

Función MINX

Compatibilidad



Microsoft Excel
Excel ≥ 2010

★★★★★

Power BI Desktop
PBI ≥ Nov 2016

★★★★★

SQL Analysis Services
SSAS ≥ 2012

DIRECTQUERY: C.Calculadas  Medidas 

ROW LEVEL SECURITY: 

Int. Contexto

Contexto de Filtro

Tiene en cuenta el contexto de filtro

★★★★★

Contexto de Fila

Ignora contextos de filas previos

Tiene en cuenta su contexto de fila

Categorías

Según Proceso Interno
Iteración

★★★★★

Según Resultado
Escalar

Recursos de Aprendizaje



MAGÍSTER EN LEGUAJE DAX

→ [Capacitación OnLine] ←

<https://bit.ly/3bzlkG0>

 MINX: Dragón Oscuro del Valor Inferior

Descripción

La función **MINX** retorna el valor más pequeño de todos los valores parciales que «salieron» como resultado de una expresión que se evalúa fila a fila en una tabla.

Sintaxis



MINX (< Tabla > , < Expresión >)

1 Tabla

Puede ser algo tan sencillo como el nombre de una tabla, o algo más elaborado mediante una expresión tabular.

 Iterador  Expresión Tabular

Tipo

Obligatorio

Atributo

No Repetible 

2 Expresión

Una expresión: aritmética o algebraica para ser evaluada en una base fila a fila.

 Contexto de Fila

Tipo

Obligatorio

Atributo

No Repetible 



Retorna el valor máximo de todos los valores calculados individualmente fila por fila en la tabla, puede ser de cualquier tipo.

OBSERVACIONES

1. El primer parámetro es versátil, puesto que, como primera posibilidad puede aceptar simplemente la llamada de una tabla en el modelo:

1. ValorVentaMáxima =
2. **MINX** (Pedidos , Pedidos[Ingresos])



Colecciona Tus Diamantes DAX



Los 7 énfasis en el Lenguaje DAX son capacitaciones online y 100% en vivo cada una de 8 horas totales, donde en cada eslabón se dedica a profundizar en un pilar del lenguaje DAX de forma real y única con gran apoyo de la metodología de infografías animadas.

Cada énfasis es representado con una gema o diamante de color distintivo que sirve de indicador de nivel de experticia de cada participante, estos son:

Énfasis I:

Fundamentos y Contexto de Evaluación

Énfasis II:

CALCULATE y Modificación de Contexto

Énfasis III:

Cálculos de Inteligencia de Tiempo

Énfasis IV:

Contexto de Filtro y Jerarquías

Énfasis V:

Funciones Tabulares y Consultas

Énfasis VI:

Manejo de DAX Superior

Énfasis VII:

Relaciones Avanzadas en DAX

Como segunda posibilidad es licito indicar cosas más interesantes mediante expresiones tabulares.

- Ejemplo I — Expresión Tabular con [FILTER](#)

Si necesitamos elaborar una expresión que devuelve el valor de la venta mínima, pero sin tener en cuenta fines de semana, podemos sostenernos en el hecho de que en el primer parámetro de [MINX](#) podemos suministrar una tabla que ya está restringida sólo para los días de lunes a viernes, con la cual, la siguiente expresión, tendremos una opción:

```

1. ValorMayorVentaNoFDS =
2. MINX (
3.     FILTER (
4.         Pedidos ,
5.         NOT ( RELATED ( Calendario[DS Número] ) IN { 6, 7 } ) ),
6.     Pedidos[Ingresos]
7. ) -- Véase que el operador NOTIN no existe en DAX, sin embargo,
8.    -- Con invertir el resultado con NOT emulamos algo similar.
    
```

La expresión de tabla del primer parámetro puede ser tan compleja como necesitemos.

- Ejemplo — Expresión Tabular con [SUMMARIZE](#)

En ciertos cálculos es necesario mostrarlo agregado por un tamaño de grano más fino que el presentado por el reporte, un ejemplo específico, exponer en un reporte para un año determinado a nivel de meses el valor del día con menores ingresos. Una posible solución:

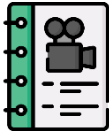
```

1. ValorDiaMenorVenta =
2. MINX (
3.     ADDCOLUMNS (
4.         SUMMARIZE ( Pedidos, Calendario[Día] ),
5.         "@Ingresos Totales", [Ing]
6.     ),
7.     [@Ingresos Totales]
8. ) -- Se asume un solo año seleccionado.
    
```





Otros Recursos de Aprendizaje



El ADN de Power Pivot

Capítulo número 6

→ [Visitar Libro] ←

El libro **El ADN de Power Pivot** es un manuscrito para estudiar los fundamentos y aspectos intermedios del lenguaje DAX, utilizando Excel y específicamente Power Pivot como herramienta cliente, para la creación de expresiones DAX cuyas soluciones se proyectan a través de tablas dinámicas, además, materializando tablas en la hoja de cálculo.

Es una primera guía para adentrarse en el mundo de DAX, sin embargo, si se requiere más información de la función TREATAS se debe optar por un video curso como el **Máster en DAX y Power Pivot** o capacitaciones online y 100% en vivo como: **Énfasis en Contexto de Filtro**.

→ <http://eladndepowerpivot.com/> ←



Detente, Un inciso:

La medida **ValorDiaMenorVenta** es diferente a:

1. **ValorVentaMínima** =
2. **MIN** (Pedidos[Ingresos])

Por el hecho de que la medida **ValorVentaMínima** devuelve el valor más pequeño de las transacciones discriminando cada una como diferente, mientras que la medida **ValorDiaMenorVenta** devuelve el valor más pequeño del agregado (en este caso suma) de las transacciones, que resulta ser a nivel de día.



Espera, Otro inciso:

Si hay varios años seleccionados, la medida **ValorDiaMenorVenta** retornaría el valor menor de la suma de los ingresos del mismo día en los años seleccionados. El motivo, se está agrupando por un campo que no tiene valores únicos, con lo cual, si se desea el valor mayor del día con menores ventas para un año especificado basta con agrupar por un campo con valores únicos, así:

1. **ValorDiaMayorVenta** =
2. **MINX** (
3. **ADDCOLUMNS** (
4. **SUMMARIZE** (Pedidos, Calendario[Fecha]),
5. **"@Ingresos Totales", [Ing]**
6. **)**,
7. **[@Ingresos Totales]**
8. **)** -- Se pueden seleccionar varios años.



Adelante:

Existe una manera de escribir la expresión anterior para una medida de una forma mucha más compacta y elegante, sin embargo, para ello primero debemos dejar estudiar el principio a continuación:



II. El segundo parámetro indica una expresión a ser ejecutada fila a fila.

Esta expresión puede ser algo tan sencillo como llamar a un campo en el modelo de datos:

• Ejemplo:

```
1. ValorVentaMáxima =
2. MINX ( Pedidos, Pedidos[Ingresos] )
```



La medida anterior es la versión larga de: **MIN(Pedidos[Ingresos])**, es decir, esta última es sólo una syntax sugar de la expresión más larga y completa: **MINX (Pedidos , Pedidos[Ingresos])**, por lo anterior, vemos que cobra mayor utilidad implementar la función **MINX** cuando el segundo parámetro involucra varias columnas a operar, por ejemplo:

• Ejemplo:

```
1. MenorCostoTotal =
2. MINX (
3.     Pedidos,
4.     Pedidos[Costo Producto] +
5.     Pedidos[Costo Empaque] +
6.     Pedidos[Costo Envió]
7. )
```

Incluso podemos involucrar funciones, ejemplo:

```
1. MenorCostoTotalComoEntero =
2. MINX (
3.     Pedidos,
4.     ROUNDUP ( Pedidos[Costo Producto], 0 ) +
5.     ROUNDUP ( Pedidos[Costo Empaque], 0 ) +
6.     ROUNDUP ( Pedidos[Costo Envió], 0 )
7. )
```





Esta capacidad de llamar funciones nos brinda una habilidad potente cuando se involucra la función [CALCULATE](#), puesto que podemos emplear la operación de transición de contextos para añadir los valores de la iteración actual como filtros o restricciones a la tabla correspondiente en el modelo de datos.

- Ejemplo:

Si recordamos nuestro escenario en el cual buscábamos exponer en un reporte para un año determinado a nivel de meses el valor del día con menor ingresos, entonces, una solución más elegante que además permite seleccionar varios años es como sigue:

```
1. ValorDiaMenorVenta =
2. MINX ( Calendario, CALCULATE ( SUM ( Pedidos[Ingresos] ) ) )
```

Y gracias al [CALCULATE](#) implícito que se añade cuando hacemos referencia a una medida, podemos escribir la expresión aún más compacta, así:

```
1. ValorDiaMenorVenta =
2. MINX (
3.     Calendario,
4.     [Ing]
5. )
6. -- La medida [Ing] es:
7. -- SUM ( Pedidos[Ingresos] )
```

III. La función **MINX** ignora la existencia de valores *null*, tomados como vacío (BLANK) por el motor DAX, realizando la determinación de valor superior exclusivamente sobre el conjunto de valores estricta y visualmente numéricos.

Por esta consideración, si la distribución de la columna o del resultado de la expresión ejecutada fila a fila tiene valores vacíos que deben ser leídos como 0, entonces, se debe asignar el valor explícitamente.

- Ejemplo:





Acerca de las Cartas DAX



Las cartas DAX del equipo de **Excel Free Blog** es un paquete de contenido de documentación y representación para un juego de todas las funciones en lenguaje DAX, compuesta por dos partes:

I. La Carta

Cada función en todo el lenguaje DAX contará con un **personaje representativo**, por ejemplo, la función SUMX será representada por el ser mitológico: el grifo.

II. La Ficha Técnica

La ficha técnica tiene **información de la función** para su manejo, consulta y entendimiento, en ella se documenta y explica: Descripción, sintaxis, parámetros y más. (Cómo la presente)

Más Información

→ <https://bit.ly/3aZiBqu> ←

→ www.CartasDax.Com ←

Última Actualización:

6 de mayo del 2021.

IV. La función MINX no admite valores de tipo TRUE/FALSE en el resultado de la expresión ejecutada fila a fila, por lo tanto, aplicar lógica booleana de forma directa para resolver el ejemplo anterior no es valido

• Ejemplo:

```
1. MinERROR =
2. MINX (
3.     Pedidos,
4.     NOT ISBLANK ( Pedidos[Ingresos] ) * VALUE ( Pedidos[Ingresos] )
5. ) -- Error: La función no admite valores booleanos
```

Aunque con funciones como **CONVERT** se puede lograr, esto hace más enrevesado e ineficiente el proceso.

V. Si la expresión pasada en su primer parámetro retorna una tabla vacía (no hay filas) entonces la función **MINX** devuelve **BLANK**.

Dado que para una visualización el contexto varia, el valor **BLANK** puede aparecer en casillas o lugares específicos del objeto visual.

REFERENCIAS Y BIBLIOGRAFÍA

Páginas Web:

- 1. DAX GUIDE: <https://dax.guide/minx/>
- 2. MICROSOFT: <https://docs.microsoft.com/en-us/dax/minx-function-dax>
- 3. EFB: <https://www.excelfreeblog.com/principios-en-funciones-de-iteracion-escalares>

Creado por:

Miguel Caballero y Fabian Torres.

Cualquier Retroalimentación:

excelfreebymcs@gmail.com

Funciones Relacionadas: **MIN** **MINA**



MINX: Dragón Oscuro del Valor Inferior