

El lenguaje DAX

DAX es el lenguaje de expresiones a través del cual construiremos la lógica de negocio en PowerPivot, tanto a nivel de tabla dinámica como de modelo de datos. Permite definir cálculos personalizados en tablas de PowerPivot (columnas calculadas) y en tablas dinámicas de Excel (medidas).

Las fórmulas DAX son muy similares a las fórmulas de Excel. Muchas funciones de DAX tienen el mismo nombre y el mismo comportamiento general que las funciones de Excel, pero se han modificado para aceptar tipos diferentes de entradas y, en algunos casos, podrían devolver un tipo de datos diferente.

Las fórmulas de DAX presentan diferencias con las fórmulas de Excel en los siguientes puntos:

- No es posible hacer referencia solo a unas cuantas celdas o a un rango de celdas. DAX siempre trabaja con columnas o tablas completas.
- Si desea personalizar los cálculos fila a fila, PowerPivot proporciona funciones que permiten usar el valor de la fila actual o un valor relacionado para realizar cálculos que varíen según el contexto.
- Muchas de las nuevas funciones de DAX devuelven una tabla de valores o realizan cálculos basados en una tabla de valores como entrada. Estas funciones se pueden usar para proporcionar la entrada a otras funciones, calculando así valores para tablas o columnas completas.
- Algunas funciones de DAX proporcionan inteligencia de tiempo, que le permite crear cálculos usando intervalos de fechas significativos y comparar los resultados por períodos paralelos.
- Las funciones de fecha y hora de DAX devuelven un tipo de datos datetime. En contraste, las funciones de fecha y hora de Excel devuelven un entero que representa una fecha como un número de serie.
- DAX no admite el tipo de datos *variant* de Excel. Se espera que los datos de una columna de basen siempre en el mismo tipo de datos. Si los datos no son del mismo tipo, DAX cambia la columna completa al tipo de datos que mejor se acomode a todos los valores.

Medidas y Columnas calculadas

Una fórmula DAX se utiliza para definir un campo que se ubicará en algún lugar de una tabla dinámica de Excel. Hay dos tipos de campos que se pueden colocar en una tabla dinámica: columnas y medidas.

Las **columnas calculadas** son columnas que se definen en la ventana de PowerPivot al proporcionar un nombre de columna y una expresión DAX. Las columnas calculadas se rellenan con valores cuando se definen y se convierten como cualquier otra columna en una tabla, excepto que las columnas regulares (no calculadas) generalmente se importan de una fuente de datos externa mientras que las columnas calculadas se calculan después de cargar las columnas normales. Los valores de una columna calculada pueden colocarse en etiquetas de fila o etiquetas de columna de una tabla dinámica o pueden colocarse en un filtro / segmentación de datos para controlar los datos que se usarán en el análisis. Finalmente, una columna calculada se puede usar como parte del cálculo que define una medida.

Ejemplo de columnas calculadas son:

[Margen] =[Ventas]-[Costes]

[CiudadProvincia] =[Ciudad] & ", " & [Provincia]

Por otra parte las **medidas son funciones agregadas** (tipo suma, promedio, etc.) que se pueden definir tanto en la ventana de PowerPivot (en el área bajo cada columna de datos de la tabla) como en Excel (en la ventana de Excel, mientras el foco está en una tabla dinámica). Una vez definidas, se pueden agregar al área de valores de la tabla dinámica.

La medida se evalúa cuando se coloca en el área valores de una tabla dinámica de Excel y luego se evalúa por separado para cada celda del área. Ejemplos de medidas son:

[Ventas] =SUM(VentasReales[ImporteVentas])

[VentaPromedio] =AVERAGE(VentasReales[ImporteVentas])

Medidas implícitas y explícitas

Si una medida se crea a través de la ventana de PowerPivot al tomar una columna y agregarla, se conoce como una **medida implícita**. Las medidas implícitas en esta ventana son las seis siguientes: SUM, COUNT, MIN, MAX, AVERAGE o DISTINCTCOUNT (cuenta diferentes).

Sin embargo, PowerPivot puede realizar cálculos que son mucho más potentes utilizando fórmulas DAX en lugar de las seis agregaciones integradas (**medidas explícitas**).

Sintaxis básica de DAX

Una fórmula de DAX está compuesta por un signo igual seguido por una **función** o una **expresión**.

Las funciones realizan operaciones y normalmente utilizan argumentos, los cuales pueden ser una referencia a una columna o tabla. Tal y como las funciones de Excel, pueden estar anidadas dentro de otras funciones.

Una expresión se puede usar para definir un valor que puede ser un valor literal o una constante, una prueba booleana, o una referencia a una columna que contiene valores. Las expresiones booleanas se pueden utilizar para definir una condición de filtro.

Ejemplo: **[Ventas] > 100**.

Operadores

Los operadores indican cómo se deben comparar o procesar los valores. La siguiente tabla muestra operadores de DAX:

Tipo de operador	Símbolo y uso
Operador de paréntesis	() orden de prioridad y agrupación de argumentos
Operadores aritméticos	+ (suma) - (resta/negación) * (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 lógicos	&& (y) (o) ! (negación)

Los valores utilizados en las fórmulas y expresiones se pueden escribir directamente en la barra de fórmulas como parte de una expresión o se pueden obtener de otras columnas, tablas o fórmulas.

Tipos de datos de DAX

Cuando se utilizan datos en una fórmula de DAX, DAX identifica automáticamente los tipos de datos en las columnas a las que se hace referencia y en los valores que se escriben, y realiza conversiones implícitas donde sea necesario para completar la operación especificada.

Sin embargo, hay algunas limitaciones que afectan a los valores que se pueden convertir correctamente. Si un valor o una columna tienen un tipo de datos incompatible con la operación especificada, DAX devuelve un error. Asimismo, DAX no proporciona funciones que permitan explícitamente cambiar o convertir el tipo de datos de los datos existentes que ha importado a un libro de PowerPivot. En general, estos tipos de datos se implementan para habilitar cálculos precisos en columnas calculadas y, para mantener la coherencia, se aplican las mismas restricciones al resto de los datos en PowerPivot.

Los formatos usados para números, moneda, fechas y horas deben seguir el formato de la configuración regional especificada en el equipo que abre el libro.

Los diferentes tipos de datos que puede admitir Excel se han ampliado en PowerPivot. Resalta, entre ellos, el tipo de datos de tabla, que se utiliza como entrada o salida para muchas nuevas funciones. 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.

Referencias a columnas y medidas

El nombre de una columna calculada siempre es único dentro de una tabla determinada, y el nombre de una medida debe ser único en todo el modelo de datos. Cuando una expresión DAX se refiere a una columna o medida, el nombre de esa columna o medida debe aparecer entre corchetes, y algunas veces estará precedido por el nombre de la tabla a la que pertenece la columna o medida (nombres completamente calificados). Ejemplos:

Nombres completamente calificados:

Table1[Column2] = SUM(Table1[Column2])

Table2[Measure1] = 2.5 * Table2[Measure1]

Nombres no calificados:

[Column2] = [Column1] + [Column2]

[Measure2] = [Measure2] / [Measure3]

Las fórmulas DAX que hacen referencia a columnas generalmente requieren que las referencias estén completamente calificadas excepto cuando una columna calculada se refiere a otras columnas dentro de la misma tabla. Sin embargo, como los nombres de medidas son únicos dentro de un modelo de datos, no se requiere que los nombres de las mismas estén completamente calificados.

Para facilitar la definición y el mantenimiento de las medidas, se recomienda en lo posible dividir las medidas en medidas separadas y usarlas en una medida nueva.

En lugar de escribir... SUM (VentasReales[ImporteVentas])/SUM(Pedidos[CantVentas]), se recomienda crear dos medidas y usarlas en una medida nueva.

Esta práctica mejorará la legibilidad, la capacidad de depurar partes de la medida y ayudará a definir la medida.

Las medidas intermedias se pueden ocultar para que no estén expuestas a los usuarios finales.

Contextos: contexto de fila y contexto del filtro

Cada fórmula de DAX se puede evaluar en el contexto de un entorno específico.

Si la fórmula se evalúa en cada fila de datos de la tabla estaremos hablando de un **contexto de fila**. Como ejemplo de fórmulas que se evalúan en un contexto de fila tenemos las columnas calculadas,

Si se evalúa en el contexto de un conjunto específico de filtros hablamos de un **contexto de filtro**. Una medida que se coloca en una tabla dinámica se evaluará para cada celda en el área de valores, y cada una de esas celdas tiene su propio "contexto de filtro" que es la combinación de etiquetas de fila, etiquetas de columna, filtros de informe y segmentaciones de datos.

Funcionalidades que no existen en las fórmulas de Excel

PowerPivot incluye capacidades tales como (relaciones entre tablas, agregación dinámica, modificación de contexto, etc. que no existen en Excel. Para ello, proporciona funciones que brinden funcionalidades que permiten explotar dichas capacidades.

Algunas de las categorías de funciones de DAX que no existen en Excel son:

- Funciones que navegan por las relaciones entre tablas (más poderosas que VLOOKUP)
- Funciones que toman tablas como argumentos (agregación sobre una tabla, filtrado de una tabla, etc.)
- Funciones que producen un resultado de tabla (este resultado debe ser una entrada para otra función)
- Funciones que permiten determinar el contexto para un cálculo.
- Funciones que modifican el contexto para un cálculo (por ejemplo, calcula la fórmula para todos los productos o años)
- Funciones de inteligencia de tiempo DAX (períodos paralelos, período anterior, YTD, etc.)

Funciones DAX simples

Funciones similares a las funciones de Excel

DAX admite aproximadamente ochenta (80) funciones de Excel. A pesar de que existen algunas diferencias específicas con respecto a cómo funcionan estas funciones en Excel, en general estas funciones son muy similares.

Las funciones lógicas AND y OR sólo permiten dos argumentos en DAX. CONCATENATE tiene la misma limitación. Para evitar estas limitaciones, se recomienda utilizar los operadores en lugar de las mismas:

Función	Operador
AND	&&
OR	
CONCATENATE	&

Las funciones de Excel admitidas en DAX son las siguientes:

Fecha y hora	Estadísticas	Texto	MAT. Y TRIG.
DATE	AVERAGE	CONCATENATE	ABS
DATEVALUE	AVERAGEA	EXACT	CEILING, ISO.CEILING
DAY	COUNT	FIND	EXP
EDATE	COUNTA	FIXED	FACT
EOMONTH	COUNTBLANK	LEFT	FLOOR
HOUR	MAX	LEN	INT
MINUTE	MAXA	LOWER	LN
MONTH	MIN	MID	LOG
NOW	MINA	REPLACE	LOG10
SECOND	RANK.EQ	REPT	MOD
TIME	STDEV.P	RIGHT	MROUND
TIMEVALUE	STDEV.S	SEARCH	PI
TODAY	VAR.P	SUBSTITUTE	POWER
WEEKDAY	VAR.S	TRIM	QUOTIENT
WEEKNUM		UPPER	RAND
YEAR		VALUE	RANDBETWEEN
YEARFRAC			ROUND
Información	Lógicas		ROUNDDOWN
ISBLANK	AND		ROUNDUP
ISERROR	IF		SIGN
ISLOGICAL	IFERROR		SQRT
ISNONTEXT	NOT		SUM
ISNUMBER	OR		TRUNC
ISTEXT	FALSE		
	TRUE		

Valores en blanco. La función BLANK ()

Cuando un campo en una base de datos tradicional no contiene un valor, el campo vacío a menudo se denomina valor NULO. En Excel, una celda vacía se conoce como una celda "en blanco".

En PowerPivot, cuando se tiene una celda en blanco en una columna de datos, Excel permite realizar aritmética en la celda y simplemente trata la celda en blanco como cero.

Por ejemplo, BLANK +3 genera un resultado de 3.

Sin embargo hay un caso en el cual DAX difiere de Excel y es el siguiente: Si tenemos una fórmula que suma una columna de números y toda la columna no contiene más que espacios en blanco, en lugar de generar un resultado de cero (que es lo que hace Excel), DAX y PowerPivot generarán un resultado en BLANCO. Siempre que haya valores en la columna, obtendremos un resultado numérico, pero cuando todos los valores estén **en blanco**, el resultado también será **en blanco**.

DAX tiene la función **BLANK()** que devuelve un valor vacío (en blanco). Esto es útil cuando se desea saber si el resultado de una expresión está en blanco. Asimismo, DAX también admite la función Excel **ISBLANK()** para el mismo propósito.

La función FORMAT(valor, formato)

En lugar de la función de TEXTO de Excel, DAX utiliza la función **FORMATO** para convertir varios valores numéricos y de fecha en cadenas. Esta función funciona igual que la función FORMATO en Visual Basic.

Las funciones "X" de agregación

DAX implementa funciones de agregación desde Excel, que incluyen SUMA, PROMEDIO, MIN, MAX, COUNT, STDEV.P, STEDEV.S, VAR.P, VAR.S, pero en lugar de tomar múltiples argumentos (una lista de rangos), pueden tomar solo una referencia a una columna. Para superar esta limitación, DAX también agrega algunas nuevas funciones de agregación que resumen cualquier expresión sobre las filas de una tabla.

Funciones de Agregación
SUMX (Table, Expression)
AVERAGEX (Table, Expression)
COUNTAX (Table, Expression)
MINX(Table, Expression)
MAXX(Table, Expression)
STDEVX.P(Table, Expression)
STDEVX.S(Table, Expression)
VARX.P (Table, Expression)
VARX.S (Table, Expression)

En cada caso, el primer argumento es una tabla sobre la que desea resumir una expresión, y el segundo argumento es la expresión que se agregará.

Ejemplo:

= SUMX(VentasReales, [UnitPrice] * [CantVentas])

Las funciones de agregación son particularmente potentes en las medidas porque el contexto en el que se evalúan cambia (se filtra) la tabla sobre la que se realiza la agregación.

Por ejemplo, si tenemos una tabla dinámica con **años** en las etiquetas de columna, y **tienda** en las etiquetas de fila, y luego colocamos esta fórmula:

=SUMX(VentasReales, [UnitPrice]*[CantVentas]) en una medida, para cada celda de la tabla dinámica, obtendremos el subtotal de las ventas que pertenecen a un año y una tienda en particular.

COUNTROWS(Table)

COUNTROWS es similar a las otras funciones COUNT (COUNT, COUNTA, COUNTX, COUNTAX, COUNTBLANK) pero toma como argumento una tabla y devuelve el recuento de filas en esa tabla. Por ejemplo, la fórmula:

=COUNTROWS (VentasReales)

devuelve el número de transacciones de ventas en la tabla VentasReales.

RELATED(Column) and RELATEDTABLE(Table)

DAX presenta las funciones RELATED y RELATEDTABLE para realizar el seguimiento de las relaciones y recuperar datos relacionados de otra tabla. Esto es más poderoso que VLOOKUP en Excel en varios sentidos. Primero, VLOOKUP solo devuelve la primera coincidencia; no hay promesa de integridad referencial y no hay seguridad de que no haya otra fila que también haya coincidido con la búsqueda. Segundo, en lugar de requerir que la tabla de búsqueda esté organizada de una manera particular, con ciertas columnas en el lado izquierdo, las funciones DAX dependen de las relaciones entre las dos tablas, y esas relaciones pueden involucrar cualquiera de las columnas en una tabla.

RELATED(Columna) sigue las relaciones existentes de muchos a uno desde muchos lados hacia un lado y devuelve el valor de coincidencia único de la otra tabla. Considera estos ejemplos:

En la tabla VentasReales, podemos agregar dos columnas calculadas usando estas fórmulas:

=RELATED(Tiendas[NombreTienda])

=RELATED(Geografia[Continente])

Estas fórmulas agregarán dos columnas a la tabla VentasReales, la primera contendrá el nombre de cada tienda y la segunda el nombre del continente. El primero de estos ejemplos sigue una relación única de VentasReales a Tiendas, mientras que la segunda fórmula debe navegar por dos relaciones: de VentasReales a Tiendas y luego de Tiendas a Geografia.

Crear esta fórmula solo requiere que conozca el nombre de la columna de la que desea obtener un valor.

RELATEDTABLE(Table) sigue una relación en cualquier dirección (de muchos a uno o de uno a muchos) y **devuelve una tabla** que contiene todas las filas relacionadas con la fila actual de la tabla especificada. Esto es muy útil cuando quiere encontrar todas las transacciones asociadas con una fila particular de una tabla relacionada.

Sin embargo, esto significa que esta función no se puede usar por sí misma para definir una columna calculada o una medida. En cambio, esta función solo puede usarse para proporcionar un resultado intermedio que a su vez es un argumento para otra función, como una función de agregación.

Ejemplo: =SUMX(RELATEDTABLE(VentasReales), VentasReales[ImporteVentas])

Esta fórmula dice que primero queremos construir una tabla que contenga las filas de VentasReales relacionadas con la fila actual.

RELATEDTABLE(VentasReales) supone que tenemos una fila actual y existe una relación entre la tabla actual y la tabla VentasReales. Una vez que tengamos la tabla que contiene las filas relacionadas de las transacciones de venta, tomaremos ImporteVentas de cada fila y luego sumaremos esos importes.

Si colocásemos esta fórmula en una columna calculada en la tabla de Productos. Luego, la misma fórmula exacta sin cambios en una columna calculada en la tabla Tiendas. Y finalmente, en la tabla Fechas, si bien obtenemos números completamente diferentes en cada una de esas tres tablas, veremos que obtenemos básicamente lo mismo: un total de los importes de ventas desglosadas por cada producto, o por cada tienda, o para cada fecha.

USERELATIONSHIP(Column1, Column2)

En el modelo de datos de DAX pueden existir múltiples relaciones entre dos tablas, pero sin embargo, solo puede existir una activa, directa o indirectamente. La función USERELATIONSHIP permite especificar el uso de una relación no activa.

La función USERELATIONSHIP solo se puede usar en funciones que toman un filtro como argumento, como CALCULATE, que se describirá más adelante en este documento dado que las relaciones activas son las que se utilizan por defecto en expresiones DAX.

La función USERELATIONSHIP usa relaciones existentes en el modelo, identificando relaciones por sus columnas de punto final. Ejemplo:

=CALCULATE([Ventas], USERELATIONSHIP(VentasReales[FechaEnvio], 'Date'[Fecha]))

FILTER(Table, Condition) and DISTINCT (Column)

FILTER y DISTINCT son dos de las funciones DAX que devuelven una tabla de resultados. Generalmente, estas funciones se usan como argumentos para otras funciones de agregación.

FILTER(Table, Condition) devuelve un subconjunto de la tabla especificada donde la condición es TRUE. Devolverá todas las columnas de la tabla especificada, y algún subconjunto de las filas en esa tabla. Ejemplo:

`FILTER (VentasReales, RELATE (Geografia[Ciudad]) = "Baltimore")`

devolverá una tabla que contiene todas las filas de VentasReales que tuvieron lugar en la ciudad de Baltimore. Para obtener el total de ventas de la ciudad de Baltimore, simplemente usaríamos esto como argumento de tabla para SUMX de la siguiente manera:

`=SUMX(FILTER(VentasReales, RELATE (Geografia[Ciudad]) = "Baltimore"), [ImporteVentas])`

DISTINCT(Column) devuelve una tabla que contiene los valores únicos dentro de la columna especificada. En otras palabras, una tabla de los valores, con cualquier valor duplicado eliminado de la lista. Ejemplo:

`DISTINCT (VentasReales[idTienda])` devolverá una tabla que contiene los diferentes Id de Tienda para los que hubo transacciones de venta.

Repasando el concepto de contexto de fila y de filtro

El contexto de filas se entiende más fácilmente como la "fila actual". Esto es más obvio en los casos en que se ingresa una fórmula en una columna calculada. Por ejemplo, dentro de la tabla VentasReales, se puede crear la siguiente fórmula para una columna calculada:

`= [CantVentas] * [CosteUnitario]`

Cuando no hay un contexto de fila, DAX no puede interpretar el nombre de una columna como un valor. Si tratamos de ingresar exactamente la misma fórmula en una medida, recibiremos un mensaje de error: "El valor de 'CantVentas' no se puede determinar. O bien 'CantVentas' no existe, o no hay una fila actual para una columna llamada 'CantVentas'."

En este caso, el problema es que no hay una fila actual (o ningún contexto de fila), por lo que no hay forma de saber qué número usar para [CantVentas].

El contexto de filtro es más complejo que el contexto de fila y puede describirse más fácilmente como el conjunto de filtros asociados con una de las celdas en el área de valores de una tabla dinámica. La fórmula que define una medida siempre se evalúa varias veces, una para cada celda en el área de valores. Y cada una de esas evaluaciones tiene su propio contexto de filtro.

La presencia del contexto de filtro para las medidas y la presencia del contexto de fila para las columnas calculadas es una de las razones por las que a menudo no se puede usar la misma fórmula para definir una medida que la que usaría para definir una columna calculada.

La mayoría de las fórmulas de medida serán de agregación (sumas, recuentos, promedios, mínimos, máximos o diferencias) de una expresión, o una fórmula que involucre otras medidas, o alguna combinación de ambas.