

## Funciones DAX Avanzadas

### **CALCULATE(Expression, SetFilter1, SetFilter2,...)**

La función CALCULATE permite que cualquier expresión DAX se evalúe en un contexto de filtro específico. Esto es equivalente a definir una medida y también definir un conjunto de filtros que se utilizarán para proporcionar un contexto en el que se va a evaluar la medida. CALCULATE hace lo siguiente:

1. Usando los argumentos de **SetFilter**, modifica el contexto del filtro
2. Si hay un contexto de fila, mueve ese contexto de fila al contexto de filtro
3. Evalúa la expresión en el contexto de filtro recientemente modificado

Aquí hay algunos ejemplos que ilustran cómo funciona esto.

#### **Ejemplos:**

##### **CALCULAR sin ningún argumento de SetFilter**

Si agregásemos una columna calculada a una tabla Inventario con la fórmula:

= COUNTROWS(VentasReales)

Obtendremos el número de filas de la tabla VentasReales para cada celda en la nueva columna calculada.

Ahora, si cambiásemos la fórmula a

= CALCULATE(COUNTROWS (VentasReales)).

Utilizaremos el contexto de la fila actual se colocara como contexto del filtro.

Es decir, el contexto del filtro tendrá un conjunto de filtros como los siguientes:

- [IdInventario] = <valor de la fila>
- [Fecha] = <valor de la fila>
- [Tienda] = <valor de la fila>
- [IdProducto] = <valor de la fila>
- [Moneda] = <valor de la fila>
- etc....

Lo que lo hace un poco más complicado es que el contexto de la fila no es simplemente la fila actual de esa tabla, sino la fila actual de esa tabla, así como la fila actual de todas las tablas en un lado de una relación de muchos a uno de esa tabla (en realidad incluye la fila actual de todo el conjunto de tablas relacionadas).

Esto significa que el contexto de la fila es la fila actual de **Inventario**, así como la fila actual de **Fechas**, la fila actual de **Productos** y la fila actual de **Tiendas**.

Es decir, usando ese contexto de filtro modificado queremos contar las filas en la tabla **VentasReales** como si estuviésemos realizando un cálculo para una medida en una tabla dinámica donde todos estos filtros se han colocado en la tabla dinámica.

## Argumentos de SetFilter

El argumento **SetFilter** se usa para modificar el contexto en el que se va a evaluar una expresión.

Digamos que queremos hacer un análisis sobre los datos de ventas. Definimos una medida simple en la tabla de VentasReales llamada [Ventas] que está definida por esta fórmula DAX:

=SUM(VentasReales[ImporteVentas]).

[Ventas] no es más que la suma del campo [ImporteVentas] para todas las transacciones de venta contenidas en un segmento particular de los datos.

Las ventas pueden ser realizadas a través de cuatro canales distintos (ventas de "catálogo", ventas "en línea", ventas de "revendedor" y ventas de "tienda") donde cada uno de los mismos es una fila separada en la tabla Canales, y cada fila de VentasReales se identifica como perteneciente a uno de estos canales por el campo [Canal] de esta tabla.

Pero si deseásemos asegurarnos de que no importa lo que elijan, solo estén viendo las ventas del canal "Tienda", definiremos una nueva **medida** llamada [VentasEnTienda] como:

=CALCULATE ([Ventas], Canales[Canal] = "Tienda").

El argumento **SetFilter** dentro de CALCULATE garantiza que se anulará cualquier filtro (si existe) en la columna Canal con un nuevo filtro que dice que solo queremos ver las ventas del canal Tienda. Al usar esta medida, garantizamos que miraremos las ventas de las tiendas y no tenemos que depender de los usuarios de la tabla dinámica para colocar el canal en algún lugar la misma.

## Sintaxis para CALCULATE (Expresión [, SetFilter1] [, SetFilter2] ...)

Cuando se tiene una función CALCULATE donde la expresión a evaluar es simplemente una medida, entonces hay un atajo disponible. Siguiendo el ejemplo anterior:

= CALCULATE ([Ventas], Canales[Canal]="Tienda")

Un acceso directo para CALCULATE (<medida>, <SetFilters> opcionales) consiste en usar el nombre de la medida como si fuera un nombre de función y escribir esto en su lugar:

=<Nombre de Medida>(<optional SetFilters>)

Cuando aplicamos esto a nuestro ejemplo, encontramos que la medida [VentasEnTienda] puede ser definido como:

= [Ventas](Canales[Canal] = "Tienda")

Un argumento SetFilter en la función CALCULATE puede ser cualquiera de los siguientes:

1. Una expresión booleana que se refiere a una sola columna como  
Canales[Canal] = "Tienda" o  
(Geografia[Ciudad]="Seattle" || Geografia[Ciudad]="Portland").

## 2. Una expresión de tabla como Filter(Table1, Condition)

Consideremos una tabla que contiene tres columnas llamadas "Ciudad", "Producto" y "Mes". Las filas en la tabla especifican las combinaciones válidas de valores para esas columnas. Cualquier combinación de valores no presente en la tabla se filtra. En otras palabras, podemos pasar una tabla como esta por un argumento SetFilter a la función CALCULATE, y eliminará cualquier combinación de ciudad, producto y mes que no esté en esta tabla.

[Ciudad]	[Producto]	[Mes]
Seattle	Bicicleta	Ene 2008
Seattle	Coche	Feb 2008
Boston	Bicicleta	Mar 2008

El escenario más común para usar una tabla como un argumento SetFilter para la función CALCULATE es la función ALL que se ilustrará más adelante.

## La función VALUES(Column)

La función **VALUES** devuelve todos los valores válidos para esa columna en el contexto de filtro actual. El resultado devuelve una tabla de valores, aunque la tabla solo tenga un valor (o incluso ningún valor). Esta función es una manera fácil de preguntar sobre el contexto de filtro actual. A menudo se usa dentro de una función IF.

Imagine un escenario en el que desea hacer un cálculo cuando el país es Estados Unidos, y un cálculo diferente cuando el país no es Estados Unidos. Esto se puede lograr con VALUES (Geografía [País])=" Estados Unidos".

Por supuesto, puede haber una tabla dinámica donde no se especifica el país, o una situación en la que se han seleccionado varios países y solo podemos comparar una tabla con un solo valor cuando la tabla tiene solo un valor.

Esto nos obliga a escribir algo como esto para determinar cuál es el país actual:

=IF (HASONEVALUE(Geografía[País]), VALUES (Geografía [País]), "No solo país seleccionado")

La función **HASONEVALUE** devuelve **TRUE** cuando el contexto de la columna se ha filtrado a un único valor distintivo. Esta función es muy conveniente para probar si la función **VALUES** devolverá un valor único, en cuyo caso se puede referenciar como un valor escalar.

Esta fórmula dice que, si hay un país seleccionado en el contexto actual, devuelva el nombre de ese país, de lo contrario, devuelva la expresión "Ningún país seleccionado".

## **CALCULATE(ExpresiónDeTabla, SetFilter1, SetFilter2,...)**

**CALCULATE** es como **CALCULATE**, excepto que tanto la expresión como el resultado son tablas ambas. Esta función se utiliza cuando se desea cambiar el contexto del filtro y luego construir una tabla en ese nuevo contexto.

## **ALL(Tabla) y ALL(Columna1 [, Columna2] ...)**

La función **ALL** es una función con valores de tabla que hace que el contexto del filtro para las columnas especificadas sea ignorado. Esta función es particularmente útil cuando se usa como un argumento **SetFilter** con valores de tabla dentro de la función **CALCULATE**.

Por ejemplo:

**ALL(Productos)** devuelve una tabla que contiene todas las filas de la tabla Productos, ignorando cualquier filtro que pueda estar presente en el contexto del filtro. Las filas duplicadas se eliminan. Como el argumento para **ALL** es una tabla, el contexto de filtro se elimina efectivamente de todas las columnas en esa tabla.

**ALL(Columna1)** devuelve una tabla que contiene todos los valores de **Columna1** (sin ningún duplicado) después de eliminar / ignorar cualquier filtro del contexto de filtro que pueda haber estado presente en esa columna.

Cuando se suministran múltiples argumentos de columna a la función **ALL**, se requiere que todas esas columnas pertenezcan a la misma tabla. Por ejemplo, **ALL(Column1, Column2)** devuelve una tabla con dos columnas de una tabla de origen, que consta de todas las combinaciones de **Column1** y **Column2** en los datos originales, e ignorando cualquier filtro del contexto de filtro que pudo haber estado presente en esas columnas.

Por ejemplo, supongamos que tenemos una medida **VentasReales[Ventas]** que se ha definido como

**=SUM(VentasReales[ImporteVentas]).**

Por sí misma, la medida **[Ventas]** es la suma de la columna **ImporteVentas** para cualquier contexto. Es decir, es un número diferente en cada celda de una tabla dinámica porque cada celda tiene un contexto diferente (una categoría de producto y un año diferentes).

Utilizando la función **ALL** podemos agregar medidas tipo Ratio a las tablas dinámicas.

Por ejemplo, podemos agregar **RatioProductos**, que es las ventas para un contexto dado dividido por ventas para todos los productos (en el mismo año). Hay dos formas de escribir esto, una usando **CALCULATE** y otra usando la sintaxis de acceso directo:

**= [Ventas] / CALCULATE([Ventas], ALL(Productos))**  
**= [Ventas] / [Ventas](ALL(Productos))**

Este es un escenario de BI muy común. Necesitamos calcular algo en comparación con lo mismo, pero para un conjunto diferente de productos, o para un año diferente, o para una región diferente, etc. Cambiar una parte del contexto de filtro para una parte de la fórmula es muy poderoso.

### **ALLEXCEPT (Tabla [, Columna1] [, Columna2] ...)**

**ALLEXCEPT** es una notación abreviada para la serie de funciones **ALL**. Imagina una tabla que tenga 43 columnas, y digamos que desea eliminar / ignorar el contexto del filtro de 40 de esas columnas, y conservar el contexto del filtro en tres de ellas. Puedes lograr esto de dos maneras:

**ALL**(Columna1, Columna2, ... <repetida para 40 columnas>)

elimina el contexto para las cuarenta columnas especificadas aquí.

**ALLEXCEPT**(Tabla, Columna1, Columna2, Columna3)

elimina el contexto para todas las columnas en la tabla especificada, excepto las tres columnas que se especifican aquí.

### **RANKX (Tabla, Expresión [, Valor] [, Orden] [, Clasificación] ...)**

La función **RANKX** devuelve una clasificación de un número en una lista de números para cada fila en el argumento **Tabla**. La función puede tomar opcionalmente un argumento de valor que represente un valor escalar cuyo rango se va a encontrar.

El argumento **Orden** especifica cómo clasificar el valor, descendente (0) o ascendente (1).

El argumento de **Clasificación** define cómo determinar el ranking cuando hay vínculos.

**Skip** (predeterminado) usará el siguiente valor de rango después de un empate, y **Dense** usará el siguiente valor de rango (es decir, no habrá espacios vacíos en los números de rango).

El siguiente ejemplo ilustra el uso de la función **RANKX** para clasificar las ventas por color de producto. Las dos expresiones de medida son:

Ventas =SUM(VentasReales[ImporteVentas])

ColorRank =RANKX(ALL(Productos[Color]), [Ventas])

En este ejemplo, la función **RANKX** compara el valor de las ventas para cada fila de una tabla dinámica y le otorga una posición a cada una en el ranking.

### **RANK.EQ(Valor, Columna [,Orden])**

Devuelve una clasificación de un número en una lista de números con su tamaño relativo a otros valores en la lista. El argumento **Orden** es un valor opcional (0 o 1) que especifica cómo clasificar el número, menor a mayor (ascendente) o de mayor a menor (descendente). Por defecto, el orden es descendente (0).

## TOPN(N\_Value, Table [,OrderBy\_Expression1] [,Order1] [,OrderBy\_Expression2] [,Order2]...)

La función **TOPN** devuelve las **N** filas superiores de una tabla específica, opcionalmente ordenadas por expresiones, ascendentes o descendentes. Los argumentos de Expresión se pueden basar en columnas o medidas. El argumento que sigue a **OrderBy\_Expression** es un valor (0 o 1) que especifica cómo ordenar la tabla. 0 ordena en orden descendente de valores del **OrderBy\_Expression**, y 1 los ordena en orden ascendente.

Como la función devuelve una tabla, deberá agregarse cuando se use en una columna o medida calculada. Los siguientes ejemplos sumarán la tabla devuelta por la función **TOPN** para calcular los tres primeros y los tres productos inferiores por ventas.

Las tres expresiones de medida son:

Ventas = SUM(VentasReales[ImporteVentas])

Mejores3Productos = SUMX (TOPN (3, Productos, [Ventas]), [Ventas])

Peores3Productos = SUMX (TOPN (3, Productos, [Ventas], 1), [Ventas])

En el siguiente ejemplo, la tabla dinámica presenta Ventas por **Tipo de Stock** en las filas, y las tres medidas Ventas, Mejores3Productos y Peores3Productos como valores.

	A	B	C	D
1				
2	<b>Tipo de Stock</b> ▼	<b>Ventas</b>	<b>Mejores3Productos</b>	<b>Peores3Productos</b>
3	Alto	6.734.689.763,77	106.075.463,69	131.925,19
4	Bajo	2.579.528.111,57	86.256.616,55	533.626,60
5	Medio	3.099.439.733,55	152.651.739,87	46.696,90
6	<b>Total General</b>	<b>12.413.657.608,89</b>	<b>344.983.820,11</b>	<b>712.248,69</b>

## LOOKUPVALUE(Result\_Column, Search\_ColumnName1, Search\_Value1 [,Search\_ColumnName2] [,Search\_Value2]...)

La función **LOOKUPVALUE** devuelve el valor en **Result\_Column** para la fila que cumple con todos los criterios especificados en los argumentos de búsqueda. Si ninguna coincidencia satisface los valores de búsqueda, se devuelve un valor en BLANCO.

Si varias filas coinciden con los valores de búsqueda, y todos los valores de **Result\_Column** son idénticos, se devuelve un error.

Esta función es útil en varios escenarios diferentes. Se usa comúnmente junto con las funciones padre-hijo para buscar el nombre del miembro con una clave.

Consulte el ejemplo de la función **PATHITEM** en la sección Padre e hijo en próximas lecciones.



## ALLSELECTED()

La función **ALLSELECTED** obtiene el contexto que representa todas las filas y columnas en la consulta, al tiempo que mantiene filtros y contextos explícitos distintos de los filtros de fila y columna. Esta función se puede usar para obtener totales visuales en consultas. No requiere argumentos ni una referencia de tabla o columna.

En el siguiente ejemplo, la tabla dinámica presenta Sales by StockTypeName en las filas e incluye la capacidad de filtrar el resultado utilizando el slicer ClassName. Cuando se seleccionan todos los valores ClassName (es decir, no hay filtro), este informe funciona bien. Las tres expresiones de medida son:

Ventas =SUM(VentasReales[ImporteVentas])  
Total\_Ventas\_ALL =CALCULATE([Ventas], ALL(VentasReales))  
RatioVentas\_ALL =[Ventas]/[Sales\_ALL]

	A	B	C	D	E	F
1	Clase		Destino	Ventas	Total_Ventas_ALL	RatioVentas_ALL
2			Europa	6.734.689.763,77	12.413.657.608,89	54,25%
3	Ejecutiva		Asia	2.579.528.111,57	12.413.657.608,89	20,78%
4	Turística		América	3.099.439.733,55	12.413.657.608,89	24,97%
5			<b>Total General</b>	<b>12.413.657.608,89</b>	<b>12.413.657.608,89</b>	<b>100,00%</b>
6	VIP					
7						

Cuando se selecciona un valor de **Clase**, la medida **Total\_Ventas\_ALL** se evalúa usando todas las filas en la tabla **VentasReales**. Los valores del **Total General** para **Ventas** y **Total\_Ventas\_ALL** difieren, y el último cálculo se basa en todas las filas de **VentasReales**.

El filtro **Clase** no se aplica al cálculo **Total\_Ventas\_ALL**.

	A	B	C	D	E	F
1	Clase		Destino	Ventas	Total_Ventas_ALL	RatioVentas_ALL
2			Europa	1.246.004.902,20	12.413.657.608,89	10,04%
3	Ejecutiva		Asia	656.379.560,65	12.413.657.608,89	5,29%
4	Turística		América	1.201.237.128,45	12.413.657.608,89	9,68%
5			<b>Total General</b>	<b>3.103.621.591,30</b>	<b>12.413.657.608,89</b>	<b>25,00%</b>
6	VIP					
7						

Ahora, veamos que sucede si agregamos dos medidas adicionales y la medida **VentasSeleccionadas** usando la función ALLSELECTED..

Ventas\_Seleccionadas =CALCULATE([Ventas], ALLSELECTED())

RatioVentas\_Seleccionadas =[Ventas]/[ VentasSeleccionadas]

	A	B	C	D	E	F	G	H
1	Clase		Destino	Ventas	Total_Ventas_ALL	RatioVentas_ALL	Ventas_Seleccionadas	RatioVentas_Seleccionadas
2			Europa	1.246.004.902,20	12.413.657.608,89	10,04%	3.103.621.591,30	40,15%
3	Ejecutiva		Asia	656.379.560,65	12.413.657.608,89	5,29%	3.103.621.591,30	21,15%
4	Turística		América	1.201.237.128,45	12.413.657.608,89	9,68%	3.103.621.591,30	38,70%
5	VIP		Total General	3.103.621.591,30	12.413.657.608,89	25,00%	3.103.621.591,30	100,00%
6								
7								

La medida **Ventas\_Seleccionadas** toma en consideración el filtro Clase y los valores del **Total General** para **Ventas** y **Ventas\_Seleccionadas** son los mismos. Asimismo, los valores **RatioVentas\_Seleccionadas** se suman al 100%. Estos cálculos adicionales que utilizan la función **ALLSELECTED** producen el efecto de los totales visuales.

Cuando se usa la función ALLSELECTED, la función siempre devolverá el total visual independientemente de lo que esté en las filas y la columna.