

Material imprimible

Curso de Power BI Desktop

## Módulo 5: Inteligencia de tiempo

### Tabla calendario

Normalmente, debería de ser el primer paso en todo informe que se precie. Si tu base de datos tiene una columna de fechas, deberías, igualmente crear una tabla maestra de fechas para poder jugar con estas y crear distintas variaciones de la misma para explotar al máximo la información disponible.

Es bastante sencillo de implementar, ya que únicamente deberemos de acudir a la opción de “Crear Tabla”, agregarla y marcarla como tabla de fechas.

Para crear nuestro calendario podemos utilizar dos funciones creadas a partir de DAX

**CALENDAR:** esta función insertará en nuestra tabla una columna que contiene fechas continuadas en el tiempo. El intervalo de fechas puede llevar una fecha de inicio y otra de fin. Por ejemplo, para insertar todo el año 2019 utilizaríamos una función como esta que vemos a continuación:

**=CALENDAR (DATE (2019, 1, 1), DATE (2019, 12, 31))**

Si queremos hacer referencia a nuestra columna de fecha insertada dentro de nuestra base de datos, a la que hemos llamado “fecha”, utilizaremos la fórmula que sigue:

**= CALENDAR([Fecha])**

**CALENDARAUTO:** establece una columna automática de calendario sin necesidad de tener una referencia de fecha de inicio y otra de fecha de fin. La fórmula a insertar en nuestra tabla sería la que vemos a continuación:

**=CALENDARAUTO()**

Una vez creada la tabla de fechas deberemos de relacionarla con la columna “fecha” de nuestra base de datos principal, en esta captura de pantalla que vemos a continuación sería en concreto la de meses. Lo haremos desde la pestaña Modelo de nuestro libro de datos de Power BI Desktop, y arrastraremos nuestra columna de fechas hasta la de

---

calendario para relacionarlas. Lo más común es establecer una relación 1:1 y en ambas direcciones.

## Inteligencia de tiempo

Data Analysis Expressions (DAX) incluye la importante funcionalidad de Time Intelligence o inteligencia de tiempo, funciones integradas en el lenguaje con las que podemos manipular y agregar datos en función de periodos temporales para construir y comparar cálculos sobre dichos periodos. Este tipo de análisis de series temporales es esencial en cualquier modelo de datos.

Aunque Time Intelligence es un tema muy amplio, vamos a centrarnos en uno de los intereses principales del estudio de las series temporales, que reside en la evaluación de los cambios de una magnitud a lo largo del tiempo. Estos cambios se valoran a través de las denominadas tasas de variación, que surgen de la comparación de los valores de la serie en dos periodos de tiempo distintos.

Uno de los puntos fuertes que tiene el lenguaje **DAX** es que nos permite comparar datos por periodos de tiempo de forma sencilla y rápida gracias a sus funciones de inteligencia de tiempo.

Describiremos las funciones de **Time Intelligence** más utilizadas con sus respectivos casos prácticos.

### 1. SAMEPERIODLASTYEAR

Posiblemente se trate de la función de inteligencia de tiempo más utilizada. Gracias a este tipo de uso, podemos comparar un importe entre un periodo de tiempo, con su importe en el mismo periodo de tiempo, pero del año anterior.

Para utilizarla simplemente tenemos que **crear una medida**, y en la parte del filtro poner la función **SAMEPERIODLASTYEAR** con el parámetro de tipo fecha por el que queramos comparar, por defecto pondremos la columna de tipo fecha de nuestra dimensión de tiempo.

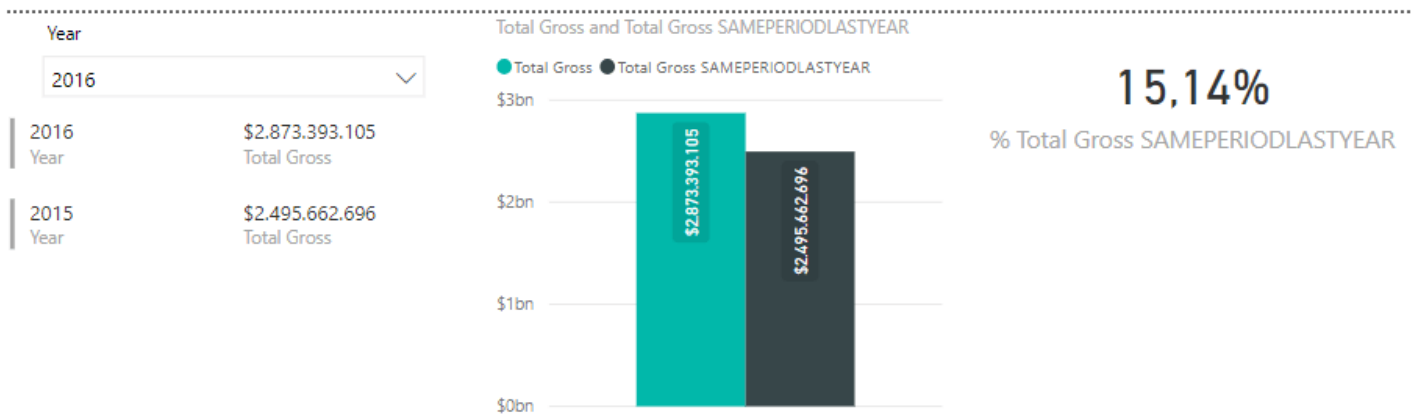
Total Gross SAMEPERIODLASTYEAR =

CALCULATE(SUM('Disney Movies'[Total Gross]);SAMEPERIODLASTYEAR(DimDate[Date]))

Una vez presentemos esta medida a nuestro cliente, lo primero que nos indicará es que quiere el **KPI** del porcentaje de mejora respecto al año anterior. Para realizar esta medidas en nuestro caso dividiremos **“Total Gross”** del periodo seleccionado / **“Total Gros”** del mismo periodo del año anterior.

% Total Gross SAMEPERIODLASTYEAR=

DIVIDE( SUM('Disney Movies'[Total Gross]);CALCULATE(SUM('Disney Movies'[Total Gross]);SAMEPERIODLASTYEAR(DimDate[Date])))-1



## 2. TOTALYTD, TOTALQTD & TOTALMTD

Cuando hablamos de importes acumulados debemos pensar siempre en las siguientes interpretaciones:

- **TOTALYTD**

Esta función evalúa la expresión a lo largo del intervalo que empieza el **primer día** del año y termina con la **última fecha** después de aplicar los filtros especificados.

Total Gross TOTALYTD =

TOTALYTD(SUM('Disney Movies'[Total Gross]);DimDate[Date])

Year  
2016

Year	Quarter	MonthNameShort	Total Gross	F. TOTALYTD
2016	Q1	Jan	\$27.569.558	\$27.569.558
2016	Q1	Feb		\$27.569.558
2016	Q1	Mar	\$341.268.248	\$368.837.806
2016	Q2	Apr	\$371.896.831	\$740.734.637
2016	Q2	May	\$485.126.730	\$1.225.861.367
2016	Q2	Jun	\$486.295.561	\$1.712.156.928
2016	Q3	Jul	\$55.483.770	\$1.767.640.698
2016	Q3	Aug	\$76.233.151	\$1.843.873.849
2016	Q3	Sep	\$21.420.368	\$1.865.294.217
2016	Q4	Oct		\$1.865.294.217
2016	Q4	Nov	\$478.614.952	\$2.343.909.169
2016	Q4	Dec	\$529.483.936	\$2.873.393.105
Total			\$2.873.393.105	\$2.873.393.105

Jan  
Jan + Feb  
Jan + Feb + Mar  
Jan + Feb + Mar + Apr  
...

- **TOTALQTD**

Evalúa la expresión a lo largo del intervalo que empieza el **primer día del trimestre** y termina con la última fecha después de aplicar los filtros especificados.

Total Gross TOTALQTD =

TOTALQTD(SUM('Disney Movies'[Total Gross]);DimDate[Date])

Year	Quarter	MonthNameShort	Total Gross	F. TOTALQTD
2016	Q1	Jan	\$27.569.558	\$27.569.558
2016	Q1	Feb		\$27.569.558
2016	Q1	Mar	\$341.268.248	\$368.837.806
2016	Q2	Apr	\$371.896.831	\$371.896.831
2016	Q2	May	\$485.126.730	\$857.023.561
2016	Q2	Jun	\$486.295.561	\$1.343.319.122
2016	Q3	Jul	\$55.483.770	\$55.483.770
2016	Q3	Aug	\$76.233.151	\$131.716.921
2016	Q3	Sep	\$21.420.368	\$153.137.289
2016	Q4	Oct		
2016	Q4	Nov	\$478.614.952	\$478.614.952
2016	Q4	Dec	\$529.483.936	\$1.008.098.888
Total			\$2.873.393.105	\$1.008.098.888

Apr  
Apr + May  
Apr + May + Jun  
Jul  
Jul + Aug  
...

- **TOTALMTD**

Al igual que las dos funciones anteriores evalúa la expresión a lo largo del intervalo que empieza al **principio del mes** y termina con la última fecha después de aplicar los filtros especificados.

Total Gross TOTALMTD =

TOTALMTD(SUM('Disney Movies'[Total Gross]);DimDate[Date])

### 3. PREVIOUSDAY, PREVIOUSMONTH, PREVIOUSQUARTER & PREVIOUSYEAR

Estas funciones, como bien indica el nombre, **devuelven la información** del día, mes, cuatrimestre o año anterior.

Total Gross PREVIOUSMONTH =

CALCULATE(SUM('Disney Movies'[Total Gross]);PREVIOUSMONTH(DimDate[Date]))

Year	MonthNameShort	Total Gross	F. PREVIOUSMONTH
2016	Jan	\$27.569.558	\$936.662.225
2016	Feb		\$27.569.558
2016	Mar	\$341.268.248	
2016	Apr	\$371.896.831	\$341.268.248
2016	May	\$485.126.730	\$371.896.831
2016	Jun	\$486.295.561	\$485.126.730
2016	Jul	\$55.483.770	\$486.295.561
2016	Aug	\$76.233.151	\$55.483.770
2016	Sep	\$21.420.368	\$76.233.151
2016	Oct		\$21.420.368
2016	Nov	\$478.614.952	
2016	Dec	\$529.483.936	\$478.614.952
<b>Total</b>		<b>\$2.873.393.105</b>	<b>\$936.662.225</b>

**NEXTDAY, NEXTMONTH, NEXTQUARTER & NEXTYEAR** son sus equivalentes para el siguiente día, mes, cuatrimestre o año.

### 4. DATEDIFF

A través de esta función podemos averiguar el **intervalo de tiempo** que hay **entre dos fechas**.

Los intervalos de tiempo que nos puede devolver esta función son los siguientes:

- **SECOND**
- **MINUTE**
- **HOURL**

- **DAY**
- **WEEK**
- **MONTH**
- **QUARTER**
- **YEAR**

Uno de los casos prácticos donde se utiliza **DATEDIFF** es para el cálculo de la **media** por un intervalo de tiempo específico, ya que, por defecto, la **tabla de hechos** está relacionada con la **dimensión de tiempo** por una columna de tipo **Date** por lo que nos calcularía la **media** por día. Si quisiéramos calcular la media por **hora** la expresión de la medida sería el siguiente:

AVG Hour =

```
DIVIDE(SUM('Disney Movies'[Total  
Gross]);DATEDIFF(MAX(DimDate[Date]);MIN(DimDate[Date]);HOUR))
```

1

2

AVG Hour =

## 5. DATEADD

Siempre que queramos añadir un **intervalo de tiempo** a una fecha, esta es la función que debemos utilizar. Un caso práctico podría ser que un cliente tuviera un problema hace 2 años y quisiera comparar la producción actual con la de hace 2 años. Para realizar esta medida utilizaríamos la función **SAMEPERIODLASTYEAR** pero en la parte donde añadimos el parámetro de la fecha, añadiríamos la fecha pero de hace 2 años.

Total Gross SAMEPERIODLAST 2 YEARS =

```
CALCULATE(SUM('Disney Movies'[Total  
Gross]);SAMEPERIODLASTYEAR(DATEADD(DimDate[Date]; -2;YEAR)))
```

Year	Total Gross SAMEPERIODLASTYEAR	Total Gross SAMEPERIODLAST 2 YEARS
2016	\$2.495.662.696	\$1.821.352.070
2015	\$1.514.179.473	\$1.452.972.057
2014	\$1.821.352.070	\$1.103.207.068
2013	\$1.452.972.057	\$1.518.975.880
2012	\$1.103.207.068	\$1.215.142.753
2011	\$1.518.975.880	\$968.276.109
2010	\$1.215.142.753	\$1.436.787.754
2009	\$968.276.109	\$1.427.356.974
2008	\$1.436.787.754	\$970.646.842

Otras funciones que también nos pueden servir de ayuda para realizar consultas **DAX** son las siguientes:

- **TODAY:** Devuelve la fecha actual.
- **NOW:** Devuelve la fecha actual con horas, minutos y segundos.
- **FIRSTDATE:** Devuelve la primera fecha.
- **LASTDATE:** Devuelve la última fecha.
- **FIRSTNONBLANK:** Devuelve el primer valor que no está en blanco.
- **LASTNONBLANK:** Devuelve el último valor que no está en blanco.
- **DATESMTD, DATESQTD & DATESYTD:** Devuelve un conjunto de fechas del mes, cuatrimestre o año hasta la fecha actual.
- **ENDOFMONTH, ENDOFQUARTER & ENDOFYEAR:** Devuelve el final del mes, cuatrimestre o año.
- **STARTOFMONTH, STARTOFQUARTER & STARTOFYEAR:** Devuelve el inicio del mes, trimestre o año.