Contents

CUMIPMT

CUMPRINC

DB

DDB

DISC

DOLLARDE

DOLLARFR

DURAÇÃO

EFFECT

FV

INTRATE

IPMT

ISPMT

MDURATION

NOMINAL

NPER

ODDFPRICE

ODDFYIELD

ODDLPRICE

ODDLYIELD

PDURATION

PMT

PPMT

PRICE

PRICEDISC

PRICEMAT

PV

RATE

RECEIVED

RRI

SLN

SYD

COMBIN

COMBINA

CONVERT

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

CHISQ.DIST

CHISQ.DIST.RT

CHISQ.INV

CHISQ.INV.RT

CONFIDENCE.NORM

CONFIDENCE.T

COT

COTH

COUNT

COUNTA

COUNTAX

COUNTBLANK

COUNTROWS

COUNTX

DISTINCTCOUNT

EXPON.DIST

GEOMEAN

GEOMEANX

MÁX.

MAXA

MAXX

MEDIAN

MEDIANX

MÍN.

MINA

MINX

NORM.DIST

NORM.INV

NORM.S.DIST

NORM.S.INV

PERCENTILE.EXC

PERCENTILE.INC

Visão Geral do DAX

17/03/2021 • 63 minutes to read

A DAX (Data Analysis Expressions) é uma linguagem de expressão de fórmula usada nos Analysis Services, no Power BI 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 DAX são usadas em medidas, colunas calculadas, tabelas calculadas e segurança no nível 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 de Dica de ferramenta mostra uma visualização rápida do que os resultados seriam para o total no contexto atual, mas, do contrário, os resultados não são produzidos imediatamente em qualquer lugar. A razão pela qual você não pode consultar os resultados filtrados do cálculo imediatamente é que 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, em seguida, 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 em um relatório e então coloca a coluna Product Category de uma tabela Product em Filtros, a soma do Valor de Vendas é calculado e exibido para cada categoria de produtos.

Diferentemente das colunas calculadas, 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 Campos do aplicativo cliente de relatório e dependendo se as perspectivas e funções estarão disponíveis a todos os usuários do modelo.

Para obter mais informações, consulte:
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. 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 expressão de fórmula, 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.

Segurança em nível de linha

Com segurança em nível de linha, uma fórmula DAX deve ser avaliada como uma condição booliana 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. A segurança em nível de linha não está disponível no Power Pivot no Excel.

Ao definir a configuração em nível de linha usando a fórmula DAX, você está criando 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 a segurança em nível de linha dessa função permitirem acesso a esse conjunto de linhas específico, o usuário poderá exibir os dados dessa linha.

As fórmulas de segurança em nível de linha aplicam-se às linhas especificadas, bem como às linhas relacionadas. Quando uma tabela tem várias relações, os filtros aplicam segurança para a relação ativa. As fórmulas de segurança em nível de linha serão interseccionadas por outras fórmulas definidas para tabelas relacionadas.

Para saber mais, confira:
Segurança no nível da linha (RLS) com Power BI
Funções 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 segurança em 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 segurança em nível de linha, use a caixa de diálogo Gerenciador de Função ou Gerenciar 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	DEFINIÇÃO
`= TODAY()`	Insere a data de hoje em cada linha de uma coluna calculada.
`= 3`	Insere o valor 3 em cada linha de uma coluna calculada.
`= [Column1] + [Column2]`	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 for 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, [Dias desde o Início do Trimestre Atual]/[Dias no Trimestre Atual] fornece a proporção decorrida no período atual.

Esta fórmula contém os seguintes elementos:

ELEMENTO DA FÓRMULA	DESCRIÇÃO
`Days in Current Quarter`	O nome da medida.
`=`	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

O Preenchimento Automático o ajuda a inserir uma sintaxe de fórmula válida fornecendo opções para cada elemento na fórmula.

- É possível usar a opção AutoCompletar Fórmula no meio de uma fórmula existente com funções aninhadas. O texto pouco antes do ponto de inserção é usado para exibir valores na lista suspensa, e todo o texto depois do ponto de inserção permanece inalterado.
- O Preenchimento Automático não adiciona parênteses de fechamento das funções, nem compara parênteses automaticamente. Você deve verificar se cada função está sintaticamente correta; caso contrário, não poderá salvar nem 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. Entretanto, o aninhamento pode dificultar a criação ou a solução de problemas em fórmulas. Muitas funções destinam-se ao uso somente como funções aninhadas. Essas funções retornam uma tabela, que não pode ser salva diretamente como um resultado; mas deve ser fornecida 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 estiver familiarizado com fórmulas do Excel, muitas dessas funções parecerão muito similares; porém, as fórmulas DAX são diferentes dos seguintes modos importantes:

- Uma função DAX sempre referencia uma coluna completa ou uma tabela. Para usar apenas valores específicos de uma tabela ou coluna, você pode adicionar filtros à fórmula.
- Se for necessário personalizar os cálculos linha por linha, o DAX fornecerá funções que permitem usar o valor da linha atual ou um valor relacionado como um tipo de parâmetro, para executar 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, você pode recuperar uma tabela e contar os valores distintos nele ou calcular somas dinâmicas em tabelas filtradas ou colunas.
- As funções do DAX incluem uma variedade de funções de *inteligência de tempo* . Estas funções permitem definir ou selecionar intervalos de datas e executar cálculos dinâmicos com base nestas datas ou intervalos. Por exemplo, você pode comparar somas em períodos paralelos.

Funções de data e hora

As funções de data e hora na DAX são semelhantes às funções de data e hora do Microsoft Excel. No entanto, as funções DAX baseiam-se em um tipo de dados **DateTime** a partir de 1º de março de 1900. Para saber mais, confira Funções de data e hora.

Funções de filtro

As funções de filtro em DAX retornam tipos de dados específicos, pesquisar valores em tabelas relacionadas e filtrar pelos valores relacionados. As funções de pesquisa funcionam com tabelas e relações, assim como um banco de dados. As funções de filtragem permitem manipular o contexto de dados para criar cálculos dinâmicos. Para saber mais, confira Funções de filtro.

Funções financeiras

As funções financeiras no DAX são usadas em fórmulas que fazem cálculos financeiros, como o valor líquido atual e a taxa de retorno. Essas funções são semelhantes às funções financeiras usadas no Microsoft Excel. Para saber mais, confira Funções financeiras.

Funções informativas

Uma função informativa verifica a célula ou linha fornecida como um argumento e indica se o valor corresponde ao tipo esperado. Por exemplo, a função ISERROR retorna TRUE quando o valor referenciado contém um erro. Para saber mais, confira Funções de informações.

Funções lógicas

As funções lógicas agem sobre uma expressão para retornar informações sobre os valores da expressão. Por exemplo, a função TRUE permite que você saiba se uma expressão que está sendo avaliada retorna um valor TRUE. Para saber mais, confira Funções lógicas.

Funções matemáticas e trigonométricas

As funções matemáticas em 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 saber mais, 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 saber mais, confira Outras funções.

Funções de relação

As funções de relação no DAX permitem retornar valores de outra tabela relacionada, especificar uma relação específica a ser usada em uma expressão e especificar a direção da filtragem cruzada. Para obter mais informações, confira Funções de relação.

Funções estatísticas

O DAX fornece funções estatísticas que executam agregações. Além de criar somas e médias, ou localizar os valores mínimo e máximo, na DAX também é possível filtrar uma coluna antes de agregar ou criar agregações com base em tabelas relacionadas. Para saber mais, confira Funções estatísticas.

Funções de texto

Funções de texto no DAX são muito semelhantes às suas equivalentes no Excel. Você pode retornar parte de uma cadeia de caracteres, pesquisar texto em uma cadeia de caracteres ou concatenar valores de cadeia de caracteres. A DAX também fornece funções para controlar os formatos de datas, horas e números. Para saber mais, confira Funções de texto.

Funções de inteligência de dados temporais

As funções de inteligência de tempo fornecidas na DAX permitem 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. Para saber mais, confira Funções de inteligência de tempo (DAX).

Funções de manipulação de tabelas

Essas funções retornam uma tabela ou manipulam tabelas existentes. Por exemplo, usando AddColumns, você pode adicionar colunas calculadas a uma tabela especificada ou pode retornar uma tabela de resumo em um conjunto de grupos com a função SUMMARIZECOLUMNS. Para saber mais, confira Funções de manipulação de tabela.

Variáveis

Você pode criar variáveis dentro de uma expressão usando VAR. Tecnicamente, o VAR não é uma função; é uma palavra-chave 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

É possível importar para um modelo de diversas fontes de dados diferentes que podem dar suporte a diferentes tipos de dados. Quando você importa os dados em um modelo, os dados são convertidos em um dos tipos de dados de modelo de tabela. Quando os dados de modelo são usados em um cálculo, os dados 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 na fórmula determinarão o tipo de dados de valor retornado automaticamente.

O DAX dá suporte aos seguintes tipos de dados:

TIPO DE DADOS EM MODELO	TIPOS DE DADOS EM DAX	DESCRIÇÃO
Número Inteiro	Um valor inteiro de 64 bits (oito bytes) [1, 2]	Números sem 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 está limitado a 17 dígitos decimais.
Booliano	Booliano	Um valor True ou False.
Texto	Cadeia de caracteres	Uma cadeia de caracteres de dados de caractere Unicode. Podem ser cadeias de caracteres, números ou datas representados em um formato de texto.
Data	Data/hora	Datas e horas em uma representação de data-hora aceita. As datas válidas são todas as datas depois de 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.

TIPO DE DADOS EM MODELO	TIPOS DE DADOS EM DAX	DESCRIÇÃO
N/D	Em branco	Um espaço em branco é um tipo de dados no DAX que representa e substitui nulos SQL. É possível 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 outra tabela que contém apenas as linhas que atendam às condições do filtro. Ao combinar funções de tabela com funções de agregação, você pode executar cálculos complexos em conjuntos de dados definidos de forma dinâmica.

Embora os tipos de dados em geral sejam definidos automaticamente, é importante entendê-los e saber como se aplicam, em particular, a fórmulas DAX. Erros em fórmulas ou resultados inesperados, por exemplo, são causados frequentemente pelo uso de 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 de 12. A fórmula `= "1" + "2"`, no entanto, retorna um resultado de inteiro de 3.

Contexto

Contexto é um conceito importante que deve ser entendido ao criar fórmulas DAX. O contexto é o que permite executar análise dinâmica, como os resultados de uma fórmula são alterados para refletir a seleção atual de linha ou célula, além de qualquer dado relacionado. Entender o que é contexto e seu uso eficiente é vital para compilar análises dinâmicas de alto desempenho e para solucionar problemas em fórmulas.

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

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

Há tipos diferentes 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 dessa outra tabela que estão relacionados à linha atual.

Por exemplo, suponha que você crie uma coluna calculada, `= [Freight] + [Tax]`, que soma os valores de duas colunas, Freight e Tax, da mesma tabela. Esta fórmula automaticamente obtém somente os valores da linha atual nas colunas especificadas.

Contexto de linha também segue os relacionamentos que foram definidos entre tabelas, incluindo relacionamentos definidos dentro de uma coluna calculada usando fórmulas DAX, para determinar quais linhas nas 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 para a qual o pedido foi enviado. O valor do imposto é determinado com o uso do valor para a região na tabela atual, pesquisando-se a região na tabela relacionada e obtendo-se o valor do imposto para essa região na tabela relacionada.

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

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

Contexto de várias linhas

A DAX inclui várias funções que iteram cálculos em uma tabela. Essas funções podem ter várias linhas atuais, cada uma com seu próprio contexto de linha. Em essência, estas funções permitem criar fórmulas que executam operações recursivamente em um loop interno e exterior.

Por exemplo, vamos supor que o modelo contenha uma tabela **Products** e uma tabela **Sales** . Talvez os usuários queiram percorrer toda a tabela de vendas, que contém várias transações que envolvem vários produtos, e encontrar a maior quantidade solicitada de cada produto em qualquer transação.

Com a DAX você pode criar uma única fórmula que retorna o valor correto e os resultados são atualizados automaticamente a qualquer momento em que um usuário adicionar dados às tabelas.

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

Para obter um exemplo detalhado desta fórmula, confira EARLIER.

Em resumo, a função EARLIER armazena o contexto de linha da operação que precedeu a operação atual. O tempo todo, 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. A DAX alimenta automaticamente valores entre os dois loops para que você possa criar agregações complexas.

Contexto de consulta

Contexto de consulta se refere ao subconjunto de dados recuperados implicitamente para uma fórmula. Por exemplo, quando um usuário coloca uma medida ou campo em um relatório, o mecanismo examina cabeçalhos de linha e coluna, as segmentações e os filtros de relatórios para determinar o contexto. As consultas necessárias então são executadas em relação aos dados de modelo para obter o subconjunto correto de dados, fazer os cálculos definidos pela fórmula e preencher os valores no relatório.

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 um relatório, o contexto é alterado por filtragem, adição ou remoção de campos e uso de segmentações. Para cada alteração, o contexto de consulta no qual a medida é avaliada. 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). 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 de uma fórmula. O contexto de filtro é aplicado sobre outros contextos, como o contexto de linha ou o contexto de consulta.

Em modelos de tabela, há muitos modos de criar 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, para filtrar tabelas que são usadas como entradas ou para obter contexto dinamicamente para os valores que são usados em cálculos. Você também pode desmarcar completa ou seletivamente os filtros em colunas específicas. Isto é muito útil ao criar fórmulas que calculam totais principais.

Para saber mais sobre como criar filtros dentro de fórmulas, confira a Função FILTER (DAX).
Para obter um exemplo de como os filtros são limpos para criar totais gerais, consulte 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 do DAX, a fórmula é testada primeiro para sintaxe válida e para ter certeza de que os nomes das colunas e tabelas incluídos na fórmula podem ser localizados no contexto atual. Se qualquer coluna ou tabela especificada pela fórmula não puder ser localizada, um erro será retornado.

O contexto durante a validação (e operações de recálculo) é determinado conforme descrito nas seções anteriores, usando as tabelas disponíveis no modelo, qualquer relacionamento entre tabela e qualquer filtro que tenha sido aplicado.

Por exemplo, se você tiver importado alguns dados em uma nova tabela e não tiver relacionado a nenhuma outra tabela (e não tiver aplicado filtros), o *contexto atual* será o conjunto inteiro de colunas da 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 tem segmentação de dados e talvez algum filtros de relatório, o contexto para a fórmula será o subconjunto de dados em cada célula do relatório.

O contexto é um conceito avançado que também pode dificultar a solução de erros de fórmulas. Nós recomendamos que você comece com fórmulas simples e relacionamentos para verificar como o contexto funciona. A seção a seguir fornece alguns exemplos de como as fórmulas usam tipos diferentes 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.

Trabalhando 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, e não com células individuais, referências de intervalos ou matrizes.
- As fórmulas podem usar relações para obter valores de tabelas relacionadas. Os valores recuperados sempre são relacionados ao valor da linha atual.

- Não pode haver dados irregulares, como ocorre em uma planilha do Excel. Cada linha de uma tabela deve conter o mesmo número de colunas. No entanto, pode haver valores vazios em algumas colunas. As tabelas de dados do Excel e as tabelas de dados do modelo tabular não são intercambiáveis.
- Como um tipo de dados é definido para cada coluna, cada valor nessa coluna deve ser do mesmo tipo.

Referenciando tabelas e colunas em fórmulas

É possível referenciar qualquer tabela e coluna usando seu nome. Por exemplo, a seguinte fórmula 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 do modelo primeiro verifica a sintaxe geral e, em seguida, os nomes das colunas e tabelas que você fornece em relação a possíveis colunas e tabelas no contexto atual. Se o nome for ambíguo ou se a coluna ou tabela não puder ser encontrada, você obterá um erro na fórmula (uma cadeia de caracteres #ERROR em lugar de um valor de dados em células nas quais o erro ocorre). Para saber mais 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 obtém a habilidade de valores relacionados em outras tabelas a serem usados em cálculos. Por exemplo, você pode usar uma coluna calculada para determinar todos os registros de envio relacionados ao revendedor atual e somá-los para cada um. Em muitos casos, porém, uma relação talvez não seja 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 uma relação existente entre as tabelas, ou entre várias tabelas, para localizar as colunas que você referenciou e retornar resultados que tenham sentido. Outras funções tentarão identificar a relação; no entanto, tendo em vista melhores resultados, você deverá criar sempre uma relação onde for 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 ocorrer, você deve estar ciente de que valores em branco ou não correspondentes pode afetar os resultados das fórmulas.

- Ao vincular tabelas em seu modelo usando relações, você amplia o escopo, ou *contexto*, no qual as fórmulas são avaliadas. As alterações no contexto resultantes da adição de novas tabelas, novas relações, ou de alterações na relação ativa podem fazer seus resultados serem alterados de maneira imprevista. Para saber mais, confira Contexto neste artigo.

Processar e atualizar

Processo e *recálculo* são duas operações separadas, mas relacionadas. Você deve compreender totalmente esses conceitos ao criar um modelo que contenha fórmulas complexas, grandes volumes 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 das fórmulas para refletir todas as alterações feitas nas próprias fórmulas e nos dados subjacentes. O recálculo pode afetar o desempenho das seguintes formas:

- O valores em uma coluna calculada são computados e armazenados no modelo. Para atualizar os valores na coluna calculada, você deverá processar o modelo usando um dos três comandos de processamento: Processar Completo, Processar Dados ou Processar Recálculo. O resultado da fórmula sempre deve ser recalculado para a coluna inteira, todas as vezes 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 segurança em nível 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.

Atualizações

O DAX está constantemente sendo aprimorado. Funções novas e atualizadas são lançadas com a próxima atualização disponível, que geralmente é mensal. Os serviços são atualizados primeiro, seguidos por aplicativos instalados, como Power BI Desktop, Excel, SSMS (SQL Server Management Studio) e a extensão de projeto Analysis Services para Visual Studio (SSDT). SQL Server Analysis Services é atualizado com a próxima atualização cumulativa. Novas funções primeiro são anunciadas e descritas na referência da função DAX que coincide com atualizações do Power BI Desktop.

Nem todas as funções têm suporte em versões anteriores do SQL Server Analysis Services e do Excel.

Solução de problemas

Se você receber um erro ao definir uma fórmula, talvez a fórmula contenha um *erro sintático*, um *erro semântico* ou *erro de cálculo*.

Erros sintáticos são os mais fáceis de resolver. Em geral, eles envolvem a falta de um parêntese ou de uma vírgula.

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 um dos seguintes problemas:

- A fórmula se refere 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 referencia uma coluna diferente que tem o erro e, por isso, os valores são inválidos.
- A fórmula refere-se a uma coluna que não foi processada, significando que tem metadados mas nenhum dado real para usar para cálculos.

Nos quatro primeiros casos, o DAX sinaliza a coluna inteira que contém a fórmula inválida. No último caso, a DAX torna a coluna indisponível para indicar que ela 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 Visual Studio com a extensão projetos do Analysis Services (VSIX) é usado para criar projetos de modelo de Analysis Services. O designer de modelo de tabela, instalado com a extensão de projetos, inclui um editor DAX.

O 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.

Tabular Editor

O Tabular Editor é uma ferramenta de open-source que fornece uma exibição intuitiva e hierárquica de cada objeto em metadados de modelo de tabela. O Tabular Editor inclui um editor DAX com realce de sintaxe, que fornece uma maneira fácil de editar medidas, colunas calculadas e expressões de tabelas calculadas.

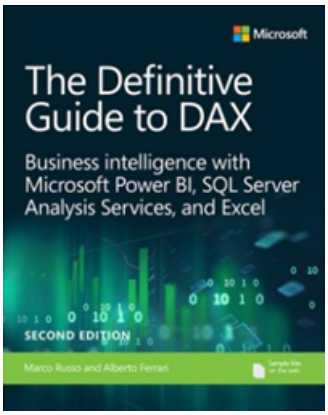
Recursos de aprendizagem

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. Estes são alguns recursos adicionais:

Vídeos

Caminho Usar o DAX no Power BI Desktop no Microsoft Learn.

The Definitive Guide to DAX (O Guia Definitivo para o DAX) de Alberto Ferrari e Marco Russo (Microsoft Press). Agora, na segunda edição, este guia extensivo fornece noções básicas para técnicas inovadoras de alto desempenho para modeladores de dados e profissionais de BI iniciantes.



Comunidade

O DAX tem uma comunidade vibrante sempre disposta a compartilhar a própria experiência. A Comunidade do Microsoft Power BI tem um fórum de discussão especial apenas para DAX, Comandos e Dicas do DAX.

Vídeos

17/03/2021 • 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

17/03/2021 • 5 minutes to read

A referência de função do DAX fornece informações detalhadas, incluindo sintaxe, parâmetros, valores retornados e exemplos para cada uma das mais de 250 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.

Nesta seção

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

Funções de data e hora – 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 filtro – 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 financeiras – essas funções são usadas em fórmulas que fazem cálculos financeiros, como o valor líquido atual e a taxa de retorno.

Funções de informações – 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 – 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 – 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 – 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 – 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 de relação – essas funções são para gerenciar e utilizar relações entre tabelas. Por exemplo, você pode definir uma relação específica a ser usada em um cálculo.

Funções estatísticas – 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 manipulação de tabela – essas funções retornam uma tabela ou manipulam tabelas existentes.

Funções de texto – 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.

Funções de inteligência de dados temporais – 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.

Consulte também

Referência de Sintaxe do DAX
Referência do operador DAX
Convenções de nomenclatura de parâmetro DAX

Novas funções do DAX

22/04/2021 • 3 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.

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 para 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
IF.EAGER	Março de 2021
ACCRINT	Julho de 2020
ACCRINTM	Julho de 2020
AMORDEGRC	Julho de 2020
AMORLINC	Julho de 2020
COUPDAYBS	Julho de 2020
COUPDAYS	Julho de 2020
COUPDAYSNC	Julho de 2020
COUPNCD	Julho de 2020
COUPNUM	Julho de 2020
COUPPCD	Julho de 2020
CUMIPMT	Julho de 2020
CUMPRINC	Julho de 2020
DB	Julho de 2020

FUNÇÃO	MÊS
DDB	Julho de 2020
DISC	Julho de 2020
DOLLARDE	Julho de 2020
DOLLARFR	Julho de 2020
DURATION	Julho de 2020
EFFECT	Julho de 2020
FV	Julho de 2020
INTRATE	Julho de 2020
IPMT	Julho de 2020
ISPMT	Julho de 2020
MDURATION	Julho de 2020
NOMINAL	Julho de 2020
NPER	Julho de 2020
ODDFPRICE	Julho de 2020
ODDFYIELD	Julho de 2020
ODDLPRICE	Julho de 2020
ODDLYIELD	Julho de 2020
PDURATION	Julho de 2020
PMT	Julho de 2020
PPMT	Julho de 2020
PRICE	Julho de 2020
PRICEDISC	Julho de 2020
PRICEMAT	Julho de 2020
PV	Julho de 2020
RATE	Julho de 2020

FUNÇÃO	MÊS
RECEIVED	Julho de 2020
RRI	Julho de 2020
SLN	Julho de 2020
SYD	Julho de 2020
TBILLEQ	Julho de 2020
TBILLPRICE	Julho de 2020
TBILLYIELD	Julho de 2020
VDB	Julho de 2020
YIELD	Julho de 2020
YIELDDISC	Julho de 2020
YIELDMAT	Julho de 2020

Funções atualizadas

FUNÇÃO	MÊS	DESCRIÇÃO
CROSSFILTER	Abril de 2021	Opções adicionais para o parâmetro Direction.
CALCULATE	Março de 2021	Suporte para o operador OR (||) quando há vários filtros.

Funções de data e hora

17/03/2021 • 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 apenas uma coluna chamada "Date" que contém um conjunto contíguo de datas.
CALENDARAUTO	Retorna uma tabela com apenas uma coluna chamada "Date" 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.
UTCNOW	Retorna a data e a hora UTC atuais
UTCTODAY	Retorna a data atual no UTC.
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

17/03/2021 • 2 minutes to read

Retorna uma tabela com apenas uma coluna chamada "Date" 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 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

28/04/2021 • 2 minutes to read

Retorna uma tabela com apenas uma coluna chamada "Date" 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 apenas uma coluna chamada "Date" 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas 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 31 de março de 2012.

DATE

17/03/2021 • 7 minutes to read

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

Sintaxe

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

Parâmetros

TERMO	DEFINIÇÃO
ano	Um número que representa o ano. 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. Há suporte para datas que começam em 1º de março de 1900. Se você inserir um número que tenha casas decimais, o número será arredondado. Para valores maiores que 9999 ou menores que zero (valores negativos), a função retorna um erro **#VALUE!** . 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.
mês	Um número que representa o mês ou um cálculo de acordo com as seguintes regras: Não há suporte para inteiros negativos. Os valores válidos são de 1 a 12. 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. 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 exemplos abaixo.

TERMO	DEFINIÇÃO
dia	Um número que representa o dia ou um cálculo de acordo com as seguintes regras: Não há suporte para inteiros negativos. Os valores válidos são de 1 a 31. Se **day** for um número de 1 até o último dia do mês especificado, ele representará um dia do mês. 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. Se **day** contiver uma parte decimal, ele será arredondado para o valor inteiro mais próximo.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo: Retornando uma data simples

Descrição

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

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

Exemplo: Anos antes de 1899

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 fórmula a seguir retorna 2 de janeiro de 1908: (1900+08).

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

Exemplo: Anos depois de 1899

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

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

Exemplo: Trabalhando com meses

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:

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

Exemplo: Trabalhando com dias

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:

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

Confira também

Funções de data e hora
função DAY
Função TODAY

DATEDIFF

17/03/2021 • 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.
intervalo	O intervalo a ser usado ao comparar datas. O valor pode ser um dos seguintes: - SECOND - MINUTE - HOUR - DAY - WEEK - MONTH - QUARTER - YEAR

Retornar valor

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

17/03/2021 • 2 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

DAY

17/03/2021 • 3 minutes to read

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
Função TODAY
Função DATE

EDATE

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

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

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

Confira também

Função EOMONTH
Funções de data e hora

EOMONTH

17/03/2021 • 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.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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)
```

Confira também

Função EDATE
Funções de data e hora

HOUR

17/03/2021 • 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 1

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

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

Exemplo 2

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

MINUTE

17/03/2021 • 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 1

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

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

Exemplo 2

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

MÊS

17/03/2021 • 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 1

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 2

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
Função HOUR
Função MINUTE
função YEAR
função SECOND

NOW

17/03/2021 • 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.
- No Serviço do Power BI, o resultado da função NOW está sempre no fuso horário UTC.
- 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
```

Confira também

Função UTCNOW
Função TODAY

QUARTER

17/03/2021 • 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.

Retornar valor

Um número inteiro de 1 a 4.

Comentários

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

Exemplo 1

A seguinte consulta DAX:

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

Retorna:

[VALOR]
1
4

Exemplo 2

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

Confira também

YEAR
MONTH
DAY

SECOND

17/03/2021 • 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 1

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

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

Exemplo 2

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
HOUR
MINUTE
YEAR

TIME

17/03/2021 • 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 aprimorar a legibilidade dos valores de hora retornados por essa função, 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo 1

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

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

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

Exemplo 2

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

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

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

Exemplo 3

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

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

Confira também

DATE
Funções de data e hora

TIMEVALUE

17/03/2021 • 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")
```

Confira também

Funções de data e hora

TODAY

17/03/2021 • 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.
- 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
NOW

UTCNOW

17/03/2021 • 2 minutes to read

Retorna a data e a hora atuais no formato UTC.

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
Função UTCTODAY

UTCTODAY

17/03/2021 • 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
Função UTCNOW

WEEKDAY

17/03/2021 • 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	Uma data no formato **datetime**. As datas devem ser inseridas com a função DATE, usando expressões que resultam em uma data ou como o resultado de outras fórmulas.
return_type	Um número que determina o valor retornado: Tipo de retorno: **1**, a semana começa no domingo (1) e termina no sábado (7). numerado de 1 a 7. Tipo de retorno: **2**, a semana começa na segunda-feira (1) e termina no domingo (7). Tipo de retorno: **3**, a semana começa na segunda-feira (0) e termina no domingo (6). numerado de 1 a 7.

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
Função WEEKNUM
função YEARFRAC

WEEKNUM

17/03/2021 • 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	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. Tipo de retorno: **1**, a semana começa no domingo. Os dias da semana são numerados de 1 a 7. Tipo de retorno: **2**, a semana começa na segunda-feira. Os dias da semana são numerados de 1 a 7.

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 1

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

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

Exemplo 2

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
função YEARFRAC
função WEEKDAY

YEAR

17/03/2021 • 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
Função HOUR
Função MINUTE
função YEAR
função SECOND

YEARFRAC

17/03/2021 • 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 1

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 2

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
Função WEEKNUM
função YEARFRAC
função WEEKDAY

Funções de filtro

17/03/2021 • 2 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
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 de filtro modificado.
CALCULATETABLE	Avalia uma expressão de tabela em um contexto de filtro modificado.
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.
FILTER	Retorna uma tabela que representa um subconjunto de outra tabela ou expressão.
KEEPFILTERS	Modifica como os filtros são aplicados durante a avaliação de uma função CALCULATE ou CALCULATETABLE.
LOOKUPVALUE	Retorna o valor da linha que atende a todos os critérios especificados pelos critérios de pesquisa. A função pode aplicar um ou mais critérios de pesquisa.
REMOVEFILTERS	Limpa filtros das tabelas ou colunas especificadas.

FUNÇÃO	DESCRIÇÃO
SELECTEDVALUE	Retorna o valor quando o contexto para columnName foi filtrado para apenas um valor distinto. Caso contrário, retorna alternateResult.

ALL

17/03/2021 • 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.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo 1

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.

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%

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 2

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** de uma 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**.

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

```
= 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 3

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**.

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

```
= 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
Função ALL
Função ALLEXCEPT
Função FILTER

ALLCROSSFILTERED

17/03/2021 • 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.

Retornar valor

N/D Consulte Observações.

Comentários

- ALLCROSSFILTERED só pode ser usado para limpar filtros, mas não para retornar uma tabela.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

DIMSALESREASON[SALESREASONNAME]	[TOTALQUANTITY1]	[TOTALQUANTITY2]
Na promoção	60398	7390
Outro	60398	3653
Preço	60398	47733
Qualidade	60398	1551
Revisão	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

17/03/2021 • 5 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.

FUNÇÃO E USO	DESCRIÇÃO
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.
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.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
Função ALL
Função FILTER

ALLNOBLANKROW

17/03/2021 • 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.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
Função ALL
Função FILTER

ALLSELECTED

17/03/2021 • 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.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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		80450596,9823	877006,7987	38786,018	
Acessórios	2001		80450596,9823	877006,7987	38786,018	
Acessórios	2002	625,7933	80450596,9823	877006,7987	38786,018	91495,3104
Acessórios	2003	26037,3132	80450596,9823	877006,7987	38786,018	572927,0136
Acessórios	2004	12122,9115	80450596,9823	877006,7987	38786,018	212584,4747
Acessórios	2005		80450596,9823	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		80450596,9823	877006,7987	38786,018	
Bikes	2000		80450596,9823	877006,7987	689287,7939	
Bikes	2001		80450596,9823	877006,7987	689287,7939	
Bikes	2002	73778,938	80450596,9823	877006,7987	689287,7939	91495,3104
Bikes	2003	439771,4136	80450596,9823	877006,7987	689287,7939	572927,0136
Bikes	2004	175737,4423	80450596,9823	877006,7987	689287,7939	212584,4747
Bikes	2005		80450596,9823	877006,7987	689287,7939	
Bikes	2006		80450596,9823	877006,7987	689287,7939	
Clothing	2000		80450596,9823	877006,7987	95090,7757	
Clothing	2001		80450596,9823	877006,7987	95090,7757	
Clothing	2002	12132,4334	80450596,9823	877006,7987	95090,7757	91495,3104
Clothing	2003	58234,2214	80450596,9823	877006,7987	95090,7757	572927,0136
Clothing	2004	24724,1209	80450596,9823	877006,7987	95090,7757	212584,4747
Clothing	2005		80450596,9823	877006,7987	95090,7757	
Clothing	2006		80450596,9823	877006,7987	95090,7757	
Componentes	2000		80450596,9823	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		80450596,9823	877006,7987	53842,2111	
Componentes	2002	4958,1457	80450596,9823	877006,7987	53842,2111	91495,3104
Componentes	2003	48884,0654	80450596,9823	877006,7987	53842,2111	572927,0136
Componentes	2004		80450596,9823	877006,7987	53842,2111	212584,4747
Componentes	2005		80450596,9823	877006,7987	53842,2111	
Componentes	2006		80450596,9823	877006,7987	53842,2111	

As colunas no relatório são:

Reseller Sales Amount
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

24/03/2021 • 9 minutes to read

Avalia uma expressão em um contexto de filtro modificado.

> **NOTE**
> Também há a função CALCULATETABLE. Ela executa exatamente a mesma funcionalidade, exceto que modifica o contexto de filtro aplicado a uma expressão que retorna um *objeto de tabela*.

Sintaxe

```
CALCULATE(<expression>[, <filter1> [, <filter2> [, …]]])
```

Parâmetros

TERMO	DEFINIÇÃO
expressão	Expressão a ser avaliada.
filter1, filter2,...	(Opcional) Expressões booleanas ou expressões de tabela que definem filtros ou funções de modificador de filtro.

A expressão usada como o primeiro parâmetro é essencialmente a mesma que uma medida.

Os filtros podem ser:

- Expressões de filtro booliano
- Expressões de filtro de tabela
- Funções de modificação de filtro

Quando há vários filtros, eles podem ser avaliados usando o operador lógico AND (&&), o que significa que todas as condições devem ser verdadeiras ou pelo operador lógico OR (| |), o que significa que qualquer condição pode ser verdadeira.

Expressões de filtro booliano

Um filtro de expressão booliana é uma expressão que é avaliada como TRUE ou FALSE. Há várias regras que elas devem cumprir:

- Elas podem referenciar colunas de uma única tabela.
- Elas não podem referenciar medidas.
- Elas não podem usar uma função CALCULATE aninhada.
- Elas não podem usar funções que examinam ou retornam uma tabela, incluindo funções de agregação.

Expressão de filtro de tabela

Um filtro de expressão de tabela aplica um objeto de tabela como um filtro. Pode ser uma referência a uma tabela de modelo, mas é mais provável que seja uma função que retorne um objeto de tabela. Você pode usar a função FILTER para aplicar condições de filtro complexas, incluindo aquelas que não podem ser definidas por uma expressão de filtro booliana.

Funções de modificador de filtro

As funções de modificação de filtro permitem que você faça mais do que simplesmente adicionar filtros. Elas fornecem controle adicional ao modificar o contexto do filtro.

FUNÇÃO	FINALIDADE
REMOVEFILTERS	Remover todos os filtros ou os filtros de uma ou mais colunas de uma tabela ou de todas as colunas de uma tabela.
ALL [1], ALLEXCEPT, ALLNOBLANKROW	Remover filtros de uma ou mais colunas ou de todas as colunas de uma tabela.
KEEPFILTERS	Adicionar filtro sem remover filtros existentes nas mesmas colunas.
USERELATIONSHIP	Envolver uma relação inativa entre as colunas relacionadas, caso em que a relação ativa se tornará inativa automaticamente.
CROSSFILTER	Modificar a direção do filtro (de ambos para único ou de um para ambos) ou desabilitar uma relação.

[1] A função ALL e suas variantes se comportam como modificadores de filtro e como funções que retornam objetos de tabela. Se houver suporte para a função REMOVEFILTERS na sua ferramenta, será melhor usá-la para remover filtros.

Retornar valor

O valor que é o resultado da expressão.

Comentários

- Quando expressões de filtro são fornecidas, a função CALCULATE modifica o contexto do filtro para avaliar a expressão. Para cada expressão de filtro, há dois resultados padrão possíveis quando a expressão de filtro não é encapsulada na função KEEPFILTERS:
  - Se as colunas (ou tabelas) não estiverem no contexto de filtro, novos filtros serão adicionados ao contexto de filtro para avaliar a expressão.
  - Se as colunas (ou tabelas) já estiverem no contexto de filtro, os filtros existentes serão substituídos pelos novos filtros para avaliar a expressão CALCULATE.
- A função CALCULATE usada *sem filtros* atinge um requisito específico. Ela faz a transição do contexto de linha para o contexto de filtro. É necessário quando uma expressão (não uma medida de modelo) que resume os dados de modelo precisa ser avaliada no contexto da linha. Esse cenário pode ocorrer em uma fórmula de coluna calculada ou quando uma expressão em uma função de iterador é avaliada. Observe que, quando uma medida de modelo é usada no contexto de linha, a transição de contexto é automática.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplos

A definição de medida da tabela **Vendas** a seguir produz um resultado de receita, mas apenas para produtos que têm a cor azul.

Os exemplos deste artigo podem ser adicionados ao modelo de exemplo do Power BI Desktop. Para obter o

modelo, confira Modelo de exemplo DAX.

```
Blue Revenue =
CALCULATE(
    SUM(Sales[Sales Amount]),
    'Product'[Color] = "Blue"
)
```

CATEGORIA	VALOR DAS VENDAS	RECEITA AZUL
Acessórios	US$ 1.272.057,89	US$ 165.406,62
Bikes	US$ 94.620.526,21	US$ 8.374.313,88
Clothing	US$ 2.117.613,45	US$ 259.488,37
Componentes	$11,799,076.66	US$ 803.642,10
Total	**US$ 109.809.274,20**	**US$ 9.602.850,97**

A função CALCULATE avalia a soma da tabela **Vendas**, coluna **Valor de vendas**, em um contexto de filtro modificado. Um novo filtro é adicionado à tabela **Product**, coluna **Color**, ou o filtro substitui qualquer filtro que já esteja aplicado à coluna.

A definição de medida da tabela **Vendas** a seguir produz uma taxa de vendas sobre vendas para todos os canais de vendas.

CANAL	VALOR DAS VENDAS	% DE RECEITA TOTAL DO CANAL
Internet	$29,358,677.22	26,74%
Reseller	$80,450,596.98	73,26%
Total	**US$ 109.809.274,20**	**100,00%**

```
Revenue % Total Channel =
DIVIDE(
    SUM(Sales[Sales Amount]),
    CALCULATE(
        SUM(Sales[Sales Amount]),
        REMOVEFILTERS('Sales Order'[Channel])
    )
)
```

A função DIVIDE divide uma expressão que soma a tabela **Vendas**, coluna **Valor das Vendas** (no contexto de filtro), pela mesma expressão em um contexto de filtro modificado. É a função CALCULATE que modifica o contexto de filtro usando a função REMOVEFILTERS, que é uma função de modificador de filtro. Ela remove filtros da tabela **Sales Order**, coluna **Channel**.

A definição de coluna calculada da tabela **Customer** a seguir classifica os clientes em uma classe de fidelidade. É um cenário muito simples: Quando a receita produzida pelo cliente é menor que US$ 2.500, é classificada como *baixa*; caso contrário, é *alta*.

```
Customer Segment =
IF(
    CALCULATE(SUM(Sales[Sales Amount]), ALLEXCEPT(Customer, Customer[CustomerKey])) < 2500,
    "Low",
    "High"
)
```

Neste exemplo, o contexto de linha é convertido no contexto de filtro. Isso é conhecido como *transição de contexto*. A função ALLEXCEPT remove filtros de todas as colunas da tabela **Customer**, exceto a coluna **CustomerKey**.

Confira também

Contexto de filtro
Contexto de linha
função CALCULATETABLE
Funções de filtro

CALCULATETABLE

17/03/2021 • 6 minutes to read

Avalia uma expressão de tabela em um contexto de filtro modificado.

> **NOTE**
> Também há a função CALCULATE. Ela executa exatamente a mesma funcionalidade, exceto que modifica o contexto de filtro aplicado a uma expressão que retorna um *valor escalar*.

Sintaxe

```
CALCULATETABLE(<expression>[, <filter1> [, <filter2> [, …]]])
```

Parâmetros

TERMO	DEFINIÇÃO
expressão	A expressão de tabela a ser avaliada.
filter1, filter2,...	(Opcional) Expressões booleanas ou expressões de tabela que definem filtros ou funções de modificador de filtro.

A expressão usada como o primeiro parâmetro deve ser uma tabela de modelo ou uma função que retorne uma tabela.

Os filtros podem ser:

- Expressões de filtro booliano
- Expressões de filtro de tabela
- Funções de modificação de filtro

Quando há vários filtros, eles são avaliados usando o operador lógico AND. Isso significa que todas as condições devem ser verdadeiras ao mesmo tempo.

Expressões de filtro booliano

Um filtro de expressão booliana é uma expressão que é avaliada como TRUE ou FALSE. Há várias regras que elas devem cumprir:

- Elas podem referenciar somente uma coluna.
- Elas não podem referenciar medidas.
- Elas não podem usar uma função CALCULATE aninhada.
- Elas não podem usar funções que examinam ou retornam uma tabela, incluindo funções de agregação.

Expressão de filtro de tabela

Um filtro de expressão de tabela aplica um objeto de tabela como um filtro. Pode ser uma referência a uma tabela de modelo, mas é mais provável que seja uma função que retorne um objeto de tabela. Você pode usar a função FILTER para aplicar condições de filtro complexas, incluindo aquelas que não podem ser definidas por uma expressão de filtro booliana.

Funções de modificador de filtro

As funções de modificação de filtro permitem que você faça mais do que simplesmente adicionar filtros. Elas fornecem controle adicional ao modificar o contexto do filtro.

FUNÇÃO	FINALIDADE
REMOVEFILTERS	Remover todos os filtros ou os filtros de uma ou mais colunas de uma tabela ou de todas as colunas de uma tabela.
ALL [1], ALLEXCEPT, ALLNOBLANKROW	Remover filtros de uma ou mais colunas ou de todas as colunas de uma tabela.
KEEPFILTERS	Adicionar filtro sem remover filtros existentes nas mesmas colunas.
USERELATIONSHIP	Envolver uma relação inativa entre as colunas relacionadas, caso em que a relação ativa se tornará inativa automaticamente.
CROSSFILTER	Modificar a direção do filtro (de ambos para único ou de um para ambos) ou desabilitar uma relação.

[1] A função ALL e suas variantes se comportam como modificadores de filtro e como funções que retornam objetos de tabela. Se houver suporte para a função REMOVEFILTERS na sua ferramenta, será melhor usá-la para remover filtros.

Retornar valor

Uma tabela de valores.

Comentários

- Quando expressões de filtro são fornecidas, a função CALCULATETABLE modifica o contexto do filtro para avaliar a expressão. Para cada expressão de filtro, há dois resultados padrão possíveis quando a expressão de filtro não é encapsulada na função KEEPFILTERS:
  - Se as colunas (ou tabelas) não estiverem no contexto de filtro, novos filtros serão adicionados ao contexto de filtro para avaliar a expressão.
  - Se as colunas (ou tabelas) já estiverem no contexto de filtro, os filtros existentes serão substituídos pelos novos filtros para avaliar a expressão CALCULATETABLE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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 na seguinte tabela:

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
Grande Total	US$ 26.054.827,45	US$ 5.681.440,58	4.59

Confira também

- Contexto de filtro
- Função CALCULATE (DAX)
- Funções de filtro (DAX)

EARLIER

17/03/2021 • 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	(Opcional) Um número positivo para a etapa de avaliação externa. O próximo nível de avaliação externa será representado por 1. Dois níveis externos além serão representados por 2 e assim por diante. Quando omitido, assumirá o valor 1 como padrão.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança

em nível de linha) ou colunas calculadas.

> **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

PRODUCTSUBCATEGORYKEY	ENGLISHPRODUCTSUBCATEGORYNAME	TOTALSUBCATEGORYSALES	SUBCATEGORYRANKING
32	Conjuntos para hidratação	US$ 96.893,78	20
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.

Confira também

função EARLIEST
Funções de filtro

EARLIEST

17/03/2021 • 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.

Retornar valor

Uma coluna com os filtros removidos.

Comentários

- A função EARLIEST é semelhante à EARLIER, mas permite que você especifique um nível adicional de recursão.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

Os dados de exemplo atuais não dão suporte a esse cenário.

```
= EARLIEST(<column>)
```

Confira também

função EARLIER
Funções de filtro

FILTER

17/03/2021 • 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, `[Amount] > 0` ou `[Region] = "France"`

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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 ao território de vendas dos 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** de um relatório ou uma Tabela Dinâmica.

Tabela 1. Comparando as vendas totais para os EUA versus todas as outras regiões

RÓTULOS DE LINHA	INTERNET SALES	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	
Grande Total	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

RÓTULOS DE LINHA	ACESSÓRIOS	BIKES	CLOTHING	GRANDE TOTAL
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
Grande Total	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])
```

Confira também

Funções de filtro
Função ALL
Função ALLEXCEPT

KEEPFILTERS

17/03/2021 • 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.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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.

Confira também

Funções de filtro
função CALCULATE
função CALCULATETABLE

LOOKUPVALUE

17/03/2021 • 3 minutes to read

Retorna o valor da linha que atende a todos os critérios especificados por um ou mais critérios de pesquisa.

Sintaxe

```
LOOKUPVALUE(
    <result_columnName>,
    <search_columnName>,
    <search_value>
    [, <search2_columnName>, <search2_value>]…
    [, <alternateResult>]
)
```

Parâmetros

TERMO	DEFINIÇÃO
result_columnName	O nome de uma coluna existente que contém o valor que você deseja retornar. Não pode ser uma expressão.
search_columnName	O nome de uma coluna existente. Ele pode estar na mesma tabela que result_columnName ou em uma tabela relacionada. Não pode ser uma expressão.
search_value	O valor a ser pesquisado em search_columnName.
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 for fornecido, a função retornará BLANK quando result_columnName for filtrado para obter o valor zero ou retornará um erro quando houve 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 exata.

Caso não haja uma correspondência que satisfaça todos os valores de pesquisa, será retornado BLANK ou **alternateResult** (se fornecido). Isso significa que a função não retornará um valor de pesquisa caso somente alguns dos critérios sejam atendidos.

Caso várias linhas correspondam aos valores de pesquisa e, em todos os casos, os valores de **result_column** forem idênticos, esse valor será retornado. No entanto, caso **result_column** retorne valores diferentes, um erro ou **alternateResult** (se fornecido) será retornado.

Comentários

- Se houver uma relação entre as tabelas de resultado e de pesquisa, na maioria dos casos, o uso da função RELATED em vez de LOOKUPVALUE será mais eficiente e fornecerá melhor desempenho.

- Os parâmetros **search_value** e **alternateResult** são avaliados antes da função iterar nas linhas da tabela de pesquisa.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

Os exemplos deste artigo podem ser adicionados ao modelo de exemplo do Power BI Desktop. Para obter o modelo, confira Modelo de exemplo DAX.

A coluna calculada a seguir definida na tabela **Sales** usa a função LOOKUPVALUE para retornar valores de canal da tabela **Sales Order**.

```
CHANNEL = LOOKUPVALUE('Sales Order'[Channel],'Sales Order'[SalesOrderLineKey],[SalesOrderLineKey])
```

No entanto, nesse caso, como há uma relação entre as tabelas **Sales Order** e **Sales**, será mais eficiente usar a função RELATED.

```
CHANNEL = RELATED('Sales Order'[Channel])
```

Confira também

Função RELATED (DAX)
Funções informativas

REMOVEFILTERS

17/03/2021 • 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.

Retornar valor

N/D Consulte Observações.

Comentários

- A função REMOVEFILTERS só pode ser usada para limpar filtros, mas não para retornar uma tabela.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
Acessórios	Falso	700.759,96	0,023868921434441
Clothing	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
Bikes	Mountain bikes	9.952.759,564	0,35146228

DIMPRODUCTCATEGORY [ENGLISHPRODUCTCATEGORYNAME]	DIMPRODUCTSUBCATEGORY [ENGLISHPRODUCTSUBCATEGORYNAME]	[TOTALSALES]	[%SALES]
Bikes	Bicicletas de estrada	14520584,04	0,51276608
Bikes	Bicicletas de passeio	3.844.801,05	0,13577164
Clothing	Bonés	19.688,1	0,05794493
Clothing	Luvas	35.020,7	0,10307099
Clothing	Camisas	172.950,68	0,5090189
Clothing	Shorts	71.319,81	0,20990453
Clothing	Meias	5.106,32	0,01502864
Clothing	Coletes	35.687	0,10503201

SELECTEDVALUE

17/03/2021 • 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>)`.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A seguinte consulta DAX:

```
DEFINE
 MEASURE DimProduct[Selected Color] = SELECTEDVALUE(DimProduct[Color], "No Single Selection")
EVALUATE
 SUMMARIZECOLUMNS
   (ROLLUPADDISSUBTOTAL(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	FALSE	Preto
Azul	FALSE	Azul
Cinza	FALSE	Cinza
Multi	FALSE	Multi
NA	FALSE	NA
Vermelho	FALSE	Vermelho
Prata	FALSE	Prata
Prata/Preto	FALSE	Prata/Preto
Branca	FALSE	Branca
Amarelo	FALSE	Amarelo
	TRUE	Nenhuma Seleção Única

Funções financeiras

17/03/2021 • 8 minutes to read

As funções financeiras no DAX são usadas em fórmulas que fazem cálculos financeiros, como o valor líquido atual e a taxa de retorno. Essas funções são semelhantes às funções financeiras usadas no Microsoft Excel.

Nesta categoria

FUNÇÃO	DESCRIÇÃO
ACCRINT	Retorna os juros acumulados de um título que paga juros periódicos.
ACCRINTM	Retorna os juros acumulados de um título que paga juros no vencimento.
AMORDEGRC	Retorna a depreciação para cada período contábil. Semelhante a AMORLINC, exceto que pelo fato de que um coeficiente de depreciação é aplicado, dependendo da vida útil dos ativos.
AMORLINC	Retorna a depreciação para cada período contábil.
COUPDAYBS	Retorna o número de dias desde o início de um período de cupom até a data de liquidação.
COUPDAYS	Retorna o número de dias no período de cupom que contém a data de liquidação.
COUPDAYSNC	Retorna o número de dias desde a data de liquidação até a próxima data do cupom.
COUPNCD	Retorna a próxima data do cupom após a data de liquidação.
COUPNUM	Retorna o número de cupons a pagar entre a data de liquidação e a data de vencimento, arredondado para o cupom inteiro mais próximo.
COUPPCD	Retorna a data do cupom anterior à data de liquidação.
CUMIPMT	Retorna os juros acumulados pagos em um empréstimo entre start_period e end_period.
CUMPRINC	Retorna o valor principal cumulativo pago em um empréstimo entre start_period e end_period.
DB	Retorna a depreciação de um ativo em um período especificado usando o método de saldo decrescente fixo.

FUNÇÃO	DESCRIÇÃO
DDB	Retorna a depreciação de um ativo em um período especificado usando o método de saldo decrescente duplo ou outro método de saldo especificado.
DISC	Retorna a taxa de desconto de um título.
DOLLARDE	Converte um preço em moeda expresso como uma parte inteira e uma parte fracionária, como 1,02, em um preço em moeda expresso como um número decimal.
DOLLARFR	Converte um preço em moeda expresso como uma parte inteira e uma parte fracionária, como 1,02, em um preço em moeda expresso como um número decimal.
DURATION	Retorna a duração de Macaulay para um valor nominal presumido de US$ 100.
EFFECT	Retorna a taxa de juros anual efetiva, conforme a taxa de juros anual nominal e o número de períodos compostos por ano.
FV	Calcula o valor futuro de um investimento com base em uma taxa de juros constante.
INTRATE	Retorna a taxa de juros de um título totalmente investido.
IPMT	Retorna o pagamento de juros de um investimento em determinado período com base em pagamentos constantes e periódicos e uma taxa de juros constante.
ISPMT	Calcula os juros pagos (ou recebidos) para o período especificado de um empréstimo (ou investimento) com pagamentos iguais do valor principal.
MDURATION	Retorna a duração de Macaulay modificada para um título com um valor nominal presumido de US$ 100.
NOMINAL	Retorna a taxa de juros anual nominal, considerando a taxa efetiva e o número de períodos compostos por ano.
NPER	Retorna o número de períodos de um investimento com base em pagamentos periódicos e constantes e uma taxa de juros constante.
ODDFPRICE	Retorna o preço por US$ 100 de valor nominal de um título com um período inicial (curto ou longo) indefinido.
ODDFYIELD	Retorna o rendimento de um título que tem um período inicial (curto ou longo) indefinido.
ODDLPRICE	Retorna o preço por US$ 100 de valor nominal de um título com um período final de cupom (curto ou longo) indefinido.

FUNÇÃO	DESCRIÇÃO
ODDLYIELD	Retorna o rendimento de um título que tem um período final (curto ou longo) indefinido.
PDURATION	Retorna o número de períodos exigidos por um investimento para alcançar um valor especificado.
PMT	Calcula o pagamento de um empréstimo com base em pagamentos e uma taxa de juros constantes.
PPMT	Retorna o pagamento do valor principal de um investimento em determinado período com base em pagamentos constantes e periódicos e uma taxa de juros constante.
PRICE	Retorna o preço por US$ 100 de valor nominal de um título que paga juros periódicos.
PRICEDISC	Retorna o preço por US$ 100 de valor nominal de um título com desconto.
PRICEMAT	Retorna o preço por US$ 100 de valor nominal de um título que paga juros no vencimento.
PV	Calcula o valor atual de um empréstimo ou de um investimento com base em uma taxa de juros constante.
RATE	Retorna a taxa de juros por período de uma anuidade.
RECEIVED	Retorna o valor recebido no vencimento de um título totalmente investido.
RRI	Retorna uma taxa de juros equivalente para o crescimento de um investimento.
SLN	Retorna a depreciação de linha reta de um ativo para um período.
SYD	Retorna uma depreciação dos dígitos da soma dos anos de um ativo para um período especificado.
TBILLEQ	Retorna o rendimento equivalente do título de dívida de uma Letra do Tesouro.
TBILLPRICE	Retorna o preço por US$ 100 de valor nominal de uma Letra do Tesouro.
TBILLYIELD	Retorna o rendimento de uma Letra do Tesouro.
VDB	Retorna a depreciação de um ativo para qualquer período especificado, incluindo períodos parciais, usando o método de saldo decrescente duplo ou outro método especificado.
XIRR	Retorna a taxa interna de retorno de um agendamento de fluxos de caixa que não é necessariamente periódico.

FUNÇÃO	DESCRIÇÃO
XNPV	Retorna o valor atual de um agendamento de fluxos de caixa que não é necessariamente periódico.
YIELD	Retorna o rendimento de um título que paga juros periódicos.
YIELDDISC	Retorna o rendimento anual de um título com desconto.
YIELDMAT	Retorna o rendimento anual de um título que paga juros no vencimento.

ACCRINT

17/03/2021 • 5 minutes to read

Retorna os juros acumulados de um título que paga juros periódicos.

Sintaxe

```
ACCRINT(<issue>, <first_interest>, <settlement>, <rate>, <par>, <frequency>[, <basis>[, <calc_method>]])
```

Parâmetros

TERMO	DEFINIÇÃO
problema	A data de emissão do título.
first_interest	A data inicial de juros do título.
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
rate	A taxa de cupom anual do título.
par	O valor nominal do título.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.
calc_method	(Opcional) Um valor lógico que especifica a maneira de calcular o total de juros acumulados quando a data de liquidação é posterior à data de first_interest. Se calc_method for omitido, ele será considerado TRUE. – Se calc_method for avaliado como TRUE ou for omitido, ACCRINT retornará o total de juros acumulados desde issue até settlement. – Se calc_method for avaliado como FALSE, ACCRINT retornará os juros acumulados desde first_interest até settlement.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360

BASE	BASE DE CONTAGEM DIÁRIA
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O juro acumulado.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- ACCRINT é calculado da seguinte maneira:

  $$\text{ACCRINT} = \text{par} \times \frac{\text{rate}}{\text{frequency}} \times \sum^{\text{NC}}_{i=1}\frac{\text{A}_{i}}{\text{NL}_{i}}$$

  em que:

  - A_{i} = número de dias acumulados para o período de cupom parcial i^{th} no período indefinido.
  - NC = número de períodos de um cupom parcial que se ajustam no período indefinido. Se esse número contiver uma fração, eleve-o para o próximo número inteiro.
  - NL_{i} = duração normal em dias do período de um cupom parcial em um período indefinido.
- issue, first_interest e settlement são arredondados para números inteiros.
- frequency e basis são arredondados para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - issue, first_interest ou settlement não for uma data válida.
  - issue ≥ settlement.
  - rate ≤ 0.
  - par ≤ 0.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplos

DADOS	DESCRIÇÃO
1 de março de 2007	Data de emissão

DADOS	DESCRIÇÃO
31 de agosto de 2008	Data inicial dos juros
1º de maio de 2008	Data de liquidação
10%	Taxa de cupom
1000	Valor nominal
2	A frequência é semestral (confira acima)
0	Base 30/360 (confira acima)

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE
{
  ACCRINT(DATE(2007,3,1), DATE(2008,8,31), DATE(2008,5,1), 0.1, 1000, 2, 0)
}
```

Retorna os juros acumulados desde issue até settlement para um título com os termos especificados acima.

[VALUE]
116,944444444444

Exemplo 2

A seguinte consulta DAX:

```
EVALUATE
{
  ACCRINT(DATE(2007,3,1), DATE(2008,8,31), DATE(2008,5,1), 0.1, 1000, 2, 0, FALSE)
}
```

Retorna os juros acumulados desde first_interest até settlement para um título com os termos especificados acima.

[VALUE]
66,9444444444445

ACCRINTM

17/03/2021 • 3 minutes to read

Retorna os juros acumulados de um título que paga juros no vencimento.

Sintaxe

```
ACCRINTM(<issue>, <maturity>, <rate>, <par>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
problema	A data de emissão do título.
maturity	A data de vencimento do título.
rate	A taxa de cupom anual do título.
par	O valor nominal do título.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O juro acumulado.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.

- ACCRINTM é calculado da seguinte maneira:

  $$\text{ACCRINTM} = \text{par} \times \text{rate} \times \frac{\text{A}}{\text{D}}$$

  em que:

  - A = número de dias acumulados contados de acordo com uma base mensal. Para os juros em itens de maturidade, é usado o número de dias da data de emissão à data de vencimento.
  - D = base anual.
- issue e maturity são arredondados para números inteiros.
- basis é arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - issue ou maturity não for uma data válida.
  - issue ≥ maturity.
  - rate ≤ 0.
  - par ≤ 0.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
1º de abril de 2008	Data de emissão
15 de junho de 2008	Data de vencimento
10%	Cupom percentual
1000	Valor nominal
3	Base real/365 (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  ACCRINTM(DATE(2008,4,1), DATE(2008,6,15), 0.1, 1000, 3)
}
```

Retorna os juros acumulados para um título com os termos especificados acima.

[VALUE]
20,5479452054795

AMORDEGRC

17/03/2021 • 4 minutes to read

Retorna a depreciação para cada período contábil. Essa função é fornecida para o sistema de contabilidade francês. Se um ativo for adquirido no meio do período contábil, será levada em conta a depreciação proporcional. A função é semelhante a AMORLINC, exceto pelo fato de que um coeficiente de depreciação é aplicado no cálculo, dependendo da vida útil dos ativos.

Sintaxe

```
AMORDEGRC(<cost>, <date_purchased>, <first_period>, <salvage>, <period>, <rate>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
cost	O custo do ativo.
date_purchased	A data da compra do ativo.
first_period	A data do final do primeiro período.
salvage	O valor de recuperação no final da vida útil do ativo.
period	O período.
rate	A taxa de depreciação.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	SISTEMA DE DATAS
0 ou omitido	360 dias (método NASD)
1	Real
3	365 dias em um ano
4	360 dias em um ano (método europeu)

Valor Retornado

A depreciação para cada período contábil.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- Essa função retornará a depreciação até o último período da vida útil dos ativos ou até que o valor total de depreciação seja maior do que o custo dos ativos menos o valor de recuperação.
- Os coeficientes de depreciação são:

VIDA ÚTIL DE ATIVOS (1/TAXA)	COEFICIENTE DE DEPRECIAÇÃO
Entre 3 e 4 anos	1.5
Entre 5 e 6 anos	2
Mais de 6 anos	2.5

- A taxa de depreciação crescerá até 50% para o período anterior ao último e crescerá até 100% para o último período.
- period e basis são arredondadas para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - cost < 0.
  - first_period ou date_purchased não for uma data válida.
  - date_purchased > first_period.
  - salvage < 0 ou salvage > cost.
  - period < 0.
  - rate ≤ 0.
  - A vida útil dos ativos estiver entre 0 (zero) e 1, 1 e 2, 2 e 3 ou 4 e 5.
  - basis for qualquer número diferente de 0, 1, 3 ou 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
2400	Cost
19 de agosto de 2008	Data de compra
31 de dezembro de 2008	Fim do primeiro período
300	Valor residual
1	Período
15%	Taxa de depreciação

DADOS	DESCRIÇÃO
1	Base real (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  AMORDEGRC(2400, DATE(2008,8,19), DATE(2008,12,31), 300, 1, 0.15, 1)
}
```

Retorna a depreciação do primeiro período, considerando os termos especificados acima.

[VALUE]
776

AMORLINC

17/03/2021 • 3 minutes to read

Retorna a depreciação para cada período contábil. Essa função é fornecida para o sistema de contabilidade francês. Se um ativo for adquirido no meio do período contábil, será levada em conta a depreciação proporcional.

Sintaxe

```
AMORLINC(<cost>, <date_purchased>, <first_period>, <salvage>, <period>, <rate>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
cost	O custo do ativo.
date_purchased	A data da compra do ativo.
first_period	A data do final do primeiro período.
salvage	O valor de recuperação no final da vida útil do ativo.
period	O período.
rate	A taxa de depreciação.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	SISTEMA DE DATAS
0 ou omitido	360 dias (método NASD)
1	Real
3	365 dias em um ano
4	360 dias em um ano (método europeu)

Valor Retornado

A depreciação para cada período contábil.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- period e basis são arredondadas para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - cost < 0.
  - first_period ou date_purchased não for uma data válida.
  - date_purchased > first_period.
  - salvage < 0 ou salvage > cost.
  - period < 0.
  - rate ≤ 0.
  - basis for qualquer número diferente de 0, 1, 3 ou 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
2400	Cost
19 de agosto de 2008	Data de compra
31 de dezembro de 2008	Fim do primeiro período
300	Valor residual
1	Período
15%	Taxa de depreciação
1	Base real (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  AMORLINC(2400, DATE(2008,8,19), DATE(2008,12,31), 300, 1, 0.15, 1)
}
```

Retorna a depreciação do primeiro período, considerando os termos especificados acima.

[VALUE]
360

COUPDAYBS

17/03/2021 • 4 minutes to read

Retorna o número de dias desde o início de um período de cupom até a data de liquidação.

Sintaxe

```
COUPDAYBS(<settlement>, <maturity>, <frequency>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O número de dias desde o início de um período de cupom até a data de liquidação.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos.

No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.

- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, 30 anos após a data de emissão em 1º de janeiro de 2008.
- settlement e maturity serão arredondados para números inteiros.
- frequency e basis são arredondados para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
25-jan-11	Data de liquidação
15-nov-11	Data de vencimento
2	Cupom semestral (confira acima)
1	Real/base real (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  COUPDAYBS(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Retorna o número de dias desde o início do período do cupom até a data de liquidação para um título de dívida com termos acima.

[VALUE]
71

COUPDAYS

17/03/2021 • 4 minutes to read

Retorna o número de dias no período de cupom que contém a data de liquidação.

Sintaxe

```
COUPDAYS(<settlement>, <maturity>, <frequency>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O número de dias no período de cupom que contém a data de liquidação.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos.

No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.

- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento será 1º de janeiro de 2038, 30 anos após a data de emissão em 1º de janeiro de 2008.
- settlement e maturity serão arredondados para números inteiros.
- frequency e basis são arredondados para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
25-jan-11	Data de liquidação
15-nov-11	Data de vencimento
2	Cupom semestral (confira acima)
1	Real/base real (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  COUPDAYS(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Retorna o número de dias no período de cupom que contém a data de liquidação para um título de dívida com os termos especificados acima.

[VALUE]
181

COUPDAYSNC

17/03/2021 • 4 minutes to read

Retorna o número de dias desde a data de liquidação até a próxima data do cupom.

Sintaxe

```
COUPDAYSNC(<settlement>, <maturity>, <frequency>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O número de dias desde a data de liquidação até a próxima data do cupom.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos.

No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.

- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- settlement e maturity serão arredondados para números inteiros.
- frequency e basis são arredondados para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
25-jan-11	Data de liquidação
15-nov-11	Data de vencimento
2	Cupom semestral (confira acima)
1	Real/base real (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  COUPDAYSNC(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Retorna o número de dias a partir da data de liquidação até a próxima data do cupom para um título de dívida com os termos especificados acima.

[VALUE]
110

COUPNCD

17/03/2021 • 4 minutes to read

Retorna a próxima data do cupom após a data de liquidação.

Sintaxe

```
COUPNCD(<settlement>, <maturity>, <frequency>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

A próxima data do cupom após a data de liquidação.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos.

No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.

- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento será 1º de janeiro de 2038, 30 anos após a data de emissão em 1º de janeiro de 2008.
- settlement e maturity serão arredondados para números inteiros.
- frequency e basis são arredondados para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
25-jan-11	Data de liquidação
15-nov-11	Data de vencimento
2	Cupom semestral (confira acima)
1	Real/base real (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  COUPNCD(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Retorna a próxima data do cupom após a data de quitação para um título de dívida com os termos especificados acima.

[VALUE]
15/05/2011 12:00:00

COUPNUM

17/03/2021 • 4 minutes to read

Retorna o número de cupons a pagar entre a data de liquidação e a data de vencimento, arredondado para o cupom inteiro mais próximo.

Sintaxe

```
COUPNUM(<settlement>, <maturity>, <frequency>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O número de cupons a pagar entre a data de liquidação e a data de vencimento.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- settlement e maturity serão arredondados para números inteiros.
- frequency e basis são arredondados para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
25-jan-07	Data de liquidação
15-nov-08	Data de vencimento
2	Cupom semestral (confira acima)
1	Real/base real (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  COUPNUM(DATE(2007,1,25), DATE(2008,11,15), 2, 1)
}
```

Retorna o número de pagamentos de cupom para um título de dívida com os termos especificados acima.

[VALUE]
4

COUPPCD

17/03/2021 • 4 minutes to read

Retorna a data do cupom anterior à data de liquidação.

Sintaxe

```
COUPPCD(<settlement>, <maturity>, <frequency>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

A data do cupom anterior à data de liquidação.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos.

No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.

- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- settlement e maturity serão arredondados para números inteiros.
- frequency e basis são arredondados para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
25-jan-11	Data de liquidação
15-nov-11	Data de vencimento
2	Cupom semestral (confira acima)
1	Real/base real (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  COUPPCD(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Retorna a data do cupom anterior antes da data de quitação para um título de dívida que usa os termos especificados acima.

[VALUE]
15/11/2010 12:00:00

CUMIPMT

17/03/2021 • 3 minutes to read

Retorna os juros acumulados pagos em um empréstimo entre start_period e end_period.

Sintaxe

```
CUMIPMT(<rate>, <nper>, <pv>, <start_period>, <end_period>, <type>)
```

Parâmetros

TERMO	DEFINIÇÃO
rate	A taxa de juros.
nper	O número total de períodos de pagamento.
pv	O valor atual.
start_period	O primeiro período no cálculo. Deve estar entre 1 e end_period (inclusive).
end_period	O último período no cálculo. Precisa estar entre start_period e nper (inclusive).
tipo	O momento do pagamento. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **type** aceitará os seguintes valores:

TIPO	TIMING
0 (zero)	Pagamento no final do período
1	Pagamento no início do período

Valor Retornado

Os juros acumulados pagos no período especificado.

Comentários

- Seja consistente em relação às unidades que você usa para especificar rate e nper. Caso faça pagamentos mensais em um empréstimo de quatro anos a uma taxa de juros anual de 10%, use 0,1/12 para rate e 4*12 para nper. Caso faça pagamentos anuais no mesmo empréstimo, use 0,1 para rate e 4 para nper.
- start_period, end_period e type são arredondados para o número inteiro mais próximo.
- Um erro será retornado se:

- rate ≤ 0.
- nper < 1.
- pv ≤0.
- start_period < 1 ou start_period > end_period.
- end_period <start_period ou end_period >nper.
- type é qualquer número diferente de 0 ou 1.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplos

DADOS	DESCRIÇÃO
9%	Taxa de juros anual
30	Anos do empréstimo
125000	Valor atual

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE
{
  CUMIPMT(0.09/12, 30*12, 125000, 13, 24, 1)
}
```

Retorna o total de juros pagos no segundo ano de pagamentos, períodos de 13 a 24, supondo que os pagamentos sejam feitos no início de cada mês.

[VALUE]
-11052,3395838718

Exemplo 2

A seguinte consulta DAX:

```
EVALUATE
{
  CUMIPMT(0.09/12, 30*12, 125000, 1, 1, 0)
}
```

Retorna os juros pagos em um pagamento no primeiro mês, supondo que o pagamento seja feito no final do mês.

[VALUE]
-937,5

CUMPRINC

17/03/2021 • 3 minutes to read

Retorna o valor principal cumulativo pago em um empréstimo entre start_period e end_period.

Sintaxe

```
CUMPRINC(<rate>, <nper>, <pv>, <start_period>, <end_period>, <type>)
```

Parâmetros

TERMO	DEFINIÇÃO
rate	A taxa de juros.
nper	O número total de períodos de pagamento.
pv	O valor atual.
start_period	O primeiro período no cálculo. Deve estar entre 1 e end_period (inclusive).
end_period	O último período no cálculo. Precisa estar entre start_period e nper (inclusive).
tipo	O momento do pagamento. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **type** aceitará os seguintes valores:

TIPO	TIMING
0 (zero)	Pagamento no final do período
1	Pagamento no início do período

Valor Retornado

O valor principal acumulado pago no período especificado.

Comentários

- Seja consistente em relação às unidades que você usa para especificar rate e nper. Caso faça pagamentos mensais em um empréstimo de quatro anos a uma taxa de juros anual de 10%, use 0,1/12 para rate e 4*12 para nper. Caso faça pagamentos anuais no mesmo empréstimo, use 0,1 para rate e 4 para nper.
- start_period, end_period e type são arredondados para o número inteiro mais próximo.
- Um erro será retornado se:

- rate ≤ 0.
- nper < 1.
- pv ≤0.
- start_period < 1 ou start_period > end_period.
- end_period <start_period ou end_period >nper.
- type é qualquer número diferente de 0 ou 1.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplos

DADOS	DESCRIÇÃO
9%	Taxa de juros anual
30	Prazo em anos
125000	Valor atual

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE
{
  CUMPRINC(0.09/12, 30*12, 125000, 13, 24, 1)
}
```

Retorna o valor principal total pago no segundo ano de pagamentos, períodos de 13 a 24, supondo que os pagamentos sejam feitos no início de cada mês.

[VALUE]
-927,153472378062

Exemplo 2

A seguinte consulta DAX:

```
EVALUATE
{
  CUMPRINC(0.09/12, 30*12, 125000, 1, 1, 0)
}
```

Retorna o valor principal pago em um pagamento no primeiro mês, supondo que o pagamento seja feito no final do mês.

[VALUE]
-68,2782711809784

DB

17/03/2021 • 3 minutes to read

Retorna a depreciação de um ativo em um período especificado usando o método de saldo decrescente fixo.

Sintaxe

```
DB(<cost>, <salvage>, <life>, <period>[, <month>])
```

Parâmetros

TERMO	DEFINIÇÃO
cost	O custo inicial do ativo.
salvage	O valor no final da depreciação (às vezes chamado de valor residual do ativo). Esse valor pode ser 0.
vida	O número de períodos sobre o qual o ativo está sendo depreciado (às vezes chamado de vida útil do ativo).
period	O período para o qual você deseja calcular a depreciação. Em period, use as mesmas unidades que life. Precisa estar entre 1 e life (inclusive).
month	(Opcional) O número de meses no primeiro ano. Se month for omitido, ele será considerado 12.

Valor Retornado

A depreciação durante o período especificado.

Comentários

- O método de saldo decrescente fixo computa a depreciação a uma taxa fixa. DB usa as seguintes fórmulas para calcular a depreciação de um período:

  $$(\text{cost} - \text{total depreciation from prior periods}) \times \text{rate}$$

  em que:

  - $\text{rate} = 1 - ((\frac{\text{salvage}}{\text{cost}})^{(\frac{1}{\text{life}})})\text{, rounded to three decimal places}$
- A depreciação do primeiro e do último período é um caso especial.
  - Para o primeiro período, DB usa esta fórmula:

    $$\frac{\text{cost} \times \text{rate} \times \text{month}}{12}$$

  - Para o último período, DB usa esta fórmula:

    $$\frac{(\text{cost} - \text{total depreciation from prior periods}) \times \text{rate} \times (12 -

\text{month}))}{12}$$

- period e month são arredondados para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - cost < 0.
  - salvage < 0.
  - life < 1.
  - period < 1 ou period >life.
  - month <1 ou month >12.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplos

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE
{
  DB(1000000, 0, 6, 1, 2)
}
```

Retorna a depreciação de um ativo nos últimos dois meses do primeiro ano, supondo que valerá \$0 após 6 anos.

[VALUE]
166666,666666667

Exemplo 2

O exemplo a seguir calcula a depreciação total de todos os ativos em anos diferentes durante seus tempos de vida. Aqui, o primeiro ano inclui apenas sete meses de depreciação e o último ano inclui apenas cinco meses.

```
DEFINE
VAR NumDepreciationPeriods = MAX(Asset[LifeTimeYears])+1
VAR DepreciationPeriods = GENERATESERIES(1, NumDepreciationPeriods)
EVALUATE
ADDCOLUMNS (
  DepreciationPeriods,
  "Current Period Total Depreciation",
  SUMX (
    FILTER (
      Asset,
      [Value] <= [LifetimeYears]+1
    ),
    DB([InitialCost], [SalvageValue], [LifetimeYears], [Value], 7)
  )
)
```

DDB

17/03/2021 • 3 minutes to read

Retorna a depreciação de um ativo em um período especificado usando o método de saldo decrescente duplo ou outro método de saldo especificado.

Sintaxe

```
DDB(<cost>, <salvage>, <life>, <period>[, <factor>])
```

Parâmetros

TERMO	DEFINIÇÃO
cost	O custo inicial do ativo.
salvage	O valor no final da depreciação (às vezes chamado de valor residual do ativo). Esse valor pode ser 0.
vida	O número de períodos sobre o qual o ativo está sendo depreciado (às vezes chamado de vida útil do ativo).
period	O período para o qual você deseja calcular a depreciação. Em period, use as mesmas unidades que life. Precisa estar entre 1 e life (inclusive).
fator	(Opcional) A taxa na qual o saldo diminui. Se factor for omitido, ele será considerado 2 (o método de saldo decrescente duplo).

Valor Retornado

A depreciação durante o período especificado.

Comentários

- O método de saldo decrescente duplo computa a depreciação a uma taxa acelerada. A depreciação é mais alta no primeiro período e diminui em períodos sucessivos. DDB usa a seguinte fórmula para calcular a depreciação de um período:

  $$\text{Min}((\text{cost} - \text{total depreciation from prior periods}) \times (\frac{\text{factor}}{\text{life}}),(\text{cost} - \text{salvage} - \text{total depreciation from prior periods}))$$

- Altere factor se você não quiser usar o método de saldo decrescente duplo.
- Use a função VDB para alternar para o método de depreciação de linha reta quando a depreciação for maior que o cálculo de saldo decrescente.
- period é arredondado para o número inteiro mais próximo.
- Um erro será retornado se:

- cost < 0.
- salvage < 0.
- life < 1.
- period < 1 ou period >life.
- factor ≤0.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplos

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE
{
  DDB(1000000, 0, 10, 5, 1.5)
}
```

Retorna a depreciação de um ativo no prazo de 5th ano, supondo que valerá \$0 após 10 anos. Esse cálculo usa um fator de 1,5.

[VALUE]
78300,9375

Exemplo 2

O exemplo a seguir calcula a depreciação total de todos os ativos em anos diferentes durante seus tempos de vida. Esse cálculo usa o fator padrão de 2 (o método de saldo decrescente duplo).

```
DEFINE
VAR NumDepreciationPeriods = MAX(Asset[LifeTimeYears])
VAR DepreciationPeriods = GENERATESERIES(1, NumDepreciationPeriods)
EVALUATE
  ADDCOLUMNS (
  DepreciationPeriods,
  "Current Period Total Depreciation",
  SUMX (
    FILTER (
      Asset,
      [Value] <= [LifetimeYears]
    ),
    DDB([InitialCost], [SalvageValue], [LifetimeYears], [Value])
  )
)
```

DISC

17/03/2021 • 4 minutes to read

Retorna a taxa de desconto de um título.

Sintaxe

```
DISC(<settlement>, <maturity>, <pr>, <redemption>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
pr	O preço do título por \US$ 100 de valor nominal.
redemption	O valor de resgate do título por \US$ 100 de valor nominal.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

A taxa de desconto.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são

39.448 dias após 30 de dezembro de 1899.

- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2018 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2018, a data de liquidação seria 1º de julho de 2018 e a data de vencimento seria 1º de janeiro de 2048, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2018.
- A função DISC será calculada da seguinte maneira:

  $$\text{DISC} = \frac{\text{redemption} - \text{par}}{\text{redemption}} \times \frac{\text{B}}{\text{DSM}}$$

  em que:

  - B = número de dias em um ano, dependendo da base anual.
  - DSM = número de dias entre a liquidação e o vencimento.
- settlement e maturity serão arredondados para números inteiros.
- basis é arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - pr ≤ 0.
  - redemption ≤ 0.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
01/07/2018	Data de liquidação
01/01/2048	Data de vencimento
97,975	Preço
100	Valor de resgate
1	Real/base real (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  DISC(DATE(2018,7,1), DATE(2048,1,1), 97.975, 100, 1)
}
```

Retorna a taxa de desconto do título de um título de dívida com os termos especificados acima.

[VALUE]
0,000686384169121348

DOLLARDE

17/03/2021 • 2 minutes to read

Converte um preço em moeda expresso como uma parte inteira e uma parte fracionária, como 1,02, em um preço em moeda expresso como um número decimal. Números fracionários em moeda às vezes são usados para obter preços de títulos.

Sintaxe

```
DOLLARDE(<fractional_dollar>, <fraction>)
```

Parâmetros

TERMO	DEFINIÇÃO
fractional_dollar	Um número expresso como uma parte inteira e uma parte fracionária, separado por um símbolo decimal.
fração	O número inteiro a ser usado no denominador da fração.

Valor Retornado

O valor decimal de *fractional_dollar*.

Comentários

- A parte fracionária do valor será dividida pelo número inteiro que você especificar. Por exemplo, caso queira que o preço seja expresso em uma precisão de 1/16 de moeda, divida a parte fracionária por 16. Nesse caso, 1,02 representará \US$ 1,125 (\US$ 1 + 2/16 = \US$ 1,125).
- A fração será arredondada para o número inteiro mais próximo.
- Um erro será retornado se:
  - fração < 1.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A seguinte consulta DAX:

```
EVALUATE
{
  DOLLARDE(1.02, 16)
}
```

Retorna 1,125, o preço decimal do preço fracionário original, 1,02, lido como 1 e 2/16. Como o valor de fração é 16, o preço tem uma precisão de 1/16 de moeda.

DOLLARFR

17/03/2021 • 2 minutes to read

Converte um preço em moeda expresso como um número decimal em um preço em moeda expresso como uma parte inteira e uma parte da fracionária, como 1,02. Números fracionários em moeda às vezes são usados para obter preços de títulos.

Sintaxe

```
DOLLARFR(<decimal_dollar>, <fraction>)
```

Parâmetros

TERMO	DEFINIÇÃO
decimal_dollar	Um número decimal.
fração	O número inteiro a ser usado no denominador da fração.

Valor Retornado

O valor fracionário de *decimal_dollar* expresso como uma parte inteira e uma parte fracionária.

Comentários

- A fração será arredondada para o número inteiro mais próximo.
- Um erro será retornado se:
  - fração < 1.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A seguinte consulta DAX:

```
EVALUATE
{
  DOLLARFR(1.125, 16)
}
```

Retorna 1,02, lido como 1 e 2/16, que é o preço fracionário correspondente ao preço decimal original de 1,125. Como o valor de fração é 16, o preço tem uma precisão de 1/16 de moeda.

DURAÇÃO

17/03/2021 • 4 minutes to read

Retorna a duração de Macaulay para um valor nominal presumido de \US$ 100. A duração será definida como a média ponderada do valor presente nos fluxos de caixa. Além disso, ela será usada como uma medida da resposta do preço de um título de dívida às alterações no rendimento.

Sintaxe

```
DURATION(<settlement>, <maturity>, <coupon>, <yld>, <frequency>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
cupom	A taxa de cupom anual do título.
yld	O rendimento anual do título.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

A duração de Macaulay.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento será 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- settlement e maturity serão arredondados para números inteiros.
- frequency e basis serão arredondadas para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - cupom < 0.
  - yld < 0
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
01/07/2018	Data de liquidação
01/01/2048	Data de vencimento
8,0%	Cupom percentual
9,0%	Percentual de rendimento
2	A frequência é semestral (confira acima)
1	Real/base real (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  DURATION(DATE(2018,7,1), DATE(2048,1,1), 0.08, 0.09, 2, 1)
}
```

Retorna a duração de Macaulay para um título de dívida com os termos especificados acima.

[VALUE]
10,9191452815919

EFFECT

17/03/2021 • 2 minutes to read

Retorna a taxa de juros anual efetiva, conforme a taxa de juros anual nominal e o número de períodos compostos por ano.

Sintaxe

```
EFFECT(<nominal_rate>, <npery>)
```

Parâmetros

TERMO	DEFINIÇÃO
nominal_rate	A taxa de juros nominal.
npery	O número de períodos compostos por ano.

Valor Retornado

A taxa de juros anual efetiva.

Comentários

- A função EFFECT será calculada da seguinte maneira:

  $$\text{EFFECT} = \bigg(1 + \frac{\text{nominal_rate}}{\text{npery}} \bigg)^{\text{npery}} - 1$$

- npery será arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - nominal_rate ≤ 0.
  - npery < 1.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
5,25%	Taxa de juros nominal
4	Número de períodos compostos por ano

A seguinte consulta DAX:

```
EVALUATE
{
  EFFECT(0.0525, 4)
}
```

Retorna uma taxa de juros efetiva usando os termos especificados acima.

[VALUE]
0,0535426673707584

FV

17/03/2021 • 3 minutes to read

Calcula o valor futuro de um investimento com base em uma taxa de juros constante. Será possível usar a função FV com pagamentos periódicos, constantes e/ou com um pagamento da quantia total.

Sintaxe

```
FV(<rate>, <nper>, <pmt>[, <pv>[, <type>]])
```

Parâmetros

TERMO	DEFINIÇÃO
rate	A taxa de juros por período.
nper	O número total de períodos de pagamento em uma anuidade.
pmt	O pagamento feito em cada período. Ele não poderá ser alterado durante a vida da anuidade. O pmt geralmente contém o valor principal e os juros, mas nenhuma outra taxa nem imposto.
pv	(Opcional) O valor atual ou o valor da quantia total de uma série de pagamentos futuros. Caso pv seja omitido, ele será considerado BLANK.
tipo	(Opcional) O número 0 ou 1 que indica quando os pagamentos vencem. Caso type seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **type** aceitará os seguintes valores:

DEFINIR TYPE COMO IGUAL A	CASO OS PAGAMENTOS ESTEJAM VENCIDOS
0 ou omitido	No final do período
1	No início do período

Observação: Para obter uma descrição completa de argumentos na função FV, bem como obter mais informações sobre as funções de anuidade, confira a função PV.

Valor Retornado

O valor futuro de um investimento.

Comentários

- Seja consistente em relação às unidades que você usa para especificar rate e nper. Caso faça pagamentos

mensais em um empréstimo de quatro anos a uma taxa anual de juros de 12%, use 0,12/12 para obter rate e 4*12 para obter nper. Caso faça pagamentos anuais no mesmo empréstimo, use 0,12 para rate e 4 para nper.

- Para todos os argumentos, o dinheiro pago, como depósitos em poupança, será representado por números negativos. Já o dinheiro recebido, como cheques de dividendos, será representado por números positivos.
- type será arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - nper < 1
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
6%	Taxa de juros anual
10	Número de pagamentos
-200	Valor do pagamento
-500	Valor atual
1	O pagamento será devido no início do período (0 indica que o pagamento será devido no fim do período)

A seguinte consulta DAX:

```
EVALUATE
{
  FV(0.06/12, 10, -200, -500, 1)
}
```

Retorna o valor futuro de um investimento usando os termos especificados acima.

[VALUE]
2581,40337406012

INTRATE

17/03/2021 • 4 minutes to read

Retorna a taxa de juros de um título totalmente investido.

Sintaxe

```
INTRATE(<settlement>, <maturity>, <investment>, <redemption>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
investment	O valor investido no título.
redemption	O valor a ser recebido no vencimento.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

A taxa de juros.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são

39.448 dias após 30 de dezembro de 1899.

- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento será 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- A função INTRATE será calculada da seguinte maneira:

  $$\text{INTRATE} = \frac{\text{redemption} - \text{investment}}{\text{investment}} \times \frac{\text{B}}{\text{DIM}}$$

  em que:

  - B = número de dias em um ano, dependendo da base anual.
  - DIM = número de dias da liquidação ao vencimento.
- settlement e maturity serão arredondados para números inteiros.
- basis é arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - investment ≤ 0.
  - redemption ≤ 0.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
15/02/2008	Data de liquidação
15/05/2008	Data de vencimento
\US$ 1.000.000	Investimento
\US$ 1.014.420	Valor de resgate
2	Base real/360

A seguinte consulta DAX:

```
EVALUATE
{
  INTRATE(DATE(2008,2,15), DATE(2008,5,15), 1000000, 1014420, 2)
}
```

Retorna a taxa de desconto de um título de dívida usando os termos especificados acima.

[VALUE]
0,05768

IPMT

17/03/2021 • 3 minutes to read

Retorna o pagamento de juros de um investimento em determinado período com base em pagamentos constantes e periódicos e uma taxa de juros constante.

Sintaxe

```
IPMT(<rate>, <per>, <nper>, <pv>[, <fv>[, <type>]])
```

Parâmetros

TERMO	DEFINIÇÃO
rate	A taxa de juros por período.
per	O período para o qual você deseja encontrar os juros. Precisa estar entre 1 e nper (inclusive).
nper	O número total de períodos de pagamento em uma anuidade.
pv	O valor atual ou o valor da quantia total de uma série de pagamentos futuros.
fv	(Opcional) O valor futuro ou um saldo de pagamento à vista que você deseja obter depois que o último pagamento for feito. Caso o fv seja omitido, ele será considerado BLANK.
tipo	(Opcional) O número 0 ou 1 que indica quando os pagamentos vencem. Caso type seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **type** aceitará os seguintes valores:

DEFINIR TYPE COMO IGUAL A	CASO OS PAGAMENTOS ESTEJAM VENCIDOS
0 ou omitido	No final do período
1	No início do período

Valor Retornado

O pagamento de juros para um determinado período.

Comentários

- Seja consistente em relação às unidades que você usa para especificar rate e nper. Caso faça pagamentos mensais em um empréstimo de quatro anos a uma taxa anual de juros de 12%, use 0,12/12 para obter

rate e 4*12 para obter nper. Caso faça pagamentos anuais no mesmo empréstimo, use 0,12 para rate e 4 para nper.

- Para todos os argumentos, o dinheiro pago, como depósitos em poupança, será representado por números negativos. Já o dinheiro recebido, como cheques de dividendos, será representado por números positivos.
- type será arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - per < 1 ou per > nper
  - nper < 1
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplos

DADOS	DESCRIÇÃO
10,00%	Juros anual
3	Anos do empréstimo
\US$ 8.000	Valor atual do empréstimo

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE
{
  IPMT(0.1/12, 1, 3*12, 8000)
}
```

Retorna os juros mensais devidos no primeiro mês de um empréstimo com os termos especificados acima.

[VALUE]
-66,6666666666667

Exemplo 2

A seguinte consulta DAX:

```
EVALUATE
{
  IPMT(0.1, 3, 3, 8000)
}
```

Retorna os juros anuais devidos no último ano de um empréstimo com os termos especificados acima, em que os pagamentos são feitos anualmente.

[VALUE]
-292,447129909366

[VALUE]

ISPMT

17/03/2021 • 4 minutes to read

Calcula os juros pagos (ou recebidos) para o período especificado de um empréstimo (ou investimento) com pagamentos iguais do valor principal.

Sintaxe

```
ISPMT(<rate>, <per>, <nper>, <pv>)
```

Parâmetros

TERMO	DEFINIÇÃO
rate	A taxa de juros do investimento.
per	O período para o qual você deseja encontrar os juros. Deve estar entre 0 e nper-1 (incluído).
nper	O número total de períodos de pagamento do investimento.
pv	O valor atual do investimento. Para um empréstimo, pv será o valor do empréstimo.

Valor Retornado

Os juros pagos (ou recebidos) pelo período especificado.

Comentários

- Seja consistente em relação às unidades que você usa para especificar rate e nper. Caso faça pagamentos mensais em um empréstimo de quatro anos a uma taxa de juros anual de 12%, use 0,12/12 para obter rate e 4*12 para obter nper. Caso faça pagamentos anuais no mesmo empréstimo, use 0,12 para rate e 4 para nper.
- Para todos os argumentos, o dinheiro pago, como depósitos em poupança ou outros saques, será representado por números negativos. Já o dinheiro recebido, como cheques de dividendos e outros depósitos, será representado por números positivos.
- O ISPMT conta cada período que começa com zero, não um.
- A maioria dos empréstimos usa uma agenda de amortização com pagamentos periódicos uniformes. A função IPMT retorna o pagamento de juros de um determinado período para esse tipo de empréstimo.
- Alguns empréstimos usam uma agenda de amortização com pagamentos uniformes do valor principal. A função ISPMT retorna o pagamento de juros de um determinado período para esse tipo de empréstimo.
- Um erro será retornado se:
  - nper = 0.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança

em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
\US$ 4.000	Valor atual
4	Número de períodos
10%	Tarifa

Para ilustrar quando usar o ISPMT, a tabela de amortização abaixo usará uma agenda de amortização uniforme do valor principal com os termos especificados acima. A cobrança de juros de cada período será igual à taxa multiplicada do saldo não pago do período anterior. E o pagamento de cada período será igual ao valor principal uniforme, além dos juros do período.

PERÍODO	PAGAMENTO DO VALOR PRINCIPAL	PAGAMENTO DE JUROS	PAGAMENTO TOTAL	SALDO
				4.000,00
1	1.000,00	400,00	1.400,00	3.000,00
2	1.000,00	300,00	1.300,00	2.000,00
3	1.000,00	200,00	1.200,00	1.000,00
4	1.000,00	100,00	1.100,00	0,00

A seguinte consulta DAX:

```
DEFINE
VAR NumPaymentPeriods = 4
VAR PaymentPeriods = GENERATESERIES(0, NumPaymentPeriods-1)
EVALUATE
ADDCOLUMNS (
  PaymentPeriods,
  "Interest Payment",
  ISPMT(0.1, [Value], NumPaymentPeriods, 4000)
)
```

Retorna os juros pagos durante cada período usando uma agenda de amortização uniforme do valor principal e os termos especificados acima. Os valores serão negativos para indicar o pagamento de juros e não os juros recebidos.

[VALUE]	[INTEREST PAYMENT]
0	-400
1	-300
2	-200

[VALUE]	[INTEREST PAYMENT]
3	-100

MDURATION

17/03/2021 • 4 minutes to read

Retorna a duração de Macaulay modificada para um título com um valor nominal presumido de \US$ 100.

Sintaxe

```
MDURATION(<settlement>, <maturity>, <coupon>, <yld>, <frequency>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
cupom	A taxa de cupom anual do título.
yld	O rendimento anual do título.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

A duração de Macaulay modificada.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento será 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- A duração modificada será definida da seguinte maneira:

  $$\text{MDURATION} = \frac{\text{DURATION}}{1 + (\frac{\text{Market yield}}{\text{Coupon payments per year}})}$$
- settlement e maturity serão arredondados para números inteiros.
- frequency e basis serão arredondadas para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - cupom < 0.
  - yld < 0
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
01/01/2008	Data de liquidação
01/01/2016	Data de vencimento
8%	Cupom percentual
9%	Percentual de rendimento
2	A frequência é semestral (confira acima)
1	Real/base real (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  MDURATION(DATE(2008,1,1), DATE(2016,1,1), 0.08, 0.09, 2, 1)
}
```

Retorna a duração de Macaulay modificada de um título de dívida usando os termos especificados acima.

[VALUE]
5,73566981391884

NOMINAL

17/03/2021 • 2 minutes to read

Retorna a taxa de juros anual nominal, considerando a taxa efetiva e o número de períodos compostos por ano.

Sintaxe

```
NOMINAL(<effect_rate>, <npery>)
```

Parâmetros

TERMO	DEFINIÇÃO
effect_rate	A taxa de juros efetiva.
npery	O número de períodos compostos por ano.

Valor Retornado

A taxa de juros anual nominal.

Comentários

- A relação entre NOMINAL e EFFECT será mostrada na seguinte equação:

  $$\text{EFFECT} = \Big(1 + \frac{\text{nominal_rate}}{\text{npery}} \Big)^{\text{npery}} - 1$$

- npery será arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - effect_rate ≤ 0.
  - npery < 1.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
5,3543%	Taxa de juros efetiva
4	Número de períodos compostos por ano

A seguinte consulta DAX:

```
EVALUATE
{
  NOMINAL(0.053543, 4)
}
```

Retorna a taxa de juros nominal usando os termos especificados acima.

[VALUE]
0,052500319868356

NPER

17/03/2021 • 2 minutes to read

Retorna o número de períodos de um investimento com base em pagamentos periódicos e constantes e uma taxa de juros constante.

Sintaxe

```
NPER(<rate>, <pmt>, <pv>[, <fv>[, <type>]])
```

Parâmetros

TERMO	DEFINIÇÃO
rate	A taxa de juros por período.
pmt	O pagamento feito em cada período. Ele não poderá ser alterado durante a vida da anuidade. O pmt geralmente contém o valor principal e os juros, mas nenhuma outra taxa nem imposto.
pv	O valor atual ou o valor da quantia total de uma série de pagamentos futuros.
fv	(Opcional) O valor futuro ou um saldo de pagamento à vista que você deseja obter depois que o último pagamento for feito. Caso o fv seja omitido, ele será considerado BLANK.
tipo	(Opcional) O número 0 ou 1 e indica quando os pagamentos estão vencidos. Caso type seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **type** aceitará os seguintes valores:

DEFINIR TYPE COMO IGUAL A	CASO OS PAGAMENTOS ESTEJAM VENCIDOS
0 ou omitido	No final do período
1	No início do período

Valor Retornado

O número de períodos de um investimento.

Comentários

- type será arredondado para o número inteiro mais próximo.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
12%	Taxa de juros anual
-100	Pagamento feito a cada período
-1000	Valor atual
10000	Valor futuro
1	O pagamento será devido no início do período (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  NPER(0.12/12, -100, -1000, 10000, 1)
}
```

Retorna o número de períodos do investimento descrito conforme os termos especificados acima.

[VALUE]
59,6738656742946

ODDFPRICE

17/03/2021 • 8 minutes to read

Retorna o preço por \US$ 100 de valor nominal de um título com um período inicial (curto ou longo) indefinido.

Sintaxe

```
ODDFPRICE(<settlement>, <maturity>, <issue>, <first_coupon>, <rate>, <yld>, <redemption>, <frequency>[,
<basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
problema	A data de emissão do título.
first_coupon	A data inicial do cupom do título.
rate	A taxa de juros do título.
yld	O rendimento anual do título.
redemption	O valor de resgate do título por \US$ 100 de valor nominal.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360

BASE	BASE DE CONTAGEM DIÁRIA
3	Real/365
4	Europeu 30/360

Valor Retornado

O preço por \US$ 100 de valor nominal.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- A função ODDFPRICE será calculada da seguinte maneira:

  Primeiro cupom de um período curto indefinido:

  $$\text{ODDFPRICE} = \bigg[\frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(N - 1 + \frac{\text{DSC}}{\text{E}})}} \bigg] + \bigg[\frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{DFC}}{\text{E}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(\frac{\text{DSC}}{\text{E}})}} \bigg] + \bigg[\sum^{N}_{k=2} \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(k - 1 + \frac{\text{DSC}}{\text{E}})}} \bigg] - \Big[100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{A}}{\text{E}} \Big] $$

  em que:

  - A = número de dias do início do período de um cupom à data de liquidação (dias acumulados).
  - DSC = número de dias da liquidação à data do próximo cupom.
  - DFC = número de dias do início do primeiro cupom indefinido à data inicial do cupom.
  - E = número de dias no período de um cupom.
  - N = número de cupons a serem pagos entre a data de liquidação e a data de resgate. (Caso esse número seja uma fração, ele será elevado para o próximo número inteiro.)

  Primeiro cupom de um período longo indefinido:

  $$\text{ODDFPRICE} = \bigg[\frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(\text{N} + \text{N}_{q} + \frac{\text{DSC}}{\text{E}})}} \bigg] + \bigg[\frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \Big[\sum^{\text{NC}}_{i=1} \frac{\text{DC}_{i}}{\text{NL}_{i}} \Big] }{(1 + \frac{\text{yld}}{\text{frequency}})^{(\text{N}_{q} + \frac{\text{DSC}}{\text{E}})}} \bigg] + \bigg[\sum^{\text{N}}_{k=1} \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(k - \text{N}_{q} + \frac{\text{DSC}}{\text{E}})}} \bigg] - \Big[100 \times \frac{\text{rate}}{\text{frequency}} \times \sum^{\text{NC}}_{i=1} \frac{\text{A}_{i}}{\text{NL}_{i}} \Big]$$

  em que:

- A_{i} = número de dias do início de i^{th} ou o período final de quase-cupom dentro de um período indefinido.
- DC_{i} = número de dias desde a data de início (ou data de emissão) até o primeiro quase-cupom ($i = 1$) ou o número de dias em um quase-cupom ($i = 2$,..., $i = \text{NC}$).
- DSC = número de dias da liquidação até a data do próximo cupom.
- E = número de dias no período de um cupom.
- N = número de cupons a serem pagos entre a data inicial e real do cupom e a data de resgate. (Caso esse número seja uma fração, ele será elevado para o próximo número inteiro.)
- NC = número de períodos de um cupom parcial que se ajustam no período indefinido. (Caso esse número seja uma fração, ele será elevado para o próximo número inteiro.)
- NL_{i} = duração normal em dias do i^{th} total ou o período final de quase-cupom dentro de um período indefinido.
- N_{q} = número de períodos inteiros de quase-cupom entre a data de liquidação e o primeiro cupom.

- settlement, maturity, issue e first_coupon serão arredondados para obter números inteiros.
- basis e frequency serão arredondados para o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement, maturity, issue ou first_coupon não for válido.
  - maturity > first_coupon > settlement > issue não é aceitável.
  - rate < 0.
  - yld < 0.
  - redemption ≤ 0.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO DE ARGUMENTO
11/11/2008	Data de liquidação
01/03/2021	Data de vencimento
15/10/2008	Data de emissão
01/03/2009	Data inicial do cupom
7,85%	Cupom percentual
6,25%	Percentual de rendimento
\US$ 100,00	Valor de quitação
2	A frequência é semestral
1	Real/base real

A seguinte consulta DAX:

```
EVALUATE
{
  ODDFPRICE(DATE(2008,11,11), DATE(2021,3,1), DATE(2008,10,15), DATE(2009,3,1), 0.0785, 0.0625, 100.00, 2,
1)
}
```

Retorna o preço por \US$ 100 de valor nominal de um título com um período inicial (curto ou longo) indefinido usando os termos especificados acima.

[VALUE]
113,597717474079

ODDFYIELD

17/03/2021 • 5 minutes to read

Retorna o rendimento de um título que tem um período inicial (curto ou longo) indefinido.

Sintaxe

```
ODDFYIELD(<settlement>, <maturity>, <issue>, <first_coupon>, <rate>, <pr>, <redemption>, <frequency>[,
<basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
problema	A data de emissão do título.
first_coupon	A data inicial do cupom do título.
rate	A taxa de juros do título.
pr	O preço do título.
redemption	O valor de resgate do título por \US$ 100 de valor nominal.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360

BASE	BASE DE CONTAGEM DIÁRIA
3	Real/365
4	Europeu 30/360

Valor Retornado

O rendimento do título.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- A função ODDFYIELD será calculada usando um método iterativo. Ela usará o método de Newton com base na fórmula usada para a função ODDFPRICE. O rendimento será alterado por meio de iterações de 100 até que o preço estimado de um determinado rendimento esteja próximo do preço. Confira ODDFPRICE para obter a fórmula que ODDFYIELD usa.
- settlement, maturity, issue e first_coupon serão arredondados para obter números inteiros.
- basis e frequency serão arredondados para o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement, maturity, issue ou first_coupon não for válido.
  - maturity > first_coupon > settlement > issue não é aceitável.
  - rate < 0.
  - pr ≤ 0.
  - redemption ≤ 0.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO DE ARGUMENTO
11 de novembro de 2008	Data de liquidação
1º de março de 2021	Data de vencimento
15 de outubro de 2008	Data de emissão

DADOS	DESCRIÇÃO DE ARGUMENTO
1º de março de 2009	Data inicial do cupom
5,75%	Cupom percentual
84,50	Preço
100	Valor de quitação
2	A frequência é semestral
0	Base de 30/360

A seguinte consulta DAX:

```
EVALUATE
{
  ODDFYIELD(DATE(2008,11,11), DATE(2021,3,1), DATE(2008,10,15), DATE(2009,3,1), 0.0575, 84.50, 100, 2, 0)
}
```

Retorna o rendimento de um título que tem um período inicial (curto ou longo) indefinido usando os termos especificados acima.

[VALUE]
0,0772455415972989

ODDLPRICE

17/03/2021 • 4 minutes to read

Retorna o preço por \US$ 100 de valor nominal de um título com um período final de cupom (curto ou longo) indefinido.

Sintaxe

```
ODDLPRICE(<settlement>, <maturity>, <last_interest>, <rate>, <yld>, <redemption>, <frequency>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
last_interest	A data final do cupom do título.
rate	A taxa de juros do título.
yld	O rendimento anual do título.
redemption	O valor de resgate do título por \US$ 100 de valor nominal.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365

BASE	BASE DE CONTAGEM DIÁRIA
4	Europeu 30/360

Valor Retornado

O preço por \US$ 100 de valor nominal.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- a liquidação, o vencimento e last_interest serão arredondados para obter números inteiros.
- basis e frequency serão arredondados para o número inteiro mais próximo.
- Um erro será retornado se:
  - a data de liquidação, vencimento ou de last_interest não for válida.
  - maturity > settlement > last_interest não é aceitável.
  - rate < 0.
  - yld < 0.
  - redemption ≤ 0.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A seguinte consulta DAX:

DADOS	DESCRIÇÃO DE ARGUMENTO
07 de fevereiro de 2008	Data de liquidação
15 de junho de 2008	Data de vencimento
15 de outubro de 2007	Data final dos juros
3,75%	Cupom percentual
4,05%	Percentual de rendimento

DADOS	DESCRIÇÃO DE ARGUMENTO
\US$ 100	Valor de quitação
2	A frequência é semestral
0	Base de 30/360

```
EVALUATE
{
  ODDLPRICE(DATE(2008,2,7), DATE(2008,6,15), DATE(2007,10,15), 0.0375, 0.0405, 100, 2, 0)
}
```

Retorna o preço por \US$ 100 de valor nominal de um título que tem um período final de cupom (curto ou longo) indefinido usando os termos especificados acima.

[VALUE]
99,8782860147213

ODDLYIELD

17/03/2021 • 6 minutes to read

Retorna o rendimento de um título que tem um período final (curto ou longo) indefinido.

Sintaxe

```
ODDLYIELD(<settlement>, <maturity>, <last_interest>, <rate>, <pr>, <redemption>, <frequency>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
last_interest	A data final do cupom do título.
rate	A taxa de juros do título.
pr	O preço do título.
redemption	O valor de resgate do título por \US$ 100 de valor nominal.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O rendimento do título.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- ODDLYIELD será calculado da seguinte maneira:

  $$\text{ODDLYIELD} = \bigg[\frac{(\text{redemption} + ((\sum^{\text{NC}}_{i=1} \frac{\text{DC}_{i}}{\text{NL}_{i}}) \times \frac{100 \times \text{rate}}{\text{frequency}})) - (\text{par} + ((\sum^{\text{NC}}_{i=1} \frac{\text{A}_{i}}{\text{NL}_{i}}) \times \frac{100 \times \text{rate}}{\text{frequency}}))}{\text{par} + ((\sum^{\text{NC}}_{i=1} \frac{\text{A}_{i}}{\text{NL}_{i}}) \times \frac{100 \times \text{rate}}{\text{frequency}}))} \bigg] \times \bigg[\frac{\text{frequency}}{(\sum^{\text{NC}}_{i=1} \frac{\text{DSC}_{i}}{\text{NL}_{i}})} \bigg]$$

  em que:

  - A_{i} = número de dias acumulados para i^{th} ou o período final de quase-cupom dentro de um período indefinido contado a partir da última data de juros antes do resgate.
  - DC_{i} = número de dias contados no i^{th} ou o período final de quase-cupom conforme delimitado pela duração do período real do cupom.
  - NC = número de períodos de quase-cupons que se ajustam a um período indefinido. Caso esse número seja uma fração, ele será elevado para o próximo número inteiro.
  - NL_{i} = duração normal em dias de i^{th} ou o período final de quase-cupom dentro de um período indefinido.
- a liquidação, o vencimento e last_interest serão arredondados para obter números inteiros.
- basis e frequency serão arredondados para o número inteiro mais próximo.
- Um erro será retornado se:
  - a data de liquidação, vencimento ou de last_interest não for válida.
  - maturity > settlement > last_interest não é aceitável.
  - rate < 0.
  - pr ≤ 0.
  - redemption ≤ 0.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A seguinte consulta DAX:

DADOS	DESCRIÇÃO DE ARGUMENTO
20/04/2008	Data de liquidação
15/06/2008	Data de vencimento
24/12/2007	Data final dos juros
3,75%	Cupom percentual
\US$ 99,875	Preço
\US$ 100	Valor de resgate
2	A frequência é semestral
0	Base de 30/360

```
EVALUATE
{
  ODDLYIELD(DATE(2008,4,20), DATE(2008,6,15), DATE(2007,12,24), 0.0375, 99.875, 100, 2, 0)
}
```

Retorna o rendimento de um título que tem um período final (curto ou longo) indefinido usando os termos especificados acima.

[VALUE]
0,0451922356291692

PDURATION

17/03/2021 • 2 minutes to read

Retorna o número de períodos exigidos por um investimento para alcançar um valor especificado.

Sintaxe

```
PDURATION(<rate>, <pv>, <fv>)
```

Parâmetros

TERMO	DEFINIÇÃO
rate	A taxa de juros por período.
pv	O valor atual do investimento.
fv	O valor futuro desejado de um investimento.

Valor Retornado

O número de períodos.

Comentários

- O PDURATION usa a seguinte equação:

  $$\text{PDURATION} = \frac{log(\text{fv}) - log(\text{pv})}{log(1 + \text{rate})}$$

- Um erro será retornado se:
  - rate ≤ 0.
  - pv ≤ 0.
  - fv ≤ 0.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE
{
  PDURATION(0.025, 2000, 2200)
}
```

Retorna o número de anos necessários para um investimento de \US$ 2.000, ganhando 2,5% ao ano, para alcançar \US$ 2.200.

[VALUE]
3.85986616262266

Exemplo 2

A seguinte consulta DAX:

```
EVALUATE
{
  PDURATION(0.025/12, 1000, 1200)
}
```

Retorna o número de meses necessários para um investimento de \US$ 1.000, ganhando 2,5% ao ano, para alcançar \US$ 1.200.

[VALUE]
87.6054764193714

PMT

17/03/2021 • 4 minutes to read

Calcula o pagamento de um empréstimo com base em pagamentos e uma taxa de juros constantes.

Sintaxe

```
PMT(<rate>, <nper>, <pv>[, <fv>[, <type>]])
```

Parâmetros

TERMO	DEFINIÇÃO
rate	A taxa de juros do empréstimo.
nper	O número total de pagamentos do empréstimo.
pv	O valor atual ou o valor total que uma série de pagamentos futuros tem no momento, também chamado de principal.
fv	(Opcional) O valor futuro ou um saldo de pagamento à vista que você deseja obter depois que o último pagamento for feito. Caso o fv seja omitido, ele será considerado BLANK.
tipo	(Opcional) O número 0 ou 1 que indica quando os pagamentos vencem. Caso type seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **type** aceitará os seguintes valores:

DEFINIR TYPE COMO IGUAL A	CASO OS PAGAMENTOS ESTEJAM VENCIDOS
0 ou omitido	No final do período
1	No início do período

Observação: Para obter uma descrição mais completa dos argumentos no PMT, confira a função PV.

Valor Retornado

O valor de um pagamento único do empréstimo.

Comentários

- O pagamento retornado por PMT inclui o valor principal e os juros, mas não impostos, pagamentos de reserva ou valores às vezes associados a empréstimos.
- Seja consistente em relação às unidades que você usa para especificar rate e nper. Caso faça pagamentos mensais em um empréstimo de quatro anos a uma taxa de juros anual de 12%, use 0,12/12 para obter rate e 4*12 para obter nper. Caso faça pagamentos anuais no mesmo empréstimo, use 0,12 para rate e 4

para nper.

- type será arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - nper < 1

Dica: Para localizar o valor total pago durante o empréstimo, multiplique o valor do PMT retornado por nper.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplos

Exemplo 1

DADOS	DESCRIÇÃO
8%	Taxa de juros anual
10	Número de meses de pagamentos
\US$ 10.000	Valor do empréstimo

A seguinte consulta DAX:

```
EVALUATE
{
  PMT(0.08/12, 10, 10000, 0, 1)
}
```

Retorna o valor do pagamento mensal, pago no início do mês, para um empréstimo com os termos especificados acima.

[VALUE]
-1030.16432717797

Observação: 1030,16432717797 é o pagamento por período. Como resultado, o valor total pago durante o empréstimo é de aproximadamente 1.030,16 * 10 = \US$ 10.301,60. Em outras palavras, aproximadamente \US$ 301,60 de juros é pago.

Exemplo 2

DADOS	DESCRIÇÃO
6%	Taxa de juros anual
18	Número de anos do pagamento
\US$ 50.000	Valor do empréstimo

A seguinte consulta DAX:

```
EVALUATE
{
  PMT(0.06/12, 18*12, 0, 50000)
}
```

[VALUE]
-129.081160867991

Retorna o valor a ser economizado a cada mês para ter \US$ 50.000 no final de 18 anos, usando os termos especificados acima.

PPMT

17/03/2021 • 3 minutes to read

Retorna o pagamento do valor principal de um investimento em determinado período com base em pagamentos constantes e periódicos e uma taxa de juros constante.

Sintaxe

```
PPMT(<rate>, <per>, <nper>, <pv>[, <fv>[, <type>]])
```

Parâmetros

TERMO	DEFINIÇÃO
rate	A taxa de juros do empréstimo.
per	Especifica o período. Precisa estar entre 1 e nper (inclusive).
nper	O número total de períodos de pagamento em uma anuidade.
pv	O valor atual, ou seja, o valor total que uma série de pagamentos futuros tem no momento.
fv	(Opcional) O valor futuro ou um saldo de pagamento à vista que você deseja obter depois que o último pagamento for feito. Caso o fv seja omitido, ele será considerado BLANK.
tipo	(Opcional) O número 0 ou 1 que indica quando os pagamentos vencem. Caso type seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **type** aceitará os seguintes valores:

DEFINIR TYPE COMO IGUAL A	CASO OS PAGAMENTOS ESTEJAM VENCIDOS
0 ou omitido	No final do período
1	No início do período

Observação: Para obter uma descrição mais completa dos argumentos no PPMT, confira a função PV.

Valor Retornado

O pagamento sobre o valor principal de um determinado período.

Comentários

- Seja consistente em relação às unidades que você usa para especificar rate e nper. Caso faça pagamentos

mensais em um empréstimo de quatro anos a uma taxa de juros anual de 12%, use 0,12/12 para obter rate e 4*12 para obter nper. Caso faça pagamentos anuais no mesmo empréstimo, use 0,12 para rate e 4 para nper.

- type será arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - per < 1 ou per > nper
  - nper < 1
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo 1

DADOS	DESCRIÇÃO DE ARGUMENTO
10%	Taxa de juros anual
2	Número de anos do empréstimo
\US$ 2.000,00	Valor do empréstimo

A seguinte consulta DAX:

```
EVALUATE
{
  PPMT(0.1/12, 1, 2*12, 2000.00)
}
```

Retorna o pagamento do valor principal feito no primeiro mês de um empréstimo com os termos especificados acima.

[VALUE]
-75.6231860083663

Exemplo 2

DADOS	DESCRIÇÃO DE ARGUMENTO
8%	Taxa de juros anual
10	Número de anos do empréstimo
\US$ 200.000,00	Valor do empréstimo

A seguinte consulta DAX:

```
EVALUATE
{
  PPMT(0.08, 10, 10, 200000.00)
}
```

Retorna o pagamento do valor principal feito no 10º ano de um empréstimo com os termos especificados acima.

[VALUE]
-27598.0534624214

PRICE

17/03/2021 • 6 minutes to read

Retorna o preço por \US$ 100 de valor nominal de um título que paga juros periódicos.

Sintaxe

```
PRICE(<settlement>, <maturity>, <rate>, <yld>, <redemption>, <frequency>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
rate	A taxa de cupom anual do título.
yld	O rendimento anual do título.
redemption	O valor de resgate do título por \US$ 100 de valor nominal.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O preço por \US$ 100 de valor nominal.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- settlement e maturity serão arredondados para números inteiros.
- basis e frequency serão arredondados para o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - rate < 0.
  - yld < 0.
  - redemption ≤ 0.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Importante:

- Quando N > 1 (N é o número de cupons a pagar entre a data de liquidação e a data do resgate), **PRICE** é calculado da seguinte maneira:

  $$\text{PRICE} = \bigg[\frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(N - 1 + \frac{\text{DSC}}{\text{E}})}} \bigg] + \bigg[\sum^{N}_{k=1} \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(k - 1 + \frac{\text{DSC}}{\text{E}})}} \bigg] - \bigg[100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{A}}{\text{E}} \bigg]$$

- Quando N = 1 (N é o número de cupons a pagar entre a data de liquidação e a data do resgate), **PRICE** é calculado da seguinte maneira:

  $$\text{DSR} = \text{E} - \text{A}$$

  $$\text{T1} = 100 \times \frac{\text{rate}}{\text{frequency}} + \text{redemption}$$

  $$\text{T2} = \frac{\text{yld}}{\text{frequency}} \times \frac{\text{DSR}}{\text{E}} + 1$$

  $$\text{T3} = 100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{A}}{\text{E}}$$

  $$\text{PRICE} = \frac{\text{T1}}{\text{T2}} - \text{T3}$$

  em que:

- DSC = número de dias da liquidação até a data do próximo cupom.
- E = número de dias no período de cupom que coincidem com a data de liquidação.
- A = número de dias desde o início do período de cupom até a data de liquidação.

Exemplo

DADOS	DESCRIÇÃO DE ARGUMENTO
15/02/2008	Data de liquidação
15/11/2017	Data de vencimento
5,75%	Percentual do cupom semestral
6,50%	Percentual de rendimento
\US$ 100	Valor de resgate
2	A frequência é semestral
0	Base de 30/360

A seguinte consulta DAX:

```
EVALUATE
{
  PRICE(DATE(2008,2,15), DATE(2017,11,15), 0.0575, 0.065, 100, 2, 0)
}
```

Retorna a preço de um título de dívida que usa os termos especificados acima.

[VALUE]
94.6343616213221

PRICEDISC

17/03/2021 • 4 minutes to read

Retorna o preço por \US$ 100 de valor nominal de um título com desconto.

Sintaxe

```
PRICEDISC(<settlement>, <maturity>, <discount>, <redemption>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
discount	A taxa de desconto do título.
redemption	O valor de resgate do título por \US$ 100 de valor nominal.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O preço por \US$ 100 de valor nominal.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são

39.448 dias após 30 de dezembro de 1899.

- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2018 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2018, a data de liquidação seria 1º de julho de 2018 e a data de vencimento seria 1º de janeiro de 2048, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2018.
- PRICEDISC será calculado da seguinte maneira:

  $$\text{PRICEDISC} = \text{redemption} - \text{discount} \times \text{redemption} \times \frac{\text{DSM}}{\text{B}}$$

  em que:

  - B = número de dias no ano, dependendo da base anual.
  - DSM = número de dias da liquidação ao vencimento.
- settlement e maturity serão arredondados para números inteiros.
- basis é arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - discount ≤ 0.
  - redemption ≤ 0.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO DE ARGUMENTO
16/2/2008	Data de liquidação
1/3/2008	Data de vencimento
5,25%	Porcentagem da taxa de desconto
\US$ 100	Valor de resgate
2	Base real/360

A seguinte consulta DAX:

```
EVALUATE
{
  PRICEDISC(DATE(2008,2,16), DATE(2008,3,1), 0.0525, 100, 2)
}
```

Retorna o preço do título por \US$ 100 de valor nominal de um título de dívida com os termos especificados acima.

[VALUE]
99.7958333333333

PRICEMAT

17/03/2021 • 4 minutes to read

Retorna o preço por \US$ 100 de valor nominal de um título que paga juros no vencimento.

Sintaxe

```
PRICEMAT(<settlement>, <maturity>, <issue>, <rate>, <yld>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
problema	A data de emissão do título.
rate	A taxa de juros do título na data de emissão.
yld	O rendimento anual do título.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O preço por \US$ 100 de valor nominal.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- PRICEMAT será calculado da seguinte maneira:

  $$\text{PRICEMAT} = \frac{100 + (\frac{\text{DIM}}{\text{B}} \times \text{rate} \times 100)}{1 + (\frac{\text{DSM}}{\text{B}} \times \text{yld})} - (\frac{\text{A}}{\text{B}} \times \text{rate} \times 100)$$

  em que:

  - B = número de dias no ano, dependendo da base anual.
  - DSM = número de dias da liquidação ao vencimento.
  - DIM = número de dias da emissão ao vencimento.
  - A = número de dias da emissão à liquidação.
- settlement, maturity e issue são arredondados para números inteiros.
- basis é arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement, maturity ou issue não for uma data válida.
  - maturity > settlement > issue não for atendido.
  - rate < 0.
  - yld < 0.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A seguinte consulta DAX:

DADOS	DESCRIÇÃO
15/02/2008	Data de liquidação
13/4/2008	Data de vencimento
11/11/2007	Data de emissão
6,10%	Percentual do cupom semestral
6,10%	Percentual de rendimento
0	Base de 30/360

```
EVALUATE
{
  PRICEMAT(DATE(2008,2,15), DATE(2008,4,13), DATE(2007,11,11), 0.061, 0.061, 0)
}
```

Retorna o preço por \US$ 100 de valor nominal de um título com os termos especificados acima.

[VALUE]
99.9844988755569

PV

17/03/2021 • 6 minutes to read

Calcula o valor atual de um empréstimo ou de um investimento com base em uma taxa de juros constante. Você pode usar o PV com pagamentos periódicos e constantes (como uma hipoteca ou outro empréstimo) e/ou com um valor futuro que seja a sua meta de investimento.

Sintaxe

```
PV(<rate>, <nper>, <pmt>[, <fv>[, <type>]])
```

Parâmetros

TERMO	DEFINIÇÃO
rate	A taxa de juros por período. Por exemplo, se você obtiver um empréstimo para comprar um automóvel a uma taxa de juros anual de 10% e fizer pagamentos mensais, a sua taxa de juros por mês será 0,1/12 ou 0,0083. Você precisaria inserir 0,1/12 ou 0,0083 na fórmula como a taxa.
nper	O número total de períodos de pagamento em uma anuidade. Por exemplo, se você obtiver um empréstimo de quatro anos para comprar um carro e fizer pagamentos mensais, o seu empréstimo terá 4*12 (ou 48) parcelas. Você precisaria inserir 48 na fórmula como nper.
pmt	O pagamento feito em cada período que não poderá ser alterado durante o tempo de vida da anuidade. O pmt geralmente inclui o valor principal e os juros, mas nenhuma outra taxa nem imposto. Por exemplo, os pagamentos mensais em um \empréstimo de quatro anos de US$ 10.000 para compra de um carro a 12% são \US$ 263,33. Você precisaria inserir -263,33 na fórmula como o pmt.
fv	(Opcional) O valor futuro ou um saldo de pagamento à vista que você deseja obter depois que o último pagamento for feito. Caso o fv seja omitido, ele será considerado BLANK. Por exemplo, se você quiser economizar \US$ 50.000 para pagar um projeto especial em 18 anos, \US$ 50.000 será o valor futuro. Em seguida, você pode fazer uma estimativa conservadora a uma taxa de juros e determinar quanto você precisa economizar cada mês.
tipo	(Opcional) O número 0 ou 1 que indica quando os pagamentos vencem. Caso type seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **type** aceitará os seguintes valores:

DEFINIR TYPE COMO IGUAL A	CASO OS PAGAMENTOS ESTEJAM VENCIDOS
0 ou omitido	No final do período
1	No início do período

Valor Retornado

O valor atual de um empréstimo ou investimento.

Comentários

- Seja consistente em relação às unidades que você usa para especificar rate e nper. Caso faça pagamentos mensais em um empréstimo de quatro anos a uma taxa anual de juros de 12%, use 0,12/12 para obter rate e 4*12 para obter nper. Caso faça pagamentos anuais no mesmo empréstimo, use 0,12 para rate e 4 para nper.
- As seguintes funções se aplicam a anuidades:
  - CUMIPMT
  - CUMPRINC
  - FV
  - IPMT
  - PMT
  - PPMT
  - PV
  - RATE
  - XIRR
  - XNPV
- Uma anuidade é uma série de pagamentos à vista constantes feitos em um período contínuo. Por exemplo, um empréstimo para compra de um carro ou uma hipoteca é uma anuidade. Para obter mais informações, confira a descrição de cada função da anuidade.
- Nas funções da anuidade, o dinheiro pago, como depósitos na poupança, será representado por números negativos. Já o dinheiro recebido, como cheques de dividendos, será representado por números positivos. Por exemplo, um depósito \US$ 1.000 para o banco será representado pelo argumento -1000 se você for o depositante e pelo argumento 1000 se você for o banco.
- Um argumento financeiro é resolvido em relação aos outros.
  - Se rate não for 0, então:

    $$\text{pv} \times (1 + \text{rate})^{\text{nper}} + \text{pmt}(1 + \text{rate} \times \text{type}) \times \bigg(\frac{(1 + \text{rate})^{\text{nper}} - 1}{\text{rate}} \bigg) + \text{fv} = 0$$
  - Se rate for 0, então:

    $$(\text{pmt} \times \text{nper}) + \text{pv} + \text{fv} = 0$$
- type será arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - nper < 1
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança

em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
\US$ 500,00	Valor em dinheiro pago para uma anuidade de seguros no final de cada mês.
8%	Taxa de juros ganha sobre o valor em dinheiro pago.
20	Quantidade de anos na qual o valor em dinheiro será pago.

A seguinte consulta DAX:

```
EVALUATE
{
  PV(0.08/12, 12*20, 500.00, 0, 0)
}
```

Retorna o valor presente de uma anuidade usando os termos especificados acima.

[VALUE]
-59777.1458511878

RATE

17/03/2021 • 4 minutes to read

Retorna a taxa de juros por período de uma anuidade. RATE é calculado por iteração e pode ter zero ou mais soluções. Se os resultados sucessivos de RATE não convergirem para 0,0000001 após 20 iterações, um erro será retornado.

Sintaxe

```
RATE(<nper>, <pmt>, <pv>[, <fv>[, <type>[, <guess>]]])
```

Parâmetros

TERMO	DEFINIÇÃO
nper	O número total de períodos de pagamento em uma anuidade.
pmt	O pagamento feito em cada período. Ele não poderá ser alterado durante o tempo de vida da anuidade. O pmt geralmente inclui o valor principal e os juros, mas nenhuma outra taxa nem imposto.
pv	O valor atual, ou seja, o valor total que uma série de pagamentos futuros tem no momento.
fv	(Opcional) O valor futuro ou um saldo de pagamento à vista que você deseja obter depois que o último pagamento for feito. Se o fv for omitido, ele será considerado 0 (o valor futuro de um empréstimo, por exemplo, é 0).
tipo	(Opcional) O número 0 ou 1 que indica quando os pagamentos vencem. Caso type seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.
guess	(Opcional) A sua estimativa de qual será a taxa. – Caso seja omitido, ele será considerado 10%. – Se RATE não convergir, tente valores diferentes para guess. O RATE geralmente convergirá se guess estiver entre 0 e 1.

O parâmetro **type** aceitará os seguintes valores:

DEFINIR TYPE COMO IGUAL A	CASO OS PAGAMENTOS ESTEJAM VENCIDOS
0 ou omitido	No final do período
1	No início do período

Valor Retornado

A taxa de juros por período.

Comentários

- Certifique-se de que as unidades que você usa para especificar guess e nper sejam consistentes. Caso faça pagamentos mensais em um empréstimo de quatro anos a uma taxa anual de 12%, use 0,12/12 para obter o guess e 4*12 para obter o nper. Caso faça pagamentos anuais no mesmo empréstimo, use 0,12 para obter o guess e 4 para obter o nper.
- type será arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - nper ≤ 0.
  - RATE não é convergido para 0,0000001 após 20 iterações
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplos

DADOS	DESCRIÇÃO
4	Anos do empréstimo
-200	Pagamento mensal
8000	Valor do empréstimo

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE
{
  RATE(4*12, -200, 8000)
}
```

Retorna a taxa mensal do empréstimo usando os termos especificados acima.

[VALUE]
0.00770147248820137

Exemplo 2

A seguinte consulta DAX:

```
EVALUATE
{
  RATE(4*12, -200, 8000) * 12
}
```

Retorna a taxa anual do empréstimo usando os termos especificados acima.

[VALUE]
0.0924176698584164

RECEIVED

17/03/2021 • 4 minutes to read

Retorna o valor recebido no vencimento de um título totalmente investido.

Sintaxe

```
RECEIVED(<settlement>, <maturity>, <investment>, <discount>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
investment	O valor investido no título.
discount	A taxa de desconto do título.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O valor recebido no vencimento.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são

39.448 dias após 30 de dezembro de 1899.

- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- RECEIVED será calculado da seguinte maneira:

  $$\text{RECEIVED} = \frac{\text{investment}}{1 - (\text{discount} \times \frac{\text{DIM}}{\text{B}})}$$

  em que:

  - B = número de dias em um ano, dependendo da base anual.
  - DIM = número de dias da emissão ao vencimento.
- settlement e maturity serão arredondados para números inteiros.
- basis é arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - investment ≤ 0.
  - discount ≤ 0.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A seguinte consulta DAX:

DADOS	DESCRIÇÃO
15-fev-08	Data de liquidação (emissão)
15-maio-08	Data de vencimento
\US$ 1.000.000,00	Investimento
5,75%	Porcentagem da taxa de desconto
2	Base real/360

```
EVALUATE
{
  RECEIVED(DATE(2008,2,15), DATE(2008,5,15), 1000000.00, 0.0575, 2)
}
```

Retorna o valor total a ser recebido no vencimento para um título de dívida com os termos especificados acima.

[VALUE]
1014584.6544071

RRI

17/03/2021 • 2 minutes to read

Retorna uma taxa de juros equivalente para o crescimento de um investimento.

Sintaxe

```
RRI(<nper>, <pv>, <fv>)
```

Parâmetros

TERMO	DEFINIÇÃO
nper	O número de períodos de um investimento.
pv	O valor atual do investimento.
fv	O valor futuro de um investimento.

Valor Retornado

A taxa de juros equivalente.

Comentários

- RRI retorna a taxa de juros quando fornecido nper (o número de períodos), pv (o valor atual) e fv (o valor futuro). Ela é calculada usando a seguinte equação:

  $$\bigg(\frac{\text{fv}}{\text{pv}} \bigg)^{(\frac{1}{\text{}nper})} - 1$$

- Um erro será retornado se:
  - nper ≤ 0.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
\US$ 10.000	Valor atual
\US$ 21.000	Valor futuro
4	Anos investidos

A seguinte consulta DAX:

```
EVALUATE
{
  RRI(4*12, 10000, 21000)
}
```

Retorna uma taxa de juros equivalente para o crescimento de um investimento com os termos especificados acima.

[VALUE]
0.0155771057566627

SLN

17/03/2021 • 2 minutes to read

Retorna a depreciação de linha reta de um ativo para um período.

Sintaxe

```
SLN(<cost>, <salvage>, <life>)
```

Parâmetros

TERMO	DEFINIÇÃO
cost	O custo inicial do ativo.
salvage	O valor no final da depreciação (às vezes chamado de valor residual do ativo).
vida	O número de períodos no qual o ativo está sendo depreciado (às vezes chamado de vida útil do ativo).

Valor Retornado

A depreciação de linha reta para um período.

Comentários

- Um erro será retornado se:
  life = 0.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
\US$ 30.000	Cost
\US$ 7.500	Valor residual
10	Anos de vida útil

A seguinte consulta DAX:

```
EVALUATE
{
  SLN(30000, 7500, 10)
}
```

Retorna a provisão para depreciação anual usando os termos especificados acima.

[VALUE]
2250

SYD

17/03/2021 • 2 minutes to read

Retorna uma depreciação dos dígitos da soma dos anos de um ativo para um período especificado.

Sintaxe

```
SYD(<cost>, <salvage>, <life>, <per>)
```

Parâmetros

TERMO	DEFINIÇÃO
cost	O custo inicial do ativo.
salvage	O valor no final da depreciação (às vezes chamado de valor residual do ativo).
vida	O número de períodos no qual o ativo está sendo depreciado (às vezes chamado de vida útil do ativo).
per	O período. Ele precisa usar as mesmas unidades que life. Precisa estar entre 1 e life (inclusive).

Valor Retornado

A depreciação da soma dos dígitos do ano para um período especificado.

Comentários

- O SYD será calculado da seguinte maneira:

  $$\text{SYD} = \frac{(\text{cost} - \text{salvage}) \times (\text{life} - \text{per} + 1) \times 2}{(\text{life}) \times (\text{life} + 1)}$$

- Um erro será retornado se:
  - life < 1.
  - per < 1 ou per > life.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplos

DADOS	DESCRIÇÃO
\US$ 30.000,00	Custo inicial
\US$ 7.500,00	Valor residual

DADOS	DESCRIÇÃO
10	Vida útil em anos

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE
{
  SYD(30000.00, 7500.00, 10, 1)
}
```

Retorna a provisão para depreciação da soma dos dígitos do ano do ativo do primeiro ano, considerando os termos especificados acima.

[VALUE]
4090.90909090909

Exemplo 2

A seguinte consulta DAX:

```
EVALUATE
{
  SYD(30000.00, 7500.00, 10, 10)
}
```

Retorna a provisão para depreciação da soma dos dígitos do ano de um ativo para o décimo ano (final), considerando os termos especificados acima.

[VALUE]
409.090909090909

TBILLEQ

17/03/2021 • 2 minutes to read

Retorna o rendimento equivalente do título de dívida de uma Letra do Tesouro.

Sintaxe

```
TBILLEQ(<settlement>, <maturity>, <discount>)
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação da Letra do Tesouro. A data de liquidação do título será a data posterior à data de emissão quando a Letra do Tesouro for negociada com o comprador.
maturity	A data de vencimento da Letra do Tesouro. A data de vencimento será a data de expiração da Letra do Tesouro.
discount	A taxa de desconto da Letra do Tesouro.

Valor Retornado

O rendimento equivalente ao título de dívida da Letra do Tesouro.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- TBILLEQ é calculado como:

  $$\text{TBILLEQ} = \frac{365 \times \text{discount}}{360 - (\text{discount} \times \text{DSM})}$$

  em que:

  - DSM é o número de dias entre a liquidação e o vencimento computados de acordo com a base de 360 dias por ano.
- settlement e maturity serão arredondados para números inteiros.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity; ou o vencimento ocorrer mais de um ano após a liquidação.
  - discount ≤ 0.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
31/03/2008	Data de liquidação
1/6/2008	Data de vencimento
9,14%	Porcentagem da taxa de desconto

A seguinte consulta DAX:

```
EVALUATE
{
  TBILLEQ(DATE(2008,3,31), DATE(2008,6,1), 0.0914)
}
```

Retorna o rendimento equivalente ao título de dívida de uma Letra do Tesouro usando os termos especificados acima.

[VALUE]
0.094151493565943

TBILLPRICE

17/03/2021 • 2 minutes to read

Retorna o preço por \US$ 100 de valor nominal de uma Letra do Tesouro.

Sintaxe

```
TBILLPRICE(<settlement>, <maturity>, <discount>)
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação da Letra do Tesouro. A data de liquidação do título será a data posterior à data de emissão quando a Letra do Tesouro for negociada com o comprador.
maturity	A data de vencimento da Letra do Tesouro. A data de vencimento será a data de expiração da Letra do Tesouro.
discount	A taxa de desconto da Letra do Tesouro.

Valor Retornado

O preço da Letra do Tesouro por \US$ 100 de valor nominal.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- A função TBILLPRICE será calculada da seguinte maneira:

  $$\text{TBILLPRICE} = 100 \times (1 - \frac{\text{discount} \times \text{DSM}}{360})$$

  em que:

  - DSM = número de dias desde a liquidação até o vencimento, excluindo qualquer data de vencimento de mais de um ano civil após a data de liquidação.
- settlement e maturity serão arredondados para números inteiros.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity; ou o vencimento ocorrer mais de um ano após a liquidação.
  - discount ≤ 0.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
31/03/2008	Data de liquidação
1/6/2008	Data de vencimento
9,0%	Porcentagem da taxa de desconto

A seguinte consulta DAX:

```
EVALUATE
{
  TBILLPRICE(DATE(2008,3,31), DATE(2008,6,1), 0.09)
}
```

Retorna o preço da Letra do Tesouro por \US$100 de valor nominal, considerando os termos especificados acima.

[VALUE]
98.45

TBILLYIELD

17/03/2021 • 2 minutes to read

Retorna o rendimento de uma Letra do Tesouro.

Sintaxe

```
TBILLYIELD(<settlement>, <maturity>, <pr>)
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação da Letra do Tesouro. A data de liquidação do título será a data posterior à data de emissão quando a Letra do Tesouro for negociada com o comprador.
maturity	A data de vencimento da Letra do Tesouro. A data de vencimento será a data de expiração da Letra do Tesouro.
pr	O preço da Letra do Tesouro por \US$ 100 de valor nominal.

Valor Retornado

O rendimento da Letra do Tesouro.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- TBILLYIELD será calculado da seguinte maneira:

  $$\text{TBILLYIELD} = \frac{100 - \text{pr}}{\text{pr}} \times \frac{360}{\text{DSM}}$$

  em que:

  - DSM = número de dias desde a liquidação até o vencimento, excluindo qualquer data de vencimento de mais de um ano civil após a data de liquidação.
- settlement e maturity serão arredondados para números inteiros.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity; ou o vencimento ocorrer mais de um ano após a liquidação.
  - pr ≤ 0.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A seguinte consulta DAX:

DADOS	DESCRIÇÃO
31/03/2008	Data de liquidação
1/6/2008	Data de vencimento
\US$ 98,45	Preço por \US$ 100 de valor nominal

```
EVALUATE
{
  TBILLYIELD(DATE(2008,3,31), DATE(2008,6,1), 98.45)
}
```

Retorna o rendimento de uma Letra do Tesouro usando os termos especificados acima.

[VALUE]
0.0914169629253426

VDB

17/03/2021 • 4 minutes to read

Retorna a depreciação de um ativo para qualquer período especificado, incluindo períodos parciais, usando o método de saldo decrescente duplo ou outro método especificado. VDB significa saldo decrescente variável.

Sintaxe

```
VDB(<cost>, <salvage>, <life>, <start_period>, <end_period>[, <factor>[, <no_switch>]])
```

Parâmetros

TERMO	DEFINIÇÃO
cost	O custo inicial do ativo.
salvage	O valor no final da depreciação (às vezes chamado de valor residual do ativo). Esse valor pode ser 0.
vida	O número de períodos sobre o qual o ativo está sendo depreciado (às vezes chamado de vida útil do ativo).
start_period	O período de início para o qual você deseja calcular a depreciação. Start_period precisa usar as mesmas unidades que life. Precisa estar entre 1 e life (inclusive).
end_period	O período final para o qual você deseja calcular a depreciação. End_period precisa usar as mesmas unidades que life. Precisa estar entre start_period e life (inclusive).
fator	(Opcional) A taxa na qual o saldo diminui. Se factor for omitido, ele será considerado 2 (o método de saldo decrescente duplo). Altere factor se você não quiser usar o método de saldo decrescente duplo. Para obter uma descrição do método de saldo decrescente duplo, confira DDB.
no_switch	(Opcional) Um valor lógico que especifica se o valor será alternado para a depreciação de linha reta quando a depreciação for maior do que o cálculo do saldo decrescente. Caso seja omitido, ele será considerado FALSE. – Se no_switch for avaliado como TRUE, o VDB não será alternado para a depreciação de linha reta, mesmo quando a depreciação for maior do que o cálculo do saldo decrescente. – Se no_switch for avaliado como FALSE ou for omitido, o VDB será alternado para a depreciação de linha reta quando a depreciação for maior do que o cálculo do saldo decrescente.

Valor Retornado

A depreciação durante o período especificado.

Comentários

- Um erro será retornado se:
  - cost < 0.
  - salvage < 0.
  - life < 1.
  - start_period < 1 ou start_period > end_period.
  - end_period < start_period ou end_period > life.
  - factor < 0.
  - no_switch não for avaliado como TRUE ou FALSE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplos

DADOS	DESCRIÇÃO
2400	Custo inicial
300	Valor residual
10	Tempo de vida em anos

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE
{
  VDB(2400, 300, 10*365, 0, 1)
}
```

Retorna a depreciação do primeiro dia de um ativo usando um fator de 2.

[VALUE]
1.31506849315068

Exemplo 2

A seguinte consulta DAX:

```
EVALUATE
{
  VDB(2400, 300, 10*12, 6, 18, 3)
}
```

Retorna a depreciação de um ativo entre o mês 6^{º} e o mês 18^{º}. Esse cálculo usa um fator de 3.

[VALUE]
540.185558199698

Exemplo 3

A seguinte consulta DAX:

```
EVALUATE
{
  VDB(2400, 300, 10, 0, 0.875, 1.5)
}
```

Retorna a depreciação de um ativo no primeiro ano fiscal após você ter adquirido ele, supondo que as leis tributárias limitem a depreciação de 150% do saldo decrescente. O ativo é adquirido no meio do primeiro trimestre do ano fiscal.

[VALUE]
315

XIRR

17/03/2021 • 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:

  $$\sum^{N}_{j=1} \frac{P_{j}}{(1 + \text{rate})^{\frac{d_{j} - d_{1}}{365}}}$$

  Em que:

  - P_{j} é o j^{th} pagamento
  - d_{j} é a data do j^{th} pagamento
  - d_{1} é a data do primeiro pagamento
- A série de valores do fluxo de caixa precisa conter pelo menos um número positivo e um número negativo.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A seguinte fórmula calcula a taxa de retorno interna da tabela CashFlows:

```
= XIRR( CashFlows, [Payment], [Date] )
```

DATA	PAGAMENTO
01/01/2014	-10.000
01/03/2014	2750
30/10/2014	4.250
15/2/2015	3250
4/1/2015	2750

Taxa de retorno = 37,49%

XNPV

17/03/2021 • 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:

  $$\sum^{N}_{j=1} \frac{P_{j}}{(1 + \text{rate})^{\frac{d_{j} - d_{1}}{365}}}$$

  Em que:

  - P_{j} é o j^{th} pagamento
  - d_{j} é a data do j^{th} pagamento
  - d_{1} é a data do primeiro pagamento
- A série de valores do fluxo de caixa precisa conter pelo menos um número positivo e um número negativo.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

Confira a seguir o cálculo do valor atual da tabela CashFlows:

```
= XNPV( CashFlows, [Payment], [Date], 0.09 )
```

DATA	PAGAMENTO
01/01/2014	-10.000
01/03/2014	2750
30/10/2014	4.250
15/2/2015	3250
4/1/2015	2750

Valor atual = 2.086,65

YIELD

17/03/2021 • 5 minutes to read

Retorna o rendimento de um título que paga juros periódicos. Use YIELD para calcular o rendimento da obrigação.

Sintaxe

```
YIELD(<settlement>, <maturity>, <rate>, <pr>, <redemption>, <frequency>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
rate	A taxa de cupom anual do título.
pr	O preço do título por \US$ 100 de valor nominal.
redemption	O valor de resgate do título por \US$ 100 de valor nominal.
frequência	O número de pagamentos de cupons por ano. Para pagamentos anuais: frequency = 1, para pagamentos semestrais: frequency = 2 e para pagamentos trimestrais: frequency = 4.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O rendimento do título.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- Se houver um período de cupom ou menos até o resgate, YIELD será calculado da seguinte maneira:

  $$\text{YIELD} = \frac{(\frac{\text{redemption}}{100} + \frac{\text{rate}}{\text{frequency}}) – (\frac{\text{par}}{100} + (\frac{\text{A}}{\text{E}} \times \frac{\text{rate}}{\text{frequency}}))}{\frac{\text{par}}{100} + (\frac{\text{A}}{\text{E}} \times \frac{\text{rate}}{\text{frequency}})} \times \frac{\text{frequency} \times \text{E}}{\text{DSR}}$$

  em que:

  - A = número de dias do início do período de um cupom à data de liquidação (dias acumulados).
  - DSR = número de dias desde a data de liquidação até a data de resgate.
  - E = número de dias no período de um cupom.
- Se houver mais de um período de cupom até o resgate, YIELD será calculado por meio de cem iterações. A resolução usa o método Newton, com base na fórmula usada para a função PRICE. O rendimento é alterado até que o preço estimado considerando o rendimento esteja perto do preço.
- settlement e maturity serão arredondados para números inteiros.
- frequency e basis serão arredondadas para obter o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - rate < 0.
  - pr ≤ 0.
  - redemption ≤ 0.
  - frequency for qualquer número diferente de 1, 2 ou 4.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
15-fev-08	Data de liquidação
15-nov-16	Data de vencimento
5,75%	Cupom percentual
95,04287	Preço
\US$ 100	Valor de resgate
2	A frequência é semestral (confira acima)
0	Base 30/360 (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  YIELD(DATE(2008,2,15), DATE(2016,11,15), 0.0575, 95.04287, 100, 2,0)
}
```

Retorna o rendimento de uma obrigação com os termos especificados acima.

[VALUE]
0,0650000068807314

YIELDDISC

17/03/2021 • 3 minutes to read

Retorna o rendimento anual de um título com desconto.

Sintaxe

```
YIELDDISC(<settlement>, <maturity>, <pr>, <redemption>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
pr	O preço do título por \US$ 100 de valor nominal.
redemption	O valor de resgate do título por \US$ 100 de valor nominal.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O rendimento anual.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são

39.448 dias após 30 de dezembro de 1899.

- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- settlement e maturity serão arredondados para números inteiros.
- basis é arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement ou maturity não forem datas válidas.
  - settlement ≥ maturity.
  - pr ≤ 0.
  - redemption ≤ 0.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	-
16-fev-08	Data de liquidação
1-mar-08	Data de vencimento
99,795	Preço
\US$ 100	Valor de resgate
2	Base real/360

A seguinte consulta DAX:

```
EVALUATE
{
  YIELDDISC(DATE(2008,2,16), DATE(2008,3,1), 99.795, 100, 2)
}
```

Retorna o rendimento anual do título, considerando os termos especificados acima.

[VALUE]
0,0528225719868583

YIELDMAT

17/03/2021 • 4 minutes to read

Retorna o rendimento anual de um título que paga juros no vencimento.

Sintaxe

```
YIELDMAT(<settlement>, <maturity>, <issue>, <rate>, <pr>[, <basis>])
```

Parâmetros

TERMO	DEFINIÇÃO
settlement	A data de liquidação do título. A data de liquidação do título será a data posterior à data de emissão quando o título for negociado com o comprador.
maturity	A data de vencimento do título. A data de vencimento é a data de expiração do título.
problema	A data de emissão do título.
rate	A taxa de juros do título na data de emissão.
pr	O preço do título por \US$ 100 de valor nominal.
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Caso basis seja omitido, ele será considerado 0. Os valores aceitos estão listados na tabela abaixo.

O parâmetro **basis** aceita os seguintes valores:

BASE	BASE DE CONTAGEM DIÁRIA
0 ou omitido	US (NASD) 30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeu 30/360

Valor Retornado

O rendimento anual.

Comentários

- As datas são armazenadas como números de série sequenciais para que possam ser usadas nos cálculos. No DAX, 30 de dezembro de 1899 será o dia 0 e 1º de janeiro de 2008 será o dia 39448 porque são 39.448 dias após 30 de dezembro de 1899.
- A data de liquidação será a data em que o comprador adquire um cupom, como um título de dívida. A data de vencimento será a data de expiração de um cupom. Por exemplo, suponha que um título de dívida de 30 anos seja emitido em 1º de janeiro de 2008 e adquirido por um comprador seis meses depois. A data de emissão seria 1º de janeiro de 2008, a data de liquidação seria 1º de julho de 2008 e a data de vencimento seria 1º de janeiro de 2038, ou seja, 30 anos após a data de emissão em 1º de janeiro de 2008.
- settlement, maturity e issue são arredondados para números inteiros.
- basis é arredondado para o número inteiro mais próximo.
- Um erro será retornado se:
  - settlement, maturity ou issue não for uma data válida.
  - maturity > settlement > issue não for atendido.
  - rate < 0.
  - pr ≤ 0.
  - basis < 0 ou basis > 4.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

DADOS	DESCRIÇÃO
15 de março de 2008	Data de liquidação
3-nov-08	Data de vencimento
8-nov-07	Data de emissão
6,25%	Percentual do cupom semestral
100,0123	Preço
0	Base 30/360 (confira acima)

A seguinte consulta DAX:

```
EVALUATE
{
  YIELDMAT(DATE(2008,3,15), DATE(2008,11,3), DATE(2007,11,8), 0.0625, 100.0123, 0)
}
```

Retorna o rendimento de um título usando os termos especificados acima.

[VALUE]
0,0609543336915387

Funções de informações

17/03/2021 • 5 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.
CONTAINSROW	Retornará TRUE se uma linha de valores existir ou estiver contida em uma tabela; caso contrário, retornará FALSE.
CONTAINSSTRING	Retorna TRUE ou FALSE, indicando se uma cadeia de caracteres contém outra cadeia de caracteres.
CONTAINSSTRINGEXACT	Retorna TRUE ou FALSE, indicando se uma cadeia de caracteres contém outra cadeia de caracteres.
CUSTOMDATA	Retorna o conteúdo da propriedade CustomData na cadeia de conexão.
HASONEFILTER	Retorna TRUE quando o número de valores filtrados diretamente em *columnName* é um; caso contrário, retorna FALSE.
HASONEVALUE	Retorna TRUE quando o contexto para *columnName* foi filtrado para apenas um valor distinto. Caso contrário, será FALSE.
ISBLANK	Verifica se um valor está em branco e retorna TRUE ou FALSE.
ISCROSSFILTERED	Retorna TRUE quando *columnName* ou outra coluna na mesma tabela ou relacionada está sendo filtrada.
ISEMPTY	Verifica se uma tabela está vazia.
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.
ISFILTERED	Retorna TRUE quando *columnName* está sendo filtrado diretamente.

FUNÇÃO	DESCRIÇÃO
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.
ISODD	Retornará TRUE se o número for ímpar ou FALSE se o número for par.
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.
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.
ISSUBTOTAL	Cria outra coluna, em uma expressão SUMMARIZE, que retornará True se a linha contiver valores de subtotal para a coluna fornecida como argumento; caso contrário, retornará False.
ISTEXT	Verifica se um valor é texto e retorna TRUE ou FALSE.
NONVISUAL	Marca um filtro de valor em uma expressão SUMMARIZECOLUMNS como não visual.
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.
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

17/03/2021 • 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 *columnName*. 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

O exemplo a seguir cria uma medida que informa se houve vendas na Internet do produto 214 e para o cliente 11185 ao mesmo tempo.

```
= CONTAINS(InternetSales, [ProductKey], 214, [CustomerKey], 11185)
```

Função CONTAINSROW

17/03/2021 • 2 minutes to read

Retornará TRUE se uma linha de valores existir ou estiver contida em uma tabela; caso contrário, retornará FALSE.

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.

Retornar valor

TRUE ou FALSE.

Comentários

- A sintaxe Except, o operador **IN** e a função CONTAINSROW são funcionalmente equivalentes.

  ```
  <scalarExpr> IN <tableExpr>
  ( <scalarExpr1>, <scalarExpr2>, … ) IN <tableExpr>
  ```

  - O número de scalarExprN deve corresponder ao número de colunas em tableExpr.
  - 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" }.
- Ao contrário do operador =, o operador IN e a função CONTAINSROW executam uma comparação estrita. Por exemplo, o valor BLANK não corresponde a 0.

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]))
```

Retorne:

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

DIMPRODUCT[COLOR]
Multi
NA
Prata
Prata\Preto
Branca

CONTAINSSTRING

17/03/2021 • 2 minutes to read

Retorna TRUE ou FALSE, indicando se uma cadeia de caracteres contém outra cadeia de caracteres.

Sintaxe

```
CONTAINSSTRING(<within_text>, <find_text>)
```

Parâmetros

TERMO	DEFINIÇÃO
within_text	O texto no qual você deseja pesquisar find_text.
find_text	O texto que você deseja encontrar.

Retornar valor

TRUE se find_text for uma substring de within_text; caso contrário, FALSE.

Comentários

- CONTAINSSTRING não diferencia maiúsculas de minúsculas.
- Você pode usar os caracteres curinga `?` e `*`. Use `~` para fazer o escapar de caracteres curinga.

Exemplo

Consulta DAX

```
EVALUATE
    ROW(
        "Case 1", CONTAINSSTRING("abcd", "bc"),
        "Case 2", CONTAINSSTRING("abcd", "BC"),
        "Case 3", CONTAINSSTRING("abcd", "a*d"),
        "Case 4", CONTAINSSTRING("abcd", "ef")
    )
```

Retorna

[MAIÚSCULAS E MINÚSCULAS 1]	[MAIÚSCULAS E MINÚSCULAS 2]	[MAIÚSCULAS E MINÚSCULAS 3]	[MAIÚSCULAS E MINÚSCULAS 4]
VERDADEIRO	VERDADEIRO	TRUE	FALSE

CONTAINSSTRINGEXACT

17/03/2021 • 2 minutes to read

Retorna TRUE ou FALSE, indicando se uma cadeia de caracteres contém outra cadeia de caracteres.

Sintaxe

```
CONTAINSSTRINGEXACT(<within_text>, <find_text>)
```

Parâmetros

TERMO	DEFINIÇÃO
within_text	O texto no qual você deseja pesquisar find_text.
find_text	O texto que você deseja encontrar.

Retornar valor

TRUE se find_text for uma substring de within_text; caso contrário, FALSE.

Comentários

CONTAINSSTRINGEXACT diferencia maiúsculas de minúsculas.

Exemplo

Consulta DAX

```
EVALUATE
    ROW(
        "Case 1", CONTAINSSTRINGEXACT("abcd", "bc"),
        "Case 2", CONTAINSSTRINGEXACT("abcd", "BC"),
        "Case 3", CONTAINSSTRINGEXACT("abcd", "a*d"),
        "Case 4", CONTAINSSTRINGEXACT("abcd", "ef")
    )
```

Retorna

[MAIÚSCULAS E MINÚSCULAS 1]	[MAIÚSCULAS E MINÚSCULAS 2]	[MAIÚSCULAS E MINÚSCULAS 3]	[MAIÚSCULAS E MINÚSCULAS 4]
TRUE	FALSO	FALSO	FALSE

CUSTOMDATA

17/03/2021 • 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.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A fórmula 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")
```

HASONEFILTER

17/03/2021 • 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

- 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 2 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`.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A fórmula de medida a seguir verificará se o contexto está sendo segmentado por um valor a fim de estimar um percentual em relação a um cenário predefinido. Nesse caso, você deseja comparar as Vendas do Revendedor com as vendas de 2007. Depois será necessário saber se o contexto será filtrado por ano. Além disso, se a comparação for inútil, você poderá retornar BLANK.

```
=
IF(HASONEVALUE(DateTime[CalendarYear]),SUM(ResellerSales_USD[SalesAmount_USD])/CALCULATE(SUM(ResellerSales_U
SD[SalesAmount_USD]),DateTime[CalendarYear]=2007),BLANK())
```

ISBLANK

17/03/2021 • 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

ISCROSSFILTERED

17/03/2021 • 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 função ISFILTERED retornará TRUE quando *columnName* for filtrado de modo direto.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Consulte também

Função ISFILTERED
Função FILTERS
Função HASONEFILTER
Função HASONEVALUE

ISEMPTY

26/04/2021 • 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.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

Para a tabela abaixo nomeada '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, [County]>25))))
```

Valor retornado: **FALSE**

ISERROR

17/03/2021 • 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.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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])
    )
```

Confira também

Funções informativas
Função IFERROR
Função IF

ISEVEN

17/03/2021 • 2 minutes to read

Retorna TRUE se o número é par ou FALSE se é ímpar.

Sintaxe

```
ISEVEN(number)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O valor a testar. Se o número não for um inteiro, ele será truncado.

Retornar valor

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!.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

ISFILTERED

17/03/2021 • 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 ISCROSSFILTERED retornará TRUE quando *columnName* ou outra coluna na mesma tabela ou em uma tabela relacionada estiver sendo filtrada.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Consulte também

Função ISCROSSFILTERED
Função FILTERS
Função HASONEFILTER
Função HASONEVALUE

ISINSCOPE

17/03/2021 • 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.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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[CATEGORY]	DIMPRODUCT[SUBCATEGORY]	[SUBTOTAL DA CATEGORIA]	[SUBTOTAL DA SUBCATEGORIA]	[VENDAS]	[% DO PAI]
		TRUE	TRUE	28.397.095,65	100,00
Acessórios		FALSO	TRUE	78.951,00	0,28
Acessórios	Racks de bicicleta	FALSE	FALSE	39.360,00	49,85
Acessórios	Suportes de bicicleta	FALSE	FALSE	39.591,00	50.15
Bikes		FALSE	TRUE	28.318.144,65	99,72
Bikes	Mountain bikes	FALSE	FALSE	9.952.759,56	35,15
Bikes	Bicicletas de estrada	FALSE	FALSE	14.520.584,04	51,28
Bikes	Bicicletas de passeio	FALSE	FALSE	3.844.801,05	13,58

Confira também

Função SUMMARIZECOLUMNS
função CALCULATE

ISLOGICAL

17/03/2021 • 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.

Retornar valor

TRUE se o valor for um valor lógico; FALSE se for qualquer valor diferente de TRUE ou FALSE.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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")
```

Confira também

Funções informativas

ISNONTEXT

17/03/2021 • 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.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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")
```

Confira também

Funções informativas

ISNUMBER

17/03/2021 • 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 retornado

TRUE se o valor for numérico; caso contrário, FALSE.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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")
```

Confira também

Funções informativas

ISODD

17/03/2021 • 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	O valor a testar. Se o número não for um inteiro, ele será truncado.

Retornar valor

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!.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

ISONORAFTER

17/03/2021 • 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.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

Nome da tabela: 'Info'

PAÍS	ESTADO	CONTAGEM	TOTAL
IND	JK	20	800

PAÍS	ESTADO	CONTAGEM	TOTAL
IND`	MH	25	1000
IND	WB	10	900
EUA	AC	5	500
EUA	WA	10	900

```
FILTER(Info, ISONORAFTER(Info[Country], "IND", ASC, Info[State], "MH", ASC))
```

ISSELECTEDMEASURE

17/03/2021 • 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.

Sintaxe

```
ISSELECTEDMEASURE( M1, M2, ... )
```

Parâmetros

TERMO	DEFINIÇÃO
M1, M2, ...	Uma lista de medidas.

Valor retornado

Um booliano 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

ISSUBTOTAL

17/03/2021 • 2 minutes to read

Cria outra coluna, em uma expressão SUMMARIZE, que retornará True se a linha contiver valores de subtotal para a coluna fornecida como argumento; caso contrário, retornará False.

Syntax

```
ISSUBTOTAL(<columnName>)
```

Com SUMMARIZE,

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]…[, ROLLUP(<groupBy_columnName>[,<
groupBy_columnName>…])][, <name>, {<expression>|ISSUBTOTAL(<columnName>)}]…)
```

Parâmetros

TERMO	DEFINIÇÃO
columnName	O nome de qualquer coluna na tabela da função SUMMARIZE ou qualquer coluna em uma tabela relacionada à tabela.

Retornar valor

Um valor True se a linha contiver um valor subtotal para a coluna fornecida como argumento; caso contrário, retornará False.

Comentários

- Essa função pode ser usada somente na expressão de uma função SUMMARIZE.
- Essa função deve ser precedida pelo nome da coluna booliana.

Exemplo

Confira SUMMARIZE.

ISTEXT

17/03/2021 • 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.

Retornar valor

TRUE se o valor for um texto. Caso contrário, FALSE.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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")
```

Confira também

Funções informativas

NONVISUAL

17/03/2021 • 2 minutes to read

Marca um filtro de valor em uma expressão SUMMARIZECOLUMNS como não visual. Essa função pode ser usada apenas dentro de uma expressão SUMMARIZECOLUMNS.

Sintaxe

```
NONVISUAL(<expression>)
```

Parâmetros

TERMO	DEFINIÇÃO
expressão	Qualquer expressão DAX que retorna um único valor (não é uma tabela).

Retornar valor

Uma tabela de valores.

Comentários

- Marca que um filtro de valor em SUMMARIZECOLUMNS não afeta os valores da medida, mas aplica-se apenas às colunas agrupar por.
- Essa função pode ser usada apenas dentro de uma expressão SUMMARIZECOLUMNS. Ela é usada como um argumento filterTable da função SUMMARIZECOLUMNS ou um argumento groupLevelFilter da função ROLLUPADDISSUBTOTAL ou ROLLUPISSUBTOTAL.

Exemplo

Confira SUMMARIZECOLUMNS.

SELECTEDMEASURE

17/03/2021 • 2 minutes to read

Usada por expressões de itens de cálculo para referenciar a medida que está no contexto.

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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.

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 2 minutes to read

Usada por expressões de itens de cálculo para determinar a medida que está no contexto por nome.

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
ISSELECTEDMEASURE

USERNAME

17/03/2021 • 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

A fórmula a seguir verifica se o logon do usuário faz parte da UsersTable.

```
= IF(CONTAINS(UsersTable,UsersTable[login], USERNAME()), "Allowed", BLANK())
```

USEROBJECTID

17/03/2021 • 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

17/03/2021 • 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

17/03/2021 • 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 uma condição e retorna um valor quando é TRUE; caso contrário, retorna um segundo valor.
IF.EAGER	Verifica uma condição e retorna um valor quando é TRUE; caso contrário, retorna um segundo valor. Usa um plano de execução *eager* que sempre executa as expressões de ramificação, independentemente da expressão de condição.
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

17/03/2021 • 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

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						
Luzes						
Bloqueios						

RÓTULOS DE LINHA	2005	2006	2007	2008	-	TOTAL GERAL
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

Funções lógicas

COALESCE

17/03/2021 • 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

17/03/2021 • 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.

RÓTULOS DE LINHA	2005	2006	2007	2008	-	TOTAL GERAL
Acessórios	FALSO	FALSO	TRUE	VERDADEIRO	FALSE	VERDADEIRO
Bicicletas	VERDADEIRO	VERDADEIRO	VERDADEIRO	VERDADEIRO	FALSE	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	FALSE	TRUE

Consulte também

Função TRUE
Função NOT
Função IF

IF

17/03/2021 • 3 minutes to read

Verifica uma condição e retorna um valor quando é TRUE; caso contrário, retorna um segundo valor.

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 é TRUE.
value_if_false	(Opcional) O valor retornado se o teste lógico é FALSE. Se omitido, BLANK será retornado.

Retornar valor

value_if_true, **value_if_false** ou BLANK.

Comentários

- A função IF pode retornar o tipo de dados de variante se **value_if_true** e **value_if_false** são de tipos de dados diferentes, mas a função tenta retornar um só tipo de dados se **value_if_true** e **value_if_false** são 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 saber mais sobre a conversão implícita de tipos de dados, confira Tipos de dados.

- Para executar as expressões de ramificação independentemente da expressão de condição, use IF.EAGER.

Exemplos

As definições de coluna calculadas da tabela **Product** usam a função IF de diferentes maneiras para classificar cada produto com base no preço de lista.

O primeiro exemplo testa se o valor da coluna **List Price** é menor que 500. Quando essa condição é verdadeira, o valor **Baixo** é retornado. Como não há nenhum valor **value_if_false**, BLANK é retornado.

Os exemplos deste artigo podem ser adicionados ao modelo de exemplo do Power BI Desktop. Para obter o modelo, confira Modelo de exemplo DAX.

```
Price Group =
IF(
    'Product'[List Price] < 500,
    "Low"
)
```

O segundo exemplo usa o mesmo teste, mas desta vez inclui um valor **value_if_false**. Portanto, a fórmula classifica cada produto como **Baixo** ou **Alto**.

```
Price Group =
IF(
    'Product'[List Price] < 500,
    "Low",
    "High"
)
```

O terceiro exemplo usa o mesmo teste, mas desta vez aninha uma função IF para executar um teste adicional. Portanto, a fórmula classifica cada produto como **Baixo**, **Médio** ou **Alto**.

```
Price Group =
IF(
    'Product'[List Price] < 500,
    "Low",
    IF(
        'Product'[List Price] < 1500,
        "Medium",
        "High"
    )
)
```

TIP

Quando você precisa aninhar várias funções IF, a função SWITCH pode ser uma opção melhor. Essa função fornece uma forma mais elegante de escrever uma expressão que retorna mais de dois valores possíveis.

Confira também

Função IF.EAGER
Função SWITCH (DAX)
Funções lógicas

IF.EAGER

17/03/2021 • 2 minutes to read

Verifica uma condição e retorna um valor quando é TRUE; caso contrário, retorna um segundo valor. Ela usa um plano de execução *eager* que sempre executa as expressões de ramificação, independentemente da expressão de condição.

Sintaxe

```
IF.EAGER(<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 é TRUE.
value_if_false	(Opcional) O valor retornado se o teste lógico é FALSE. Se omitido, BLANK será retornado.

Retornar valor

value_if_true, **value_if_false** ou BLANK.

Comentários

- A função IF.EAGER pode retornar o tipo de dados de variante se value_if_true e value_if_false são de tipos de dados diferentes, mas a função tenta retornar um só tipo de dados se **value_if_true** e **value_if_false** são de tipos de dados numéricos. No último caso, a função IF.EAGER converterá implicitamente os tipos de dados para acomodar os dois valores.

  Por exemplo, a fórmula `IF.EAGER(<condition>, TRUE(), 0)` retorna TRUE ou 0, mas a fórmula `IF.EAGER(<condition>, 1.0, 0)` retorna apenas valores decimais, embora **value_if_false** seja do tipo de dados número inteiro. Para saber mais sobre a conversão implícita de tipos de dados, confira Tipos de dados.

- IF.EAGER tem o mesmo comportamento funcional que a função IF, mas o desempenho pode ser diferente devido a diferenças nos planos de execução.
  `IF.EAGER(<logical_test>, <value_if_true>, <value_if_false>)` tem o mesmo plano de execução que a seguinte expressão DAX:

  ```
  VAR _value_if_true = <value_if_true>
  VAR _value_if_false = <value_if_false>
  RETURN
  IF (<logical_test>, _value_if_true, _value_if_false)
  ```

Observação: as duas expressões de ramificação são avaliadas independentemente da expressão de condição.

Exemplos

Confira exemplos de IF.

Confira também

Função IF
Funções lógicas

IFERROR

17/03/2021 • 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)
  ```

  Os valores retornados para A e B devem ter o mesmo tipo de dados, portanto, a coluna ou a expressão usada para **value** e o valor retornado para **value_if_error** devem ter o mesmo tipo de dados.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

NOT

17/03/2021 • 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
Função FALSE
Função IF

OU

17/03/2021 • 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

RÓTULOS DE LINHA	2005	2006	2007	2008	-	TOTAL GERAL
Abbas, Syed E						

RÓTULOS DE LINHA	2005	2006	2007	2008	-	TOTAL GERAL
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						
Varkey Chudukatil, Ranjit R						Círculo de Excelência
Grande Total	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência

Consulte também

Funções lógicas

SWITCH

17/03/2021 • 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).
value	Um valor constante a ser correspondido com os resultados da *expressão*.
result	Qualquer expressão escalar a ser avaliada se os resultados da *expressão* corresponderem ao *valor* correspondente.
else	Qualquer expressão escalar a ser avaliada se os resultados da *expressão* não corresponderem a nenhum dos argumentos *value*.

Retornar valor

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

17/03/2021 • 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 ou uma visualização, com 'ProductCategory'[ProductCategoryName] nos rótulos de linha e 'DateTime'[CalendarYear] nos rótulos de coluna.

RÓTULOS DE LINHA	2005	2006	2007	2008	-	TOTAL GERAL
Acessórios	FALSO	FALSO	TRUE	VERDADEIRO	FALSE	VERDADEIRO
Bicicletas	VERDADEIRO	VERDADEIRO	VERDADEIRO	VERDADEIRO	FALSE	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	FALSE	TRUE

Consulte também

FALSE
NOT
IF

Funções matemáticas e trigonométricas

17/03/2021 • 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.
CONVERT	Converte uma expressão de um tipo de dados para outro.
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.

FUNÇÃO	DESCRIÇÃO
DIVIDE	Efetua a divisão e retorna um resultado alternativo ou BLANK() na divisão por zero.
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.

FUNÇÃO	DESCRIÇÃO
RAND	Retorna um número aleatório maior ou igual a 0 e menor que 1, distribuído uniformemente.
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

17/03/2021 • 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
função SIGN

ACOS

17/03/2021 • 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

17/03/2021 • 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.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
= ACOSH(1)	Cosseno hiperbólico inverso de 1.	0
= ACOSH(10)	Cosseno hiperbólico inverso de 10.	2,993228

ACOT

17/03/2021 • 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.

Retornar valor

Um valor decimal único.

ACOTH

17/03/2021 • 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.

Retornar valor

Um valor decimal único.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

ASIN

17/03/2021 • 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

17/03/2021 • 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.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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

17/03/2021 • 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.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
= ATANH(0.76159416)	Tangente hiperbólica inversa de 0,76159416	1,00000001
= ATANH(-0.1)		-0,100335348

Confira também

função ATAN

CEILING

17/03/2021 • 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 úmeros 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 1

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 2

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
função FLOOR
Função ISO.CEILING
função ROUNDUP

COMBIN

17/03/2021 • 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:

  $${n \choose k} = \frac{P_{k,n}}{k!} = \frac{n!}{k!(n-k)!}$$

  Where

  $$P_{k,n} = \frac{n!}{(n-k)!}$$

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO

FÓRMULA	DESCRIÇÃO	RESULTADO
= COMBIN(8,2)	Possíveis equipes de duas pessoas que podem ser formadas com oito candidatos.	28

COMBINA

17/03/2021 • 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 seguinte equação é usada, em que N é Number e M é Number_chosen:

  $${N+M-1 \choose N-1}$$

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

CONVERT

17/03/2021 • 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 <Expression>, convertido em <Datatype>.

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))`.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

Consulta DAX

```
EVALUATE { CONVERT(DATE(1900, 1, 1), INTEGER) }
```

Retorna

[VALOR]
2

COS

17/03/2021 • 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 PI()/180 ou use a função RADIANS para converter o ângulo em radianos.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
= COS(1,047)	Cosseno de 1,047 radianos	0,5001711
= COS(60*PI()/180)	Cosseno de 60 graus	0,5
= COS(RADIANS(60))	Cosseno de 60 graus	0,5

COSH

17/03/2021 • 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:

  $$\text{COSH}(z) = \frac{e^{z} + e^{-z}}{2}$$

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

22/04/2021 • 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.

Retornar valor

O valor da expressão avaliada e retornada como um valor de tipo de moeda.

Comentários

- A função CURRENCY arredonda o quinto decimal significativo, em valor, para retornar o quarto dígito decimal. O arredondamento ocorrerá se o quinto decimal significativo for igual ou maior que cinco. Por exemplo, se o valor for 3,6666666666666, a conversão em moeda retornará \$ 3,6667. No entanto, se o valor for 3,0123456789, a conversão em moeda retornará \$ 3,0123.
- Se o tipo de dados da expressão for TrueFalse, CURRENCY(<TrueFalse>) retornará \$ 1,0000 para valores True e \$ 0,0000 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

17/03/2021 • 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

17/03/2021 • 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 1

O exemplo a seguir retorna BLANK.

```
= DIVIDE(5,0)
```

Exemplo 2

O exemplo a seguir retorna 1.

```
= DIVIDE(5,0,1)
```

Consulte também

Função QUOTIENT

Funções matemáticas e trigonométricas

EVEN

17/03/2021 • 2 minutes to read

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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
função LN
função EXP
função LOG
função LOG

FACT

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

VALORES	RESULTADOS
5	120
170	7.257415615308E+306

Consulte também

Funções matemáticas e trigonométricas
Função TRUNC

FLOOR

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

VALORES	RESULTADO ESPERADO
2,9733	2.9

Consulte também

Funções matemáticas e trigonométricas

GCD

17/03/2021 • 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 >= 2^53, GCD retornará o valor de erro #VALUE!.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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
função ROUND
função ROUNDUP
Função ROUNDDOWN
função MROUND

ISO.CEILING

17/03/2021 • 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 úmeros 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

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.

```
= ISO.CEILING(4.42,0.05)
```

Exemplo: números negativos

A seguinte fórmula retorna o valor de teto ISO de -4,40.

```
= ISO.CEILING(-4.42,0.05)
```

Confira também

Funções matemáticas e trigonométricas
função FLOOR
Função CEILING
função ROUNDUP

LCM

17/03/2021 • 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 LCM(a,b) >=2^53, LCM retornará o valor de erro #VALUE!.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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
função EXP

LOG

17/03/2021 • 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
função EXP
função LOG
função LOG

LOG10

17/03/2021 • 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
função EXP
função LOG
função LOG

MOD

17/03/2021 • 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 1

A fórmula a seguir retorna 1, o resto de 3 dividido por 2.

```
= MOD(3,2)
```

Exemplo 2

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
função ROUND
função ROUNDUP
Função ROUNDDOWN

função MROUND
função INT

MROUND

17/03/2021 • 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

A expressão a seguir arredonda 1,3 para o múltiplo mais próximo de 0,2. O resultado esperado é 1,4.

```
= MROUND(1.3,0.2)
```

Exemplo: números negativos

A expressão a seguir arredonda -10 para o múltiplo mais próximo de -3. O resultado esperado é -9.

```
= MROUND(-10,-3)
```

Exemplo: erro

A expressão a seguir retorna um erro, pois os números têm sinais diferentes.

```
= MROUND(5,-2)
```

Confira também

ODD

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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

POWER

17/03/2021 • 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

PRODUTO

17/03/2021 • 2 minutes to read

Retorna o produto dos números em uma coluna.

Sintaxe

```
PRODUCT(<column>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna que contém os números para os quais o produto deve ser calculado.

Retornar valor

Um número decimal.

Comentários

- Para retornar o produto de uma expressão avaliada para cada linha de uma tabela, use a função PRODUCTX.
- Somente os números na coluna são contados. Espaços em branco, valores lógicos e texto são ignorados. Por exemplo,

  `PRODUCT( Table[Column] )` é equivalente a `PRODUCTX( Table, Table[Column] )`.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

O seguinte calcula o produto da coluna AdjustedRates em uma tabela de Anuidade:

```
= PRODUCT( Annuity[AdjustedRates] )
```

Confira também

PRODUCTX

PRODUCTX

17/03/2021 • 2 minutes to read

Retorna o produto de uma expressão avaliada para cada linha de uma tabela.

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.

Retornar valor

Um número decimal.

Comentários

- Para retornar o produto dos números em uma coluna, use PRODUCT.
- 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

O exemplo a seguir computa o valor futuro de um investimento:

```
= [PresentValue] * PRODUCTX( AnnuityPeriods, 1+[FixedInterestRate] )
```

Consulte também

PRODUCT

QUOTIENT

17/03/2021 • 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

RADIANS

17/03/2021 • 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 3π/2 radianos)	4,712389

RAND

17/03/2021 • 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()
```

Retornar valor

Um número decimal.

Comentários

- O recálculo depende de vários fatores, incluindo se o modelo está definido com o modo de recálculo **Manual** ou **Automático** e se os dados foram atualizados.
- 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.
- RAND sempre é recalculado se a função é usada na definição de uma medida.
- A função RAND não pode retornar um resultado igual a zero, a fim de evitar erros como a divisão por zero.

Exemplos

Para gerar um número real aleatório entre dois outros números, use:

```
= RAND()*(b-a)+a
```

Para gerar um número aleatório maior que 0 e menor que 1:

```
= RAND()
```

Para gerar um número aleatório maior que 0 e menor que 100

```
= RAND()*100
```

Para gerar um número inteiro aleatório maior que 0 e menor que 100

```
INT(RAND()*100)
```

Consulte também

Funções matemáticas e trigonométricas
Funções estatísticas

RANDBETWEEN

17/03/2021 • 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A fórmula a seguir retorna um número aleatório entre 1 e 10.

```
= RANDBETWEEN(1,10)
```

Confira também

Funções matemáticas e trigonométricas
Funções estatísticas

ROUND

17/03/2021 • 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.
  - Use as funções TRUNC e INT para obter a parte inteira do número.

Exemplo 1

A fórmula a seguir arredonda 2,15 para cima, para uma casa decimal. O resultado esperado é 2,2.

```
= ROUND(2.15,1)
```

Exemplo 2

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
ROUND
ROUNDDOWN
MROUND
INT
TRUNC

ROUNDDOWN

17/03/2021 • 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.
- 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 1

O exemplo a seguir arredonda 3,14159 para baixo até três casas decimais. O resultado esperado é 3,141.

```
= ROUNDDOWN(3.14159,3)
```

Exemplo 2

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)
```

Confira também

Funções matemáticas e trigonométricas
ROUND
ROUNDUP
ROUNDDOWN
MROUND
INT

ROUNDUP

17/03/2021 • 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

- 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.
- 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

A fórmula a seguir Arredonda 1,3 para o múltiplo mais próximo de 0,2. O resultado esperado é 2.

```
= ROUNDUP(1.3,0.2)
```

Exemplo: Número negativo como segundo argumento

A fórmula a seguir Arredonda o valor na coluna, **FreightCost**, com os resultados esperados mostrados na tabela a seguir:

```
= ROUNDUP([Values],-1)
```

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

Confira também

Funções matemáticas e trigonométricas
ROUND
ROUNDDOWN
MROUND
INT

SIGN

17/03/2021 • 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

SQRT

17/03/2021 • 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

```
= SQRT(25)
```

Consulte também

Funções matemáticas e trigonométricas

SUM

17/03/2021 • 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.

Retornar valor

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])
```

Confira também

SUMX

SUMX

17/03/2021 • 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.

Retornar valor

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 exemplos mais complexos de SUMX em fórmulas, confira ALL e CALCULATETABLE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

SUM

TRUNC

17/03/2021 • 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 1

A fórmula a seguir retorna 3, a parte inteira de PI.

```
= TRUNC(PI())
```

Exemplo 2

A fórmula a seguir retorna -8, a parte inteira de -8,9.

```
= TRUNC(-8.9)
```

Consulte também

Funções matemáticas e trigonométricas
ROUND
ROUNDUP
ROUNDDOWN

MROUND
INT

Outras funções

17/03/2021 • 2 minutes to read

Essas funções executam ações exclusivas que não podem ser definidas por nenhuma das categorias.

Nesta categoria

FUNÇÃO	DESCRIÇÃO
BLANK	Retorna um espaço em branco.
ERROR	Gera um erro com uma mensagem de erro.

BLANK

17/03/2021 • 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.

RÓTULOS DE LINHA	ACESSÓRIOS	BIKES	CLOTHING	GRANDE TOTAL
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
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
função ISBLANK

ERRO

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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 uma 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.

Funções pai e filho

17/03/2021 • 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 *item* especificado existir no *path* especificado.
PATHITEM	Retorna o item na *posição* especificada de uma cadeia de caracteres resultante da avaliação de uma função PATH.
PATHITEMREVERSE	Retorna o item na *posição* 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

17/03/2021 • 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 – retorna um texto delimitado com os identificadores de todos os pais da linha atual, começando com o mais antigo ou mais alto até o 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 – 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])`', e 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 – retorna o item na posição especificada de um resultado semelhante a PATH(), contando da esquerda para a direita. No exemplo a seguir, a coluna PathItem, que é a 4ª partindo da esquerda, é definida como "`= PATHITEM([Path], 4)`". Esse exemplo retorna o EmployeeKey na quarta posição na cadeia de caracteres Path começando da esquerda, usando os mesmos dados 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 – retorna o item na *posição* de um resultado de função semelhante a PATH(), contando de maneira inversa da direita para a esquerda.
No exemplo a seguir, a coluna PathItemReverse, que é a 3ª partindo da direita, é definida como '`= PATHITEMREVERSE([Path], 3)`'. Esse exemplo retorna o EmployeKey na terceira posição na cadeia de caracteres Path começando da direita, usando os mesmos dados 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 – retornará **TRUE** se o *item* especificado existir no *caminho* especificado. No exemplo a seguir, na coluna PathContains, o 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

17/03/2021 • 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 *ID_columnName* deve ser texto ou inteiro e também deve ser o mesmo tipo de dados que a coluna referenciada em *parent_columnName*.
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 *parent_columnName* deve ser texto ou inteiro e deve ser o mesmo tipo de dados que o valor em *parent_columnName*.

Retornar valor

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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

O exemplo a seguir cria uma coluna calculada que lista todos os gerentes de cada funcionário.

```
= PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey])
```

PATHCONTAINS

17/03/2021 • 2 minutes to read

Retornará **TRUE** se o *item* especificado existir no *path* especificado.

Sintaxe

```
PATHCONTAINS(<path>, <item>)
```

Parâmetros

TERMO	DEFINIÇÃO
caminho	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.

Retornar valor

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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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

TERMO	DEFINIÇÃO
caminho	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.
tipo	(Opcional) Uma enumeração que define o tipo de dados do resultado:

enumeração de tipo

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.

Retornar valor

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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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

TERMO	DEFINIÇÃO
caminho	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.
tipo	(Opcional) Uma enumeração que define o tipo de dados do resultado:

enumeração de tipo

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.

Retornar valor

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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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
caminho	Uma expressão de texto resultante da avaliação de uma função PATH.

Retornar valor

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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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 de relação

17/03/2021 • 2 minutes to read

As funções nessa categoria são para o gerenciamento e a utilização de relações entre tabelas.

Nesta categoria

FUNÇÃO	DESCRIÇÃO
CROSSFILTER	Especifica a direção de filtragem cruzada a ser usada em um cálculo para uma relação existente entre duas colunas.
RELATED	Retorna um valor relacionado de outra tabela.
RELATEDTABLE	Avalia uma expressão de tabela em um contexto modificado pelos filtros especificados.
USERELATIONSHIP	Especifica a relação a ser usada em um cálculo específico como aquela que existe entre columnName1 e columnName2.

CROSSFILTER

22/04/2021 • 6 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. Deve ser uma destas opções: **None** – nenhuma filtragem cruzada ocorre nessa relação. **Both** – os filtros em qualquer um dos lados e filtram o outro lado. **OneWay** – os filtros de um lado ou o lado de pesquisa de uma relação filtram o outro lado. Esta opção não pode ser usada com uma relação de um para um. Não use esta opção em uma relação de muitos para muitos porque não está claro qual lado é o lado de pesquisa; em vez disso, use OneWay_LeftFiltersRight ou OneWay_RightFiltersLeft. **OneWay_LeftFiltersRight** – os filtros no lado de <columnName1> filtram o lado de <columnName2>. Esta opção não pode ser usada com uma relação de um para um ou de muitos para um. **OneWay_RightFiltersLeft** – os filtros no lado de <columnName2> filtram o lado de <columnName1>. Esta opção não pode ser usada com uma relação de um para um ou de muitos para um.

Retornar valor

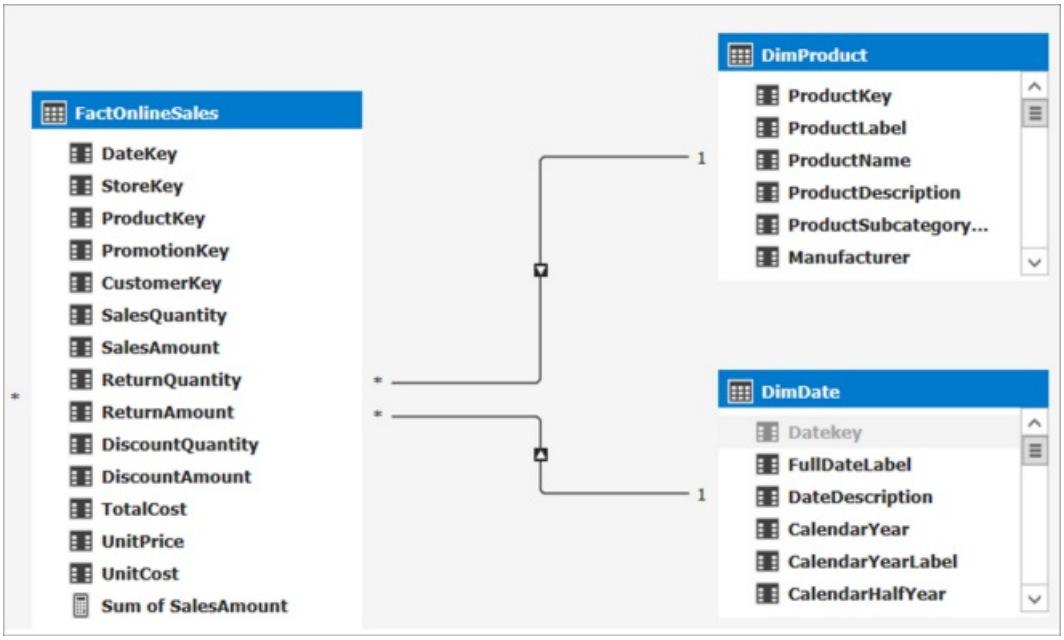
A função não retorna nenhum valor; a função define apenas a direção de filtragem cruzada para a relação indicada, durante o tempo da consulta.

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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**

RELATED

17/03/2021 • 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.
- 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.
- A função RELATED não pode ser usada para buscar uma coluna em uma relação limitada.

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. Os Estados Unidos, como um país, são exibidos 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	INTERNET SALES	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	
Grande Total	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:

RÓTULOS DE LINHA	ACESSÓRIOS	BIKES	CLOTHING	GRANDE TOTAL
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
Grande Total	US$ 384.639,63	US$ 16.107.890,23	US$ 172.817,81	US$ 16.665.347,67

Confira também

RELATEDTABLE
Funções de filtro

RELATEDTABLE

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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 seguinte tabela mostra os resultados:

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

CALCULATETABLE
Funções de filtro

USERELATIONSHIP

17/03/2021 • 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 USEREALTIONSHIP:

  ```
  =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, crie uma medida, [InternetSales by ShippingDate], usando a seguinte expressão:

```
= CALCULATE(SUM(InternetSales[SalesAmount]), USERELATIONSHIP(InternetSales[ShippingDate], DateTime[Date]))
```

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.

Funções estatísticas

17/03/2021 • 7 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
APPROXIMATEDISTINCTCOUNT	Retorna o número *aproximado* 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.
COUNT	Conta o número de células de uma coluna que contém números.

FUNÇÃO	DESCRIÇÃO
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.
DATATABLE	Fornece 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.
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.
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.

FUNÇÃO	DESCRIÇÃO
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 *table*.
SAMPLE	Retorna uma amostra de N linhas da tabela especificada.
SIN	Retorna o seno do ângulo determinado.
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).
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.

FUNÇÃO	DESCRIÇÃO
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.
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.

APPROXIMATEDISTINCTCOUNT

17/03/2021 • 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.

Retornar valor

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.

AVERAGE

17/03/2021 • 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.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
Função AVERAGEX
Funções estatísticas

AVERAGEX

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
Função AVERAGEA
Funções estatísticas

BETA.DIST

17/03/2021 • 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.
Um	Opcional. Um limite inferior para o intervalo de x.
B	Opcional. Um limite superior para o intervalo de x.

Retornar valor

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 alfa ≤0 ou beta ≤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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

BETA.INV

17/03/2021 • 2 minutes to read

Retorna o inverso da função de densidade de probabilidade cumulativa beta (BETA.DIST).

Se probability = BETA.DIST(x,...TRUE), then BETA.INV(probability,...) = 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.
Um	Opcional. Um limite inferior para o intervalo de x.
B	Opcional. Um limite superior para o intervalo de x.

Retornar valor

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 alfa ≤0 ou beta ≤0, BETA.INV retornará o valor de erro #VALUE!.
- Se probabilidade ≤ 0 ou 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

CHISQ.DIST

17/03/2021 • 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 deg_freedom < 1 ou deg_freedom > 10^10, um erro é retornado.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A seguinte consulta DAX:

```
EVALUATE { CHISQ.DIST(2, 2, TRUE) }
```

Retorna

[VALOR]
0.632120558828558

CHISQ.DIST.RT

17/03/2021 • 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 deg_freedom < 1 ou deg_freedom > 10^10, um erro é retornado.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A seguinte consulta DAX:

```
EVALUATE { CHISQ.DIST.RT(2, 5) }
```

Retorna

[VALOR]
0.84914503608461

CHISQ.INV

17/03/2021 • 2 minutes to read

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 or probability > 1, CHISQ.INV retorna o #NUM! #VALUE!.
- Se deg_freedom não for um inteiro, ele será arredondado.
- Se deg_freedom < 0 or deg_freedom > 10^10, CHISQ.INV retorna o #NUM! #VALUE!.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 2 minutes to read

Retorna o inverso da probabilidade de cauda direita da distribuição qui-quadrada.

Se probability = CHISQ.DIST.RT(x,...), CHISQ.INV.RT(probability,...) = 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 CHISQ.DIST.RT(x, deg_freedom) = 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

CONFIDENCE.NORM

17/03/2021 • 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 ± CONFIDENCE.NORM. Por exemplo, se x é a média amostral dos tempos de entrega para os produtos solicitados pelo correio, x ± CONFIDENCE.NORM é um intervalo de médias populacionais. Para qualquer média populacional, μ0, nesse intervalo, a probabilidade de obter uma média amostral mais distante de μ0 que x é maior que alfa; para qualquer média populacional, μ0, que não esteja nesse intervalo, a probabilidade de obter uma média amostral mais distante de μ0 que x é menor que alfa. Em outras palavras, suponha que usemos x, standard_dev e size para construir um teste bicaudal no nível de significância alfa da hipótese de que a média populacional é μ0. Em seguida, não rejeitaremos essa hipótese se μ0 estiver no intervalo de confiança e rejeitaremos essa hipótese se μ0 não estiver no intervalo de confiança. O intervalo de confiança não nos permite inferir que há a probabilidade 1 – alfa 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*(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.
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 alfa ≤ 0 ou ≥ 1, CONFIDENCE.NORM retornará o valor de erro #VALUE!.
- Se standard_dev ≤ 0, CONFIDENCE.NORM retornará o valor de erro #VALUE!.
- Se size não for um inteiro, ele será arredondado.
- Se size < 1, CONFIDENCE.NORM retornará o valor de erro #VALUE!.
- Se pressupormos que alfa seja igual a 0,05, precisaremos calcular a área sob a curva normal padrão que é igual a (1 - alfa) ou 95%. Esse valor é ± 1,96. Portanto, o intervalo de confiança é:

$$\overline{x} \pm 1.96 \bigg(\frac{\sigma}{\sqrt{n}} \bigg) $$

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

CONFIDENCE.T

17/03/2021 • 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*(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 alfa ≤ 0 ou ≥ 1, CONFIDENCE.T retornará o valor de erro #VALUE!.
- Se standard_dev ≤ 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!.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO

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

17/03/2021 • 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

17/03/2021 • 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:

  $$\text{COTH}(N) = \frac{1}{\text{TANH}(N)} = \frac{\text{COSH(N)}}{\text{SINH(N)}} = \frac{e^{N} + e^{-N}}{e^{N} - e^{-N}}$$

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A seguinte consulta DAX:

```
EVALUATE { COTH(2) }
```

Retorna

[VALOR]
1.03731472072755

CONTAGEM

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
Função COUNTAX
função COUNTX

COUNTA

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
Função COUNTAX
função COUNTX
Funções estatísticas

COUNTAX

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
Função COUNTA
função COUNTX

COUNTBLANK

17/03/2021 • 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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.

Confira também

Função COUNT

Função COUNTA
Função COUNTAX
função COUNTX
Funções estatísticas

COUNTROWS

17/03/2021 • 3 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo 1

O exemplo a seguir mostra como contar o número de linhas da tabela Orders. O resultado esperado é 52761.

```
= COUNTROWS('Orders')
```

Exemplo 2

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
Função COUNTA
Função COUNTAX
função COUNTX
Funções estatísticas

COUNTX

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
Função COUNTA
Função COUNTAX
Funções estatísticas

DISTINCTCOUNT

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

O exemplo a seguir mostra como contar o número de pedidos de vendas distintos na coluna ResellerSales_USD[SalesOrderNumber].

```
= DISTINCTCOUNT(ResellerSales_USD[SalesOrderNumber])
```

O uso da medida anterior em uma tabela com ano calendário na lateral e categoria de produto na parte superior retorna os seguintes resultados:

RÓTULOS DE LINHA	ACESSÓRIOS	BIKES	CLOTHING	COMPONENTES	-	GRANDE TOTAL
2005	135	345	242	205		366
2006	356	850	644	702		1015
2007	531	1234	963	1138		1521

RÓTULOS DE LINHA	ACESSÓRIOS	BIKES	CLOTHING	COMPONENTES	-	GRANDE TOTAL
2008	293	724	561	601		894
					1	1
Grande Total	**1315**	**3153**	**2410**	**2646**	**1**	**3797**

No exemplo acima, observe que os números das linhas em Grande Total não são somados. Isso acontece porque o mesmo pedido poderá conter itens de linha de categorias de produtos diferentes.

Confira também

Função COUNT
Função COUNTA
Função COUNTAX
função COUNTX
Funções estatísticas

EXPON.DIST

17/03/2021 • 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 ≤ 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:

  $$F(x; \lambda) = 1 - e^{-\lambda x}$$

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

GEOMEAN

17/03/2021 • 2 minutes to read

Retorna a média geométrica dos números em uma coluna.

Use a função GEOMEANX para retornar a média geométrica de uma expressão avaliada para cada linha em uma tabela.

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])
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

O seguinte computa a média geométrica da coluna Retorno na tabela Investimentos:

```
= GEOMEAN( Investment[Return] )
```

Consulte também

Função GEOMEANX

GEOMEANX

17/03/2021 • 2 minutes to read

Retorna a média geométrica de uma expressão avaliada para cada linha de uma tabela.

Use a função GEOMEAN para retornar a média geométrica de números em uma coluna.

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.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

O seguinte computa a média geométrica da coluna ReturnPct na tabela Investimentos:

```
= GEOMEANX( Investments, Investments[ReturnPct] + 1 )
```

Consulte também

Função GEOMEAN

MAX

17/03/2021 • 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.

Retornar valor

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

MAXA

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
Função MAXX
Funções estatísticas

MAXX

17/03/2021 • 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.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
Função MAXA
Funções estatísticas

MEDIAN

17/03/2021 • 2 minutes to read

Retorna a mediana dos números de uma coluna.

Use a função MEDIANX para retornar a mediana de uma expressão avaliada para cada linha em uma tabela.

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]).
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

MEDIANX)

17/03/2021 • 2 minutes to read

Retorna o número mediano de uma expressão avaliada para cada linha de uma tabela.

Use a função MEDIAN para retornar a mediana dos números em uma coluna.

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
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

O seguinte calcular a idade mediana dos clientes que moram nos EUA.

```
= MEDIANX( FILTER(Customers, RELATED( Geography[Country]="USA" ) ), Customers[Age] )
```

Consulte também

Função MEDIAN

MIN

17/03/2021 • 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
Função MINX
Funções estatísticas

MINA

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

MINX

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
função MINA
Funções estatísticas

NORM.DIST

17/03/2021 • 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.

Retornar valor

A distribuição normal para a média especificada e o desvio padrão.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

```
EVALUATE { NORM.DIST(42, 40, 1.5, TRUE) }
```

Retorna

[VALOR]
0,908788780274132

Confira também

Função NORM.S.DIST
Função NORM.INV
NORM.S.INV

NORM.INV

17/03/2021 • 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.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

```
EVALUATE { NORM.INV(0.908789, 40, 1.5) }
```

Retorna

[VALOR]
42,000002009566287802741 32

Confira também

NORM.S.INV
Função NORM.S.DIST
Função NORM.DIST

NORM.S.DIST

17/03/2021 • 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.

Retornar valor

A distribuição normal padrão (tem uma média igual a zero e um desvio padrão de um).

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

```
EVALUATE { NORM.S.DIST(1.333333, TRUE) }
```

Retorna

[VALOR]
0,908788725604095

Confira também

Função NORM.INV
Função NORM.DIST
NORM.S.INV

NORM.S.INV

17/03/2021 • 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.

Retornar valor

O inverso da distribuição cumulativa normal padrão. A distribuição tem uma média igual a zero e um desvio padrão de um.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

```
EVALUATE { NORM.S.INV(0.908789) }
```

Retorna

[VALOR]
1,33333467304411

Confira também

NORM.INV
Função NORM.S.DIST
Função NORM.DIST

PERCENTILE.EXC

17/03/2021 • 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.

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Consulte também

PERCENTILEX.EXC

PERCENTILE.INC

17/03/2021 • 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 PERCENTILEX.INC.

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Consulte também

PERCENTILEX.INC

PERCENTILEX.EXC

17/03/2021 • 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 a função PERCENTILE.EXC.

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Consulte também

PERCENTILE.EXC

PERCENTILEX.INC

17/03/2021 • 2 minutes to read

Retorna o número de percentil de uma expressão avaliada para cada linha de uma tabela.

Para retornar o percentil de números em uma coluna, use PERCENTILE.INC.

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Consulte também

PERCENTILE.INC

PERMUT

17/03/2021 • 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 ≤ 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:

  $$P_{k,n} = \frac{n!}{(n-k)!}$$

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

Na seguinte fórmula, as permutas possíveis para um grupo de 3 objetos em que 2 são escolhidos:

```
= PERMUT(3,2)
```

Resultado,

POISSON.DIST

17/03/2021 • 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:

    $$\text{POISSON} = \frac{e^{-\lambda} \lambda^{x}}{x!}$$

  - Para cumulative = TRUE:

    $$\text{CUMPOISSON} = \sum^{x}_{k=0} \frac{e^{-\lambda} \lambda^{x}}{k!}$$

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

RANK.EQ

17/03/2021 • 3 minutes to read

Retorna a classificação de um número em uma lista de números.

Sintaxe

```
RANK.EQ(<value>, <columnName>[, <order>])
```

Parâmetros

TERMO	DEFINIÇÃO
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 o valor dela deve ser 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 nem uma coluna criada usando estas funções: ADDCOLUMNS, ROW ou RESUME.
ordem	(Opcional) Um valor que especifica como classificar *number*, de menor para maior ou de maior para menor:

Valores de order

VALUE	VALOR ALTERNATIVO	DESCRIÇÃO
0 (zero)	FALSE	Classifica em ordem decrescente de *columnName*. Se *value* for igual ao maior número em *columnName*, **RANK.EQ** será 1.
1	TRUE	Classifica em ordem crescente de *columnName*. Se *value* for igual ao menor número em *columnName*, **RANK.EQ** será 1.

Retornar valor

Um número que indica a classificação de *value* entre os números em *columnName*.

Comentários

- *columnName* não pode fazer referência a nenhuma coluna criada usando essas funções: ADDCOLUMNS, ROW ou SUMMARIZE.I
- 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).
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo 1

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 2

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

17/03/2021 • 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

table
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.

value
(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 for 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

ENUMERAÇÃO	DESCRIÇÃO
Ignorar	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). É o valor padrão quando o parâmetro *ties* é omitido.
Denso	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.

Valor retornado

O número de classificação de *value* entre todos os valores possíveis de *expression* avaliados para todas as linhas dos números de *table*.

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")
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

SAMPLE

17/03/2021 • 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

TERMO	DEFINIÇÃO
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.
table	Qualquer expressão DAX que retorna uma tabela de dados das quais extrair 'n' linhas de amostra.
orderBy_expression	(Opcional) Qualquer expressão DAX escalar em que o valor de resultado seja avaliado para cada linha de *table*.
ordem	(Opcional) Um valor que especifica como classificar valores *orderBy_expression* em ordem crescente ou decrescente: 0 (zero), classifica em ordem decrescente os valores de *order_by*. 1, classifica em ordem crescente *order_by*.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

SIN

17/03/2021 • 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 um argumento estiver em graus, multiplique-o por PI()/180 ou use a função RADIANS para convertê-lo em radianos.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
= SIN(PI())	Seno de Pi radianos (aproximadamente 0).	0,0
= SIN(PI()/2)	Seno de Pi/2 radianos.	1.0
= SIN(30*PI()/180)	Seno de 30 graus.	0,5
= SIN(RADIANS(30))	Seno de 30 graus.	0,5

SINH

17/03/2021 • 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:

  $$\texto{SINH}(z) = \frac{e^{z} - e^{-z}}{2}$$

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

Probabilidade de obter um resultado de menor de 1,03 segundos.

```
= 2.868*SINH(0.0342\*1.03)
```

Retorna 0.1010491

STDEV.S

17/03/2021 • 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

- 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.
- STDEV.S usa a seguinte fórmula:

  Ã[∑(x – $\tilde{x}$)^2/(n-1)]

  em que $\tilde{x}$ é o valor médio de x para a população da amostra e n é o tamanho da população.
- As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.
- Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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.

Comentários

- 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.
- STDEV.P usa a seguinte fórmula:

  $\tilde{A}[\sum(x - \tilde{x})^2/n]$

  em que $\tilde{x}$ é o valor médio de x para a população inteira e n é o tamanho da população.
- As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.
- Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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

- 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.
- STDEVX.S usa a seguinte fórmula:

  Ã[∑(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.
- As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.
- Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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

- 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.
- STDEVX.P usa a seguinte fórmula:

  $\tilde{A}[\sum(x - \tilde{x})^2/n]$

  em que $\tilde{x}$ é o valor médio de x para a população inteira e n é o tamanho da população.
- As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.
- Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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).

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
= SQRTPI(1)	Raiz quadrada de pi.	1,772454
= SQRTPI(2)	Raiz quadrada de 2 * pi.	2,506628

T.DIST

17/03/2021 • 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.

Retornar valor

A distribuição t do teste unilateral à esquerda do Student.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

```
EVALUATE { T.DIST(60, 1, TRUE) }
```

Retorna:

[VALOR]
0,994695326367377

Confira também

T.DIST.2T
T.DIST.RT
T.INV
T.INV.2t

T.DIST.2T

17/03/2021 • 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.

Retornar valor

A distribuição t do Student bicaudal.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

```
EVALUATE { T.DIST.2T(1.959999998, 60) }
```

Retorna

[VALOR]
0,054644929975921

Confira também

T.DIST
T.DIST.RT
T.INV
T.INV.2t

T.DIST.RT

17/03/2021 • 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.

Retornar valor

O teste unilateral à direita da distribuição t do Student.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

```
EVALUATE { T.DIST.RT(1.959999998, 60) }
```

Retorna

[VALOR]
0,0273224649879605

Confira também

T.DIST
T.DIST.2T
T.INV
T.INV.2t

T.INV

17/03/2021 • 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.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

```
EVALUATE { T.INV(0.75, 2) }
```

Retorna

[VALOR]
0,816496580927726

Confira também

T.INV.2T
T.DIST
T.DIST.2T
T.DIST.RT

T.INV.2T

17/03/2021 • 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.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

```
EVALUATE { T.INV.2T(0.546449, 60) }
```

Retorna

[VALOR]
0,606533075825759

Confira também

T.INV
T.DIST
T.DIST.2T
T.DIST.RT

TAN

17/03/2021 • 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 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

17/03/2021 • 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:

  $$\texto{TANH}(z) = \frac{\text{SINH}(z)}{\text{COSH}(z)}$$

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

VAR.S

17/03/2021 • 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.

Comentários

- 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.
- 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
- As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.
- Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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

- 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.
- 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
- As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.
- Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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.

Comentários

- 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.
- 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
- As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.
- Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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.

Comentários

- 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.
- 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
- As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.
- Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Funções de manipulação de tabelas

17/03/2021 • 5 minutes to read

Essas funções retornam uma tabela ou manipulam tabelas existentes.

Nesta categoria

FUNÇÃO	DESCRIÇÃO
ADDCOLUMNS	Adiciona colunas calculadas à tabela ou à expressão de tabela fornecida.
ADDMISSINGITEMS	Adiciona combinações de itens de várias colunas a uma tabela caso ainda não existam.
CROSSJOIN	Retorna uma tabela que contém o produto cartesiano de todas as linhas de todas as tabelas nos argumentos.
CURRENTGROUP	Retorna um conjunto de linhas do argumento de tabela de uma expressão GROUPBY.
DATATABLE	Fornece um mecanismo para declarar um conjunto embutido de valores de dados.
DETAILROWS	Avalia uma Expressão das Linhas de Detalhes definida para uma medida e retorna os dados.
Coluna DISTINCT	Retorna uma tabela de coluna única que contém os valores distintos da coluna especificada.
Tabela DISTINCT	Retorna uma tabela removendo linhas duplicadas de outra tabela ou expressão.
EXCEPT	Retorna as linhas de uma tabela que não aparecem em outra tabela.
FILTERS	Retorna uma tabela de valores aplicados diretamente como filtros a *columnName*.
GENERATE	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*.
GENERATEALL	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*.
GENERATESERIES	Retorna uma tabela de coluna única que contém os valores de uma série aritmética.

FUNÇÃO	DESCRIÇÃO
GROUPBY	De maneira semelhante à função SUMMARIZE, GROUPBY não executa um CALCULATE implícito em nenhuma coluna de extensão que adiciona.
IGNORE	Modifica SUMMARIZECOLUMNS omitindo expressões específicas da avaliação BLANK/NULL.
INTERSECT	Retorna a interseção de linha entre duas tabelas, retendo as duplicatas.
NATURALINNERJOIN	Executa uma junção interna de uma tabela com outra tabela.
NATURALLEFTOUTERJOIN	Executa uma junção interna de uma tabela com outra tabela.
ROLLUP	Modifica o comportamento de SUMMARIZE adicionando linhas de valores acumulados ao resultado nas colunas definidas pelo parâmetro groupBy_columnName.
ROLLUPADDISSUBTOTAL	Modifica o comportamento de SUMMARIZECOLUMNS adicionando linhas de valores acumulados/subtotal ao resultado com base nas colunas groupBy_columnName.
ROLLUPISSUBTOTAL	Emparelha grupos de valores acumulados com a coluna adicionada por ROLLUPADDISSUBTOTAL dentro de uma expressão ADDMISSINGITEMS.
ROLLUPGROUP	Modifica o comportamento de SUMMARIZE e SUMMARIZECOLUMNS adicionando linhas de valores acumulados ao resultado nas colunas definidas pelo parâmetro groupBy_columnName.
ROW	Retorna uma tabela com uma única linha contendo valores que resultam das expressões fornecidas para cada coluna.
SELECTCOLUMNS	Adiciona colunas calculadas à tabela ou à expressão de tabela fornecida.
SUBSTITUTEWITHINDEX	Retorna uma tabela que representa uma semijunção à esquerda das duas tabelas fornecidas como argumentos.
SUMMARIZE	Retorna uma tabela de resumo para os totais solicitados sobre um conjunto de grupos.
SUMMARIZECOLUMNS	Retorna uma tabela de resumo por um conjunto de grupos.
Construtor de Tabela	Retorna uma tabela de uma ou mais colunas.
TOPN	Retorna as N linhas superiores da tabela especificada.
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.

FUNÇÃO	DESCRIÇÃO
VALUES	Retorna uma tabela de coluna única que contém os valores distintos da tabela ou coluna especificada.

ADDCOLUMNS

17/03/2021 • 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 *tabela*.

Retornar valor

Uma tabela com todas as suas colunas originais e as adicionadas.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	PRODUCTCATEGORY[PRODUCTCATEGORYALTERNATEKEY]	PRODUCTCATEGORY[PRODUCTCATEGORYKEY]	[INTERNET SALES]	[VENDAS DO REVENDEDOR]
Clothing	3	3	306157,5829	1669943,267
Acessórios	4	4	640920,1338	534301,9888

ADDMISSINGITEMS

17/03/2021 • 2 minutes to read

Adiciona linhas com valores vazios a uma tabela retornada por SUMMARIZECOLUMNS.

Sintaxe

```
ADDMISSINGITEMS ( [<showAll_columnName> [, <showAll_columnName> [, … ] ] ], <table> [, <groupBy_columnName>
[, [<filterTable>] [, <groupBy_columnName> [, [<filterTable>] [, … ] ] ] ] ] )
```

Parâmetros

TERMO	DEFINIÇÃO
showAll_columnName	(Opcional) Uma coluna para a qual retornar itens sem dados para as medidas usadas. Se não for especificada, todas as colunas serão retornadas.
table	Uma tabela de SUMMARIZECOLUMNS.
groupBy_columnName	(Opcional) Uma coluna segundo a qual fazer o agrupamento no argumento de tabela fornecido.
filterTable	(Opcional) Uma expressão de tabela que define quais linhas são retornadas.

Retornar valor

Uma tabela com uma ou mais colunas.

Comentários

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Com SUMMARIZECOLUMNS

Uma tabela retornada por SUMMARIZECOLUMNS incluirá somente linhas com valores. Ao encapsular uma expressão SUMMARIZECOLUMNS dentro de uma expressão ADDMISSINGITEMS, linhas que não contêm nenhum valor também serão retornadas.

Exemplo

Sem ADDMISSINGITEMS, a seguinte consulta:

```
SUMMARIZECOLUMNS(
    'Sales'[CustomerId],
    "Total Qty", SUM ( Sales[TotalQty] )
)
```

Retorna:

CUSTOMERID	TOTALQTY
Um	5
B	3
C	3
E	2

Com ADDMISSINGITEMS, a seguinte consulta:

```
EVALUATE
ADMISSINGITEMS (
    'Sales'[CustomerId],
    SUMMARIZECOLUMNS(
        'Sales'[CustomerId],
        "Total Qty", SUM ( Sales[TotalQty] )
    ),
    'Sales'[CustomerId]
)
```

Retorna:

CUSTOMERID	TOTALQTY
Um	5
B	3
C	3
D	
E	2
F	

CROSSJOIN

17/03/2021 • 3 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

Retornar valor

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**.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

CURRENTGROUP

17/03/2021 • 2 minutes to read

Retorna um conjunto de linhas do argumento de tabela de uma expressão GROUPBY que pertencem à linha atual do resultado de GROUPBY.

Sintaxe

```
CURRENTGROUP ( )
```

Parâmetros

Nenhum

Retornar valor

As linhas no argumento de tabela da função GROUPBY correspondentes a um grupo de valores dos argumentos groupBy_columnName.

Comentários

- Essa função pode ser usada apenas dentro de uma expressão GROUPBY.
- Essa função 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

Confira GROUPBY.

DATATABLE

17/03/2021 • 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.
Tipo de dados	Uma enumeração que inclui: INTEGER, DOUBLE, STRING, BOOLEAN, CURRENCY, DATETIME
Valor	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 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. 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. 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}

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança

em nível de linha) ou colunas calculadas.

Exemplo

```
= DataTable("Name", STRING,
               "Region", STRING
               ,{
                        {" User1","East"},
                        {" User2","East"},
                        {" User3","West"},
                        {" User4","West"},
                        {" User4","East"}
                }
           )
```

DETAILROWS

17/03/2021 • 2 minutes to read

Avalia uma Expressão das Linhas de Detalhes definida para uma medida e retorna os dados.

Sintaxe

```
DETAILROWS([Measure])
```

Parâmetros

TERMO	DEFINIÇÃO
Medida	Nome de uma medida.

Retornar valor

Uma tabela com os dados retornados pela Expressão das Linhas de Detalhes. Se nenhuma Expressão das Linhas de Detalhes for definida, os dados da tabela que contém a medida serão retornados.

DISTINCT (coluna)

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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 a função VALUES.

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.

RÓTULOS DE LINHA	ACESSÓRIOS	BIKES	CLOTHING	GRANDE TOTAL
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
Função FILTER
função RELATED
Função VALUES

DISTINCT (tabela)

17/03/2021 • 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	Um
2	B

Consulte também

Funções de filtro
DISTINCT (coluna)
Função FILTER
função RELATED
Função VALUES

EXCEPT

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

States1

ESTADO
Um

ESTADO
B
B
B
C
D
D

States2

ESTADO
B
C
D
D
D
E
E
E

Except(States1, States2)

ESTADO
Um

Except(States2, States1)

ESTADO
E
E
E

FILTERS

17/03/2021 • 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

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

O exemplo a seguir mostra como determinar o número de filtros diretos que uma coluna tem.

```
= COUNTROWS(FILTERS(ResellerSales_USD[ProductKey]))
```

Este exemplo permite saber quantos filtros diretos em ResellerSales_USD[ProductKey] foram aplicados ao contexto em que a expressão está sendo avaliada.

GENERATE

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

SALESTERRITORY[SALESTERRITORYGROUP]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VENDAS DO REVENDEDOR]
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

A seguinte fórmula produzirá 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

PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VENDAS DO REVENDEDOR]
Vestuário	US$ 1.669.943,27
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

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

SALESTERRITORY[SALESTERRITORYGROUP]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VENDAS DO REVENDEDOR]
NA	Acessórios	
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

A seguinte fórmula produzirá 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
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.

GENERATESERIES

17/03/2021 • 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.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE GENERATESERIES(1, 5)
```

Retorna a tabela a seguir com uma única coluna:

[VALOR]
1

[VALOR]
2
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

17/03/2021 • 5 minutes to read

A função GROUPBY é semelhante à função SUMMARIZE. No entanto, GROUPBY não executa um CALCULATE 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 é usado para executar várias agregações em uma só verificação de tabela.

Sintaxe

```
GROUPBY (<table> [, <groupBy_columnName> [, <groupBy_columnName> [, …]]] [, <name>, <expression> [, <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	Uma das funções de agregação X, com o primeiro argumento sendo CURRENTGROUP(). Confira na seção Com CURRENTGROUP a seguir a lista completa de funções de agregação X com suporte.

Retornar valor

Uma tabela com as colunas selecionadas para os argumentos groupBy_columnName e as colunas de extensão 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. Diferente da função SUMMARIZE, um CALCULATE implícito não é executado e o grupo não é colocado no contexto do filtro.
- 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.
- GROUPBY é usado principalmente para executar agregações de resultados intermediários de expressões de tabela DAX. Para agregações eficientes em tabelas físicas no modelo, considere usar a função SUMMARIZECOLUMNS ou SUMMARIZE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Com CURRENTGROUP

CURRENTGROUP pode ser usado somente em uma expressão que define uma coluna de extensão dentro da função GROUPBY. Em vigor, CURRENTGROUP retorna um conjunto de linhas do argumento de tabela de GROUPBY que pertencem à linha atual do resultado de 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

O exemplo a seguir calcula primeiro o total de vendas agrupadas por país e por categoria de produto em tabelas físicas usando a função SUMMARIZECOLUMNS. Em seguida, ele usa a função GROUPBY para verificar o resultado intermediário da primeira etapa para encontrar o máximo de vendas em cada país entre as categorias de produto.

```
DEFINE
VAR SalesByCountryAndCategory =
SUMMARIZECOLUMNS(
Geography[Country],
Product[Category],
"Total Sales", SUMX(Sales, Sales[Price] * Sales[Qty])
)

EVALUATE
GROUPBY(
SalesByCountryAndCategory,
Geography[Country],
"Max Sales", MAXX(CURRENTGROUP(), [Total Sales])
)
```

Confira também

Função SUMMARIZE
Função SUMMARIZECOLUMNS

IGNORE

17/03/2021 • 2 minutes to read

Modifica 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. Essa função pode ser usada apenas dentro de uma expressão SUMMARIZECOLUMNS.

Syntax

```
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).

Retornar valor

A função não retorna um valor.

Comentários

IGNORE pode ser usado somente como um argumento de expressão para SUMMARIZECOLUMNS.

Exemplo

Confira SUMMARIZECOLUMNS.

INTERSECT

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

States1

ESTADO
A
Um
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
C
D
D
D

NATURALINNERJOIN

17/03/2021 • 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.

Retornar valor

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[ProductdID], StoreSales[ProductdID], 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

NATURALLEFTOUTERJOIN

17/03/2021 • 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.

Retornar valor

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[ProductdID], StoreSales[ProductdID], 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

ROLLUP

17/03/2021 • 2 minutes to read

Modifica o comportamento da função SUMMARIZE adicionando linhas de valores acumulados ao resultado nas colunas definidas pelo parâmetro groupBy_columnName. Essa função pode ser usada apenas dentro de uma expressão SUMMARIZE.

Syntax

```
ROLLUP ( <groupBy_columnName> [, <groupBy_columnName> [, … ] ] )
```

Com SUMMARIZE,

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]…[, ROLLUP(<groupBy_columnName>[,<
groupBy_columnName>…])][, <name>, <expression>]…)
```

Parâmetros

TERMO	DEFINIÇÃO
groupBy_columnName	O nome qualificado de uma coluna existente ou da função ROLLUPGROUP a ser usada para criar grupos de resumo com base nos valores encontrados nela. Esse parâmetro não pode ser uma expressão.

Retornar valor

Essa função não retorna um valor. Ele especifica apenas o conjunto de colunas a ser subtotalizado.

Comentários

Essa função pode ser usada apenas dentro de uma expressão SUMMARIZE.

Exemplo

Confira SUMMARIZE.

ROLLUPADDISSUBTOTAL

17/03/2021 • 2 minutes to read

Modifica o comportamento da função SUMMARIZECOLUMNS adicionando linhas de valores acumulados/subtotal ao resultado com base nas colunas groupBy_columnName. Essa função pode ser usada apenas dentro de uma expressão SUMMARIZECOLUMNS.

Sintaxe

```
ROLLUPADDISSUBTOTAL ( [<grandtotalFilter>], <groupBy_columnName>, <name> [, [<groupLevelFilter>] [,
<groupBy_columnName>, <name> [, [<groupLevelFilter>] [, … ] ] ] ] )
```

Parâmetros

TERMO	DEFINIÇÃO
grandtotalFilter	(Opcional) Filtro a ser aplicado ao nível do total geral.
groupBy_columnName	Nome de uma coluna existente usada para criar grupos de resumo com base nos valores encontrados nela. Não pode ser uma expressão.
name	Nome de uma coluna ISSUBTOTAL. Os valores da coluna são calculados usando a função ISSUBTOTAL.
groupLevelFilter	(Opcional) Filtro a ser aplicado ao nível atual.

Retornar valor

A função não retorna um valor.

Comentários

Nenhum

Exemplo

Confira SUMMARIZECOLUMNS.

ROLLUPGROUP

17/03/2021 • 2 minutes to read

Modifica o comportamento das funções SUMMARIZE e SUMMARIZECOLUMNS adicionando linhas de valores acumulados ao resultado nas colunas definidas pelo parâmetro groupBy_columnName. Essa função pode ser usada apenas dentro de uma expressão SUMMARIZE ou SUMMARIZECOLUMNS.

Sintaxe

```
ROLLUPGROUP ( <groupBy_columnName> [, <groupBy_columnName> [, … ] ] )
```

Parâmetros

TERMO	DEFINIÇÃO
groupBy_columnName	O nome qualificado de uma coluna existente ou da função ROLLUPGROUP a ser usada para criar grupos de resumo com base nos valores encontrados nela. Esse parâmetro não pode ser uma expressão.

Retornar valor

Essa função não retorna um valor. Ela marca um conjunto de colunas a serem tratadas como um só grupo durante o cálculo de subtotal por ROLLUP ou ROLLUPADDISSUBTOTAL.

Comentários

ROLLUPGROUP pode ser usado somente como um argumento groupBy_columnName para ROLLUP, ROLLUPADDISSUBTOTAL ou ROLLUPISSUBTOTAL.

Exemplo

Confira SUMMARIZE e SUMMARIZECOLUMNS.

ROLLUPISSUBTOTAL

17/03/2021 • 2 minutes to read

Emparelha grupos de acúmulo com a coluna adicionada por ROLLUPADDISSUBTOTAL. Essa função pode ser usada apenas dentro de uma expressão ADDMISSINGITEMS.

Sintaxe

```
ROLLUPISSUBTOTAL ( [<grandTotalFilter>], <groupBy_columnName>, <isSubtotal_columnName> [,
[<groupLevelFilter>] [, <groupBy_columnName>, <isSubtotal_columnName> [, [<groupLevelFilter>] [, … ] ] ] ] )
```

Parâmetros

TERMO	DEFINIÇÃO
grandTotalFilter	(Opcional) Filtro a ser aplicado ao nível do total geral.
groupBy_columnName	Nome de uma coluna existente usada para criar grupos de resumo com base nos valores encontrados nela. Não pode ser uma expressão.
isSubtotal_columnName	Nome de uma coluna ISSUBTOTAL. Os valores da coluna são calculados usando a função ISSUBTOTAL.
groupLevelFilter	(Opcional) Filtro a ser aplicado ao nível atual.

Retornar valor

Nenhum

Comentários

Essa função pode ser usada apenas dentro de uma expressão ADDMISSINGITEMS.

Função ROW

17/03/2021 • 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. *name*.

Retornar valor

Uma tabela com uma única linha

Comentários

- Os argumentos sempre devem vir em pares de *name* e *expression*.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

SELECTCOLUMNS

17/03/2021 • 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 <name>, argumentos <scalar_expression> e cada expressão é avaliada no contexto de uma linha do argumento <table> especificado.

Comentários

SELECTCOLUMNS tem a mesma assinatura e comportamento que ADDCOLUMNS, com a exceção de que, em vez de começar com a <table> especificada, SELECTCOLUMNS começa com uma tabela vazia antes da adição de colunas.

Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

PAÍS	ESTADO	CONTAGEM	TOTAL
EUA	WA	10	900

```
SELECTCOLUMNS(Info, "StateCountry", [State]&", "&[Country])
```

Retorna:

STATECOUNTRY
IND, JK
IND, MH
IND, WB
USA, CA
USA, WA

SUBSTITUTEWITHINDEX

17/03/2021 • 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.
ordem	(Opcional) Um valor que especifica como classificar valores orderBy_expression em ordem crescente ou decrescente: 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. Valor: **ASC**. Valor alternativo: **1**/**TRUE**. Classifica em ordem crescente de orderBy_expression.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

SUMMARIZE

17/03/2021 • 8 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 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

- 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 a *table*.
- 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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	Bikes	12968255,42	36167,6592
2005	Bikes	6958251,043	4231,1621
2006	Bikes	18901351,08	178175,8399
2007	Bikes	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	Clothing	366507,844	4151,1235
2005	Clothing	31851,1628	90,9593
2006	Clothing	455730,9729	4233,039
2007	Clothing	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

Com ROLLUP

A adição da sintaxe ROLLUP modifica o comportamento da função SUMMARIZE ao acrescentar linhas de acúmulo ao resultado nas colunas groupBy_columnName. ROLLUP pode ser usada somente dentro de uma expressão SUMMARIZE.

Exemplo

O seguinte exemplo adiciona linhas de valores acumulados 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])
)
```

Retorna a tabela a seguir,

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
2008	Bikes	12968255,42	36167,6592
2005	Bikes	6958251,043	4231,1621
2006	Bikes	18901351,08	178175,8399
2007	Bikes	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	Clothing	366507,844	4151,1235
2005	Clothing	31851,1628	90,9593
2006	Clothing	455730,9729	4233,039
2007	Clothing	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
2008		15496115,89	41224,3038
2005		7582953,67	4326,4144
2006		22871907,85	184419,1335

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
2007		30543780,84	297538,0745
		76494758,25	527507,9262

Com ROLLUPGROUP

A adição de ROLLUPGROUP dentro de uma sintaxe ROLLUP pode ser usada para impedir subtotais parciais em linhas de acúmulo. ROLLUPGROUP pode ser usado somente dentro de uma expressão ROLLUP, ROLLUPADDISSUBTOTAL ou ROLLUPISSUBTOTAL.

Exemplo

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])
)
```

Retorna a tabela a seguir,

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
2008	Bikes	12968255,42	36167,6592
2005	Bikes	6958251,043	4231,1621
2006	Bikes	18901351,08	178175,8399
2007	Bikes	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	Clothing	366507,844	4151,1235
2005	Clothing	31851,1628	90,9593
2006	Clothing	455730,9729	4233,039
2007	Clothing	815853,2868	12489,3835
2008	Acessórios	153299,924	865,5945

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
2005	Acessórios	18594,4782	4,293
2006	Acessórios	86612,7463	1061,4872
2007	Acessórios	275794,8403	4756,6546
		76494758,25	527507,9262

Com ISSUBTOTAL

Com ISSUBTOTAL, você poderá criar outra coluna na expressão SUMMARIZE que retornará True se a linha contiver valores de subtotal para a coluna fornecida como argumento para ISSUBTOTAL; caso contrário, retornará False. ISSUBTOTAL pode ser usada somente dentro de uma expressão SUMMARIZE.

Exemplo

O seguinte exemplo 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])
)
```

Retorna a tabela a seguir,

[É O SUBTOTAL PARA DATETIMECALENDARYEAR]	[É O SUBTOTAL PARA PRODUCTCATEGORYNAME]	DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
FALSE	FALSE				
FALSE	FALSE	2008	Bikes	12968255,42	36167,6592
FALSE	FALSE	2005	Bikes	6958251,043	4231,1621
FALSO	FALSE	2006	Bikes	18901351,08	178175,8399
FALSE	FALSE	2007	Bikes	24256817,5	276065,992
FALSE	FALSE	2008	Componentes	2008052,706	39,9266
FALSE	FALSE	2005	Componentes	574256,9865	0
FALSE	FALSE	2006	Componentes	3428213,05	948,7674
FALSE	FALSE	2007	Componentes	5195315,216	4226,0444

[É O SUBTOTAL PARA DATETIMECALENDARYEAR]	[É O SUBTOTAL PARA PRODUCTCATEGORYNAME]	DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
FALSE	FALSE	2008	Clothing	366507,844	4151,1235
FALSE	FALSE	2005	Clothing	31851,1628	90,9593
FALSE	FALSE	2006	Clothing	455730,9729	4233,039
FALSE	FALSE	2007	Clothing	815853,2868	12489,3835
FALSE	FALSE	2008	Acessórios	153299,924	865,5945
FALSE	FALSE	2005	Acessórios	18594,4782	4,293
FALSE	FALSE	2006	Acessórios	86612,7463	1061,4872
FALSE	FALSE	2007	Acessórios	275794,8403	4756,6546
FALSE	TRUE				
FALSE	TRUE	2008		15496115,89	41224,3038
FALSE	TRUE	2005		7582953,67	4326,4144
FALSE	TRUE	2006		22871907,85	184419,1335
FALSE	TRUE	2007		30543780,84	297538,0745
TRUE	TRUE			76494758,25	527507,9262

Confira também

SUMMARIZECOLUMNS

SUMMARIZECOLUMNS

17/03/2021 • 9 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

- Essa função 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, a fórmula a seguir é inválida.

  ```
  SUMMARIZECOLUMNS( Sales[StoreId], Sales[StoreId] )
  ```

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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 FILTER (por exemplo, da tabela Customer). O filtro não é aplicado às colunas groupBy. As tabelas Sales Territory e Customer podem estar indiretamente relacionadas por meio da tabela de fatos Reseller sales. 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.

Com 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, quer as expressões que usam IGNORE sejam avaliadas como BLANK/NULL ou não. IGNORE pode ser usada somente dentro de uma expressão 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
Um	5	
B	3	3
C	3	3

Com IGNORE,

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
B	3	3

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
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 um valor em branco para algumas linhas, elas são incluídas, já que não há nenhuma expressão não ignorada que retorne em branco.

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
Um		5
B		
C		

Com NONVISUAL

A função NONVISUAL marca que um filtro de valor na função SUMMARIZECOLUMNS não afeta os valores da medida, mas aplica-se apenas às colunas groupBy. NONVISUAL pode ser usada somente dentro de uma expressão SUMMARIZECOLUMNS.

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]
```

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]
```

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

Com ROLLUPADDISSUBTOTAL

A adição da sintaxe ROLLUPADDISSUBTOTAL modifica o comportamento da função SUMMARIZECOUMNS adicionando linhas de acúmulo/subtotal ao resultado com base nas colunas groupBy_columnName. ROLLUPADDISSUBTOTAL pode ser usada somente dentro de uma expressão SUMMARIZECOLUMNS.

Exemplo com apenas um subtotal

```
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]
```

Retorna a tabela a seguir,

CATEGORIA	SUBCATEGORIA	ISCATEGORYSUBTOTAL	ISSUBCATEGORYSUBTOTAL	TOTAL QTY
		verdadeiro	verdadeiro	60398
Acessórios		Falso	verdadeiro	36092
Acessórios	Racks de bicicleta	Falso	Falso	328

CATEGORIA	SUBCATEGORIA	ISCATEGORYSUBTOTAL	ISSUBCATEGORYSUBTOTAL	TOTAL QTY
Bikes	Mountain bikes	Falso	Falso	4970
Clothing		Falso	verdadeiro	9101

Exemplo com vários subtotais

```
SUMMARIZECOUMNS (
    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.

Retorna a tabela a seguir,

CUSTOMERID	ISCUSTOMERSUBTOTAL	ESTADO	TOTAL QTY	DATA	ISDATESUBTOTAL
Um	FALSE	WA	5	10/7/2014	
B	FALSE	WA	1	10/7/2014	
B	FALSE	WA	2	11/7/2014	
C	FALSE	OU	2	10/7/2014	
C	FALSE	OU	1	11/7/2014	
	TRUE	WA	6	10/7/2014	
	TRUE	WA	2	11/7/2014	
	TRUE	OU	2	10/7/2014	
	TRUE	OU	1	11/7/2014	
Um	FALSE	WA	5		TRUE
B	FALSE	WA	3		TRUE
C	FALSE	OU	3		VERDADEIRO
	TRUE	WA	8		VERDADEIRO
	TRUE	OU	3		TRUE

Com ROLLUPGROUP

Assim como com a função SUMMARIZE, ROLLUPGROUP pode ser usado com ROLLUPADDISSUBTOTAL para especificar quais grupos/granularidades de resumo (subtotais) incluir, reduzindo o número de linhas de subtotal

retornadas. ROLLUPGROUP pode ser usada apenas dentro de uma expressão SUMMARIZECOLUMNS ou SUMMARIZE.

Exemplo com vários subtotais

```
SUMMARIZECOLUMNS(
    ROLLUPADDISSUBTOTAL( Sales[CustomerId], "IsCustomerSubtotal" ),
    ROLLUPADDISSUBTOTAL(ROLLUPGROUP(Regions[City], Regions[State]), "IsCityStateSubtotal"),"Total Qty", SUM(
Sales[Qty] )
)
```

Ainda agrupados por City e State, mas acumulados ao relatar um subtotal, retornando a tabela a seguir,

ESTADO	CUSTOMERID	ISCUSTOMERSUBTOTAL	TOTAL QTY	CITY	ISCITYSTATESUBTOTAL
WA	Um	FALSE	2	Bellevue	FALSE
WA	B	FALSE	2	Bellevue	FALSE
WA	Um	FALSE	3	Redmond	FALSE
WA	B	FALSE	1	Redmond	FALSE
OU	C	FALSE	3	Portland	FALSE
WA		TRUE	4	Bellevue	FALSE
WA		TRUE	4	Redmond	FALSE
OU		TRUE	3	Portland	FALSE
	Um	FALSE	5		FALSO
	B	FALSE	3		TRUE
	C	FALSE	3		TRUE
		TRUE	11		TRUE

Confira também

SUMMARIZE

Construtor de tabela

17/03/2021 • 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.

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
2

[VALOR]
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")
    }
```

Retorna:

[VALUE1]	[VALUE2]	[VALUE3]	[VALUE4]
1.5	01/01/2017	199,99	A
2.5	02/01/2017	249,99	B
3,5	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

TOPN

17/03/2021 • 2 minutes to read

Retorna as N linhas superiores da tabela especificada.

Sintaxe

```
TOPN(<n_value>, <table>, <orderBy_expression>, [<order>[, <orderBy_expression>, [<order>]]…])
```

Parâmetros

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 os valores de *orderBy_expression* em ordem crescente ou decrescente:

VALUE	VALOR ALTERNATIVO	DESCRIÇÃO
0 (zero)	FALSE	Classifica em ordem decrescente de valores de *order_by*. É o valor padrão quando o parâmetro *order* é omitido.
1	TRUE	Classifica em ordem crescente de *order_by*.

Retornar valor

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 enésima 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.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

TREATAS

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

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

INTERSECT
FILTER
USERELATIONSHIP

UNION

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

VALUES

17/03/2021 • 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, os valores exclusivos 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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
Função COUNTROWS
Funções de filtro

Funções de texto

17/03/2021 • 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
COMBINEVALUES	une duas cadeias de cadeia de caracteres de texto em uma.
CONCATENATE	Une duas cadeias de texto em uma.
CONCATENATEX	Concatena o resultado de uma expressão avaliada para cada linha de uma tabela.
EXACT	Compara duas cadeias de texto e retorna TRUE se elas são exatamente iguais; caso contrário, FALSE.
FIND	Retorna a posição inicial de uma cadeia de texto em outra cadeia de texto.
FIXED	Arredonda um número para o número especificado de decimais e retorna o resultado como texto.
FORMAT	Converte um valor em texto de acordo com o formato especificado.
LEFT	Retorna o número especificado de caracteres do início de uma cadeia de texto.
LEN	Retorna o número de caracteres em uma cadeia de texto.
LOWER	Converte todas as letras de uma cadeia de texto em minúsculas.
MID	Retorna uma cadeia de caracteres do meio de uma cadeia de texto, dados um ponto inicial e um comprimento.
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.

FUNÇÃO	DESCRIÇÃO
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.

COMBINEVALUES

17/03/2021 • 4 minutes to read

une duas cadeias de cadeia de caracteres de texto em uma. 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
Maio 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

17/03/2021 • 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 boolianos 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo: concatenação de literais

A fórmula de exemplo cria um novo valor de cadeia de caracteres combinando dois valores de cadeia de caracteres que você fornece como argumentos.

```
= CONCATENATE("Hello ", "World")
```

Exemplo: concatenação de cadeias de caracteres em colunas

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.

```
= CONCATENATE(Customer[LastName], CONCATENATE(", ", Customer[FirstName]))
```

Exemplo: concatenação condicional de cadeias de caracteres em colunas

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.

```
= CONCATENATE( [FirstName]&" ", CONCATENATE( IF( LEN([MiddleName])>1, LEFT([MiddleName],1)&" ", ""),
[LastName]))
```

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	ABREVIAÇÃO 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

```
= CONCATENATE('Products'[Product abbreviation],'Products'[Product number])
```

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]
```

Confira também

Funções de texto

CONCATENATEX

17/03/2021 • 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

Tabela Funcionários

FIRSTNAME	LASTNAME
Alan	Brewer
Michael	Blythe

A seguinte fórmula:

```
CONCATENATEX(Employees, [FirstName] & " " & [LastName], ",")
```

Retorna:

"Alan Brewer, Michael Blythe"

EXACT

17/03/2021 • 2 minutes to read

Compara duas cadeias de texto e retorna TRUE se elas são exatamente iguais; caso contrário, FALSE. EXATO 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.

Retornar valor

True ou false. (Booliano)

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

FIND

17/03/2021 • 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 **within_text**.
within_text	O texto que contém o texto que você deseja encontrar.
start_num	(opcional) O caractere no qual iniciar a pesquisa; se omitido, **start_num** = 1. O primeiro caractere em **within_text** 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().

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.
- 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")
```

Confira também

Funções de texto

FIXED

17/03/2021 • 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.

Retornar valor

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 está função converte seu resultado em texto. Um número formatado com um comando no menu de formatação ainda é um número.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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.

Confira também

Funções de texto
Funções matemáticas e trigonométricas

FORMAT

17/03/2021 • 27 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**.

> **NOTE**
>
> Caso o **valor** seja 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 de dados de **value**).

Comentários

- Cadeias de caracteres de formato predefinido usam a propriedade de cultura do modelo ao formatar o resultado. Por padrão, a propriedade de cultura do modelo é definida de acordo com a localidade do usuário do computador. Para novos modelos do Power BI Desktop, a propriedade de cultura do modelo pode ser alterada em Opções > Configurações regionais > Idioma do modelo. Para o Analysis Services, a cultura do modelo é definida de acordo com a propriedade de idioma definida inicialmente pela instância.
- 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 "Percentual". 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 processadas como parte de uma cadeia de caracteres de formato personalizada ou como um literal de cadeia de caracteres.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplos

```
= 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")
```

Retorna o seguinte:

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 "Padrão": exibe pelo menos um dígito à esquerda e dois dígitos à direita do separador decimal, além de incluir 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.

Formatos numéricos predefinidos

Os seguintes formatos numéricos predefinidos poderão ser especificados no argumento **format_string**:

FORMATAR	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.

FORMATAR	DESCRIÇÃO
`"On/Off"`	Exibe Desligado o número é 0; caso contrário, exibe Ligado.

Formatos numéricos personalizados

Uma expressão de formato personalizada para números pode ter de uma a três seções separadas por ponto e vírgulas. Caso o argumento da cadeia de caracteres de formato contenha um dos formatos numéricos nomeados, somente uma seção será permitida.

SE VOCÊ USAR	O RESULTADO É
Apenas uma seção	A expressão de formato é aplicada 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.

```
"$#,##0;($#,##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"
```

Caso inclua ponto e vírgulas sem nada entre eles, a seção ausente será mostrada usando o formato do valor positivo.

Caracteres de formato numérico e personalizado

Os seguintes caracteres de formato numérico e personalizado poderão ser especificados no argumento **format_string**:

CARACTERE	DESCRIÇÃO
Nenhum	Exibe o número sem formatação.
(**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.

CARACTERE	DESCRIÇÃO
(**#**)	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.
(,)	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.

CARACTERE	DESCRIÇÃO
(**/**)	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 (" ").
(****)	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 o mesmo que 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 (" ").

Formatos de data/hora predefinidos

Os formatos a seguir de data/hora predefinidos poderão ser especificados no argumento **format_string**. Ao usar outros formatos, eles serão interpretados como um formato de data/hora personalizado:

FORMATAR	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.

FORMATAR	DESCRIÇÃO
`"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.

Formatos de data/hora personalizados

Os seguintes caracteres de formato poderão ser especificados como **format_string** para criar formatos de data/hora personalizados:

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.
(\ )	Barra invertida. Exibe o próximo caractere como um caractere literal. Portanto, ele não é interpretado como um caractere de formatação.
(")	Aspas duplas. O texto é exibido entre aspas duplas. Portanto, elas não são interpretadas como caracteres de formatação.
c	Exibe a data como `ddddd` e a hora como `ttttt`, 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 um 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.

CARACTERE	DESCRIÇÃO
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 `mm/dd/yyyy`.
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 `dddd, mmmm d, yyyy`.
w	Exibe o dia da semana como um número (de 1 para domingo a 7 para sábado).
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 `m` vier imediatamente após `h` ou `hh`, o minuto, em vez do mês, será exibido.
MM	Exibe o mês como um número com um zero à esquerda (01-12). Se `mm` vier imediatamente após `h` ou `hh`, o minuto, em vez do mês, será exibido.
mmm	Exibe o mês como uma abreviação (jan-dez). Localizado.
mmmm	Exibe o mês como um nome de mês completo (janeiro-dezembro). Localizado.
q	Exibe o trimestre do ano como um número (1-4).
s	Exibe o dia do ano como um número (1-366).
yy	Exibe o ano como um número de 2 dígitos (00-99).
yyyy	Exibe o ano como um número de 4 dígitos (100-9999).
h	Exibe a hora como um número sem um 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 um 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 um zero à esquerda (0-59).

CARACTERE	DESCRIÇÃO
ss	Exibe o segundo como um número com um zero à esquerda (00-59).
ttttt	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 `h:mm:ss`.
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.
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 P.M. 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 normalmente é definida como uma cadeia de caracteres vazia.

A formatação de data/hora usa a localidade atual do usuário para formatar a cadeia de caracteres. Por exemplo, considere a data 25 de junho de 2020. Quando é formatada usando a cadeia de caracteres de formato "m/d/aaaa", ela passa a ser:

- A localidade do usuário são os Estados Unidos da América (en-US): "6/25/2020"
- A localidade do usuário é a Alemanha (de-DE): "6.25.2020"

Exemplos de formatos de data/hora personalizados

Os exemplos a seguir usam a data/hora quinta-feira, 25 de junho de 2020, às 13:23:45. A Alemanha (de-DE) usa um sistema de 24 horas. Não há um formato equivalente ao sistema de 12 horas (AM/PM).

FORMATAR	RESULTADO (EN-US)	RESULTADO (DE-DE)
`"c"`	06/25/2020 13:23:45	25.06.2020 13:23:45
`"d"`	25	25

FORMATAR	RESULTADO (EN-US)	RESULTADO (DE-DE)
`"dd"`	25	25
`"ddd"`	Thu	O que fazer
`"dddd"`	Quinta-feira	Donnerstag
`"ddddd"`	06/25/2020	25.06.2020
`"dddddd"`	Thursday, June 25, 2020	Donnerstag, 25. Juni 2020
`"w"`	5	5
`"ww"`	26	26
`"m"`	6	6
`"mm"`	06	06
`"mmm"`	Jun	Jun
`"mmmm"`	Junho	Juni
`"q"`	2	2
`"y"`	177	177
`"yy"`	20	20
`"yyyy"`	2020	2020
`"""Year"" yyyy"`	Year 2020	Year 2020
`"yyyy \Qq"`	2020 Q2	2020 Q2
`"dd/mm/yyyy"`	25/06/2020	25.06.2020
`"mm/dd/yyyy"`	06/25/2020	06.25.2020
`"h:nn:ss"`	13:23:45	13:23:45
`"h:nn:ss AMPM"`	1:23:45 PM	1:23:45
`"hh:nn:ss"`	13:23:45	13:23:45
`"hh:nn:ss AMPM"`	01:23:45 PM	01:23:45
`"ttttt"`	13:23:45	13:23:45

FORMATAR	RESULTADO (EN-US)	RESULTADO (DE-DE)
`"ttttt AMPM"`	13:23:45 PM	13:23:45
`"mm/dd/yyyy hh:nn:ss AMPM"`	06/25/2020 01:23:45 PM	6.25.2020 01:23:45

LEFT

17/03/2021 • 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.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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.

Confira também

Funções de texto

LEN

17/03/2021 • 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.
- 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 boolianos, 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

17/03/2021 • 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.

Retornar valor

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

MID

17/03/2021 • 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.

Retornar valor

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

REPLACE

17/03/2021 • 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 **old_text** que você deseja substituir por **new_text**.
num_chars	O número de caracteres a serem substituídos. **Aviso:** Se o argumento *num_chars* for um espaço em branco ou fizer referência a uma coluna que seja avaliada como um espaço em branco, a cadeia de caracteres para *new_text* será inserida na posição, *start_num*, sem que nenhum caractere seja substituído. Esse é o mesmo comportamento do Excel.
new_text	O texto de substituição dos caracteres especificados em **old_text**.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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")
```

Confira também

Funções de texto
função SUBSTITUTE

REPT

17/03/2021 • 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.

Retornar valor

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

O exemplo a seguir retorna a cadeia de caracteres 85 repetida três vezes.

```
= REPT("85",3)
```

Exemplo: repetindo valores de coluna

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.

```
= REPT([MyText],[MyNumber])
```

MYTEXT	MYNUMBER	CALCULATEDCOLUMN1
Texto	2	TextText
Número	0	
85	3	858585

Confira também

Funções de texto

RIGHT

17/03/2021 • 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.

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo: retornando um número fixo de caracteres

A fórmula a seguir retorna os dois últimos dígitos do código do produto na tabela Novos Produtos.

```
= RIGHT('New Products'[ProductCode],2)
```

Exemplo: usando uma referência de coluna para especificar a contagem de caracteres

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.

```
= RIGHT('New Products'[ProductCode],[MyCount])
```

Confira também

Funções de texto
LEFT
MID

SEARCH

17/03/2021 • 5 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 **find_text**. 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 **find_text** ou uma coluna contendo texto.
start_num	(opcional) A posição do caractere em **within_text** 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

- A função de pesquisa não diferencia maiúsculas de minúsculas. A pesquisa por "N" encontrará a primeira ocorrência de "N" ou "n".
- 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".
- 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.
- 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.

- 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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo: pesquisar em uma cadeia de caracteres

A fórmula a seguir localiza a posição da letra "n" na palavra "impressora".

```
= SEARCH("n","printer")
```

A fórmula retorna 4 porque "n" é o quarto caractere da palavra "impressora".

Exemplo: pesquisar em uma coluna

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].

```
= SEARCH("-",[PostalCode])
```

O resultado retornado é uma coluna de números, indicando a posição do índice do hífen.

Exemplo: manipulação de erro com SEARCH

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.

```
= IFERROR(SEARCH("-",[PostalCode]),-1)
```

O tipo de dados do valor que você usa como uma saída de erro precisa 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.

Confira também

MID
REPLACE
Funções de texto

SUBSTITUTE

17/03/2021 • 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 **old_text**.
instance_num	(Opcional) A ocorrência de **old_text** que você deseja substituir. Se omitido, todas as instâncias de **old_text** serão substituídas

Retornar valor

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.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo: substituição dentro de uma cadeia de caracteres

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.

```
= SUBSTITUTE([Product Code], "NW", "PA")
```

Confira também

Funções de texto
REPLACE

TRIM

17/03/2021 • 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.

Retornar valor

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, . 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

UNICHAR

17/03/2021 • 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

17/03/2021 • 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.

Retornar valor

Um código numérico para o primeiro caractere em uma cadeia de texto.

UPPER

17/03/2021 • 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.

Retornar valor

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
função LOWER

VALUE

17/03/2021 • 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

Funções de inteligência de dados temporais

17/03/2021 • 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 uma data de início especificada e continua até uma data de término especificada.
DATESINPERIOD	Retorna uma tabela que contém uma coluna de datas que começa com uma data de início especificada e continua até o número e o tipo de intervalo de datas especificados.
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

17/03/2021 • 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 boolianas são descritas em função CALCULATE.

    > **NOTE**
    > A expressão **filter** tem as restrições descritas em Função CALCULATE.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

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

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

Confira também

Funções de inteligência de dados temporais
função CLOSINGBALANCEYEAR
Função CLOSINGBALANCEQUARTER

CLOSINGBALANCEQUARTER

17/03/2021 • 2 minutes to read

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

Sintaxe

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

Parâmetros

TERMO	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 boolianas são descritas em função CALCULATE.

    > **NOTE**
    > A expressão **filter** tem as restrições descritas em Função CALCULATE.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]),DateT
ime[DateKey])
```

Confira também

Funções de inteligência de dados temporais
função CLOSINGBALANCEYEAR
função CLOSINGBALANCEMONTH

CLOSINGBALANCEYEAR

17/03/2021 • 2 minutes to read

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

Sintaxe

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

Parâmetros

TERMO	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 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.
- 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 em função CALCULATE.

    NOTE
    A expressão **filter** tem as restrições descritas em Função CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança

em nível de linha) ou colunas calculadas.

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])
```

Confira também

Funções de inteligência de dados temporais
função CLOSINGBALANCEYEAR
Função CLOSINGBALANCEQUARTER
função CLOSINGBALANCEMONTH

DATEADD

17/03/2021 • 2 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
datas	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: `year`, `quarter`, `month` ou `day`

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 boolianas são descritas no tópico função CALCULATE.

- 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo – Como deslocar um conjunto de datas

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

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

Confira também

Funções de inteligência de dados temporais
Funções de data e hora

DATESBETWEEN

17/03/2021 • 4 minutes to read

Retorna uma tabela que contém uma coluna de datas que começa com uma data de início especificada e continua até uma data de término especificada.

Essa função é adequada para ser transmitida como um filtro para a função CALCULATE. Use-a para filtrar uma expressão por um intervalo de datas personalizado.

> **NOTE**
> Se você estiver trabalhando com intervalos de datas padrão como dias, meses, trimestres ou anos, recomendamos usar a função mais adequada DATESINPERIOD.

Sintaxe

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

Parâmetros

TERMO	DEFINIÇÃO
datas	Uma coluna de data.
start_date	Uma expressão de data.
end_date	Uma expressão de data.

Retornar valor

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

Comentários

- No caso de uso mais comum, **dates** é uma referência à coluna de data de uma tabela de data marcada.
- Se **start_date** for BLANK, **start_date** será o valor mais antigo na coluna **dates**.
- Se **end_date** for BLANK, **end_date** será o valor mais recente na coluna **dates**.
- As datas usadas como **start_date** e **end_date** são inclusivas. Então, por exemplo, se o valor **start_date** for 1º de julho de 2019, essa data será incluída na tabela retornada (desde que a data exista na coluna **dates**).
- A tabela retornada só pode conter datas armazenadas na coluna **dates**. Então, por exemplo, se a coluna **dates** começar a partir de 1º de julho de 2017 e o valor de **start_date** for 1º de julho de 2016, a tabela retornada começará a partir de 1º de julho de 2017.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A definição de medida de tabela **Sales** a seguir usa a função DATESBETWEEN para produzir um cálculo de *LTD* (vida útil até o momento). A vida útil até o momento representa o acúmulo de uma medida ao longo do tempo desde o início do tempo.

Observe que a fórmula usa a função MAX. Essa função retorna a data mais recente que está no contexto de filtro. Portanto, a função DATESBETWEEN retorna uma tabela de datas começando na data mais antiga até a data mais recente que está sendo relatada.

Os exemplos deste artigo podem ser adicionados ao modelo de exemplo do Power BI Desktop. Para obter o modelo, confira Modelo de exemplo DAX.

```
Customers LTD =
CALCULATE(
    DISTINCTCOUNT(Sales[CustomerKey]),
    DATESBETWEEN(
        'Date'[Date],
        BLANK(),
        MAX('Date'[Date])
    )
)
```

Considere que a data mais antiga armazenada na tabela **Date** seja 1º de julho de 2017. Assim, quando um relatório filtrar a medida pelo mês de junho de 2020, a função DATESBETWEEN retornará um intervalo de datas de 1º de julho de 2017 a 30 de junho de 2020.

Confira também

- Funções de inteligência de dados temporais (DAX)
- Funções de data e hora (DAX)
- Função DATESINPERIOD (DAX)

DATESINPERIOD

17/03/2021 • 4 minutes to read

Retorna uma tabela que contém uma coluna de datas que começa com uma data de início especificada e continua até o número e o tipo de intervalo de datas especificados.

Essa função é adequada para ser transmitida como um filtro para a função CALCULATE. Use-a para filtrar uma expressão por intervalos de datas padrão, como dias, meses, trimestres ou anos.

Sintaxe

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

Parâmetros

TERMO	DEFINIÇÃO
datas	Uma coluna de data.
start_date	Uma expressão de data.
number_of_intervals	Um inteiro que especifica o número de intervalos a serem adicionados às datas ou subtraídos delas.
intervalo	O intervalo pelo qual as datas serão deslocadas. O valor do intervalo pode ser um dos seguintes: `DAY`, `MONTH`, `QUARTER` e `YEAR`

Retornar valor

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

Comentários

- No caso de uso mais comum, **dates** é uma referência à coluna de data de uma tabela de data marcada.
- 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. Os valores válidos são `DAY`, `MONTH`, `QUARTER` e `YEAR`. Como ele é uma enumeração, os valores não são transmitidos como cadeias de caracteres. Portanto, não coloque-os entre aspas.
- A tabela retornada só pode conter datas armazenadas na coluna **dates**. Então, por exemplo, se a coluna **dates** começar a partir de 1º de julho de 2017 e o valor de **start_date** for 1º de julho de 2016, a tabela retornada começará a partir de 1º de julho de 2017.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

A definição de medida da tabela **Sales** a seguir usa a função DATESINPERIOD para calcular a receita do PY (ano anterior).

Observe que a fórmula usa a função MAX. Essa função retorna a data mais recente que está no contexto de filtro. Portanto, a função DATESINPERIOD retorna uma tabela de datas começando a partir da data mais recente do último ano.

Os exemplos deste artigo podem ser adicionados ao modelo de exemplo do Power BI Desktop. Para obter o modelo, confira Modelo de exemplo DAX.

```
Revenue PY =
CALCULATE(
    SUM(Sales[Sales Amount]),
    DATESINPERIOD(
        'Date'[Date],
        MAX('Date'[Date]),
        -1,
        YEAR
    )
)
```

Considere que o relatório é filtrado pelo mês de junho de 2020. A função MAX retorna 30 de junho de 2020. A função DATESINPERIOD retorna um intervalo de datas de 1º de julho de 2019 até 30 de junho de 2020. É um ano de valores de data que começam a partir de 30 de junho de 2020 no último ano.

Confira também

Funções de inteligência de dados temporais (DAX)
Funções de data e hora (DAX)
Função DATESBETWEEN (DAX)

DATESMTD

17/03/2021 • 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.

Retornar valor

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 boolianas são descritas no tópico função CALCULATE.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
função DATESYTD
função DATESQTD

DATESQTD

17/03/2021 • 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.

Retornar valor

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 boolianas são descritas no tópico função CALCULATE.

- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
função DATESYTD
função DATESMTD

DATESYTD

17/03/2021 • 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
datas	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.

Retornar valor

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 boolianas são descritas no tópico função CALCULATE.

- 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
função DATESMTD
função DATESQTD

ENDOFMONTH

17/03/2021 • 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 booliana que define uma tabela de coluna única de valores de data/hora.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

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

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

Confira também

Funções de data e hora
Funções de inteligência de dados temporais
Função ENDOFYEAR
Função ENDOFQUARTER

ENDOFQUARTER

17/03/2021 • 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.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

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

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

Confira também

Funções de data e hora
Funções de inteligência de dados temporais
Função ENDOFYEAR
Função ENDOFMONTH

ENDOFYEAR

17/03/2021 • 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
datas	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.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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")
```

Confira também

Funções de data e hora
Funções de inteligência de dados temporais

FIRSTDATE

17/03/2021 • 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.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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])
```

Confira também

Funções de data e hora
Funções de inteligência de dados temporais

função LASTDATE

Função FIRSTNONBLANK

FIRSTNONBLANK

17/03/2021 • 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**.

Retornar valor

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.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Confira também

Função LASTNONBLANK
Funções estatísticas

FIRSTNONBLANKVALUE

17/03/2021 • 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 que não está em branco de <expression> correspondente 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.
- Essa função é diferente de FIRSTNONBLANK, pois <column> será adicionado ao contexto de filtro para a avaliação de <expression>.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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])
```

Confira também

Funções de data e hora
Funções de inteligência de dados temporais
função FIRSTDATE
Função LASTNONBLANK

LASTNONBLANK

17/03/2021 • 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**.

Retornar valor

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
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Confira também

Função FIRSTNONBLANK
Funções estatísticas

LASTNONBLANKVALUE

17/03/2021 • 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 que não está em branco de <expression> correspondente 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.
- Essa função é diferente de LASTNONBLANK, pois <column> será adicionado ao contexto de filtro para obter a avaliação de <expression>.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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

17/03/2021 • 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
datas	Uma coluna que contém datas.

Retornar valor

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.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
função NEXTQUARTER
função NEXTMONTH

NEXTMONTH

17/03/2021 • 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
datas	Uma coluna que contém datas.

Retornar valor

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.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
Função NEXTDAY

NEXTQUARTER

17/03/2021 • 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
datas	Uma coluna que contém datas.

Retornar valor

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.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
Função NEXTDAY

função NEXTMONTH
Função NEXTYEAR

NEXTYEAR

17/03/2021 • 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
datas	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.

Retornar valor

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.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
Função NEXTDAY
função NEXTQUARTER
função NEXTMONTH

OPENINGBALANCEMONTH

17/03/2021 • 2 minutes to read

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

Sintaxe

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

Parâmetros

TERMO	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.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- A expressão **filter** tem as restrições descritas no tópico Função CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]),DateTim
e[DateKey])
```

Confira também

OPENINGBALANCEQUARTER

17/03/2021 • 2 minutes to read

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

Sintaxe

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

Parâmetros

TERMO	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.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- A expressão **filter** tem as restrições descritas no tópico Função CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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])
```

Confira também

função OPENINGBALANCEYEAR
função OPENINGBALANCEMONTH
Funções de inteligência de dados temporais
Função CLOSINGBALANCEQUARTER

OPENINGBALANCEYEAR

17/03/2021 • 2 minutes to read

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

Sintaxe

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

Parâmetros

TERMO	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 booliana que define uma tabela de coluna única de valores de data/hora.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- A expressão **filter** tem as restrições descritas no tópico Função CALCULATE.
- 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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])
```

Confira também

função OPENINGBALANCEQUARTER
função OPENINGBALANCEMONTH
Funções de inteligência de dados temporais
função CLOSINGBALANCEYEAR

PARALLELPERIOD

17/03/2021 • 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
datas	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: `year`, `quarter`, `month`.

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 booliana que define uma tabela de coluna única de valores de data/hora.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
função DATEADD

PREVIOUSDAY

17/03/2021 • 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
datas	Uma coluna que contém datas.

Retornar valor

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.
- As restrições em expressões boolianas são descritas no tópico função CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
função PREVIOUSMONTH
Função PREVIOUSQUARTER

PREVIOUSMONTH

17/03/2021 • 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.

Retornar valor

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.
- As restrições em expressões boolianas são descritas no tópico CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
PREVIOUSDAY
PREVIOUSQUARTER

PREVIOUSYEAR

PREVIOUSQUARTER

17/03/2021 • 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
datas	Uma coluna que contém datas.

Retornar valor

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.
- As restrições em expressões boolianas são descritas no tópico CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
PREVIOUSMONTH

PREVIOUSDAY
PREVIOUSYEAR

PREVIOUSYEAR

17/03/2021 • 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
datas	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.

Retornar valor

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.
- As restrições em expressões boolianas são descritas no tópico CALCULATE.
- 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
PREVIOUSMONTH
PREVIOUSDAY
PREVIOUSQUARTER

SAMEPERIODLASTYEAR

17/03/2021 • 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.

Retornar valor

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.
- As restrições em expressões boolianas são descritas no tópico CALCULATE.
- As datas retornadas são as mesmas que as datas retornadas por esta fórmula equivalente: `DATEADD(dates, -1, year)`
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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]))
```

Confira também

Funções de inteligência de dados temporais
Funções de data e hora
PREVIOUSYEAR
PARALLELPERIOD

STARTOFMONTH

17/03/2021 • 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.
- As restrições em expressões boolianas são descritas no tópico CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

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

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

Confira também

Funções de data e hora
Funções de inteligência de dados temporais
STARTOFYEAR
STARTOFQUARTER

STARTOFQUARTER

17/03/2021 • 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.
- As restrições em expressões boolianas são descritas no tópico CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

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

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

Confira também

Funções de data e hora
Funções de inteligência de dados temporais
STARTOFYEAR
STARTOFMONTH

STARTOFYEAR

17/03/2021 • 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.
- As restrições em expressões boolianas são descritas no tópico CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

Exemplo

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

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

Confira também

Funções de data e hora
Funções de inteligência de dados temporais
STARTOFQUARTER
STARTOFMONTH

TOTALMTD

17/03/2021 • 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.
- As restrições em expressões boolianas são descritas no tópico CALCULATE.
- A expressão **filter** tem as restrições descritas no tópico CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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])
```

Confira também

ALL

CALCULATE
TOTALYTD
TOTALQTD

TOTALQTD

17/03/2021 • 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.
- As restrições em expressões boolianas são descritas no tópico CALCULATE.
- A expressão **filter** tem as restrições descritas no tópico CALCULATE.
- Não há suporte para a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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])
```

Confira também

ALL

CALCULATE
TOTALYTD
TOTALMTD

TOTALYTD

17/03/2021 • 3 minutes to read

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.
- As restrições em expressões boolianas são descritas no tópico CALCULATE.
- A expressão **filter** tem as restrições descritas no tópico CALCULATE.
- 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 a função ser usada no modo DirectQuery quando usada em regras RLS (segurança em nível de linha) ou colunas calculadas.

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])
```

Confira também

ALL
CALCULATE
DATESYTD
TOTALMTD
TOTALQTD

Instruções

17/03/2021 • 2 minutes to read

Nesta categoria

INSTRUÇÃO	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

17/03/2021 • 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.
name	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 do DAX
ORDER BY
VAR

AVALIAR

17/03/2021 • 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 do DAX
DEFINE
ORDER BY

ORDER BY

17/03/2021 • 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.

Retornar valor

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 do DAX
EVALUATE

VAR

17/03/2021 • 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	O nome da variável (identificador). Não há suporte para delimitadores. Por exemplo, "varName" ou [varName] resultará em erro. Conjunto de caracteres compatíveis: a-z, A-Z, 0-9. Os caracteres 0-9 não são válidos como primeiro caractere. O sublinhado duplo (**_**) **é permitido como prefixo do nome do identificador.** Nenhum outro caractere especial é compatível. Palavras-chave reservadas não são permitidas. Nomes de tabelas existentes não são permitidos. Espaços vazios não são permitidos.
expressão	Uma expressão DAX que retorna um valor escalar ou de tabela.

Valor retornado

Uma variável com nome que contém o resultado do argumento da expressão.

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 recalcular o resultado cada vez que ele for usado, como ocorre quando não se usa uma variável. Isso pode melhorar o desempenho da medida.

Glossário do DAX

17/03/2021 • 14 minutes to read

Consulta analítica

Os visuais do Power BI consultam um modelo de dados por meio de uma *consulta analítica*. Uma consulta analítica se esforça para reduzir os volumes de dados potencialmente grandes e as complexidades de modelo usando três fases distintas: Filtrar, agrupar e resumir. Uma consulta analítica é criada automaticamente quando os campos são atribuídos às caixas de visuais do relatório. Os autores de relatórios podem controlar o comportamento das atribuições de campo renomeando os campos, modificando a técnica de resumo ou desabilitando o resumo para obter o agrupamento. No momento do design do relatório, os filtros podem ser adicionados ao relatório, a uma página de relatório ou a um visual. No Modo de Exibição de Leitura, os filtros podem ser modificados no painel **Filtros** ou por interações com segmentações e outros visuais (filtragem cruzada).

BLANK

O DAX define a ausência de um valor como BLANK. Ele é o equivalente de SQL NULL, mas não se comporta exatamente da mesma forma. Ele está mais alinhado ao Excel e como ele define uma célula vazia. BLANK é avaliado como zero ou uma cadeia de caracteres vazia quando combinado com outras operações. Por exemplo, BLANK + 20 = 20. Sempre use letras maiúsculas; o plural é BLANKs, com um "s" minúsculo.

Coluna calculada

Um cálculo de modelo usado para adicionar uma coluna a um modelo de tabela pela escrita de uma fórmula DAX. A fórmula precisa retornar um valor escalar e é avaliada para cada linha na tabela. Uma coluna calculada pode ser adicionada a uma tabela de modo de armazenamento de Importação ou do DirectQuery.

Medida calculada

Na modelagem de tabela, não há nenhum conceito como uma *medida calculada*. Em vez disso, use *medida*. A palavra *calculada* é usada para descrever as tabelas calculadas e as colunas calculadas. Ela os distingue de tabelas e colunas originadas no Power Query. O Power Query não tem o conceito de medida.

Tabela calculada

Um cálculo de modelo usado para adicionar uma tabela a um modelo de tabela pela escrita de uma fórmula DAX. A fórmula precisa retornar um objeto de tabela. Ela resulta em uma tabela que usa o modo de armazenamento de Importação.

Cálculo

Um processo deliberado que transforma uma ou mais entradas em um ou mais resultados. Em um modelo de dados de tabela, um cálculo pode ser um objeto de modelo; uma tabela calculada, uma coluna calculada ou uma medida.

Contexto

Descreve o ambiente no qual uma fórmula DAX é avaliada. Há dois tipos de contexto: *Contexto de linha* e *contexto de filtro*. O contexto de linha representa a "linha atual" e é usado para avaliar fórmulas de coluna

calculada e expressões usadas por iteradores de tabela. O contexto de filtro é usado para avaliar medidas e representa filtros aplicados diretamente a colunas de modelo e filtros propagados por relações de modelo.

Cubo

Confira modelo multidimensional.

Modelo de dados

Um recurso de dados que é especificamente preparado para relatórios e análises. Ele permite que os usuários de relatórios procurem e explorem dados de maneira simples e intuitiva. O mais importante é que ele fornece resultados de consulta de alto desempenho, mesmo em grandes volumes de dados. Ele pode integrar dados de várias fontes e usar cálculos para transformar os dados. Ele pode impor a permissão em nível de linha para garantir que diferentes usuários tenham acesso a dados diferentes. Às vezes, é chamado de *Modelo semântico* ou apenas *Modelo*.

Modelador de dados

Um profissional de dados especializado que cria modelos de dados. No SSBI (BI de autoatendimento), eles podem ser chamados de analistas de negócios. No BI corporativo, eles podem ser chamados de desenvolvedores de BI.

DAX

A linguagem DAX (Data Analysis Expressions) é uma linguagem de fórmula para o Power Pivot no Excel, o Power BI, o Azure Analysis Services e a modelagem de tabela do SQL Server Analysis Services. Use também o DAX para adicionar cálculos de modelo de dados e definir regras de RLS (Segurança em Nível de Linha).

Segurança dinâmica

Quando as regras de RLS (Segurança em Nível de Linha) são impostas por meio da identidade do usuário de relatório. As regras filtram as tabelas do modelo usando o nome da conta do usuário, o que pode ser feito com as funções USERNAME ou USERPRINCIPALNAME. Confira Segurança em nível de linha.

Expression

Uma unidade de lógica DAX que é avaliada e retorna um resultado. As expressões podem declarar variáveis e, nesse caso, recebem uma subexpressão e precisam incluir uma instrução RETURN que produz uma expressão final. Elas são construídas por meio de objetos de modelo (tabelas, colunas ou medidas), funções, operadores ou constantes.

Campo

Recurso de modelo de dados apresentado no painel **Campos**. Os campos são usados para configurar filtros de relatório e visuais. Os campos consistem em colunas de modelo, níveis de hierarquia e medidas.

Fórmula

Uma ou mais expressões DAX usadas para definir um cálculo de modelo. As expressões internas são chamadas subexpressões. O plural é *fórmulas*.

Função

As funções DAX têm argumentos que permitem a passagem de parâmetros. As fórmulas podem usar muitas

chamadas de função, possivelmente, aninhando funções em outras funções. Em uma fórmula, os nomes de função devem ser seguidos por parênteses. Dentro dos parênteses, os parâmetros são transmitidos.

Medida implícita

Um cálculo gerado automaticamente, obtido pela configuração de um visual do Power BI para resumir os valores de coluna. As colunas **numéricas** dão suporte à maior variedade de resumos, incluindo: Soma, Média, Mínimo, Máximo, Contagem (Distinta), Contagem, Desvio padrão, Variância ou Mediana. Colunas de outros tipos de dados também podem ser resumidas. As colunas de **texto** podem ser resumidas por: Primeiro (em ordem alfabética), Último (em ordem alfabética), Contagem (Distinta) ou Contagem. As colunas de **data** podem ser resumidas por: Mais antiga, Mais recente, Contagem (Distinta) ou Contagem. As colunas **boolianas** podem ser resumidas por: Contagem (Distinta) ou Contagem.

Função de iterador

Uma função DAX que enumera todas as linhas de determinada tabela e avalia um expressão especificada para cada linha. Ela fornece flexibilidade e controle sobre como os cálculos de modelo resumem os dados.

MDX

A linguagem MDX (Multidimensional Expressions) é uma linguagem de fórmula para os modelos multidimensionais do SQL Server Analysis Services (também conhecidos como *cubos*). O MDX pode ser usado para consultar modelos de tabela, mas não pode definir medidas implícitas. Ele só pode consultar medidas já definidas no modelo.

Medida

Um cálculo que faz o resumo. As medidas são *implícitas* ou *explícitas*. Uma medida explícita é um cálculo adicionado a um modelo de dados de tabela pela escrita de uma fórmula DAX. Uma fórmula de medida precisa retornar um valor escalar. No painel **Campos**, as medidas explícitas são adornadas com um ícone de calculadora. As medidas explícitas são necessárias quando o modelo é consultado por meio do MDX (Multidimensional Expressions), como é o caso ao usar o recurso Analisar no Excel. Normalmente, uma medida explícita é chamada apenas de medida.

Grupo de medidas

Uma tabela de modelo que contém, pelo menos, uma medida e não tem hierarquias nem colunas visíveis. No painel **Campos**, cada grupo de medidas é adornado com um ícone de várias calculadoras. Os grupos de medidas são listados em conjunto na parte superior do painel **Campos** e classificados em ordem alfabética por nome.

Modelo

Confira Modelo de dados.

Modelador

Confira Modelador de dados.

Cálculo do modelo

Uma fórmula nomeada usada para adicionar uma tabela calculada, uma coluna calculada ou uma medida a um modelo de dados de tabela. A estrutura dela é <NAME> = <FORMULA>. A maioria dos cálculos é adicionada por modeladores de dados no Power BI Desktop, mas as medidas também podem ser adicionadas a um

relatório de conexão dinâmica. Confira Medidas do relatório.

Modelo multidimensional

Um modelo de dados desenvolvido para o SQL Server Analysis Services (modo multidimensional). Consiste em dimensões e medidas. Geralmente, ele é chamado de *cubo*.

Medidas rápidas

Um recurso do Power BI Desktop que elimina a necessidade de escrever fórmulas DAX para medidas normalmente definidas. As medidas rápidas incluem média por categoria, classificação e diferença da linha de base.

Medidas de relatório

Também chamadas de *medidas no nível de relatório*. Elas são adicionadas a um relatório de conexão dinâmica no Power BI Desktop pela escrita de uma fórmula DAX, mas somente para conexões com modelos do Power BI ou modelos de tabela do Analysis Services.

Segurança em nível de linha

Também chamada de *RLS*. Técnica de design para restringir o acesso a subconjuntos de dados a usuários específicos. Em um modelo de tabela, é possível criar funções de modelo. As funções têm regras, que são expressões DAX usadas para filtrar linhas de tabela.

Modelo semântico

Confira Modelo de dados.

Resumo

Uma operação aplicada aos valores de uma coluna. Confira Medida.

Cubo de tabela

Esse conceito de *cubo de tabela* não existe. Em vez disso, use modelo de tabela.

Modelo de tabela

Um modelo de dados desenvolvido para Power Pivot no Excel, no Power BI, no Azure Analysis Services ou no SQL Server Analysis Services (modo de tabela).

Inteligência de dados temporais

A inteligência de dados temporais se relaciona com os cálculos ao longo do tempo, como YTD (total acumulado no ano).

Função de inteligência de dados temporais

O DAX inclui muitas funções de inteligência de tempo. Cada função de inteligência de dados temporais atinge o resultado modificando o contexto de filtro para filtros de data. Funções de exemplo: TOTALYTD e SAMEPERIODLASTYEAR.

Valor, valores

Dados a serem visualizados.

Parâmetro de hipóteses

Um recurso do Power BI Desktop que fornece a capacidade de aceitar a entrada do usuário por meio de segmentações. Cada parâmetro cria uma tabela calculada de coluna única e uma medida que retorna um só valor selecionado. A medida pode ser usada em cálculos de modelo para responder à entrada do usuário.

Operadores DAX

17/03/2021 • 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.

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 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 booliano. 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
^	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
Booliano	Booliano
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 dão 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
Booliano	Booliano

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, recomenda-se 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

17/03/2021 • 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>
```

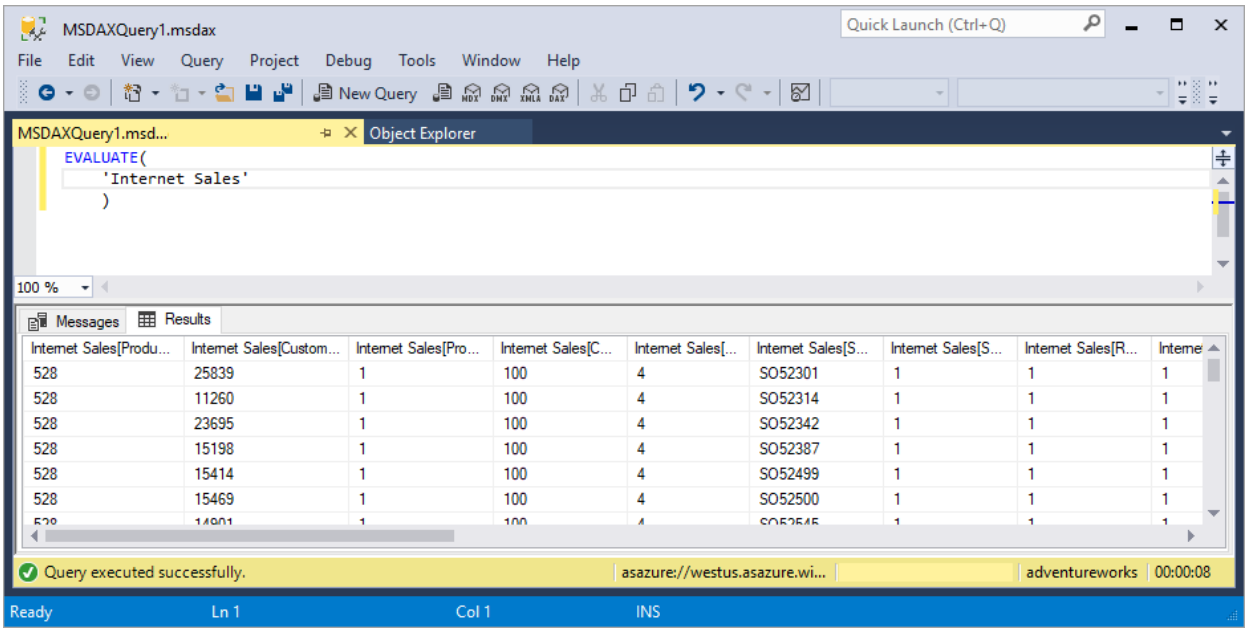
Argumentos

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}]}[, …]
```

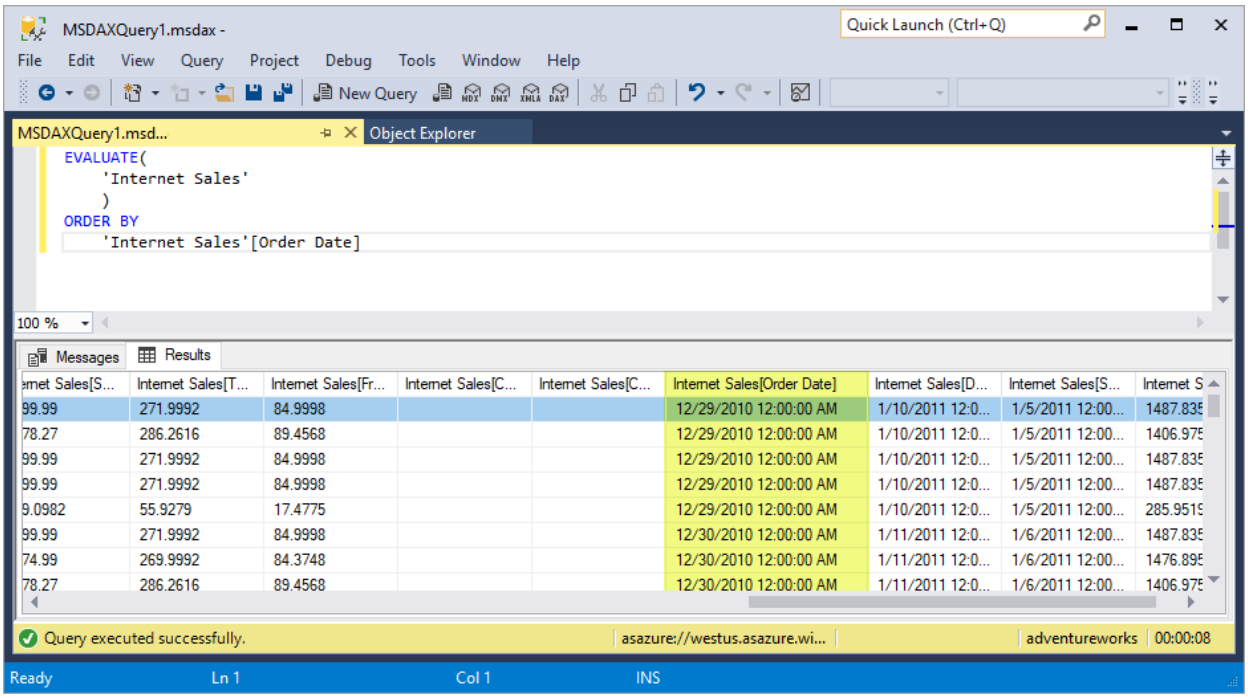
Argumentos

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

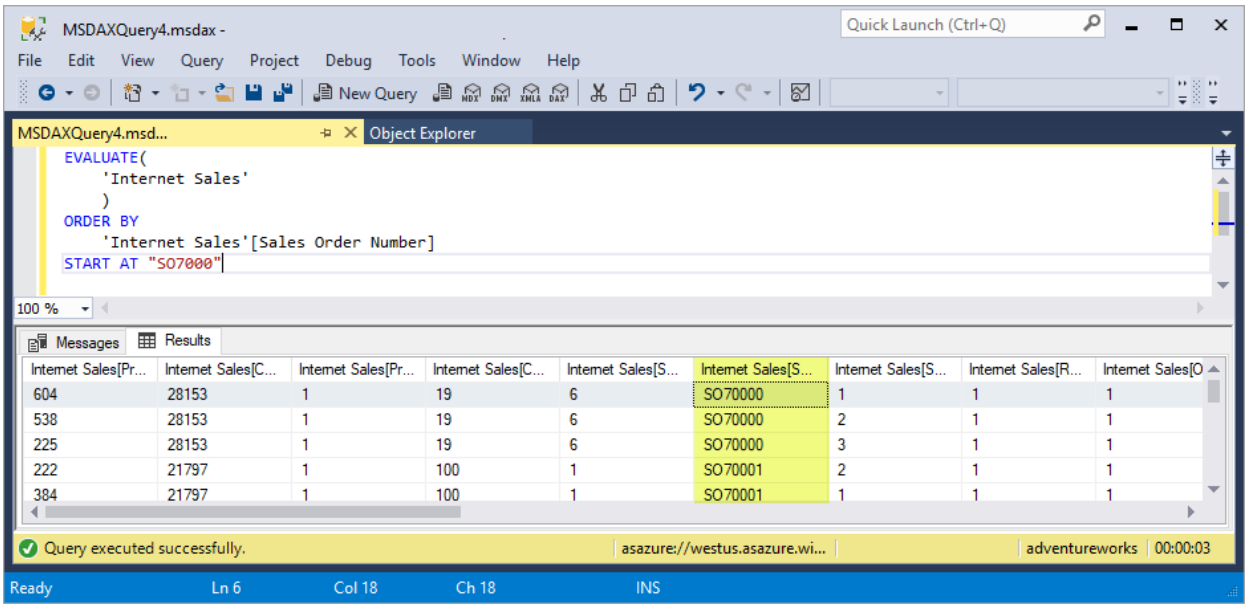
TERMO	DEFINIÇÃO
valor	Um valor constante. Não pode ser uma expressão.
parâmetro	O nome de um parâmetro em uma instrução XMLA prefixada com um caractere `@`.

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>
```

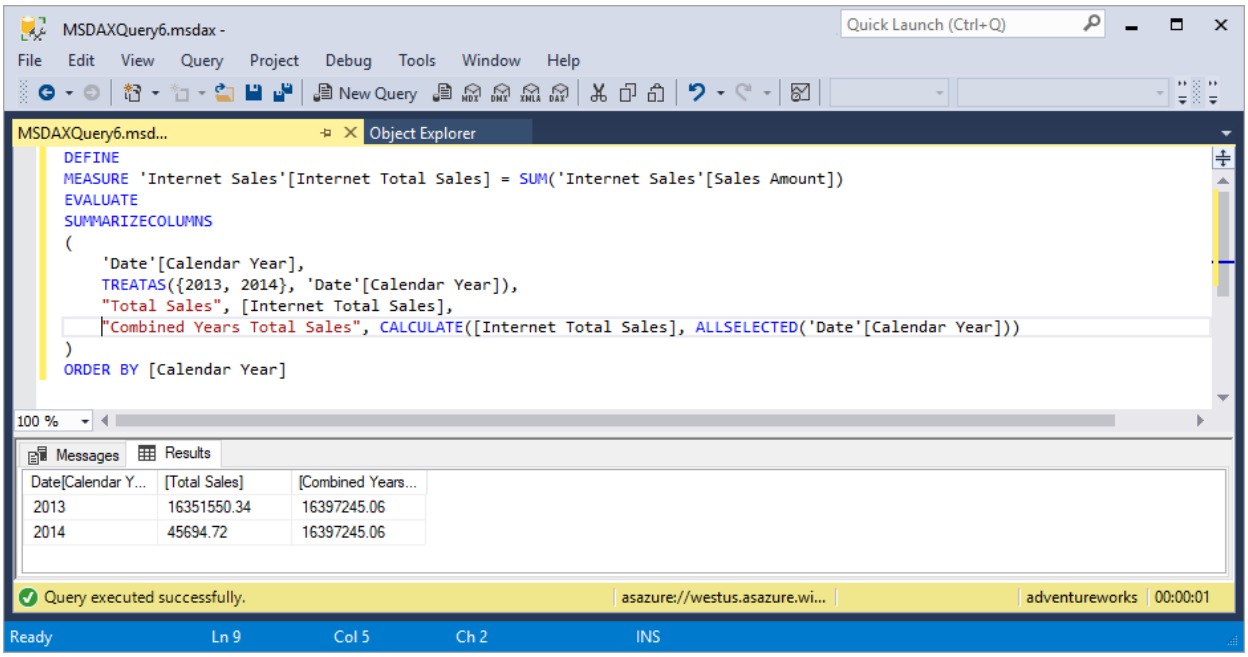
Argumentos

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 VAR 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.



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 `@`. Em qualquer lugar na sintaxe em que um valor é permitido, esse valor 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. 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

17/03/2021 • 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

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).
value	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.
ordem	Uma enumeração usada para determinar a ordem de classificação.
empates	Uma enumeração usada para determinar a manipulação de valores de ligação.
tipo	Uma enumeração usada para determinar o tipo de dados para PathItem e PathItemReverse.

Como prefixar nomes de parâmetro ou usar apenas o prefixo

TERMO	DEFINIÇÃO
como prefixar	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: Result_ColumnName – refere-se a uma coluna existente usada para obter os valores de resultado na função LOOKUPVALUE(). Search_ColumnName – refere-se a uma coluna existente usada para pesquisar um valor de resultado na função LOOKUPVALUE().

TERMO	DEFINIÇÃO
como omitir	Os nomes de parâmetro serão omitidos se o prefixo for claro o suficiente para descrever o parâmetro. 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. 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.

Sintaxe do DAX

17/03/2021 • 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 que contém 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	**Vendas**	Se o nome da tabela não contiver espaços ou outros caracteres especiais, o nome não precisará ser colocado entre aspas.
Nome da tabela	**'Canada Sales'**	Se o nome contiver espaços, tabulações ou outros caracteres especiais, coloque o nome entre aspas simples.

TIPOS DE OBJETO	EXEMPLOS	COMENTÁRIO
Nome da coluna totalmente qualificado	**Sales[Amount]**	O nome da tabela precede o nome da coluna e o nome da coluna é colocado entre colchetes.
Nome de medida totalmente qualificado	**Sales[Profit]**	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	**[Amount]**	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	**'Canada Sales'[Qty]**	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

TIPO DE OPERADOR	SÍMBOLO E USO
Operadores aritméticos	+ (adição) - (subtração/ sinal) * (multiplicação) / (divisão) ^ (exponenciação)
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.