

VARIABLES

Una variable puede almacenar un VALOR ESCALAR o una TABLA.

```
VAR SalesAmt =  
    SUMX (  
        Sales,  
        Sales[Quantity] * Sales[Net Price]  
    )  
RETURN  
    IF (  
        SalesAmt > 100000,  
        SalesAmt,  
        SalesAmt * 1.2  
    )
```

Agregar más definiciones de VAR dentro del mismo bloque permite la definición de múltiples variables, mientras que el bloque RETURN debe ser único. **Es importante señalar que el bloque VAR / RETURN es, de hecho, una expresión.** Como tal, una definición de variable tiene sentido siempre que se pueda usar una expresión. Esto hace posible definir variables durante una iteración, o como parte de expresiones más complejas, como en el siguiente ejemplo:

```
VAR SalesAmt =  
    SUMX (  
        Sales,  
        VAR Quantity = Sales[Quantity]  
        VAR Price = Sales[Price]  
        RETURN  
            Quantity * Price  
    )  
RETURN  
    ...
```

Un uso muy frecuente de las variables es dividir el cálculo de una fórmula compleja en pasos lógicos, asignando el resultado de cada paso a una variable. Por ejemplo, en el siguiente código, las variables se utilizan para almacenar resultados parciales del cálculo:

```
Margin% :=
VAR SalesAmount =
    SUMX ( Sales, Sales[Quantity] * Sales[Net Price] )

VAR TotalCost =
    SUMX ( Sales, Sales[Quantity] * Sales[Unit Cost] )

VAR Margin =
    SalesAmount - TotalCost
VAR MarginPerc =
    DIVIDE ( Margin, TotalCost )
RETURN
    MarginPerc
-----
Sales,
VAR Quantity = Sales[Quantity]
VAR CurrentPrice = RELATED ( 'Product'[Unit Price] )
VAR AmountAtCurrentPrice = Quantity * CurrentPrice
RETURN
    AmountAtCurrentPrice
)
-- Any reference to Quantity, CurrentPrice, or AmountAtCurrentPrice
-- would be invalid outside of SUMX
```

Puede escribir cualquier expresión después de RETURN. Sin embargo, el uso de una sola variable para la parte RETURN se considera la mejor práctica.

Las variables se evalúan una vez en el alcance de la definición (VAR) y no cuando se utiliza su valor. Por ejemplo, la siguiente medida siempre devuelve 100% porque la variable **SalesAmount** no se ve afectada por CALCULATE. Su valor solo se calcula una vez. Cualquier referencia al nombre de la variable devuelve el mismo valor independientemente del contexto del filtro donde se usa el valor de la variable.

```
% of Product :=
VAR SalesAmount = SUMX ( Sales, Sales[Quantity] * Sales[Net Price] )
RETURN
    DIVIDE (
        SalesAmount,
        CALCULATE (
            SalesAmount,
            ALL ( 'Product' )
        )
    )
)
```

En este último ejemplo, usamos una variable donde deberíamos haber usado una medida. En el siguiente código, el porcentaje correcto se obtiene definiendo dos medidas:

```
Sales Amount :=  
SUMX ( Sales, Sales[Quantity] * Sales[Net Price] )  
  
% of Product :=  
DIVIDE (  
    [Sales Amount],  
    CALCULATE (  
        [Sales Amount],  
        ALL ( 'Product' )  
    )  
)
```

Una definición de variable puede acceder a las variables definidas en declaraciones VAR anteriores, pero no a las variables definidas en las declaraciones siguientes. Por lo tanto, este código funciona bien:

```
Margin :=  
VAR SalesAmount =  
    SUMX ( Sales, Sales[Quantity] * Sales[Net Price] )  
VAR TotalCost =  
    SUMX ( Sales, Sales[Quantity] * Sales[Unit Cost] )  
VAR Margin = SalesAmount - TotalCost  
RETURN  
    Margin
```

```
Margin :=  
SUMX (  
    Sales,  
    VAR LineAmount = Sales[Quantity] * Sales[Net Price]  
    VAR LineCost = Sales[Quantity] * Sales[Unit Cost]  
    RETURN  
        LineAmount - LineCost  
)
```

Cuando una variable contiene una tabla, es probable que se quiera iterar sobre ella. Es importante tener en cuenta que, durante dicha iteración, se debe acceder a las columnas de una variable de tabla utilizando sus nombres originales. En otras palabras, un nombre de variable no es un alias de la tabla subyacente en las referencias de columna:

```
Filtered Amount :=  
VAR  
    MultiSales = FILTER ( Sales, Sales[Quantity] > 1 )  
RETURN  
    SUMX (  
        MultiSales,  
        -- MultiSales is not a table name for column references  
        -- Trying to access MultiSales[Quantity] would generate an error  
        Sales[Quantity] * Sales[Net Price]  
    )
```

Aunque SUMX itera sobre *MultiSales*, debe usar el nombre de la tabla *Sales* para acceder a las columnas *Quantity* y *Net Price*. Una referencia de columna como *MultiSales[Quantity]* no es válida.

Una limitación actual de DAX es que una variable no puede tener el mismo nombre que cualquier tabla en el modelo de datos.

1. Ejemplo 1 para columna calculada

```

1 Demora en envío =
2 VAR Diferencia = Sales[Delivery Date] - Sales[Order Date]
3 RETURN
4 IF(
5     Diferencia <= 0, "En proceso",
6     IF(
7         Diferencia <= 7, "Una semana",
8         IF(
9             Diferencia <= 14, "Dos semanas", "Por encima de las dos semanas"
10        )
11    )
12 )

```

customerKey	OrderDateKey	Order Date	Delivery Date	Order Number	Order Line Number	Quantity	Unit Price	Unit Discount	Unit Cost	Net Price	Demora en envío
18074	20070424	24-04-2007	01-05-2007	20070424729073	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Una semana
18075	20070424	24-04-2007	03-05-2007	20070424729074	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18076	20070424	24-04-2007	05-05-2007	20070424729075	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18078	20070424	24-04-2007	02-05-2007	20070424729077	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18079	20070424	24-04-2007	04-05-2007	20070424729078	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18080	20070424	24-04-2007	06-05-2007	20070424729079	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18090	20070424	24-04-2007	03-05-2007	20070424729089	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18105	20070424	24-04-2007	05-05-2007	20070424729104	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18107	20070424	24-04-2007	02-05-2007	20070424729106	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18108	20070424	24-04-2007	04-05-2007	20070424729107	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18109	20070424	24-04-2007	06-05-2007	20070424729108	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18110	20070424	24-04-2007	01-05-2007	20070424729109	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Una semana
18112	20070424	24-04-2007	05-05-2007	20070424729111	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18113	20070424	24-04-2007	30-04-2007	20070424729112	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Una semana
18125	20070424	24-04-2007	06-05-2007	20070424729124	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18126	20070424	24-04-2007	01-05-2007	20070424729125	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Una semana
18127	20070424	24-04-2007	03-05-2007	20070424729126	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18131	20070424	24-04-2007	01-05-2007	20070424729130	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Una semana
18132	20070424	24-04-2007	03-05-2007	20070424729131	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18134	20070424	24-04-2007	30-04-2007	20070424729133	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Una semana
18135	20070424	24-04-2007	02-05-2007	20070424729134	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18136	20070424	24-04-2007	04-05-2007	20070424729135	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18137	20070424	24-04-2007	06-05-2007	20070424729136	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18139	20070424	24-04-2007	03-05-2007	20070424729138	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas
18140	20070424	24-04-2007	05-05-2007	20070424729139	1	1	\$8,99	\$0,00	\$4,13	\$8,99	Dos semanas

2. Ejemplo 2 para columna calculada

```

1 Mayúscula Month =
2 VAR NombreMonth =
3 |     FORMAT('Date'[Month], "MMMM")
4 RETURN
5 |     UPPER(LEFT(NombreMonth,1)) & UPPER(RIGHT(NombreMonth,LEN(NombreMonth)-1))

```

Month Number	Month	Week Day Number	Week Day	Year Month Number	Year Month	Year Quarter Number	Year Quarter	DateKey	Mayúscula Month
7	July	3	Tuesday	200807	July 2008	200803	Q3-2008	20080701	JULY
7	July	4	Wednesday	200807	July 2008	200803	Q3-2008	20080702	JULY
7	July	5	Thursday	200807	July 2008	200803	Q3-2008	20080703	JULY
7	July	6	Friday	200807	July 2008	200803	Q3-2008	20080704	JULY
7	July	7	Saturday	200807	July 2008	200803	Q3-2008	20080705	JULY
7	July	1	Sunday	200807	July 2008	200803	Q3-2008	20080706	JULY
7	July	2	Monday	200807	July 2008	200803	Q3-2008	20080707	JULY
7	July	3	Tuesday	200807	July 2008	200803	Q3-2008	20080708	JULY
7	July	4	Wednesday	200807	July 2008	200803	Q3-2008	20080709	JULY
7	July	5	Thursday	200807	July 2008	200803	Q3-2008	20080710	JULY
7	July	6	Friday	200807	July 2008	200803	Q3-2008	20080711	JULY
7	July	7	Saturday	200807	July 2008	200803	Q3-2008	20080712	JULY
7	July	1	Sunday	200807	July 2008	200803	Q3-2008	20080713	JULY
7	July	2	Monday	200807	July 2008	200803	Q3-2008	20080714	JULY
7	July	3	Tuesday	200807	July 2008	200803	Q3-2008	20080715	JULY
7	July	4	Wednesday	200807	July 2008	200803	Q3-2008	20080716	JULY
7	July	5	Thursday	200807	July 2008	200803	Q3-2008	20080717	JULY
7	July	6	Friday	200807	July 2008	200803	Q3-2008	20080718	JULY
7	July	7	Saturday	200807	July 2008	200803	Q3-2008	20080719	JULY
7	July	1	Sunday	200807	July 2008	200803	Q3-2008	20080720	JULY
7	July	2	Monday	200807	July 2008	200803	Q3-2008	20080721	JULY
7	July	3	Tuesday	200807	July 2008	200803	Q3-2008	20080722	JULY
7	July	4	Wednesday	200807	July 2008	200803	Q3-2008	20080723	JULY
7	July	5	Thursday	200807	July 2008	200803	Q3-2008	20080724	JULY
7	July	6	Friday	200807	July 2008	200803	Q3-2008	20080725	JULY