

DAX (II)

Funciones básicas DAX

Jose Ignacio González Gómez
Departamento de Economía Contabilidad y Finanzas - Universidad de La Laguna
www.jggomez.eu

Ejercicio Base: Operaciones Inmobiliarias Registradas

1. Operaciones con DAX y clasificación de las principales funciones
2. Principales funciones DAX
 - a. Funciones de agregación (SUM, AVERAGE, MAX y otras)
 - b. Funciones de conteo (COUNT, COUNTA, COUNTROWS y otras)
 - c. Funciones de Relación. Navegación entre tablas (RELATED y RELATEDTABLE)
 - d. Funciones Lógicas y Condicionales (AND, , IF, SWITCH y otras)
3. Funciones especiales DAX. Funciones de iteración (X) y Filtro (CALCULATE y otras)
 - a. Funciones de iteración (X). Características generales de las funciones de iteración (X)
 - b. Principales funciones de iteración escalares (SUMX, AVERAGEX, y otras)
 - c. Principales funciones de interacción tipo tabla (FILTER y otras)
 - d. Funciones de Filtro. CALCULATE
4. La tabla de medidas y el administrador de medidas

0. Recordando la necesidad de un adecuado modelo de datos

Antes de adentrarnos en el mundo de DAX, es importante tener presente la siguiente consideración:

«Muchos de los problemas que se nos pueden presentar en Power BI y Power Pivot y en la aplicación formulas DAX se pueden resolver con un buen Modelo de Datos»



En otras palabras, si uno construye un Modelo de Datos adecuado para su modelo de negocio, la cantidad y dificultad de las fórmulas DAX que tendrá que escribir es muy poca. Si por el contrario el Modelo de Datos es malo (léase: poco adecuado para el problema en cuestión), uno tendrá que escribir fórmulas monstruosas y, en consecuencia, ser una especie de «semidiós» del DAX.

Para evitar estos modelos de datos complejos como el de la ilustración conviene modelar tipo «estrella» en lugar de «copo de nieve»

1. Tipos de operaciones en DAX: aritméticas, de comparación, lógicas, operaciones de texto y otras

El tipo de operaciones que podemos realizar en nuestro modelo de datos son muy similares a las disponibles en Excel y que pasamos a presentar.

Aritmética

Suma	+
Sustracción	-
Multipliación	*
División	/
Potenciación	^

De comparación

Igualdad	=
Diferente de	<>
Mayor Que	>
Mayor o Igual Que	>=
Menor Que	<
Menor o Igual Que	<=

Operadores lógicos

Conjunción (Y)	&&	Similar a AND
Negación	!	Similar a NOT
Disyunción Inclusiva (O)		Similar a OR

Operadores de Texto Concatenación &

Presentación de las principales funciones DAX y su clasificación

Principales Funciones

1. Fecha y Hora
2. Matemáticas y Trigonométricas
3. Estadísticas
4. Texto
5. Lógicas
6. Filtro
7. Información
8. Primario / Secundario
9. **Time Intelligence**
10. **Misceláneas**

Categoría personal de Funciones

- a) Funciones de Agregación y Conteo
- b) Funciones de Filtrado
- c) Funciones de Relación. Navegación entre tablas.
- d) Funciones Lógicas y Condicionales
- e) Funciones de Fecha y Hora. Time Intelligence
- f) Otras funciones
 - Funciones de Información
 - Funciones de Texto

2. Principales Funciones DAX. 2.1 Funciones de agregación (SUM, AVERAGE, MAX..)

Se encargan de realizar un cálculo sobre una columna afectando en principio a todas ellas, salvo que indiquemos algún tipo de filtro o desglose descriptivo. Las funciones de agregación siempre se aplican a la columna entera y se expanden si se agregan más registros.

Vamos a crear la siguiente columna calculada con la función SUM:

=SUM(TABLA_Pedidos[Costo Total]).

Principales funciones

Ejemplo-SUM

= SUM(Sales[Amt])

- Se agregan todos los números contenidos en la columna Amt de la tabla Sales.

SUM(<Columna>)	Suma todas las filas de una columna. = SUM([ImporteVenta])
AVERAGE(<Columna>)	Devuelve el promedio de todos los números de una columna. = AVERAGE([ImporteVenta])
MIN(<Columna>)	Devuelve el menor valor numérico de una columna que tenga datos numéricos o fechas. = MIN(Ventas[TiendaSK])
MAX(<Columna>)	Devuelve el mayor valor numérico de una columna que tenga datos numéricos o fechas. = MAX(Ventas[TiendaSK])

Ejemplo SUM - Si queremos saber el total de ventas de la empresa, crearemos una medida utilizando la función SUM. Si creamos una tabla dinámica conectada a nuestro modelo en Power Pivot y arrastramos “TotalVentas” al área de valores, nos muestra el resultado de sumar “[ImporteVenta]” para todas las filas de la tabla y ya formateado.

fx TotalVentas:=SUM(Ventas[ImporteVenta])	
Cantidad	ImporteVenta
1	0,74
1	0,74
1	0,74
1	0,74
1	0,74
TotalVentas: 1.078.749,83	

Aplicar a ejercicio base

Ejemplo – AVERAGE

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

AVERAGE(InternetSales[ExtendedSalesAmount])

Ejemplo – MAX y MIN

En este caso se devuelve el escalar del valor mayor encontrado en la columna ExtendedAmount de la tabla InternetSales.

= MAX(InternetSales[ExtendedAmount])

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

= Max([TotalSales], [TotalPurchases])

Notas relacionadas: se admiten los valores TRUE o FALSE. Si quiere evaluar una columna de valores TRUE/FALSE, use la función MAXAç

2.2 Funciones de conteo (COUNT, COUNTA, COUNTROWS y otras..)

Estas funciones tienen como objetivo contar todos los elementos de una columna o tabla que cumplan una serie de condiciones. Las más relevantes son las siguientes:

COUNT(<Columna>)	Cuenta el número de filas de una columna que tenga números o fechas. = COUNT([ImporteVenta])
COUNTA (<Columna>)	Cuenta el número de celdas que tienen números, valores lógicos y texto en columna.
COUNTBLANK (<Columna>)	Cuenta el número de celdas vacías en una columna.
COUNTROWS(<Tabla>)	Cuenta el número de filas de una tabla. = COUNTROWS('Cliente') = COUNTROWS(RELATEDTABLE('Ventas'))
DISTINCTCOUNT(<Columna>)	Cuenta el número de filas con valores distintos de una columna. = DISTINCTCOUNT(Ventas[Ticket])

Aplicar a ejercicio base

Ejemplos de funciones de conteo

= COUNT([ShipDate])

Cuenta el número de valores de la columna ShipDate

= COUNTA('Reseller'[Phone])

Devuelven todas las filas de la tabla Reseller que tienen cualquier tipo de valor en la columna que almacena números de teléfono y que no están vacías. Dado que el nombre de la tabla no contiene espacios, las comillas son opcionales

Ejemplos de funciones de conteo

[Número de Registros]:=COUNTROWS
('Pedidos')

Cuenta el número de filas de la tabla de pedidos

Ejemplo de COUNTROWS combinada con FILTER

[Ventas de Alta Rentabilidad]:=COUNTROWS (FILTER (TABLA_Pedidos; TABLA_Pedidos[Precio de Venta] >= 40))

La función FILTER retorna una tabla con solamente aquellas filas que en la columna Precio de Venta tienen un valor mayor o igual a 40 y, como esta encapsulada en la función COUNTROWS, entonces la medida retorna el número de ventas con un precio mayor o igual a 40 para el contexto actual

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

Otro ejemplo

[Fecha] 4 Countrows:=COUNTROWS(Tabla_D_partes)							
	id	Fecha	Cliente	Activ...	Horas	Personal	
1	1	15/01/2016 0:00:00	6	7	150	5	
2	2	22/02/2016 0:00:00		7	20	5	
3	3	03/02/2016 0:00:00	0	7	160	5	
4	4	04/04/2016 0:00:00	8	7	90	5	
		1 Count: 33					
		2 CountA: 33					
		3 Countblank: 1					
		4 Countrows: 33					
		5 Distinctcount: 20					

- 1 Count:=Count(Tabla_D_partes[Horas])
- 2 CountA:=COUNTA(Tabla_D_partes[Horas])
- 3 Countblank:=COUNTBLANK(Tabla_D_partes[Cliente])
- 4 Countrows:=COUNTROWS(Tabla_D_partes).
- 5 Distinctcount:=DISTINCTCOUNT(Tabla_D_partes[Horas])

2.3 Funciones de Relación. Navegación entre tablas (RELATED y RELATEDTABLE)

Estas funciones nos permiten navegar y obtener valores de columnas de diversas tablas del modelo siempre que existan relaciones entre ellas.

RELATEDTABLE se utiliza en conjunto con otras funciones

Ejemplo RELATED(<Columna>)

= RELATED(Geografia[Poblacion])

Devuelve un valor relacionado de la columna de otra tabla siguiendo la relación M->1. Ejemplo: agregar una columna calculada con la “[Poblacion]” a la tabla “Tienda”

Ejemplo RELATEDTABLE(<Tabla>)

= COUNTROWS(RELATEDTABLE('Cliente'))

Devuelve una tabla en un contexto especificado por los filtros indicados, siguiendo la relación 1->M. Ejemplo: agregar una columna con el número de clientes a la tabla “Geografia”:

2.4 Funciones Lógicas y Condicionales (AND, , IF, SWITCH y otras).

Estas funciones son similares a las de Excel. Presentamos las más relevantes y su equivalente en Excel

Función Excel	Power Pivot DAX	Descripción
Y	AND	Comprueba si todos los argumentos son VERDADEROS y devuelve VERDADERO o FALSO. Devuelve FALSO si alguno de los argumentos es FALSO
Falso	FALSE	Devuelve el valor lógico FALSO
SI	IF	Comprueba si se cumple una condición y devuelve un valor si se evalúa como VERDADERO y otro valor si se evalúa como FALSO.
SI.ERROR	IFERROR	Retorna lo especificado en el segundo argumento si hay un error. Devuelve un valor si la expresión es un error y otro valor si no lo es
NO	NOT	Cambia FALSO por VERDADERO y VERDADERO por FALSO.
O	OR	Comprueba si alguno de los argumentos es VERDADERO y devuelve VERDADERO o FALSO. Devuelve FALSO si todos los argumentos son FALSO
Verdadero	TRUE	Devuelve el valor lógico VERDADERO

Ejemplo IF con RELATED(<Columna>)

En el siguiente ejemplo vamos a crear una columna calculada (Comision) en la tabla D-Partes de nuestro modelo de datos, que en función de la categoría del empleado se asigne un porcentaje de comisión, si es Socio el 10% y si es Personal de Apoyo el 3%. Por tanto tendremos que establecer una formula y en función del código del empleado me consulte en la tabla Personal que categoría tiene y en base a ello aplique la comisión correspondiente.

=IF(RELATED(Tabla_A_personal[Categoria])="Socio";0,1;0,03)

	Fecha	Cliente	Activ...	Horas	Personal	Comision	Objetivo Medio	Objetivo Alto	Obje
1	15/01/2016 0:00:00	6	7	150	3	10,00 %	367	403,7	
2	22/02/2016 0:00:00		7	20	2	10,00 %	367	403,7	
3	03/02/2016 0:00:00	0	7	160	5	3,00 %	367	403,7	
4	04/04/2016 0:00:00	8	7	90	5	3,00 %	367	403,7	
5	05/05/2016 0:00:00	9	8	180	1	10,00 %	367	403,7	
1 Count: 33									
2 CountA: 33									

Se podría leer la fórmula de la siguiente forma:

Relaciona la tabla Personal y su columna Categoría con el valor de la Columna Personal de la tabla Partes y si el valor de correspondencia es Socio, en ese caso le asignas la comisión del 10% y en caso contrario le asigna un 3%. En este caso se utiliza la propiedad RELATED para relacionar campos de otras tablas relacionadas.

Ejemplo uso de la función IF, SWITCH y TRUE

IF - Solo una sencilla columna que acorta las palabras, una abreviación, esto capta la esencia de la función IF. Creemos la siguiente Columna Calculada.

[Abreviación T. Compra] =IF([Tipo de Compra]="Normal"; "V"; IF([Tipo de Compra]="Devolución";"D"))

SWITCH - Ahora la misma expresión, pero utilizando SWITCH.

[Abreviación T. Compra]=SWITCH([Tipo de Compra]; "Normal"; "N"; "Devolución";"D")

La función SWITCH es mucho más fácil de leer que su “melliza” IF, sobre todo cuando hay una gran cantidad de condicionales. Pero tenga en cuenta que la función IF es más eficiente, si tiene una base de datos voluminosa (10 millones de datos más o menos) utiliza IF.

SWITCH con TRUE - Si deseas probar múltiples condicionales con la función SWITCH utilice TRUE() en el primer argumento y en los demás realiza las comparaciones usando los operadores lógicos.

=SWITCH(TRUE(); [Campo1]="A" &&[Campo2]="B", “Resultado 1”[Campo1]="B" &&[Campo2]="C", “Resultado 2”)

Otro ejemplo de Swith

En el siguiente ejemplo se evalúa el país al que se asocia la venta para generar una columna personalizada que contenga el continente.

Fuente:

<http://www.interactivechaos.com/dax/function/switch>



```
Continente = SWITCH(
    Tabla1[País];
    "España"; "Europa";
    "Alemania"; "Europa";
    "Argentina"; "América";
    "Brasil"; "América";
    "Estados Unidos"; "América";
    "Desconocido"
```

Fecha	Precio	Ciudad	País	Continente
miércoles, 17 de febrero de 2016	1.480 €	Toledo	España	Europa
viernes, 25 de marzo de 2016	2.530 €	Berlín	Alemania	Europa
sábado, 30 de abril de 2016	650 €	Barcelona	España	Europa
sábado, 2 de abril de 2016	920 €	Buenos Aires	Argentina	América

Otro ejemplo de Switch

Si quisiéramos clasificar cada categoría según la cifra total de ventas con una escala del 1 al 3 (indicándolo con símbolos "\$", por ejemplo), podríamos hacerlo usando la función SWITCH de la siguiente manera:

```
Clasificación = SWITCH(  
    TRUE();  
    [Ventas]>1000000000;"$$$";  
    [Ventas]>100000000;"$";  
    [Ventas]>0;"$";  
    BLANK()  
)
```

En esta medida, comparamos las ventas totales (del contexto actual) con una cifra, y devolvemos un valor u otro en función de que la comparación se cumpla o no. El resultado es el siguiente:

ProductSubcategoryName	Ventas	Clasificación
Bluetooth Headphones	41.907.488,91 €	\$
Camcorders	1.335.302.769,92 €	\$\$\$
Cameras & Camcorders Accessories	51.643.775,83 €	\$
Car Video	306.818.844,52 €	\$\$

3. Funciones especiales DAX. Funciones de iteración (X) y Filtro (CALCULATE y otras)

3.1 Funciones de iteración (X).), características generales

Ya hemos visto las funciones simples **SUM**, **AVERAGE**, **COUNT.**; pero también es importante comprender cómo usar las funciones iterativas que tienen una X al final, como: **SUMX**, **AVERAGEX** o **COUNTX**.

Una función de iteración es una función de DAX que generalmente tiene una X al final. También existen funciones de tabla que iteran a través de una tabla como **FILTER** o **VALUES**, pero en este caso nos ocuparemos únicamente de las que terminan en X.

Las funciones de iteración nos brindan mucha más flexibilidad en términos de los cálculos que puede hacer a nivel de fila en comparación con las agregaciones que solo le permiten agregar una columna completa. Ahora, dentro de las funciones de iteración, primero debe ingresar una referencia de tabla y luego una expresión. Si aprende esto, entonces las oportunidades de ejecutar cálculos internos en la memoria a través de medidas se vuelve una herramienta realmente poderosa.

3. Funciones especiales DAX. Funciones de iteración (X) y Filtro (CALCULATE y otras)

3.1 Funciones de iteración (X), características generales.

Función	Principales Funciones - Descripción
SUMX	Retorna la suma de los valores generados en un proceso iterativo.
AVERAGEX	Retorna la media aritmética de los valores generados en un proceso iterativo.
MAXX	Retorna el valor más grandes de los valores generados en un proceso iterativo.
CONCATENATEX	Une los textos generados en un proceso iterativo.
COUNTX	Cuenta el número de valores generados en un proceso iterativo.
MEDIANX	Retorna la mediana de los valores generados en un proceso iterativo.
PRODUCTX	Retorna el producto de los valores generados en un proceso iterativo.
RANKX	Retorna la clasificación (1 a N) de un valor contra los valores generados en un proceso iteraitvo.
STDEVX.P	Retorna la desviación estándar poblacional de los valores generados en un proceso iterativo.

3.2 Principales funciones de iteración escalares (SUMX, AVERAGEX, y otras)

Objetivo: Para operar (por ejemplo, multiplicar) dos columnas de una misma fila de una tabla (precio unitario por cantidad). También permite filtrar los valores que va a sumar, puede usar la función SUMX y especificar una expresión que sumar. Devuelve la suma de una expresión evaluada para cada fila de una tabla.

Sintaxis: [SUMX](#)(<table>, <expression>), como podemos observar tiene dos parámetros

Ejemplo SUMX operando sola. *Creacion de la medida, total ventas como precio por cantidad*

Disponemos de la siguiente tabla, donde tenemos el precio unitario y cantidad de cada producto. Si quisiéramos obtener el total de ventas como multiplicación de la columna precio por cantidad utilizando la función SUM como:

$$\text{SUM}(\text{Ventas}[\text{CANTIDAD}]*\text{Ventas}[\text{PPRECIO}])$$

No se puede porque SUM solo permite una columna

Producto	Precio	Cantidad
Mandarina	15	17
Melón	20	10
Naranja	25	5
Total	60	32

Si para salvar esta limitación creamos la siguiente medida obtendríamos la siguiente tabla.

```
1 Total Ventas = SUM( Ventas[Precio] ) * SUM( Ventas[Cantidad] )
```

Si para salvar esta limitación creamos la siguiente medida obtendríamos la siguiente tabla.

Hemos utilizado dos (2) veces la función SUM para realizar la multiplicación. Y a nivel de filas se presenta adecuadamente, pero ¿Qué ocurre con la totalización? El valor proviene de la multiplicación de $60 \times 32 = 1.920$. Y aunque matemáticamente es correcto, no es el resultado esperado. El problema se origina porque la función SUM no trabaja en un contexto de fila.

Producto	Precio	Cantidad	Total Ventas
Mandarina	15	17	255
Melón	20	10	200
Naranja	25	5	125
Total	60	32	1.920

Podríamos leer lo que ha sucedido de la siguiente forma:

Vamos a sumar todos los registros visibles de la columna precio, luego haremos lo mismo en la columna cantidad, y al final multiplicaremos los totales.

Solución: Hemos creado la siguiente media para calcular el total de ventas como resultado de la multiplicación de la columna precio unitario por cantidad. Tal y como se muestra a continuación

```
1 Total Ventas = SUMX( Ventas; Ventas[PRECIO] * Ventas[CANTIDAD] )
```

Producto	Precio	Cantidad	Total Ventas
Mandarina	15	17	255
Melón	20	10	200
Naranja	25	5	125
Total	60	32	580

Ejemplo SUMX con RELATED I

Vamos a ver como actúa una función de iteración combinada con RELATED, navegando a un campo de otra tabla y operando.

ProductID	Fecha	Código Postal	Unidades	Ingresos	País	CPPaís	Producto
1086	viernes, 27 de diciembre de 2019	02771	1	\$89.20	USA	USA-02771	Pirum RP-32
1086	viernes, 27 de diciembre de 2019	06812	1	\$89.20	USA	USA-06812	Pirum RP-32
1086	viernes, 27 de diciembre de 2019	08230	1	\$89.20	USA	USA-08230	Pirum RP-32
1086	viernes, 27 de diciembre de 2019	21120	1	\$89.20	USA	USA-21120	Pirum RP-32
1086	viernes, 27 de diciembre de 2019	21225	1	\$89.20	USA	USA-21225	Pirum RP-32
1086	viernes, 27 de diciembre de 2019	29588	1	\$89.20	USA	USA-29588	Pirum RP-32
1086	viernes, 27 de diciembre de 2019	32025	1	\$89.20	USA	USA-32025	Pirum RP-32
1086	viernes, 27 de diciembre de 2019	34990	1	\$89.20	USA	USA-34990	Pirum RP-32
1086	viernes, 27 de diciembre de 2019	35592	1	\$89.20	USA	USA-35592	Pirum RP-32

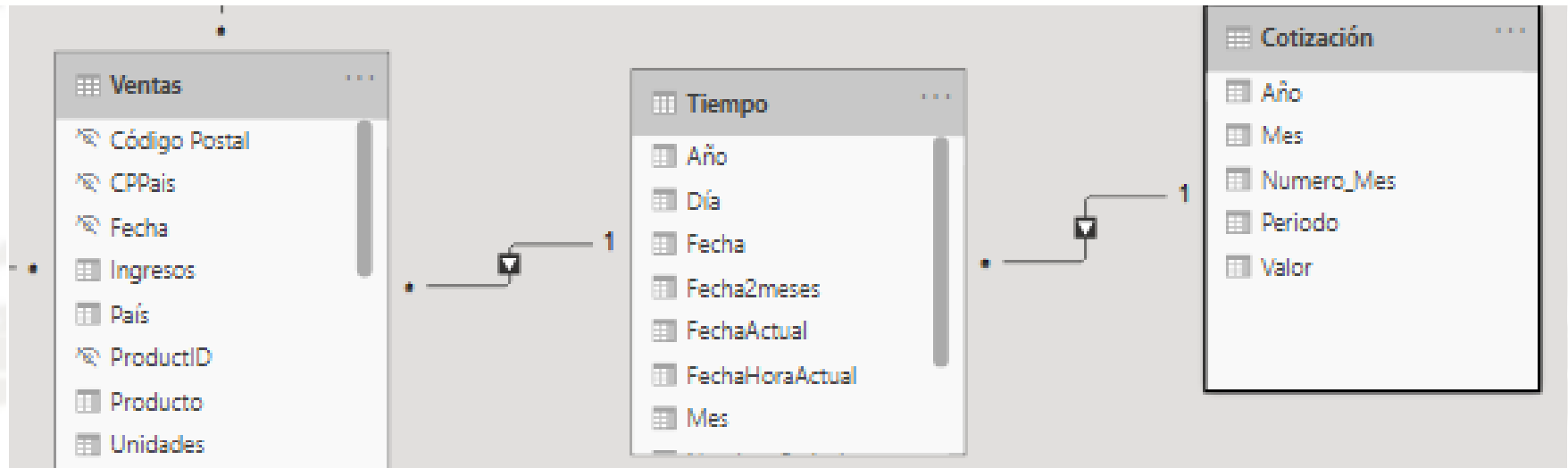
Fuente: <https://inteligencia.dds.pe/dax/funciones-de-iteracion-en-dax/>

Contamos con una tabla de ventas con los ingresos en dólares y otra tabla de cotización de €-Soles y ambas tablas están relacionadas con una tabla calendario (Tiempo). Para obtener los ingresos en la moneda soles se usará la función de iteración **SUMX**.

En este caso, para cada fila de la tabla Ventas, multiplicamos los ingresos en dólares por la tasa de conversión en ese periodo usando la función **RELATE** para obtener la tasa de conversión de cada fila de la tabla ventas.

Año	Mes	Valor	Numero_Mes	Periodo
1992	enero	0.98	1	199201
1992	febrero	0.96	2	199202
1992	marzo	0.96	3	199203
1992	abril	1.05	4	199204
1992	mayo	1.13	5	199205
1992	junio	1.19	6	199206
1992	julio	1.23	7	199207
1992	agosto	1.29	8	199208
1992	setiembre	1.47	9	199209
1992	octubre	1.58	10	199210
1992	noviembre	1.61	11	199211
1992	diciembre	1.63	12	199212

```
IngresosSoles = SUMX(Ventas,Ventas[Ingresos]*RELATED('Cotización'[Valor]))
```



Para este mismo caso veamos otra aplicación interesante relacionada con el uso de iteradores, en concreto vamos a crear una medida que nos de los **ingresos superiores a 200\$**.

```
Ingreso>200 =  
SUMX(  
    FILTER(Ventas,Ventas[Ingresos]>200),[IngresosSoles])
```

Con Filter se crea primero una tabla o subtabla de ventas con aquellos registros cuyas ventas son mayores de 200\$ y mostramos las ventas en soles para cada fila.

3.3 Principales funciones de iteración tipo tabla (Filter y otras)

Presentamos una relación básica de las funciones de iteración tipo tabla más relevantes.

Función	Descripción
<u>FILTER</u>	Retorna una tabla filtrada de acuerdo con una expresión booleana ejecutada fila a fila.
SUMMARIZE	Retorna una tabla de resumen de acuerdo a las columnas especificadas.
TOPN	Retorna el top de elementos de acuerdo a la expresión señalada.

Ejemplo SUMX con FILTER I.

Vamos a ver como actúa una función de iteración combinada con FILTER, filtrando y operando.. Supongamos que queremos tener el Importe de las ventas del 2013 para poder compararlo con los otros años, o sea ventas 2014/ventas 2013, ventas 2015/ventas2013.

Creemos una nueva medida y ponemos:

Importe 2013 = sumx(FILTER('Facturas ventas';'Facturas ventas'[Ejercicio]=2013);'Facturas ventas'[Neto])

Donde el primer parámetro es un subconjunto de la tabla Facturas ventas usando la función FILTER

Si ponemos este Importe 2013 en la tabla tenemos:

Representante	Neto	Importe	Importe sumado	Importe 2013
Fernando Gómez Garcia	2.588.605,26	2.588.605,26	2.588.605,26	988.477,46
Juanjo Pérez Romero	1.942.113,32	1.942.113,32	1.942.113,32	678.026,84
Luis Fernandez Pérez	1.623.809,34	1.623.809,34	1.623.809,34	600.274,86
Maria López López	1.710.700,00	1.710.700,00	1.710.700,00	602.974,66
Total	7.865.227,92	7.865.227,92	7.865.227,92	2.869.753,82

Ejemplo SUMX con FILTER II

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

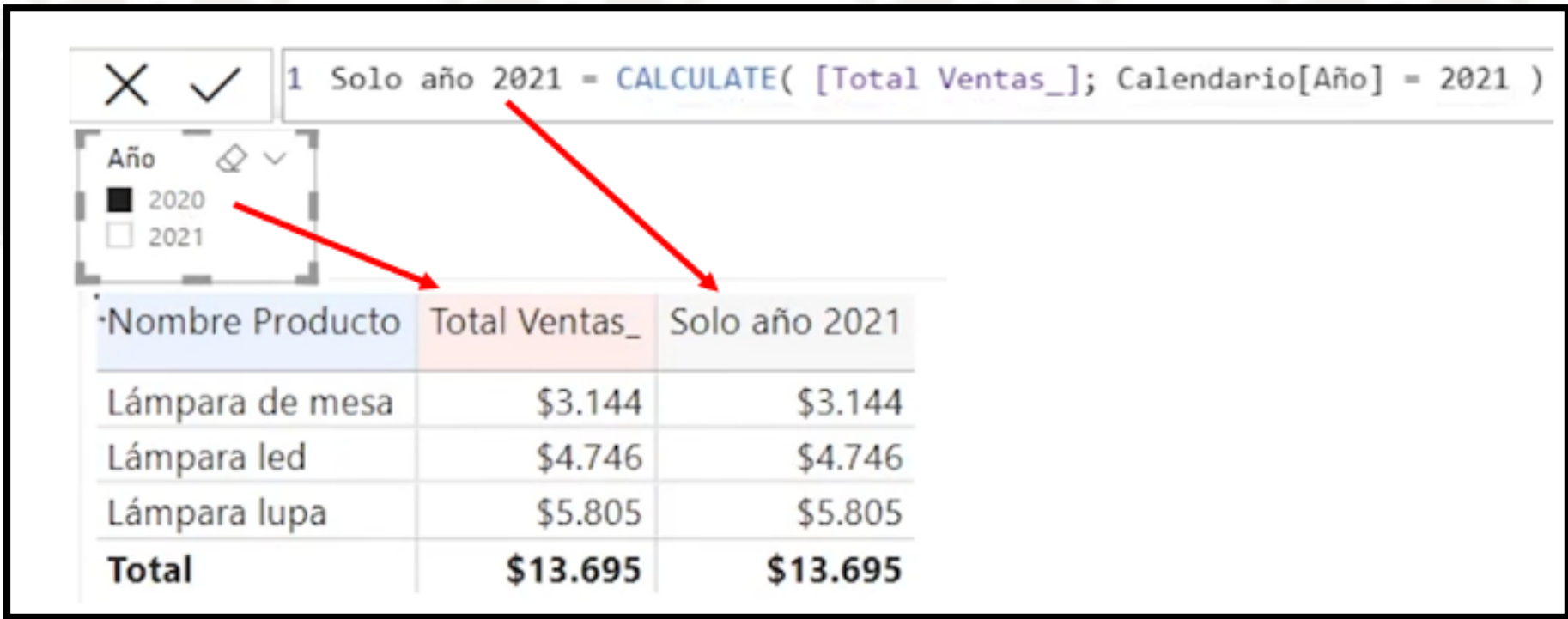
En este ejemplo se filtra primero la tabla, InternetSales, en la expresión SalesTerritoryID = 5 y, después, se devuelve la suma de todos los valores de la columna Freight. En otras palabras, la expresión devuelve la suma de los cargos de flete solo para el área de ventas especificada.

3.4 Funciones de Filtro. CALCULATE

La función mas importante de DAX

La función CALCULATE es considerada una de las más relevantes de DAX por la capacidad que tiene de cambiar o modificar el contexto de filtrado y la cantidad de escenarios en la cual actúa son innumerables.

Ejemplo base: En este caso hemos generado una medida para que nos de las ventas totales del año 2021 sin que le afecte en filtro del campo fecha y la hemos dispuesto la tabla, tal y como se muestra



1 Solo año 2021 = CALCULATE([Total Ventas_]; Calendario[Año] = 2021)

Año

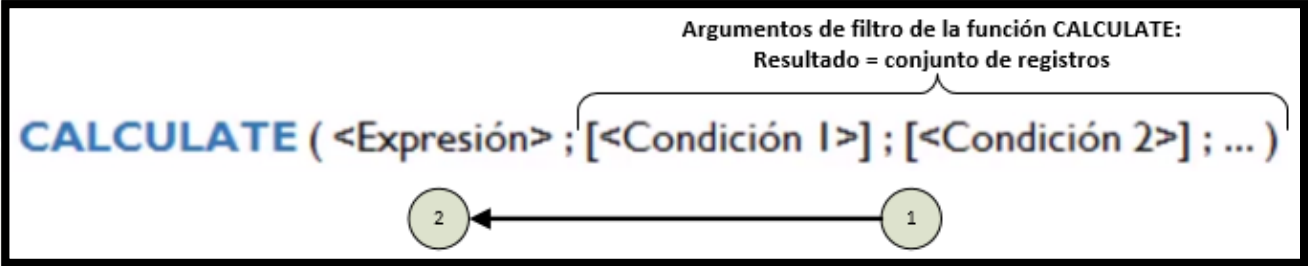
☒ 2020

☐ 2021

Nombre Producto	Total Ventas_	Solo año 2021
Lámpara de mesa	\$3.144	\$3.144
Lámpara led	\$4.746	\$4.746
Lámpara lupa	\$5.805	\$5.805
Total	\$13.695	\$13.695

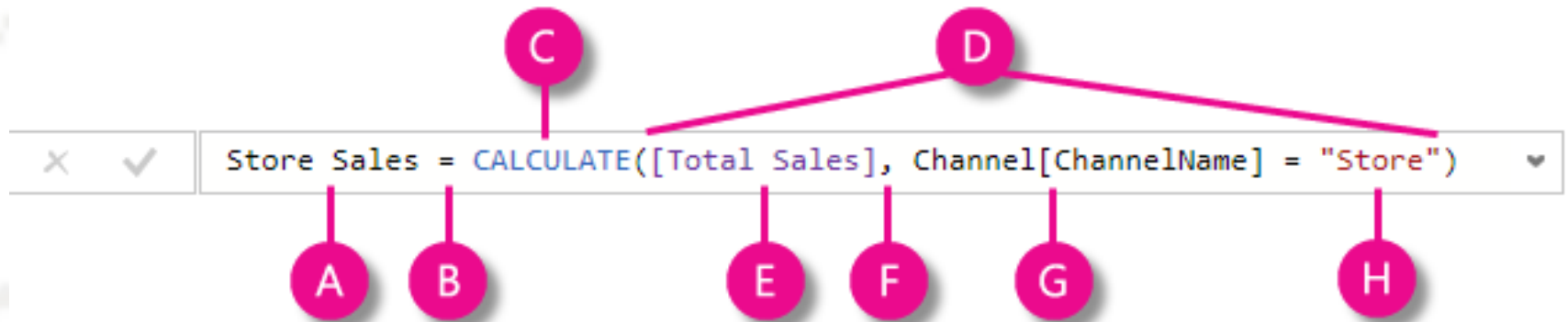
Definición y sintaxis:

La función CALCULATE evalúa una expresión, como, por ejemplo: SUMA, CUENTA, MEDIANA, etc. Añadiendo, removiendo o modificando coordenadas del query context. La función CALCULATE tiene un número indefinido de *argumentos, parámetros o condiciones*:



El único parámetro obligatorio es la expresión que puede ser SUM o cualquier otro e incluso anidaciones de funciones y por otro lado el grupo de condiciones que pueden ser infinitas

La función se “procesa” de la siguiente forma, primero se evalúa el contexto de filtro que da como resultado un registro o conjunto de registros de valores y sobre estos se aplica la expresión, de esta forma se procede de derecha a izquierda.



Esta fórmula incluye los siguientes elementos de sintaxis:

- A. El nombre de medida, Store Sales.
- B. El operador signo igual (=) que indica el principio de la fórmula.
- C. La función CALCULATE, que evalúa una expresión, como argumento, en un contexto modificado por los filtros especificados.
- D. Los paréntesis () que envuelven una expresión que contiene uno o más argumentos.
- E. Una medida [Total Sales] en la misma tabla como una expresión. La medida Total Sales tiene la fórmula: =SUM(Sales[SalesAmount]).
- F. Una coma (,), que separa el primer argumento de expresión del argumento de filtro.
- G. La columna completa a la que se hace referencia, Channel[ChannelName] . Se trata de nuestro contexto de fila. Cada fila de esta columna especifica un canal como Store, Online, etc.
- H. El valor concreto Store como filtro. Este es el contexto de filtro.

Ejemplos con CALCULATE

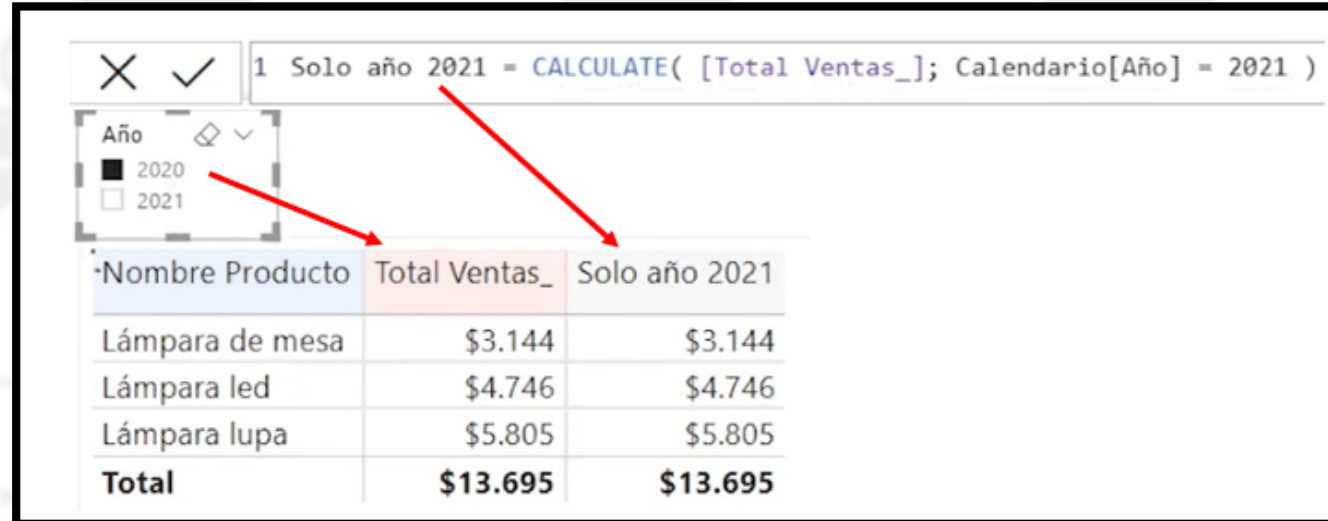
Ejemplo CALCULATE

Ejemplo: `CALCULATE(SUM(Sales[Margin]), Sales[Year]=2001)`

Ejemplo: `CALCULATE([Sales per Day], Sales[Year]=2002, Sales[ProductKey]=313)`

Ejemplo CALCULATE

En este caso hemos generado una medida para que nos de las ventas totales del año 2021 sin que le afecte el filtro del campo fecha y la hemos dispuesto en nuestra tabla, tal y como se muestra.



1 Solo año 2021 = CALCULATE([Total Ventas_]; Calendario[Año] = 2021)

Nombre Producto	Total Ventas_	Solo año 2021
Lámpara de mesa	\$3.144	\$3.144
Lámpara led	\$4.746	\$4.746
Lámpara lupa	\$5.805	\$5.805
Total	\$13.695	\$13.695

Ejemplo CALCULATE

Filtrar para un país concreto, queremos crear una medida que siempre devuelva los ingresos para el país Colombia independientemente de la selección hecha para un país.

[Ingresos COLOMBIA]:=

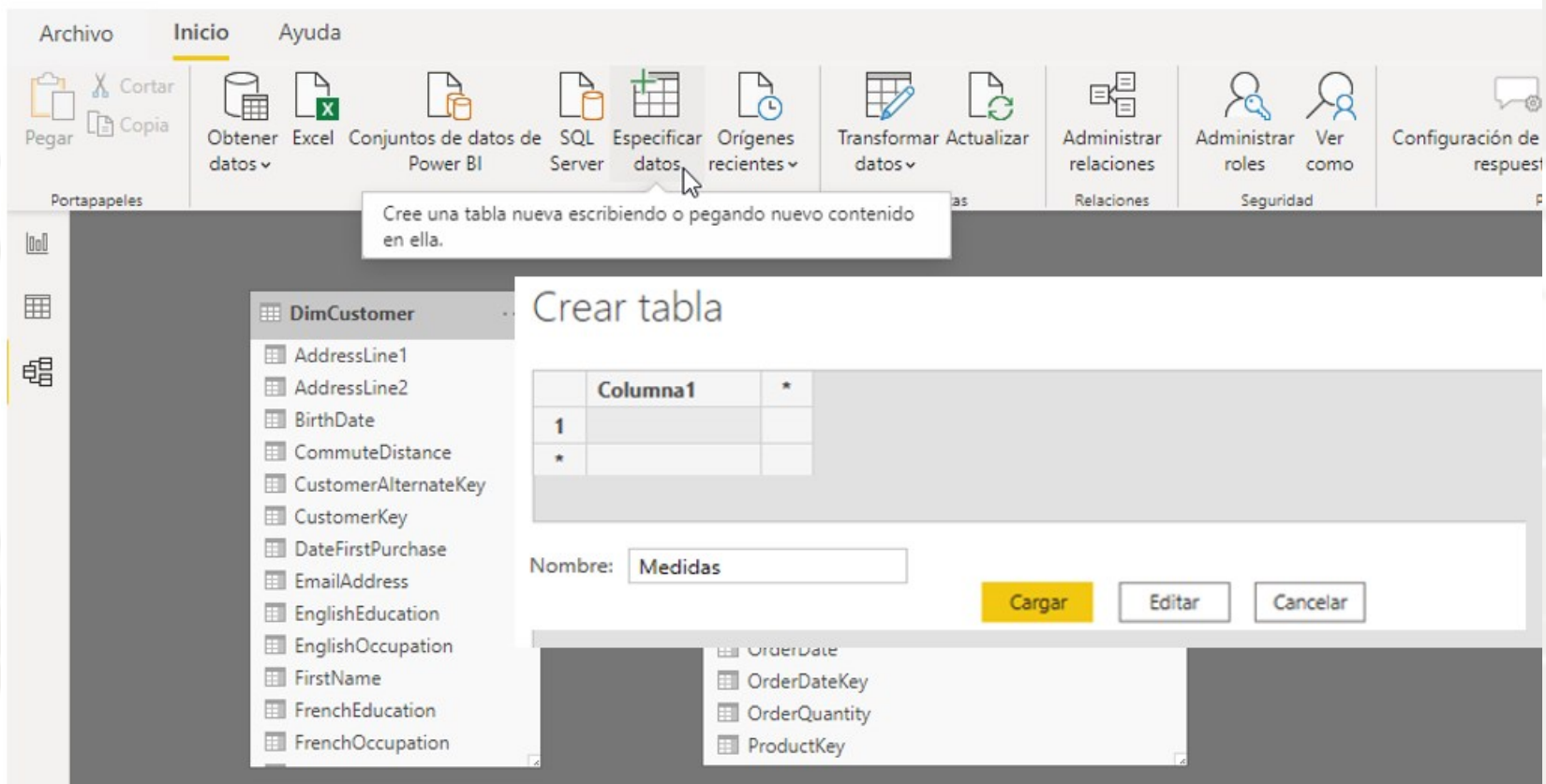
`CALCULATE ( SUM ( Pedidos[Ingresos] ); Pedidos[País] = "Colombia" )`

4. La tabla de medidas y el administrador de medidas

Por defecto cualquier medida al crearse se coloca en la tabla desde la que la hemos generado y en la mayoría de ocasiones esto podría convertirse en una situación caótica a medida que estas van aumentando ya que estarían dispersas en las diversas tablas de nuestro modelo y mezcladas con los múltiples campos..

Por tanto, es aconsejable disponer de un “almacén central” que agrupe las distintas medidas creadas que facilite su localización y empleo creando para ello una Tabla de Medidas.

Para su creación accedemos a Power Query o bien a través de la vista modelo de Power BI y con la misma opción, “Especificar Datos” procedemos a crear la tabla que agrupara nuestras medidas.



Archivo Inicio Ayuda

Pegar Cortar Copia

Obtener datos Excel Conjuntos de datos de Power BI SQL Server Especificar datos Orígenes recientes Transformar Actualizar

Administración de relaciones Administración de roles Ver como Configuración de respuestas

Portapapeles

Cree una tabla nueva escribiendo o pegando nuevo contenido en ella.

Crear tabla

	Columna1	*
1		
*		

Nombre: Medidas

Cargar Editar Cancelar

DimCustomer

- AddressLine1
- AddressLine2
- BirthDate
- CommuteDistance
- CustomerAlternateKey
- CustomerKey
- DateFirstPurchase
- EmailAddress
- EnglishEducation
- EnglishOccupation
- FirstName
- FrenchEducation
- FrenchOccupation

OrderDate

- OrderDateKey
- OrderQuantity
- ProductKey

Esta nueva tabla “Medidas” por tanto servirá de contenedor de nuestras medidas y a medida que vayamos trabajando y creando nuestras visualizaciones y necesitemos nuevas medias estas se deberían ir alojando en este almacén

Por defecto nos crea una primera columna que posteriormente eliminaremos una vez hayamos creado las medidas necesarias.

El Administrador de Medidas.

	Medidas
	Cantidad Clientes
	Columna1
	Total Ventas

Configuración de medida

Nombre de la tabla:

Sales

Nombre de medida (todas las tablas dinámicas):

Customers
Sales
Products
Medida 1

Nombre personalizado (esta tabla dinámica):

Descripción:

Fórmula:

f_x

Comprobar fórmula

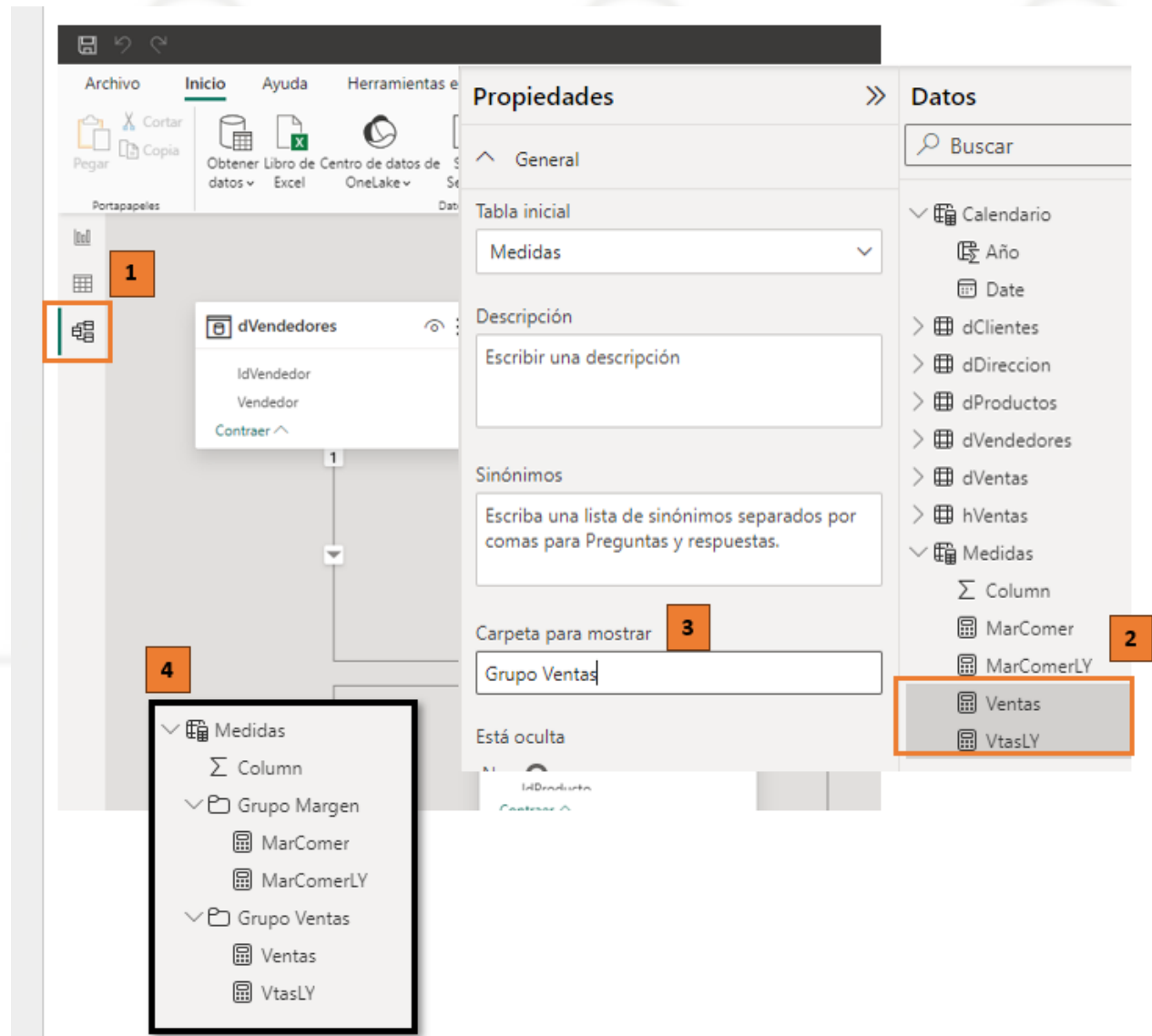
=

Opciones de formato

Categoría:

General
Número
Moneda
Fecha
TRUE/FALSE

Organizando las medidas dentro de la tabla medida, creando grupos o carpetas



1. Accedemos a la vista de modelo
2. Seleccionamos las medidas agrupar en carpeta (con Ctrl seleccionamos varias)
3. Vamos a la pestaña Propiedades-General y elegimos el nombre de la carpeta
4. Resultado final del agrupamiento.