

Referencia de expresiones de análisis de datos (DAX)

Las expresiones de análisis de datos (DAX) son una biblioteca de funciones y operadores que se pueden combinar para compilar fórmulas y expresiones en Power BI, Analysis Services y Power Pivot en modelos de datos de Excel.

Functions

REFERENCIA

[Funciones de DAX](#)

[Instrucciones DAX](#)

NOVEDADES

[Nuevas funciones de DAX](#)

Más información sobre DAX

INFORMACIÓN GENERAL

[Información general sobre DAX](#)

VIDEO

[Vídeos](#)

CURSOS

[Uso de DAX en Power BI Desktop](#)

Descripción

CONCEPTO

[Glosario de DAX](#)

[Operadores de DAX](#)

[Consultas DAX](#)

[Sintaxis DAX](#)

Información general sobre DAX

Artículo • 20/10/2023

Expresiones de análisis de datos (DAX) es un lenguaje de expresiones de fórmulas que se usa en Analysis Services, Power BI y Power Pivot en Excel. Las fórmulas DAX abarcan funciones, operadores y valores para realizar cálculos avanzados y consultas en los datos de las tablas y columnas relacionadas de los modelos de datos tabulares.

Este artículo sirve de introducción básica a los conceptos más importantes de DAX. En él se describe DAX y es aplicable a todos los productos que lo usan. Puede que algunas funciones no sean válidas en determinados productos o casos de uso. Consulte la documentación del producto en la que se especifique su implementación de DAX específica.

Cálculos

Las fórmulas DAX se usan en medidas, columnas calculadas, tablas calculadas y seguridad de nivel de fila.

Medidas

Las medidas son fórmulas de cálculo dinámico en las que los resultados cambian en función del contexto. Las medidas se usan en informes en los que se pueden combinar y filtrar datos del modelo mediante varios atributos, como un informe de Power BI o una tabla dinámica o un gráfico dinámico de Excel. Las medidas se crean con la barra de fórmulas DAX del diseñador de modelos.

Una fórmula en una medida puede usar las funciones de agregación estándar creadas automáticamente con la característica de Autosuma (como COUNT o SUM), aunque también podemos definir nuestra propia fórmula con la barra de fórmulas DAX. Las medidas con nombre se pueden pasar como argumento a otras medidas.

Al definir una fórmula para una medida en la barra de fórmulas, una característica de información sobre herramientas muestra una vista previa de cuáles serían los resultados para total en el contexto actual, pero de lo contrario no se generan los resultados inmediatamente en ninguna parte. La razón por la que no se pueden ver los resultados (filtrados) del cálculo inmediatamente es que el resultado de una medida no se puede determinar sin el contexto. Evaluar una medida requiere una aplicación cliente de informes que pueda proporcionar el contexto necesario para recuperar los datos pertinentes de cada celda y, a continuación, evaluar la expresión para cada celda. Ese

cliente podría ser una tabla dinámica o un gráfico dinámico de Excel, un informe de Power BI o una expresión de tabla en una consulta DAX en SQL Server Management Studio (SSMS).

Independientemente del cliente, se ejecuta una consulta distinta por cada celda de los resultados. Es decir, cada combinación de encabezados de fila y de columna de una tabla dinámica, o cada selección de segmentación de datos y filtros de un informe de Power BI, genera un subconjunto de datos diferente sobre el que se calcula la medida. Por ejemplo, con esta fórmula de medida muy sencilla:

DAX

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

Cuando un usuario coloca la medida TotalSales en un informe y, después, coloca la columna Product Category de una tabla Product en Filters, la suma de Sales Amount se calcula y se muestra en cada categoría de producto.

A diferencia de las columnas calculadas, la sintaxis de una medida incluye el nombre de la medida antes de la fórmula. En el ejemplo que acabamos de proporcionar, el nombre **Total Sales** aparece delante de la fórmula. Después de crear una medida, el nombre y su definición aparecen en la lista de campos de la aplicación cliente de informes y, en función de las perspectivas y roles, estarán disponibles para todos los usuarios del modelo.

Para obtener más información, consulte:

[Medidas en Power BI Desktop](#)

[Medidas en Analysis Services](#)

[Medidas en Power Pivot](#) ↗

Columnas calculadas

Una columna calculada es una columna que se agrega a una tabla existente (en el diseñador de modelos) y, después, se crea una fórmula DAX que define los valores de esa columna. Cuando una columna calculada contiene una fórmula DAX válida, se calculan valores para cada fila en cuanto la fórmula se escribe y, tras ello, los valores se almacenan en el modelo de datos en memoria. Por ejemplo, en una tabla Date, cuando se escribe la fórmula en la barra de fórmulas:

DAX

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

Se calcula un valor para cada fila de la tabla, tomando para ello los valores de la columna Calendar Year (de la propia tabla Date), agregando un espacio y la letra mayúscula Q y, finalmente, agregando los valores de la columna Calendar Quarter (de la propia tabla Date). El resultado de cada fila en la columna calculada se calcula inmediatamente y se muestra, por ejemplo, como 2017 Q1. Los valores de columna solo se recalculan si la tabla o cualquier tabla relacionada se procesa (actualiza), o si el modelo se descarga de la memoria y se vuelve a cargar, como al cerrar y volver a abrir un archivo de Power BI Desktop.

Para más información, vea:

[Columnas calculadas en Power BI Desktop](#)

[Columnas calculadas en Analysis Services](#)

[Columnas calculadas en Power Pivot](#).

Tablas calculadas

Una tabla calculada es un objeto calculado, basado en una expresión de fórmula, que se deriva de todas las tablas (o parte de ellas) del mismo modelo. En lugar de consultar y cargar valores en las columnas de la nueva tabla desde un origen de datos, una fórmula DAX define los valores de la tabla.

Las tablas calculadas pueden ser útiles en una dimensión realizadora de roles. Un ejemplo es la tabla Date, como OrderDate, ShipDate o DueDate, según la relación de clave externa. Al crear una tabla calculada para ShipDate explícitamente, se obtiene una tabla independiente disponible para las consultas y plenamente funcional, como cualquier otra tabla. Las tablas calculadas también son útiles al configurar un conjunto de filas filtrado o un subconjunto o superconjunto de columnas a partir de otras tablas existentes, ya que permiten mantener intacta la tabla original mientras se crean variaciones de dicha tabla para dar cabida a escenarios concretos.

Las tablas calculadas admiten relaciones con otras tablas. Las columnas de la tabla calculada tienen tipos de datos y formato, y pueden pertenecer a una categoría de datos. Las tablas calculadas se pueden denominar, mostrar u ocultar, como cualquier otra tabla. Las tablas calculadas vuelven a calcularse si alguna de las tablas desde la que extraen datos se actualiza.

Para más información, vea:

[Tablas calculadas en Power BI Desktop](#)

[Tablas calculadas en Analysis Services](#).

Seguridad de nivel de fila

Con la seguridad de nivel de fila, una fórmula DAX se debe evaluar como una condición booleana "true" o "false", que define qué filas pueden devolver los resultados de una consulta realizada por miembros de un rol determinado. Por ejemplo, en el caso de los miembros del rol Sales, dada una tabla Customers con la siguiente fórmula DAX:

DAX

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

Los miembros de ese rol Sales solo podrán ver los datos de los clientes de Estados Unidos, y solo se devuelven agregados (como SUM) relativos a los clientes de Estados Unidos. La seguridad de nivel de fila no está disponible en Power Pivot en Excel.

Al definir la seguridad de nivel de fila con una fórmula DAX, se crea un conjunto de filas permitido. Esto no deniega el acceso a otras filas, simplemente estas no se devuelven como parte del conjunto de filas permitido. Otros roles pueden permitir el acceso a las filas excluidas por la fórmula DAX. Si un usuario es miembro de otro rol y la seguridad de nivel de fila de ese rol permite el acceso a ese conjunto de filas específico, el usuario podrá ver los datos de esa fila.

Las fórmulas de seguridad de nivel de fila se aplican tanto a las filas especificadas como a las filas relacionadas. Si una tabla tiene varias relaciones, los filtros aplican seguridad a la relación que esté activa. Las fórmulas de seguridad de nivel de fila se entrecruzan con otras fórmulas definidas para las tablas relacionadas.

Para más información, vea:

[Seguridad de nivel de fila \(RLS\) con Power BI](#)
[Roles en Analysis Services](#)

Consultas

Se pueden crear y ejecutar consultas DAX también en SQL Server Management Studio (SSMS) y en herramientas de código abierto como DAX Studio (daxstudio.org). A diferencia de las fórmulas de cálculo DAX, que solo se pueden crear en modelos de datos tabulares, se pueden ejecutar consultas DAX también en modelos multidimensionales de Analysis Services. Las consultas DAX suelen ser más fáciles de escribir y más eficaces que las consultas de Expresiones de datos multidimensionales (MDX).

Una consulta DAX es una instrucción, como, por ejemplo, una instrucción SELECT en T-SQL. El tipo más básico de consulta DAX es una instrucción *evaluate*. Por ejemplo,

DAX

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

Devuelve como resultados una tabla que muestra solo los productos con un nivel SafetyStockLevel inferior a 200, en orden ascendente según el elemento EnglishProductName.

Se pueden crear medidas como parte de la consulta. Las medidas existen únicamente lo que dura la consulta. Para más información, vea [Consultas DAX](#).

Fórmulas

Las fórmulas DAX son fundamentales para crear cálculos en columnas y medidas calculadas, así como para proteger los datos con seguridad de nivel de fila. Para crear fórmulas de columnas y medidas calculadas, use la barra de fórmulas situada en la parte superior de la ventana del diseñador de modelos o del editor DAX. A fin de crear fórmulas para la seguridad de nivel de fila, use el cuadro de diálogo Administrador de roles o Administrar roles. La información de esta sección está pensada para empezar a conocer los conceptos básicos de las fórmulas DAX.

Conceptos básicos de las fórmulas

Las fórmulas DAX pueden ser muy simples o bastante complejas. En la siguiente tabla se muestran algunos ejemplos de fórmulas sencillas que se podrían usar en una columna calculada.

Fórmula	Definición
= TODAY()	Inserta la fecha actual en cada fila de una columna calculada.
= 3	Inserta el valor 3 en cada fila de una columna calculada.
= [Column1] + [Column2]	Agrega los valores de la misma fila de [Column1] y [Column2] y coloca los resultados en la columna calculada de la misma fila.

Independientemente de si la fórmula que se cree sea sencilla o compleja, se pueden usar los siguientes pasos para crear una fórmula:

1. Cada fórmula debe comenzar por un signo igual (=).

2. Se puede escribir o seleccionar un nombre de función, o bien escribir una expresión.
3. Empiece a escribir las primeras letras de la función o el nombre que quiera, y Autocompletar mostrará una lista de las funciones, tablas y columnas disponibles. Presione la tecla TAB para agregar un elemento de la lista Autocompletar a la fórmula.

También puede hacer clic en el botón Fx para ver una lista de las funciones disponibles. Para seleccionar una función de la lista desplegable, use las teclas de dirección para resaltar el elemento y haga clic en **Aceptar** para agregar la función a la fórmula.
4. Para proporcionar los argumentos de la función, selecciónelos de una lista desplegable de posibles tablas y columnas, o bien escriba valores.
5. Compruebe si hay errores de sintaxis: asegúrese de que todos los paréntesis están emparejados y de que se hace referencia correctamente a las columnas, tablas y valores.
6. Presione Entrar para aceptar la fórmula.

ⓘ Nota

En una columna calculada, en cuanto se escribe la fórmula y esta se valida, la columna se rellena con valores. En una medida, cuando se presiona Entrar, se guarda la definición de la medida en la tabla. Si una fórmula no es válida, se mostrará un error.

En este ejemplo, echemos un vistazo a una fórmula en una medida denominada **Days in Current Quarter**:

DAX

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

Esta medida se usa para crear una relación de comparación entre un período incompleto y el período anterior. La fórmula debe tener en cuenta la proporción del período que ha transcurrido y compararla con la misma proporción del período anterior. En este caso, [Days Current Quarter to Date]/[Days in Current Quarter] nos da la proporción transcurrida en el período actual.

Esta fórmula contiene los siguientes elementos:

Elemento de la fórmula	Descripción
Days in Current Quarter	Nombre de la medida.
=	El signo igual (=) inicia la fórmula.
COUNTROWS	COUNTROWS cuenta el número de filas de la tabla Date.
()	Los paréntesis de apertura y de cierre especifican argumentos.
DATESBETWEEN	La función DATESBETWEEN devuelve las fechas entre la última fecha de cada valor de la columna Date en la tabla Date.
'Date'	Especifica la tabla Date. Las tablas se especifican entre comillas simples.
[Date]	Especifica la columna Date de la tabla Date. Las columnas se especifican entre corchetes.
,	
STARTOFQUARTER	La función STARTOFQUARTER devuelve la fecha de inicio del trimestre.
LASTDATE	La función LASTDATE devuelve la última fecha del trimestre.
'Date'	Especifica la tabla Date.
[Date]	Especifica la columna Date de la tabla Date.
,	
ENDOFQUARTER	La función ENDOFQUARTER.
'Date'	Especifica la tabla Date.
[Date]	Especifica la columna Date de la tabla Date.

Uso de la función Autocompletar en fórmulas

La función Autocompletar ayuda a escribir una sintaxis de fórmula válida proporcionando las opciones para cada elemento de la fórmula.

- Puede usar la función Autocompletar fórmula en medio de una fórmula existente con funciones anidadas. El texto situado inmediatamente antes del punto de inserción se usa para mostrar los valores de la lista desplegable, mientras que todo el texto situado después del punto de inserción se mantiene inalterado.

- La función Autocompletar no agrega el paréntesis de cierre de las funciones, ni hace coincidir automáticamente los paréntesis. Debe asegurarse de que cada función sea correcta sintácticamente ya que, de lo contrario, no podrá guardar o usar la fórmula.

Uso de varias funciones en una fórmula

Las funciones se pueden anidar, es decir, puede usar los resultados de una función como argumento de otra función. Puede anidar hasta 64 niveles de funciones en columnas calculadas. Sin embargo, el anidamiento puede dificultar la creación de fórmulas o la solución de sus problemas. Muchas funciones están diseñadas para usarse exclusivamente como funciones anidadas. Estas funciones devuelven una tabla, que no se puede guardar directamente como un resultado, pero que se debe proporcionar como entrada de una función de tabla. Por ejemplo, las funciones SUMX, AVERAGEX y MINX requieren una tabla como primer argumento.

Functions

Una función es una fórmula con nombre dentro de una expresión. La mayoría de las funciones tienen argumentos obligatorios y opcionales, también conocidos como parámetros, como entrada. Cuando se ejecuta la función, se devuelve un valor. DAX incluye funciones que sirven para realizar cálculos usando fechas y horas, para crear valores condicionales, para trabajar con cadenas y para realizar búsquedas basadas en relaciones. También ofrece la posibilidad de iterar por una tabla para realizar cálculos recursivos. Si conoce las fórmulas de Excel, muchas de estas funciones le parecerán muy similares; sin embargo, las fórmulas DAX son diferentes en los siguientes aspectos importantes:

- Una función de DAX siempre hace referencia a una columna completa o una tabla. Si solo desea usar valores concretos de una tabla o columna, puede agregar filtros a la fórmula.
- Si necesita personalizar los cálculos fila a fila, DAX dispone de funciones que permiten usar el valor de la fila actual o un valor relacionado como un tipo de parámetro, para realizar cálculos que varían según el contexto. Para saber cómo funcionan estas funciones, vea [Contexto](#) en este artículo.
- DAX incluye muchas funciones que devuelven una tabla, en lugar de un valor. La tabla no se muestra en un cliente del informes, sino que se utiliza para proporcionar la entrada a otras funciones. Por ejemplo, puede recuperar una tabla

y, a continuación, contar los valores distintos que contiene, o calcular sumas dinámicas en tablas o columnas filtradas.

- Las funciones DAX incluyen una serie de funciones de *inteligencia de tiempo*. Estas funciones le permiten definir o seleccionar rangos de fechas y realizar cálculos dinámicos basados en dichas fechas o rangos. Por ejemplo, puede comparar sumas en períodos paralelos.

Funciones de agregación

Las funciones de agregación calculan un valor (escalar) como count, sum, average, minimum o maximum para todas las filas de una columna o tabla, según se define en la expresión. Para más información, consulte [Funciones de agregación](#).

Funciones de fecha y hora

Las funciones de fecha y hora en DAX son similares a las funciones de fecha y hora en Microsoft Excel. Sin embargo, las funciones DAX se basan en un tipo de datos **datetime** a partir del 1 de marzo de 1900. Para obtener más información, vea [Funciones de fecha y hora](#).

Funciones de filtro

Las funciones de filtro de DAX devuelven tipos de datos específicos, valores de búsqueda en tablas relacionadas, además de la capacidad de filtrar por valores relacionados. Las funciones de búsqueda usan tablas y relaciones, como una base de datos. Las funciones de filtrado permiten manipular el contexto de los datos para crear cálculos dinámicos. Para obtener más información, vea [Funciones de filtro](#).

Funciones financieras

Estas funciones en DAX se usan en fórmulas que realizan cálculos financieros, como el valor neto presente y la tasa de devolución. Estas funciones son similares a las funciones financieras usadas en Microsoft Excel. Para obtener más información, vea [Funciones financieras](#).

Funciones de información

Una función de información examina la celda o fila que se proporciona como argumento e indica si el valor coincide con el tipo esperado. Por ejemplo, la función

`ISERROR` devuelve TRUE si el valor al que se hace referencia contiene un error. Para obtener más información, vea [Funciones de información](#).

Funciones lógicas

Las funciones lógicas actúan sobre una expresión para devolver información acerca de los valores de la expresión. Por ejemplo, la función TRUE permite saber si una expresión que se está evaluando devuelve un valor TRUE. Para obtener más información, vea [Funciones lógicas](#).

Funciones matemáticas y trigonométricas

Las funciones matemáticas en DAX son muy parecidas a las funciones matemáticas y trigonométricas de Excel. Existen pequeñas diferencias en los tipos de datos numéricos usados por funciones de DAX. Para obtener más información, vea [Funciones matemáticas y trigonométricas](#).

Otras funciones

Estas funciones realizan acciones únicas que no se pueden definir por medio de ninguna de las categorías a las que la mayoría de las otras funciones pertenecen. Para obtener más información, vea [Otras funciones](#).

Funciones de relación

Las funciones de relación en DAX permiten devolver valores de otra tabla relacionada, especificar una relación concreta para usarla en una expresión y especificar la dirección de filtro cruzado. Para más información, consulte [Funciones de relación](#).

Funciones estadísticas

Las funciones estadísticas calculan valores relacionados con las distribuciones estadísticas y la probabilidad, como la desviación estándar y el número de permutaciones. Para obtener más información, vea [Funciones estadísticas](#).

Funciones de texto

Las funciones de texto en DAX son muy parecidas a sus homólogas en Excel. Puede devolver parte de una cadena, buscar texto dentro de una cadena o concatenar valores

de una cadena. DAX también proporciona funciones para controlar los formatos para las fechas, horas y números. Para obtener más información, vea [Funciones de texto](#).

Funciones de inteligencia de tiempo

Las funciones de inteligencia de tiempo proporcionadas en DAX le permiten crear cálculos que usan el conocimiento integrado acerca de calendarios y fechas. El uso de intervalos de fecha y de hora en combinación con agregaciones o cálculos permite crear comparaciones significativas a lo largo de períodos de tiempo comparables relativos a ventas, inventarios, etc. Para obtener más información, vea [Funciones de inteligencia de tiempo \(DAX\)](#).

Funciones de manipulación de tablas

Estas funciones devuelven una tabla o manipulan las tablas existentes. Por ejemplo, mediante el uso de ADDCOLUMNS puede agregar columnas calculadas a una tabla especificada o devolver una tabla de resumen sobre un conjunto de grupos con la función SUMMARIZECOLUMN. Para obtener más información, vea [Funciones de manipulación de tablas](#).

Variables

Se pueden crear variables dentro de una expresión mediante [VAR](#). Técnicamente, VAR no es una función, sino una palabra clave que sirve para almacenar el resultado de una expresión como una variable con nombre. De este modo, esa variable se puede pasar como argumento a otras expresiones de medida. Por ejemplo:

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

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

En este ejemplo, TotalQty se puede pasar como una variable con nombre a otras expresiones. Las variables pueden ser de cualquier tipo de datos escalar, tablas incluidas. El uso de variables en las fórmulas DAX puede ser increíblemente eficaz.

Tipos de datos

Puede importar datos en un modelo de varios orígenes de datos diferentes que podrían admitir tipos de datos diferentes. Al importar los datos en un modelo, se convierten a uno de los tipos de datos del modelo tabular. Cuando se usa el modelo de datos en un cálculo, los datos se convierten a un tipo de datos DAX para la duración y el resultado del cálculo. Cuando se crea una fórmula DAX, los términos usados en la fórmula determinarán automáticamente el tipo de datos de valor devuelto.

DAX admite los siguientes tipos de datos:

Tipo de datos en el modelo	Tipo de datos en DAX	Descripción
Whole Number	Valor entero de 64 bits (ocho bytes) 1, 2	Números que no tienen posiciones decimales. Los enteros pueden ser números positivos o negativos, pero deben ser números enteros comprendidos entre -9.223.372.036.854.775.808 (-2^63) y 9.223.372.036.854.775.807 (2^63-1).
Decimal Number	Número real de 64 bits (ocho bytes) 1, 2	<p>Los números reales son aquellos que pueden tener posiciones decimales. Abarcan un amplio intervalo de valores:</p> <p>Valores negativos de -1,79E +308 a -2,23E -308</p> <p>Cero</p> <p>Valores positivos desde 2,23E -308 hasta 1,79E + 308</p> <p>Sin embargo, el número de dígitos significativos se limita a 17 dígitos decimales.</p>
Booleano	Boolean	Valor True o False.
Texto	String	Cadena de datos de carácter Unicode. Pueden ser cadenas, números o fechas representados en un formato de texto.
Fecha	Fecha y hora	<p>Fechas y horas en una representación de fecha y hora aceptada.</p> <p>Las fechas válidas son todas las fechas posteriores al 1 de marzo de 1900.</p>
Moneda	Moneda	El tipo de datos de moneda permite los valores comprendidos entre -922.337.203.685.477,5808 y 922.337.203.685.477,5807 con cuatro dígitos decimales de precisión fija.
N/D	En blanco	Un tipo en blanco es un tipo de datos de DAX que representa y reemplaza los valores NULL de SQL. Un valor en blanco se puede

Tipo de datos en el modelo	Tipo de datos en DAX	Descripción
		crear con la función BLANK y se puede comprobar si es tal con la función lógica ISBLANK.

Los modelos de datos tabulares también incluyen el tipo de datos *Tabla* como entrada o salida en muchas funciones DAX. Por ejemplo, la función FILTER toma una tabla como entrada y genera otra tabla de salida que contiene solo las filas que cumplen las condiciones del filtro. Mediante la combinación de funciones de tabla con funciones de agregación, se pueden realizar cálculos complejos en conjuntos de datos definidos dinámicamente.

Como los tipos de datos suelen establecerse automáticamente, es importante entender los tipos de datos y cómo se aplican, en particular, a las fórmulas DAX. Los errores en fórmulas o los resultados inesperados, por ejemplo, suelen producirse cuando se usa un operador determinado que no se puede utilizar con un tipo de datos especificado en un argumento. por ejemplo, la fórmula `= 1 & 2` devuelve un resultado de cadena de 12. Sin embargo, la fórmula `= "1" + "2"` devuelve un resultado entero de 3.

Context

El *contexto* es un concepto importante que se debe comprender al crear fórmulas DAX. El contexto es lo que permite realizar análisis dinámicos, ya que los resultados de una fórmula cambian para reflejar la selección de fila o celda actual, y también los datos relacionados. Entender lo que es el contexto y usarlo eficazmente es esencial para generar análisis dinámicos y muy eficaces, y para solucionar los posibles problemas de las fórmulas.

Las fórmulas de modelos tabulares se pueden evaluar en un contexto diferente, dependiendo de otros elementos de diseño:

- Filtros aplicados en una Tabla dinámica o informe
- Filtros definidos dentro de una fórmula
- Relaciones especificadas utilizando funciones especiales dentro de una fórmula

Hay diferentes tipos de contexto: *contexto de fila*, *contexto de consulta* y *contexto de filtro*.

Contexto de fila

Se puede pensar en un *contexto de fila* como "la fila actual". Si crea una fórmula en una columna calculada, el contexto de la fila para esa fórmula incluye los valores de todas

las columnas en la fila actual. Si la tabla se relaciona con otra tabla, el contenido también incluye todos los valores de la otra tabla que están relacionados con la fila actual.

Por ejemplo, suponga que crea una columna calculada = [Freight] + [Tax] que suma los valores de dos columnas, Freight y Tax, de la misma tabla. Esta fórmula obtiene automáticamente solo los valores de la fila actual en las columnas especificadas.

El contexto de fila también sigue cualquier relación definida entre las tablas, incluidas las relaciones definidas dentro de una columna calculada utilizando fórmulas DAX, para determinar qué filas de las tablas relacionadas están asociadas a la fila actual.

Por ejemplo, la fórmula siguiente utiliza la función RELATED para capturar un valor de impuesto de una tabla relacionada, en función de la región a la que se envió el pedido. El valor del impuesto se determina utilizando el valor para la región en la tabla actual, para ello, se busca la región en la tabla relacionada y, posteriormente, se obtiene la tasa impositiva para esa región de la tabla relacionada.

DAX

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

Esta fórmula obtiene la tasa impositiva para la región actual de la tabla de regiones y la agrega al valor de la columna Freight. En las fórmulas de DAX, no necesita conocer o especificar la relación específica que conecta las tablas.

Contexto de varias filas

DAX incluye funciones que iteran los cálculos sobre una tabla. Estas funciones pueden tener varias filas actuales, cada una con su propio contexto de fila. Básicamente, estas funciones permiten crear fórmulas que realizan operaciones sobre un bucle interno y externo de forma recursiva.

Por ejemplo, suponga que un modelo contiene una tabla **Products** y una tabla **Sales**. Es posible que los usuarios deseen pasar por la tabla de ventas completa, la cual está llena de transacciones que implican a varios productos, y encontrar la cantidad más grande que se haya pedido para cada producto en cualquiera de las transacciones.

Con DAX puede compilar una fórmula única que devuelve el valor correcto y los resultados se actualizan automáticamente cada vez que un usuario agrega datos a las tablas.

DAX

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

Para obtener un ejemplo detallado de esta fórmula, vea [EARLIER](#).

En resumen, la función EARLIER almacena el contexto de fila de la operación anterior a la operación actual. En todo momento, la función almacena en memoria dos conjuntos de contexto: un conjunto de contexto representa la fila actual para el bucle interno de la fórmula y el otro conjunto de contexto representa la fila actual para el bucle externo de la fórmula. DAX alimenta automáticamente los valores entre los dos bucles de forma que puede crear agregados complejos.

Contexto de consulta

Contexto de la consulta hace referencia al subconjunto de datos que se recuperan implícitamente para una fórmula. Por ejemplo, cuando un usuario coloca una medida o un campo en un informe, el motor examina los encabezados de fila y de columna, las segmentaciones y los filtros de informe para determinar el contexto. Después, se ejecutan las consultas necesarias en los datos del modelo para obtener el subconjunto de datos correcto, se realizan los cálculos que define la fórmula y se rellenan los valores en el informe.

Como el contexto cambia según dónde se coloque la fórmula, los resultados de la fórmula también pueden variar. Por ejemplo, supongamos que creamos una fórmula que suma los valores de la columna **Profit** de la tabla **Sales**: `= SUM('Sales'[Profit])`. Si usamos esta fórmula en una columna calculada dentro de la tabla **Sales**, los resultados de la fórmula serán los mismos en toda la tabla, ya que el contexto de consulta de la fórmula es siempre el conjunto de datos completo de la tabla **Sales**. Los resultados reflejarán las ganancias de todas las regiones, de todos los productos, de todos los años, etc.

No obstante, normalmente los usuarios no quieren ver el mismo resultado cientos de veces, sino obtener las ganancias de un año determinado, de un país determinado, de un producto determinado o de alguna combinación de estos elementos para, luego, obtener un total general.

En un informe, el contexto se cambia mediante el filtrado, la incorporación o la eliminación de campos, y con el uso de segmentaciones. Para cada cambio, el contexto de consulta en el que se evalúa la medida. Por consiguiente, la misma fórmula, que se utiliza en una medida, se evalúa en un *contexto de la consulta* diferente para cada celda.

Contexto de filtro

El *contexto de filtro* es el conjunto de valores permitido en cada columna o en los valores recuperados de una tabla relacionada. Los filtros se pueden aplicar a la columna en el diseñador o en el nivel de presentación (informes y tablas dinámicas). Las expresiones de filtro también pueden definir explícitamente filtros dentro de la fórmula.

El contexto del filtro se agrega al especificar las restricciones de filtro en el conjunto de valores permitido en una columna o tabla, utilizando los argumentos para una fórmula. El contexto del filtro se aplica sobre otros contextos, como el contexto de la fila o el de la consulta.

En los modelos tabulares hay muchas maneras de crear el contexto de filtro. En el contexto de los clientes que pueden consumir el modelo, como los informes de Power BI, los usuarios pueden crear filtros sobre la marcha agregando segmentaciones o filtros de informe en los encabezados de columna y fila. También puede especificar directamente las expresiones de filtro dentro de la fórmula, para especificar valores relacionados, filtrar las tablas que se usan como entradas u obtener dinámicamente el contexto de los valores utilizados en los cálculos. También puede borrar por completo o de forma selectiva los filtros en columnas específicas. Esto resulta muy útil al crear fórmulas que calculan totales generales.

Para obtener más información sobre cómo crear filtros en las fórmulas, vea [Función FILTER \(DAX\)](#).

Para ver un ejemplo de cómo los filtros se pueden borrar para crear totales generales, vea [Función ALL \(DAX\)](#).

Para obtener ejemplos sobre cómo borrar y aplicar filtros de forma selectiva en las fórmulas, vea [ALLEXCEPT](#).

Determinar el contexto de las fórmulas

Al crear una fórmula DAX, se comprueba primero que la fórmula tiene una sintaxis válida y después se prueba para asegurarse de que los nombres de las columnas y tablas incluidas en la fórmula se pueden encontrar en el contexto actual. Si no se puede encontrar alguna columna o tabla especificada por la fórmula, se devuelve un error.

El contexto durante la validación (y las operaciones de recálculo) se determina, según se describió en las secciones anteriores, utilizando las tablas disponibles en el modelo, cualquier relación entre las tablas y cualquier filtro que se haya aplicado.

Por ejemplo, si ha importado recientemente algunos datos en una tabla nueva y no está relacionada con ninguna otra tabla (y no ha aplicado ningún filtro), el *contexto actual* será todo el conjunto de columnas de la tabla. Si la tabla está vinculada mediante relaciones con otras tablas, el contexto actual incluirá las tablas relacionadas. Si agrega

una columna de la tabla a un informe que tiene segmentación de datos y quizá algún filtro de informe, el contexto de la fórmula es el subconjunto de datos de cada celda del informe.

El contexto es un concepto eficaz que también puede dificultar la solución de los problemas con las fórmulas. Se recomienda comenzar con fórmulas y relaciones simples para ver cómo funciona el contexto. La siguiente sección proporciona algunos ejemplos de cómo las fórmulas utilizan tipos diferentes de contexto para devolver resultados de forma dinámica.

Operadores

El lenguaje DAX usa cuatro tipos diferentes de operadores de cálculo en las fórmulas:

- Operadores de comparación para comparar valores y devolver un valor lógico TRUE\FALSE.
- Operadores aritméticos para realizar cálculos aritméticos que devuelven valores numéricos.
- Operadores de concatenación de texto para combinar dos o más cadenas de texto.
- Operadores lógicos que combinan dos o más expresiones para devolver un único resultado.

Para más información detallada sobre los operadores usados en las fórmulas DAX, vea [Operadores DAX](#).

Trabajar con tablas y columnas

Las tablas de los modelos de datos tabulares son similares a las de Excel, pero son diferentes en la forma en que trabajan con datos y con fórmulas:

- Las fórmulas solo funcionan con tablas y columnas, pero no con celdas individuales, referencias a rangos ni matrices.
- Las fórmulas pueden usar relaciones para obtener valores de las tablas relacionadas. Los valores que se recuperan siempre se relacionan con el valor de fila actual.
- No puede tener datos irregulares o "desiguales", como puede haber en una hoja de cálculo de Excel. Cada fila de una tabla debe contener el mismo número de columnas. Sin embargo, puede tener valores vacíos en algunas columnas. Las tablas de datos de Excel y de los modelos tabulares no son intercambiables.
- Debido a que se establece un tipo de datos para cada columna, cada valor de esa columna debe ser del mismo tipo.

Hacer referencia a tablas y columnas en fórmulas

Puede hacer referencia a cualquier tabla y columna mediante el nombre. Por ejemplo, en la siguiente fórmula se muestra cómo hacer referencia a las columnas de dos tablas utilizando el nombre *completo* :

DAX

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

Al evaluar una fórmula, el diseñador de modelos comprueba primero la sintaxis general y, a continuación, compara los nombres de las columnas y las tablas proporcionadas con las posibles columnas y las tablas del contexto actual. Si el nombre es ambiguo o si no se puede encontrar la columna o tabla, obtendrá un error en su fórmula (una #cadena ERROR en lugar de un valor de datos en las celdas donde el error se produce). Para obtener más información sobre los requisitos de nomenclatura de tablas, columnas y otros objetos, vea la sección Requisitos de nomenclatura en [Sintaxis de DAX](#).

Relaciones de tablas

Al crear relaciones entre las tablas, se obtiene la capacidad de usar los valores relacionados de otras tablas en los cálculos. Por ejemplo, se puede usar una columna calculada para determinar todos los registros de envío relacionados con el distribuidor actual y, luego, sumar los costos de envío de cada uno. En muchos casos, sin embargo, puede no ser necesaria una relación. Puede usar la función [LOOKUPVALUE](#) en una fórmula para devolver el valor de *result_columnName* de la fila que cumple los criterios especificados en los argumentos *search_column* y *search_value*.

Muchas funciones de DAX requieren que exista una relación entre las tablas, o entre varias tablas, para localizar las columnas a las que se ha hecho referencia y devolver resultados que tengan sentido. Otras funciones intentarán identificar la relación; sin embargo, para obtener los mejores resultados, debería crear una relación siempre que sea posible. Los modelos de datos tabulares admiten varias relaciones entre tablas. Para evitar confusiones o resultados incorrectos, solo una relación se designa como la relación activa cada vez, si bien esta relación activa se puede cambiar según sea necesario para recorrer las distintas conexiones de los datos en los cálculos. La función [USERELATIONSHIP](#) sirve para especificar una o más relaciones que se van a usar en un cálculo específico.

Cuando se usan relaciones, es importante respetar las siguientes reglas de diseño de fórmulas:

- Si las tablas están conectadas mediante una relación, hay que garantizar que las dos columnas que se usan como claves tienen valores que coinciden. No se aplica integridad referencial, por lo que es posible tener valores no coincidentes en una columna de clave y seguir creando una relación. Si ocurre esto, debe ser consciente de que la presencia de espacios en blanco o valores no coincidentes podría afectar a los resultados de las fórmulas.
- Al vincular tablas en el modelo mediante relaciones, amplía el ámbito, o *contexto*, en el que se evalúan las fórmulas. Los cambios del contexto como resultado de la incorporación de nuevas tablas, nuevas relaciones o de cambios en la relación activa pueden hacer que los resultados cambien de forma imprevista. Para obtener más información, vea la sección [Contexto](#) en este artículo.

Proceso y actualización

Proceso y *recálculo* son dos operaciones independientes, pero que se relacionan entre sí. Debe entender perfectamente estos conceptos a la hora de diseñar un modelo que contiene fórmulas complejas, cantidades grandes de datos o datos que se obtienen de orígenes de datos externos.

El *proceso* (*actualización*) consiste en actualizar los datos de un modelo con datos nuevos de un origen de datos externo.

El *recálculo* es el proceso de actualizar los resultados de las fórmulas para reflejar cualquier cambio en las propias fórmulas y cualquier cambio en los datos subyacentes. El recálculo puede afectar al rendimiento de las siguientes maneras:

- Los valores de una columna calculada se calculan y se almacenan en el modelo. Para actualizar los valores de la columna calculada, debe procesar el modelo mediante uno de los tres comandos de procesamiento: Proceso completo, Procesar datos o Procesar recalc. El resultado de la fórmula se debe recalculiar siempre para la columna completa, cada vez que cambia la fórmula.
- Los valores calculados mediante medidas se evalúan dinámicamente siempre que un usuario agrega la medida a una tabla dinámica o abre un informe; a medida que el usuario modifique el contexto, los valores devueltos por la medida cambiarán. Los resultados de la medida siempre reflejan el valor más reciente de la memoria caché en memoria.

El proceso y el recálculo no tienen ningún efecto en las fórmulas de seguridad de nivel de fila, a menos que el resultado de un recálculo devuelva un valor diferente, lo que hace que los miembros del rol puedan o no realizar consultas en esa fila.

Actualizaciones

DAX mejora constantemente. Las [funciones nuevas y actualizadas](#) se publican con la actualización disponible siguiente, que suele ser mensual. En primer lugar se actualizan los servicios, seguido de las aplicaciones instaladas como Power BI Desktop, Excel, SQL Server Management Studio (SSMS) y la extensión del proyecto de Analysis Services para Visual Studio (SSDT). SQL Server Analysis Services se actualiza en la actualización acumulativa siguiente. Las funciones nuevas se anuncian y se describen en primer lugar en la referencia de funciones DAX, que coincide con las actualizaciones de Power BI Desktop.

No todas las funciones se admiten en versiones anteriores de SQL Server Analysis Services y Excel.

Solución de problemas

Si recibe un error al definir una fórmula, esta podría contener un *error sintáctico*, un *error semántico* o un *error de cálculo*.

Los errores sintácticos son más fáciles de resolver. Normalmente, se deben a que falta un paréntesis o una coma.

El otro tipo de error sucede cuando la sintaxis es correcta, pero el valor o una columna a la que se hace referencia no tienen sentido en el contexto de la fórmula. Estos errores semánticos y de cálculo se pueden deber a uno de los problemas siguientes:

- La fórmula hace referencia a una columna, tabla o función que no existe.
- La fórmula parece ser correcta, pero cuando el motor de datos captura los datos, detecta un error de coincidencia de tipos y genera un error.
- La fórmula pasa un número o un tipo de argumentos incorrecto a una función.
- La fórmula hace referencia a otra columna que tiene un error y, en consecuencia, sus valores no son válidos.
- La fórmula hace referencia a una columna que no se ha procesado, lo que significa que tiene metadatos pero ningún dato real que se pueda utilizar para los cálculos.

En los cuatro primeros casos, DAX marca la columna completa que contiene la fórmula no válida. En el último caso, DAX hace que la columna se muestre en gris para indicar que se encuentra en estado no procesado.

Aplicaciones y herramientas

Power BI Desktop



Power BI Desktop [↗](#) es una aplicación de creación de informes y modelado de datos gratuita. El diseñador de modelos incluye un editor DAX para crear fórmulas de cálculo DAX.

Power Pivot en Excel



El diseñador de modelos de Power Pivot en Excel [↗](#) incluye un editor DAX para crear fórmulas de cálculo DAX.

Visual Studio



Visual Studio con la extensión de [proyectos de Analysis Services](#) [↗](#) (VSIX) se usa para crear proyectos de modelos de Analysis Services. El diseñador de modelos tabulares, instalado con la extensión de proyectos, incluye un editor de DAX.

SQL Server Management Studio



[SQL Server Management Studio](#) (SSMS) es una herramienta esencial para trabajar con Analysis Services. SSMS incluye un editor de consultas DAX para consultar modelos

tanto tabulares como multidimensionales.

DAX Studio



[DAX Studio](#) es una herramienta de cliente de código abierto para crear y ejecutar consultas DAX en Analysis Services, Power BI Desktop y Power Pivot en modelos de Excel.

Tabular Editor



[Tabular Editor](#) es una herramienta de código abierto que proporciona una vista jerárquica intuitiva de cada objeto en los metadatos del modelo tabular. Tabular Editor incluye un editor DAX con resaltado de sintaxis, que proporciona una manera sencilla de editar medidas, columnas calculadas y expresiones de tabla calculada.

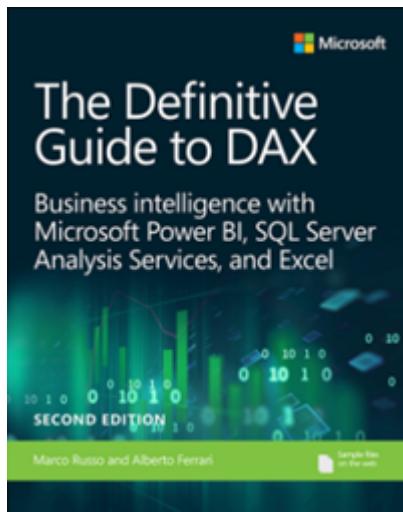
Recursos de aprendizaje

Al familiarizarse con DAX, lo mejor es usar la aplicación que se va a usar para crear los modelos de datos. Analysis Services, Power BI Desktop y Power Pivot en Excel cuentan con artículos y tutoriales que incluyen lecciones sobre cómo crear medidas, columnas calculadas y filtros de fila mediante DAX. Estos son algunos recursos adicionales:

Vídeos

[Uso de DAX en Power BI Desktop](#) ruta de aprendizaje.

La [guía definitiva sobre DAX](#), por Alberto Ferrari y Marco Russo (Microsoft Press). En la segunda edición de esta guía exhaustiva, los modeladores de datos noveles y los profesionales de la inteligencia empresarial encontrarán los conceptos básicos de técnicas de alto rendimiento muy innovadoras.



Comunidad

DAX cuenta con una comunidad muy activa que siempre está dispuesta a compartir sus conocimientos. La comunidad de Microsoft [Power BI](#) cuenta con un foro de debate específico para DAX, [DAX Commands and Tips](#).

Comentarios

¿Le ha resultado útil esta página?

Yes

No

Vídeos

Artículo • 20/10/2023

Tanto si usa Power BI Desktop, Power Pivot en Excel o Analysis Services, el aprendizaje de Expresiones de análisis de datos (DAX) es esencial para crear modelos de datos eficientes. Estos son algunos vídeos que le ayudarán a empezar a usar este lenguaje de expresiones potente.

DAX 101

En este vídeo DAX 101 Alberto Ferrari, partner de Microsoft, presenta los conceptos básicos de DAX. Con ejemplos prácticos y claros, obtendrá información sobre las medidas, las columnas calculadas y las expresiones básicas de modelado de datos con DAX.

<https://www.youtube-nocookie.com/embed/kIQAZLr5vxA> ↗

Advanced DAX (DAX avanzado)

En este vídeo Advanced DAX (DAX avanzado) Alberto Ferrari, partner de Microsoft, describe la teoría DAX, el contexto de filtro y de fila, y otros conceptos básicos de DAX.

<https://www.youtube-nocookie.com/embed/6ncHnWMEdic> ↗

Practical DAX for Power BI (DAX práctico para Power BI)

En este vídeo, el administrador de programas principal de Microsoft, Phil Seemark, le guía por algunos procedimientos recomendados para usos prácticos de DAX en Power BI.

<https://www.youtube-nocookie.com/embed/1fGfqzS37qs> ↗

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

Escritura de consultas DAX con Copilot

Artículo • 02/04/2024

Pruebe a usar Copilot para crear una consulta DAX. Escriba un mensaje para describir qué consulta DAX desea generar y seleccione **Enviar** o presione **Entrar**. Para ejecutar lo que se devuelve, seleccione **Conservarla** para agregarlo a la pestaña de consulta. A continuación, seleccione **Ejecutar** o presione **F5** para ver los resultados de la consulta DAX.

Características de una consulta DAX en Copilot

Echemos un vistazo a lo que Copilot puede ayudarle en la [vista de consulta DAX](#):

- **Lenguaje natural a consulta DAX.** Solo tiene que pedir lo que desea de su modelo semántico y hacer que Copilot genere la consulta DAX.
- **Consulta DAX a lenguaje natural.** Pida a Copilot que explique lo que hace cualquier consulta DAX.
- **Descripción de los conceptos de DAX.** Tal vez haya oído hablar de una determinada función DAX o necesita encontrar la función DAX correcta para el trabajo. Copilot también puede ayudarle aquí.

Habilitación de Copilot para consultas DAX

En Power BI Desktop, en la sección **Características de vista previa de las opciones**, habilite la **vista de consulta DAX con Copilot**.

Compruebe que cumple los [requisitos](#) de Fabric Copilot.

A continuación, vaya a la [vista de consulta DAX](#) y seleccione el botón **Copilot** o **CTRL+I** para iniciar Copilot en línea.

Vamos a probar algunos avisos de Copilot, con datos de ejemplo en Power BI Desktop, con algunas medidas adicionales.

Solicitudes de ejemplo para escribir consultas DAX

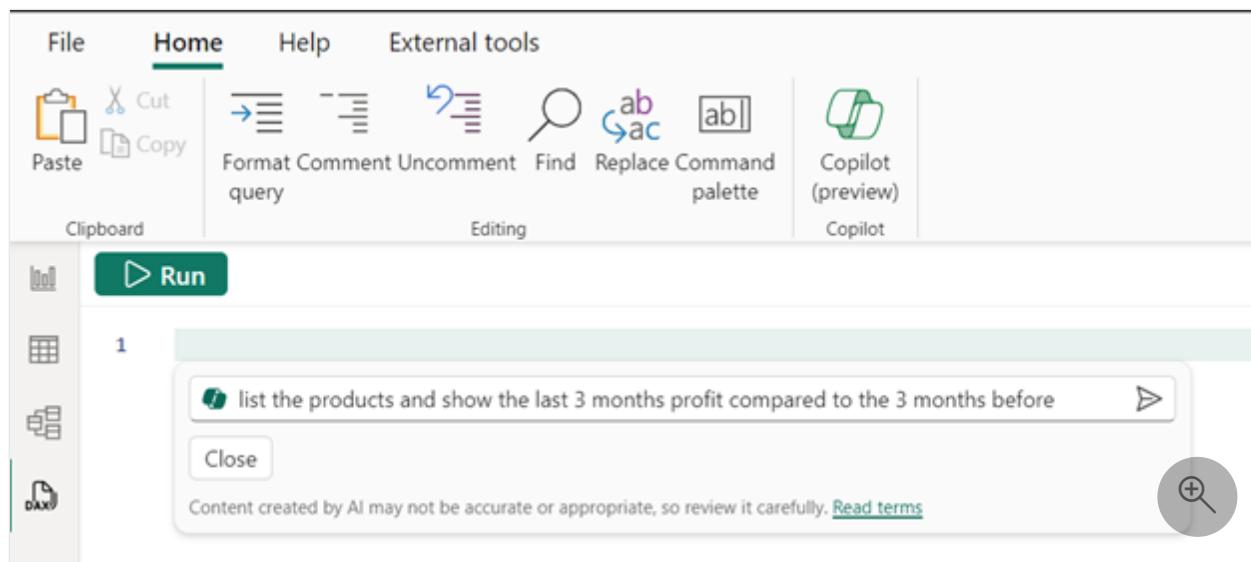
Esta es una colección de indicaciones de ejemplo para ilustrar cómo obtener Copilot para escribir las consultas DAX que desee.

- Enumere los productos y muestre el beneficio de los últimos tres meses en comparación con los tres meses anteriores.
- Agregue una columna para mostrar la diferencia.
- Explique lo que hace esta consulta.
- ¿Para qué se usa SUMMARIZECOLUMNS?

Enumere los productos y muestre el beneficio de los últimos tres meses en comparación con los tres meses anteriores

En una pestaña de consulta en blanco, invoque Copilot con **CTRL+I** y escriba esta solicitud:

"enumere los productos y muestre los últimos 3 meses de beneficio en comparación con los 3 meses anteriores".



Después de unos instantes aparece una consulta DAX detallada.

The screenshot shows the Fabric Copilot interface with the following components:

- Left Sidebar:** Icons for Home, Recent, and New.
- Top Bar:** A green button labeled "Run".
- Main Area:** A code editor containing DAX code. The code defines two variables: "Last3Months" and "Previous3Months", both using CALCULATETABLE to filter dates. It then evaluates the profit for each product in these periods and summarizes the results by product. The code is numbered from 1 to 25.

```
1  -
1+ // DAX query generated by Fabric Copilot with "list the products and show the last 3 months profit compared
2+
3+ DEFINE
4+     // Define the last 3 months period
5+     VAR Last3Months =
6+         CALCULATETABLE (
7+             VALUES ( 'Date'[Date] ),
8+             'Date'[Date] > EOMONTH ( MAX ( 'Date'[Date] ), -4 ),
9+             'Date'[Date] <= EOMONTH ( MAX ( 'Date'[Date] ), -1 )
10+
11+     // Define the 3 months before the last 3 months period
12+     VAR Previous3Months =
13+         CALCULATETABLE (
14+             VALUES ( 'Date'[Date] ),
15+             'Date'[Date] > EOMONTH ( MAX ( 'Date'[Date] ), -7 ),
16+             'Date'[Date] <= EOMONTH ( MAX ( 'Date'[Date] ), -4 )
17+
18+ EVALUATE
19+     // Calculate the profit for each product in the last 3 months and the 3 months before
20+     SUMMARIZECOLUMNS (
21+         'Fact'[Product],
22+         "Last 3 Months Profit", CALCULATE ( [Profit], Last3Months ),
23+         "Previous 3 Months Profit", CALCULATE ( [Profit], Previous3Months )
24+
25+ ORDER BY
26+     'Fact'[Product]
```
- Bottom Panel:**
 - A description box with a "Keep it" button, a trash icon, and a refresh icon.
 - A note: "Content created by AI may not be accurate or appropriate, so review it carefully. [Read terms](#)"
 - A magnifying glass icon.

Seleccione **Consevarla** y, a continuación, **Ejecutar la consulta**.

Run

```

1 // DAX query generated by Fabric Copilot with "add a column to show the difference"
2 // DAX query generated by Fabric Copilot with "list the products and show the last 3 months profit compared to the 3 months before"
3 DEFINE
4     // Define the last 3 months period
5     VAR Last3Months =
6         CALCULATETABLE (
7             VALUES ( 'Date'[Date] ),
8             'Date'[Date] > EOMONTH ( MAX ( 'Date'[Date] ), -4 ),
9             'Date'[Date] <= EOMONTH ( MAX ( 'Date'[Date] ), -1 )
10        )
11    // Define the 3 months before the last 3 months period
12    VAR Previous3Months =
13        CALCULATETABLE (
14            VALUES ( 'Date'[Date] ),
15            'Date'[Date] > EOMONTH ( MAX ( 'Date'[Date] ), -7 ),
16            'Date'[Date] <= EOMONTH ( MAX ( 'Date'[Date] ), -4 )
17        )
18 EVALUATE
19     // Calculate the profit for each product in the last 3 months and the 3 months before
20     ADDCOLUMNS (
21         SUMMARIZECOLUMNS (
22             "Fact'[Product]",
23             "Last 3 Months Profit", CALCULATE { [Profit], Last3Months },
24             "Previous 3 Months Profit", CALCULATE { [Profit], Previous3Months }
25         ),
26         "Profit Difference", [Last 3 Months Profit] - [Previous 3 Months Profit] // Add a column to show the difference in profit
27     )
28 ORDER BY
29     'Fact'[Product]
30

```

Results | Result 1 of 1 ▾ | Copy

Fact[Product]	[Last 3 Months Profit]	[Previous 3 Months Prof...]	[Profit Difference]
1 Amarilla	788764.46	260552.97	528211.49
2 Carretera	340020.57	204468.44	135552.13
3 Montana	293406.44	322609.87	-29203.43
4 Paseo	622377.71	1120643.28	-498265.57
5 Velo	538961.03	790779.8	-251818.77
6 VTT	826188.15	489631.56	336556.59



Veamos la diferencia entre los últimos tres meses y los tres meses anteriores. Seleccione la consulta e inicie Copilot de nuevo con **CTRL+I** o seleccione el botón de la cinta de opciones.

Agregue una columna para mostrar la diferencia

Esta vez la selección se muestra ajustada al texto y el sombreado de fondo resalta lo que se envía con la solicitud a Copilot.

```

1 // DAX query generated by Fabric Copilot with "list the products and show the last 3 months profit compared to the 3 months before."
2 DEFINE
3     // Define the last 3 months period
4     VAR Last3Months =
5         CALCUTATETABLE (
6             VALUES ('Date'[Date]),
7             'Date'[Date] > EOMONTH(-MAX('Date'[Date]), -4),
8             'Date'[Date] <= EOMONTH(-MAX('Date'[Date]), -1)
9         )
10    // Define the 3 months before the last 3 months period
11    VAR Previous3Months =
12        CALCUTATETABLE (
13            VALUES ('Date'[Date]),
14            'Date'[Date] > EOMONTH(-MAX('Date'[Date]), -7),
15            'Date'[Date] <= EOMONTH(-MAX('Date'[Date]), -4)
16        )
17 EVALUATE
18     // Calculate the profit for each product in the last 3 months and the 3 months before
19     SUMMARIZECOLUMNS (
20         'Fact'[Product],
21         "Last 3 Months Profit", CALCULATE ([Profit], Last3Months),
22         "Previous 3 Months Profit", CALCULATE ([Profit], Previous3Months)
23     )
24 ORDER BY
25     'Fact'[Product]

```

Working on it... Cancel

+

Ahora el estilo de diferencias en el editor de consultas DAX muestra exactamente lo que Copilot ha cambiado.

```

1 // DAX query generated by Fabric Copilot with "list the products and show the last 3 months profit compared to the 3 months before."
2 DEFINE
3     // Define the last 3 months period
4     VAR Last3Months =
5         CALCUTATETABLE (
6             VALUES ('Date'[Date]),
7             'Date'[Date] + EOMONTH(1, MAX('Date'[Date])) - 4,
8             'Date'[Date] + EOMONTH(1, MAX('Date'[Date])) - 1
9         )
10    // Define the 3 months before the last 3 months period
11    VAR Previous3Months =
12        CALCUTATETABLE (
13            VALUES ('Date'[Date]),
14            'Date'[Date] + EOMONTH(1, MAX('Date'[Date])) - 7,
15            'Date'[Date] + EOMONTH(1, MAX('Date'[Date])) - 4
16        )
17 EVALUATE
18     // calculate the profit for each product in the last 3 months and the 3 months before
19     SUMMARIZECOLUMNS (
20         'Fact'[Product],
21         "Last 3 Months Profit", CALCULATE ([Profit], Last3Months),
22         "Previous 3 Months Profit", CALCULATE ([Profit], Previous3Months)
23     )
24 ORDER BY
25     'Fact'[Product]

```

Original DAX code (left) vs. Modified DAX code (right). The changes made by Copilot are highlighted in green.

+

Seleccione Conservarla y Ejecutar de nuevo.

The screenshot shows the DAX Studio interface with a DAX query and its results.

```

1 // DAX query generated by Fabric Copilot with "add a column to show the difference"
2 // DAX query generated by Fabric Copilot with "list the products and show the last 3 months profit compared to the 3 months before"
3 DEFINE
4     // Define the last 3 months period
5     VAR Last3Months =
6         CALCULATETABLE (
7             VALUES ( 'Date'[Date] ),
8             'Date'[Date] > EOMONTH ( MAX ( 'Date'[Date] ), -4 ),
9             'Date'[Date] <= EOMONTH ( MAX ( 'Date'[Date] ), -1 )
10        )
11    // Define the 3 months before the last 3 months period
12    VAR Previous3Months =
13        CALCULATETABLE (
14            VALUES ( 'Date'[Date] ),
15            'Date'[Date] > EOMONTH ( MAX ( 'Date'[Date] ), -7 ),
16            'Date'[Date] <= EOMONTH ( MAX ( 'Date'[Date] ), -4 )
17        )
18 EVALUATE
19     // Calculate the profit for each product in the last 3 months and the 3 months before
20     ADDCOLUMNS (
21         SUMMARIZECOLUMNS (
22             "Fact'[Product]",
23             "Last 3 Months Profit", CALCULATE { [Profit], Last3Months },
24             "Previous 3 Months Profit", CALCULATE { [Profit], Previous3Months }
25         ),
26         "Profit Difference", [Last 3 Months Profit] - [Previous 3 Months Profit] // Add a column to show the difference in profit
27     )
28 ORDER BY
29     'Fact'[Product]
30

```

Results | Result 1 of 1 ▾ | Copy

	Fact[Product]	[Last 3 Months Profit]	[Previous 3 Months Prof...]	[Profit Difference]
1	Amarilla	788764.46	260552.97	528211.49
2	Carretera	340020.57	204468.44	135552.13
3	Montana	293406.44	322609.87	-29203.43
4	Paseo	622377.71	1120643.28	-498265.57
5	Velo	538961.03	790779.8	-251818.77
6	VTT	826188.15	489631.56	336556.59

Aparece la consulta DAX y se ven las solicitudes enviadas a Copilot en la parte superior de la consulta. Cada bloque principal de la consulta también se comenta, explicando lo que está haciendo en cada parte.

Quizá esos comentarios no le sirvan. Vamos a obtener una explicación completa.

Explique lo que hace esta consulta

De nuevo, resalte toda la consulta e invoque Copilot con esta solicitud:

"Explíqueme lo que hace esta consulta".

The screenshot shows the Power BI DAX Editor interface. At the top, there are icons for file, copy, paste, and run. Below the toolbar is a code editor window containing a DAX query. The code defines two periods: 'Last3Months' and 'Previous3Months', calculates the profit for each product in these periods, and then adds a column to show the difference. A tooltip from Fabric Copilot provides an explanation of the query's purpose.

```
1 // DAX query generated by Fabric Copilot with "add a column to show the difference"
2 // DAX query generated by Fabric Copilot with "list the products and show the last 3 months profit compared to the 3 months before"
3 DEFINE
4     -- Define the last 3 months period
5     VAR Last3Months =
6         CALCULATETABLE (
7             VALUES ('Date'[Date]),
8             'Date'[Date] > EOMONTH ( MAX ( 'Date'[Date] ), -4 ),
9             'Date'[Date] <= EOMONTH ( MAX ( 'Date'[Date] ), -1 )
10        )
11    -- Define the 3 months before the last 3 months period
12    VAR Previous3Months =
13        CALCULATETABLE (
14            VALUES ('Date'[Date]),
15            'Date'[Date] > EOMONTH ( MAX ( 'Date'[Date] ), -7 ),
16            'Date'[Date] <= EOMONTH ( MAX ( 'Date'[Date] ), -4 )
17        )
18 EVALUATE
19     -- Calculate the profit for each product in the last 3 months and the 3 months before
20     ADDCOLUMNS (
21         SUMMARIZECOLUMNS (
22             "Fact'[Product]",
23             "Last 3 Months Profit", CALCULATE ( [Profit], Last3Months ),
24             "Previous 3 Months Profit", CALCULATE ( [Profit], Previous3Months )
25         ),
26         "Profit Difference", [Last 3 Months Profit] - [Previous 3 Months Profit] // Add a column to show the difference in profit
27     )
28 ORDER BY
29     [Profit Difference] desc
```

explain to me what this query does

Content created by AI may not be accurate or appropriate, so review it carefully. [Read terms](#)

Ahora, en lugar de escribir consultas DAX, responde en la ventana en línea.

This screenshot shows the AI-generated explanation for the DAX query. It includes a button to explain the query, a close button, and a refresh icon. The main text explains that the query compares profit over three months and three months before. A 'Show more' button is available to expand the explanation. A note at the bottom indicates that the content is created by AI and should be reviewed.

explain to me what this query does

This query uses DAX, a language for data analysis and manipulation, to compare the profit of each product in the last 3 months and the 3 months before that. It does this by following these steps:

Show more ▾

Content created by AI may not be accurate or appropriate, so review it carefully. [Read terms](#)

Copilot genera algunas líneas. Seleccione Mostrar más para ver la explicación completa.

explain to me what this query does



Close



This query uses DAX, a language for data analysis and manipulation, to compare the profit of each product in the last 3 months and the 3 months before that. It does this by following these steps:

- It defines two variables, Last3Months and Previous3Months, that store the date values for the two periods of interest. It uses the CALCULATETABLE and VALUES functions to filter and return the dates that match the criteria. For example, Last3Months contains the dates that are greater than the end of the fourth month before the maximum date in the table and less than or equal to the end of the first month before the maximum date.
- It evaluates the query by using the ADDCOLUMNS function to add a new column to the result of the SUMMARIZECOLUMNNS function. The SUMMARIZECOLUMNNS function groups the data by the product name and calculates the profit for each product in the last 3 months and the 3 months before using the CALCULATE function and the variables defined earlier. The profit measure is assumed to be defined elsewhere in the model.
- The ADDCOLUMNS function adds a new column called Profit Difference that subtracts the profit in the previous 3 months from the profit in the last 3 months. This shows how much each product's profit has changed over time.
- The query orders the results by the Profit Difference column in descending order, so that the products with the highest increase in profit are shown first.

Show Less ^

Content created by AI may not be accurate or appropriate, so review it carefully. [Read terms](#)

Ahora se entienden cada parte y las funciones DAX usadas.

¿Para qué se usa SUMMARIZECOLUMNNS?

Hablando de estas funciones DAX, sería genial obtener también una explicación de esas funciones. Vamos a probar Copilot para obtener más información sobre una función DAX específica.

En esta solicitud, vamos a averiguar qué hace SUMMARIZECOLUMNNS.

▷ Run

```

1 // DAX query generated by Fabric Copilot with "add a column to show the difference"
2 // DAX query generated by Fabric Copilot with "list the products and show the last 3 months profit compared to the 3 months before"
3 DEFINE
4     // Define the last 3 months period
5     VAR Last3Months =
6         CALCULATETABLE (
7             VALUES ('Date'[Date]),
8             'Date'[Date] > EOMONTH ( MAX ( 'Date'[Date] ), -4 ),
9             'Date'[Date] <= EOMONTH ( MAX ( 'Date'[Date] ), -1 )
10        )
11    // Define the 3 months before the last 3 months period
12    VAR Previous3Months =
13        CALCULATETABLE (
14            VALUES ('Date'[Date]),
15            'Date'[Date] > EOMONTH ( MAX ( 'Date'[Date] ), -7 ),
16            'Date'[Date] <= EOMONTH ( MAX ( 'Date'[Date] ), -4 )
17        )
18 EVALUATE
19     // Calculate the profit for each product in the last 3 months and the 3 months before
20     ADDCOLUMNS (
21         SUMMARIZECOLUMNS (
22             'Fact'[Product],
23             "Last 3 Months Profit", CALCULATE ( [Profit], Last3Months ),
24             "Previous 3 Months Profit", CALCULATE ( [Profit], Previous3Months )
25         ),
26         "Profit Difference", [Last 3 Months Profit] - [Previous 3 Months Profit] // Add a column to show the difference in profit
27     )
28 ORDER BY
29     [Profit Difference] desc
30

```

what is SUMMARIZECOLUMNS used for

Working on it...

Cancel



De nuevo, Copilot genera una breve descripción:

what is SUMMARIZECOLUMNS used for

[Close](#) [↻](#)

SUMMARIZECOLUMNS is a DAX function that returns a table with the specified columns and measures, grouped by the columns and applying any filters or expressions that are specified. It is similar to the GROUP BY clause in SQL, but it allows more flexibility and control over the...

[Show more ▾](#)

Content created by AI may not be accurate or appropriate, so review it carefully. [Read terms](#)

Seleccione Mostrar más.

what is SUMMARIZECOLUMNS used for



Close



SUMMARIZECOLUMNS is a DAX function that returns a table with the specified columns and measures, grouped by the columns and applying any filters or expressions that are specified. It is similar to the GROUP BY clause in SQL, but it allows more flexibility and control over the aggregation and filtering logic.

In the code example, SUMMARIZECOLUMNS is used to create a table with the product name, the profit for the last 3 months, and the profit for the 3 months before the last 3 months. The profit measures are calculated using the CALCULATE function, which applies the filter context from the variables Last3Months and Previous3Months, which are defined using CALCULATETABLE and VALUES. The result is a table like this:

Product	Last 3 Months Profit	Previous 3 Months Profit
A	1000	800
B	900	700
C	800	900
D	700	1000

Show Less ^

Content created by AI may not be accurate or appropriate, so review it carefully. [Read terms](#)

Como le preguntamos a Copilot en el contexto de una consulta DAX que resaltaba, nos ha dado la respuesta en el contexto de esa consulta también.

Obtenga más información con estos recursos

- [Información general de Copilot en Fabric y Power BI](#)
- [Vista de consulta DAX](#)
- [Consultas DAX ↗](#)

Comentarios

¿Le ha resultado útil esta página?

Sí

No

Modelo de ejemplo de DAX

Artículo • 14/12/2023

El modelo de ejemplo Adventure Works DW 2020 de Power BI Desktop está diseñado para admitir el aprendizaje de DAX. El modelo se basa en el [ejemplo de almacenamiento de datos de Adventure Works](#) para AdventureWorksDW2017, pero los datos se modificaron para satisfacer los objetivos del modelo de ejemplo.

El modelo de ejemplo no contiene ninguna fórmula DAX. Sin embargo, admite cientos o, incluso, miles de posibles fórmulas de cálculo y consultas. Algunas funciones de ejemplo, como las de CALCULATE, DATESBETWEEN, DATESIN PERIOD, IF y LOOKUPVALUE, se pueden agregar al modelo de ejemplo sin modificaciones. Estamos trabajando para incluir más ejemplos en otros artículos de referencia de funciones que operan con el modelo de ejemplo.

Escenario



La empresa Adventure Works representa a un fabricante de bicicletas que vende bicicletas y accesorios en mercados globales. La empresa tiene sus datos de almacenamiento de datos almacenados en una instancia de Azure SQL Database.

Estructura del modelo

El modelo tiene siete tablas:

Expandir tabla

Tabla	Descripción
Customer	Describe los clientes y su ubicación geográfica. Los clientes compran productos en línea (ventas por Internet).
Date	Hay tres relaciones entre las tablas Date y Sales, para la fecha del pedido, la fecha de envío y la fecha de vencimiento. La relación de fecha de pedido está activa. Los

Tabla	Descripción
	informes de ventas de la empresa usan un año fiscal que comienza el 1 de julio de cada año. La tabla se marca como una tabla de fechas con la columna Date .
Producto	Almacena solo los productos terminados.
Reseller	Describe los revendedores y su ubicación geográfica. Revendedor en la venta de productos a sus clientes.
Sales	Almacena filas en la línea de pedido de ventas. Todos los valores financieros están en dólares estadounidenses (USD). La fecha de pedido más temprana es el 1 de julio de 2017 y la fecha del último pedido es el 15 de junio de 2020.
Sales Order	Describe los números de pedido de ventas y los números de línea de pedido, además del canal de ventas, que es Revendedor o Internet . Esta tabla tiene una relación uno a uno con la tabla Sales .
Sales Territory	Los territorios de ventas están organizados en grupos (Norteamérica, Europa y Pacífico), países y regiones. Solo el grupo de Estados Unidos vende productos en el nivel de región.

Descarga de un ejemplo

Descargue el archivo de modelo de ejemplo de Power BI Desktop [aquí](#).

Contenido relacionado

- Ruta de aprendizaje: [Uso de DAX en Power BI Desktop](#)
- ¿Tiene alguna pregunta? [Pruebe a preguntar a la comunidad de Power BI](#)
- ¿Sugerencias? [Ideas para contribuir a mejorar Power BI](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Descripción de las funciones ORDERBY, PARTITIONBY y MATCHBY

Artículo • 28/07/2023

Las funciones [ORDERBY](#), [PARTITIONBY](#) y [MATCHBY](#) de DAX son funciones especiales que solo se pueden usar junto con funciones de ventana DAX: [INDEX](#), [OFFSET](#), [WINDOW](#), [RANK](#), [ROWNUMBER](#).

Comprender ORDERBY, PARTITIONBY y MATCHBY es fundamental para usar correctamente las funciones de ventana. En los ejemplos que se proporcionan aquí se usa OFFSET, pero se aplican de forma similar en las demás funciones de ventana.

Escenario

Comencemos con un ejemplo que no usa funciones de ventana en absoluto. A continuación se muestra una tabla que devuelve las ventas totales por color, por año natural. Hay varias maneras de definir esta tabla, pero dado que estamos interesados en comprender lo que sucede en DAX, usaremos una tabla calculada. Esta es la expresión de tabla:

```
DAX

BasicTable =
    SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
    0 )
    )
```

Verá que esta expresión de la tabla calculada usa [SUMMARIZECOLUMNS](#) para calcular la SUMA de la columna SalesAmount en la tabla FactInternetSales, por la columna Color de la tabla DimProduct y la columna CalendarYear de la tabla DimDate. Este es el resultado:

[+] Expandir tabla

Color	CalendarYear	CurrentYearSales
«Negro»	2017	393885

Color	CalendarYear	CurrentYearSales
«Negro»	2018	1818835
«Negro»	2019	3981638
«Negro»	2020	2644054
"Blue"	2019	994448
"Blue"	2020	1284648
«Multi»	2019	48622
«Multi»	2020	57849
«N/D»	2019	207822
«N/D»	2020	227295
«Rojo»	2017	2961198
«Rojo»	2018	3686935
«Rojo»	2019	900175
«Rojo»	2020	176022
«Plata»	2017	326399
«Plata»	2018	750026
«Plata»	2019	2165176
«Plata»	2020	1871788
«Blanco»	2019	2517
«Blanco»	2020	2589
«Amarillo»	2018	163071
«Amarillo»	2019	2072083
«Amarillo»	2020	2621602

Ahora, imaginemos que estamos tratando de resolver la cuestión empresarial de calcular la diferencia en ventas, año a año para cada color. De hecho, necesitamos una manera de encontrar ventas para el mismo color en el año anterior y restar eso de las ventas en el año actual, en contexto. Por ejemplo, para la combinación [Rojo, 2019] estamos buscando ventas para [Rojo, 2018]. Una vez que lo tengamos, podemos restarlo de las ventas actuales y devolver el valor necesario.

Uso de OFFSET

OFFSET es perfecto para la *comparación típica con los tipos anteriores* de cálculos necesarios para responder a la pregunta empresarial descrita anteriormente, ya que nos permite hacer un movimiento relativo. Nuestro primer intento podría ser:

DAX

```
1stAttempt =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
    0 )
)
RETURN
ADDCOLUMNS (
    vRelation,
    "PreviousColorSales",
    SELECTCOLUMNS (
        OFFSET (
            -1,
            vRelation
        ),
        [CurrentYearSales]
    )
)
```

Muchas cosas suceden con esta expresión. Hemos usado **ADDCOLUMNS** para expandir la tabla de antes con una columna denominada PreviousColorSales. El contenido de esa columna se establece en CurrentYearSales, que es SUM(FactInternetSales[SalesAmount]), para el color anterior (recuperado mediante OFFSET).

El resultado es el siguiente:

 Expandir tabla

Color	CalendarYear	CurrentYearSales	PreviousColorSales
«Negro»	2017	393885	
«Negro»	2018	1818835	393885
«Negro»	2019	3981638	1818835
«Negro»	2020	2644054	3981638
"Blue"	2019	994448	2644054

Color	CalendarYear	CurrentYearSales	PreviousColorSales
"Blue"	2020	1284648	994448
«Multi»	2019	48622	1284648
«Multi»	2020	57849	48622
«N/D»	2019	207822	57849
«N/D»	2020	227295	207822
«Rojo»	2017	2961198	227295
«Rojo»	2018	3686935	2961198
«Rojo»	2019	900175	3686935
«Rojo»	2020	176022	900175
«Plata»	2017	326399	176022
«Plata»	2018	750026	326399
«Plata»	2019	2165176	750026
«Plata»	2020	1871788	2165176
«Blanco»	2019	2517	1871788
«Blanco»	2020	2589	2517
«Amarillo»	2018	163071	2589
«Amarillo»	2019	2072083	163071
«Amarillo»	2020	2621602	2072083

Estamos un paso más cerca de nuestro objetivo, pero si miramos bien, no coincide exactamente con lo de después. Por ejemplo, para [Plata, 2017], PreviousColorSales se establece en [Rojo, 2020].

Adición de ORDERBY

Esa definición anterior equivale a:

DAX

```
1stAttemptWithORDERBY =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
```

```

        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
0 )
    )
)
RETURN
ADDCOLUMNS (
    vRelation,
    "PreviousColorSales",
SELECTCOLUMNS (
    OFFSET (
        -1,
        vRelation,
        ORDERBY ([Color], ASC, [CalendarYear], ASC,
[CurrentYearSales], ASC)
    ),
    [CurrentYearSales]
)
)

```

En este caso, la llamada a OFFSET usa ORDERBY para ordenar la tabla por Color y CalendarYear en orden ascendente, que determina lo que se considera la fila anterior que se devuelve.

La razón por la que estos dos resultados son equivalentes es que ORDERBY contiene automáticamente todas las columnas de la relación que no están en PARTITIONBY. Dado que no se especificó PARTITIONBY, ORDERBY se establece en Color, CalendarYear y CurrentYearSales. Sin embargo, dado que los pares Color y CalendarYear en la relación son únicos, agregar CurrentYearSales no cambia el resultado. De hecho, incluso si solo se especificara Color en ORDERBY, los resultados son los mismos, ya que CalendarYear se agregaría automáticamente. Esto se debe a que la función agregará tantas columnas como sea necesario a ORDERBY para asegurarse de que cada fila se pueda identificar de forma única mediante las columnas ORDERBY y PARTITIONBY:

DAX

```

1stAttemptWithORDERBY =
VAR vRelation = SUMMARIZECOLUMNS (
    DimProduct[Color],
    DimDate[CalendarYear],
    "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
0 )
)
)
RETURN
ADDCOLUMNS(
    vRelation,
    "PreviousColorSales",
SELECTCOLUMNS (
    OFFSET (
        -1,

```

```

        vRelation,
        ORDERBY ([Color])
    ),
    [CurrentYearSales]
)
)

```

Adición de PARTITIONBY

Ahora, para obtener *casi* el resultado que buscamos podemos usar PARTITIONBY como se muestra en la siguiente expresión de tabla calculada:

DAX

```

UsingPARTITIONBY =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
        0 )
    )
    RETURN
    ADDCOLUMNS (
        vRelation,
        "PreviousColorSales",
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation,
                ORDERBY ([CalendarYear]),
                PARTITIONBY ([Color])
            ),
            [CurrentYearSales]
        )
    )
)

```

Observe que especificar ORDERBY es opcional aquí porque ORDERBY contiene automáticamente todas las columnas de la relación que no se especifican en PARTITIONBY. Por lo tanto, la siguiente expresión devuelve los mismos resultados porque ORDERBY se establece en CalendarYear y CurrentYearSales automáticamente:

DAX

```

UsingPARTITIONBYWithoutORDERBY =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],

```

```

    "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
0 )
)
RETURN
ADDCOLUMNS (
vRelation,
"PreviousColorSales",
SELECTCOLUMNS (
OFFSET (
-1,
vRelation,
PARTITIONBY ([Color])
),
[CurrentYearSales]
)
)

```

⚠ Nota

Aunque ORDERBY se establece en CalendarYear y CurrentYearSales automáticamente, no se proporciona ninguna garantía sobre el orden en el que se agregarán. Si CurrentYearSales se agrega antes de CalendarYear, el orden resultante no está alineado como se espera. **Sea explícito al especificar ORDERBY y PARTITIONBY para evitar confusiones y resultados inesperados.**

Ambas expresiones devuelven el resultado que buscamos:

[] Expandir tabla

Color	CalendarYear	CurrentYearSales	PreviousYearSalesForSameColor
«Negro»	2017	393885	
«Negro»	2018	1818835	393885
«Negro»	2019	3981638	1818835
«Negro»	2020	2644054	3981638
"Blue"	2019	994448	
"Blue"	2020	1284648	994448
«Multi»	2019	48622	
«Multi»	2020	57849	48622

Color	CalendarYear	CurrentYearSales	PreviousYearSalesForSameColor
«N/D»	2019	207822	
«N/D»	2020	227295	207822
«Rojo»	2017	2961198	
«Rojo»	2018	3686935	2961198
«Rojo»	2019	900175	3686935
«Rojo»	2020	176022	900175
«Plata»	2017	326399	
«Plata»	2018	750026	326399
«Plata»	2019	2165176	750026
«Plata»	2020	1871788	2165176
«Blanco»	2019	2517	
«Blanco»	2020	2589	2517
«Amarillo»	2018	163071	
«Amarillo»	2019	2072083	163071
«Amarillo»	2020	2621602	2072083

Como ve en esta tabla, la columna PreviousYearSalesForSameColor muestra las ventas del año anterior del mismo color. Para [Rojo, 2020], devuelve las ventas de [Rojo, 2019], etc. Si no hay ningún año anterior, por ejemplo, en el caso de [Rojo, 2017], no se devuelve ningún valor.

Puede considerar PARTITIONBY como una manera de dividir la tabla en partes en las que ejecutar el cálculo OFFSET. En el ejemplo anterior, la tabla se divide en tantas partes como colores, una por cada color. A continuación, dentro de cada parte, se calcula el OFFSET, ordenado por CalendarYear.

Visualmente, lo que sucede es esto:

Color	CalendarYear	CurrentYearSales	PreviousYearSalesForSameColor
"Black"	2017	{ 393885 }	
"Black"	2018	{ 1818835 }	393885
"Black"	2019	{ 3981638 }	1818835
"Black"	2020	2644054	3981638
"Blue"	2019	{ 994448 }	
"Blue"	2020	1284648	994448
"Multi"	2019	{ 48622 }	
"Multi"	2020	57849	48622
"NA"	2019	{ 207822 }	
"NA"	2020	227295	207822
"Red"	2017	{ 2961198 }	
"Red"	2018	{ 3686935 }	2961198
"Red"	2019	{ 900175 }	3686935
"Red"	2020	176022	900175
"Silver"	2017	{ 326399 }	
"Silver"	2018	{ 750026 }	326399
"Silver"	2019	{ 2165176 }	750026
"Silver"	2020	1871788	2165176
"White"	2019	{ 2517 }	
"White"	2020	2589	2517
"Yellow"	2018	{ 163071 }	
"Yellow"	2019	{ 2072083 }	163071
"Yellow"	2020	2621602	2072083

En primer lugar, la llamada a PARTITIONBY da como resultado que la tabla se divide en partes, una para cada color. Esto se representa mediante los cuadros azules claros de la imagen de la tabla. A continuación, ORDERBY se asegura de que cada parte está ordenada por CalendarYear (representado por las flechas naranjas). Por último, dentro de cada parte ordenada, para cada fila, OFFSET busca la fila encima de ella y devuelve ese valor en la columna PreviousYearSalesForSameColor. Puesto que para cada primera fila de cada parte no hay ninguna fila anterior en esa misma parte, el resultado de esa fila para la columna PreviousYearSalesForSameColor está vacía.

Para lograr el resultado final, simplemente tenemos que restar CurrentYearSales de las ventas del año anterior para el mismo color devuelto por la llamada a OFFSET. Puesto que no estamos interesados en mostrar las ventas del año anterior para el mismo color, sino solo en las ventas del año actual y en la diferencia año a año. Esta es la expresión de tabla final calculada:

DAX

```

FinalResult =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
        0 )

```

```

)
RETURN
ADDCOLUMNS (
    vRelation,
    "YoYSalesForSameColor",
    [CurrentYearSales] -
SELECTCOLUMNS (
    OFFSET (
        -1,
        vRelation,
        ORDERBY ([CalendarYear]),
        PARTITIONBY ([Color])
    ),
    [CurrentYearSales]
)
)

```

Y este es el resultado de esa expresión:

[] Expandir tabla

Color	CalendarYear	CurrentYearSales	YoYSalesForSameColor
«Negro»	2017	393885	393885
«Negro»	2018	1818835	1424950
«Negro»	2019	3981638	2162803
«Negro»	2020	2644054	-1337584
"Blue"	2019	994448	994448
"Blue"	2020	1284648	290200
«Multi»	2019	48622	48622
«Multi»	2020	57849	9227
«N/D»	2019	207822	207822
«N/D»	2020	227295	19473
«Rojo»	2017	2961198	2961198
«Rojo»	2018	3686935	725737
«Rojo»	2019	900175	-2786760
«Rojo»	2020	176022	-724153

Color	CalendarYear	CurrentYearSales	YoySalesForSameColor
«Plata»	2017	326399	326399
«Plata»	2018	750026	423627
«Plata»	2019	2165176	1415150
«Plata»	2020	1871788	-293388
«Blanco»	2019	2517	2517
«Blanco»	2020	2589	72
«Amarillo»	2018	163071	163071
«Amarillo»	2019	2072083	1909012
«Amarillo»	2020	2621602	549519

Uso de MATCHBY

Es posible que haya observado que no hemos especificado MATCHBY en absoluto. En este caso, no es necesario. Las columnas de ORDERBY y PARTITIONBY (en la medida en que se especificaron en los ejemplos anteriores) son suficientes para identificar de forma única cada fila. Dado que no se ha especificado MATCHBY, las columnas especificadas en ORDERBY y PARTITIONBY se usan para identificar de forma única cada fila para que se puedan comparar para permitir que OFFSET proporcione un resultado significativo. Si las columnas de ORDERBY y PARTITIONBY no pueden identificar de forma única cada fila, se pueden agregar columnas adicionales a la cláusula ORDERBY si esas columnas adicionales permiten identificar cada fila de forma única. Si no es posible, se devuelve un error. En este último caso, especificar MATCHBY puede ayudar a resolver el error.

Si se especifica MATCHBY, las columnas de MATCHBY y PARTITIONBY se usan para identificar de forma única cada fila. Si no es posible, se devuelve un error. Incluso si NO se requiere MATCHBY, considere la posibilidad de especificar explícitamente MATCHBY para evitar cualquier confusión.

Siguiendo con los ejemplos anteriores, esta es la última expresión:

```
DAX
FinalResult =
VAR vRelation = SUMMARIZECOLUMNS (
    DimProduct[Color],
    DimDate[CalendarYear],
    "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
```

```

0 )
)
RETURN
ADDCOLUMNS (
    vRelation,
    "YoYSalesForSameColor",
    [CurrentYearSales] -
SELECTCOLUMNS (
    OFFSET (
        -1,
        vRelation,
        ORDERBY ([CalendarYear]),
        PARTITIONBY ([Color])
    ),
    [CurrentYearSales]
)
)
)

```

Si queremos ser explícitos sobre cómo se deben identificar las filas de forma única, podemos especificar MATCHBY como se muestra en la siguiente expresión equivalente:

DAX

```

FinalResultWithExplicitMATCHBYOnColorAndCalendarYear =
VAR vRelation = SUMMARIZECOLUMNS (
    DimProduct[Color],
    DimDate[CalendarYear],
    "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ) ,
0 )
)
RETURN
ADDCOLUMNS (
    vRelation,
    "YoYSalesForSameColor",
    [CurrentYearSales] -
SELECTCOLUMNS (
    OFFSET (
        -1,
        vRelation,
        ORDERBY ([CalendarYear]),
        PARTITIONBY ([Color]),
        MATCHBY ([Color], [CalendarYear])
    ),
    [CurrentYearSales]
)
)
```

Dado que se especifica MATCHBY, tanto las columnas especificadas en MATCHBY como en PARTITIONBY se usan para identificar filas de forma única. Dado que Color se

especifica en MATCHBY y en PARTITIONBY, la siguiente expresión es equivalente a la expresión anterior:

DAX

```
FinalResultWithExplicitMATCHBYOnCalendarYear =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
0 )
    )
    RETURN
    ADDCOLUMNS (
        vRelation,
        "YoYSalesForSameColor",
        [CurrentYearSales] -
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation,
                ORDERBY ([CalendarYear]),
                PARTITIONBY ([Color]),
                MATCHBY ([CalendarYear])
            ),
            [CurrentYearSales]
        )
    )
)
```

Dado que la especificación de MATCHBY no es necesaria en los ejemplos que hemos examinado hasta ahora, echemos un vistazo a un ejemplo ligeramente diferente que requiere MATCHBY. En este caso, tenemos una lista de líneas de pedido. Cada fila representa una línea de pedido para un pedido. Un pedido puede tener varias líneas de pedido y la línea de pedido 1 aparece en muchos pedidos. Además, para cada línea de pedido tenemos un ProductKey y un SalesAmount. Un ejemplo de las columnas pertinentes de la tabla tiene este aspecto:

[\[\] Expandir tabla](#)

SalesOrderNumber	SalesOrderLineNumber	ProductKey	SalesAmount
SO51900	1	528	4,99
SO51948	1	528	5,99
SO52043	1	528	4,99
SO52045	1	528	4,99

SalesOrderNumber	SalesOrderLineNumber	ProductKey	SalesAmount
SO52094	1	528	4,99
SO52175	1	528	4,99
SO52190	1	528	4,99
SO52232	1	528	4,99
SO52234	1	528	4,99
SO52234	2	529	3,99

Observe que SalesOrderNumber y SalesOrderLineNumber son necesarios para identificar filas de forma única.

Para cada pedido, queremos devolver el importe de ventas anterior del mismo producto (representado por ProductKey) ordenado por SalesAmount en orden descendente. La siguiente expresión no funcionará porque hay potencialmente varias filas en vRelation, ya que se pasa a OFFSET:

DAX

```
ThisExpressionFailsBecauseMATCHBYIsMissing =
ADDCOLUMNS (
    FactInternetSales,
    "Previous Sales Amount",
    SELECTCOLUMNS (
        OFFSET (
            -1,
            FactInternetSales,
            ORDERBY ( FactInternetSales[SalesAmount], DESC ),
            PARTITIONBY ( FactInternetSales[ProductKey] )
        ),
        FactInternetSales[SalesAmount]
    )
)
```

Esta expresión devuelve un error: "El parámetro Relation de OFFSET puede tener filas duplicadas, lo que no es viable".

Para que esta expresión funcione, debe especificarse MATCHBY y debe incluir todas las columnas que definen una fila de forma única. MATCHBY es necesario aquí porque la relación, FactInternetSales, no contiene ninguna clave explícita ni columnas únicas. Sin embargo, las columnas SalesOrderNumber y SalesOrderLineNumber forman una *clave compuesta*, donde su existencia conjunta es única en la relación y, por tanto, pueden

identificar de forma única cada fila. Solo especificar SalesOrderNumber o SalesOrderLineNumber no es suficiente, ya que ambas columnas contienen valores repetidos. La siguiente expresión resuelve el problema:

DAX

```
ThisExpressionWorksBecauseOfMATCHBY =
    ADDCOLUMNS (
        FactInternetSales,
        "Previous Sales Amount",
        SELECTCOLUMNS (
            OFFSET (
                -1,
                FactInternetSales,
                ORDERBY ( FactInternetSales[SalesAmount], DESC ),
                PARTITIONBY ( FactInternetSales[ProductKey] ),
                MATCHBY ( FactInternetSales[SalesOrderNumber],
                          FactInternetSales[SalesOrderLineNumber] )
            ),
            FactInternetSales[SalesAmount]
        )
    )
```

Y esta expresión ya sí devuelve los resultados que buscamos:

 Expandir tabla

SalesOrderNumber	SalesOrderLineNumber	ProductKey	SalesAmount	Importe de ventas anterior
SO51900	1	528	5,99	
SO51948	1	528	4,99	5,99
SO52043	1	528	4,99	4,99
SO52045	1	528	4,99	4,99
SO52094	1	528	4,99	4,99
SO52175	1	528	4,99	4,99
SO52190	1	528	4,99	4,99
SO52232	1	528	4,99	4,99
SO52234	1	528	4,99	4,99
SO52234	2	529	3,99	

Contenido relacionado

[ORDERBY](#)

[PARTITIONBY](#)

[MATCHBY](#)

[INDEX](#)

[OFFSET](#)

[WINDOW](#)

[RANK](#)

[ROWNUMBER](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Uso adecuado de las funciones de error

Artículo • 06/04/2023

Como modelador de datos, al escribir una expresión DAX que podría generar un error en tiempo de evaluación, puede considerar el uso de dos funciones DAX útiles.

- La función **ISERROR**, que toma una sola expresión y devuelve TRUE si se produce un error.
- La función **IFERROR**, que toma dos expresiones. Si la primera expresión genera un error, se devuelve el valor de la segunda expresión. En realidad, es una implementación más optimizada del anidamiento de la función ISERROR dentro de una función **IF**.

Sin embargo, aunque estas funciones pueden ser útiles y pueden contribuir a escribir expresiones fáciles de entender, también pueden degradar significativamente el rendimiento de los cálculos. Puede suceder ya que estas funciones aumentan el número de exámenes del motor de almacenamiento necesarios.

La mayoría de los errores en tiempo de evaluación se deben a valores BLANK o cero inesperados, o bien a una conversión de tipo de datos no válida.

Recomendaciones

Es mejor evitar el uso de las funciones ISERROR e IFERROR. En su lugar, aplique estrategias defensivas al desarrollar el modelo y escribir expresiones. Las estrategias pueden incluir lo siguiente:

- **Garantizar que se cargan datos de calidad en el modelo:** Utilice transformaciones de Power Query para quitar o sustituir valores no válidos o ausentes, y para establecer los tipos de datos correctos. También se puede utilizar una transformación de Power Query para filtrar las filas cuando se producen errores, como la conversión de datos no válida.

También se puede controlar la calidad de los datos si se establece la propiedad **Is Nullable** de la columna del modelo en Off, lo que producirá un error en la actualización de los datos en caso de que se encuentren valores BLANK. Si se produce este error, los datos cargados como resultado de una actualización correcta permanecerán en las tablas.

- **Uso de la función IF:** La expresión de la prueba lógica de la función IF puede determinar si se produciría un resultado de error. Tenga en cuenta que, al igual

que las funciones ISERROR e IFERROR, esta función puede producir exámenes adicionales del motor de almacenamiento, pero es probable que funcione mejor que estos ya que no es necesario generar ningún error.

- **Usar funciones tolerantes a errores:** Algunas funciones de DAX probarán y compensarán las condiciones de error. Estas funciones permiten especificar un resultado alternativo que se devolvería en su lugar. La función [DIVIDE](#) es un ejemplo de este tipo. Para obtener más información sobre esta función, lea el artículo [DAX: función DIVIDE frente al operador de división \(/\)](#).

Ejemplo

La expresión de medida siguiente comprueba si se produciría un error. Devuelve BLANK en esta instancia (que es el caso cuando no se proporciona la función IF con una expresión de valor por si es falso).

DAX

```
Profit Margin  
= IF(ISERROR([Profit] / [Sales]))
```

La siguiente versión de la expresión de medida se ha mejorado mediante el uso de la función IFERROR en lugar de las funciones IF e ISERROR.

DAX

```
Profit Margin  
= IFERROR([Profit] / [Sales], BLANK())
```

Sin embargo, esta versión final de la expresión de medición consigue el mismo resultado, pero de forma más eficaz y elegante.

DAX

```
Profit Margin  
= DIVIDE([Profit], [Sales])
```

Consulte también

- Ruta de aprendizaje: [Uso de DAX en Power BI Desktop](#)
- ¿Tiene alguna pregunta? [Pruebe a preguntar a la comunidad de Power BI ↗](#)
- ¿Sugerencias? [Ideas para contribuir a mejorar Power BI ↗](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

[Obtener ayuda en Microsoft Q&A](#)

Impedimento de la conversión de BLANK en valores

Artículo • 06/04/2023

Como modelador de datos, al escribir expresiones de medida es posible que se encuentre casos en los que no se puede devolver un valor significativo. En tales casos, es posible que le tiente devolver un valor (por ejemplo, cero). Se recomienda decidir detenidamente si este diseño es eficaz y práctico.

Tenga en cuenta la siguiente definición de medida, que convierte de forma explícita los resultados en blanco en cero.

DAX

```
Sales (No Blank) =  
IF(  
    ISBLANK([Sales]),  
    0,  
    [Sales]  
)
```

Tenga en cuenta otra definición de medida, que también convierte los resultados en blanco en cero.

DAX

```
Profit Margin =  
DIVIDE([Profit], [Sales], 0)
```

La función **DIVIDE** divide la medida **Profit** por la medida **Sales**. Si el resultado es cero o está en blanco, se devolverá el tercer argumento: el resultado alternativo (que es opcional). En este ejemplo, dado que se pasa cero como resultado alternativo, se garantiza que la medida siempre devolverá un valor.

Estos diseños de medida son ineficaces y dan lugar a diseños de informes deficientes.

Cuando se agregan a un objeto visual de informe, Power BI intenta recuperar todas las agrupaciones del contexto del filtro. La evaluación y la recuperación de resultados de consultas de gran tamaño suelen comportar una representación lenta de los informes. Cada medida de ejemplo convierte de forma eficaz un cálculo disperso en uno denso, lo que fuerza que Power BI use más memoria de la necesaria.

Además, si hay demasiadas agrupaciones, los usuarios de los informes podrían saturarse.

Veamos lo que sucede cuando se agrega la medida **Profit Margin** a un objeto visual de tabla, agrupando por cliente.

Customer	Sales	Profit Margin
AW00011000		0.00%
AW00011001		0.00%
AW00011002		0.00%
AW00011003		0.00%
AW00011004		0.00%
AW00011005		0.00%
AW00011006		0.00%
AW00011007		0.00%
AW00011008		0.00%
AW00011009		0.00%

El objeto visual de tabla muestra un enorme número de filas (en realidad hay 18 484 clientes en el modelo, por lo que la tabla intenta mostrarlos todos). Tenga en cuenta que los clientes de la vista no han conseguido ninguna venta. Pero como la medida **Profit Margin** siempre devuelve un valor, estos se muestran.

ⓘ Nota

Cuando hay demasiados puntos de datos para mostrarlos en un objeto visual, Power BI puede usar las estrategias de reducción de datos para quitar o resumir los resultados de consultas de gran tamaño. Para obtener más información, consulte [Límites de punto de datos y estrategias por tipo de objeto visual](#).

Veamos lo que sucede cuando se mejora la definición de la medida **Profit Margin**.

Ahora devuelve un valor solo si la medida **Sales** no está en blanco (o cero).

DAX

```
Profit Margin =  
DIVIDE([Profit], [Sales])
```

El objeto visual de tabla ahora muestra solo los clientes que han tenido ventas en el contexto de filtro actual. La medida mejorada da como resultado una experiencia más eficaz y práctica para los usuarios de los informes.

Customer	Sales	Profit Margin
AW00011034	1,264.51	6.72%
AW00011131	427.92	19.59%
AW00024100	1,329.08	21.07%
AW00028194	248.19	11.37%
Total	3,269.70	17.57%

💡 Sugerencia

Si es necesario, puede configurar un objeto visual para mostrar todas las agrupaciones (que devuelven valores o en blanco) dentro del contexto de filtro habilitando la opción **Mostrar elementos sin datos**.

Recomendación

Se recomienda que las medidas devuelvan un valor en blanco si no se puede devolver un valor significativo.

Este enfoque de diseño es eficaz, lo que permite a Power BI representar los informes con mayor rapidez. Además, la devolución de un valor BLANK es más indicada ya que, de forma predeterminada, los objetos visuales de los informes eliminan las agrupaciones cuando los resúmenes están en blanco.

Consulte también

- Ruta de aprendizaje: [Uso de DAX en Power BI Desktop](#)
- ¿Tiene alguna pregunta? [Pruebe a preguntar a la comunidad de Power BI ↗](#)
- ¿Sugerencias? [Ideas para contribuir a mejorar Power BI ↗](#)

Comentarios

¿Le ha resultado útil esta página?

Yes

No

[Obtener ayuda en Microsoft Q&A](#)

Evitar el uso de FILTER como argumento de filtro

Artículo • 06/04/2023

Como modelador de datos, es habitual que escriba expresiones DAX que se deban evaluar en un contexto de filtro modificado. Por ejemplo, puede escribir una definición de medida para calcular las ventas de "productos de margen alto". Este cálculo se describe más adelante.

ⓘ Nota

Este artículo resulta especialmente pertinente para los cálculos de modelos que aplican filtros a las tablas de importación.

Las funciones DAX **CALCULATE** y **CALCULATETABLE** son importantes y útiles. Le permiten escribir cálculos que quitan o agregan filtros, o que modifican las rutas de acceso de las relaciones. Se realiza pasando argumentos de filtro, que son expresiones booleanas, expresiones de tabla o funciones de filtro especiales. En este artículo solo trataremos las expresiones booleanas y de tabla.

Valore la siguiente definición de medida, que calcula las ventas de productos de color rojo mediante una expresión de tabla. Reemplazará todos los filtros que se puedan aplicar a la tabla **Producto**.

DAX

```
Red Sales =  
CALCULATE(  
    [Sales],  
    FILTER('Product', 'Product'[Color] = "Red")  
)
```

La función **CALCULATE** acepta una expresión de tabla devuelta por la función DAX **FILTER**, que evalúa su expresión de filtro para cada fila de la tabla **Producto**. Obtiene el resultado correcto: el de ventas de los productos rojos. Sin embargo, se podría lograr de forma mucho más eficaz mediante el uso de una expresión booleana.

Esta es una definición de medida mejorada, que usa una expresión booleana en lugar de la expresión de tabla. La función DAX **KEEPFILTERS** garantiza que se conservan los filtros existentes aplicados a la columna de **Color** y no se sobrescriban.

DAX

```
Red Sales =  
CALCULATE(  
    [Sales],  
    KEEPFILTERS('Product'[Color] = "Red")  
)
```

Se recomienda pasar argumentos de filtro como expresiones booleanas, siempre que sea posible. Se debe a que las tablas de modelo de importación son almacenes de columnas en memoria. Están optimizadas explícitamente para filtrar de forma eficaz las columnas de esta manera.

Sin embargo, hay restricciones que se aplican a expresiones booleanas cuando se usan como argumentos de filtro. Son las siguientes:

- No se puede hacer referencia a columnas de varias tablas
- No pueden hacer referencia a una medida.
- No pueden usar funciones CALCULATE anidadas.
- No pueden usar funciones que examinen o devuelvan una tabla.

Esto significa que necesitará usar expresiones de tabla para requisitos de filtro más complejos.

Tome ahora una definición de medida diferente. Esta vez, el requisito es calcular las ventas, pero solo para los meses en los que se haya logrado un beneficio.

DAX

```
Sales for Profitable Months =  
CALCULATE(  
    [Sales],  
    FILTER(  
        VALUES('Date'[Month]),  
        [Profit] > 0  
)  
)
```

En este ejemplo se debe usar la función FILTER. Se debe a que es necesario evaluar la medida **Beneficio** para eliminar los meses en los que no se logró un beneficio. No es posible usar una medida en una expresión booleana cuando se usa como un argumento de filtro.

Recomendaciones

Para obtener el mejor rendimiento, se recomienda usar expresiones booleanas como argumentos de filtro, siempre que sea posible.

Por lo tanto, la función FILTER solo debe usarse cuando sea necesario. Puede utilizarla para realizar comparaciones de columnas complejas de filtro. Estas comparaciones de columnas pueden implicar:

- Medidas
- Otras columnas
- Uso de la función DAX **OR** o el operador lógico OR (||)

Consulte también

- [Funciones de filtro \(DAX\)](#)
 - Ruta de aprendizaje: [Uso de DAX en Power BI Desktop](#)
 - ¿Tiene alguna pregunta? [Pruebe a preguntar a la comunidad de Power BI](#) ↗
 - ¿Sugerencias? [Ideas para contribuir a mejorar Power BI](#) ↗
-

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

[Obtener ayuda en Microsoft Q&A](#)

Referencias de columnas y medidas

Artículo • 06/04/2023

Como modelador de datos, las expresiones DAX harán referencia a las columnas y medidas del modelo. Las columnas y las medidas siempre están asociadas a las tablas del modelo, pero estas asociaciones son diferentes, por lo que son diferentes las recomendaciones sobre cómo hacer referencia a ellas en las expresiones.

Columnas

Una columna es un objeto de nivel de tabla y los nombres de columna deben ser únicos en una tabla. Por tanto, se puede usar el mismo nombre de columna varias veces en el modelo, siempre que pertenezca a tablas diferentes. Hay una regla más: un nombre de columna no puede ser el mismo que el de una medida o una jerarquía de la misma tabla.

Por lo general, DAX no forzará el uso de una referencia *completa* a una columna. Una referencia completa significa que el nombre de la tabla precede al nombre de la columna.

Este es un ejemplo de una definición de columna calculada en la que solo se usan referencias de nombre de columna. Las columnas **Sales** y **Cost** pertenecen a una tabla denominada **Orders**.

DAX

```
Profit = [Sales] - [Cost]
```

Se puede volver a escribir la misma definición con referencias de columna completas.

DAX

```
Profit = Orders[Sales] - Orders[Cost]
```

Pero, en ocasiones, se le pedirá que use referencias de columna completas cuando Power BI detecte ambigüedades. Al escribir una fórmula, se le avisará mediante un mensaje de error con una línea ondulada de color rojo. Además, algunas funciones DAX, como [LOOKUPVALUE](#), requieren el uso de columnas completas.

Se recomienda usar siempre nombres completos para las referencias de columna. Los motivos se proporcionan en la sección [Recomendaciones](#).

Medidas

Una medida es un objeto de nivel de modelo. Por esta razón, los nombres de medida deben ser únicos en el modelo. Pero en el panel **Campos**, los autores del informe verán cada medida asociada a una sola tabla de modelo. Esta asociación se establece por motivos cosméticos y puede configurarla si establece la propiedad **Tabla inicial** de la medida. Para obtener más información, vea [Medidas en Power BI Desktop \(organización de las medidas\)](#).

Es posible usar una medida completa en las expresiones. IntelliSense para DAX ofrecerá incluso la sugerencia. Pero no es necesario y no es un procedimiento recomendado. Si cambia la tabla inicial de una medida, se interrumpirán todas las expresiones en las que se use una referencia de medida completa. Después, tendrá que editar cada fórmula interrumpida para quitar (o actualizar) la referencia de la medida.

Se recomienda no calificar nunca las referencias de medidas. Los motivos se proporcionan en la sección [Recomendaciones](#).

Recomendaciones

Nuestras recomendaciones son sencillas y fáciles de recordar:

- Usar siempre nombres completos de referencias de columna
- No usar nunca nombres completos de referencias de medida

Aquí se detallan los motivos:

- **Entrada de fórmula:** se aceptarán expresiones, ya que no habrá ninguna referencia ambigua que resolver. Además, cumplirá los requisitos para las funciones DAX que requieren referencias de columna completas.
- **Solidez:** las expresiones seguirán funcionando, incluso cuando cambie una propiedad de la tabla inicial de la medida.
- **Legibilidad:** las expresiones serán rápidas y fáciles de entender: determinará rápidamente que es una columna o medida, en función de si está completa o no.

Consulte también

- Ruta de aprendizaje: [Uso de DAX en Power BI Desktop](#)
- ¿Tiene alguna pregunta? [Pruebe a preguntar a la comunidad de Power BI ↗](#)
- ¿Sugerencias? [Ideas para contribuir a mejorar Power BI ↗](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

[Obtener ayuda en Microsoft Q&A](#)

Función DIVIDE frente al operador de división (/)

Artículo • 06/04/2023

Como modelador de datos, al escribir una expresión DAX para dividir un numerador por un denominador, puede usar la función **DIVIDE** o el operador de división (/: barra diagonal).

Cuando se usa la función **DIVIDE**, se deben pasar expresiones de numerador y denominador. Opcionalmente, puede pasar un valor que representa un *resultado alternativo*.

DAX

```
DIVIDE(<numerator>, <denominator> [,<alternateresult>])
```

La función **DIVIDE** se ha diseñado para controlar de forma automática los casos de división entre cero. Si no se pasa un resultado alternativo y el denominador es cero o está en blanco, la función devuelve un valor en blanco. Al pasar un resultado alternativo, se devuelve, en lugar de ofrecer un valor en blanco.

La función **DIVIDE** es útil porque guarda la expresión para que no tenga que probar primero el valor del denominador. La función también está mejor optimizada para probar el valor del denominador que la función **IF**. La mejora del rendimiento es importante, ya que la búsqueda de la división entre cero resulta costosa. El uso adicional de **DIVIDE** da lugar a una expresión más concisa y elegante.

Ejemplo

La siguiente expresión de medición genera una división segura, pero implica el uso de cuatro funciones DAX.

DAX

```
Profit Margin =  
IF(  
    OR(  
        ISBLANK([Sales]),  
        [Sales] == 0  
    ),  
    BLANK(),
```

```
[Profit] / [Sales]
```

```
)
```

Esta expresión de medición consigue el mismo resultado, pero de forma más eficaz y elegante.

DAX

```
Profit Margin =  
DIVIDE([Profit], [Sales])
```

Recomendaciones

Se recomienda usar la función DIVIDE siempre que el denominador sea una expresión que *pueda* devolver cero o BLANK.

En el caso de que el denominador sea un valor constante, se recomienda el uso del operador de división. En este caso, la división tiene la garantía de que se realizará correctamente y la expresión funcionará mejor porque evitará pruebas innecesarias.

Considere detenidamente si la función DIVIDE debe devolver un valor alternativo. En el caso de las medidas, normalmente se trata de un mejor diseño, ya que devuelven un valor BLANK. La devolución de un valor BLANK es más indicada ya que, de forma predeterminada, los objetos visuales de los informes eliminan las agrupaciones cuando los resúmenes están en blanco. Esto permite que el objeto visual se centre en los grupos en los que existen los datos. Si es necesario, en Power BI puede configurar el objeto visual para mostrar todos los grupos (que devuelven valores o BLANK) dentro del contexto de filtro si habilita la opción [Mostrar elementos sin datos](#).

Consulte también

- Ruta de aprendizaje: [Uso de DAX en Power BI Desktop](#)
- ¿Tiene alguna pregunta? [Pruebe a preguntar a la comunidad de Power BI](#)
- ¿Sugerencias? [Ideas para contribuir a mejorar Power BI](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

[Obtener ayuda en Microsoft Q&A](#)

Uso de SELECTEDVALUE en lugar de VALUES

Artículo • 06/04/2023

Como modelador de datos, es posible que en ocasiones tenga que escribir una expresión DAX que compruebe si una columna está filtrada por un valor específico.

En versiones anteriores de DAX, este requisito se satisfacía de forma segura con un patrón que implicaba tres funciones DAX: **IF**, **HASONEVALUE** y **VALUES**. La siguiente definición de medida presenta un ejemplo. Calcula el importe del impuesto de ventas, pero solo de las ventas efectuadas a los clientes de Australia.

DAX

```
Australian Sales Tax =  
IF(  
    HASONEVALUE(Customer[Country-Region]),  
    IF(  
        VALUES(Customer[Country-Region]) = "Australia",  
        [Sales] * 0.10  
    )  
)
```

En el ejemplo, la función **HASONEVALUE** devuelve TRUE solo cuando un único valor de la columna **Country-Region** es visible en el contexto de filtro actual. Si es TRUE, la función **VALUES** se compara con el texto literal "Australia". Si la función **VALUES** devuelve TRUE, la medida **Sales** se multiplica por 0,10 (que representa el 10%). Si la función **HASONEVALUE** devuelve FALSE (ya que más de un valor filtra la columna), la primera función **IF** devolverá un valor en blanco.

El uso de **HASONEVALUE** es una técnica defensiva. Es necesario porque es posible que varios valores filtren la columna **Country-Region**. En este caso, la función **VALUES** devuelve una tabla de varias filas. La comparación de una tabla de varias filas con un valor escalar genera un error.

Recomendación

Se recomienda usar la función **SELECTEDVALUE**. Logra el mismo resultado que el patrón que se describe en este artículo, pero de forma más eficaz y elegante.

Con la función **SELECTEDVALUE**, ahora se reescribe la definición de la medida de ejemplo.

DAX

```
Australian Sales Tax =  
IF(  
    SELECTEDVALUE(Customer[Country-Region]) = "Australia",  
    [Sales] * 0.10  
)
```

💡 Sugerencia

Es posible pasar un valor de *resultado alternativo* a la función SELECTEDVALUE. Se devuelve el valor de resultado alternativo cuando no se aplica ningún filtro (o se aplican varios) a la columna.

Consulte también

- Ruta de aprendizaje: [Uso de DAX en Power BI Desktop](#)
- ¿Tiene alguna pregunta? [Pruebe a preguntar a la comunidad de Power BI ↗](#)
- ¿Sugerencias? [Ideas para contribuir a mejorar Power BI ↗](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

[Obtener ayuda en Microsoft Q&A](#)

Uso de COUNTROWS en lugar de COUNT

Artículo • 06/04/2023

Como modelador de datos, en ocasiones es posible que tenga que escribir una expresión DAX que cuente las filas de una tabla. La tabla puede ser una tabla modelo o una expresión que devuelve una tabla.

Su requisito se puede satisfacer de dos formas. Puede usar la función **COUNT** para contar los valores de la columna o bien puede usar la función **COUNTROWS** para contar las filas de la tabla. Ambas funciones lograrán el mismo resultado, siempre y cuando la columna contada no contenga ningún valor en blanco.

La siguiente definición de medida presenta un ejemplo. Calcula el número de valores de columna **OrderDate**.

DAX

```
Sales Orders =  
COUNT(Sales[OrderDate])
```

Siempre que la granularidad de la tabla **Sales** sea una fila por pedido de venta y la columna **OrderDate** no contenga ningún valor en blanco, la medida devolverá un resultado correcto,

pero la siguiente definición de medida es una solución mejor.

DAX

```
Sales Orders =  
COUNTROWS(Sales)
```

Hay tres motivos por los que la segunda definición de medida es mejor:

- Es más eficaz y, por lo tanto, funcionará mejor.
- No tiene en cuenta los valores en blanco incluidos en las columnas de la tabla.
- La intención de la fórmula es más clara, hasta el punto de ser autodescriptiva.

Recomendación

Si su intención es contar las filas de una tabla, se recomienda usar siempre la función COUNTROWS.

Consulte también

- Ruta de aprendizaje: [Uso de DAX en Power BI Desktop](#)
 - ¿Tiene alguna pregunta? [Pruebe a preguntar a la comunidad de Power BI ↗](#)
 - ¿Sugerencias? [Ideas para contribuir a mejorar Power BI ↗](#)
-

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

[Obtener ayuda en Microsoft Q&A](#)

Uso de variables para mejorar las fórmulas DAX

Artículo • 06/04/2023

Como modelador de datos, escribir y depurar algunos cálculos DAX puede ser complejo. Es habitual que los requisitos de cálculos complejos impliquen a menudo la escritura de expresiones compuestas o complejas. Las expresiones compuestas pueden implicar el uso de muchas funciones anidadas y, posiblemente, la reutilización de la lógica de expresión.

El uso de variables en las fórmulas DAX puede ayudarle a escribir cálculos complejos y eficaces. Las variables pueden mejorar el rendimiento, la fiabilidad y la legibilidad, y reducir la complejidad.

En este artículo se mostrarán las tres primeras ventajas con una medida de ejemplo para el crecimiento de ventas de año a año (YoY). (La fórmula del crecimiento de ventas YoY es la siguiente: ventas por periodo, menos ventas del mismo periodo del año pasado, dividido por las ventas del mismo periodo del año pasado).

Comencemos con la siguiente definición de medida.

```
DAX

Sales YoY Growth % =
DIVIDE(
    [Sales] - CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12,
MONTH))),
    CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))
)
```

La medida genera el resultado correcto, pero veamos cómo se puede mejorar.

Mejorar el rendimiento

Observe que la fórmula repite la expresión que calcula "el mismo período del año pasado". Esta fórmula es ineficaz, ya que fuerza a Power BI a evaluar la misma expresión dos veces. La definición de la medida se puede hacer más eficaz usando una variable, VAR.

La siguiente definición de medida representa una mejora. Usa una expresión para asignar el resultado "del mismo período del año pasado" a una variable denominada SalesPriorYear. Entonces, la variable se usa dos veces en la expresión RETURN.

DAX

```
Sales YoY Growth % =  
VAR SalesPriorYear =  
    CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))  
RETURN  
    DIVIDE(([Sales] - SalesPriorYear), SalesPriorYear)
```

La medida sigue generando el resultado correcto y lo hace en más o menos la mitad del tiempo de consulta.

Mejorar la legibilidad

En la definición de medida anterior, observe cómo la elección del nombre de la variable hace que la expresión RETURN sea más fácil de entender. La expresión es breve y autodescriptiva.

Simplificar la depuración

Las variables también pueden ayudarle a depurar una fórmula. Para probar una expresión asignada a una variable, debe reescribir temporalmente la expresión RETURN para generar la variable.

La siguiente definición de medida solo devuelve la variable **SalesPriorYear**. Observe cómo quita la marca de comentario de la expresión RETURN deseada. Con esta técnica puede revertirla fácilmente una vez concluida la depuración.

DAX

```
Sales YoY Growth % =  
VAR SalesPriorYear =  
    CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))  
RETURN  
    --DIVIDE(([Sales] - SalesPriorYear), SalesPriorYear)  
    SalesPriorYear
```

Reducir la complejidad

En versiones anteriores de DAX, las variables aún no eran compatibles. Las expresiones complejas que incorporaban nuevos contextos de filtro eran necesarias para usar las funciones DAX **EARLIER** o **EARLIEST** para hacer referencia a contextos de filtro externos.

Desafortunadamente, para los modeladores de datos estas funciones eran difíciles de comprender y de usar.

Las variables siempre se evalúan fuera de los filtros a los que se aplica la expresión RETURN. Por este motivo, a la hora de usar una variable dentro de un contexto de filtro modificado, se obtiene el mismo resultado que con la función EARLIERST. Por lo tanto, se puede evitar el uso de las funciones EARLIER o EARLIERST. Es decir, ahora puede escribir fórmulas menos complejas y más fáciles de entender.

Observe la siguiente definición de columna calculada que se ha agregado a la tabla **Subcategoría**. Evalúa un rango para cada subcategoría de producto en función de los valores de la columna **Subcategory Sales** (Ventas por subcategoría).

DAX

```
Subcategory Sales Rank =  
COUNTRows(  
    FILTER(  
        Subcategory,  
        EARLIER(Subcategory[Subcategory Sales]) < Subcategory[Subcategory  
Sales]  
    )  
) + 1
```

La función EARLIER sirve para hacer referencia al valor de la columna **Subcategory Sales** en el contexto de fila actual.

La definición de la columna calculada se puede mejorar usando una variable en lugar de la función EARLIER. La variable **CurrentSubcategorySales** almacena el valor de la columna **Subcategory Sales** en el contexto de fila actual y la expresión RETURN la usa en un contexto de filtro modificado.

DAX

```
Subcategory Sales Rank =  
VAR CurrentSubcategorySales = Subcategory[Subcategory Sales]  
RETURN  
    COUNTRows(  
        FILTER(  
            Subcategory,  
            CurrentSubcategorySales < Subcategory[Subcategory Sales]  
        )  
    ) + 1
```

Consulte también

- Artículo DAX [VAR](#)
 - Ruta de aprendizaje: [Uso de DAX en Power BI Desktop](#)
 - ¿Tiene alguna pregunta? [Pruebe a preguntar a la comunidad de Power BI](#) ↗
-

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

[Obtener ayuda en Microsoft Q&A](#)

Referencia de funciones DAX

Artículo • 20/10/2023

La referencia de funciones de DAX proporciona información detallada (como la sintaxis, los parámetros, los valores devueltos y ejemplos) de cada una de las más de 250 funciones usadas en las fórmulas de Expresiones de análisis de datos (DAX).

ⓘ Importante

No todas las funciones DAX se admiten o incluyen en las versiones anteriores de Power BI Desktop, Analysis Services y Power Pivot en Excel.

En esta sección

[Nuevas funciones DAX](#): estas funciones son nuevas o son funciones existentes que se han actualizado de forma significativa.

[Funciones de agregación](#): estas funciones calculan un valor (escalar) como count, sum, average, minimum o maximum para todas las filas de una columna o tabla, según se define en la expresión.

[Funciones de fecha y hora](#): estas funciones en DAX son similares a las funciones de fecha y hora de Microsoft Excel, si bien las funciones DAX se basan en los tipos de datos datetime usados por Microsoft SQL Server.

[Funciones de filtro](#): estas funciones ayudan a devolver tipos de datos específicos, buscar valores en tablas relacionadas y filtrar por valores relacionados. Las funciones de búsqueda funcionan usando tablas y relaciones entre ellas. Las funciones de filtro permiten manipular el contexto de datos para crear cálculos dinámicos.

[Funciones financieras](#): estas funciones se usan en fórmulas que realizan cálculos financieros, como el valor neto presente y la tasa de devolución.

[Funciones de información](#): estas funciones examinan una tabla o columna proporcionada como argumento para otra función y devuelven resultados si el valor coincide con el tipo esperado. Por ejemplo, la función ISERROR devuelve TRUE si el valor al que se hace referencia contiene un error.

[Funciones lógicas](#): estas funciones devuelven información sobre los valores de una expresión. Por ejemplo, la función TRUE permite saber si una expresión que se está evaluando devuelve un valor TRUE.

Funciones matemáticas y trigonométricas: las funciones matemáticas en DAX son similares a las funciones matemáticas y trigonométricas de Excel, si bien existen algunas diferencias en los tipos de datos numéricos que las funciones DAX usan.

Otras funciones: estas funciones realizan acciones únicas que no se pueden definir por medio de ninguna de las categorías a las que la mayoría de las otras funciones pertenecen.

Funciones primarias y secundarias: estas funciones ayudan a los usuarios a administrar los datos que se presentan como una jerarquía de elementos primarios y secundarios en sus modelos de datos.

Funciones de relación: estas funciones sirven para administrar y usar las relaciones entre las tablas. Por ejemplo, se puede especificar una relación determinada que vaya a usarse en un cálculo.

Funciones estadísticas: estas funciones calculan valores relacionados con las distribuciones estadísticas y la probabilidad, como la desviación estándar y el número de permutaciones.

Funciones de manipulación de tablas: estas funciones devuelven una tabla o manipulan tablas existentes.

Funciones de texto: con estas funciones, se puede devolver parte de una cadena, buscar texto dentro de una cadena o concatenar valores de cadena. Existen más funciones para controlar los formatos de fechas, horas y números.

Funciones de inteligencia de tiempo: estas funciones ayudan a crear cálculos que usan el conocimiento integrado sobre calendarios y fechas. El uso de intervalos de fecha y de hora en combinación con agregaciones o cálculos permite crear comparaciones significativas a lo largo de períodos de tiempo comparables relativos a ventas, inventarios, etc.

Vea también

[Referencia de sintaxis de DAX](#)

[Referencia de operadores DAX](#)

[Convenciones de nomenclatura de parámetros DAX](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

Nuevas funciones de DAX

Artículo • 22/02/2024

DAX mejora continuamente con nuevas funciones y funcionalidades para admitir nuevas características. Las nuevas funciones y actualizaciones se incluyen en las actualizaciones del servicio, la aplicación y las herramientas que, en la mayoría de los casos, son mensuales.

Aunque las funciones y la funcionalidad se actualizan continuamente, solo se describen en la documentación aquellas actualizaciones que implican un cambio visible y funcional para los usuarios. Aquí se muestran nuevas funciones y actualizaciones de las funciones existentes en el año pasado.

ⓘ Importante

No todas las funciones se admiten en todas las versiones de Power BI Desktop, Analysis Services y Power Pivot en Excel. Normalmente, las funciones nuevas y actualizadas se presentan primero en Power BI Desktop y, después, se incluyen en Analysis Services, Power Pivot en Excel y herramientas.

Nuevas funciones

 Expandir tabla

Función	Month (Mes)	Descripción
FIRST	Enero de 2024	Solo se usa en cálculos visuales. Recupera un valor de la matriz visual a partir de la primera fila de un eje.
LAST	Enero de 2024	Solo se usa en cálculos visuales. Recupera un valor de la matriz visual de la última fila de un eje.
SIGUIENTE	Enero de 2024	Solo se usa en cálculos visuales. Recupera un valor de la siguiente fila de un eje de la matriz visual.
PREVIOUS	Enero de 2024	Solo se usa en cálculos visuales. Recupera un valor de la fila anterior de un eje de la matriz visual.
MATCHBY	Mayo de 2023	Define las columnas que se usan para buscar coincidencias con los datos e identificar la fila actual, en una expresión de función window.

Función	Month (Mes)	Descripción
RANK	Abril de 2023	Devuelve la clasificación del contexto actual dentro de la partición especificada ordenada según lo especificado.
ROWNUMBER	Abril de 2023	Devuelve la clasificación única del contexto actual dentro de la partición especificada ordenada según lo especificado.
LINEST	Febrero, 2023	Usa el método de mínimos cuadrados para calcular una línea recta con el mejor ajuste a los datos especificados.
LINESTX	Febrero, 2023	Usa el método de mínimos cuadrados para calcular una línea recta con el mejor ajuste a los datos especificados. Los datos se obtienen de las expresiones evaluadas para cada fila de una tabla.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Funciones de agregación

Artículo • 20/10/2023

Las funciones de agregación calculan un valor (escalar) como count, sum, average, minimum o maximum para todas las filas de una columna o tabla, según se define en la expresión.

En esta categoría

Función	Descripción
APPROXIMATEDISTINCTCOUNT	Devuelve un recuento <i>estimado</i> de valores únicos en una columna.
AVERAGE	Devuelve el promedio (media aritmética) de todos los números de una columna.
AVERAGEA	Devuelve el promedio (media aritmética) de los valores de una columna.
AVERAGEX	Calcula el promedio (media aritmética) de un conjunto de expresiones evaluadas en una tabla.
COUNT	Cuenta el número de filas de la columna especificada que contienen valores que no están en blanco.
COUNTA	Cuenta el número de filas de la columna especificada que contienen valores que no están en blanco.
COUNTAX	Cuenta los resultados que no están en blanco al evaluar el resultado de una expresión en una tabla.
COUNTBLANK	Cuenta el número de celdas en blanco de una columna.
COUNTROWS	Cuenta el número de filas de la tabla especificada o de una tabla definida por una expresión.
COUNTX	Cuenta el número de filas que contienen un número o una expresión que se evalúa como un número, al evaluar una expresión en una tabla.
DISTINCTCOUNT	Cuenta el número de valores distintos de una columna.
DISTINCTCOUNTNOBLANK	Cuenta el número de valores distintos de una columna.
MAX	Devuelve el valor numérico mayor de una columna, o entre dos expresiones escalares.

Función	Descripción
MAXA	Devuelve el valor mayor de una columna.
MAXX	Evalúa una expresión para cada fila de una tabla y devuelve el valor numérico mayor.
MIN	Devuelve el valor numérico menor de una columna, o entre dos expresiones escalares.
MINA	Devuelve el valor menor de una columna, incluidos los valores lógicos y los números representados como texto.
MINX	Devuelve el valor numérico menor que se obtiene al evaluar una expresión para cada fila de una tabla.
PRODUCT	Devuelve el producto de los números de una columna.
PRODUCTX	Devuelve el producto de una expresión evaluada para cada fila de una tabla.
SUM	Suma todos los números de una columna.
SUMX	Devuelve la suma de una expresión evaluada para cada fila de una tabla.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

APPROXIMATEDISTINCTCOUNT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve un recuento *estimado* de valores únicos en una columna. Esta función invoca una operación de agregación correspondiente en el origen de datos, que está optimizada para el rendimiento de las consultas, pero con una precisión ligeramente reducida. Esta función se puede usar con los siguientes orígenes de datos: Azure SQL, grupo de SQL dedicado de Azure Synapse, BigQuery, Databricks y Snowflake. Esta función requiere el modo DirectQuery. No se admite el modo de importación ni el modo de almacenamiento dual.

Sintaxis

DAX

```
APPROXIMATEDISTINCTCOUNT(<columnName>)
```

Parámetros

Expandir tabla

Término	Descripción
columna	Columna que contiene los valores que se van a contar. No puede ser una expresión.

Valor devuelto

Número aproximado de valores distintos de *column*.

Observaciones

El único argumento para esta función es una columna. Puede usar columnas que contengan cualquier tipo de datos. Si la función no encuentra ninguna fila que contar, devuelve el valor BLANK; de lo contrario, devuelve el recuento de valores distintos.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

AVERAGE

Artículo • 20/10/2023

Devuelve el promedio (media aritmética) de todos los números de una columna.

Sintaxis

DAX

`AVERAGE(<column>)`

Parámetros

Término	Definición
columna	Columna que contiene los números cuyo promedio se quiere obtener.

Valor devuelto

Devuelve un número decimal que representa la media aritmética de los números de la columna.

Observaciones

- Esta función toma la columna especificada como argumento y busca el promedio de los valores de esa columna. Si quiere buscar el promedio de una expresión que se evalúa como un conjunto de números, use en su lugar la función AVERAGEX.
- Los valores no numéricos de la columna se tratan de la siguiente manera:
 - Si la columna contiene texto, no se puede realizar ninguna agregación y las funciones devuelven espacios en blanco.
 - Si la columna contiene valores lógicos o celdas vacías, dichos valores se omiten.
 - Se incluyen las celdas con el valor cero.
- Al calcular el promedio de las celdas, debe tener en cuenta la diferencia entre una celda vacía y una celda que contiene el valor 0 (cero). Si una celda contiene 0, se agrega a la suma de números y la fila se cuenta entre el número de filas que se ha usado como divisor. En cambio, si una celda contiene un espacio en blanco, no se cuenta la fila.

- Si no hay filas que agregar, la función devuelve un espacio en blanco. Pero si hay filas, y ninguna de ellas cumple los criterios especificados, la función devuelve 0. Excel también devuelve cero si no se encuentra ninguna fila que cumpla las condiciones.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula siguiente devuelve el promedio de los valores de la columna ExtendedSalesAmount de la tabla InternetSales.

DAX

```
= AVERAGE(InternetSales[ExtendedSalesAmount])
```

Funciones relacionadas

La función AVERAGEX puede tomar como argumento una expresión que se evalúa para cada fila de una tabla. Esto le permite realizar cálculos y después tomar el promedio de los valores calculados.

La función AVERAGEA toma una columna como argumento, pero, por lo demás, es como la función de Excel del mismo nombre. Al usar la función AVERAGEA, puede calcular una media en una columna que contiene valores vacíos.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

AVERAGEA

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el promedio (media aritmética) de los valores de una columna. Controla valores de texto y no numéricos.

Sintaxis

DAX

`AVERAGEA(<column>)`

Parámetros

Expandir tabla

Término	Definición
columna	Columna que contiene los valores cuyo promedio quiere obtener.

Valor devuelto

Número decimal.

Notas

- La función AVERAGEA toma una columna y calcula el promedio de los números que contiene, pero también trata tipos de datos no numéricos de acuerdo con las siguientes reglas:
 - Los valores que se evalúan como TRUE cuentan como 1.
 - Los valores que se evalúan como FALSE cuentan como 0 (cero).
 - Los valores que contienen texto no numérico cuentan como 0 (cero).
 - El texto vacío ("") cuenta como 0 (cero).
- Si no quiere incluir valores lógicos y representaciones de texto de los números en una referencia como parte del cálculo, use la función AVERAGE.

- Si no hay filas que agregar, la función devuelve un espacio en blanco. Pero si hay filas, y ninguna de ellas cumple los criterios especificados, la función devuelve 0. Microsoft Excel también devuelve cero si no se encuentra ninguna fila que cumpla las condiciones.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se devuelve el promedio de las celdas que no están en blanco en la columna a la que se hace referencia, dada la siguiente tabla. Si ha usado la función AVERAGE, la media sería 21/2; con la función AVERAGEA, el resultado es 22/5.

 Expandir tabla

Identificador de transacción	Cantidad	Resultado
0000123	1	Cuenta como 1
0000124	20	Cuenta como 20
0000125	n/d	Cuenta como 0
0000126		Cuenta como 0
0000126	VERDADERO	Cuenta como 1

DAX

```
= AVERAGEA([Amount])
```

Contenido relacionado

[Función AVERAGE](#)

[AVERAGEX, función](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

AVERAGEX

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Calcula el promedio (media aritmética) de un conjunto de expresiones evaluadas en una tabla.

Sintaxis

DAX

```
AVERAGEX(<table>,<expression>)
```

Parámetros

Expandir tabla

Término	Definición
tabla	Nombre de una tabla, o una expresión que especifica la tabla, en la que se puede realizar la agregación.
expression	Expresión con un resultado escalar que se va a evaluar para cada fila de la tabla en el primer argumento.

Valor devuelto

Número decimal.

Notas

- La función AVERAGEX permite evaluar expresiones para cada fila de una tabla y luego tomar el conjunto de valores resultante y calcular su media aritmética. Por lo tanto, la función toma una tabla como primer argumento y una expresión como segundo argumento.
- En todos los demás aspectos, AVERAGEX sigue las mismas reglas que AVERAGE. No se pueden incluir celdas no numéricas ni nulas. Los argumentos de expresión y tabla son obligatorios.

- Si no hay filas para agregar, la función devuelve un espacio en blanco. Si hay filas, pero ninguna de ellas cumple los criterios especificados, la función devuelve 0.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se calcula el flete y el impuesto medios de cada pedido de la tabla InternetSales, para lo que primero se suman Freight y TaxAmt en cada fila y luego se calcula la media de esas sumas.

DAX

```
= AVERAGEX(InternetSales, InternetSales[Freight]+ InternetSales[TaxAmt])
```

Si usa varias operaciones en la expresión empleada como segundo argumento, debe usar paréntesis para controlar el orden de los cálculos. Para obtener más información, vea [Sintaxis de DAX](#).

Contenido relacionado

[Función AVERAGE](#)

[Función AVERAGEA](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COUNT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Cuenta el número de filas de la columna especificada que contienen valores que no están en blanco.

Sintaxis

DAX

`COUNT(<column>)`

Parámetros

Expandir tabla

Término	Definición
columna	Columna que contiene los valores que se van a contar.

Valor devuelto

Número entero.

Notas

- El único argumento permitido para esta función es una columna. La función COUNT cuenta las filas que contienen los siguientes tipos de valores:
 - Números
 - Fechas
 - Cadenas
- Si la función no encuentra ninguna fila que contar, devuelve un espacio en blanco.
- Los valores en blanco se omiten. Los valores TRUE/FALSE no se admiten.
- Si quiere evaluar una columna de valores TRUE/FALSE, use la función COUNTA.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).
- Para conocer los procedimientos recomendados al usar COUNT, consulte [Uso de COUNTROWS en lugar de COUNT](#).

Ejemplo

En el siguiente ejemplo se muestra cómo contar el número de valores de la columna ShipDate.

DAX

```
= COUNT([ShipDate])
```

Para contar valores lógicos o texto, use las funciones COUNTA o COUNTAX.

Contenido relacionado

[Función COUNTA](#)

[Función COUNTAX](#)

[COUNTX, función](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COUNTA

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Cuenta el número de filas de la columna especificada que contienen valores que no están en blanco.

Sintaxis

DAX

`COUNTA(<column>)`

Parámetros

[+] Expandir tabla

Término	Definición
columna	Columna que contiene los valores que se van a contar.

Valor devuelto

Número entero.

Notas

- Si la función no encuentra ninguna fila para contar, devuelve un espacio en blanco.
- A diferencia de [COUNT](#), COUNTA admite el tipo de datos booleano.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se devuelven todas las filas de la tabla `Reseller` que tienen cualquier tipo de valor en la columna que almacena números de teléfono.

DAX

= COUNTA(Reseller[Phone])

Contenido relacionado

[Función COUNT](#)

[Función COUNTAX](#)

[COUNTX, función](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COUNTAX

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

La función COUNTAX cuenta los resultados no vacíos al evaluar el resultado de una expresión en una tabla. Es decir, funciona igual que la función COUNTA, pero se usa para recorrer en iteración las filas de una tabla y contar las filas en las que las expresiones especificadas dan un resultado que no está en blanco.

Sintaxis

DAX

`COUNTAX(<table>,<expression>)`

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla que contiene las filas para las que se evaluará la expresión.
expresión	Expresión que se debe evaluar para cada fila de la tabla.

Valor devuelto

Número entero.

Notas

- Al igual que la función COUNTA, la función COUNTAX cuenta las celdas que contienen cualquier tipo de información, incluidas otras expresiones. Por ejemplo, si la columna contiene una expresión que se evalúa como una cadena vacía, la función COUNTAX considera que ese resultado no está en blanco. Normalmente, la función COUNTAX no cuenta las celdas vacías, pero en este caso la celda contiene una fórmula, de modo que sí se cuenta.
- Si la función no encuentra filas que agregar, devuelve un espacio en blanco.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En este ejemplo se cuenta el número de filas que no están en blanco en la columna Phone, usando la tabla que resulta de filtrar la tabla Reseller en [Status] = **Activo**.

DAX

```
= COUNTAX(FILTER('Reseller',[Status]="Active"),[Phone])
```

Contenido relacionado

[Función COUNT](#)

[Función COUNTA](#)

[COUNTX, función](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COUNTBLANK

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Cuenta el número de celdas en blanco de una columna.

Sintaxis

DAX
<code>COUNTBLANK(<column>)</code>

Parámetros

Expandir tabla

Término	Definición
columna	Columna que contiene las celdas en blanco que se van a contar.

Valor devuelto

Número entero. Si no se encuentra ninguna fila que cumpla las condiciones, se devuelven espacios en blanco.

Notas

- El único argumento permitido para esta función es una columna. Puede usar columnas que contengan cualquier tipo de datos, pero solo se cuentan las celdas en blanco. No se cuentan las celdas que tienen el valor cero (0), puesto que cero se considera un valor numérico y no un espacio en blanco.
- Si no hay filas que agregar, la función devuelve un espacio en blanco. Pero si hay filas, y ninguna de ellas cumple los criterios especificados, la función devuelve 0. Microsoft Excel también devuelve cero si no se encuentra ninguna fila que cumpla las condiciones.

- En otras palabras, si la función COUNTBLANK no encuentra espacios en blanco, el resultado será cero; pero, si no hay ninguna fila que comprobar, el resultado estará en blanco.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En este ejemplo se muestra cómo contar el número de filas de la tabla Reseller que tienen valores en blanco en BankName.

DAX

```
= COUNTBLANK(Reseller[BankName])
```

Para contar valores lógicos o texto, use las funciones COUNTA o COUNTAX.

Contenido relacionado

[Función COUNT](#)

[Función COUNTA](#)

[Función COUNTAX](#)

[COUNTX, función](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COUNTROWS

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

La función COUNTROWS cuenta el número de filas de la tabla especificada o de una tabla definida por una expresión.

Sintaxis

DAX

`COUNTROWS([<table>])`

Parámetros

Expandir tabla

Término	Definición
tabla	(Opcional) Nombre de la tabla que contiene las filas que se van a contar o una expresión que devuelve una tabla. Cuando no se proporciona, el valor predeterminado es la tabla principal de la expresión actual.

Valor devuelto

Número entero.

Notas

- Esta función se puede usar para contar el número de filas de una tabla base, pero se usa con más frecuencia para contar el número de filas resultante del filtrado de una tabla o de la aplicación de contexto a una tabla.
- Cuando el argumento tabla no contiene filas, la función devuelve BLANK.
- Para más información sobre los procedimientos recomendados al usar COUNT y COUNTROWS, consulte [Uso de COUNTROWS en lugar de COUNT en DAX](#).

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

En este ejemplo se muestra cómo contar el número de filas de la tabla Orders. El resultado esperado es 52 761.

DAX

```
= COUNTROWS('Orders')
```

Ejemplo 2

En este ejemplo se muestra cómo usar COUNTROWS con un contexto de fila. En este escenario, hay dos conjuntos de datos que están relacionados por número de pedido. La tabla Reseller contiene una fila por cada distribuidor; la tabla ResellerSales contiene varias filas para cada pedido, cada una de las cuales contiene un pedido para un distribuidor determinado. Las tablas están conectadas por una relación en la columna ResellerKey.

La fórmula obtiene el valor de ResellerKey y luego cuenta el número de filas de la tabla relacionada que tienen el mismo identificador de distribuidor. El resultado se muestra en la columna **CalculatedColumn1**.

DAX

```
= COUNTROWS(RELATEDTABLE(ResellerSales))
```

En esta tabla se muestra una parte de los resultados esperados:

 Expandir tabla

ResellerKey	CalculatedColumn1
1	73
2	70
3	394

Contenido relacionado

[Función COUNT](#)

[Función COUNTA](#)

[Función COUNTAX](#)

[COUNTX, función](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COUNTX

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Cuenta el número de filas que contienen un valor que no está en blanco o una expresión que se evalúa como un valor que no está en blanco, al evaluar una expresión en una tabla.

Sintaxis

DAX

`COUNTX(<table>,<expression>)`

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla que contiene las filas que se van a contar.
expression	Expresión que devuelve el conjunto de valores que contiene los valores que quiere contar.

Valor devuelto

Entero.

Notas

- La función COUNTX toma dos argumentos. El primero siempre debe ser una tabla o cualquier expresión que devuelva una tabla. El segundo es la columna o expresión que busca COUNTX.
- La función COUNTX solo cuenta valores, fechas o cadenas. Si la función no encuentra ninguna fila que contar, devuelve un valor en blanco.
- Si quiere contar valores lógicos, use la función COUNTAX.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

La fórmula siguiente devuelve un recuento de todas las filas de la tabla Product que tienen un precio de venta.

DAX

```
= COUNTX(Product,[ListPrice])
```

Ejemplo 2

La fórmula siguiente muestra cómo pasar una tabla filtrada a COUNTX para el primer argumento. La fórmula usa una expresión de filtro para obtener solo las filas de la tabla Product que cumplen la condición ProductSubCategory = "Caps" y, después, cuenta las filas de la tabla resultante que tienen un precio de venta. La expresión FILTER se aplica a la tabla Products, pero usa un valor que se busca en la tabla relacionada ProductSubCategory.

DAX

```
=  
COUNTX(FILTER(Product, RELATED(ProductSubcategory[EnglishProductSubcategoryNa  
me])="Caps"), Product[ListPrice])
```

Contenido relacionado

[Función COUNT](#)

[Función COUNTA](#)

[Función COUNTAX](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DISTINCTCOUNT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Cuenta el número de valores distintos de una columna.

Sintaxis

DAX
<code>DISTINCTCOUNT(<column>)</code>

Parámetros

Expandir tabla

Término	Descripción
columna	Columna que contiene los valores que se van a contar.

Valor devuelto

Número de valores distintos en *column*.

Notas

- El único argumento permitido para esta función es una columna. Puede usar columnas que contengan cualquier tipo de datos. Si la función no encuentra ninguna fila que contar, devuelve el valor BLANK; de lo contrario, devuelve el recuento de valores distintos.
- La función DISTINCTCOUNT cuenta los valores BLANK. Para omitir el valor BLANK, use la función [DISTINCTCOUNTNOBLANK](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se muestra cómo contar el número de diferentes pedidos de ventas en la columna ResellerSales_USD[SalesOrderNumber].

DAX

```
= DISTINCTCOUNT(ResellerSales_USD[SalesOrderNumber])
```

Si se usa la medida anterior en una tabla con el año en el lado y la categoría de producto en la parte superior, se devuelven los siguientes resultados:

 Expandir tabla

Etiquetas de fila	Accesorios	Bicicletas	Ropa	Componentes	-	Total general
2005	135	345	242	205		366
2006	356	850	644	702		1015
2007	531	1234	963	1138		1521
2008	293	724	561	601		894
					1	1
Total general	1315	3153	2410	2646	1	3797

En el ejemplo anterior, tenga en cuenta que la suma de los números de Total general de las filas no es correcta; esto sucede porque en el mismo pedido puede haber elementos de línea, en el mismo orden, de varias categorías de producto.

Contenido relacionado

[Función COUNT](#)

[Función COUNTA](#)

[Función COUNTAX](#)

[COUNTX, función](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DISTINCTCOUNTNOBLANK

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Cuenta el número de valores distintos de una columna.

Sintaxis

DAX

`DISTINCTCOUNTNOBLANK (<column>)`

Parámetros

Expandir tabla

Término	Descripción
columna	Columna que contiene los valores que se van a contar.

Valor devuelto

Número de valores distintos en *column*.

Notas

- A diferencia de la función [DISTINCTCOUNT](#), DISTINCTCOUNTNOBLANK no cuenta el valor BLANK.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se muestra cómo contar el número de diferentes pedidos de ventas en la columna ResellerSales_USD[SalesOrderNumber].

DAX

```
= DISTINCTCOUNT(ResellerSales_USD[SalesOrderNumber])
```

Consulta DAX

DAX

```
EVALUATE  
ROW(  
    "DistinctCountNoBlank", DISTINCTCOUNTNOBLANK(DimProduct[EndDate]),  
    "DistinctCount", DISTINCTCOUNT(DimProduct[EndDate])  
)
```

 Expandir tabla

[DistinctCountNoBlank]	[DistinctCount]
2	3

Contenido relacionado

[DISTINCTCOUNT](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MÁX

Artículo • 02/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el valor mayor de una columna, o entre dos expresiones escalares.

Sintaxis

DAX
<code>MAX(<column>)</code>

DAX
<code>MAX(<expression1>, <expression2>)</code>

Parámetros

Expandir tabla

Término	Definición
columna	Columna en la que se quiere buscar el valor mayor.
expression	Cualquier expresión DAX que devuelva un valor único.

Valor devuelto

Valor mayor.

Notas

- Al comparar dos expresiones, el espacio en blanco se trata como 0. Es decir, Max(1, Blank()) devuelve 1 y Max(-1, Blank()) devuelve -0. Si ambos argumentos están en blanco, MAX devuelve un espacio en blanco. Si alguna de las expresiones devuelve un valor no permitido, MAX devuelve un error.
- No se admiten los valores TRUE o FALSE. Si quiere evaluar una columna de valores TRUE/FALSE, use la función MAXA.

Ejemplo 1

En el ejemplo siguiente se devuelve el valor mayor encontrado en la columna ExtendedAmount de la tabla InternetSales.

DAX

```
= MAX(InternetSales[ExtendedAmount])
```

Ejemplo 2

En el ejemplo siguiente se devuelve el valor mayor entre el resultado de dos expresiones.

DAX

```
= Max([TotalSales], [TotalPurchases])
```

Contenido relacionado

[MAXA, función](#)

[MAXX, función](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MAXA

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el valor mayor de una columna.

Sintaxis

DAX
<code>MAXA(<column>)</code>

Parámetros

Expandir tabla

Término	Definición
columna	Columna en la que se quiere buscar el valor mayor.

Valor devuelto

Valor mayor.

Notas

- La función MAXA toma como argumento una columna y busca el valor mayor entre los siguientes tipos de valores:
 - Números
 - Fechas
- Valores lógicos como TRUE y FALSE. Las filas que se evalúan como TRUE cuentan como 1; las que se evalúan como FALSE cuentan como 0 (cero).
- Las celdas vacías se omiten. Si la columna no contiene valores que se puedan usar, MAXA devuelve 0 (cero).
- Si quiere comparar valores de texto, use la función MAX.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

En el ejemplo siguiente se devuelve el valor mayor de una columna calculada denominada **ResellerMargin** que calcula la diferencia entre el precio de venta y el precio del distribuidor.

DAX

```
= MAXA([ResellerMargin])
```

Ejemplo 2

En el ejemplo siguiente se devuelve el valor mayor de una columna que contiene fechas y horas. Por lo tanto, esta fórmula obtiene la fecha de transacción más reciente.

DAX

```
= MAXA([TransactionDate])
```

Contenido relacionado

[Función MAX](#)

[MAXX, función](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MAXX

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el valor máximo que se obtiene al evaluar una expresión para cada fila de una tabla.

Sintaxis

DAX

```
MAXX(<table>,<expression>,[<variant>])
```

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla que contiene las filas para las que se evaluará la expresión.
expresión	Expresión que se debe evaluar para cada fila de la tabla.
variant	(Opcional) Si es TRUE y si hay tipos de valores variantes o mixtos, se devuelve el valor máximo basado en ORDER BY DESC.

Valor devuelto

El valor más alto.

Observaciones

- El argumento **table** para la función MAXX puede ser un nombre de tabla o una expresión que se evalúa como una tabla. El segundo argumento indica la expresión que se va a evaluar para cada fila de la tabla.
- De los valores que se van a evaluar, solo se cuentan los siguientes:
 - Números
 - Textos

- Fechas
- Los valores en blanco se omiten. Los valores TRUE/FALSE no se admiten.
- Si la expresión tiene tipos de valores mixtos o variantes, como texto y número, MAXX solo tiene en cuenta números de forma predeterminada. Si es <variant> = TRUE, se devuelve el valor máximo.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

La siguiente fórmula usa una expresión como segundo argumento para calcular el importe total de impuestos y gastos de envío de cada pedido de la tabla InternetSales. El resultado esperado es 375,7184.

DAX

```
= MAXX(InternetSales, InternetSales[TaxAmt]+ InternetSales[Freight])
```

Ejemplo 2

La siguiente fórmula primero filtra la tabla InternetSales mediante una expresión FILTER para devolver un subconjunto de pedidos de una región de ventas específica definida como [SalesTerritory] = 5. Luego la función MAXX evalúa la expresión usada como segundo argumento para cada fila de la tabla filtrada y devuelve el importe más alto de impuestos y gastos de envío solo de esos pedidos. El resultado esperado es 250,3724.

DAX

```
= MAXX(FILTER(InternetSales,[SalesTerritoryCode]="5"),  
InternetSales[TaxAmt]+ InternetSales[Freight])
```

Contenido relacionado

[Función MAX](#)

[MAXA, función](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?



Sí



No

MÍN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el valor inferior de una columna, o entre dos expresiones escalares.

Sintaxis

DAX
<code>MIN(<column>)</code>

DAX
<code>MIN(<expression1>, <expression2>)</code>

Parámetros

Expandir tabla

Término	Definición
columna	Columna en la que se quiere encontrar el valor inferior.
expression	Cualquier expresión DAX que devuelva un valor único.

Valor devuelto

Valor menor.

Notas

- La función MIN toma una columna o dos expresiones como argumento y devuelve el valor inferior. Se cuentan los siguientes tipos de valores de las columnas:
 - Números
 - Textos
 - Fechas
 - Espacios en blanco

- Al comparar expresiones, el espacio en blanco se trata como 0. Es decir, Min(1,Blank()) devuelve 0 y Min(-1, Blank()) devuelve -1. Si ambos argumentos están en blanco, MIN devuelve un espacio en blanco. Si alguna de las expresiones devuelve un valor no permitido, MIN devuelve un error.
- No se admiten los valores TRUE o FALSE. Si quiere evaluar una columna de valores TRUE/FALSE, use la función MINA.

Ejemplo 1

En el ejemplo siguiente se devuelve el valor inferior de la columna calculada, ResellerMargin.

DAX

```
= MIN([ResellerMargin])
```

Ejemplo 2

En el ejemplo siguiente se devuelve el valor inferior de una columna que contiene fechas y horas, TransactionDate. Por lo tanto, esta fórmula devuelve la fecha más antigua.

DAX

```
= MIN([TransactionDate])
```

Ejemplo 3

En el ejemplo siguiente se devuelve el valor inferior del resultado de dos expresiones escalares.

DAX

```
= Min([TotalSales], [TotalPurchases])
```

Contenido relacionado

[Función MINA](#)

[Función MINX](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MINA

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el valor menor de una columna.

Sintaxis

DAX
<code>MINA(<column>)</code>

Parámetros

Expandir tabla

Término	Definición
columna	Columna para la que se quiere encontrar el valor mínimo.

Valor devuelto

Valor menor.

Notas

- La función MINA toma como argumento una columna que contiene números y determina el valor menor de la manera siguiente:
 - Si la columna no contiene ningún valor, MINA devuelve 0 (cero).
 - Las filas de la columna que se evalúan como valores lógicos, como TRUE y FALSE, se tratan como 1 si es TRUE y 0 (cero) si es FALSE.
 - Las celdas vacías se omiten.
- Si quiere comparar valores de texto, use la función MIN.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

La siguiente expresión devuelve el gasto de flete mínimo de la tabla InternetSales.

DAX

```
= MINA(InternetSales[Freight])
```

Ejemplo 2

La expresión siguiente devuelve el valor mínimo de la columna PostalCode. Dado que el tipo de datos de la columna es texto, la función no encuentra ningún valor y la fórmula devuelve cero (0).

DAX

```
= MINA([PostalCode])
```

Contenido relacionado

[Función MIN](#)

[Función MINX](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MINX

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el valor mínimo que se obtiene al evaluar una expresión para cada fila de una tabla.

Sintaxis

DAX

```
MINX(<table>, < expression>,[<variant>])
```

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla que contiene las filas para las que se evaluará la expresión.
expresión	Expresión que se debe evaluar para cada fila de la tabla.
variant	(Opcional) Si es TRUE y si hay tipos de valores variantes o mixtos, se devuelve el valor mínimo basado en ORDER BY ASC.

Valor devuelto

El valor más bajo.

Observaciones

- La función MINX toma como primer argumento una tabla, o bien una expresión que devuelve una tabla. El segundo argumento contiene la expresión que se evalúa para cada fila de la tabla.
- Los valores en blanco se omiten. Los valores TRUE/FALSE no se admiten.
- Si la expresión tiene tipos de valores mixtos o variantes, como texto y número, MINX solo tiene en cuenta números de forma predeterminada. Si es `<variant> =`

`TRUE`, se devuelve el valor mínimo.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

En el ejemplo siguiente se filtra la tabla InternetSales y solo se devuelven las filas de un territorio de ventas específico. Después, la fórmula busca el valor mínimo de la columna Freight.

DAX

```
= MINX( FILTER(InternetSales, [SalesTerritoryKey] = 5),[Freight])
```

Ejemplo 2

En el ejemplo siguiente se usa la misma tabla filtrada que en el anterior, pero en lugar de buscar simplemente valores en la columna para cada fila de la tabla filtrada, la función calcula la suma de dos columnas (Freight y TaxAmt) y devuelve el valor más bajo que resulta de dicho cálculo.

DAX

```
= MINX( FILTER(InternetSales, InternetSales[SalesTerritoryKey] = 5),  
InternetSales[Freight] + InternetSales[TaxAmt])
```

En el primer ejemplo, los nombres de las columnas son incompletos. En el segundo ejemplo, los nombres de columna son completos.

Contenido relacionado

[Función MIN](#)

[Función MINA](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PRODUCTO

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el producto de los números de una columna.

Sintaxis

DAX
<code>PRODUCT(<column>)</code>

Parámetros

Expandir tabla

Término	Definición
columna	Columna que contiene los números para los que se va a calcular el producto.

Valor devuelto

Número decimal.

Comentarios

- Para devolver el producto de una expresión evaluada para cada fila de una tabla, use la función [PRODUCTX](#).
- Solo se cuentan los números de la columna. Se omiten los espacios en blanco, los valores lógicos y el texto. Por ejemplo,

`PRODUCT(Table[Column])` equivale a `PRODUCTX(Table, Table[Column])`.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Aquí se calcula el producto de la columna AdjustedRates de la tabla Annuity:

DAX

```
= PRODUCT( Annuity[AdjustedRates] )
```

Contenido relacionado

[PRODUCTX](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PRODUCTX

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el producto de una expresión evaluada para cada fila de una tabla.

Sintaxis

DAX

`PRODUCTX(<table>, <expression>)`

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla que contiene las filas para las que se evaluará la expresión.
expresión	Expresión que se debe evaluar para cada fila de la tabla.

Valor devuelto

Número decimal.

Comentarios

- Para devolver el producto de los números de una columna, use [PRODUCT](#).
- La función PRODUCTX toma como primer argumento una tabla o una expresión que devuelve una tabla. El segundo argumento es una columna que contiene los números cuya mediana se quiere calcular, o una expresión que se evalúa como una columna.
- Solo se cuentan los números de la columna. Se omiten los espacios en blanco, los valores lógicos y el texto.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Aquí se calcula el valor futuro de una inversión:

DAX

```
= [PresentValue] * PRODUCTX( AnnuityPeriods, 1+[FixedInterestRate] )
```

Contenido relacionado

[PRODUCT](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SUM

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Suma todos los números de una columna.

Sintaxis

DAX

```
SUM(<column>)
```

Parámetros

Expandir tabla

Término	Definición
columna	Columna que contiene los números que se van a sumar.

Valor devuelto

Número decimal.

Comentarios

Si quiere filtrar los valores que va a sumar, puede usar la función SUMX y especificar una expresión que sumar.

Ejemplo

En el ejemplo siguiente se agregan todos los números contenidos en la columna Amt de la tabla Sales.

DAX

```
= SUM(Sales[Amt])
```

Contenido relacionado

SUMX

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SUMX

Artículo • 03/04/2024

Devuelve la suma de una expresión evaluada para cada fila de una tabla.

Sintaxis

DAX

```
SUMX(<table>, <expression>)
```

Parámetros

 Expandir tabla

Término	Definición
tabla	Tabla que contiene las filas para las que se evaluará la expresión.
expresión	Expresión que se debe evaluar para cada fila de la tabla.

Valor devuelto

Número decimal.

Notas

- La función SUMX toma como primer argumento una tabla, o bien una expresión que devuelve una tabla. El segundo argumento es una columna que contiene los números que quiere sumar, o una expresión que se evalúa en una columna.
- La SUMX es una [función iteradora](#).
- Solo se cuentan los números de la columna. Se omiten los espacios en blanco, los valores lógicos y el texto.
- Para ver ejemplos más complejos de SUMX en fórmulas, vea [ALL](#) y [CALCULATETABLE](#).

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En este ejemplo se filtra primero la tabla, InternetSales, en la expresión "InternetSales[SalesTerritoryID] = 5" y, después, se devuelve la suma de todos los valores de la columna Freight. En otras palabras, la expresión devuelve la suma de los cargos de flete solo para el área de ventas especificada.

DAX

```
= SUMX(FILTER(InternetSales, InternetSales[SalesTerritoryID]=5),[Freight])
```

Si no necesita filtrar la columna, use la función SUM. La función SUM es similar a la función de Excel del mismo nombre, salvo que toma una columna como referencia.

Contenido relacionado

[SUM](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Funciones de fecha y hora

Artículo • 20/10/2023

Estas funciones le ayudan a crear cálculos basados en fechas y horas. Muchas de las funciones de DAX son similares a las funciones de fecha y hora de Excel. Sin embargo, las funciones de DAX usan un tipo de datos **datetime** y pueden tomar valores de una columna como argumento.

En esta categoría

Función	Descripción
CALENDAR	Devuelve una tabla con una sola columna denominada "Date" que contiene un conjunto contiguo de fechas.
CALENDARAUTO	Devuelve una tabla con una sola columna denominada "Date" que contiene un conjunto contiguo de fechas.
DATE	Devuelve la fecha especificada en formato de fecha y hora.
DATEDIFF	Devuelve el número de límites de intervalos entre dos fechas.
DATEVALUE	Convierte una fecha en formato de texto a una fecha en formato de fecha y hora.
DAY	Devuelve el día del mes, un número entre 1 y 31.
EDATE	Devuelve la fecha que es el número de meses indicado antes o después de la fecha inicial.
EOMONTH	Devuelve la fecha, en formato de fecha y hora, del último día del mes antes o después de un número de meses especificado.
HOUR	Devuelve la hora como un número entre 0 (00:00) y 23 (23:00).
MINUTE	Devuelve el minuto como un número entre 0 y 59, a partir de un valor de fecha y hora.
MONTH	Devuelve el mes como un número entre 1 (enero) y 12 (diciembre).
NETWORKDAYS	Devuelve el número de días laborables completos entre dos fechas.
NOW	Devuelve la fecha y la hora actuales en formato datetime.
QUARTER	Devuelve el trimestre como un número del 1 al 4.
SECOND	Devuelve los segundos de un valor temporal como un número entre 0 y 59.

Función	Descripción
TIME	Convierte las horas, los minutos y los segundos proporcionados como números en una hora en formato de fecha y hora.
TIMEVALUE	Convierte una hora en formato de texto a una hora en formato de fecha y hora.
TODAY	Devuelve la fecha actual.
UTCNOW	Devuelve la fecha y la hora UTC actuales.
UTCTODAY	Devuelve la fecha UTC actual.
WEEKDAY	Devuelve un número del 1 al 7 que identifica el día de la semana de una fecha.
WEEKNUM	Devuelve el número de semana de la fecha y el año especificados según el valor return_type (el tipo de valor devuelto).
YEAR	Devuelve el año de una fecha como un número entero de cuatro dígitos entre 1900 y 9999.
YEARFRAC	Calcula la fracción del año representada por el número de días enteros entre dos fechas.

Comentarios

¿Le ha resultado útil esta página?

Yes

No

CALENDAR

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve una tabla con una sola columna denominada "Date" que contiene un conjunto contiguo de fechas. El intervalo de fechas va desde la fecha de inicio especificada hasta la fecha de finalización especificada, incluidas las dos fechas.

Sintaxis

DAX
<code>CALENDAR(<start_date>, <end_date>)</code>

Parámetros

[+] Expandir tabla

Término	Definición
start_date	Cualquier expresión DAX que devuelve un valor datetime.
end_date	Cualquier expresión DAX que devuelve un valor datetime.

Valor devuelto

Devuelve una tabla con una sola columna denominada "Date" que contiene un conjunto contiguo de fechas. El intervalo de fechas va desde la fecha de inicio especificada hasta la fecha de finalización especificada, incluidas las dos fechas.

Notas

- Se devuelve un error si start_date es mayor que end_date.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

Esta fórmula devuelve una tabla con fechas entre el 1 de enero de 2015 y el 31 de diciembre de 2021.

DAX

```
= CALENDAR (DATE (2015, 1, 1), DATE (2021, 12, 31))
```

Para un modelo de datos que incluye datos de ventas reales y previsiones de ventas futuras, la expresión siguiente devuelve una tabla de fechas que abarca el intervalo de fechas de las tablas Sales y Forecast.

DAX

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

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CALENDARAUTO

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla con una sola columna denominada "Date" que contiene un conjunto contiguo de fechas. El rango de fechas se calcula automáticamente según los datos del modelo.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
fiscal_year_end_month	Cualquier expresión DAX que devuelva un entero entre 1 y 12. Si se omite, el valor predeterminado es el valor especificado en la plantilla de tabla de calendario del usuario actual, si está presente; de lo contrario, el valor predeterminado es 12.

Valor devuelto

Devuelve una tabla con una sola columna denominada "Date" que contiene un conjunto contiguo de fechas. El rango de fechas se calcula automáticamente según los datos del modelo.

Notas

- El rango de fechas se calcula de la manera siguiente:
 - La fecha más antigua del modelo que no se encuentra en una columna o en una tabla calculada se toma como MinDate.
 - La fecha más reciente del modelo que no se encuentra en una columna o en una tabla calculada se toma como MaxDate.
 - El rango de fechas devuelto son las fechas comprendidas entre el principio del año fiscal asociado a MinDate y el final del año fiscal asociado a MaxDate.
- Se devuelve un error si el modelo no contiene valores de fecha y hora que no estén en columnas o tablas calculadas.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En este ejemplo, MinDate y MaxDate en el modelo de datos son el 1 de julio de 2010 y el 30 de junio de 2011.

`CALENDARAUTO()` devuelve todas las fechas entre el 1 de enero de 2010 y el 31 de diciembre de 2011.

`CALENDARAUTO(3)` devuelve todas las fechas entre el 1 de abril de 2010 y el 31 de marzo de 2012.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DATE

Artículo • 20/10/2023

Devuelve la fecha especificada en formato **datetime**.

Sintaxis

DAX

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

Parámetros

Término	Definición
año	<p>Un número que representa el año.</p> <p>El valor del argumento year puede incluir de uno a cuatro dígitos. El argumento year se interpreta según el sistema de fechas que usa su equipo.</p> <p>Se admiten las fechas a partir del 1 de marzo de 1900.</p> <p>Si escribe un número que tiene posiciones decimales, este se redondea.</p> <p>Para los valores mayores que 9999 o menores que cero (valores negativos), la función devuelve un error #VALUE!.</p> <p>Si el valor de year está comprendido entre 0 y 1899, se suma a 1900 para generar el valor final. Consulte los ejemplos más abajo. Nota: Siempre que sea posible, use cuatro dígitos para el argumento year para evitar resultados no deseados. Por ejemplo, si usa 07, se devuelve 1907 como el valor de año.</p>
mes	<p>Un número que representa el mes o un cálculo según las siguientes reglas:</p> <p>No se admiten enteros negativos. Los valores válidos varían entre 1 y 12.</p> <p>Si month es un número comprendido entre 1 y 12, representa un mes del año. El 1 representa a enero, el 2 representa a febrero, y así sucesivamente hasta el 12, que representa a diciembre.</p> <p>Si escribe un número entero mayor que 12, se produce el siguiente cálculo: la fecha se calcula sumando el valor de month al de year. Por ejemplo, si tiene DATE(2008, 18, 1), la función devuelve un valor de fecha y hora equivalente al 1 de junio de 2009, ya que</p>

Término	Definición
	<p>se suman 18 meses al inicio de 2008, lo que da como resultado un valor de junio de 2009. Vea los ejemplos siguientes.</p>
day	<p>Un número que representa el día o un cálculo según las siguientes reglas:</p> <p>No se admiten enteros negativos. Los valores válidos varían entre 1 y 31.</p> <p>Si day es un número comprendido entre 1 y el último día del mes especificado, entonces representa un día del mes.</p> <p>Si escribe un número entero mayor que el último día del mes especificado, se produce el siguiente cálculo: la fecha se calcula sumando el valor de day al de month. Por ejemplo, en la fórmula <code>DATE(2008, 3, 32)</code>, la función DATE devuelve un valor datetime equivalente al 1 de abril de 2008, ya que se suman 32 días al comienzo de marzo, lo que da como resultado un valor de 1 de abril.</p> <p>Si day contiene un valor decimal, se redondea al valor entero más próximo.</p>

Valor devuelto

Devuelve la fecha especificada (**datetime**).

Comentarios

- La función DATE toma los valores enteros que se introducen como argumentos y genera la fecha correspondiente. La función DATE es más útil en situaciones en las que las fórmulas proporcionan el año, el mes y el día. Por ejemplo, los datos subyacentes pueden contener una fecha en un formato que no se reconoce como tal, como AAAAMMDD. Puede usar la función DATE junto con otras funciones para convertir los datos en un número que se pueda reconocer como una fecha.
- A diferencia de Microsoft Excel, que almacena las fechas como un número de serie, las funciones de fecha de DAX siempre devuelven un tipo de datos **datetime**. Aunque, si quiere, puede cambiar el formato para mostrar las fechas como números de serie.
- Date y datetime también se pueden especificar como literal con el formato `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` o `dt"YYYY-MM-DD hh:mm:ss"`. Cuando se especifica como literal, no es necesario usar la función DATE en la expresión. Para más información, consulte [Sintaxis DAX | Fecha y hora](#).

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

Fecha simple

La fórmula siguiente devuelve la fecha 8 de julio de 2009:

DAX

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

Años anteriores a 1899

Si especifica un valor para el argumento **year** entre 0 (cero) y 1899 (incluido), ese valor se suma a 1900 para calcular el año. La fórmula siguiente devuelve el 2 de enero de 1908 ($1900+08$).

DAX

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

Años posteriores a 1899

Si **year** es un valor entre 1900 y 9999 (incluido), ese valor se usa como año. La fórmula siguiente devuelve la fecha 2 de enero de 2008:

DAX

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

Meses

Si **month** tiene un valor mayor que 12, **month** suma ese número de meses al primer mes del año especificado. La fórmula siguiente devuelve la fecha 2 de febrero de 2009:

DAX

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

Días

Si **day** es mayor que el número de días del mes especificado, **day** suma ese número de días al primer día del mes. La fórmula siguiente devuelve la fecha 4 de febrero de 2008:

DAX

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

Vea también

[Funciones de fecha y hora](#)

[DAY, función](#)

[Función TODAY](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

DATEDIFF

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve el número de límites de intervalos entre dos fechas.

Sintaxis

DAX

```
DATEDIFF(<Date1>, <Date2>, <Interval>)
```

Parámetros

Expandir tabla

Término	Definición
Date1	Un valor de fecha y hora escalar.
Date2	Un valor de fecha y hora escalar.
Intervalo	El intervalo que se va a usar al comparar fechas. El valor puede ser uno de los siguientes: <ul style="list-style-type: none">- SECOND- MINUTE- HOUR- DAY- WEEK- MONTH- QUARTER- YEAR

Valor devuelto

Número de límite de intervalos entre dos fechas.

Comentarios

Se devuelve un resultado positivo si Date2 es mayor que Date1. Se devuelve un resultado negativo si Date1 es mayor que Date2.

Ejemplo

Los ejemplos de este artículo se pueden usar con el modelo de ejemplo de Power BI Desktop de Adventure Works DW 2020. Para obtener el modelo, vea [Modelo de ejemplo de DAX](#).

La consulta DAX siguiente:

```
DAX

EVALUATE
VAR StartDate = DATE ( 2019, 07, 01 )
VAR EndDate =     DATE ( 2021, 12, 31 )
RETURN
{
    ( "Year",      DATEDIFF ( StartDate, EndDate, YEAR ) ),
    ( "Quarter",   DATEDIFF ( StartDate, EndDate, QUARTER ) ),
    ( "Month",     DATEDIFF ( StartDate, EndDate, MONTH ) ),
    ( "Week",      DATEDIFF ( StartDate, EndDate, WEEK ) ),
    ( "Day",       DATEDIFF ( StartDate, EndDate, DAY ) )
}
```

Devuelve lo siguiente:

 Expandir tabla

Valor1	Valor2
Año	2
Quarter (Trimestre)	9
Month (Mes)	29
Semana	130
Día	914

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DATEVALUE

Artículo • 20/10/2023

Convierte una fecha en formato de texto a una fecha en formato de fecha y hora.

Sintaxis

DAX

```
DATEVALUE(date_text)
```

Parámetros

Término	Definición
date_text	Texto que representa una fecha.

Valor de propiedad o valor devuelto

Fecha en formato **datetime**.

Notas

- Al convertirse, DATEVALUE usa la configuración regional y los valores de fecha y hora del modelo para determinar un valor de fecha. Si la configuración de fecha y hora del modelo representa fechas con el formato mes/día/año, la cadena "1/8/2009" se convierte a un valor **datetime** equivalente al 8 de enero de 2009. Pero si la configuración de fecha y hora del modelo representa fechas con el formato día/mes/año, la misma cadena se convierte a un valor **datetime** equivalente al 1 de agosto de 2009.
- Si se produce un error en la conversión con la configuración regional y los valores de fecha y hora del modelo, DATEVALUE intentará usar otros formatos de fecha. En este caso, algunas filas se pueden convertir con un formato mientras que otras filas se convierten con un formato diferente. Por ejemplo, "5/4/2018" puede convertirse en el 4 de mayo de 2018 y "20/4/2018" puede convertirse al 20 de abril.
- Si se omite la parte del año del argumento **date_text**, la función DATEVALUE utiliza el año actual del reloj integrado del equipo. En el argumento **date_text** se omite la

información de hora.

- La configuración regional y los valores de fecha y hora del modelo se determinan inicialmente por la aplicación y el equipo cuando se crea el modelo.
- Date y datetime también se pueden especificar como literal con el formato `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` o `dt"YYYY-MM-DD hh:mm:ss"`. Cuando se especifica como literal, no es necesario usar la función DATEVALUE en la expresión. Para más información, consulte [Sintaxis DAX | Fecha y hora](#).

Ejemplo

En el ejemplo siguiente se devuelve un valor **datetime** distinto en función de la configuración regional del modelo y de la configuración de cómo se presentan las fechas y horas.

- En la configuración de fecha y hora en la que el día precede al mes, el ejemplo devuelve un valor **datetime** correspondiente al 8 de enero de 2009.
- En la configuración de fecha y hora en la que el mes precede al día, el ejemplo devuelve un valor **datetime** correspondiente al 1 de agosto de 2009.

DAX

```
= DATEVALUE("8/1/2009")
```

Vea también

[Funciones de fecha y hora](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

DAY

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve el día del mes, un número entre 1 y 31.

Sintaxis

DAX
<code>DAY(<date>)</code>

Parámetros

Expandir tabla

Término	Definición
fecha	Fecha con formato datetime o una representación de texto de una fecha.

Valor devuelto

Número entero que indica el día del mes.

Notas

- La función DAY toma como argumento la fecha del día que se intenta buscar. Se pueden proporcionar fechas a la función mediante otra función de fecha, una expresión que devuelve una fecha o escribiendo una fecha con un formato **datetime**. También se puede escribir una fecha en uno de los formatos de cadena aceptados para las fechas.
- Los valores que devuelven las funciones YEAR, MONTH y DAY son valores gregorianos, independientemente del formato de presentación del valor de fecha proporcionado. Por ejemplo, si el formato de presentación de la fecha proporcionada en Hijri, los valores devueltos de las funciones YEAR, MONTH y DAY serán valores asociados a la fecha gregoriana equivalente.

- Si el argumento de fecha es una representación de texto de la fecha, la función DAY usará la configuración regional y de fecha y hora del equipo cliente para entender el valor de texto con el fin de realizar la conversión. Si la configuración de fecha y hora actual representa fechas con el formato mes/día/año, la cadena "1/8/2009" se interpretará como un valor **datetime** equivalente al 8 de enero de 2009 y la función devolverá 8. Pero si la configuración de fecha y hora actual representa fechas con el formato día/mes/año, la misma cadena se interpretará como un valor **datetime** equivalente al 1 de agosto de 2009 y la función devolverá 1.

Ejemplo: obtener el día a partir de una columna de fecha

La fórmula siguiente devuelve el día a partir de la fecha de la columna [Cumpleaños].

DAX

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

Ejemplo: obtener el día a partir de una fecha de cadena

Las fórmulas siguientes devuelven el día, 4, con las fechas que se han proporcionado como cadenas en un formato de texto aceptado.

DAX

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

Ejemplo: usar un valor de día como una condición

La expresión siguiente devuelve el día en que se ha realizado cada pedido de venta y marca la fila como un elemento de promoción si el pedido se ha realizado el día 10 del mes.

DAX

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Función TODAY](#)

[DATE, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

EDATE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la fecha que es el número de meses indicado antes o después de la fecha inicial. Use EDATE para calcular las fechas de vencimiento que caigan en el mismo día del mes que la fecha de emisión.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
start_date	Fecha en formato datetime o text que representa la fecha de inicio.
months	Entero que representa el número de meses antes o después de start_date .

Valor devuelto

Fecha (**datetime**).

Notas

- A diferencia de Microsoft Excel, que almacena las fechas como números de serie secuenciales, DAX trabaja con fechas en formato **datetime**. Las fechas almacenadas en otros formatos se convierten de forma implícita.
- Si **start_date** no es una fecha válida, EDATE devuelve un error. Asegúrese de que la fecha o la referencia de columna proporcionada como primer argumento sea una fecha.
- Si **months** no es un entero, se trunca.

- Si el argumento de fecha es una representación de texto de la fecha, la función EDATE usa la configuración regional y de fecha y hora del equipo cliente para entender el valor de texto con el fin de realizar la conversión. Si la configuración de fecha y hora actual representa una fecha en el formato mes/día/año, la siguiente cadena "1/8/2009" se interpreta como un valor datetime equivalente al 8 de enero de 2009. Pero si la configuración de fecha y hora actual representa una fecha en el formato día/mes/año, la misma cadena se interpretaría como un valor datetime equivalente al 1 de agosto de 2009.
- Si la fecha solicitada es posterior al último día del mes correspondiente, se devuelve el último día del mes. Por ejemplo, las siguientes funciones: EDATE("2009-01-29", 1), EDATE("2009-01-30", 1), EDATE("2009-01-31", 1) devuelven 28 de febrero de 2009, que corresponde a un mes después de la fecha de inicio.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se devuelve la fecha tres meses después de la fecha de pedido almacenada en la columna [TransactionDate].

DAX

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

Contenido relacionado

[EOMONTH, función](#)

[Funciones de fecha y hora](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

EOMONTH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve la fecha, en formato **datetime**, del último día del mes, antes o después de un número de meses especificado. Use EOMONTH para calcular las fechas de vencimiento que caigan en el último día del mes.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
start_date	Fecha de inicio con formato datetime o con una representación de texto aceptado de una fecha.
months	Número de meses antes o después de start_date . Nota: Si escribe un número que no es un entero, el número se redondea al alza o a la baja al entero más próximo.

Valor devuelto

Fecha (**datetime**).

Notas

- A diferencia de Microsoft Excel, que almacena las fechas como números de serie secuenciales, DAX trabaja con fechas en formato **datetime**. La función EOMONTH puede aceptar fechas en otros formatos, con las restricciones siguientes:
- Si **start_date** no es una fecha válida, EOMONTH devuelve un error.

- Si **start_date** es un valor numérico que no está en formato **datetime**, EOMONTH convertirá el número en una fecha. Para evitar resultados inesperados, convierta el número en formato **datetime** antes de usar la función EOMONTH.
- Si **start_date** más meses da lugar a una fecha no válida, EOMONTH devolverá un error. Las fechas anteriores al 1 de marzo de 1900 y posteriores al 31 de diciembre de 9999 no son válidas.
- Si el argumento de fecha es una representación de texto de la fecha, la función EDATE usa la configuración regional y de fecha y hora del equipo cliente para entender el valor de texto con el fin de realizar la conversión. Si la configuración de fecha y hora actual representa una fecha con el formato mes/día/año, la siguiente cadena "1/8/2009" se interpreta como un valor datetime equivalente al 8 de enero de 2009. Pero si la configuración de fecha y hora actual representa una fecha con el formato día/mes/año, la misma cadena se interpretaría como un valor datetime equivalente al 1 de agosto de 2009.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La expresión siguiente devuelve 31 de mayo de 2008, porque el argumento **months** se ha redondeado a 2.

DAX

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

Contenido relacionado

[EDATE, función](#)

[Funciones de fecha y hora](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

HOUR

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve la hora como un número entre 0 (00:00) y 23 (23:00).

Sintaxis

DAX
<code>HOUR(<datetime>)</code>

Parámetros

Expandir tabla

Término	Definición
datetime	Un valor datetime , como 16:48:00 o 4:48 PM.

Valor devuelto

Un número entero comprendido entre 0 y 23.

Notas

- La función HOUR toma como argumento el período que contiene la hora que se desea encontrar. Puede proporcionar la hora usando una función de fecha y hora, una expresión que devuelva un valor **datetime**, o bien escribiendo el valor directamente en uno de los formatos de hora aceptados. Las horas también se pueden especificar como cualquier representación de texto aceptada de hora.
- Si el argumento **datetime** es una representación de texto de la fecha y la hora, la función usa la configuración regional y de fecha y hora del equipo cliente para entender el valor de texto con el fin de realizar la conversión. La mayoría de las configuraciones regionales usan los dos puntos (:) como separador de hora, por lo que cualquier texto de entrada en el que se usen dos puntos como separadores de

hora se analizará correctamente. Revise la configuración regional para entender los resultados.

Ejemplo 1

En el siguiente ejemplo se devuelve la hora de la columna **TransactionTime** de una tabla denominada **Orders**.

```
DAX
```

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

Ejemplo 2

En el siguiente ejemplo se devuelve 15, lo que significa que la hora corresponde a 3 P.M. en un reloj de 24 horas. El valor de texto se analiza automáticamente y se convierte en un valor de fecha y hora.

```
DAX
```

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Función MINUTE](#)

[YEAR, función](#)

[Función SECOND](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MINUTE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el minuto como un número entre 0 y 59, a partir de un valor de fecha y hora.

Sintaxis

DAX
<code>MINUTE(<datetime>)</code>

Parámetros

Expandir tabla

Término	Definición
datetime	Un valor datetime o de texto en un formato de hora aceptado, como 16:48:00 o 4:48 p. m.

Valor devuelto

Un número entero comprendido entre 0 y 59.

Notas

- A diferencia de Microsoft Excel, que almacena las fechas y horas en un formato de número de serie, DAX usa un tipo de datos **datetime** para las fechas y las horas. Puede proporcionar el valor **datetime** a la función MINUTE si hace referencia a una columna que almacena fechas y horas, mediante una función de fecha y hora, o bien mediante una expresión que devuelve una fecha y una hora.
- Si el argumento **datetime** es una representación de texto de la fecha y la hora, la función usa la configuración regional y de fecha y hora del equipo cliente para entender el valor de texto con el fin de realizar la conversión. La mayoría de las configuraciones regionales usan los dos puntos (:) como separador de hora, por lo que cualquier texto de entrada en el que se usen dos puntos como separadores de

hora se analizará correctamente. Compruebe la configuración regional para entender los resultados.

Ejemplo 1

En el ejemplo siguiente se devuelve el minuto del valor almacenado en la columna **TransactionTime** de la tabla **Orders**.

DAX

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

Ejemplo 2

En el ejemplo siguiente se devuelve 45, que es el número de minutos de la hora 13:45.

DAX

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Función HOUR](#)

[YEAR, función](#)

[Función SECOND](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MONTH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el mes como un número entre 1 (enero) y 12 (diciembre).

Sintaxis

DAX
<code>MONTH(<datetime>)</code>

Parámetros

Expandir tabla

Término	Definición
fecha	Una fecha en formato datetime de o texto.

Valor devuelto

Un número entero comprendido entre 1 y 12.

Notas

- A diferencia de Microsoft Excel, que almacena las fechas como números de serie, DAX usa el formato **datetime** cuando trabaja con fechas. Puede especificar la fecha que se usa como argumento para la función MONTH si escribe un formato **datetime** aceptado y proporciona una referencia a una columna que contiene fechas, o bien mediante una expresión que devuelva una fecha.
- Los valores que devuelven las funciones YEAR, MONTH y DAY son valores gregorianos, con independencia del formato de presentación del valor de fecha proporcionado. Por ejemplo, si el formato de presentación de la fecha proporcionada en Hijri, los valores devueltos para las funciones YEAR, MONTH y DAY serán valores asociados a la fecha gregoriana equivalente.

- Si el argumento de fecha es una representación de texto de la fecha, la función usa la configuración regional y de fecha y hora del equipo cliente para entender el valor de texto con el fin de realizar la conversión. Si la configuración de fecha y hora actual representa una fecha en el formato mes/día/año, la cadena "1/8/2009" siguiente se interpreta como un valor de fecha y hora equivalente al 8 de enero de 2009, y la función devuelve un resultado de 1. Pero si la configuración de fecha y hora actual representa una fecha con el formato día/mes/año, la misma cadena se interpretará como un valor de fecha y hora equivalente al 1 de agosto de 2009 y la función devolverá 8.
- Si la representación de texto de la fecha no se puede convertir correctamente en un valor de fecha y hora, la función devuelve un error.

Ejemplo 1

La expresión siguiente devuelve 3, que es el entero que corresponde a marzo, el mes del argumento **date**.

DAX

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

Ejemplo 2

La expresión siguiente devuelve el mes de la fecha de la columna **TransactionDate** de la tabla **Orders**.

DAX

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Función HOUR](#)

[Función MINUTE](#)

[YEAR, función](#)

[Función SECOND](#)

Comentarios

¿Le ha resultado útil esta página?



Sí



No

NETWORKDAYS

Artículo • 21/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de días laborables completos entre dos fechas (inclusive). Los parámetros especifican cuáles y cuántos días son de fin de semana. Los días de fin de semana y los días especificados como días festivos no se consideran días laborables.

Sintaxis

DAX

```
NETWORKDAYS(<start_date>, <end_date>[, <weekend>, <holidays>])
```

Parámetros

Expandir tabla

Término	Definición
start_date	Fecha que representa la fecha de inicio. Fechas para las que se va a calcular la diferencia. El valor de start_date puede ser anterior, igual o posterior a end_date.
end_date	Fecha que representa la fecha de finalización. Fechas para las que se va a calcular la diferencia. El valor de start_date puede ser anterior, igual o posterior a end_date.
weekend	Indica los días de la semana que son días de fin de semana y no se incluyen en el número de días laborables completos entre start_date y end_date. weekend es un número de fin de semana que especifica cuándo se producen los fines de semana. Los valores numéricos del fin de semana indican los siguientes días del fin de semana: 1 u omitido: sábado, domingo 2: domingo, lunes 3: lunes, martes 4: martes, miércoles 5: miércoles, jueves 6: jueves, viernes 7: viernes, sábado 11: solo domingo 12: solo lunes 13: solo martes 14: solo miércoles 15: solo jueves

Término	Definición
16: solo viernes 17: solo sábado	
vacaciones	Tabla de columnas de una o varias fechas que se excluirán del calendario de días laborables.

Valor devuelto

Número entero de días laborables enteros.

Observaciones

- Esta función DAX es similar a las funciones NETWORKDAYS.INTL y NETWORKDAYS de Excel.
- Si start_date y end_date ambos son BLANK, el valor de salida también es BLANK.
- Si start_date o end_date es BLANK, el valor start_date o end_date BLANK se tratará como Date(1899, 12, 30).
- Las fechas se deben especificar mediante la [función DATE](#) o como resultado de otra expresión. Por ejemplo, `DATE (2022, 5, 30)`, especifica el 30 de mayo de 2022. Las fechas también pueden especificarse como un [literal](#) en formato, `(dt"2022-05-30")`. No especifique fechas como texto.

Ejemplo

La expresión siguiente:

```
DAX
= NETWORKDAYS (
    DATE ( 2022, 5, 28 ),
    DATE ( 2022, 5, 30 ),
    1,
    {
        DATE ( 2022, 5, 30 )
    }
)
```

Devuelve:

 Expandir tabla

[Value]
0

En este ejemplo, se devuelve 0 porque la fecha de inicio es un sábado y la fecha de finalización es un lunes. El parámetro weekend especifica que el fin de semana es sábado y domingo, por lo que no son días laborables. El parámetro holiday marca el 30 (la fecha de finalización) como festivo, por lo que no quedan días laborables.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NOW

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la fecha y la hora actuales en formato **datetime**.

La función NOW resulta útil si necesita mostrar la fecha y la hora actuales en una hoja de cálculo, o bien si tiene que calcular un valor en función de la fecha y hora actuales y hacer que ese valor se actualice cada vez que se abra la hoja de cálculo.

Sintaxis

DAX

```
NOW()
```

Valor devuelto

Fecha (**datetime**).

Notas

- El resultado de la función NOW solo cambia cuando se actualiza la columna que contiene la fórmula. No se actualiza continuamente.
- En el servicio Power BI, el resultado de la función NOW siempre está en la zona horaria UTC.
- La función TODAY devuelve la misma fecha, pero no es precisa con respecto a la hora; la hora devuelta es siempre 12:00:00 a. m. y solo se actualiza la fecha.

Ejemplo

En el ejemplo siguiente se devuelve la fecha y la hora actuales más 3,5 días:

DAX

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

Contenido relacionado

[Función UTCNOW](#)

[Función TODAY](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

QUARTER

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve el trimestre como un número del 1 (enero-marzo) al 4 (octubre-diciembre).

Sintaxis

DAX
QUARTER(<date>)

Parámetros

[] Expandir tabla

Término	Definición
fecha	Fecha.

Valor devuelto

Número entero comprendido entre 1 y 4.

Observaciones

Si el valor de entrada es BLANK, el valor de salida es también BLANK.

Ejemplo 1

La consulta DAX siguiente:

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

Devuelve:

 Expandir tabla

[Valor]
1
4

Ejemplo 2

La consulta DAX siguiente:

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

Devuelve:

 Expandir tabla

FactInternetSales[OrderDate]	[Quarter]
3/31/2008	1
4/1/2008	2

Contenido relacionado

[YEAR](#)

[MONTH](#)

[DAY](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SECOND

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve los segundos de un valor temporal, como un número del 0 al 59.

Sintaxis

DAX
<code>SECOND(<time>)</code>

Parámetros

Expandir tabla

Término	Definición
time	Una hora en formato datetime , como 16:48:23 o 4:48:47 p. m.

Valor devuelto

Un número entero comprendido entre 0 y 59.

Notas

- A diferencia de Microsoft Excel, que almacena las fechas y horas como números de serie, DAX usa el formato **datetime** cuando trabaja con fechas y horas. Si los datos de origen no están en este formato, DAX convierte los datos implícitamente. Si es necesario, puede usar el formato para mostrar las fechas y las horas como un número de serie.
- El valor de fecha y hora que se proporciona como argumento a la función SECOND se puede escribir como una cadena de texto entre comillas (por ejemplo, "6:45 p. m."). También puede proporcionar un valor temporal como resultado de otra expresión o como una referencia a una columna que contenga horas.

- Si proporciona un valor numérico de otro tipo de datos, como 13,60, el valor se interpreta como un número de serie y se representa como un tipo de datos **datetime** antes de extraer el valor de segundos. Para que sea más fácil comprender los resultados, puede interesarle representar estos números como fechas antes de usarlos en la función SECOND. Por ejemplo, si usa SECOND con una columna que contiene un valor numérico como, 25,56, la fórmula devuelve 24. Esto se debe a que, cuando el formato es una fecha, el valor 25,56 equivale a 25 de enero de 1900, 1:26:24 de la tarde.
- Si el argumento **time** es una representación de texto de una fecha, la función usa la configuración regional y de fecha y hora del equipo cliente para entender el valor de texto con el fin de realizar la conversión. La mayoría de las configuraciones regionales usan los dos puntos (:) como separador de hora, por lo que cualquier texto de entrada que use dos puntos como separadores de hora se analizará correctamente. Revise la configuración regional para entender los resultados.

Ejemplo 1

Esta fórmula devuelve el número de segundos en el tiempo contenido en la columna **TransactionTime** de una tabla denominada **Orders**.

DAX

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

Ejemplo 2

Esta fórmula devuelve 3, que es el número de segundos de la hora representada por el valor 3 de marzo de 2008, 12:00:03.

DAX

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

Contenido relacionado

[Funciones de fecha y hora](#)

[HOUR](#)

MINUTE

YEAR

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TIME

Artículo • 14/12/2023

Convierte las horas, minutos y segundos proporcionados como números en una hora en formato **datetime**.

Sintaxis

DAX

`TIME(hour, minute, second)`

Parámetros

 Expandir tabla

Término	Definición
hora	<p>Modo de importación: número del 0 al 32767 que representa la hora. Cualquier valor mayor que 23 se dividirá entre 24 y el resto se tratará como el valor de hora, representado como una fracción de un día. Por ejemplo, TIME(27,0,0) = TIME(3,0,0) = 3:00:00 a. m.</p> <p>Modo DirectQuery: número del 0 al 23 que representa la hora.</p>
minute	<p>Modo de importación: número del 0 al 32767 que representa el minuto. Cualquier valor mayor que 59 minutos se convertirá a horas y minutos. Cualquier valor mayor que 1440 (24 horas) no modifica la parte de la fecha; en su lugar, se dividirá entre 1440 y el resto se tratará como el valor de minutos, representado como una fracción de un día. Por ejemplo, TIME(0,2190,0) = TIME(0,750,0) = TIME(12,30,0) = 12:30:00 p. m.</p> <p>Modo DirectQuery: número del 0 al 59 que representa el minuto.</p>
second	<p>Modo de importación: número del 0 al 32767 que representa el segundo. Cualquier valor mayor que 59 se convertirá a horas, minutos y segundos. Por ejemplo, TIME(0,0,2000) = TIME(0,33,20) = 12:33:20 a. m.</p> <p>Modo DirectQuery: número del 0 al 59 que representa el segundo.</p>

Valor devuelto

Una hora (**datetime**) que va desde las 00:00:00 (12:00:00 a. m.) hasta las 23:59:59 (11:59:59 p. m.).

Notas

- A diferencia de Microsoft Excel, que almacena las fechas y horas como números de serie, DAX trabaja con fechas y horas en formato **datetime**. Los números en otros formatos se convierten implícitamente cuando se usa un valor de fecha y hora en una función DAX. Si se necesitan usar números de serie, se puede usar el formato para cambiar la forma en la que se muestran los números.
- Los valores de hora son una parte de un valor de fecha y en el sistema de números de serie se representan mediante un número decimal. Por lo tanto, el valor de **datetime** 12:00 p. m. es equivalente a 0,5, ya que es la mitad de un día.
- Se pueden proporcionar los argumentos a la función TIME como valores introducidos directamente, como resultado de otra expresión o mediante una referencia a una columna que contiene un valor numérico.
- Date y datetime también se pueden especificar como literal con el formato `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` o `dt"YYYY-MM-DD hh:mm:ss"`. Cuando se especifica como literal, no es necesario usar la función TIME en la expresión. Para más información, consulte [Sintaxis DAX | Fecha y hora](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

Los ejemplos siguientes devuelven la hora, 3:00 a. m.:

DAX

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

DAX

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

Ejemplo 2

Los ejemplos siguientes devuelven la hora, 12:30 p. m.:

DAX

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

DAX

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

Ejemplo 3

En el ejemplo siguiente se crea una hora basada en los valores de las columnas

`intHours`, `intMinutes` y `intSeconds`:

DAX

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

Contenido relacionado

[FECHA](#)

[Funciones de fecha y hora](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TIMEVALUE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Convierte una hora en formato de texto a una hora en formato datetime.

Sintaxis

DAX

```
TIMEVALUE(time_text)
```

Parámetros

Expandir tabla

Término	Definición
time_text	Cadena de texto que representa una hora determinada del día. Se omite cualquier información de fecha incluida en el argumento time_text .

Valor devuelto

Fecha (datetime).

Notas

- Los valores de hora son una parte de un valor de fecha y se representan mediante un número decimal. Por ejemplo, 12:00 p. m. se representa como 0,5 porque es la mitad de un día.
- Si el argumento **time_text** es una representación de texto de la fecha, la función usa la configuración regional y los valores de fecha y hora del modelo para entender el valor de texto con el fin de realizar la conversión. La mayoría de las configuraciones regionales usan los dos puntos (:) como separador de hora, por lo que cualquier texto de entrada que use dos puntos como separadores de hora se analizará correctamente. Revise la configuración regional para entender los resultados.

- Date y datetime también se pueden especificar como literal con el formato `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` o `dt"YYYY-MM-DD hh:mm:ss"`. Cuando se especifica como literal, no es necesario usar la función TIMEVALUE en la expresión. Para más información, consulte [Sintaxis DAX | Fecha y hora](#).

Ejemplo

DAX

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

Contenido relacionado

[Funciones de fecha y hora](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TODAY

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la fecha actual.

Sintaxis

DAX

`TODAY()`

Valor devuelto

Fecha (`datetime`).

Notas

- La función TODAY es útil cuando es necesario que la fecha actual se muestre en una hoja de cálculo, independientemente del momento en que se abra el libro. También es útil para calcular intervalos.
- Si la función TODAY no actualiza la fecha cuando es de esperar, es posible que deba cambiar la configuración que controla cuándo se actualiza la columna o el libro.
- La función NOW es parecida, pero devuelve la hora exacta, mientras que TODAY devuelve el valor de hora 12:00:00 p. m. para todas las fechas.

Ejemplo

Si sabe que alguien nació en 1963, puede usar la fórmula siguiente para buscar la edad de esa persona a partir del cumpleaños de este año:

DAX

`= YEAR(TODAY())-1963`

Esta fórmula usa la función TODAY como argumento de la función YEAR para obtener el año actual y, después, resta 1963, lo que devuelve la edad de la persona.

Contenido relacionado

[Funciones de fecha y hora](#)

[NOW](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

UTCNOW

Artículo • 28/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la fecha y hora UTC actuales.

Sintaxis

DAX

```
UTCNOW()
```

Valor devuelto

(datetime) .

Notas

El resultado de la función UTCNOW solo cambia cuando se actualiza la fórmula. No se actualiza continuamente.

Ejemplo

Lo siguiente:

DAX

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

Devuelve:

Expandir tabla

[Valor]
2/2/2018 4:48:08 AM

Contenido relacionado

[Función NOW](#)

[Función UTCTODAY](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

UTCTODAY

Artículo • 28/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve la fecha UTC actual.

Sintaxis

DAX

```
UTCTODAY()
```

Valor devuelto

Fecha.

Notas

- UTCTODAY devuelve el valor de hora 12:00:00 p. m. para todas las fechas.
- La función UTCNOW es similar, pero devuelve la fecha y la hora exactas.

Ejemplo

Lo siguiente:

DAX

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

Devuelve:

[] Expandir tabla

[Valor]
2/2/2018

Contenido relacionado

[Función NOW](#)

[Función UTCNOW](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

WEEKDAY

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve un número del 1 al 7 que identifica el día de la semana de una fecha. De forma predeterminada, el día está comprendido entre 1 (domingo) y 7 (sábado).

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
fecha	Fecha en formato datetime . Las fechas se deben escribir con la función DATE, mediante expresiones que dan lugar a una fecha o como resultado de otras fórmulas.
return_type	Número que determina el valor devuelto: Tipo de valor devuelto: 1, la semana comienza el domingo (1) y termina el sábado (7), numerado del 1 al 7. Tipo de valor devuelto: 2, la semana comienza el lunes (1) y termina el domingo (7). Tipo de valor devuelto: 3, la semana comienza el lunes (0) y termina el domingo (6), numerado del 1 al 7.

Valor devuelto

Número entero comprendido entre 1 y 7.

Notas

- A diferencia de Microsoft Excel, que almacena las fechas como números de serie, DAX trabaja con las fechas y las horas en formato **datetime**. Si necesita mostrar las fechas como números de serie, puede usar las opciones de formato de Excel.
- También puede escribir las fechas en una representación de texto aceptada de una fecha, pero, para evitar resultados inesperados, es mejor convertir primero la fecha de texto en un formato **datetime**.
- Si el argumento de fecha es una representación de texto de la fecha, la función usa la configuración regional y de fecha y hora del equipo cliente para entender el valor de texto con el fin de llevar a cabo la conversión. Si la configuración de fecha y hora actual representa las fechas con el formato mes/día/año, la cadena "1/8/2009" se interpretará como un valor **datetime** equivalente al 8 de enero de 2009. En cambio, si la configuración de fecha y hora actual representa las fechas con el formato día/mes/año, la misma cadena se interpretaría como un valor **datetime** equivalente al 1 de agosto de 2009.

Ejemplo

En el ejemplo siguiente se obtiene la fecha de la columna [HireDate], se agrega 1 y se muestra el día de la semana correspondiente a esa fecha. Dado que se ha omitido el argumento **return_type**, se usa el formato predeterminado, en el que el 1 es domingo y el 7 es sábado. Si el resultado es 4, el día sería miércoles.

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Función WEEKNUM](#)

[YEARFRAC, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

WEEKNUM

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve el número de semana de la fecha especificada según el valor de `return_type` (el tipo de valor devuelto). El número de semana indica la posición numérica de la semana dentro de un año.

Hay dos *sistemas* usados para esta función:

- **Sistema 1:** la semana que contiene el 1 de enero es la primera semana del año y se numera la semana 1.
- **Sistema 2:** la semana que contiene el primer jueves del año es la primera semana del año y se numera como semana 1. Este sistema es la metodología especificada en ISO 8601, que se conoce normalmente como sistema de numeración de semana europea.

Sintaxis

DAX

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

Parámetros

[] Expandir tabla

Término	Definición
fecha	Fecha en formato <code>datetime</code> .
return_type	(Opcional) Número que determina el día en que comienza la semana. El valor predeterminado es 1. Vea la sección Comentarios.

Valor devuelto

Número entero.

Notas

- De forma predeterminada, la función WEEKNUM usa una convención de calendario por la que la semana que contiene el 1 de enero se considera la primera semana del año. Pero el estándar de calendario ISO 8601, ampliamente usado en Europa, define la primera semana como la que tiene la mayoría de días (cuatro o más) en el nuevo año. Esto significa que si **return_type** es cualquier valor válido distinto de 21, para los años en los que haya tres días o menos en la primera semana de enero, la función WEEKNUM devuelve números de semana que son diferentes de la definición ISO 8601.
- Para **return_type**, excepto 21, es posible que algunos orígenes de datos de DirectQuery no admitan los siguientes valores válidos:

 Expandir tabla

return_type	Cuándo comienza la semana	Sistema
1 u omitido	Domingo	1
2	Lunes	1
11	Lunes	1
12	Martes	1
13	Miércoles	1
14	Jueves	1
15	Viernes	1
16	Sábado	1
17	Domingo	1
21	Lunes	2

Ejemplo 1

En el ejemplo siguiente se devuelve el número de semana para el 14 de febrero de 2010. En este cálculo se da por supuesto que las semanas comienzan el lunes.

DAX

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

Ejemplo 2

En el ejemplo siguiente se devuelve el número de semana de la fecha almacenada en la columna **HireDate** de la tabla **Employees**. En este cálculo se da por supuesto que las semanas comienzan el domingo.

DAX

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

Contenido relacionado

[YEARFRAC, función](#)

[Función WEEKDAY](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

YEAR

Artículo • 14/12/2023

Devuelve el año de una fecha como un entero de cuatro dígitos del rango 1900-9999.

Sintaxis

DAX

`YEAR(<date>)`

Parámetros

 Expandir tabla

Término	Definición
fecha	Fecha en formato datetime o de texto que contiene el año que se quiere buscar.

Valor devuelto

Entero del rango 1900-9999.

Notas

- A diferencia de Microsoft Excel, que almacena las fechas como números de serie, DAX usa un tipo de datos **datetime** para trabajar con fechas y horas.
- Las fechas se deben especificar mediante la función DATE, o como resultados de otras fórmulas o funciones. También puede especificar fechas en representaciones de texto aceptadas de una fecha, como 3 de marzo, 2007, o 3-marzo-2003.
- Los valores devueltos por las funciones YEAR, MONTH y DAY son valores gregorianos independientemente del formato de presentación del valor de fecha proporcionado. Por ejemplo, si el formato de presentación de la fecha proporcionada usa el calendario Hijri, los valores devueltos de las funciones YEAR, MONTH y DAY son valores asociados a la fecha gregoriana equivalente.

- Si el argumento de fecha es una representación de texto de la fecha, la función usa la configuración regional y de fecha y hora del equipo cliente para entender el valor de texto con el fin de realizar la conversión. Pueden producirse errores si el formato de las cadenas no es compatible con la configuración regional actual. Por ejemplo, si la configuración regional define las fechas a las que se va a dar formato como mes/día/año, y la fecha se proporciona como día/mes/año, 25/1/2009 no se interpreta como 25 de enero de 2009, sino como una fecha no válida.

Ejemplo

En el ejemplo siguiente se devuelve 2007.

DAX

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

Ejemplo: fecha como resultado de expresión

Descripción

En el ejemplo siguiente se devuelve el año de la fecha actual.

DAX

```
= YEAR(TODAY())
```

Contenido relacionado

[Funciones de fecha y hora](#)

[Función HOUR](#)

[Función MINUTE](#)

[YEAR, función](#)

[Función SECOND](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

YEARFRAC

Artículo • 28/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Calcula la fracción del año representada por el número de días enteros entre dos fechas. Use la función de hoja de cálculo YEARFRAC para identificar la proporción de los beneficios o las obligaciones de un año completo para asignarla a un trimestre específico.

Sintaxis

DAX

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

Parámetros

[] Expandir tabla

Término	Definición
start_date	Fecha de inicio en formato datetime .
end_date	Fecha final en formato de datetime .
basis	(Opcional) Tipo de base en que deben contarse los días. Todos los argumentos se truncan en enteros. Base: Descripción 0 - US (NASD) 30/360 (valor predeterminado) 1 - Actual/actual 2 - Actual/360 3 - Actual/365 4 - European 30/360

Valor devuelto

Número decimal. El tipo de datos interno es un número de punto flotante de precisión doble IEEE de 64 bits (8 bytes).

Notas

- A diferencia de Microsoft Excel, que almacena las fechas como números de serie, DAX usa un formato **datetime** para trabajar con fechas y horas. Si necesita ver fechas como números de serie, puede usar las opciones de formato de Excel.
- Si **start_date** o **end_date** no son fechas válidas, YEARFRAC devuelve un error.
- Si **basis**< 0 o si **basis**> 4, YEARFRAC devuelve un error.

Ejemplo 1

En este ejemplo se devuelve la fracción de un año representada por la diferencia entre las fechas de las dos columnas, `TransactionDate` y `ShippingDate`:

DAX

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

Ejemplo 2

En este ejemplo se devuelve la fracción de un año representada por la diferencia entre las fechas, 1 de enero y 1 de marzo:

DAX

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

Use el formato de cuatro dígitos para los años siempre que sea posible, para evitar obtener resultados inesperados. Cuando se trunca el año, se supone que se trata del año actual. Cuando se omite la fecha, se supone que es la primera fecha del mes.

El segundo argumento, **basis**, también se ha omitido. Por tanto, la fracción del año se calcula según el estándar US (NASD) 30/360.

Contenido relacionado

[Funciones de fecha y hora](#)

[Función WEEKNUM](#)

[YEARFRAC, función](#)

[Función WEEKDAY](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Funciones de filtro

Artículo • 02/04/2024

Las funciones de filtro y valor de DAX son algunas de las más complejas y eficaces, y se diferencian mucho de las funciones de Excel. Las funciones de búsqueda trabajan mediante tablas y relaciones, como una base de datos. Las funciones de filtro permiten manipular el contexto de datos para crear cálculos dinámicos.

En esta categoría

 Expandir tabla

Función	Descripción
ALL	Devuelve todas las filas de una tabla, o todos los valores de una columna, omitiendo los filtros que se puedan haber aplicado.
ALLCROSSFILTERED	Borra todos los filtros que se han aplicado a una tabla.
ALLEXCEPT	Quita todos los filtros de contexto de la tabla, excepto los que se han aplicado a las columnas especificadas.
ALLNOBLANKROW	A partir de la tabla principal de una relación, devuelve todas las filas, menos la fila en blanco, o todos los valores distintos de una columna, menos la fila en blanco, y no tiene en cuenta los filtros de contexto que puedan existir.
ALLSELECTED	Quita los filtros de contexto de las columnas y las filas de la consulta actual, a la vez que conserva todos los demás filtros de contexto o filtros explícitos.
CALCULATE	Evalúa una expresión en un contexto de filtro modificado.
CALCULATETABLE	Evalúa una expresión de tabla en un contexto de filtro modificado.
EARLIER	Devuelve el valor actual de la columna especificada en un paso de evaluación externo de la columna mencionada.
EARLIEST	Devuelve el valor actual de la columna especificada en un paso de evaluación externo de la columna especificada.
FILTER	Devuelve una tabla que representa un subconjunto de otra tabla o expresión.
FIRST	Solo se usa en cálculos visuales. Recupera un valor de la matriz visual a partir de la primera fila de un eje.
INDEX	Devuelve una fila en una posición absoluta, especificada por el parámetro position, dentro de la partición especificada con el orden o en el eje

Función	Descripción
	especificados.
KEEPFILTERS	Modifica el modo en que se aplican los filtros mientras se evalúa una función CALCULATE o CALCULATETABLE.
LAST	Solo se usa en cálculos visuales. Recupera un valor de la matriz visual de la última fila de un eje.
LOOKUPVALUE	Devuelve el valor para la fila que cumple todos los criterios que especifican las condiciones de búsqueda. La función puede aplicar una o más condiciones de búsqueda.
MATCHBY	En las funciones window, define las columnas que se usan para determinar cómo hacer coincidir los datos e identificar la <i>fila actual</i> .
MOVINGAVERAGE	Devuelve un promedio móvil calculado a lo largo del eje especificado de la matriz visual.
SIGUIENTE	Solo se usa en cálculos visuales. Recupera un valor de la siguiente fila de un eje de la matriz visual.
OFFSET	Devuelve una sola fila que se coloca antes o después de la <i>fila actual</i> en la misma tabla, con un desplazamiento determinado.
ORDERBY	Define las columnas que determinan el criterio de ordenación dentro de cada una de las particiones de una función WINDOW.
PARTITIONBY	Define las columnas que se usan para particionar el parámetro <relation> de una función window.
PREVIOUS	Solo se usa en cálculos visuales. Recupera un valor de la fila anterior de un eje de la matriz visual.
RANGE	Devuelve un intervalo de filas dentro del eje especificado, en relación con la fila actual. Un acceso directo para WINDOW.
RANK	Devuelve la clasificación de una fila dentro del intervalo especificado.
REMOVEFILTERS	Borra los filtros de las tablas o columnas especificadas.
ROWNUMBER	Devuelve la clasificación única de una fila dentro del intervalo especificado.
RUNNINGSUM	Devuelve una suma en ejecución calculada a lo largo del eje especificado de la matriz visual.
SELECTEDVALUE	Devuelve el valor si el contexto de columnName se ha filtrado a un solo valor distinto. De lo contrario, devuelve alternateResult.
WINDOW	Devuelve varias filas que se colocan dentro del intervalo especificado.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ALL

Artículo • 20/10/2023

Devuelve todas las filas de una tabla, o todos los valores de una columna, omitiendo los filtros que se puedan haber aplicado. Esta función es útil para borrar filtros y crear cálculos en todas las filas de una tabla.

Sintaxis

DAX

```
ALL( <table> | <column>[, <column>[, <column>[,...]]] )
```

Parámetros

Término	Definición
tabla	Tabla en la que se van a borrar los filtros.
columna	Columna en la que se van a borrar los filtros.

El argumento de la función ALL debe ser una referencia a una tabla base o a una columna base. No se pueden usar expresiones de tabla ni de columna con la función ALL.

Valor devuelto

Tabla o columna con filtros quitados.

Notas

- Esta función no se usa por sí sola, sino que sirve como función intermedia que se puede usar para cambiar el conjunto de resultados en el que se realiza otro cálculo.
- El comportamiento normal de las expresiones DAX que contienen la función ALL() es omitir cualquier filtro que se haya aplicado. Sin embargo, hay algunas situaciones en las que este no es el caso debido a *auto-exist*, una tecnología de DAX que optimiza el filtrado para reducir la cantidad de procesamiento necesaria

para ciertas consultas DAX. Un ejemplo en el que auto-exist y ALL() proporcionan resultados inesperados es al aplicar filtros en una o varias columnas de la misma tabla (por ejemplo, al usar sectores), y hay una medida en dicha tabla que usa ALL(). En este caso, auto-exist *combina* los diferentes filtros en uno y solo aplica el filtrado en las combinaciones de valores existentes. Debido a esta combinación, la medición se calcula sobre las combinaciones de valores existentes, y el resultado se basa en los valores filtrados, en lugar de en todos ellos, como se esperaría. Para obtener más información sobre auto-exist y sus implicaciones en relación con los cálculos, consulte el artículo del Microsoft MVP Alberto Ferrari, en el que se abordan [auto-exist en DAX](#), en [sql.bi.com](#).

- En la tabla siguiente se describen los posibles usos de las funciones ALL y ALLEXCEPT en diferentes situaciones.

Función y uso	Descripción
ALL()	Quita todos los filtros de todas partes. ALL() solo se puede usar para borrar filtros, pero no para devolver una tabla.
ALL(Table)	Quita todos los filtros de la tabla especificada. En efecto, ALL(Table) devuelve todos los valores de la tabla, quitando los filtros de contexto que pudieran haberse aplicado. Esta función resulta útil cuando se trabaja con muchos niveles de agrupación y se quiere crear un cálculo que cree una proporción de un valor agregado al valor total. El primer ejemplo ilustra este escenario.
ALL (Column[, Column[, ...]])	Quita todos los filtros de las columnas especificadas de la tabla. Todos los demás filtros de las demás columnas de la tabla se siguen aplicando. Todos los argumentos de columna deben proceder de la misma tabla. La variante ALL(Column) resulta útil cuando se quieren quitar filtros de contexto de una o varias columnas específicas y conservar todos los demás filtros de contexto. El segundo y el tercer ejemplo ilustran este escenario.
ALLEXCEPT(Table, Column1 [,Column2]...)	Quita todos los filtros de contexto de la tabla, excepto los que se aplican a las columnas especificadas. Se trata de un cómodo acceso directo para las situaciones en las que se quieren quitar filtros en muchas columnas de una tabla, pero no en todas.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

Calcular la proporción de ventas por categoría con respecto a las ventas totales

Supongamos que quiere hallar la cantidad de ventas de la celda actual, en la tabla dinámica, dividida entre las ventas totales de todos los distribuidores. Para asegurarse de que el denominador es el mismo, independientemente de cómo el usuario de la tabla dinámica pueda filtrar o agrupar los datos, define una fórmula que usa ALL para crear el total general correcto.

En la siguiente tabla se muestran los resultados cuando se crea una medida, **Proporción de todas las ventas de distribuidores**, con la fórmula mostrada en la sección de código. Para ver cómo funciona esto, agregue el campo CalendarYear al área **Etiquetas de fila** de la tabla dinámica y el campo ProductCategoryName al área **Etiquetas de columna**. Luego, arrastre la medida, **Proporción de todas las ventas de distribuidores**, al área **Valores** de la tabla dinámica. Para ver los resultados como porcentajes, use las características de formato de Excel para aplicar un formato de número de porcentaje a las celdas que contienen la medida.

Etiquetas de fila	Accesorios	Bicicletas	Ropa	Componentes	Total general
2005	0,02 %	9,10 %	0,04 %	0,75 %	9,91 %
2006	0,11 %	24,71 %	0,60 %	4,48 %	29,90 %
2007	0,36 %	31,71 %	1,07 %	6,79 %	39,93 %
2008	0,20 %	16,95 %	0,48 %	2,63 %	20,26 %
Total general	0,70 %	82,47 %	2,18 %	14,65 %	100,00 %

Fórmula

DAX

```
= SUMX(ResellerSales_USD,
    ResellerSales_USD[SalesAmount_USD])/SUMX(ALL(ResellerSales_USD),
    ResellerSales_USD[SalesAmount_USD])
```

La fórmula se construye de la siguiente manera:

1. El numerador, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, es la suma de los valores de `ResellerSales_USD[SalesAmount_USD]` de la celda actual de la tabla dinámica, con filtros de contexto aplicados en `CalendarYear` y `ProductCategoryName`.
2. Respecto al denominador, empieza especificando una tabla, `ResellerSales_USD`, y usa la función `ALL` para quitar todos los filtros de contexto de la tabla.

3. Luego, usa la función SUMX para sumar los valores de la columna ResellerSales_USD[SalesAmount_USD]. En otras palabras, obtiene la suma de ResellerSales_USD[SalesAmount_USD] de todas las ventas de distribuidores.

Ejemplo 2

Calcular la proporción de ventas de productos con respecto a las ventas totales a lo largo del año actual

Supongamos que quiere crear una tabla que muestre el porcentaje de ventas en comparación con los años de cada categoría de producto (ProductCategoryName). Para obtener el porcentaje de cada año relativo a cada valor de ProductCategoryName, debe dividir la suma de ventas de ese año en particular y de la categoría de producto entre la suma de las ventas de la misma categoría de producto a lo largo de todos los años. En otras palabras, quiere conservar el filtro en ProductCategoryName, pero quitar el filtro de año al calcular el denominador del porcentaje.

En la siguiente tabla se muestran los resultados cuando se crea una medida, **Año de ventas de distribuidores**, con la fórmula mostrada en la sección de código. Para ver cómo funciona, agregue el campo CalendarYear al área **Etiquetas de fila** de la tabla dinámica y el campo ProductCategoryName al área **Etiquetas de columna**. Para ver los resultados como porcentajes, use las características de formato de Excel para aplicar un formato de número de porcentaje a las celdas que contienen la medida, **Año de ventas de distribuidores**.

Etiquetas de fila	Accesorios	Bicicletas	Ropa	Componentes	Total general
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 %
Total general	100,00 %	100,00 %	100,00 %	100,00 %	100,00 %

Fórmula

DAX

```
= SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])/CALCULATE(  
SUM( ResellerSales_USD[SalesAmount_USD]), ALL(DateTime[CalendarYear]))
```

La fórmula se crea de la siguiente manera:

1. El numerador, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, es la suma de los valores de `ResellerSales_USD[SalesAmount_USD]` de la celda actual de la tabla dinámica, con filtros de contexto aplicados en las columnas `CalendarYear` y `ProductCategoryName`.
2. En el denominador, quita el filtro existente en `CalendarYear` mediante la función `ALL(Column)`. Esto calcula la suma de las filas restantes en la tabla `ResellerSales_USD`, después de aplicar los filtros de contexto existentes de las etiquetas de columna. El efecto neto es que, para el denominador, la suma se calcula sobre el valor de `ProductCategoryName` seleccionado (el filtro de contexto implícito) y para todos los valores de año.

Ejemplo 3

Calcular la contribución de las categorías de producto a las ventas totales por año

Supongamos que quiere crear una tabla que muestra el porcentaje de ventas de cada categoría de producto de cada año. Para obtener el porcentaje de cada categoría de producto en un año determinado, debe calcular la suma de ventas de esa categoría de producto en particular (`ProductCategoryName`) en un año n y, después, dividir el valor resultante entre la suma de ventas del año n de todas las categorías de producto. En otras palabras, quiere conservar el filtro de año, pero quitar el filtro en `ProductCategoryName` al calcular el denominador del porcentaje.

En la siguiente tabla se muestran los resultados cuando se crea una medida, **Nombre de categoría de ventas de distribuidores**, con la fórmula mostrada en la sección de código. Para ver cómo funciona esto, agregue el campo `CalendarYear` al área **Etiquetas de fila** de la tabla dinámica y el campo `ProductCategoryName` al área **Etiquetas de columna**. Luego, agregue la nueva medida al área **Valores** de la tabla dinámica. Para ver los resultados como porcentajes, use las características de formato de Excel para aplicar un formato de número de porcentaje a las celdas que contienen la nueva medida, **Nombre de categoría de ventas de distribuidores**.

Etiquetas de fila	Accesorios	Bicicletas	Ropa	Componentes	Total general
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 %

Etiquetas de fila	Accesorios	Bicicletas	Ropa	Componentes	Total general
2008	0,99 %	83,69 %	2,37 %	12,96 %	100,00 %
Total general	0,70 %	82,47 %	2,18 %	14,65 %	100,00 %

Fórmula

DAX

```
= SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])/CALCULATE(
SUM( ResellerSales_USD[SalesAmount_USD]),
ALL(ProductCategory[ProductCategoryName]))
```

La fórmula se crea de la siguiente manera:

1. El numerador, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, es la suma de los valores de `ResellerSales_USD[SalesAmount_USD]` de la celda actual de la tabla dinámica, con filtros de contexto aplicados en los campos `CalendarYear` y `ProductCategoryName`.
2. Respecto al denominador, usa la función `ALL(Column)` para quitar el filtro en `ProductCategoryName` y calcular la suma de las filas restantes en la tabla `ResellerSales_USD`, después de aplicar los filtros de contexto existentes de las etiquetas de fila. El efecto neto es que, para el denominador, la suma se calcula sobre el valor de año seleccionado (el filtro de contexto implícito) y para todos los valores de `ProductCategoryName`.

Vea también

[Funciones de filtro](#)

[Función ALL](#)

[ALLEXCEPT, función](#)

[FILTER \(función\)](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

ALLCROSSFILTERED

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Borra todos los filtros que se han aplicado a una tabla.

Sintaxis

DAX

```
ALLCROSSFILTERED(<table>)
```

Parámetros

[] Expandir tabla

Término	Definición
tabla	Tabla en la que se van a borrar los filtros.

Valor devuelto

N/D Vea Notas.

Observaciones

- ALLCROSSFILTERED solo se puede usar para borrar filtros, pero no para devolver una tabla.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

DAX

```
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]

```

Vea que devuelve lo siguiente:

 Expandir tabla

DimSalesReason[SalesReasonName]	[TotalQuantity1]	[TotalQuantity2]
Evento de demostración	60 398	
Anuncio en revista	60 398	
Fabricante	60 398	1818
En promoción	60 398	7390
Otros	60 398	3 653
Precio	60 398	47 733
Calidad	60 398	1 551
Revisar	60 398	1640
Patrocinio	60 398	
Anuncio en televisión	60 398	730

Nota

Existe una relación de varios a varios directa o indirecta entre las tablas FactInternetSales y DimSalesReason.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ALLEXCEPT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Quita todos los filtros de contexto de la tabla, excepto los que se han aplicado a las columnas especificadas.

Sintaxis

DAX

```
ALLEXCEPT(<table>,<column>[,<column>[,...]])
```

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla en la que se quitan todos los filtros de contexto, excepto los filtros de las columnas especificadas en los argumentos siguientes.
columna	Columna para la que se deben conservar los filtros de contexto.

El primer argumento de la función ALLEXCEPT debe ser una referencia a una tabla base. Todos los argumentos subsiguientes deben ser referencias a las columnas base. No se pueden usar expresiones de tabla ni de columna con la función ALLEXCEPT.

Valor devuelto

Una tabla con todos los filtros quitados, excepto los filtros de las columnas especificadas.

Notas

- Esta función no se usa por sí sola, sino que sirve como función intermedia que se puede usar para cambiar el conjunto de resultados en el que se realiza otro cálculo.

- ALL y ALLEXCEPT se pueden usar en escenarios diferentes:

 Expandir tabla

Función y uso	Descripción
ALL(Table)	Quita todos los filtros de la tabla especificada. En efecto, ALL(Table) devuelve todos los valores de la tabla, quitando los filtros de contexto que pudieran haberse aplicado. Esta función resulta útil cuando se trabaja con muchos niveles de agrupación y se quiere crear un cálculo que cree una proporción de un valor agregado al valor total.
ALL (Column[, Column[, ...]])	Quita todos los filtros de las columnas especificadas de la tabla. Todos los demás filtros de las demás columnas de la tabla se siguen aplicando. Todos los argumentos de columna deben proceder de la misma tabla. La variante ALL(Column) resulta útil cuando se quieren quitar filtros de contexto de una o varias columnas específicas y conservar todos los demás filtros de contexto.
ALLEXCEPT(Table, Column1 [,Column2]...)	Quita todos los filtros de contexto de la tabla, excepto los que se aplican a las columnas especificadas. Se trata de un cómodo acceso directo para las situaciones en las que se quieren quitar filtros en muchas columnas de una tabla, pero no en todas.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula de medida siguiente suma SalesAmount_USD y usa la función ALLEXCEPT para quitar todos los filtros de contexto en la tabla DateTime, excepto si el filtro se ha aplicado a la columna CalendarYear.

DAX

```
= CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]), ALLEXCEPT(DateTime,
DateTime[CalendarYear]))
```

Como la fórmula usa ALLEXCEPT, cada vez que se use una columna, salvo CalendarYear, de la tabla DateTime para segmentar la visualización, la fórmula quitará los filtros de segmentación y proporcionará un valor igual a la suma de SalesAmount_USD. Pero si se usa la columna CalendarYear para segmentar la visualización, los resultados son diferentes. Como se ha especificado CalendarYear como el argumento para ALLEXCEPT,

cuando los datos se segmenten en el año, se aplicará un filtro en los años en el nivel de fila.

Contenido relacionado

[Funciones de filtro](#)

[Función ALL](#)

[FILTER \(función\)](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ALLNOBLANKROW

Artículo • 20/10/2023

A partir de la tabla primaria de una relación, devuelve todas las filas, menos la fila en blanco, o todos los valores distintos de una columna, menos la fila en blanco, y no tiene en cuenta los filtros de contexto que puedan existir.

Sintaxis

DAX

```
ALLNOBLANKROW( {<table> | <column>[, <column>[, <column>[,...]]]} )
```

Parámetros

Término	Definición
tabla	Tabla en la que se quitan todos los filtros de contexto.
columna	Columna en la que se quitan todos los filtros de contexto.

Solo se debe pasar un parámetro, que es una tabla o una columna.

Valor devuelto

Una tabla, cuando el parámetro pasado ha sido una tabla, o una columna de valores, cuando el parámetro pasado ha sido una columna.

Notas

- La función ALLNOBLANKROW solo filtra la fila en blanco que mostrará una tabla primaria, en una relación, cuando haya una o más filas en la tabla secundaria que tengan valores no coincidentes en la columna primaria. Vea el ejemplo siguiente para obtener una explicación detallada.
- En la tabla siguiente se resumen las variaciones de ALL, que se proporcionan en DAX, y sus diferencias:

Función y uso	Descripción
ALL(Column)	Quita todos los filtros de la columna especificada de la tabla. El resto de filtros de la tabla que hay en otras columnas se siguen aplicando.
ALL(Table)	Quita todos los filtros de la tabla especificada.
ALLEXCEPT(Table,Col1,Col2...)	Invalida todos los filtros de contexto de la tabla, excepto los de las columnas especificadas.
ALLNOBLANK(table column)	A partir de la tabla primaria de una relación, devuelve todas las filas, menos la fila en blanco, o todos los valores distintos de una columna, menos la fila en blanco, y no tiene en cuenta los filtros de contexto que puedan existir.

Para obtener una descripción general de cómo opera la función ALL, junto con ejemplos paso a paso que usan ALL(Table) y ALL(Column), vea [Función ALL](#).

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En los datos de ejemplo, la tabla ResellerSales_USD contiene una fila que no tiene valores y, por tanto, no se puede relacionar con ninguna de las tablas primarias en las relaciones dentro del libro. Usará esta tabla en una tabla dinámica para que pueda ver el comportamiento de las filas en blanco y cómo controlar los recuentos de los datos no relacionados.

Paso 1: comprobación de los datos no relacionados

Abra la **ventana de Power Pivot** y luego seleccione la tabla ResellerSales_USD. En la columna ProductKey, filtre por los valores en blanco. Se conservará una fila. En esa fila, todos los valores de columna deben estar en blanco, excepto SalesOrderLineNumber.

Paso 2: creación de una tabla dinámica

Cree una nueva tabla dinámica y, después, arrastre la columna datetime.[Calendar Year], al panel Etiquetas de fila. En la tabla siguiente se muestran los resultados esperados:

Etiquetas de fila
2005
2006

Etiquetas de fila
2007
2008
Total general

Tenga en cuenta la etiqueta en blanco entre **2008** y **Total general**. Esta etiqueta en blanco representa el miembro desconocido, que es un grupo especial que se crea para tener en cuenta los valores de la tabla secundaria que no tengan ningún valor coincidente en la tabla primaria; en este ejemplo, la columna datetime.[Calendar Year].

Cuando vea esta etiqueta en blanco en la tabla dinámica, sabrá que en algunas de las tablas relacionadas con la columna datetime.[Calendar Year] hay valores en blanco o valores no coincidentes. La tabla primaria es la que muestra la etiqueta en blanco, pero las filas que no coinciden se encuentran en una o varias de las tablas secundarias.

Las filas que se agregan a este grupo de etiquetas en blanco son valores que no coinciden con ningún valor de la tabla primaria (por ejemplo, una fecha que no existe en la tabla datetime) o valores NULL, lo que significa que no hay ningún valor para fecha. En este ejemplo, se ha colocado un valor en blanco en todas las columnas de la tabla de ventas secundaria. Tener más valores en la tabla primaria que en las tablas secundarias no provoca ningún problema.

Paso 3: recuento de filas con ALL y ALLNOBLANK

Agregue las dos medidas siguientes a la tabla datetime para contar las filas de la tabla: **CountRows ALLNOBLANK of datetime**, **CountRows ALL of datetime**. Las fórmulas que se pueden usar para definir estas medidas son las siguientes:

DAX

```
// 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'))
```

En una tabla dinámica en blanco, agregue la columna datetime.[Calendar Year] a las etiquetas de fila y, después, agregue las medidas recientemente creadas. Los resultados deben tener un aspecto similar a la tabla siguiente:

Etiquetas de fila	CountRows ALLNOBLANK of datetime	CountRows ALL of datetime
2005	1280	1281
2006	1280	1281
2007	1280	1281
2008	1280	1281
	1280	1281
Total general	1280	1281

Los resultados muestran una diferencia de una fila en el recuento de filas de la tabla. Sin embargo, si abre la **ventana de Power Pivot** y selecciona la tabla datetime, no se puede encontrar ninguna fila en blanco porque la fila en blanco especial mencionada aquí es el miembro desconocido.

Paso 4: comprobación de que el recuento es preciso

Para demostrar que ALLNOBLANKROW no cuenta ninguna fila realmente en blanco y que solo controla la fila en blanco especial de la tabla primaria, agregue las dos medidas siguientes a la tabla ResellerSales_USD: **CountRows ALLNOBLANKROW of ResellerSales_USD**, **CountRows ALL of ResellerSales_USD**.

Cree una nueva tabla dinámica y arrastre la columna datetime.[Calendar Year] al panel Etiquetas de fila. Ahora, agregue las medidas que se acaban de crear. Los resultados deberían parecerse a lo siguiente:

Etiquetas de fila	CountRows ALLNOBLANKROW of ResellerSales_USD	CountRows ALL of ResellerSales_USD
2005	60 856	60 856
2006	60 856	60 856
2007	60 856	60 856
2008	60 856	60 856
	60 856	60 856
Total general	60 856	60 856

Ahora las dos medidas tienen los mismos resultados. Esto se debe a que la función ALLNOBLANKROW no cuenta filas realmente en blanco en una tabla, sino que solo controla la fila en blanco que es un caso especial generado en una tabla primaria, cuando una o varias de las tablas secundarias de la relación contienen valores no coincidentes o valores en blanco.

Vea también

[Funciones de filtro](#)

[Función ALL](#)

[FILTER \(función\)](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

ALLSELECTED

Artículo • 20/10/2023

Quita los filtros de contexto de las columnas y las filas de la consulta actual, a la vez que conserva todos los demás filtros de contexto o filtros explícitos.

La función ALLSELECTED obtiene el contexto que representa todas las filas y columnas de la consulta, a la vez que mantiene los filtros y contextos explícitos distintos de los filtros de fila y columna. Esta función se puede usar para obtener totales visuales en las consultas.

Sintaxis

DAX

```
ALLSELECTED([<tableName> | <columnName>[ , <columnName>[ , <columnName>[ ,...]]]])
```

Parámetros

Término	Definición
tableName	Nombre de una tabla existente con la sintaxis DAX estándar. Este parámetro no puede ser una expresión. Este parámetro es opcional.
columnName	Nombre de una columna existente, con la sintaxis DAX estándar, normalmente completa. No puede ser una expresión. Este parámetro es opcional.

Valor devuelto

Contexto de la consulta sin filtros de fila y columna.

Observaciones

- Si hay un argumento, el argumento es *tableName* o *columnName*. Si hay más de un argumento, deben ser columnas de la misma tabla.
- Esta función es diferente de ALL() porque conserva todos los filtros establecidos explícitamente en la consulta, además de todos los filtros de contexto distintos de los filtros de fila y columna.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En este ejemplo se muestra cómo generar diferentes niveles de totales visuales en un informe de tabla mediante expresiones DAX. En el informe se han aplicado dos (2) filtros anteriores a los datos de Reseller Sales: uno en Sales Territory Group = *Europe* y el otro en Promotion Type = *Volume Discount*. Una vez aplicados los filtros, se pueden calcular los totales visuales para todo el informe, para las categorías All Years o para All Products. Además, con fines ilustrativos, también se obtiene el total general All Reseller Sales y se quitan todos los filtros del informe. La evaluación de esta expresión DAX devuelve una tabla con toda la información necesaria para generar una tabla con totales visuales.

DAX

```

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]

```

Después de ejecutar la expresión anterior en SQL Server Management Studio en el modelo tabular AdventureWorks DW, se obtienen estos resultados:

[Product Category Name]	[Calendar Year]	[Reseller Sales Amount]	[Reseller Grand Total]	[Reseller Visual Total]	[Reseller Visual Total for All of Calendar Year]	[Reseller Visual Total for All of Product Category Name]
Accesorios	2000		80450596,9823	877006,7987	38786,018	
Accesorios	2001		80450596,9823	877006,7987	38786,018	
Accesorios	2002	625,7933	80450596,9823	877006,7987	38786,018	91495,3104
Accesorios	2003	26037,3132	80450596,9823	877006,7987	38786,018	572927,0136
Accesorios	2004	12122,9115	80450596,9823	877006,7987	38786,018	212584,4747
Accesorios	2005		80450596,9823	877006,7987	38786,018	
Accesorios	2006		80450596,9823	877006,7987	38786,018	
Bicicletas	2000		80450596,9823	877006,7987	689287,7939	
Bicicletas	2001		80450596,9823	877006,7987	689287,7939	
Bicicletas	2002	73778,938	80450596,9823	877006,7987	689287,7939	91495,3104
Bicicletas	2003	439771,4136	80450596,9823	877006,7987	689287,7939	572927,0136
Bicicletas	2004	175737,4423	80450596,9823	877006,7987	689287,7939	212584,4747
Bicicletas	2005		80450596,9823	877006,7987	689287,7939	
Bicicletas	2006		80450596,9823	877006,7987	689287,7939	
Ropa	2000		80450596,9823	877006,7987	95090,7757	
Ropa	2001		80450596,9823	877006,7987	95090,7757	
Ropa	2002	12132,4334	80450596,9823	877006,7987	95090,7757	91495,3104
Ropa	2003	58234,2214	80450596,9823	877006,7987	95090,7757	572927,0136

[Product Category Name]	[Calendar Year]	[Reseller Sales Amount]	[Reseller Grand Total]	[Reseller Visual Total]	[Reseller Visual Total for All of Calendar Year]	[Reseller Visual Total for All of Product Category Name]
Ropa	2004	24724,1209	80450596,9823	877006,7987	95090,7757	212584,4747
Ropa	2005		80450596,9823	877006,7987	95090,7757	
Ropa	2006		80450596,9823	877006,7987	95090,7757	
Componentes	2000		80450596,9823	877006,7987	53842,2111	
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	

Las columnas del informe son:

Reseller Sales Amount

Valor real de Reseller Sales para el año y la categoría de producto. Este valor aparece en una celda del centro del informe, en la intersección de año y categoría.

Reseller Visual Total for All of Calendar Year

Valor total de una categoría de producto en todos los años. Este valor aparece al final de una columna o fila para una categoría de producto determinada y en todos los años del informe.

Reseller Visual Total for All of Product Category Name

Valor total de un año en todas las categorías de productos. Este valor aparece al final de una columna o fila para un año determinado y en todas las categorías de producto del informe.

Reseller Visual Total

Valor total de todos los años y categorías de producto. Este valor suele aparecer en la esquina inferior derecha de la tabla.

Reseller Grand Total

Este es el total general de todos los valores de Reseller Sales, antes de que se aplique cualquier filtro; observe la diferencia con [Reseller Visual Total]. Recuerde que este informe incluye dos (2) filtros, uno en Product Category Group y otro en Promotion Type.

ⓘ Nota

Si tiene filtros explícitos en la expresión, estos filtros también se aplican a la expresión.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

CALCULATE

Artículo • 20/10/2023

Evalúa una expresión en un contexto de filtro modificado.

ⓘ Nota

También está la función **CALCULATETABLE**. Hace exactamente lo mismo, salvo que modifica el **contexto de filtro** aplicado a una expresión que devuelve un *objeto de tabla*.

Sintaxis

DAX

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

Parámetros

Término	Definición
expression	Expresión que se va a evaluar.
filter1, filter2,...	(Opcional) Expresiones booleanas o expresiones de tabla que definen filtros o funciones de modificador de filtro.

La expresión usada como primer parámetro es esencialmente la misma que una medida.

Los filtros pueden ser:

- Expresiones de filtro booleanas
- Expresiones de filtro de tabla
- Funciones de modificación de filtro

Cuando hay varios filtros, se pueden evaluar mediante el **operador lógico AND** (`&&`), lo que significa que todas las condiciones deben ser verdaderas, o bien el operador lógico **OR** (`||`), lo que significa que cualquiera de las condiciones puede ser true.

Expresiones de filtro booleanas

Un filtro de expresión booleana es una expresión que se evalúa como TRUE o FALSE.

Hay varias reglas que deben cumplir lo siguiente:

- Deben hacer referencia a columnas de una sola tabla.
- No pueden hacer referencia a medidas.
- No pueden usar una función CALCULATE anidada.

A partir de la versión de septiembre de 2021 de Power BI Desktop, también se aplica lo siguiente:

- No pueden usar funciones que analicen o devuelvan una tabla a menos que se pasen como argumentos a las funciones de agregación.
- *Pueden* contener una función de agregación que devuelve un valor escalar. Por ejemplo:

DAX

```
Total sales on the last selected date =  
CALCULATE (  
    SUM ( Sales[Sales Amount] ),  
    'Sales'[OrderDateKey] = MAX ( 'Sales'[OrderDateKey] )  
)
```

Expresión de filtro de tabla

Un filtro de expresión de tabla aplica un objeto de tabla como un filtro. Podría ser una referencia a una tabla de modelo, pero lo más probable es que sea una función que devuelva un objeto de tabla. Puede usar la función FILTER para aplicar condiciones de filtro complejas, incluidas las que no se pueden definir mediante una expresión de filtro booleana.

Funciones de modificador de filtro

Las funciones del modificador de filtro permiten hacer algo más que simplemente agregar filtros. Proporcionan un control adicional al modificar el contexto de filtro.

Función	Finalidad
REMOVEFILTERS	Quitar todos los filtros, o los filtros de una o varias columnas de una tabla o de todas las columnas de una sola tabla.
ALL ¹ , ALLEXCEPT, ALLNOBLANKROW	Quitar los filtros de una o varias columnas, o de todas las columnas de una sola tabla.

Función	Finalidad
KEEPFILTERS	Agregar un filtro sin quitar los existentes en las mismas columnas.
USERELATIONSHIP	Activar una relación inactiva entre las columnas relacionadas, en cuyo caso la relación activa se convertirá automáticamente en inactiva.
CROSSFILTER	Modificar la dirección del filtro (de ambas a única o de única a ambas) o deshabilitar una relación.

¹ La función ALL y sus variantes se comportan como modificadores de filtro y como funciones que devuelven objetos de tabla. Si la herramienta admite la función REMOVEFILTERS, es mejor usarla para quitar filtros.

Valor devuelto

Valor que es el resultado de la expresión.

Comentarios

- Cuando se proporcionan expresiones de filtro, la función CALCULATE modifica el contexto de filtro para evaluar la expresión. Cuando la expresión de filtro no está incluida en la función KEEPFILTERS, para cada expresión de filtro hay dos posibles resultados estándar:
 - Si las columnas (o tablas) no están en el contexto de filtro, para evaluar la expresión se agregarán filtros nuevos al contexto de filtro.
 - Si las columnas (o tablas) ya están en el contexto de filtro, para evaluar la expresión CALCULATE los filtros nuevos sobrescribirán los existentes.
- La función CALCULATE usada *sin filtros* alcanza un requisito específico. Realiza la transición del contexto de fila al contexto de filtro. Es necesario cuando una expresión (no una medida de modelo) que resume los datos del modelo debe evaluarse en el contexto de fila. Este escenario puede producirse en una fórmula de columna calculada o cuando se evalúa una expresión en una función de iterador. Tenga en cuenta que, cuando se usa una medida de modelo en el contexto de fila, la transición de contexto es automática.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

La definición de medida de la tabla **Sales** siguiente genera un resultado de ingresos, pero solo para los productos en color azul.

Los ejemplos de este artículo se pueden usar con el modelo de ejemplo de Power BI Desktop de Adventure Works DW 2020. Para obtener el modelo, vea [Modelo de ejemplo de DAX](#).

DAX

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

Category	Sales Amount	Ingresos en azul
Accessories	1 272 057,89 USD	165 406,62 USD
Bikes	94 620 526,21 USD	8 374 313,88 USD
Clothing	2 117 613,45 USD	259 488,37 USD
Componentes	\$11,799,076.66	803 642,10 USD
Total	109 809 274,20 USD	9 602 850,97 USD

La función CALCULATE evalúa la suma de la columna **Sales Amount** de la tabla **Sales** en un contexto de filtro modificado. Se agrega un nuevo filtro a la columna **Color** de la tabla **Product**, o bien el filtro sobrescribe cualquier filtro que ya se haya aplicado a la columna.

La siguiente definición de medida de la tabla **Sales** genera una relación de ventas sobre ventas en todos los canales de ventas.

Canal	Sales Amount	Porcentaje total de ingresos del canal
Internet	\$29,358,677.22	26,74 %
Reseller	\$80,450,596.98	73,26 %
Total	109 809 274,20 USD	100 %

DAX

```
Revenue % Total Channel =  
DIVIDE(  
    SUM(Sales[Sales Amount]),
```

```
CALCULATE(  
    SUM(Sales[Sales Amount]),  
    REMOVEFILTERS('Sales Order'[Channel])  
)  
)
```

La función **DIVIDE** divide una expresión que suma el valor de la columna **Sales Amount** de la tabla **Sales** (en el contexto de filtro) entre la misma expresión en un contexto de filtro modificado. La función **CALCULATE** modifica el contexto de filtro mediante la función **REMOVEFILTERS**, que es una función de modificador de filtro. Quita los filtros de la columna **Channel** de la tabla **Sales Order**.

En la siguiente definición de columna calculada de la tabla **Customer** se clasifican los clientes en una clase de fidelidad. Se trata de un escenario muy sencillo: Cuando los ingresos que genera el cliente son inferiores a 2500 USD, se clasifican como *Low*; de lo contrario, como *High*.

DAX

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

En este ejemplo, el contexto de fila se convierte en el contexto de filtro. Esto se conoce como *transición de contexto*. La función **ALLEXCEPT** quita los filtros de todas las columnas de la tabla **Customer**, excepto los de la columna **CustomerKey**.

Vea también

[Contexto de filtro](#)

[Contexto de fila](#)

[Función CALCULATETABLE](#)

[Funciones de filtro](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

CALCULATETABLE

Artículo • 20/10/2023

Evalúa una expresión de tabla en un contexto de filtro modificado.

ⓘ Nota

También está la función **CALCULATE**. Hace exactamente lo mismo, salvo que modifica el **contexto de filtro** aplicado a una expresión que devuelve un *valor escalar*.

Sintaxis

DAX

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

Parámetros

Término	Definición
expression	Expresión de tabla que se va a evaluar.
filter1, filter2,...	(Opcional) Expresiones booleanas o expresiones de tabla que definen filtros o funciones de modificador de filtro.

La expresión usada como primer parámetro debe ser una tabla de modelo o una función que devuelve una tabla.

Los filtros pueden ser:

- Expresiones de filtro booleanas
- Expresiones de filtro de tabla
- Funciones de modificación de filtro

Cuando hay varios filtros, se evalúan mediante el [operador lógico AND](#). Esto significa que todas las condiciones deben ser "true" a la vez.

Expresiones de filtro booleanas

Un filtro de expresión booleana es una expresión que se evalúa como TRUE o FALSE. Hay varias reglas que deben cumplir lo siguiente:

- Solo pueden hacer referencia a una única columna.
- No pueden hacer referencia a medidas.
- No pueden usar una función CALCULATE anidada.

A partir de la versión de septiembre de 2021 de Power BI Desktop, también se aplica lo siguiente:

- No pueden usar funciones que analicen o devuelvan una tabla a menos que se pasen como argumentos a las funciones de agregación.
- *Pueden* contener una función de agregación que devuelve un valor escalar.

Expresión de filtro de tabla

Un filtro de expresión de tabla aplica un objeto de tabla como un filtro. Podría ser una referencia a una tabla de modelo, pero lo más probable es que sea una función que devuelva un objeto de tabla. Puede usar la función **FILTER** para aplicar condiciones de filtro complejas, incluidas las que no se pueden definir mediante una expresión de filtro booleana.

Funciones de modificador de filtro

Las funciones del modificador de filtro permiten hacer algo más que simplemente agregar filtros. Proporcionan un control adicional al modificar el contexto de filtro.

Función	Finalidad
REMOVEFILTERS	Quitar todos los filtros, o los filtros de una o varias columnas de una tabla o de todas las columnas de una sola tabla.
ALL ¹ , ALLEXCEPT, ALLNOBLANKROW	Quitar los filtros de una o varias columnas, o de todas las columnas de una sola tabla.
KEEPFILTERS	Agregar un filtro sin quitar los existentes en las mismas columnas.
USERELATIONSHIP	Activar una relación inactiva entre las columnas relacionadas, en cuyo caso la relación activa se convertirá automáticamente en inactiva.
CROSSFILTER	Modificar la dirección del filtro (de ambas a única o de única a ambas) o deshabilitar una relación.

¹ La función ALL y sus variantes se comportan como modificadores de filtro y como funciones que devuelven objetos de tabla. Si la herramienta admite la función REMOVEFILTERS, es mejor usarla para quitar filtros.

Valor devuelto

Una tabla de valores.

Comentarios

- Cuando se proporcionan expresiones de filtro, la función CALCULATETABLE modifica el contexto de filtro para evaluar la expresión. Cuando la expresión de filtro no está incluida en la función KEEPFILTERS, para cada expresión de filtro hay dos posibles resultados estándar:
 - Si las columnas (o tablas) no están en el contexto de filtro, para evaluar la expresión se agregarán filtros nuevos al contexto de filtro.
 - Si las columnas (o tablas) ya están en el contexto de filtro, para evaluar la expresión CALCULATETABLE los filtros nuevos sobrescribirán los existentes.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se usa la función CALCULATETABLE para obtener la suma de las ventas por Internet de 2006. Este valor se usa posteriormente para calcular la relación de ventas por Internet con respecto a todas las ventas del año 2006.

La fórmula siguiente:

```
DAX  
= SUMX(  
    CALCULATETABLE(  
        'InternetSales_USD',  
        'DateTime'[CalendarYear] = 2006  
    ),  
    [SalesAmount_USD]  
)
```

Da como resultado la tabla siguiente:

Etiquetas de fila	Internet SalesAmount_USD	CalculateTable 2006 Internet Sales	Internet Sales to 2006 ratio
2005	2 627 031,40 \$	5 681 440,58 \$	0,46
2006	5 681 440,58 \$	5 681 440,58 \$	1.00
2007	8 705 066,67 \$	5 681 440,58 \$	1,53
2008	9 041 288,80 \$	5 681 440,58 \$	1,59
Total general	26 054 827,45 \$	5 681 440,58 \$	4.59

Vea también

- [Contexto de filtro](#)
- [Función CALCULATE \(DAX\)](#)
- [Funciones de filtro \(DAX\)](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

EARLIER

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✗ Cálculo visual

Devuelve el valor actual de la columna especificada en un paso de evaluación externo de la columna mencionada.

EARLIER es útil para los cálculos anidados en los que se quiere usar un determinado valor como entrada y generar cálculos basados en esa entrada. En Microsoft Excel, estos cálculos solo se pueden realizar dentro del contexto de la fila actual. Sin embargo, en DAX puede almacenar el valor de la entrada y luego hacer cálculos con los datos de toda la tabla.

EARLIER se usa principalmente en el contexto de columnas calculadas.

Sintaxis

DAX

`EARLIER(<column>, <number>)`

Parámetros

[] Expandir tabla

Término	Definición
columna	Columna o expresión que se resuelve en una columna.
num	(Opcional) Número positivo para el paso de evaluación externo. El siguiente nivel de evaluación externo se representa mediante 1; dos niveles externos se representan mediante 2, y así sucesivamente. Si se omite, el valor predeterminado es 1.

Valor devuelto

Valor actual de fila, desde **column**, en **number** de pasos de evaluación externos.

Excepciones

Descripción de errores

Notas

- **EARLIER** se ejecuta correctamente si hay un contexto de fila antes del inicio del recorrido de tabla. De lo contrario, devuelve un error.
- El rendimiento de **EARLIER** puede ser lento porque, en teoría, puede tener que realizar una serie de operaciones cercana al número total de filas (en la columna) por el mismo número (en función de la sintaxis de la expresión). Por ejemplo, si tiene 10 filas en la columna, se podrían necesitar aproximadamente 100 operaciones; si tiene 100 filas, podrían realizarse casi 10 000 operaciones.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

① Nota

En la práctica, el motor de análisis en memoria VertiPaq realiza optimizaciones para reducir el número real de cálculos, pero debe tener cuidado al crear fórmulas que impliquen recursividad.

Ejemplo

Para mostrar el uso de **EARLIER**, es necesario crear un escenario que calcule un valor de clasificación y luego use ese valor en otros cálculos.

El ejemplo siguiente se basa en esta tabla simple, **ProductSubcategory**, que muestra las ventas totales de cada ProductSubcategory.

La tabla final, incluida la columna de clasificación, se muestra aquí.

 Expandir tabla

ProductSubcategoryKey	EnglishProductSubcategoryName	TotalSubcategorySales	SubcategoryRanking
18	Pantalones de ciclismo	156 167,88 \$	18
26	Bastidores de bicicletas	220 720,70 \$	14
27	Soportes de bicicletas	35 628,69 \$	30
28	Bidones y soportes	59 342,43 \$	24
5	Ejes pedalier	48 643,47 \$	27
6	Frenos	62 113,16 \$	23
19	Gorras	47 934,54 \$	28
7	Cadenas	8 847,08 \$	35

ProductSubcategoryKey	EnglishProductSubcategoryName	TotalSubcategorySales	SubcategoryRanking
29	Limpiadores	16 882,62 \$	32
8	Bielas	191 522,09 \$	15
9	Desviadores	64 965,33 \$	22
30	Guardabarros	41 974,10 \$	29
10	Horquillas	74 727,66 \$	21
20	Guantes	228 353,58 \$	12
4	Manillares	163 257,06 \$	17
11	Tubos de dirección	57 659,99 \$	25
31	Cascos	451 192,31 \$	9
32	Paquetes de hidratación	96 893,78 \$	20
21	Maillots	699 429,78 \$	7
33	Luces		36
34	Bloqueos	15 059,47 \$	33
1	Bicicletas de montaña	34 305 864,29 \$	2
12	Cuadros de bicicletas de montaña	4 511 170,68 \$	4
35	Alforjas		36
13	Pedales	140 422,20 \$	19
36	Bombas	12 695,18 \$	34
2	Bicicletas de carretera	40 551 696,34 \$	1
14	Cuadros de bicicletas de carretera	3 636 398,71 \$	5
15	Sillines	52 526,47 \$	26
22	Shorts	385 707,80 \$	10
23	Calcetines	28 337,85 \$	31
24	Medias	189 179,37 \$	16
37	Cubiertas y tubulares	224 832,81 \$	13
3	Bicicletas de paseo	13 334 864,18 \$	3
16	Cuadros de bicicletas de paseo	1 545 344,02 \$	6
25	Chalecos	240 990,04 \$	11

ProductSubcategoryKey	EnglishProductSubcategoryName	TotalSubcategorySales	SubcategoryRanking
17	Ruedas	648 240,04 \$	8

Creación de un valor de clasificación

Una manera de obtener un valor de clasificación de un valor determinado de una fila es contar el número de filas de la misma tabla que tienen un valor mayor (o menor) que el que se está comparando. Esta técnica devuelve un espacio en blanco o un valor cero para el valor más alto de la tabla, mientras que los valores iguales tienen el mismo valor de clasificación y el siguiente valor (después de los valores iguales) tiene un valor de clasificación no consecutivo. Vea el ejemplo siguiente.

Se crea una nueva columna calculada, **SubCategorySalesRanking**, con la fórmula siguiente.

DAX

```
= COUNTROWS(FILTER(ProductSubcategory,
EARLIER(ProductSubcategory[TotalSubcategorySales])
<ProductSubcategory[TotalSubcategorySales]))+1
```

En los pasos siguientes se describe el método de cálculo más detalladamente.

1. La función **EARLIER** obtiene el valor de *TotalSubcategorySales* de la fila actual de la tabla. En este caso, dado que el proceso se está iniciando, es la primera fila de la tabla.
2. **EARLIER([TotalSubcategorySales])** se evalúa en 156 167,88 \$, la fila actual del bucle externo.
3. La función **FILTER** devuelve ahora una tabla en la que todas las filas tienen un valor de *TotalSubcategorySales* mayor que 156 167,88 \$ (que es el valor actual de **EARLIER**).
4. La función **COUNTROWS** cuenta las filas de la tabla filtrada y asigna ese valor a la nueva columna calculada de la fila actual más 1. Es necesario agregar 1 para evitar que el valor de clasificación superior se convierta en un espacio en blanco.
5. La fórmula de la columna calculada se desplaza a la siguiente fila y repite los pasos del 1 al 4. Estos pasos se repiten hasta que se alcanza el final de la tabla.

La función **EARLIER** siempre obtiene el valor de la columna antes de la operación de tabla actual. Si necesita obtener un valor del bucle antes de eso, establezca el segundo argumento en 2.

Contenido relacionado

[Función EARLIEST](#)

[Funciones de filtro](#)

Comentarios

¿Le ha resultado útil esta página?



EARLIEST

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✗ Cálculo visual

Devuelve el valor actual de la columna especificada en un paso de evaluación externo de la columna especificada.

Sintaxis

DAX

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

Parámetros

[+] Expandir tabla

Término	Definición
columna	Referencia a una columna.

Valor devuelto

Columna con filtros quitados.

Notas

- La función EARLIEST es similar a EARLIER, aunque permite especificar un nivel de recursividad adicional.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Los datos del ejemplo actual no admiten este escenario.

DAX

```
= EARLiest(<column>)
```

Contenido relacionado

[Función EARLIER](#)

[Funciones de filtro](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

FILTER

Artículo • 14/12/2023

Devuelve una tabla que representa un subconjunto de otra tabla o expresión.

Sintaxis

DAX

```
FILTER(<table>,<filter>)
```

Parámetros

 Expandir tabla

Término	Definición
tabla	Tabla que se va a filtrar. La tabla también puede ser una expresión que da lugar a una tabla.
filter	Expresión booleana que se va a evaluar para cada fila de la tabla. Por ejemplo, <code>[Amount] > 0 O [Region] = "France".</code>

Valor devuelto

Tabla que contiene solo las filas filtradas.

Notas

- Puede usar FILTER para reducir el número de filas de la tabla con la que está trabajando y emplear solo datos concretos en los cálculos. FILTER no se usa independientemente, sino como una función insertada en otras funciones que requieren una tabla como argumento.
- Para conocer los procedimientos recomendados al usar FILTER, consulte [Impedimento del uso de FILTER como argumento de filtro](#).

Uso de COUNTROWS en lugar de COUNT en DAX

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se crea un informe de ventas por Internet fuera de los Estados Unidos mediante una medida que filtra las ventas en los Estados Unidos y luego el desglose por año natural y categorías de producto. Para crear esta medida, se filtra la tabla, Internet Sales USD, mediante Sales Territory y luego se usa la tabla filtrada en una función SUMX.

En este ejemplo, la expresión:

DAX

```
FILTER('InternetSales_USD', RELATED('SalesTerritory'[SalesTerritoryCountry])
<>"United States")
```

Devuelve una tabla que es un subconjunto de ventas por Internet menos todas las filas que pertenecen al territorio de ventas de Estados Unidos. La función RELATED es lo que vincula la clave Territory de la tabla Internet Sales a SalesTerritoryCountry en la tabla SalesTerritory.

En la tabla siguiente se muestra la prueba de concepto de la medida, ventas por Internet fuera de los Estados Unidos, la fórmula para la cual se proporciona en la sección de código siguiente. En la tabla se comparan todas las ventas por Internet con las ventas por Internet fuera de los Estados Unidos, para mostrar que la expresión de filtro funciona, al excluir las ventas de los Estados Unidos del cálculo.

Para volver a crear esta tabla, agregue el campo SalesTerritoryCountry al área **Etiquetas de fila** de un informe o una tabla dinámica.

Tabla 1. Comparación de las ventas totales para EE. UU. y todas las demás regiones

 Expandir tabla

Etiquetas de fila	Internet Sales	Ventas por Internet fuera de EE. UU.
Australia	4 999 021,84 \$	4 999 021,84 \$
Canadá	1 343 109,10 \$	1 343 109,10 \$

Etiquetas de fila	Internet Sales	Ventas por Internet fuera de EE. UU.
Francia	2 490 944,57 \$	2 490 944,57 \$
Alemania	2 775 195,60 \$	2 775 195,60 \$
Reino Unido	5 057 076,55 \$	5 057 076,55 \$
Estados Unidos	9 389 479,79 \$	
Total general	26 054 827,45 \$	16 665 347,67 \$

En la tabla de informe final se muestran los resultados al crear una tabla dinámica mediante la medida, Ventas por Internet fuera de EE. UU. Agregue el campo CalendarYear al área **Etiquetas de fila** de la tabla dinámica y el campo ProductCategoryName al área **Etiquetas de columna**.

Tabla 2. Comparación de ventas fuera de EE. UU. por categorías de producto

 Expandir tabla

Etiquetas de fila	Accesorios	Bicicletas	Ropa	Total general
2005		1 526 481,95 \$		1 526 481,95 \$
2006		3 554 744,04 \$		3 554 744,04 \$
2007	156 480,18 \$	5 640 106,05 \$	70 142,77 \$	5 866 729,00 \$
2008	228 159,45 \$	5 386 558,19 \$	102 675,04 \$	5 717 392,68 \$
Total general	384 639,63 \$	16 107 890,23 \$	172 817,81 \$	16 665 347,67 \$

DAX

```
SUMX(FILTER('InternetSales_USD',
RELATED('SalesTerritory'[SalesTerritoryCountry]))<>"United States")
,'InternetSales_USD'[SalesAmount_USD])
```

Contenido relacionado

[Funciones de filtro](#)

[Función ALL](#)

[ALLEXCEPT, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

INDEX

Artículo • 21/02/2024

Devuelve una fila en una posición absoluta, especificada por el parámetro position, dentro de la partición especificada con el orden especificado. Si la partición actual no se puede deducir como una sola partición, se pueden devolver varias filas.

Sintaxis

DAX

```
INDEX(<position>[, <relation> or <axis>][, <orderBy>][, <blanks>][, <partitionBy>][, <matchBy>][, <reset>] )
```

Parámetros

 Expandir tabla

Término	Definición
position	Posición absoluta (de base 1) desde la que se obtienen los datos: - <position> es una posición positiva: 1 es la primera fila, 2 es la segunda fila, etc. - <position> es una posición negativa: -1 es la última fila, -2 es la penúltima fila, etc. Cuando <position> está fuera del límite, o es cero o BLANK(), INDEX devuelve una tabla vacía. Puede ser cualquier expresión DAX que devuelva un valor escalar.
relation	(Opcional) Expresión de tabla a partir de la cual se devuelve la salida. Si se especifica, todas las columnas de <partitionBy> deben proceder de ella o de una tabla relacionada. Si se omite: - <orderBy> debe especificarse explícitamente. - Todas las expresiones de <orderBy> y <partitionBy> deben tener el nombre completo de la columna y proceder de una sola tabla. - Toma el valor predeterminado ALLSELECTED() de todas las columnas de <orderBy> y <partitionBy>.
ejes	(Opcional) Un eje en la forma visual. Solo está disponible en cálculos visuales y reemplaza la <relación>.
orderBy	(Opcional) Cláusula ORDERBY() que contiene las expresiones que definen cómo se ordena cada partición. Si se omite: - <relation> debe especificarse explícitamente. - De forma predeterminada, ordena por cada columna de <relation> que aún no se haya especificado en <partitionBy>.

Término	Definición
espacios en blanco	<p>(Opcional) Enumeración que define cómo controlar los valores en blanco en la ordenación.</p> <p>Este parámetro se reserva para uso futuro.</p> <p>Actualmente, el único valor admitido es DEFAULT, donde el comportamiento de los valores numéricos es que los valores en blanco se ordenan entre cero y los valores negativos. El comportamiento de las cadenas es que los valores en blanco se ordenan delante de todas las cadenas, incluidas las cadenas vacías.</p>
partitionBy	<p>(Opcional) Cláusula PARTITIONBY() que contiene las columnas que definen cómo se partitiona <relation>.</p> <p>Si se omite, <relation> se trata como una sola partición.</p>
matchBy	(Opcional) Una cláusula MATCHBY() que contiene las columnas que definen cómo hacer coincidir los datos e identificar la fila actual.
reset	<p>(Opcional) Solo está disponible en cálculos visuales. Indique si el cálculo se restablece y en qué nivel de la jerarquía de columnas de la forma visual. Los valores aceptados son: NONE, LOWESTPARENT, HIGHESTPARENT o un entero. El comportamiento depende del signo entero:</p> <ul style="list-style-type: none"> : si se omite cero, el cálculo no se restablece. Equivalente a NONE. : si es positivo, el entero identifica la columna a partir de la más alta, independiente del intervalo de agregación. HIGHESTPARENT es equivalente a 1. : si es negativo, el entero identifica la columna a partir del más bajo, en relación con el intervalo de agregación actual. LOWESTPARENT es equivalente a -1.

Valor devuelto

Una fila en una posición absoluta.

Observaciones

Cada columna de <partitionBy> y <matchBy> debe tener un valor externo correspondiente para ayudar a definir la "partición actual" en la que operar, con el siguiente comportamiento:

- Si hay exactamente una columna externa correspondiente, se utiliza su valor.
- Si no hay ninguna columna externa correspondiente:
 - INDEX determina primero todas las columnas de <partitionBy> y <matchBy> que no tienen ninguna columna externa correspondiente.
 - Para cada combinación de valores de estas columnas en el contexto primario de INDEX, se evalúa INDEX y se devuelve una fila.
 - La salida final de INDEX es una unión de estas filas.
- Si hay más de una columna externa correspondiente, se devuelve un error.

Si <matchBy> está presente, INDEX intentará usar las columnas <matchBy> y <partitionBy> para identificar la fila.

Si <matchBy> no está presente y las columnas especificadas en <orderBy> y <partitionBy> no pueden identificar de forma única todas las filas de <relation>:

- INDEX intenta buscar el menor número de columnas adicionales necesarias para identificar de forma única cada fila.
- Si se encuentran estas columnas, INDEX anexa automáticamente estas nuevas columnas a <orderBy> y cada partición se ordena usando este nuevo conjunto de columnas OrderBy.
- Si no se encuentran estas columnas, se devuelve un error.

Se devuelve una tabla vacía si:

- El valor externo correspondiente de una columna PartitionBy no existe en <relation>.
- El valor de <position> hace referencia a una posición que no existe en la partición.

Si se utiliza INDEX en una columna calculada definida en la misma tabla que <relation> y se omite <orderBy>, se devuelve un error.

<reset> solo se puede usar en cálculos visuales y no se puede usar en combinación con <orderBy> o <partitionBy>. Si <reset> está presente, se puede especificar el <eje>, pero no se puede especificar la <relación>.

Ejemplo 1: columna calculada

La consulta DAX siguiente:

DAX
EVALUATE INDEX(1, ALL(DimDate[CalendarYear]))

Devuelve la tabla siguiente:

 Expandir tabla

DimDate[CalendarYear]
2005

Ejemplo 2: columna calculada

La consulta DAX siguiente:

DAX

```
EVALUATE
SUMMARIZECOLUMNS (
    FactInternetSales[ProductKey],
    DimDate[MonthNumberOfYear],
    FILTER (
        VALUES(FactInternetSales[ProductKey]),
        [ProductKey] < 222
    ),
    "CurrentSales", SUM(FactInternetSales[SalesAmount]),
    "LastMonthSales",
    CALCULATE (
        SUM(FactInternetSales[SalesAmount]),
        INDEX(-1, ORDERBY(DimDate[MonthNumberOfYear]))
    )
)
ORDER BY [ProductKey], [MonthNumberOfYear]
```

Devuelve la tabla siguiente:

 Expandir tabla

FactInternetSales[ProductKey]	DimDate[MonthNumberOfYear]	[CurrentSales]	[LastMonthSales]
214	1	5423.45	8047.7
214	2	4968.58	8047.7
214	3	5598.4	8047.7
214	4	5073.55	8047.7
214	5	5248.5	8047.7
214	6	7487.86	8047.7
214	7	7382.89	8047.7
214	8	6543.13	8047.7
214	9	6788.06	8047.7
214	10	6858.04	8047.7
214	11	8607.54	8047.7
214	12	8047.7	8047.7
217	1	5353.47	7767.78

FactInternetSales[ProductKey]	DimDate[MonthNumberOfYear]	[CurrentSales]	[LastMonthSales]
217	2	4268.78	7767.78
217	3	5773.35	7767.78
217	4	5738.36	7767.78
217	5	6158.24	7767.78
217	6	6998	7767.78
217	7	5563.41	7767.78
217	8	5913.31	7767.78
217	9	5913.31	7767.78
217	10	6823.05	7767.78
217	11	6683.09	7767.78
217	12	7767.78	7767.78

Ejemplo 3: cálculo visual

Las siguientes consultas DAX de cálculo visual:

```
DAX

SalesComparedToBeginningOfYear = [SalesAmount] -
CALCULATE(SUM([SalesAmount]), INDEX(1, ROWS, HIGHESTPARENT))

SalesComparedToBeginningOfQuarter = [SalesAmount] -
CALCULATE(SUM([SalesAmount]), INDEX(1, , -1))
```

Mejorar una tabla para que contenga, para cada mes:

- el importe total de las ventas;
- la diferencia con el primer mes del año respectivo;
- : y la diferencia con el primer mes del trimestre respectivo.

En la siguiente captura de pantalla se muestra la matriz visual y la primera expresión de cálculo visual:

				<code>1 SalesComparedToBeginningOfYear = [SalesAmount] - CALCULATE(SUM([SalesAmount]), INDEX(1, ROWS, HIGHESTPARENT))</code>		
CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	SalesComparedToBeginningOfYear	SalesComparedToBeginningOfQuarter	
2011	1	1	\$323,743.2056	0.00	0.00	
		2	\$437,372.4476	113,629.24	113,629.24	
		3	\$514,918.2284	191,175.02	191,175.02	
		Total	\$1,276,033.8816	0.00	0.00	
		4	\$523,950.0258	200,206.82	0.00	
		5	\$539,469.0004	215,725.79	15,518.97	
		6	\$647,524.214	323,781.01	123,574.19	

Contenido relacionado

[OFFSET](#)

[ORDERBY](#)

[PARTITIONBY](#)

[WINDOW](#)

[RANK](#)

[ROWNUMBER](#)

Comentarios

¿Le ha resultado útil esta página?

 [Sí](#)

 [No](#)

KEEPFILTERS

Artículo • 20/10/2023

Modifica el modo en que se aplican los filtros mientras se evalúa una función CALCULATE o CALCULATETABLE.

Sintaxis

DAX

`KEEPFILTERS(<expression>)`

Parámetros

Término	Definición
expression	Cualquier expresión.

Valor devuelto

Una tabla de valores.

Notas

- Puede usar KEEPFILTERS en las funciones de contexto CALCULATE y CALCULATETABLE para invalidar el comportamiento estándar de esas funciones.
- De forma predeterminada, los argumentos de filtro en funciones como CALCULATE se usan como contexto para evaluar la expresión y, por tanto, los argumentos de filtro de CALCULATE reemplazan todos los filtros existentes en las mismas columnas. El nuevo contexto que ha realizado el argumento de filtro para CALCULATE solo afecta a los filtros existentes en las columnas mencionadas como parte del argumento de filtro. Los filtros en columnas distintas de las mencionadas en los argumentos de CALCULATE u otras funciones relacionadas permanecen en vigor y sin modificar.
- La función KEEPFILTERS permite modificar este comportamiento. Si usa KEEPFILTERS, todos los filtros existentes en el contexto actual se comparan con las columnas de los argumentos de filtro y la intersección de esos argumentos se usa

como contexto para evaluar la expresión. El efecto real en cualquier columna es que se aplican ambos conjuntos de argumentos: los argumentos de filtro que se usan en CALCULATE y los filtros en los argumentos de la función KEEPFILTER. En otras palabras, mientras que los filtros de CALCULATE reemplazan el contexto actual, KEEPFILTERS agrega filtros al contexto actual.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se muestran algunos escenarios comunes en los que se muestra el uso de la función KEEPFILTERS como parte de una fórmula CALCULATE o CALCULATETABLE.

Las tres primeras expresiones obtienen datos simples que se usarán para las comparaciones:

- Ventas por Internet del estado de Washington.
- Ventas por Internet de los estados de Washington y Oregón (ambos estados combinados).
- Ventas por Internet del estado de Washington y la provincia de Columbia Británica (ambas regiones combinadas).

La cuarta expresión calcula las ventas por Internet de Washington y Oregón, mientras que se aplica el filtro de Washington y Columbia Británica.

La siguiente expresión calcula las ventas por Internet de Washington y Oregón, pero usa KEEPFILTERS; el filtro de Washington y Columbia Británica forma parte del contexto anterior.

DAX

```
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"
        )
    )
)

```

Cuando esta expresión se evalúa en la base de datos de ejemplo AdventureWorks DW, se obtienen los siguientes resultados.

Columna	Valor
[\$\$ in WA]	2 467 248,34 USD
[\$\$ in WA and OR]	3 638 239,88 USD
[\$\$ in WA and BC]	4 422 588,44 USD
[\$\$ in WA and OR ??]	3 638 239,88 USD
[\$\$ in WA !!!]	2 467 248,34 USD

ⓘ Nota

Los resultados anteriores tenían el formato de una tabla, en lugar de una sola fila, con fines educativos.

En primer lugar, examine la expresión **[\$\$ in WA and OR ??]**. Tal vez se pregunte cómo esta fórmula puede devolver el valor de las ventas en Washington y Oregón, ya que la expresión CALCULATE externa incluye un filtro para Washington y Columbia Británica. La respuesta es que el comportamiento predeterminado de CALCULATE invalida los filtros

externos de 'Geography'[State Province Code] y sustituye sus propios argumentos de filtro, porque los filtros se aplican a la misma columna.

Después, examine la expresión `[$$ in WA !!]`. Tal vez se pregunte cómo esta fórmula puede devolver el valor de las ventas en Washington y nada más, ya que el filtro de argumentos incluye Oregón y la expresión CALCULATE externa incluye un filtro en Washington y Columbia Británica. La respuesta es que KEEPFILTERS modifica el comportamiento predeterminado de CALCULATE y agrega un filtro adicional. Dado que se usa la intersección de filtros, ahora se agrega el filtro externo `'Geography'[State Province Code]="WA" || 'Geography'[State Province Code]="BC"`) al argumento de filtro `'Geography'[State Province Code]="WA" || 'Geography'[State Province Code]="OR"`. Dado que ambos filtros se aplican a la misma columna, el filtro resultante `'Geography'[State Province Code]="WA"` es el que se aplica al evaluar la expresión.

Vea también

[Funciones de filtro](#)

[CALCULATE, función](#)

[Función CALCULATETABLE](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

LOOKUPVALUE

Artículo • 20/10/2023

Devuelve el valor para la fila que cumple todos los criterios que especifica al menos una de las condiciones de búsqueda.

Sintaxis

DAX

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

Parámetros

Término	Definición
result_columnName	Nombre de una columna existente que contiene el valor que se quiere devolver. No puede ser una expresión.
search_columnName	Nombre de una columna existente. Puede estar en la misma tabla que result_columnName o en una relacionada. No puede ser una expresión. Se pueden especificar varios pares de search_columnName y search_value.
search_value	Valor que se buscará en search_columnName. Se pueden especificar varios pares de search_columnName y search_value.
alternateResult	(Opcional) Valor devuelto cuando el contexto de result_columnName se ha filtrado a cero o a más de un valor distinto. Si no se especifica, la función devuelve BLANK cuando result_columnName se filtra a cero valores o un error cuando hay más de un valor distinto en el contexto para result_columnName.

Valor devuelto

El valor de **result_columnName** en la fila donde todos los pares de **result_columnName** y **search_value** tienen una coincidencia exacta.

Si no hay ninguna coincidencia que cumpla todos los valores de búsqueda, se devuelve BLANK o **alternateResult**, si se proporciona. En otras palabras, la función no devolverá un valor de búsqueda si solo coinciden algunos de los criterios.

Si varias filas coinciden con los valores de búsqueda y los valores de **result_columnName** para estas filas son idénticos, se devuelve ese valor. En cambio, si **result_columnName** devuelve valores distintos, se devuelve un error o **alternateResult**, si se proporciona.

Comentarios

- Si existe una relación entre la tabla que contiene la columna de resultados y las tablas que contienen las columnas de búsqueda, el uso de la función **RELATED** en lugar de **LOOKUPVALUE** resulta más eficiente y proporciona un rendimiento mejor.
- Se pueden especificar varios pares de **search_columnName** y **search_value**.
- Los parámetros **search_value** y **alternateResult** se evalúan antes de que la función itere en las filas de la tabla de búsqueda.
- Evite usar las funciones **ISERROR** o **IFERROR** para capturar un error devuelto por **LOOKUPVALUE**. Si algunas entradas de la función producen un error cuando no se puede determinar un valor de salida único, proporcionar un parámetro **alternateResult** es la manera más confiable y que mejor funciona para administrar el error.
- El parámetro **alternateResult** devolverá un error si se especifica en una columna calculada de Power Pivot.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

En este ejemplo, **LOOKUPVALUE** se emplea para buscar la tasa media de la moneda utilizada para pagar el pedido el día en que se realizó el pedido:

DAX

```
Exchange Rate =  
LOOKUPVALUE ( 'Currency Rate'[Average Rate],  
'Currency Rate'[CurrencyKey], [CurrencyKey],  
'Currency Rate'[DateKey], [OrderDateKey]
```

)

Tanto la fecha de pedido como la moneda son necesarias para encontrar la tasa media de la fecha y la moneda correctas. OrderDateKey y CurrencyKey son las claves que se usan para buscar la tasa media en la tabla Tipo de moneda.

Puede usar el tipo de cambio para calcular el importe de Sales ("ventas") en moneda local con:

DAX

```
Sales Amount Local Currency = [Sales Amount] * [Exchange Rate]
```

Ejemplo 2

En este ejemplo, la siguiente columna calculada que se define en la tabla **Sales** utiliza la función LOOKUPVALUE para devolver los valores de canal de la tabla **Sales Order**.

DAX

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

Sin embargo, en este caso, dado que hay una relación entre las tablas **Sales Order** y **Sales**, resulta más eficaz usar la función **RELATED**.

DAX

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

Vea también

[Función RELATED \(DAX\)](#)

[Funciones de información](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

MATCHBY

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

En las funciones window, define las columnas que se usan para determinar cómo hacer coincidir los datos e identificar la fila actual.

Sintaxis

DAX

```
MATCHBY ( [<matchBy_columnName>[, matchBy_columnName [, ...]]] )
```

Parámetros

[+] Expandir tabla

Término	Definición
matchBy_columnName	(Opcional) Nombre de una columna existente que se va a usar para particionar el parámetro <relation> de una función window. RELATED() también se puede usar para hacer referencia a una columna de una tabla relacionada con <relation>.

Valor devuelto

Esta función no devuelve ningún valor.

Observaciones

Esta función solo se puede usar en una expresión de una función WINDOW.

Ejemplo

Consulte [OFFSET](#).

Contenido relacionado

[Descripción de las funciones ORDERBY, PARTITIONBY y MATCHBY](#)

[INDEX](#)

[OFFSET](#)

[ORDERBY](#)

[PARTITIONBY](#)

[WINDOW](#)

[RANK](#)

[ROWNUMBER](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MOVINGAVERAGE

Artículo • 21/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve un promedio móvil calculado a lo largo del eje especificado de la matriz visual. Es decir, el promedio de la columna especificada calculado en las últimas filas <windowSize>.

Sintaxis

DAX

```
MOVINGAVERAGE ( <column>, <windowSize>[, <includeCurrent>][, <axis>][, <blanks>][, <reset>] )
```

Parámetros

Expandir tabla

Término	Definición
columna	Columna que proporciona el valor de cada elemento.
windowSize	Número de filas que se van a incluir en el cálculo. Debe ser un valor constante.
includeCurrent	(Opcional) Valor lógico que especifica si se va a incluir o no la fila actual en el rango. El valor predeterminado es Verdadero.
ejes	(Opcional) Referencia del eje, dirección a lo largo del cual se calculará la media móvil.
espacios en blanco	(Opcional) Enumeración que define cómo controlar los valores en blanco en la ordenación. Los valores admitidos son: <ul style="list-style-type: none">• DEFAULT (el valor predeterminado), donde el comportamiento de los valores numéricos es que los valores en blanco se ordenan entre cero y valores negativos. El comportamiento de las cadenas es que los valores en blanco se ordenan delante de todas las cadenas, incluidas las cadenas vacías.• FIRST, los espacios en blanco siempre se ordenan al principio, independientemente del criterio de ordenación ascendente o descendente.

Término	Definición
	<ul style="list-style-type: none"> • LAST, los espacios en blanco siempre se ordenan al final, independientemente del criterio de ordenación ascendente o descendente.
reset	<p>(Opcional) Indica si el cálculo se restablece y en qué nivel de la jerarquía de columnas de la forma visual. Los valores aceptados son: NONE, LOWESTPARENT, HIGHESTPARENT o un entero. El comportamiento depende del signo entero:</p> <ul style="list-style-type: none"> : si se omite cero, el cálculo no se restablece. Equivalente a NONE. : si es positivo, el entero identifica la columna a partir de la más alta, independiente del intervalo de agregación. HIGHESTPARENT es equivalente a 1. : si es negativo, el entero identifica la columna a partir del más bajo, en relación con el intervalo de agregación actual. LOWESTPARENT es equivalente a -1.

Valor devuelto

Valor escalar, la media móvil en el elemento actual del eje.

Comentarios

Esta función solo se puede usar en cálculos visuales.

Se pueden omitir los parámetros <includeCurrent>, <axis>, <blanks> y <reset>.

Ejemplo 1

Dada una tabla que resume las ventas totales de cada categoría de producto y mes natural, la siguiente consulta DAX agrega una columna con el promedio de ventas totales de esa categoría en los últimos 6 meses.

DAX

```
AvgSalesLast6Months = MOVINGAVERAGE([SalesAmount], 6, Rows)
```

En la siguiente captura de pantalla se muestra la matriz visual y la expresión de cálculo visual:

1 AvgSalesLast6Months = MOVINGAVERAGE([SalesAmount], 6, Rows)

CalendarYear	MonthNumberOfYear	EnglishProductCategoryName	Accessories		Bikes		Clothing		Total	
			SalesAmount	AvgSalesLast6 Months	SalesAmount	AvgSalesLast6 Months	SalesAmount	AvgSalesLast6 Months	SalesAmount	AvgSalesLast6 Months
2012	11			\$513,987.7576	499,021.53		\$513,987.7576	499,021.53		
	12			\$595,333.0034	512,488.63		\$595,333.0034	512,488.63		
	Total			\$5,836,345.8175	6,344,417.60		\$5,836,345.8175	6,344,417.60		
2013	1	\$10,545.97	10,545.97	\$766,754.1686	561,259.44	\$4,775.28	4,775.28	\$782,075.4186	563,812.98	
	2	\$36,924.35	23,735.16	\$739,000.87	598,749.66	\$16,696.66	10,735.97	\$792,621.88	610,240.04	
	3	\$54,689.48	34,053.27	\$933,134.99	674,486.47	\$26,341.8	15,937.91	\$1,014,166.27	699,482.06	
	4	\$52,204.71	38,591.13	\$930,705.79	746,486.10	\$25,222.8	18,259.14	\$1,008,133.3	784,386.27	
	5	\$54,600.84	41,793.07	\$1,033,128.45	833,009.55	\$25,598.1	19,726.93	\$1,113,327.39	884,276.21	
	6	\$62,495.09	45,243.41	\$1,512,801.23	985,920.92	\$28,925.62	21,260.04	\$1,604,221.94	1,052,424.37	
	7	\$58,364.86	53,213.22	\$1,342,684.25	1,081,909.26	\$30,661.88	25,574.48	\$1,431,710.99	1,160,696.96	
	8	\$60,752.56	57,184.59	\$1,455,104.25	1,201,259.83	\$29,639.91	27,731.69	\$1,545,496.72	1,286,176.10	
	9	\$57,717.29	57,689.23	\$1,347,106.88	1,270,255.14	\$27,614.56	27,943.81	\$1,432,438.73	1,355,888.18	

Ejemplo 2

Dada la misma tabla, la siguiente consulta DAX agrega una columna con el promedio de ventas totales de esa categoría en los 12 meses anteriores (sin incluir el mes actual):

DAX

AvgSalesPrev12Months = MOVINGAVERAGE([SalesAmount], 12, FALSE, Rows, KEEP)
--

Consulte también

[INDEX](#)

[ORDERBY](#)

[PARTITIONBY](#)

[RUNNINGSUM](#)

[WINDOW](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

OFFSET

Artículo • 21/02/2024

Devuelve una sola fila que se coloca antes o después de la *fila actual* en la misma tabla, con un desplazamiento determinado. Si la fila actual no se puede deducir como una sola fila, se pueden devolver varias filas.

Sintaxis

DAX

```
OFFSET ( <delta>[, <relation> or <axis>][, <orderBy>][, <blanks>][, <partitionBy>][, <matchBy>][, <reset>] )
```

Parámetros

 Expandir tabla

Término	Definición
delta	Número de filas delante (valor negativo) o después (valor positivo) de la fila actual desde la que se van a obtener los datos. Puede ser cualquier expresión DAX que devuelva un valor escalar.
relation	(Opcional) Expresión de tabla desde la que se devuelve la fila de salida. Si se especifica, todas las columnas de <partitionBy> deben proceder de ella o de una tabla relacionada. Si se omite: <ul style="list-style-type: none">- <orderBy> debe especificarse explícitamente.- Todas las expresiones de <orderBy> y <partitionBy> deben tener el nombre completo de la columna y proceder de una sola tabla.- Toma el valor predeterminado ALLSELECTED() de todas las columnas de <orderBy> y <partitionBy>.
ejes	(Opcional) Un eje en la forma visual. Solo está disponible en cálculos visuales y reemplaza la <relación>.
orderBy	(Opcional) Cláusula ORDERBY() que contiene las expresiones que definen cómo se ordena cada partición. Si se omite: <ul style="list-style-type: none">- <relation> debe especificarse explícitamente.- De forma predeterminada, ordena por cada columna de <relation> que aún no se haya especificado en <partitionBy>.

Término	Definición
espacios en blanco	(Opcional) Enumeración que define cómo controlar los valores en blanco en la ordenación. Este parámetro se reserva para uso futuro. Actualmente, el único valor admitido es DEFAULT, donde el comportamiento de los valores numéricos es que los valores en blanco se ordenan entre cero y los valores negativos. El comportamiento de las cadenas es que los valores en blanco se ordenan delante de todas las cadenas, incluidas las cadenas vacías.
partitionBy	(Opcional) Cláusula PARTITIONBY() que contiene las columnas que definen cómo se partitiona <relation>. Si se omite, <relation> se trata como una sola partición.
matchBy	(Opcional) Una cláusula MATCHBY() que contiene las columnas que definen cómo hacer coincidir los datos e identificar la fila actual.
reset	(Opcional) Solo está disponible en cálculos visuales. Indica si el cálculo se restablece y en qué nivel de la jerarquía de columnas de la forma visual. Los valores aceptados son: NONE, LOWESTPARENT, HIGHESTPARENT o un entero. El comportamiento depende del signo entero: : si se omite cero, el cálculo no se restablece. Equivalente a NONE. : si es positivo, el entero identifica la columna a partir de la más alta, independiente del intervalo de agregación. HIGHESTPARENT es equivalente a 1. : si es negativo, el entero identifica la columna a partir del más bajo, en relación con el intervalo de agregación actual. LOWESTPARENT es equivalente a -1.

Valor devuelto

Una o varias filas de <relation>.

Observaciones

Excepto las columnas agregadas por las funciones de tabla DAX, cada columna de <relación>, cuando <matchBy> no está presente o cada columna de <matchBy> y <partitionBy>, cuando <matchBy> está presente, debe tener un valor externo correspondiente para ayudar a definir la fila actual en la que se va a operar, con el siguiente comportamiento:

- Si hay exactamente una columna externa correspondiente, se utiliza su valor.
- Si no hay ninguna columna externa correspondiente:
 - OFFSET determina primero todas las columnas que no tienen ninguna columna externa correspondiente.
 - Para cada combinación de valores de estas columnas en el contexto primario de OFFSET, se evalúa OFFSET y se devuelve una fila.

- La salida final de OFFSET es una unión de estas filas.
- Si hay más de una columna externa correspondiente, se devuelve un error.

Si todas las columnas de <relation> se agregaron mediante funciones de tabla DAX, se devuelve un error.

Si <matchBy> está presente, OFFSET intentará usar las columnas <matchBy> y <partitionBy> para identificar la fila.

Si <matchBy> no está presente y las columnas especificadas en <orderBy> y <partitionBy> no pueden identificar de forma única todas las filas de <relation>:

- OFFSET intenta buscar el menor número de columnas adicionales necesarias para identificar de forma única cada fila.
- Si se encuentran estas columnas, OFFSET anexa automáticamente estas nuevas columnas a <orderBy> y cada partición se ordena usando este nuevo conjunto de columnas OrderBy.
- Si no se encuentran estas columnas, se devuelve un error.

Se devuelve una tabla vacía si:

- El valor externo correspondiente de una columna OrderBy o PartitionBy no existe en <relation>.
- El valor <delta> provoca un desplazamiento a una fila que no existe en la partición.

Si se utiliza OFFSET en una columna calculada definida en la misma tabla que <relation> y se omite <orderBy>, se devuelve un error.

<reset> solo se puede usar en cálculos visuales y no se puede usar en combinación con <orderBy> o <partitionBy>. Si <reset> está presente, se puede especificar el <eje>, pero no se puede especificar la <relación>.

Ejemplo 1: columna calculada

La consulta DAX siguiente:

```
DAX

DEFINE
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProductCategory[EnglishProductName],
        DimDate[CalendarYear],
        "CurrentYearSales", SUM(FactInternetSales[SalesAmount])
    )
EVALUATE
```

```

ADDCOLUMNS (
    vRelation,
    "PreviousYearSales",
    SELECTCOLUMNS(
        OFFSET (
            -1,
            vRelation,
            ORDERBY([CalendarYear]),
            PARTITIONBY([EnglishProductCategoryName])
        ),
        [CurrentYearSales]
    )
)

```

Devuelve una tabla que resume las ventas totales de cada categoría de producto y año natural, así como las ventas totales de esa categoría en el año anterior.

Ejemplo 2: medida

La consulta DAX siguiente:

```

DAX

DEFINE
MEASURE DimProduct[CurrentYearSales] = SUM(FactInternetSales[SalesAmount])
MEASURE DimProduct[PreviousYearSales] =
CALCULATE(SUM(FactInternetSales[SalesAmount]), OFFSET(-1, ,
ORDERBY(DimDate[CalendarYear])))
EVALUATE
SUMMARIZECOLUMNS (
    DimDate[CalendarYear],
    "CurrentYearSales", DimProduct[CurrentYearSales],
    "PreviousYearSales", DimProduct[PreviousYearSales]
)

```

Utiliza OFFSET() en una medida para devolver una tabla que resume las ventas totales de cada año natural y las ventas totales del año anterior.

Ejemplo 3: columna calculada

La consulta DAX siguiente:

```

DAX

EVALUATE
ADDCOLUMNS (

```

```

FactInternetSales,
"Previous Sales Amount",
SELECTCOLUMNS (
    OFFSET (
        -1,
        FactInternetSales,
        ORDERBY ( FactInternetSales[SalesAmount], DESC ),
        PARTITIONBY ( FactInternetSales[ProductKey] ),
        MATCHBY( FactInternetSales[SalesOrderNumber],
        FactInternetSales[SalesOrderLineNumber] )
    ),
    FactInternetSales[SalesAmount]
)
)

```

Devuelve la tabla FactInternetSales con la adición de una columna, que indica, para cada venta, el importe de la venta anterior, del mismo producto, en orden descendente según el importe de ventas, con la venta actual identificada mediante su SalesOrderNumber y SalesOrderLineNumber. Sin MATCHBY, la consulta devolvería un error porque no hay columnas de clave en la tabla FactInternetSales.

Ejemplo 4: cálculo visual

La siguiente consulta DAX de cálculo visual:

DAX

```
SalesRelativeToPreviousMonth = [SalesAmount] - CALCULATE(SUM([SalesAmount]),
OFFSET(-1, ROWS, HIGHESTPARENT))
```

Devuelve la diferencia de ventas totales entre cada mes y el anterior dentro del mismo año.

En la siguiente captura de pantalla se muestra la matriz visual y la expresión de cálculo visual:

X ✓ fx 1 SalesRelativeToPreviousMonth = [SalesAmount] - CALCULATE(SUM([SalesAmount]), OFFSET(-1, ROWS, HIGHESTPARENT))

CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	SalesRelativeToPreviousMonth
2011	1	1	\$323,743.2056	323,743.21
		2	\$437,372.4476	113,629.24
		3	\$514,918.2284	77,545.78
	2	Total	\$1,276,033.8816	1,276,033.88
		4	\$523,950.0258	9,031.80
		5	\$539,469.0004	15,518.97
		6	\$647,524.214	108,055.21

Contenido relacionado

[INDEX](#)

[ORDERBY](#)

[PARTITIONBY](#)

[MATCHBY](#)

[WINDOW](#)

[RANK](#)

[ROWNUMBER](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ORDERBY

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Define las expresiones que determinan el criterio de ordenación dentro de cada una de las particiones de una función window.

Sintaxis

DAX

```
ORDERBY ( [<orderBy_expression>[, <order>[, <orderBy_expression>[, <order>]] ...]] )
```

Parámetros

Expandir tabla

Término	Definición
orderBy_expression	(Opcional) Cualquier expresión escalar que se usará para ordenar los datos dentro de cada una de las particiones de una función window.
orden	(Opcional) Valor de dos partes del formulario "<OrderDirection> [<BlankHandling>]". <OrderDirection> especifica cómo ordenar los valores <orderBy_expression> (es decir, orden ascendente o descendente). Los valores válidos son: DESC. Valor alternativo: 0(cero)/FALSE . Ordena de forma descendente los valores de <orderBy_expression>. ASC. Valor alternativo: 1/TRUE . Ordena de forma ascendente los valores de <orderBy_expression>. Este es el valor predeterminado si se omite <order>. El elemento <BlankHandling> es opcional. Especifica cómo se ordenan los espacios en blanco. Los valores válidos son: BLANKS DEFAULT. Este es el valor predeterminado. El comportamiento de los valores numéricos es que los valores en blanco se ordenan entre cero y valores negativos. El comportamiento de las cadenas es que los valores en blanco se ordenan delante de todas las cadenas, incluidas las cadenas vacías. BLANKS FIRST. Los espacios en blanco siempre se ordenan al principio, independientemente del criterio de ordenación ascendente o descendente.

Término	Definición
	BLANKS LAST. Los espacios en blanco siempre se ordenan al final, independientemente del criterio de ordenación ascendente o descendente.

Valor devuelto

Esta función no devuelve ningún valor.

Observaciones

Esta función solo se puede usar en una expresión de una función WINDOW.

Ejemplo

Consulte [OFFSET](#).

Contenido relacionado

[Descripción de las funciones ORDERBY, PARTITIONBY y MATCHBY](#)

[INDEX](#)

[OFFSET](#)

[PARTITIONBY](#)

[MATCHBY](#)

[WINDOW](#)

[RANK](#)

[ROWNUMBER](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PARTITIONBY

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Define las columnas que se usan para particionar el parámetro <relation> de una función window.

Sintaxis

DAX

```
PARTITIONBY ( [<partitionBy_columnName>[, partitionBy_columnName [, ...]]] )
```

Parámetros

Expandir tabla

Término	Definición
partitionBy_columnName	(Opcional) Nombre de una columna existente que se va a usar para particionar el parámetro <relation> de una función window. RELATED() también se puede usar para hacer referencia a una columna de una tabla relacionada con <relation>.

Valor devuelto

Esta función no devuelve ningún valor.

Observaciones

Esta función solo se puede usar en una expresión de una función WINDOW.

Ejemplo

Consulte [OFFSET](#).

Contenido relacionado

[Descripción de las funciones ORDERBY, PARTITIONBY y MATCHBY](#)

[INDEX](#)

[OFFSET](#)

[ORDERBY](#)

[MATCHBY](#)

[WINDOW](#)

[RANK](#)

[ROWNUMBER](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

RANGE

Artículo • 21/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve un intervalo de filas dentro del eje especificado, en relación con la fila actual. Este intervalo se compone de las últimas filas <step> antes de la actual o de las primeras filas <step> después de la actual.

Sintaxis

DAX

```
RANGE ( <step>[, <includeCurrent>][, <axis>][, <blanks>][, <reset>] )
```

Parámetros

Expandir tabla

Término	Definición
paso	Número de filas delante (valor negativo) o después (valor positivo) de la fila actual para incluir en el rango. Debe ser un valor constante. - Si es negativo, la ventana contendrá las últimas filas -step antes de la fila actual. - De lo contrario, la ventana contendrá las primeras filas step después de la fila actual.
includeCurrent	(Opcional) Valor lógico que especifica si se va a incluir o no la fila actual en el rango. El valor predeterminado es Verdadero.
ejes	(Opcional) Referencia del eje, dirección a lo largo del cual se creará el intervalo.
espacios en blanco	(Opcional) Enumeración que define cómo controlar los valores en blanco en la ordenación. Los valores admitidos son: <ul style="list-style-type: none">• DEFAULT (el valor predeterminado), donde el comportamiento de los valores numéricos es que los valores en blanco se ordenan entre cero y valores negativos. El comportamiento de las cadenas es que los valores en blanco se ordenan delante de todas las cadenas, incluidas las cadenas vacías.• FIRST, los espacios en blanco siempre se ordenan al principio, independientemente del criterio de ordenación ascendente o descendente.

Término	Definición
	<ul style="list-style-type: none"> • LAST, los espacios en blanco siempre se ordenan al final, independientemente del criterio de ordenación ascendente o descendente.
reset	<p>(Opcional) Indica si el cálculo se restablece y en qué nivel de la jerarquía de columnas de la forma visual. Los valores aceptados son: NONE, LOWESTPARENT, HIGHESTPARENT o un entero. El comportamiento depende del signo entero:</p> <ul style="list-style-type: none"> : si se omite cero, el cálculo no se restablece. Equivalente a NONE. : si es positivo, el entero identifica la columna a partir de la más alta, independiente del intervalo de agregación. HIGHESTPARENT es equivalente a 1. : si es negativo, el entero identifica la columna a partir del más bajo, en relación con el intervalo de agregación actual. LOWESTPARENT es equivalente a -1.

Valor devuelto

Intervalo de filas de datos.

Comentarios

Esta función solo se puede usar en cálculos visuales.

Se pueden omitir los parámetros <includeCurrent>, <axis>, <blanks> y <reset>.

Ejemplo 1

Dada una tabla que resume las ventas totales de cada categoría de producto y mes, la siguiente consulta DAX agrega una columna con el total de ventas en los últimos 6 meses:

DAX

```
TotalSalesLast6Months = CALCULATE(SUM([SalesAmount]), RANGE(-5, Rows))
```

En la siguiente captura de pantalla se muestra la matriz visual y la expresión de cálculo visual:

1 TotalSalesLast6Months = CALCULATE(SUM([SalesAmount]), RANGE(-5, Rows))										
CalendarYear	EnglishProductName	ProductCategoryName	Accessories	Bikes	Clothing	Total Sales	Total Sales	Total Sales	Total Sales	Total Sales
	MonthNumberOfYear	SalesAmount	TotalSalesLast6 Months	SalesAmount	TotalSalesLast6 Months	SalesAmount	TotalSalesLast6 Months	SalesAmount	TotalSalesLast6 Months	TotalSalesLast6 Months
2012	11			\$513,987.7576	2,994,129.18			\$513,987.7576	2,994,129.18	
	12			\$595,333.0034	3,074,931.78			\$595,333.0034	3,074,931.78	
	Total			\$5,836,345.8175	12,688,835.20			\$5,836,345.8175	12,688,835.20	
2013	1	\$10,545.97	10,545.97	\$766,754.1686	3,367,556.63	\$4,775.28	4,775.28	\$782,075.4186	3,382,877.88	
	2	\$36,924.35	47,470.32	\$739,000.87	3,592,497.96	\$16,696.66	21,471.94	\$792,621.88	3,661,440.22	
	3	\$54,689.48	102,159.80	\$933,134.99	4,046,918.83	\$26,341.8	47,813.74	\$1,014,166.27	4,196,892.37	
	4	\$52,204.71	154,364.51	\$930,705.79	4,478,916.58	\$25,222.8	73,036.54	\$1,008,133.3	4,706,317.63	
	5	\$54,600.84	208,965.35	\$1,033,128.45	4,998,057.27	\$25,598.1	98,634.64	\$1,113,327.39	5,305,657.26	
	6	\$62,495.09	271,460.44	\$1,512,801.23	5,915,525.50	\$28,925.62	127,560.26	\$1,604,221.94	6,314,546.20	
	7	\$58,364.86	319,279.33	\$1,342,684.25	6,491,455.58	\$30,661.88	153,446.86	\$1,431,710.99	6,964,181.77	
	8	\$60,752.56	343,107.54	\$1,455,104.25	7,207,558.96	\$29,639.91	166,390.11	\$1,545,496.72	7,717,056.61	
	9	\$57,717.29	346,135.35	\$1,347,106.88	7,621,530.85	\$27,614.56	167,662.87	\$1,432,438.73	8,135,329.07	

Ejemplo 2

Dada la misma tabla, la siguiente consulta DAX agrega una columna con las ventas totales en los siguientes 12 meses, sin incluir el mes actual:

DAX

TotalSalesFollowingYear = <code>CALCULATE(SUM([SalesAmount]), RANGE(12, FALSE, Rows, KEEP))</code>
--

Consulte también

[INDEX](#)

[ORDERBY](#)

[PARTITIONBY](#)

[WINDOW](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

RANK

Artículo • 21/02/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve la clasificación del contexto actual dentro de la partición especificada ordenada según lo especificado. Si no se encuentra una coincidencia, la clasificación está en blanco.

Sintaxis

DAX

```
RANK ( [<ties>][, <relation> or <axis>][, <orderBy>][, <blanks>][, <partitionBy>][, <matchBy>][, <reset>] )
```

Parámetros

[] Expandir tabla

Término	Definición
ties	(Opcional) Define cómo controlar la clasificación cuando se vinculan dos o más filas. Si se especifica, el valor admitido es DENSE o SKIP. Si se omite: el valor predeterminado es SKIP.
relation	(Opcional) Expresión de tabla desde la que se devuelve la fila de salida. Si se especifica, todas las columnas de <orderBy> y <partitionBy> deben proceder de ella. Si se omite: <ul style="list-style-type: none">- <orderBy> debe especificarse explícitamente.- Todas las columnas de <orderBy> y <partitionBy> deben tener el nombre completo y proceder de una sola tabla.- Toma el valor predeterminado ALLSELECTED() de todas las columnas de <orderBy> y <partitionBy>.
ejes	(Opcional) Un eje en la forma visual. Solo está disponible en cálculos visuales y reemplaza la <relación>.
orderBy	(Opcional) Cláusula ORDERBY() que contiene las columnas que definen cómo se ordena cada partición. Si se omite:

Término	Definición
	<ul style="list-style-type: none"> - <relation> debe especificarse explícitamente. - De forma predeterminada, ordena por cada columna de <relation> que aún no se haya especificado en <partitionBy>.
espacios en blanco	<p>(Opcional) Enumeración que define cómo controlar los valores en blanco en la ordenación.</p> <p>Los valores admitidos son:</p> <ul style="list-style-type: none"> • KEEP (el valor predeterminado), donde el comportamiento de los valores numéricos es que los valores en blanco se ordenan entre cero y valores negativos. El comportamiento de las cadenas es que los valores en blanco se ordenan delante de todas las cadenas, incluidas las cadenas vacías. • FIRST, los espacios en blanco siempre se ordenan al principio, independientemente del criterio de ordenación ascendente o descendente. • LAST, los espacios en blanco siempre se ordenan al final, independientemente del criterio de ordenación ascendente o descendente. <p>Tenga en cuenta que cuando se especifican parámetros <blanks> y espacios en blanco en la función ORDERBY() en una expresión individual, los <espacios en blanco> de la expresión orderBy individuales tienen prioridad para la expresión orderBy pertinente y las expresiones orderBy sin que se especifiquen <espacios en blanco> respetarán el parámetro <blanks> en la función primaria Window.</p>
partitionBy	<p>(Opcional) Cláusula PARTITIONBY() que contiene las columnas que definen cómo se partitiona <relation>.</p> <p>Si se omite, <relation> se trata como una sola partición.</p>
matchBy	(Opcional) Una cláusula MATCHBY() que contiene las columnas que definen cómo hacer coincidir los datos e identificar la fila actual.
reset	<p>(Opcional) Solo está disponible en cálculos visuales. Indique si el cálculo se restablece y en qué nivel de la jerarquía de columnas de la forma visual. Los valores aceptados son: NONE, LOWESTPARENT, HIGHESTPARENT o un entero. El comportamiento depende del signo entero:</p> <ul style="list-style-type: none"> : si se omite cero, el cálculo no se restablece. Equivalente a NONE. : si es positivo, el entero identifica la columna a partir de la más alta, independiente del intervalo de agregación. HIGHESTPARENT es equivalente a 1. : si es negativo, el entero identifica la columna a partir del más bajo, en relación con el intervalo de agregación actual. LOWESTPARENT es equivalente a -1.

Valor devuelto

Número de clasificación del contexto actual.

Observaciones

- Cada columna de <orderBy>, <partitionBy> y <matchBy> debe tener un valor externo correspondiente para ayudar a definir la fila actual en la que operar, con el siguiente comportamiento:
 - Si hay exactamente una columna externa correspondiente, se utiliza su valor.
 - Si no hay ninguna columna externa correspondiente:
 - RANK determina primero todas las columnas de <orderBy>, <partitionBy> y <matchBy> que no tienen ninguna columna externa correspondiente.
 - Para cada combinación de valores de estas columnas existentes en el contexto primario de RANK, se evalúa RANK y se devuelve una fila.
 - La salida final de RANK es un número de clasificación.
- Si <matchBy> está presente, RANK intentará usar columnas en <matchBy> y <partitionBy> para identificar la fila actual.
- Si las columnas especificadas en <orderBy> y <partitionBy> no pueden identificar de forma única cada fila en <relation>, dos o más filas pueden tener la misma clasificación y el parámetro ties determinará la clasificación.
- RANK devuelve un valor en blanco para el total de filas. Se recomienda probar exhaustivamente la expresión.
- RANK no se compara con RANKX como SUM comparado con SUMX.
- <reset> solo se puede usar en cálculos visuales y no se puede usar en combinación con <orderBy> o <partitionBy>. Si <reset> está presente, se puede especificar el <eje>, pero no se puede especificar la <relación>.

Ejemplo 1: columna calculada

La consulta DAX siguiente:

```
DAX

EVALUATE
ADDCOLUMNS(
    'DimGeography',
    "Rank",
    RANK(
        DENSE,
        'DimGeography',
        ORDERBY(
            'DimGeography'[StateProvinceName], desc,
            'DimGeography'[City], asc),
        LAST,
        PARTITIONBY(
            'DimGeography'[EnglishCountryRegionName])))
```

```
ORDER BY [EnglishCountryRegionName] asc, [StateProvinceName] desc, [City]  
asc
```

Devuelve una tabla que clasifica cada geografía con el mismo englishCountryRegionName, por stateProvinceName y City. Los valores de columna <orderBy> en blanco se ordenan al final.

Ejemplo 2: cálculo visual

Las siguientes consultas DAX de cálculo visual:

DAX

```
SalesRankWithinYear = RANK(DENSE, ORDERBY([SalesAmount], DESC),  
PARTITIONBY([CalendarYear]))
```

```
SalesRankAllHistory = RANK(DENSE, ORDERBY([SalesAmount], DESC))
```

Cree dos columnas que clasifiquen cada mes por el total de ventas, tanto dentro de cada año como por todo el historial.

En la siguiente captura de pantalla se muestra la matriz visual y la primera expresión de cálculo visual:

X ✓ fx 1 SalesRankWithinYear = RANK(DENSE, ORDERBY([SalesAmount], DESC), PARTITIONBY([CalendarYear]))

CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	SalesRankWithinYear	SalesRankAllHistory
2011	1	1	\$323,743.2056	12.00	37.00
		2	\$437,372.4476	11.00	34.00
		3	\$514,918.2284	10.00	25.00
		Total	\$1,276,033.8816	4.00	11.00
	2	4	\$523,950.0258	9.00	24.00
		5	\$539,469.0004	8.00	23.00
		6	\$647,524.214	4.00	16.00
		Total	\$1,710,943.2402	3.00	7.00
	3	7	\$648,343.8132	3.00	15.00
		8	\$599,874.2732	7.00	20.00
		9	\$609,121.865	6.00	19.00
		Total	\$1,857,339.9514	2.00	6.00
	4	10	\$655,170.3294	2.00	14.00
		11	\$716,748.7178	1.00	13.00
		12	\$636,253.2642	5.00	17.00
		Total	\$2,008,172.3114	1.00	5.00
Total			\$6,852,489.3846	1.00	2.00

Contenido relacionado

[INDEX](#)

[ORDERBY](#)

[PARTITIONBY](#)

[WINDOW](#)

[ROWNUMBER](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

REMOVEFILTERS

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Borra los filtros de las tablas o columnas especificadas.

Sintaxis

DAX

```
REMOVEFILTERS([<table> | <column>[, <column>[, <column>[,...]]]])
```

Parámetros

[] Expandir tabla

Término	Definición
tabla	Tabla en la que se van a borrar los filtros.
columna	Columna en la que se van a borrar los filtros.

Valor devuelto

N/D Vea Notas.

Observaciones

- REMOVEFILTERS solo se puede usar para borrar filtros, pero no para devolver una tabla.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

Consulta DAX

DAX

```
DEFINE  
MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])  
MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales],  
CALCULATE([TotalSales],REMOVEFILTERS()))
```

```

EVALUATE
    SUMMARIZECOLUMNS(
        ROLLUPADDISSTOTAL(DimProductCategory[EnglishProductName], 
        "IsGrandTotal"),
        "%Sales", [%Sales]
    )
ORDER BY
    [IsGrandTotal] DESC, [TotalSales] DESC

```

Devuelve

Expandir tabla

DimProductCategory[EnglishProductName]	[IsGrandTotal]	[TotalSales]	[%Sales]
Fila1	Verdadero	29 358 677,2207	1
Bicicletas	Falso	28 318 144,6507	0,964557920570538
Accessories	False	700 759,96	0,023868921434441
Clothing	False	339 772,61	0,0115731579950215

Ejemplo 2

Consulta DAX

```

DAX

DEFINE
    MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])
    MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales],
    CALCULATE([TotalSales], REMOVEFILTERS(DimProductSubcategory[EnglishProductSubcategoryName])))

EVALUATE
    SUMMARIZECOLUMNS(
        DimProductCategory[EnglishProductName],
        DimProductSubcategory[EnglishProductSubcategoryName],
        "%Sales", [%Sales]
    )
ORDER BY
    DimProductCategory[EnglishProductName] ASC,
    DimProductSubcategory[EnglishProductSubcategoryName] ASC

```

Devuelve

Expandir tabla

DimProductCategory [EnglishProductName]	DimProductSubcategory [EnglishProductSubcategoryName]	[TotalSales]	[%Sales]
Accessories	Marcos de bicicletas	39 360	0,05616759
Accesorios	Soportes de bicicletas	39 591	0,05649723
Accesorios	Bidones y soportes	56 798,19	0,08105228
Accesorios	Limpiadores	7 218,6	0,0103011
Accesorios	Guardabarros	46 619,58	0,06652717
Accesorios	Cascos	225 335,6	0,3215589
Accesorios	Paquetes de hidratación	40 307,67	0,05751994
Accesorios	Cubiertas y tubulares	245 529,32	0,35037578
Bikes	Bicicletas de montaña	9 952 759,564	0,35146228
Bicicletas	Bicicletas de carretera	14 520 584,04	0,51276608
Bicicletas	Bicicletas de paseo	3 844 801,05	0,13577164
Ropa	Límites	19 688,1	0,05794493
Ropa	Guantes	35 020,7	0,10307099
Ropa	Maillots	172 950,68	0,5090189
Ropa	Shorts	71 319,81	0,20990453
Ropa	Calcetines	5106,32	0,01502864
Ropa	Chalecos	35 687	0,10503201

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ROWNUMBER

Artículo • 21/02/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve la clasificación única del contexto actual dentro de la partición especificada ordenada según lo especificado. Si no se encuentra una coincidencia, rownumber está en blanco.

Sintaxis

DAX

```
ROWNUMBER ( [<relation> or <axis>][, <orderBy>][, <blanks>][, <partitionBy>]
[, <matchBy>][, <reset>] )
```

Parámetros

Expandir tabla

Término	Definición
relation	(Opcional) Expresión de tabla desde la que se devuelve la fila de salida. Si se especifica, todas las columnas de <orderBy> y <partitionBy> deben proceder de ella. Si se omite: <ul style="list-style-type: none">- <orderBy> debe especificarse explícitamente.- Todas las columnas de <orderBy> y <partitionBy> deben tener el nombre completo y proceder de una sola tabla.- Toma el valor predeterminado ALLSELECTED() de todas las columnas de <orderBy> y <partitionBy>.
ejes	(Opcional) Un eje en la forma visual. Solo está disponible en cálculos visuales y reemplaza la <relación>.
orderBy	(Opcional) Cláusula ORDERBY() que contiene las columnas que definen cómo se ordena cada partición. Si se omite: <ul style="list-style-type: none">- <relation> debe especificarse explícitamente.- De forma predeterminada, ordena por cada columna de <relation> que aún no se haya especificado en <partitionBy>.
espacios en blanco	(Opcional) Enumeración que define cómo controlar los valores en blanco en la ordenación.

Término	Definición
	<p>Los valores admitidos son:</p> <ul style="list-style-type: none"> • DEFAULT (el valor predeterminado), donde el comportamiento de los valores numéricos es que los valores en blanco se ordenan entre cero y valores negativos. El comportamiento de las cadenas es que los valores en blanco se ordenan delante de todas las cadenas, incluidas las cadenas vacías. • FIRST, los espacios en blanco siempre se ordenan al principio, independientemente del criterio de ordenación ascendente o descendente. • LAST, los espacios en blanco siempre se ordenan al final, independientemente del criterio de ordenación ascendente o descendente. <p>Tenga en cuenta que cuando se especifican parámetros <blanks> y espacios en blanco en la función ORDERBY() en una expresión individual, los <espacios en blanco> de la expresión orderBy individuales tienen prioridad para la expresión orderBy pertinente y las expresiones orderBy sin que se especifiquen <espacios en blanco> respetarán el parámetro <blanks> en la función primaria Window.</p>
partitionBy	(Opcional) Cláusula PARTITIONBY() que contiene las columnas que definen cómo se partitiona <relation>. Si se omite, <relation> se trata como una sola partición.
matchBy	(Opcional) Una cláusula MATCHBY() que contiene las columnas que definen cómo hacer coincidir los datos e identificar la fila actual.
reset	(Opcional) Solo está disponible en cálculos visuales. Indique si el cálculo se restablece y en qué nivel de la jerarquía de columnas de la forma visual. Los valores aceptados son: NONE, LOWESTPARENT, HIGHESTPARENT o un entero. El comportamiento depende del signo entero: <ul style="list-style-type: none"> : si se omite cero, el cálculo no se restablece. Equivalente a NONE. : si es positivo, el entero identifica la columna a partir de la más alta, independiente del intervalo de agregación. HIGHESTPARENT es equivalente a 1. : si es negativo, el entero identifica la columna a partir del más bajo, en relación con el intervalo de agregación actual. LOWESTPARENT es equivalente a -1.

Valor devuelto

Número de rownumber del contexto actual.

Observaciones

Cada columna de <orderBy>, <partitionBy> y <matchBy> debe tener un valor externo correspondiente para ayudar a definir la fila actual en la que operar, con el siguiente comportamiento:

- Si hay exactamente una columna externa correspondiente, se utiliza su valor.
- Si no hay ninguna columna externa correspondiente:
 - ROWNUMBER determina primero todas las columnas de <orderBy>, <partitionBy> y <matchBy> que no tienen ninguna columna externa correspondiente.
 - Para cada combinación de valores de estas columnas en el contexto primario de ROWNUMBER, se evalúa ROWNUMBER y se devuelve una fila.
 - La salida final de ROWNUMBER es una unión de estas filas.
- Si hay más de una columna externa correspondiente, se devuelve un error.

Si <matchBy> está presente, ROWNUMBER intentará usar columnas en <matchBy> y <partitionBy> para identificar la fila actual.

Si las columnas especificadas en <orderBy> y <partitionBy> no pueden identificar de forma única todas las filas de <relation>:

- ROWNUMBER intenta buscar el menor número de columnas adicionales necesarias para identificar de forma única cada fila.
- Si se encuentran estas columnas, ROWNUMBER
 - intentará buscar el menor número de columnas adicionales necesarias para identificar de forma única cada fila.
 - Anexe automáticamente estas nuevas columnas a la cláusula <orderBy>.
 - Ordene cada partición con este nuevo conjunto de columnas orderBy.
- Si no se encuentran estas columnas y la función detecta un empate en tiempo de ejecución, se devuelve un error.

<reset> solo se puede usar en cálculos visuales y no se puede usar en combinación con <orderBy> o <partitionBy>. Si <reset> está presente, se puede especificar el <eje>, pero no se puede especificar la <relación>.

Ejemplo 1: columna calculada

La consulta DAX siguiente:

```
DAX

EVALUATE
ADDCOLUMNS(
    'DimGeography',
    "UniqueRank",
    ROWNUMBER(
        'DimGeography',
        ORDERBY(
            'DimGeography'[StateProvinceName], desc,
            'DimGeography'[City], asc),
        "UniqueRank"
    )
)
```

```

PARTITIONBY(
    'DimGeography'[EnglishCountryRegionName]))
ORDER BY [EnglishCountryRegionName] asc, [StateProvinceName] desc, [City]
asc

```

Devuelve una tabla que clasifica exclusivamente cada geografía con el mismo englishCountryRegionName, por stateProvinceName y City.

Ejemplo 2: cálculo visual

Las siguientes consultas DAX de cálculo visual:

DAX

```
SalesRankWithinYear = ROWNUMBER(ORDERBY([SalesAmount], DESC),
PARTITIONBY([CalendarYear]))
```

```
SalesRankAllHistory = ROWNUMBER(ORDERBY([SalesAmount], DESC))
```

Cree dos columnas que clasifiquen de forma única cada mes por el total de ventas, tanto dentro de cada año como por todo el historial.

En la siguiente captura de pantalla se muestra la matriz visual y la primera expresión de cálculo visual:

X ✓ fx 1 SalesRankWithinYear = ROWNUMBER(ORDERBY([SalesAmount], DESC), PARTITIONBY([CalendarYear]))						
CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	SalesRankWithinYear	SalesRankAllHistory	
2011	1	1	\$323,743.2056	12.00	37.00	
		2	\$437,372.4476	11.00	34.00	
		3	\$514,918.2284	10.00	25.00	
	2	Total	\$1,276,033.8816	4.00	11.00	
		4	\$523,950.0258	9.00	24.00	
		5	\$539,469.0004	8.00	23.00	
		6	\$647,524.214	4.00	16.00	
	3	Total	\$1,710,943.2402	3.00	7.00	
		7	\$648,343.8132	3.00	15.00	
		8	\$599,874.2732	7.00	20.00	
		9	\$609,121.865	6.00	19.00	
	4	Total	\$1,857,339.9514	2.00	6.00	
		10	\$655,170.3294	2.00	14.00	
		11	\$716,748.7178	1.00	13.00	
		12	\$636,253.2642	5.00	17.00	
Total		Total	\$2,008,172.3114	1.00	5.00	
			\$6,852,489.3846	1.00	2.00	

Contenido relacionado

INDEX

ORDERBY

PARTITIONBY

WINDOW

RANK

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

RUNNINGSUM

Artículo • 21/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve una suma en ejecución calculada a lo largo del eje especificado de la matriz visual. Es decir, la suma de la columna especificada calculada sobre todos los elementos hasta el elemento actual de un eje.

Sintaxis

DAX

```
RUNNINGSUM ( <column>[, <axis>][, <blanks>][, <reset>] )
```

Parámetros

Expandir tabla

Término	Definición
columna	Columna que proporciona el valor de cada elemento.
ejes	(Opcional) Referencia del eje, dirección a lo largo del cual se calculará la suma en ejecución.
espacios en blanco	(Opcional) Enumeración que define cómo controlar los valores en blanco en la ordenación. Los valores admitidos son: <ul style="list-style-type: none">• DEFAULT (el valor predeterminado), donde el comportamiento de los valores numéricos es que los valores en blanco se ordenan entre cero y valores negativos. El comportamiento de las cadenas es que los valores en blanco se ordenan delante de todas las cadenas, incluidas las cadenas vacías.• FIRST, los espacios en blanco siempre se ordenan al principio, independientemente del criterio de ordenación ascendente o descendente.• LAST, los espacios en blanco siempre se ordenan al final, independientemente del criterio de ordenación ascendente o descendente.
reset	(Opcional) Indica si el cálculo se restablece y en qué nivel de la jerarquía de columnas de la forma visual. Los valores aceptados son: NONE, LOWESTPARENT, HIGHESTPARENT o un entero. El comportamiento depende del signo entero: : si se omite cero, el cálculo no se restablece. Equivalente a NONE. : si es positivo, el entero identifica la columna a partir de la más alta, independiente

Término	Definición
	del intervalo de agregación. HIGHESTPARENT es equivalente a 1. : si es negativo, el entero identifica la columna a partir del más bajo, en relación con el intervalo de agregación actual. LOWESTPARENT es equivalente a -1.

Valor devuelto

Un valor escalar, la suma acumulada hasta el elemento actual del eje.

Comentarios

Esta función solo se puede usar en cálculos visuales.

Se pueden omitir los parámetros <eje>, <espacios en blanco> y <restablecer>.

Ejemplo

Dada una tabla que resume las ventas totales de cada categoría de producto y año natural, la siguiente consulta DAX agrega una columna con las ventas totales de esa categoría hasta un año determinado.

DAX
SalesUpToDate = RUNNINGSUM([SalesAmount], Rows)

En la captura de pantalla siguiente se muestra la matriz visual y la expresión de cálculo visual:

1 SalesUpToDate = RUNNINGSUM([SalesAmount], Rows)								
EnglishProductCategoryName	Accessories	Bikes	Clothing	Total	SalesAmount	SalesUpToDate		
CalendarYear	SalesAmount	SalesUpToDate	SalesAmount	SalesUpToDate	SalesAmount	SalesUpToDate		
2011		\$6,852,489.3846	6,852,489.38		\$6,852,489.3846	6,852,489.38		
2012		\$5,836,345.8175	12,688,835.20		\$5,836,345.8175	12,688,835.20		
2013	\$647,604.83	647,604.83	\$15,085,083.2586	27,773,918.46	\$312,059.21	312,059.21	\$16,044,747.2986	28,733,582.50
2014	\$53,155.13	700,759.96	\$544,226.19	28,318,144.65	\$27,713.4	339,772.61	\$625,094.72	29,358,677.22
Total	\$700,759.96	700,759.96	\$28,318,144.6507	28,318,144.65	\$339,772.61	339,772.61	\$29,358,677.2207	29,358,677.22

Consulte también

[INDEX](#)

[MOVINGAVERAGE](#)

[ORDERBY](#)

PARTITIONBY
WINDOW

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SELECTEDVALUE

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve el valor si el contexto de columnName se ha filtrado a un solo valor distinto. De lo contrario, devuelve alternateResult.

Sintaxis

DAX

```
SELECTEDVALUE(<columnName>[, <alternateResult>])
```

Parámetros

[] Expandir tabla

Término	Definición
columnName	Nombre de una columna existente, con la sintaxis de DAX estándar. No puede ser una expresión.
alternateResult	(Opcional) El valor devuelto cuando el contexto de columnName se ha filtrado a cero o a más de un valor distinto. Si no se proporciona, el valor predeterminado es BLANK().

Valor devuelto

El valor cuando el contexto de columnName se ha filtrado a un solo valor distinto. En caso contrario, alternateResult.

Observaciones

- `IF(HASONEVALUE(<columnName>), VALUES(<columnName>), <alternateResult>)` es una expresión equivalente para `SELECTEDVALUE(<columnName>, <alternateResult>)`.
- Para más información sobre los procedimientos recomendados al usar SELECTEDVALUE, consulte [Uso de SELECTEDVALUE en lugar de VALUES en DAX](#).

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La consulta DAX siguiente:

```
DAX

DEFINE
    MEASURE DimProduct[Selected Color] = SELECTEDVALUE(DimProduct[Color], "No
Single Selection")
EVALUATE
SUMMARIZECOLUMNS
    (ROLLUPADDISSTOTAL(DimProduct[Color], "Is Total"),
    "Selected Color", [Selected Color])ORDER BY [Is Total] ASC,
    [Color] ASC
```

Devuelve lo siguiente:

[Expandir tabla](#)

DimProduct[Color]	[Is Total]	[Selected Color]
Negro	FALSE	Negro
Azul	FALSE	Azul
Gris	FALSE	Gris
Multi	FALSE	Multi
N/D	FALSE	NA
Rojo	FALSE	Rojo
Plata	FALSE	Plata
Plata/negro	FALSE	Plata/negro
Blanco	FALSE	Blanco
Amarillo	FALSE	Amarillo
	true	Sin selección única

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

WINDOW

Artículo • 22/02/2024

Devuelve varias filas que se colocan dentro del intervalo especificado.

Sintaxis

DAX

```
WINDOW ( from[, from_type], to[, to_type][, <relation> or <axis>][,
<orderBy>][, <blanks>][, <partitionBy>][, <matchBy>][, <reset>] )
```

Parámetros

 Expandir tabla

Término	Definición
desde	<p>Indica dónde comienza el período. Puede ser cualquier expresión DAX que devuelva un valor escalar.</p> <p>El comportamiento depende del parámetro <from_type>:</p> <ul style="list-style-type: none">- Si <from_type> es REL, es el número de filas que se retrocede (valor negativo) o se avanza (valor positivo) desde la fila actual para obtener la primera fila del período.- Si <from_type> es ABS y <from> es positivo, es la posición donde comienza el período respecto al principio de la partición. La indexación es de base 1. Por ejemplo, 1 significa que el período comienza al principio de la partición. Si <from> es negativo, es la posición donde comienza el período respecto al final de la partición. -1 significa la última fila de la partición.
from_type	Modifica el comportamiento del parámetro <from>. Los valores posibles son ABS (absoluto) y REL (relativo). El valor predeterminado es REL.
to	Es igual que <from>, pero indica el final del período. La última fila se incluye en el período.
to_type	Es igual que <from_type>, pero modifica el comportamiento de <to>.
relation	<p>(Opcional) Expresión de tabla desde la que se devuelve la fila de salida.</p> <p>Si se especifica, todas las columnas de <partitionBy> deben proceder de ella o de una tabla relacionada.</p> <p>Si se omite:</p> <ul style="list-style-type: none">- <orderBy> debe especificarse explícitamente.- Todas las expresiones de <orderBy> y <partitionBy> deben tener el nombre completo de la columna y proceder de una sola tabla.

Término	Definición
	<ul style="list-style-type: none"> - Toma el valor predeterminado ALLSELECTED() de todas las columnas de <orderBy> y <partitionBy>.
ejes	(Opcional) Un eje en la forma visual. Solo está disponible en cálculos visuales y reemplaza la <relación>.
orderBy	<p>(Opcional) Cláusula ORDERBY() que contiene las expresiones que definen cómo se ordena cada partición.</p> <p>Si se omite:</p> <ul style="list-style-type: none"> - <relation> debe especificarse explícitamente. - De forma predeterminada, ordena por cada columna de <relation> que aún no se haya especificado en <partitionBy>.
espacios en blanco	<p>(Opcional) Enumeración que define cómo controlar los valores en blanco en la ordenación.</p> <p>Este parámetro se reserva para uso futuro.</p> <p>Actualmente, el único valor admitido es DEFAULT, donde el comportamiento de los valores numéricos es que los valores en blanco se ordenan entre cero y los valores negativos. El comportamiento de las cadenas es que los valores en blanco se ordenan delante de todas las cadenas, incluidas las cadenas vacías.</p>
partitionBy	(Opcional) Cláusula PARTITIONBY() que contiene las columnas que definen cómo se partitiona <relation>. Si se omite, <relation> se trata como una sola partición.
matchBy	(Opcional) Una cláusula MATCHBY() que contiene las columnas que definen cómo hacer coincidir los datos e identificar la fila actual.
reset	<p>(Opcional) Solo está disponible en cálculos visuales. Indique si el cálculo se restablece y en qué nivel de la jerarquía de columnas de la forma visual. Los valores aceptados son: NONE, LOWESTPARENT, HIGHESTPARENT o un entero. El comportamiento depende del signo entero:</p> <ul style="list-style-type: none"> : si se omite cero, el cálculo no se restablece. Equivalente a NONE. : si es positivo, el entero identifica la columna a partir de la más alta, independiente del intervalo de agregación. HIGHESTPARENT es equivalente a 1. : si es negativo, el entero identifica la columna a partir del más bajo, en relación con el intervalo de agregación actual. LOWESTPARENT es equivalente a -1.

Valor devuelto

Todas las filas del período.

Observaciones

Excepto las columnas agregadas por las funciones de tabla DAX, cada columna de <relación>, cuando <matchBy> no está presente o cada columna de <matchBy> y

<partitionBy>, cuando <matchBy> está presente, debe tener un valor externo correspondiente para ayudar a definir la fila actual en la que se va a operar. Si <from_type> y <to_type> tienen el valor ABS, lo siguiente solo se aplica a las columnas de <partitionBy>:

- Si hay exactamente una columna externa correspondiente, se utiliza su valor.
- Si no hay ninguna columna externa correspondiente:
 - WINDOW determina primero todas las columnas que no tienen ninguna columna externa correspondiente.
 - Para cada combinación de valores de estas columnas en el contexto primario de WINDOW, se evalúa WINDOW y se devuelven las filas correspondientes.
 - La salida final de WINDOW es una unión de estas filas.
- Si hay más de una columna externa correspondiente, se devuelve un error.

Si todas las columnas de <relation> se agregaron mediante funciones de tabla DAX, se devuelve un error.

Si <matchBy> está presente, WINDOW intentará usar las columnas <matchBy> y <partitionBy> para identificar la fila.

Si <matchBy> no está presente y las columnas especificadas en <orderBy> y <partitionBy> no pueden identificar de forma única todas las filas de <relation>:

- WINDOW intenta buscar el menor número de columnas adicionales necesarias para identificar de forma única cada fila.
- Si se encuentran estas columnas, WINDOW anexa automáticamente estas nuevas columnas a <orderBy> y cada partición se ordena usando este nuevo conjunto de columnas OrderBy.
- Si no se encuentran estas columnas, se devuelve un error.

Se devuelve una tabla vacía si:

- El valor externo correspondiente de una columna de <orderBy> o <partitionBy> no existe en <relation>.
- Todo el período está fuera de la partición o el principio del período está después de su final.

Si se utiliza WINDOW en una columna calculada definida en la misma tabla que <relation> y se omite <orderBy>, se devuelve un error.

Si el principio del período está antes que la primera fila, se establece en la primera fila. Del mismo modo, si el final del período está después de la última fila de la partición, se establece en la última fila.

<reset> solo se puede usar en cálculos visuales y no se puede usar en combinación con <orderBy> o <partitionBy>. Si <reset> está presente, se puede especificar el <eje>, pero no se puede especificar la <relación>.

Ejemplo 1: medida

La medida siguiente:

DAX

```
3-day Average Price =  
AVERAGEX(  
    WINDOW(  
        -2,REL,0,REL,  
        SUMMARIZE(ALLSELECTED('Sales'), 'Date'[Date], 'Product'[Product]),  
        ORDERBY('Date'[Date]),  
        KEEP,  
        PARTITIONBY('Product'[Product])  
    ),  
    CALCULATE(AVERAGE(Sales[Unit Price]))  
)
```

Devuelve el promedio de 3 días de los precios unitarios de cada producto. Tenga en cuenta que el período de 3 días consta de tres días en los que el producto tiene ventas, no necesariamente tres días naturales consecutivos.

Ejemplo 2: medida

La medida siguiente:

DAX

```
RunningSum =  
SUMX (  
    WINDOW (  
        1, ABS, 0, REL,  
        ALLSELECTED (  
            'Date'[Fiscal Year],  
            'Date'[Month Number Of Year]  
        ),  
        PARTITIONBY ('Date'[Fiscal Year] )  
    ),  
    [Total Sales]  
)
```

Devuelve la suma en ejecución de Ventas totales por número de mes de año, reiniciando para cada año fiscal:

[\[+\] Expandir tabla](#)

Year	mes número de año	Sales Amount	RunningSum
FY2018	1	1 327 675 \$	1 327 675 \$
FY2018	2	3 936 463 \$	5 264 138 \$
FY2018	3	700 873 \$	5 965 011 \$
FY2018	4	1 519 275 \$	7 484 286 \$
FY2018	5	2 960 378 \$	10 444 664 \$
FY2018	6	1 487 671 \$	11 932 336 \$
FY2018	7	1 423 357 \$	13 355 693 \$
FY2018	8	2 057 902 \$	15 413 595 \$
FY2018	9	2 523 948 \$	17 937 543 \$
FY2018	10	561 681 \$	18 499 224 \$
FY2018	11	4 764 920 \$	23 264 145 \$
FY2018	12	596 747 \$	23 860 891 \$
FY2019	1	1 847 692 \$	1 847 692 \$
FY2019	2	2 829 362 \$	4 677 054 \$
FY2019	3	2 092 434 \$	6 769 488 \$
FY2019	4	2 405 971 \$	9 175 459 \$
FY2019	5	3 459 444 \$	12 634 903 \$
FY2019	6	2 850 649 \$	15 485 552 \$
FY2019	7	2 939 691 \$	18 425 243 \$
FY2019	8	3 964 801 \$	22 390 045 \$
FY2019	9	3 287 606 \$	25 677 650 \$
FY2019	10	2 157 287 \$	27 834 938 \$
FY2019	11	3 611 092 \$	31 446 030 \$

Year	mes número de año	Sales Amount	RunningSum
FY2019	12	2 624 078 \$	34 070 109 \$
FY2020	1	3 235 187 \$	3 235 187 \$
FY2020	2	4 070 046 \$	7 305 233 \$
FY2020	3	4 429 833 \$	11 735 066 \$
FY2020	4	4 002 614 \$	15 737 680 \$
FY2020	5	5 265 797 \$	21 003 477 \$
FY2020	6	3 465 241 \$	24 468 717 \$
FY2020	7	3 513 064 \$	27 981 781 \$
FY2020	8	5 247 165 \$	33 228 947 \$
FY2020	9	5 104 088 \$	38 333 035 \$
FY2020	10	3 542 150 \$	41 875 184 \$
FY2020	11	5 151 897 \$	47 027 081 \$
FY2020	12	4 851 194 \$	51 878 275 \$

Ejemplo 3: cálculo visual

La siguiente consulta DAX de cálculo visual:

```
DAX
TotalSalesRunningSumByYear = SUMX(WINDOW(0, ABS, 0, REL, ROWS,
HIGHESTPARENT), [SalesAmount])
```

Devuelve las ventas totales acumuladas por mes, calculadas a lo largo de cada año. Los valores 1 y -2 se pueden usar en lugar de HIGHESTPARENT, con los mismos resultados.

En la siguiente captura de pantalla se muestra la matriz visual y la expresión de cálculo visual:

X ✓ fx 1 TotalSalesRunningSumByYear = SUMX(WINDOW(0, ABS, 0, REL, ROWS, HIGHESTPARENT), [SalesAmount])

CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	TotalSalesRunningSumByYear	TotalSalesRunningSumByQuarter
2011	1	1	\$323,743.2056	323,743.21	323,743.21
		2	\$437,372.4476	761,115.65	761,115.65
		3	\$514,918.2284	1,276,033.88	1,276,033.88
	2	Total	\$1,276,033.8816	1,276,033.88	1,276,033.88
		4	\$523,950.0258	1,799,983.91	523,950.03
		5	\$539,469.0004	2,339,452.91	1,063,419.03
		6	\$647,524.214	2,986,977.12	1,710,943.24

Ejemplo 4: cálculo visual

La siguiente consulta DAX de cálculo visual:

DAX

TotalSalesRunningSumByQuarter = SUMX(WINDOW(0, ABS, 0, REL, , -1), [SalesAmount])
--

Devuelve las ventas totales acumuladas por mes, calculadas a lo largo de cada trimestre.

Contenido relacionado

[INDEX](#)
[MOVINGAVERAGE](#)
[OFFSET](#)
[ORDERBY](#)
[PARTITIONBY](#)
[RANGE](#)
[RANK](#)
[ROWNUMBERRUNNINGSUM](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Funciones financieras

Artículo • 20/10/2023

Las funciones financieras en DAX se usan en fórmulas que realizan cálculos financieros, como el valor neto presente y la tasa de devolución. Estas funciones son similares a las funciones financieras usadas en Microsoft Excel.

En esta categoría

Función	Descripción
ACCRINT	Devuelve el interés acumulado de un valor negociable que paga intereses periódicos.
ACCRINTM	Devuelve los intereses acumulado de un valor negociable que paga intereses al vencimiento.
AMORDEGRC	Devuelve la amortización de cada período contable. Es parecido a AMORLINC, salvo por el hecho de que se aplica un coeficiente de amortización en función de la duración de los activos.
AMORLINC	Devuelve la amortización de cada período contable.
COUPDAYBS	Devuelve el número de días desde el inicio de un período de cupón hasta su fecha de liquidación.
COUPDAYS	Devuelve el número de días del período de cupón que contiene la fecha de liquidación.
COUPDAYSNC	Devuelve el número de días desde la fecha de liquidación hasta la fecha del siguiente cupón.
COUPNCD	Devuelve la fecha del siguiente cupón después de la fecha de liquidación.
COUPNUM	Devuelve el número de cupones pagaderos entre la fecha de liquidación y la fecha de vencimiento, redondeado al cupón completo más cercano.
COUPPCD	Devuelve la fecha del cupón anterior a la fecha de liquidación.
CUMIPMT	Devuelve los intereses acumulados pagados de un préstamo entre start_period y end_period.
CUMPRINC	Devuelve capital acumulado pagado de un préstamo entre start_period y end_period.
DB	Devuelve la amortización de un activo en un período específico con el método de amortización por disminución fija de saldo.

Función	Descripción
DDB	Devuelve la amortización de un activo en un período específico con el método de amortización por doble disminución de saldo u otro método que se especifique.
DISC	Devuelve la tasa de descuento de un valor negociable.
DOLLARDE	Convierte un precio en dólares expresado como una parte entera y una parte fraccionaria, como 1,02, en un precio en dólares expresado como un número decimal.
DOLLARFR	Convierte un precio en dólares expresado como una parte entera y una parte fraccionaria, como 1,02, en un precio en dólares expresado como un número decimal.
DURATION	Devuelve la duración de Macauley de un valor nominal asumido de 100 USD.
EFFECT	Devuelve el tipo de interés anual efectiva si se conocen el tipo de interés anual nominal y el número de períodos de interés compuesto al año.
FV	Calcula el valor futuro de una inversión basado en un tipo de interés constante.
INTRATE	Devuelve el tipo de interés de un valor negociable invertido en su totalidad.
IPMT	Devuelve el pago de los intereses durante un período determinado con base en pagos periódicos y constantes, y en un tipo de interés constante.
ISPMT	Calcula los intereses pagados (o recibidos) del período especificado de un préstamo (o inversión) con pagos fijos de capital.
MDURATION	Devuelve la duración de Macauley modificada de un valor negociado con un valor nominal de 100 USD.
NOMINAL	Devuelve el tipo de interés anual nominal si se conocen el tipo efectivo y el número de períodos de interés compuesto al año.
NPER	Devuelve el número de períodos de una inversión basándose en los pagos periódicos y constantes, y en un tipo de interés constante.
ODDFPRICE	Devuelve el precio por cada 100 USD de valor nominal de un valor negociable que tiene un primer período impar (corto o largo).
ODDFYIELD	Devuelve el rendimiento de un valor negociable que tiene un primer período impar (corto o largo).
ODDLPRICE	Devuelve el precio por cada 100 USD de valor nominal de un valor negociable que tiene un último período de cupón impar (corto o largo).
ODDLYIELD	Devuelve el rendimiento de un valor negociable que tiene un último período impar (corto o largo).

Función	Descripción
PDURATION	Devuelve el número de períodos que requiere una inversión para alcanzar un valor especificado.
PMT	Calcula el pago de un préstamo basado en pagos constantes y un tipo de interés constante.
PPMT	Devuelve el pago del capital de una inversión durante un período determinado con base en pagos periódicos y constantes, y en un tipo de interés constante.
PRICE	Devuelve el precio por cada 100 USD de valor nominal de un valor negociable que paga intereses periódicos.
PRICEDISC	Devuelve el precio por cada 100 USD de valor nominal de un valor negociable con descuento.
PRICEMAT	Devuelve el precio por cada 100 USD de valor nominal de un valor negociable que paga intereses al vencimiento.
PV	Calcula el valor actual de un préstamo o una inversión, basado en un tipo de interés constante.
RATE	Devuelve el tipo de interés por período de una anualidad.
RECEIVED	Devuelve el importe recibido al vencimiento para un valor negociable invertido en su totalidad.
RRI	Devuelve un tipo de interés equivalente para el crecimiento de una inversión.
SLN	Devuelve la amortización lineal de un activo para un período.
SYD	Devuelve la amortización por método de anualidades de un activo para un período especificado.
TBILLEQ	Devuelve el rendimiento equivalente de un bono de una factura de tesorería.
TBILLPRICE	Devuelve el precio por cada 100 USD de valor nominal de una factura de tesorería.
TBILLYIELD	Devuelve el rendimiento de una factura de tesorería.
VDB	Devuelve la amortización de un activo en un período específico, incluidos los períodos parciales, con el método de amortización por doble disminución de saldo u otro método que se especifique.
XIRR	Devuelve la tasa interna de devolución de una programación de flujos de efectivo que no es necesariamente periódica.
XNPV	Devuelve el valor presente de una programación de flujos de efectivo que no es necesariamente periódica.

Función	Descripción
YIELD	Devuelve el rendimiento de un valor negociable que paga intereses periódicos.
YIELDDISC	Devuelve el rendimiento anual de un valor negociable con descuento.
YIELDMAT	Devuelve el rendimiento anual de un valor negociable que paga intereses al vencimiento.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

ACCRINT

Artículo • 20/10/2023

Devuelve el interés acumulado de un valor negociable que paga intereses periódicos.

Sintaxis

DAX

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

Parámetros

Término	Definición
issue	Fecha de emisión del valor negociable.
first_interest	Fecha del primer interés del valor negociable.
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
rate	Tasa del cupón anual del valor negociable.
par	Valor nominal del valor negociable.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.
calc_method	(Opcional) Valor lógico que especifica la manera de calcular el interés acumulado total cuando la fecha de liquidación es posterior a la fecha de first_interest. Si el parámetro calc_method se omite, se da por hecho que es "true". - Si calc_method da como resultado "true" o se omite, ACCRINT devuelve el interés total acumulado desde la emisión hasta la liquidación. - Si calc_method da como resultado "false", ACCRINT devuelve el interés acumulado desde first_interest hasta la liquidación.

El parámetro **basis** acepta los valores siguientes:

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Interés acumulado.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- ACCRINT se calcula de la siguiente manera:

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

donde:

- A_i = número de días acumulados para el período i^{th} de cuasicupón en un período impar.
- NC = número de períodos de cuasicupón que caben en un período impar. Si este número contiene una fracción, se eleva al siguiente número entero.
- NL_i = duración normal en días del período de cuasicupón en un período impar.
- Los parámetros issue, first_interest y settlement se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros issue, first_interest o settlement no son una fecha válida.
 - El parámetro issue es mayor o igual que settlement.
 - El parámetro rate es menor o igual que 0.
 - El parámetro par es menor o igual que 0.
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.

- basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

Data	Descripción
1 de marzo de 2007	Fecha de emisión
31 de agosto de 2008	Fecha del primer interés
1 de mayo de 2008	Fecha de liquidación
10 %	Tasa del cupón
1000	Valor nominal
2	Frecuencia semestral (ver más arriba)
0	Base 30/360 (vea más arriba)

Ejemplo 1

La consulta DAX siguiente:

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

Devuelve el interés acumulado desde la emisión hasta la liquidación, para un valor negociable con las condiciones especificadas anteriormente.

[Value]
116,944444444444

Ejemplo 2

La consulta DAX siguiente:

DAX

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

Devuelve el interés acumulado desde first_interest hasta la liquidación, para un valor con las condiciones especificadas anteriormente.

[Value]

66,9444444444445

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

ACCRINTM

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve los intereses acumulado de un valor negociable que paga intereses al vencimiento.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
issue	Fecha de emisión del valor negociable.
maturity	Fecha de vencimiento del valor negociable.
rate	Tasa del cupón anual del valor negociable.
par	Valor nominal del valor negociable.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360

Basis	Day count basis
3	Actual/365
4	European 30/360

Valor devuelto

Interés acumulado.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- ACCRINTM se calcula de la siguiente manera:

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

donde:

- A = número de días acumulados contados según una base mensual. En el caso de los elementos de intereses al vencimiento, se usa el número de días desde la fecha de emisión hasta la de vencimiento.
- D = base anual.
- Los parámetros issue y maturity se truncan en enteros.
- El parámetro basis se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros issue o maturity no son una fecha válida.
 - El parámetro issue es mayor o igual que maturity.
 - El parámetro rate es menor o igual que 0.
 - El parámetro par es menor o igual que 0.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

[Expandir tabla](#)

Data	Descripción
1 de abril de 2008	Fecha de emisión
15 de junio de 2008	Fecha de vencimiento
10 %	Porcentaje del cupón
1000	Valor nominal
3	Base real/365 (ver más arriba)

La consulta DAX siguiente:

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

Devuelve los intereses acumulados para un valor negociable con las condiciones especificadas anteriormente.

[Expandir tabla](#)

[Value]
20,5479452054795

Comentarios

¿Le ha resultado útil esta página?

[Sí](#)[No](#)

AMORDEGRC

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la amortización de cada período contable. Esta función se proporciona para el sistema de contabilidad francés. Si se adquiere un activo en mitad del período contable, se tendrá en cuenta la amortización prorrataeda. La función es parecida a AMORLINC, salvo por el hecho de que, en el cálculo, se aplica un coeficiente de amortización en función de la vida útil de los activos.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
cost	Costo del activo.
date_purchased	Fecha de compra del activo.
first_period	Fecha de finalización del primer período.
salvage	Valor residual al final de la vida del activo.
period	Período.
rate	Tasa de amortización.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Date system
Omitido o 0	360 días (método NASD)
1	Real
3	365 días en un año
4	360 días en un año (método europeo)

Valor devuelto

Amortización de cada período contable.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- Esta función devolverá la amortización hasta el último período de la vida de los activos o hasta que el valor acumulado de la amortización sea mayor que el costo de los activos menos el valor residual.
- Los coeficientes de amortización son los siguientes:

 Expandir tabla

Vida de los activos (1/tasa)	Coeficiente de amortización
Entre 3 y 4 años	1.5
Entre 5 y 6 años	2
Más de 6 años	2.5

- La tasa de amortización aumentará al 50 % durante el penúltimo período y al 100 % en el último período.
- Los parámetros period y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - cost < 0.
 - Los parámetros first_period o date_purchased no son una fecha válida.
 - date_purchased > first_period.

- salvage < 0 o salvage > cost.
- period < 0.
- El parámetro rate es menor o igual que 0.
- La vida útil de los activos oscila entre 0 (cero) y 1, 1 y 2, 2 y 3, o 4 y 5.
- El parámetro basis es cualquier número distinto de 0, 1, 3 o 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

[+] Expandir tabla

Data	Descripción
2400	Costos
19 de agosto de 2008	Fecha de compra
31 de diciembre de 2008	Fin del primer período
300	Valor residual
1	Período
15 %	Tasa de amortización
1	Base real (ver más arriba)

La consulta DAX siguiente:

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

Devuelve la amortización del primer período, dadas las condiciones especificadas anteriormente.

[+] Expandir tabla

[Value]
776

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

AMORLINC

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la amortización de cada período contable. Esta función se proporciona para el sistema de contabilidad francés. Si se adquiere un activo en mitad del período contable, se tendrá en cuenta la amortización prorrataeada.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
cost	Costo del activo.
date_purchased	Fecha de compra del activo.
first_period	Fecha de finalización del primer período.
salvage	Valor residual al final de la vida del activo.
period	Período.
rate	Tasa de amortización.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Date system
Omitido o 0	360 días (método NASD)
1	Real
3	365 días en un año
4	360 días en un año (método europeo)

Valor devuelto

Amortización de cada período contable.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- Los parámetros period y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - cost < 0.
 - Los parámetros first_period o date_purchased no son una fecha válida.
 - date_purchased > first_period.
 - salvage < 0 o salvage > cost.
 - period < 0.
 - El parámetro rate es menor o igual que 0.
 - El parámetro basis es cualquier número distinto de 0, 1, 3 o 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
2400	Costos
19 de agosto de 2008	Fecha de compra

Data	Descripción
31 de diciembre de 2008	Fin del primer período
300	Valor residual
1	Período
15 %	Tasa de amortización
1	Base real (ver más arriba)

La consulta DAX siguiente:

DAX
<pre>EVALUATE { AMORLINC(2400, DATE(2008,8,19), DATE(2008,12,31), 300, 1, 0.15, 1) }</pre>

Devuelve la amortización del primer período, dadas las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
360

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COUPDAYBS

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de días desde el inicio de un período de cupón hasta su fecha de liquidación.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual

Basis	Day count basis
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Número de días desde el inicio de un período de cupón hasta su fecha de liquidación.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- Los parámetros settlement y maturity se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
25-Ene-11	Fecha de liquidación
15-Nov-11	Fecha de vencimiento
2	Cupón semestral (ver más arriba)
1	Base real/real (ver más arriba)

La consulta DAX siguiente:

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

Devuelve el número de días desde el inicio de un período de cupón hasta su fecha de liquidación, para un bono con las condiciones anteriores.

[Expandir tabla](#)

[Value]
71

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COUPDAYS

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de días del período de cupón que contiene la fecha de liquidación.

Sintaxis

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

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360

Basis	Day count basis
3	Actual/365
4	European 30/360

Valor devuelto

Número de días del período de cupón que contiene la fecha de liquidación.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- Los parámetros settlement y maturity se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
25-Ene-11	Fecha de liquidación
15-Nov-11	Fecha de vencimiento
2	Cupón semestral (ver más arriba)
1	Base real/real (ver más arriba)

La consulta DAX siguiente:

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

Devuelve el número de días en el período de cupón que contiene la fecha de liquidación para un bono con las condiciones especificadas anteriormente.

[Expandir tabla](#)

[Value]
181

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COUPDAYSNC

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de días desde la fecha de liquidación hasta la fecha del siguiente cupón.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual

Basis	Day count basis
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Número de días desde la fecha de liquidación hasta la fecha del siguiente cupón.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- Los parámetros settlement y maturity se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
25-Ene-11	Fecha de liquidación
15-Nov-11	Fecha de vencimiento
2	Cupón semestral (ver más arriba)
1	Base real/real (ver más arriba)

La consulta DAX siguiente:

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

Devuelve el número de días desde la fecha de liquidación hasta la fecha del siguiente cupón, para un bono con las condiciones especificadas anteriormente.

[Expandir tabla](#)

[Value]
110

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COUPNCD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la fecha del siguiente cupón después de la fecha de liquidación.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360

Basis	Day count basis
3	Actual/365
4	European 30/360

Valor devuelto

La fecha del siguiente cupón después de la fecha de liquidación.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- Los parámetros settlement y maturity se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
25-Ene-11	Fecha de liquidación
15-Nov-11	Fecha de vencimiento
2	Cupón semestral (ver más arriba)
1	Base real/real (ver más arriba)

La consulta DAX siguiente:

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

Devuelve la fecha del siguiente cupón después de la fecha de liquidación para un bono con las condiciones especificadas anteriormente.

[Expandir tabla](#)

[Value]
15/5/2011 12:00:00

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COUPNUM

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de cupones pagaderos entre la fecha de liquidación y la fecha de vencimiento, redondeado al cupón completo más cercano.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual

Basis	Day count basis
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Número de cupones pagaderos entre la fecha de liquidación y la fecha de vencimiento.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- Los parámetros settlement y maturity se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
25-Ene-07	Fecha de liquidación
15-Nov-08	Fecha de vencimiento
2	Cupón semestral (ver más arriba)
1	Base real/real (ver más arriba)

La consulta DAX siguiente:

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

Devuelve el número de cupones pagaderos para un bono con las condiciones especificadas anteriormente.

[Expandir tabla](#)

[Value]
4

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COUPPCD

Artículo • 20/10/2023

Devuelve la fecha del cupón anterior a la fecha de liquidación.

Sintaxis

DAX

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

Parámetros

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

La fecha de cupón anterior a la fecha de liquidación.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- Los parámetros settlement y maturity se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Data	Descripción
25-Ene-11	Fecha de liquidación
15-Nov-11	Fecha de vencimiento
2	Cupón semestral (ver más arriba)
1	Base real/real (ver más arriba)

La consulta DAX siguiente:

DAX

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

Devuelve la fecha del siguiente cupón anterior a la fecha de liquidación para un bono con las condiciones especificadas anteriormente.

[Value]

15/11/2010 12:00:00

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

CUMIPMT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve los intereses acumulados pagados de un préstamo entre start_period y end_period.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
rate	Tipo de interés.
nper	Número total de períodos de pago.
pv	Valor actual.
start_period	Primer período del cálculo. Debe estar entre 1 y end_period (incluidos).
end_period	Último período del cálculo. Debe estar entre start_period y nper (incluidos).
tipo	Tiempo del pago. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **type** acepta los valores siguientes:

Expandir tabla

Tipo	Control de tiempo
0 (cero)	Pago al final del período
1	Pago al inicio del período

Valor devuelto

Los intereses acumulados pagados en el período especificado.

Comentarios

- Asegúrese de ser coherente con las unidades que se usan para especificar los parámetros rate y nper. Si realiza pagos mensuales en un préstamo de cuatro años con un tipo de interés anual del 10 %, use 0,1/12 en rate y 4*12 en nper. Si realiza pagos anuales en el mismo préstamo, use 0,1 en rate y 4 en nper.
- start_period, end_period y type se redondean al entero más próximo.
- Se devuelve un error si ocurre lo siguiente:
 - El parámetro rate es menor o igual que 0.
 - nper < 1.
 - El parámetro pv es menor o igual que 0.
 - start_period < 1 o start_period > end_period.
 - end_period < start_period o end_period > nper.
 - El parámetro type es cualquier número distinto de 0 o 1.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

 Expandir tabla

Data	Descripción
9%	Tipo de interés anual
30	Años del préstamo
12 5000	Valor actual

Ejemplo 1

La consulta DAX siguiente:

```
DAX
```

```
EVALUATE  
{
```

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

Devuelve los intereses totales pagados en el segundo año de pagos, los períodos comprendidos entre el 13 y el 24, suponiendo que los pagos se realizan al inicio de cada mes.

 Expandir tabla

[Value]
-11052,3395838718

Ejemplo 2

La consulta DAX siguiente:

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

Devuelve los intereses pagados en un único pago en el primer mes, suponiendo que el pago se realiza al final del mes.

 Expandir tabla

[Value]
-937,5

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CUMPRINC

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve capital acumulado pagado de un préstamo entre start_period y end_period.

Sintaxis

DAX

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

Parámetros

[] Expandir tabla

Término	Definición
rate	Tipo de interés.
nper	Número total de períodos de pago.
pv	Valor actual.
start_period	Primer período del cálculo. Debe estar entre 1 y end_period (incluidos).
end_period	Último período del cálculo. Debe estar entre start_period y nper (incluidos).
tipo	Tiempo del pago. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **type** acepta los valores siguientes:

[] Expandir tabla

Tipo	Control de tiempo
0 (cero)	Pago al final del período
1	Pago al inicio del período

Valor devuelto

El capital acumulado pagado en el período especificado.

Comentarios

- Asegúrese de ser coherente con las unidades que se usan para especificar los parámetros rate y nper. Si realiza pagos mensuales en un préstamo de cuatro años con un tipo de interés anual del 10 %, use 0,1/12 en rate y 4*12 en nper. Si realiza pagos anuales en el mismo préstamo, use 0,1 en rate y 4 en nper.
- start_period, end_period y type se redondean al entero más próximo.
- Se devuelve un error si ocurre lo siguiente:
 - El parámetro rate es menor o igual que 0.
 - nper < 1.
 - El parámetro pv es menor o igual que 0.
 - start_period < 1 o start_period > end_period.
 - end_period < start_period o end_period > nper.
 - El parámetro type es cualquier número distinto de 0 o 1.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

 Expandir tabla

Data	Descripción
9%	Tipo de interés anual
30	Plazo en años
12 5000	Valor actual

Ejemplo 1

La consulta DAX siguiente:

```
DAX
```

```
EVALUATE  
{
```

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

Devuelve el capital total pagado en el segundo año de pagos, los períodos comprendidos entre el 13 y el 24, suponiendo que los pagos se realizan al inicio de cada mes.

 Expandir tabla

[Value]
-927,153472378062

-927,153472378062

Ejemplo 2

La consulta DAX siguiente:

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

Devuelve el capital pagado en un único pago en el primer mes, suponiendo que el pago se realiza al final del mes.

 Expandir tabla

[Value]
-68,2782711809784

-68,2782711809784

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DB

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve la amortización de un activo en un período específico con el método de amortización por disminución fija de saldo.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
cost	Costo inicial del activo.
salvage	Valor al final de la amortización (también conocido como valor residual del activo). Este valor puede ser 0.
vida	Número de períodos durante los cuales se amortiza el activo (también conocido como vida útil del activo).
period	Período para el que se quiere calcular la amortización. El período debe usar las mismas unidades que la vida del activo. Debe estar entre 1 y la vida del activo (incluida).
month	(Opcional) Número de meses del primer año. Si el parámetro month se omite, se da por hecho que es 12.

Valor devuelto

Amortización en el período especificado.

Comentarios

- El método de amortización por disminución fija de saldo calcula la amortización a una tasa fija. DB usa las fórmulas siguientes para calcular la amortización de un período:

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

donde:

- $\text{rate} = 1 - ((\frac{\text{salvage}}{\text{cost}})^{\frac{1}{\text{life}}})$, rounded to three decimal places

- La amortización del primer y el último período es un caso especial.
- En el primer período, DB usa esta fórmula:

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

- En el último período, DB usa esta fórmula:

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

- Los parámetros period y month se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - cost < 0.
 - salvage < 0.
 - life < 1.
 - period < 1 o period > life.
 - month < 1 o month > 12.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

Ejemplo 1

La consulta DAX siguiente:

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

Devuelve la amortización de un activo en los dos últimos meses del primer año, suponiendo que valdrá 0 USD después de 6 años.

[] Expandir tabla

[Value]
166666,666666667

Ejemplo 2

A continuación se calcula la amortización total de todos los activos en años diferentes a lo largo de su vida útil. Aquí, el primer año solo incluye 7 meses de amortización, y el último, tan solo 5 meses.

```
DAX

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)
        )
    )
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DDB

Artículo • 20/10/2023

Devuelve la amortización de un activo en un período específico con el método de amortización por doble disminución de saldo u otro método que se especifique.

Sintaxis

DAX

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

Parámetros

Término	Definición
cost	Costo inicial del activo.
salvage	Valor al final de la amortización (también conocido como valor residual del activo). Este valor puede ser 0.
vida	Número de períodos durante los cuales se amortiza el activo (también conocido como vida útil del activo).
period	Período para el que se quiere calcular la amortización. El período debe usar las mismas unidades que la vida del activo. Debe estar entre 1 y la vida del activo (incluida).
factor	(Opcional) Tasa a la que disminuye el saldo. Si el argumento factor se omite, se interpretará como 2 (método de amortización por doble disminución de saldo).

Valor devuelto

Amortización en el período especificado.

Comentarios

- El método de amortización por doble disminución de saldo calcula la amortización a una tasa acelerada. La amortización es mayor en el primer período disminuye en períodos sucesivos. DDB usa la fórmula siguiente para calcular la amortización de un período:

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

- Cambie el parámetro factor si no quiere usar el método de amortización por doble disminución de saldo.
- Use la función VDB si quiere cambiar al método de amortización lineal cuando dicha amortización sea mayor que el cálculo del saldo decreciente.
- El parámetro period se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - cost < 0.
 - salvage < 0.
 - life < 1.
 - period < 1 o period > life.
 - El parámetro factor es menor o igual que 0.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

Ejemplo 1

La consulta DAX siguiente:

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

Devuelve la amortización de un activo en el año 5th, suponiendo que valdrá 0 USD después de 10 años. Este cálculo utiliza un parámetro factor de 1,5.

[Value]
78300,9375

Ejemplo 2

A continuación se calcula la amortización total de todos los activos en años diferentes a lo largo de su vida útil. Este cálculo usa 2 como valor predeterminado del parámetro factor (método de amortización por doble disminución de saldo).

```
DAX  
  
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])  
    )  
)
```

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

DISC

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la tasa de descuento de un valor negociable.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
pr	Precio del valor negociable por cada 100 USD de valor nominal.
redemption	Valor de canje del valor negociable por cada 100 USD de valor nominal.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual

Basis	Day count basis
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Tasa de descuento.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2018 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2018; la de liquidación, el 1 de julio de 2018; y la de vencimiento, el 1 de enero de 2048, 30 años después del 1 de enero de 2018, la fecha de la emisión.
- DISC se calcula de la siguiente manera:

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

donde:

- B = número de días en un año, en función de la base del año utilizada.
- DSM = número de días comprendidos entre la liquidación y el vencimiento.
- Los parámetros settlement y maturity se truncan en enteros.
- El parámetro basis se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - El parámetro pr es menor o igual que 0.

- El parámetro redemption es menor o igual que 0.
- basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
01/07/2018	Fecha de liquidación
01/01/2048	Fecha de vencimiento
97,975	Price
100	Valor de canje
1	Base real/real (ver más arriba)

La consulta DAX siguiente:

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

Devuelve la tasa de descuento del bono con las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
0,000686384169121348

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DOLLARDE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Convierte un precio en dólares expresado como una parte entera y una parte fraccionaria, como 1,02, en un precio en dólares expresado como un número decimal. En ocasiones, los números en dólares fraccionarios se usan para los precios de los valores negociables.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
fractional_dollar	Número expresado como una parte entera y una parte fraccionaria, separadas por un símbolo decimal.
fraction	Entero que se usa como denominador de la fracción.

Valor devuelto

Valor decimal de *fractional_dollar*.

Comentarios

- La parte fraccionaria del valor se divide entre un entero que se especifique. Por ejemplo, si quiere que el precio se exprese en una precisión de 1/16 de un dólar, divida la parte fraccionaria entre 16. En este caso, 1,02 representa 1,125 USD ($1 \text{ USD} + 2/16 = 1,125 \text{ USD}$).
- El parámetro *fraction* se redondea al entero más cercano.

- Se devuelve un error si ocurre lo siguiente:
 - fraction < 1.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La consulta DAX siguiente:

```
DAX

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

Devuelve 1,125, el precio decimal del precio fraccionario original (1,02), leído como 1 y 2/16. Dado que el valor fraccionario es 16, el precio tiene una precisión de 1/16 de dólar.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DOLLARFR

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Convierte un precio en dólares expresado como número decimal en un precio en dólares expresado con una parte entera y otra fraccionaria, como 1,02. En ocasiones, los números en dólares fraccionarios se usan para los precios de los valores negociables.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
decimal_dollar	Número decimal.
fraction	Entero que se usa como denominador de la fracción.

Valor devuelto

Valor fraccionario de *decimal_dollar*, expresado como una parte entera y una parte fraccionaria.

Comentarios

- El parámetro *fraction* se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - *fraction* < 1.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La consulta DAX siguiente:

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

Devuelve 1,02, leído como 1 y 2/16, que es el precio fraccionario correspondiente al precio decimal original, 1,125. Dado que el valor fraccionario es 16, el precio tiene una precisión de 1/16 de dólar.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DURATION

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la duración de Macauley de un valor nominal asumido de 100 USD. La duración se define como la media ponderada del valor actual de los flujos de efectivo y se utiliza como medida de la respuesta del precio de un bono a los cambios en el rendimiento.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
coupon	Tasa del cupón anual del valor negociable.
yld	Rendimiento anual del valor negociable.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Duración de Macauley.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- Los parámetros settlement y maturity se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - coupon < 0.
 - yld < 0
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
01/07/2018	Fecha de liquidación
01/01/2048	Fecha de vencimiento
8,0 %	Porcentaje del cupón
9,0 %	Tipo de interés
2	Frecuencia semestral (ver más arriba)
1	Base real/real (ver más arriba)

La consulta DAX siguiente:

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

Devuelve la duración de Macauley para un bono con las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
10,9191452815919

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

EFFECT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el tipo de interés anual efectiva si se conocen el tipo de interés anual nominal y el número de períodos de interés compuesto al año.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
nominal_rate	Tipo de interés nominal.
nper	Número de períodos de interés compuesto al año.

Valor devuelto

Tipo de interés anual efectivo.

Comentarios

- EFFECT se calcula de la siguiente manera:

$$\text{EFFECT} = \left(1 + \frac{\text{nominal_rate}}{\text{nper}}\right)^{\text{nper}} - 1$$

- El parámetro nper se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - El parámetro nominal_rate es menor o igual que 0.
 - nper < 1.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
5,25 %	Tipo de interés nominal
4	Número de períodos de interés compuesto al año.

La consulta DAX siguiente:

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

Devuelve el tipo de interés efectivo mediante las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
0,0535426673707584

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

FV

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Calcula el valor futuro de una inversión basado en un tipo de interés constante. Puede usar FV con pagos periódicos, constantes o con un único pago de suma total.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
rate	Tipo de interés por período.
nper	Número total de períodos de pago en una anualidad.
pmt	Pago realizado en cada período; no puede cambiar durante la vida útil de la anualidad. Normalmente, pmt incluye el capital y los intereses, pero no otros honorarios o impuestos.
pv	(Opcional) Valor actualizado o suma total de una serie de pagos futuros en el momento actual. Si el parámetro pv se omite, se da por hecho que es BLANK.
tipo	(Opcional) Número 0 o 1 que indica cuándo vencen los pagos. Si el parámetro type se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **type** acepta los valores siguientes:

Expandir tabla

Establecer type igual a	Si los pagos han vencido
Omitido o 0	Al final del período
1	Al inicio del período

Nota: Para obtener una descripción más completa de los argumentos de FV y más información sobre las funciones de anualidad, vea la función PV.

Valor devuelto

Valor futuro de una inversión.

Comentarios

- Asegúrese de ser coherente con las unidades que se usan para especificar los parámetros rate y nper. Si realiza pagos mensuales en un préstamo de cuatro años con un interés anual del 12 %, use 0,12/12 en rate y 4*12 en nper. Si realiza pagos anuales en el mismo préstamo, use 0,12 en rate y 4 en nper.
- Para todos los argumentos, el efectivo que se paga, como los depósitos de ahorro, se representa mediante números negativos y el que se recibe, como los dividendos, se representa mediante números positivos.
- El parámetro type se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - nper < 1
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
6 %	Tipo de interés anual
10	Número de pagos
-200	Importe del pago
-500	Valor actual
1	El pago vence al inicio del período (0 indica que el pago vence al final del período).

La consulta DAX siguiente:

DAX

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

Devuelve el valor futuro de una inversión con las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
2581,40337406012

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

INTRATE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve el tipo de interés de un valor negociable invertido en su totalidad.

Sintaxis

DAX
<code>INTRATE(<settlement>, <maturity>, <investment>, <redemption>[, <basis>])</code>

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
investment	Importe invertido en el valor negociable.
redemption	Importe que se recibirá al vencimiento.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual

Basis	Day count basis
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Tipo de interés.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- INTRATE se calcula de la siguiente manera:

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

donde:

- B = número de días en un año, en función de la base del año utilizada.
- DIM = número de días desde la liquidación hasta el vencimiento.
- Los parámetros settlement y maturity se truncan en enteros.
- El parámetro basis se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - El parámetro investment es menor o igual que 0.
 - El parámetro redemption es menor o igual que 0.
 - basis < 0 o basis > 4.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

[\[+\] Expandir tabla](#)

Data	Descripción
15/2/2008	Fecha de liquidación
15/5/2008	Fecha de vencimiento
1 000 000 USD	Inversión
1 014 420 USD	Valor de canje
2	Base real/360

La consulta DAX siguiente:

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

Devuelve la tasa de descuento para un bono con las condiciones especificadas anteriormente.

[\[+\] Expandir tabla](#)

[Value]
0,05768

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

IPMT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve el pago de los intereses durante un período determinado con base en pagos periódicos y constantes, y en un tipo de interés constante.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
rate	Tipo de interés por período.
per	Período para el que quiere encontrar el tipo de interés. Debe estar entre 1 y nper (incluido).
nper	Número total de períodos de pago en una anualidad.
pv	Valor actualizado o suma total de una serie de pagos futuros en el momento actual.
fv	(Opcional) Valor futuro o saldo en efectivo que se quiere obtener después del último pago. Si el parámetro fv se omite, se da por hecho que es BLANK.
tipo	(Opcional) Número 0 o 1 que indica cuándo vencen los pagos. Si el parámetro type se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **type** acepta los valores siguientes:

Expandir tabla

Establecer type igual a	Si los pagos han vencido
Omitido o 0	Al final del período

Establecer type igual a	Si los pagos han vencido
1	Al inicio del período

Valor devuelto

Pago del interés para el período determinado.

Comentarios

- Asegúrese de ser coherente con las unidades que se usan para especificar los parámetros rate y nper. Si realiza pagos mensuales en un préstamo de cuatro años con un interés anual del 12 %, use 0,12/12 en rate y 4*12 en nper. Si realiza pagos anuales en el mismo préstamo, use 0,12 en rate y 4 en nper.
- Para todos los argumentos, el efectivo que se paga, como los depósitos de ahorro, se representa mediante números negativos y el que se recibe, como los dividendos, se representa mediante números positivos.
- El parámetro type se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - per < 1 o per > nper
 - nper < 1
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

 Expandir tabla

Data	Descripción
10 %	Interés anual
3	Años del préstamo
8000 USD	Valor actual del préstamo

Ejemplo 1

La consulta DAX siguiente:

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

Devuelve el interés mensual vencido en el primer mes para un préstamo con las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
-66,6666666666667

Ejemplo 2

La consulta DAX siguiente:

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

Devuelve el interés anual vencido en el último año para un préstamo con las condiciones especificadas anteriormente, donde los pagos se realizan anualmente.

 Expandir tabla

[Value]
-292,447129909366

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISPMT

Artículo • 20/10/2023

Calcula los intereses pagados (o recibidos) del período especificado de un préstamo (o inversión) con pagos fijos de capital.

Sintaxis

DAX

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

Parámetros

Término	Definición
rate	Tipo de interés de la inversión.
per	Período para el que quiere encontrar el tipo de interés. Debe estar entre 0 y nper-1 (incluidos).
nper	Número total de períodos de pago de la inversión.
pv	Valor actual de la inversión. Para un préstamo, pv es el importe del préstamo.

Valor devuelto

Intereses pagados (o recibidos) del período especificado.

Comentarios

- Asegúrese de ser coherente con las unidades que se usan para especificar los parámetros rate y nper. Si realiza pagos mensuales en un préstamo de cuatro años con un tipo de interés anual del 12 %, use 0,12/12 para rate y 4*12 para nper. Si realiza pagos anuales en el mismo préstamo, use 0,12 en rate y 4 en nper.
- Para todos los argumentos, los efectivos que se pagan, como los depósitos de ahorro u otros reembolsos, se representan mediante números negativos y los que se reciben, como los dividendos u otros depósitos, se representan mediante números positivos.

- ISPMT cuenta cada período a partir de cero, no de uno.
- La mayoría de préstamos usan un calendario de reembolsos con pagos periódicos fijos. La función IPMT devuelve el pago de intereses de un período determinado para este tipo de préstamo.
- Algunos préstamos usan un calendario de reembolsos con pagos periódicos fijos. La función ISPMT devuelve el pago de intereses de un período determinado para este tipo de préstamo.
- Se devuelve un error si ocurre lo siguiente:
 - El parámetro nper es igual a 0.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Data	Descripción
4000 USD	Valor actual
4	Número de períodos
10 %	Tarifa

Para ilustrar cuándo se debe usar ISPMT, en la tabla de amortización siguiente se usa un calendario de reembolsos fijos de capital con las condiciones especificadas anteriormente. El importe de los intereses de cada período es igual a la tasa multiplicada por el saldo no pagado del período anterior. Y el pago de cada período es igual a la parte fija de capital más los intereses del período.

Período	Pago de capital	Pago de intereses	Pago total	Saldo
				4000
1	1000	400	1400	3000
2	1000	300	1300	2000
3	1000	200,00	1200	1000
4	1000	100,00	1100	0,00

La consulta DAX siguiente:

DAX

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

Devuelve los intereses pagados durante cada período, utilizando el calendario de reembolsos fijos de capital y las condiciones especificadas anteriormente. Los valores son negativos para indicar que se trata de intereses pagados, no recibidos.

[Value]	[Interest Payment]
0	-400
1	-300
2	-200
3	-100

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

MDURATION

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la duración de Macauley modificada de un valor negociado con un valor nominal de 100 USD.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
coupon	Tasa del cupón anual del valor negociable.
yld	Rendimiento anual del valor negociable.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

La duración de Macauley modificada.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- La duración modificada se define de la siguiente manera:

$$\text{MDURATION} = \frac{\text{DURATION}}{1 + \left(\frac{\text{Market yield}}{\text{Coupon payments per year}} \right)}$$

- Los parámetros settlement y maturity se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - coupon < 0.
 - yld < 0
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.

- basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

[+] Expandir tabla

Data	Descripción
1/1/2008	Fecha de liquidación
1/1/2016	Fecha de vencimiento
8 %	Porcentaje del cupón
9%	Tipo de interés
2	Frecuencia semestral (ver más arriba)
1	Base real/real (ver más arriba)

La consulta DAX siguiente:

DAX

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

Devuelve la duración de Macauley modificada de un bono con las condiciones especificadas anteriormente.

[+] Expandir tabla

[Value]
5,73566981391884

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NOMINAL

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el tipo de interés anual nominal si se conocen el tipo efectivo y el número de períodos de interés compuesto al año.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
effect_rate	Tipo interés efectivo.
nper	Número de períodos de interés compuesto al año.

Valor devuelto

Tipo de interés anual nominal.

Comentarios

- La relación entre NOMINAL y EFFECT se muestra en la ecuación siguiente:

$$\text{EFFECT} = \left(1 + \frac{\text{nominal_rate}}{\text{nper}}\right)^{\text{nper}} - 1$$

- El parámetro nper se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - El parámetro effect_rate es menor o igual que 0.
 - nper < 1.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
5,3543 %	Tipo de interés efectivo.
4	Número de períodos de interés compuesto al año.

La consulta DAX siguiente:

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

Devuelve el tipo de interés nominal con las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
0,052500319868356

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NPER

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de períodos de una inversión basándose en los pagos periódicos y constantes, y en un tipo de interés constante.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
rate	Tipo de interés por período.
pmt	Pago realizado en cada período; no puede cambiar durante la vida útil de la anualidad. Normalmente, pmt incluye el capital y los intereses, pero no otros honorarios o impuestos.
pv	Valor actualizado o suma total de una serie de pagos futuros en el momento actual.
fv	(Opcional) Valor futuro o saldo en efectivo que se quiere obtener después del último pago. Si el parámetro fv se omite, se da por hecho que es BLANK.
tipo	(Opcional) Número 0 o 1 que indica cuándo vencen los pagos. Si el parámetro type se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **type** acepta los valores siguientes:

Expandir tabla

Establecer type igual a	Si los pagos han vencido
Omitido o 0	Al final del período
1	Al inicio del período

Valor devuelto

Número de períodos de una inversión.

Comentarios

- El parámetro type se redondea al entero más cercano.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
12 %	Tipo de interés anual
-100	Pago realizado en cada período
-1000	Valor actual
10000	Valor futuro
1	El pago vence al inicio del período (ver más arriba)

La consulta DAX siguiente:

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

Devuelve el número de períodos de la inversión que describen las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
59,6738656742946

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ODDFPRICE

Artículo • 20/10/2023

Devuelve el precio por cada 100 USD de valor nominal de un valor negociable que tiene un primer período impar (corto o largo).

Sintaxis

DAX

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

Parámetros

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
issue	Fecha de emisión del valor negociable.
first_coupon	Fecha del primer cupón del valor negociable.
rate	Tipo de interés del valor negociable.
yld	Rendimiento anual del valor negociable.
redemption	Valor de canje del valor negociable por cada 100 USD de valor nominal.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Precio por cada 100 USD de valor nominal.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- ODDFPRICE se calcula de la siguiente manera:

Primer cupón corto impar:

$$\text{ODDFPRICE} = \left[\frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(N-1+\frac{\text{DSC}}{\text{E}})}} \right] + \left[\frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{DFC}}{\text{E}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(\frac{\text{DSC}}{\text{E}})}} \right] + \left[\sum_{k=2}^N \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(k-1+\frac{\text{DSC}}{\text{E}})}} \right] - \left[100 \times \frac{\text{rate}}{\text{frequency}} \right]$$

donde:

- o A = número de días desde el inicio del período de cupón hasta la fecha de liquidación (días acumulados).
- o DSC = número de días desde la liquidación hasta la fecha del siguiente cupón.
- o DFC = número de días desde el inicio del primer cupón impar hasta la fecha del primer cupón.
- o E = número de días del período del cupón.
- o N = número de cupones pagaderos entre la fecha de liquidación y la fecha de canje. (Si este número contiene una fracción, se eleva al siguiente número entero).

Primer cupón impar largo:

$$\text{ODDFPRICE} = \left[\frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(N+N_q+\frac{\text{DSC}}{\text{E}})}} \right] + \left[\frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \left[\sum_{i=1}^{\text{NC}} \frac{\text{DC}_i}{\text{NL}_i} \right]}{(1 + \frac{\text{yld}}{\text{frequency}})^{(N_q+\frac{\text{DSC}}{\text{E}})}} \right] + \left[\sum_{k=1}^N \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(k-N_q+\frac{\text{DSC}}{\text{E}})}} \right] - \left[100 \times \frac{\text{rate}}{\text{frequency}} \right]$$

donde:

- o A_i = número de días desde el inicio de i^{th} o último período de cuasicupón en un período impar.
- o DC_i = número de días desde la fecha a partir de la cual se calculan los intereses devengados (o fecha de emisión) hasta el primer cuasicupón ($i = 1$) o el número de días del cuasicupón ($i = 2, \dots, i = \text{NC}$).
- o DSC = número de días desde la liquidación hasta la fecha del siguiente cupón.
- o E = número de días del período del cupón.
- o N = número de cupones pagaderos entre la fecha del primer cupón real y la fecha de canje. (Si este número contiene una fracción, se eleva al siguiente número entero).
- o NC = número de períodos de cuasicupón que caben en un período impar. (Si este número contiene una fracción, se eleva al siguiente número entero).
- o NL_i = duración normal en días de i^{th} completo o último período de cuasicupón en un período impar.
- o N_q = número de períodos de cuasicupón completos entre la fecha de liquidación y el primer cupón.

- Los parámetros settlement, maturity, issue y first_coupon se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - o Los parámetros settlement, maturity, issue o first_coupon no son una fecha válida.
 - o No se cumple que maturity > first_coupon > settlement > issue.
 - o rate < 0.
 - o yld < 0.
 - o El parámetro redemption es menor o igual que 0.
 - o El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - o basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Data	Descripción del argumento
11/11/2008	Fecha de liquidación
01/03/2021	Fecha de vencimiento
15/10/2008	Fecha de emisión
01/03/2009	Fecha del primer cupón
7,85 %	Porcentaje del cupón
6,25 %	Tipo de interés
100 USD	Valor de canje
2	Frecuencia semestral
1	Base real/real

La consulta DAX siguiente:

```
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)  
}
```

Devuelve el precio por cada 100 USD de valor nominal de un valor negociable que tiene un primer período impar (corto o largo), con las condiciones especificadas anteriormente.

[Value]
113,597717474079

Comentarios

¿Le ha resultado útil esta página? [!\[\]\(f184c0dccff2ca4c061b4f3965fcea8f_img.jpg\) Yes](#) [!\[\]\(5243d3ed45b578cb43eef704a188139f_img.jpg\) No](#)

ODDFYIELD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el rendimiento de un valor negociable que tiene un primer período impar (corto o largo).

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
issue	Fecha de emisión del valor negociable.
first_coupon	Fecha del primer cupón del valor negociable.
rate	Tipo de interés del valor negociable.
pr	Precio del valor negociable.
redemption	Valor de canje del valor negociable por cada 100 USD de valor nominal.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

 Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Rendimiento del valor negociable.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- ODDFYIELD se calcula con un método iterativo. Usa el método de Newton, basado en la fórmula utilizada para la función ODDFPRICE. El rendimiento se cambia a través de 100 iteraciones hasta que el precio estimado con el rendimiento determinado se aproxima al precio. Vea ODDFPRICE para la fórmula que usa ODDFYIELD.
- Los parámetros settlement, maturity, issue y first_coupon se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:

- Los parámetros settlement, maturity, issue o first_coupon no son una fecha válida.
 - No se cumple que maturity > first_coupon > settlement > issue.
 - rate < 0.
 - El parámetro pr es menor o igual que 0.
 - El parámetro redemption es menor o igual que 0.
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción del argumento
11 de noviembre de 2008	Fecha de liquidación
1 de marzo de 2021	Fecha de vencimiento
15 de octubre de 2008	Fecha de emisión
1 de marzo de 2009	Fecha del primer cupón
5,75 %	Porcentaje del cupón
84,50	Price
100	Valor de canje
2	Frecuencia semestral
0	Base 30/360

La consulta DAX siguiente:

```
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)
}
```

Devuelve el rendimiento de un valor negociable que tiene un primer período impar (corto o largo), con las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
0,0772455415972989

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ODDLPRICE

Artículo • 20/10/2023

Devuelve el precio por cada 100 USD de valor nominal de un valor negociable que tiene un último período de cupón impar (corto o largo).

Sintaxis

DAX

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

Parámetros

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
last_interest	Fecha del último cupón del valor negociable.
rate	Tipo de interés del valor negociable.
yld	Rendimiento anual del valor negociable.
redemption	Valor de canje del valor negociable por cada 100 USD de valor nominal.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Basis	Day count basis
Omitido o 0	US (NASD) 30/360

Basis	Day count basis
1	Actual/actual
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Precio por cada 100 USD de valor nominal.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- Los parámetros settlement, maturity y last_interest se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement, maturity o last_interest no son una fecha válida.
 - maturity > settlement > last_interest is not satisfied.
 - rate < 0.
 - yld < 0.
 - El parámetro redemption es menor o igual que 0.
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La consulta DAX siguiente:

Data	Descripción del argumento
7 de febrero de 2008	Fecha de liquidación
15 de junio de 2008	Fecha de vencimiento
15 de octubre de 2007	Fecha del último interés
3,75 %	Porcentaje del cupón
4,05 %	Tipo de interés
100 USD	Valor de canje
2	Frecuencia semestral
0	Base 30/360

DAX

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

Devuelve el precio por cada 100 USD de valor nominal de un valor negociable que tiene un último período de cupón impar (corto o largo), con las condiciones especificadas anteriormente.

[Value]

99,8782860147213

Comentarios

¿Le ha resultado útil esta página?

Yes

No

ODDLYIELD

Artículo • 20/10/2023

Devuelve el rendimiento de un valor negociable que tiene un último período impar (corto o largo).

Sintaxis

DAX

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

Parámetros

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
last_interest	Fecha del último cupón del valor negociable.
rate	Tipo de interés del valor negociable.
pr	Precio del valor negociable.
redemption	Valor de canje del valor negociable por cada 100 USD de valor nominal.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Rendimiento del valor negociable.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de

enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.

- ODDLYIELD se calcula de la siguiente manera:

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

donde:

- A_i = número de días acumulados de i^{th} o último período de cuasicupón en un período impar contando hacia delante a partir de la fecha de los últimos intereses antes del canje.
- DC_i = número de días contados en i^{th} o último período de cuasicupón, tal y como lo delimita la duración del período de cupón real.
- NC = número de períodos de cuasicupón que caben en un período impar. Si este número contiene una fracción, aumentará al siguiente número entero.
- NL_i = duración normal en días de i^{th} o último período de cuasicupón en un período de cupón impar.
- Los parámetros settlement, maturity y last_interest se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement, maturity o last_interest no son una fecha válida.
 - maturity > settlement > last_interest is not satisfied.
 - rate < 0.
 - El parámetro pr es menor o igual que 0.
 - El parámetro redemption es menor o igual que 0.
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La consulta DAX siguiente:

Data	Descripción del argumento
20/4/2008	Fecha de liquidación
15/6/2008	Fecha de vencimiento
24/12/2007	Fecha del último interés
3,75 %	Porcentaje del cupón
99,875 USD	Precio
100 USD	Valor de canje
2	Frecuencia semestral
0	Base 30/360

DAX

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

Devuelve el rendimiento de un valor negociable que tiene un último período impar (corto o largo), con las condiciones especificadas anteriormente.

[Value]

0,0451922356291692

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

PDURATION

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de períodos que requiere una inversión para alcanzar un valor especificado.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
rate	Tipo de interés por período.
pv	Valor actual de la inversión.
fv	Valor futuro deseado de la inversión.

Valor devuelto

Número de períodos.

Comentarios

- PDURATION usa la ecuación siguiente:

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

- Se devuelve un error si ocurre lo siguiente:
 - El parámetro rate es menor o igual que 0.
 - El parámetro pv es menor o igual que 0.
 - El parámetro fv es menor o igual que 0.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

La consulta DAX siguiente:

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

Devuelve el número de años necesarios para que una inversión de 2000 USD, con un rendimiento anual del 2,5 %, alcance 2200 USD.

[Expandir tabla](#)

[Value]
3,85986616262266

Ejemplo 2

La consulta DAX siguiente:

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

Devuelve el número de meses necesarios para que una inversión de 1000 USD, con un rendimiento anual del 2,5 %, alcance 1200 USD.

[Expandir tabla](#)

[Value]
87,6054764193714

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PMT

Artículo • 20/10/2023

Calcula el pago de un préstamo basado en pagos constantes y un tipo de interés constante.

Sintaxis

DAX

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

Parámetros

Término	Definición
rate	Tipo de interés del préstamo.
nper	Número total de pagos del préstamo.
pv	Valor actual o importe total de una serie de pagos futuros; también se conoce como "capital".
fv	(Opcional) Valor futuro o saldo en efectivo que se quiere obtener después del último pago. Si el parámetro fv se omite, se da por hecho que es BLANK.
tipo	(Opcional) Número 0 o 1 que indica cuándo vencen los pagos. Si el parámetro type se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **type** acepta los valores siguientes:

Establecer type igual a	Si los pagos han vencido
Omitido o 0	Al final del período
1	Al inicio del período

Nota: Para obtener una descripción más completa de los argumentos de PMT, vea la función PV.

Valor devuelto

El importe de un pago único de préstamo.

Comentarios

- El pago que devuelve PMT incluye el capital y los intereses, pero no los impuestos, los pagos de reserva ni los honorarios asociados a menudo con los préstamos.
- Asegúrese de ser coherente con las unidades que se usan para especificar los parámetros rate y nper. Si realiza pagos mensuales en un préstamo de cuatro años con un tipo de interés anual del 12 %, use 0,12/12 para rate y 4*12 para nper. Si realiza pagos anuales en el mismo préstamo, use 0,12 en rate y 4 en nper.
- El parámetro type se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - nper < 1

Sugerencia: para averiguar el importe total pagado a lo largo de la duración del préstamo, multiplique el valor de PMT devuelto por nper.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

Ejemplo 1

Data	Descripción
8 %	Tipo de interés anual
10	Número de meses en los que se realizan pagos.
10 000 USD	Importe del préstamo

La consulta DAX siguiente:

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

Devuelve el importe de pago mensual, pagado al inicio del mes, para un préstamo con las condiciones especificadas anteriormente.

[Value]
-1030,16432717797

Nota: 1030,16432717797 es el pago por período. Como resultado, el importe total pagado a lo largo de la vida del préstamo es aproximadamente $1030,16 \times 10 = 10\,301,60$ USD. En otras palabras, se pagan aproximadamente 301,60 USD de intereses.

Ejemplo 2

Data	Descripción
6 %	Tipo de interés anual
18	Número de años en los que se realizan pagos.
50 000 USD	Importe del préstamo

La consulta DAX siguiente:

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

[Value]
-129,081160867991

Devuelve la cantidad que se tiene que ahorrar cada mes para disponer de 50 000 USD al cabo de 18 años, con las condiciones especificadas anteriormente.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

PPMT

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el pago del capital de una inversión durante un período determinado con base en pagos periódicos y constantes, y en un tipo de interés constante.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
rate	Tipo de interés del préstamo.
per	Especifica el período. Debe estar entre 1 y nper (incluido).
nper	Número total de períodos de pago en una anualidad.
pv	Valor actualizado: importe total de una serie de pagos futuros en el momento actual.
fv	(Opcional) Valor futuro o saldo en efectivo que se quiere obtener después del último pago. Si el parámetro fv se omite, se da por hecho que es BLANK.
tipo	(Opcional) Número 0 o 1 que indica cuándo vencen los pagos. Si el parámetro type se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **type** acepta los valores siguientes:

Expandir tabla

Establecer type igual a	Si los pagos han vencido
Omitido o 0	Al final del período
1	Al inicio del período

Nota: Para obtener una descripción más completa de los argumentos de PPMT, vea la función PV.

Valor devuelto

Pago de capital para un período determinado.

Comentarios

- Asegúrese de ser coherente con las unidades que se usan para especificar los parámetros rate y nper. Si realiza pagos mensuales en un préstamo de cuatro años con un tipo de interés anual del 12 %, use 0,12/12 para rate y 4*12 para nper. Si realiza pagos anuales en el mismo préstamo, use 0,12 en rate y 4 en nper.
- El parámetro type se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - per < 1 o per > nper
 - nper < 1
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

 Expandir tabla

Data	Descripción del argumento
10 %	Tipo de interés anual
2	Número de años del préstamo
2000 USD	Importe del préstamo

La consulta DAX siguiente:

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

Devuelve el pago de capital realizado en el primer mes para un préstamo con las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
-75,6231860083663

Ejemplo 2

 Expandir tabla

Data	Descripción del argumento
8 %	Tipo de interés anual
10	Número de años del préstamo
200 000 USD	Importe del préstamo

La consulta DAX siguiente:

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

Devuelve el pago de capital realizado en el décimo año para un préstamo con las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
-27598,0534624214

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PRICE

Artículo • 20/10/2023

Devuelve el precio por cada 100 USD de valor nominal de un valor negociable que paga intereses periódicos.

Sintaxis

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

Parámetros

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
rate	Tasa del cupón anual del valor negociable.
yld	Rendimiento anual del valor negociable.
redemption	Valor de canje del valor negociable por cada 100 USD de valor nominal.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Precio por cada 100 USD de valor nominal.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- Los parámetros settlement y maturity se truncan en enteros.
- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - rate < 0.
 - yld < 0.
 - El parámetro redemption es menor o igual que 0.
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Importante:

- Cuando N > 1 (N es el número de cupones pagaderos entre la fecha de liquidación y la fecha de canje), **PRICE** se calcula de la siguiente manera:

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

- Cuando N es igual a 1 (N es el número de cupones pagaderos entre la fecha de liquidación y la fecha de canje), **PRICE** se calcula de la siguiente manera:

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

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

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

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

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

donde:

- DSC = número de días desde la liquidación hasta la fecha del siguiente cupón.

- E = número de días del período de cupón en el que entra la fecha de liquidación.
- A = número de días desde el inicio del período de cupón hasta la fecha de liquidación.

Ejemplo

Data	Descripción del argumento
15/2/2008	Fecha de liquidación
15/11/2017	Fecha de vencimiento
5,75 %	Porcentaje del cupón semestral
6,50 %	Tipo de interés
100 USD	Valor de canje
2	Frecuencia semestral
0	Base 30/360

La consulta DAX siguiente:

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

Devuelve el precio de un bono para un bono con las condiciones especificadas anteriormente.

[Value]
94,6343616213221

Comentarios

¿Le ha resultado útil esta página? Yes No

PRICEDISC

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el precio por cada 100 USD de valor nominal de un valor negociable con descuento.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
discount	Tasa de descuento del valor negociable.
redemption	Valor de canje del valor negociable por cada 100 USD de valor nominal.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual

Basis	Day count basis
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Precio por cada 100 USD de valor nominal.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2018 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2018; la de liquidación, el 1 de julio de 2018; y la de vencimiento, el 1 de enero de 2048, 30 años después del 1 de enero de 2018, la fecha de la emisión.
- PRICEDISC se calcula de la siguiente manera:

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

donde:

- B = número de días en el año, en función de la base del año utilizada.
- DSM = número de días desde la liquidación hasta el vencimiento.
- Los parámetros settlement y maturity se truncan en enteros.
- El parámetro basis se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - El parámetro discount es menor o igual que 0.
 - El parámetro redemption es menor o igual que 0.
 - basis < 0 o basis > 4.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

[\[+\] Expandir tabla](#)

Data	Descripción del argumento
16/02/2008	Fecha de liquidación
01/03/2008	Fecha de vencimiento
5,25 %	Tasa de descuento porcentual
100 USD	Valor de canje
2	Base real/360

La consulta DAX siguiente:

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

Devuelve el precio del bono por cada 100 USD de valor nominal, para un bono con las condiciones especificadas anteriormente.

[\[+\] Expandir tabla](#)

[Value]
99,7958333333333

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PRICEMAT

Artículo • 20/10/2023

Devuelve el precio por cada 100 USD de valor nominal de un valor negociable que paga intereses al vencimiento.

Sintaxis

DAX

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

Parámetros

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
issue	Fecha de emisión del valor negociable.
rate	Tipo de interés del valor negociable en la fecha de emisión.
yld	Rendimiento anual del valor negociable.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360
3	Actual/365

Basis	Day count basis
4	European 30/360

Valor devuelto

Precio por cada 100 USD de valor nominal.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- PRICEMAT se calcula de la siguiente manera:

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

donde:

- B = número de días en el año, en función de la base del año utilizada.
- DSM = número de días desde la liquidación hasta el vencimiento.
- DIM = número de días desde la emisión hasta el vencimiento.
- A = número de días desde la emisión hasta la liquidación.
- Los parámetros settlement, maturity e issue se truncan en enteros.
- El parámetro basis se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement, maturity o issue no son una fecha válida.
 - No se cumple que maturity > settlement > issue.
 - rate < 0.
 - yld < 0.
 - basis < 0 o basis > 4.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La consulta DAX siguiente:

Data	Descripción
15/2/2008	Fecha de liquidación
13/4/2008	Fecha de vencimiento
11/11/2007	Fecha de emisión
6,10 %	Porcentaje del cupón semestral
6,10 %	Tipo de interés
0	Base 30/360

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

Devuelve el precio por cada 100 USD de valor nominal de un valor negociable con las condiciones especificadas anteriormente.

[Value]
99,9844988755569

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

PV

Artículo • 20/10/2023

Calcula el valor actual de un préstamo o una inversión, basado en un tipo de interés constante. Puede usar PV con pagos periódicos y constantes (como una hipoteca u otro préstamo), y con un valor futuro que sea el objetivo de la inversión.

Sintaxis

DAX

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

Parámetros

Término	Definición
rate	Tipo de interés por período. Por ejemplo, si obtiene un préstamo para la compra de un automóvil con un tipo de interés anual del 10 % y realiza pagos mensuales, el tipo de interés mensual es de 0,1/12 o 0,0083. En la fórmula, para rate escribiría 0,1/12 o 0,0083.
nper	Número total de períodos de pago en una anualidad. Por ejemplo, si obtiene un préstamo para un automóvil de cuatro años y realiza pagos mensuales, el préstamo tiene 4*12 (o 48) períodos. En la fórmula, para nper escribiría 48.
pmt	Pago realizado en cada período que no puede cambiar durante la vida útil de la anualidad. Normalmente, pmt incluye el capital y los intereses, pero no otros honorarios o impuestos. Por ejemplo, los pagos mensuales de un préstamo para un automóvil de cuatro años y 10 000 USD al 12 % son de 263,33 USD. En la fórmula, para pmt escribiría -263,33.
fv	(Opcional) Valor futuro o saldo en efectivo que se quiere obtener después del último pago. Si el parámetro fv se omite, se da por hecho que es BLANK. Por ejemplo, si quiere ahorrar 50 000 USD para pagar un proyecto especial en 18 años, 50 000 USD es el valor futuro. Después, se puede hacer una estimación conservadora con un tipo de interés y determinar la cantidad que debe ahorrar cada mes.
tipo	(Opcional) Número 0 o 1 que indica cuándo vencen los pagos. Si el parámetro type se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **type** acepta los valores siguientes:

Establecer type igual a	Si los pagos han vencido
Omitido o 0	Al final del período
1	Al inicio del período

Valor devuelto

Valor actual del préstamo o la inversión.

Comentarios

- Asegúrese de ser coherente con las unidades que se usan para especificar los parámetros rate y nper. Si realiza pagos mensuales en un préstamo de cuatro años con un interés anual del 12 %, use 0,12/12 en rate y 4*12 en nper. Si realiza pagos anuales en el mismo préstamo, use 0,12 en rate y 4 en nper.
- Las siguientes funciones se aplican a las anualidades:
 - CUMIPMT
 - CUMPRINC
 - FV
 - IPMT
 - PMT
 - PPMT
 - PV
 - RATE
 - XIRR
 - XNPV
- Una anualidad es una serie de pagos constantes en efectivo realizados durante un período continuo. Por ejemplo, un préstamo para un automóvil o una hipoteca son una anualidad. Para obtener más información, vea la descripción de las funciones de anualidad.
- En las funciones de anualidad, el efectivo que se paga, como un depósito de ahorro, se representa mediante un número negativo y el que se recibe, como un dividendo, se representa mediante un número positivo. Por ejemplo, un depósito de 1000 USD en el banco se representaría mediante el argumento -1000 si es el depositante y 1000 si es el banco.
- Un argumento financiero se resuelve en términos de los demás.

- Si rate no es 0, entonces:

$$pv \times (1 + \text{rate})^{\text{nper}} + \text{pmt}(1 + \text{rate} \times \text{type}) \times \left(\frac{(1 + \text{rate})^{\text{nper}} - 1}{\text{rate}} \right) + \text{fv} = 0$$

- Si rate es 0, entonces:

$$(\text{pmt} \times \text{nper}) + \text{pv} + \text{fv} = 0$$

- El parámetro type se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - nper < 1 i está en blanco
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Data	Descripción
500 USD	Dinero pagado de la anualidad de un seguro al final de cada mes.
8 %	Tipo de interés ganado sobre el dinero pagado.
20	Años que se pagará el dinero.

La consulta DAX siguiente:

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

Devuelve el valor actual de una anualidad con las condiciones especificadas anteriormente.

[Value]
-59777,1458511878

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

RATE

Artículo • 20/10/2023

Devuelve el tipo de interés por período de una anualidad. RATE se calcula mediante iteración y puede tener cero o más soluciones. Si los resultados sucesivos de RATE no convergen en 0,0000001 después de 20 iteraciones, devuelve un error.

Sintaxis

DAX

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

Parámetros

Término	Definición
nper	Número total de períodos de pago en una anualidad.
pmt	Pago realizado en cada período; no puede cambiar durante la vida útil de la anualidad. Normalmente, pmt incluye el capital y los intereses, pero no otros honorarios o impuestos.
pv	Valor actualizado: importe total de una serie de pagos futuros en el momento actual.
fv	(Opcional) Valor futuro o saldo en efectivo que se quiere obtener después del último pago. Si se omite fv, se supone que es 0 (el valor futuro de un préstamo, por ejemplo, es 0).
tipo	(Opcional) Número 0 o 1 que indica cuándo vencen los pagos. Si el parámetro type se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.
guess	(Opcional) Su estimación de cuál será la tasa. <ul style="list-style-type: none">- Si se omite, se da por hecho que es un 10 %.- Si el valor de RATE no converge, pruebe con valores diferentes para guess. Normalmente, RATE converge si guess oscila entre 0 y 1.

El parámetro **type** acepta los valores siguientes:

Establecer type igual a	Si los pagos han vencido
Omitido o 0	Al final del período

Establecer type igual a	Si los pagos han vencido
1	Al inicio del período

Valor devuelto

Tipo de interés por período.

Comentarios

- Asegúrese de ser coherente con las unidades que se usan para especificar los parámetros guess y nper. Si realiza pagos mensuales en un préstamo de cuatro años con un interés anual del 12 %, use 0,12/12 en guess y 4*12 en nper. Si realiza pagos anuales en el mismo préstamo, use 0,12 en guess y 4 en nper.
- El parámetro type se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - El parámetro nper es menor o igual que 0.
 - RATE no converge en 0,0000001 después de 20 iteraciones.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

Data	Descripción
4	Años del préstamo
-200	Pago mensual
8000	Importe del préstamo

Ejemplo 1

La consulta DAX siguiente:

```
DAX
```

```
EVALUATE
{
```

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

Devuelve la tasa mensual del préstamo usando las condiciones especificadas anteriormente.

[Value]

0,00770147248820137

Ejemplo 2

La consulta DAX siguiente:

DAX

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

Devuelve la tasa anual del préstamo con las condiciones especificadas anteriormente.

[Value]

0,0924176698584164

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

RECEIVED

Artículo • 20/10/2023

Devuelve el importe recibido al vencimiento para un valor negociable invertido en su totalidad.

Sintaxis

DAX

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

Parámetros

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
investment	Importe invertido en el valor negociable.
discount	Tasa de descuento del valor negociable.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Importe recibido al vencimiento.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- RECEIVED se calcula de la siguiente manera:

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

donde:

- B = número de días en un año, en función de la base del año utilizada.
- DIM = número de días desde la emisión hasta el vencimiento.
- Los parámetros settlement y maturity se truncan en enteros.
- El parámetro basis se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - El parámetro investment es menor o igual que 0.
 - El parámetro discount es menor o igual que 0.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La consulta DAX siguiente:

Data	Descripción
15-Feb-08	Fecha de liquidación (emisión)
15-May-08	Fecha de vencimiento
1 000 000,00 USD	Inversión
5,75 %	Tasa de descuento porcentual
2	Base real/360

DAX

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

Devuelve el importe total que se va a recibir al vencimiento, para un bono con las condiciones especificadas anteriormente.

[Value]

1014584,6544071

Comentarios

¿Le ha resultado útil esta página?

Yes

No

RRI

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve un tipo de interés equivalente para el crecimiento de una inversión.

Sintaxis

DAX
<code>RRI(<nper>, <pv>, <fv>)</code>

Parámetros

Expandir tabla

Término	Definición
nper	Número de períodos de la inversión.
pv	Valor actual de la inversión.
fv	Valor futuro de la inversión.

Valor devuelto

Tipo de interés equivalente.

Comentarios

- RRI devuelve el tipo de interés dada con nper (número de períodos), pv (valor actual) y fv (valor futuro), calculada mediante la ecuación siguiente:

$$\left(\frac{fv}{pv} \right)^{\left(\frac{1}{nper} \right)} - 1$$

- Se devuelve un error si ocurre lo siguiente:
 - El parámetro nper es menor o igual que 0.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
10 000 USD	Valor actual
21 000 USD	Valor futuro
4	Años invertidos

La consulta DAX siguiente:

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

Devuelve un tipo de interés equivalente para el crecimiento de una inversión con las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
0,0155771057566627

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SLN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la amortización lineal de un activo para un período.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
cost	Costo inicial del activo.
salvage	Valor al final de la amortización (también conocido como valor residual del activo).
vida	Número de períodos durante los cuales se produce la amortización del activo (también conocido como "vida útil del activo").

Valor devuelto

La amortización lineal para un período.

Comentarios

- Se devuelve un error si ocurre lo siguiente:
El parámetro life es igual a 0.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

[Expandir tabla](#)

Data	Descripción
30 000 USD	Coste
7500 USD	Valor residual
10	Años de vida útil

La consulta DAX siguiente:

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

Devuelve la provisión de la amortización anual con las condiciones especificadas anteriormente.

[Expandir tabla](#)

[Value]
2250

Comentarios

¿Le ha resultado útil esta página?

[Sí](#)[No](#)

SYD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la amortización por método de anualidades de un activo para un período especificado.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
cost	Costo inicial del activo.
salvage	Valor al final de la amortización (también conocido como valor residual del activo).
vida	Número de períodos durante los cuales se produce la amortización del activo (también conocido como "vida útil del activo").
per	Período. Debe usar las mismas unidades que el parámetro life. Debe estar entre 1 y la vida del activo (incluida).

Valor devuelto

La amortización por método de anualidades para un período especificado.

Comentarios

- SYD se calcula de la siguiente manera:

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

- Se devuelve un error si ocurre lo siguiente:
 - life < 1.
 - per < 1 o per > life.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

 Expandir tabla

Data	Descripción
30 000 USD	Costo inicial
7500 USD	Valor residual
10	Duración en años

Ejemplo 1

La consulta DAX siguiente:

DAX
<pre> EVALUATE { SYD(30000.00, 7500.00, 10, 1) } </pre>

Devuelve la provisión de amortización por método de anualidades del primer año, dadas las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
4090,90909090909

Ejemplo 2

La consulta DAX siguiente:

--

DAX

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

Devuelve la provisión de amortización por método de anualidades para el décimo año (último), dadas las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
409,090909090909

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TBILLEQ

Artículo • 20/10/2023

Devuelve el rendimiento equivalente de un bono de una factura de tesorería.

Sintaxis

DAX

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

Parámetros

Término	Definición
settlement	Fecha de liquidación de la factura de tesorería. La fecha de liquidación del valor negociable es la fecha después de la fecha de emisión en la que se negocia la factura de tesorería del comprador.
maturity	Fecha de vencimiento de la factura de tesorería. La fecha de vencimiento es la fecha en la que expira la factura de tesorería.
discount	Tasa de descuento de la factura de tesorería.

Valor devuelto

El rendimiento equivalente de un bono de la factura de tesorería.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- TBILLEQ se calcula de la forma siguiente:

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

donde:

- DSM es el número de días comprendidos entre la liquidación y el vencimiento, calculados según la base de que un año tiene 360 días.
- Los parámetros settlement y maturity se truncan en enteros.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity, o maturity es más de un año después de settlement.
 - El parámetro discount es menor o igual que 0.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Data	Descripción
3/31/2008	Fecha de liquidación
1/6/2008	Fecha de vencimiento
9,14 %	Tasa de descuento porcentual

La consulta DAX siguiente:

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

Devuelve el rendimiento equivalente de un bono de una factura de tesorería con las condiciones especificadas anteriormente.

[Value]
0,094151493565943

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

TBILLPRICE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el precio por cada 100 USD de valor nominal de una factura de tesorería.

Sintaxis

DAX
<code>TBILLPRICE(<settlement>, <maturity>, <discount>)</code>

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación de la factura de tesorería. La fecha de liquidación del valor negociable es la fecha después de la fecha de emisión en la que se negocia la factura de tesorería del comprador.
maturity	Fecha de vencimiento de la factura de tesorería. La fecha de vencimiento es la fecha en la que expira la factura de tesorería.
discount	Tasa de descuento de la factura de tesorería.

Valor devuelto

Precio de la factura de tesorería por cada 100 USD de valor nominal.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- TBILLPRICE se calcula de la siguiente manera:

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

donde:

- DSM = número de días desde la liquidación hasta el vencimiento, excluida cualquier fecha de vencimiento superior a un año natural después de la fecha de liquidación.
- Los parámetros settlement y maturity se truncan en enteros.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity, o maturity es más de un año después de settlement.
 - El parámetro discount es menor o igual que 0.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	Descripción
3/31/2008	Fecha de liquidación
1/6/2008	Fecha de vencimiento
9,0 %	Tasa de descuento porcentual

La consulta DAX siguiente:

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

Devuelve el precio de la factura de tesorería por cada 100 USD de valor nominal, dadas las condiciones especificadas anteriormente.

 Expandir tabla

[Value]

98,45

Comentarios

¿Le ha resultado útil esta página?



TBILLYIELD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el rendimiento de una factura de tesorería.

Sintaxis

DAX
<pre>TBILLYIELD(<settlement>, <maturity>, <pr>)</pre>

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación de la factura de tesorería. La fecha de liquidación del valor negociable es la fecha después de la fecha de emisión en la que se negocia la factura de tesorería del comprador.
maturity	Fecha de vencimiento de la factura de tesorería. La fecha de vencimiento es la fecha en la que expira la factura de tesorería.
pr	Precio de la factura de tesorería por cada 100 USD de valor nominal.

Valor devuelto

El rendimiento de la factura de tesorería.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- TBILLYIELD se calcula de la siguiente manera:

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

donde:

- DSM = número de días desde la liquidación hasta el vencimiento, excluida cualquier fecha de vencimiento superior a un año natural después de la fecha de liquidación.
- Los parámetros settlement y maturity se truncan en enteros.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity, o maturity es más de un año después de settlement.
 - El parámetro pr es menor o igual que 0.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La consulta DAX siguiente:

 Expandir tabla

Data	Descripción
3/31/2008	Fecha de liquidación
1/6/2008	Fecha de vencimiento
98,45 USD	Precio por cada 100 USD de valor nominal

DAX
<pre> EVALUATE { TBILLYIELD(DATE(2008,3,31), DATE(2008,6,1), 98.45) } </pre>

Devuelve el rendimiento de una factura de tesorería con las condiciones especificadas anteriormente.

 Expandir tabla

[Value]

0,0914169629253426

Comentarios

¿Le ha resultado útil esta página?



Sí



No

VDB

Artículo • 20/10/2023

Devuelve la amortización de un activo en un período específico, incluidos los períodos parciales, con el método de amortización por doble disminución de saldo u otro método que se especifique. VDB significa disminución variable de saldo.

Sintaxis

DAX

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

Parámetros

Término	Definición
cost	Costo inicial del activo.
salvage	Valor al final de la amortización (también conocido como valor residual del activo). Este valor puede ser 0.
vida	Número de períodos durante los cuales se amortiza el activo (también conocido como vida útil del activo).
start_period	Período de inicio para el que se quiere calcular la amortización. El parámetro start_period debe usar las mismas unidades que life. Debe estar entre 1 y la vida del activo (incluida).
end_period	Período de inicio para el que se quiere calcular la amortización. El parámetro end_period debe usar las mismas unidades que life. Debe estar entre start_period y life (incluidos).
factor	(Opcional) Tasa a la que disminuye el saldo. Si el argumento factor se omite, se interpretará como 2 (método de amortización por doble disminución de saldo). Cambie el parámetro factor si no quiere usar el método de amortización por doble disminución de saldo. Para obtener una descripción del método de amortización por doble disminución de saldo, vea DDB.
no_switch	(Opcional) Valor lógico que especifica si se debe cambiar a la amortización lineal cuando esta sea mayor que el cálculo de la disminución de saldo. Si se omite, se da por hecho que es "false". - Si no_switch se evalúa como "true", VDB no cambiará a amortización lineal cuando esta es mayor que el cálculo de la disminución de saldo.

Término	Definición
	<ul style="list-style-type: none"> - Si no_switch se evalúa como "false" o se omite, VDB cambiará a amortización lineal cuando esta sea mayor que el cálculo de la disminución de saldo.

Valor devuelto

Amortización en el período especificado.

Comentarios

- Se devuelve un error si ocurre lo siguiente:
 - cost < 0.
 - salvage < 0.
 - life < 1.
 - start_period < 1 o start_period > end_period.
 - end_period < start_period o end_period > life.
 - factor < 0.
 - El parámetro no_switch no se evalúa como "true" o "false".
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

Data	Descripción
2400	Costo inicial
300	Valor residual
10	Duración en años

Ejemplo 1

La consulta DAX siguiente:

DAX

```
EVALUATE
{
```

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

Devuelve la amortización del primer día de un activo con un valor del parámetro factor de 2.

[Value]
1,31506849315068

Ejemplo 2

La consulta DAX siguiente:

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

Devuelve la amortización de un activo entre los meses 6th y 18th. Este cálculo utiliza un valor del parámetro factor de 3.

[Value]
540,185558199698

Ejemplo 3

La consulta DAX siguiente:

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

Devuelve la amortización de un activo en el primer año fiscal de su propiedad, suponiendo que las leyes fiscales limiten al 150 % la amortización de la disminución de saldo. El activo se compra en la mitad del primer trimestre del año fiscal.

[Value]

315

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

XIRR

Artículo • 18/12/2023

Devuelve la tasa interna de devolución de una programación de flujos de efectivo que no es necesariamente periódica.

Sintaxis

DAX

```
XIRR(<table>, <values>, <dates>, [, <guess>[, <alternateResult>]]])
```

Parámetros

 Expandir tabla

Término	Definición
tabla	Tabla para la que se deben calcular las expresiones de valores y fechas.
valores	Expresión que devuelve el valor del flujo de efectivo de cada fila de la tabla.
fechas	Expresión que devuelve la fecha del flujo de efectivo de cada fila de la tabla.
guess	(Opcional) Estimación inicial de la tasa interna de devolución. Si se omite, se usa la estimación predeterminada de 0,1.
alternateResult	(Opcional) Valor devuelto en lugar de un error cuando no se puede determinar una solución.

Valor devuelto

Tasa interna de devolución de las entradas especificadas. Si el cálculo no devuelve un resultado válido, se devuelve un error o un valor especificado como alternateResult.

Notas

- El valor se calcula como la tasa que satisface la siguiente función:

$$0 = \sum_{j=1}^N \frac{P_j}{(1 + \text{rate})^{\frac{d_j - d_1}{365}}}$$

Donde:

- P_j es el pago de j^{th}
 - d_j es la fecha de pago de j^{th}
 - d_1 es la fecha del primer pago
- La serie de valores de flujo de efectivo debe contener al menos un número positivo y un número negativo.
 - Evite el uso de las funciones ISERROR o IFERROR para capturar un error devuelto por XIRR. Si algunas entradas de la función pueden dar lugar a un error sin solución, proporcionar un parámetro alternateResult es la manera más confiable y que mejor funciona para administrar el error.
 - Para obtener más información sobre el uso del parámetro alternateResult, consulte este [vídeo](#).
 - Cuando el valor absoluto del pago inicial es pequeño, es probable que el cálculo no devuelva un resultado válido.
 - Evite incluir 0 valores en los pagos. No afectan al resultado final y el uso de 0 como pago inicial siempre producirá un error en el cálculo de XIRR().
 - Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula siguiente calcula la tasa interna de devolución de la tabla CashFlows:

DAX

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

[] Expandir tabla

Date	Payment
1/1/2014	-10 000
1/3/2014	2750
30/10/2014	4250
15/2/2015	3250

Date	Payment
4/1/2015	2750

Tasa de devolución = 37,49 %

Comentarios

¿Le ha resultado útil esta página?



XNPV

Artículo • 28/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el valor presente de una programación de flujos de efectivo que no es necesariamente periódica.

Sintaxis

DAX

`XNPV(<table>, <values>, <dates>, <rate>)`

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla para la que se deben calcular las expresiones de valores y fechas.
valores	Expresión que devuelve el valor del flujo de efectivo de cada fila de la tabla.
fechas	Expresión que devuelve la fecha del flujo de efectivo de cada fila de la tabla.
rate	Tarifa de descuento que se aplica al flujo de efectivo de cada fila de la tabla.

Valor devuelto

Valor neto actual.

Notas

- El valor se calcula como la suma siguiente:

$$\sum_{j=1}^N \frac{P_j}{(1 + \text{rate})^{\frac{d_j - d_1}{365}}}$$

Donde:

- P_j es el pago de j^{th}

- d_j es la fecha de pago de j^{th}
- d_1 es la fecha del primer pago
- La serie de valores de flujo de efectivo debe contener al menos un número positivo y un número negativo.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

A continuación se calcula el valor actual de la tabla CashFlows:

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

 Expandir tabla

Date	Payment
1/1/2014	-10 000
1/3/2014	2750
30/10/2014	4250
15/2/2015	3250
4/1/2015	2750

Valor actual = 2086,65

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

YIELD

Artículo • 20/10/2023

Devuelve el rendimiento de un valor negociable que paga intereses periódicos. Use YIELD para calcular el rendimiento de los bonos.

Sintaxis

DAX

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

Parámetros

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
rate	Tasa del cupón anual del valor negociable.
pr	Precio del valor negociable por cada 100 USD de valor nominal.
redemption	Valor de canje del valor negociable por cada 100 USD de valor nominal.
frequency	Número de cupones pagaderos al año. Para pagos anuales, frequency = 1; para semestrales, frequency = 2; para trimestrales, frequency = 4.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual

Basis	Day count basis
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

Rendimiento del valor negociable.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- Si hay un período de cupón o menos hasta el canje, YIELD se calcula de la manera siguiente:

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

donde:

- A = número de días desde el inicio del período de cupón hasta la fecha de liquidación (días acumulados).
- DSR = número de días desde la fecha de liquidación hasta la fecha de canje.
- E = número de días del período del cupón.
- Si hay más de un período de cupón hasta el canje, YIELD se calcula a través de cien iteraciones. La resolución usa el método de Newton, basado en la fórmula utilizada para la función PRICE. El rendimiento se cambia hasta el precio estimado dado que el rendimiento es cercano al precio.
- Los parámetros settlement y maturity se truncan en enteros.

- Los parámetros frequency y basis se redondean al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - rate < 0.
 - El parámetro pr es menor o igual que 0.
 - El parámetro redemption es menor o igual que 0.
 - El parámetro frequency es cualquier número distinto de 1, 2 o 4.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Data	Descripción
15-Feb-08	Fecha de liquidación
15-Nov-16	Fecha de vencimiento
5,75 %	Porcentaje del cupón
95,04287	Price
100 USD	Valor de canje
2	Frecuencia semestral (ver más arriba)
0	Base 30/360 (vea más arriba)

La consulta DAX siguiente:

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

Devuelve el rendimiento de un bono con las condiciones especificadas anteriormente.

[Value]
0,0650000068807314

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

YIELDDISC

Artículo • 28/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el rendimiento anual de un valor negociable con descuento.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
pr	Precio del valor negociable por cada 100 USD de valor nominal.
redemption	Valor de canje del valor negociable por cada 100 USD de valor nominal.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360
1	Actual/actual

Basis	Day count basis
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

El rendimiento anual.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- Los parámetros settlement y maturity se truncan en enteros.
- El parámetro basis se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement o maturity no son una fecha válida.
 - El parámetro settlement es mayor o igual que maturity.
 - El parámetro pr es menor o igual que 0.
 - El parámetro redemption es menor o igual que 0.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Data	-
16-Feb-08	Fecha de liquidación
1-Mar-08	Fecha de vencimiento
99,795	Price
100 USD	Valor de canje
2	Base real/360

La consulta DAX siguiente:

DAX
<pre>EVALUATE { YIELDDISC(DATE(2008,2,16), DATE(2008,3,1), 99.795, 100, 2) }</pre>

Devuelve el rendimiento anual del valor negociable, dadas las condiciones especificadas anteriormente.

 Expandir tabla

[Value]
0,0528225719868583

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

YIELDMAT

Artículo • 28/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el rendimiento anual de un valor negociable que paga intereses al vencimiento.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
settlement	Fecha de liquidación del valor negociable. La fecha de liquidación del valor negociable es la posterior a la de emisión, cuando el valor negociable se transfiere al comprador.
maturity	Fecha de vencimiento del valor negociable. La fecha de vencimiento es la fecha en la que expira el valor negociable.
issue	Fecha de emisión del valor negociable.
rate	Tipo de interés del valor negociable en la fecha de emisión.
pr	Precio del valor negociable por cada 100 USD de valor nominal.
basis	(Opcional) Tipo de base en que deben contarse los días. Si el parámetro basis se omite, se da por hecho que es 0. Los valores aceptados se muestran debajo de esta tabla.

El parámetro **basis** acepta los valores siguientes:

Expandir tabla

Basis	Day count basis
Omitido o 0	US (NASD) 30/360

Basis	Day count basis
1	Actual/actual
2	Real/360
3	Actual/365
4	European 30/360

Valor devuelto

El rendimiento anual.

Comentarios

- Las fechas se almacenan como números de serie secuenciales para que se puedan usar en los cálculos. En DAX, el 30 de diciembre de 1899 es el día 0 y el 1 de enero de 2008 es el 39 448 porque es 39 448 días después del 30 de diciembre de 1899.
- La fecha de liquidación es la fecha en la que un comprador adquiere un cupón, por ejemplo, un bono. La fecha de vencimiento es la fecha en la que expira el cupón. Por ejemplo, supongamos que un bono de 30 años se emite el 1 de enero de 2008 y lo adquiere un comprador seis meses más tarde. La fecha de emisión sería el 1 de enero de 2008; la de liquidación, el 1 de julio de 2008; y la de vencimiento, el 1 de enero de 2038, 30 años después del 1 de enero de 2008, la fecha de la emisión.
- Los parámetros settlement, maturity e issue se truncan en enteros.
- El parámetro basis se redondea al entero más cercano.
- Se devuelve un error si ocurre lo siguiente:
 - Los parámetros settlement, maturity o issue no son una fecha válida.
 - No se cumple que maturity > settlement > issue.
 - rate < 0.
 - El parámetro pr es menor o igual que 0.
 - basis < 0 o basis > 4.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

[Expandir tabla](#)

Data	Descripción
15-Mar-08	Fecha de liquidación
3-Nov-08	Fecha de vencimiento
8-Nov-07	Fecha de emisión
6,25 %	Porcentaje del cupón semestral
100,0123	Price
0	Base 30/360 (vea más arriba)

La consulta DAX siguiente:

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

Devuelve el rendimiento de un valor negociable con las condiciones especificadas anteriormente.

[Expandir tabla](#)

[Value]
0,0609543336915387

Comentarios

¿Le ha resultado útil esta página?

[Sí](#)[No](#)

Funciones de información

Artículo • 20/10/2023

Las funciones de información de DAX examinan la celda o fila que se proporciona como argumento e indican si el valor coincide con el tipo esperado. Por ejemplo, la función ISERROR devuelve TRUE si el valor al que se hace referencia contiene un error.

En esta categoría

Función	Descripción
COLUMNSTATISTICS	Devuelve una tabla de estadísticas referente a cada columna de cada tabla del modelo.
CONTAINS	Devuelve true si los valores de todas las columnas a las que se hace referencia existen o están contenidos en esas columnas; de lo contrario, la función devuelve false.
CONTAINSROW	Devuelve TRUE si una fila de valores existe o está incluida en una tabla; de lo contrario, devuelve FALSE.
CONTAINSSTRING	Devuelve TRUE o FALSE para indicar si una cadena contiene otra cadena.
CONTAINSSTRINGEXACT	Devuelve TRUE o FALSE para indicar si una cadena contiene otra cadena.
CUSTOMDATA	Devuelve el contenido de la propiedad CustomData de la cadena de conexión.
HASONEFILTER	Devuelve TRUE si el número de valores filtrados directamente en <i>columnName</i> es uno; de lo contrario, devuelve FALSE.
HASONEVALUE	Devuelve TRUE si el contexto de <i>columnName</i> se ha filtrado a un solo valor distinto. De lo contrario, es FALSE.
ISAFTER	Una función booleana que emula el comportamiento de una cláusula "Fecha de inicio" y devuelve true para una fila que cumple todos los parámetros de la condición.
ISBLANK	Comprueba si un valor está en blanco y devuelve TRUE o FALSE.
ISCROSSFILTERED	Devuelve TRUE si se está filtrando <i>columnName</i> u otra columna de la misma tabla o de una relacionada.

Función	Descripción
ISEMPTY	Comprueba si una tabla está vacía.
ISERROR	Comprueba si un valor es un error y devuelve TRUE o FALSE.
ISEVEN	Devuelve TRUE si el número es par o FALSE si el número es impar.
ISFILTERED	Devuelve TRUE si <i>columnName</i> se está filtrando directamente.
ISINSCOPE	Devuelve true si la columna especificada es el nivel de una jerarquía de niveles.
ISLOGICAL	Comprueba si un valor es lógico (TRUE o FALSE), y devuelve TRUE o FALSE.
ISNONTEXT	Comprueba si un valor no es de texto (las celdas en blanco no son texto) y devuelve TRUE o FALSE.
ISNUMBER	Comprueba si un valor es un número y devuelve TRUE o FALSE.
ISODD	Devuelve TRUE si el número es impar o FALSE si es par.
ISONORAFTER	Una función booleana que emula el comportamiento de una cláusula "Fecha de inicio" y devuelve true para una fila que cumple todos los parámetros de la condición.
ISSELECTEDMEASURE	Se usa en expresiones para que los elementos de cálculo determinen que la medida que está en contexto es una de las especificadas en la lista de medidas.
ISSUBTOTAL	Crea otra columna en una expresión SUMMARIZE que devuelve "true" si la fila contiene valores de subtotal de la columna especificada como argumento; de lo contrario, devuelve "false".
ISTEXT	Comprueba si un valor es texto y devuelve TRUE o FALSE.
NONVISUAL	Marca un filtro de valores en una expresión SUMMARIZECOLUMNS como no visual.
SELECTEDMEASURE	Se usa en expresiones para que los elementos de cálculo hagan referencia a la medida que está en contexto.
SELECTEDMEASUREFORMATSTRING	Se usa en expresiones para que los elementos de cálculo recuperen la cadena de formato de la medida que está en contexto.

Función	Descripción
SELECTEDMEASURENAME	Se usa en expresiones para que los elementos de cálculo determinen por nombre la medida que está en contexto.
USERCULTURE	Devuelve la configuración regional del usuario actual.
USERNAME	Devuelve el nombre de dominio y el nombre de usuario de las credenciales proporcionadas al sistema en el momento de la conexión.
USEROBJECTID	Devuelve el identificador de objeto o SID del usuario actual.
USERPRINCIPALNAME	Devuelve el nombre principal de usuario.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

COLUMNSTATISTICS

Artículo • 24/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla de estadísticas referente a cada columna de cada tabla del modelo.

Sintaxis

DAX

```
COLUMNSTATISTICS ()
```

Parámetros

Esta función no toma ningún parámetro.

Valor devuelto

Tabla de estadísticas. Cada fila de esta tabla representa una columna diferente en el modelo. Las columnas de la tabla incluyen lo siguiente:

- **Nombre de la tabla:** la tabla de la columna actual.
- **Nombre de la columna:** el nombre de la columna actual.
- **Mínimo:** el valor mínimo que se encuentra en la columna actual.
- **Máximo:** el valor máximo que se encuentra en la columna actual.
- **Cardinalidad:** el número de valores distintos que se encuentran en la columna actual.
- **Longitud máxima:** la longitud de la cadena más larga que se encuentra en la columna actual (solo se aplica a las columnas de cadena).

Observaciones

- Las columnas con un estado de error y las de las tablas calculadas del ámbito de consulta no aparecen en la tabla de resultados.
- Si se aplica un filtro del contexto de filtro a COLUMNSTATISTICS(), se devuelve un error.
- Para las columnas con tipo binario, las estadísticas Min y Max tendrán valores BLANK.

Ejemplo

Los ejemplos de este artículo se pueden usar con el modelo de ejemplo de Power BI Desktop de Adventure Works DW 2020. Para obtener el modelo, vea [Modelo de ejemplo de DAX](#).

La consulta DAX siguiente:

```
DAX

DEFINE
    TABLE FilteredProduct =
        FILTER (
            Product,
            [Color] == "Blue"
        )
    COLUMN Customer[Location] = [State-Province] & " " & [Country-Region]

EVALUATE
COLUMNSTATISTICS ()
```

Devuelve una tabla con estadísticas relativas a todas las columnas de todas las tablas del modelo. La tabla también incluye estadísticas para la columna calculada del ámbito de consulta, Customer[Location]. Pero la tabla no incluye las columnas de la tabla calculada del ámbito de consulta, FilteredProduct.

Table Name	Column Name	Min	Max	Cardinality	Max Length
Customer	RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61			18485	
Customer	CustomerKey	-1	29483	18485	
Customer	Customer ID	[Not Applicable]	AW00029483	18485	16
Customer	Customer	[Not Applicable]	Zoe Watson	18401	26
Customer	City	[Not Applicable]	York	270	21
Customer	State-Province	[Not Applicable]	Yveline	54	19
Customer	Country-Region	[Not Applicable]	United States	7	16
Customer	Postal Code	[Not Applicable]	YO15	324	16
Customer	Location	[Not Applicable] [Not Applicable]	Yveline France	54	33
Date	RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61			1461	
Date	DateKey	20170701	20210630	1461	
Date	Date	7/1/2017 12:00:00 AM	6/30/2021 12:00:00 AM	1461	
Date	Fiscal Year	FY2018	FY2021	4	6
Date	Fiscal Quarter	FY2018 Q1	FY2021 Q4	16	9
Date	Month	2017 Aug	2021 May	48	8
Date	MonthKey	201707	202106	48	
Date	Full Date	2017 Aug, 01	2021 May, 31	1461	12
Sales Territory	RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61			11	
Sales Territory	SalesTerritoryKey	1	11	11	
Sales Territory	Review	Australia	United Kingdom	11	

Contenido relacionado

[Contexto de filtro](#)

[Función CALCULATETABLE](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CONTAINS

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve true si los valores de todas las columnas a las que se hace referencia existen o están incluidos en esas columnas; de lo contrario, la función devuelve false.

Sintaxis

DAX

```
CONTAINS(<table>, <columnName>, <value>[, <columnName>, <value>]...)
```

Parámetros

Expandir tabla

Término	Definición
tabla	Cualquier expresión DAX que devuelve una tabla de datos.
columnName	Nombre de una columna existente, con la sintaxis de DAX estándar. No puede ser una expresión.
valor	Cualquier expresión DAX que devuelva un único valor escalar que se va a buscar en <i>columnName</i> . La expresión se va a evaluar exactamente una vez antes de pasarse a la lista de argumentos.

Valor devuelto

Un valor de **TRUE** si cada *value* especificado se puede encontrar en el *columnName* correspondiente o está incluido en esas columnas; de lo contrario, la función devuelve **FALSE**.

Notas

- Los argumentos *columnName* y *value* deben ir en parejas; de lo contrario, se devuelve un error.

- *columnName* debe pertenecer a la *table* especificada o a una tabla relacionada con *table*.
- Si *columnName* hace referencia a una columna de una tabla relacionada, debe ser completo; de lo contrario, se devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se crea una medida que indica si ha habido ventas por Internet del producto 214 y al cliente 11185 al mismo tiempo.

DAX

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

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Función CONTAINSROW

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve TRUE si hay al menos una fila con valores especificados en todas las columnas.

Sintaxis

DAX

```
CONTAINSROW(<Table>, <Value> [, <Value> [, ...] ] )
```

Parámetros

[] Expandir tabla

Término	Definición
Tabla	Una tabla que se va a probar.
Value	Cualquier expresión DAX válida que devuelve un valor escalar.

Valor devuelto

TRUE o FALSE.

Comentarios

- Excepto en la sintaxis, el operador IN y la función CONTAINSROW son funcionalmente equivalentes.

DAX

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

- El número de valores scalarExprN debe coincidir con el número de columnas de tableExpr.

- NOT IN no es un operador en DAX. Para realizar la negación lógica del operador IN, ponga NOT delante de la expresión completa. Por ejemplo, NOT [Color] IN {"Rojo", "Amarillo", "Azul"}.
- A diferencia del operador =, el operador IN y la función CONTAINSROW realizan una comparación estricta. Por ejemplo, el valor BLANK no coincide con 0.

Ejemplos

Los ejemplos de este artículo se pueden usar con el modelo de ejemplo de Power BI Desktop de Adventure Works DW 2020. Para obtener el modelo, vea [Modelo de ejemplo de DAX](#).

Ejemplo 1

Las consultas DAX siguientes:

```
DAX

EVALUATE
FILTER (
    ALL ( Product[Color] ),
    ( [Color] )
        IN {
            "Red",
            "Yellow",
            "Blue"
        }
)
ORDER BY [Color]
```

y

```
DAX

EVALUATE
FILTER (
    ALL ( Product[Color] ),
    CONTAINSROW (
        {
            "Red",
            "Yellow",
            "Blue"
        },
        [Color]
    )
)
```

```
)  
ORDER BY [Color]
```

Devuelva la siguiente tabla con una sola columna:

 Expandir tabla

[Color]
Azul
Rojo
Amarillo

Ejemplo 2

Las siguientes consultas DAX equivalentes:

```
DAX  
  
EVALUATE  
FILTER (  
    ALL ( Product[Color] ),  
    NOT [Color]  
        IN {  
            "Red",  
            "Yellow",  
            "Blue"  
        }  
)  
ORDER BY [Color]
```

y

```
DAX  
  
EVALUATE  
FILTER (  
    ALL ( Product[Color] ),  
    NOT CONTAINSROW (  
        {  
            "Red",  
            "Yellow",  
            "Blue"  
        },  
        [Color]  
)
```

```
)  
ORDER BY [Color]
```

Devuelva la siguiente tabla con una sola columna:

 Expandir tabla

[Color]
Negro
Gris
Multi
NA
Plata
Plata\Negro
Blanco

Contenido relacionado

[Operador IN](#)

[Consultas DAX](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CONTAINSSTRING

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve TRUE o FALSE para indicar si una cadena contiene otra cadena.

Sintaxis

DAX

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

Parámetros

[] Expandir tabla

Término	Definición
within_text	Texto en el que se quiere buscar find_text.
find_text	Texto que se desea buscar.

Valor devuelto

TRUE si find_text es una subcadena de within_text; de lo contrario, FALSE.

Observaciones

- CONTAINSSTRING no distingue entre mayúsculas y minúsculas.
- Puede usar los caracteres comodín `?` y `*`. Use `~` para eludir caracteres comodín.

Ejemplo

Consulta DAX

DAX

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

Devuelve

 Expandir tabla

[Caso 1]	[Caso 2]	[Caso 3]	[Caso 4]
VERDADERO	TRUE	TRUE	FALSE

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CONTAINSSTRINGEXACT

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve TRUE o FALSE para indicar si una cadena contiene otra cadena.

Sintaxis

DAX

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

Parámetros

[] Expandir tabla

Término	Definición
within_text	Texto en el que se quiere buscar find_text.
find_text	Texto que se desea buscar.

Valor devuelto

TRUE si find_text es una subcadena de within_text; de lo contrario, FALSE.

Observaciones

CONTAINSSTRINGEXACT distingue mayúsculas de minúsculas.

Ejemplo

Consulta DAX

DAX

```
EVALUATE  
ROW(  
    "Case 1", CONTAINSSTRINGEXACT("abcd", "bc"),
```

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

Devuelve

 Expandir tabla

[Caso 1]	[Caso 2]	[Caso 3]	[Caso 4]
true	false	false	FALSE

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CUSTOMDATA

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve el contenido de la propiedad **CustomData** de la cadena de conexión.

Sintaxis

DAX

```
CUSTOMDATA()
```

Valor devuelto

Contenido de la propiedad **CustomData** de la cadena de conexión.

En blanco si la propiedad **CustomData** no se ha definido en el momento de la conexión.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la fórmula DAX siguiente se comprueba si la propiedad CustomData se ha establecido en "OK".

DAX

```
= IF(CUSTOMDATA()="OK", "Correct Custom data in connection string", "No custom data in connection string property or unexpected value")
```

Comentarios

¿Le ha resultado útil esta página?

👍 Sí

👎 No

HASONEFILTER

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve **TRUE** si el número de valores filtrados directamente en *columnName* es uno; de lo contrario, devuelve **FALSE**.

Sintaxis

DAX

`HASONEFILTER(<columnName>)`

Parámetros

Expandir tabla

Término	Definición
columnName	Nombre de una columna existente, con la sintaxis de DAX estándar. No puede ser una expresión.

Valor devuelto

TRUE si el número de valores filtrados directamente en *columnName* es uno; de lo contrario, devuelve **FALSE**.

Observaciones

- Esta función es similar a HASONEVALUE(), con la diferencia de que el funcionamiento de HASONEVALUE() se basa en filtros cruzados, mientras que HASONEFILTER() funciona con un filtro directo.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el siguiente ejemplo se ilustra cómo usar HASONEFILTER() para devolver el filtro de ResellerSales_USD[ProductKey]) si hay un filtro, o para devolver BLANK si no hay filtros o más de un filtro en ResellerSales_USD[ProductKey]).

DAX

```
=  
IF(HASONEFILTER(ResellerSales_USD[ProductKey]),FILTERS(ResellerSales_USD[Pro  
ductKey]),BLANK())
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

HASONEVALUE

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medición ✓ Cálculo visual

Devuelve **TRUE** si el contexto de *columnName* se ha filtrado a un solo valor distinto. De lo contrario, es **FALSE**.

Sintaxis

HTML

```
HASONEVALUE(<columnName>)
```

Parámetros

[+] Expandir tabla

Término	Definición
columnName	Nombre de una columna existente, con la sintaxis de DAX estándar. No puede ser una expresión.

Valor devuelto

TRUE si el contexto de *columnName* se ha filtrado a un solo valor distinto. De lo contrario, es **FALSE**.

Notas

- Una expresión equivalente de HASONEVALUE() es
`COUNTROWS(values(<columnName>)) = 1.`
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la siguiente fórmula de medida se comprueba si el contexto se segmenta mediante un valor para calcular un porcentaje en un escenario predefinido; en este caso, quiere comparar ResellerSales con las ventas de 2007 y, después, necesita saber si el contexto está filtrado por años individuales. Además, si la comparación no tiene sentido, quiere que se devuelva BLANK.

DAX

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

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISAFTER

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Una función booleana que emula el comportamiento de una cláusula "Fecha de inicio" y devuelve "true" para una fila que cumple todos los parámetros de la condición.

Según el criterio de ordenación, el primer parámetro se compara con el segundo. Si el criterio de ordenación es ascendente, la comparación que se va a hacer es: primer parámetro mayor que el segundo. Si el criterio de ordenación es descendente, la comparación que se va a hacer es: segundo parámetro menor que el primero.

Sintaxis

DAX

```
ISAFTER(<scalar_expression>, <scalar_expression>[, sort_order [, <scalar_expression>, <scalar_expression>[, sort_order]]]...)
```

Parámetros

Expandir tabla

Término	Definición
expresión escalar	Cualquier expresión que devuelve un valor escalar como una referencia de columna o un valor de cadena o entero. Normalmente, el primer parámetro es una referencia de columna y el segundo es un valor escalar.
criterio de ordenación	(opcional) El orden en que se ordena la columna. Puede ser ascendente (ASC) o descendente (DEC). De forma predeterminada, el criterio de ordenación es ascendente.

Valor devuelto

True o false.

Comentarios

Esta función es similar a [ISONORAFTER](#). La diferencia es que ISAFTER devuelve true para los valores ordenados estrictamente *después* de los valores de filtro, donde ISONORAFTER devuelve true para los valores ordenados *en o después* de los valores de filtro.

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Nombre de la tabla: "Info"

 Expandir tabla

País/región	State	Count	Total
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900
EE. UU.	CA	5	500
EE. UU.	WA	10	900

La expresión siguiente:

DAX
<pre>FILTER (Info, ISAFTER (Info[Country], "IND", ASC, Info[State], "MH", ASC))</pre>

Devuelve:

 Expandir tabla

País	State	Count	Total
IND	WB	10	900
EE. UU.	CA	5	500

País	State	Count	Total
EE. UU.	WA	10	900

Contenido relacionado

[ISONORAFTER](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISBLANK

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Comprueba si un valor está en blanco y devuelve TRUE o FALSE.

Sintaxis

DAX
<code>ISBLANK(<value>)</code>

Parámetros

Expandir tabla

Término	Definición
valor	Valor o expresión que se quiere probar.

Valor devuelto

Valor booleano de TRUE si el valor está en blanco; de lo contrario, FALSE.

Comentarios

Para más información sobre los procedimientos recomendados al trabajar con BLANKS, consulte [Impedimento de la conversión de BLANK en valores en DAX](#).

Ejemplo

Esta fórmula calcula la relación de aumento o disminución de las ventas en comparación con el año anterior. En el ejemplo se usa la función IF para comprobar el valor de las ventas del año anterior a fin de evitar un error de división por cero.

DAX
<code>IF(ISBLANK([Sales]), 0, ([Sales] - [Sales] <sup>Año anterior</sup>) / [Sales] <sup>Año anterior</sup>))</code>

```
//Sales to Previous Year Ratio

= IF( ISBLANK('CalculatedMeasures'[PreviousYearTotalSales])
    , BLANK()
    , ( 'CalculatedMeasures'[Total Sales]-
'CalculatedMeasures'[PreviousYearTotalSales] )
    /'CalculatedMeasures'[PreviousYearTotalSales])
```

Result,

 Expandir tabla

Etiquetas de fila	Total Sales	Ventas totales del año anterior	Relación de ventas con el año anterior
2005	10 209 985,08 \$		
2006	28 553 348,43 \$	10 209 985,08 \$	179,66 %
2007	39 248 847,52 \$	28 553 348,43 \$	37,46 %
2008	24 542 444,68 \$	39 248 847,52 \$	-37,47 \$
Total general	102 554 625,71 \$		

Contenido relacionado

[Funciones de información](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISCROSSFILTERED

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve TRUE cuando la tabla o columna especificada se filtran de forma cruzada.

Sintaxis

DAX

`ISCROSSFILTERED(<TableNameOrColumnName>)`

Parámetros

Expandir tabla

Término	Definición
TableNameOrColumnName	Nombre de una tabla o columna existente. No puede ser una expresión.

Valor devuelto

TRUE cuando se filtra de forma cruzada *ColumnName* o una columna de *TableName*. En caso contrario, devuelve FALSE.

Observaciones

- Se dice que una columna o tabla se filtra de forma cruzada cuando se aplica un filtro a *ColumnName*, a cualquier columna de *TableName* o a cualquier columna de una tabla relacionada.
- Se dice que una columna o tabla se filtra directamente cuando se aplica un filtro a *ColumnName* o a cualquier columna de *TableName*. Por lo tanto, la función [ISFILTERED](#) también devuelve TRUE cuando se filtra *ColumnName* o cualquier columna de *TableName*.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Contenido relacionado

[ISFILTERED, función](#)

[Función FILTERS](#)

[HASONEFILTER, función](#)

[HASONEVALUE, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISEMPTY

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Comprueba si una tabla está vacía.

Sintaxis

DAX

```
ISEMPTY(<table_expression>)
```

Parámetros

[] Expandir tabla

Término	Definición
table_expression	Una referencia de tabla o una expresión DAX que devuelve una tabla.

Valor devuelto

True si la tabla está vacía (no tiene filas); en caso contrario, es false.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Para esta tabla llamada "info":

[] Expandir tabla

País/región	State	County	Total
IND	JK	20	800

País/región	State	County	Total
IND	MH	25	1000
IND	WB	10	900
EE. UU.	CA	5	500
EE. UU.	WA	10	900

DAX

```
EVALUATE
ROW("Any countries with count > 25?", NOT(ISEMPTY(FILTER(Info,
[County]>25))))
```

Valor devuelto: FALSE

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISERROR

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Comprueba si un valor es un error y devuelve TRUE o FALSE.

Sintaxis

DAX
<code>ISERROR(<value>)</code>

Parámetros

[] Expandir tabla

Término	Definición
valor	Valor que se quiere probar.

Valor devuelto

Valor booleano de TRUE si el valor es un error; en caso contrario, FALSE.

Comentarios

- Para conocer los procedimientos recomendados al usar ISERROR, consulte [Uso adecuado de funciones de error](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En este ejemplo se calcula la relación entre el total de ventas por Internet y el total de ventas de distribuidores. La función ISERROR se usa para comprobar si hay errores, como la división por cero. Si se produce un error, se devuelve un espacio en blanco; en caso contrario, se devuelve la proporción.

DAX

```
= IF( ISERROR(
    SUM('ResellerSales_USD'[SalesAmount_USD])
    /SUM('InternetSales_USD'[SalesAmount_USD])
)
, BLANK()
, SUM('ResellerSales_USD'[SalesAmount_USD])
/SUM('InternetSales_USD'[SalesAmount_USD])
)
```

Contenido relacionado

[Funciones de información](#)

[IFERROR, función](#)

[Función IF](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISEVEN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve TRUE si el número es par o FALSE si el número es impar.

Sintaxis

DAX
<code>ISEVEN(number)</code>

Parámetros

Expandir tabla

Término	Definición
número	Valor que se va a comprobar. Si el número no es un entero, se trunca.

Valor devuelto

Devuelve TRUE si el número es par o FALSE si el número es impar.

Observaciones

- Si el número no es un valor numérico, ISEVEN devuelve el valor de error #NUM!
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Comentarios

¿Le ha resultado útil esta página?

Sí

No

ISFILTERED

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medición ✓ Cálculo visual

Devuelve TRUE cuando la tabla o columna especificada se filtra directamente.

Sintaxis

DAX

```
ISFILTERED(<TableNameOrColumnName>)
```

Parámetros

[] Expandir tabla

Término	Definición
TableNameOrColumnName	Nombre de una tabla o columna existente. No puede ser una expresión.

Valor devuelto

TRUE cuando *ColumnName* o una columna de *TableName* se filtra directamente. En caso contrario, devuelve FALSE.

Observaciones

- Se dice que una columna o tabla se filtra directamente cuando se aplica un filtro a *ColumnName* o a cualquier columna de *TableName*.
- Se dice que una columna o tabla se filtra de forma cruzada cuando se aplica un filtro a *ColumnName*, a cualquier columna de *TableName* o a cualquier columna de una tabla relacionada. Por lo tanto, la función [ISCROSSFILTERED](#) también devuelve TRUE cuando se filtra *ColumnName*, cualquier columna de *TableName* o una columna de una tabla relacionada.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Contenido relacionado

[ISCROSSFILTERED, función](#)

[Función FILTERS](#)

[HASONEFILTER, función](#)

[HASONEVALUE, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISINSCOPE

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medición ✓ Cálculo visual

Devuelve true si la columna especificada es el nivel de una jerarquía de niveles.

Sintaxis

DAX

```
ISINSCOPE(<columnName>)
```

Parámetros

[] Expandir tabla

Término	Definición
columnName	Nombre de una columna existente, con la sintaxis de DAX estándar. No puede ser una expresión.

Valor devuelto

TRUE si la columna especificada es el nivel de una jerarquía de niveles.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

DAX

```
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]

```

Vea que devuelve lo siguiente:

[\[\] Expandir tabla](#)

DimProduct[Category]	DimProduct[SubCategory]	[Category Subtotal]	[Subcategory Subtotal]	[Sales]	[% of Parent]
		TRUE	TRUE	28 397 095,65	100,00
Accesorios		FALSO	true	78 951,00	0,28
Accessories	Marcos de bicicletas	FALSE	FALSE	39 360,00	49,85
Accesorios	Soportes de bicicletas	FALSE	FALSE	39 591,00	50,15
Bicicletas		false	true	28 318 144,65	99,72
Bikes	Bicicletas de montaña	FALSE	FALSE	9 952 759,56	35,15
Bicicletas	Bicicletas de carretera	FALSE	FALSE	14 520 584,04	51,28
Bicicletas	Bicicletas de paseo	FALSE	FALSE	3 844 801,05	13,58

Contenido relacionado

[Función SUMMARIZECOLUMNS](#)

[CALCULATE, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISLOGICAL

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Comprueba si un valor es lógico (TRUE o FALSE), y devuelve TRUE o FALSE.

Sintaxis

DAX

```
ISLOGICAL(<value>)
```

Parámetros

Expandir tabla

Término	Definición
valor	El valor que se quiere probar.

Valor devuelto

TRUE si el valor es un valor lógico; FALSE si es cualquier valor distinto de TRUE o FALSE.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En los tres ejemplos siguientes se muestra el comportamiento de ISLOGICAL.

DAX

```
//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")
```

Contenido relacionado

[Funciones de información](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISNOTTEXT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Comprueba si un valor no es de texto (las celdas en blanco no son texto) y devuelve TRUE o FALSE.

Sintaxis

DAX

```
ISNOTTEXT(<value>)
```

Parámetros

Expandir tabla

Término	Definición
valor	Valor que se quiere comprobar.

Valor devuelto

TRUE si el valor no es texto o está en blanco; FALSE si el valor es texto.

Observaciones

- Una cadena vacía se considera texto.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En los ejemplos siguientes se muestra el comportamiento de la función ISNOTTEXT.

DAX

```
//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")
```

Contenido relacionado

[Funciones de información](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISNUMBER

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medición ✓ Cálculo visual

Comprueba si un valor es un número y devuelve TRUE o FALSE.

Sintaxis

DAX

```
ISNUMBER(<value>)
```

Parámetros

[] Expandir tabla

Término	Definición
valor	Valor que se quiere probar.

Valor devuelto

TRUE si el valor es numérico; de lo contrario, FALSE.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En los tres ejemplos siguientes se muestra el comportamiento de ISNUMBER.

DAX

```
//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")
```

Contenido relacionado

[Funciones de información](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISODD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve TRUE si el número es impar o FALSE si es par.

Sintaxis

DAX
<code>ISODD(number)</code>

Parámetros

Expandir tabla

Término	Definición
número	Valor que se va a comprobar. Si el número no es un entero, se trunca.

Valor devuelto

Devuelve TRUE si el número es impar o FALSE si es par.

Observaciones

- Si el número no es un valor numérico, ISODD devuelve el valor de error "#VALUE!" #NUM!
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Comentarios

¿Le ha resultado útil esta página?

Sí

No

ISONORAFTER

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Una función booleana que emula el comportamiento de una cláusula "Fecha de inicio" y devuelve true para una fila que cumple todos los parámetros de la condición.

Según el criterio de ordenación, el primer parámetro se compara con el segundo. Si el criterio de ordenación es ascendente, la comparación que se va a hacer es: primer parámetro mayor que el segundo. Si el criterio de ordenación es descendente, la comparación que se va a hacer es: segundo parámetro menor que el primero.

Sintaxis

DAX

```
ISONORAFTER(<scalar_expression>, <scalar_expression>[, sort_order [, <scalar_expression>, <scalar_expression>[, sort_order]]]...)
```

Parámetros

Expandir tabla

Término	Definición
expresión escalar	Cualquier expresión que devuelve un valor escalar como una referencia de columna o un valor de cadena o entero. Normalmente, el primer parámetro es una referencia de columna y el segundo es un valor escalar.
criterio de ordenación	(opcional) El orden en que se ordena la columna. Puede ser ascendente (ASC) o descendente (DESC). De forma predeterminada, el criterio de ordenación es ascendente.

Valor devuelto

True o false.

Comentarios

Esta función es similar a [ISAFTER](#). La diferencia es que ISONORAFTER devuelve true para los valores ordenados *en o después* de los valores de filtro, donde ISAFTER devuelve true para los valores ordenados estrictamente *después* de los valores de filtro.

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Para la tabla siguiente denominada Info:

 Expandir tabla

País/región	State	Count	Total
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900
EE. UU.	CA	5	500
EE. UU.	WA	10	900

La expresión siguiente:

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

Devuelve:

 Expandir tabla

País/región	State	Count	Total
IND	MH	25	1000
IND	WB	10	900
EE. UU.	CA	5	500

País/región	State	Count	Total
EE. UU.	WA	10	900

Contenido relacionado

[ISAFTER](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISSELECTEDMEASURE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Se usa en expresiones para que los elementos de cálculo determinen que la medida que está en contexto es una de las especificadas en la lista de medidas.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
M1, M2, ...	Lista de medidas.

Valor devuelto

Valor booleano que indica si la medida que está actualmente en contexto es una de las especificadas en la lista de parámetros.

Notas

- Solo se puede hacer referencia en la expresión para un elemento de cálculo.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Esta expresión de elemento de cálculo comprueba si la medida actual es una de las especificadas en la lista de parámetros. Si se cambia el nombre de las medidas, la corrección de fórmula reflejará los cambios de nombre en la expresión.

DAX

```
IF (
    ISSELECTEDMEASURE ( [Expense Ratio 1], [Expense Ratio 2] ),
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```

Contenido relacionado

[SELECTEDMEASURE](#)

[SELECTEDMEASURENAME](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISSUBTOTAL

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✗ Cálculo visual

Crea otra columna en una expresión [SUMMARIZE](#) que devuelve "true" si la fila contiene valores de subtotal de la columna especificada como argumento; de lo contrario, devuelve "false".

Syntax

DAX

```
ISSUBTOTAL(<columnName>)
```

Con [SUMMARIZE](#):

DAX

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[,  
ROLLUP(<groupBy_columnName>[,< groupBy_columnName>...])][, <name>,  
{<expression>|ISSUBTOTAL(<columnName>)})]...)
```

Parámetros

 Expandir tabla

Término	Definición
columnName	Nombre de cualquier columna de table de la función SUMMARIZE o de cualquier columna de una tabla relacionada con table.

Valor devuelto

"true" si la fila contiene un valor de subtotal para la columna especificada como argumento; de lo contrario, devuelve "false".

Comentarios

- Esta función solo se puede usar en la expresión de una función [SUMMARIZE](#).

- Esta función debe ir precedida del nombre de la columna booleana.

Ejemplo

Vea [SUMMARIZE](#).

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISTEXT

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Comprueba si un valor es texto y devuelve TRUE o FALSE.

Sintaxis

DAX

```
ISTEXT(<value>)
```

Parámetros

[] Expandir tabla

Término	Definición
valor	Valor que se quiere comprobar.

Valor devuelto

"true" si el valor es texto; de lo contrario, "false".

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En los ejemplos siguientes se muestra el comportamiento de la función ISTEXT.

DAX

```
//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")
```

Contenido relacionado

Funciones de información

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NONVISUAL

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✗ Cálculo visual

Marca un filtro de valores en una expresión [SUMMARIZECOLUMNS](#) como no visual. Esta función solo se puede usar en una expresión [SUMMARIZECOLUMNS](#).

Sintaxis

DAX

`NONVISUAL(<expression>)`

Parámetros

[+] Expandir tabla

Término	Definición
expression	Cualquier expresión DAX que devuelva un único valor (no una tabla).

Valor devuelto

Una tabla de valores.

Comentarios

- Marca un filtro de valor en la función [SUMMARIZECOLUMNS](#) como que no afecta a los valores de medida, sino que solo se aplica a las columnas group-by.
- Esta función solo se puede usar en una expresión [SUMMARIZECOLUMNS](#). Se utiliza como argumento filterTable de la función [SUMMARIZECOLUMNS](#), o groupLevelFilter de las funciones [ROLLUPADDISSUBTOTAL](#) o [ROLLUPISSUBTOTAL](#).

Ejemplo

Vea [SUMMARIZECOLUMNS](#).

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SELECTEDMEASURE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Se usa en expresiones para que los elementos de cálculo hagan referencia a la medida que está en contexto.

Sintaxis

DAX

```
SELECTEDMEASURE()
```

Parámetros

None

Valor devuelto

Referencia a la medida que está actualmente en contexto cuando se evalúa el elemento de cálculo o la cadena de formato.

Observaciones

- Solo se puede hacer referencia en la expresión para un elemento de cálculo o cadena de formato.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Esta expresión de elemento de cálculo calcula el año hasta la fecha de la medida que se encuentra en contexto.

DAX

```
CALCULATE(SELECTEDMEASURE(), DATESYTD(DimDate[Date]))
```

La expresión siguiente se puede usar para ajustar dinámicamente la cadena de formato de una medida en función de si un valor es los cientos, miles o millones.

```
DAX

SWITCH(
    TRUE(),
    SELECTEDMEASURE() < 1000, "#,##0", //Values less than 1000 have
    no text after them
    SELECTEDMEASURE() < 1000000, "#,##0,.0 K", //Values between 1000 and
    1000000 are formatted as #.## K
    "#,##0,,.0 M" //Values greater than 1000000
    are formatted as #.## M
)
```

Contenido relacionado

[SELECTEDMEASURENAME](#)
[ISSELECTEDMEASURE](#)

Comentarios

¿Le ha resultado útil esta página?



SELECTEDMEASUREFORMATSTRING

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Se usa en expresiones para que los elementos de cálculo recuperen la cadena de formato de la medida que está en contexto.

Sintaxis

DAX

```
SELECTEDMEASUREFORMATSTRING()
```

Parámetros

None

Valor devuelto

Una cadena que contiene la cadena de formato de la medida que está actualmente en contexto cuando se evalúa el elemento de cálculo.

Notas

- Solo se puede hacer referencia a esta función en expresiones de elementos de cálculo en grupos de cálculo. Está diseñada para ser utilizada por la propiedad **Expresión de cadena de formato** de los elementos de cálculo.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La expresión siguiente se evalúa mediante la propiedad Expresión de cadena de formato para un elemento de cálculo. Si hay una sola moneda en el contexto de filtro, la cadena de formato se recupera de la columna DimCurrency[FormatString]; de lo contrario, se usa la cadena de formato de la medida en contexto.

DAX

```
SELECTEDVALUE( DimCurrency[FormatString], SELECTEDMEASUREFORMATSTRING() )
```

Contenido relacionado

[SELECTEDMEASURE](#)

[ISSELECTEDMEASURE](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SELECTEDMEASURENAME

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Se usa en expresiones para que los elementos de cálculo determinen por nombre la medida que está en contexto.

Sintaxis

DAX

```
SELECTEDMEASURENAME()
```

Parámetros

None

Valor devuelto

Valor de cadena que contiene el nombre de la medida que está actualmente en contexto cuando se evalúa el elemento de cálculo.

Notas

- Solo se puede hacer referencia en la expresión para un elemento de cálculo.
- Esta función se suele usar con fines de depuración al crear grupos de cálculo.

Ejemplo

La expresión de elemento de cálculo siguiente comprueba si la medida actual es Expense Ratio y aplica condicionalmente la lógica de cálculo. Como la comprobación se basa en una comparación de cadenas, no está sujeta a la corrección de fórmulas y no se beneficiará del cambio de nombre de objeto que se refleja automáticamente. Para obtener una comparación similar que se beneficiaría de la corrección de fórmulas, vea la función ISSELECTEDMEASURE en su lugar.

DAX

```
IF (
    SELECTEDMEASURENAME = "Expense Ratio",
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```

Contenido relacionado

[SELECTEDMEASURE](#)

[ISSELECTEDMEASURE](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

USERCULTURE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve la configuración regional (código de idioma y código de país) para el usuario actual, determinada por el sistema operativo, la configuración del explorador o el servicio Power BI.

Nota: Actualmente esta función solo se admite en Power BI Premium por capacidad, Power BI Premium por usuario y Power BI Embedded.

Sintaxis

DAX

```
USERCULTURE()
```

Parámetros

Esta expresión no tiene parámetros.

Valor devuelto

Configuración regional como una cadena.

Comentarios

- En el servicio Power BI, la configuración regional se determina en **Configuración>Idioma>Configuración de idioma**. El valor predeterminado se determina mediante la configuración de idioma del explorador del usuario.
- Cuando se usa en expresiones de tabla calculada y columna calculada, el resultado puede diferir en función de si la tabla está en modo DirectQuery o Import. Cuando está en modo DirectQuery, el resultado viene determinado por el idioma (configuración regional) especificado en la Configuración de idioma del servicio Power BI. El valor predeterminado en Configuración de idioma especifica la configuración regional y viene determinado por la configuración de idioma del explorador del usuario, lo que significa que la misma tabla o columna calculada

puede devolver resultados diferentes en función de la configuración de idioma del explorador para cada usuario. Cuando está en modo Import, el resultado se determina estáticamente durante la actualización y no variará en el momento de la consulta. En el caso de las actualizaciones administradas, como las programadas o interactivas, la configuración regional no se basa en la configuración de idioma del explorador del usuario, sino que usa una configuración regional invariable. Sin embargo, la configuración regional invariable se puede invalidar mediante el punto de conexión XMLA para especificar una configuración regional personalizada.

- Cuando se combina con la característica Parámetros de campo en Power BI, USERCULTURE se puede usar para traducir de forma confiable subtítulos y títulos de visualización dinámicos cuando se usan en expresiones de objeto de seguridad de nivel de fila (RLS) y medida dentro del mismo modelo. Sin embargo, en el caso de títulos y subtítulos traducidos correctamente, no debe recurrirse a expresiones que contengan USERCULTURE a las que se llame desde fuera del modelo, como consultas y medidas de informe de Live Connect.
- USERCULTURE devuelve la configuración regional de usuario correcta cuando se usa en expresiones de objeto llamadas desde dentro del modelo, como medidas, seguridad de nivel de fila (RLS) y elementos de cálculo. Sin embargo, es posible que no devuelva la configuración regional del usuario correcta cuando se use en expresiones desde fuera del modelo, como consultas y medidas de informe de Live Connect.
- En los informes de Live Connect, USERCULTURE no puede devolver la configuración regional de usuario correcta cuando se llama desde una expresión de medida de informe.

Ejemplo

Para la siguiente expresión,

DAX

```
FORMAT(TODAY(), "dddd", USERCULTURE())
```

Según la configuración de idioma del usuario actual, USERCULTURE devuelve el día actual, por ejemplo:

 Expandir tabla

Configuración regional	Día de la semana con formato
de-DE	Dienstag
es-ES	Tuesday
es-ES_tradnl	martes
eu-ES	asteartea
it-IT	martedì
nl-NL	dinsdag
pl-PL	wtorek
ro-RO	marți
ru-RU	вторник
uk-UA	вівторок

Contenido relacionado

[Títulos basados en expresiones en Power BI](#)

[USERNAME](#)

[USERPRINCIPALNAME](#)

[USEROBJECTID](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

USERNAME

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el nombre de dominio y el nombre de usuario de las credenciales proporcionadas al sistema en el momento de la conexión.

Sintaxis

DAX

```
USERNAME()
```

Parámetros

Esta expresión no tiene parámetros.

Valor devuelto

Nombre de usuario de las credenciales proporcionadas al sistema en el momento de la conexión.

Ejemplo

En la fórmula siguiente se comprueba si el inicio de sesión del usuario forma parte de UsersTable.

DAX

```
= IF(CONTAINS(UsersTable,UsersTable[login], USERNAME()), "Allowed", BLANK())
```

Comentarios

¿Le ha resultado útil esta página?

Sí

No

USEROBJECTID

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el identificador de objeto del usuario actual de Microsoft Entra ID o el identificador de seguridad (SID).

Sintaxis

DAX

```
USEROBJECTID()
```

Parámetros

Esta expresión no tiene parámetros.

Valor devuelto

Identificador de objeto del usuario actual de Microsoft Entra ID para Power BI o modelos de Azure Analysis Services o SID para modelos de SQL Server Analysis Services.

Comentarios

¿Le ha resultado útil esta página?

Sí

No

USERPRINCIPALNAME

Artículo • 28/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el nombre principal de usuario.

Sintaxis

DAX

```
USERPRINCIPALNAME()
```

Parámetros

Esta expresión no tiene parámetros.

Valor devuelto

UserPrincipalName en el momento de la conexión.

Comentarios

¿Le ha resultado útil esta página?

Sí

No

Funciones lógicas

Artículo • 20/10/2023

Las funciones lógicas actúan sobre una expresión para devolver información sobre los valores o los conjuntos de la expresión. Por ejemplo, puede usar la función IF para comprobar el resultado de una expresión y crear resultados condicionales.

En esta categoría

Función	Descripción
AND	Comprueba si los dos argumentos son TRUE y devuelve TRUE si todos lo son.
BITAND	Devuelve una operación 'AND' bit a bit de dos números.
BITLSHIFT	Devuelve un número desplazado a la izquierda el número de bits especificado.
BITOR	Devuelve una operación 'OR' bit a bit de dos números.
BITRSHIFT	Devuelve un número desplazado a la derecha el número de bits especificado.
BITXOR	Devuelve una operación 'XOR' bit a bit de dos números.
COALESCE	Devuelve la primera expresión que no se evalúe como BLANK.
FALSE	Devuelve el valor lógico FALSE.
IF	Comprueba una condición y devuelve un valor cuando es "TRUE"; en caso contrario, devuelve un segundo valor.
IF.EAGER	Comprueba una condición y devuelve un valor cuando es "TRUE"; en caso contrario, devuelve un segundo valor. Utiliza un plan de ejecución <i>diligente</i> que siempre ejecuta las expresiones de bifurcación independientemente de la expresión de condición.
IFERROR	Evalúa una expresión y devuelve un valor especificado si la expresión devuelve un error.
NOT	Cambia FALSE por TRUE o TRUE por FALSE.
OR	Comprueba si uno de los argumentos es TRUE para devolver TRUE.
SWITCH	Evalúa una expresión en una lista de valores y devuelve una de varias expresiones de resultado posibles.
TRUE	Devuelve el valor lógico TRUE.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

AND

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Comprueba si los dos argumentos son TRUE y devuelve TRUE si todos lo son. De lo contrario, devuelve false.

Sintaxis

DAX

```
AND(<logical1>,<logical2>)
```

Parámetros

Expandir tabla

Término	Definición
logical_1, logical_2	Valores lógicos que se quieren probar.

Valor devuelto

Devuelve true o false en función de la combinación de valores que se prueba.

Notas

La función AND de DAX acepta solo dos (2) argumentos. Si necesita realizar una operación AND en varias expresiones, puede crear una serie de cálculos o, mejor aún, usar el operador AND (**&&**) para combinarlas en una expresión más sencilla.

Ejemplo 1

La fórmula siguiente muestra la sintaxis de la función AND.

DAX

```
= IF(AND(10 > 9, -10 < -1), "All true", "One or more false")
```

Dado que ambas condiciones que se pasan como argumentos a la función AND son true, la fórmula devuelve "All true".

Ejemplo 2

En el ejemplo siguiente se usa la función AND con fórmulas anidadas para comparar dos conjuntos de cálculos a la vez. Para cada categoría de producto, la fórmula determina si las ventas del año actual y las del año anterior del canal de Internet son mayores que las del canal de revendedores durante los mismos períodos. Si ambas condiciones son true, la fórmula devuelve el valor "Internet Hit" para cada categoría.

DAX

```
= 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"  
        , ""  
        )
```

Devuelve

 Expandir tabla

Etiquetas de fila	2005	2006	2007	2008	- Total general
Pantalones de ciclismo					
Bastidores de bicicletas					
Soportes de bicicletas				Internet Hit	
Botellas y contenedores				Internet Hit	
Ejes pedalier					
Frenos					
Gorras					

Etiquetas de fila	2005	2006	2007	2008	- Total general
Cadenas					
Limpiadores					
Bielas					
Desviadores					
Guardabarros					Internet Hit
Horquillas					
Guantes					
Manillares					
Tubos de dirección					
Cascos					
Paquetes de hidratación					
Maillots					
Luces					
Bloqueos					
Bicicletas de montaña					
Cuadros de bicicletas de montaña					
Alforjas					
Pedales					
Bombas					
Bicicletas de carretera					
Cuadros de bicicletas de carretera					
Sillines					
Pantalones cortos					
Calcetines					
Medias					
Cubiertas y llantas					Internet Hit

Etiquetas de fila	2005	2006	2007	2008	- Total general
Bicicletas de paseo					
Cuadros de bicicletas de paseo					
Chalecos					
Ruedas					
Total general					

Contenido relacionado

[Funciones lógicas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

BITAND

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve una operación AND bit a bit de dos números.

Sintaxis

DAX

```
BITAND(<number>, <number>)
```

Parámetros

Expandir tabla

Término	Definición
Número	Cualquier expresión escalar que devuelva un número. Si el valor no es un entero, se trunca.

Valor devuelto

Operación AND bit a bit de dos números.

Comentarios

- Esta función admite números positivos y negativos.

Ejemplo

La consulta DAX siguiente:

DAX

```
EVALUATE { BITAND(13, 11) }
```

Devuelve 9.

Contenido relacionado

[BITLSHIFT](#)

[BITRSHIFT](#)

[BITOR](#)

[BITXOR](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

BITLSHIFT

Artículo • 30/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve un número desplazado a la izquierda el número de bits especificado.

Sintaxis

DAX
<code>BITLSHIFT(<Number>, <Shift_Amount>)</code>

Parámetros

[] Expandir tabla

Término	Definición
Número	Cualquier expresión DAX que devuelva una expresión de entero.
Shift_Amount	Cualquier expresión DAX que devuelva una expresión de entero.

Valor devuelto

Valor de entero.

Comentarios

- Asegúrese de comprender la naturaleza de las operaciones de desplazamiento de bits y el desbordamiento o subdesbordamiento de enteros antes de usar las funciones de desplazamiento de bits DAX.
- Si Shift_Amount es un valor negativo, el desplazamiento se realiza en la dirección opuesta.
- Si el valor absoluto de Shift_Amount es mayor que 64, no habrá ningún error, pero dará lugar a un desbordamiento o subdesbordamiento.
- No hay ningún límite para el valor de Number, pero el resultado puede desbordarse o subdesbordarse.

Ejemplos

Ejemplo 1

La consulta DAX siguiente:

```
DAX  
  
EVALUATE  
{ BITLSHIFT(2, 3) }
```

Devuelve 16.

Ejemplo 2

La consulta DAX siguiente:

```
DAX  
  
EVALUATE  
{ BITLSHIFT(128, -1) }
```

Devuelve 64.

Ejemplo 3

La consulta DAX siguiente:

```
DAX  
  
Define  
    Measure Sales[LeftShift] = BITLSHIFT(SELECTEDVALUE(Sales[Amount]), 3)  
  
EVALUATE  
SUMMARIZECOLUMNS(  
    Sales[Amount],  
    "LEFTSHIFT",  
    [LeftShift]  
)
```

Desplaza a la izquierda cada importe de ventas con 3 bits y devuelve el importe de ventas desplazado.

Contenido relacionado

[BITSHIFT](#)

[BITAND](#)

[BITOR](#)

[BITXOR](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

BITOR

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve una operación OR bit a bit de dos números.

Sintaxis

DAX

```
BITOR(<number>, <number>)
```

Parámetros

Expandir tabla

Término	Definición
Número	Cualquier expresión escalar que devuelva un número. Si el valor no es un entero, se trunca.

Valor devuelto

Operación OR bit a bit de dos números.

Comentarios

- Esta función admite números positivos y negativos.

Ejemplo

La consulta DAX siguiente:

DAX

```
EVALUATE
```

```
{ BITOR(9, 10) }
```

Devuelve 11.

Contenido relacionado

[BITAND](#)

[BITXOR](#)

[BITLSHIFT](#)

[BITRSHIFT](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

BITRSHIFT

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve un número desplazado a la derecha el número de bits especificado.

Sintaxis

DAX
<code>BITRSHIFT(<Number>, <Shift_Amount>)</code>

Parámetros

[] Expandir tabla

Término	Definición
Número	Cualquier expresión DAX que devuelva una expresión de entero.
Shift_Amount	Cualquier expresión DAX que devuelva una expresión de entero.

Valor devuelto

Valor de entero.

Comentarios

- Asegúrese de comprender la naturaleza de las operaciones de desplazamiento de bits y el desbordamiento o subdesbordamiento de enteros antes de usar las funciones de desplazamiento de bits DAX.
- Si Shift_Amount es un valor negativo, el desplazamiento se realiza en la dirección opuesta.
- Si el valor absoluto de Shift_Amount es mayor que 64, no habrá ningún error, pero dará lugar a un desbordamiento o subdesbordamiento.
- No hay ningún límite para el valor de Number, pero el resultado puede desbordarse o subdesbordarse.

Ejemplos

Ejemplo 1

La consulta DAX siguiente:

```
DAX  
EVALUATE  
{ BITRSHIFT(16, 3) }
```

Devuelve 2.

Ejemplo 2

La consulta DAX siguiente:

```
DAX  
EVALUATE  
{ BITRSHIFT(1024, -3) }
```

Devuelve 8192.

Ejemplo 3

La consulta DAX siguiente:

```
DAX  
Define  
    Measure Sales[RightShift] = BITRSHIFT(SELECTEDVALUE(Sales[Amount]), 3)  
  
EVALUATE  
SUMMARIZECOLUMNS(  
    Sales[Amount],  
    "RIGHTSHIFT",  
    [RightShift]  
)
```

Desplaza a la derecha cada importe de ventas con 3 bits y devuelve el importe de ventas desplazado.

Contenido relacionado

[BITSHIFT](#)

[BITAND](#)

[BITOR](#)

[BITXOR](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

BITXOR

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve una operación XOR bit a bit de dos números.

Sintaxis

DAX

```
BITXOR(<number>, <number>)
```

Parámetros

Expandir tabla

Término	Definición
Número	Cualquier expresión escalar que devuelva un número. Si el valor no es un entero, se trunca.

Valor devuelto

Operación XOR bit a bit de dos números.

Comentarios

- Esta función admite números positivos y negativos.

Ejemplo

La consulta DAX siguiente:

DAX

```
EVALUATE { BITXOR(9, 10) }
```

Devuelve 3.

Contenido relacionado

[BITOR](#)

[BITAND](#)

[BITLSHIFT](#)

[BITRSHIFT](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COALESCE

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la primera expresión que no se evalúe como BLANK. Si todas las expresiones se evalúan como BLANK, se devuelve BLANK.

Sintaxis

DAX

```
COALESCE(<expression>, <expression>[, <expression>]...)
```

Parámetros

Expandir tabla

Término	Definición
expression	Cualquier expresión DAX que devuelve una expresión escalar.

Valor devuelto

Un valor escalar procedente de una de las expresiones o BLANK si todas las expresiones se evalúan como BLANK.

Notas

Las expresiones de entrada pueden ser de tipos de datos diferentes.

Ejemplo 1

La consulta DAX siguiente:

DAX

```
EVALUATE { COALESCE(BLANK(), 10, DATE(2008, 3, 3)) }
```

Devuelve `10`, que es la primera expresión que no se evalúe como BLANK.

Ejemplo 2

La expresión DAX siguiente:

DAX

```
= COALESCE(SUM(FactInternetSales[SalesAmount]), 0)
```

Devuelve la suma de todos los valores de la columna SalesAmount de la tabla FactInternetSales, o `0`. Se puede usar para convertir los valores en blanco de las ventas totales en `0`.

Comentarios

¿Le ha resultado útil esta página?



FALSE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el valor lógico FALSE.

Sintaxis

DAX

```
FALSE()
```

Valor devuelto

Siempre FALSE.

Notas

La palabra FALSE también se interpreta como el valor lógico FALSE.

Ejemplo

La fórmula devuelve el valor lógico FALSE cuando el valor de la columna 'InternetSales_USD'[SalesAmount_USD] es menor o igual que 200000.

DAX

```
= IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())
```

En la tabla siguiente se muestran los resultados cuando se usa la fórmula de ejemplo con 'ProductCategory'[ProductName] en las etiquetas de fila y 'DateTime'[CalendarYear] en las etiquetas de columna.

Expandir tabla

Etiquetas de fila	2005	2006	2007	2008	-	Total general
Accesorios	FALSO	false	VERDADERO	TRUE	false	VERDADERO

Etiquetas de fila	2005	2006	2007	2008	-	Total general
Bicicletas	VERDADERO	TRUE	TRUE	TRUE	false	VERDADERO
Ropa	FALSO	false	false	false	false	VERDADERO
Componentes	FALSO	false	false	false	false	false
	false	false	false	false	false	FALSO
Total general	VERDADERO	TRUE	TRUE	TRUE	false	VERDADERO

Contenido relacionado

[TRUE, función](#)

[NOT, función](#)

[Función IF](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

IF

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Comprueba una condición y devuelve un valor cuando es "true"; en caso contrario, devuelve un segundo valor.

Sintaxis

DAX

```
IF(<logical_test>, <value_if_true>[, <value_if_false>])
```

Parámetros

Expandir tabla

Término	Definición
logical_test	Cualquier valor o expresión que pueda evaluarse como TRUE o FALSE.
value_if_true	Valor que se devuelve si la prueba lógica es "true".
value_if_false	(Opcional) Valor que se devuelve si la prueba lógica es "false". Si se omite, se devuelve BLANK.

Valor devuelto

value_if_true, value_if_false o BLANK.

Comentarios

- La función IF puede devolver un tipo de datos variante si value_if_true y value_if_false son de diferentes tipos de datos, pero la función intenta devolver un solo tipo de datos si tanto value_if_true como value_if_false son de tipos de datos numéricos. En el último caso, la función IF convertirá de forma implícita los tipos de datos para alojar ambos valores.

Por ejemplo, la fórmula `IF(<condition>, TRUE(), 0)` devuelve TRUE o 0, pero la fórmula `IF(<condition>, 1.0, 0)` devuelve solo valores decimales aunque `value_if_false` sea de todo el tipo de datos numéricos. Para obtener más información sobre la conversión implícita de tipos de datos, vea [Tipos de datos](#).

- Para ejecutar las expresiones de bifurcación independientemente de la expresión de condición, utilice `IF.EAGER` en su lugar.

Ejemplos

En las siguientes definiciones de columna calculada de la tabla **Product** se usa la función IF de maneras diferentes para clasificar cada producto en función de su precio de venta.

En el primer ejemplo se prueba si el valor de la columna **List Price** es menor que 500. Cuando esta condición es "true", se devuelve el valor **Low**. Dado que no hay ningún valor en `value_if_false`, se devuelve BLANK.

Los ejemplos de este artículo se pueden usar con el modelo de ejemplo de Power BI Desktop de Adventure Works DW 2020. Para obtener el modelo, vea [Modelo de ejemplo de DAX](#).

DAX

```
Price Group =  
IF(  
    'Product'[List Price] < 500,  
    "Low"  
)
```

En el segundo ejemplo se usa la misma prueba, pero esta vez incluye un valor en `value_if_false`. Así pues, la fórmula clasifica cada producto como **Low** o **High**.

DAX

```
Price Group =  
IF(  
    'Product'[List Price] < 500,  
    "Low",  
    "High"  
)
```

En el tercer ejemplo se usa la misma prueba, pero esta vez anida una función IF para realizar una prueba adicional. Así pues, la fórmula clasifica cada producto como **Low**, **Medium** o **High**.

DAX

```
Price Group =  
IF(  
    'Product'[List Price] < 500,  
    "Low",  
    IF(  
        'Product'[List Price] < 1500,  
        "Medium",  
        "High"  
    )  
)
```

💡 Sugerencia

Cuando se necesiten anidar varias funciones IF, la función **SWITCH** podría ser una opción mejor. Esta función proporciona una manera más elegante de escribir una expresión que devuelva más de dos valores posibles.

Contenido relacionado

[Función IF.EAGER](#)

[Función SWITCH \(DAX\)](#)

[Funciones lógicas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

IF.EAGER

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Comprueba una condición y devuelve un valor cuando es "TRUE"; en caso contrario, devuelve un segundo valor. Utiliza un plan de ejecución *diligente* que siempre ejecuta las expresiones de bifurcación independientemente de la expresión de condición.

Sintaxis

DAX

```
IF.EAGER(<logical_test>, <value_if_true>[, <value_if_false>])
```

Parámetros

Expandir tabla

Término	Definición
logical_test	Cualquier valor o expresión que pueda evaluarse como TRUE o FALSE.
value_if_true	Valor que se devuelve si la prueba lógica es "true".
value_if_false	(Opcional) Valor que se devuelve si la prueba lógica es "false". Si se omite, se devuelve BLANK.

Valor devuelto

value_if_true, value_if_false o BLANK.

Comentarios

- La función IF.EAGER puede devolver un tipo de datos variante si value_if_true y value_if_false son de diferentes tipos de datos, pero la función intenta devolver un solo tipo de datos si tanto value_if_true como value_if_false son de tipos de datos numéricos. En el último caso, la función IF.EAGER convertirá de forma implícita los tipos de datos para alojar ambos valores.

Por ejemplo, la fórmula `IF.EAGER(<condition>, TRUE(), 0)` devuelve TRUE o 0, pero la fórmula `IF.EAGER(<condition>, 1.0, 0)` devuelve solo valores decimales aunque `value_if_false` sea de todo el tipo de datos numéricos. Para obtener más información sobre la conversión implícita de tipos de datos, vea [Tipos de datos](#).

- IF.EAGER muestra el mismo comportamiento funcional que la función IF, pero el rendimiento puede variar debido a las diferencias de los planes de ejecución. `IF.EAGER(<logical_test>, <value_if_true>, <value_if_false>)` tiene el mismo plan de ejecución que la expresión DAX siguiente:

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)
```

Nota: Las dos expresiones de bifurcación se evalúan independientemente de la expresión de condición.

Ejemplos

Vea [Ejemplos de IF](#).

Contenido relacionado

[Función IF](#)

[Funciones lógicas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

IFERROR

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Evaluá una expresión y devuelve un valor especificado si la expresión devuelve un error; de lo contrario, devuelve el valor de la propia expresión.

Sintaxis

DAX

```
IFERROR(value, value_if_error)
```

Parámetros

Expandir tabla

Término	Definición
valor	Cualquier valor o expresión.
value_if_error	Cualquier valor o expresión.

Valor devuelto

Un escalar del mismo tipo que **value**

Notas

- Puede usar la función IFERROR para interceptar y controlar los errores en una expresión.
- Si **value** o **value_if_error** es una celda vacía, IFERROR lo trata como un valor de cadena vacía ("").
- La función IFERROR se basa en la función IF y usa los mismos mensajes de error, pero tiene menos argumentos. La relación entre la función IFERROR y la función IF es la siguiente:

```
IFERROR(A,B) := IF(ISERROR(A), B, A)
```

Los valores que se devuelven para A y B deben ser del mismo tipo de datos; por lo tanto, la columna o expresión utilizada para **value** y el valor devuelto para **value_if_error** deben ser del mismo tipo de datos.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).
- Para conocer los procedimientos recomendados al usar IFERROR, consulte [Uso adecuado de funciones de error](#).

Ejemplo

En el ejemplo siguiente se devuelve 9999 si la expresión 25/0 se evalúa como un error. Si la expresión devuelve un valor que no es un error, ese valor se pasa a la expresión de invocación.

DAX

```
= IFERROR(25/0,9999)
```

Contenido relacionado

[Funciones lógicas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NOT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Cambia FALSE por TRUE o TRUE por FALSE.

Sintaxis

DAX
<code>NOT(<logical>)</code>

Parámetros

Expandir tabla

Término	Definición
logical	Valor o expresión que se puede evaluar como TRUE o FALSE.

Valor devuelto

TRUE o FALSE.

Ejemplo

En el ejemplo siguiente se recuperan los valores de la columna calculada que se ha creado para ilustrar la función IF. En ese ejemplo, el nombre de la columna calculada era el predeterminado, **Calculated Column1**, y esta contiene la siguiente fórmula: `=`
`IF([Orders]<300,"true","false")`.

La fórmula comprueba el valor de la columna [Orders] y devuelve "true" si el número de pedidos es inferior a 300.

Ahora, cree una columna calculada denominada **Calculated Column2** y escriba la fórmula siguiente.

DAX
<code>NOT([Calculated Column1])</code>

```
= NOT([CalculatedColumn1])
```

Para cada fila de **Calculated Column1**, los valores "true" y "false" se interpretan como los valores lógicos TRUE o FALSE, y la función NOT devuelve el opuesto lógico de ese valor.

Contenido relacionado

[TRUE, función](#)

[Función FALSE](#)

[Función IF](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

O BIEN

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Comprueba si uno de los argumentos es TRUE para devolver TRUE. La función devuelve FALSE si ambos argumentos son FALSE.

Sintaxis

DAX

```
OR(<logical1>,<logical2>)
```

Parámetros

Expandir tabla

Término	Definición
logical_1, logical_2	Valores lógicos que se quieren probar.

Valor devuelto

Valor booleano. El valor es TRUE si alguno de los dos argumentos es TRUE; el valor es FALSE si ambos argumentos son FALSE.

Observaciones

- La función OR de DAX acepta solo dos (2) argumentos. Si necesita realizar una operación OR en varias expresiones, puede crear una serie de cálculos o, mejor, usar el operador OR (||) para combinarlas en una expresión más sencilla.
- La función evalúa los argumentos hasta el primer argumento TRUE y luego devuelve TRUE.

Ejemplo

En el ejemplo siguiente se muestra cómo usar la función OR para obtener los vendedores que pertenecen al Círculo de excelencia. El Círculo de excelencia reconoce a quienes han logrado más de un millón de dólares en ventas de bicicletas de paseo o ventas de más de dos millones y medio de dólares en 2007.

DAX

```
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"
    ,
    "")
```

Devuelve

 Expandir tabla

Etiquetas de fila	2005	2006	2007	2008	-	Total general
Abbas, Syed E						
Alberts, Amy E						
Ansman-Wolfe, Pamela O						
Blythe, Michael G	Círculo de excelencia					
Campbell, David R						
Carson, Jillian	Círculo de excelencia					
Ito, Shu K						
Jiang, Stephen Y						
Mensa-Annan, Tete						

Etiquetas de fila	2005	2006	2007	2008	-	Total general
A						
Mitchell, Linda C	Círculo de excelencia					
Pak, Jae B	Círculo de excelencia					
Reiter, Tsvi Michael						
Saraiva, José Edvaldo	Círculo de excelencia					
Tsoflias, Lynn N						
Valdez, Rachel B						
Vargas, Garrett R						
Varkey Chudukatil, Ranjit R						Círculo de excelencia
Total general	Círculo de excelencia					

Contenido relacionado

[Funciones lógicas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SWITCH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Evaluá una expresión en una lista de valores y devuelve una de varias expresiones de resultado posibles. Esta función se puede usar para evitar tener varias instrucciones **IF** anidadas.

Sintaxis

DAX

```
SWITCH(<expression>, <value>, <result>[, <value>, <result>]...[, <else>])
```

Parámetros

Expandir tabla

Término	Definición
expression	Cualquier expresión DAX que devuelve un único valor escalar, donde la expresión se va a evaluar varias veces (para cada fila o contexto).
value	Valor constante con el que van a coincidir los resultados de expression .
resultado	Cualquier expresión escalar que se evaluará si los resultados de expression coinciden con el argumento value correspondiente.
else	Cualquier expresión escalar que se evaluará si el resultado de expression no coincide con ninguno de los argumentos value .

Valor devuelto

Si hay una coincidencia con un **valor**, se devuelve un valor escalar del **resultado** correspondiente. Si no hay ninguna coincidencia con un **valor**, se devuelve un valor de **else**. Si ninguno de los **valores** coincide y **else** no se especifica, se devuelve BLANK.

Observaciones

- La expresión que se va a evaluar puede ser un valor constante o una expresión. Un uso común de esta función es establecer el primer parámetro en TRUE. Vea los ejemplos siguientes.
- Todas las expresiones **result** y la expresión **else** deben ser del mismo tipo de datos.
- El orden de las condiciones es importante. En cuanto un **valor** coincide, se devuelve el **resultado** correspondiente y no se evalúan otros **valores** posteriores. Asegúrese de que los **valores** más restrictivos que se van a evaluar se especifican antes de los **valores** menos restrictivos. Vea los ejemplos siguientes.

Ejemplos

Un uso común de SWITCH es comparar la expresión con valores constantes. En este ejemplo se crea una columna calculada de nombres de meses:

```
DAX

= SWITCH (
    [Month Number Of Year],
    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"
)
```

Otro uso común de SWITCH es reemplazar varias instrucciones IF anidadas. Esto se logra estableciendo la expresión en TRUE, como se muestra en el ejemplo siguiente, que compara Punto de nuevo pedido y Nivel de inventario de seguridad en productos para identificar posibles riesgos de quedarse sin existencias:

```
DAX

= SWITCH (
    TRUE,
    [Reorder Point] > [Safety Stock Level], "Good: Safety stock level exceeded",
    [Reorder Point] = [Safety Stock Level], "Minimal: Safety stock level met",
    [Reorder Point] < [Safety Stock Level], "At risk: Safety stock level low"
```

```
not met",
    ISBLANK ( [Reorder Point] ), "Incomplete: Reorder point not set",
    ISBLANK ( [Safety Stock Level] ), "Incomplete: Safety stock level
not set",
    "Unknown"
)
```

El orden de los **valores** es importante. En el ejemplo siguiente, el segundo **resultado** nunca se devuelve porque el primer valor es menos restrictivo que el segundo. El resultado de este ejemplo es siempre "A" o "C", pero nunca "B".

DAX

```
= SWITCH (
    TRUE,
    Product[Standard Cost] < 100, "A",
    Product[Standard Cost] < 10, "B",
    "C"
)
```

La siguiente instrucción devuelve un error porque los tipos de datos de los argumentos de **resultado** son diferentes. Tenga en cuenta que los tipos de datos de todos los argumentos de **resultado** y **else** deben ser los mismos.

DAX

```
= SWITCH (
    [Class],
    "L", "Large",
    "H", 0.1
)
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TRUE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el valor lógico TRUE.

Sintaxis

DAX

`TRUE()`

Valor devuelto

Siempre TRUE.

Notas

La palabra TRUE también se interpreta como el valor lógico TRUE.

Ejemplo

La fórmula devuelve el valor lógico TRUE cuando el valor de la columna 'InternetSales_USD'[SalesAmount_USD] es mayor que 200000.

DAX

`= IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())`

En la tabla siguiente se muestran los resultados cuando se usa la fórmula de ejemplo en un informe, con "ProductCategory"[ProductName] en las etiquetas de fila y "DateTime"[CalendarYear] en las etiquetas de columna.

Expandir tabla

Etiquetas de fila	2005	2006	2007	2008	-	Total general
Accesorios	FALSO	false	VERDADERO	TRUE	false	VERDADERO

Etiquetas de fila	2005	2006	2007	2008	-	Total general
Bicicletas	VERDADERO	TRUE	TRUE	TRUE	false	VERDADERO
Ropa	FALSO	false	false	false	false	VERDADERO
Componentes	FALSO	false	false	false	false	false
	false	false	false	false	false	FALSO
Total general	VERDADERO	TRUE	TRUE	TRUE	false	VERDADERO

Contenido relacionado

[FALSO](#)

[NOT](#)

[IF](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Funciones matemáticas y trigonométricas

Artículo • 20/10/2023

Las funciones matemáticas de las expresiones de análisis de datos (DAX) son muy similares a las funciones matemáticas y trigonométricas de Excel. En esta sección se enumeran las funciones matemáticas que proporciona DAX.

En esta categoría

Función	Descripción
ABS	Devuelve el valor absoluto de un número.
ACOS	Devuelve el arcocoseno, o coseno inverso, de un número.
ACOSH	Devuelve el coseno hiperbólico inverso de un número.
ACOT	Devuelve la arcocotangente, o cotangente inversa, de un número.
ACOTH	Devuelve la cotangente hiperbólica inversa de un número.
ASIN	Devuelve el arcoseno, o seno inverso, de un número.
ASINH	Devuelve el seno hiperbólico inverso de un número.
ATAN	Devuelve la arcotangente, o tangente inversa, de un número.
ATANH	Devuelve la tangente hiperbólica inversa de un número.
CEILING	Redondea al alza un número al entero o al múltiplo significativo más próximo.
CONVERT	Convierte una expresión de un tipo de datos a otro.
COS	Devuelve el coseno del ángulo dado.
COSH	Devuelve el coseno hiperbólico de un número.
COT	Devuelve la cotangente de un ángulo especificado en radianes.
COTH	Devuelve la cotangente hiperbólica de un ángulo hiperbólico.
CURRENCY	Evalúa el argumento y devuelve el resultado como un tipo de datos de moneda.
DEGREES	Convierte los radianes en grados.

Función	Descripción
DIVIDE	Realiza la división y devuelve un resultado alternativo o BLANK() al dividirlo entre 0.
EVEN	Devuelve el número redondeado al alza al entero par más cercano.
EXP	Devuelve e elevado a la potencia de un número determinado.
FACT	Devuelve el factorial de un número, igual a las series $1*2*3 *...*$, que finaliza en el número dado.
FLOOR	Redondea un número a la baja, hacia cero, al múltiplo significance más próximo.
GCD	Devuelve el máximo común divisor de dos o más enteros.
INT	Redondea un número a la baja hasta el entero más próximo.
ISO.CEILING	Redondea un número hacia arriba, al entero o al múltiplo significativo más próximo.
LCM	Devuelve el mínimo común múltiplo de enteros.
LN	Devuelve el logaritmo natural de un número.
LOG	Devuelve el logaritmo de un número en la base especificada.
LOG10	Devuelve el logaritmo en base 10 de un número.
MOD	Devuelve el resto después de dividir un número entre un divisor. El resultado siempre tiene el mismo signo que el divisor.
MROUND	Devuelve un número redondeado al múltiplo deseado.
ODD	Devuelve un número redondeado al alza al entero impar más cercano.
PI	Devuelve el valor de Pi (3,14159265358979) con precisión de 15 dígitos.
POWER	Devuelve el resultado de un número elevado a una potencia.
QUOTIENT	Realiza la división y solo devuelve la parte entera del resultado.
RADIANS	Convierte los grados en radianes.
RAND	Devuelve un número aleatorio mayor o igual que 0 y menor que 1 distribuido uniformemente.
RANDBETWEEN	Devuelve un número aleatorio en el intervalo que se encuentra entre dos números que especifique.
ROUND	Redondea un número al número especificado de dígitos.

Función	Descripción
ROUNDDOWN	Redondea un número a la baja, hacia el cero.
ROUNDUP	Redondea un número al alza, alejándose de 0 (cero).
SIGN	Determina el signo de un número, el resultado de un cálculo o un valor en una columna.
SIN	Devuelve el seno del ángulo dado.
SINH	Devuelve el seno hiperbólico de un número.
SQRT	Devuelve la raíz cuadrada de un número.
SQRTPI	Devuelve la raíz cuadrada de (número * pi).
TAN	Devuelve la tangente del ángulo dado.
TANH	Devuelve la tangente hiperbólica de un número.
TRUNC	Trunca un número decimal en uno entero al quitar la parte decimal o fraccionaria del número.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

ABS

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el valor absoluto de un número.

Sintaxis

DAX
<code>ABS(<number>)</code>

Parámetros

Expandir tabla

Término	Definición
número	Número cuyo valor absoluto se quiere obtener.

Valor devuelto

Número decimal.

Notas

El valor absoluto de un número es un número decimal, entero o decimal, sin su signo.

Puede usar la función ABS para asegurarse de que solo se devuelvan números no negativos de expresiones cuando estén anidadas en funciones que requieran un número positivo.

Ejemplo

En el ejemplo siguiente se devuelve el valor absoluto de la diferencia entre el precio de venta y el precio del distribuidor, que puede usar en una nueva columna calculada, **DealerMarkup**.

DAX
<code>ABS([Price] - [Cost])</code>

= ABS([DealerPrice]-[ListPrice])

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[SIGN, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ACOS

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el arcocoseno, o coseno inverso, de un número. El arcocoseno es el ángulo cuyo coseno es *number*. El ángulo devuelto se expresa en radianes en el intervalo de 0 (cero) a pi.

Sintaxis

DAX

`ACOS(number)`

Parámetros

Expandir tabla

Término	Definición
Número	El coseno del ángulo deseado y debe ser de -1 a 1.

Valor devuelto

Devuelve el arcocoseno, o coseno inverso, de un número.

Notas

Si quiere convertir el resultado de radianes a grados, multiplíquelo por 180/PI() o use la función DEGREES.

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= ACOS(-0.5)	Arcocoseno de -0,5 en radianes, $2\pi/3$.	2,094395102
= ACOS(-0.5)*180/PI()	Arcocoseno de -0,5 en grados.	120

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ACOSH

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el coseno hiperbólico inverso de un número. El número de fila debe ser mayor o igual que 1. El coseno hiperbólico inverso es el valor cuyo coseno hiperbólico es *number*, por lo que ACOSH(COSH(number)) es igual a *number*.

Sintaxis

DAX

`ACOSH(number)`

Parámetros

Expandir tabla

Término	Definición
número	Cualquier número real igual o mayor que 1.

Valor devuelto

Devuelve el coseno hiperbólico inverso de un número.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= ACOSH(1)	Coseno hiperbólico inverso de 1.	0
= ACOSH(10)	Coseno hiperbólico inverso de 10.	2,993228

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ACOT

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el valor principal del arco tangente, o cotangente inverso, de un número.

Sintaxis

DAX
ACOT(number)

Parámetros

Expandir tabla

Término	Definición
Número	El coseno del ángulo que desea. Debe ser un número real.

Valor devuelto

Un valor decimal sencillo.

Comentarios

¿Le ha resultado útil esta página?

Sí

No

ACOTH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la cotangente hiperbólica inversa de un número.

Sintaxis

DAX
ACOTH(number)

Parámetros

Expandir tabla

Término	Definición
Número	El valor absoluto del número debe ser mayor que 1.

Valor devuelto

Un valor decimal sencillo.

Observaciones

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Comentarios

¿Le ha resultado útil esta página?

Sí

No

ASIN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el arcoseno, o seno inverso, de un número. El arcoseno es el ángulo cuyo seno es *number*. El ángulo devuelto se expresa en radianes en el intervalo de -pi/2 a pi/2.

Sintaxis

DAX

`ASIN(number)`

Parámetros

Expandir tabla

Término	Definición
número	El seno del ángulo deseado y debe ser de -1 a 1.

Valor devuelto

Devuelve el arcoseno, o seno inverso, de un número.

Notas

Para expresar el arcoseno en grados, multiplique el resultado por 180/PI() o use la función DEGREES.

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= ASIN(-0.5)	Arcoseno de 0,5 en radianes, -pi/6	-0,523598776

Fórmula	Description	Resultado
= ASIN(-0.5)*180/PI()	Arcoseno de -0,5 en grados	-30
= DEGREES(ASIN(-0.5))	Arcoseno de -0,5 en grados	-30

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ASINH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el seno hiperbólico inverso de un número. El seno hiperbólico inverso es el valor cuyo seno hiperbólico es *number*, por lo que ASINH(SINH(*number*)) es igual a *number*.

Sintaxis

DAX

`ASINH(number)`

Parámetros

Expandir tabla

Término	Definición
número	Cualquier número real.

Valor devuelto

Devuelve el seno hiperbólico inverso de un número.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= ASINH(-2.5)	Seno hiperbólico inverso de -2,5	-1,647231146
= ASINH(10)	Seno hiperbólico inverso de 10	2,99822295

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ATAN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la arcotangente, o tangente inversa, de un número. La arcotangente es el ángulo cuya tangente es *number*. El ángulo devuelto se expresa en radianes en el intervalo de -pi/2 a pi/2.

Sintaxis

DAX

`ATAN(number)`

Parámetros

Expandir tabla

Término	Definición
número	Tangente del ángulo que quiere obtener.

Valor devuelto

Devuelve la tangente hiperbólica inversa de un número.

Notas

Para expresar la arcotangente en grados, multiplique el resultado por 180/PI() o use la función DEGREES.

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= ATAN(1)	Arcotangente de 1 en radianes, pi/4	0,785398163
= ATAN(1)*180/PI()	Arcotangente de 1 en grados	45

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ATANH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la tangente hiperbólica inversa de un número. El número debe estar comprendido entre -1 y 1 (se excluyen -1 y 1). La tangente hiperbólica inversa es el valor cuya tangente hiperbólica es *number*, por lo que ATANH(TANH(*number*)) es igual a *number*.

Sintaxis

DAX
<code>ATANH(number)</code>

Parámetros

Expandir tabla

Término	Definición
número	Cualquier número real entre 1 y -1.

Valor devuelto

Devuelve la tangente hiperbólica inversa de un número.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= ATANH(0,76159416)	Tangente hiperbólica inversa de 0,76159416	1,00000001
= ATANH(-0,1)		-0,100335348

Contenido relacionado

[Función ATAN](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CEILING

Artículo • 20/10/2023

Redondea al alza un número al entero o al múltiplo significativo más próximo.

Sintaxis

DAX

```
CEILING(<number>, <significance>)
```

Parámetros

Término	Definición
número	Número que se quiere redondear, o bien referencia a una columna que contiene números.
significance	El múltiplo significativo al que se quiere redondear. Por ejemplo, para redondear al entero más próximo, escriba 1.

Valor devuelto

Un número redondeado como se ha especificado.

Notas

- Hay dos funciones CEILING en DAX, con las diferencias siguientes:
 - La función CEILING emula el comportamiento de la función CEILING de Excel.
 - La función ISO.CEILING sigue el comportamiento definido por ISO para determinar el valor del límite superior.
- Las dos funciones devuelven el mismo valor para los números positivos, pero valores distintos para los números negativos. Cuando se usa un múltiplo significativo positivo, tanto CEILING como ISO.CEILING redondean los números negativos al alza (hacia el infinito positivo). Cuando se usa un múltiplo significativo negativo, CEILING redondea los números negativos hacia abajo (hacia el infinito negativo), mientras que ISO.CEILING lo hace hacia arriba (hacia el infinito positivo).

- Normalmente, el tipo de valor devuelto es del mismo tipo que el argumento significativo, con las excepciones siguientes:
 - Si el tipo del argumento del número es moneda, el tipo del valor devuelto es moneda.
 - Si el tipo del argumento significativo es booleano, el tipo del valor devuelto es entero.
 - Si el tipo del argumento significativo no es numérico, el tipo del valor devuelto es real.

Ejemplo 1

La fórmula siguiente devuelve 4,45. Esto puede resultar útil si quiere evitar el uso de unidades más pequeñas en los precios. Si un producto existente tiene un precio de 4,42 USD, puede usar CEILING para redondear hacia arriba los precios hasta la unidad de cinco céntimos más cercana.

DAX

```
= CEILING(4.42,0.05)
```

Ejemplo 2

La fórmula siguiente devuelve resultados similares a los del ejemplo anterior, pero usa valores numéricos almacenados en la columna **ProductPrice**.

DAX

```
= CEILING([ProductPrice],0.05)
```

Vea también

[Funciones matemáticas y trigonométricas](#)

[Función FLOOR](#)

[ISO.CEILING, función](#)

[ROUNDUP, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

CONVERT

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Convierte una expresión de un tipo de datos en otro.

Sintaxis

DAX
<code>CONVERT(<Expression>, <Datatype>)</code>

Parámetros

[] Expandir tabla

Término	Definición
Expresión	Cualquier expresión válida.
Datatype	Enumeración que incluye: INTEGER (número entero), DOUBLE (número decimal), STRING (texto), BOOLEAN (True/False), CURRENCY (número decimal fijo), DATETIME (fecha, hora, etc.).

Valor devuelto

Devuelve el valor de <Expression> convertido a <Datatype>.

Notas

- La función devuelve un error si no se puede convertir un valor al tipo de datos especificado.
- Las columnas calculadas de DAX deben ser de un solo tipo de datos. Dado que las funciones MEDIAN y MEDIANX en una columna de enteros devuelven tipos de datos mixtos, ya sea integer o double, la expresión de columna calculada siguiente devuelve un error como resultado: `MedianNumberCarsOwned =`
`MEDIAN(DimCustomer[NumberCarsOwned])`.

- Para evitar tipos de datos mixtos, cambie la expresión para que siempre devuelva el tipo de datos double, por ejemplo:

```
MedianNumberCarsOwned = MEDIANX(DimCustomer, CONVERT([NumberCarsOwned],  
DOUBLE)).
```

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Consulta DAX

```
DAX  
  
EVALUATE { CONVERT(DATE(1900, 1, 1), INTEGER) }
```

Devuelve

 Expandir tabla

[Valor]
2

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COS

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el coseno del ángulo dado.

Sintaxis

DAX
<code>COS(number)</code>

Parámetros

Expandir tabla

Término	Definición
número	Requerido. Ángulo en radianes cuyo coseno se quiere obtener.

Valor devuelto

Devuelve el coseno del ángulo dado.

Notas

Si el ángulo está en grados, multiplíquelo por PI()/180 o use la función RADIANS para convertirlo en radianes.

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= COS(1,047)	Coseno de 1,047 radianes	0,5001711
= COS(60*PI()/180)	Coseno de 60 grados	0,5

Fórmula	Description	Resultado
= COS(RADIANS(60))	Coseno de 60 grados	0,5

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COSH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el coseno hiperbólico de un número.

Sintaxis

DAX

`COSH(number)`

Parámetros

Expandir tabla

Término	Definición
número	Obligatorio. Cualquier número real cuyo coseno hiperbólico se quiere obtener.

Valor devuelto

Coseno hiperbólico de un número.

Notas

- La fórmula del coseno hiperbólico es:

$$\text{COSH}(z) = \frac{e^z + e^{-z}}{2}$$

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= COSH(4)	Coseno hiperbólico de 4.	27,308233
= COSH(EXP(1))	Coseno hiperbólico de la base del logaritmo natural.	7,6101251

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la cotangente de un ángulo especificado en radianes.

Sintaxis

DAX

`COT (<number>)`

Parámetros

Expandir tabla

Término	Definición
número	Ángulo en radianes para el que se quiere obtener la cotangente.

Valor devuelto

Cotangente del ángulo dado.

Notas

- El valor absoluto del número debe ser mayor que 2^{27} y no puede ser 0.
- Si el número está fuera de sus restricciones, se devuelve un error.
- Si el número es un valor no numérico, se devuelve un error.

Ejemplo

Consulte la siguiente consulta DAX:

DAX

```
EVALUATE { COT(30) }
```

Devuelve

 Expandir tabla

[Valor]
-0.156119952161659

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COTH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la cotangente hiperbólica de un ángulo hiperbólico.

Sintaxis

DAX
<code>COTH (<number>)</code>

Parámetros

Expandir tabla

Término	Definición
número	Ángulo hiperbólico en radianes para el que se desea la cotangente hiperbólica.

Valor devuelto

Cotangente hiperbólica del ángulo dado.

Notas

- La cotangente hiperbólica es una analogía de la cotangente ordinaria (circular).
- El valor absoluto del número debe ser menor que 2^{27} y no puede ser 0.
- Si el número está fuera de sus restricciones, se devuelve un error.
- Si el número es un valor no numérico, se devuelve un error.
- Se usa la siguiente ecuación:

$$\text{COTH}(N) = \frac{1}{\text{TANH}(N)} = \frac{\text{COSH}(N)}{\text{SINH}(N)} = \frac{e^N + e^{-N}}{e^N - e^{-N}}$$

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Consulte la siguiente consulta DAX:

```
DAX  
EVALUATE { COTH(2) }
```

Devuelve

 Expandir tabla

[Valor]
1.03731472072755

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MONEDA

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Evalúa el argumento y devuelve el resultado como un tipo de datos de moneda.

Sintaxis

DAX
<code>CURRENCY(<value>)</code>

Parámetros

Expandir tabla

Término	Definición
valor	Cualquier expresión DAX que devuelve un único valor escalar, donde la expresión se va a evaluar exactamente una vez antes de todas las demás operaciones.

Valor devuelto

Valor de la expresión que se ha evaluado y devuelto como un valor de tipo de moneda.

Observaciones

- La función CURRENCY redondea el quinto decimal significativo, en valor, para devolver el cuarto dígito decimal. Si el quinto decimal significativo es igual o mayor que 5 se redondeará. Por ejemplo, si el valor es 3,6666666666666, entonces al convertirlo en moneda se devolverá 3,6667 USD. Sin embargo, si el valor es 3,0123456789, entonces al convertirlo en moneda se devolverá 3,0123 USD.
- Si el tipo de datos de la expresión es TrueFalse, CURRENCY(<TrueFalse>) devolverá 1,0000 USD para los valores True y 0,0000 USD para los valores False.
- Si el tipo de datos de la expresión es Text, entonces CURRENCY(<Text>) intentará convertir texto en un número. Si la conversión se realiza correctamente, el número se convertirá en una moneda; de lo contrario, se devolverá un error.

- Si el tipo de datos de la expresión es DateTime, CURRENCY(<DateTime>) convertirá el valor datetime en un número y ese número en una moneda. Los valores datetime tienen una parte entera que representa el número de días entre la fecha dada y 1900-03-01, y una fracción que representa la fracción de un día (donde 12 horas o mediodía es 0,5 días). Si el valor de la expresión no es un valor datetime correcto, se devuelve un error.

Ejemplo

Convierta el número 1234,56 en el tipo de datos de moneda.

DAX

```
= CURRENCY(1234.56)
```

Devuelve el valor 1234,56000.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DEGREES

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Convierte los radianes en grados.

Sintaxis

DAX
<code>DEGREES(angle)</code>

Parámetros

Expandir tabla

Término	Definición
angle	Obligatorio. Ángulo en radianes que se quiere convertir.

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= DEGREES(PI())	Grados de pi radianes	180

Comentarios

¿Le ha resultado útil esta página?

Sí

No

DIVIDIR

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Realiza la división y devuelve el resultado alternativo o BLANK() si se divide entre 0.

Sintaxis

DAX

```
DIVIDE(<numerator>, <denominator> [,<alternateresult>])
```

Parámetros

Expandir tabla

Término	Definición
numerator	Dividendo o número que se va a dividir.
denominator	Divisor o número por el que se va a dividir.
alternateresult	(Opcional) Valor devuelto si la división entre cero da como resultado un error. Si no se proporciona, el valor predeterminado es BLANK().

Valor devuelto

- Número decimal.

Notas

- El resultado alternativo en la división por 0 debe ser una constante.
- Para ver los procedimientos recomendados al usar DIVIDE, consulte [Función DIVIDE frente al operador de división \(/\) en DAX](#).

Ejemplo

En el ejemplo siguiente se devuelve 2,5.

DAX

```
= DIVIDE(5,2)
```

Ejemplo 1

En el ejemplo siguiente se devuelve BLANK.

DAX

```
= DIVIDE(5,0)
```

Ejemplo 2

En el ejemplo siguiente se devuelve 1.

DAX

```
= DIVIDE(5,0,1)
```

Contenido relacionado

[QUOTIENT, función](#)

[Funciones matemáticas y trigonométricas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

EVEN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número redondeado al alza al entero par más próximo. Puede usar esta función para procesar elementos que vienen de dos en dos. Por ejemplo, un contenedor de empaquetado acepta filas de uno o dos elementos. El contenedor estará lleno cuando el número de elementos, redondeados al alza a la cantidad dos más cercana, coincida con la capacidad del contenedor.

Sintaxis

DAX
<code>EVEN(number)</code>

Parámetros

Expandir tabla

Término	Definición
número	Valor que se redondea.

Valor devuelto

Devuelve el número redondeado al alza al entero par más próximo.

Notas

- Si el número no es un valor numérico, EVEN devuelve el valor de error .
- Independientemente del signo del número, un valor se redondea al alza cuando se ajusta a partir de cero. Si el número es un entero par, no se produce ningún redondeo.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Fórmula	Description	Resultado
= EVEN(1.5)	Redondea 1,5 al entero par más próximo.	2
= EVEN(3)	Redondea 3 al entero par más próximo.	4
= EVEN(2)	Redondea 2 al entero par más próximo.	2
= EVEN(-1)	Redondea -1 al entero par más próximo.	-2

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

EXP

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve e elevado a la potencia de un número determinado. La constante e es igual a 2,71828182845904, la base del logaritmo natural.

Sintaxis

DAX

`EXP(<number>)`

Parámetros

Expandir tabla

Término	Definición
número	Exponente aplicado a la base e. La constante e es igual a 2,71828182845904, la base del logaritmo natural.

Valor devuelto

Un número decimal.

Excepciones

Notas

- EXP es el inverso de LN, que es el logaritmo natural del número especificado.
- Para calcular las potencias de las bases distintas de e, utilice el operador de exponentiación (^). Para obtener más información, vea [Referencia del operador DAX](#).

Ejemplo

La fórmula siguiente calcula e elevado a la potencia del número contenido en la columna [Power].

DAX

```
= EXP([Power])
```

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[LN, función](#)

[EXP, función](#)

[LOG, función](#)

[LOG, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

FACT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el factorial de un número, igual a las series $1*2*3 *...*$, que finaliza en el número dado.

Sintaxis

DAX

`FACT(<number>)`

Parámetros

Expandir tabla

Término	Definición
número	Número no negativo cuyo factorial se quiere calcular.

Valor devuelto

Número decimal.

Notas

- Si el número no es un entero, se trunca y se devuelve un error. Si el resultado es demasiado grande, se devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la fórmula siguiente se devuelve el factorial de la serie de enteros de la columna, `[Values]`.

DAX

= FACT([Values])

En la tabla siguiente se muestran los resultados esperados:

 Expandir tabla

Valores	Resultados
0	1
1	1
2	2
3	6
4	24
5	120
170	7,257415615308E+306

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[Función TRUNC](#)

Comentarios

¿Le ha resultado útil esta página?



Sí



No

FLOOR

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Redondea un número a la baja, hacia cero, al múltiplo significance más próximo.

Sintaxis

DAX

`FLOOR(<number>, <significance>)`

Parámetros

Expandir tabla

Término	Definición
número	Valor numérico que se quiere redondear.
significance	Múltiplo al que se quiere redondear. Los argumentos <code>number</code> y <code>significance</code> deben ser ambos positivos o negativos.

Valor devuelto

Número decimal.

Comentarios

- Si alguno de los argumentos no es numérico, FLOOR devuelve el valor de error `**#VALUE!**`.
- Si `number` y `significance` tienen signos distintos, FLOOR devuelve el valor de error `**#NUM!**`.
- Independientemente del signo del número, un valor se redondea a la baja cuando se ajusta a partir de cero. Si el número es un múltiplo exacto de `significance`, no se produce ningún redondeo.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula siguiente toma los valores de la columna [Coste total del producto] de la tabla InternetSales y redondea a la baja al múltiplo más próximo a 1.

DAX

```
= FLOOR(InternetSales[Total Product Cost], .1)
```

En la tabla siguiente se muestran los resultados esperados de algunos valores de ejemplo:

 Expandir tabla

Valores	Resultado esperado
10,8423	10,8
8,0373	8
2,9733	2,9

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

GCD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve el máximo común divisor de dos o más enteros. El máximo común divisor es el entero más grande que divide tanto number1 como number2 sin un resto.

Sintaxis

DAX

`GCD(number1, [number2], ...)`

Parámetros

Expandir tabla

Término	Definición
number1,	Number1 es obligatorio, los números siguientes son opcionales. De 1 a 255
number2,...	valores. Si algún valor no es un entero, se trunca.

Valor devuelto

El máximo común divisor de dos o más enteros.

Notas

- Si algún argumento no es numérico, GCD devuelve el valor de error #VALUE!
- Si algún argumento es menor que cero, GCD devuelve el valor de error .
- El número uno divide cualquier valor uniformemente.
- Un número primo solo tiene el mismo número y uno como divisores.
- Si un parámetro de GCD es $>= 2^{53}$, GCD devuelve el valor de error #NUM!
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Fórmula	Description	Resultado
= GCD(5, 2)	Máximo común divisor de 5 y 2.	1
= GCD(24, 36)	Máximo común divisor de 24 y 36.	12
= GCD(7, 1)	Máximo común divisor de 7 y 1.	1

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

INT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Redondea un número a la baja hasta el entero más próximo.

Sintaxis

DAX
<code>INT(<number>)</code>

Parámetros

Expandir tabla

Término	Definición
número	Número que se quiere redondear a la baja a un entero.

Valor devuelto

Número entero.

Notas

TRUNC e INT son similares en el aspecto de que devuelven enteros. TRUNC quita la parte fraccionaria del número. INT redondea los números a la baja al entero más cercano en función del valor de la parte fraccionaria del número. INT y TRUNC solo son diferentes cuando se usan números negativos: `TRUNC(-4.3)` devuelve -4, pero `INT(-4.3)` devuelve -5 porque -5 es el número inferior.

Ejemplo

La expresión siguiente redondea el valor a 1. Si se usa la función ROUND, el resultado sería 2.

DAX
<code>INT(1.2)</code>

= INT(1.5)

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[Función ROUND](#)

[ROUNDUP, función](#)

[ROUNDDOWN, función](#)

[Función MROUND](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ISO.CEILING

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Redondea al alza un número al entero o al múltiplo significativo más próximo.

Sintaxis

DAX
<code>ISO.CEILING(<number>[, <significance>])</code>

Parámetros

Expandir tabla

Término	Definición
número	Número que se quiere redondear, o bien referencia a una columna que contiene números.
significance	(Opcional) Múltiplo significativo al que se quiere redondear. Por ejemplo, para redondear al entero más próximo, escriba 1. Si no se especifica la unidad significativa, el número se redondea al alza al entero más próximo.

Valor devuelto

Número, del mismo tipo que el argumento *number*, redondeado según lo especificado.

Notas

Hay dos funciones CEILING en DAX, con las diferencias siguientes:

- La función CEILING emula el comportamiento de la función CEILING de Excel.
- La función ISO.CEILING sigue el comportamiento definido por ISO para determinar el valor del límite superior.

Las dos funciones devuelven el mismo valor para los números positivos, pero valores distintos para los números negativos. Cuando se usa un múltiplo significativo positivo,

tanto CEILING como ISO.CEILING redondean los números negativos al alza (hacia el infinito positivo). Cuando se usa un múltiplo significativo negativo, CEILING redondea los números negativos a la baja (hacia el infinito negativo), mientras que ISO.CEILING lo hace hacia arriba (hacia el infinito positivo).

Normalmente, el tipo de resultado es del mismo tipo que el significado usado como argumento, con las excepciones siguientes:

- Si el primer argumento es de tipo moneda, el resultado es de tipo moneda.
- Si el argumento opcional no está incluido, el resultado es de tipo entero.
- Si el argumento significativo es de tipo booleano, el resultado es de tipo entero.
- Si el argumento significativo es de tipo no numérico, el resultado es de tipo real.

Ejemplo: números positivos

La fórmula siguiente devuelve 4,45. Esto puede resultar útil si quiere evitar el uso de unidades más pequeñas en los precios. Si un producto existente tiene un precio de 4,42 USD, puede usar ISO.CEILING para redondear al alza los precios hasta la unidad de cinco céntimos más cercana.

DAX

```
= ISO.CEILING(4.42,0.05)
```

Ejemplo: Números negativos

La fórmula siguiente devuelve el valor de límite superior ISO de -4,40.

DAX

```
= ISO.CEILING(-4.42,0.05)
```

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[Función FLOOR](#)

[Función CEILING](#)

[ROUNDUP, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

LCM

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el mínimo común múltiplo de enteros. El mínimo común múltiplo es el entero positivo más pequeño que es múltiplo de todos los argumentos enteros number1, number2, etc. Use LCM para agregar fracciones con diferentes denominadores.

Sintaxis

DAX

`LCM(number1, [number2], ...)`

Parámetros

Expandir tabla

Término	Definición
number1, number2,...	Number1 es obligatorio, los números siguientes son opcionales. De 1 a 255 valores cuyo mínimo común múltiplo quiere. Si el valor no es un entero, se trunca.

Valor devuelto

Devuelve el mínimo común múltiplo de enteros.

Notas

- Si algún argumento no es numérico, LCM devuelve el valor de error #VALUE! .
- Si algún argumento es menor que cero, LCM devuelve el valor de error #NUM! .
- Si $LCM(a,b) >= 2^{53}$, LCM devuelve el valor de error #NUM! .
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Fórmula	Description	Resultado
= LCM(5, 2)	Mínimo común múltiplo de 5 y 2.	10
= LCM(24, 36)	Mínimo común múltiplo de 24 y 36.	72

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

LN

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el logaritmo natural de un número. Los logaritmos naturales se basan en la constante e (2,71828182845904).

Sintaxis

DAX

`LN(<number>)`

Parámetros

Expandir tabla

Término	Definición
número	El número positivo del que se quiere obtener el logaritmo natural.

Valor devuelto

Número decimal.

Notas

LN es la función inversa a EXP.

Ejemplo

En el ejemplo siguiente se devuelve el logaritmo natural del número de la columna, `[Values]`.

DAX

`= LN([Values])`

Contenido relacionado

Funciones matemáticas y trigonométricas
EXP, función

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

REGISTRO

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el logaritmo de un número en la base especificada.

Sintaxis

DAX

`LOG(<number>,<base>)`

Parámetros

Expandir tabla

Término	Definición
número	El número positivo para el que se quiere obtener el logaritmo.
base	La base del logaritmo. Si se omite, la base es 10.

Valor devuelto

Número decimal.

Notas

Es posible que reciba un error si el valor es demasiado grande para mostrarlo.

La función LOG10 es similar, pero siempre devuelve el logaritmo común, es decir, el logaritmo para la base 10.

Ejemplo

Las fórmulas siguientes devuelven el mismo resultado, 2.

DAX

```
= LOG(100,10)
= LOG(100)
= LOG10(100)
```

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[EXP, función](#)

[LOG, función](#)

[LOG, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

LOG10

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el logaritmo en base 10 de un número.

Sintaxis

DAX

```
LOG10(<number>)
```

Parámetros

Expandir tabla

Término	Definición
número	Número positivo para el que se quiere obtener el logaritmo en base 10.

Valor devuelto

Número decimal.

Notas

La función LOG permite cambiar la base del logaritmo, en lugar de usar la base 10.

Ejemplo

Las fórmulas siguientes devuelven el mismo resultado, 2:

DAX

```
= LOG(100,10)  
= LOG(100)  
= LOG10(100)
```

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[EXP, función](#)

[LOG, función](#)

[LOG, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MOD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el resto después de dividir un número entre un divisor. El resultado siempre tiene el mismo signo que el divisor.

Sintaxis

DAX

`MOD(<number>, <divisor>)`

Parámetros

Expandir tabla

Término	Definición
número	El número para el que se quiere encontrar el resto después de realizar la división.
divisor	El número por el que se quiere dividir.

Valor devuelto

Número entero.

Notas

- Si el divisor es 0 (cero), MOD devuelve un error. No se puede dividir por 0.
- La función MOD se puede expresar en términos de la función INT: $MOD(n, d) = n - d * INT(n/d)$

Ejemplo 1

La fórmula siguiente devuelve 1, el resto de 3 dividido entre 2.

DAX

= **MOD(3,2)**

Ejemplo 2

La fórmula siguiente devuelve -1, el resto de 3 dividido entre 2. Tenga en cuenta que el signo es siempre el mismo que el del divisor.

DAX

= **MOD(-3,-2)**

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[Función ROUND](#)

[ROUNDUP, función](#)

[ROUNDDOWN, función](#)

[Función MROUND](#)

[Función INT](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MROUND

Artículo • 30/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve un número redondeado al múltiplo que se quiere obtener.

Sintaxis

DAX

```
MROUND(<number>, <multiple>)
```

Parámetros

[] Expandir tabla

Término	Definición
número	Número que se va a redondear.
multiple	Múltiplo significativo al que se quiere redondear el número.

Valor devuelto

Número decimal.

Notas

MROUND se redondea al alza, alejándose de cero, si el resto de la división de **number** por el **multiple** especificado es mayor o igual que la mitad del valor de **multiple**.

Ejemplo: Posiciones decimales

Esta expresión redondea 1,3 al múltiplo más próximo de 0,2. El resultado esperado es 1,4.

DAX

```
= MROUND(1.3,0.2)
```

Ejemplo: Números negativos

Esta expresión redondea -10 al múltiplo más próximo de -3. El resultado esperado es -9.

DAX

```
= MROUND(-10,-3)
```

Ejemplo: Error

Esta expresión devuelve un error, porque los números tienen signos diferentes.

DAX

```
= MROUND(5,-2)
```

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[Función ROUND](#)

[ROUNDUP, función](#)

[ROUNDDOWN, función](#)

[Función MROUND](#)

[Función INT](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ODD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve un número redondeado al alza al entero impar más cercano.

Sintaxis

DAX
<code>ODD(number)</code>

Parámetros

Expandir tabla

Término	Definición
número	Obligatorio. Valor que se redondea.

Valor devuelto

Devuelve un número redondeado al alza al entero impar más cercano.

Notas

- Si el número no es un valor numérico, ODD devuelve el valor de error "#VALUE!" .
- Independientemente del signo del número, un valor se redondea al alza cuando se ajusta a partir de cero. Si el número es un entero impar, no se produce ningún redondeo.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Fórmula	Description	Resultado
= ODD(1.5)	Redondea 1,5 al alza al entero impar más cercano.	3
= ODD(3)	Redondea 3 al alza al entero impar más cercano.	3
= ODD(2)	Redondea 2 al alza al entero impar más cercano.	3
= ODD(-1)	Redondea -1 al alza al entero impar más cercano.	-1
= ODD(-2)	Redondea -2 al alza (alejándose de 0) al entero impar más cercano.	-3

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PI

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el valor de Pi (3,14159265358979) con precisión de 15 dígitos.

Sintaxis

DAX

`PI()`

Valor devuelto

Un número decimal con el valor de Pi, 3,14159265358979, con una precisión de 15 dígitos.

Notas

Pi es una constante matemática. En DAX, Pi se representa como un número real con una precisión de 15 dígitos, igual que en Excel.

Ejemplo

Esta fórmula calcula el área de un círculo dado el radio de la columna, `[Radius]`.

DAX

`= PI()*([Radius]*2)`

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

POWER

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el resultado de un número elevado a una potencia.

Sintaxis

DAX

`POWER(<number>, <power>)`

Parámetros

Expandir tabla

Término	Definición
número	Número base, que puede ser cualquier número real.
potencia	Exponente al que se eleva el número base.

Valor devuelto

Número decimal.

Ejemplo

En el ejemplo siguiente se devuelve 25.

DAX

`= POWER(5,2)`

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

Comentarios

¿Le ha resultado útil esta página?



Sí



No

QUOTIENT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Realiza la división y solo devuelve la parte entera del resultado. Use esta función cuando quiera descartar el resto de la división.

Sintaxis

DAX

`QUOTIENT(<numerator>, <denominator>)`

Parámetros

Expandir tabla

Término	Definición
numerator	Dividendo o número que se va a dividir.
denominator	Divisor o número por el que se va a dividir.

Valor devuelto

Número entero.

Notas

- Si alguno de los argumentos es no numérico, QUOTIENT devuelve el valor de error **#VALUE!**.
- Puede usar una referencia de columna en lugar de un valor literal para cualquiera de los argumentos. En cambio, si la columna a la que se hace referencia contiene un 0 (cero), se devuelve un error para toda la columna de valores.

Ejemplo

Las fórmulas siguientes devuelven el mismo resultado, 2.

DAX

```
= QUOTIENT(5,2)
```

DAX

```
= QUOTIENT(10/2,2)
```

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

RADIANS

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Convierte los grados en radianes.

Sintaxis

DAX
<code>RADIANS(angle)</code>

Parámetros

Expandir tabla

Término	Definición
angle	Obligatorio. El ángulo en grados que quiere convertir.

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= RADIANS(270)	270 grados como radianes (4,712389 o $3\pi/2$ radianes)	4,712389

Comentarios

¿Le ha resultado útil esta página?

Sí

No

RAND

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve un número aleatorio mayor o igual que 0 y menor que 1 distribuido uniformemente. El número que se devuelve cambia cada vez que se vuelve a calcular la celda que contiene esta función.

Sintaxis

DAX

`RAND()`

Valor devuelto

Número decimal.

Notas

- El recálculo depende de varios factores, por ejemplo si el modelo está establecido en el modo de recálculo **Manual** o **Automático**, y si se han actualizado los datos.
- RAND y otras funciones volátiles que no tienen valores fijos no siempre se vuelven a calcular. Por ejemplo, la ejecución de una consulta o el filtrado normalmente no hacen que se vuelvan a evaluar esas funciones. Pero los resultados de esas funciones se vuelven a calcular cuando se recalcula toda la columna. Estas situaciones incluyen la actualización de un origen de datos externo o la edición manual de datos que hace que se vuelvan a evaluar las fórmulas que contienen esas funciones.
- RAND siempre se vuelve a calcular si la función se usa en la definición de una medida.
- La función RAND no puede devolver un resultado de cero, para evitar errores como la división por cero.

Ejemplos

Para generar un número real aleatorio entre otros dos números, use lo siguiente:

DAX

```
= RAND()*(b-a)+a
```

Para generar un número aleatorio mayor que 0 y menor que 1, use lo siguiente:

DAX

```
= RAND()
```

Para generar un número aleatorio mayor que 0 y menor que 100, use lo siguiente:

DAX

```
= RAND()*100
```

Para generar un número entero aleatorio mayor que 0 y menor que 100, use lo siguiente:

DAX

```
INT(RAND()*100)
```

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

RANDBETWEEN

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve un número aleatorio en el intervalo que se encuentra entre dos números que especifique.

Sintaxis

DAX

`RANDBETWEEN(<bottom>, <top>)`

Parámetros

[+] Expandir tabla

Término	Definición
Bottom	Entero menor que devolverá la función.
Superior	Entero mayor que devolverá la función.

Valor devuelto

Número entero.

Notas

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La siguiente fórmula devuelve un número aleatorio entre 1 y 10.

DAX

= RANDBETWEEN(1,10)

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ROUND

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Redondea un número al número especificado de dígitos.

Sintaxis

DAX

`ROUND(<number>, <num_digits>)`

Parámetros

Expandir tabla

Término	Definición
número	Número que quiere redondear.
num_digits	Número de dígitos al que quiere redondear. Un valor negativo redondea los dígitos a la izquierda del separador decimal; un valor de cero se redondea al entero más próximo.

Valor devuelto

Número decimal.

Notas

- Si **num_digits** es mayor que 0 (cero), el número se redondea al número especificado de posiciones decimales.
- Si **num_digits** es 0, el número se redondea al entero más próximo.
- Si **num_digits** es menor que 0, el número se redondea a la izquierda del separador decimal.
- Funciones relacionadas
 - Para redondear siempre al alza (hacia arriba del cero), use la función ROUNDUP.

- Para redondear siempre a la baja (hacia el cero), use la función ROUNDOWN.
- Para redondear un número a un múltiplo específico (por ejemplo, para redondear al múltiplo más próximo de 0,5), use la función MROUND.
- Use las funciones TRUNC e INT para obtener la parte entera del número.

Ejemplo 1

La fórmula siguiente redondea 2,15 al alza a una posición decimal. El resultado esperado es 2,2.

```
DAX  
= ROUND(2.15,1)
```

Ejemplo 2

La fórmula siguiente redondea 21,5 a una posición decimal a la izquierda del separador decimal. El resultado esperado es 20.

```
DAX  
= ROUND(21.5,-1)
```

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[ROUND](#)

[ROUNDDOWN](#)

[MROUND](#)

[INT](#)

[TRUNC](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ROUNDDOWN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Redondea un número a la baja, hacia el cero.

Sintaxis

DAX

`ROUNDDOWN(<number>, <num_digits>)`

Parámetros

Expandir tabla

Término	Definición
número	Número real que quiera redondear a la baja.
num_digits	Número de dígitos al que quiere redondear. Los valores negativos se redondean a la izquierda del separador decimal; cero al entero más próximo.

Valor devuelto

Número decimal.

Notas

- Si **num_digits** es mayor que 0 (cero), el valor de **number** se redondea a la baja al número especificado de posiciones decimales.
- Si **num_digits** es 0, el valor de **number** se redondea a la baja al entero más próximo.
- Si **num_digits** es menor que 0, el valor de **number** se redondea a la baja a la izquierda del separador decimal.

- ROUNDDOWN se comporta como ROUND, con la excepción de que siempre redondea un número a la baja. La función INT también redondea a la baja, pero con INT el resultado siempre es un entero, mientras que con ROUNDDOWN se puede controlar la precisión del resultado.

Ejemplo 1

En el ejemplo siguiente se redondea 3,14159 a la baja a tres posiciones decimales. El resultado esperado es 3,141.

```
DAX  
= ROUNDDOWN(3.14159,3)
```

Ejemplo 2

En el ejemplo siguiente se redondea el valor de 31 415,92654 a la baja a dos posiciones decimales a la izquierda del separador decimal. El resultado esperado es 31 400.

```
DAX  
= ROUNDDOWN(31415.92654, -2)
```

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[ROUND](#)

[ROUNDUP](#)

[ROUNDDOWN](#)

[MROUND](#)

[INT](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ROUNDUP

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Redondea un número al alza, alejándose de 0 (cero).

Sintaxis

DAX

```
ROUNDUP(<number>, <num_digits>)
```

Parámetros

Expandir tabla

Término	Definición
número	Número real que se quiere redondear al alza.
num_digits	Número de dígitos al que se quiere redondear. Un valor negativo de num_digits redondea a la izquierda del separador decimal; si num_digits es cero o se omite, number se redondea al entero más próximo.

Valor devuelto

Número decimal.

Notas

- Si **num_digits** es mayor que 0 (cero), el número se redondea al alza al número especificado de posiciones decimales.
- Si **num_digits** es 0, el número se redondea al alza al entero más próximo.
- Si **num_digits** es menor que 0, el número se redondea al alza a la izquierda del separador decimal.
- ROUNDUP se comporta como ROUND, con la excepción de que siempre redondea un número al alza.

Ejemplo

La fórmula siguiente redondea Pi a cuatro posiciones decimales. El resultado esperado es 3,1416.

```
DAX
```

```
= ROUNDUP(PI(),4)
```

Ejemplo: decimales como segundo argumento

La fórmula siguiente redondea 1,3 al múltiplo más próximo de 0,2. El resultado esperado es 2.

```
DAX
```

```
= ROUNDUP(1.3,0.2)
```

Ejemplo: número negativo como segundo argumento

La fórmula siguiente redondea el valor de la columna **FreightCost**, con los resultados esperados mostrados en la tabla siguiente:

```
DAX
```

```
= ROUNDUP([Values],-1)
```

Si **num_digits** es menor que cero, el número de posiciones a la izquierda del separador decimal aumenta según el valor que se especifique.

 Expandir tabla

FreightCost	Resultado previsto
13,25	20
2.45	10
25,56	30
1,34	10

FreightCost	Resultado previsto
345,01	350

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[ROUND](#)

[ROUNDDOWN](#)

[MROUND](#)

[INT](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SIGN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Determina el signo de un número, el resultado de un cálculo o un valor en una columna. La función devuelve 1 si el número es positivo, 0 (cero) si el número es cero y -1 si el número es negativo.

Sintaxis

DAX

`SIGN(<number>)`

Parámetros

Expandir tabla

Término	Definición
número	Cualquier número real, una columna que contenga números o una expresión que se evalúe como un número.

Valor devuelto

Número entero. Los valores devueltos posibles son 1, 0 y -1.

Expandir tabla

Valor devuelto	Descripción
1	El número es positivo.
0	El número es cero.
-1	El número es negativo.

Ejemplo

La fórmula siguiente devuelve el signo del resultado de la expresión que calcula el precio de venta menos el costo.

DAX

```
= SIGN( ([Sale Price] - [Cost]) )
```

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SIN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el seno del ángulo dado.

Sintaxis

DAX
<code>SIN(number)</code>

Parámetros

Expandir tabla

Término	Definición
número	Obligatorio. Ángulo en radianes cuyo seno se quiere obtener.

Valor devuelto

Devuelve el seno del ángulo dado.

Notas

Si un argumento está en grados, multiplíquelo por PI()/180 o use la función RADIANS para convertirlo en radianes.

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= SIN(PI())	Seno de pi radianes (0, aproximadamente).	0,0
= SIN(PI()/2)	Seno de pi/2 radianes.	1,0

Fórmula	Description	Resultado
= SIN(30*PI()/180)	Seno de 30 grados.	0,5
= SIN(RADIANS(30))	Seno de 30 grados.	0,5

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SINH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el seno hiperbólico de un número.

Sintaxis

DAX

`SINH(number)`

Parámetros

Expandir tabla

Término	Definición
número	Obligatorio. Cualquier número real.

Valor devuelto

Devuelve el seno hiperbólico de un número.

Notas

- La fórmula del seno hiperbólico es:

$$\text{SINH}(z) = \frac{e^z - e^{-z}}{2}$$

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Probabilidad de obtener un resultado de menos de 1,03 segundos.

DAX

$$= 2.868 * \text{SINH}(0.0342 * 1.03)$$

Devuelve 0,1010491.

Comentarios

¿Le ha resultado útil esta página?



SQRT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve la raíz cuadrada de un número.

Sintaxis

DAX

`SQRT(<number>)`

Parámetros

Expandir tabla

Término	Definición
número	Número para el que se quiere obtener la raíz cuadrada, una columna que contiene números o una expresión que se evalúa como un número.

Valor devuelto

Número decimal.

Notas

Si el número es negativo, la función SQRT devuelve un error.

Ejemplo

La fórmula siguiente:

DAX

`= SQRT(25)`

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SQRTPI

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la raíz cuadrada de (número * pi).

Sintaxis

DAX
<code>SQRTPI(number)</code>

Parámetros

Expandir tabla

Término	Definición
número	Obligatorio. Número por el que se multiplica pi.

Valor devuelto

Devuelve la raíz cuadrada de (número * pi).

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= SQRTPI(1)	Raíz cuadrada de pi.	1,772454
= SQRTPI(2)	Raíz cuadrada de 2 * pi.	2,506628

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TAN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la tangente del ángulo dado.

Sintaxis

DAX
<code>TAN(number)</code>

Parámetros

Expandir tabla

Término	Definición
número	Obligatorio. Ángulo en radianes para el que se quiere obtener la tangente.

Valor devuelto

Devuelve la tangente del ángulo dado.

Notas

Si el argumento está en grados, multiplíquelo por PI()/180 o use la función RADIANS para convertirlo en radianes.

Ejemplo

Expandir tabla

Fórmula	Description	Resultado
= TAN(0.785)	Tangente de 0,785 radianes (0,99920)	0,99920
= TAN(45*PI()/180)	Tangente de 45 grados (1)	1

Fórmula	Description	Resultado
= TAN(RADIANS(45))	Tangente de 45 grados (1)	1

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TANH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve la tangente hiperbólica de un número.

Sintaxis

DAX
<code>TANH(number)</code>

Parámetros

Expandir tabla

Término	Definición
número	Obligatorio. Cualquier número real.

Valor devuelto

Devuelve la tangente hiperbólica de un número.

Notas

- La fórmula de la tangente hiperbólica es:

$$\text{TANH}(z) = \frac{\text{SINH}(z)}{\text{COSH}(z)}$$

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Expandir tabla

Fórmula	Description	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

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TRUNC

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Trunca un número decimal en uno entero al quitar la parte decimal o fraccionaria del número.

Sintaxis

DAX

```
TRUNC(<number>,<num_digits>)
```

Parámetros

Expandir tabla

Término	Definición
número	El número que se quiere truncar.
num_digits	Un número que especifica la precisión de truncado; si se omite, es 0 (cero)

Valor devuelto

Número entero.

Notas

TRUNC e INT son similares en el aspecto de que devuelven enteros. TRUNC quita la parte fraccionaria del número. INT redondea los números a la baja al entero más próximo en función del valor de la parte fraccionaria del número. INT y TRUNC solo son diferentes cuando se usan números negativos: `TRUNC(-4.3)` devuelve -4, pero `INT(-4.3)` devuelve -5 porque -5 es el número inferior.

Ejemplo 1

La fórmula siguiente devuelve 3, la parte entera de PI.

DAX

```
= TRUNC(PI())
```

Ejemplo 2

La fórmula siguiente devuelve -8, la parte entera de -8,9.

DAX

```
= TRUNC(-8.9)
```

Contenido relacionado

[Funciones matemáticas y trigonométricas](#)

[ROUND](#)

[ROUNDUP](#)

[ROUNDDOWN](#)

[MROUND](#)

[INT](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Otras funciones

Artículo • 20/10/2023

Estas funciones realizan acciones únicas que no se pueden definir por medio de ninguna de las categorías.

En esta categoría

Función	Descripción
BLANK	Devuelve un espacio en blanco.
ERROR	Genera un error con un mensaje de error.
EVALUATEANDLOG	Devuelve el valor del primer argumento y lo registra en un evento del generador de perfiles del registro de evaluación DAX.
TOCSV	Devuelve una tabla como una cadena en formato CSV.
TOJSON	Devuelve una tabla como una cadena en formato JSON.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

BLANK

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve un espacio en blanco.

Sintaxis

DAX

```
BLANK()
```

Valor devuelto

Espacio en blanco.

Notas

- Los espacios en blanco no son equivalentes a los valores NULL. DAX usa espacios en blanco para los valores NULL de base de datos y para las celdas en blanco de Excel.
- Algunas funciones de DAX tratan las celdas en blanco de forma ligeramente diferente a la de Microsoft Excel. Los espacios en blanco y las cadenas vacías ("") no siempre son equivalentes, pero algunas operaciones pueden tratarlos como tales.

Ejemplo

En el ejemplo siguiente se muestra cómo trabajar con espacios en blanco en fórmulas. La fórmula calcula la proporción de ventas entre los canales de revendedores e Internet. Pero antes de intentar calcular la proporción, se debe comprobar si hay valores cero en el denominador. Si el denominador es cero, se debe devolver un valor en blanco; de lo contrario, se calcula la proporción.

DAX

```
= IF( SUM(InternetSales_USD[SalesAmount_USD])= 0 , BLANK() ,  
SUM(ResellerSales_USD[SalesAmount_USD])/SUM(InternetSales_USD[SalesAmount_US
```

En la tabla se muestran los resultados esperados cuando se usa esta fórmula para crear una visualización de tablas.

 Expandir tabla

Etiquetas de fila	Accesorios	Bicicletas	Ropa	Total general
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 general	0,83	2,51	5,45	2,94

En el origen de datos original, la columna evaluada por la función BLANK puede haber incluido texto, cadenas vacías o valores NULL. Si el origen de datos original es una base de datos de SQL Server, los valores NULL y las cadenas vacías son diferentes tipos de datos. Pero para esta operación se realiza una conversión de tipo implícita y DAX los trata como iguales.

Contenido relacionado

[Funciones de texto](#)

[Función ISBLANK](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ERROR

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Genera un error con un mensaje de error.

Sintaxis

DAX
ERROR(<text>)

Parámetros

[] Expandir tabla

Término	Definición
texto	Cadena de texto que contiene un mensaje de error.

Valor devuelto

None

Notas

- La función ERROR se puede colocar en una expresión DAX en cualquier lugar en el que se espere un valor escalar.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

La consulta DAX siguiente:

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]

```

Produce un error y genera un mensaje de error que contiene "color rojo encontrado".

Ejemplo 2

La consulta DAX siguiente:

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]

```

Devuelve la tabla siguiente:

 Expandir tabla

DimProduct[Color]	[Medida]
Negro	Negro
Azul	Azul
Gris	Gris
Multi	Multi
N/D	NA
Rojo	Rojo
Plata	Plata

DimProduct[Color]	[Medida]
Plata\Negro	Plata\Negro
Blanco	Blanco
Amarillo	Amarillo

Dado que magenta no es uno de los colores del producto, la función ERROR no se ejecuta.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

EVALUATEANDLOG

Artículo • 20/10/2023

Devuelve el valor del primer argumento y lo registra en un evento del generador de perfiles del registro de evaluación DAX. Esta función solo es totalmente funcional en Power BI Desktop. Actúa como una simple función de acceso directo en otros entornos.

Sintaxis

DAX
<pre>EVALUATEANDLOG(<Value>, [Label], [MaxRows])</pre>

Parámetros

Término	Definición
Value	Cualquier expresión escalar o expresión de tabla que se va a evaluar y registrar.
Etiqueta	(Opcional) Cadena constante incluida en el texto JSON y la columna Label (Etiqueta) del evento del registro de evaluación DAX que se puede usar para identificar fácilmente una instancia de la llamada de función.
MaxRows	(Opcional) Número máximo de filas en el texto JSON del evento del registro de evaluación DAX cuando el primer argumento es una expresión de tabla. El valor predeterminado es 10.

Valor devuelto

Valor del primer argumento.

La estructura JSON registrada en un evento del generador de perfiles del registro de evaluación DAX incluye lo siguiente:

- "expression" es la versión de texto del primer argumento.
- "label" es el parámetro Label cuando se especifica en la expresión.
- "inputs" es una lista de columnas en el contexto de evaluación que afecta a los valores del primer argumento.
- "outputs" es una lista de una sola columna ("Value") cuando el primer argumento es una expresión escalar, y una lista de columnas de salida cuando el primer argumento es una expresión de tabla.

- "data" es una lista de valores de entrada y de salida cuando el primer argumento es una expresión escalar, y una lista de valores de entrada y las filas de salida correspondientes cuando el primer argumento es una expresión de tabla.
- "rowCount" es el número de filas cuando el primer argumento es una expresión de tabla. Aunque el número de filas de la salida JSON lo trunca el parámetro MaxRows, rowCount es el número real de filas sin truncamiento.

Observaciones

- Los eventos de seguimiento se pueden capturar mediante [SQL Server Profiler](#) y la herramienta [DAX Debug Output](#) de código abierto.
- Esta función se puede usar con casi cualquier subexpresión en una expresión DAX, y toda la expresión seguirá siendo válida.
- Cuando el primer argumento se evalúa varias veces en una sola consulta, la función genera un único evento del registro de evaluación DAX que contiene los valores de entrada junto con los valores de salida correspondientes.
- Cuando se especifica el parámetro label, su valor se devuelve tanto en la salida JSON como en la columna Label del evento del registro de evaluación DAX.
- Si el primer argumento es una expresión de tabla, solo se muestran las filas MaxRows superiores en el evento del registro de evaluación DAX.
- En algunos casos, esta función no se ejecuta debido a las optimizaciones.
- Si el evento del registro de evaluación DAX tiene más de un millón de caracteres, se trunca para conservar la estructura JSON correcta.

Ejemplo 1

La consulta DAX siguiente:

```
DAX

evaluate
SUMMARIZE(
    EVALUATEANDLOG(FILTER(Sales, [ProductKey] = 528)),
    Sales[SalesTerritoryKey],
    "sum",
    sum(Sales[Sales Amount])
)
```

Devuelve el siguiente evento del registro de evaluación DAX:

JSON

```
{  
    "expression": "FILTER(Sales, [ProductKey] = 528)",  
    "inputs": [],  
    "outputs": ["'Sales'[SalesOrderLineKey]", "'Sales'[ResellerKey]",  
    "'Sales'[CustomerKey]", "'Sales'[ProductKey]", "'Sales'[OrderDateKey]",  
    "'Sales'[DueDateKey]", "'Sales'[ShipDateKey]", "'Sales'[SalesTerritoryKey]",  
    "'Sales'[Order Quantity]", "'Sales'[Unit Price]", "'Sales'[Extended  
Amount]", "'Sales'[Product Standard Cost]", "'Sales'[Total Product Cost]",  
    "'Sales'[Sales Amount]", "'Sales'[Unit Price Discount Pct]"],  
    "data": [  
        {  
            "input": [],  
            "rowCount": 3095,  
            "output": [  
                [52174001, -1, 23785, 528, 20190707, 20190717, 20190714, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [52173001, -1, 26278, 528, 20190707, 20190717, 20190714, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [52082001, -1, 23831, 528, 20190705, 20190715, 20190712, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [52054002, -1, 11207, 528, 20190704, 20190714, 20190711, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [52036001, -1, 25337, 528, 20190704, 20190714, 20190711, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [51939002, -1, 23670, 528, 20190702, 20190712, 20190709, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [51911002, -1, 11746, 528, 20190701, 20190711, 20190708, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [51379003, -1, 13745, 528, 20190612, 20190622, 20190619, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [51264002, -1, 11282, 528, 20190605, 20190615, 20190612, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [51184003, -1, 11263, 528, 20190531, 20190610, 20190607, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0]  
            ]  
        }  
    ]  
}
```

Ejemplo 2

La siguiente consulta DAX con un argumento escalar y atributos variables:

DAX

```
evaluate  
SELECTCOLUMNS(
```

```
TOPN(5, Customer),
[Customer],
"Customer",
EVALUATEANDLOG([Customer] & ", " & [Country-Region], "customerLog")
)
```

Devuelve el siguiente evento del registro de evaluación DAX:

JSON

```
{
  "expression": "[Customer] & \", \" & [Country-Region]",
  "label": "customerLog",
  "inputs": ["'Customer'[Customer]", "'Customer'[Country-Region]"],
  "data": [
    {
      "input": ["Russell Xie", "United States"],
      "output": "Russell Xie, United States"
    },
    {
      "input": ["Savannah Baker", "United States"],
      "output": "Savannah Baker, United States"
    },
    {
      "input": ["Maurice Tang", "United States"],
      "output": "Maurice Tang, United States"
    },
    {
      "input": ["Emily Wood", "United States"],
      "output": "Emily Wood, United States"
    },
    {
      "input": ["Meghan Hernandez", "United States"],
      "output": "Meghan Hernandez, United States"
    }
  ]
}
```

Consulte también

[TOCSV](#)

[TOJSON](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

TOCSV

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve una tabla como una cadena en formato CSV.

Sintaxis

DAX

```
TOCSV(<Table>, [MaxRows], [Delimiter], [IncludeHeaders])
```

Parámetros

[] Expandir tabla

Término	Definición
Tabla	Tabla que se va a convertir a CSV.
MaxRows	(Opcional) Número máximo de filas que se van a convertir. El valor predeterminado es 10 filas.
Delimitador	(Opcional) Delimitador de columna. El valor predeterminado es una coma (",").
IncludeHeaders	(Opcional) Especifica un encabezado con el nombre de columna como primera fila. El valor predeterminado es True.

Valor devuelto

Cadena con una representación en CSV de la tabla.

Ejemplo

La consulta DAX siguiente:

DAX

```
EVALUATE  
{TOCSV(DimSalesTerritory)}
```

Devuelve:

```
'DimSalesTerritory'[SalesTerritoryKey], 'DimSalesTerritory'[SalesTerritoryAlt  
ernateKey], 'DimSalesTerritory'[SalesTerritoryRegion], 'DimSalesTerritory'[Sal  
esTerritoryCountry], 'DimSalesTerritory'[SalesTerritoryGroup]  
1,1,Northwest,United States,North America  
2,2,Northeast,United States,North America  
3,3,Central,United States,North America  
4,4,Southwest,United States,North America  
5,5,Southeast,United States,North America  
6,6,Canada,Canada,North America  
7,7,France,France,Europe  
8,8,Germany,Germany,Europe  
9,9,Australia,Australia,Pacific  
10,10,United Kingdom,United Kingdom,Europe
```

Contenido relacionado

[TOJSON](#)

[EVALUATEANDLOG](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TOJSON

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medición ✓ Cálculo visual

Devuelve una tabla como una cadena en formato JSON.

Sintaxis

DAX

```
TOJSON(<Table>, [MaxRows])
```

Parámetros

[] Expandir tabla

Término	Definición
Tabla	Tabla que se va a convertir a JSON.
MaxRows	(Opcional) Número máximo de filas que se van a convertir. El valor predeterminado es 10 filas.

Valor devuelto

Cadena con una representación en JSON de la tabla. La representación contiene los nombres de columna como "header", el recuento de filas como "rowCount" y los valores como "data".

Ejemplo

La consulta DAX siguiente:

DAX

```
EVALUATE  
{TOJSON(DimSalesTerritory)}
```

Devuelve:

JSON

```
{  
  "header": ["'DimSalesTerritory'[SalesTerritoryKey]",  
             "'DimSalesTerritory'[SalesTerritoryAlternateKey]",  
             "'DimSalesTerritory'[SalesTerritoryRegion]",  
             "'DimSalesTerritory'[SalesTerritoryCountry]",  
             "'DimSalesTerritory'[SalesTerritoryGroup]"],<br>  
  "rowCount": 11,  
  "data": [  
    [1, 1, "Northwest", "United States", "North America"],  
    [2, 2, "Northeast", "United States", "North America"],  
    [3, 3, "Central", "United States", "North America"],  
    [4, 4, "Southwest", "United States", "North America"],  
    [5, 5, "Southeast", "United States", "North America"],  
    [6, 6, "Canada", "Canada", "North America"],  
    [7, 7, "France", "France", "Europe"],  
    [8, 8, "Germany", "Germany", "Europe"],  
    [9, 9, "Australia", "Australia", "Pacific"],  
    [10, 10, "United Kingdom", "United Kingdom", "Europe"]  
  ]  
}
```

Contenido relacionado

[TOCSV](#)

[EVALUATEANDLOG](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Funciones primarias y secundarias

Artículo • 20/10/2023

Estas funciones administran los datos que se presentan como jerarquías de elementos primarios y secundarios. Para más información, vea [Descripción de las funciones para jerarquías de elementos primarios y secundarios en DAX](#).

En esta categoría

Función	Descripción
PATH	Devuelve una cadena de texto delimitada con los identificadores de todos los elementos primarios del identificador actual.
PATHCONTAINS	Devuelve TRUE si el <i>elemento</i> especificado existe en la <i>ruta de acceso</i> especificada.
PATHITEM	Devuelve el elemento situado en el valor de <i>position</i> especificado de una cadena que resulta de la evaluación de una función PATH.
PATHITEMREVERSE	Devuelve el elemento situado en el valor de <i>position</i> especificado de una cadena que resulta de la evaluación de una función PATH.
PATHLENGTH	Devuelve el número de elementos primarios del elemento especificado en un resultado de PATH determinado, incluido a sí mismo.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

Descripción de las funciones para jerarquías de elementos primarios y secundarios en DAX

Artículo • 20/10/2023

DAX proporciona cinco funciones para ayudar a los usuarios a administrar los datos presentados como una jerarquía de elementos primarios y secundarios en sus modelos. Con estas funciones, un usuario puede obtener el linaje completo de elementos primarios que tiene una fila, cuántos niveles tiene el linaje hasta el elemento primario superior, quién es el elemento primario n niveles por encima de la fila actual, quién es el descendiente n desde la parte superior de la jerarquía de filas actual y si un determinado elemento primario es un elemento primario de la jerarquía de filas actual.

Funciones de elementos primarios y secundarios en DAX

La tabla siguiente contiene una jerarquía de elementos primarios y secundarios en las columnas: **EmployeeKey** y **ParentEmployeeKey** que se usa en todos los ejemplos de funciones.

EmployeeKey	ParentEmployeeKey
112	
14	112
3	14
11	3
13	3
162	3
117	162
221	162
81	162

En la tabla anterior puede ver que el empleado 112 no tiene ningún elemento primario definido, el empleado 14 tiene al empleado 112 como jefe (ParentEmployeeKey), el

empleado 3 tiene al empleado 14 como jefe y los empleados 11, 13 y 162 tienen al empleado 3 como jefe. Lo anterior ayuda a comprender que el empleado 112 no tiene ningún jefe por encima, sino que es el jefe superior de todos los empleados que se muestran aquí; además, el empleado 3 depende del empleado 14 y los empleados 11, 13 y 162 depende del 3.

En la tabla siguiente se presentan las funciones disponibles, una breve descripción de la función y un ejemplo de ella con los mismos datos mostrados arriba.

Función PATH: devuelve un texto delimitado con los identificadores de todos los elementos primarios para la fila actual, a partir del más antiguo o superior hasta el actual.

EmployeeKey	ParentEmployeeKey	Ruta de acceso
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

Función PATHLENGTH: devuelve el número de niveles de un argumento PATH() determinado a partir del nivel actual hasta el nivel primario más antiguo o superior. En el ejemplo siguiente, la columna PathLength se define como "`= PATHLENGTH([Path])`"; en el ejemplo se incluyen todos los datos del ejemplo de Path() para ayudar a comprender cómo funciona esta función.

EmployeeKey	ParentEmployeeKey	Ruta de acceso	PathLength
112		112	1
14	112	112 14	2
3	14	112 14 3	3
11	3	112 14 3 11	4

EmployeeKey	ParentEmployeeKey	Ruta de acceso	PathLength
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

Función PATHITEM: devuelve el elemento situado en la posición especificada de un resultado similar a PATH(), contando de izquierda a derecha. En el ejemplo siguiente, la columna PathItem - 4th from left se define como "`= PATHITEM([Path], 4)`"; en este ejemplo se devuelve el argumento EmployeeKey situado en la cuarta posición de la cadena Path desde la izquierda, con los mismos datos del ejemplo de Path().

EmployeeKey	ParentEmployeeKey	Ruta de acceso	PathItem - 4th from left
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

Función PATHITEMREVERSE: devuelve el elemento situado en *position* de un resultado de función similar a PATH(), contando hacia atrás de derecha a izquierda.

En el ejemplo siguiente, la columna PathItemReverse - 3rd from right se define como "`= PATHITEMREVERSE([Path], 3)`"; en este ejemplo se devuelve el argumento EmployeeKey situado en la tercera posición de la cadena Path desde la derecha, con los mismos datos del ejemplo de Path().

EmployeeKey	ParentEmployeeKey	Ruta de acceso	PathItemReverse - 3rd from right
112		112	

EmployeeKey	ParentEmployeeKey	Ruta de acceso	PathItemReverse - 3rd from right
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

Función PATHCONTAINS: devuelve **TRUE** si el *elemento* especificado existe en la *ruta de acceso* indicada. En el ejemplo siguiente, la columna PathContains - employee 162 se define como "`= PATHCONTAINS([Path], "162")`"; en este ejemplo se devuelve **TRUE** si la ruta de acceso dada contiene el empleado 162. En este ejemplo se usan los resultados del ejemplo de Path() anterior.

EmployeeKey	ParentEmployeeKey	Ruta de acceso	PathContains - employee 162
112		112	FALSO
14	112	112 14	false
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	VERDADERO
117	162	112 14 3 162 117	VERDADERO

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

PATH

Artículo • 20/10/2023

Devuelve una cadena de texto delimitada con los identificadores de todos los elementos primarios del identificador actual, a partir del más antiguo hasta el actual.

Sintaxis

DAX

```
PATH(<ID_columnName>, <parent_columnName>)
```

Parámetros

Término	Definición
ID_columnName	Nombre de una columna existente que contiene el identificador único de las filas de la tabla. No puede ser una expresión. El tipo de datos del valor de <i>ID_columnName</i> debe ser texto o entero y también debe ser el mismo tipo de datos que la columna a la que se hace referencia en <i>parent_columnName</i> .
parent_columnName	Nombre de una columna existente que contiene el identificador único del elemento primario de la fila actual. No puede ser una expresión. El tipo de datos del valor de <i>parent_columnName</i> debe ser texto o entero y también debe ser el mismo tipo de datos que el valor de <i>ID_columnName</i> .

Valor devuelto

Cadena de texto delimitada que contiene los identificadores de todos los elementos primarios del identificador actual.

Observaciones

- Esta función se usa en tablas que tienen algún tipo de jerarquía interna para devolver los elementos relacionados con el valor de fila actual. Por ejemplo, en una tabla Employees que contiene empleados, los jefes de los empleados y los jefes de los jefes, puede devolver la ruta de acceso que conecta a un empleado con su jefe.

- La ruta de acceso no está restringida a un único nivel de relaciones de primario a secundario; puede devolver filas relacionadas que estén varios niveles por encima de la fila inicial especificada.
 - El delimitador usado para separar los antecesores es la barra vertical, "|".
 - Los valores de *ID_columnName* y *parent_columnName* deben tener el mismo tipo de datos, texto o entero.
 - Los valores de *parent_columnName* deben estar presentes en *ID_columnName*. Es decir, no se puede buscar un elemento primario si no hay ningún valor en el nivel secundario.
 - Si *parent_columnName* es BLANK, PATH() devuelve el valor *ID_columnName*. Es decir, si busca el jefe de un empleado pero la columna *parent_columnName* no tiene datos, la función PATH devuelve solo el identificador de empleado.
 - Si *ID_columnName* tiene duplicados y *parent_columnName* es igual para esos duplicados, PATH() devuelve el valor común *parent_columnName*; pero si el valor *parent_columnName* es diferente a esos duplicados, PATH() devuelve un error. Es decir, si tiene dos listas para el mismo identificador de empleado y tienen el mismo identificador de jefe, la función PATH devuelve el identificador de ese jefe. Pero si hay dos identificadores de empleado idénticos que tienen identificadores de jefe diferentes, la función PATH devuelve un error.
 - Si *ID_columnName* es BLANK, PATH() devuelve BLANK.
 - Si *ID_columnName* contiene una barra vertical "|", PATH() devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se crea una columna calculada que enumera todos los jefes de cada empleado.

DAX

```
= PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey])
```

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

PATHCONTAINS

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve TRUE si el *elemento* especificado existe en la *ruta de acceso* especificada.

Sintaxis

DAX

```
PATHCONTAINS(<path>, <item>)
```

Parámetros

[] Expandir tabla

Término	Definición
path	Cadena creada como resultado de la evaluación de una función PATH.
item	Expresión de texto que se va a buscar en el resultado de la ruta de acceso.

Valor devuelto

Valor TRUE si *item* existe en *path*; de lo contrario, FALSE.

Observaciones

- Si *item* es un número entero, se convierte en texto y luego se evalúa la función. Si se produce un error de conversión, la función devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se crea una columna calculada que toma un identificador de jefe y comprueba un conjunto de empleados. Si el identificador de jefe se encuentra entre la

lista de jefes devuelta por la función PATH, la función PATHCONTAINS devuelve true; de lo contrario, devuelve false.

DAX

```
= PATHCONTAINS(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]),  
"23")
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PATHITEM

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve el elemento situado en el valor de *position* especificado de una cadena que resulta de la evaluación de una función PATH. Las posiciones se cuentan de izquierda a derecha.

Sintaxis

DAX

```
PATHITEM(<path>, <position>[, <type>])
```

Parámetros

[] Expandir tabla

Término	Definición
path	Cadena de texto en forma de los resultados de una función PATH.
position	Expresión entera con la posición del elemento que se va a devolver.
tipo	(Opcional) Enumeración que define el tipo de datos del resultado:

type enumeration

[] Expandir tabla

Enumeración	Enumeración alternativa	Descripción
TEXT	0	Los resultados se devuelven con el tipo de datos de texto (valor predeterminado).
INTEGER	1	Los resultados se devuelven como números enteros.

Valor devuelto

Identificador devuelto por la función PATH en la posición especificada de la lista de identificadores. Los elementos devueltos por la función PATH se ordenan desde el más lejano hasta el actual.

Comentarios

- Esta función sirve para devolver un nivel específico de una jerarquía devuelta por una función PATH. Por ejemplo, podría devolver solo los gerentes en la posición más alta de la jerarquía con respecto a los empleados de una empresa.
- Si especifica un número para *position* que es menor que uno (1) o mayor que el número de elementos de *path*, la función PATHITEM devuelve BLANK.
- Si *type* no es un elemento de enumeración válido, se devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se devuelve el gerente de tercer nivel con respecto al empleado actual; toma los identificadores del empleado y el gerente como entrada de una función PATH que devuelve una cadena con la jerarquía de elementos primarios para el empleado actual. Desde esa cadena, PATHITEM devuelve la tercera entrada como un entero.

DAX

```
= PATHITEM(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), 3, 1)
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PATHITEMREVERSE

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve el elemento situado en la *position* especificada de una cadena que resulta de la evaluación de una función PATH. Las posiciones se cuentan hacia atrás de derecha a izquierda.

Sintaxis

DAX

```
PATHITEMREVERSE(<path>, <position>[, <type>])
```

Parámetros

[] Expandir tabla

Término	Definición
path	Cadena de texto resultante de la evaluación de una función PATH.
position	Expresión entera con la posición del elemento que se va a devolver. La posición se cuenta hacia atrás de derecha a izquierda.
tipo	(Opcional) Enumeración que define el tipo de datos del resultado:

type enumeration

[] Expandir tabla

Enumeración	Enumeración alternativa	Descripción
TEXT	0	Los resultados se devuelven con el tipo de datos de texto (valor predeterminado).
INTEGER	1	Los resultados se devuelven como números enteros.

Valor devuelto

Antecesor de la posición n de la ruta de acceso dada, contando desde el actual al más antiguo.

Comentarios

- Esta función se puede usar para obtener un elemento individual de una jerarquía resultante de una función PATH.
- En esta función se invierte el orden estándar de la jerarquía, de modo que los elementos más cercanos se enumeran en primer lugar; por ejemplo, si la función PATH devuelve una lista de jefes por encima de un empleado en una jerarquía, la función PATHITEMREVERSE devuelve el jefe inmediato del empleado en la posición 2, ya que la posición 1 contiene el identificador del empleado.
- Si el número especificado para *position* es menor que uno (1) o mayor que el número de elementos de *path*, la función PATHITEM devuelve BLANK.
- Si *type* no es un elemento de enumeración válido, se devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se toma una columna de identificador de empleado como entrada de una función PATH y se invierte la lista de elementos principales que se devuelven. La posición especificada es 3 y el tipo devuelto es 1; por lo tanto, la función PATHITEMREVERSE devuelve un entero que representa el jefe dos niveles por encima del empleado.

DAX

```
= PATHITEMREVERSE(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]),  
3, 1)
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PATHLENGTH

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de elementos primarios del elemento especificado en un resultado de PATH determinado, incluido a sí mismo.

Sintaxis

DAX

`PATHLENGTH(<path>)`

Parámetros

Expandir tabla

Término	Definición
path	Una expresión de texto resultante de la evaluación de una función PATH.

Valor devuelto

El número de elementos primarios del elemento especificado en un resultado de PATH determinado, incluido el elemento especificado.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se toma un identificador de empleado como entrada de una función PATH y se devuelve una lista de los administradores por encima de ese empleado en la jerarquía; la función PATHLENGTH toma ese resultado y cuenta los distintos niveles de empleados y administradores, incluido el empleado inicial.

DAX

```
= PATHLENGTH(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]))
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Funciones de relación

Artículo • 20/10/2023

Las funciones de esta categoría sirven para administrar y usar las relaciones entre las tablas.

En esta categoría

Función	Descripción
CROSSFILTER	Especifica la dirección de filtrado cruzado que se va a usar en un cálculo de una relación que existe entre dos columnas.
RELATED	Devuelve un valor relacionado de otra tabla.
RELATEDTABLE	Evalúa una expresión de tabla en un contexto modificado por los filtros especificados.
USERELATIONSHIP	Especifica la relación que se va a usar en un cálculo concreto como la que existe entre columnName1 y columnName2.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

CROSSFILTER

Artículo • 24/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Especifica la dirección de filtrado cruzado que se va a usar en un cálculo de una relación que existe entre dos columnas.

Sintaxis

DAX

`CROSSFILTER(<columnName1>, <columnName2>, <direction>)`

Parámetros

Expandir tabla

Término	Definición
columnName1	Nombre de una columna existente, con la sintaxis DAX estándar y completo, que normalmente representa el lado "varios" de la relación que se va a usar. Si los argumentos se proporcionan en orden inverso, la función los intercambia antes de usarlos. Este argumento no puede ser una expresión.
columnName2	Nombre de una columna existente, con la sintaxis DAX estándar y completo, que normalmente representa el lado "uno" o de búsqueda de la relación que se va a usar. Si los argumentos se proporcionan en orden inverso, la función los intercambia antes de usarlos. Este argumento no puede ser una expresión.
Dirección	<p>Dirección del filtro cruzado que se va a usar. Debe ser una de las siguientes:</p> <p>None: no se produce ningún filtrado cruzado en esta relación.</p> <p>Both: todos los filtros se aplican a ambos lados.</p> <p>OneWay: los filtros de un lado o el lado de búsqueda de una relación filtra el otro lado. Esta opción no se puede usar con una relación uno a uno. No use esta opción en una relación de varios a varios porque no está claro cuál es el lado de la búsqueda. use OneWay_LeftFiltersRight o OneWay_RightFiltersLeft en su lugar.</p> <p>OneWay_LeftFiltersRight: los filtros del lado de <columnName1> filtran el lado de <columnName2>. Esta opción no se puede usar con una relación uno a uno ni varios a uno.</p>

Término	Definición
	OneWay_RightFiltersLeft: los filtros del lado de <columnName2> filtran el lado de <columnName1>. Esta opción no se puede usar con una relación uno a uno ni varios a uno.

Valor devuelto

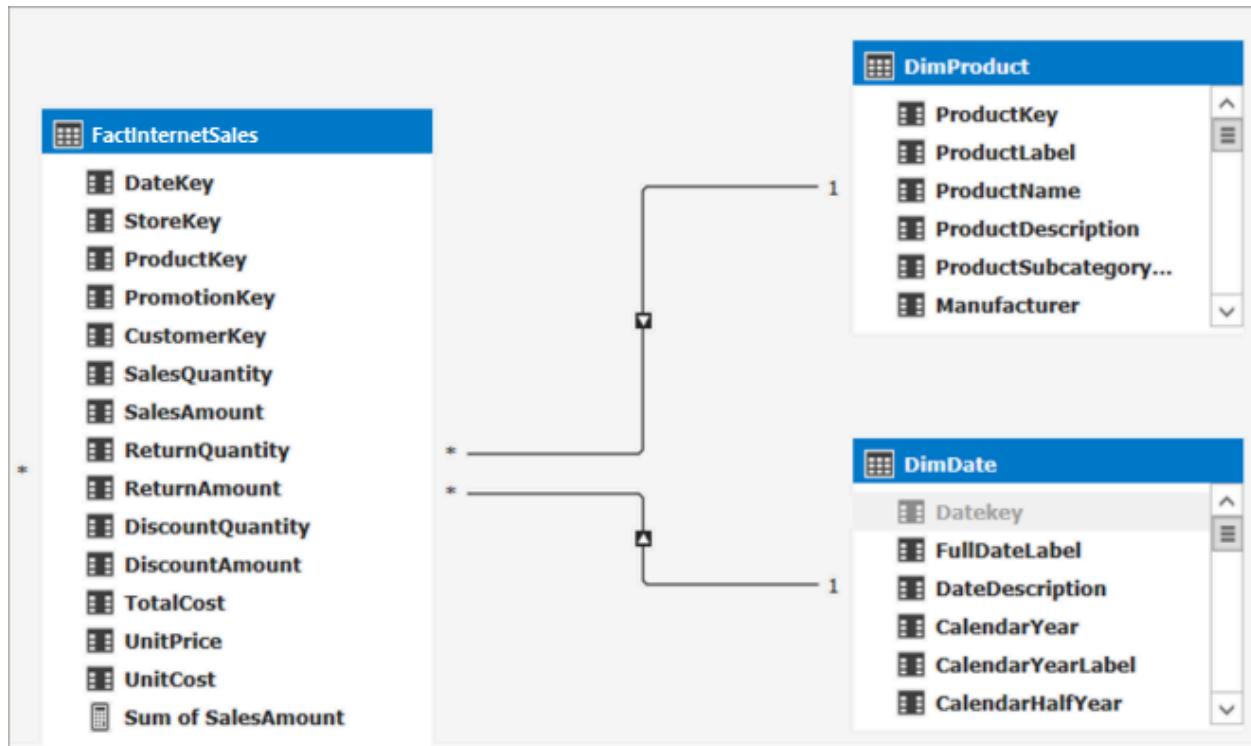
La función no devuelve ningún valor; solo establece la dirección del filtrado cruzado de la relación indicada mientras dura la consulta.

Observaciones

- En el caso de una relación 1:1, no hay ninguna diferencia entre las direcciones one y both.
- CROSSFILTER solo se puede usar en funciones que toman un filtro como argumento, por ejemplo: funciones CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD y TOTALYTD.
- CROSSFILTER usa las relaciones existentes en el modelo y las identifica por sus columnas de punto final.
- En CROSSFILTER, la configuración del filtrado cruzado de una relación no es importante; es decir, independientemente de si la relación está establecida para filtrar una o ambas direcciones del modelo, esto no afecta al uso de la función. CROSSFILTER invalidará cualquier configuración de filtrado cruzado existente.
- Se devuelve un error si alguna de las columnas denominadas como argumento no forma parte de una relación o los argumentos pertenecen a relaciones diferentes.
- Si las expresiones de CALCULATE están anidadas y más de una expresión de CALCULATE contiene una función CROSSFILTER, la instancia de CROSSFILTER más interna es la que prevalece en caso de conflicto o ambigüedad.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el siguiente diagrama de modelo, tanto DimProduct como DimDate tienen una relación de dirección única con FactInternetSales.



De forma predeterminada, no se puede obtener el número de productos vendidos por año:

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

Hay dos maneras de obtener el número de productos por año:

- Active el filtrado cruzado bidireccional en la relación. Esto cambiará el modo en que funcionan los filtros para todos los datos entre estas dos tablas.
- Use la función CROSSFILTER para cambiar el modo en que funcionan las relaciones solo para esta medida.

Al usar DAX, podemos utilizar la función CROSSFILTER para cambiar el comportamiento de la dirección del filtro cruzado entre dos columnas definidas por una relación. En este caso, la expresión DAX tiene el siguiente aspecto:

DAX

```
BiDi:= CALCULATE([Distinct Count of ProductKey],  
CROSSFILTER(FactInternetSales[ProductKey], DimProduct[ProductKey] , Both))
```

Mediante el uso de la función CROSSFILTER en nuestra expresión de medida, obtenemos los 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

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

RELATED

Artículo • 20/10/2023

Devuelve un valor relacionado de otra tabla.

Sintaxis

DAX

```
RELATED(<column>)
```

Parámetros

Término	Definición
columna	Columna que contiene los valores que quiere recuperar.

Valor devuelto

Valor único que está relacionado con la fila actual.

Notas

- La función RELATED requiere que haya una relación entre la tabla actual y la tabla con información relacionada. Especifique la columna que contiene los datos que quiere; la función seguirá una relación de varios a uno existente para capturar el valor de la columna especificada en la tabla relacionada. Si no existe ninguna relación, debe crear una.
- Cuando la función RELATED realiza una búsqueda, examina todos los valores de la tabla especificada, independientemente de los filtros que se puedan haber aplicado.
- La función RELATED necesita un contexto de fila; por lo tanto, solo se puede usar en una expresión de columna calculada, en la que el contexto de la fila actual es inequívoco, o bien como una función anidada en una expresión que usa una función de recorrido de tabla. Una función de recorrido de tabla, como SUMX, obtiene el valor de la fila actual y, después, busca instancias de ese valor en otra tabla.

- La función RELATED no se puede utilizar para capturar una columna a través de una [relación limitada](#).

Ejemplo

En el ejemplo siguiente, se crea la medida Ventas por Internet fuera de EE. UU. para generar un informe de ventas que excluya las ventas en Estados Unidos. Para crear la medida, la tabla InternetSales_USD se debe filtrar para excluir todas las ventas que pertenecen a Estados Unidos de la tabla SalesTerritory. Los Estados Unidos, como país, aparece cinco veces en la tabla SalesTerritory; una vez para cada una de las siguientes regiones: Northwest, Northeast, Central, Southwest y Southeast.

El primer enfoque para filtrar las ventas por Internet, con el fin de crear la medida, podría ser agregar una expresión de filtro similar a la siguiente:

DAX

```
FILTER('InternetSales_USD'
, 'InternetSales_USD'[SalesTerritoryKey]<>1 &&
'InternetSales_USD'[SalesTerritoryKey]<>2 &&
'InternetSales_USD'[SalesTerritoryKey]<>3 &&
'InternetSales_USD'[SalesTerritoryKey]<>4 &&
'InternetSales_USD'[SalesTerritoryKey]<>5)
```

Pero este enfoque es contradictorio, propenso a errores tipográficos y podría no funcionar si alguna de las regiones existentes se divide en el futuro.

Un mejor enfoque sería usar la relación existente entre InternetSales_USD y SalesTerritory, y establecer explícitamente que el país debe ser diferente de Estados Unidos. Para ello, cree una expresión de filtro similar a la siguiente:

DAX

```
FILTER( 'InternetSales_USD',
RELATED('SalesTerritory'[SalesTerritoryCountry])<>"United States")
```

Esta expresión usa la función RELATED para buscar el valor de país en la tabla SalesTerritory, empezando por el valor de la columna de clave, SalesTerritoryKey, en la tabla InternetSales_USD. La función de filtro usa el resultado de la búsqueda para determinar si la fila InternetSales_USD está filtrada o no.

ⓘ Nota

Si el ejemplo no funciona, puede que tenga que crear una relación entre las tablas.

DAX

```
= SUMX(FILTER('InternetSales_USD'  
    , RELATED('SalesTerritory'[SalesTerritoryCountry])  
    <>"United States"  
)  
'InternetSales_USD'[SalesAmount_USD])
```

En la tabla siguiente se muestran solo los totales de cada región, para demostrar que la expresión de filtro de la medida, Ventas por Internet fuera de EE. UU., funciona según lo previsto.

Row Labels	Internet Sales	Non USA Internet Sales
Australia	4 999 021,84 \$	4 999 021,84 \$
Canadá	1 343 109,10 \$	1 343 109,10 \$
Francia	2 490 944,57 \$	2 490 944,57 \$
Alemania	2 775 195,60 \$	2 775 195,60 \$
Reino Unido	5 057 076,55 \$	5 057 076,55 \$
Estados Unidos	9 389 479,79 \$	
Total general	26 054 827,45 \$	16 665 347,67 \$

En la tabla siguiente se muestra lo que podría obtener si ha usado esta medida en un objeto visual de tabla de informe:

Row Labels	Accessories	Bikes	Clothing	Grand Total
2005		1 526 481,95 \$		1 526 481,95 \$
2006		3 554 744,04 \$		3 554 744,04 \$
2007	156 480,18 \$	5 640 106,05 \$	70 142,77 \$	5 866 729,00 \$
2008	228 159,45 \$	5 386 558,19 \$	102 675,04 \$	5 717 392,68 \$
Total general	384 639,63 \$	16 107 890,23 \$	172 817,81 \$	16 665 347,67 \$

Vea también

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

RELATEDTABLE

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✗ Cálculo visual

Evalúa una expresión de tabla en un contexto modificado por los filtros especificados.

Sintaxis

DAX
<code>RELATEDTABLE(<tableName>)</code>

Parámetros

[] Expandir tabla

Término	Definición
tableName	Nombre de una tabla existente que usa la sintaxis DAX estándar. No puede ser una expresión.

Valor devuelto

Una tabla de valores.

Comentarios

- La función RELATEDTABLE cambia el contexto en el que se filtran los datos y evalúa la expresión en el nuevo contexto que se especifique.
- Esta función es un acceso directo a la función CALCULATETABLE sin ninguna expresión lógica.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se usa la función RELATEDTABLE para crear una columna calculada con las ventas por Internet en la tabla Product Category:

DAX

```
= SUMX( RELATEDTABLE('InternetSales_USD')
      , [SalesAmount_USD])
```

En la tabla siguiente se muestran los resultados:

 Expandir tabla

Product Category Key	Product Category AlternateKey	Product Category Name	Internet Sales
1	1	Bicicletas	28 318 144,65 \$
2	2	Componentes	
3	3	Ropa	339 772,61 \$
4	4	Accesorios	700 759,96 \$

Contenido relacionado

[CALCULATETABLE](#)

[Funciones de filtro](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

USERELATIONSHIP

Artículo • 20/10/2023

Especifica la relación que se va a usar en un cálculo concreto como la que existe entre columnName1 y columnName2.

Sintaxis

DAX

```
USERELATIONSHIP(<columnName1>,<columnName2>)
```

Parámetros

Término	Definición
columnName1	Nombre de una columna existente, con la sintaxis DAX estándar y completo, que normalmente representa el lado "varios" de la relación que se va a usar. Si los argumentos se proporcionan en orden inverso, la función los intercambia antes de usarlos. Este argumento no puede ser una expresión.
columnName2	Nombre de una columna existente, con la sintaxis DAX estándar y completo, que normalmente representa el lado "uno" o de búsqueda de la relación que se va a usar. Si los argumentos se proporcionan en orden inverso, la función los intercambia antes de usarlos. Este argumento no puede ser una expresión.

Valor devuelto

La función no devuelve ningún valor; solo habilita la relación indicada mientras dura el cálculo.

Notas

- USERELATIONSHIP solo se puede usar en funciones que toman un filtro como argumento, por ejemplo: funciones CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD y TOTALYTD.

- No se puede usar USERELATIONSHIP si se ha definido la seguridad de nivel de fila para la tabla en la que está incluida la medida. Por ejemplo, `CALCULATE(SUM([SalesAmount]), USERELATIONSHIP(FactInternetSales[CustomerKey], DimCustomer[CustomerKey]))` devuelve un error si se ha definido la seguridad de nivel de fila para DimCustomer.
- USERELATIONSHIP usa las relaciones existentes en el modelo y las identifica por sus columnas de punto final.
- En USERELATIONSHIP, el estado de una relación no es importante; es decir, si la relación está activa o no, no afecta al uso de la función. Se usa aun cuando la relación esté inactiva e invalida cualquier otra relación activa que esté presente en el modelo, pero que no se mencione en los argumentos de la función.
- Se devuelve un error si alguna de las columnas denominadas como argumento no forma parte de una relación o los argumentos pertenecen a relaciones diferentes.
- Si se necesitan varias relaciones para combinar la tabla A con la tabla B en un cálculo, cada relación debe indicarse en una función USERELATIONSHIP diferente.
- Si las expresiones de CALCULATE están anidadas y más de una expresión de CALCULATE contiene una función USERELATIONSHIP, la instancia de USERELATIONSHIP más interna es la que prevalece en caso de conflicto o ambigüedad.
- Se pueden anidar hasta 10 funciones USERELATIONSHIP; pero es posible que la expresión tenga un nivel más profundo de anidamiento, es decir, que la siguiente expresión de ejemplo esté anidada a tres niveles de profundidad, pero solo a dos para USERELATIONSHIP: `=CALCULATE(CALCULATE(CALCULATE(<anyExpression>, USERELATIONSHIP(t1[colA], t2[colB])), t99[colZ]=999), USERELATIONSHIP(t1[colA], t2[colA]))`.
- En el caso de las relaciones 1 a 1, USERELATIONSHIP solo activará la relación en una dirección. En concreto, los filtros solo podrán fluir de la tabla *columnName2* a la tabla *columnName1*. Si se desea el filtrado cruzado bidireccional, se pueden usar dos USERELATIONSHIP con direccionalidad opuesta en el mismo cálculo. Por ejemplo, `CALCULATE(..., USERELATIONSHIP(T1[K], T2[K]), USERELATIONSHIP(T2[K], T1[K]))`.

Ejemplo

En el ejemplo siguiente se muestra cómo invalidar la relación predeterminada activa entre las tablas InternetSales y DateTime. La relación predeterminada existe entre la columna OrderDate de la tabla InternetSales y la columna Date de la tabla DateTime.

Para calcular la suma de las ventas por Internet y permitir la segmentación por ShippingDate en lugar del tradicional OrderDate, debe crear una medida, [InternetSales por ShippingDate] mediante la expresión siguiente:

DAX

```
= CALCULATE(SUM(InternetSales[SalesAmount]),  
USERELATIONSHIP(InternetSales[ShippingDate], DateTime[Date]))
```

Las relaciones entre InternetSales[ShipmentDate] y DateTime[Date] deben existir y no deben ser la relación activa; además, la relación entre InternetSales[OrderDate] y DateTime[Date] debe existir y debe ser la relación activa.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

Funciones estadísticas

Artículo • 20/10/2023

Las funciones estadísticas calculan valores relacionados con las distribuciones estadísticas y la probabilidad, como la desviación estándar y el número de permutaciones.

En esta categoría

Función	Descripción
BETA.DIST	Devuelve la distribución beta.
BETA.INV	Devuelve el inverso de la función de densidad de probabilidad acumulativa beta (BETA.DIST).
CHISQ.DIST	Devuelve la distribución chi cuadrado.
CHISQ.DIST.RT	Devuelve la probabilidad de cola derecha de la distribución chi cuadrado.
CHISQ.INV	Devuelve el inverso de la probabilidad de cola izquierda de la distribución chi cuadrado.
CHISQ.INV.RT	Devuelve el inverso de la probabilidad de cola derecha de la distribución chi cuadrado.
COMBIN	Devuelve el número de combinaciones de una cantidad determinada de elementos.
COMBINA	Devuelve la cantidad de combinaciones (con repeticiones) de una cantidad determinada de elementos.
CONFIDENCE.NORM	El intervalo de confianza es un rango de valores.
CONFIDENCE.T	Devuelve el intervalo de confianza de una media de población con una distribución t de Student.
EXPON.DIST	Devuelve la distribución exponencial.
GEOMEAN	Devuelve la media geométrica de los números de una columna.
GEOMEANX	Devuelve la media geométrica de una expresión evaluada de cada fila de una tabla.
LINEST	Usa el método de mínimos cuadrados para calcular una línea recta con el mejor ajuste a los datos especificados.

Función	Descripción
LINESTX	Usa el método de mínimos cuadrados para calcular una línea recta con el mejor ajuste a los datos especificados. Los datos se obtienen de las expresiones evaluadas para cada fila de una tabla.
MEDIAN	Devuelve la mediana de números de una columna.
MEDIANX	Devuelve la mediana de una expresión evaluada para cada fila de una tabla.
NORM.DIST	Devuelve la distribución normal para la media y la desviación estándar especificadas.
NORM.INV	Inverso de la distribución acumulativa normal para la media y la desviación estándar especificadas.
NORM.S.DIST	Devuelve la distribución normal estándar (tiene una media de cero y una desviación estándar de uno).
NORM.S.INV	Devuelve el inverso de la distribución acumulativa normal estándar.
PERCENTILE.EXC	Devuelve el percentil k-ésimo de los valores de un intervalo, donde k está en el intervalo 0..1, exclusive.
PERCENTILE.INC	Devuelve el percentil k-ésimo de los valores de un intervalo, donde k está en el intervalo 0..1, inclusive.
PERCENTILEX.EXC	Devuelve el número de percentil de una expresión evaluada para cada fila de una tabla.
PERCENTILEX.INC	Devuelve el número de percentil de una expresión evaluada para cada fila de una tabla.
PERMUT	Devuelve el número de permutaciones de un número determinado de objetos que se pueden seleccionar de objetos number.
POISSON.DIST	Devuelve la distribución de Poisson.
RANK.EQ	Devuelve la clasificación de un número en una lista de números.
RANKX	Devuelve la clasificación de un número en una lista de números para cada fila del argumento <i>table</i> .
SAMPLE	Devuelve una muestra de N filas de la tabla especificada.
STDEV.P	Devuelve la desviación estándar de toda la población.
STDEV.S	Devuelve la desviación estándar de una población de muestra.
STDEVX.P	Devuelve la desviación estándar de toda la población.

Función	Descripción
STDEVX.S	Devuelve la desviación estándar de una población de muestra.
T.DIST	Devuelve la distribución t de cola izquierda de Student.
T.DIST.2T	Devuelve la distribución t bilateral de Student.
T.DIST.RT	Devuelve la distribución t de Student de cola derecha.
T.INV	Devuelve el inverso de cola izquierda de la distribución t de Student.
T.INV.2T	Devuelve el inverso bilateral de la distribución t de Student.
VAR.P	Devuelve la varianza del total de la población.
VAR.S	Devuelve la varianza de una población de muestra.
VARX.P	Devuelve la varianza del total de la población.
VARX.S	Devuelve la varianza de una población de muestra.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

BETA.DIST

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la distribución beta. La distribución beta se suele usar para estudiar la variación en el porcentaje de algo en varias muestras, como la fracción del día que las personas pasan viendo la televisión.

Sintaxis

DAX

`BETA.DIST(x,alpha,beta,cumulative,[A],[B])`

Parámetros

Expandir tabla

Término	Definición
x	Valor entre A y B en el cual se evalúa la función.
Alpha	Parámetro de la distribución.
Beta	Parámetro de la distribución.
A	Opcional. Límite inferior para el intervalo de x.
B	Opcional. Límite superior para el intervalo de x.

Valor devuelto

Devuelve la distribución beta.

Observaciones

- Si algún argumento no es numérico, BETA.DIST devuelve el #NUM!
- Si algún argumento no es un entero, se redondea.

- Si alpha ≤ 0 o beta ≤ 0, BETA.DIST devuelve el valor de error #NUM!
 - Si x < A, x > B o A = B, BETA.DIST devuelve el valor de error #NUM!
 - Si omite los valores de A y B, BETA.DIST usa la distribución beta acumulativa estándar, de modo que A = 0 y B = 1.
 - Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).
-

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

BETA.INV

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el inverso de la función de densidad de probabilidad acumulativa beta (BETA.DIST).

Si probability = BETA.DIST(x,...TRUE), BETA.INV(probability,...) = x. La distribución beta se puede usar en la planeación de proyectos para modelar las horas de finalización probables según una hora de finalización y una variabilidad esperadas.

Sintaxis

DAX
<code>BETA.INV(probability,alpha,beta,[A],[B])</code>

Parámetros

Expandir tabla

Término	Definición
Probabilidad	Probabilidad asociada a la distribución beta.
Alpha	Parámetro de la distribución.
Beta	Parámetro de la distribución.
A	Opcional. Límite inferior para el intervalo de x.
B	Opcional. Límite superior para el intervalo de x.

Valor devuelto

Devuelve el inverso de la función de densidad de probabilidad acumulativa beta (BETA.DIST).

Observaciones

- Si algún argumento no es numérico, BETA.INV devuelve el valor de error #VALUE! #NUM!
 - Si algún argumento no es un entero, se redondea.
 - Si alpha ≤ 0 o beta ≤ 0, BETA.INV devuelve el valor de error #NUM! #NUM!
 - Si probability ≤ 0 o probability > 1, BETA.INV devuelve el valor de error #NUM! #NUM!
 - Si omite los valores de A y B, BETA.INV usa la distribución beta acumulativa estándar, de modo que A = 0 y B = 1.
 - Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).
-

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CHISQ.DIST

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la distribución chi cuadrado.

La distribución chi cuadrado se suele usar para estudiar la variación en el porcentaje de algo en diversas muestras, como la fracción del día que las personas pasan viendo la televisión.

Sintaxis

DAX

```
CHISQ.DIST(<x>, <deg_freedom>, <cumulative>)
```

Parámetros

Expandir tabla

Término	Definición
x	El valor en el que desea evaluar la distribución.
Deg_freedom	Número de grados de libertad.
cumulative	Valor lógico que determina el formato de la función. Si cumulative es TRUE, CHISQ.DIST devuelve la función de distribución acumulativa; si es FALSE, devuelve la función de densidad de probabilidad.

Valor devuelto

Distribución chi cuadrado.

Notas

- Si x o deg_freedom no son numéricos, se devuelve un error.
- Si deg_freedom no es un entero, se redondea.

- Si $x < 0$, se devuelve un error.
- Si $\text{deg_freedom} < 1$ o $\text{deg_freedom} > 10^{10}$, se devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Consulte la siguiente consulta DAX:

```
DAX  
EVALUATE { CHISQ.DIST(2, 2, TRUE) }
```

Devuelve

[Expandir tabla](#)

[Valor]
0.632120558828558

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CHISQ.DIST.RT

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la probabilidad de cola derecha de la distribución chi cuadrado.

La distribución chi cuadrado se asocia a una prueba chi cuadrado. Use la prueba chi cuadrado para comparar los valores observados y esperados. Por ejemplo, un experimento genético podría plantear la hipótesis de que la próxima generación de plantas exhibirá un cierto conjunto de colores. Mediante la comparación de los resultados observados con los esperados, puede decidir si la hipótesis original es válida.

Sintaxis

DAX

```
CHISQ.DIST.RT(<x>, <deg_freedom>)
```

Parámetros

Expandir tabla

Término	Definición
x	El valor en el que desea evaluar la distribución.
Deg_freedom	Número de grados de libertad.

Valor devuelto

La probabilidad de cola derecha de la distribución chi cuadrado.

Notas

- Si x o deg_freedom no son numéricos, se devuelve un error.
- Si deg_freedom no es un entero, se redondea.
- Si x < 0, se devuelve un error.

- Si `deg_freedom` < 1 o `deg_freedom` > 10^{10} , se devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Consulte la siguiente consulta DAX:

```
DAX  
EVALUATE { CHISQ.DIST.RT(2, 5) }
```

Devuelve

 Expandir tabla

[Valor]
0.84914503608461

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CHISQ.INV

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el inverso de la probabilidad de cola izquierda de la distribución chi cuadrado.

La distribución chi cuadrado se suele usar para estudiar la variación en el porcentaje de algo en diversas muestras, como la fracción del día que las personas pasan viendo la televisión.

Sintaxis

DAX

`CHISQ.INV(probability,deg_freedom)`

Parámetros

Expandir tabla

Término	Definición
Probabilidad	Probabilidad asociada a la distribución chi cuadrado.
Deg_freedom	Número de grados de libertad.

Valor devuelto

Devuelve el inverso de la probabilidad de cola izquierda de la distribución chi cuadrado.

Notas

- Si el argumento no es numérico, CHISQ.INV devuelve el valor de error "#VALUE!" .
- Si probability < 0 o probability > 1, CHISQ.INV devuelve el valor de error #NUM! .
- Si deg_freedom no es un entero, se redondea.
- Si deg_freedom es < 0 o > 10^10, CHISQ.INV devuelve el valor de error #NUM!

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Fórmula	Description	Resultado
= CHISQ.INV(0,93,1)	Inverso de la probabilidad de cola izquierda de la distribución chi cuadrado de 0,93, con 1 grado de libertad.	5,318520074
= CHISQ.INV(0,6,2)	Inverso de la probabilidad de cola izquierda de la distribución chi cuadrado de 0,6, con 2 grados de libertad.	1,832581464

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CHISQ.INV.RT

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el inverso de la probabilidad de cola derecha de la distribución chi cuadrado.

Si `probability` = CHISQ.DIST.RT(`x`,...), entonces `CHISQ.INV.RT(probability,...)` = `x`. Use esta función para comparar los resultados observados con los esperados a fin de decidir si la hipótesis original es válida.

Sintaxis

DAX

`CHISQ.INV.RT(probability,deg_freedom)`

Parámetros

Expandir tabla

Término	Definición
Probabilidad	Probabilidad asociada a la distribución chi cuadrado.
Deg_freedom	Número de grados de libertad.

Valor devuelto

Devuelve el inverso de la probabilidad de cola derecha de la distribución chi cuadrado.

Observaciones

- Si alguno de los argumentos no es numérico, CHISQ.INV.RT devuelve el #NUM!
- Si `probability` < 0 o `probability` > 1, CHISQ.INV.RT devuelve el .
- Si `deg_freedom` no es un entero, se redondea.
- Si `deg_freedom` < 1, CHISQ.INV.RT devuelve el #NUM!

- Dado un valor de probabilidad, CHISQ.INV.RT busca ese valor x, de modo que CHISQ.DIST.RT(x, deg_freedom) = probability. Por tanto, la precisión de CHISQ.INV.RT depende de la precisión de CHISQ.DIST.RT. CHISQ.INV.RT usa una técnica de búsqueda iterativa. Si la búsqueda no converge después de 64 iteraciones, la función devuelve el valor de error #N/A.
 - Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).
-

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COMBIN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de combinaciones de una cantidad determinada de elementos. Use COMBIN para determinar el número total posible de grupos para una cantidad determinada de elementos.

Sintaxis

DAX

`COMBIN(number, number_chosen)`

Parámetros

Expandir tabla

Término	Definición
número	Número de elementos.
number_chosen	Número de elementos de cada combinación.

Valor devuelto

Devuelve el número de combinaciones de una cantidad determinada de elementos.

Notas

- Los argumentos numéricos se truncan en enteros.
- Si alguno de los argumentos no es numérico, COMBIN devuelve el valor de error "#VALUE!" .
- Si $number < 0$, $number_chosen < 0$ o $number < number_chosen$, COMBIN devuelve el valor de error "#NUM!" .

- Una combinación es cualquier conjunto o subconjunto de elementos, independientemente de su orden interno. Las combinaciones son distintas de las permutaciones, para las que el orden interno es significativo.
- El número de combinaciones es el siguiente, donde `number = n` y `number_chosen = k`:

$$\binom{n}{k} = \frac{P_{k,n}}{k!} = \frac{n!}{k!(n-k)!}$$

Where

$$P_{k,n} = \frac{n!}{(n-k)!}$$

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Fórmula	Description	Resultado
= COMBIN(8,2)	Posibles equipos de dos personas que se pueden formar con 8 candidatos.	28

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

COMBINA

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la cantidad de combinaciones (con repeticiones) de una cantidad determinada de elementos.

Sintaxis

DAX

`COMBINA(number, number_chosen)`

Parámetros

Expandir tabla

Término	Definición
número	Debe ser mayor o igual que 0 y mayor o igual que Number_chosen. Los valores no enteros se truncan.
number_chosen	Debe ser mayor o igual que 0. Los valores no enteros se truncan.

Valor devuelto

Devuelve la cantidad de combinaciones (con repeticiones) de una cantidad determinada de elementos.

Notas

- Si el valor de cualquier argumento está fuera de sus restricciones, COMBINA devuelve el valor de error .
- Si alguno de los argumentos es un valor no numérico, COMBINA devuelve el valor de error .
- Se usa la siguiente ecuación, donde N es Number y M es Number_chosen:

$$\binom{N+M-1}{N-1}$$

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Fórmula	Description	Resultado
= COMBINA(4,3)	Devuelve el número de combinaciones (con repeticiones) para 4 y 3.	20
= COMBINA(10,3)	Devuelve el número de combinaciones (con repeticiones) para 10 y 3.	220

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CONFIDENCE.NORM

Artículo • 20/10/2023

El intervalo de confianza es un rango de valores. La media de muestra, x , está en el centro de este rango, que es $x \pm \text{CONFIDENCE.NORM}$. Por ejemplo, si x es la media de muestra de tiempos de entrega de productos pedidos por correo, $x \pm \text{CONFIDENCE.NORM}$ es un rango de medias de población. En cualquier media de población, μ_0 , de este rango, la probabilidad de obtener una media de muestra más allá de μ_0 que x es mayor que alpha; en cualquier media de población, μ_0 , que no es de este rango, la probabilidad de obtener una media de muestra más allá de μ_0 que x es menor que alfa. Es decir, imagine que se usan x , `standard_dev` y `size` para construir una prueba de dos colas en el nivel de significación alpha de la hipótesis de que la media de población es μ_0 . Luego no se rechaza esa hipótesis si μ_0 está en el intervalo de confianza y se rechaza esa hipótesis si μ_0 no está en el intervalo de confianza. El intervalo de confianza no permite deducir que hay una probabilidad $1 - \text{alpha}$ de que el siguiente paquete tarde un tiempo de entrega que se encuentre en el intervalo de confianza.

Sintaxis

DAX

```
CONFIDENCE.NORM(alpha,standard_dev,size)
```

Parámetros

Término	Definición
alpha	Nivel de significación usado para calcular el nivel de confianza. El nivel de confianza es igual a $100*(1 - \text{alpha})\%$, es decir, un valor alfa de 0,05 indica un nivel de confianza del 95 por ciento.
standard_dev	Desviación estándar de población del rango de datos, que se asume que es conocida.
standard_dev,size	Tamaño de la muestra.

Valor devuelto

Rango de valores.

Notas

- Si algún argumento no es numérico, CONFIDENCE.NORM devuelve el valor de error #VALUE! .
- Si alpha ≤ 0 o alpha ≥ 1, CONFIDENCE.NORM devuelve el valor de error #NUM! .
- Si standard_dev ≤ 0, CONFIDENCE.NORM devuelve el valor de error #NUM! .
- Si size no es un entero, se redondea.
- Si size < 1, CONFIDENCE.NORM devuelve el valor de error #NUM! .
- Si imagina que alpha es igual a 0,05, es necesario calcular el área bajo la curva normal estándar que sea igual a (1 - alpha) o 95 por ciento. Este valor es ± 1,96. Por lo tanto, el intervalo de confianza es:

$$\bar{x} \pm 1.96 \left(\frac{\sigma}{\sqrt{n}} \right)$$

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

CONFIDENCE.T

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el intervalo de confianza de una media de población con una distribución t de Student.

Sintaxis

DAX

`CONFIDENCE.T(alpha,standard_dev,size)`

Parámetros

Expandir tabla

Término	Definición
alpha	Nivel de significación usado para calcular el nivel de confianza. El nivel de confianza es igual a $100*(1 - \text{alpha})\%$, es decir, un valor alfa de 0,05 indica un nivel de confianza del 95 por ciento.
standard_dev	Desviación estándar de población del rango de datos, que se asume que es conocida.
size	Tamaño de la muestra.

Valor devuelto

Devuelve el intervalo de confianza de una media de población con una distribución t de Student.

Notas

- Si algún argumento no es numérico, CONFIDENCE.T devuelve el valor de error #VALUE! .
- Si alpha ≤ 0 o alpha ≥ 1, CONFIDENCE.T devuelve el valor de error #NUM! .

- Si standard_dev ≤ 0, CONFIDENCE.T devuelve el valor de error #NUM! .
- Si size no es un entero, se redondea.
- Si size es igual a 1, CONFIDENCE.T devuelve el valor de error #DIV/0! .
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

 Expandir tabla

Fórmula	Description	Resultado
= CONFIDENCE.T(0,05,1,50)	Intervalo de confianza de la media de una población basado en un tamaño de muestra de 50, con un nivel de significación del 5 % y una desviación estándar de 1. Se basa en la distribución t de Student.	0,284196855

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

EXPON.DIST

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la distribución exponencial. Use EXPON.DIST para modelar el tiempo entre eventos, por ejemplo, el tiempo que tarda un cajero automático en entregar el dinero. Por ejemplo, puede usar EXPON.DIST para determinar la probabilidad de que el proceso tarde como máximo 1 minuto.

Sintaxis

DAX

`EXPON.DIST(x,lambda,cumulative)`

Parámetros

Expandir tabla

Término	Definición
x	Obligatorio. Valor de la función.
lambda	Obligatorio. Valor del parámetro.
cumulative	Obligatorio. Valor lógico que indica qué forma de la función exponencial se va a proporcionar. Si cumulative es TRUE, EXPON.DIST devuelve la función de distribución acumulativa; si es FALSE, devuelve la función de densidad de probabilidad.

Valor devuelto

Devuelve la distribución exponencial.

Notas

- Si x o lambda no son numéricos, EXPON.DIST devuelve el valor de error #VALUE! .
- Si x o lambda no son enteros, se redondean.

- If $x < 0$, EXPON.DIST devuelve el valor de error #NUM! .
- If $\lambda \leq 0$, EXPON.DIST devuelve el valor de error #NUM! .
- La ecuación de la función de densidad de probabilidad es:

$$f(x; \lambda) = \lambda e^{-\lambda x}$$

- La ecuación de la función de distribución acumulativa es:

$$F(x; \lambda) = 1 - e^{-\lambda x}$$

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).
-

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

GEOMEAN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve la media geométrica de los números de una columna.

Para devolver la media geométrica de una expresión evaluada para cada fila de una tabla, use la función [GEOMEANX](#).

Sintaxis

DAX

`GEOMEAN(<column>)`

Parámetros

Expandir tabla

Término	Definición
columna	Columna que contiene los números para los que se va a calcular la media geométrica.

Valor devuelto

Número decimal.

Notas

- Solo se cuentan los números de la columna. Se omiten los espacios en blanco, los valores lógicos y el texto.
- GEOMEAN(Table[Column]) es equivalente a GEOMEANX(Table, Table[Column])
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Lo siguiente calcula la media geométrica de la columna devuelta en la tabla Inversiones:

DAX

```
= GEOMEAN( Investment[Return] )
```

Contenido relacionado

[Función GEOMEANX](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

GEOMEANX

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la media geométrica de una expresión evaluada de cada fila de una tabla.

Para devolver la media geométrica de los números de una columna, use la función [GEOMEAN](#).

Sintaxis

DAX

```
GEOMEANX(<table>, <expression>)
```

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla que contiene las filas para las que se evaluará la expresión.
expresión	Expresión que se debe evaluar para cada fila de la tabla.

Valor devuelto

Número decimal.

Notas

- La función GEOMEANX toma como primer argumento una tabla o una expresión que devuelve una tabla. El segundo argumento es una columna que contiene los números cuya media geométrica se quiere calcular, o una expresión que se evalúa como una columna.
- Solo se cuentan los números de la columna. Se omiten los espacios en blanco, los valores lógicos y el texto.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Lo siguiente calcula la media geométrica de la columna ReturnPct devuelta en la tabla Investments:

DAX

```
= GEOMEANX( Investments, Investments[ReturnPct] + 1 )
```

Contenido relacionado

[Función GEOMEAN](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

LINEST

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Uso el método de mínimos cuadrados para calcular una línea recta con el mejor ajuste a los datos especificados y, a continuación, devuelve una tabla que describe la línea. La ecuación de la línea adopta la forma: $y = \text{Slope1} * x_1 + \text{Slope2} * x_2 + \dots + \text{Intercept}$.

Sintaxis

DAX

```
LINEST ( <columnY>, <columnX>[ , ...][, <const>] )
```

Parámetros

[] Expandir tabla

Término	Definición
columnY	Columna de valores y conocidos. Debe tener un tipo escalar.
columnX	Columnas de valores x conocidos. Debe tener un tipo escalar. Se debe proporcionar al menos una.
const	(Opcional) Valor TRUE/FALSE constante que especifica si se va a forzar que la constante Intercept sea igual a 0. Si es TRUE o se omite, el valor Intercept se calcula de forma normal; Si es FALSE, el valor Intercept se establece en cero.

Valor devuelto

Tabla de una sola fila que describe la línea, además de estadísticas adicionales. Estas son las columnas disponibles:

- **Slope1, Slope2, ..., SlopeN:** los coeficientes correspondientes a cada valor x;
- **Intercept:** valor de intersección;
- **StandardErrorSlope1, StandardErrorSlope2, ..., StandardErrorSlopeN:** los valores de error estándar para los coeficientes Slope1, Slope2, ..., SlopeN;
- **StandardErrorIntercept:** el valor de error estándar para la constante Intercept;
- **CoefficientOfDetermination:** el coeficiente de determinación (r^2). Compara los valores y estimados y reales, y los intervalos en un valor de 0 a 1: cuanto mayor sea el valor, mayor será la correlación en la muestra;
- **StandardError:** el error estándar de la estimación y;
- **FStatistic:** la estadística F o el valor observado F. Utilice la estadística F para determinar si la relación observada entre las variables dependientes e independientes ocurre por casualidad;
- **DegreesOfFreedom:** los grados de libertad. Use este valor para ayudarle a encontrar valores críticos de F en una tabla estadística y determinar un nivel de confianza para el modelo;

- **RegressionSumOfSquares**: la suma de regresión de los cuadrados;
- **ResidualSumOfSquares**: la suma residual de los cuadrados.

Comentarios

<columnY> y columnX <>deben pertenecer a la misma tabla.

Ejemplo 1

La consulta DAX siguiente:

```
DAX

EVALUATE LINEST(
    'FactInternetSales'[SalesAmount],
    'FactInternetSales'[TotalProductCost]
)
```

Devuelve una tabla de una sola fila con diez columnas:

[Expandir tabla](#)

Slope1	Interceptar	StandardErrorSlope1	StandardErrorIntercept	CoefficientOfDetermination
1,67703250456677	6,34550460373026	0,000448675725548806	0,279131821917317	0,995695557281456

[Expandir tabla](#)

StandardError	FStatistic	DegreesOfFreedom	RegressionSumOfSquares	ResidualSumOfSquares
60,9171030357485	13970688,6139993	60396	51843736761,658	224123120,339218

- **Slope1 e Intercept**: los coeficientes del modelo lineal calculado;
- **StandardErrorSlope1 y StandardErrorIntercept**: los valores de error estándar para los coeficientes anteriores;
- **CoefficientOfDetermination, StandardError, FStatistic, DegreesOfFreedom, RegressionSumOfSquares y ResidualSumOfSquares**: estadísticas de regresión sobre el modelo.

Para una venta por Internet determinada, este modelo predice el importe de la venta mediante la fórmula siguiente:

```
SalesAmount = Slope1 * TotalProductCost + Intercept
```

Ejemplo 2

La consulta DAX siguiente:

```
DAX
```

```
EVALUATE LINEST(  
    'DimCustomer'[TotalSalesAmount],  
    'DimCustomer'[YearlyIncome],  
    'DimCustomer'[TotalChildren],  
    'DimCustomer'[BirthDate]  
)
```

Devuelve una tabla de una sola fila con catorce columnas:

- Slope1
- Slope2
- Slope3
- Interceptar
- StandardErrorSlope1
- StandardErrorSlope2
- StandardErrorSlope3
- StandardErrorIntercept
- CoefficientOfDetermination
- StandardError
- FStatistic
- DegreesOfFreedom
- RegressionSumOfSquares
- ResidualSumOfSquares

Para un cliente determinado, este modelo predice las ventas totales mediante la fórmula siguiente (la fecha de nacimiento se convierte automáticamente en un número):

```
TotalSalesAmount = Slope1 * YearlyIncome + Slope2 * TotalChildren + Slope3 * BirthDate +  
Intercept
```

Contenido relacionado

[LINESTX](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

LINESTX

Artículo • 30/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Usa el método de mínimos cuadrados para calcular una línea recta con el mejor ajuste a los datos especificados y, a continuación, devuelve una tabla que describe la línea. Los datos se obtienen de las expresiones evaluadas para cada fila de una tabla. La ecuación de la línea adopta la forma: $y = \text{Slope1} * x_1 + \text{Slope2} * x_2 + \dots + \text{Intercept}$.

Sintaxis

DAX

```
LINESTX ( <table>, <expressionY>, <expressionX>[, ...][, <const>] )
```

Parámetros

[] Expandir tabla

Término	Definición
tabla	Tabla que contiene las filas para las que se evaluarán las expresiones.
expressionY	Expresión que se evaluará para cada fila de la tabla para obtener los valores y conocidos. Debe tener un tipo escalar.
expressionX	Expresiones que se evaluarán para cada fila de la tabla para obtener los valores x conocidos. Debe tener un tipo escalar. Se debe proporcionar al menos una.
const	(Opcional) Valor TRUE/FALSE constante que especifica si se va a forzar que la constante Intercept sea igual a 0. Si es TRUE o se omite, el valor Intercept se calcula de forma normal; Si es FALSE, el valor Intercept se establece en cero.

Valor devuelto

Tabla de una sola fila que describe la línea, además de estadísticas adicionales. Estas son las columnas disponibles:

- **Slope1, Slope2, ..., SlopeN**: los coeficientes correspondientes a cada valor x;
- **Intercept**: valor de intersección;
- **StandardErrorSlope1, StandardErrorSlope2, ..., StandardErrorSlopeN**: los valores de error estándar para los coeficientes Slope1, Slope2, ..., SlopeN;
- **StandardErrorIntercept**: el valor de error estándar para la constante Intercept;

- **CoefficientOfDetermination**: el coeficiente de determinación (r^2). Compara los valores y estimados y reales, y los intervalos en un valor de 0 a 1: cuanto mayor sea el valor, mayor será la correlación en la muestra;
- **StandardError**: el error estándar de la estimación y;
- **FStatistic**: la estadística F o el valor observado F. Utilice la estadística F para determinar si la relación observada entre las variables dependientes e independientes ocurre por casualidad;
- **DegreesOfFreedom**: los grados de libertad. Use este valor para ayudarle a encontrar valores críticos de F en una tabla estadística y determinar un nivel de confianza para el modelo;
- **RegressionSumOfSquares**: la suma de regresión de los cuadrados;
- **ResidualSumOfSquares**: la suma residual de los cuadrados.

Ejemplo 1

La consulta DAX siguiente:

```
DAX

DEFINE VAR TotalSalesByRegion = SUMMARIZECOLUMNS(
    'Sales Territory'[Sales Territory Key],
    'Sales Territory'[Population],
    "Total Sales", SUM(Sales[Sales Amount]))
)
EVALUATE LINESTX(
    'TotalSalesByRegion',
    [Total Sales],
    [Population]
)
```

Devuelve una tabla de una sola fila con diez columnas:

[] Expandir tabla

Slope1	Interceptar	StandardErrorSlope1	StandardErrorIntercept	CoefficientOfDetermination
6,42271517588	-410592,76216	0,24959467764561	307826,343996223	0,973535860750193

[] Expandir tabla

StandardError	FStatistic	DegreesOfFreedom	RegressionSumOfSquares	ResidualSumOfSquares
630758,1747292	662,165707642	18	263446517001130	7161405749781,07

- **Slope1 e Intercept**: los coeficientes del modelo lineal calculado;
- **StandardErrorSlope1 y StandardErrorIntercept**: los valores de error estándar para los coeficientes anteriores;
- **CoefficientOfDetermination, StandardError, FStatistic, DegreesOfFreedom, RegressionSumOfSquares y ResidualSumOfSquares**: estadísticas de regresión sobre el modelo.

Para un determinado territorio de ventas, este modelo predice las ventas totales mediante la fórmula siguiente:

```
Total Sales = Slope1 * Population + Intercept
```

Ejemplo 2

La consulta DAX siguiente:

```
DAX

DEFINE VAR TotalSalesByCustomer = SUMMARIZECOLUMNS(
    'Customer'[Customer ID],
    'Customer'[Age],
    'Customer'[NumOfChildren],
    "Total Sales", SUM(Sales[Sales Amount])
)
EVALUATE LINESTX(
    'TotalSalesByCustomer',
    [Total Sales],
    [Age],
    [NumOfChildren]
)
```

Devuelve una tabla de una sola fila con doce columnas:

[\[+\] Expandir tabla](#)

Slope1	Slope2	Interceptar	StandardErrorSlope1
69,0435458093763	33,005949841721	-871,118539339539	0,872588875481658

[\[+\] Expandir tabla](#)

StandardErrorSlope2	StandardErrorIntercept	CoefficientOfDetermination	StandardError
6,21158863903435	26,726292527427	0,984892920482022	68,5715034014342

[\[+\] Expandir tabla](#)

FStatistic	DegreesOfFreedom	RegressionSumOfSquares	ResidualSumOfSquares
3161,91535144391	97	29734974,9782379	456098,954637092

Para un determinado cliente, este modelo predice las ventas totales mediante la fórmula siguiente:

Total Sales = Slope1 * Age + Slope2 * NumOfChildren + Intercept

Contenido relacionado

[LINEST](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MEDIAN

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la mediana de números de una columna.

Para devolver la mediana de una expresión evaluada para cada fila de una tabla, use la función [MEDIANX](#).

Sintaxis

DAX

`MEDIAN(<column>)`

Parámetros

Expandir tabla

Término	Definición
columna	Columna que contiene los números cuya mediana se va a calcular.

Valor devuelto

Número decimal.

Notas

- Solo se cuentan los números de la columna. Los espacios en blanco se ignoran. No se admiten valores lógicos, fechas y texto.
- MEDIAN(Table[Column]) es equivalente a MEDIANX(Table, Table[Column]).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el siguiente ejemplo se calcula la mediana de una columna denominada Age de una tabla denominada Customers:

DAX

```
= MEDIAN( Customers[Age] )
```

Contenido relacionado

[Función MEDIANX](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MEDIANX

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la mediana de una expresión evaluada para cada fila de una tabla.

Para devolver la mediana de números de una columna, use [Función MEDIAN](#).

Sintaxis

DAX

`MEDIANX(<table>, <expression>)`

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla que contiene las filas para las que se evaluará la expresión.
expresión	Expresión que se debe evaluar para cada fila de la tabla.

Valor devuelto

Número decimal.

Notas

- La función MEDIANX toma como primer argumento una tabla o una expresión que devuelve una tabla. El segundo argumento es una columna que contiene los números cuya mediana se quiere calcular, o una expresión que se evalúa como una columna.
- Solo se cuentan los números de la columna.
- Se omiten los valores lógicos y el texto.
- MEDIANX no omite los espacios en blanco, aunque MEDIAN sí.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se calcula la edad media de los clientes que viven en los Estados Unidos.

DAX

```
= MEDIANX( FILTER(Customers, RELATED( Geography[Country]="USA" ) ),  
Customers[Age] )
```

Contenido relacionado

[Función MEDIAN](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NORM.DIST

Article • 05/20/2024

Applies to: Calculated column Calculated table Measure Visual calculation

Returns the normal distribution for the specified mean and standard deviation.

Syntax

DAX

`NORM.DIST(X, Mean, Standard_dev, Cumulative)`

Parameters

Expand table

Term	Definition
X	The value for which you want the distribution.
Mean	The arithmetic mean of the distribution.
Standard_dev	The standard deviation of the distribution.
Cumulative*	A logical value that determines the form of the function. If cumulative is TRUE, NORM.DIST returns the cumulative distribution function; if FALSE, it returns the probability density function.

Return value

The normal distribution for the specified mean and standard deviation.

Remarks

This function is not supported for use in DirectQuery mode when used in calculated columns or row-level security (RLS) rules.

Example

DAX

```
EVALUATE { NORM.DIST(42, 40, 1.5, TRUE) }
```

Returns

 Expand table

[Value]
0.908788780274132

Related content

[NORM.S.DIST function](#)

[NORM.INV function](#)

[NORM.S.INV](#)

Feedback

Was this page helpful?

 Yes

 No

NORM.INV

Artículo • 21/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Inverso de la distribución acumulativa normal para la media y la desviación estándar especificadas.

Sintaxis

DAX

```
NORM.INV(Probability, Mean, Standard_dev)
```

Parámetros

Expandir tabla

Término	Definición
Probabilidad	Probabilidad correspondiente a la distribución normal.
Media	Media aritmética de la distribución.
Standard_dev	Desviación estándar de la distribución.

Valor devuelto

Devuelve el inverso de la distribución acumulativa normal para la media y la desviación estándar especificadas.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

DAX

```
EVALUATE { NORM.INV(0.908789, 40, 1.5) }
```

Devuelve

 Expandir tabla

[Valor]
42,00000200956628780274132

Contenido relacionado

[NORM.S.INV](#)

[Función NORM.S.DIST](#)

[Función NORM.DIST](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NORM.S.DIST

Artículo • 21/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la distribución normal estándar (tiene una media de cero y una desviación estándar de uno).

Sintaxis

DAX

`NORM.S.DIST(Z, Cumulative)`

Parámetros

Expandir tabla

Término	Definición
Z	Valor cuya distribución se quiere obtener.
Cumulative	Cumulative es un valor lógico que determina el formato de la función. Si cumulative es TRUE, NORM.S.DIST devuelve la función de distribución acumulativa; si es FALSE, devuelve la función de densidad de probabilidad.

Valor devuelto

Distribución normal estándar (tiene una media de cero y una desviación estándar de uno).

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

DAX

```
EVALUATE { NORM.S.DIST(1.333333, TRUE) }
```

Devuelve

 Expandir tabla

[Valor]
0,908788725604095

Contenido relacionado

[NORM.INV \(función\)](#)

[Función NORM.DIST](#)

[NORM.S.INV](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NORM.S.INV

Artículo • 21/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la inversa de la distribución acumulativa normal estándar. La distribución tiene una media de cero y una desviación estándar de uno.

Sintaxis

DAX

`NORM.S.INV(Probability)`

Parámetros

Expandir tabla

Término	Definición
Probabilidad	Probabilidad correspondiente a la distribución normal.

Valor devuelto

Inverso de la distribución acumulativa normal estándar. La distribución tiene una media de cero y una desviación estándar de uno.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

DAX

`EVALUATE { NORM.S.INV(0.908789) }`

Devuelve

 Expandir tabla

[Valor]
1,33333467304411

Contenido relacionado

[NORM.INV](#)

[Función NORM.S.DIST](#)

[Función NORM.DIST](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PERCENTILE.EXC

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el percentil k-ésimo de los valores de un intervalo, donde k está en el intervalo 0..1, exclusive.

Para devolver el número de percentil de una expresión evaluada para cada fila de una tabla, use la función [PERCENTILE.EXC](#).

Sintaxis

DAX

`PERCENTILE.EXC(<column>, <k>)`

Parámetros

Expandir tabla

Término	Definición
columna	Una columna que contiene los valores que definen la posición relativa.
k	El valor de percentil en el intervalo 0..1, exclusive.

Valor devuelto

El percentil k-ésimo de los valores de un intervalo, donde k está en el intervalo 0..1, exclusive.

Notas

- Si la columna está vacía, se devuelve BLANK().
- Si k es cero o está en blanco, el intervalo percentil de $1/(n+1)$ devuelve el valor más pequeño. Si es cero, está fuera del intervalo y se devuelve un error.

- Si k es un valor no numérico o está fuera del intervalo comprendido entre 0 y 1, se devuelve un error.
- Si k no es un múltiplo de $1/(n + 1)$, PERCENTILE.EXC se interpolará para determinar el valor en el percentil k-ésimo.
- PERCENTILE.EXC se interpolará cuando el valor del percentil especificado esté entre dos valores de la matriz. Si no se puede interpolar para el percentil k especificado, se devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Contenido relacionado

[PERCENTILE.EXC](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PERCENTILE.INC

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el percentil k-ésimo de los valores de un intervalo, donde k está en el intervalo 0..1, inclusive.

Para devolver el número de percentil de una expresión evaluada para cada fila de una tabla, use la función [PERCENTILEX.INC](#).

Sintaxis

DAX

`PERCENTILE.INC(<column>, <k>)`

Parámetros

Expandir tabla

Término	Definición
columna	Una columna que contiene los valores que definen la posición relativa.
k	El valor de percentil en el intervalo 0..1, inclusive.

Valor devuelto

El percentil k-ésimo de los valores de un intervalo, donde k está en el intervalo 0..1, inclusive.

Notas

- Si la columna está vacía, se devuelve BLANK().
- Si k es cero o está en blanco, el intervalo percentil de $1/(n+1)$ devuelve el valor más pequeño. Si es cero, está fuera del intervalo y se devuelve un error.

- Si k es un valor no numérico o está fuera del intervalo comprendido entre 0 y 1, se devuelve un error.
- Si k no es un múltiplo de $1/(n + 1)$, PERCENTILE.INC se interpolará para determinar el valor en el percentil k-ésimo.
- PERCENTILE.INC se interpolará cuando el valor del percentil especificado esté entre dos valores de la matriz. Si no se puede interpolar para el percentil k especificado, se devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Contenido relacionado

[PERCENTILEX.INC](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PERCENTILEX.EXC

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de percentil de una expresión evaluada para cada fila de una tabla.

Para devolver el percentil de números de una columna, use la [función PERCENTILE.EXC](#).

Sintaxis

DAX

`PERCENTILEX.EXC(<table>, <expression>, k)`

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla que contiene las filas para las que se evaluará la expresión.
expresión	Expresión que se debe evaluar para cada fila de la tabla.
k	El valor de percentil deseado en el intervalo de 0 a 1 exclusivo.

Valor devuelto

Número de percentil de una expresión evaluada para cada fila de una tabla.

Notas

- Si k es cero o está en blanco, el intervalo percentil de $1/(n+1)$ devuelve el valor más pequeño. Si es cero, está fuera del intervalo y se devuelve un error.
- Si k es un valor no numérico o está fuera del intervalo comprendido entre 0 y 1, se devuelve un error.
- Si k no es un múltiplo de $1/(n + 1)$, PERCENTILEX.EXC se interpolará para determinar el valor en el percentil k-ésimo.

- PERCENTILEX.EXC se interpolará cuando el valor del percentil especificado esté entre dos valores de la matriz. Si no se puede interpolar para el percentil k especificado, se devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Contenido relacionado

[PERCENTILE.EXC](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PERCENTILEX.INC

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de percentil de una expresión evaluada para cada fila de una tabla.

Para devolver el percentil de números de una columna, use [PERCENTILE.INC](#).

Sintaxis

DAX

`PERCENTILEX.INC(<table>, <expression>;, k)`

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla que contiene las filas para las que se evaluará la expresión.
expresión	Expresión que se debe evaluar para cada fila de la tabla.
k	Valor de percentil deseado en el rango de 0 a 1 incluido.

Valor devuelto

Número de percentil de una expresión evaluada para cada fila de una tabla.

Notas

- Si k es cero o está en blanco, la clasificación de percentil de $1/(n - 1)$ devuelve el valor más pequeño. Si es cero, está fuera del rango y se devuelve un error.
- Si k es un valor no numérico o está fuera del rango de 0 a 1, se devuelve un error.
- Si k no es un múltiplo de $1/(n - 1)$, PERCENTILEX.EXC se interpola para determinar el valor en el percentil k-ésimo.

- PERCENTILEX.INC se interpola cuando el valor del percentil especificado está entre dos valores de la matriz. Si no se puede interpolar para el percentil k especificado, se devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Contenido relacionado

[PERCENTILE.INC](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PERMUT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de permutaciones de un número determinado de objetos que se pueden seleccionar de objetos number. Una permutación es cualquier conjunto o subconjunto de objetos o eventos donde el orden interno es significativo. Las permutaciones son distintas de las combinaciones, para las que el orden interno no es significativo. Use esta función para los cálculos de probabilidad del estilo de la lotería.

Sintaxis

DAX

`PERMUT(number, number_chosen)`

Parámetros

Expandir tabla

Término	Definición
número	Obligatorio. Entero que describe el número de objetos.
number_chosen	Obligatorio. Entero que describe el número de objetos de cada permutación.

Valor devuelto

Devuelve el número de permutaciones de un número determinado de objetos que se pueden seleccionar de objetos number.

Notas

- Ambos argumentos se truncan en enteros.
- Si number o number_chosen no son numéricos, PERMUT devuelve el valor de error #VALUE! .
- Si number ≤ 0 o number_chosen < 0, PERMUT devuelve el valor de error #NUM! .

- Si number < number_chosen, PERMUT devuelve el valor de error #NUM! .
- La ecuación del número de permutaciones es:

$$P_{k,n} = \frac{n!}{(n - k)!}$$

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la fórmula siguiente, las permutaciones son posibles para un grupo de 3 objetos donde se eligen 2:

DAX

```
= PERMUT(3,2)
```

Result,

6

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

POISSON.DIST

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la distribución de Poisson. Una aplicación común de la distribución de Poisson es predecir el número de eventos en un tiempo específico, como el número de automóviles que llegan a una estación de peaje en 1 minuto.

Sintaxis

DAX

`POISSON.DIST(x,mean,cumulative)`

Parámetros

Expandir tabla

Término	Definición
x	Obligatorio. Número de eventos.
mean	Obligatorio. Valor numérico esperado.
cumulative	Obligatorio. Valor lógico que determina la forma de la distribución de probabilidad devuelta. Si el valor acumulativo es TRUE, POISSON.DIST devuelve la probabilidad de Poisson acumulativa de que el número de eventos aleatorios que se producen estará comprendido entre cero y x inclusive; si es FALSE, devuelve la función de probabilidad bruta de Poisson de que el número de eventos que se producen será exactamente x.

Valor devuelto

Devuelve la distribución de Poisson.

Notas

- Si x no es un entero, se redondea.
- Si x o mean no es numérico, POISSON.DIST devuelve el .

- Si $x < 0$, POISSON.DIST devuelve el .
- Si $mean < 0$, POISSON.DIST devuelve el .
- POISSON.DIST se calcula como se muestra aquí.
 - Para cumulative = FALSE:

$$\text{POISSON} = \frac{e^{-\lambda} \lambda^x}{x!}$$

- Para cumulative = TRUE:

$$\text{CUMPOISSON} = \sum_{k=0}^x \frac{e^{-\lambda} \lambda^k}{k!}$$

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).
-

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

RANK.EQ

Artículo • 20/10/2023

Devuelve la clasificación de un número en una lista de números.

Sintaxis

DAX

```
RANK.EQ(<value>, <columnName>[, <order>])
```

Parámetros

Término	Definición
valor	Cualquier expresión DAX que devuelva un único valor escalar cuya clasificación se quiere obtener. La expresión se va a evaluar exactamente una vez antes de que se evalúe la función y de que su valor se pase a la lista de argumentos.
columnName	Nombre de una columna existente en la que se va a determinar la clasificación. No puede ser una expresión ni una columna creada mediante estas funciones: ADDCOLUMNS, ROW o SUMMARIZE.
orden	(Opcional) Valor que especifica cómo clasificar <i>number</i> , de inferior a superior o de superior a inferior:

valores de orden

value	valor alternativo	Descripción
0 (cero)	false	Clasifica <i>columnName</i> en orden descendente. Si <i>value</i> es igual al número superior de <i>columnName</i> , RANK.EQ es 1.
1	true	Clasifica <i>columnName</i> en orden ascendente. Si <i>value</i> es igual al número inferior de <i>columnName</i> , RANK.EQ es 1.

Valor devuelto

Número que indica la clasificación de *value* entre los números de *columnName*.

Comentarios

- *columnName* no puede hacer referencia a ninguna columna creada mediante estas funciones: ADDCOLUMNS, ROW o SUMMARIZE.I
- Si *value* no está en *columnName* o el valor es un espacio en blanco, *RANK.EQ* devuelve un valor en blanco.
- Los valores duplicados de *value* reciben el mismo valor de clasificación; el siguiente valor de clasificación asignado es el valor de clasificación más el número de valores duplicados. Por ejemplo, si cinco (5) valores están asociados a una clasificación de 11, el valor siguiente recibe una clasificación de 16 (11 + 5).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

En el ejemplo siguiente se crea una columna calculada que clasifica los valores de SalesAmount_USD, de la tabla *InternetSales_USD*, con respecto a todos los números de la misma columna.

DAX

```
= RANK.EQ(InternetSales_USD[SalesAmount_USD],  
InternetSales_USD[SalesAmount_USD])
```

Ejemplo 2

En el ejemplo siguiente se clasifica un subconjunto de valores en un ejemplo determinado. Imagine que tiene una tabla de alumnos locales con su rendimiento en una prueba nacional determinada y, además, tiene el conjunto completo de puntuaciones de esa prueba nacional. La siguiente columna calculada proporciona la clasificación nacional de cada uno de los alumnos locales.

DAX

```
= RANK.EQ(Students[Test_Score], NationalScores[Test_Score])
```

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

RANKX

Artículo • 20/10/2023

Devuelve la clasificación de un número en una lista de números de cada fila del argumento *table*.

Sintaxis

DAX

```
RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])
```

Parámetros

table

Cualquier expresión DAX que devuelve una tabla de datos en la que se evalúa la expresión.

expression

Cualquier expresión DAX que devuelve un valor escalar único. La expresión se evalúa para cada fila de *table*, para generar todos los valores posibles para la clasificación. Vea la sección Notas para comprender el comportamiento de la función cuando *expression* se evalúa como BLANK.

value

(Opcional) Cualquier expresión DAX que devuelva un único valor escalar cuya clasificación se quiere obtener. Vea la sección Notas para conocer el comportamiento de la función cuando no se encuentra *value* en la expresión.

Cuando se omite el valor de *value*, se usa en su lugar el valor de la expresión de la fila actual.

order

(Opcional) Valor que especifica cómo clasificar *value*, de inferior a superior o de superior a inferior:

valor	valor alternativo	Descripción
0 (cero)	FALSO	Clasifica en orden descendente los valores de la expresión. Si el valor es igual al número más alto de la expresión, RANKX devuelve 1.

valor	valor alternativo	Descripción
		Este es el valor predeterminado si se omite el parámetro <i>order</i> .
1	VERDADERO	Clasifica en orden ascendente los valores de la expresión. Si el valor es igual al número más bajo de la expresión, RANKX devuelve 1.

ties

Enumeración que define cómo determinar la clasificación cuando hay asociaciones.

enumeración	Descripción
Omitir	<p>El siguiente valor de clasificación, después de una asociación, es el valor de rango de la asociación más el recuento de valores asociados. Por ejemplo, si cinco (5) valores están asociados a una clasificación de 11, el valor siguiente recibe una clasificación de 16 (11 + 5).</p> <p>Este es el valor predeterminado cuando se omite el parámetro <i>ties</i>.</p>
Dense	<p>El siguiente valor de clasificación, después de una asociación, es el siguiente valor de rango. Por ejemplo, si cinco (5) valores están asociados a una clasificación de 11, el valor siguiente recibe una clasificación de 12.</p>

Valor devuelto

El número de rango de *value* entre todos los valores posibles de *expression* evaluado para todas las filas de números de *table*.

Comentarios

- Si *expression* o *value* se evalúa como BLANK, se trata como 0 (cero) para todas las expresiones que dan como resultado un número, o como texto vacío para todas las expresiones de texto.
- Si *value* no se encuentra entre todos los valores posibles de *expression*, entonces RANKX agrega temporalmente *value* a los valores de *expression* y vuelve a evaluar RANKX para determinar el rango adecuado de *value*.
- Se pueden omitir los argumentos opcionales si se coloca una coma vacía (,) en la lista de argumentos, como aquí: RANKX(Inventory, [InventoryCost],,"Dense")
- Es posible que las comparaciones relacionadas con la igualdad (=, <>, >= y <=) entre valores con el tipo de datos *Número decimal* puedan devolver resultados

inesperados al usar la función RANKX. Se pueden producir resultados incorrectos porque los valores con el tipo de datos *Número decimal* se almacenan como números de punto flotante del estándar IEEE 754 y tienen limitaciones inherentes en su precisión. Para evitar resultados inesperados, cambie el tipo de datos a *Número decimal fijo* o realice un redondeo forzado mediante ROUND.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Esta columna calculada de la tabla Products calcula la clasificación de ventas de cada producto del canal de Internet.

DAX

```
= RANKX(ALL(Products), SUMX(RELATEDTABLE(InternetSales), [SalesAmount]))
```

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

SAMPLE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve una muestra de N filas de la tabla especificada.

Sintaxis

DAX

```
SAMPLE(<n_value>, <table>, <orderBy_expression>, [<order>[,<orderBy_expression>, [<order>]]...])
```

Parámetros

Expandir tabla

Término	Definición
n_value	Número de filas que se va a devolver. Es cualquier expresión DAX que devuelve un único valor escalar, donde la expresión se va a evaluar varias veces (para cada fila o contexto). Si se especifica un valor (o expresión) no entero, el resultado se convierte en un entero.
mesa	Cualquier expresión DAX que devuelve una tabla de datos de donde extraer las "n" filas de muestra.
orderBy_expression	Cualquier expresión DAX escalar donde se evalúa el valor del resultado de cada fila de la <i>tabla</i> .
orden	(Opcional) Valor que especifica cómo ordenar valores <i>orderBy_expression</i> , de forma ascendente o descendente: 0 (cero), ordena de forma descendente los valores de <i>order_by</i> . 1, clasifica en orden ascendente los valores de <i>order_by</i> .

Valor devuelto

Tabla que consta de una muestra de N filas de *table* o tabla vacía si *n_value* es 0 (cero) o menos. Si se proporcionan argumentos OrderBy, el ejemplo será estable y determinista, devolverá la primera fila, la última y las filas distribuidas uniformemente entre ellas. Si no se especifica ningún orden, el ejemplo será aleatorio, no estable y no determinista.

Comentarios

- Si n_value es 0 (cero) o menos, SAMPLE devuelve una tabla vacía.
 - Para evitar valores duplicados en el ejemplo, la tabla proporcionada como segundo argumento se debe agrupar por la columna usada para la ordenación.
 - Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).
-

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

STDEV.S

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la desviación estándar de una población de muestra.

Sintaxis

DAX
<code>STDEV.S(<ColumnName>)</code>

Parámetros

Expandir tabla

Término	Definición
columnName	Nombre de una columna existente, con la sintaxis DAX estándar, normalmente completa. No puede ser una expresión.

Valor devuelto

Número que representa la desviación estándar de una población de muestra.

Excepciones

Notas

- STDEV.S da por hecho que la columna se refiere a una muestra de la población. Si los datos representan a toda la población, calcule la desviación estándar mediante STDEV.P.
- STDEV.S usa la fórmula siguiente:

$$\sqrt{\sum(x - \bar{x})^2 / (n-1)}$$

donde \tilde{x} es el valor medio de x para la muestra de la población y n es el tamaño de la población.

- Las filas en blanco se excluyen de *columnName* y no se tienen en cuenta en los cálculos.
- Se devuelve un error si *columnName* contiene menos de dos filas que no están en blanco.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se muestra la fórmula para una medida que calcula la desviación estándar de la columna, SalesAmount_USD, cuando la tabla InternetSales_USD es la población de muestra.

DAX

```
= STDEV.S(InternetSales_USD[SalesAmount_USD])
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Función STDEV.P

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la desviación estándar de toda la población.

Sintaxis

DAX
<code>STDEV.P(<ColumnName>)</code>

Parámetros

Expandir tabla

Término	Definición
columnName	Nombre de una columna existente, con la sintaxis DAX estándar, normalmente completa. No puede ser una expresión.

Valor devuelto

Número que representa la desviación estándar de toda la población.

Comentarios

- STDEV.P supone que la columna hace referencia a todo el relleno. Si los datos representan una muestra de la población, calcule la desviación estándar mediante STDEV.S.
- STDEV.P utiliza la fórmula siguiente:

$$\sqrt{\sum(x - \bar{x})^2/n}$$

donde \bar{x} es el valor medio de x para toda la población y n es el tamaño de la población.

- Las filas en blanco se excluyen de *columnName* y no se tienen en cuenta en los cálculos.
- Se devuelve un error si *columnName* contiene menos de dos filas que no están en blanco.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En este ejemplo se muestra la fórmula para una medida que calcula la desviación estándar de la columna, SalesAmount_USD, cuando la tabla InternetSales_USD representa toda la población.

DAX

```
= STDEV.P(InternetSales_USD[SalesAmount_USD])
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

STDEVX.S

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la desviación estándar de una población de muestra.

Sintaxis

DAX
<code>STDEVX.S(<table>, <expression>)</code>

Parámetros

Expandir tabla

Término	Definición
tabla	Cualquier expresión DAX que devuelve un único valor escalar, donde la expresión se va a evaluar varias veces (para cada fila o contexto).
expression	Cualquier expresión DAX que devuelve un único valor escalar, donde la expresión se va a evaluar varias veces (para cada fila o contexto).

Valor devuelto

Número con la desviación estándar de una población de muestra.

Excepciones

Notas

- STDEVX.S evalúa *expression* para cada fila de *table* y devuelve la desviación estándar de *expression* suponiendo que *table* hace referencia a una muestra de la población. Si *table* representa a toda la población, calcule la desviación estándar mediante STDEVX.P.
- STDEVX.S usa esta fórmula:

$$\sqrt{\sum(x - \bar{x})^2/(n-1)}$$

donde \bar{x} es el valor medio de x para toda la población y n es el tamaño de la población.

- Las filas en blanco se excluyen de *columnName* y no se tienen en cuenta en los cálculos.
- Se devuelve un error si *columnName* contiene menos de dos filas que no están en blanco.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En este ejemplo se muestra la fórmula de una columna calculada que calcula la desviación estándar del precio unitario por producto de una población de muestra, cuando se usa la fórmula en la tabla Product.

DAX

```
= STDEVX.S(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD]
- (InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Función STDEVX.P

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la desviación estándar de toda la población.

Sintaxis

DAX
<code>STDEVX.P(<table>, <expression>)</code>

Parámetros

Expandir tabla

Término	Definición
tabla	Cualquier expresión DAX que devuelve un único valor escalar, donde la expresión se va a evaluar varias veces (para cada fila o contexto).
expression	Cualquier expresión DAX que devuelve un único valor escalar, donde la expresión se va a evaluar varias veces (para cada fila o contexto).

Valor devuelto

Número que representa la desviación estándar de toda la población.

Notas

- STDEVX.P evalúa *expression* para cada fila de *table* y devuelve la desviación estándar de la expresión suponiendo que la tabla hace referencia a toda la población. Si los datos de *table* representan una muestra de la población, debe calcular la desviación estándar mediante STDEVX.S.
- STDEVX.P utiliza la fórmula siguiente:

$$\sqrt{\frac{\sum(x - \bar{x})^2}{n}}$$

donde \tilde{x} es el valor medio de x para toda la población y n es el tamaño de la población.

- Las filas en blanco se excluyen de *columnName* y no se tienen en cuenta en los cálculos.
- Se devuelve un error si *columnName* contiene menos de dos filas que no están en blanco.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se muestra la fórmula de una columna calculada que calcula la desviación estándar del precio unitario por producto, cuando la fórmula se usa en la tabla *Product*.

DAX

```
= STDEVX.P(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD]
- (InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

T.DIST

Artículo • 21/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la distribución t de cola izquierda de Student.

Sintaxis

DAX

```
T.DIST(X,Deg_freedom,Cumulative)
```

Parámetros

Expandir tabla

Término	Definición
X	Valor numérico en el que se debe evaluar la distribución.
Deg_freedom	Número entero que indica la cantidad de grados de libertad.
Cumulative	Valor lógico que determina el formato de la función. Si cumulative es TRUE, T.DIST devuelve la función de distribución acumulativa; si es FALSE, devuelve la función de densidad de probabilidad.

Valor devuelto

Distribución t de cola izquierda de Student.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

DAX

```
EVALUATE { T.DIST(60, 1, TRUE) }
```

Vea que devuelve lo siguiente:

 Expandir tabla

[Valor]
0,994695326367377

Contenido relacionado

[T.DIST.2T](#)

[T.DIST.RT](#)

[T.INV](#)

[T.INV.2t](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

T.DIST.2T

Artículo • 21/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la distribución t bilateral de Student.

Sintaxis

DAX

```
T.DIST.2T(X,Deg_freedom)
```

Parámetros

Expandir tabla

Término	Definición
X	Valor numérico en el que se debe evaluar la distribución.
Deg_freedom	Número entero que indica la cantidad de grados de libertad.

Valor devuelto

Distribución T bilateral de Student.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

DAX

```
EVALUATE { T.DIST.2T(1.959999998, 60) }
```

Devuelve

 Expandir tabla

[Valor]
0,054644929975921

Contenido relacionado

[T.DIST](#)
[T.DIST.RT](#)
[T.INV](#)
[T.INV.2t](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

T.DIST.RT

Artículo • 21/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la distribución t de Student de cola derecha.

Sintaxis

DAX

```
T.DIST.RT(X,Deg_freedom)
```

Parámetros

Expandir tabla

Término	Definición
X	Valor numérico en el que se debe evaluar la distribución.
Deg_freedom	Número entero que indica la cantidad de grados de libertad.

Valor devuelto

Distribución t de Student de cola derecha.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

DAX

```
EVALUATE { T.DIST.RT(1.95999998, 60) }
```

Devuelve

 Expandir tabla

[Valor]
0,0273224649879605

Contenido relacionado

[T.DIST](#)
[T.DIST.2T](#)
[T.INV](#)
[T.INV.2t](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

T.INV

Artículo • 21/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el inverso de cola izquierda de la distribución t de Student.

Sintaxis

DAX

```
T.INV(Probability,Deg_freedom)
```

Parámetros

Expandir tabla

Término	Definición
Probabilidad	Probabilidad asociada a la distribución t de Student.
Deg_freedom	Cantidad de grados de libertad con la que se debe caracterizar la distribución.

Valor devuelto

Inverso de cola izquierda de la distribución t de Student.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

DAX

```
EVALUATE { T.INV(0.75, 2) }
```

Devuelve

 Expandir tabla

[Valor]
0,816496580927726

Contenido relacionado

[T.INV.2T](#)

[T.DIST](#)

[T.DIST.2T](#)

[T.DIST.RT](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

T.INV.2T

Artículo • 21/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el inverso bilateral de la distribución t de Student.

Sintaxis

DAX

```
T.INV.2T(Probability,Deg_freedom)
```

Parámetros

Expandir tabla

Término	Definición
Probabilidad	Probabilidad asociada a la distribución t de Student.
Deg_freedom	Cantidad de grados de libertad con la que se debe caracterizar la distribución.

Valor devuelto

Inverso bilateral de la distribución t de Student.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

DAX

```
EVALUATE { T.INV.2T(0.546449, 60) }
```

Devuelve

 Expandir tabla

[Valor]
0,606533075825759

Contenido relacionado

[T.INV](#)
[T.DIST](#)
[T.DIST.2T](#)
[T.DIST.RT](#)

Comentarios

¿Le ha resultado útil esta página?



VAR.S

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la varianza de una población de muestra.

Sintaxis

DAX
<code>VAR.S(<columnName>)</code>

Parámetros

Expandir tabla

Término	Definición
columnName	Nombre de una columna existente, con la sintaxis DAX estándar, normalmente completa. No puede ser una expresión.

Valor devuelto

Número con la varianza de una población de ejemplo.

Comentarios

- VAR.S da por hecho que la columna se refiere a una muestra de la población. Si los datos representan una muestra de la población, calcule la varianza mediante VAR.P.
- VAR.S usa esta fórmula:

$$\sum(x - \bar{x})^2/(n-1)$$

donde \bar{x} es el valor medio de x para la población de muestra

y n es el tamaño de la población.

- Las filas en blanco se excluyen de *columnName* y no se tienen en cuenta en los cálculos.
- Se devuelve un error si *columnName* contiene menos de dos filas que no están en blanco.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En este ejemplo se muestra la fórmula de una medida que calcula la varianza de la columna SalesAmount_USD a partir de InternetSales_USD de una población de muestra.

DAX

```
= VAR.S(InternetSales_USD[SalesAmount_USD])
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Función VAR.P

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la varianza del total de la población.

Sintaxis

DAX
<code>VAR.P(<columnName>)</code>

Parámetros

Expandir tabla

Término	Definición
columnName	Nombre de una columna existente, con la sintaxis DAX estándar, normalmente completa. No puede ser una expresión.

Valor devuelto

Número con la varianza del total de la población.

Comentarios

- VAR.P supone que la columna hace referencia a todo el relleno. Si los datos representan una muestra de la población, calcule la varianza mediante VAR.S.
- VAR.P usa la siguiente fórmula:

$$\sum(x - \bar{x})^2/n$$

donde \bar{x} es el valor medio de x para toda la población

y n es el tamaño de la población.

- Las filas en blanco se excluyen de *columnName* y no se tienen en cuenta en los cálculos.

- Se devuelve un error si *columnName* contiene menos de dos filas que no están en blanco.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se muestra la fórmula de una medida que estima la varianza de la columna SalesAmount_USD de la tabla InternetSales_USD de toda la población.

DAX

```
= VAR.P(InternetSales_USD[SalesAmount_USD])
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

VARX.S

Artículo • 28/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve la varianza de una población de muestra.

Sintaxis

DAX
<code>VARX.S(<table>, <expression>)</code>

Parámetros

[] Expandir tabla

Término	Definición
tabla	Cualquier expresión DAX que devuelve una tabla de datos.
expression	Cualquier expresión DAX que devuelve un único valor escalar, donde la expresión se va a evaluar varias veces (para cada fila o contexto).

Valor devuelto

Número que representa la varianza de una población de muestra.

Comentarios

- VARX.S evalúa el valor de *expression* para cada fila de *table* y devuelve la varianza de *expression*, suponiendo que *table* haga referencia a una muestra de la población. Si *table* representa toda la población, debe calcular la varianza mediante VARX.P.
- VAR.S usa esta fórmula:

$$\sum(x - \bar{x})^2/(n-1)$$

donde \bar{x} es el valor medio de x para la población de muestra

y n es el tamaño de la población.

- Las filas en blanco se excluyen de *columnName* y no se tienen en cuenta en los cálculos.
- Se devuelve un error si *columnName* contiene menos de dos filas que no están en blanco.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En este ejemplo se muestra la fórmula de una columna calculada que calcula la varianza del precio unitario por producto de una población de muestra, cuando se usa la fórmula en la tabla Product.

DAX

```
= VARX.S(InternetSales_USD, InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Función VARX.P

Artículo • 28/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve la varianza del total de la población.

Sintaxis

DAX
<code>VARX.P(<table>, <expression>)</code>

Parámetros

Expandir tabla

Término	Definición
tabla	Cualquier expresión DAX que devuelve una tabla de datos.
expression	Cualquier expresión DAX que devuelve un único valor escalar, donde la expresión se va a evaluar varias veces (para cada fila o contexto).

Valor devuelto

Número con la varianza del total de la población.

Comentarios

- VARX.P evalúa `<expression>` para cada fila de `<table>` y devuelve la desviación de `<expression>` suponiendo que `<table>` hace referencia a todo el relleno. Si `<table>` representa una muestra de la población, calcule la varianza mediante VARX.S.
- VARX.P usa la siguiente fórmula:

$$\sum(x - \bar{x})^2/n$$

donde \bar{x} es el valor medio de x para toda la población

y n es el tamaño de la población.

- Las filas en blanco se excluyen de *columnName* y no se tienen en cuenta en los cálculos.
- Se devuelve un error si *columnName* contiene menos de dos filas que no están en blanco.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En este ejemplo se muestra la fórmula de una columna calculada que calcula la varianza del precio unitario por producto, cuando se usa la fórmula en la tabla Product.

DAX

```
= VARX.P(InternetSales_USD, InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Funciones de manipulación de tablas

Artículo • 20/10/2023

Estas funciones devuelven una tabla o manipulan las tablas existentes.

En esta categoría

Función	Descripción
ADDCOLUMNS	Agrega columnas calculadas a la tabla o expresión de tabla dada.
ADDMISSINGITEMS	Agrega combinaciones de elementos de varias columnas a una tabla, si aún no existen.
CROSSJOIN	Devuelve una tabla que contiene el producto cartesiano de todas las filas de todas las tablas de argumentos.
CURRENTGROUP	Devuelve un conjunto de filas del argumento de tabla de una expresión GROUPBY.
DATATABLE	Proporciona un mecanismo para declarar un conjunto insertado de valores de datos.
DETAILROWS	Evalúa una Expresión de filas de detalles definida para una medida y devuelve los datos.
Columna DISTINCT	Devuelve una tabla de una columna que contiene los valores distintos de la columna especificada.
Tabla DISTINCT	Devuelve una tabla quitando las filas duplicadas a partir de otra tabla o expresión.
EXCEPT	Devuelve las filas de una tabla que no aparecen en otra tabla.
FILTERS	Devuelve una tabla de valores que se aplican directamente como filtros a <i>columnName</i> .
GENERATE	Devuelve una tabla con el producto cartesiano entre cada fila de <i>table1</i> y la tabla que se obtiene al evaluar <i>table2</i> en el contexto de la fila actual de <i>table1</i> .
GENERATEALL	Devuelve una tabla con el producto cartesiano entre cada fila de <i>table1</i> y la tabla que se obtiene al evaluar <i>table2</i> en el contexto de la fila actual de <i>table1</i> .
GENERATESERIES	Devuelve una tabla de una sola columna que contiene los valores de una serie aritmética.

Función	Descripción
GROUPBY	De forma similar a la función SUMMARIZE, GROUPBY no realiza una operación CALCULATE implícita para las columnas de extensión que agrega.
IGNORE	Modifica SUMMARIZECOLUMNS mediante la omisión de expresiones específicas de la evaluación BLANK/NULL.
INTERSECT	Devuelve la intersección de fila de dos tablas y conserva los duplicados.
NATURALINNERJOIN	Realiza una combinación interna de una tabla con otra.
NATURALLEFTOUTERJOIN	Realiza una combinación de LeftTable con RightTable.
ROLLUP	Modifica el comportamiento de SUMMARIZE mediante la incorporación de filas de acumulación al resultado en las columnas que define el parámetro groupBy_columnName.
ROLLUPADDISSUBTOTAL	Modifica el comportamiento de SUMMARIZECOLUMNS mediante la incorporación de filas de acumulación o subtotal al resultado con base en las columnas de groupBy_columnName.
ROLLUPISSUBTOTAL	Asocia los grupos de acumulación con la columna que agrega ROLLUPADDISSUBTOTAL en una expresión ADDMISSINGITEMS.
ROLLUPGROUP	Modifica el comportamiento de SUMMARIZE y SUMMARIZECOLUMNS mediante la incorporación de filas de acumulación al resultado en las columnas que define el parámetro groupBy_columnName.
ROW	Devuelve una tabla con una sola fila que contiene los valores resultantes de las expresiones proporcionadas a cada columna.
SELECTCOLUMNS	Agrega columnas calculadas a la tabla o expresión de tabla dada.
SUBSTITUTEWITHINDEX	Devuelve una tabla que representa una semicombinación izquierda de las dos tablas proporcionadas como argumentos.
SUMMARIZE	Devuelve una tabla de resumen de los totales solicitados en un conjunto de grupos.
SUMMARIZECOLUMNS	Devuelve una tabla de resumen sobre un conjunto de grupos.
Constructor de tabla	Devuelve una tabla de una o varias columnas.
TOPN	Devuelve las N filas superiores de la tabla especificada.
TREATAS	Aplica el resultado de una expresión de tabla como filtros a las columnas de una tabla no relacionada.

Función	Descripción
UNION	Crea una tabla de unión (combinación) a partir de un par de tablas.
VALUES	Devuelve una tabla de una columna que contiene los valores distintos de la tabla o columna especificada.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

ADDCOLUMNS

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Agrega columnas calculadas a la tabla o expresión de tabla dada.

Sintaxis

DAX

```
ADDCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]...)
```

Parámetros

Expandir tabla

Término	Definición
table	Cualquier expresión DAX que devuelve una tabla de datos.
name	Nombre dado a la columna, entre comillas dobles.
expression	Cualquier expresión DAX que devuelve una expresión escalar, evaluada para cada fila de <i>table</i> .

Valor devuelto

Tabla con todas sus columnas originales y las agregadas.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se devuelve una versión ampliada de la tabla Product Category que incluye los valores de ventas totales del canal de revendedores y las ventas por Internet.

DAX

```
ADDCOLUMNS(ProductCategory
    , "Internet Sales", SUMX(RELATEDTABLE(InternetSales_USD), InternetSales_USD[SalesAmount_USD])
    , "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))
```

En la tabla siguiente se muestra una vista previa de los datos tal como la recibiría cualquier función en espera de recibir una tabla:

Expandir tabla

ProductCategory[ProductCategoryName]	ProductCategory[ProductCategoryAlternateKey]	ProductCategory[ProductCategoryKey]	[Internet Sales]	[Reseller Sales]
Bicicletas	1	1	25107749,77	6308
Componentes	2	2		1120
Ropa	3	3	306157,5829	1669
Accesorios	4	4	640920,1338	5343

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ADDMISSINGITEMS

Artículo • 30/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✗ Cálculo visual

Agrega filas con valores vacíos a una tabla que devuelve [SUMMARIZECOLUMNS](#).

Sintaxis

DAX

```
ADDMISSINGITEMS ( [<showAll_columnName> [, <showAll_columnName> [, ... ] ] ],  
<table> [, <groupBy_columnName> [, [<filterTable>] [, <groupBy_columnName>  
[, [<filterTable>] [, ... ] ] ] ] ] )
```

Parámetros

[] Expandir tabla

Término	Definición
showAll_columnName	(Opcional) Columna para la que se van a devolver elementos sin datos para las medidas usadas. Si no se especifica, se devolverán todas las columnas.
mesa	Tabla SUMMARIZECOLUMNS .
groupBy_columnName	(Opcional) Columna por la que se va a realizar una agrupación en el argumento de tabla proporcionado.
filterTable	(Opcional) Expresión de tabla que define las filas que se devuelven.

Valor devuelto

Una tabla con una o varias columnas.

Comentarios

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Con SUMMARIZECOLUMNS

Una tabla que devuelva `SUMMARIZECOLUMNS` incluirá únicamente las filas con valores. Al incluir una expresión `SUMMARIZECOLUMNS` en una expresión `ADDMISSINGITEMS`, también se devuelven las filas que no contienen valores.

Ejemplo

Sin `ADDMISSINGITEMS`, la consulta siguiente:

```
DAX  
SUMMARIZECOLUMNS(  
    'Sales'[CustomerId],  
    "Total Qty", SUM ( Sales[TotalQty] )  
)
```

Vea que devuelve lo siguiente:

 Expandir tabla

CustomerId	TotalQty
A	5
B	3
C	3
E	2

Con `ADDMISSINGITEMS`, la consulta siguiente:

```
DAX  
EVALUATE  
ADDMISSINGITEMS (  
    'Sales'[CustomerId],  
    SUMMARIZECOLUMNS(  
        'Sales'[CustomerId],  
        "Total Qty", SUM ( Sales[TotalQty] )  
,  
        'Sales'[CustomerId]  
)
```

Vea que devuelve lo siguiente:

 Expandir tabla

CustomerId	TotalQty
A	5
B	3
C	3
D	
E	2
F	

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CROSSJOIN

Artículo • 30/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve una tabla que contiene el producto cartesiano de todas las filas de todas las tablas de argumentos. Las columnas de la nueva tabla son todas las columnas de todas las tablas de argumentos.

Sintaxis

DAX

`CROSSJOIN(<table>, <table>[, <table>]...)`

Parámetros

[] Expandir tabla

Término	Definición
tabla	Cualquier expresión DAX que devuelve una tabla de datos.

Valor devuelto

Una tabla que contiene el producto cartesiano de todas las filas de todas las tablas de argumentos.

Observaciones

- Los nombres de columna de los argumentos *table* deben ser diferentes en todas las tablas o se devuelve un error.
- El número total de filas devueltas por CROSSJOIN() es igual al producto del número de filas de todas las tablas de los argumentos. Además, el número total de columnas de la tabla de resultados es la suma del número de columnas de todas las tablas. Por ejemplo, si **TableA** tiene r_A filas y c_A columnas, **TableB** tiene r_B filas y c_B columnas, y **TableC** tiene r_C filas y c_C columnas, entonces la tabla resultante tiene $r_A \times r_B \times r_C$ filas y $c_A + c_B + c_C$ columnas.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En este ejemplo se muestran los resultados de aplicar CROSSJOIN() a dos tablas: **Colores** y **Diseño de fondo**.

La tabla **Colores** contiene colores y patrones:

 Expandir tabla

Color	Patrón
Rojo	Franja horizontal
Verde	Franja vertical
Azul	Trama

La tabla **Diseño de fondo** contiene fuentes y presentación:

 Expandir tabla

Fuente	Presentación
serif	embossed
sans-serif	engraved

Aquí se muestra la expresión para generar la combinación cruzada:

DAX

`CROSSJOIN(Colors, Stationery)`

Cuando se usa la expresión anterior siempre que se espera una expresión de tabla, los resultados de la expresión serían los siguientes:

 Expandir tabla

Color	Patrón	Fuente	Presentación
Rojo	Franja horizontal	serif	embossed

Color	Patrón	Fuente	Presentación
Verde	Franja vertical	serif	embossed
Azul	Trama	serif	embossed
Rojo	Franja horizontal	sans-serif	engraved
Verde	Franja vertical	sans-serif	engraved
Azul	Trama	sans-serif	engraved

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CURRENTGROUP

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve un conjunto de filas del argumento table de una expresión [GROUPBY](#) que pertenecen a la fila actual del resultado de [GROUPBY](#).

Sintaxis

DAX

```
CURRENTGROUP ( )
```

Parámetros

None

Valor devuelto

Filas del argumento table de la función [GROUPBY](#) correspondiente a un grupo de valores de los argumentos groupBy_columnName.

Comentarios

- Esta función solo se puede usar en una expresión [GROUPBY](#).
- Esta función no toma ningún argumento y solo se admite como primer argumento en una de las funciones de agregación siguientes: [AVERAGEX](#), [COUNTAX](#), [COUNTX](#), [GEOMEANX](#), [MAXX](#), [MINX](#), [PRODUCTX](#), [STDEVX.S](#), [STDEVX.P](#), [SUMX](#), [VARX.S](#) y [VARX.P](#).

Ejemplo

Vea GROUPBY.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DATABASE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Proporciona un mecanismo para declarar un conjunto insertado de valores de datos.

Sintaxis

DAX
<pre>DATABASE (ColumnName1, DataType1, ColumnName2, DataType2..., {{Value1, Value2...}, {ValueN, ValueN+1...}}...)</pre>

Parámetros

Expandir tabla

Término	Definición
ColumnName	Cualquier expresión DAX que devuelve una tabla.
DataType	Enumeración que incluye: INTEGER, DOUBLE, STRING, BOOLEAN, CURRENCY, DATETIME
Valor	<p>Un solo argumento que usa la sintaxis de Excel para una constante de matriz unidimensional anidada para proporcionar una matriz de matrices. Este argumento representa el conjunto de valores de datos que se van a incluir en la tabla.</p> <p>Por ejemplo,</p> <pre>{ {values in row1}, {values in row2}, {values in row3}, etc. }</pre> <p>Donde {values in row1} es un conjunto delimitado por comas de expresiones constantes, es decir, una combinación de constantes, combinado con una serie de funciones básicas que incluyen DATE, TIME y BLANK, así como un operador más entre DATE y TIME y un operador menos unario para que se puedan expresar valores negativos.</p> <p>Los siguientes son todos los valores válidos: 3, -5, BLANK(), "2009-04-15 02:45:21". Puede que los valores no hagan referencia a nada fuera de la expresión inmediata, y no pueden hacer referencia a columnas, tablas, relaciones ni nada más.</p>

Término	Definición
	Un valor que falta se trata de forma idéntica a BLANK(). Por ejemplo, los siguientes son iguales: {1,2,BLANK(),4} {1,2,,4}

Valor devuelto

Tabla que declara un conjunto insertado de valores.

Observaciones

- A diferencia de DATATABLE, el [constructor de tabla](#) permite cualquier expresión escalar como valor de entrada.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

DAX

```
= DataTable("Name", STRING,
            "Region", STRING
            ,{
                {"User1", "East"},  

                {"User2", "East"},  

                {"User3", "West"},  

                {"User4", "West"},  

                {"User4", "East"}  

            }  

        )
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DETAILROWS

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Evalúa una expresión de filas de detalles definida para una medida y devuelve los datos.

Sintaxis

DAX
DETAILROWS([Measure])

Parámetros

Expandir tabla

Término	Definición
Measure	Nombre de una medida.

Valor devuelto

Una tabla con los datos que devuelve la expresión de filas de detalles. Si no hay definida ninguna expresión de filas de detalles, se devuelven los datos de la tabla que contiene la medida.

Comentarios

¿Le ha resultado útil esta página?

Sí

No

DISTINCT (columna)

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve una tabla de una columna que contiene los valores distintos de la columna especificada. En otras palabras, los valores duplicados se quitan y solo se devuelven los valores únicos.

Nota

Esta función no se puede usar para Valores devueltos en una celda o columna de una hoja de cálculo, sino que se anida la función DISTINCT en una fórmula para obtener una lista de valores distintos que se pueden pasar a otra función y, después, se cuentan, suman o usan para otras operaciones.

Sintaxis

DAX

`DISTINCT(<column>)`

Parámetros

Expandir tabla

Término	Definición
columna	Columna de la que se van a devolver valores únicos. O una expresión que devuelve una columna.

Valor devuelto

Una columna de valores únicos.

Notas

- Los resultados de DISTINCT se ven afectados por el contexto de filtro actual. Por ejemplo, si utiliza la fórmula en el ejemplo siguiente para crear una medida, los resultados cambiarán cada vez que se filtre la tabla para mostrar solo una región determinada o un periodo de tiempo.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Funciones relacionadas

Existe otra versión de la función DISTINCT, [DISTINCT \(tabla\)](#), que devuelve una tabla quitando las filas duplicadas de otra tabla o expresión.

La función VALUES es similar a DISTINCT; también se puede utilizar para devolver una lista de valores únicos y, por lo general, devolverá exactamente los mismos resultados que DISTINCT. Sin embargo, en algún contexto VALUES devolverá un valor especial adicional. Para obtener más información, vea [Función VALUES](#).

Ejemplo

La fórmula siguiente cuenta el número de clientes únicos que han generado pedidos a través del canal de Internet. En la tabla siguiente se muestran los posibles resultados cuando se agrega la fórmula a un informe.

DAX

```
= COUNTROWS(DISTINCT(InternetSales_USD[CustomerKey]))
```

No se puede pegar la lista de valores que DISTINCT devuelve directamente en una columna. En su lugar, se pasan los resultados de la función DISTINCT a otra función que cuenta, filtra o agrega valores mediante la lista. Para que el ejemplo sea lo más sencillo posible, se ha pasado la tabla de valores distintos a la función COUNTROWS.

Expandir tabla

Etiquetas de fila	Accesorios	Bicicletas	Ropa	Total general
2005		1013		1013
2006		2677		2677
2007	6792	4875	2867	9309

Etiquetas de fila	Accesorios	Bicicletas	Ropa	Total general
2008	9435	5451	4196	11 377
Total general	15114	9132	6852	18 484

Además, tenga en cuenta que los resultados no son aditivos. Es decir, el número total de clientes únicos en *2007* no es la suma de los clientes únicos de *Accesorios*, *Bicicletas* y *Ropa* de ese año. El motivo es que un cliente se puede contar en varios grupos.

Contenido relacionado

[Funciones de filtro](#)

[FILTER \(función\)](#)

[Función RELATED](#)

[Función VALUES](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DISTINCT (tabla)

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve una tabla quitando las filas duplicadas a partir de otra tabla o expresión.

Sintaxis

DAX

```
DISTINCT(<table>)
```

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla de la que se van a devolver filas únicas. La tabla también puede ser una expresión que da como resultado una tabla.

Valor devuelto

Tabla que contiene solo filas distintas.

Funciones relacionadas

Existe otra versión de la función DISTINCT, [DISTINCT \(columna\)](#), que toma un nombre de columna como parámetro de entrada.

Ejemplo

La consulta siguiente:

DAX

```
EVALUATE DISTINCT( { (1, "A"), (2, "B"), (1, "A") } )
```

Devuelve la tabla:

 Expandir tabla

[Value1]	[Value2]
1	A
2	B

Contenido relacionado

[Funciones de filtro](#)

[DISTINCT \(columna\)](#)

[FILTER \(función\)](#)

[Función RELATED](#)

[Función VALUES](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

EXCEPT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve las filas de la primera tabla de la expresión que no aparecen en la segunda tabla.

Sintaxis

DAX

```
EXCEPT(<table_expression1>, <table_expression2>)
```

Parámetros

Expandir tabla

Término	Definición
Table_expression	Cualquier expresión DAX que devuelve una tabla.

Valor devuelto

Tabla que contiene las filas de una tabla menos todas las filas de otra tabla.

Observaciones

- Si aparece una fila en ambas tablas, esta y sus duplicados no están presentes en el conjunto de resultados. Si aparece una fila solo en table_expression1, esta y sus duplicados aparecerán en el conjunto de resultados.
- Los nombres de columna coincidirán con los nombres de columna de table_expression1.
- La tabla devuelta tiene linaje basado en las columnas de table_expression1, independientemente del linaje de las columnas de la segunda tabla. Por ejemplo, si la primera columna de la primera table_expression tiene linaje en la columna base C1 del modelo, Except reducirá las filas en función de la disponibilidad de los

valores de la primera columna de la segunda table_expression y mantendrá intacto el linaje de la columna base C1.

- Las dos tablas deben tener el mismo número de columnas.
- Las columnas se comparan en función de la posición y la comparación de datos sin coerción de tipos.
- El conjunto de filas devuelto depende del orden de las dos expresiones.
- La tabla devuelta no incluye las columnas de las tablas relacionadas con table_expression1.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

States1

 Expandir tabla

Estado
A
B
B
B
C
D
D

States2

 Expandir tabla

Estado
B
C

Estado
D
D
D
E
E
E

Except(States1, States2)

 Expandir tabla

Estado
A

Except(States2, States1)

 Expandir tabla

Estado
E
E
E

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

FILTERS

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve los valores que se aplican directamente como filtros a *columnName*.

Sintaxis

DAX

```
FILTERS(<columnName>)
```

Parámetros

Expandir tabla

Término	Descripción
columnName	Nombre de una columna existente, con la sintaxis de DAX estándar. No puede ser una expresión.

Valor devuelto

Valores que se aplican directamente como filtros a *columnName*.

Notas

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se muestra cómo determinar el número de filtros directos que tiene una columna.

DAX

```
= COUNTROWS(FILTERS(ResellerSales_USD[ProductKey]))
```

Este ejemplo permite saber cuántos filtros directos en ResellerSales_USD[ProductKey] se han aplicado en el contexto en el que se evalúa la expresión.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

GENERATE

Artículo • 20/10/2023

Devuelve una tabla con el producto cartesiano entre cada fila de *table1* y la tabla que se obtiene al evaluar *table2* en el contexto de la fila actual de *table1*.

Sintaxis

DAX

```
GENERATE(<table1>, <table2>)
```

Parámetros

Término	Definición
table1	Cualquier expresión DAX que devuelve una tabla.
table2	Cualquier expresión DAX que devuelve una tabla.

Valor devuelto

Una tabla con el producto cartesiano entre cada fila de *table1* y la tabla que se obtiene al evaluar *table2* en el contexto de la fila actual de *table1*.

Notas

- Si la evaluación de *table2* de la fila actual en *table1* devuelve una tabla vacía, la tabla de resultados no contendrá la fila actual de *table1*. Esto difiere de GENERATEALL(), donde la fila actual de *table1* se incluirá en los resultados, y las columnas correspondientes a *table2* tendrán valores null para esa fila.
- Todos los nombres de columna de *table1* y *table2* deben ser diferentes; si no, se devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el siguiente ejemplo, el usuario quiere una tabla de resumen de las ventas por región y categoría de producto para el canal distribuidores, como en la tabla siguiente:

SalesTerritory[SalesTerritoryGroup]	ProductCategory[ProductName]	[ResellerSales]
Europa	Accesorios	142 227,27 USD
Europa	Bicicletas	9 970 200,44 USD
Europa	Ropa	365 847,63 USD
Europa	Componentes	2 214 440,19 USD
Norteamérica	Accesorios	379 305,15 USD
Norteamérica	Bicicletas	52 403 796,85 USD
Norteamérica	Ropa	1 281 193,26 USD
Norteamérica	Componentes	8 882 848,05 USD
Pacífico	Accesorios	12 769,57 USD
Pacífico	Bicicletas	710 677,75 USD
Pacífico	Ropa	22 902,38 USD
Pacífico	Componentes	108 549,71 USD

La fórmula siguiente genera la tabla anterior:

```
DAX

GENERATE(
    SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])
    ,SUMMARIZE(ProductCategory
    , [ProductName]
    , "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD),
    ResellerSales_USD[SalesAmount_USD])
    )
)
```

1. La primera instrucción SUMMARIZE, `SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])`, genera una tabla de grupos de territorios, donde cada fila es un grupo de territorios, como se muestra aquí:

SalesTerritory[SalesTerritoryGroup]
Norteamérica

SalesTerritory[SalesTerritoryGroup]

Europa

Pacífico

NA

2. La segunda instrucción SUMMARIZE, `SUMMARIZE(ProductCategory, [ProductName], "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))`, genera una tabla de grupos de categoría de producto con las ventas de los distribuidores de cada grupo, como se muestra aquí:

ProductCategory[ProductName]	[ResellerSales]
Bicicletas	63 084 675,04 USD
Componentes	11 205 837,96 USD
Clothing	1 669 943,27 USD
Accessories	534 301,99 USD

3. Pero, cuando se toma la tabla anterior y se evalúa en el contexto de cada fila de la tabla de grupos de territorios, se obtienen resultados distintos de cada territorio.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

GENERATEALL

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve una tabla con el producto cartesiano entre cada fila de *table1* y la tabla que se obtiene al evaluar *table2* en el contexto de la fila actual de *table1*.

Sintaxis

DAX

```
GENERATEALL(<table1>, <table2>)
```

Parámetros

Expandir tabla

Término	Definición
table1	Cualquier expresión DAX que devuelve una tabla.
table2	Cualquier expresión DAX que devuelve una tabla.

Valor devuelto

Una tabla con el producto cartesiano entre cada fila de *table1* y la tabla que se obtiene al evaluar *table2* en el contexto de la fila actual de *table1*.

Notas

- Si la evaluación de *table2* de la fila actual de *table1* devuelve una tabla vacía, la fila actual de *table1* se incluirá en los resultados y las columnas correspondientes a *table2* tendrán valores NULL para esa fila. Esto es diferente que GENERATE(), donde la fila actual de *table1* no se incluirá en los resultados.
- Todos los nombres de columna de *table1* y *table2* deben ser diferentes; si no, se devuelve un error.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente, el usuario quiere una tabla de resumen de las ventas por región y categoría de producto para el canal distribuidores, como en la tabla siguiente:

 Expandir tabla

SalesTerritory[SalesTerritoryGroup]	ProductCategory[ProductName]	[ResellerSales]
Europa	Accesorios	142 227,27 USD
Europa	Bicicletas	9 970 200,44 USD
Europa	Ropa	365 847,63 USD
Europa	Componentes	2 214 440,19 USD
NA	Accesorios	
NA	Bicicletas	
NA	Ropa	
NA	Componentes	
Norteamérica	Accesorios	379 305,15 USD
Norteamérica	Bicicletas	52 403 796,85 USD
Norteamérica	Ropa	1 281 193,26 USD
Norteamérica	Componentes	8 882 848,05 USD
Pacífico	Accesorios	12 769,57 USD
Pacífico	Bicicletas	710 677,75 USD
Pacífico	Ropa	22 902,38 USD
Pacífico	Componentes	108 549,71 USD

La fórmula siguiente genera la tabla anterior:

DAX

```
GENERATEALL(
    SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup]))
```

```

, SUMMARIZE(ProductCategory
, [ProductCategoryName]
, "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD),
ResellerSales_USD[SalesAmount_USD])
)
)

```

1. El primer elemento SUMMARIZE genera una tabla de grupos de territorios, donde cada fila es un grupo de territorios, como los que se muestran a continuación:

 Expandir tabla

SalesTerritory[SalesTerritoryGroup]
Norteamérica
Europa
Pacífico
NA

2. El segundo elemento SUMMARIZE genera una tabla de grupos de categoría de producto con las ventas de los distribuidores de cada grupo, tal como se muestra a continuación:

 Expandir tabla

ProductCategory[ProductName]	[ResellerSales]
Bicicletas	63 084 675,04 USD
Componentes	11 205 837,96 USD
Clothing	1 669 943,27 USD
Accessories	534 301,99 USD

3. Sin embargo, cuando se toma la tabla anterior y se evalúa la tabla en el contexto de cada fila de la tabla de grupos de territorios, se obtienen resultados distintos para cada territorio.

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

GENERATESERIES

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve una tabla de una sola columna que contiene los valores de una serie aritmética, es decir, una secuencia de valores en la que cada valor difiere del anterior por una cantidad constante. El nombre de la columna devuelta es Value.

Sintaxis

DAX

```
GENERATESERIES(<startValue>, <endValue>[, <incrementValue>])
```

Parámetros

[] Expandir tabla

Término	Definición
startValue	Valor inicial que se usa para generar la secuencia.
endValue	Valor final que se usa para generar la secuencia.
incrementValue	(Opcional) Valor de incremento de la secuencia. Cuando no se proporciona, el valor predeterminado es 1.

Valor devuelto

Una tabla de una sola columna que contiene los valores de una serie aritmética. El nombre de la columna es Value.

Observaciones

- Cuando endValue es menor que startValue, se devuelve una tabla vacía.
- incrementValue debe ser un valor positivo.
- La secuencia se detiene en el último valor menor o igual que endValue.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo 1

La consulta DAX siguiente:

DAX
EVALUATE GENERATESERIES(1, 5)

Devuelve la tabla siguiente con una sola columna:

 Expandir tabla

[Valor]
1
2
3
4
5

Ejemplo 2

La consulta DAX siguiente:

DAX
EVALUATE GENERATESERIES(1.2, 2.4, 0.4)

Devuelve la tabla siguiente con una sola columna:

 Expandir tabla

[Valor]
1.2
1.6

[Valor]
2
2,4

Ejemplo 3

La consulta DAX siguiente:

DAX
EVALUATE GENERATESERIES(CURRENCY(10), CURRENCY(12.4), CURRENCY(0.5))

Devuelve la tabla siguiente con una sola columna:

 Expandir tabla

[Valor]
10
10,5
11
11,5
12

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

GROUPBY

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ! Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

La función GROUPBY es similar a la función [SUMMARIZE](#). Sin embargo, GROUPBY no realiza una operación [CALCULATE](#) implícita de las columnas de extensión que agrega. GROUPBY permite usar una nueva función, [CURRENTGROUP](#), dentro de las funciones de agregación en las columnas de extensión que agrega. GROUPBY se usa para realizar varias agregaciones en un solo recorrido de tabla.

Sintaxis

DAX

```
GROUPBY (<table> [, <groupBy_columnName> [, <groupBy_columnName> [, ...]]] [, <name>, <expression> [, <name>, <expression> [, ...]]])
```

Parámetros

[] Expandir tabla

Término	Definición
tabla	Cualquier expresión DAX que devuelve una tabla de datos.
groupBy_columnName	Nombre de una columna existente de la tabla (o de una tabla relacionada) por la que se van a agrupar los datos. Este parámetro no puede ser una expresión.
name	Nombre dado a una nueva columna que se va a agregar a la lista de columnas GroupBy, entre comillas dobles.
expression	Una de las funciones de agregación X cuyo primer argumento es CURRENTGROUP(). Consulte la sección siguiente Con CURRENTGROUP para obtener la lista completa de funciones de agregación X admitidas.

Valor devuelto

Una tabla con las columnas seleccionadas de los argumentos `groupBy_columnName` y las columnas de extensión diseñadas por los argumentos de nombre.

Comentarios

- La función GROUPBY hace lo siguiente:
 1. Empieza por la tabla especificada (y todas las tablas relacionadas en la dirección "a una").
 2. Crea una agrupación con todas las columnas GroupBy (que deben existir en la tabla del paso 1).
 3. Cada grupo es una fila en el resultado, pero representa un conjunto de filas en la tabla original.
 4. Por cada grupo, evalúa las columnas de extensión que se agregan. A diferencia de la función SUMMARIZE, no se realiza una operación CALCULATE implícita y el grupo no se coloca en el contexto de filtro.
- Cada columna para la que se define un nombre debe tener una expresión correspondiente; de lo contrario, se devuelve un error. El primer argumento, `name`, define el nombre de la columna en los resultados. El segundo argumento, `expression`, define el cálculo realizado para obtener el valor de cada fila de esa columna.
- `groupBy_columnName` debe estar en la tabla o en una tabla relacionada.
- Cada nombre debe ir entre comillas dobles.
- La función agrupa un conjunto de filas seleccionado en un conjunto de filas de resumen de los valores de una o más columnas `groupBy_columnName`. Se devuelve una fila de cada grupo.
- GROUPBY se usa principalmente para realizar agregaciones sobre resultados intermedios a partir de expresiones de tabla DAX. Para obtener agregaciones eficientes sobre tablas físicas del modelo, considere la posibilidad de usar las funciones [SUMMARIZECOLUMNS](#) o [SUMMARIZE](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Con CURRENTGROUP

[CURRENTGROUP](#) solo se puede usar en una expresión que define una columna de extensión dentro de la función GROUPBY. En efecto, [CURRENTGROUP](#) devuelve un conjunto de filas del argumento table de GROUPBY, que pertenecen a la fila actual del resultado de GROUPBY. La función [CURRENTGROUP](#) no toma ningún argumento y solo se admite como primer argumento en una de las funciones de agregación siguientes: [AVERAGEX](#), [COUNTAX](#), [COUNTX](#), [GEOMEANX](#), [MAXX](#), [MINX](#), [PRODUCTX](#), [STDEVX.S](#), [STDEVX.P](#), [SUMX](#), [VARX.S](#) y [VARX.P](#).

Ejemplo

En el ejemplo siguiente se calculan primero las ventas totales agrupadas por país y categoría de producto a través de las tablas físicas mediante la función [SUMMARIZECOLUMNNS](#). Después, usa la función GROUPBY para examinar el resultado intermedio desde el primer paso, a fin de encontrar el máximo de ventas en cada país en todas las categorías de productos.

```
DAX

DEFINE
    VAR SalesByCountryAndCategory =
        SUMMARIZECOLUMNNS(
            Geography[Country],
            Product[Category],
            "Total Sales", SUMX(Sales, Sales[Price] * Sales[Qty])
        )

EVALUATE
    GROUPBY(
        SalesByCountryAndCategory,
        Geography[Country],
        "Max Sales", MAXX(CURRENTGROUP(), [Total Sales])
    )
```

Contenido relacionado

[Función SUMMARIZE](#)
[Función SUMMARIZECOLUMNNS](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

IGNORE

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✗ Cálculo visual

Modifica el comportamiento de la función [SUMMARIZECOLUMNS](#) mediante la omisión de expresiones específicas de la evaluación BLANK/NULL. Las filas para las que todas las expresiones que no usan IGNORE devuelven BLANK/NULL se excluirán aunque las expresiones que usen IGNORE se evalúen como BLANK/NULL. Esta función solo se puede usar en una expresión [SUMMARIZECOLUMNS](#).

Syntax

DAX

```
IGNORE(<expression>)
```

Con [SUMMARIZECOLUMNS](#):

DAX

```
SUMMARIZECOLUMNS(<groupBy_columnName>[ , < groupBy_columnName >]...,  
[<filterTable>]...[ , <name>, IGNORE(...)]...)
```

Parámetros

[] Expandir tabla

Término	Definición
expression	Cualquier expresión DAX que devuelva un único valor (no una tabla).

Valor devuelto

Esta función no devuelve ningún valor.

Comentarios

Solo se puede usar IGNORE como argumento de expresión en [SUMMARIZECOLUMNS](#).

Ejemplo

Vea [SUMMARIZECOLUMNS](#).

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

INTERSECT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve la intersección de fila de dos tablas y conserva los duplicados.

Sintaxis

DAX

```
INTERSECT(<table_expression1>, <table_expression2>)
```

Parámetros

Expandir tabla

Término	Definición
Table_expression	Cualquier expresión DAX que devuelve una tabla.

Valor devuelto

Tabla que contiene todas las filas de table_expression1 que también están en table_expression2.

Excepciones

Notas

- Intersect no es conmutativa. En general, Intersect(T1, T2) tendrá un conjunto de resultados diferente que Intersect(T2, T1).
- Las filas duplicadas se conservan. Si aparece una fila en table_expression1 y table_expression2, esta y todos los duplicados de table_expression_1 se incluyen en el conjunto de resultados.
- Los nombres de columna coincidirán con los nombres de columna de table_expression1.

- La tabla devuelta tiene linaje basado en las columnas de table_expression1, independientemente del linaje de las columnas de la segunda tabla. Por ejemplo, si la primera columna de la primera table_expression tiene linaje en la columna base C1 del modelo, la intersección reducirá las filas en función de la intersección de la primera columna de la segunda table_expression y mantendrá intacto el linaje de la columna base C1.
- Las columnas se comparan en función de la posición y la comparación de datos sin coerción de tipos.
- La tabla devuelta no incluye las columnas de las tablas relacionadas con table_expression1.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

States1

 Expandir tabla

Estado
A
A
B
B
B
C
D
D

States2

 Expandir tabla

Estado
B
C
D
D
D
E

Intersect(States1, States2)

 Expandir tabla

Estado
B
B
B
C
D
D

Intersect(States2, States1)

 Expandir tabla

Estado
B
C
D
D
D

Comentarios

¿Le ha resultado útil esta página?



Sí



No

NATURALINNERJOIN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Realiza una combinación interna de una tabla con otra.

Sintaxis

DAX

```
NATURALINNERJOIN(<LeftTable>, <RightTable>)
```

Parámetros

Expandir tabla

Término	Definición
LeftTable	Expresión de tabla que define la tabla del lado izquierdo de la combinación.
RightTable	Expresión de tabla que define la tabla del lado derecho de la combinación.

Valor devuelto

Tabla que incluye solo las filas para las que los valores de las columnas comunes especificadas están presentes en ambas tablas. La tabla devuelta tiene las columnas comunes de la tabla izquierda y otras columnas de ambas tablas.

Observaciones

- Las tablas se combinan en columnas comunes (por nombre) de las dos tablas. Si las dos tablas no tienen nombres de columna comunes, se devuelve un error.
- No hay garantía de criterio de ordenación para los resultados.
- Las columnas que se combinan deben tener el mismo tipo de datos en ambas tablas.

- Solo se combinan las columnas de la misma tabla de origen (tienen el mismo linaje). Por ejemplo, Products[ProductID], WebSales[ProductdID], StoreSales[ProductdID] con relaciones de varios a uno entre WebSales y StoreSales y la tabla Products basada en la columna ProductID, las tablas WebSales y StoreSales se combinan en [ProductID].
- Durante la combinación se usa semántica de comparación estricta. No hay ninguna coerción de tipos; por ejemplo, 1 no es igual a 1,0.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Contenido relacionado

[NATURALLEFTOUTERJOIN](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NATURALLEFTOUTERJOIN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Realiza una combinación de LeftTable con RightTable mediante la semántica de combinación externa izquierda.

Sintaxis

DAX

`NATURALLEFTOUTERJOIN(<LeftTable>, <RightTable>)`

Parámetros

Expandir tabla

Término	Definición
LeftTable	Expresión de tabla que define la tabla del lado izquierdo de la combinación.
RightTable	Expresión de tabla que define la tabla del lado derecho de la combinación.

Valor devuelto

Tabla que incluye solo las filas de RightTable para las que los valores de las columnas comunes especificadas también están presentes en LeftTable. La tabla devuelta tendrá las columnas comunes de la tabla izquierda y las otras columnas de las dos tablas.

Observaciones

- Las tablas se combinan en columnas comunes (por nombre) de las dos tablas. Si las dos tablas no tienen nombres de columna comunes, se devuelve un error.
- No hay garantía de criterio de ordenación para los resultados.
- Las columnas que se combinan deben tener el mismo tipo de datos en ambas tablas.

- Solo se combinan las columnas de la misma tabla de origen (tienen el mismo linaje). Por ejemplo, Products[ProductID], WebSales[ProductdID], StoreSales[ProductdID] con relaciones de varios a uno entre WebSales y StoreSales y la tabla Products basada en la columna ProductID, las tablas WebSales y StoreSales se combinan en [ProductID].
- Durante la combinación se usa semántica de comparación estricta. No hay ninguna coerción de tipos; por ejemplo, 1 no es igual a 1,0.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Contenido relacionado

[NATURALINNERJOIN](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ROLLUP

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✗ Cálculo visual

Modifica el comportamiento de la función [SUMMARIZE](#) mediante la incorporación de filas de acumulación al resultado en las columnas que define el parámetro `groupBy_columnName`. Esta función solo se puede usar en una expresión [SUMMARIZE](#).

Syntax

DAX

```
ROLLUP ( <groupBy_columnName> [, <groupBy_columnName> [, ...] ] )
```

Con [SUMMARIZE](#):

DAX

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[,  
ROLLUP(<groupBy_columnName>[, <groupBy_columnName>...]])[, <name>,  
<expression>]...)
```

Parámetros

 Expandir tabla

Término	Definición
groupBy_columnName	Nombre completo de una columna existente o función ROLLUPGROUP que se va a usar para crear grupos de resumen basados en los valores que incluye. Este parámetro no puede ser una expresión.

Valor devuelto

Esta función no devuelve ningún valor. Solo especifica el conjunto de columnas de las que se va a calcular el subtotal.

Comentarios

Esta función solo se puede usar en una expresión [SUMMARIZE](#).

Ejemplo

Vea [SUMMARIZE](#).

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ROLLUPADDISSUBTOTAL

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medición ✗ Cálculo visual

Modifica el comportamiento de la función [SUMMARIZECOLUMNS](#) mediante la incorporación de filas de acumulación o subtotal al resultado con base en las columnas de `groupBy_columnName`. Esta función solo se puede usar en una expresión [SUMMARIZECOLUMNS](#).

Sintaxis

DAX

```
ROLLUPADDISSUBTOTAL ( [<grandtotalFilter>], <groupBy_columnName>, <name> [, [<groupLevelFilter>] [, <groupBy_columnName>, <name> [, [<groupLevelFilter>] [, ... ] ] ] ] )
```

Parámetros

[+] Expandir tabla

Término	Definición
grandtotalFilter	(Opcional) Filtro que se debe aplicar al nivel general.
groupBy_columnName	Nombre de una columna existente que se va a usar para crear grupos de resumen basados en los valores que incluye. No puede ser una expresión.
name	Nombre de una columna ISSUBTOTAL. Los valores de la columna se calculan mediante la función ISSUBTOTAL.
groupLevelFilter	(Opcional) Filtro que se debe aplicar al nivel actual.

Valor devuelto

Esta función no devuelve ningún valor.

Observaciones

None

Ejemplo

Vea [SUMMARIZECOLUMNS](#).

Comentarios

¿Le ha resultado útil esta página?



Sí



No

ROLLUPGROUP

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✗ Cálculo visual

Modifica el comportamiento de las funciones [SUMMARIZE](#) y [SUMMARIZECOLUMNS](#) mediante la incorporación de filas de resumen al resultado en las columnas que define el parámetro `groupBy_columnName`. Esta función solo se puede usar en una expresión [SUMMARIZE](#) o [SUMMARIZECOLUMNS](#).

Sintaxis

DAX

```
ROLLUPGROUP ( <groupBy_columnName> [, <groupBy_columnName> [, ... ] ] )
```

Parámetros

[] Expandir tabla

Término	Definición
groupBy_columnName	Nombre completo de una columna existente o función ROLLUPGROUP que se va a usar para crear grupos de resumen basados en los valores que incluye. Este parámetro no puede ser una expresión.

Valor devuelto

Esta función no devuelve ningún valor. Marca un conjunto de columnas que [ROLLUP](#) o [ROLLUPADDISSUBTOTAL](#) van a tratar como un solo grupo durante los subtotales.

Comentarios

ROLLUPGROUP solamente se puede usar como argumento de `groupBy_columnName` para [ROLLUP](#), [ROLLUPADDISSUBTOTAL](#) o [ROLLUPISSUBTOTAL](#).

Ejemplo

Vea [SUMMARIZE](#) y [SUMMARIZECOLUMNS](#).

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ROLLUPISSUBTOTAL

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✗ Cálculo visual

Asocia los grupos de acumulación con la columna que agrega [ROLLUPADDISSTOTAL](#). Esta función solo se puede usar en una expresión [ADDMISSINGITEMS](#).

Sintaxis

DAX

```
ROLLUPISSUBTOTAL ( [<grandTotalFilter>], <groupBy_columnName>,
<isSubtotal_columnName> [, [<groupLevelFilter>] [, <groupBy_columnName>,
<isSubtotal_columnName> [, [<groupLevelFilter>] [, ... ] ] ] ] )
```

Parámetros

[] Expandir tabla

Término	Definición
grandTotalFilter	(Opcional) Filtro que se debe aplicar al nivel general.
groupBy_columnName	Nombre de una columna existente que se va a usar para crear grupos de resumen basados en los valores que incluye. No puede ser una expresión.
isSubtotal_columnName	Nombre de una columna ISSUBTOTAL. Los valores de la columna se calculan mediante la función ISSUBTOTAL.
groupLevelFilter	(Opcional) Filtro que se debe aplicar al nivel actual.

Valor devuelto

None

Observaciones

Esta función solo se puede usar en una expresión [ADDMISSINGITEMS](#).

Comentarios

¿Le ha resultado útil esta página?



Sí



No

Función ROW

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve una tabla con una sola fila que contiene los valores resultantes de las expresiones proporcionadas a cada columna.

Sintaxis

DAX

```
ROW(<name>, <expression>[ [ ,<name>, <expression>]... ])
```

Parámetros

Expandir tabla

Término	Definición
name	Nombre dado a la columna, entre comillas dobles.
expression	Cualquier expresión DAX que devuelve un valor escalar único para llenar <i>name</i> .

Valor devuelto

Tabla de una sola fila.

Observaciones

- Los argumentos deben ir siempre en pares de *name* y *expression*.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se devuelve una tabla de una sola fila con las ventas totales de canales de Internet y revendedores.

DAX

```
ROW("Internet Total Sales (USD)", SUM(InternetSales_USD[SalesAmount_USD]),  
    "Resellers Total Sales (USD)",  
    SUM(ResellerSales_USD[SalesAmount_USD]))
```

Comentarios

¿Le ha resultado útil esta página?



SELECTCOLUMNS

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve una tabla con las columnas seleccionadas de la tabla y las columnas nuevas especificadas por las expresiones DAX.

Sintaxis

DAX

```
SELECTCOLUMNS(<Table>, [<Name>], <Expression>, <Name>], ...)
```

Parámetros

Expandir tabla

Término	Definición
Tabla	Cualquier expresión DAX que devuelve una tabla.
Nombre	Nombre dado a la columna, entre comillas dobles.
Expression	Cualquier expresión que devuelve un valor escalar como una referencia de columna o un valor de cadena o entero.

Valor devuelto

Una tabla con el mismo número de filas que la tabla especificada como primer argumento. La tabla devuelta tiene una columna para cada par de argumentos `<Name>` y `<Expression>`, y cada expresión se evalúa en el contexto de una fila a partir del argumento `<Table>` especificado.

Comentarios

`SELECTCOLUMNS` tiene la misma firma que `ADDCOLUMNS` y el mismo comportamiento, pero en lugar de comenzar con la `<Table>` especificada, `SELECTCOLUMNS` comienza con una tabla vacía antes de agregar columnas.

Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Para la tabla siguiente, denominada **Customer**:

 Expandir tabla

País	State	Count	Total
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900
EE. UU.	CA	5	500
EE. UU.	WA	10	900

DAX

```
SELECTCOLUMNS(Customer, "Country, State", [Country]&", "&[State])
```

Vea que devuelve lo siguiente:

 Expandir tabla

País, estado
IND, JK
IND, MH
IND, WB
USA, CA
USA, WA

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SUBSTITUTEWITHINDEX

Artículo • 20/10/2023

Devuelve una tabla que representa una semicombinación izquierda de las dos tablas proporcionadas como argumentos. La semicombinación se realiza mediante columnas comunes y se determina por los nombres de columna comunes y el tipo de datos común. Las columnas que se van a combinar se reemplazan por una sola columna en la tabla devuelta, que es de tipo entero y contiene un índice. El índice es una referencia a la tabla de combinación derecha según un criterio de ordenación especificado.

Las columnas de la tabla derecha o segunda proporcionada que no existen en la tabla izquierda o primera ofrecida no se incluyen en la tabla devuelta y no se usan para combinar.

El índice comienza en 0 (basado en 0) y se incrementa en uno por cada fila adicional de la tabla de combinación derecha o segunda proporcionada. El índice se basa en el criterio de ordenación especificado para la tabla de combinación derecha o segunda.

Sintaxis

DAX

```
SUBSTITUTEWITHINDEX(<table>, <indexColumnName>, <indexColumnsTable>, [<orderBy_expression>, [<order>][, <orderBy_expression>, [<order>]]...])
```

Parámetros

Término	Definición
tabla	Tabla que se va a filtrar mediante una semicombinación izquierda con la tabla especificada como tercer argumento (indexColumnsTable). Esta es la tabla del lado izquierdo de la semicombinación izquierda, por lo que la tabla devuelta incluye las mismas columnas que esta tabla, salvo que todas las columnas comunes de las dos tablas se reemplazan por una sola columna de índice en la tabla devuelta.
indexColumnName	Cadena que especifica el nombre de la columna de índice que reemplaza todas las columnas comunes de las dos tablas proporcionadas como argumentos para esta función.
indexColumnsTable	Segunda tabla de la semicombinación izquierda. Esta es la tabla del lado derecho de la semicombinación izquierda. La función solo devuelve los valores presentes en esta tabla. Además, las columnas de esta tabla

Término	Definición
	(basadas en nombres de columna) se reemplazan por una sola columna de índice en la tabla devuelta por esta función.
orderBy_expression	Cualquier expresión DAX donde el valor de resultado se use para especificar el criterio de ordenación deseado de la tabla indexColumnsTable para generar valores de índice correctos. El criterio de ordenación especificado para la tabla indexColumnsTable define el índice de cada fila de la tabla y ese índice se usa en la tabla devuelta para representar combinaciones de valores de indexColumnsTable tal y como aparecen en la tabla proporcionada como primer argumento para esta función.
orden	<p>(Opcional) Valor que especifica cómo ordenar valores orderBy_expression, de forma ascendente o descendente:</p> <p>Valor: Desc. Valor alternativo: 0(cero)/FALSE. Ordena de forma descendente los valores de orderBy_expression. Este es el valor predeterminado si se omite el parámetro order.</p> <p>Valor: ASC. Valor alternativo: 1/TRUE. Clasifica en orden ascendente de orderBy_expression.</p>

Valor devuelto

Tabla que incluye solo los valores presentes en la tabla indexColumnsTable y que tiene una columna de índice en lugar de todas las columnas presentes (por nombre) en la tabla indexColumnsTable.

Observaciones

- Esta función no garantiza ningún criterio de ordenación del resultado.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

SUMMARIZE

Artículo • 20/10/2023

Devuelve una tabla de resumen de los totales solicitados en un conjunto de grupos.

Sintaxis

DAX

```
SUMMARIZE (<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, <name>, <expression>]...)
```

Parámetros

Término	Definición
tabla	Cualquier expresión DAX que devuelve una tabla de datos.
groupBy_ColumnName	(Opcional) Nombre completo de una columna existente usada para crear grupos de resumen basados en los valores que incluye. Este parámetro no puede ser una expresión.
name	Nombre dado a una columna de total o de resumen, entre comillas dobles.
expression	Cualquier expresión DAX que devuelve un único valor escalar, donde la expresión se va a evaluar varias veces (para cada fila o contexto).

Valor devuelto

Tabla con las columnas seleccionadas de los argumentos *groupBy_columnName* y las columnas resumidas diseñadas por los argumentos de nombre.

Notas

- Cada columna para la que se define un nombre debe tener una expresión correspondiente; de lo contrario, se devuelve un error. El primer argumento, name, define el nombre de la columna en los resultados. El segundo argumento, expression, define el cálculo realizado para obtener el valor de cada fila de esa columna.
- groupBy_columnName debe estar en *table* o en una tabla relacionada con *table*.
- Cada nombre debe ir entre comillas dobles.
- La función agrupa un conjunto de filas seleccionado en un conjunto de filas de resumen de los valores de una o más columnas groupBy_columnName. Se devuelve una fila de cada grupo.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se devuelve un resumen de las ventas del revendedor agrupadas por año natural y nombre de la categoría de producto. Esta tabla de resultados permite realizar análisis de las ventas del revendedor por año y por categoría de producto.

DAX

```
SUMMARIZE(ResellerSales_USD  
    , DateTime[CalendarYear]  
    , ProductCategory[ProductCategoryName]  
    , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])  
    , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])  
)
```

En la tabla siguiente se muestra una vista previa de los datos tal como la recibiría cualquier función en espera de recibir una tabla:

DateTime[CalendarYear]	ProductCategory[ProductName]	[Sales Amount (USD)]	[Discount Amount (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	Accessories	153299,924	865,5945
2005	Accessories	18594,4782	4,293
2006	Accessories	86612,7463	1061,4872
2007	Accessories	275794,8403	4756,6546

Con ROLLUP

La incorporación de la sintaxis **ROLLUP** modifica el comportamiento de la función SUMMARIZE al agregar filas de resumen al resultado de las columnas `groupBy_columnName`. **ROLLUP** solo se puede usar en una expresión SUMMARIZE.

Ejemplo

En el ejemplo siguiente se agregan filas de resumen a las columnas Group-By de la llamada de función SUMMARIZE:

DAX

```
SUMMARIZE(ResellerSales_USD
    , ROLLUP( DateTime[CalendarYear], ProductCategory[ProductName])
    , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
    , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

Devuelve la tabla siguiente:

DateTime[CalendarYear]	ProductCategory[ProductName]	[Sales Amount (USD)]	[Discount Amount (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

DateTime[CalendarYear]	ProductCategory[ProductName]	[Sales Amount (USD)]	[Discount Amount (USD)]
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	Accessories	153299,924	865,5945
2005	Accessories	18594,4782	4,293
2006	Accessories	86612,7463	1061,4872
2007	Accessories	275794,8403	4756,6546
2008		15496115,89	41224,3038
2005		7582953,67	4326,4144
2006		22871907,85	184419,1335
2007		30543780,84	297538,0745
		76494758,25	527507,9262

Con ROLLUPGROUP

La incorporación de **ROLLUPGROUP** dentro de una sintaxis **ROLLUP** se puede usar para evitar subtotalés parciales en filas de resumen. **ROLLUPGROUP** solo se puede usar en una expresión **ROLLUP**, **ROLLUPADDISSTOTAL** o **ROLLUPISSSTOTAL**.

Ejemplo

En el ejemplo siguiente se muestra solo el total general de todos los años y categorías sin el subtotal de cada año con todas las categorías:

```
DAX
SUMMARIZE(ResellerSales_USD
    , ROLLUP(ROLLUPGROUP( DateTime[CalendarYear], ProductCategory[ProductName]))
    , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
    , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

Devuelve la tabla siguiente:

DateTime[CalendarYear]	ProductCategory[ProductName]	[Sales Amount (USD)]	[Discount Amount (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

DateTime[CalendarYear]	ProductCategory[ProductName]	[Sales Amount (USD)]	[Discount Amount (USD)]
2005	Clothing	31851,1628	90,9593
2006	Clothing	455730,9729	4233,039
2007	Clothing	815853,2868	12489,3835
2008	Accessories	153299,924	865,5945
2005	Accessories	18594,4782	4,293
2006	Accessories	86612,7463	1061,4872
2007	Accessories	275794,8403	4756,6546
		76494758,25	527507,9262

Con ISSUBTOTAL

Con [ISSUBTOTAL](#), se puede crear otra columna en la expresión SUMMARIZE que devuelve "true" si la fila contiene valores de subtotal para la columna especificada como argumento para [ISSUBTOTAL](#); de lo contrario, devuelve "false". [ISSUBTOTAL](#) solo se puede usar en una expresión SUMMARIZE.

Ejemplo

En el ejemplo siguiente se genera una columna [ISSUBTOTAL](#) para cada una de las columnas [ROLLUP](#) de la llamada de función SUMMARIZE especificada:

DAX

```
SUMMARIZE(ResellerSales_USD
    , ROLLUP( DateTime[CalendarYear], ProductCategory[ProductName])
    , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
    , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
    , "Is Sub Total for DateTime[CalendarYear]", ISSUBTOTAL(DateTime[CalendarYear])
    , "Is Sub Total for ProductCategoryName", ISSUBTOTAL(ProductCategory[ProductName]))
)
```

Devuelve la tabla siguiente:

[Is Sub Total for DateTime[CalendarYear]]	[Is Sub Total for ProductCategoryName]	DateTime[CalendarYear]	ProductCategory[ProductName]	[Sales Amount (USD)]	[Discount Amount (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
false	false	2008	Clothing	366507,844	4151,1235
false	false	2005	Clothing	31851,1628	90,9593
false	false	2006	Clothing	455730,9729	4233,039

[Is Sub Total for DateTimeCalendarYear]	[Is Sub Total for ProductCategoryName]	DateTime[CalendarYear]	ProductCategory[ProductCategoryName]	[Sales Amount (USD)]	[Discount Amount (USD)]
false	false	2007	Clothing	815853,2868	12489,3835
false	false	2008	Accessories	153299,924	865,5945
false	false	2005	Accessories	18594,4782	4,293
false	false	2006	Accessories	86612,7463	1061,4872
false	false	2007	Accessories	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

Vea también

[SUMMARIZECOLUMNS](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

SUMMARIZECOLUMNS

Artículo • 20/07/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✖ Cálculo visual

Devuelve una tabla de resumen sobre un conjunto de grupos.

Sintaxis

DAX

```
SUMMARIZECOLUMNS( <groupBy_columnName> [, <groupBy_columnName>]...,  
[<filterTable>]...[, <name>, <expression>]...)
```

Parámetros

[+] Expandir tabla

Término	Definición
groupBy_columnName	Referencia de columna completa (Table[Column]) a una tabla base para la que se incluyen los valores diferenciados en la tabla devuelta. En cada columna groupBy_columnName se realiza una operación de combinación cruzada (tablas diferentes) o de existencia automática (la misma tabla) con las columnas siguientes especificadas.
filterTable	Expresión de tabla que se agrega al contexto de filtro de todas las columnas especificadas como argumentos groupBy_columnName. Los valores presentes en la tabla de filtros se usan para filtrar antes de que se realice la operación de combinación cruzada o de existencia automática.
name	Cadena que representa el nombre de columna que se va a usar para la expresión siguiente especificada.
expression	Cualquier expresión DAX que devuelva un único valor (no una tabla).

Valor devuelto

Una tabla que incluye combinaciones de valores de las columnas proporcionadas en función de la agrupación especificada. Solo se incluyen en la tabla devuelta las filas para las que al menos una de las expresiones proporcionadas devuelve un valor que no está

en blanco. Si todas las expresiones se evalúan como BLANK/NULL para una fila, esa fila no se incluye en la tabla devuelta.

Notas

- Esta función no garantiza ningún criterio de ordenación para los resultados.
- No se puede especificar una columna más de una vez en el parámetro groupBy_columnName. Por ejemplo, la fórmula siguiente no es válida.

```
SUMMARIZECOLUMNS( Sales[StoreId], Sales[StoreId] )
```
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Contexto de filtro

Considere la consulta siguiente:

DAX

```
SUMMARIZECOLUMNS (
    'Sales Territory'[Category],
    FILTER('Customer', 'Customer' [First Name] = "Alicia")
)
```

En esta consulta, sin una medida, las columnas groupBy no contienen ninguna columna de la expresión FILTER (por ejemplo, de la tabla Customer). El filtro no se aplica a las columnas groupBy. Las tablas Sales Territory y Customer pueden estar indirectamente relacionadas a través de la tabla de hechos Reseller sales. Como no están relacionadas directamente, la expresión de filtro es no-op y las columnas groupBy no se ven afectadas.

Pero con esta consulta:

DAX

```
SUMMARIZECOLUMNS (
    'Sales Territory'[Category], 'Customer' [Education],
    FILTER('Customer', 'Customer'[First Name] = "Alicia")
)
```

Las columnas groupBy contienen una columna que se ve afectada por el filtro y dicho filtro se aplica a los resultados de groupBy.

Con IGNORE

La sintaxis **IGNORE** se puede usar para modificar el comportamiento de la función **SUMMARIZECOLUMNS** omitiendo expresiones específicas de la evaluación **BLANK/NULL**. Las filas para las que todas las expresiones que no usen **IGNORE** devuelvan **BLANK/NULL** se excluirán, aunque las expresiones que usen **IGNORE** se evalúen como **BLANK/NULL**. **IGNORE** solo se puede usar en una expresión **SUMMARIZECOLUMNS**.

Ejemplo

DAX

```
SUMMARIZECOLUMNS(  
    Sales[CustomerId], "Total Qty",  
    IGNORE( SUM( Sales[Qty] ) ),  
    "BlankIfTotalQtyIsNot3", IF( SUM( Sales[Qty] )=3, 3 )  
)
```

Acumula la columna **Sales[CustomerId]** y crea un subtotal para todos los clientes de la agrupación especificada. Sin **IGNORE**, el resultado es el siguiente:

 Expandir tabla

CustomerId	Cantidad total	BlankIfTotalQtyIsNot3
A	5	
B	3	3
C	3	3

Con **IGNORE**:

 Expandir tabla

CustomerId	Cantidad total	BlankIfTotalQtyIsNot3
B	3	3
C	3	3

Todas las expresiones omitidas:

DAX

```

SUMMARIZECOLUMNS(
    Sales[CustomerId], "Blank",
    IGNORE( BLANK(), "BlankIfTotalQtyIsNot5",
    IGNORE( IF( SUM( Sales[Qty] )=5, 5 ) )
)

```

Aunque ambas expresiones devuelven un valor en blanco para algunas filas, se incluyen porque no hay expresiones no omitidas que devuelven valores en blanco.

 Expandir tabla

CustomerId	En blanco	BlankIfTotalQtyIsNot5
A		5
B		
C		

Con NONVISUAL

La función **NONVISUAL** marca un filtro de valor en la función SUMMARIZECOLUMNS como que no afecta a los valores de medida, sino que solo se aplica a las columnas groupBy. **NONVISUAL** solo se puede usar en una expresión SUMMARIZECOLUMNS.

Ejemplo

DAX

```

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]

```

Devuelve un resultado en el que [Visual Total Sales] es el total entre todos los años:

[+] Expandir tabla

DimDate[CalendarYear]	[Sales]	[Visual Total Sales]
2007	9 791 060,30	29 358 677,22
2008	9 770 899,74	29 358 677,22

En cambio, la misma consulta *sin* la función NONVISUAL:

DAX

```
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]
```

Devuelve un resultado en el que [Visual Total Sales] es el total entre los dos años seleccionados:

[+] Expandir tabla

DimDate[CalendarYear]	[Sales]	[Visual Total Sales]
2007	9 791 060,30	19 561 960,04
2008	9 770 899,74	19 561 960,04

Con ROLLUPADDISSUBTOTAL

La suma de la sintaxis [ROLLUPADDISSUBTOTAL](#) modifica el comportamiento de la función SUMMARIZECOLUMNS al agregar filas de acumulación o subtotal al resultado en función de las columnas de groupBy_columnName. [ROLLUPADDISSUBTOTAL](#) solo se puede usar en una expresión SUMMARIZECOLUMNS.

Ejemplo con un solo subtotal

DAX

```

DEFINE
VAR vCategoryFilter =
    TREATAS({ "Accessories", "Clothing"}, Product[Category])
VAR vSubcategoryFilter =
    TREATAS({ "Bike Racks", "Mountain Bikes"}, Product[Subcategory])
EVALUATE
SUMMARIZECOLUMNS
(
    ROLLUPADDISSTOTAL
    (
        Product[Category], "IsCategorySubtotal", vCategoryFilter,
        Product[Subcategory], "IsSubcategorySubtotal", vSubcategoryFilter
    ),
    "Total Qty", SUM(Sales[Qty])
)
ORDER BY
[IsCategorySubtotal] DESC, [Category],
[IsSubcategorySubtotal] DESC, [Subcategory]

```

Devuelve la tabla siguiente:

Expandir tabla

Category	Subcategoría	IsCategorySubtotal	IsSubcategorySubtotal	Cantidad total
		True	True	60 398
Accessories		False	True	36 092
Accessories	Marcos de bicicletas	False	False	328
Bikes	Bicicletas de montaña	False	False	4970
Clothing		False	True	9101

Ejemplo con varios subtotales

DAX

```

SUMMARIZECOLUMNS (
    Regions[State], ROLLUPADDISSTOTAL ( Sales[CustomerId],
    "IsCustomerSubtotal" ),
    ROLLUPADDISSTOTAL ( Sales[Date], "IsDateSubtotal" ), "Total Qty", SUM(

```

```
Sales[Qty] )  
)
```

Las ventas se agrupan por estado, por cliente, por fecha, con subtotales de 1. Ventas por estado, por fecha 2. Ventas por estado, por cliente 3. Se acumula tanto en el cliente como en la fecha, lo que conduce a las ventas por estado.

Devuelve la tabla siguiente:

 Expandir tabla

CustomerID	IsCustomerSubtotal	State	Cantidad total	Date	IsDateSubtotal
A	false	WA	5	10/07/2014	
B	false	WA	1	10/07/2014	
B	false	WA	2	11/07/2014	
C	false	O BIEN	2	10/07/2014	
C	false	O BIEN	1	11/07/2014	
	true	WA	6	10/07/2014	
	true	WA	2	11/07/2014	
	true	O BIEN	2	10/07/2014	
	true	O BIEN	1	11/07/2014	
A	false	WA	5		true
B	false	WA	3		true
C	false	O BIEN	3		TRUE
	VERDADERO	WA	8		TRUE
	VERDADERO	O BIEN	3		true

Con ROLLUPGROUP

Al igual que con la función [SUMMARIZE](#), [ROLLUPGROUP](#) se puede usar junto con [ROLLUPADDISSTOTAL](#) para especificar qué grupos de resumen o granularidades (subtotales) se incluirán, lo que reduce el número de filas de subtotal devueltas. [ROLLUPGROUP](#) solo se puede usar en una expresión SUMMARIZECOLUMNS o [SUMMARIZE](#).

Ejemplo con varios subtotales

DAX

```
SUMMARIZECOLUMNS(  
    ROLLUPADDISSTOTAL( Sales[CustomerId], "IsCustomerSubtotal" ),  
    ROLLUPADDISSTOTAL(ROLLUPGROUP(Regions[City], Regions[State]),  
    "IsCityStateSubtotal"), "Total Qty", SUM( Sales[Qty] )  
)
```

Todavía se agrupan por ciudad y estado, pero se acumulan cuando notificar un subtotal devuelve la tabla siguiente:

 Expandir tabla

State	CustomerId	IsCustomerSubtotal	Cantidad total	City (Ciudad)	IsCityStateSubtotal
WA	A	false	2	Bellevue	false
WA	B	false	2	Bellevue	false
WA	A	false	3	Redmond	false
WA	B	false	1	Redmond	false
O BIEN	C	false	3	Portland	false
WA		true	4	Bellevue	false
WA		true	4	Redmond	false
O BIEN		true	3	Portland	false
	A	false	5		FALSO
	B	false	3		true
	C	false	3		TRUE
	TRUE		11		VERDADERO

SummarizeColumns contextual

Fondo

Hasta febrero de 2023, SUMMARIZECOLUMNS no admitía la evaluación en una transición de contexto. En los productos publicados antes de ese mes, esta limitación hacía que SUMMARIZECOLUMNS no fuera útil en la mayoría de las medidas: no era posible llamar SUMMARIZECOLUMNS a una medida en ningún caso de transición de contexto, incluidas otras declaraciones SUMMARIZECOLUMNS.

A partir de febrero de 2023, la transición de contexto se admite en algunos escenarios, pero no en todas las condiciones. Los casos admitidos y restringidos son los siguientes:

 Expandir tabla

Tipo SummarizeColumns	Filtro externo con una sola columna	Filtro externo con más de una columna	Columnas AgruparPor externas
SummarizeColumns solo con AgruparPor	Aceptar	Aceptar	Aceptar
SummarizeColumns con Filtros/Medidas	OK (CORRECTO)	ERROR	ERROR

Desde junio de 2024, habilitamos SummarizeColumns contextuales que permiten que SummarizeColumns se evalúe en cualquier transición de contexto. SummarizeColumns en medida es ahora totalmente compatible:

 Expandir tabla

Tipo SummarizeColumns	Filtro externo con una sola columna	Filtro externo con más de una columna	Columnas AgruparPor externas
SummarizeColumns solo con AgruparPor	Aceptar	Aceptar	Aceptar
SummarizeColumns con Filtros/Medidas	Aceptar	Aceptar	Aceptar

Sin embargo, esta actualización también incluye cambios en el comportamiento de SummarizeColumns, que pueden modificar los resultados de las expresiones existentes:

Semántica de SelfValue para filtros externos

Presentamos un concepto semántico denominado SelfValue, que modifica cómo interactúan los filtros de las tablas externas con las columnas AgruparPor en SummarizeColumns. Este cambio no permite que los filtros de una tabla diferente

afecten a las columnas AgruparPor, incluso si las tablas están relacionadas a través de una relación de filtrar por. Un ejemplo que ilustra el impacto de este cambio implica la siguiente expresión:

```
CalculateTable(
    SummarizeColumns(
        'Reseller Sales'[ResellerKey],
        'Reseller Sales'[ProductKey]
    ),
    Treatas({(229)}, 'Product'[Product Key])
)
```

Antes de esta actualización, el filtro TreatAs se aplicaría a la operación AgruparPor dentro de SummarizeColumns, aprovechando la relación entre 'Producto'[Product Key] y 'Ventas a revendedores'[ProductKey]. Por lo tanto, los resultados de la consulta solo incluirían filas en las que "Ventas a revendedores" [ProductKey] es igual a 229. Sin embargo, después de la actualización, las columnas AgruparPor dentro de SummarizeColumns ya no se filtrarán por columnas de tablas externas, incluso si existe una relación entre ellas. Por lo tanto, en el ejemplo anterior, la columna AgruparPor "Ventas a revendedores" [ProductKey] no se filtrará por la columna "Producto" [ProductKey]. Como resultado, la consulta incluirá filas en las que "Ventas a revendedores" [ProductKey] no es igual a 229.

Si prefiere conservar el comportamiento anterior, puede volver a escribir la expresión mediante AddColumns o SelectColumns en lugar de SummarizeColumns, como se muestra a continuación:

```
CalculateTable(
    Filter(
        SelectColumns(
            'Reseller Sales',
            "ResellerKey",
            [ResellerKey],
            "ProductKey",
            [ProductKey]
        ),
        And(Not IsBlank([ResellerKey]), Not IsBlank([ProductKey]))
    ),
    Treatas({(229)}, 'Product'[Product Key])
)
```

Esta expresión reescrita conserva la semántica original en la que la operación AgruparPor no se ve afectada por la restricción SelfValue introducida por la actualización.

Validación de filas para columnas AgruparPor totalmente cubiertas por Treatas

Antes de esta actualización, dentro de una función SummarizeColumns, si todas las columnas AgruparPor de una tabla específica estaban totalmente cubiertas por un único filtro Treatas de esa misma tabla, como se muestra a continuación:

```
SummarizeColumns(
    Geography[Country],
    Geography[State],
    Treatas(
        {"United States", "Alberta"}),
    Geography[Country],
    Geography[State]
)
)
```

El resultado de la consulta anterior incluiría las filas especificadas en el filtro Treatas, independientemente de si eran válidas o no. Por ejemplo, el resultado sería una tabla de una sola fila ("Estados Unidos", "Alberta"), incluso si no existía dicha fila con [País] = "Estados Unidos" y [Estado] = "Alberta" existía en la tabla "Geografía".

Se ha conocido este problema y se ha solucionado mediante la actualización. Después de la actualización, estas filas no válidas se filtrarán y solo se devolverán filas válidas de la tabla AgruparPor. Por lo tanto, el resultado de la consulta anterior estaría vacío, ya que no hay filas válidas que coincidan con los valores [Country] y [State] especificados en la tabla "Geografía".

No permitir keepfilters/overridedefilters mixtos en la misma tabla o clúster

La actualización reciente ha introducido una restricción temporal que desencadena un mensaje de error que indica:

```
"SummarizeColumns filters with keepfilters behavior and overridedefilters behavior are mixed within one cluster, which is not allowed. Consider adding
```

```
keepfilters() to all filters of summarizecolumns."
```

Este error se produce cuando los filtros normales (que invalidan los filtros existentes) y los filtros con KeepFilters especificados están presentes en la misma tabla o clúster. Por ejemplo:

```
Evaluate CalculateTable(
    SummarizeColumns(
        Product[Color],
        KeepFilters(
            TreatAs(
                {{ "Washington" }}
                , Geography[State]
            )
        ),
        TreatAs(
            {"United States"}, {"Canada"})
            , Geography[Country]
        )
    )
    ,TreatAs({{"Alberta"}}, Geography[State])
    ,TreatAs({{"Canada"}}, Geography[Country])
)
```

En la expresión anterior, hay dos filtros en la tabla "Geografía": uno con KeepFilters especificado y otro sin él. Estos filtros se superponen con filtros externos en columnas diferentes. Actualmente, no se permite esta configuración porque internamente, los dos filtros se agrupan en uno y el sistema no puede determinar el comportamiento correcto de invalidación del filtro en clúster en estos casos.

Tenga en cuenta que esta restricción es temporal. Estamos desarrollando soluciones activamente para quitar esta limitación en futuras actualizaciones. Si se encuentra con este error, le recomendamos que ajuste los filtros dentro de SummarizeColumns agregando o eliminando KeepFilters según sea necesario para garantizar un comportamiento de invalidación coherente en cada tabla.

Contenido relacionado

[SUMMARIZE](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Constructor de tablas

Artículo • 20/10/2023

Devuelve una tabla de una o varias columnas.

Sintaxis

DAX

```
{ <scalarExpr1>, <scalarExpr2>, ... }  
{ ( <scalarExpr1>, <scalarExpr2>, ... ), ( <scalarExpr1>, <scalarExpr2>, ... ),  
... }
```

Parámetros

Término	Definición
scalarExprN	Cualquier expresión DAX que devuelve un valor escalar.

Valor devuelto

Tabla de una o varias columnas. Si solo hay una columna, el nombre de la columna es Value. Si hay N columnas en las que $N > 1$, los nombres de las columnas de izquierda a derecha son Value1, Value2, ..., ValueN.

Notas

- La primera sintaxis devuelve una tabla de una sola columna. La segunda sintaxis devuelve una tabla de una o varias columnas.
- El número de expresiones escalares debe ser el mismo en todas las filas.
- Si los tipos de datos de los valores de una columna son diferentes en filas distintas, todos los valores se convierten a un tipo de datos común.

Ejemplo 1

Las consultas DAX siguientes:

```
DAX
```

```
EVALUATE { 1, 2, 3 }
```

y

```
DAX
```

```
EVALUATE { (1), (2), (3) }
```

Devuelven la siguiente tabla de una sola columna:

[Valor]
1
2
3

Ejemplo 2

La consulta DAX siguiente:

```
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")
}
```

Vea que devuelve lo siguiente:

[Value1]	[Value2]	[Value3]	[Value4]
1.5	1/1/2017	199,99	A
2.5	1/2/2017	249,99	B
3,5	1/3/2017	299,99	C

Ejemplo 3

La consulta DAX siguiente:

DAX

```
EVALUATE { 1, DATE(2017, 1, 1), TRUE, "A" }
```

Devuelve la siguiente tabla de una sola columna de tipo de datos de cadena:

[Valor]
1
1/1/2017
VERDADERO
A

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

TOPN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve las N filas superiores de la tabla especificada.

Sintaxis

DAX

```
TOPN(<N_Value>, <Table>, <OrderBy_Expression>, [<Order>[, <OrderBy_Expression>, [<Order>]]...])
```

Parámetros

Expandir tabla

Parámetro	Definición
N_Value	Número de filas que se va a devolver. Cualquier expresión DAX que devuelve un valor escalar, donde la expresión se va a evaluar varias veces (para cada fila o contexto). Vea Notas para comprender mejor cuándo el número de filas devueltas podría ser mayor que <i>n_value</i> .
Tabla	Cualquier expresión DAX que devuelve una tabla de datos de donde extraer las "n" filas superiores. Vea Notas para comprender mejor cuándo se devuelve una tabla vacía.
OrderBy_Expression	(Opcional) Cualquier expresión DAX donde se use el valor de resultado para ordenar la tabla y evaluar para cada fila de <i>tabla</i> .
Orden	(Opcional) Valor que especifica cómo ordenar valores <i>OrderBy_Expression</i> : <ul style="list-style-type: none">- 0 (cero) o FALSE. Ordena de forma descendente los valores de <i>Order</i>. Es el valor predeterminado si se omite el parámetro <i>Order</i>.- 1 o TRUE. Clasifica en orden ascendente los valores de <i>OrderBy</i>.

Valor devuelto

Una tabla con las N filas superiores de *Table* o una tabla vacía si *N_Value* es 0 (cero) o menos. Las filas no se ordenan en ningún orden determinado.

Comentarios

- Si hay una asociación en los valores de *Order_By*, en la enésima fila de la tabla, se devuelven todas las filas asociadas. Después, cuando haya asociaciones en la enésima fila, es posible que la función devuelva más de n filas.
- Si N_Value es 0 (cero) o menos, TOPN devuelve una tabla vacía.
- TOPN no garantiza ningún criterio de ordenación para los resultados.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula de medida siguiente devuelve los 10 principales productos vendidos por cantidad de ventas.

```
DAX

= SUMX(
    TOPN(
        10,
        SUMMARIZE(
            InternetSales,
            InternetSales[ProductKey],
            "TotalSales", SUM(InternetSales[SalesAmount])
        ),
        [TotalSales], DESC
    ),
    [TotalSales]
)
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TREATAS

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Aplica el resultado de una expresión de tabla como filtros a las columnas de una tabla no relacionada.

Sintaxis

DAX

```
TREATAS(table_expression, <column>[, <column>[, <column>[,...]]]} )
```

Parámetros

Expandir tabla

Término	Definición
table_expression	Expresión que da lugar a una tabla.
columna	Una o más columnas existentes. No puede ser una expresión.

Valor devuelto

Tabla que contiene todas las filas de las columnas que también están en table_expression.

Notas

- El número de columnas especificado debe coincidir con el número de columnas de la expresión de tabla y estar en el mismo orden.
- Si un valor devuelto en la expresión de tabla no existe en la columna, se omite. Por ejemplo, TREATAS({"Red", "Green", "Yellow"}, DimProduct[Color]) establece un filtro en la columna DimProduct[Color] con tres valores "Red", "Green" y "Yellow". Si "Yellow" no existe en DimProduct[Color], los valores de filtro efectivos serían "Red" y "Green".

- Se recomienda para cuando no existe una relación entre las tablas. Si tiene varias relaciones entre las tablas implicadas, valore la posibilidad de utilizar [USERELATIONSHIP](#) en su lugar.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente, el modelo contiene dos tablas de productos no relacionadas. Si un usuario aplica un filtro a DimProduct1[ProductCategory] seleccionando Bikes, Seats, Tires, se aplica el mismo filtro, Bikes, Seats, Tires, a DimProduct2[ProductCategory].

DAX

```
CALCULATE(  
    SUM(Sales[Amount]),  
    TREATAS(VALUES(DimProduct1[ProductCategory]), DimProduct2[ProductCategory])  
)
```

Contenido relacionado

[INTERSECT](#)

[FILTER](#)

[USERELATIONSHIP](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

UNION

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Crea una tabla de unión (combinación) a partir de un par de tablas.

Sintaxis

DAX

```
UNION(<table_expression1>, <table_expression2> [,<table_expression>]...)
```

Parámetros

Expandir tabla

Término	Definición
table_expression	Cualquier expresión DAX que devuelve una tabla.

Valor devuelto

Tabla que contiene todas las filas de cada una de las dos expresiones de tabla.

Notas

- Las dos tablas deben tener el mismo número de columnas.
- Las columnas se combinan según la posición que tienen en sus respectivas tablas.
- Los nombres de columna de la tabla devuelta coincidirán con los de table_expression1.
- Las filas duplicadas se conservan.
- La tabla devuelta tiene linaje siempre que sea posible. Por ejemplo, si la primera columna de cada table_expression tiene linaje en la misma columna base C1 del modelo, la primera columna del resultado de UNION tendrá linaje en C1. En

cambio, si las columnas combinadas tienen linaje en otras columnas base, o bien si hay una columna de extensión, la columna resultante en UNION no tendrá linaje.

- Si los tipos de datos difieren, el tipo de datos resultante se determina en función de las reglas para la coerción de tipos de datos.
- La tabla devuelta no contendrá columnas de tablas relacionadas.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La expresión siguiente crea una unión al combinar la tabla USAInventory y la tabla INDInventory en una sola tabla:

DAX
<code>UNION(UsaInventory, IndInventory)</code>

USAInventory

[\[+\] Expandir tabla](#)

País/región	State	Count	Total
EE. UU.	CA	5	500
EE. UU.	WA	10	900

INDInventory

[\[+\] Expandir tabla](#)

País/región	State	Count	Total
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900

Tabla devuelta

[\[+\] Expandir tabla](#)

País/región	State	Count	Total
EE. UU.	CA	5	500
EE. UU.	WA	10	900
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

VALUES

Artículo • 20/10/2023

Si el parámetro de entrada es un nombre de columna, devuelve una tabla de una columna que contiene los valores distintos de la columna especificada. Los valores duplicados se quitan y solo se devuelven los valores únicos. Se puede agregar un valor BLANK. Si el parámetro de entrada es un nombre de tabla, devuelve las filas de la tabla especificada. Las filas duplicadas se conservan. Se puede agregar una fila BLANK.

ⓘ Nota

Esta función no se puede usar para devolver valores en una celda o columna de una hoja de cálculo, sino que se usa como una función intermedia, anidada en una fórmula, para obtener una lista de valores distintos que se pueden contar o usar para filtrar o sumar otros valores.

Sintaxis

DAX

`VALUES(<TableNameOrColumnName>)`

Parámetros

Término	Definición
TableName o ColumnName	Columna de la que se van a devolver valores únicos o tabla de la que se van a devolver filas.

Valor devuelto

Si el parámetro de entrada es un nombre de columna, una tabla de una sola columna. Si el parámetro de entrada es un nombre de tabla, se devuelve una tabla de las mismas columnas.

Notas

- Cuando se usa la función VALUES en un contexto que se ha filtrado, los valores únicos que devuelve VALUES se ven afectados por el filtro. Por ejemplo, si filtra por región y devuelve una lista de los valores de ciudad, la lista solo incluye las ciudades de las regiones permitidas por el filtro. Para devolver todas las ciudades, independientemente de los filtros existentes, debe usar la función ALL para quitar los filtros de la tabla. En el segundo ejemplo se muestra el uso de ALL con VALUES.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).
- Para conocer los procedimientos recomendados al usar VALUES, consulte [Uso de SELECTEDVALUE en lugar de VALUES](#).

Funciones relacionadas

En la mayoría de los escenarios, si el argumento es un nombre de columna, los resultados de la función VALUES son idénticos a los de la función **DISTINCT**. Ambas funciones quitan duplicados y devuelven una lista de los valores posibles de la columna especificada. Pero la función VALUES también puede devolver un valor en blanco. Este valor en blanco es útil en aquellos casos en los que se buscan valores distintos de una tabla relacionada, pero falta un valor usado en la relación de una tabla. En la terminología de bases de datos, esto se denomina una infracción de integridad referencial. Tales discrepancias en los datos pueden producirse cuando se actualiza una tabla y la tabla relacionada no.

Si el argumento es un nombre de tabla, el resultado de la función VALUES devuelve todas las filas de la tabla especificada más una fila en blanco, si hay una infracción de integridad referencial. La función DISTINCT quita las filas duplicadas y devuelve las filas únicas de la tabla especificada.

Nota

La función DISTINCT permite que un nombre de columna o cualquier expresión de tabla válida sea su argumento, pero la función VALUES solo acepta un nombre de columna o un nombre de tabla como argumento.

En la tabla siguiente se resume la discrepancia entre los datos que se puede producir en dos tablas relacionadas cuando no se preserva la integridad referencial.

Tabla MyOrders	Tabla MySales
1 de junio	Ventas del 1 de junio

Tabla MyOrders

(no se ha especificado ninguna fecha de pedido)

Tabla MySales junio

Ventas del 3 de junio

Si usa la función DISTINCT para devolver una lista de fechas, solo se devolverán dos fechas. Pero si usa la función VALUES, la función devuelve las dos fechas más un miembro en blanco adicional. Además, cualquier fila de la tabla MySales que no tenga una fecha coincidente en la tabla MyOrders se "empareja" con este miembro desconocido.

Ejemplo

La fórmula siguiente cuenta el número de facturas únicas (pedidos de ventas) y genera los siguientes resultados cuando se usa en un informe que incluye los nombres de categoría de producto:

DAX

```
= COUNTROWS(VALUES('InternetSales_USD'[SalesOrderNumber]))
```

Devuelve

Etiquetas de fila	Recuento de facturas
Accesorios	18 208
Bicicletas	15 205
Ropa	7 461
Total general	27 659

Vea también

[FILTER \(función\)](#)

[Función COUNTROWS](#)

[Funciones de filtro](#)

Comentarios

¿Le ha resultado útil esta página?

Yes

No

Funciones de texto

Artículo • 20/10/2023

Las expresiones de análisis de datos (DAX) incluyen un conjunto de funciones de texto basadas en la biblioteca de funciones de cadena de Excel, pero que se han modificado para trabajar con tablas y columnas en modelos tabulares. En esta sección se describen las funciones de texto que hay disponibles en el lenguaje DAX.

En esta categoría

Función	Descripción
COMBINEVALUES	combina dos o más cadenas de texto en una sola.
CONCATENATE	Combina dos cadenas de texto en una sola.
CONCATENATEX	Concatena el resultado de una expresión evaluada para cada fila de una tabla.
EXACT	Compara dos cadenas de texto y devuelve TRUE si son exactamente iguales, o FALSE en caso contrario.
FIND	Devuelve la primera posición de una cadena de texto en otra.
FIXED	Redondea un número al número de decimales especificado y devuelve el resultado como texto.
FORMAT	Convierte un valor en texto según el formato especificado.
LEFT	Devuelve el número de caracteres especificados desde el principio de una cadena de texto.
LEN	Devuelve el número de caracteres de una cadena de texto.
LOWER	Convierte todas las letras de una cadena de texto a minúsculas.
MID	Devuelve una cadena de caracteres de la mitad de una cadena de texto a partir de una posición inicial y una longitud.
REPLACE	REPLACE reemplaza la parte de una cadena de texto, según el número de caracteres que se especifique, por otra cadena de texto.
REPT	Repite el texto un número determinado de veces.
RIGHT	Devuelve el último carácter o caracteres de una cadena de texto, en función del número de caracteres que especifique.
SEARCH	Devuelve el número del carácter en el que se encuentra por primera vez un carácter o una cadena de texto específicos, de izquierda a derecha.

Función	Descripción
SUBSTITUTE	Reemplaza texto existente por otro nuevo en una cadena de texto.
TRIM	Quita todos los espacios del texto, excepto los espacios individuales entre palabras.
UNICHAR	Devuelve el carácter Unicode al que hace referencia el valor numérico.
UNICODE	Devuelve el código numérico que corresponde al primer carácter de la cadena de texto.
UPPER	Convierte una cadena de texto a letras mayúsculas.
VALUE	Convierte en un número una cadena de texto que representa un número.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

COMBINEVALUES

Artículo • 20/10/2023

combina dos o más cadenas de texto en una sola. El propósito principal de esta función es admitir relaciones de varias columnas en los modelos DirectQuery. Consulte [Comentarios](#) para obtener más detalles.

Sintaxis

DAX

```
COMBINEVALUES(<delimiter>, <expression>, <expression>[ , <expression>]...)
```

Parámetros

Término	Definición
delimiter	Separador que se va a usar durante la concatenación. Debe ser un valor constante.
expresión	Expresión DAX cuyo valor se va a combinar en una sola cadena de texto.

Valor devuelto

Cadena concatenada.

Notas

- La función COMBINEVALUES da por hecho, aunque no valida, que si los valores de entrada son diferentes, las cadenas de salida también son diferentes. En función de esta premisa, cuando se usa COMBINEVALUES para crear columnas calculadas con el fin de generar una relación que combina varias columnas de dos tablas de DirectQuery, se genera una condición de combinación optimizada en el momento de la consulta. Por ejemplo, si los usuarios quieren crear una relación entre Table1(Column1, Column2) y Table2(Column1, Column2), pueden crear dos columnas calculadas, una en cada tabla, como:

DAX

```
Table1[CalcColumn] = COMBINEVALUES(", ", Table1[Column1],
```

```
Table1[Column2])
```

y

```
DAX
```

```
Table2[CalcColumn] = COMBINEVALUES(", ", Table2[Column1],  
Table2[Column2])
```

Y luego crear una relación entre `Table1[CalcColumn]` y `Table2[CalcColumn]`. A diferencia de otras funciones y operadores de DAX, que se traducen literalmente en los operadores y funciones de SQL correspondientes, la relación anterior genera un predicado de combinación de SQL como:

```
DAX
```

```
(Table1.Column1 = Table2.Column1 OR Table1.Column1 IS NULL AND  
Table2.Column1 IS NULL)
```

y

```
DAX
```

```
(Table1.Column2 = Table2.Column2 OR Table1.Column2 IS NULL AND  
Table2.Column2 IS NULL)
```

- El predicado de combinación puede ofrecer potencialmente un rendimiento de consulta mucho mejor que uno que implique operadores y funciones de SQL complejos.
- La función COMBINEVALUES se basa en los usuarios para elegir el delimitador adecuado con el fin de asegurarse de que combinaciones únicas de valores de entrada produzcan cadenas de salida distintas, pero no valida que la premisa sea verdadera. Por ejemplo, si los usuarios eligen `" | "` como delimitador, pero una fila de Table1 tiene `Table1[Column1] = " | "` y `Table2[Column2] = " "`, mientras que una fila de Table2 tiene `Table2[Column1] = " "` y `Table2[Column2] = " | "`, las dos salidas concatenadas generan `" || "`, lo que parece indicar que las dos filas son una coincidencia de la operación de combinación. Las dos filas no se combinan si ambas tablas proceden del mismo origen de DirectQuery, aunque sí lo hacen si ambas tablas se importan.

Ejemplo

La consulta DAX siguiente:

DAX

```
EVALUATE  
DISTINCT (  
    SELECTCOLUMNS ( Date, "Month", COMBINEVALUES ( ", ", [MonthName],  
[CalendarYear] ) )  
)
```

Devuelve la tabla siguiente de una sola columna:

[Mes]
Enero de 2020
Febrero de 2020
Marzo de 2020
Abril de 2020
Mayo de 2020
Junio de 2020
Julio de 2020
Agosto de 2020
Septiembre de 2020
Octubre de 2020
Noviembre de 2020
Diciembre de 2020
Enero de 2021
Enero de 2021
Febrero de 2021
Marzo de 2021
Abrial de 2021
Mayo de 2021
Junio de 2021

[Mes]

Julio de 2021

Agosto de 2021

Septiembre de 2021

Octubre de 2021

Noviembre de 2021

diciembre de 2021

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

CONCATENATE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Combina dos cadenas de texto en una sola.

Sintaxis

DAX

```
CONCATENATE(<text1>, <text2>)
```

Parámetros

Expandir tabla

Término	Definición
text1	La primera cadena de texto que se va a combinar en una sola. La cadena puede incluir texto o números. También puede usar referencias de columna.
text2	La segunda cadena de texto que se va a combinar en una sola. La cadena puede incluir texto o números. También puede usar referencias de columna.

Valor devuelto

Cadena concatenada.

Notas

- La función CONCATENATE combina dos cadenas de texto en una sola. Los elementos combinados pueden ser texto, números, valores booleanos representados como texto o una combinación de esos elementos. También puede usar una referencia de columna si la columna contiene los valores adecuados.
- La función CONCATENATE de DAX acepta solo dos argumentos, mientras que la función CONCATENATE de Excel acepta hasta 255 argumentos. Si necesita concatenar varias columnas, puede crear una serie de cálculos o usar el operador de concatenación (&) para combinarlas en una expresión más sencilla.

- Si quiere usar cadenas de texto directamente, en lugar de usar una referencia de columna, debe incluir cada cadena entre comillas dobles.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo: concatenación de literales

La fórmula de ejemplo crea un valor de cadena mediante la combinación de dos valores de cadena que se proporcionan como argumentos.

DAX

```
= CONCATENATE("Hello ", "World")
```

Ejemplo: concatenación de cadenas en columnas

La fórmula de ejemplo devuelve el nombre completo del cliente tal como aparece en una agenda de teléfonos. Observe cómo se usa una función anidada como segundo argumento. Es una manera de concatenar varias cadenas, si tiene más de dos valores que quiere usar como argumentos.

DAX

```
= CONCATENATE(Customer[LastName], CONCATENATE(", ", Customer[FirstName]))
```

Ejemplo: concatenación condicional de cadenas en columnas

La fórmula de ejemplo crea una columna calculada en la tabla Customer con el nombre completo del cliente como una combinación del nombre, la inicial del segundo nombre y los apellidos. Si no hay ningún segundo nombre, el apellido aparece directamente después del nombre. Si hay un segundo nombre, solo se usa la primera letra del segundo nombre y la letra inicial va seguida de un punto.

DAX

```
= CONCATENATE([FirstName]&" ", CONCATENATE(IF(LEN([MiddleName])>1,
```

```
LEFT([MiddleName],1)&" ", ""), [LastName]))
```

Esta fórmula usa funciones CONCATENATE e IF anidadas, junto con el operador Y comercial (&), para concatenar condicionalmente tres valores de cadena y agregar espacios como separadores.

Ejemplo: concatenación de columnas con tipos de datos distintos

En el ejemplo siguiente se muestra cómo concatenar valores en columnas que tienen tipos de datos diferentes. Si el valor que se va a concatenar es numérico, se convertirá en texto de forma implícita. Si los dos valores son numéricos, se convertirán en texto y se concatenarán como si fueran cadenas.

 Expandir tabla

Descripción del producto	Abreviatura del producto (columna 1 de la clave compuesta)	Número del producto (columna 2 de la clave compuesta)	Nueva columna de clave generada
Bicicleta de montaña	MTN	40	MTN40
Bicicleta de montaña	MTN	42	MTN42

DAX

```
= CONCATENATE('Products'[Product abbreviation], 'Products'[Product number])
```

La función CONCATENATE de DAX acepta solo dos argumentos, mientras que la función CONCATENATE de Excel acepta hasta 255 argumentos. Si necesita agregar más argumentos, puede usar el operador Y comercial (&). Por ejemplo, la fórmula siguiente genera los resultados MTN-40 y MTN-42.

DAX

```
= [Product abbreviation] & "-" & [Product number]
```

Contenido relacionado

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CONCATENATEX

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Concatena el resultado de una expresión evaluada para cada fila de una tabla.

Sintaxis

DAX

```
CONCATENATEX(<table>, <expression>[, <delimiter> [, <orderBy_expression> [, <order>]]...])
```

Parámetros

Expandir tabla

Término	Definición
tabla	Tabla que contiene las filas para las que se evaluará la expresión.
expresión	Expresión que se debe evaluar para cada fila de la <i>tabla</i> .
delimiter	(Opcional) Separador que se va a usar durante la concatenación.
orderBy_expression	(Opcional) Cualquier expresión DAX en la que se utilice el valor de resultado para ordenar los valores concatenados en la cadena de salida. Se evalúa para cada fila de la <i>tabla</i> .
orden	(Opcional) Valor que especifica cómo ordenar valores <i>orderBy_expression</i> , de forma ascendente o descendente:

El parámetro **orden** opcional acepta los valores siguientes:

Expandir tabla

Value	Valor alternativos	Descripción
0 (cero)	FALSE, DESC	Ordena de forma descendente los valores de <i>orderBy_expression</i> . Es el valor predeterminado si se omite el parámetro <i>order</i> .
1	TRUE, ASC	Ordena de forma ascendente los valores de <i>orderBy_expression</i> .

Valor devuelto

Cadena concatenada.

Notas

- Esta función toma como primer argumento una tabla o una expresión que devuelve una tabla. El segundo argumento es una columna que contiene los valores que se quieren concatenar, o una expresión que devuelve un valor.
- Los valores concatenados no se ordenan necesariamente en un orden determinado, a menos que *orderBy_expression* se especifique.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Tabla de empleados

 Expandir tabla

FirstName	LastName
Alan	Brewer
Michael	Blythe

La fórmula siguiente:

DAX

```
= CONCATENATEX(Employees, [FirstName] & " " & [LastName], ",")
```

Devuelve:

"Alan Brewer, Michael Blythe"

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

EXACT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Compara dos cadenas de texto y devuelve TRUE si son exactamente iguales, o FALSE en caso contrario. EXACT distingue mayúsculas de minúsculas, pero omite las diferencias de formato. EXACT distingue entre mayúsculas y minúsculas.

Sintaxis

DAX

```
EXACT(<text1>,<text2>)
```

Parámetros

Expandir tabla

Término	Definición
text1	Primera cadena o columna de texto que contiene el texto.
text2	Segunda cadena o columna de texto que contiene el texto.

Valor devuelto

Verdadero o Falso. (Booleano)

Ejemplo

La fórmula siguiente usada en una columna calculada de la tabla Product comprueba el valor de Product para la fila actual con el valor de Model para la fila actual; devuelve True si son iguales y devuelve False si son diferentes.

Los ejemplos de este artículo se pueden usar con el modelo de ejemplo de Power BI Desktop de Adventure Works DW 2020. Para obtener el modelo, vea [Modelo de ejemplo de DAX](#).

DAX

```
= EXACT([Product], [Model])
```

Contenido relacionado

Funciones de texto

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

FIND

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Devuelve la primera posición de una cadena de texto en otra. FIND distingue mayúsculas de minúsculas

Sintaxis

DAX

```
FIND(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

Parámetros

Expandir tabla

Término	Definición
find_text	Texto que se desea buscar. Use comillas dobles (texto vacío) para hacer coincidir el primer carácter de <code>within_text</code> .
within_text	Texto que contiene lo que se quiere buscar.
start_num	(Opcional) Carácter en el que se va a iniciar la búsqueda; si se omite, <code>start_num</code> = 1. El primer carácter de <code>within_text</code> es el número de carácter 1.
NotFoundValue	(Opcional, pero recomendado encarecidamente) Valor que se debe devolver cuando la operación no encuentra una subcadena coincidente, normalmente 0, -1 o <code>BLANK()</code> . Si no se especifica, se devolverá un error.

Valor devuelto

Número que muestra el punto de inicio de la cadena de texto que se quiere buscar.

Observaciones

- Mientras que Microsoft Excel tiene varias versiones de la función FIND para dar cabida a los lenguajes de juego de caracteres de byte único (SBCS) y de juego de caracteres de doble byte (DBCS), DAX usa Unicode y cuenta cada carácter de la

misma manera. Por lo tanto, no es necesario usar una versión distinta en función del tipo de carácter.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).
- FIND no admite caracteres comodín. Para usar caracteres comodín, use [SEARCH](#).

Ejemplo

La siguiente consulta DAX encuentra la posición de la primera letra de "Bike" en la cadena que contiene el nombre del distribuidor. Si no se encuentra, se devuelve Blank.

Tenga en cuenta que FIND distingue mayúsculas de minúsculas. En este ejemplo, si se usara "bike" en el argumento <find_text>, no se devolvería ningún resultado. Use [SEARCH](#) para no distinguir mayúsculas de minúsculas.

Los ejemplos de este artículo se pueden usar con el modelo de ejemplo de Power BI Desktop de Adventure Works DW 2020. Para obtener el modelo, vea [Modelo de ejemplo de DAX](#).

```
DAX

EVALUATE
CALCULATETABLE (
    ADDCOLUMNS (
        TOPN ( 10, SUMMARIZE('Reseller', [Reseller], [Business Type])),
        "Position of Bike", FIND ( "Bike", 'Reseller'[Reseller], 1, BLANK ())
    ),
    'Reseller'[Business Type] IN { "Specialty Bike Shop", "Value Added Reseller", "Warehouse" }
)
```

Vea que devuelve lo siguiente:

[Expandir tabla](#)

Reseller	Business Type (Tipo de empresa)	Position of Bike
Volume Bike Sellers	Warehouse	8
Mass Market Bikes	Value Added Reseller	13
Twin Cycles	Value Added Reseller	

Reseller	Business Type (Tipo de empresa)	Position of Bike
Rich Department Store	Warehouse	
Rental Gallery	Specialty Bike Shop	
Budget Toy Store	Warehouse	
Global Sports Outlet	Warehouse	
Online Bike Catalog	Warehouse	8
Helmets and Cycles	Value Added Reseller	
Jumbo Bikes	Specialty Bike Shop	7

Contenido relacionado

[SEARCH](#)

[Funciones de texto](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

FIXED

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Redondea un número al número de decimales especificado y devuelve el resultado como texto. Se puede especificar que el resultado se devuelva con o sin comas.

Sintaxis

DAX

```
FIXED(<number>, <decimals>, <no_commas>)
```

Parámetros

Expandir tabla

Término	Definición
número	Número que se quiere redondear y convertir en texto, o una columna que contenga un número.
decimals	(Opcional) Número de dígitos a la derecha de la coma decimal; si se omite, se asume 2.
no_commas	(Opcional) Valor lógico: si es 1, no se muestran comas en el texto devuelto; si es 0 o se omite, se muestran comas en el texto devuelto.

Valor devuelto

Número representado como texto.

Notas

- Si el valor usado para el parámetro **decimals** es negativo, **number** se redondea a la izquierda de la coma decimal.
- Si se omite **decimals**, se supone que es 2.

- Si `no_commas` es 0 o se omite, el texto devuelto incluye las comas de la forma habitual.
- La diferencia principal entre dar formato a una celda que contiene un número utilizando un comando y dar formato directamente a un número con la función `FIXED` es que `FIXED` convierte su resultado en texto. Un número al que se da formato con un comando desde el menú Formato sigue siendo un número.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula siguiente usada en una columna calculada obtiene el valor numérico de la fila actual en `Product[List Price]` y lo devuelve como texto con 2 posiciones decimales y sin comas.

Los ejemplos de este artículo se pueden usar con el modelo de ejemplo de Power BI Desktop de Adventure Works DW 2020. Para obtener el modelo, vea [Modelo de ejemplo de DAX](#).

DAX

```
= FIXED([List Price],2,1)
```

Contenido relacionado

[CEILING](#)
[FLOOR](#)
[ISO.CEILING](#)
[MROUND](#)
[ROUND](#)
[ROUNDDOWN](#)
[ROUNDUP](#)

Comentarios

¿Le ha resultado útil esta página?

Sí

No

FORMAT

Artículo • 20/10/2023

Convierte un valor en texto según el formato especificado.

Sintaxis

DAX

```
FORMAT(<value>, <format_string>[, <locale_name>])
```

Parámetros

Término	Definición
valor	Valor o expresión que se evalúa como un valor único.
format_string	Cadena con la plantilla de formato.
nombreDeConfiguraciónRegional	(Opcional) Nombre de la configuración regional que va a usar la función. Los valores posibles son cadenas aceptadas por la función API de Windows LocaleNameToLCID() .

Valor devuelto

Cadena que contiene **value** con el formato tal como lo define **format_string**.

ⓘ Nota

Si **value** es BLANK, la función devuelve una cadena vacía.

Si **format_string** es BLANK, se da formato al valor con un formato "Número general" o "Fecha general" (según el tipo de datos **value**).

Notas

- Las cadenas de formato predefinidas utilizan la propiedad Referencia cultural de modelo al aplicar formato al resultado. De forma predeterminada, la propiedad Referencia cultural de modelo se establece de acuerdo con la configuración

regional de usuario del equipo. Para los nuevos modelos de Power BI Desktop, la propiedad Referencia cultural de modelo se puede cambiar en Opciones > Configuración regional > Idioma del modelo. Para Analysis Services, la referencia cultural de modelo se puede establecer de acuerdo con la propiedad de idioma definida inicialmente por la instancia.

- Las cadenas de formato admitidas como un argumento para la función FORMAT de DAX se basan en las cadenas de formato que utiliza Visual Basic (automatización OLE), no en las cadenas de formato que utiliza .NET Framework. Por lo tanto, podría obtener resultados inesperados o un error si el argumento no coincide con ninguna cadena de formato definida. Por ejemplo, no se admite "p" como abreviatura de "Porcentaje". Las cadenas que se proporcionan como argumento a la función FORMAT y que no están incluidas en la lista de cadenas de formato predefinidas se administran como parte de una cadena de formato personalizado o como un literal de cadena.
- Para obtener más información sobre cómo especificar una configuración regional con FORMAT, vea este [vídeo](#).
- El uso de FORMAT cambia un resultado de medida a un tipo de datos de texto. Si el resultado de la medida es originalmente del tipo de datos numérico, con FORMAT, la medida no se puede usar en objetos visuales en los que la sección de valores requiere un tipo de datos numérico, como con gráficos. En Power BI, también puede usar **cadenas de formato dinámico para las medidas** o especificar una cadena de formato condicional que mantenga el tipo de datos numérico de la medida.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplos

Cadenas de formato

DAX

```
= 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")
```

Devuelve:

12345,67 "Número general" muestra el número sin formato.

\$12.345,67 "Moneda" muestra el número con el formato de configuración regional de moneda. En este ejemplo se muestra el formato de moneda predeterminado de Estados Unidos.

12345,67 "Fijo" muestra al menos un dígito a la izquierda y dos a la derecha del separador decimal.

12.345,67 "Estándar" muestra al menos un dígito a la izquierda y dos a la derecha del separador decimal, e incluye separadores de miles. En este ejemplo se muestra el formato de número predeterminado de Estados Unidos.

1.234.567,00 % "Porcentaje" muestra el número como porcentaje (multiplicado por 100) con formato y el signo de porcentaje a la derecha del número separado por un solo espacio.

1,23E+04 "Científica" muestra el número en notación científica con dos dígitos decimales.

Datetime con locale_name opcional

DAX

```
= FORMAT( dt"2020-12-15T12:30:59", BLANK(), "en-US" )  
= FORMAT( dt"2020-12-15T12:30:59", BLANK(), "en-GB" )  
= FORMAT( dt"2020-12-15T12:30:59", "mm/dd/yyyy", "en-GB" )
```

Devuelve:

12/15/2020 12:30:59 PM Donde el mes precede al día y la hora tiene un formato de 12 horas.

15/12/2020 12:30:59 Donde el día precede al mes y la hora tiene un formato de 24 horas.

12/15/2020 12:30:59 Donde el mes precede al día y la hora tiene un formato de 24 horas. Dado que se especifica una cadena de formato no dependiente de la configuración regional, no se aplica la configuración regional y se devuelve el formato sin configuración regional.

Formatos numéricos predefinidos

Los siguientes formatos numéricicos predefinidos se pueden especificar en el argumento **format_string**:

Formato	Descripción
"General" "Number"	Muestra el número sin separadores de miles.
"Currency"	Muestra el número con separadores de miles, si procede; muestra dos dígitos a la derecha del separador decimal. La salida se basa en la configuración regional del sistema.
"Fixed"	Muestra al menos un dígito a la izquierda y dos a la derecha del separador decimal.
"Standard"	Muestra el número con separador de miles, al menos un dígito a la izquierda y dos a la derecha del separador decimal.
"Percent"	Muestra el número multiplicado por 100 con un signo de porcentaje (%) anexado justo a la derecha; siempre muestra dos dígitos a la derecha del separador decimal.
"Scientific"	Usa la notación científica estándar, que proporciona dos dígitos significativos.
"Yes/No"	Muestra No si el número es 0; de lo contrario, muestra Sí.
"True/False"	Muestra False si el número es 0; de lo contrario, muestra True.
"On/Off"	Muestra Desactivado si el número es 0; de lo contrario, muestra Activado.

Formatos numéricos personalizados

Una expresión de formato personalizado para los números puede tener de una a tres secciones separadas por punto y coma. Si el argumento de cadena de formato contiene uno de los formatos numéricos con nombre, únicamente se permite una sección.

Si se usa	El resultado es
Solo una sección	La expresión de formato se aplica a todos los valores.
Dos secciones	La primera sección se aplica a valores positivos y ceros; la segunda se aplica a valores negativos.
Tres secciones.	La primera sección se aplica a valores positivos, la segunda a valores negativos y la tercera a los ceros.

```
"$#,##0; ($#,##0)"
```

Si incluye puntos y coma sin nada entre ellos, la sección que falta se define con el formato del valor positivo. Por ejemplo, el siguiente formato muestra los valores positivos y negativos utilizando el formato de la primera sección y muestra "Zero" si el valor es cero.

DAX

```
"$#,##0"
```

Si incluye puntos y coma sin nada entre ellos, la sección que falta se muestra con el formato del valor positivo.

Caracteres de formato numérico personalizados

Los siguientes caracteres de formato numérico personalizados se pueden especificar en el argumento **format_string**:

Carácter	Descripción
Ninguno	Muestra el número sin formato.
(0)	Marcador de posición de dígitos. Muestra un dígito o un cero. Si la expresión tiene un dígito en la posición donde aparece el 0 en la cadena de formato, se muestra; en caso contrario, se muestra un cero en esa posición. Si el número tiene menos dígitos que ceros (en cualquier lado del separador decimal) en la expresión de formato, se muestran ceros a la izquierda o a la derecha. Si el número tiene más dígitos a la derecha del separador decimal que ceros en la expresión de formato, se redondea el número a tantos lugares decimales como ceros haya. Si el número tiene más dígitos a la izquierda del separador decimal que ceros en la expresión de formato, se muestran los dígitos adicionales sin modificación.
(#)	Marcador de posición de dígitos. Muestra un dígito o nada. Si la expresión tiene un dígito en la posición donde aparece este símbolo en la cadena de formato, se muestra; en caso contrario, no se muestra nada en esa posición. Este símbolo funciona como marcador de posición de 0 dígitos, excepto que los ceros a la izquierda y a la derecha no se muestran si el número tiene los mismos dígitos o menos que caracteres # en cualquier lado del separador decimal en la expresión de formato.
(.)	Marcador de posición decimal. En algunas configuraciones regionales, se usa una coma como separador decimal. El marcador de posición decimal determina el número de dígitos que se muestran a la izquierda y a la derecha del separador decimal. Si la expresión de formato solo contiene signos de número a la izquierda de este símbolo, los números menores que 1 comienzan con un separador decimal. Para mostrar un cero a la izquierda con números fraccionarios, use 0 como primer marcador de

Carácter	Descripción
	posición de dígitos a la izquierda del separador decimal. El carácter real usado como marcador de posición de decimales en la salida con formato depende del formato de número que reconozca el sistema.
(%)	Marcador de posición de porcentaje. La expresión se multiplica por 100. El carácter de porcentaje (%) se inserta en la posición en la que aparece en la cadena de formato.
(,)	Separador de miles. En algunas configuraciones regionales, se usa un punto como separador de miles. El separador de miles separa miles de cientos en un número que tiene cuatro o más posiciones a la izquierda del separador decimal. El uso estándar del separador de miles se especifica si el formato contiene un separador de miles rodeado de marcadores de posición de dígitos (0 o #). Dos separadores de miles adyacentes o un separador de miles inmediatamente a la izquierda del separador decimal (tanto si se especifica o no un decimal) significa "dividir el número entre 1000 para escalarlo y redondearlo según sea necesario." Por ejemplo, puede usar la cadena de formato "###0,," para representar 100 millones como 100. Los números menores que 1 millón se muestran como 0. Dos separadores de miles adyacentes en cualquier posición que no sea la inmediatamente a la izquierda del separador decimal se tratan simplemente como si se especificara el uso de un separador de miles. El carácter real que se usa como separador de miles en la salida con formato depende del formato de número que reconozca el sistema.
(:)	Separador de líneas. En algunas configuraciones regionales, se pueden usar otros caracteres para representar el separador de hora. El separador de hora separa las horas, los minutos y los segundos cuando se aplica formato a los valores de hora. La configuración del sistema determina el carácter real que se usa como separador de hora en la salida con formato.
(/)	Separador de fecha. En algunas configuraciones regionales, se pueden usar otros caracteres para representar el separador de fecha. El separador de fecha separa el día, el mes y el año cuando se aplica formato a los valores de fecha. La configuración del sistema determina el carácter real que se usa como separador de fecha en la salida con formato.
(E- E+ e- e+)	Formato científico. Si la expresión de formato contiene al menos un marcador de posición de dígitos (0 o #) a la derecha de E-, E+, e- o e+, el número se muestra en formato científico y se inserta E o e entre el número y su exponente. El número de marcadores de posición de dígitos a la derecha determina el número de dígitos en el exponente. Use E- o e- para colocar un signo menos junto a los exponentes negativos. Use E + o e + para colocar un signo menos junto a los exponentes negativos y un signo más junto a los exponentes positivos.
- + \$ ()	Muestra un carácter literal. Para mostrar un carácter que no sea uno de los indicados, debe ir precedido de una barra diagonal inversa (\) o entre comillas dobles ("").
\	Muestra el siguiente carácter en la cadena de formato. Para mostrar un carácter con un significado especial como carácter literal, debe ir precedido de una barra diagonal inversa (\). La barra diagonal inversa no se muestra. Utilizar una barra diagonal inversa es equivalente a delimitar el siguiente carácter con comillas dobles. Para

Carácter	Descripción
mostrar una barra diagonal inversa, use dos barras diagonales inversas (\ \). Algunos ejemplos de caracteres que no se pueden mostrar como caracteres literales son los caracteres de formato de fecha y hora (a, c, d, h, m, n, p, q, s, t, w, y, / y :), los caracteres de formato numérico (#, 0, %, E, e, coma y punto) y los caracteres de formato de cadena (@, &, <, > y !).	
("ABC")	Muestra la cadena entre comillas dobles (" ").

Formatos predefinidos de fecha y hora

Los siguientes formatos de fecha y hora predefinidos se pueden especificar en el argumento `format_string`. Si se usan formatos distintos a estos, se interpretarán como un formato personalizado de fecha y hora:

Formato	Descripción
"General Date"	Muestra una fecha, una hora o ambas. Por ejemplo, 12/3/2008 11:07:31. La representación de la fecha está determinada por el valor de la referencia cultural actual de la aplicación.
"Long Date" o "Medium Date"	Muestra una fecha según el formato de fecha larga de la referencia cultural actual. Por ejemplo, miércoles, 12 de marzo de 2008.
"Short Date"	Muestra una fecha con el formato de fecha corta de la referencia cultural actual. Por ejemplo, 12/3/2008.
"Long Time" o "Medium Time"	Muestra una hora con el formato de hora larga de la referencia cultural actual; normalmente incluye horas, minutos y segundos. Por ejemplo, 11:07:31.
"Short Time"	Muestra una hora en formato de 12 horas. Por ejemplo, 11:07.

Formatos personalizados de fecha y hora

Los caracteres de formato siguientes pueden especificarse en `format_string` para crear formatos personalizados de fecha y hora:

Carácter	Descripción
(:)	Separador de líneas. En algunas configuraciones regionales, se pueden usar otros caracteres para representar el separador de hora. El separador de hora separa las horas, los minutos y los segundos cuando se aplica formato a los valores de hora. La

Carácter	Descripción
	configuración del sistema determina el carácter real que se usa como separador de hora en la salida con formato.
(/)	Separador de fecha. En algunas configuraciones regionales, se pueden usar otros caracteres para representar el separador de fecha. El separador de fecha separa el día, el mes y el año cuando se aplica formato a los valores de fecha. La configuración del sistema determina el carácter real que se usa como separador de fecha en la salida con formato.
(\)	Barra diagonal inversa. Muestra el carácter siguiente como literal. Por lo tanto, no se interpreta como un carácter de formato.
("")	Comillas dobles. Se muestra el texto entre comillas dobles. Por lo tanto, no se interpreta como caracteres de formato.
c	Muestra la fecha como <code>ddddd</code> y muestra la hora como <code>ttttt</code> , en ese orden. Muestra solo la información de fecha si no hay ninguna parte fraccionaria en el número de serie de fecha; muestra solo la información de hora si no hay ninguna parte entera.
d	Muestra el día como un número sin un cero interlineado (1 a 31).
dd	Muestra el día como un número con un cero interlineado (01 a 31).
ddd	Muestra el día abreviado (dom. a sáb.). Localizado.
dddd	Muestra el día como un nombre completo (domingo a sábado). Localizado.
ddddd	Muestra la fecha como una fecha completa (día, mes y año), con el formato que especifique la configuración de formato de fecha corta del sistema. El formato de fecha corta predeterminado es <code>mm/dd/yyyy</code> .
ddyyyy	Muestra un número de serie de fecha como fecha completa (día, mes y año), con el formato de fecha larga que reconozca el sistema. El formato de fecha larga predeterminado es <code>ddd, mmmm d, yyyy</code> .
w	Muestra el día de la semana como un número (del 1 para el domingo al 7 para el sábado).
ww	Muestra la semana del año como un número (1 a 54).
m	Muestra el mes como un número sin un cero interlineado (1 a 12). Si <code>m</code> sigue inmediatamente a <code>h</code> o <code>hh</code> , se muestra el minuto en lugar del mes.
MM	Muestra el mes como un número con un cero interlineado (01 a 12). Si <code>mm</code> sigue inmediatamente a <code>h</code> o <code>hh</code> , se muestra el minuto en lugar del mes.
mmm	Muestra el mes abreviado (en. a dic.). Localizado.
mmmm	Muestra el mes con el nombre completo (enero a diciembre). Localizado.

Carácter	Descripción
q	Muestra el trimestre del año como un número (1 a 4).
s	Muestra el día del año como un número (1 a 366).
yy	Muestra el año como un número de 2 dígitos (00 a 99).
aaaa	Muestra el año como un número de 4 dígitos (100 a 9999).
h	Muestra la hora como un número sin un cero interlineado (0 a 23).
hh	Muestra la hora como un número con un cero interlineado (00 a 23).
n	Muestra el minuto como un número sin un cero interlineado (0 a 59).
nn	Muestra el minuto como un número con un cero interlineado (00 a 59).
s	Muestra el segundo como un número sin un cero interlineado (0 a 59).
ss	Muestra el segundo como un número con un cero interlineado (00 a 59).
tttt	Muestra la hora como una hora completa (hora, minuto y segundo), utilizando el separador de hora definido en el formato de hora que reconozca el sistema. Se muestra un cero a la izquierda si se selecciona la opción de cero a la izquierda y la hora es anterior a las 10:00. El formato predeterminado es <code>h:mm:ss</code> .
AM/PM	Usa el reloj de 12 horas y muestra AM en mayúsculas con cualquier hora antes del mediodía; muestra PM en mayúsculas con cualquier hora entre el mediodía y las 11:59 p.m.
AM/PM	Usa el reloj de 12 horas y muestra AM en minúsculas con cualquier hora antes del mediodía; muestra PM en minúsculas con cualquier hora entre el mediodía y las 11:59 p. m.
A/P	Usa el reloj de 12 horas y muestra una A mayúscula con cualquier hora antes del mediodía; muestra una P mayúscula con cualquier hora entre el mediodía y las 11:59 p. m.
A/P	Usa el reloj de 12 horas y muestra una A minúscula con cualquier hora antes del mediodía; muestra una P minúscula con cualquier hora entre el mediodía y las 11:59 p. m.
AMPM	Usa el reloj de 12 horas y muestra el literal de cadena AM tal y como lo define el sistema con cualquier hora antes del mediodía; muestra el literal de cadena PM tal y como lo define el sistema con cualquier hora entre el mediodía y las 11:59 p. m. AMPM puede estar en mayúsculas o en minúsculas, pero coincidirá con la cadena tal y como la defina la configuración del sistema. El formato predeterminado es AM/PM. Si el sistema se define en el reloj de 24 horas, la cadena se establece normalmente en una cadena vacía.

El formato de fecha y hora usa la configuración regional actual del usuario para dar formato la cadena. Por ejemplo, considere la fecha 25 de junio de 2020. Cuando se aplica formato con la cadena de formato "m/d/yyyy", será:

- La configuración regional del usuario es la de los Estados Unidos de América (en-US): "6/25/2020"
- La configuración regional del usuario es la de Alemania (de-DE): "6.25.2020"

Ejemplos de formatos personalizados de fecha y hora

En los ejemplos siguientes se usa la fecha y hora jueves, 25 de junio de 2020, a las 13:23:45. Alemania (de-DE) utiliza el sistema de 24 horas. No hay equivalentes de A. M. ni P. M.

Formato	Resultado (en-US)	Resultado (de-DE)
"c"	06/25/2020 13:23:45	25.06.2020 13:23:45
"d"	25	25
"dd"	25	25
"ddd"	Ju.	Cosas que hacer
"dddd"	Jueves	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"	Junio	Juni
"q"	2	2
"y"	177	177
"yy"	20	20
"yyyy"	2020	2020

Formato	Resultado (en-US)	Resultado (de-DE)
"""Year"" yyyy"	Año 2020	Año 2020
"yyyy \Qq"	2020 T2	2020 T2
"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
"tttt"	13:23:45	13:23:45
"tttt 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

Comentarios

¿Le ha resultado útil esta página?

Yes

No

LEFT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de caracteres especificados desde el principio de una cadena de texto.

Sintaxis

DAX

```
LEFT(<text>, <num_chars>)
```

Parámetros

Expandir tabla

Término	Definición
texto	La cadena de texto que contiene los caracteres que se quieren extraer, o bien una referencia a una columna que contiene texto.
num_chars	(Opcional) El número de caracteres que se quiere que LEFT extraiga; si se omite, es 1.

Valor devuelto

Una cadena de texto.

Notas

- Mientras que Microsoft Excel contiene otras funciones para trabajar con texto en lenguajes de caracteres de un solo byte y de doble byte, DAX trabaja con Unicode y almacena todos los caracteres con la misma longitud; por tanto, una sola función es suficiente.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En el ejemplo siguiente se devuelven los cinco primeros caracteres del nombre de la empresa en la columna [ResellerName] y las cinco primeras letras del código geográfico de la columna [GeographyKey] y se concatenan para crear un identificador.

DAX

```
= CONCATENATE(LEFT('Reseller'[ResellerName],LEFT(GeographyKey,3))
```

Si el argumento **num_chars** es un número mayor que el número de caracteres disponible, la función devuelve los caracteres máximos disponibles y no genera un error. Por ejemplo, la columna [GeographyKey] contiene números como 1, 12 y 311; por tanto, el resultado también tiene una longitud variable.

Contenido relacionado

[Funciones de texto](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

LEN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número de caracteres de una cadena de texto.

Sintaxis

DAX
<code>LEN(<text>)</code>

Parámetros

Expandir tabla

Término	Definición
texto	El texto cuya longitud se quiere buscar, o bien una columna que contiene texto. Los espacios cuentan como caracteres.

Valor devuelto

Un número entero que indica el número de caracteres de la cadena de texto.

Notas

- Mientras que Microsoft Excel tiene otras funciones para trabajar con lenguajes de caracteres de un solo byte y de doble byte, DAX usa Unicode y almacena todos los caracteres con la misma longitud.
- LEN siempre cuenta cada carácter como 1, con independencia de la configuración de idioma predeterminada.
- Si usa LEN con una columna que contiene valores que no son de texto, como fechas o valores booleanos, la función convierte de forma implícita el valor en texto, con el formato de columna actual.

Ejemplo

En la fórmula siguiente se suman las longitudes de las direcciones de las columnas [AddressLine1] y [AddressLine2].

DAX

```
= LEN([AddressLine1])+LEN([AddressLine2])
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

LOWER

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Convierte todas las letras de una cadena de texto a minúsculas.

Sintaxis

DAX
<code>LOWER(<text>)</code>

Parámetros

Expandir tabla

Término	Definición
texto	Texto que se quiere convertir a minúsculas o referencia a una columna que contiene texto.

Valor devuelto

Texto en minúsculas.

Observaciones

Los caracteres que no son letras no cambian. Por ejemplo, la fórmula `= LOWER("123ABC")` devuelve `123abc`.

Ejemplo

Esta fórmula obtiene cada fila de la columna [ProductCode] y convierte el valor a minúsculas. Los números de la columna no se ven afectados.

DAX
<code>LOWER([ProductCode])</code>

```
= LOWER('New Products'[ProductCode])
```

Contenido relacionado

[Funciones de texto](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MID

Artículo • 30/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve una cadena de caracteres de la mitad de una cadena de texto a partir de una posición inicial y una longitud.

Sintaxis

DAX

```
MID(<text>, <start_num>, <num_chars>)
```

Parámetros

[+] Expandir tabla

Término	Definición
texto	Cadena de texto de la que se quieren extraer los caracteres o una columna que contiene texto.
start_num	Posición del primer carácter que se quiere extraer. Las posiciones empiezan en 1.
num_chars	Número de caracteres que se va a devolver.

Valor devuelto

Cadena de texto de la longitud especificada.

Observaciones

Mientras que Microsoft Excel tiene diferentes funciones para trabajar con lenguajes de caracteres de un solo byte y de doble byte, DAX usa Unicode y almacena todos los caracteres con la misma longitud.

Ejemplos

La expresión siguiente,

DAX

```
MID("abcde",2,3))
```

Devuelve "bcd".

La expresión siguiente,

DAX

```
MID('Reseller'[ResellerName],1,5))
```

Devuelve el mismo resultado que `LEFT([ResellerName],5)`. Ambas expresiones devuelven las cinco primeras letras de la columna, `[ResellerName]`.

Contenido relacionado

[Funciones de texto](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

REPLACE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

REPLACE reemplaza la parte de una cadena de texto, según el número de caracteres que se especifique, por otra cadena de texto.

Sintaxis

DAX

```
REPLACE(<old_text>, <start_num>, <num_chars>, <new_text>)
```

Parámetros

Expandir tabla

Término	Definición
old_text	Cadena de texto que contiene los caracteres que se quieren reemplazar, o una referencia a una columna que contiene texto.
start_num	Posición del carácter en old_text que se quiere reemplazar por new_text .
num_chars	Número de caracteres que se quiere reemplazar. Advertencia: Si el argumento num_chars es un espacio en blanco o hace referencia a una columna que se evalúa como un espacio en blanco, la cadena de new_text se inserta en la posición start_num , sin reemplazar ningún carácter. Es el mismo comportamiento que en Excel.
new_text	Texto de reemplazo para los caracteres especificados en old_text .

Valor devuelto

Una cadena de texto.

Notas

- Mientras que Microsoft Excel tiene otras funciones para usar con lenguajes de caracteres de un solo byte y de doble byte, DAX usa Unicode y, por tanto,

almacena todos los caracteres con la misma longitud.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La siguiente fórmula crea una nueva columna calculada que reemplaza los dos primeros caracteres del código de producto de la columna [ProductCode] por un nuevo código de dos letras, OB.

DAX

```
= REPLACE('New Products'[Product Code],1,2,"OB")
```

Contenido relacionado

[Funciones de texto](#)

[Función SUBSTITUTE](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

REPT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Repite el texto un número determinado de veces. REPT se usa para llenar una celda con un número de instancias de una cadena de texto.

Sintaxis

DAX

```
REPT(<text>, <num_times>)
```

Parámetros

Expandir tabla

Término	Definición
texto	Texto que quiere repetir.
num_times	Número positivo que especifica el número de veces que se repetirá el texto.

Valor devuelto

Cadena que contiene los cambios.

Observaciones

- Si **number_times** es 0 (cero), REPT devuelve un valor en blanco.
- Si **number_times** no es un entero, se trunca.
- El resultado de la función REPT no puede tener más de 32 767 caracteres; si los tiene, REPT devuelve un error.

Ejemplo: repetir cadenas literales

En este ejemplo se devuelve la cadena 85, repetida tres veces.

DAX

```
= REPT("85",3)
```

Ejemplo: repetir valores de columna

En este ejemplo se devuelve la cadena de la columna [MyText], que se repite el número de veces que hay en la columna [MyNumber]. Como la fórmula se extiende para toda la columna, la cadena resultante depende del texto y el valor numérico de cada fila.

DAX

```
= REPT([MyText],[MyNumber])
```

 Expandir tabla

MyText	MyNumber	CalculatedColumn1
Texto	2	TextText
Número	0	
85	3	858 585

Contenido relacionado

[Funciones de texto](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

RIGHT

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

RIGHT devuelve el último carácter o caracteres de una cadena de texto, en función del número de caracteres que se especifique.

Sintaxis

DAX

```
RIGHT(<text>, <num_chars>)
```

Parámetros

Expandir tabla

Término	Definición
texto	Cadena de texto que contiene los caracteres que se quieren extraer, o una referencia a una columna que contiene texto.
num_chars	(Opcional) Número de caracteres que se quiere que RIGHT extraiga; si se omite, 1. También se puede usar una referencia a una columna que contenga números.

Si la referencia a la columna no contiene texto, se convierte implícitamente en texto.

Valor devuelto

Cadena de texto que contiene los caracteres especificados situados más a la derecha.

Notas

- RIGHT siempre cuenta cada carácter, ya sea de un solo byte o de doble byte, como 1, con independencia de cuál sea la configuración de idioma predeterminada.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo: devolver un número fijo de caracteres

La siguiente fórmula devuelve los últimos dos dígitos del código del producto de la tabla de nuevos productos.

DAX

```
= RIGHT('New Products'[ProductCode],2)
```

Ejemplo: usar una referencia de columna para especificar el recuento de caracteres

La siguiente fórmula devuelve un número variable de dígitos del código del producto de la tabla de nuevos productos, en función del número de la columna MyCount. Si no hay ningún valor en la columna MyCount o el valor es un espacio en blanco, RIGHT también devuelve un espacio en blanco.

DAX

```
= RIGHT('New Products'[ProductCode],[MyCount])
```

Contenido relacionado

[Funciones de texto](#)

[LEFT](#)

[MID](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SEARCH

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve el número del carácter en el que se encuentra por primera vez un carácter o una cadena de texto específicos, de izquierda a derecha. La búsqueda no distingue entre mayúsculas y minúsculas, y distingue los acentos.

Sintaxis

DAX

```
SEARCH(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

Parámetros

[] Expandir tabla

Término	Definición
find_text	El texto que se quiere buscar. En find_text puede usar caracteres comodín: el signo de interrogación (?) y el asterisco (*). Un signo de interrogación coincide con cualquier carácter individual; un asterisco coincide con cualquier secuencia de caracteres. Si quiere buscar un signo de interrogación o un asterisco reales, escriba una tilde (~) antes del carácter.
within_text	El texto en el que se quiere buscar find_text , o bien una columna que contiene texto.
start_num	(Opcional) La posición del carácter en within_text en la que se quiere iniciar la búsqueda. Si se omite, es 1.
NotFoundValue	(Opcional, pero recomendado encarecidamente) Valor que se debe devolver cuando la operación no encuentra una subcadena coincidente, normalmente 0, -1 o BLANK(). Si no se especifica, se devolverá un error.

Valor devuelto

El número de la posición inicial de la primera cadena de texto a partir del primer carácter de la segunda cadena de texto.

Notas

- La función de búsqueda no distingue mayúsculas de minúsculas. Si se busca "N", se encontrará la primera repetición de "N" o "n".
- La función de búsqueda distingue los acentos. La búsqueda de "á" encontrará la primera repetición de "á", pero no repeticiones de "a", "à" ni de las versiones en mayúsculas "A", "Á".
- Se puede usar la función SEARCH para determinar la ubicación de un carácter o cadena de texto dentro de otra cadena de texto y, después, usar la función MID para devolver el texto, o bien usar la función REPLACE para cambiarlo.
- Si no se encuentra **find_text** en **within_text**, la fórmula devuelve un error. Este comportamiento es similar al de Excel, que devuelve #VALUE si no se encuentra la subcadena. Los valores NULL de **within_text** se interpretarán como una cadena vacía en este contexto.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La siguiente consulta DAX encuentra la posición de la primera letra de "cycle" en la cadena que contiene el nombre del distribuidor. Si no se encuentra, se devuelve Blank.

SEARCH no distingue entre mayúsculas y minúsculas. En este ejemplo, si se usa "cycle" o "Cycle" en el argumento <find_text>, se devuelven resultados para ambos casos. Use **FIND** para distinguir mayúsculas de minúsculas.

Los ejemplos de este artículo se pueden usar con el modelo de ejemplo de Power BI Desktop de Adventure Works DW 2020. Para obtener el modelo, vea [Modelo de ejemplo de DAX](#).

DAX

```
EVALUATE  
CALCULATETABLE (  
    ADDCOLUMNS (  
        TOPN ( 10, SUMMARIZE('Reseller', [Reseller], [Business Type])),  
        "Position of cycle", SEARCH ( "cycle", 'Reseller'[Reseller], 1,
```

```

    BLANK () )
),
'Reseller'[Business Type] IN { "Specialty Bike Shop", "Value Added
Reseller", "Warehouse"}
)

```

Vea que devuelve lo siguiente:

 Expandir tabla

Reseller	Business Type (Tipo de empresa)	Position of cycle
Volume Bike Sellers	Warehouse	
Mass Market Bikes	Value Added Reseller	
Twin Cycles	Value Added Reseller	6
Rich Department Store	Warehouse	
Rental Gallery	Specialty Bike Shop	
Budget Toy Store	Warehouse	
Global Sports Outlet	Warehouse	
Online Bike Catalog	Warehouse	
Helmets and Cycles	Value Added Reseller	13
Jumbo Bikes	Specialty Bike Shop	

Contenido relacionado

[FIND](#)

[REPLACE](#)

[Funciones de texto](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SUBSTITUTE

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Reemplaza texto existente por otro nuevo en una cadena de texto.

Sintaxis

DAX

```
SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)
```

Parámetros

[] Expandir tabla

Término	Definición
texto	Texto en el que se quieren sustituir caracteres, o una referencia a una columna que contiene texto.
old_text	Texto existente que se quiere reemplazar.
new_text	Texto con el que se quiere reemplazar old_text .
instance_num	(Opcional) La instancia de old_text que se quiere reemplazar. Si se omite, se reemplazan todas las instancias de old_text .

Valor devuelto

Cadena de texto.

Observaciones

- Use la función SUBSTITUTE cuando quiera reemplazar texto específico en una cadena de texto. Use la función REPLACE si quiere reemplazar cualquier texto de longitud variable que se encuentre en una ubicación específica de una cadena de texto.

- La función SUBSTITUTE distingue entre mayúsculas y minúsculas. En el caso de que no haya ninguna coincidencia entre **text** y **old_text**, SUBSTITUTE no reemplazará el texto.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo: sustitución dentro de una cadena

Esta fórmula crea una copia de la columna [Product Code] que sustituye el nuevo código de producto **NW** por el código de producto anterior **PA** cada vez que aparezca en la columna.

DAX

```
= SUBSTITUTE([Product Code], "NW", "PA")
```

Contenido relacionado

[Funciones de texto](#)

[REPLACE](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TRIM

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Quita todos los espacios del texto, excepto los espacios individuales entre palabras.

Sintaxis

DAX
<code>TRIM(<text>)</code>

Parámetros

Expandir tabla

Término	Definición
<code>text</code>	El texto del que se quieren quitar los espacios, o bien una columna que contiene texto.

Valor devuelto

La cadena con los espacios quitados.

Observaciones

- Use TRIM en texto que haya recibido de otra aplicación que pueda tener un espaciado irregular.
- La función TRIM se diseñó originalmente para recortar el carácter de espacio ASCII de 7 bits (valor 32) del texto. En el juego de caracteres Unicode, hay un carácter de espacio adicional denominado carácter de espacio de no separación que tiene un valor decimal de 160. Este carácter se usa normalmente en páginas web como la entidad HTML, . Por sí sola, la función TRIM no quita este carácter de espacio de no separación. Para obtener un ejemplo de cómo recortar los dos caracteres de espacio del texto, vea Eliminación de espacios y caracteres no imprimibles del texto.

Ejemplo

La fórmula siguiente crea una cadena que no tiene ningún espacio en blanco al final.

DAX

```
= TRIM("A column with trailing spaces.    ")
```

Cuando se crea la fórmula, se propaga a través de la fila tal y como se ha escrito, de modo que se vea la cadena original en cada fórmula y los resultados no sean aparentes. Pero cuando se evalúa la fórmula, la cadena se recorta.

Puede comprobar que la fórmula genera el resultado correcto si comprueba la longitud de la columna calculada creada por la fórmula anterior, como se muestra a continuación:

DAX

```
= LEN([Calculated Column 1])
```

Contenido relacionado

[Funciones de texto](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

UNICHAR

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ✓ Cálculo visual

Devuelve el carácter Unicode al que hace referencia el valor numérico.

Sintaxis

DAX

`UNICHAR(number)`

Parámetros

[] Expandir tabla

Término	Definición
número	Número Unicode que representa el carácter.

Valor devuelto

Un carácter representado por el número Unicode.

Notas

- Si los caracteres XML no son válidos, UNICHAR devuelve un error.
- Si los números Unicode son suplementos parciales y los tipos de datos no son válidos, UNICHAR devuelve un error.
- Si los números son valores numéricos que se encuentran fuera del rango permitido, UNICHAR devuelve un error.
- Si el número es cero (0), UNICHAR devuelve un error.
- El carácter Unicode devuelto puede ser una cadena de caracteres, por ejemplo en códigos UTF-8 o UTF-16.

Ejemplo

En el ejemplo siguiente se devuelve el carácter representado por el número Unicode 66 (A mayúscula).

```
DAX  
= UNICHAR(65)
```

En el ejemplo siguiente se devuelve el carácter representado por el número Unicode 32 (carácter de espacio).

```
DAX  
= UNICHAR(32)
```

En el ejemplo siguiente se devuelve el carácter representado por el número Unicode 9733 (carácter ★).

```
DAX  
= UNICHAR(9733)
```

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

UNICODE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Devuelve el número (punto de código) que corresponde al primer carácter del texto.

Sintaxis

DAX
UNICODE(<Text>)

Parámetros

Expandir tabla

Término	Definición
Texto	El texto es el carácter para el que quiere el valor Unicode.

Valor devuelto

Código numérico para el primer carácter de una cadena de texto.

Comentarios

¿Le ha resultado útil esta página?

Sí

No

UPPER

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Convierte una cadena de texto a letras mayúsculas.

Sintaxis

DAX

`UPPER (<text>)`

Parámetros

Expandir tabla

Término	Definición
texto	Texto que se quiere convertir a mayúsculas o referencia a una columna que contiene texto.

Valor devuelto

Mismo texto en mayúsculas.

Ejemplo

La siguiente fórmula convierte la cadena de la columna [ProductCode] a mayúsculas. Los caracteres no alfabéticos no se ven afectados.

DAX

`= UPPER(['New Products'[Product Code])`

Contenido relacionado

[Funciones de texto](#)

[LOWER, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

VALOR

Artículo • 28/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Convierte en un número una cadena de texto que representa un número.

Sintaxis

DAX
<code>VALUE(<text>)</code>

Parámetros

Expandir tabla

Término	Definición
text	Texto que se va a convertir.

Valor devuelto

Número convertido en tipo de datos decimal.

Observaciones

- El valor que se pasa como parámetro **text** puede estar en cualquiera de los formatos de constante, número, fecha u hora reconocidos por la aplicación o los servicios que se están usando. Si **text** no está en uno de estos formatos, se devuelve un error.
- Normalmente no es necesario usar la función VALUE en una fórmula porque el motor convierte de forma implícita el texto en números según necesidad.
- También puede usar referencias de columna. Por ejemplo, si tiene una columna que contiene tipos de números mixtos, se puede usar VALUE para convertir todos los valores en un único tipo de datos numérico. Pero si usa la función VALUE con una columna que contiene números y texto mezclados, toda la columna se marca

con un error, ya que no todos los valores de todas las filas se pueden convertir en números.

Ejemplo

En la siguiente fórmula se convierte la cadena especificada, "3", en el valor numérico 3.

DAX

```
= VALUE("3")
```

Contenido relacionado

[Funciones de texto](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Funciones de inteligencia de tiempo

Artículo • 20/10/2023

La característica Expresiones de análisis de datos (DAX) también incluye funciones de inteligencia de tiempo que permiten manipular datos mediante períodos de tiempo (incluidos días, meses, trimestres y años) y, luego, compilar y comparar cálculos durante esos períodos. Antes de usar cualquier función de inteligencia de tiempo, asegúrese de marcar una de las tablas que contienen la columna de fecha como Tabla de fechas.

En esta categoría

Función	Descripción
CLOSINGBALANCEMONTH	Evalúa el valor de expression en la última fecha del mes en el contexto actual.
CLOSINGBALANCEQUARTER	Evalúa el valor de expression en la última fecha del trimestre en el contexto actual.
CLOSINGBALANCEYEAR	Evalúa el valor de expression en la última fecha del año en el contexto actual.
DATEADD	Devuelve una tabla que contiene una columna de fechas, desplazada hacia delante o hacia atrás en el tiempo, por el número especificado de intervalos desde las fechas del contexto actual.
DATESBETWEEN	Devuelve una tabla que contiene una columna de fechas que comienza con una fecha de inicio especificada y continúa hasta una fecha de finalización especificada.
DATESINPERIOD	Devuelve una tabla que contiene una columna de fechas que empieza por una fecha de inicio específica y sigue hasta el número y tipo de intervalos de fechas especificados.
DATESMTD	Devuelve una tabla que contiene una columna con las fechas del mes hasta la fecha, en el contexto actual.
DATESQTD	Devuelve una tabla que contiene una columna con las fechas del trimestre hasta la fecha, en el contexto actual.
DATESYTD	Devuelve una tabla que contiene una columna con las fechas del año hasta la fecha, en el contexto actual.
ENDOFMONTH	Devuelve la última fecha del mes del contexto actual para la columna de fechas especificada.

Función	Descripción
ENDOFQUARTER	Devuelve la última fecha del trimestre del contexto actual para la columna de fechas especificada.
ENDOFYEAR	Devuelve la última fecha del año del contexto actual para la columna de fechas especificada.
FIRSTDATE	Devuelve la primera fecha del contexto actual para la columna de fechas especificada.
FIRSTNONBLANK	Devuelve el primer valor de la columna, column, filtrado por el contexto actual, donde expression no está en blanco.
LASTDATE	Devuelve la última fecha del contexto actual para la columna de fechas especificada.
LASTNONBLANK	Devuelve el último valor de la columna, column, filtrado por el contexto actual, donde expression no está en blanco.
NEXTDAY	Devuelve una tabla que contiene una columna de todas las fechas del día siguiente, en función de la primera fecha especificada en la columna de fechas del contexto actual.
NEXTMONTH	Devuelve una tabla que contiene una columna de todas las fechas del mes siguiente, en función de la primera fecha de la columna de fechas del contexto actual.
NEXTQUARTER	Devuelve una tabla que contiene una columna de todas las fechas del trimestre siguiente, en función de la primera fecha de la columna de fechas del contexto actual.
NEXTYEAR	Devuelve una tabla que contiene una columna de todas las fechas del año siguiente, en función de la primera fecha de la columna de fechas del contexto actual.
OPENINGBALANCEMONTH	Evalúa expression en la primera fecha del mes del contexto actual.
OPENINGBALANCEQUARTER	Evalúa expression en la primera fecha del trimestre del contexto actual.
OPENINGBALANCEYEAR	Evalúa expression en la primera fecha del año en el contexto actual.
PARALLELPERIOD	Devuelve una tabla que contiene una columna de fechas que representa un período paralelo a las fechas de la columna de fechas especificada, en el contexto actual, con las fechas desplazadas un número de intervalos hacia delante o hacia atrás en el tiempo.
PREVIOUSDAY	Devuelve una tabla que contiene una columna de todas las fechas que representan el día anterior a la primera fecha de la columna

Función	Descripción
	de fechas, en el contexto actual.
PREVIOUSMONTH	Devuelve una tabla que contiene una columna de todas las fechas del mes anterior, en función de la primera fecha de la columna de fechas del contexto actual.
PREVIOUSQUARTER	Devuelve una tabla que contiene una columna de todas las fechas del trimestre anterior, en función de la primera fecha de la columna de fechas del contexto actual.
PREVIOUSYEAR	Devuelve una tabla que contiene una columna de todas las fechas del año anterior, en función de la última fecha de la columna de fechas del contexto actual.
SAMEPERIODLASTYEAR	Devuelve una tabla que contiene una columna de fechas desplazadas un año atrás en el tiempo desde las fechas de la columna de fechas especificada, en el contexto actual.
STARTOFMONTH	Devuelve la primera fecha del mes del contexto actual para la columna de fechas especificada.
STARTOFQUARTER	Devuelve la primera fecha del trimestre del contexto actual para la columna de fechas especificada.
STARTOFYEAR	Devuelve la primera fecha del año del contexto actual para la columna de fechas especificada.
TOTALMTD	Evalúa el valor de expression del mes hasta la fecha en el contexto actual.
TOTALQTD	Evalúa el valor de expression de las fechas del trimestre hasta la fecha en el contexto actual.
TOTALYTD	Evalúa el valor de year-to-date de expression en el contexto actual.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

CLOSINGBALANCEMONTH

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ! Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Evalúa la **expresión** en la última fecha del mes en el contexto actual.

Sintaxis

DAX

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

Parámetros

[+] Expandir tabla

Parámetro	Definición
expression	Expresión que devuelve un valor escalar.
fechas	Columna que contiene fechas.
filter	(Opcional) Expresión que especifica un filtro que se va a aplicar al contexto actual.

Valor devuelto

Un valor escalar que representa la **expresión** evaluada en la última fecha del mes en el contexto actual.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.

- Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.

① Nota

Las restricciones de las expresiones booleanas se describen en la [función CALCULATE](#).

① Nota

La expresión **filter** tiene restricciones descritas en el tema [Función CALCULATE](#).

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la fórmula del ejemplo siguiente se crea una medida que calcula el "valor de inventario de final de mes" del inventario de productos.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)
[CLOSINGBALANCEYEAR, función](#)
[Función CLOSINGBALANCEQUARTER](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CLOSINGBALANCEQUARTER

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ! Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Evalúa el valor de **expression** en la última fecha del trimestre del contexto actual.

Sintaxis

DAX

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

Parámetros

[+] Expandir tabla

Término	Definición
expression	Expresión que devuelve un valor escalar.
fechas	Columna que contiene fechas.
filter	(Opcional) Expresión que especifica un filtro que se va a aplicar al contexto actual.

Valor devuelto

Valor escalar que representa el valor de **expression** evaluado en la última fecha del trimestre del contexto actual.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.

- Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.

① Nota

Las restricciones de las expresiones booleanas se describen en la [función CALCULATE](#).

① Nota

La expresión **filter** tiene restricciones descritas en el tema [Función CALCULATE](#).

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del siguiente ejemplo crea una medida que calcula el "valor de inventario de fin de trimestre" del inventario de productos.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[CLOSINGBALANCEYEAR, función](#)

[Función CLOSINGBALANCEMONTH](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

CLOSINGBALANCEYEAR

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Evalúa la **expresión** en la última fecha del año en el contexto actual.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
expression	Expresión que devuelve un valor escalar.
fechas	Columna que contiene fechas.
filter	(Opcional) Expresión que especifica un filtro que se va a aplicar al contexto actual.
year_end_date	(Opcional) Cadena literal con una fecha que define la fecha de finalización del año. El valor predeterminado es el 31 de diciembre.

Valor devuelto

Un valor escalar que representa la **expresión** evaluada en la última fecha del año en el contexto actual.

Notas

- El parámetro `year_end_date` es un literal de cadena de una fecha, en la misma configuración regional que la del cliente donde se ha creado el libro. Se omite la parte de año de la fecha.
- El argumento `dates` puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.

① Nota

Las restricciones de las expresiones booleanas se describen en la [función CALCULATE](#).

① Nota

La expresión `filter` tiene restricciones descritas en el tema [Función CALCULATE](#).

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del siguiente ejemplo crea una medida que calcula el "valor de inventario de final de año" del inventario de productos.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[CLOSINGBALANCEYEAR, función](#)

[Función CLOSINGBALANCEQUARTER](#)

[Función CLOSINGBALANCEMONTH](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DATEADD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna de fechas que se han desplazado hacia delante o hacia atrás en el tiempo según el número especificado de intervalos desde las fechas del contexto actual.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
fechas	Una columna que contiene fechas.
number_of_intervals	Entero que especifica el número de intervalos que se van a sumar o restar a las fechas.
interval	Intervalo por el que se van a desplazar las fechas. El valor del intervalo puede ser uno de los siguientes: <code>year</code> , <code>quarter</code> , <code>month</code> o <code>day</code> .

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

El argumento **dates** puede ser cualquiera de los siguientes:

- Una referencia a una columna de fecha y hora.
- Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.

ⓘ Nota

Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).

- Si el número de intervalos especificado para **number_of_intervals** es positivo, las fechas de **dates** se mueven hacia delante en el tiempo; si el número es negativo, las fechas de **dates** se desplazan hacia atrás en el tiempo.
- El parámetro **interval** es una enumeración, no un conjunto de cadenas, por lo tanto, los valores no deben ir entre comillas. Además, los valores `year`, `quarter`, `month` y `day` deben escribirse completos cuando se usan.
- La tabla de resultados solo incluye fechas que existen en la columna **dates**.
- Si las fechas del contexto actual no forman un intervalo contiguo, la función devuelve un error.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo: Desplazar un conjunto de fechas

La fórmula siguiente calcula las fechas del año anterior equivalentes a las fechas del contexto actual.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DATESBETWEEN

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna de fechas que comienza con una fecha de inicio especificada y continúa hasta una fecha de finalización especificada.

Esta función es adecuada para pasar como filtro en la función [CALCULATE](#). Úsela para filtrar una expresión por un intervalo de fechas personalizado.

Nota

Si trabaja con intervalos de fechas estándar como días, meses, trimestres o años, se recomienda usar la función [DATESINPERIOD](#), más adecuada para este propósito.

Sintaxis

DAX

`DATESBETWEEN(<Dates>, <StartDate>, <EndDate>)`

Parámetros

Expandir tabla

Término	Definición
Fechas	Columna de fecha.
StartDate	Una expresión de fecha.
EndDate	Una expresión de fecha.

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Comentarios

- En el caso de uso más común, **Dates** es una referencia a la columna de fecha de una tabla de fechas marcada.
- Si **StartDate** es BLANK, **StartDate** será el primer valor de la columna **Dates**.
- Si **EndDate** es BLANK, **EndDate** será el último valor de la columna **Dates**.
- Las fechas usadas como **StartDate** y **EndDate** son inclusivas. Por ejemplo, si el valor de **StartDate** es el 1 de julio de 2019, esa fecha se incluirá en la tabla devuelta (siempre que la fecha exista en la columna **Dates**).
- La tabla devuelta solo puede incluir fechas almacenadas en la columna **Dates**. Por ejemplo, si la columna **Dates** comienza el 1 de julio de 2017 y el valor **StartDate** es el 1 de julio de 2016, la tabla devuelta comenzará a partir del 1 de julio de 2017.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La siguiente definición de medida de la tabla **Sales** usa la función DATESBETWEEN para generar un cálculo *hasta la fecha* (LTD). El cálculo hasta la fecha representa la acumulación de una medida a lo largo del tiempo desde el mismo inicio.

Observe que la fórmula usa la función **MAX**. Esta función devuelve la última fecha que se encuentra en el contexto de filtro. Por lo tanto, la función DATESBETWEEN devuelve una tabla de fechas que empieza en la fecha más antigua y termina en la última fecha que se va a notificar.

Los ejemplos de este artículo se pueden usar con el modelo de ejemplo de Power BI Desktop de Adventure Works DW 2020. Para obtener el modelo, vea [Modelo de ejemplo de DAX](#).

DAX

```
Customers LTD =  
CALCULATE(  
DISTINCTCOUNT(Sales[CustomerKey]),
```

```
DATESBETWEEN(  
    'Date'[Date],  
    BLANK(),  
    MAX('Date'[Date])  
)  
)
```

Tenga en cuenta que la fecha más antigua almacenada en la tabla **Date** tabla es el 1 de julio de 2017. Por lo tanto, cuando un informe filtra la medida por el mes de junio de 2020, la función DATESBETWEEN devuelve un intervalo de fechas que va desde el 1 de julio de 2017 hasta el 30 de junio de 2020.

Contenido relacionado

- [Funciones de inteligencia de tiempo \(DAX\)](#)
- [Funciones de fecha y hora \(DAX\)](#)
- [Función DATESINPERIOD \(DAX\)](#)

Comentarios

¿Le ha resultado útil esta página?



DATESINPERIOD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna de fechas que empieza por una fecha de inicio específica y sigue hasta el número y tipo de intervalos de fechas especificados.

Esta función es adecuada para pasar como filtro en la función [CALCULATE](#). Úsela para filtrar una expresión por intervalos de fechas estándar, como días, meses, trimestres o años.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
fechas	Columna de fecha.
start_date	Una expresión de fecha.
number_of_intervals	Entero que especifica el número de intervalos que se van a sumar o restar a las fechas.
interval	El intervalo según el cual se van a desplazar las fechas. El valor del intervalo puede ser uno de los siguientes: <code>DAY</code> , <code>MONTH</code> , <code>QUARTER</code> y <code>YEAR</code> .

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Comentarios

- En el caso de uso más común, **dates** es una referencia a la columna de fecha de una tabla de fechas marcada.
- Si el número especificado para **number_of_intervals** es positivo, las fechas se mueven hacia delante en el tiempo; si el número es negativo, las fechas se desplazan hacia atrás en el tiempo.
- El parámetro **interval** es una enumeración. Los valores válidos son **DAY**, **MONTH**, **QUARTER** y **YEAR**. Dado que es una enumeración, los valores no se pasan como cadenas. Por lo que no debe ponerlas entre comillas.
- La tabla devuelta solo puede incluir fechas almacenadas en la columna **dates**. Por ejemplo, si la columna **dates** comienza el 1 de julio de 2017 y el valor **start_date** es el 1 de julio de 2016, la tabla devuelta comenzará a partir del 1 de julio de 2017.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La definición siguiente de medida de la tabla **Sales** usa la función **DATESINPERIOD** para calcular los ingresos del año anterior (PY).

Observe que la fórmula usa la función **MAX**. Esta función devuelve la última fecha que se encuentra en el contexto de filtro. Por lo tanto, la función **DATESINPERIOD** devuelve una tabla de fechas que empieza a partir de la fecha más reciente del último año.

Los ejemplos de este artículo se pueden usar con el modelo de ejemplo de Power BI Desktop de Adventure Works DW 2020. Para obtener el modelo, vea [Modelo de ejemplo de DAX](#).

DAX

```
Revenue PY =  
CALCULATE(  
    SUM(Sales[Sales Amount]),  
    DATESINPERIOD(  
        'Date'[Date],  
        MAX('Date'[Date]),  
        -1,
```

```
YEAR  
)  
)
```

Tenga en cuenta que el informe se filtra por el mes de junio de 2020. La función MAX devuelve el 30 de junio de 2020. Después, la función DATESINPERIOD devuelve un intervalo de fechas desde el 1 de julio de 2019 hasta el 30 de junio de 2020. Se trata de un año de valores de fecha, a partir del 30 de junio de 2020, para el último año.

Contenido relacionado

[Funciones de inteligencia de tiempo \(DAX\)](#)

[Funciones de fecha y hora \(DAX\)](#)

[Función DATESBETWEEN \(DAX\)](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DATESMTD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna con las fechas del mes hasta la fecha, en el contexto actual.

Sintaxis

DAX

`DATESMTD(<dates>)`

Parámetros

Expandir tabla

Término	Definición
fechas	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

El argumento **dates** puede ser cualquiera de los siguientes:

- Una referencia a una columna de fecha y hora.
- Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.

- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.

ⓘ Nota

Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del ejemplo siguiente crea una medida que calcula el "Total del mes hasta la fecha" para las Ventas por Internet.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[DATESYTD, función](#)

[DATESQTD, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DATESQTD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna con las fechas del trimestre hasta la fecha, en el contexto actual.

Sintaxis

DAX

`DATESQTD(<dates>)`

Parámetros

Expandir tabla

Término	Definición
fechas	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

El argumento **dates** puede ser cualquiera de los siguientes:

- Una referencia a una columna de fecha y hora.
- Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.

- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.

ⓘ Nota

Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del ejemplo siguiente crea una medida que calcula el "Total trimestral acumulado" de las Ventas por Internet.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[DATESYTD, función](#)

[DATESMTD, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

DATESYTD

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ! Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna con las fechas del año hasta la fecha, en el contexto actual.

Sintaxis

DAX

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

Parámetros

[] Expandir tabla

Término	Definición
fechas	Columna que contiene fechas.
year_end_date	(Opcional) Cadena literal con una fecha que define la fecha de finalización del año. El valor predeterminado es el 31 de diciembre.

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

El argumento **dates** puede ser cualquiera de los siguientes:

- Una referencia a una columna de fecha y hora.

- Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.

 Nota

Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).

- El parámetro `year_end_date` es un literal de cadena de una fecha, en la misma configuración regional que la del cliente donde se ha creado el libro. Se omite la parte de año de la fecha. En función de la configuración regional, el formato puede ser similar a "m-dd" o "dd-m".
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del ejemplo siguiente crea una medida que calcula el "Total acumulado" para las Ventas por Internet.

DAX

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

En la fórmula de ejemplo siguiente se crea una medida que calcula el "Total acumulado del año fiscal" para las ventas por Internet, que emplea una configuración regional de EE. UU. para el formato de fecha.

DAX

```
= CALCULATE(
    SUM(InternetSales_USD[SalesAmount_USD]),
    DATESYTD(DateTime[DateKey],
        "6-30"
    )
)
```

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[DATESMTD, función](#)

[DATESQTD, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ENDOFMONTH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve la última fecha del mes del contexto actual para la columna de fechas especificada.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
fechas	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una sola columna y una sola fila con un valor de fecha.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.

- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del ejemplo siguiente crea una medida que devuelve el final del mes del contexto actual.

DAX

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Funciones de inteligencia de tiempo](#)

[Función ENDOFYEAR](#)

[ENDOFQUARTER, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ENDOFQUARTER

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve la última fecha del trimestre del contexto actual para la columna de fechas especificada.

Sintaxis

DAX

`ENDOFQUARTER(<dates>)`

Parámetros

Expandir tabla

Término	Definición
fechas	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una sola columna y una sola fila con un valor de fecha.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.

- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del ejemplo siguiente crea una medida que devuelve el final del trimestre del contexto actual.

DAX

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Funciones de inteligencia de tiempo](#)

[Función ENDOFYEAR](#)

[Función ENDOFMONTH](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

ENDOFYEAR

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve la última fecha del año del contexto actual para la columna de fechas especificada.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
fechas	Columna que contiene fechas.
year_end_date	(Opcional) Cadena literal con una fecha que define la fecha de finalización del año. El valor predeterminado es el 31 de diciembre.

Valor devuelto

Tabla que contiene una sola columna y una sola fila con un valor de fecha.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.

- Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
 - El parámetro **year_end_date** es un literal de cadena de una fecha, en la misma configuración regional que la del cliente donde se ha creado el libro. Se omite la parte de año de la fecha.
 - Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del ejemplo siguiente crea una medida que devuelve el final del año fiscal, que acaba el 30 de junio, para el contexto actual.

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Funciones de inteligencia de tiempo](#)

[Función ENDOFMONTTH](#)

[ENDOFQUARTER, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

FIRSTDATE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve la primera fecha del contexto actual para la columna de fechas especificada.

Sintaxis

DAX

`FIRSTDATE(<dates>)`

Parámetros

Expandir tabla

Término	Definición
fechas	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una sola columna y una sola fila con un valor de fecha.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.

- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- Si el contexto actual es una sola fecha, la fecha que devuelven las funciones FIRSTDATE y LASTDATE es igual.
- El valor devuelto es una tabla que contiene una sola columna y un único valor. Por lo tanto, esta función se puede usar como argumento para cualquier función que requiera una tabla en sus argumentos. Además, el valor devuelto se puede usar siempre que se requiera un valor de fecha.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del ejemplo siguiente crea una medida que obtiene la primera fecha en la que se ha realizado una venta en el canal de Ventas por Internet para el contexto actual.

DAX

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Funciones de inteligencia de tiempo](#)

[Función LASTDATE](#)

[FIRSTNONBLANK, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

FIRSTNONBLANK

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve el primer valor de la columna, **column**, filtrado por el contexto actual, donde la expresión no está en blanco.

Sintaxis

DAX

FIRSTNONBLANK(<column>,<expression>)

Parámetros

Expandir tabla

Término	Definición
columna	Expresión de columna.
expression	Expresión evaluada para detectar espacios en blanco para cada valor de column .

Valor devuelto

Tabla que contiene una sola columna y una sola fila con el primer valor calculado.

Notas

- El argumento **column** puede ser cualquiera de los siguientes:
 - Una referencia a cualquier columna.
 - Una tabla con una sola columna.

- Una expresión booleana que define una tabla de una sola columna.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- Esta función se usa normalmente para devolver el primer valor de una columna cuya expresión no está en blanco. Por ejemplo, se podría obtener el último valor para el que hubiera ventas de un producto.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Contenido relacionado

[LASTNONBLANK, función](#)

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

FIRSTNONBLANKVALUE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Evalúa una expresión filtrada por los valores ordenados de una columna y devuelve el primero de la expresión que no esté vacío.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
columna	Columna o expresión que devuelve una tabla de columna única.
expression	Expresión evaluada para cada valor de <column>.

Valor devuelto

Primer valor que no esté vacío de <expression> correspondiente a los valores ordenados de <column>.

Notas

- El argumento de columna puede ser cualquiera de los siguientes:
 - Una referencia a cualquier columna.
 - Una tabla con una sola columna.

- Esta función difiere de la función FIRSTNONBLANK, ya que la columna <column> se agrega al contexto de filtro para la evaluación de <expression>.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Consulte la siguiente consulta DAX:

```
DAX

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

Vea que devuelve lo siguiente:

 Expandir tabla

DimProduct[Class]	[FNBV]
L	699,0982
H	13 778,24
M	1000,4375
	533,83

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

LASTDATE

Artículo • 26/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ! Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve la última fecha del contexto actual para la columna de fechas especificada.

Sintaxis

DAX

`LASTDATE(<dates>)`

Parámetros

[+] Expandir tabla

Término	Definición
fechas	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una sola columna y una sola fila con un valor de fecha.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.

- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- Si el contexto actual es una sola fecha, la fecha devuelta por las funciones FIRSTDATE y LASTDATE es igual.
- Técnicamente, el valor devuelto es una tabla que contiene una sola columna y un único valor. Por lo tanto, esta función se puede usar como argumento para cualquier función que requiera una tabla en sus argumentos. Además, el valor devuelto se puede usar siempre que se requiera un valor de fecha.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la fórmula del ejemplo siguiente se crea una medida que obtiene la última fecha del contexto actual, cuando se ha realizado una venta en el canal de ventas por Internet.

DAX

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Funciones de inteligencia de tiempo](#)

[Función FIRSTDATE](#)

[LASTNONBLANK, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

LASTNONBLANK

Artículo • 30/04/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ! Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve el último valor de la columna, **column**, filtrado por el contexto actual, donde la expresión no está en blanco.

Sintaxis

DAX

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

Parámetros

[] Expandir tabla

Término	Definición
columna	Expresión de columna.
expression	Expresión evaluada para detectar espacios en blanco para cada valor de column .

Valor devuelto

Tabla que contiene una sola columna y una sola fila con el último valor calculado.

Notas

- El argumento **column** puede ser cualquiera de los siguientes:
 - Una referencia a cualquier columna.
 - Una tabla con una sola columna.
 - Una expresión booleana que define una tabla de una sola columna.

- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- Esta función se usa normalmente para devolver el último valor de una columna cuya expresión no está en blanco. Por ejemplo, se podría obtener el último valor para el que hubiera ventas de un producto.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Contenido relacionado

[FIRSTNONBLANK](#), función

[Funciones estadísticas](#)

Comentarios

¿Le ha resultado útil esta página?



LASTNONBLANKVALUE

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Evalúa una expresión filtrada por los valores ordenados de una columna y devuelve el último de la expresión que no esté vacío.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
columna	Columna o expresión que devuelve una tabla de columna única.
expression	Expresión evaluada para cada valor de <column>.

Valor devuelto

Último valor que no esté vacío de <expression> correspondiente a los valores ordenados de <column>.

Notas

- El argumento de columna puede ser cualquiera de los siguientes:
 - Una referencia a cualquier columna.
 - Una tabla con una sola columna.

- Esta función difiere de la función LASTNONBLANK, ya que la columna <column> se agrega al contexto de filtro para la evaluación de <expression>.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

Consulte la siguiente consulta DAX:

```
DAX

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

Vea que devuelve lo siguiente:

 Expandir tabla

DimProduct[Class]	[LNBV]
L	132,44
H	137,6
M	84,97
	2288,6

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NEXTDAY

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna de todas las fechas del día siguiente, en función de la primera fecha especificada en la columna **dates** del contexto actual.

Sintaxis

DAX

`NEXTDAY(<dates>)`

Parámetros

Expandir tabla

Término	Definición
dates	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

- Esta función devuelve todas las fechas del día siguiente hasta la primera fecha del parámetro de entrada. Por ejemplo, si la primera fecha del argumento **dates** se refiere al 10 de junio 2009, esta función devuelve todas las fechas que coincidan con 11 de junio de 2009.

- El argumento **dates** puede ser cualquiera de estos:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En esta fórmula de ejemplo se crea una medida que calcula las "ventas del día siguiente" de las ventas por Internet.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[NEXTQUARTER, función](#)

[Función NEXTMONTH](#)

[Función NEXTYEAR](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NEXTMONTH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna de todas las fechas del mes siguiente, en función de la primera fecha de la columna **dates** del contexto actual.

Sintaxis

DAX

`NEXTMONTH(<dates>)`

Parámetros

Expandir tabla

Término	Definición
dates	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

- Esta función devuelve todas las fechas del día siguiente a la primera fecha del parámetro de entrada. Por ejemplo, si la primera fecha del argumento **dates** se refiere al 10 de junio de 2009, esta función devuelve todas las fechas del mes de julio de 2009.

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del siguiente ejemplo crea una medida que calcula las "ventas del mes siguiente" para las ventas por Internet.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[NEXTDAY, función](#)

[NEXTQUARTER, función](#)

[Función NEXTYEAR](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NEXTQUARTER

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Nota

No se recomienda usar esta función en cálculos visuales, ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna de todas las fechas del trimestre siguiente en función de la primera fecha especificada en la columna **dates** del contexto actual.

Sintaxis

DAX

`NEXTQUARTER(<dates>)`

Parámetros

Expandir tabla

Término	Definición
dates	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

- Esta función devuelve todas las fechas del trimestre siguiente en función de la primera fecha del parámetro de entrada. Por ejemplo, si la primera fecha de la columna **dates** se refiere al 10 de junio de 2009, esta función devuelve todas las fechas del trimestre comprendido entre julio y septiembre de 2009.

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del siguiente ejemplo crea una medida que calcula las "ventas del trimestre siguiente" para las ventas por Internet.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[NEXTDAY, función](#)

[Función NEXTMONTH](#)

[Función NEXTYEAR](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

NEXTYEAR

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna de todas las fechas del año siguiente en función de la primera fecha de la columna **dates** del contexto actual.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
dates	Columna que contiene fechas.
year_end_date	(Opcional) Cadena literal con una fecha que define la fecha de finalización del año. El valor predeterminado es el 31 de diciembre.

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

- Esta función devuelve todas las fechas del año siguiente en función de la primera fecha de la columna de entrada. Por ejemplo, si la primera fecha de la columna **dates** se refiere al año 2007, esta función devuelve todas las fechas del año 2008.

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- El parámetro **year_end_date** es un literal de cadena de una fecha, en la misma configuración regional que la del cliente donde se ha creado el libro. Se omite la parte de año de la fecha.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del siguiente ejemplo crea una medida que calcula las "ventas del año siguiente" para las ventas por Internet.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[NEXTDAY, función](#)

[NEXTQUARTER, función](#)

[Función NEXTMONTH](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

OPENINGBALANCEMONTH

Artículo • 30/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Evalúa la **expresión** en la primera fecha del mes del contexto actual.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
expression	Expresión que devuelve un valor escalar.
fechas	Columna que contiene fechas.
filter	(Opcional) Expresión que especifica un filtro que se va a aplicar al contexto actual.

Valor devuelto

Valor escalar que representa la **expresión** evaluada en la primera fecha del mes del contexto actual.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.

- Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
 - La expresión **filter** tiene restricciones descritas en el tema [Función CALCULATE](#).
 - Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del siguiente ejemplo crea una medida que calcula el "valor de inventario de inicio de mes" del inventario de productos.

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

Contenido relacionado

[OPENINGBALANCEYEAR, función](#)
[Función OPENINGBALANCEQUARTER](#)
[Funciones de inteligencia de tiempo](#)
[Función CLOSINGBALANCEMONTH](#)

Comentarios

¿Le ha resultado útil esta página?



OPENINGBALANCEQUARTER

Artículo • 15/06/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Evalúa la **expresión** en la fecha correspondiente al final del trimestre anterior en el contexto actual.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
expression	Expresión que devuelve un valor escalar.
fechas	Columna que contiene fechas.
filte*	(Opcional) Expresión que especifica un filtro que se va a aplicar al contexto actual.

Valor devuelto

Valor escalar que representa la **expresión** evaluada en la primera fecha del trimestre del contexto actual.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:

- Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
 - La expresión **filter** tiene restricciones descritas en el tema [Función CALCULATE](#).
 - Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del siguiente ejemplo crea una medida que calcula el "valor de inventario de inicio de trimestre" del inventario de productos.

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

Contenido relacionado

[OPENINGBALANCEYEAR, función](#)

[Función OPENINGBALANCEMONTH](#)

[Funciones de inteligencia de tiempo](#)

[Función CLOSINGBALANCEQUARTER](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

OPENINGBALANCEYEAR

Artículo • 15/06/2024

Se aplica a: ✓ Columna calculada ✓ Tabla calculada ✓ Medida ! Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Evaluá la **expresión** en la fecha correspondiente al final del año anterior en el contexto actual.

Sintaxis

DAX

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

Parámetros

[] Expandir tabla

Término	Definición
expression	Expresión que devuelve un valor escalar.
fechas	Columna que contiene fechas.
filter	(Opcional) Expresión que especifica un filtro que se va a aplicar al contexto actual.
year_end_date	(Opcional) Cadena literal con una fecha que define la fecha de finalización del año. El valor predeterminado es el 31 de diciembre.

Valor devuelto

Un valor escalar que representa la **expresión** evaluada en la primera fecha del año en el contexto actual.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- La expresión **filter** tiene restricciones descritas en el tema [Función CALCULATE](#).
- El parámetro **year_end_date** es un literal de cadena de una fecha, en la misma configuración regional que la del cliente donde se ha creado el libro. Se omite la parte de año de la fecha.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la fórmula del ejemplo siguiente se crea una medida que calcula el "valor de inventario de inicio de año" del inventario de productos.

DAX

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

Contenido relacionado

[Función OPENINGBALANCEQUARTER](#)

[Función OPENINGBALANCEMONTH](#)

[Funciones de inteligencia de tiempo](#)

[CLOSINGBALANCEYEAR, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PARALLELPERIOD

Artículo • 20/10/2023

Devuelve una tabla que contiene una columna de fechas que representa un período paralelo a las fechas de la columna **dates** especificada, en el contexto actual, con las fechas desplazadas un número de intervalos hacia delante o hacia atrás en el tiempo.

Sintaxis

DAX

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

Parámetros

Término	Definición
fechas	Una columna que contiene fechas.
number_of_intervals	Entero que especifica el número de intervalos que se van a sumar o restar a las fechas.
interval	El intervalo según el cual se van a desplazar las fechas. El valor del intervalo puede ser uno de los siguientes: <code>year</code> , <code>quarter</code> , <code>month</code> .

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

- Esta función toma el conjunto actual de fechas de la columna especificada por **dates**, desplaza la primera fecha y la última el número de intervalos especificado y, después, devuelve todas las fechas contiguas entre las dos fechas desplazadas. Si el intervalo es un rango parcial de mes, trimestre o año, los meses parciales del resultado también se llenan para completar todo el intervalo.
- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.

- Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- Si el número de intervalos especificado para **number_of_intervals** es positivo, las fechas de **dates** se mueven hacia delante en el tiempo; si el número es negativo, las fechas de **dates** se desplazan hacia atrás en el tiempo.
- El parámetro **interval** es una enumeración, no un conjunto de cadenas; por tanto, los valores no deben ir entre comillas. Además, los valores **year**, **quarter** y **month** se deben escribir al completo cuando se usan.
- La tabla de resultados solo incluye las fechas que aparecen en los valores de la columna de la tabla subyacente.
- La función PARALLELPERIOD es similar a la función DATEADD, salvo que PARALLELPERIOD siempre devuelve períodos completos en el nivel de granularidad dado, en lugar de los períodos parciales que devuelve DATEADD. Por ejemplo, si tiene una selección de fechas que empieza el 10 de junio y finaliza el 21 de junio del mismo año, y quiere desplazar esa selección hacia delante en un mes, la función PARALLELPERIOD devolverá todas las fechas del mes siguiente (del 1 de al 31 de julio); pero, si se usa DATEADD en su lugar, el resultado incluirá únicamente fechas del 10 al 21 de julio.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la fórmula del ejemplo siguiente se crea una medida que calcula las ventas por Internet del año anterior.

DAX

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

Vea también

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[Función DATEADD](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

PREVIOUSDAY

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna de todas las fechas que representan el día anterior a la primera fecha de la columna **dates**, en el contexto actual.

Sintaxis

DAX

`PREVIOUSDAY(<dates>)`

Parámetros

Expandir tabla

Término	Definición
dates	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

- Esta función determina la primera fecha del parámetro de entrada y, luego, devuelve todas las fechas correspondientes al día anterior a esa primera fecha. Por ejemplo, si la primera fecha del argumento **dates** hace referencia al 10 de junio de 2009, esta función devuelve todas las fechas iguales al 9 de junio de 2009.

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones de las expresiones booleanas se describen en el tema [Función CALCULATE](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La siguiente fórmula de ejemplo crea una medida que calcula las "ventas del día anterior" por Internet.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[Función PREVIOUSMONTH](#)

[PREVIOUSQUARTER, función](#)

[PREVIOUSYEAR, función](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PREVIOUSMONTH

Artículo • 02/05/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna de todas las fechas del mes anterior, en función de la primera fecha de la columna **Dates** del contexto actual.

Sintaxis

DAX

`PREVIOUSMONTH(<Dates>)`

Parámetros

Expandir tabla

Término	Definición
Fechas	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

- Esta función devuelve todas las fechas del mes anterior mediante la primera fecha de la columna usada como entrada. Por ejemplo, si la primera fecha del argumento **Dates** se refiere al 10 de junio de 2009, esta función devuelve todas las fechas del mes de mayo de 2009.

- El argumento **Dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones en expresiones booleanas se describen en el tema [CALCULATE](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la fórmula del ejemplo siguiente se crea una medida que calcula las "ventas del mes anterior" para las ventas por Internet.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[PREVIOUSDAY](#)

[PREVIOUSQUARTER](#)

[PREVIOUSYEAR](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PREVIOUSQUARTER

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna de todas las fechas del trimestre anterior, en función de la primera fecha de la columna **dates** del contexto actual.

Sintaxis

DAX

`PREVIOUSQUARTER(<dates>)`

Parámetros

Expandir tabla

Término	Definición
dates	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

- Esta función devuelve todas las fechas del trimestre siguiente, mediante la primera fecha de la columna de entrada. Por ejemplo, si la primera fecha del argumento **dates** se refiere al 10 de junio de 2009, esta función devuelve todas las fechas del trimestre de enero a marzo de 2009.

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones en expresiones booleanas se describen en el tema [CALCULATE](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la fórmula del ejemplo siguiente se crea una medida que calcula las "ventas del trimestre anterior" para las ventas por Internet.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[PREVIOUSMONTH](#)

[PREVIOUSDAY](#)

[PREVIOUSYEAR](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

PREVIOUSYEAR

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna de todas las fechas del año anterior, en función de la última fecha de la columna **dates** del contexto actual.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Término	Definición
dates	Columna que contiene fechas.
year_end_date	(Opcional) Cadena literal con una fecha que define la fecha de finalización del año. El valor predeterminado es el 31 de diciembre.

Valor devuelto

Tabla que contiene una única columna de valores de fecha.

Notas

- Esta función devuelve todas las fechas del año anterior dada la última fecha del parámetro de entrada. Por ejemplo, si la última fecha del argumento **dates** hace

referencia al año 2009, esta función devuelve todas las fechas del año 2008, hasta el valor de **year_end_date** especificado.

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones en expresiones booleanas se describen en el tema [CALCULATE](#).
- El parámetro **year_end_date** es un literal de cadena de una fecha, en la misma configuración regional que la del cliente donde se ha creado el libro. Se omite la parte de año de la fecha.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la fórmula del ejemplo siguiente se crea una medida que calcula las ventas por Internet del año anterior.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[PREVIOUSMONTH](#)

[PREVIOUSDAY](#)

[PREVIOUSQUARTER](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

SAMEPERIODLASTYEAR

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve una tabla que contiene una columna de fechas desplazadas un año atrás en el tiempo desde las fechas de la columna **dates** especificada, en el contexto actual.

Sintaxis

DAX

`SAMEPERIODLASTYEAR(<dates>)`

Parámetros

Expandir tabla

Término	Definición
<code>dates</code>	Columna que contiene fechas.

Valor devuelto

Tabla de valores de fecha de una sola columna.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.

- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones en expresiones booleanas se describen en el tema [CALCULATE](#).
- Las fechas devueltas son las mismas que las que devuelve esta fórmula equivalente: `DATEADD(dates, -1, year)`.
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la fórmula del ejemplo siguiente se crea una medida que calcula las ventas del año anterior de los revendedores.

DAX

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

Contenido relacionado

[Funciones de inteligencia de tiempo](#)

[Funciones de fecha y hora](#)

[PREVIOUSYEAR](#)

[PARALLELPERIOD](#)

Comentarios

¿Le ha resultado útil esta página?



STARTOFTMONTH

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medición Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve la primera fecha del mes del contexto actual para la columna de fechas especificada.

Sintaxis

DAX

`STARTOFTMONTH(<dates>)`

Parámetros

Expandir tabla

Término	Definición
fechas	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una sola columna y una sola fila con un valor de fecha.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.

- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones en expresiones booleanas se describen en el tema [CALCULATE](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del ejemplo siguiente crea una medida que devuelve el inicio del mes del contexto actual.

DAX

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Funciones de inteligencia de tiempo](#)

[STARTOFTYEAR](#)

[STARTOFQUARTER](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

STARTOFQUARTER

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve la primera fecha del trimestre del contexto actual para la columna de fechas especificada.

Sintaxis

DAX

`STARTOFQUARTER(<dates>)`

Parámetros

Expandir tabla

Término	Definición
fechas	Columna que contiene fechas.

Valor devuelto

Tabla que contiene una sola columna y una sola fila con un valor de fecha.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.

- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones en expresiones booleanas se describen en el tema [CALCULATE](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del siguiente ejemplo crea una medida que devuelve el inicio del trimestre del contexto actual.

DAX

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Funciones de inteligencia de tiempo](#)

[STARTOFCYEAR](#)

[STARTOFCMONTH](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

STARTOFTYEAR

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Devuelve la primera fecha del año del contexto actual para la columna de fechas especificada.

Sintaxis

DAX

`STARTOFTYEAR(<dates>)`

Parámetros

Expandir tabla

Término	Definición
fechas	Columna que contiene fechas.
YearEndDate	(Opcional) Valor de fecha de finalización del año.

Valor devuelto

Tabla que contiene una sola columna y una sola fila con un valor de fecha.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.

- Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones en expresiones booleanas se describen en el tema [CALCULATE](#).
- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del ejemplo siguiente crea una medida que devuelve el inicio del año del contexto actual.

DAX

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

Contenido relacionado

[Funciones de fecha y hora](#)

[Funciones de inteligencia de tiempo](#)

[STARTOFTQUARTER](#)

[STARTOFTMONTH](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TOTALMTD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Evalúa el valor de la **expresión** del mes hasta la fecha en el contexto actual.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Parámetro	Definición
expression	Expresión que devuelve un valor escalar.
fechas	Columna que contiene fechas.
filter	(Opcional) Expresión que especifica un filtro que se va a aplicar al contexto actual.

Valor devuelto

Valor escalar que representa la **expression** evaluada para las fechas en el mes actual hasta la fecha, especificadas las fechas en **dates**.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.

- Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones en expresiones booleanas se describen en el tema [CALCULATE](#).
 - La expresión **filter** tiene restricciones descritas en el tema [CALCULATE](#).
 - Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del ejemplo siguiente crea una medida que calcula el "total acumulado del mes" o la "suma acumulada del mes" de las ventas por Internet.

DAX

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

Contenido relacionado

[ALL](#)

[CALCULATE](#)

[TOTALYTD](#)

[TOTALQTD](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TOTALQTD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Evalúa el valor de **expression** de las fechas del trimestre hasta la fecha del contexto actual.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Parámetro	Definición
expression	Expresión que devuelve un valor escalar.
fechas	Columna que contiene fechas.
filter	(Opcional) Expresión que especifica un filtro que se va a aplicar al contexto actual.

Valor devuelto

Valor escalar que representa el valor de **expression** evaluado para todas las fechas del trimestre actual hasta la fecha, dadas las fechas de **dates**.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:

- Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones en expresiones booleanas se describen en el tema [CALCULATE](#).
 - La expresión **filter** tiene restricciones descritas en el tema [CALCULATE](#).
 - Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

La fórmula del ejemplo siguiente crea una medida que calcula el "total del trimestre" o la "suma total del trimestre" de las ventas por Internet.

DAX

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

Contenido relacionado

[ALL](#)
[CALCULATE](#)
[TOTALYTD](#)
[TOTALMTD](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

TOTALYTD

Artículo • 26/04/2024

Se aplica a: Columna calculada Tabla calculada Medida Cálculo visual

ⓘ Nota

No se recomienda usar esta función en [cálculos visuales](#), ya que es probable que devuelva resultados sin sentido.

Evalúa el valor anual hasta la fecha de la **expresión** en el contexto actual.

Sintaxis

DAX

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

Parámetros

Expandir tabla

Parámetro	Definición
expression	Expresión que devuelve un valor escalar.
fechas	Columna que contiene fechas.
filter	(Opcional) Expresión que especifica un filtro que se va a aplicar al contexto actual.
year_end_date	(Opcional) Cadena literal con una fecha que define la fecha de finalización del año. El valor predeterminado es el 31 de diciembre.

Valor devuelto

Un valor escalar que representa la **expresión** evaluada para las **fechas** en el año actual hasta la fecha.

Notas

- El argumento **dates** puede ser cualquiera de los siguientes:
 - Una referencia a una columna de fecha y hora.
 - Una expresión de tabla que devuelve una sola columna de valores de fecha y hora.
 - Una expresión booleana que define una tabla de una sola columna de valores de fecha y hora.
- Las restricciones en expresiones booleanas se describen en el tema [CALCULATE](#).
- La expresión **filter** tiene restricciones descritas en el tema [CALCULATE](#).
- El parámetro **year_end_date** es un literal de cadena de una fecha, en la misma configuración regional que la del cliente donde se ha creado el libro. La parte del año de la fecha no es obligatoria y se omite. Por ejemplo, en la fórmula siguiente se especifica un valor **year_end_date** (fiscal) de 30/6 en un libro con la configuración regional EN-US.

DAX

```
= TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey],  
ALL('DateTime'), "6/30")
```

En este ejemplo, **year_end_date** se puede especificar como "30/6", "junio 30", "30 de junio" o cualquier cadena que se resuelva como un mes y día. Pero se recomienda especificar **year_end_date** con "mes/día" (tal como se muestra) para asegurarse de que la cadena se resuelve como una fecha.

- Esta función no se admite para su uso en el modo DirectQuery cuando se utiliza en columnas calculadas o en reglas de seguridad de nivel de fila (RLS).

Ejemplo

En la fórmula del ejemplo siguiente se crea una medida que calcula el "total acumulado del año" o la "suma acumulada del año" de las ventas por Internet.

DAX

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

Contenido relacionado

ALL

CALCULATE

DATESYTD

TOTALMTD

TOTALQTD

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

Instrucciones

Artículo • 20/10/2023

En esta categoría

.	Descripción
DEFINE	(Palabra clave) Presenta una o varias definiciones de entidad que se pueden aplicar a una o varias instrucciones EVALUATE.
EVALUATE	(Palabra clave) Presenta una instrucción que contiene una expresión de tabla necesaria para ejecutar una consulta DAX.
MEASURE	(Palabra clave) Presenta una definición de medida que se puede usar en una o varias instrucciones EVALUATE de una consulta.
ORDER BY	(Palabra clave) Presenta una instrucción que define el criterio de ordenación de los resultados de la consulta devueltos por una instrucción EVALUATE.
START AT	(Palabra clave) Presenta una instrucción que define el valor inicial en el que se devuelven los resultados de la consulta de una instrucción ORDER BY.
VAR	(Palabra clave) Almacena el resultado de una expresión como una variable con nombre que después se puede pasar como argumento a otras expresiones de medida.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

DEFINIR

Artículo • 14/12/2023

Presenta una instrucción con una o varias definiciones de entidades que se puede aplicar a una o más instrucciones EVALUATE de una [consulta DAX](#).

Sintaxis

```
DAX

[DEFINE
(
    (MEASURE <table name>[<measure name>] = <scalar expression>) |
    (VAR <var name> = <table or scalar expression>) |
    (TABLE <table name> = <table expression>) |
    (COLUMN <table name>[<column name>] = <scalar expression>) |
) +
]

(EVALUATE <table expression>) +
```

Parámetros

[\[\] Expandir tabla](#)

Término	Definición
Entidad	MEASURE, VAR, TABLE ¹ o COLUMN ¹ .
name	Nombre de una definición de medida, variable, tabla o columna. No puede ser una expresión. El nombre no tiene que ser único. El nombre existe únicamente lo que dura la consulta.
expression	Cualquier expresión DAX que devuelve un valor de tabla o escalar. La expresión puede usar cualquiera de las entidades definidas. Si es necesario convertir una expresión escalar en una expresión de tabla, encapsule la expresión dentro de un constructor de tabla con llaves {} o use la función ROW() para devolver una tabla de una sola fila.

[1]Precaución: las definiciones TABLE y COLUMN con ámbito de consulta están pensadas solo para uso interno. Aunque puede definir expresiones TABLE y COLUMN para una consulta sin errores de sintaxis, pueden producir errores en tiempo de ejecución, por lo que no se recomienda.

Observaciones

- Una consulta DAX puede tener varias instrucciones EVALUATE, pero solo puede tener una instrucción DEFINE. Las definiciones de la instrucción DEFINE se pueden aplicar a cualquier instrucción EVALUATE de la consulta.
- Se requiere al menos una definición en una instrucción DEFINE.
- Definiciones de medida para las medidas del modelo de invalidación de consultas con el mismo nombre.
- Los nombres VAR tienen restricciones únicas. Para obtener más información, consulte [VAR: parámetros](#).
- Para obtener más información sobre cómo se usa una instrucción DEFINE, consulte [Consultas DAX](#).

Contenido relacionado

[EVALUATE](#)

[VAR](#)

[MEASURE](#)

[Consultas DAX](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

EVALUAR

Artículo • 14/12/2023

Introduce una instrucción que contiene una expresión de tabla necesaria en una consulta DAX.

Sintaxis

DAX

```
EVALUATE <table>
```

Parámetros

 Expandir tabla

Término	Definición
tabla	Una expresión de tabla.

Valor devuelto

El resultado de una expresión de tabla.

Notas

- Una consulta DAX puede contener varias instrucciones EVALUATE.
- Para obtener más información sobre cómo se usan las instrucciones EVALUATE, consulte [Consultas DAX](#).

Ejemplo

DAX

```
EVALUATE  
    'Internet Sales'
```

Devuelve todas las filas y columnas de la tabla Ventas por Internet, como una tabla.

Contenido relacionado

[ORDER BY](#)

[START AT](#)

[DEFINE](#)

[VAR](#)

[Consultas DAX](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

MEASURE

Artículo • 14/12/2023

Presenta una definición de medida en una instrucción DEFINE de una consulta DAX.

Sintaxis

DAX

```
[DEFINE
(
    MEASURE <table name>[<measure name>] = <scalar expression>
) +
]

(EVALUATE <table expression>) +
```

Parámetros

 Expandir tabla

Término	Definición
Nombre de la tabla	Nombre de una tabla que contiene la medida.
nombre de medida	Nombre de la medida. No puede ser una expresión. El nombre no tiene que ser único. El nombre existe únicamente lo que dura la consulta.
expresión escalar	Expresión DAX que devuelve un valor escalar.

Valor devuelto

Resultado calculado de la expresión de medida.

Observaciones

- Las definiciones de medida de una consulta invalidan las medidas del modelo del mismo nombre mientras dura la consulta. No afectarán a la medida del modelo.

- La expresión de medida se puede usar con cualquier otra expresión de la misma consulta.
- Para obtener más información sobre cómo se usan las instrucciones MEASURE, consulte [Consultas DAX](#).

Contenido relacionado

[DEFINE](#)

[EVALUATE](#)

[VAR](#)

[Consultas DAX](#)

Comentarios

¿Le ha resultado útil esta página?



ORDER BY

Artículo • 14/12/2023

Introduce una instrucción que define el criterio de ordenación de los resultados de la consulta devueltos por una instrucción EVALUATE en una [consulta DAX](#).

Sintaxis

DAX

```
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]]
```

Parámetros

 Expandir tabla

Término	Definición
expression	Cualquier expresión DAX que devuelve un valor escalar único.
ASC	(Predeterminado) Criterio de ordenación ascendente.
DESC	Criterio de ordenación descendente.

Valor devuelto

El resultado de una instrucción EVALUATE en orden ascendente (ASC) o descendente (DESC).

Observaciones

Para obtener más información sobre cómo se usan las instrucciones ORDER BY, vea [Consultas DAX](#).

Contenido relacionado

[START AT](#)

[EVALUATE](#)

[VAR](#)

Comentarios

¿Le ha resultado útil esta página?

 Sí

 No

START AT

Artículo • 20/10/2023

Presenta una instrucción que define el valor inicial en el que se devuelven los resultados de la consulta de una cláusula ORDER BY en una instrucción EVALUATE de una [consulta DAX](#).

Sintaxis

DAX

```
[START AT {<value>|<parameter>} [, ...]]
```

Parámetros

Término	Definición
valor	Valor constante No puede ser una expresión.
parameter	Nombre de un parámetro en una instrucción XMLA con el carácter @ como prefijo.

Observaciones

- Los argumentos START AT tienen una correspondencia uno a uno con las columnas de la instrucción ORDER BY. En la instrucción START AT puede haber tantos argumentos como en la instrucción ORDER BY, pero no más. El primer argumento de la instrucción START AT define el valor inicial de la columna 1 de las columnas ORDER BY. El segundo argumento de la instrucción START AT define el valor inicial de la columna 2 de las columnas ORDER BY dentro de las filas que cumplen el primer valor de la columna 1.
- Para obtener más información sobre cómo se usan las instrucciones START AT, vea [Consultas DAX](#).

Vea también

[ORDER BY](#)

[EVALUATE](#)

[VAR](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

VAR

Artículo • 20/10/2023

Almacena el resultado de una expresión como una variable con nombre que después se puede pasar como argumento a otras expresiones de medida. Una vez que se han calculado los valores resultantes de una expresión variable, esos valores no cambian, aunque se haga referencia a la variable en otra expresión.

Sintaxis

DAX

```
VAR <name> = <expression>
```

Parámetros

Término	Definición
name	Nombre de la variable (identificador). No se admiten delimitadores. Por ejemplo, "varName" o [varName] da lugar a un error. Juego de caracteres admitido: a-z, A-Z, 0-9. 0-9 no son válidos como primer carácter. <u>_</u> (doble subrayado) se permite como prefijo del nombre del identificador. No se admiten otros caracteres especiales. No se permiten palabras clave reservadas. No se permiten nombres de tablas existentes. No se permiten espacios vacíos.
expression	Expresión DAX que devuelve un valor escalar o de tabla.

Valor devuelto

Variable con nombre que contiene el resultado del argumento de expresión.

Comentarios

- Una expresión pasada como argumento a VAR puede contener otra declaración VAR.
- Al hacer referencia a una variable:

- Las medidas no pueden hacer referencia a variables definidas fuera de la expresión de medida, pero pueden hacer referencia a variables de ámbito funcional definidas dentro de la expresión.
 - Las variables pueden hacer referencia a medidas.
 - Las variables pueden hacer referencia a variables definidas previamente.
 - No se puede hacer referencia a columnas de variables de tabla mediante la sintaxis TableName[ColumnName].
- Para conocer los procedimientos recomendados al usar VAR, consulte [Uso de variables para mejorar las fórmulas DAX](#).
 - Para obtener más información sobre cómo se usa VAR dentro de una consulta DAX, consulte [Consultas DAX](#).

Ejemplo

Para calcular un porcentaje de crecimiento interanual sin usar una variable, puede crear tres medidas independientes. Esta primera medida calcula Sum of Sales Amount:

DAX

```
Sum of SalesAmount = SUM(SalesTable[SalesAmount])
```

Una segunda medida calcula el importe de ventas del año anterior:

DAX

```
SalesAmount PreviousYear =
CALCULATE([Sum of SalesAmount],
SAMEPERIODLASTYEAR(Calendar[Date]))
)
```

Luego puede crear una tercera medida que combine las otras dos medidas a fin de calcular un porcentaje de crecimiento. Observe que la medida Sum of SalesAmount se usa en dos lugares: primero para determinar si hay una venta y luego para calcular un porcentaje.

DAX

```
Sum of SalesAmount YoY%: =
IF([Sum of SalesAmount] ,
DIVIDE(([Sum of SalesAmount] - [SalesAmount PreviousYear]), [Sum of
```

```
SalesAmount])  
)
```

Mediante una variable, puede crear una única medida que calcule el mismo resultado:

DAX

```
YoY% = VAR Sales = SUM(SalesTable[SalesAmount])  
  
VAR SalesLastYear =  
    CALCULATE ( SUM ( SalesTable[SalesAmount] ), SAMEPERIODLASTYEAR ( 'Calendar'[Date] ) )  
  
return if(Sales, DIVIDE(Sales - SalesLastYear, Sales))
```

Mediante una variable, puede obtener el mismo resultado, pero de forma más legible. Y dado que el resultado de la expresión se almacena en la variable, el rendimiento de la medida se puede mejorar significativamente porque no tiene que volver a calcularse cada vez que se usa.

Consulte también

[Uso de variables para mejorar las fórmulas DAX](#)

[Consultas DAX](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

Glosario de DAX

Artículo • 20/10/2023

Consulta analítica

Los objetos visuales de Power BI consultan un modelo de datos mediante una *consulta analítica*. Una consulta analítica se esfuerza por reducir los volúmenes de datos potencialmente grandes y las complejidades del modelo mediante tres fases distintas: Filtrado, agrupación y resumen. Una consulta analítica se crea automáticamente cuando se asignan campos a los cuadros de los objetos visuales del informe. Los autores de informes pueden controlar el comportamiento de las asignaciones de campo mediante el cambio del nombre de los campos, la modificación de la técnica de resumen o la deshabilitación del resumen para alcanzar la agrupación. En el momento de diseñar el informe, los filtros se pueden agregar al informe, a una página del informe o a un objeto visual. En la vista de lectura, los filtros se pueden modificar en el panel **Filtros** o mediante interacciones con segmentaciones y otros objetos visuales (filtrado cruzado).

BLANK

DAX define la ausencia de un valor como BLANK. Es el equivalente a NULL de SQL, pero no se comporta exactamente igual. Se alinea más estrechamente con Excel y la forma de definir una celda vacía. BLANK se evalúa como cero o como una cadena vacía cuando se combina con otras operaciones. Por ejemplo, $\text{BLANK} + 20 = 20$. Use siempre letras mayúsculas; el plural es BLANKs, con una "s" minúscula.

Columna calculada

Un cálculo del modelo que se usa para agregar una columna a un modelo tabular mediante la escritura de una fórmula DAX. La fórmula debe devolver un valor escalar y se evalúa para cada fila de la tabla. Se puede agregar una columna calculada a una tabla de modo de almacenamiento DirectQuery o de importación.

Medida calculada

En los modelos tabulares, no existe el concepto de *medida calculada*. En su lugar, use *medida*. La palabra *calculada* se utiliza para describir las tablas calculadas y las columnas calculadas. Los distingue de las tablas y columnas que se originan en Power Query. Power Query no tiene el concepto de una medida.

Tabla calculada

Un cálculo del modelo que se usa para agregar una tabla a un modelo tabular mediante la escritura de una fórmula DAX. La fórmula debe devolver un objeto de tabla. Da como resultado una tabla que usa el modo de almacenamiento de importación.

Cálculo

Un proceso deliberado que transforma una o más entradas en uno o más resultados. En un modelo de datos tabular, un cálculo puede ser un objeto de modelo, ya sea una tabla calculada, una columna calculada o una medida.

Context

Describe el entorno en el que se evalúa una fórmula DAX. Existen dos tipos de contexto: *Contexto de fila* y *contexto de filtro*. El contexto de fila representa la "fila actual", y se utiliza para evaluar las fórmulas y expresiones de columna calculada que usan los iteradores de la tabla. El contexto de filtro se usa para evaluar las medidas, y representa los filtros que se aplican directamente a las columnas y filtros del modelo que propagan las relaciones del modelo.

DAX

El lenguaje Expresiones de análisis de datos (DAX) es un lenguaje de fórmulas para Power Pivot en Excel, Power BI, Azure Analysis Services y el modelado tabular en SQL Server Analysis Services. También se puede usar DAX para agregar cálculos del modelo de datos y definir reglas de seguridad de nivel de fila (RLS).

Seguridad dinámica

Cuándo se aplican las reglas de seguridad de nivel de fila (RLS) mediante la identidad del usuario del informe. Las reglas filtran las tablas del modelo mediante el nombre de cuenta del usuario. Esto se puede realizar con las funciones USERNAME o USERPRINCIPALNAME. Vea [Seguridad de nivel de fila](#).

Expression

Una unidad de lógica de DAX que se evalúa y devuelve un resultado. Las expresiones pueden declarar variables, en cuyo caso se les asigna una subexpresión y deben incluir

una instrucción RETURN que genere una expresión final. Las expresiones se construyen mediante objetos del modelo (tablas, columnas o medidas), funciones, operadores o constantes.

Campo

Recurso del modelo de datos que se presenta en el panel **Campos**. Los campos se usan para configurar los filtros y los objetos visuales del informe. Los campos se componen de columnas de modelo, niveles de jerarquía y medidas.

Fórmula

Una o más expresiones DAX que se usan para definir un cálculo de modelo. Las expresiones internas se denominan subexpresiones. El plural es *fórmulas*.

Función

Las funciones DAX tienen argumentos que permiten pasar parámetros. Las fórmulas pueden usar muchas llamadas de función, posiblemente mediante el anidamiento de funciones dentro de otras funciones. En una fórmula, los nombres de función deben ir seguidos de paréntesis. Dentro de los paréntesis los parámetros se pasan.

Medida implícita

Un cálculo generado automáticamente que se logra mediante la configuración de un objeto visual de Power BI para resumir los valores de columna. Las columnas **numéricas** admiten la mayor variedad de resúmenes, entre los que se incluyen los siguientes: Suma, promedio, mínimo, máximo, recuento (distinto), recuento, desviación estándar, varianza o mediana. También se pueden resumir las columnas de otros tipos de datos. Las columnas de **texto** se pueden resumir mediante lo siguiente: Primero (alfabéticamente), último (alfabéticamente), recuento (distinto) o recuento. Las columnas de **fechas** se pueden resumir mediante lo siguiente: Más antiguo, más reciente, recuento (distinto) o recuento. Las columnas **booleanas** se pueden resumir mediante lo siguiente: Recuento (distinto) o recuento.

Función de iterador

Función DAX que enumera todas las filas de una tabla determinada y evalúa una expresión determinada para cada fila. Proporciona flexibilidad y control sobre cómo los

cálculos del modelo resumen los datos.

MDX.

El lenguaje Expresiones multidimensionales (MDX) es un lenguaje de fórmulas para los modelos multidimensionales de SQL Server Analysis Services (también conocidos como *cubos*). MDX se puede usar para consultar modelos tabulares, pero no puede definir medidas implícitas. Solo puede consultar medidas que ya están definidas en el modelo.

Measure

Un cálculo que alcanza el resumen. Las medidas son *implícitas* o *explícitas*. Una medida explícita es un cálculo que se agrega a un modelo de datos tabular mediante la escritura de una fórmula DAX. Una fórmula de medida debe devolver un valor escalar. En el panel **Campos**, las medidas explícitas se indican con un icono de calculadora. Las medidas explícitas son necesarias cuando el modelo se consulta mediante Expresiones multidimensionales (MDX), como sucede cuando se usa Analizar en Excel. Normalmente, a una medida explícita se le llama simplemente medida.

Grupo de medida

Una tabla de modelo que contiene al menos una medida y no tiene jerarquías ni columnas visibles. En el panel **Campos**, cada grupo de medida se indica con un icono de varias calculadoras. Los grupos de medida se muestran juntos en la parte superior del panel **Campos** y se ordenan alfabéticamente por nombre.

Cálculo del modelo

Una fórmula con nombre que se usa para agregar una tabla calculada, una columna calculada o una medida a un modelo de datos tabular. Su estructura es <NAME> = <FORMULA>. Los modeladores de datos agregan la mayoría de cálculos en Power BI Desktop, pero también se pueden agregar medidas a un informe de conexión dinámica. Vea [Medidas de informe](#).

Medidas rápidas

Una característica de Power BI Desktop que elimina la necesidad de escribir fórmulas DAX para las medidas definidas habitualmente. Las medidas rápidas incluyen el promedio por categoría, el rango y la diferencia de la línea base.

Medidas de informe

También se denominan *medidas de nivel de informe*. Se agregan a un informe de conexión dinámica en Power BI Desktop con la escritura de una fórmula DAX, pero solo para las conexiones a modelos de Power BI o a modelos tabulares de Analysis Services.

Seguridad de nivel de fila

También se denomina *RLS*. Técnica de diseño para restringir el acceso a subconjuntos de datos para usuarios específicos. En un modelo tabular, esto se logra mediante la creación de roles de modelo. Los roles tienen reglas, que son expresiones DAX para filtrar filas de tabla.

Escalar

En DAX, un escalar es un valor único. Un escalar puede ser de cualquier tipo de datos: Decimal, Integer, DateTime, String, Currency, Boolean. Un valor escalar puede ser el resultado de una expresión calculada a partir de varios valores. Por ejemplo, una función de agregación como MAX() devuelve un único valor máximo de un conjunto de valores a partir del cual se va a evaluar.

Resumen

Una operación aplicada a los valores de una columna. Vea [medida](#).

Inteligencia de tiempo

La inteligencia de tiempo está relacionada con los cálculos a lo largo del tiempo, como el año hasta la fecha (YTD).

Función de inteligencia de tiempo

DAX incluye una gran variedad de funciones de inteligencia de tiempo. Cada función de inteligencia de tiempo obtiene su resultado mediante la modificación del contexto de filtro para los filtros de fecha. Funciones de ejemplo: TOTALYTD y SAMEPERIODLASTYEAR.

Valor, valores

Datos que se visualizarán.

Parámetro What-if

Una característica de Power BI Desktop que proporciona la capacidad de aceptar la entrada del usuario a través de segmentaciones. Cada parámetro crea una tabla calculada de una sola columna y una medida que devuelve un valor de selección única. La medida se puede usar en los cálculos del modelo para responder a la entrada del usuario.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

Operadores de DAX

Artículo • 20/10/2023

El lenguaje DAX (Expresiones de análisis de datos) usa operadores para crear expresiones que comparan valores, realizan cálculos aritméticos o trabajan con cadenas.

Tipos de operadores

Existen cuatro tipos distintos de operadores de cálculo: aritméticos, de comparación, de concatenación de texto y lógicos.

Operadores aritméticos

Use los siguientes operadores aritméticos para realizar operaciones matemáticas básicas (como sumas, restas o multiplicaciones), para combinar números y para producir resultados numéricos.

Operadores aritméticos	Significado	Ejemplo
+ (signo más)	Suma	3+3
- (signo menos)	Resta o inversión del signo	3-1-1
* (asterisco)	Multiplicación	3*3
/ (barra diagonal)	División	3/3
^ (símbolo de intercalación)	Exponenciación	16^4

ⓘ Nota

El signo más puede funcionar como *operador binario* y como *operador unario*. Un operador binario requiere números en ambos lados del operador y realiza la suma. Cuando se usan valores en una fórmula DAX a ambos lados del operador binario, DAX intenta convertir los valores a tipos de datos numéricos si aún no son números. En cambio, el operador unario se puede aplicar a cualquier tipo de argumento. El signo más no afecta al tipo o al valor y simplemente se pasa por alto, mientras que el operador menos crea un valor negativo, si se aplica a un valor numérico.

Operadores de comparación

Puede comparar dos valores con los siguientes operadores. Cuando se comparan dos valores con estos operadores, el resultado es un valor lógico, ya sea TRUE o FALSE.

Operadores de comparación	Significado	Ejemplo
=	Igual a	[Region] = "USA"
==	Estrictamente igual a	[Region] == "USA"
>	Mayor que	[Sales Date] > "Jan 2009"
<	Menor que	[Sales Date] < "Jan 1 2009"
>=	Mayor o igual que	[Amount] >= 20000
<=	Menor o igual que	[Amount] <= 100
<>	No igual a	[Region] <> "USA"

Todos los operadores de comparación, excepto ==, tratan BLANK como equivalente al número 0, una cadena vacía "", DATE(1899, 12, 30), o FALSE. Como resultado, [Column] = 0 será true cuando el valor de [Column] sea 0 o BLANK. Por el contrario, [Column] == 0 solo es true cuando el valor de [Column] es 0.

Operador de concatenación de texto

Use la "Y comercial" (&) para unir, o concatenar, dos o más cadenas de texto para generar un único fragmento de texto.

Operador de texto	Significado	Ejemplo
& ("Y comercial")	Conecta, o concatena, dos valores para generar un valor de texto continuo.	[Region] & ", " & [City]

Operadores lógicos

Use los operadores lógicos (&&) y (||) para combinar expresiones para generar un solo resultado.

Operador de texto	Significado	Ejemplos
&& (doble Y comercial)	Crea una condición AND entre dos expresiones que tienen ambas un resultado booleano. Si ambas expresiones devuelven TRUE, la combinación de las	([Region] = "France") && ([BikeBuyer] = "yes"))

Operador de texto	Significado	Ejemplos
	expresiones también devuelven TRUE; de lo contrario, la combinación devuelve FALSE.	
(doble barra vertical)	Crea una condición OR entre dos expresiones lógicas. Si alguna de las expresiones devuelve TRUE, el resultado es TRUE; solo cuando ambas expresiones son FALSE, el resultado es FALSE.	(([Region] = "France") ([BikeBuyer] = "yes"))
IN	Crea una condición OR lógica entre cada fila que se compara con una tabla. Nota: En la sintaxis del constructor de tabla se emplean llaves.	'Product'[Color] IN { "Red", "Blue", "Black" }

Operadores y orden de precedencia

En algunos casos, el orden en el que se realiza el cálculo puede afectar al valor devuelto, de ahí que sea importante comprender cómo se determina el orden y cómo este se puede cambiar para obtener los resultados deseados.

Orden del cálculo

Una expresión evalúa los operadores y los valores en un orden específico. Todas las expresiones comienzan siempre con un signo igual (=). Este signo igual indica que los caracteres tras él constituyen una expresión.

Después del signo igual están los elementos que se van a calcular (los operandos), que se separan con operadores de cálculo. Las expresiones siempre se leen de izquierda a derecha, pero el orden en el que los elementos se agrupan se puede controlar en cierto grado mediante el uso de paréntesis.

Prioridad de los operadores

Si se combinan varios operadores en una misma fórmula, las operaciones se ordenan según la siguiente tabla. Si los operadores tienen el mismo valor de precedencia, se ordenan de izquierda a derecha. Por ejemplo, si una expresión contiene un operador de multiplicación y división, se evalúan en el orden en que aparecen en la expresión, de izquierda a derecha.

Operador	Descripción
^	Exponenciación

Operador	Descripción
-	Inversión del signo (como en -1)
* y /	Multiplicación y división
+ y -	Suma y resta
&	Conecta dos cadenas de texto (concatenación)
=, ==, <, >, <=, >=, <>, IN	De comparación
NOT	NOT (operador unario)

Uso de paréntesis para controlar el orden del cálculo

Para cambiar el orden de evaluación, debe incluir entre paréntesis la parte de la fórmula que deba calcularse en primer lugar. Por ejemplo, la siguiente fórmula da como resultado 11 porque la multiplicación se calcula antes que la suma. La fórmula multiplica 2 por 3 y, tras ello, suma 5 al resultado.

DAX

```
=5+2*3
```

Por el contrario, si se usan paréntesis para cambiar la sintaxis, el orden se cambia de modo que 5 y 2 se suman y el resultado de esto se multiplica por 3, dando como resultado 21.

DAX

```
=(5+2)*3
```

En el siguiente ejemplo, los paréntesis que rodean la primera parte de la fórmula hacen que el cálculo evalúe la expresión $(3 + 0.25)$ en primer lugar y, luego, divida el resultado entre el resultado de la expresión $(3 - 0.25)$.

DAX

```
=(3 + 0.25)/(3 - 0.25)
```

En el siguiente ejemplo, el operador de exponentiación se aplica en primer lugar, según las reglas de precedencia de operadores, y, tras ello, se aplica el operador de inversión del signo. El resultado de esta expresión es -4.

```
DAX
```

```
--2^2
```

Para garantizar que el operador de inversión del signo se aplica primero al valor numérico, se pueden usar paréntesis para controlar los operadores, como se muestra en el siguiente ejemplo. El resultado de esta expresión es 4.

```
DAX
```

```
= (-2)^2
```

Compatibilidad

DAX controla y compara fácilmente varios tipos de datos, de forma muy similar a como lo hace Microsoft Excel, pero el motor de cálculo subyacente se basa en SQL Server Analysis Services, con lo cual proporciona las características avanzadas adicionales de los almacenes de datos relacionales, incluida una mayor compatibilidad con los tipos de fecha y hora. Por lo tanto, es posible que, en algunos casos, los resultados de los cálculos o el comportamiento de las funciones no sean los mismos que en Excel. Además, DAX admite más tipos de datos que Excel. En esta sección se describen las principales diferencias.

Coerción de tipos de datos de operandos

En general, los dos operandos de los lados izquierdo y derecho de cualquier operador deben ser del mismo tipo de datos. Si los tipos de datos son diferentes, DAX los convertirá a un tipo de datos común para aplicar el operador en algunos casos:

1. Ambos operandos se convierten al tipo de datos común más grande posible.
2. El operador se aplica, si es posible.

Por ejemplo, supongamos que tiene dos números que quiere combinar. Un número es el resultado de una fórmula, como = [Price] * .20, y dicho resultado puede contener muchos decimales, mientras que el otro número es un entero que se ha proporcionado como valor de cadena.

En este caso, DAX convertirá ambos números en números reales en formato numérico, usando el formato numérico más grande que pueda almacenar ambos tipos de números. Tras ello, DAX aplicará la multiplicación.

Dependiendo de la combinación de tipos de datos, es posible que la coerción de tipos no pueda aplicarse a las operaciones de comparación. Para obtener una lista completa de los tipos de datos admitidos en DAX, consulte [Tipos de datos admitidos en los modelos tabulares](#) y [Tipos de datos en Power BI Desktop](#).

Los tipos Integer, Real Number, Currency, Date/time y Blank se consideran numéricos con fines de comparación. Blank evalúa como cero cuando se realiza una comparación. En las operaciones de comparación se pueden usar las siguientes combinaciones de tipos de datos.

Tipo de datos del lado izquierdo	Tipo de datos del lado derecho
Numérica	Numérica
Booleano	Boolean
String	Cadena

Cualquier otra combinación de comparaciones de tipos de datos devolverá un error. Por ejemplo, una fórmula como `= "1" > 0` devuelve un error que indica que *las operaciones de comparación DAX no permiten la comparación de valores de tipo Text con valores de tipo Integer*.

Tipos de datos usados en DAX	Tipos de datos usados en Excel
Números (I8, R8)	Números (R8)
Cadena	Cadena
Booleano	Boolean
DateTime	Variante
Moneda	Moneda

Diferencias en el orden de precedencia

El orden de precedencia de las operaciones en las fórmulas DAX es básicamente el mismo que el de Microsoft Excel, pero algunos operadores de Excel no se admiten, como, por ejemplo, el de porcentaje. Tampoco se admiten intervalos.

Por lo tanto, siempre que copie y pegue fórmulas de Excel, asegúrese de revisarla cuidadosamente, ya que es posible que algunos operadores o elementos de las fórmulas no sean válidos. Cuando tenga alguna duda sobre el orden en el que se

realizan las operaciones, se recomienda usar paréntesis para controlar el orden de las operaciones y acabar con cualquier ambigüedad en el resultado.

Vea también

[Sintaxis DAX](#)

[Nomenclatura de parámetros DAX](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

Consultas DAX

Artículo • 20/10/2023

Los clientes de informes como Power BI y Excel ejecutan consultas DAX cada vez que se coloca un campo en un informe o cuando se aplica un filtro. Mediante [SQL Server Management Studio](#) (SSMS), el [Generador de informes de Power BI](#) y herramientas de código abierto como [DAX Studio](#), puede crear y ejecutar sus propias consultas DAX. Las consultas DAX devuelven resultados como una tabla directamente dentro de la herramienta, lo que le permite crear y probar rápidamente el rendimiento de las fórmulas DAX.

Antes de aprender sobre las consultas, es importante tener un conocimiento sólido de los aspectos básicos de DAX. Si aún no lo ha hecho, asegúrese de consultar [Información general sobre DAX](#).

Palabras clave

Las consultas DAX tienen una sintaxis simple formada por una sola palabra clave necesaria, EVALUATE, y varias palabras clave opcionales: ORDER BY, START AT, DEFINE, MEASURE, VAR, TABLE y COLUMN. Cada palabra clave define una instrucción usada durante la consulta.

EVALUATE (necesario)

En el nivel más básico, una consulta DAX es una instrucción EVALUATE que contiene una expresión de tabla. Se requiere al menos una instrucción EVALUATE, pero una consulta puede contener cualquier número de instrucciones EVALUATE.

Sintaxis de EVALUATE

DAX
EVALUATE <table>

Parámetros de EVALUATE

Término	Definición
tabla	Expresión de tabla.

Ejemplo de EVALUATE

```
DAX

EVALUATE
    'Internet Sales'
```

Devuelve todas las filas y columnas de la tabla Ventas por Internet, como una tabla.

The screenshot shows the Microsoft Power BI DAX Query Editor interface. The query 'MSDAXQuery1.msdax' contains the DAX code: 'EVALUATE('Internet Sales')'. The results pane displays a table with multiple columns, each labeled 'Internet Sales[...]' followed by a truncated column name. The data consists of several rows of numerical values. A message bar at the bottom indicates 'Query executed successfully.' and provides connection information: 'asazure://westus.asazure.wi...', 'adventureworks', and '00:00:08'.

ORDER BY (opcional)

La palabra clave opcional **ORDER BY** define una o varias expresiones que se usan para ordenar los resultados de la consulta. Cualquier expresión que se puede evaluar para cada fila del resultado es válida.

Sintaxis de ORDER BY

```
DAX

EVALUATE <table>
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]]
```

Parámetros de ORDER BY

Término	Definición
expression	Cualquier expresión DAX que devuelve un valor escalar único.

Término	Definición
ASC	(Predeterminado) Criterio de ordenación ascendente.
DESC	Criterio de ordenación descendente.

Ejemplo de ORDER BY

```
DAX

EVALUATE
    'Internet Sales'

ORDER BY
    'Internet Sales'[Order Date]
```

Devuelve todas las filas y columnas de la tabla Ventas por Internet, en orden ascendente por Fecha de pedido, como una tabla.

The screenshot shows the Microsoft Power BI Data Editor interface. The top navigation bar includes File, Edit, View, Query, Project, Debug, Tools, Window, Help, and a Quick Launch search bar. Below the menu is a toolbar with various icons for file operations like New Query, MDX, DMX, XSLA, DAX, and others. The main workspace is titled 'MSDAXQuery1.msdax...' and contains the DAX code:

```
EVALUATE(
    'Internet Sales'
)
ORDER BY
    'Internet Sales'[Order Date]
```

Below the code is a results grid showing data from the 'Internet Sales' table. The columns are: Internet Sales[Sales], Internet Sales[Tax Rate], Internet Sales[Freight], Internet Sales[Carrier], Internet Sales[Customer ID], Internet Sales[Order Date], Internet Sales[Due Date], Internet Sales[Ship Date], and Internet Sales[Status]. The data is sorted by Order Date in ascending order. A status bar at the bottom indicates 'Query executed successfully.' and shows connection details: 'asazure://westus.asazure.wi...', 'adventureworks', and '00:00:08'.

START AT (opcional)

La palabra clave opcional **START AT** se usa dentro de una cláusula **ORDER BY**. Define el valor en el que comienzan los resultados de la consulta.

Sintaxis de START AT

```
DAX
```

```
EVALUATE <table>
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]
[START AT {<value>|<parameter>} [, ...]]]
```

Parámetros de START AT

Término	Definición
valor	Valor constante No puede ser una expresión.
parameter	Nombre de un parámetro en una instrucción XMLA con el carácter @ como prefijo.

Comentarios sobre START AT

Los argumentos START AT tienen una correspondencia uno a uno con las columnas de la cláusula ORDER BY. En la cláusula START AT puede haber tantos argumentos como en la cláusula ORDER BY, pero no más. El primer argumento START AT define el valor inicial de la columna 1 de las columnas ORDER BY. El segundo argumento START AT define el valor inicial de la columna 2 de las columnas ORDER BY dentro de las filas que cumplen el primer valor de la columna 1.

Ejemplo de START AT

DAX

```
EVALUATE
    'Internet Sales'

    ORDER BY
        'Internet Sales'[Sales Order Number]
    START AT "SO7000"
```

Devuelve todas las filas y columnas de la tabla Internet Sales, en orden ascendente por Número de pedido de venta, empezando en SO7000.

The screenshot shows the Microsoft Power BI Data Editor interface. The top menu bar includes File, Edit, View, Query, Project, Debug, Tools, Window, and Help. The ribbon tabs are New Query, DAX, XMLA, DAX, and DAX. The main area displays a DAX query:

```
EVALUATE
    'Internet Sales'
)
ORDER BY
    'Internet Sales'[Sales Order Number]
START AT "S07000"
```

The results pane shows a table with columns: Internet Sales[Pr...], Internet Sales[C...], Internet Sales[Pr...], Internet Sales[C...], Internet Sales[S...], Internet Sales[S...], Internet Sales[S...], Internet Sales[R...], and Internet Sales[O...]. The data rows are:

Internet Sales[Pr...]	Internet Sales[C...]	Internet Sales[Pr...]	Internet Sales[C...]	Internet Sales[S...]	Internet Sales[S...]	Internet Sales[S...]	Internet Sales[R...]	Internet Sales[O...]
604	28153	1	19	6	S070000	1	1	1
538	28153	1	19	6	S070000	2	1	1
225	28153	1	19	6	S070000	3	1	1
222	21797	1	100	1	S070001	2	1	1
384	21797	1	100	1	S070001	1	1	1

At the bottom, a message says "Query executed successfully." and shows connection details: "asazure://westus.asazure.wi..." and "adventureworks | 00:00:03". The status bar indicates "Ready", "Ln 6", "Col 18", "Ch 18", and "INS".

DEFINE (opcional)

La palabra clave opcional **DEFINE** presenta una o varias definiciones de entidad calculadas que solo existen durante la consulta. Las definiciones preceden a la instrucción **EVALUATE** y son válidas para todas las instrucciones **EVALUATE** de la consulta. Las definiciones pueden ser variables, medidas, tablas¹ y columnas¹. Las definiciones pueden hacer referencia a otras definiciones que aparecen antes o después de la definición actual. Se requiere al menos una definición si la palabra clave **DEFINE** se incluye en una consulta.

Sintaxis de DEFINE

DAX

```
[DEFINE
(
    (MEASURE <table name>[<measure name>] = <scalar expression>) |
    (VAR <var name> = <table or scalar expression>) |
    (TABLE <table name> = <table expression>) |
    (COLUMN <table name>[<column name>] = <scalar expression>) |
)
+
]

(EVALUATE <table expression>) +
```

Parámetros de DEFINE

Término	Definición
Entidad	MEASURE, VAR, TABLE ¹ o COLUMN ¹ .

Término	Definición
name	Nombre de una definición de medida, variable, tabla o columna. No puede ser una expresión. El nombre no tiene que ser único. El nombre existe únicamente lo que dura la consulta.
expression	Cualquier expresión DAX que devuelve un valor de tabla o escalar. La expresión puede usar cualquiera de las entidades definidas. Si es necesario convertir una expresión escalar en una expresión de tabla, encapsule la expresión dentro de un constructor de tabla con llaves {} o use la función <code>ROW()</code> para devolver una tabla de una sola fila.

[1] **Precaución:** las definiciones TABLE y COLUMN con ámbito de consulta están pensadas solo para uso interno. Aunque puede definir expresiones TABLE y COLUMN para una consulta sin errores de sintaxis, pueden producir errores en tiempo de ejecución, por lo que no se recomienda.

Comentarios sobre DEFINE

- Una consulta DAX puede tener varias instrucciones EVALUATE, pero solo puede tener una instrucción DEFINE. Las definiciones de la instrucción DEFINE se pueden aplicar a cualquier instrucción EVALUATE de la consulta.
- Se requiere al menos una definición en una instrucción DEFINE.
- Las definiciones de medida de una consulta invalidan las medidas del modelo con el mismo nombre, pero solo se usan dentro de la consulta. No afectarán a la medida del modelo.
- Los nombres VAR tienen restricciones únicas. Para obtener más información, consulte [VAR: parámetros](#).

Ejemplo de DEFINE

```
DAX

DEFINE
    MEASURE 'Internet Sales'[Internet Total Sales] =
        SUM ( 'Internet Sales'[Sales Amount] )

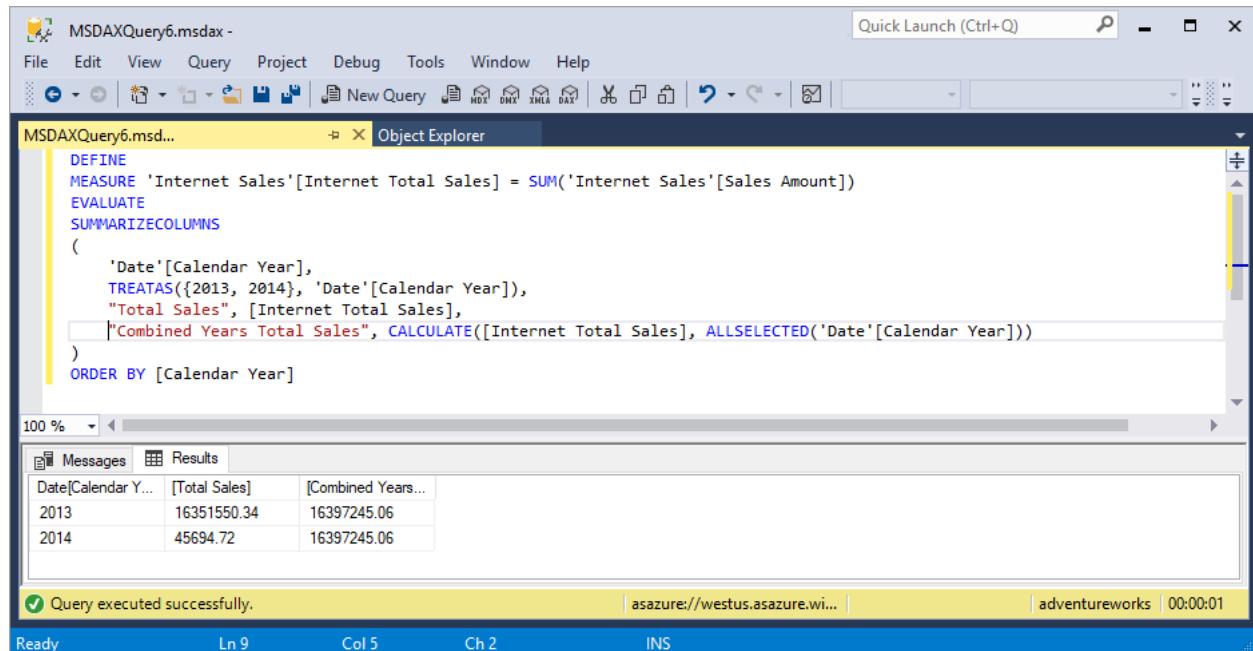
EVALUATE
SUMMARIZECOLUMNS (
    'Date'[Calendar Year],
    TREATAS (
        {
            2013,
            2014
        },
        'Date'[Calendar Year]
    )
)
ORDER BY 'Date'[Calendar Year] DESC
```

```

'Date'[Calendar Year]
),
"Total Sales", [Internet Total Sales],
"Combined Years Total Sales",
CALCULATE (
    [Internet Total Sales],
    ALLSELECTED ( 'Date'[Calendar Year] )
)
)
ORDER BY [Calendar Year]

```

Devuelve como una tabla las ventas totales calculadas para los años 2013 y 2014, y las ventas totales calculadas combinadas para los años 2013 y 2014. La medida de la instrucción DEFINE, Internet Total Sales, se usa tanto en las expresiones Total Sales como en Combined Years Total Sales.



```

MSDAXQuery6.msdax - Quick Launch (Ctrl+Q) 
File Edit View Query Project Debug Tools Window Help 
File New Query MDX DAX XMLA DAX 
MSDAXQuery6.msd... Object Explorer 
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]

100 % 
Messages Results 
Date[Calendar Y... [Total Sales] [Combined Years...
2013 16351550.34 16397245.06 
2014 45694.72 16397245.06 

Query executed successfully. | asazure://westus.asazure.wi... | adventureworks | 00:00:01 
Ready Ln 9 Col 5 Ch 2 INS 

```

Parámetros en consultas DAX

Se puede parametrizar una instrucción de consulta DAX bien definida y, después, usarla una y otra vez con tan solo realizar cambios en los valores de parámetro.

El [Método Execute \(XMLA\)](#) tiene un elemento de la colección [Elemento de parámetros \(XMLA\)](#) que permite definir parámetros y asignarles un valor. Dentro de la colección, cada elemento de [Elemento de parámetros \(XMLA\)](#) define el nombre del parámetro y un valor.

Para hacer referencia a los parámetros XMLA, debe añadir un carácter @ como prefijo al nombre del parámetro. En cualquier lugar de la sintaxis donde se permita un valor, este

se podrá reemplazar por una llamada de parámetro. Todos los parámetros XMLA se escriben como texto.

Importante

Los parámetros definidos en la sección parámetros y que no se usan en el elemento <STATEMENT> generan una respuesta de error en XMLA. Los parámetros usados y no definidos en el elemento <Parameters> generan una respuesta de error en XMLA.

Vea también

[Instrucciones DAX](#)

[SUMMARIZECOLUMNS](#)

[TREATAS](#)

[FILTER](#)

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

Convenciones de nomenclatura de parámetros DAX

Artículo • 20/10/2023

Los nombres de parámetro se normalizan en la referencia DAX para facilitar el uso y la comprensión de las funciones.

Nombres de parámetro:

Término	Definición
expression	Cualquier expresión DAX que devuelve un único valor escalar, donde la expresión se va a evaluar varias veces (para cada fila o contexto).
value	Cualquier expresión DAX que devuelve un único valor escalar, donde la expresión se va a evaluar exactamente una vez antes de todas las demás operaciones.
tabla	Cualquier expresión DAX que devuelve una tabla de datos.
tableName	Nombre de una tabla existente que usa la sintaxis DAX estándar. No puede ser una expresión.
columnName	Nombre de una columna existente, con la sintaxis DAX estándar, normalmente completa. No puede ser una expresión.
name	Constante de cadena que se utilizará para proporcionar el nombre de un objeto nuevo.
orden	Enumeración que se usa para determinar el criterio de ordenación.
ties	Enumeración que se usa para determinar el control de los valores de empate.
tipo	Enumeración que se usa para determinar el tipo de datos de PathItem y PathItemReverse.

Inclusión de prefijos en nombres de parámetros o uso de solo el prefijo

Término	Definición
prefixing	Los nombres de los parámetros se pueden calificar aún más con un prefijo que describa cómo se usa el argumento y cómo evitar la lectura ambigua de los parámetros. Por ejemplo:

Término Definición

Result_ColumnName: hace referencia a una columna existente que se usa para obtener los valores de resultado de la función LOOKUPVALUE().

Search_ColumnName: hace referencia a una columna existente que se usa para buscar un valor de la función LOOKUPVALUE().

omitting Los nombres de los parámetros se omitirán si el prefijo es lo suficientemente claro como para describir el parámetro.

Por ejemplo, en lugar de tener la sintaxis DATE siguiente (Year_Value, Month_Value, Day_Value) es más claro que el usuario lea DATE (Year, Month, Day). Repetir tres veces el valor del sufijo no mejora en nada la comprensión de la función y satura la lectura de forma innecesaria.

Sin embargo, si el parámetro con prefijo es Year_columnName, el nombre del parámetro y el prefijo permanecerán para garantizar que el usuario entiende que el parámetro requiere una referencia a una columna de años existente.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No

Sintaxis DAX

Artículo • 20/10/2023

En este artículo se describen la sintaxis y los requisitos del lenguaje de expresiones de fórmula DAX.

Requisitos de la sintaxis

Una fórmula DAX siempre comienza con un signo igual (=). Después del signo igual, se puede proporcionar cualquier expresión que se evalúe como un valor escalar o una que se pueda convertir en un valor escalar. que incluyen la siguiente información:

- Una constante escalar o expresión que usa un operador escalar (+,-,*,/,>=,...,&&, ...)
- Referencias a columnas o tablas. El lenguaje DAX siempre usa tablas y columnas como entradas a funciones, nunca una matriz o un conjunto arbitrario de valores.
- Operadores, constantes y valores proporcionados como parte de una expresión.
- El resultado de una función y sus argumentos necesarios. Algunas funciones DAX devuelven una tabla en lugar de un valor escalar y se deben ajustar en una función que evalúa la tabla y devuelve un valor escalar. A menos que la tabla sea una tabla de una sola fila, se trata como un valor escalar.

La mayoría de las funciones DAX requieren uno o más argumentos, que pueden incluir tablas, columnas, expresiones y valores. Sin embargo, algunas funciones, como PI, no requieren ningún argumento, pero siempre requieren paréntesis para indicar el argumento NULL. Por ejemplo, siempre se debe escribir PI() en lugar de PI. También se pueden anidar funciones dentro de otras funciones.

- Expresiones. Una expresión puede contener alguno de los elementos siguientes o todos ellos: operadores, constantes o referencias a columnas.

Por ejemplo, todas las fórmulas siguientes son válidas.

Fórmula	Resultado
= 3	3
= "Sales"	Sales
= "Sales"	Si utiliza esta fórmula dentro de la tabla Sales, obtendrá el valor de la columna

Fórmula	Resultado
[Importe]	Importe en la tabla Sales de la fila actual.
= (0,03 * [Importe])	Tres por ciento del valor de la columna Importe de la tabla actual.
=0.03 * [Amount]	Aunque esta fórmula se puede usar para calcular un porcentaje, el resultado no se muestra como un porcentaje a menos que se aplique el formato en la tabla.
= PI()	Valor de la constante pi.

Las fórmulas pueden comportarse de forma diferente en función de cómo se utilicen. Siempre debe tener en cuenta el contexto y el modo en que los datos que se usan en la fórmula están relacionados con otros datos que podrían utilizarse en el cálculo.

Requisitos de nomenclatura

A menudo, un modelo de datos contiene varias tablas. Las tablas y sus columnas, de forma conjunta, componen una base de datos almacenada en el motor de análisis en memoria (VertiPaq). En esa base de datos, todas las tablas deben tener nombres únicos. Los nombres de las columnas también deben ser únicos en cada tabla. Los nombres de objeto *no distinguen mayúsculas de minúsculas*; por ejemplo, los nombres **SALES** y **Sales** representarían la misma tabla.

Cada columna y medida que se agregan a un modelo de datos existente deben pertenecer a una tabla específica. La tabla que contiene la columna se especifica implícitamente, cuando se crea una columna calculada en una tabla, o explícitamente, cuando se crea una medida y se especifica el nombre de la tabla donde se debe almacenar la definición de la medida.

Cuando se usa una tabla o una columna como entrada para una función, generalmente se debe *calificar* el nombre de la columna. El nombre *completo* de una columna es el nombre de la tabla, seguido del nombre de la columna entre corchetes; por ejemplo, 'Ventas de EE. UU.'[Productos]. Siempre se requiere un nombre completo cuando se hace referencia a una columna en los contextos siguientes:

- Como argumento de la función, VALUES
- Como argumento de la función, ALL o EXCEPT
- En un argumento de filtro para las funciones, CALCULATE o CALCULATETABLE
- Como argumento de la función, RELATEDTABLE

- Como argumento para cualquier función de inteligencia de tiempo

Un nombre de columna *incompleto* es simplemente el nombre de la columna, entre corchetes: por ejemplo, [Importe de ventas]. Por ejemplo, si se hace referencia a un valor escalar de la misma fila de la tabla actual, se puede usar el nombre de la columna no completo.

Si el nombre de una tabla contiene espacios, palabras clave reservadas o caracteres no permitidos, deberá incluir el nombre de la tabla entre comillas simples. También deberá escribir los nombres de tabla entre comillas si el nombre contiene algún carácter fuera del rango de caracteres alfanuméricos ANSI, independientemente de si la configuración regional es compatible con el conjunto de caracteres o no. Por ejemplo, si abre un libro que contiene nombres de tablas escritos en caracteres cirílicos, como "Таблица", el nombre de la tabla debe ir entre comillas, aunque no contenga espacios.

ⓘ Nota

Para facilitar la entrada de los nombres completos de las columnas, use la característica AutoComplete en el editor de fórmulas.

Tablas

- Los nombres de tabla son necesarios siempre que la columna sea de una tabla distinta de la actual. Los nombres de tabla deben ser únicos en la base de datos.
- Los nombres de tabla deben ir entre comillas simples si contienen espacios, otros caracteres especiales o cualquier carácter alfanumérico que no sea el inglés.

Medidas

- Los nombres de medida siempre deben ir entre corchetes.
- Los nombres de medida pueden contener espacios.
- Cada nombre de medida debe ser único en el modelo. Por lo tanto, el nombre de la tabla es opcional delante de un nombre de medida cuando se hace referencia a una medida existente. Sin embargo, cuando se crea una medida, siempre se debe especificar una tabla en la que se almacenará la definición de la medida.

Columnas

Los nombres de columna deben ser únicos en el contexto de una tabla. Sin embargo, varias tablas pueden tener columnas con los mismos nombres (la desambiguación viene con el nombre de la tabla).

En general, se puede hacer referencia a las columnas sin hacer referencia a la tabla base a la que pertenecen, excepto cuando pueda haber un conflicto de nombres para resolver o con ciertas funciones que requieren que los nombres de columna estén completos.

Palabras clave reservadas

Si el nombre que se utiliza para una tabla es el mismo que el de una palabra clave reservada de Analysis Services, se genera un error y se debe cambiar el nombre de la tabla. Sin embargo, puede utilizar palabras clave en nombres de objetos si el nombre del objeto está entre corchetes (en las columnas) o con comillas (en las tablas).

ⓘ Nota

Las comillas se pueden representar con varios caracteres diferentes, en función de la aplicación. Si se pegan fórmulas de un documento o una página web externos, asegúrese de comprobar el código ASCII del carácter que se usa para las comillas de apertura y cierre, con el fin de garantizar que son iguales. De lo contrario, puede que DAX no reconozca los símbolos como comillas, lo que haría que la referencia no fuera válida.

Caracteres especiales

Los caracteres y tipos de caracteres siguientes no son válidos en los nombres de tablas, columnas o medidas:

- Espacios iniciales o finales; a menos que los espacios se incluyan entre delimitadores de nombre, corchetes o apóstrofes únicos.
- Caracteres de control
- Los caracteres siguientes no son válidos en los nombres de objetos:

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

Ejemplos de nombres de objeto

En la tabla siguiente se muestran ejemplos de algunos nombres de objeto:

Tipos de objeto	Ejemplos	Comentario
Nombre de tabla	Sales	Si el nombre de la tabla no contiene espacios ni otros caracteres especiales, no es necesario que el nombre se incluya entre comillas.
Nombre de tabla	"Canada Sales"	Si el nombre contiene espacios, tabulaciones u otros caracteres especiales, incluya el nombre entre comillas simples.
Nombre completo de la columna	Ventas[Importe]	El nombre de la tabla precede al nombre de la columna y el nombre de la columna se incluye entre corchetes.
Nombre completo de la medida	Ventas[Beneficio]	El nombre de la tabla precede al nombre de la medida y el nombre de la columna se incluye entre corchetes. En determinados contextos, siempre es necesario un nombre completo.
Nombre de columna no completo	[Importe]	El nombre no completo es simplemente el nombre de la columna, entre corchetes. Entre los contextos en los que se puede usar el nombre no completo se incluyen las fórmulas de una columna calculada en la misma tabla o en una función de agregación que está buscando en la misma tabla.
Columna completa de la tabla con espacios	"Canada Sales" [Cantidad]	El nombre de la tabla contiene espacios, por lo que debe ir entre comillas simples.

Otras restricciones

La sintaxis necesaria para cada función y el tipo de operación que puede realizar varían en gran medida dependiendo de la función. Pero, por lo general, las reglas siguientes se aplican a todas las fórmulas y expresiones:

- Las fórmulas y expresiones DAX no pueden modificar o insertar valores individuales en las tablas.
- No se pueden crear filas calculadas con DAX. Solo se pueden crear medidas y columnas calculadas.
- Al definir columnas calculadas, se pueden anidar funciones en cualquier nivel.
- DAX tiene varias funciones que devuelven una tabla. Normalmente, se usan los valores que devuelven estas funciones como entrada a otras funciones, que

requieren una tabla como entrada.

Operadores y constantes DAX

En la tabla siguiente se enumeran los operadores que son compatibles con DAX. Para obtener más información sobre la sintaxis de los operadores individuales, vea

[Operadores DAX](#).

Tipo de operador	Símbolo y uso
Operador de paréntesis	() orden de prioridad y agrupación de argumentos
Operadores aritméticos	+ (suma) - (resta/ signo) * (multiplicación) / (división) ^ (exponenciación)
Operadores de comparación	= (igual a) > (mayor que) < (menor que) >= (mayor o igual que) <= (menor o igual que) <> (no es igual a)
Operador de concatenación de texto	& (concatenación)
Operadores de lógica	&& (y) (or)

Tipos de datos

No es necesario convertir o especificar de cualquier otro modo el tipo de datos de una columna o valor que se usa en una fórmula DAX. Cuando se usan datos en una fórmula

DAX, este identifica automáticamente los tipos de datos de las columnas a las que se hace referencia y de los valores que se escriben, y realiza las conversiones implícitas, cuando sea necesario, para completar la operación especificada.

Por ejemplo, si se intenta agregar un número a un valor de fecha, el motor interpretará la operación en el contexto de la función y convertirá los números a un tipo de datos común y, después, presentará el resultado en el formato previsto, una fecha.

Sin embargo, existen algunas limitaciones en los valores que se pueden convertir correctamente. Si un valor o una columna tienen un tipo de datos que es incompatible con la operación actual, DAX devuelve un error. Además, DAX no proporciona funciones que permitan cambiar o convertir explícitamente el tipo de datos de los datos existentes que se han importado en un modelo de datos.

Importante

DAX no admite el uso del tipo de datos variable. Por lo tanto, cuando se cargan o importan datos en un modelo de datos, se espera que los datos de cada columna sean normalmente de un tipo de datos coherente.

Algunas funciones devuelven valores escalares, incluidas cadenas, mientras que otras funcionan con números, enteros y reales, o con fechas y horas. El tipo de datos necesario para cada función se describe en la sección [Funciones DAX](#).

Se pueden usar tablas que contengan varias columnas y varias filas de datos como argumento para una función. Algunas funciones también devuelven tablas, que se almacenan en memoria y se pueden usar como argumentos para otras funciones.

Fecha y hora

DAX almacena los valores de fecha y hora con el tipo de datos datetime utilizado por Microsoft SQL Server. El formato datetime usa un número de punto flotante donde los valores de fecha corresponden a la parte entera que representa el número de días desde el 30 de diciembre de 1899. Los valores de hora corresponden a la parte decimal de un valor de fecha donde las horas, los minutos y los segundos se representan mediante fracciones decimales de un día. Las funciones de fecha y hora de DAX convierten implícitamente los argumentos al tipo de datos datetime.

Nota

El valor DateTime máximo exacto admitido por DAX es el 31 de diciembre de 9999 00:00:00.

Literal de fecha y hora

A partir de la versión de agosto de 2021 de Power BI Desktop, los valores date y datetime de DAX se pueden especificar como literales con el formato `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` o `dt"YYYY-MM-DD hh:mm:ss"`. Cuando se especifican como literal, no es necesario usar las funciones `DATE`, `TIME`, `DATEVALUE` y `TIMEVALUE` en la expresión.

Por ejemplo, la expresión siguiente usa las funciones DATE y TIME para filtrar por OrderDate:

```
DAX  
  
EVALUATE  
FILTER (  
    FactInternetSales,  
    [OrderDate] > (DATE(2015,1,9) + TIME(2,30,0)) && [OrderDate] <  
    (DATE(2015,12,31) + TIME(11,59,59))  
)
```

La misma expresión de filtro se puede especificar como literal:

```
DAX  
  
EVALUATE  
FILTER (  
    FactInternetSales,  
    [OrderDate] > dt"2015-1-9T02:30:00" && [OrderDate] < dt"2015-12-  
31T11:59:59"  
)
```

ⓘ Nota

El formato literal con tipo date y datetime de DAX no se admite en todas las versiones de Power BI Desktop, Analysis Services y Power Pivot en Excel. Normalmente, las funcionalidades de DAX nuevas y actualizadas se presentan primero en Power BI Desktop y, después, se incluyen en Analysis Services y Power Pivot en Excel.

Comentarios

¿Le ha resultado útil esta página?

 Yes

 No