Funciones de inteligencia de tiempo

Uno de los cálculos más comunes que se realizan en el análisis de datos es comparar valores en períodos de tiempo diferentes. Por esta razón, DAX incluye 35 funciones con el fin de trabajar con datos basados en el tiempo.

Siempre que se definen medidas y se trabaja con períodos de tiempo, hay muchas oportunidades para hacer suposiciones erróneas acerca de cómo se vería la tabla dinámica en la que se utilizará la medida.

Como práctica, se recomienda crear una tabla de fechas en el modelo de datos de PowerPivot que contenga una fila para cada fecha que pudiese existir en sus datos. La idea es que esta tabla tenga una fila para cada fecha, y que exista una relación de muchos a uno desde cualquier columna de fecha en la base de datos hasta esta tabla de fechas. Crear una tabla de este tipo en Excel es bastante trivial, y puede parecerse a la siguiente tabla:

Archivo Inicio Insertar Dibujar Diseño de página Fórmulas Datos Revisar Vista Programador ¿Qué desea hacer?

J8

	A	B	C	D	E	F	G	H
1	Fecha	Año	Mes	Día	MesLargo	MesCorto	DíaLargo	DíaCorto
2	01/01/2016	2016	1	1	enero	ene	viernes	vi
3	02/01/2016	2016	1	2	enero	ene	sábado	sá
4	03/01/2016	2016	1	3	enero	ene	domingo	do
5	04/01/2016	2016	1	4	enero	ene	lunes	lu
6	05/01/2016	2016	1	5	enero	ene	martes	ma
7	06/01/2016	2016	1	6	enero	ene	miércoles	mi
8	07/01/2016	2016	1	7	enero	ene	jueves	ju
9	08/01/2016	2016	1	8	enero	ene	viernes	vi
10	09/01/2016	2016	1	9	enero	ene	sábado	sá

DAX usa esta tabla para construir un conjunto de fechas para cada cálculo. La única columna que importa es la columna de fecha misma. No hay requisitos en absoluto sobre ninguna de las otras columnas, y DAX no las necesita, aunque es posible que las encuentre útiles cuando construya Tablas Dinámicas. Debe proporcionar una referencia a la columna **Fecha** como argumento para cada una de las 35 funciones de inteligencia de tiempo en DAX.

Se deberá definir qué columna de tipo de fecha en la tabla Fechas, expresa la fecha. Esto permite que las tablas modelo usen una clave de tipo que no sea Fecha, lo cual es común en las tablas de hechos que usan claves enteras (típicamente fechas en formato ISO, por ejemplo, 20160821). Asimismo, hay que asegurar que la tabla **Fechas** contenga una fila por fecha y que no haya espacios vacíos en los datos de la tabla, así como de que la tabla esté definida como una tabla de fechas.

Esta configuración simple asegurará que las funciones de Inteligencia de Tiempo funcionarán correctamente y de que Excel sepa que las columnas en la tabla de datos pueden usar filtros relacionados con el tiempo.

En DAX, siempre se calcula un conjunto de fechas como una tabla y luego se utiliza este resultado como si fuera un argumento de **SetFilter** para la función **CALCULATE**.

Un usuario puede haber seleccionado varias fechas en una tabla dinámica, por lo que el contexto para un cálculo podría ser cualquiera de los siguientes:

- Una sola fecha
- Un conjunto de fechas contiguas
- Un conjunto de fechas no contiguas
- Fechas que coinciden con un mes calendario o trimestre o año (muy común)

Entonces considere que podríamos necesitar cambiar esas fechas para encontrar lo siguiente:

- Las fechas que componen el día, mes, trimestre o año anterior
- Las mismas fechas a un mes, trimestre o año anterior
- Fechas calculadas desplazando un intervalo arbitrario hacia adelante o hacia atrás en el tiempo.

DAX, puede lograr todo esto al trabajar con conjuntos de fechas para todo.

Las funciones integradas permiten manejan tanto el calendario natural como ejercicios fiscales donde la fecha de finalización no es el 31 de diciembre.

Las funciones de Inteligencia de tiempo de DAX se pueden clasificar en:

Funciones que devuelven una fecha única

Las funciones en esta categoría devuelven una sola fecha. El resultado de estas funciones se puede usar como argumentos para otras funciones.

Las dos primeras funciones simplemente devuelven la primera (o última) fecha en la columna de fecha en el contexto actual. Esto puede ser útil cuando quiere encontrar la primera **FIRSTDATE** (o última **LASTDATE**) fecha en la que vendió cada producto, o la primera (o última) fecha en la que tuvo una transacción de un tipo en particular. Estas funciones toman solo un argumento, el nombre de la columna de fecha en su tabla de fecha (o hora).

FIRSTDATE(ColumnaDeFecha)

LASTDATE(ColumnaDeFecha)

Las siguientes dos funciones no son estrictamente funciones de inteligencia temporal, ya que pueden usarse para otros fines, pero con mayor frecuencia se usarán para calcular el tiempo.

Se usan para encontrar la primera **FIRSTNONBLANK** (o última **LASTNONBLANK**) fecha (o cualquier otro valor de columna) donde una expresión tiene un valor no en blanco. Esto se usa con más frecuencia en situaciones como inventario, donde desea obtener el último importe de inventario y no sabe cuándo se tomó el último inventario.

FIRSTNONBLANK(ColumnaDeFecha, Expresión)

LASTNONBLANK(ColumnaDeFecha, Expresión)

Las siguientes seis funciones devuelven la primera o la última fecha de un mes, trimestre o año dentro del contexto actual del cálculo.

STARTOFMONTH (ColumnaDeFecha)	Inicio de mes
STARTOFQUARTER (ColumnaDeFecha)	Inicio de trimestre
STARTOFYEAR (ColumnaDeFecha [, YE_Date])	Inicio de año
ENDOFMONTH (ColumnaDeFecha)	Fin de mes
ENDOFQUARTER (ColumnaDeFecha)	Fin de trimestre
ENDOFYEAR (ColumnaDeFecha [, YE_Date])	Fin de año

Funciones que devuelven una tabla de fechas

Hay dieciséis funciones de inteligencia de tiempo que devuelven una tabla de fechas. Estas funciones se usan generalmente como argumento de **SetFilter** en la función **CALCULATE**. Al igual que el resto de las funciones de Inteligencia de Tiempo en DAX, cada función toma una columna de fecha como uno de sus argumentos.

Las primeras ocho funciones calculan el día, mes, trimestre o año anterior (o siguiente) en el contexto actual y devuelven esas fechas en forma de una tabla de columna única. Tenga en cuenta que las funciones "*previous*" funcionan hacia atrás desde la primera fecha en el contexto actual, y las funciones "*next*" avanzan desde la última fecha en el contexto actual.

PREVIOUSDAY (ColumnaDeFecha)	Día anterior
PREVIOUSMTH (ColumnaDeFecha)	Mes anterior
PREVIOUSQUARTER (ColumnaDeFecha)	Trimestre anterior
PREVIOUSYEAR (ColumnaDeFecha [, YE_Date])	Año anterior
NEXTDAY (ColumnaDeFecha)	Día siguiente
NEXTMONTH (ColumnaDeFecha)	Mes siguiente
NEXTQUARTER (ColumnaDeFecha)	Trimestre siguiente
NEXTYEAR (ColumnaDeFecha [, YE_Date])	Año siguiente

Las siguientes cuatro funciones en esta categoría realizan sus cálculos utilizando la última fecha en el contexto actual. Devuelven una tabla con una única columna que contiene las fechas desde el principio de mes, trimestre o año del contexto actual hasta la última fecha del contexto actual.

La función SAMEPERIODLASTYEAR requiere que el contexto actual contenga un conjunto contiguo de fechas. Si el contexto actual no es un conjunto contiguo de fechas, SAMEPERIODLASTYEAR devolverá un error.

DATESMTD(ColumnaDeFecha)	Desde principio de mes
DATESQTD(ColumnaDeFecha)	Desde principio de trimestre
DATESYTD(ColumnaDeFecha [, YE_Date])	Desde principio de año
SAMEPERIODLASTYEAR(ColumnaDeFecha)	El mismo período en el año anterior

Las últimas cuatro funciones en esta categoría son un poco más complejas y también un poco más potentes. Estas funciones se utilizan para pasar del conjunto de fechas que están en el contexto actual a un nuevo conjunto de fechas.

DATEADD(ColumnaDeFecha, númeroDeIntervalos, Intervalo)

DATESBETWEEN(ColumnaDeFecha, Fecha_Inicio, Fecha_Fin)

DATESINPERIOD(ColumnaDeFecha, Fecha_Inicio, númeroDeIntervalos, Intervalo)

PARALELPERIOD(ColumnaDeFecha, númeroDeIntervalos, Intervalo)

La función **DATEADD** devuelve una tabla con una columna de fechas que coinciden con las implicadas en el contexto actual desplazadas hacia adelante o hacia atrás un **númeroDeIntervalos** determinado. Los valores posibles para el argumento **Intervalo** son: DAY (días), MONTH (meses), QUARTER (trimestres) o YEAR (años), y deberán escribirse en mayúsculas y sin comillas. Un intervalo positivo avanzará en el tiempo, mientras que un intervalo negativo retrocederá en el tiempo.

DATESBETWEEN calcula el conjunto de fechas entre la **Fecha_Inicio** y la **Fecha_Fin** especificadas.

Las dos funciones restantes cambian algunos intervalos de tiempo del contexto actual.

Estas funciones hacen que sea muy fácil cambiar el intervalo de tiempo para un cálculo por cualquiera de los siguientes:

- Retrocede dos años
- Retroceder un mes
- Avanzar tres cuartas partes
- Retroceder 14 días
- Avanzar 28 días

En cada caso, solo necesitamos especificar qué intervalo y cuántos de esos intervalos queremos cambiar. Un intervalo positivo avanzará en el tiempo, mientras que un intervalo negativo retrocederá en el tiempo. El argumento intervalo es una palabra clave: DAY, MONTH, QUARTER o YEAR, como en la función DATEADD. Las palabras clave no son cadenas de texto, por lo que no deben estar entre comillas. Ejemplo:

Crecimiento año tras año

El problema que queremos resolver es calcular el crecimiento interanual de las Ventas del canal **Retail**. La forma de calcular las ventas para este canal es:

```
[VentasRetail] =
CALCULATE (SUMA (VentasReales[ImporteVentas]), Canales [Canal] = "Retail")
```

En otras palabras, **VentasRetail** se define como la suma de la columna **ImporteVentas** en la tabla VentasReales, con el contexto modificado para incluir solo las ventas donde el **Canal** en la tabla **Canales** es "**Retail**".

Ahora queremos calcular **VentasRetail** para el año anterior.

```
[VentasRetailPrv] = CALCULATE ([VentasRetail], DATEADD (Fechas[Fecha], -1, AÑO))
```

Dado que estamos calculando una sola medida en un contexto modificado, esto se puede escribir usando un atajo de sintaxis.

```
[VentasRetailPrv] = [VentasRetail] (DATEADD (Fechas[Fecha], -1, AÑO))
```

Ahora podemos calcular el crecimiento año tras año simplemente restando las ventas del año pasado de las ventas de este año y mostrando esa diferencia como un porcentaje de las ventas del año pasado.

```
[Crecimiento] = ([VentasRetail] - [VentasRetailPrv]) / [VentasRetailPrv]
```

Finalmente, tenemos que ajustar esto en una declaración IF, por lo que no obtenemos división por cero en el primer año.

```
[Crecimiento] =
 IF ([VentasRetailPrv], ([VentasRetail] - [VentasRetailPrv]) / [VentasRetailPrv], BLANK())
```

Al colocar [VentasRetail] y [Crecimiento] en una tabla dinámica con países en filas y años en columnas podemos obtener lo siguiente:

	A	B	C	D	E	F	G
1		**Años**					
2		**2014**		**2015**		**2016**	
3	**Paises**	**VentasRetail**	**Crecimiento**	**VentasRetail**	**Crecimiento**	**VentasRetail**	**Crecimiento**
4	Antigua y Barbuda	10.697.643		12.805.079	19,7%	13.765.459	7,5%
5	Argentina	57.320.581		71.880.009	25,4%	74.395.809	3,5%
6	Bahamas	47.841.079		35.211.034	-26,4%	27.887.139	-20,8%
7	Barbados	31.064.936		43.366.651	39,6%	51.953.247	19,8%
8	Belice	72.771.283		77.865.273	7,0%	84.094.495	8,0%
9	Bolivia	33.205.796		54.789.563	65,0%	94.785.945	73,0%
10	Brasil	85.586.043		104.414.972	22,0%	154.534.159	48,0%
11	Canadá	89.377.073		161.772.502	81,0%	241.041.028	49,0%

Funciones que evalúan expresiones durante un período de tiempo

Existe un conjunto más de funciones de inteligencia de Tiempo en DAX: estas son las funciones que evalúan una expresión en un período de tiempo específico. Puede lograr lo mismo usando CALCULATE y otras funciones de Inteligencia de Tiempo. Por ejemplo,

= TOTALMTD (Expresión, ColumnaDeFecha [, SetFilter])

Esta expresión es exactamente la misma que esta combinación de funciones:

= CALCULATE (Expresión, DATESMTD (ColumnaDeFecha) [, SetFilter])

Pero será más simple y más fácil para los usuarios usar estas funciones DAX cuando encajan bien en el problema que debe ser resuelto. Las funciones DAX que hacen esto son:

TOTALMTD (Expresión, ColumnaDeFecha [, SetFilter])

TOTALQTD (Expresión, ColumnaDeFecha [, SetFilter])

TOTALYTD (Expresión, ColumnaDeFecha [, SetFilter] [, YE_Date])

Estas tres funciones, realizan sumas acumuladas sobre la expresión, de los meses (TOTALMTD), trimestres (TOTALQTD) o años (TOTALYTD) en un conjunto de datos especificado.

Funciones que calculan los saldos de apertura y cierre

Para las funciones que calculan los saldos de apertura y cierre, hay ciertos conceptos que son útiles. En primer lugar, el saldo inicial (OPENINGBALANCE) de cualquier período es el mismo que el saldo final del período anterior. El saldo de cierre (CLOSINGBALANCE) incluye todos los datos hasta el final del período, mientras que el saldo de apertura no incluye datos del período actual.

Estas funciones siempre devuelven el valor de una expresión evaluada para un punto específico en el tiempo, el último valor de fecha posible en un período de calendario. El saldo inicial se basa en la última fecha del período anterior, mientras que el saldo final se basa en la última fecha del período actual. El período actual siempre está determinado por la última fecha en el contexto de fecha actual.

OPENINGBALANCEMONTH (Expresión, ColumnaDeFecha [, SetFilter])

OPENINGBALANCEQUARTER (Expresión, ColumnaDeFecha [, SetFilter])

OPENINGBALANCEYEAR (Expresión, ColumnaDeFecha [, SetFilter] [, YE_Date])

CLOSINGBALANCEMONTH (Expresión, ColumnaDeFecha [, SetFilter])

CLOSINGBALANCEQUARTER (Expresión, ColumnaDeFecha [, SetFilter])

CLOSINGBALANCEYEAR (Expresión, ColumnaDeFecha [, SetFilter] [, YE_Date])