Iniciamos en Breve

• Archivos: https://github.com/ExcelFreeBlog/SemanaDeDAX







Clasificación de Funciones

Contexto de Filtro + Contexto de Fila



Clasificación de Funciones

Aunque se pueden clasificar de muchas maneras, <u>existen por lo menos tres clasificaciones mínimas que si interiorizan nos facilitará en el aprendizaje</u> del lenguaje DAX y desarrollar un entendimiento pleno

Según Resultado

Esta forma de clasificación de las funciones del lenguaje DAX consiste en poder identificar que devuelve como resultado final.

- ♥ Escalar Ejemplo: SUM, LEFT, YEAR, ...
- ▼ Tabla Ejemplo: FILTER, SUMMARIZE, ...

Según Compartimiento RC

Esta forma de clasificación se refiere a cómo interactúa una función especifica con los contextos de evaluación.

Contexto de Fila

- ♀ Ignora Ejemplo: SUM, COUNTROWS, ...
- Respeta Ejemplo: OR, MONTH, ...

Contexto de Filtro

- Ignora Ejemplo: ALL, ALLEXCEPT, ...
- Respeta Ejemplo: SUMX, AVERAGEX, ...

© www.ExcelFreeBlog.Com | www.EscuelaDeInteligenciaDeNegocios.Com

<u>Según el Proceso</u>

Se refiere a cómo hace la función de manera interna para llegar al resultado final.

- Prunciones de Cálculos Directo
 - Ejemplo: AND, OR, ...
- Funciones de Escaneo
 - Ejemplo: ALL, SUMMARIZE
- Funciones de Iteración
 - Ejemplo: SUMX, FILTER
- Funciones de Producto Cartesiano
 - Ejemplo: SUMMARIZECOLUMNS, ...

Repaso y Síntesis

Clasificación de Funciones

Contexto de Filtro + Contexto de Fila



Funciones de Iteración: Son un conjunto de funciones que <u>se ejecutan en una base fila a fila</u>, es decir, <u>iteran en una tabla o expresión de tabla</u> y va calculando para dicha fila, <u>una expresión de tipo escalar generalmente</u>, cuyo valor es guardado para luego realizar la semántica de la función a todos los valores parciales una vez culminado el proceso.

(1): Funciones de Iteración de Agregación \rightarrow (Retorna un Escalar)

(2): Funciones de Iteración Tabualares → (Retorna una Tabla)

EL CONTEXTO DE FILA: Es la fila actual, la cual itera en una tabla y restringe las filas visible sólo al del momento.

Contexto de Fila Automático.

Contexto de Fila Programable.

Creado por las Columnas Calculadas

Creado por cualquier Función de Iteración

Funciones de Iteración de Agregación o Funciones de «Sufijo» X

AVERAGEX, CONCATENATEX, COUNTAX, COUNTX, GEOMEANX, MAXX, MEDIANX, MINX, PERCENTILX.EXC, PERCENTILX.INC, PRODUCTX, RANKX, STDEVX.P, STDEVX.S, SUMX, VARX.P, VARX.S, XIRR y XNPV.

Funciones Tabulares de Iteración

ADDCOLUMNS. FILTER. FIRSTNONBLANK, FIRSTNONBLANKVALUE, GENERATE. GENERATEALL. LASTNOBLANK, LASTNONBLANKVALUE, SAMPLE, SELECTCOLUMNS, SUBSTITUTEWITHINDEX, SUMMARIZE, TOPN y TOPNSKIP.

Aunque este par de funcione devuelve un valor escalar, no tienen el sufijo X puesto que no hay que distinguirlas de otro par con los mismo nombres.



LA PRIMER SYNTAX SUGAR:

Las funciones más simples, AVERAGE, CONCATENATE..... VAR.S son una syntax sugar de sus pares iterativos con una tabla y expresión imple, es decir: FunciónX (Tabla : Tabla[Columna]), no obstante, están quedan por fuera de la clasificación de funciones de iteración porque no hay posibilidad de acceder a la expresión, para ello su utiliza la función expandida explícitamente.

© www.ExcelFreeBlog.Com | www.EscuelaDeInteligenciaDeNegocios.Com



Acceso con EARLIER y EARLIEST





Escalar Un valor único de cualquier tipo

 $XL \ge 2010$, $PBI \ge Nov 2016$, $SSAS \ge 2012$

Retorna el valor de una columna especifica en un contexto de fila previo, es decir, anterior al contexto de fila generado más reciente en el proceso de ejecución de una expresión. Por defecto se asocia al penúltimo contexto de fila generado, que es identificado por el número I.

• SINTAXIS:

EARLIER (< Nombre de Columna > ; [< Número >])

Acceso con EARLIER y EARLIEST



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna de índice o ID consecutivo).

=

ID	Valor	Acumulado
	40	
2	30	
3	40	
4	70	
5	50	
6	10	
7	10	
8	90	

Acceso con EARLIER y EARLIEST



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna de índice o ID consecutivo).

$$=$$
 SUMX (A ;

ID	Valor	Acumulado
Ĺ	40	
2	30	
3	40	
4	70	
5	50	
6	10	
7	10	
8	90	

SUMX Ignora cualquier contexto de fila previa generado antes que el propio, es decir, ignora el contexto de fila automático pero trabajo en su contexto de fila programable.

Acceso con EARLIER y EARLIEST



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna de índice o ID consecutivo).

$$=$$
 SUMX (A ;

ID	Valor	Acumulado
I	40	
2	30	
3	40	
4	70	
5	50	
6	10	
7	10	
8	90	

SUMX Ignora cualquier contexto de fila previa generado antes que el propio, es decir, ignora el contexto de fila automático pero trabajo en su contexto de fila programable.

Acceso con EARLIER y EARLIEST



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna de índice o ID consecutivo).

ID	Valor	Acumulado
I	40	
2	30	
3	40	
4	70	
5	50	
6	10	
7	10	
8	90	

SUMX Ignora cualquier contexto de fila previa generado antes que el propio, es decir, ignora el contexto de fila automático pero trabajo en su contexto de fila programable.

Acceso con EARLIER y EARLIEST



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna de índice o ID consecutivo).

ID	Valor	Acumulado
I	40	
2	30	
3	40	
4	70	
5	50	
6	10	
7	10	
8	90	

SUMX Ignora cualquier contexto de fila previa generado antes que el propio, es decir, ignora el contexto de fila automático pero trabajo en su contexto de fila programable.

Acceso con EARLIER y EARLIEST



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna de índice o ID consecutivo).

$$=$$
 SUMX (A ;

ID	Valor	Acumulado
I	40	
2	30	
3	40	
4	70	
5	50	
6	10	
7	10	
8	90	

SUMX Ignora cualquier contexto de fila previa generado antes que el propio, es decir, ignora el contexto de fila automático pero trabajo en su contexto de fila programable.

Acceso con EARLIER y EARLIEST



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna de índice o ID consecutivo).

ID	Valor	Acumulado
I	40	
2	30	
3	40	
4	70	
5	50	
6	10	
7	10	
8	90	

SUMX Ignora cualquier contexto de fila previa generado antes que el propio, es decir, ignora el contexto de fila automático pero trabajo en su contexto de fila programable.

Acceso con EARLIER y EARLIEST



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna de índice o ID consecutivo).

$$=$$
 SUMX (A ;

ID	Valor	Acumulado
I	40	
2	30	
3	40	
4	70	
5	50	
6	10	
7	10	
8	90	

SUMX Ignora cualquier contexto de fila previa generado antes que el propio, es decir, ignora el contexto de fila automático pero trabajo en su contexto de fila programable.

Acceso con EARLIER y EARLIEST



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna de índice o ID consecutivo).

$$=$$
 SUMX (A ;

ID	Valor	Acumulado
I	40	
2	30	
3	40	
4	70	
5	50	
6	10	
7	10	
8	90	

SUMX Ignora cualquier contexto de fila previa generado antes que el propio, es decir, ignora el contexto de fila automático pero trabajo en su contexto de fila programable.

Acceso con EARLIER y EARLIEST



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna de índice o ID consecutivo).

$$=$$
 SUMX (A ;

ID	Valor	Acumulado
I	40	
2	30	
3	40	
4	70	
5	50	
6	10	
7	10	
8	90	

SUMX Ignora cualquier contexto de fila previa generado antes que el propio, es decir, ignora el contexto de fila automático pero trabajo en su contexto de fila programable.

Acceso con EARLIER y EARLIEST



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna de índice o ID consecutivo).

=
$$SUMX$$
 (A; IF (A[ID] <= $EARLIER$ (A[ID]); A[Valor]; 0)

ID	Valor	Acumulado
I	40	
2	30	
3	40	
4	70	
5	50	
6	10	
7	10	
8	90	

Acceso con EARLIER y EARLIEST - Cardinalidad





La iteración más externa da la columna calculada se corresponde con el contexto de fila automático, por lo tanto, recorre cada registro de la tabla, en el cual realiza un total de 8 iteraciones.

La iteración interna que se corresponde con el contexto de fila programable de SUMX, donde itera en toda la tabla nuevamente por cada iteración externa, es decir, 8 iteraciones internas por cada iteración externa.

Para cada iteración externa se realiza el recorrido de las iteraciones internas.

8 Filas ITERACIÓN EXTERNA
X 8 Filas ITERACIÓN ITERNA

64 Iteraciones

$$=(\dot{\mathbf{u}})$$



Acceso con EARLIER y EARLIEST



FACILIDADES PARA ACCESO A MÚLTIPLES CONTEXTO DE FILA

Para tener facilidad al acceso a contextos de fila previos (porque pueden existir múltiples contexto de fila programables a la vez con iteradores anidados), es crucial que exista una columna de índice, y que la misma que cumpla con las siguiente propiedades:

- 1. Debe ser una columna de valores que no se repitan, por lo tanto identificada cada fila de forma única.
- 2. Dicha columna debe ser de valores consecutivos, ejemplo: 1,2,3,4,56, ...1000
- 3. Preferiblemente creada con agregar columna y columna de índice con Power Query



Es posible omitir las propiedad 2 y 3, sin embargo, esto hace que las expresiones sean vuelvan innecesariamente mucho más complejas y muchos más ineficientes en términos de rendimiento, porque si este tipo de cálculos tiene un gran impacto en el rendimiento del modelo por la cantidad iteraciones que devuelve realizar la expresión.



Algunos Primeros Retos con EARLIER y EARLIEST





VALOR ANTERIOR

. Crear una expresión DAX para un columna calculada devuelva el valor en la casilla inmediatamente anterior.



```
Valor Anterior =
SUMX ( A; IF ( A[ID] = EARLIER ( A[ID] ) - 1; A[Valor]; BLANK () ) )
```

VALOR SIGUIENTE

2. Crear una expresión DAX para un columna calculada devuelva el valor en la casilla inmediatamente siguiente.



```
Valor Anterior =
SUMX ( A; IF ( A[ID] = EARLIER ( A[ID] ) + I; A[Valor]; BLANK () ) )
```

Algunos Primeros Retos con EARLIER y EARLIEST





VALOR ANTERIOR

. Crear una expresión DAX para un columna calculada devuelva el valor en la casilla inmediatamente anterior.



```
Valor Anterior =
SUMX ( A; IF ( A[ID] = EARLIER ( A[ID] ) - 1; A[Valor]; BLANK () ) )
```

VALOR SIGUIENTE

2. Crear una expresión DAX para un columna calculada devuelva el valor en la casilla inmediatamente siguiente.



```
Valor Anterior =
SUMX ( A; IF ( A[ID] = EARLIER ( A[ID] ) + I; A[Valor]; BLANK () ) )
```

Algunos Primeros Retos con EARLIER y EARLIEST





SUMA DE LOS DOS ANTERIORES

3. Crear una expresión DAX para un columna calculada devuelva el valor en que corresponda a la suma de los valores inmediatamente anteriores

Acceso con EARLIER y EARLIEST: Suma de Los Dos Anteriores





Escalar Un v

Un valor único de cualquier tipo

 $XL \geq 2010$, $PBI \geq Nov 2016$, $SSAS \geq 2012$

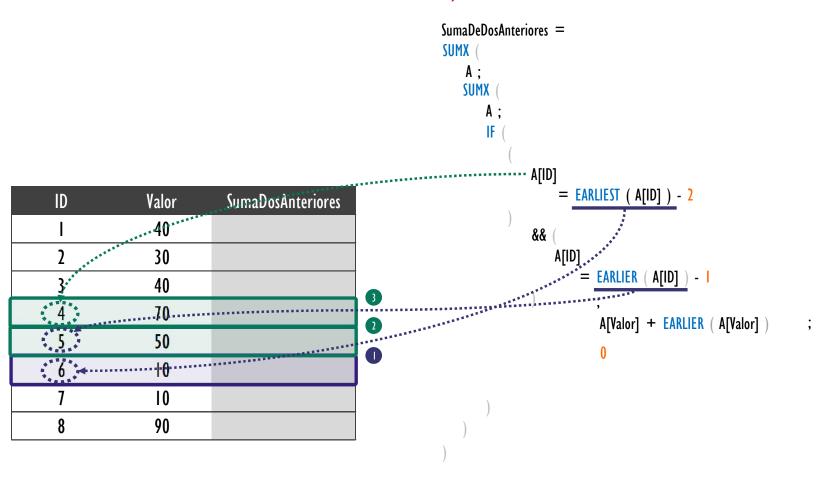
Retorna el valor de una columna especifica en el primer contexto de fila generado, que si es en una columna calculada es el contexto de fila automático, pero si es una expresión con sólo contextos de fila programables, entonces, es el generado en un primer punto.

• SINTAXIS:

EARLIEST (< Nombre de Columna >)

Acceso con EARLIER y EARLIEST: Suma de Los Dos Anteriores





Acceso con EARLIER y EARLIEST: Suma de Los Dos Anteriores





Y si tenemos más de tres contexto de fila y queremos hacer referencia a uno especifico, cómo se puede conseguir.

	ID	Valor	SumaDosAnteriores	
		40		
	2	30		
	3	40		E
	4	70		6
)	5	50		
)	6	10		
,	7	10		
	8	90		



```
SumaDeDosAnteriores =
SUMX
    A;
    SUMX
        A;
        IF
               A[ID]
                    = EARLIEST (A[ID]) - 2
                &&
                    A[ID]
                        = EARLIER (A[ID]) - I
           A[Valor] + EARLIER ( A[Valor] );
```

Acceso con EARLIER y EARLIEST: Suma de Los Dos Anteriores





Y si tenemos más de tres contexto de fila y queremos hacer referencia a uno especifico, cómo se puede conseguir.

	ID	Valor	SumaDosAnteriores	
		40		
	2	30		
	3	40		E
	4	70		6
)	5	50		
)	6	10		
,	7	10		
	8	90		



```
SumaDeDosAnteriores =
SUMX
    A;
    SUMX
        A;
        IF
               A[ID]
                    = EARLIEST (A[ID]) - 2
                &&
                    A[ID]
                        = EARLIER (A[ID]) - I
           A[Valor] + EARLIER ( A[Valor] );
```

Acceso con EARLIER y EARLIEST: Suma de Los Dos Anteriores





Y si tenemos más de tres contexto de fila y queremos hacer referencia a uno especifico, cómo se puede conseguir.

	ID	Valor	SumaDosAnteriores	
	I	40		
	2	30		
	3	40		
	4	70		
)	5	50		
)	6	10		
	7	10		
	8	90		



```
SumaDeDosAnteriores-2 =
SUMX
    A;
    SUMX
        A;
        IF
                A[ID]
                    = EARLIER ( A[ID]; 2 ) - 2
                &&
                    A[ID]
                        = EARLIER (A[ID]) - I
            A[Valor] + EARLIER ( A[Valor] );
```

Acceso con EARLIER y EARLIEST: Suma de Los Dos Anteriores



ERROR

```
EjemploEncColumnaCalculada =
SUMX
      A;
      SUMX
            A;
            SUMX
                   <...>
```

	ID	Valor	SumaDosAnteriores
	I	40	
	2	30	
	3	40	
)	4	70	
)	5	50	
)	6	10	
	7	10	
	8	90	

```
A[ID]

EARLIER ( A[ID] ; I ) = EARLIER ( A[ID] )

EARLIER ( A[ID] ; 2 )

EARLIER ( A[ID] ; 3 )

EARLIEST ( A[ID] )

EARLIER ( A[ID] ; 4 )
```

Algunos Retos Adicionales EARLIER y EARLIEST



SUMA DE LOS TRES VALORES ANTERIORES

 Crear una expresión DAX para un columna calculada devuelva la suma de los tres valores anteriores.





Acceso con Variables y la Despedida de EARLIER y EARLIEST Forever . .



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna con ID), pero ahora utilizando variables.

-			
	, ID	Valor	Acumulado
		40	
	Ì	30	
	3	40	
	4	70	
	5	50	
	6	10	
	7	10	
	8	90	

```
AcumuladoVariables =

VAR ValorActual = A[ID]

RETURN

SUMX (A; IF (A[ID]) <= ValorActual; A[Valor]; 0)
```

Acceso con Variables y la Despedida de EARLIER y EARLIEST Forever



Acumulado en Columna Calculada: Imaginemos que queremos realizar el acumulado en una columna calculada (asumiendo que tenemos una columna con ID), pero ahora utilizando variables.

-			
	, ID	Valor	Acumulado
		40	
	Ž	30	
	3	40	
•	4	70	
	5	50	
	6	10	
	7	10	
	8	90	

AcumuladoVariables =

3 VAR ValorActual = A[ID]

RETURN

SUMX (A; IF (A[ID]) <= ValorActual; A[Valor]; 0)



Será posible mediante variables almacenar los valores de varios contextos de fila y así tener acceso, por ejemplo: será posible emular rescribir la expresión de la suma de los dos valores netamente con variables sin necesidad de EARLIER y EARLIEST.

Acceso con Variables y la Despedida de EARLIER y EARLIEST Forever . . .



```
SumaDosAnterioreswVAR =
VAR IDActual = A[ID]
RETURN
   SUMX (
        A;
       VAR IDPrevio = A[ID]
       VAR ValorPrevio = A[Valor]
        RETURN
            SUMX
                A;
                IF
                    A[ID] = IDActual - 2
                        && A[ID] = IDPrevio - I;
                    A[Valor] + ValorPrevio;
                    BLANK ()
```



Acceso con Variables y la Despedida de EARLIER y EARLIEST Forever . . .





EARLIER y EARLIEST I

EXTINTOS

COMPLEJIDAD

La implementación de la función EARLIER y su primo cercano EARLIEST puede ser bastante difícil de entender plenamente, de hecho, es una de las funciones que más les cuesta aprender a las personas que se incursionan en el mundo de DAX.

RENDIMIENTO

La implementación de variables para el acceso de valores en las columnas en contextos de filas generados previamente, optimiza un poco la expresión, dado que las variables almacena dicho valor como fijo, y más adelante si es requeridos más de una vez ya esta calculado y no tiene que ejecutarse *n* veces.

Considere EARLIER (Mamut del los Contextos de Filas Previos) y EARLIEST (Colmillos de Sable del Primer Contexto de fila) como:

Funciones extintas

© www.ExcelFreeBlog.Com | www.EscuelaDeInteligenciaDeNegocios.Com — Creado por Miguel Caballero Sierra y Fabian Torres Hernández

Acceso con Variables y la Despedida de EARLIER y EARLIEST Forever . .





NADA DE MIEDOS

Te dejamos el siguiente reto:

Pasar la expresión de la suma de los tres valores anteriores escrita con las funciones EARLIEST y EARLIER con implementación del segundo parámetro

Y comprobar y demostrar por ti mismo, la obsolescencia de EARLIEST y EARLIER.

