

# POWER BI MASTER EN DAX

APRENDE TODO SOBRE DAX



POR: OMAR WALLY



# TABLA DE CONTENIDO

INTRODUCCIÓN AL MODELO DE DATOS 01

QUÉ ES EL MODELO DE DATOS 02

CARGANDO DATOS 03

CÓMO CREAR UNA RELACIÓN 04


MEDIDAS 05

FÓRMULAS ESENCIALES: SUM, COUNT,  
COUNTROWS, MIN, MAX, COUNTBLANK, DIVIDE 06

TABLA DE DATOS Y TABLA DE BUSQUEDA 07

LOS ITERADORES: SUMX Y AVERAGEX 08





<b>DAX CALCULATE</b>	<b>09</b>
<b>DAX IF Y SWITCH</b>	<b>10</b>
<b>DAX VALUES, HASONEVALUE Y SELECTVALUE</b>	<b>11</b>
<b>ALL, ALLEXCEPT, ALLSELECTED</b>	<b>12</b>
<b>LA FUNCIÓN FILTER</b>	<b>13</b>
<b>INTELIGENCIA DE TIEMPO (TIME INTELLIGENCE)</b>	<b>14</b>
<b>LA FUNCIÓN RELATED</b>	<b>15</b>
<b>EL USO DE TABLAS DESCONECTADAS</b>	<b>16</b>
<b>USO DE VARIABLES</b>	<b>17</b>
<b>MÚLTIPLES TABLAS Y PRESUPUESTO</b>	<b>18</b>

## 1. Introducción al modelo de datos

El motor de modelado de datos que se usa dentro de Power BI es el mismo que se usa en Power Pivot para Excel. El modelado de datos no es un término que a menudo es familiar para los usuarios de negocios, ya que normalmente es el dominio de los profesionales de IT BI. Pero este ya no es el caso para los usuarios comunes de Excel y analistas, gracias a la introducción de Power BI y Power Pivot para Excel.

## 2. Qué es el modelo de datos

El modelado de datos es el proceso de tomar datos de varias fuentes; cargar, estructurar y relacionar datos lógicamente con otros datos; y mejorar, embellecer y, en general, preparar los datos para su uso. El objetivo es permitir que los datos se utilicen sin tener que escribir una consulta personalizada cada vez que desee ver un subconjunto de datos diferente.

El proceso de modelado de datos incluye:

Determinación de la estructura y forma óptimas de los datos de origen para analizar, pueden ser todos los datos, datos completos o datos de resumen.

- Cargar los datos de la fuente en el modelo de datos (Power BI en este caso).
- Definir las relaciones lógicas entre las distintas tablas (que es similar a lo que hace con BUSCARV en Excel, excepto que los datos permanecen en la tabla de origen en Power BI).
- Definir tipos de datos (por ejemplo, especificar si una columna de datos es numérica o una columna de valores es de moneda o una columna de campos de texto).
- Crear nuevos conocimientos a partir de los datos de origen para que pueda analizar conceptos que no existen de forma nativa en los datos de origen, pero que se pueden calcular o crear dentro del modelo de datos. Por ejemplo, si tiene una tabla de datos transaccionales con precio de costo y precio de venta, puede extender el modelo de datos para incluir cálculos de margen, porcentaje de margen, etc., aunque estos conceptos no estén explícitamente en los datos de origen. Una vez que haya modelado estos nuevos datos en el modelo de datos, las personas que usen sus archivos modelos pueden reutilizarlos una y otra vez.
- Dar nombres significativos a sus nuevas ideas de negocios (es decir, a sus medidas).

Cuando aprende el idioma de DAX y une sus tablas de datos en Power BI, en realidad está aprendiendo el **modelado de datos**. El término puede dar un poco de miedo, pero no hay razón para preocuparse. Para cuando haya terminado de leer este libro, estará bien encaminado para convertirse en un experto en la modelación y transformación de datos con Power BI. Solo use las técnicas cubiertas en este libro y tenga en cuenta que lo que realmente está haciendo es aprender a ser un modelador de datos.

## Cómo obtener Power BI Desktop

Todas las lecciones dadas en este libro utilizan Power BI Desktop como herramienta de modelado de datos. Para descargar esta herramienta gratuita, debe dirigirse al siguiente enlace para descargar Power BI Desktop e instalar en su equipo:

<http://powerbi.com>, siga todos los pasos indicados en la página de Power BI.



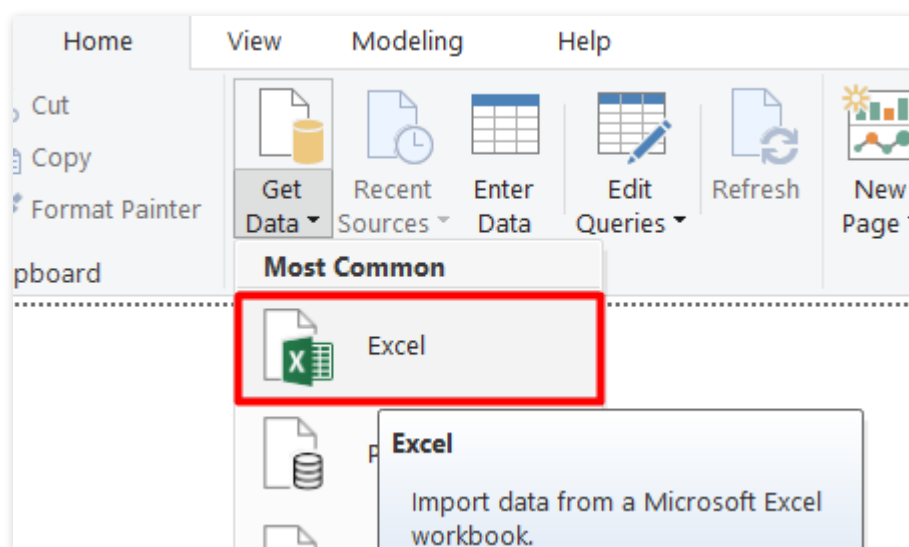
## Power BI Pro vs. Power BI (Cuenta Gratis)

Power BI Desktop es una herramienta gratuita de creación de modelos de datos que se utiliza para crear modelos de datos e informes. Después de crear un informe, puede compartirlo como un archivo con otros usuarios de Power BI Desktop (al igual que puede hacerlo con un archivo de Excel xlsx). Power BI Desktop también puede publicar sus informes en PowerBI.com, donde los informes se pueden compartir fácilmente con otros usuarios de Power BI; Esta es la forma normal de compartir.

Necesita una cuenta de Power BI para usar PowerBI.com. Hay dos tipos de cuentas Power BI: gratis y Pro. Una cuenta gratuita le permite usar la mayoría de las funciones de PowerBI.com, pero hay algunas excepciones notables. No puede compartir su trabajo con otras personas que no sean públicas (no seguras), y no puede acceder a los datos utilizando Analizar en Excel o exportar a PowerPoint. Si desea compartir informes y cuadros de mandos con otros usuarios, todos los usuarios que quieran ser parte del intercambio deben tener cuentas de Power BI Pro.

### 3. Cargando Datos

La imagen a continuación muestra el conector de datos que aparece cuando se conecta a una base de datos en Excel. (Hay un conector de datos diferente para cada fuente de datos. Verá cómo llegar a las distintas pantallas de conectores de datos más adelante en este capítulo). Hay dos modos que puede usar en Power BI Desktop al cargar datos de una base de datos, una es por importación de datos y otra por conexión directa a una fuente de datos a un sistema ERP o mediante SQL Server.



Este libro se centra en el modo Importar, que realiza una copia física de los datos de la fuente y la carga en Power BI Desktop. Cuando usa el modo Importar, Power BI Desktop carga una copia completa de los datos de origen en el modelo de datos como el primer paso en el proceso. Una vez que está cargado, puede compartir su libro de trabajo pbix con otros, y no es necesario que nadie más tenga acceso directo a los datos de origen. Alternativamente, puede publicar sus informes en PowerBI.com y compartir los contenidos con otros desde allí. Cuando publica un informe en PowerBI.com, también se carga una copia completa de los datos en la nube, sin la necesidad de acceder a los datos de origen.

Cuando carga datos, tiene que decidir qué datos cargar, incluidas qué tablas, qué columnas en cada tabla, etc. Esto es lo que se llama "la forma de los datos". La siguiente sección "Cargando datos desde una fuente" muestra cómo cargar los datos que se han preparado para usted. Pero debe tener en cuenta que el proceso de decidir qué datos cargar es una parte importante del proceso de modelado de datos, como se explica más adelante en este capítulo.

## Cargando datos desde una fuente

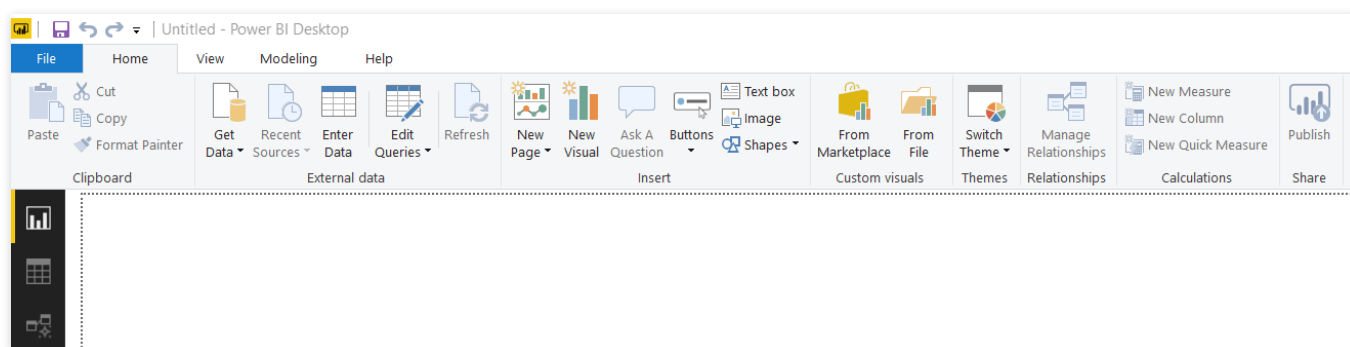
A continuación, le indicamos cómo: cargar datos de una nueva fuente. Para acceder al material de trabajo, favor de dar clic en el botón de abajo para que descargue una copia de la base de datos AmericanAdventure utilizada en este libro, una vez descargado el material, descomprímalo y colóquelo en un lugar que sea fácil de encontrar. Comenzará cargando las siguientes tablas de la base de datos de AmericanAdventure Access:

- Ventas
- Productos
- Países
- Clientes
- Calendario

**Descargar Ahora!**  
Material de trabajo

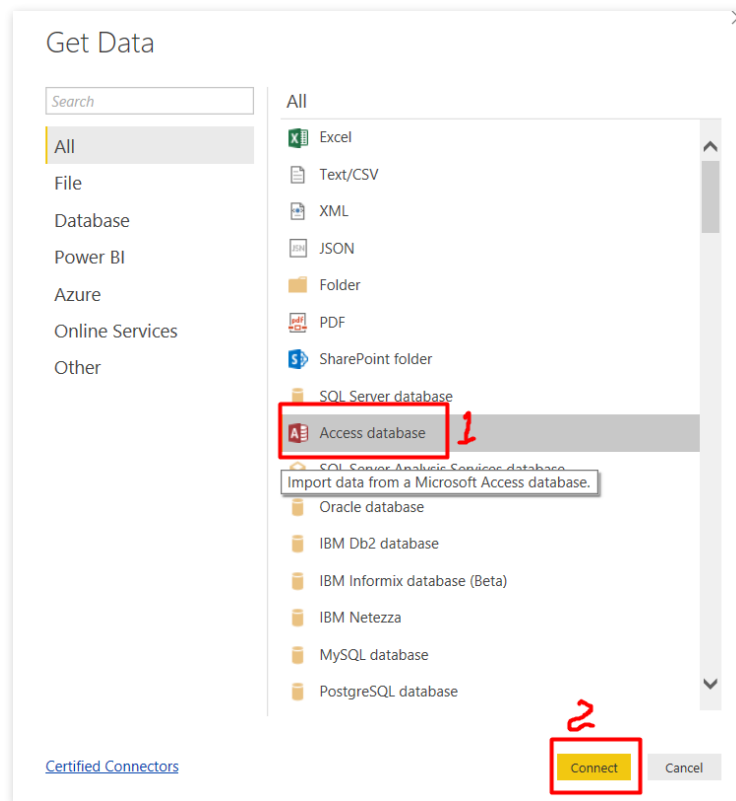
Los siguientes pasos le muestran cómo cargar estas tablas y cómo prepararlas para su uso en Power BI:

1. Abra Power BI Desktop. Debería ver un archivo de escritorio de Power BI en blanco con un menú en la parte superior, como se muestra a continuación.

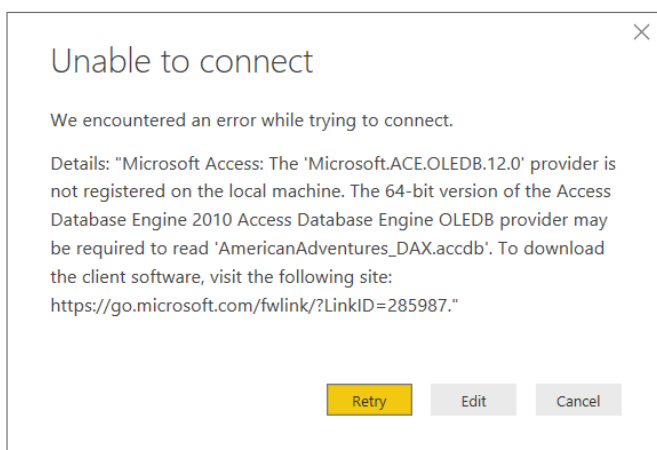


2. Desde el menú de inicio en Power BI Desktop, seleccione Obtener datos, Más (More) y Todos (All), Acceder a la base de datos (fijese en el número 1 a continuación) y luego haga clic en Conectar (mire el paso el número 2 a continuación).

2. Desde el menú de inicio en Power BI Desktop, seleccione Obtener datos, Más (More) y Todos (All), Acceder a la base de datos (fijese en el número 1 a continuación) y luego haga clic en Conectar (mire el paso el número 2 a continuación).



En este punto, es posible que las cosas salgan mal, especialmente la primera vez que cargue datos desde Access. La causa más común de problemas es que tiene Microsoft Office de 32 bits instalado en su computadora y Power BI Desktop de 64 bits. En ese caso, puede ver un mensaje similar a este:

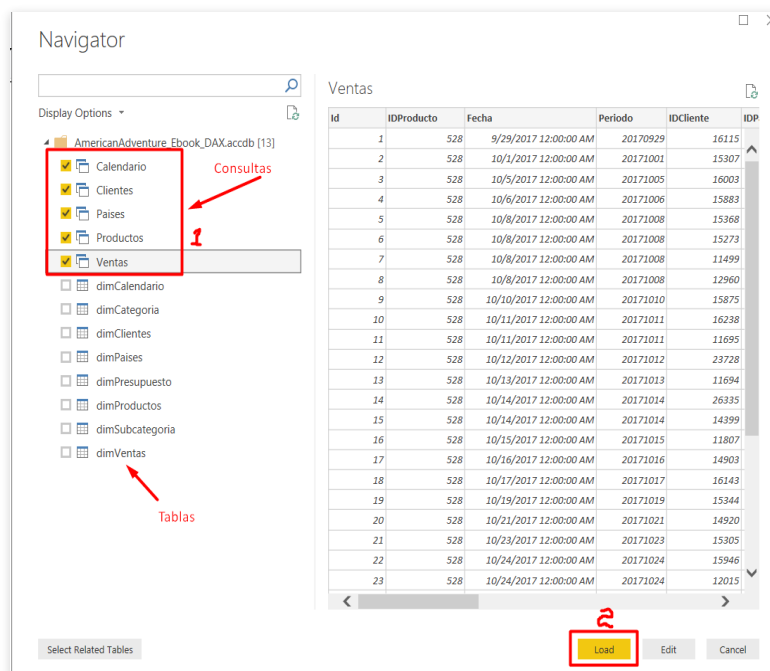




Si esto sucede, no entre en panico, le recomiendo encarecidamente que mantenga la versión de 64 bits de Power BI Desktop, ya que la necesitará para realizar cualquier procesamiento de datos serio con grandes conjuntos de datos. Puedes resolver el problema instalando la versión de access que sugiere Power BI que es la versión de access de 64bits. Ir al enlace que se menciona en la imagen.  
<https://go.microsoft.com/fwlink/?LinkID=285987>."

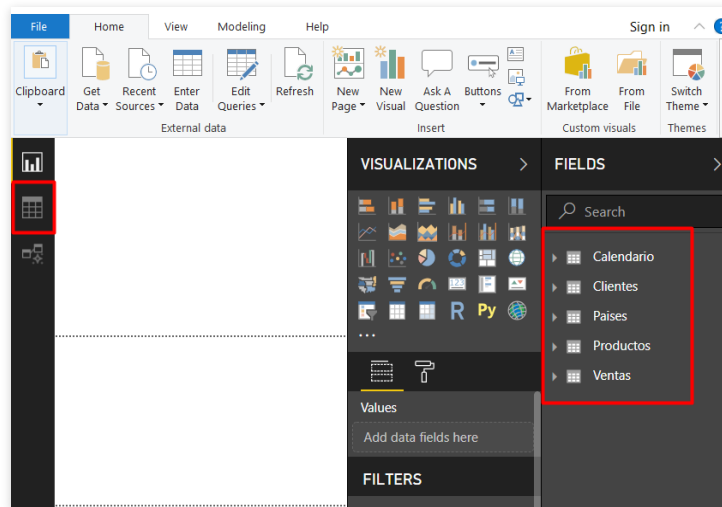
3. Busque la ubicación de la base de datos de ejemplo que descargó y que descomprimió anteriormente y haga clic en Abrir.

4. Seleccione las cinco vistas en la parte superior de la lista colocando una marca de verificación en el cuadro junto a cada una, como se muestra a continuación. El panel del navegador muestra diferentes iconos para consultas / vistas y para tablas, como se puede ver a continuación. Seleccione solo las listas de "Consultas"



Luego de hacer Clic en Cargar, Power BI Desktop cargará sus datos en el panel. Después que se cierre el asistente para importar tablas, verá las cinco tablas que acaba de importar en Power BI en el lado derecho, como se muestra a continuación. Cada una de las tablas es una copia completa de los datos que importó de los archivos de origen (una base de datos de Access, en este ejemplo). No necesitará volver a los archivos de origen hasta que esté listo para actualizar los datos, generalmente cuando los datos cambian en algún momento en el futuro. Ahora bien esta es una de las muchas cosas geniales de Power BI: simplemente tendrá que actualizar los datos, haciendo clic en el icono actualizar y cuando los datos cambian, sus libros de trabajo se actualizan con los nuevos datos en pocos segundos.

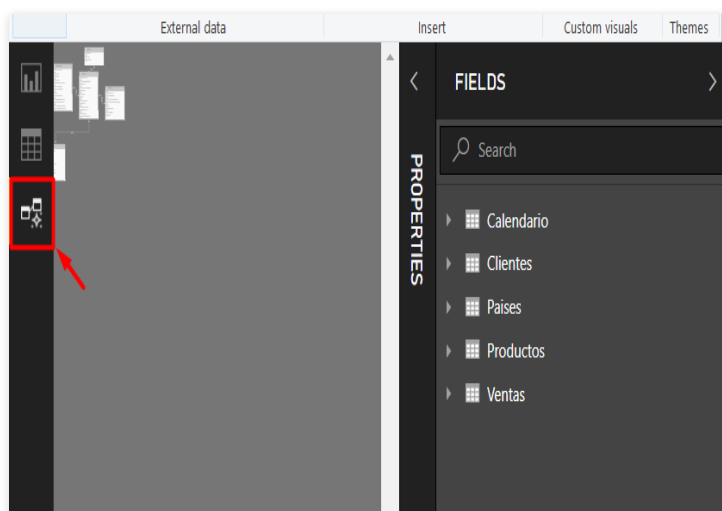
Luego de hacer Clic en Cargar, Power BI Desktop cargará sus datos en el panel. Después que se cierre el asistente para importar tablas, verá las cinco tablas que acaba de importar en Power BI en el lado derecho, como se muestra a continuación. Cada una de las tablas es una copia completa de los datos que importó de los archivos de origen (una base de datos de Access, en este ejemplo). No necesitará volver a los archivos de origen hasta que esté listo para actualizar los datos, generalmente cuando los datos cambian en algún momento en el futuro. Ahora bien esta es una de las muchas cosas geniales de Power BI: simplemente tendrá que actualizar los datos, haciendo clic en el icono actualizar y cuando los datos cambian, sus libros de trabajo se actualizan con los nuevos datos en pocos segundos.



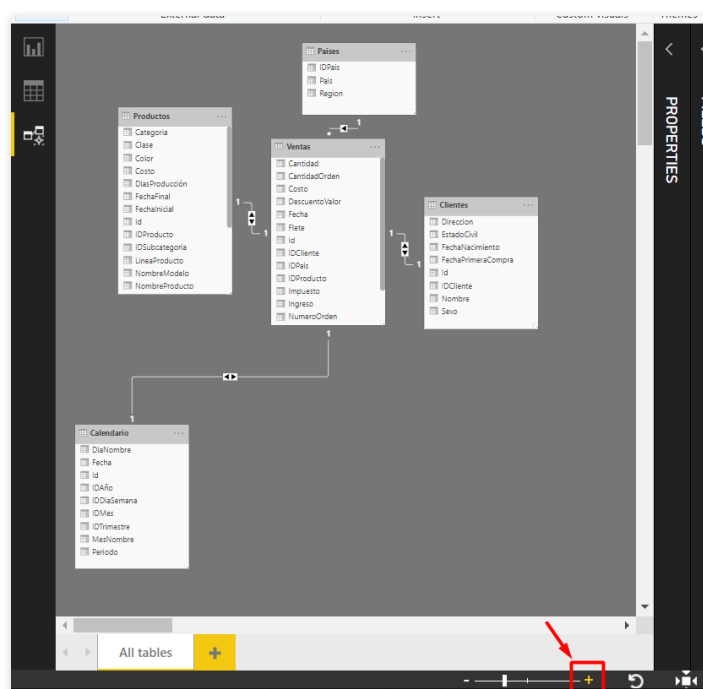
6. Ahora cambie a la vista de Datos seleccionando el ícono de Datos (ícono remarcado a su mano izquierda). En esta vista, puede ver los datos en las tablas.

Id	Fecha	IDDiaSemana	DíaNombre	MesNombre
1	7/1/2015 12:00:00 AM	4	miércoles	jul
2	7/2/2015 12:00:00 AM	5	jueves	jul
3	7/3/2015 12:00:00 AM	6	viernes	jul
4	7/4/2015 12:00:00 AM	7	sábado	jul
5	7/5/2015 12:00:00 AM	1	domingo	jul
6	7/6/2015 12:00:00 AM	2	lunes	jul
7	7/7/2015 12:00:00 AM	3	martes	jul
8	7/8/2015 12:00:00 AM	4	miércoles	jul
9	7/9/2015 12:00:00 AM	5	jueves	jul
10	7/10/2015 12:00:00 AM	6	viernes	jul
11	7/11/2015 12:00:00 AM	7	sábado	jul
12	7/12/2015 12:00:00 AM	1	domingo	jul
13	7/13/2015 12:00:00 AM	2	lunes	jul

7. La siguiente etapa del proceso de modelado de datos implica crear las relaciones lógicas entre las tablas. Cambie a la vista Relaciones haciendo clic en el ícono Relación



8. Si no puede ver las cinco tablas en la pantalla, haga clic en el botón Zoom para ajustar, que se muestra a continuación, para revelar las tablas ocultas.

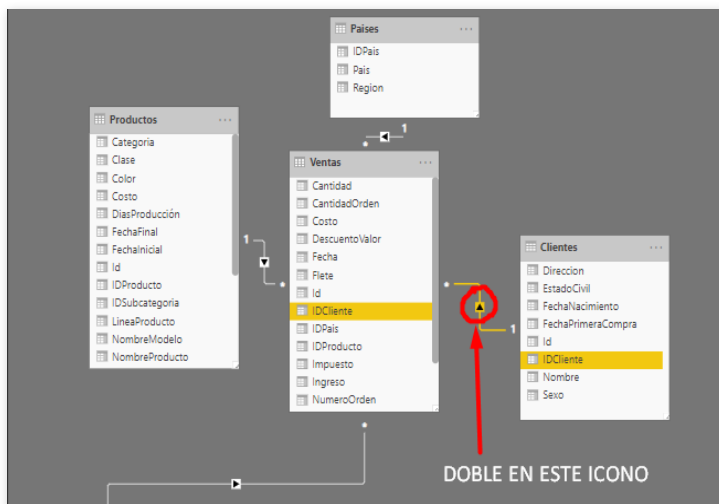


## 4. Cómo crear una relación

Una tabla de clientes normalmente tiene una lista de todos los clientes que una empresa tiene archivada. Pero es posible que algunos de estos clientes solo haya hecho una compra y algunos clientes hayan realizado muchas compras. Por lo tanto, para cada entrada en la tabla Clientes solo se tiene un registro único. Mientras que en la tabla de ventas pueden haber muchos registros de los clientes y puede que solo aparezca el ID de cliente y no su nombre.

La tabla de ventas se puede unir de forma lógica a la tabla de clientes utilizando la clave de cliente (a menudo llamada número de cliente o ID). Cuando estas tablas se unen a través de la clave del cliente, habrá una relación de uno a muchos o de muchos a uno (Clientes-Ventas) entre estas dos tablas.

Para realizar la relación mire las siguientes imágenes.



Edit relationship

Select tables and columns that are related.

Ventas

ID	IDProducto	Fecha	Periodo	IDCliente	IDPais	NumeroOrden	CantidadOrden	C
1	528	9/29/2017 12:00:00 AM	20170929	16115	4	SO55161		1
2	528	10/1/2017 12:00:00 AM	20171001	15307	4	SO55352		1
3	528	10/5/2017 12:00:00 AM	20171005	16003	4	SO55578		1

Clientes

ID	IDCliente	Nombre	FechaNacimiento	EstadoCivil	Sexo	Direccion	FechaPrimeraC
103	14691	Derrick Torres	11/13/1979 12:00:00 AM	M	M	4195, rue Léo Delibes	3/5/2018 12:00:00 AM
104	14997	Angel Cooper	6/5/1975 12:00:00 AM	M	M	Brunnenstr 6466	6/8/2016 12:00:00 AM
106	16454	Marco Suri	6/5/1977 12:00:00 AM	M	M	8510 G St.	8/16/2017 12:00:00 AM

Cardinality: Many to one (\*:1)

Cross filter direction: Single

☒ Make this relationship active

☐ Assume referential integrity

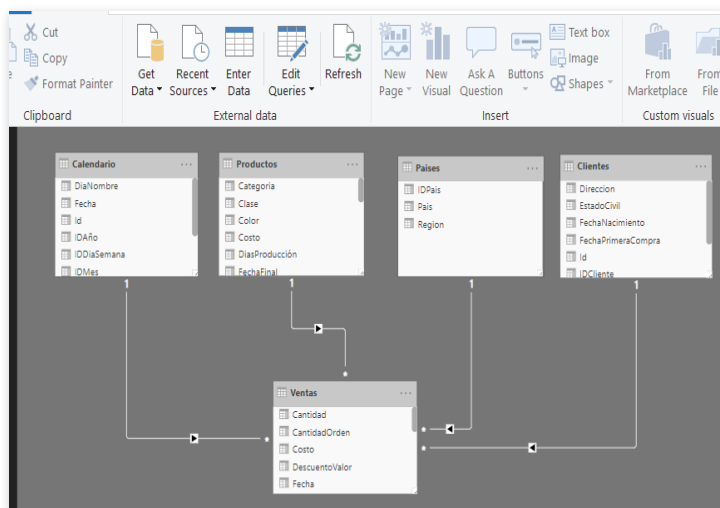
☐ Apply security filter in both directions

OK Cancel

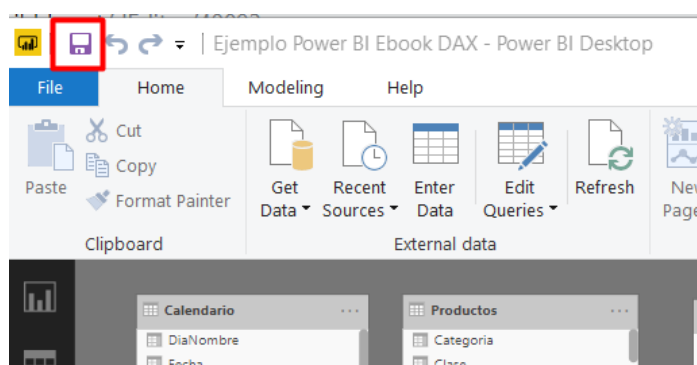
Luego proceda a realizar las siguientes relaciones con las demás tablas de búsqueda:

- Relacionar la tabla Calendario con la tabla de ventas, debe seleccionar en la tabla de ventas la columna fecha y en la tabla calendario la columna fecha y clic en Aceptar.
- Relacionar la tabla Productos con la tabla de ventas, debe seleccionar en la tabla de ventas la columna IDProducto y en la tabla Producto la columna IDProducto y clic en Aceptar.
- Relacionar la tabla País con la tabla de ventas, debe seleccionar en la tabla de ventas la columna IDPaís y en la tabla País la columna IDPaís y clic en Aceptar.
- Relacionar la tabla Clientes con la tabla de ventas, debe seleccionar en la tabla de ventas la columna IDCliente y en la tabla Clientes la columna IDCliente y clic en Aceptar.

Como puede ver en la imagen de abajo, hay un asterisco (\*) al final de la relación que apunta a la tabla de datos, y hay un (1) al final que apunta a la tabla de búsqueda, y hay una flecha que apunta hacia la Tabla de venta (más sobre esas flechas más adelante). Al colocar la tabla de datos de ventas en la parte inferior, se obtiene una pista visual de que las tablas en la parte superior de la pantalla son tablas de búsqueda.



Puede proceder a guardar el archivo de Power BI Desktop haciendo clic en el icono Guardar, que se muestra aquí, y especificando un nombre y una ubicación adecuados donde lo va a guardar.



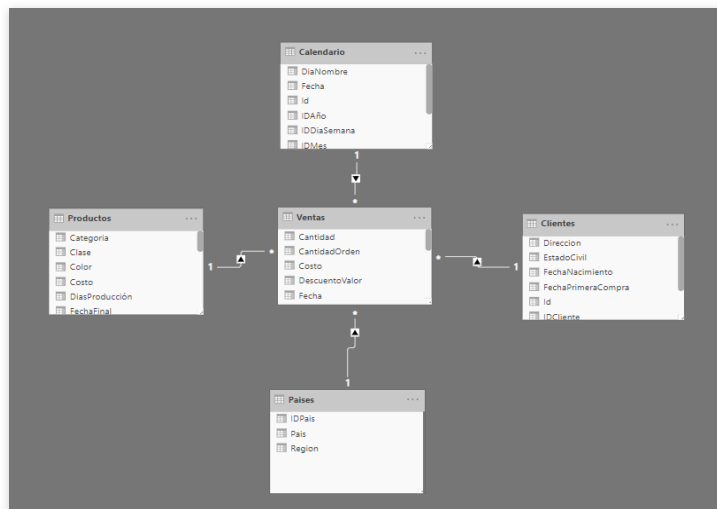
## Nota importante

Hagamos una pequeña pausa por un minuto, ya que quiero sugerir la forma óptima de los datos para Power BI. Cuando digo "forma" de datos, me refiero a cosas como cuántas tablas importas al modelo, cuántas columnas hay en cada tabla, qué columnas están en cada una de las tablas, etc. La configuración de datos es un tema enorme, y No tengo espacio aquí para discutirlo completamente. Pero quiero dar algunos consejos fundamentales para que comiences.

Una razón por la que este consejo es importante es que la forma de los datos en los sistemas transaccionales (o bases de datos relacionales) rara vez es la forma ideal para Power BI. Cuando el departamento de TI ejecuta un proyecto de BI empresarial, uno de los primeros pasos importantes es configurar los datos para que sean óptimos para la generación de informes. Este paso normalmente es completamente transparente para el usuario final (es decir, usted) y, por lo tanto, el usuario final está protegido de la necesidad de hacer esto. Pero estoy compartiendo esta información importante con usted aquí y ahora porque necesita entender la configuración de datos si desea tener modelos de datos Power BI eficientes y efectivos. Es poco probable que la información de sus datos de origen que usa en sus informes y reportes sea óptima.

## Elegir un Esquema (Estructura de la Tabla)

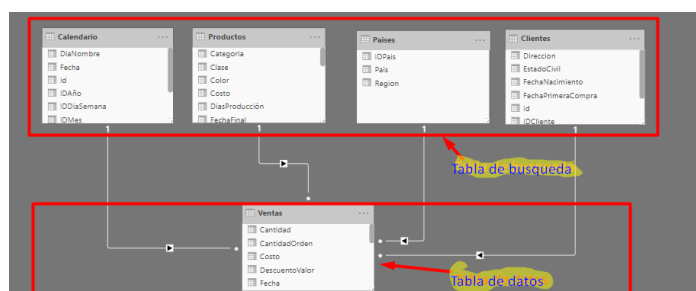
El enfoque generalmente aceptado para llevar datos a Power BI es traer los datos en lo que se conoce como un esquema en estrella. Este es un término técnico que proviene de la metodología Kimball (también conocido como modelado dimensional) y describe la forma lógica en que se deben estructurar los datos para un rendimiento óptimo de los informes. El objetivo del modelado dimensional es permitir al usuario visualizar los datos sin la necesidad de escribir una nueva consulta en la base de datos para cada informe. El diseño visual de las tablas en la siguiente imagen (que incluye exactamente los mismos datos que acaba de importar) lo ayuda a ver por qué se llama un esquema en estrella.



En este esquema, las tablas de datos (solo hay una en este ejemplo, tabla de Ventas) están rodeadas por tablas de búsqueda (Calendario, Clientes, Productos, Países en este ejemplo), y juntas forman una forma de estrella.

## El diseño visual de tablas en la vista de relaciones

Cuando se trata de posicionar visualmente las tablas en la vista Relaciones, He aprendido en estos últimos años a colocar las tablas de manera que las tablas de búsqueda estén ubicadas en la parte superior y lateral de la ventana y las tablas de datos estén en la parte inferior o en el centro de la ventana (como se muestra a continuación), las tablas de búsqueda en la parte de arriba y la de datos en la parte inferior.



## Comprensión de los dos tipos de tablas: tablas de búsqueda y tablas de datos

En el mundo de TI, las tablas de búsqueda se denominan tablas de dimensión y las tablas de datos se denominan tablas de hechos. Sin embargo, para los usuarios de negocios (analistas), sugiero usar como terminología : tablas de búsqueda y las tablas de datos. Una tabla de datos contiene información transaccional. En este libro, la tabla de datos contiene transacciones de ventas. Las tablas de búsqueda contienen información sobre grupos lógicos de objetos, como clientes, productos, tiempo ('Calendario'), etc.

Antes de Power BI y Power Pivot, un usuario de Excel necesitaba crear una tabla grande y plana en Excel antes de crear una tabla dinámica. A menudo, eso significaba escribir fórmulas con BUSCARV para llevar otros datos de otras tablas a la tabla de datos. Bueno, ya no es necesario traer datos de las tablas de búsqueda a las tablas de datos usando BUSCARV. En su lugar, simplemente puede cargar las tablas de búsqueda y unirlos con una relación.

### Tabla de Búsqueda

Debe tener una tabla de búsqueda para cada "objeto" que necesita para fines de informes. Por ejemplo, en los datos que se utilizan aquí, estos objetos son tiempo (calendario), clientes, productos y países. Una característica clave de una tabla de búsqueda es que contiene una y solo una fila para cada elemento individual en la tabla, y tiene tantas columnas como sea necesario para describir el objeto. Por lo tanto, solo hay una fila para cada cliente único en la tabla Clientes.

La tabla de Clientes tiene muchas columnas que describen a cada cliente, como el número de cliente (clave), el nombre del cliente, la dirección del cliente, etc., pero solo hay una fila para cada cliente. Cada fila es única, según el número de cliente, y no se permiten duplicados de número de cliente (clave o llave).

### Tabla de Datos

Es posible tener muchas tablas de datos, pero solo hay una en este ejemplo: la tabla Ventas. Esta tabla de datos contiene muchas filas (más de 60,000 en este caso) y todos los registros transaccionales de ventas que se produjeron durante varios años. Es importante destacar que la tabla de datos se puede unir a cada una de las tablas de búsqueda. En este caso, la tabla de ventas contiene una columna (técnicamente llamada clave foránea) que coincide con cada una de las claves de cada tabla de búsqueda (técnicamente denominada clave principal). Dicho de otra manera, la tabla de datos de ventas tiene cuatro columnas de clave foránea, una fecha, un número de cliente, un número de producto y una clave de país.

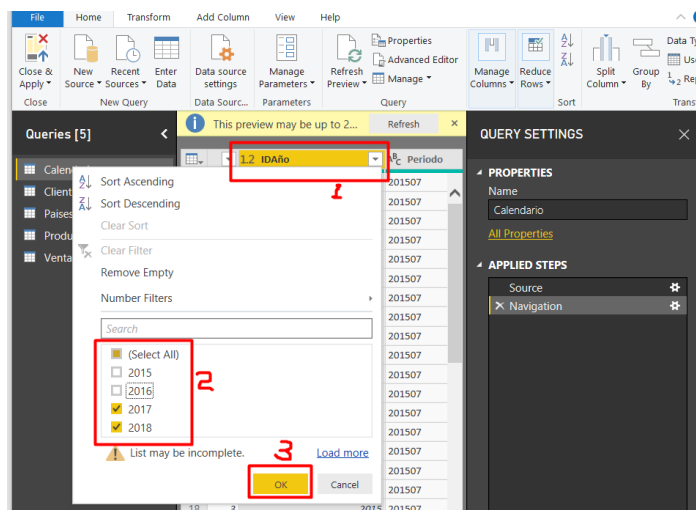


Estas columnas permiten que la tabla de datos de ventas se pueda unir de forma lógica a cada una de las tablas de búsqueda. Idealmente, las tablas de datos deberían tener muy pocas columnas pero tantas filas como sea necesario para traer todos los registros de datos. Las tablas de datos normalmente tienen muchas filas (a veces en decenas de millones).

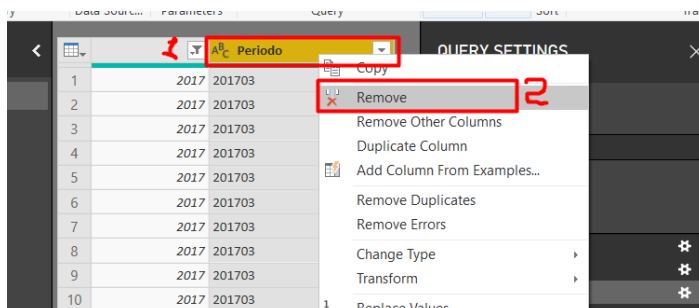
## Cómo hacer cambios en una tabla que ya está cargada

Suponga que desea realizar cambios en la tabla Calendario para que solo incluya fechas para los años 2017 y 2018, y también desea eliminar las columnas de Periodo de la tabla. Puedes hacer esto usando el Editor de consultas. Los siguientes pasos lo guían a través de cómo realizar cambios como estos en una tabla que ya está cargada:

1. A su mano izquierda seleccione el icono Data y luego en la lista de campos a la derecha (en la vista Informe o Vista de datos), haga clic sobre los tres puntitos en la tabla Calendario y seleccione Editar consulta (Edit Query).
2. En el Editor de consultas que aparece, navegue a la columna Calendario y haga clic en la flecha desplegable de la columna IDAño (ver imagen) y siga los pasos.



3. Deseleccione los años 2015 y 2016 de la lista desplegable y luego haga clic en Aceptar.
4. Elimine la columna de periodo, se posiciona sobre la columna, luego clic con el botón derecho y seleccione Eliminar columnas (# 2).



5. Haga clic en Cerrar y aplicar (Close & Apply) y luego guarde el libro pbix.

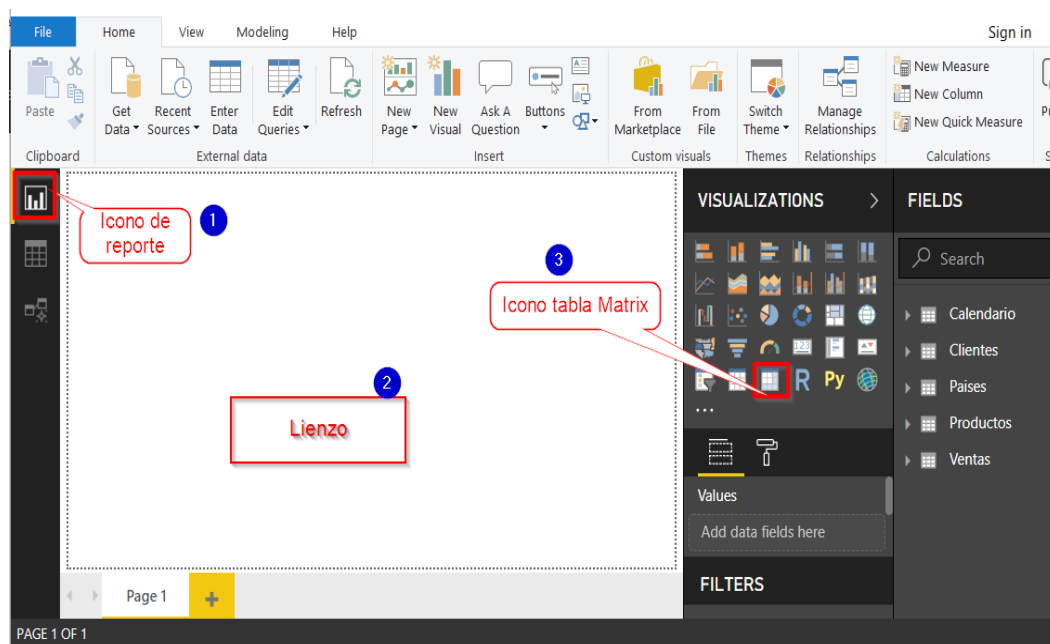
## Cómo Insertar una Matrix

El lienzo de informes de Power BI es muy diferente de lo que está acostumbrado a usar en Excel. El lienzo del informe se parece mucho más a una diapositiva de PowerPoint que a Excel. Esto puede confrontar bastante a los usuarios de Excel que se están iniciando con Power BI. Pero la buena noticia es que se sentirá cómodo con él en poco tiempo. En este momento, no hay ningún objeto de tabla dinámica en Power BI, pero hay una matriz.

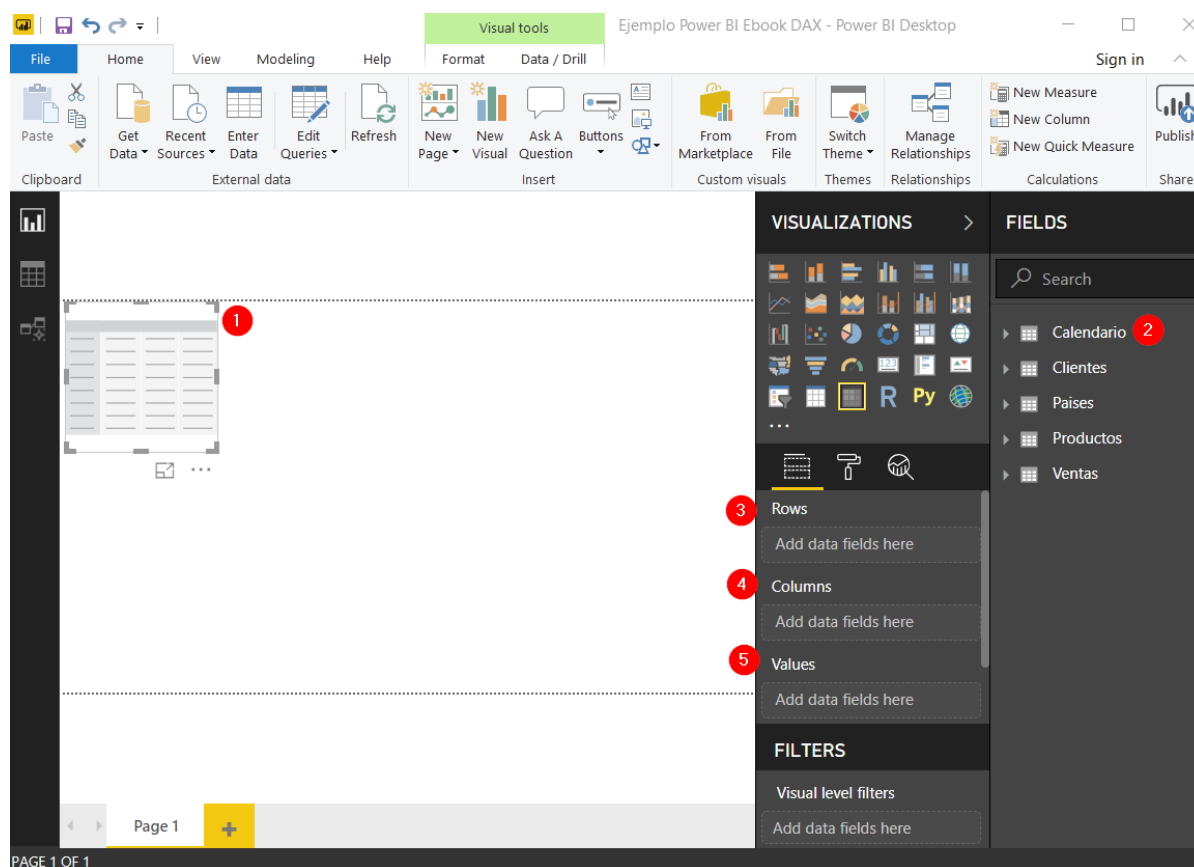
Una matriz es un sustituto muy cercano a una tabla dinámica, y es la mejor visualización para usar cuando se está iniciando. También hay una tabla visual que puedes usar. La tabla es similar a una matriz, pero no tiene una opción para agregar una columna. Puede explorar la diferencia usted mismo cambiando entre los dos tipos de visualización en el informe.

Hay varias formas de insertar una matriz en un informe. Le sugiero que lo haga así:

1. Abra la vista Informe haciendo clic en el ícono Informe.



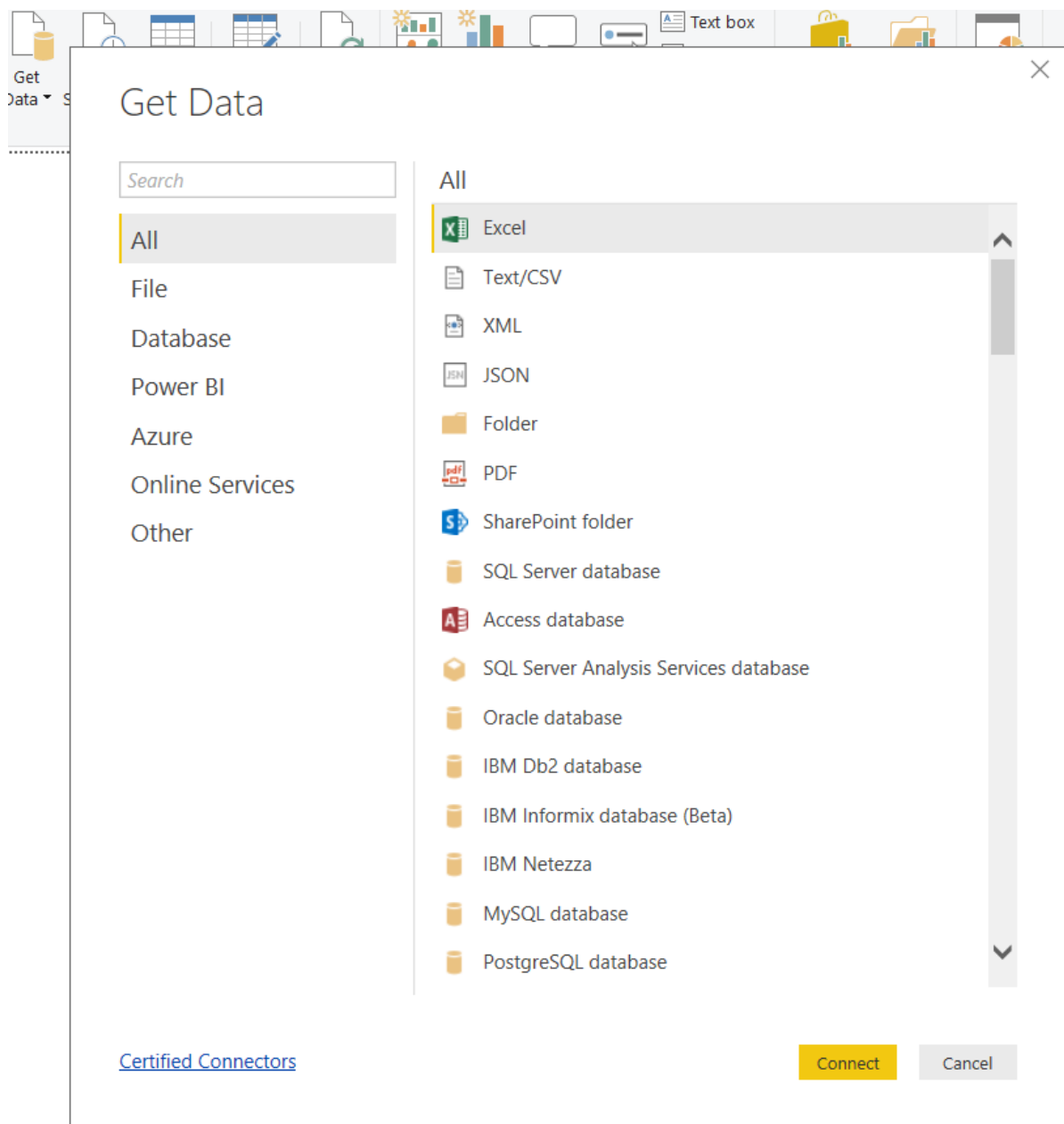
2. Haga clic una vez en una sección en blanco del lienzo del informe (# 2 arriba) y luego haga clic en el ícono de Matrix (# 3).



3. Ahora verá que aparece una nueva tabla de matriz en el lienzo (vea el número 1 arriba). Tenga en cuenta la lista de campos (# 2), las Filas (# 3), Columnas (# 4), y Valores (# 5) zonas de arrastrar y soltar en el lado derecho. Si tiene experiencia en el uso de tablas dinámicas en Excel, reconocerá que es muy similar a la experiencia de la tabla dinámica de Excel.

## Otras fuentes de datos

En este libro, le enseñé cómo importar datos de la base de datos de AmericanAdventure en Access, pero esto es, por supuesto, solo una de las muchas fuentes de datos a las que puede acceder. Hay muchos otros conectores de fuente de datos disponibles en Power BI. Para ver una lista completa, simplemente haga clic en Obtener datos, Más y luego en Todos. Luego puede ver la lista completa de conectores de datos actualmente admitidos en Power BI, como se muestra a continuación.

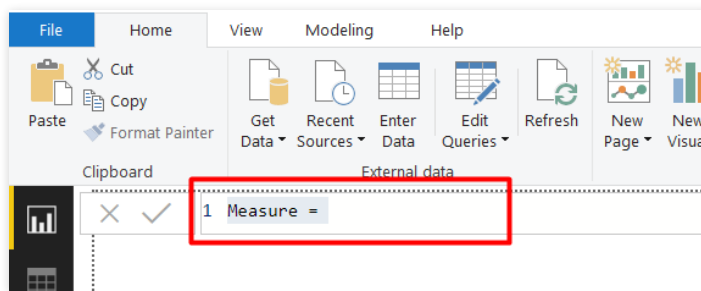


## 5. Medidas

Las medidas han existido durante muchos años en las versiones empresariales de las herramientas de Microsoft BI. Las medidas ahora han llegado al mundo de los usuarios empresariales y analistas de negocios que desean aprender a crear informes en Power BI. No hay nada confuso o difícil de aprender acerca de las medidas. Una medida es simplemente una fórmula DAX que le indica a Power BI que haga un cálculo en los datos. En cierto sentido, una medida se parece mucho a una fórmula en una celda de Excel. Sin embargo, la principal diferencia entre una fórmula en una celda en Excel y una medida es que una medida siempre opera en todo el modelo de datos, no solo en unas pocas celdas en una hoja de cálculo, sino en toda la columna de datos. Aprenderá más sobre esto más adelante, pero por ahora solo debe saber que una medida es una fórmula que calcula un resultado a partir de los datos cargados.

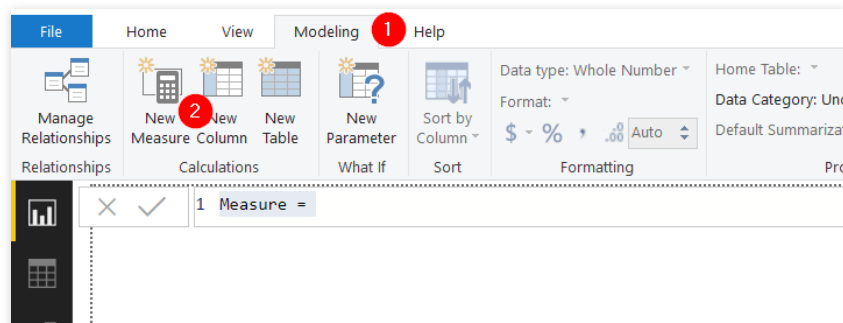
### Técnicas para escribir medidas

Las medidas en Power BI siempre se escriben en la barra de fórmulas, como se muestra a continuación.

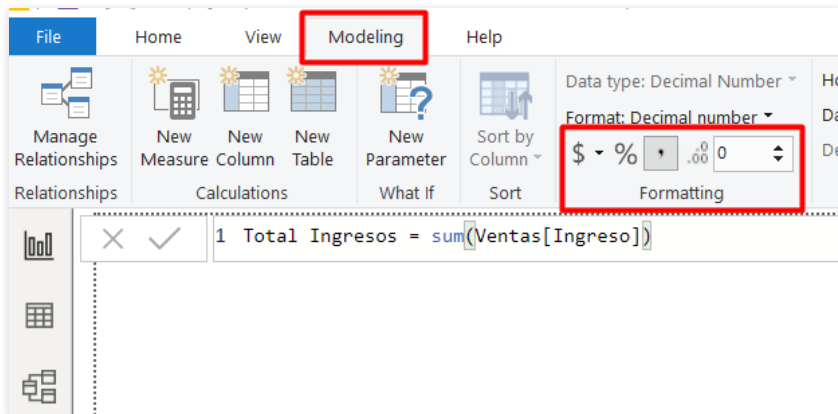
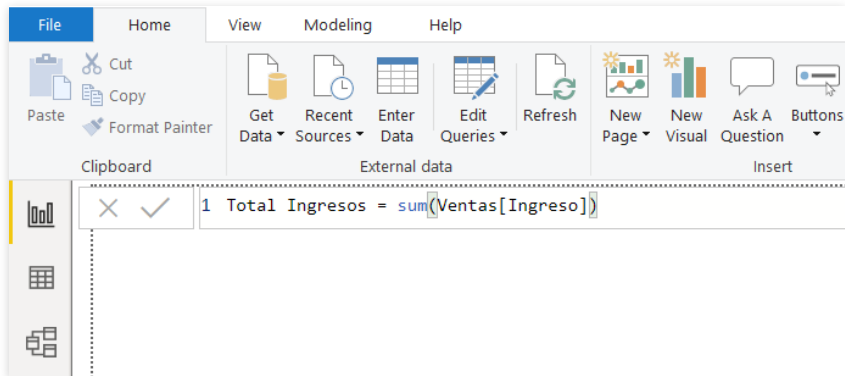


La barra de fórmulas no está visible a menos que tenga una medida seleccionada. Cuando selecciona una medida, la barra de fórmulas aparece justo debajo de la cinta, como se muestra arriba. Hay dos formas de iniciar el proceso de escritura de una nueva medida en Power BI:

- Primero, puede seleccionar la pestaña Modelado (ver # 1 a continuación) y luego hacer clic en el botón Nueva medida (# 2).



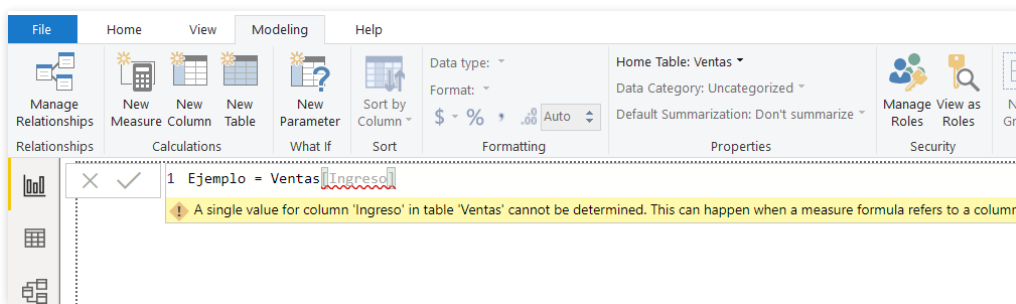
- En la barra de fórmulas, escriba la fórmula DAX Total Ingresos = SUM (Ventas [Ingreso]) y presione Enter. Escribilo directamente sobre el texto resaltado Medir = o Measure =.
- Para darle formato numérico, elige la opción "Modeling" en la barra de herramienta, luego formato, selecciona el que necesita, separar por comas y decimales y clic en Enter. Acostumbrate a aplicar el formato de forma inmediata para que te acostumbres.



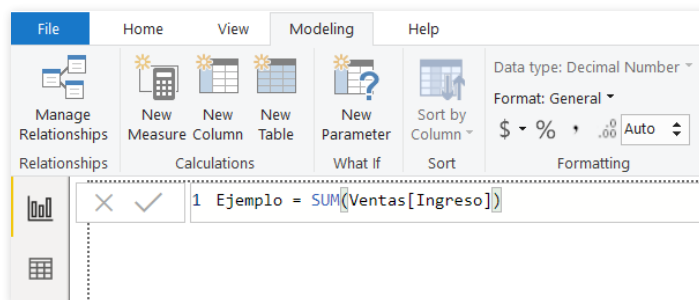
## 6. Fórmulas Esenciales: SUM(), COUNT(), COUNTROWS(), MIN(), MAX(), COUNTBLANK() y DIVIDE()

Este capítulo comienza con algunas fórmulas básicas de DAX para comenzar. La mayoría de las funciones DAX en este capítulo aceptan una columna como único parámetro, como este: = FORMULA (ColumnName). Las excepciones son = COUNTROWS (Tabla), que toma una tabla (no una columna) como parámetro, y DIVIDE (), que cubriremos más adelante en este capítulo. Todas las funciones en este capítulo (excepto DIVIDE ()) son funciones de agregación, o agregadores. Es decir, toman entradas de una columna o tabla y de alguna manera agregan los contenidos (de manera diferente para cada fórmula). Piense en la columna Ventas [Ingreso], que tiene más de 60,000 filas de datos. No se puede simplemente colocar toda la columna en una sola celda en una matriz porque Power BI no puede "encajar" en una columna de 60,000 números en una sola celda en la matriz.

El siguiente ejemplo muestra una fórmula DAX que usa una columna de forma "directa", sin ninguna función de agregación. Esto no funciona cuando está escribiendo una medida, como lo indica el mensaje de error.



Tiene que decirle a Power BI cómo agregar los datos de esta columna para que devuelva un solo valor a cada celda de la matriz. Todas las funciones de agregación en este capítulo convierten efectivamente una columna de valores en un solo valor. La forma correcta de escribir esta medida se muestra a continuación.



## Reutilizar medidas

Una capacidad importante en DAX es que puede reutilizar medidas para escribir otras medidas.

Digamos que creas una nueva medida llamada [Total Ingresos]. Una vez que existe esta medida en el modelo de datos de Power BI, se puede hacer referencia y reutilizar dentro de otras medidas. Por ejemplo, después de crear la medida [Total Ingresos], puede utilizar la siguiente fórmula para crear una nueva medida para el impuesto del 10% sobre la venta o los ingresos:  $\text{Impuesto total} = [\text{Total Ingresos}] * 0.1$ . Tenga en cuenta que la nueva medida [Impuesto total] es un cálculo basado en la medida original [Total Ingresos] multiplicado por 0.1.

Esto es una buena práctica en Power BI reutilizar las medidas dentro de otras medidas.

## Escribir DAX

Es hora de comenzar a escribir tu propio DAX para obtener algo de práctica. Cuando digo escribir, me refiero a sentarse frente a su PC, abrir el archivo de trabajo con los datos del Capítulo 1 cargados y realmente escribir algún DAX. Especialmente si nunca ha escrito fórmulas usando estas funciones, debería hacerlo físicamente ahora, mientras lee esta sección. Imaginarte a ti mismo haciéndolo en tu mente no es suficiente. Si aún no lo ha hecho, siga adelante y cargue los datos de prueba siguiendo los pasos del Capítulo 1. Una vez que esté cargado y preparado, estará listo para crear las nuevas medidas en los siguientes ejercicios prácticos.

## Practicar

Periódicamente, durante el resto de este libro, encontrará ejercicios de práctica diseñados para ayudarlo a aprender. Debes completar cada ejercicio a medida que llegues a él. Las respuestas a todos estos ejercicios de práctica se proporcionan en el archivo de PBI.

### [Total Ventas]

Ya deberías haber escrito esta medida anteriormente en este libro. Si no, escriba una nueva medida que sea el total de las ventas en la columna Ingresos de la tabla Ventas.  $\text{Total Ventas} = \text{SUM}(\text{Ventas}[\text{Ingreso}])$

### [Total Costos]

Cree una medida que sea la suma de la columna de costos en la tabla Ventas. Esta medida usa exactamente la misma estructura que la medida anterior, pero agrega el costo del producto en lugar del monto de los ingresos. Utilice la columna Costo de la tabla Ventas, para crear la medida Total Costos.  $\text{Total Costos} = \text{SUM}(\text{Ventas}[\text{Costo}])$

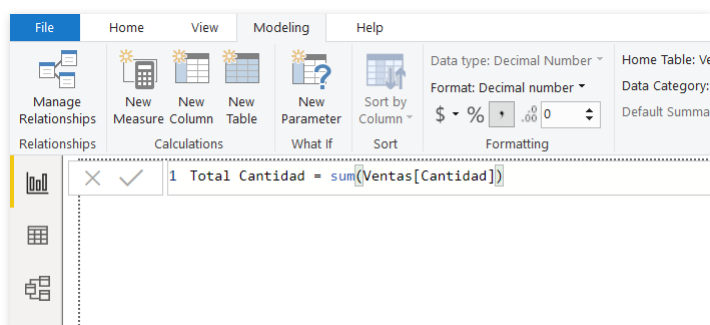
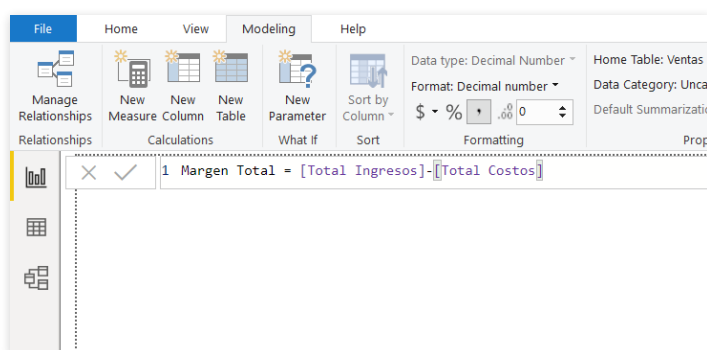


### [Margen Total]

Cree una nueva medida para el margen total, que es el total de ventas menos el costo total. Asegúrese de reutilizar las dos medidas que creó anteriormente en esta nueva medida. Es decir [Total Ventas] - [Total Costos]

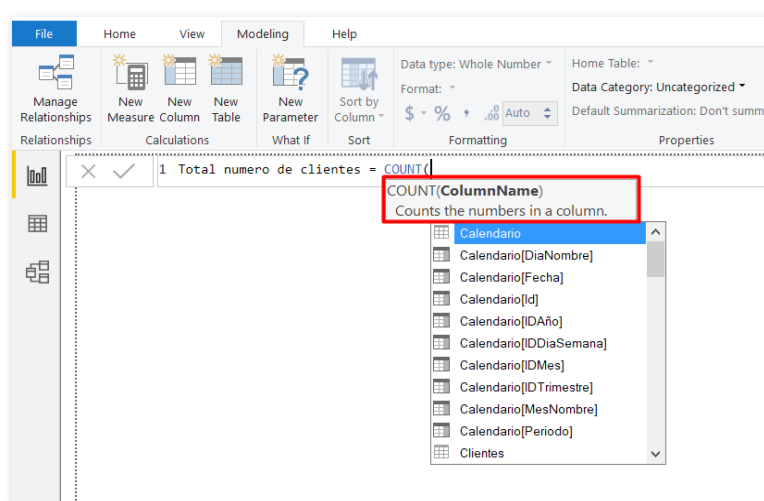
### [Total Cantidad]

Cree una medida que sea la suma de la columna de Cantidad de la tabla Ventas. Esta medida es Total cantidad. Total Cantidad=SUM(Ventas[Cantidad])



## La función COUNT

A medida que escribe la fórmula que se muestra a continuación usando COUNT (), tómese el tiempo para ver de nuevo cómo la "autoayuda y autocompletar" puede ayudarlo a escribir DAX. Recuerde que cada vez que escribe una nueva fórmula, puede pausar, y la autoayuda muestra la sintaxis y una descripción de la función. La descripción incluye información muy útil. Por ejemplo, en la siguiente figura, la información sobre herramientas dice que esta función "cuenta los números en una columna". Esto le brinda tres datos muy útiles. Ya has elaborado el primero: cuenta. Además, esta información sobre herramientas le dice que la función cuenta los números y también que los números deben estar en una columna. Esta debería ser suficiente información sobre la función COUNT () para que usted escriba algunas medidas al usarla.



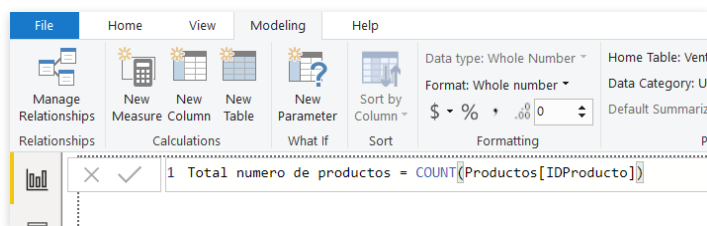
### [Total numero de productos]

Utilice la tabla de búsqueda de productos al escribir esta medida. Solo cuenta cuántos números de productos hay. Los números de producto y las claves de producto son lo mismo en este ejemplo. Total numero de productos =COUNT(Productos[IDProducto])

### [Total numero de clientes]

Utilice la tabla de clientes. Una vez más, solo cuenta los números de los clientes. Los números de cliente y las claves de cliente son lo mismo en este ejemplo.

Cómo te fue hasta aquí? Bueno inserte una tabla Matrix para validar los calculos, como se muestra en la siguiente imagen. Total numero de clientes =COUNT(Clientes[IDCliente])



Visualizations

Fields

Values

- Categoria
- Total numero de productos
- Total numero de clientes

Filters

Visual level filters

- Categoria is (All)
- Total numero de clientes is (All)
- Total numero de productos is (All)

Categoria	Total numero de productos	Total numero de clientes
Accesorio	35	18484
Bicicleta	125	18484
Componente	189	18484
Prenda	48	18484
<b>Total</b>	<b>397</b>	<b>18484</b>

Categoria	Total numero de productos	Total numero de clientes
Accesorio	35	18484
Bicicleta	125	18484
Componente	189	18484
Prenda	48	18484
<b>Total</b>	<b>397</b>	<b>18484</b>

## La función COUNTROWS

Vamos a conocer ahora la función, COUNTROWS (). Prefiero usar COUNTROWS () en lugar de COUNT (). Simplemente me parece más natural. Estas funciones no son exactamente las mismas, aunque a veces se pueden usar indistintamente. Si usa COUNT () con TableName [ColumnName] y a la columna le falta un número en una de las filas (por algún motivo), esa fila no se contabilizará o contará. Por el contrario COUNTROWS () cuenta cada fila en la tabla, independientemente de si todas las columnas tienen un valor en cada fila. Así que tenga cuidado y asegúrese de seleccionar la mejor fórmula para la tarea que va a realizar.

### [Total numero de productos version COUNTROWS]

Cuenta el número de productos en la tabla Productos, usando la función COUNTROWS ().  
Total numero de productos version COUNTROWS =COUNTROWS(Productos)

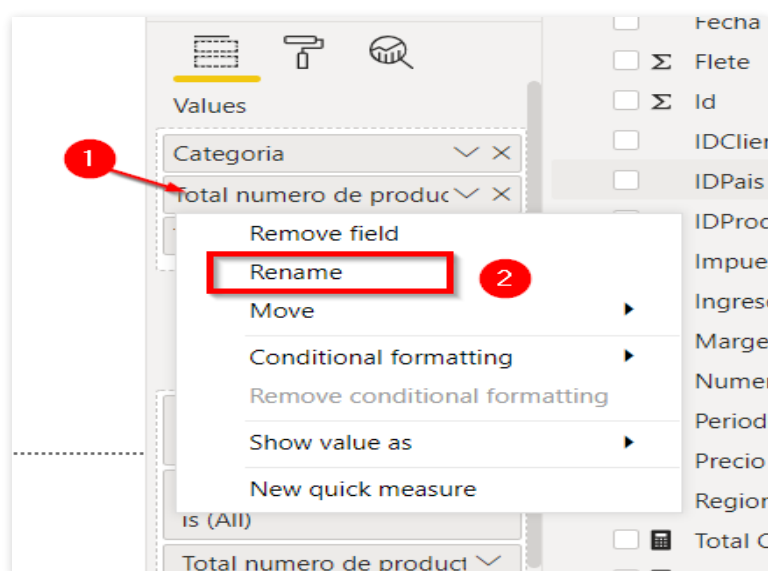
### [Total numero de clientes version COUNTROWS]

Cuenta el número de clientes en la tabla Clientes, usando la función COUNTROWS ().  
Total numero de clientes version COUNTROWS =COUNTROWS(Clientes)

Categoria	Total numero de productos version COUNTROWS	Total numero de clientes version COUNTROWS
Accesorio	35	18484
Bicicleta	125	18484
Componente	189	18484
Prenda	48	18484
<b>Total</b>	<b>397</b>	<b>18484</b>

## Importancia al nombrar medidas

Es importante usar nombre de medidas cortos y no tan descriptivos como hemos hechos en las últimas dos medidas. Sin embargo también puede hacer los nombres de las medidas descriptivos siempre que sean necesarios para dejar en claro cuáles son realmente las medidas y luego editar el nombre cuando utilice la medida en una tabla o gráfica.



## La función DISTINCTCOUNT

DISTINCTCOUNT () cuenta cada valor en una columna y solo una vez. Si un valor aparece más de una vez en una columna, todavía se cuenta una sola vez. Considere la tabla Clientes. En este caso, la clave del cliente es única y, por definición, cada clave del cliente aparece solo una vez en la tabla. (Tenga en cuenta que la clave de cliente = número de cliente). En este caso, usar DISTINCTCOUNT () con la clave de cliente en la tabla Clientes le da la misma respuesta que usar COUNTROWS () con la tabla Clientes. Pero si usara DISTINCTCOUNT () con la clave de cliente en la tabla de Ventas, en realidad estaría contando el número total único de clientes que alguna vez compraron algo, aunque se repita varias veces, lo cual no es lo mismo y es lo que hace esta función especial.

### [Total numero de clientes BaseVentas DISTINCTCOUNT]

Utilice la tabla de búsqueda de Ventas para escribir esta medida. Solo cuenta cuántos números de clientes únicos que han comprado algún producto. Total numero de clientes baseventas `DISTINCTCOUNT =DISTINCTCOUNT(Ventas[IDClientes])`

### [Total numero de clientes DISTINCTCOUNT]

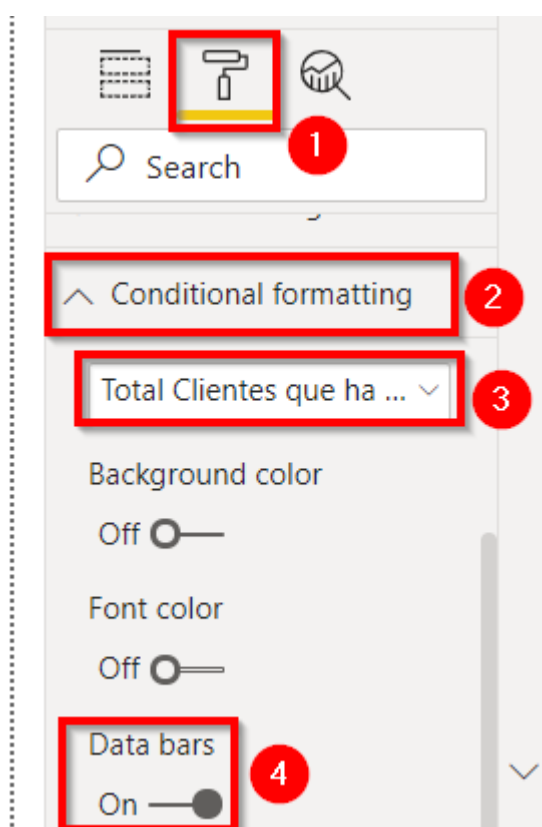
Utilice la tabla de búsqueda de Clientes para escribir esta medida. Solo cuenta el número de clientes únicos que están en la base de datos de clientes. Total numero de clientes `DISTINCTCOUNT =DISTINCTCOUNT(Clientes[IDClientes])`

Como verá, hay una diferencia en los valores, ya que en la base de datos está el registro único de todos los clientes, sin embargo en la data de ventas solo están registrados los clientes que algún momento realizaron una o varias compras, y la función DISTINCTCOUNT lo que hace es solo contarlos una vez, aunque haya realizado varias compras y así evitar la duplicidad de valores.

Categoría	Total Cantidad	Total numero clientes DISTINCTCOUNT	Total numero clientes BaseVentas DISTINCTCOUNT
Accesorio	36,003	18,484	15,080
Bicicleta	15,181	18,484	9,126
Componente		18,484	
Prenda	9,074	18,484	6,831
<b>Total</b>	<b>60,258</b>	<b>18,484</b>	<b>18,456</b>

### [Total clientes que hicieron una compra]

Crea una nueva matriz y coloca Productos [subcategoría] en Filas. Luego, utilizando DISTINCTCOUNT () en los datos de la tabla de Ventas, y cree una nueva medida llamada [Total de clientes que han comprado]. También aplique el formato condicional en la matriz y luego ordene de mayor a menor, (haciendo clic en el encabezado). A continuación, puede ver que las cubiertas y cámaras tienen la mayor cantidad de clientes que han comprado al menos una vez.



SubCategoría	Total Clientes que ha comprado
Cubierta y cámara	8,467
Bicicleta de carretera	6,395
Casco	5,947
Portabotellas y botella	4,539
Bicicleta de montaña	4,084
Jersey	3,182
Bicicleta de paseo	2,135
Gorra	2,126
Guardabarros	2,107
Guantes	1,373
Pantalones cortos	1,017
Limpiador	873
Sistema de hidratación	715
Calcetines	557
<b>Total</b>	<b>18,456</b>

## La funciones MAX(), MIN() y AVERAGE()

Estas funciones son agregadores. Toman varios valores en una columna como entrada y devuelven un solo valor a la matriz. En estos próximos ejercicios de práctica, creará nuevas medidas utilizando estos agregadores.

### [Precio Maximo]

Utilice la tabla de Ventas para escribir esta medida. Esta medida mostrará el precio Máximo de venta. Precio Promedio Maximo =MAX(Ventas[Precio])

### [Precio Minimo]

Utilice la tabla de Ventas para escribir esta medida. Esta medida mostrará el precio Mínimo de venta. Precio Promedio Maximo =MIN(Ventas[Precio])

### [Precio Promedio]

Utilice la tabla de Ventas para escribir esta medida. Esta medida mostrará el precio Promedio de venta. Precio Promedio Maximo =AVERAGE(Ventas[Precio])

Categoria	Precio Maximo	Precio Minimo	Precio Promedio
Bicicleta	3,578.27	539.99	1862.39
Prenda	69.99	8.99	37.34
Accesorio	159.00	2.29	19.42
<b>Total</b>	<b>3,578.27</b>	<b>2.29</b>	<b>486.42</b>

## La función DIVIDE()

DIVIDE () es una función simple pero poderosa que también se conoce como "división segura". DIVIDE () te protege contra los errores de división por cero en tus efectos visuales. Una matriz, por diseño, oculta las filas o columnas que no tienen datos. Si recibe un error en una medida dentro de una matriz, es posible que vea muchas filas que de otra manera no vería, y posiblemente también vea algunos mensajes de error. La función DIVIDE () está diseñada específicamente para resolver este problema. Si usa DIVIDE () en lugar del operador de barra diagonal (/) para la división, DAX devuelve un espacio en blanco donde de lo contrario obtendría un error de división por cero. Dado que una matriz filtrará las filas en blanco por defecto, una fila en blanco es una opción mucho mejor que un error.

### [Total Margen %]

Utilice las medidas [Total Margen] y [Total Ingresos] para crear esta medida. Esta medida mostrará el margen %.  $\text{Total Margen \%} = \text{DIVIDE}([\text{Margen Total}], [\text{Total Ingresos}], 0)$

Categoría	Total Ingresos	Margen Total	Margen Total %
Accesorio	699,107	437,640	62.6%
Bicicleta	28272,920	11487,524	40.6%
Prenda	338,781	136,043	40.2%
<b>Total</b>	<b>29310,809</b>	<b>12061,207</b>	<b>41.1%</b>

## 7. Tabla de Datos y Tabla de búsqueda

Todos los datos de muestra y ejercicio en este libro hasta ahora han sido preparados para usted; no ha habido nada que hacer excepto seguir las instrucciones. Pero la simplicidad de seguir mis instrucciones lo protege de un tema profundo e importante: cómo debe estructurar las tablas de datos que carga. Este capítulo cubre los diversos tipos de tablas que puede cargar en Power BI.

Este tema es fácil de repasar y descartar como trivial, pero según mi experiencia, es una de las cosas más fáciles de equivocarse, especialmente si no sabes por qué es importante o cómo hacerlo correctamente. Si se equivoca en la estructura de la tabla, todo lo demás se vuelve más difícil en órdenes de magnitud.





## Tabla de Datos

Aunque las tablas de datos no tienen que ser las tablas más grandes cargadas en Power BI, por lo general lo son. La tabla de ventas utilizada en este libro es una tabla transaccional que contiene detalles de transacciones individuales que se llevaron a cabo en los puntos de venta minoristas de AmericanAdventure en todo el mundo. Cada fila en esta tabla representa una partida individual en un recibo de registro para una transacción de compra individual.

Las tablas de datos pueden constar de millones (o incluso miles de millones) de filas de datos. Algunos ejemplos de tablas de datos incluyen Ventas, Presupuesto, inventarios, Libro mayor, Resultados de exámenes y Recuento de existencias.

No hay limitación sobre la frecuencia con la que pueden ocurrir transacciones similares y almacenarse en una tabla de datos. Considere una cadena de hamburguesas que vende hamburguesas y papas fritas. Puede haber literalmente cientos de transacciones cada día que son casi idénticas porque el mismo tipo de hamburguesa se puede vender muchas veces en un día determinado.

## Tabla de búsqueda

Las tablas de búsqueda tienden a ser más pequeñas que las tablas de datos (con menos filas) y, a menudo, pueden ser más anchas (con más columnas). Algunos ejemplos de tablas de búsqueda incluyen Clientes, Productos, Calendario y Plan de Cuentas.

Las tablas de búsqueda tienen una característica especial que las hace diferentes a las tablas de datos: una tabla de búsqueda debe tener un código de identificación único de algún tipo para diferenciar de forma única cada fila de la tabla. Este código único a menudo se denomina clave (o clave principal, en el mundo de la base de datos). Consideremos la tabla de productos utilizada en este libro. AmericanAdventure vende muchos productos diferentes, +250 productos. Cada uno de estos productos tiene un código de producto único, un número de tres dígitos que es único para ese producto. Por ejemplo, IDProducto 362 es una bicicleta mantañera 200 en color negro 44. Ningún otro producto en la tabla Producto tiene el mismo código. Si lo piensas, así es como debe ser; habría caos si una empresa utilizara el mismo código de producto para diferentes productos. Lo mismo es cierto para los clientes y los números de identificación de la tienda. De hecho, lo mismo se aplica a la tabla Calendario, dado que el campo de fecha es una ID única para cada día en el calendario.

## 8. Los iteradores básicos SUMX () y AVERAGEX ()

Las funciones cubiertas anteriormente, son todas las funciones de agregación. Cada una de esas funciones de agregación actúa en una columna o tabla completa y utiliza una técnica de agregación específica para devolver un solo valor a una celda en una visualización de un informe.

Otra clase de funciones posiblemente puede devolver las mismas respuestas que las funciones de agregación pero utilizando un enfoque diferente. Estas son "funciones X" (es decir, cualquier función que tenga una X al final del nombre) son parte de la familia llamada iteradores.

La principal diferencia entre los iteradores y las otras funciones que hemos analizado hasta ahora es que los iteradores tienen lo que se denomina contexto de fila, lo que significa que una función es "consciente" de a qué fila está haciendo referencia en cualquier momento. En lugar de entrar en una explicación teórica, pasemos a trabajar con el iterador SUMX () y hablemos sobre el contexto de las filas mientras usamos esta función.

### Usando SUMX (table, expression)

SUMX () toma dos parámetros: un nombre de tabla y una expresión para evaluar. SUMX () crea un contexto de fila en la tabla especificada y luego itera a través de cada fila de la tabla, fila por fila, y evalúa la expresión de cada fila a medida que llega antes de finalmente agregar los resultados provisionales de cada fila. El contexto de fila es un concepto en DAX que implica crear "conciencia" de la existencia de las filas en la tabla para que una función pueda recorrerlas una por una hasta que haya tocado cada fila una vez y solo una vez. Puede pensar en el contexto de la fila como una lista de verificación de todas las filas que SUMX () usa para mantener un registro de dónde está. SUMX () puede trabajar a través de las filas una a la vez, metafóricamente "marcando" cada fila para asegurarse de que no se haya perdido ninguna. Este contexto de fila solo existe en ciertas fórmulas DAX, incluidas las funciones X (analizadas en este capítulo).

Para demostrar el punto, veamos cómo escribir una nueva versión de la medida Total ingresos pero incluyendo los fletes [Total Ingresos con fletes]. Primero, cree una nueva matriz, ponga Productos [Categoría] en Filas y luego escriba esta medida: Total Ingresos con fletes = SUMX(Ventas,Ventas[Ingreso]+Ventas[Flete])

Tenga en cuenta que, en esta medida, no está envolviendo las columnas en una función de agregación. Ya que la forma en que funciona una función X es que va a la tabla especificada (en este caso, Ventas), crea un contexto de fila para que use como referencia y luego toma cada fila de la tabla, una a la vez, y evalúa la expresión para esa única fila. Una vez que ha creado un resultado provisional para cada fila en la tabla, suma todos los resultados provisionales.

## Cuándo usar funciones X vs. agregadores

Ahora sabe que puede usar funciones X como SUMX () y AVERAGEX (), y también puede usar agregadores como SUM () y AVERAGE(), y hacen cosas similares pero usando diferentes enfoques. ¿Qué debes usar? Los siguientes ejemplos te ayudarán a resolver esto. Las siguientes muestras son tablas minúsculas simplistas solo para fines de ilustración. En la vida real, las tablas de datos son, por supuesto, mucho más grandes y robustas.

Fecha	Producto	Cantidad	Precio
1-jun-19	A	2	2.7
2-jun-19	B	1	2.9
3-jun-19	C	3	3.1
4-jun-19	D	2	3.5
5-jun-19	A	4	2.7
6-jun-19	B	5	2.9
7-jun-19	C	2	3.1

Si esta es la estructura de sus datos, entonces simplemente debe usar SUMX (), así:

Ingresos totales 1 = SUMX (Ventas, Ventas [Cantidad] \* Ventas [Precio])

En este ejemplo, primero debe calcular los totales de cada fila, una fila a la vez. Esto es para lo que están diseñadas las funciones del iterador.

Pero, si sus datos contienen una sola columna con los ingresos totales, entonces, puede usar SUM () para sumar los valores: Ingresos totales 2 = SUMA (Ventas [Ingresos])

Fecha	Producto	Cantidad	Ingresos
1-jun-19	A	2	5.4
2-jun-19	B	1	2.9
3-jun-19	C	3	9.3
4-jun-19	D	2	7
5-jun-19	A	4	10.8
6-jun-19	B	5	14.5
7-jun-19	C	2	6.2

## 9. DAX CALCULATE

CALCULATE () es la función más importante y poderosa de DAX. Es la única función que tiene la habilidad para modificar el contexto de filtro que viene de tus visuales.

**Nota:** En realidad, hay otra función que puede modificar el contexto del filtro: CALCULATETABLE (). Esta función se usa normalmente dentro de las consultas de DAX, aunque su discusión está fuera del alcance de este libro.

Voy a proporcionarle una sólida comprensión de cómo funciona CALCULATE () en este libro, pero tendrá que seguir aprendiendo en el futuro. Hay mucho que aprender. Si quieres ser un experto, necesitas practicar, practicar y practicar mucho, además de incluir lectura de otros libros y blogs para construir sobre la base que obtienes de este libro.

Por ejemplo, cuando ordena una ensalada, usted modifica la opción de menú estándar y en su lugar dice: "Voy a pedir la ensalada César, sin anchoas". CALCULATE () es muy parecido a eso: le permite modificar la oferta estándar (que recibe desde un visual en lugar de un menú) para que pueda obtener alguna variación que termine siendo exactamente lo que desea.

Técnicamente hablando, CALCULATE () altera el contexto del filtro. Modifica una expresión (que puede ser una medida u otra fórmula DAX) aplicando / eliminando / modificando filtros. La sintaxis de CALCULATE () es: = CALCULATE (expresión, filtro 1, filtro 2, filtro n ...)

CALCULATE () altera el contexto del filtro que proviene de lo visual aplicando ninguno, uno o más filtros antes de evaluar la expresión. CALCULATE () "vuelve a ejecutar" el motor de filtro integrado en Power BI, el que hace que los filtros se propaguen automáticamente desde las tablas de búsqueda y fluyan cuesta abajo hacia las tablas de datos. Cuando CALCULATE () vuelve a ejecutar el motor del filtro, si hay algún filtro dentro de la función CALCULATE (), estos filtros se convierten en parte del contexto del filtro antes de que el motor del filtro se active.

### Filtros simples

CALCULATE () puede usar dos tipos de filtros. Un filtro simple (o filtro crudo) tiene un nombre de columna a la izquierda y un valor a la derecha, como en estos ejemplos: Clientes [Sexo] = "F", Productos [Color] = "Azul", Fecha [Año] = 2017

Puede usar estos filtros simples como segundo y subsiguientes parámetros para CALCULATE () para alterar el significado original de una expresión (que es el primer parámetro). Los filtros simples son realmente importantes en Power BI porque fueron diseñados para ser fáciles de usar y comprender.

Cree las siguientes medidas:

Ingresos totales de productos azules. = CALCULATE ([Total Ingresos], Productos [Color] = "Azul")

Ingresos totales de mujeres. = CALCULATE ([Total Ingresos], Clientes [Sexo] = "F")

En la imagen de abajo, ¿puede ver cómo el filtro simple usado aquí, Productos [Color] = "Azul", y la otra medida Clientes [Sexo] = "F" ha alterado el contexto del filtro inicial proveniente de la matriz y ha dado una variación a la medida regular [Total Ingresos]? Es como si hubiera cambiado la receta para el producto estándar en el menú y en su lugar recibió una variación de ese elemento del menú regular. Piensa en la ensalada César sin anchoas.

Categoría	Total Ingresos	Ingresos total de productos azules	Ingresos total a mujeres
Accesorio	699,107	74,249	348,636
Bicicleta	28272,920	2159,946	14267,298
Prenda	338,781	35,560	166,443
<b>Total</b>	<b>29310,809</b>	<b>2269,755</b>	<b>14782,377</b>

### [Total Ingresos clientes hombres]

Utilice la medida [Total Ingresos] y la tabla clientes [Sexo] Masculino. Total Ingresos clientes hombre =CALCULATE([Total Ingresos], Clientes[Sexo] = "M")

### [Total Ingresos año 2017]

Utilice la medida [Total Ingresos] y la tabla Fecha [Año] 2017. Total Ingresos año 2017 =CALCULATE([Total Ingresos], Fecha[Año] = 2017)[Precio Promedio]

### [Total Ingresos Sur America]

Utilice la medida [Total Ingresos] y la tabla Países [Region] Sur America. Total Ingresos Sur America =CALCULATE([Total Ingresos], Países[Region] = "Sur America")

## 10. DAX: IF y SWITCH

DAX tiene una serie de funciones útiles que le permiten aplicar una prueba y luego ramificar la fórmula en función de los resultados de esa prueba. Lo más probable es que esté familiarizado con este concepto de la función IF () en Excel.

### La función IF ()

La función IF () en DAX es casi idéntica a IF () en Excel:

IF (Prueba lógica, Resultado si es verdadero, [Resultado si es falso])

Tenga en cuenta que el último parámetro, [Resultado si es falso], es opcional. Si omite este parámetro y el resultado es FALSO, la fórmula IF () devuelve BLANK (). Esto es muy útil porque una matriz o gráfico en Power BI no mostrará valores si el resultado en la sección de Valores es BLANK ().

En la tabla Cliente, elija New Column, para ingresar una nueva columna en esa tabla y luego Ingrese la formula siguiente:

ColumnSexo = IF(Cliente[Sexo]="M", "Hombre", "Mujer")

### La función SWITCH ()

La sintaxis de una función SWITCH ()

es como sigue:

SWITCH(expresión, valor, resultado [, valor, resultado] ... [, else])

Esta sintaxis es un poco confusa, así que veamos un ejemplo simple con otra columna calculada. Haga clic con el botón derecho en la tabla Clientes, seleccione Nueva columna e ingrese la siguiente fórmula:

ColumnEstadoCivil = SWITCH (Clientes [EstadoCivil], "M", "Casado",  
"S", "Soltero", "Sin Información")

Los valores posibles en la columna EstadoCivil son M y S en esta instancia. Por lo tanto, si el valor en EstadoCivil es M, entonces se devuelve "Casado" y si el resultado es S, se devuelve "Soltero", pero si no hay ninguno de estos valores, entonces se devuelve "Sin Información".

1 ColumnEstadoCivil = SWITCH(Clientes[EstadoCivil], "M", "Casado", "S", "Soltero", "Sin Información")

IDCliente	Nombre	FechaNacimiento	EstadoCivil	Sexo	Direccion	FechaPrimeraCompra	ColumnEstadoCivil
14691	Derrick Torres	11/13/1979 12:00:00 AM	M	M	4195, rue Léo Delibes	3/5/2018 12:00:00 AM	Casado
14997	Angel Cooper	6/5/1975 12:00:00 AM	M	M	Brunnenstr 6466	6/8/2016 12:00:00 AM	Casado
16454	Marco Suri	6/5/1977 12:00:00 AM	M	M	8510 G St.	8/16/2017 12:00:00 AM	Casado
18039	Logan Clark	10/19/1975 12:00:00 AM	M	M	40, rue Royale	3/7/2016 12:00:00 AM	Casado
19512	Marco Kapoor	7/11/1976 12:00:00 AM	M	M	2893 Vera	10/12/2017 12:00:00 AM	Casado
24383	Ronald Prasad	6/25/1974 12:00:00 AM	M	M	12, avenue Foch	2/15/2018 12:00:00 AM	Casado

1 ColumnSexo = IF(Clientes[Sexo]="M", "Hombre", "Mujer")

IDCliente	Nombre	FechaNacimiento	EstadoCivil	Sexo	Direccion	FechaPrimeraCompra	ColumnEstadoCivil	ColumnSexo
14691	Derrick Torres	11/13/1979 12:00:00 AM	M	M	4195, rue Léo Delibes	3/5/2018 12:00:00 AM	Casado	Hombre
14997	Angel Cooper	6/5/1975 12:00:00 AM	M	M	Brunnenstr 6466	6/8/2016 12:00:00 AM	Casado	Hombre
16454	Marco Suri	6/5/1977 12:00:00 AM	M	M	8510 G St.	8/16/2017 12:00:00 AM	Casado	Hombre
18039	Logan Clark	10/19/1975 12:00:00 AM	M	M	40, rue Royale	3/7/2016 12:00:00 AM	Casado	Hombre

## 11. DAX: VALUES, HASONEVALUE y SELECTVALUE

En el Capítulo de CALCULATE, hablamos de los filtros simples, CALCULATE puede usar dos tipos de filtros: filtros simples y filtros avanzados. Los filtros simples están en esta forma:

TablaNombre [ColumnaNombre = algún valor

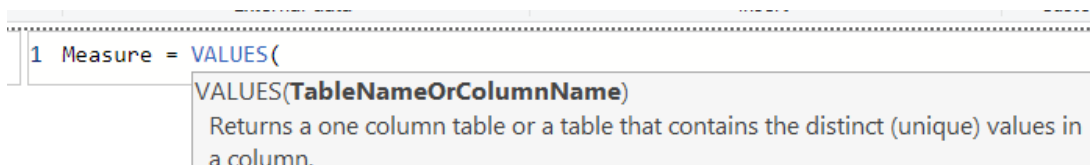
Por otro lado, un filtro avanzado toma una tabla como entrada de filtro. En otras palabras, utiliza una tabla existente (o crea una tabla virtual sobre la marcha) que contiene las filas que desea incluir en el filtro, y CALCULATE () aplica ese filtro al modelo de datos antes de completar la evaluación de la principal expresión.

### Creando tablas virtuales

Se puede considerar que las tablas que crea con las funciones son "virtuales" porque no se almacenan físicamente como parte del modelo de datos. Se crean sobre la marcha dentro de sus fórmulas DAX y se pueden usar durante la evaluación de solo las fórmulas específicas que contienen la tabla virtual. Es importante destacar que, al crear una tabla virtual con una fórmula, la nueva tabla virtual tendrá una relación virtual con el modelo de datos, y esa relación virtual propagará el contexto del filtro exactamente de la misma manera que las relaciones permanentes. (Aprenderá más sobre esto más adelante en el capítulo). Se dice que las tablas virtuales retienen el linaje con sus tablas de origen.

### La función de VALUES ()

VALUES () es la primera función a la que ha llegado en este libro que devuelve una tabla (virtual). Si escribe la palabra VALORES en la barra de fórmulas y lee la información sobre herramientas, puede ver que esta función devuelve una tabla, como se muestra a continuación.



Una cosa importante a tener en cuenta acerca de VALUES () es que respeta el contexto del filtro inicial que proviene de sus imágenes. Entonces, si combina este hecho de que VALUES () respeta el contexto del filtro actual con la información provista por IntelliSense en la imagen de arriba, verá que VALUES () devuelve una tabla de una sola columna que contiene la lista de todos los valores posibles en la versión actual. filtro de contexto.

Vamos a trabajar algunos ejemplos para demostrar el punto.

## Un ejemplo de calendario

Configure una nueva matriz como se muestra a continuación y ponga el año calendario en filas. A continuación, escribir

La siguiente medida en la tabla de calendario:

Total numero de meses = COUNTROWS(VALUES(Calendario[MesNombre]))

IDAño
2015
2016
2017
2018

Cuando coloca 'Calendario' [MesNombre] dentro de la función VALUES (), esta única columna que forma parte de la tabla del Calendario se convierte en una tabla por derecho propio y conserva una relación (linaje) con la tabla del Calendario original. Esta nueva tabla devuelta por VALUES () sigue siendo una sola columna, pero ahora técnicamente es una tabla y una columna en lugar de ser simplemente una columna de alguna otra tabla (Calendario, en este caso).

Entonces, VALUES ('Calendario' [MesNombre]) devuelve una tabla de una sola columna de valores posibles que respeta el contexto del filtro inicial que proviene de la matriz. No es posible colocar esta nueva tabla creada por VALUES () en una medida a menos que la incluya dentro de alguna otra fórmula (por ejemplo, un agregador). En el ejemplo anterior, primero crea la tabla (la parte de VALORES de la fórmula) y luego cuenta cuántas filas hay en la tabla usando la función COUNTROWS ():

Total numero de meses = COUNTROWS(VALUES(Calendario[MesNombre]))

IDAño	Total numero de meses
2015	6
2016	12
2017	12
2018	12
<b>Total</b>	<b>12</b>

Observe que el año 2015 tiene solo 6 meses, y los otros años tienen 12. Esta es una prueba de que la función VALUES () respeta el contexto del filtro inicial de la matriz. El contexto del filtro inicial para la primera fila en la matriz es 'Calendario' [IDAño] = 2015. Ese filtro se aplica antes de que se calcule la fórmula [Total numero de meses]. VALUES () toma esta tabla "prefiltrada" (solo las fechas donde no se filtra el año = 2015) y devuelve una tabla de una sola columna que contiene una lista distinta de los valores posibles, es decir la cantidad de meses que tiene cada año.



**HASONEVALUE()**

La función HASONEVALUE devuelve TRUE cuando el contexto ha reducido el número de valores (es decir elementos distintos) de la columna indicada como argumento a un único valor, y devuelve FALSE cuando el número de valores de la columna es mayor que uno o presenta dos o mas valores.

Ahora bien, escribe la siguiente fórmula, debe proporcionar protección para el otro escenario posible, donde VALUES ('Calendario' [MesNombre]) tiene más de una fila en la tabla. Esto se hace usando la función HASONEVALUE (), así:

MedidaNombreMes =

IF(HASONEVALUE(Calendario[MesNombre]),VALUES(Calendario[MesNombre]))

MesNombre	Total numero de meses	MedidaNombreMes
abr	1	abr
ago	1	ago
dic	1	dic
ene	1	ene
feb	1	feb
jul	1	jul
jun	1	jun
mar	1	mar
may	1	may
nov	1	nov
oct	1	oct
sep	1	sep
<b>Total</b>	<b>12</b>	

Recuerde que la estructura de una instrucción IF () es la siguiente: = IF (Logical Test, Result if True, [Result if False]). El último parámetro es opcional. Si lo deja fuera, está aceptando el valor predeterminado, EN BLANCO ().

Pero, si escribe la fórmula anterior sin la función HASONEVALUE (), arrojará un error. Incluso si elimina el total general de la matriz, aún arrojará un error. DAX le permite usar el valor único devuelto en una sola fila de la tabla de una sola columna solo si protege la fórmula con HASONEVALUE ().

**SELECTEDVALUE()**

La función SELECTEDVALUE devuelve el valor de la referencia columna filtrada como primer argumento, si es el único valor habilitado o disponible en el contexto de filtro; de lo contrario, devuelve un valor en blanco o el valor predeterminado pasado como segundo argumento. En resumen esta función simplifica la sintaxis de requerida cuando necesita leer un solo valor seleccionando el contexto de filtro. La sintaxis es: = SELECTEDVALUE(ColumnName, AlternateResult).

En definitiva, SELECTEDVALUE () se creó para reemplazar la fórmula compleja en la sección anterior. Aquí está la fórmula

desde arriba: MedidaNombreMes =

IF(HASONEVALUE(Calendario[MesNombre]),VALUES(Calendario[MesNombre]))

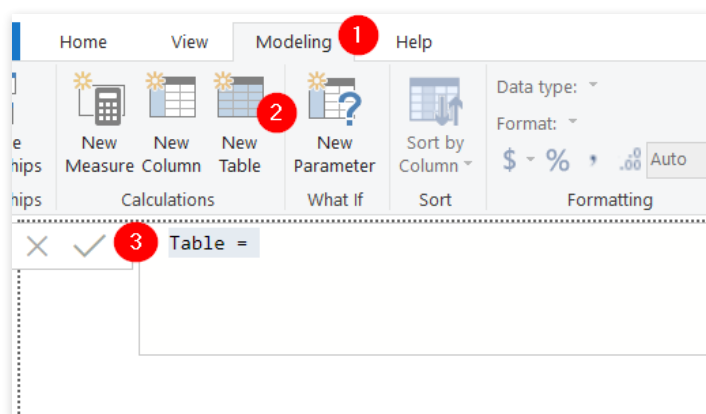
Con la función SELECTEDVALUE () usted puede reescribir la misma fórmula de la siguiente manera: MedidaNombreMes2 = SELECTEDVALUE(Calendario[MesNombre])

MesNombre	Total numero de meses	MedidaNombreMes	MedidaNombreMes2
abr	1	abr	abr
ago	1	ago	ago
dic	1	dic	dic
ene	1	ene	ene
feb	1	feb	feb
jul	1	jul	jul
jun	1	jun	jun
mar	1	mar	mar
may	1	may	may
nov	1	nov	nov
oct	1	oct	oct
sep	1	sep	sep
<b>Total</b>	<b>12</b>		

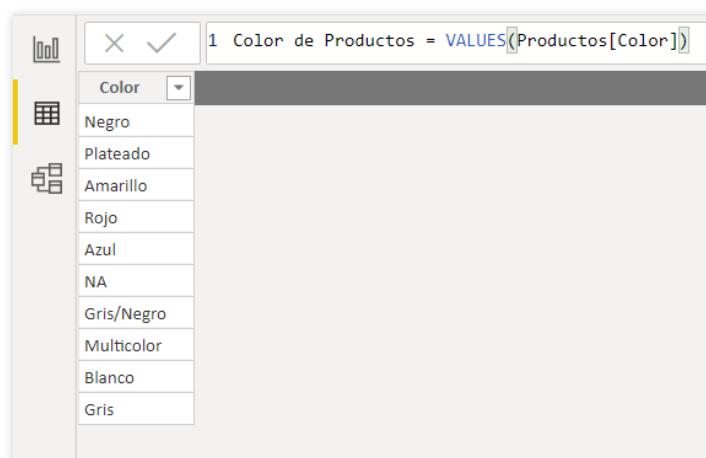
## La opción de Nueva Tabla

Los usuarios de negocios (es decir, aquellos que no tienen experiencia profesional en TI) a menudo tienen dificultades para comprender la función VALUES () porque en realidad no se puede "ver" la tabla. Hasta ahora, le he demostrado que, aunque no puede ver la tabla, puede envolverla o anidarla dentro de la función COUNTROWS (), por lo que al menos puede ver el número de filas en la tabla.

Ahora bien, Power BI Desktop tiene una característica llamada Nueva tabla que actualmente no está disponible en Power Pivot para Excel.



Cuando hace clic en este botón (# 2), ve que puede escribir una fórmula en la barra de fórmulas (# 3). Cuando crea una nueva tabla con este botón, puede escribir una fórmula que devuelva una tabla (por ejemplo, Color de Productos =VALUES(Productos[Color])), y la tabla se agrega al modelo de datos y obtiene la nueva tabla que se muestra a continuación en el modelo de datos (visible en la vista de datos)



## 12. ALL(), ALLEXCEPT() y ALLSELECTED()

Las funciones DAX ALL (), ALLEXCEPT () y ALLSELECTED () son muy similares en lo que hacen. Vamos a comenzar con ALL () y luego veremos las otras dos variantes.

### La función ALL ()

La función ALL () elimina todos los filtros actuales del contexto de filtro actual. Por esta razón, ALL () puede ser considerada la función "eliminar filtros". La forma más fácil de entender esto es con un ejemplo.

Cree una nueva matriz y ponga Productos [Categoría] en Filas y ponga la medida [Total numero de productos] , medida que creó anteriormente en Valores. Obtiene la matriz que se muestra a continuación.

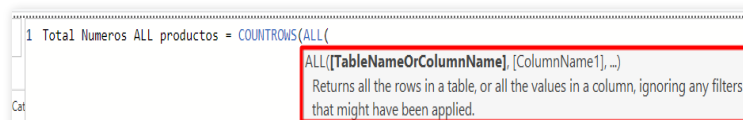
Categoría	Total numero de productos
Accesorio	35
Bicicleta	125
Componente	189
Prenda	48
<b>Total</b>	<b>397</b>

Técnicamente, lo que está sucediendo arriba es que la primera fila de la matriz está filtrando la tabla Productos para que solo los productos que sean del tipo Productos [Categoría] = "Accesorios" estén visibles en la tabla subyacente; de igual forma para bicicletas, componentes y prendas. En el ejemplo podemos apreciar la cantidad de productos que tiene cada categoría. Después de que la matriz aplica un filtro a las tablas subyacentes, la medida [Número total de productos] cuenta las filas que sobreviven al filtro. Hace esto para cada celda de la matriz, una a la vez, incluida la celda total. No hay ningún filtro aplicado a la celda total, por lo que la medida cuenta todas las filas de la tabla.

Ahora cree una nueva medida que use la función ALL ():

Total de todos los productos = COUNTROWS (ALL(Productos))

La función ALL () devuelve una tabla. No puede ver la tabla, pero puede ajustar COUNTROWS () a su alrededor para que pueda ver cuántas filas hay en la tabla. A medida que escribe esta fórmula, si hace una pausa mientras escribe, IntelliSense muestra la sintaxis de ALL (), como se muestra a continuación.



Categoría	Total numero de productos	Total Numeros ALL productos
Accesorio	35	397
Bicicleta	125	397
Componente	189	397
Prenda	48	397
<b>Total</b>	<b>397</b>	<b>397</b>

Puede ver en esta matriz que la nueva medida (la de la derecha) ignora el contexto de filtro inicial que proviene de las filas de la matriz. Lo que sucede aquí es que, en primer lugar, el contexto de filtro inicial lo establece la fila Productos [Categoría], pero la función ALL () siempre devuelve una copia sin filtrar de la tabla y, por lo tanto, devuelve la tabla completa de Productos en lugar de la Tabla de productos filtrados. Entonces COUNTROWS () devuelve 397 por cada fila de la matriz, incluido el total.

### Usar ALL () como entrada de filtro avanzada

El uso más común de ALL () es como una entrada de tabla de filtro avanzada a una función CALCULATE (). Hagamos un ejemplo usando ALL () como una entrada de tabla para CALCULATE ().

Un buen uso de ALL () dentro de CALCULATE () es eliminar los filtros que se aplican naturalmente a una visualización para que pueda acceder al número que normalmente está en la línea total de la visualización. Una vez que puede acceder al total equivalente en un visual (como una matriz) desde cualquier fila de la matriz, puede crear fácilmente una medida que encuentre el porcentaje del total, lo que sería muy útil. El concepto tendrá más sentido a medida que trabaje en el siguiente ejemplo.

### Cálculo del porcentaje de ventas totales por país

Digamos que desea calcular el porcentaje de las ventas por país del total de las ventas. En primer lugar, configure una matriz con Territorios [País] en Filas y [Ventas totales] en Valores, como se muestra a continuación.

País	Total Ingresos
Argentina	5705,927
Brasil	2890,907
Canada	3649,475
Chile	12,239
Colombia	2640,882
Costa Rica	3379,702
Estados Unidos	6,532
Mexico	3,001
Panama	9047,019
Peru	1975,126
<b>Total</b>	<b>29310,809</b>

### Paso 1: crear una medida de gran total

Haga clic en New Measure para crear la siguiente medida nueva:

Total Ventas globales = CALCULATE([Total Ingresos], ALL(Paises))

Como sabe, el primer parámetro de CALCULATE () es una expresión, y los parámetros siguientes son filtros que modifican el contexto del filtro. En este caso, está pasando una tabla como contexto de filtro. Esta tabla es ALL (Paises), que en realidad es una copia sin filtrar de toda la tabla Paises.

Después de agregar la nueva medida a la matriz, su matriz se ve como se muestra a continuación. ¿Ves que la nueva medida ignora el contexto de filtro inicial que proviene de la matriz? CALCULATE () es la única función que puede modificar el contexto del filtro. En este caso, CALCULATE () está reemplazando el contexto de filtro inicial en Paises [País] con un nuevo contexto de filtro (una copia sin filtrar de la tabla Paises).

Pais	Total Ingresos	Total Ventas globales
Argentina	5705,927	29310,809
Brasil	2890,907	29310,809
Canada	3649,475	29310,809
Chile	12,239	29310,809
Colombia	2640,882	29310,809
Costa Rica	3379,702	29310,809
Estados Unidos	6,532	29310,809
Guatemala		29310,809
Mexico	3,001	29310,809
Panama	9047,019	29310,809
Peru	1975,126	29310,809
<b>Total</b>	<b>29310,809</b>	<b>29310,809</b>

### Paso 2: crear el porcentaje del total

Después de haber creado la medida [Total Ventas globales], es fácil crear una nueva medida para calcular el porcentaje de ventas globales del país, de la siguiente manera:

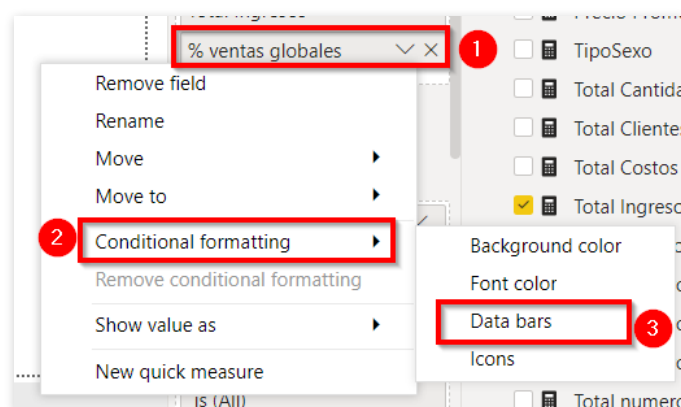
% de ventas globales

= DIVIDIR ([Total Ingresos], [Total Ventas globales ])

Asegúrese de formatear esta medida para que el Formato esté configurado en Porcentaje y con decimales en 1. Usted debe terminar con la matriz que se muestra a continuación.

Pais	Total Ingresos	Total Ventas globales	% ventas globales
Argentina	5705,927	29310,809	19.5%
Brasil	2890,907	29310,809	9.9%
Canada	3649,475	29310,809	12.5%
Chile	12,239	29310,809	0.0%
Colombia	2640,882	29310,809	9.0%
Costa Rica	3379,702	29310,809	11.5%
Estados Unidos	6,532	29310,809	0.0%
Guatemala		29310,809	
Mexico	3,001	29310,809	0.0%
Panama	9047,019	29310,809	30.9%
Peru	1975,126	29310,809	6.7%
<b>Total</b>	<b>29310,809</b>	<b>29310,809</b>	<b>100.0%</b>

Como último paso puede quitar la medida [Total Ventas globales] de la tabla matriz. También puede incluir un formato condicional de barra o lo que desee realizar para tener una visualización gráfica que facilite el análisis de los datos.



Pais	Total Ingresos	% ventas globales
Argentina	5705,927	19.5%
Brasil	2890,907	9.9%
Canada	3649,475	12.5%
Chile	12,239	0.0%
Colombia	2640,882	9.0%
Costa Rica	3379,702	11.5%
Estados Unidos	6,532	0.0%
Mexico	3,001	0.0%
Panama	9047,019	30.9%
Peru	1975,126	6.7%
<b>Total</b>	<b>29310,809</b>	<b>100.0%</b>

## La función ALLEXCEPT()

ALLEXCEPT () le permite eliminar filtros de todas las columnas de una tabla, excepto los filtros que se han aplicado a las columnas especificadas. Considere el siguiente ejemplo:

Total ingresos a región o país

= CALCULATE ([Total Ingresos], ALLEXCEPT (Países, Países [País]))

El primer argumento de ALLEXCEPT debe ser una referencia a una tabla base; el resto de los argumentos deben ser referencias a columnas de la tabla base. Para este ejemplo la tabla es Países y la columna "País". Esta función no se usa por sí misma, pero sirve como una función intermedia que se puede usar para cambiar el conjunto de resultados sobre los cuales se realiza algún otro cálculo.



País	Total Ingresos	Total Ventas globales	Total Ingresos region
Argentina	5705,927	29310,809	13225,081
Brasil	2890,907	29310,809	13225,081
Chile	12,239	29310,809	13225,081
Colombia	2640,882	29310,809	13225,081
Peru	1975,126	29310,809	13225,081
<b>Total</b>	<b>13225,081</b>	<b>29310,809</b>	<b>13225,081</b>

Como puedes apreciar en la imagen de arriba hemos excluido del filtro con ALLEXCEPT(), el filtro por region, es decir que al utilizar una segmentación de datos se puede visualizar la información por región de la medida creada Total Ingresos región, sin embargo con la medida Total ventas globales al filtrar presenta toda la información ignorando cualquier tipo de filtro o segmentación que se haga en una visualización utilizando la función ALL().



## La función ALLSELECTED ()

La función ALLSELECTED es útil cuando desea calcular porcentajes, como se muestra arriba en el ejemplo utilizado con ALL(), y tiene un filtro aplicado (digamos, a través de una segmentación) pero desea que el total en su matriz sume 100%.

Ahora bien, digamos que está trabajando con la misma matriz utilizada anteriormente en este capítulo, pero ahora con una segmentación de datos que se filtra en Países[Region]. Observe a continuación que [% de ventas globales] suma hasta 45.1%; esto es correcto porque los otros países que conforman el 54.9% restante han sido filtrados por la segmentación de región.

Region	Pais	Total Ingresos	% ventas globales
<input type="checkbox"/> Centro America	Argentina	5705,927	19.5%
<input type="checkbox"/> Norte America	Brasil	2890,907	9.9%
<input checked="" type="checkbox"/> Sur America	Chile	12,239	0.0%
	Colombia	2640,882	9.0%
	Peru	1975,126	6.7%
	<b>Total</b>	<b>13225,081</b>	<b>45.1%</b>

Pero supongamos que desea ver el porcentaje de cada región de todos los valores de la matriz (en este ejemplo, solo las regiones del grupo Sur America). Aquí es donde entra ALLSELECTED (). ALLSELECTED () elimina los filtros de la matriz pero respeta los filtros de la segmentación.

Agregue la siguiente medida a la matriz anterior:

Total seleccion paises

= CALCULAR ([Total Ingresos], ALLSELECTED (Países))

Region	Pais	Total Ingresos	% ventas globales	Total seleccion paises
<input type="checkbox"/> Centro America	Argentina	5705,927	19.5%	13225,081
<input type="checkbox"/> Norte America	Brasil	2890,907	9.9%	13225,081
<input checked="" type="checkbox"/> Sur America	Chile	12,239	0.0%	13225,081
	Colombia	2640,882	9.0%	13225,081
	Peru	1975,126	6.7%	13225,081
	<b>Total</b>	<b>13225,081</b>	<b>45.1%</b>	<b>13225,081</b>

Observe cómo la medida creada [Total seleccion paises] devuelve el mismo valor que el total de los elementos en la matriz. Con los mismos pasos que antes, ahora puede escribir una nueva medida [% seleccion paises] y luego quitar la medida provisional [Total seleccion paises] de la tabla matriz.

Ahora escribe la siguiente medida:

% seleccion paises = DIVIDIR ([Total Ingresos], [Total seleccion paises])

Recuerde formatear esta nueva medida usando porcentaje y un decimal.

Region	Pais	Total Ingresos	% ventas globales	% seleccion paises
<input type="checkbox"/> Centro America	Argentina	5705,927	19.5%	43.1%
<input type="checkbox"/> Norte America	Brasil	2890,907	9.9%	21.9%
<input checked="" type="checkbox"/> Sur America	Chile	12,239	0.0%	0.1%
	Colombia	2640,882	9.0%	20.0%
	Peru	1975,126	6.7%	14.9%
	<b>Total</b>	<b>13225,081</b>	<b>45.1%</b>	<b>100.0%</b>

## Realizar los siguientes ejemplos prácticos

Es hora de practicar un poco. Cree una nueva matriz y ponga Productos [Categoria] en Filas y la medida [Ingresos Totales] en Valores. Obtenga la matriz que se muestra a continuación.

Categoria	Total Ingresos
Accesorio	699,107
Bicicleta	28272,920
Prenda	338,781
<b>Total</b>	<b>29310,809</b>

## Realizar las siguientes medidas

[Total Ingresos ALL Productos]

[% Total Ingresos ALL Productos]

[Total Ingresos seleccion Productos]

[% Total Ingresos seleccion Productos]

### 13. La función FILTER()

FILTER () es una función muy poderosa en DAX. Cuando FILTER () y CALCULATE () se combinan, estas dos funciones le permiten alterar el contexto del filtro en sus matrices de la forma que desee. Sin embargo, antes de pasar a usar FILTER () con CALCULATE (), creo que vale la pena mirar FILTER () solo con un par de ejemplos simples.

La sintaxis de FILTER () es la siguiente:

= FILTER (Tabla, myFilter)

Tabla es cualquier tabla (o función que devuelve una tabla, como ALL ()), y myFilter es cualquier expresión que se evalúa como una respuesta VERDADERO / FALSO.

La función FILTER () devuelve una tabla que contiene cero o más filas de la tabla original. Dicho de otra manera, la tabla devuelta por FILTER () puede contener cero filas, una fila, dos filas o cualquier otro número de filas, hasta incluir el número total de filas en la tabla original. El propósito de FILTER (), por lo tanto, es determinar qué filas se devolverán al resultado de la tabla final después de usar la prueba myFilter.

FILTER () es un iterador y, por lo tanto, puede completar un análisis granular para determinar qué filas se incluirán en la tabla final. En general, está bien usar FILTER () sobre tablas de búsqueda, pero es algo más arriesgado usarlo sobre tablas de datos, particularmente si son muy grandes (millones de filas). Si debe usar FILTER () depende de sus datos, de la calidad de sus fórmulas DAX dentro de FILTER () y de lo que necesita lograr.

Analicemos un ejemplo. Configure una nueva matriz con Ventas[Region] en Filas y la medida [Total Clientes que ha comprado] en Valores. La matriz, que se muestra a continuación, indica cuántos clientes hay en toda la base de datos de clientes para cada tipo de sexo.

Region	Total Clientes que ha comprado
Centro America	5,498
Norte America	3,353
Sur America	9,605
<b>Total</b>	<b>18,456</b>

Pero, ¿qué sucede si desea saber cuántas regiones en la base de datos tienen una facturación por más de 10,000 por año? Considere la siguiente fórmula:  
 = FILTER (Países, Países, [Total Ingresos]> = 10000)

El resultado de esta fórmula es una tabla de ventas, y esta nueva tabla virtual de Ventas incluye a todas las regiones que tienen un ingreso mayor o igual a \$ 10,000 por año. Pero tenga en cuenta que es una tabla y no puede poner una tabla de valores en una matriz. Entonces, si desea ver el resultado (en este caso, el recuento total de filas) dentro de una matriz, debe ajustar esta tabla que fue devuelta por FILTER () dentro de una función que devuelve un valor en lugar de una tabla de valores ( como un agregador).

Es posible sumar el número de filas en esta tabla envolviendo la fórmula anterior dentro de otra fórmula, como esta:

Regiones totales ingresos \$ 10,000 o más

=COUNTROWS(FILTER (Países, [Total Ingresos]> = 10000))

Region	Total Clientes que ha comprado	Regiones total ingresos 10,000 o mas
Centro America	5,498	2
Norte America	3,353	1
Sur America	9,605	5
<b>Total</b>	<b>18,456</b>	<b>8</b>

La medida [Regiones total ingresos de 10,000 o más] itera a través de la tabla Países usando la función FILTER (), verificando el valor de la medida [Total Ingresos] de cada cliente individual para ver si el valor es mayor o igual a \$ 10,000 . FILTER () devuelve una tabla de todos los clientes que pasaron esta prueba, y luego COUNTROWS () los cuenta.

Como se muestra a continuación, la fórmula determina que la región Centro America son 2, que la región Norte America 1 y que la región Sur America 5 clientes pasan esta prueba, y 8 clientes en total pasan la prueba.

Analicemos otro ejemplo: Ahora en este caso deseamos ahora saber cuanto es el ingresos por producto con una facturación menor a 10,000 y cuales con una facturación mayor a 10,000.

La formula es la siguiente:

ingresos menores a 10000 = CALCULATE([Total Ingresos],FILTER(Productos,[Total Ingresos]<10000))

ingresos mayores a 10000 = CALCULATE([Total Ingresos],FILTER(Productos,[Total Ingresos]>10000))

Podemos ver en la tabla el siguiente resultado. Como puedes apreciar la suma de ambas medida da como resultado el total de los ingresos.

Categoria	Total Ingresos	ingresos menores a 10000	ingresos mayores a 10000
Accesorio	214,796	81,413	133,383
Bicicleta	12109,744	124,495	11985,249
Prenda	102,181	81,394	20,786
<b>Total</b>	<b>12426,720</b>	<b>287,302</b>	<b>12139,418</b>

## 14. Inteligencia de Tiempo (Time Intelligence)

La inteligencia de tiempo es una característica muy importante y poderosa en DAX. La inteligencia de tiempo se refiere a la capacidad de escribir fórmulas que se refieren a otros períodos de tiempo dentro de una visualización sin necesidad de cambiar los filtros de tiempo.

Las funciones de inteligencia de tiempo permite satisfacer las necesidades del análisis de Business Intelligence o Inteligencia de Negocio al facilitarle manipular datos utilizando períodos de tiempo, como: días, meses, trimestres y años, y luego construir y comparar cálculos durante esos períodos.

Puede usar las funciones de inteligencia de tiempo para crear nuevas medidas relativas, como [Ventas totales el año pasado], es decir que con DAX usted puede dentro de una visualización del periodo seleccionado compararse con el periodo del año anterior sin tener ninguno tipo de complicación al utilizar las funciones de inteligencia de tiempo.

**Es muy importante** saber que cuando utilizamos funciones de inteligencia de tiempo se recomienda utilizar una tabla de fecha o master de fecha para que funcionen de forma efectiva.

Hagamos el siguiente ejemplo: Crear una tabla que se visualice los ingresos actuales y año anterior, como también su comparación. Cree las siguientes medidas

Ingresos Periodo anterior

= CALCULATE([Total Ingresos],SAMEPERIODLASTYEAR(Calendario[Fecha]))

Desv Act vs AAnt. = [Total Ingresos]-[Ingresos Periodo anterior]

IDAño	Categoria	Total Ingresos	Ingresos Periodo anterior	Desv Act vs AAnt.
<input type="checkbox"/> 2015	Accesorio	124,266	90,530	33,735
<input type="checkbox"/> 2016	Bicicleta	3564,365	4198,869	-634,504
<input type="checkbox"/> 2017	Prenda	59,547	42,634	16,913
<input checked="" type="checkbox"/> 2018	<b>Total</b>	<b>3748,178</b>	<b>4332,033</b>	<b>-583,855</b>

IDAño	Total Ingresos
2015	1600,638
2016	2745,872
2017	4332,033
2018	3748,178
<b>Total</b>	<b>12426,720</b>

Como puedes ver en las imagenes de arriba, en la primera imagen se visualiza el análisis comparativo de los ingresos por Categoría, un filtro por año, donde se ha seleccionado el año 2018 que corresponde al año actual, y en la tabla puedes visualizar los ingresos por categoria del año actual (2018) y los ingresos del año anterior (2017) y su comparativo, con solo seleccionar el actual 2018.

Si en el filtro se selecciona el año 2017, entonces automaticamente se va a visualizar en la medida año anterior la información del 2016. Este es el gran valor que ofrece las funciones de inteligencia de tiempo DAX.

## Calculo acumulado del periodo

Una necesidad comercial y del area financiera muy común es calcular las cifras de forma anual o acumulado a la fecha (YTD). Afortunadamente DAX tiene una función incorporada para esto, y es la función TOTALYTD. Antes de escribir cualquier fórmula YTD, es una buena idea configurar una matriz que le brinde retroalimentación inmediata si su fórmula está funcionando como se esperaba. Por lo tanto configure una nueva matriz como la que se muestra a continuación antes de crear la medida acumulada.

IDAño	MesNombre	Total Ingresos
<input type="checkbox"/> 2015	ene	1340,245
<input type="checkbox"/> 2016	feb	1414,611
<input type="checkbox"/> 2017	mar	1480,905
<input checked="" type="checkbox"/> 2018	abr	1608,751
	may	1878,318
	jun	1949,361
	jul	50,841
	<b>Total</b>	<b>9723,031</b>

Ahora cree la siguiente medida:

Ingresos Acumulados = TOTALYTD([Total Ingresos],Calendario[Fecha])

IDAño	MesNombre	Total Ingresos	Ingresos Acumulados
<input type="checkbox"/> 2015	ene	1340,245	1340,245
<input type="checkbox"/> 2016	feb	1414,611	2754,856
<input type="checkbox"/> 2017	mar	1480,905	4235,761
<input checked="" type="checkbox"/> 2018	abr	1608,751	5844,512
	may	1878,318	7722,829
	jun	1949,361	9672,191
	jul	50,841	9723,031
	ago		9723,031
	sep		9723,031
	oct		9723,031
	nov		9723,031
	dic		9723,031
	<b>Total</b>	<b>9723,031</b>	<b>9723,031</b>

## Practicar funciones de inteligencia de tiempo

Calcule el mes anterior, para calcular el mes anterior y tener un comparativo del mes actual vs el mes anterior, ahora puedes usar la función PREVIOUSMONTH.

La sintaxis es PREVIOUSMONTH(<dates>). Esta función al igual que muchas deben utilizarse junto con CALCULATE, para un desempeño correcto. Hagamos el siguiente ejemplo:

Ingresos mes anterior =

CALCULATE([Total Ingresos],PREVIOUSMONTH(Calendario[Fecha]))

ID Año	MesNombre	Total Ingresos	Ingresos mes anterior	Ingresos Acumulados
<input type="checkbox"/> 2015				
<input type="checkbox"/> 2016				
<input type="checkbox"/> 2017				
<input checked="" type="checkbox"/> 2018	ene	1340,245	1731,788	1340,245
	feb	1414,611	1340,245	2754,856
	mar	1480,905	1414,611	4235,761
	abr	1608,751	1480,905	5844,512
	may	1878,318	1608,751	7722,829
	jun	1949,361	1878,318	9672,191
	jul	50,841	1949,361	9723,031
	ago		50,841	9723,031
	sep			9723,031
	oct			9723,031
	nov			9723,031
	dic			9723,031

Como se puede apreciar en la imagen de arriba si seleccionamos el mes de febrero en la columna "Ingresos mes anterior" presenta en esa fila los ingresos que corresponde al mes de enero. Hay muchas funciones de inteligencia de tiempo que puedes utilizar realizar los análisis que deseas visualizar con DAX. A medida que vas aprendiendo a utilizar las funciones de inteligencia de tiempo DAX puedes realizar una búsqueda en su catalogo de funciones en la web de Microsoft, aquí te voy a compartir el enlace.

### Lista de funciones Inteligencia de tiempo DAX

- [PREVIOUSDAY](#) (Date\_Column)
- [PREVIOUSMONTH](#) (Date\_Column)
- [PREVIOUSQUARTER](#) (Date\_Column)
- [PREVIOUSYEAR](#) (Date\_Column [, YE\_Date])
- [Nextday](#) (Date\_Column)
- [NEXTMONTH](#) (Date\_Column)
- [NEXTQUARTER](#) (Date\_Column)
- [NEXTYEAR](#) (Date\_Column [, YE\_Date])



Hay otras funciones que tienen similitud en sus calculos pero que ofrecen una mayor flexibilidad en cuanto al calculo deseado. Hemos realizado un ejemplo de inteligencia de tiempo con la función SAMEPERIOLASTYEAR, ahora vamos a realizar un ejemplo con la función DATEADD que permite realizar calculos desplazando el tiempo, ya sea hacia atrás o hacia adelante por el número especificado de intervalos desde la fecha del contexto actual.

La sintaxis es DATEADD(<dates>, <number\_of\_intervals>,<intervals>).

Si el número de intervalos es positivo las fechas se adelantan en el tiempo; si el número es negativo las fechas se desplazan o retroceden en el tiempo. Y el parametro intervalo se refiere al periodo que se desea desplazar, esdecir: Año, trimestre, mes o día. Es importante que esta condición sea especifica.

Hagamos el siguiente ejemplo:

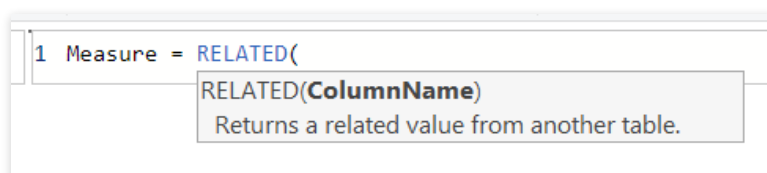
Ingresos año anterior DATEADD=

CALCULATE([Total Ingresos],DATEADD(Calendario[Fecha],-1,YEAR))

IDAño	MesNombre	Total Ingresos	Ingresos Periodo anterior	Ingresos año anterior DATEADD
<input type="checkbox"/> 2015	▲			
<input type="checkbox"/> 2016				
<input type="checkbox"/> 2017	ene	1340,245	438,865	438,865
<input checked="" type="checkbox"/> 2018	feb	1414,611	489,090	489,090
	mar	1480,905	485,575	485,575
	abr	1608,751	506,399	506,399
	may	1878,318	562,773	562,773
	jun	1949,361	554,799	554,799
	jul	50,841	886,669	886,669
	ago		847,414	847,414
	sep		1010,258	1010,258
	oct		1080,450	1080,450
	nov		1196,981	1196,981
	dic		1731,788	1731,788
	<b>Total</b>	<b>9723,031</b>	<b>9791,060</b>	<b>9791,060</b>

## 15. La función RELATED

La función RELATED se usa típicamente en columnas calculadas para hacer referencia a datos y listas relevantes en otras tablas, aunque también se pueden usar en medidas. Son un poco como VLOOKUP () para tablas que tienen una relación. Como se mencionó brevemente en capítulos anteriores, el contexto de una fila no sigue una relación. Por lo tanto, aunque puede haber una relación entre dos tablas, un contexto de fila no puede usar esta relación, a menos que use una de estas dos funciones que sí puede. Básicamente, RELATED () y RELATEDTABLE () permiten que un contexto de fila aproveche una relación existente para que pueda acceder a columnas en tablas relacionadas



Esta sección proporciona un ejemplo de traer un valor de una columna en una tabla de búsqueda a una tabla en el lado "muchos" de la relación. En aras de este ejemplo, suponga que desea incluir en la tabla de ventas la columna de Grupo de la tabla Países para realizar un filtro.

Para lograr esto, cree un campo calculado con la función RELATED, Teniendo en cuenta que la tabla Países y la tabla Ventas están relacionadas:

1. Elegir el icono Data.
2. Seleccionar la tabla Ventas.
3. Seleccionar campo calculado y realizar la siguiente medida.
4. Grupo Region = RELATED(Paises[Grupo]).

1 Grupo Region = RELATED(Paises[Grupo])

Costo	Ingreso	Impuesto	Flete	Region	Grupo Region
0	1.87	4.99	0.3992	Sur America	America Latina
0	1.87	4.99	0.3992	Sur America	America Latina
0	1.87	4.99	0.3992	Sur America	America Latina
0	1.87	4.99	0.3992	Sur America	America Latina
0	1.87	4.99	0.3992	Sur America	America Latina
0	1.87	4.99	0.3992	Sur America	America Latina
0	1.87	4.99	0.3992	Sur America	America Latina
0	1.87	4.99	0.3992	Sur America	America Latina

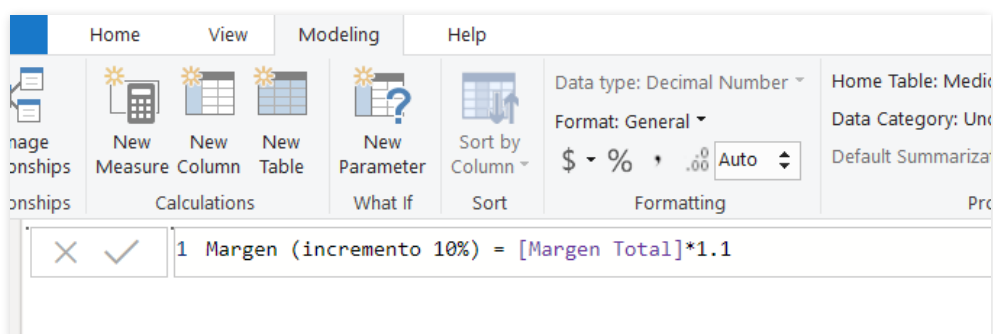
## 16. El uso de tablas desconectadas

Todo lo que hemos trabajado hasta este momento en este libro, siempre ha sido cargado las tablas necesarias en el modelo de datos para conectarlas a otras tablas. Esta es una técnica esencial a utilizar Power BI, ya que le permite trabajar en varias tablas sin usar BUSCARV (). Sin embargo, Power BI también permite trabajar en el modelo de datos sin tener que unir todas las tablas, de hecho hay algunos casos en los que no tiene sentido hacerlo. Esta sección analiza dos técnicas que no implican tablas de conexión:

- La función análisis What-If (New Parameter)
- Uso de tablas desconectadas

El análisis "What-If" Qué pasaría sí?, permite crear variables utilizando parametros para definir o cuantificar diferentes escenarios en sus informes. Esta es una función muy poderosa porque le permite crear analisis avanzados, tipo predicción o pronostico de lo que podría suceder en el futuro.

Por ejemplo, imagine que en sus datos, el resultado de las ventas es directamente proporcional al resultado del beneficio. Tiene los datos de ventas actuales y desea ver qué impacto tendrá un aumento en las ventas en su beneficio total. Podría escribir una nueva medida codificada con un aumento del 10%, como se muestra a continuación



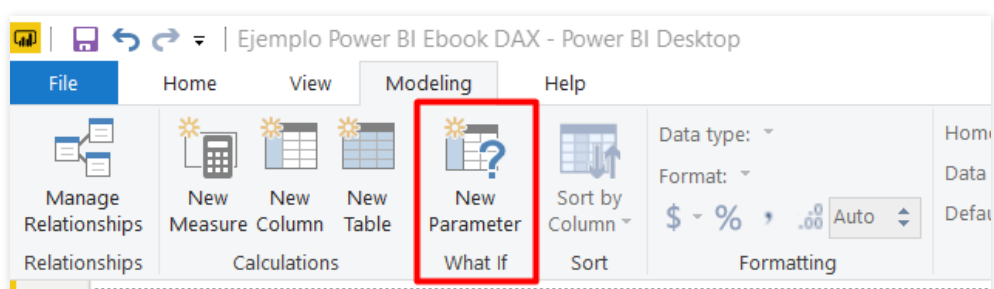
Y la Tabla matriz se vería como se muestra a continuación:

Categoría	Total Ingresos	Margen Total	Margen (incremento 10%)
Accesorio	699,107	437,640	481,404
Bicicleta	28272,920	11487,524	12636,277
Prenda	338,781	136,043	149,647
<b>Total</b>	<b>29310,809</b>	<b>12061,207</b>	<b>13267,327</b>

Pero, ¿qué pasaría si quisiera ver cómo se ve un aumento del 15% en las ventas, o del 20%, o del 5%, o algún otro porcentaje? Con una formula estatica no sería eficiente, porque se requeriría el tener que crear una nueva medida para cada valor, y como comentamos, eso no es nada eficiente. Por lo tanto la alternativa que ofrece Power BI es utilizar la capacidad y el poder de análisis What-If.

A continuación, explicamos como utilizar What-If

1. Cree una nueva página en blanco en su libro de Power BI.
2. Navegue a la pestaña Modelado (Modeling) y haga clic en Nuevo parámetro como se muestra a continuación.



3. Ingrese un valor mínimo y máximo que se utilizará dentro del análisis Qué pasa si, junto con el incremento. Como puede ver a continuación, he ingresado un rango de enteros de 0 a 10. También le he dado a mi parámetro un nombre "Incremento".

What-if parameter

Name  
Incremento

Data type  
Whole number

Minimum  
0

Maximum  
10

Increment  
1

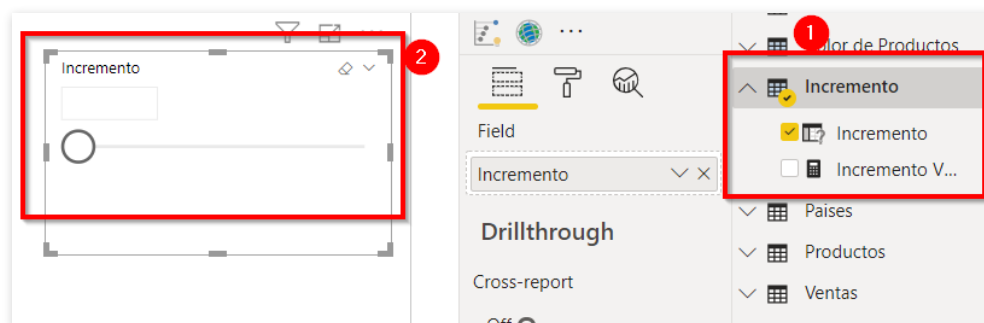
Default

☒ Add slicer to this page

OK Cancel

4. Haga clic en Aceptar.

5. Observe que hay una nueva Tabla (que incluye una nueva medida [Incremento Value] en la lista de Campos (# 1) y también un nuevo Slicer o segmentación (# 2) en el Informe.



6. Escriba la siguiente nueva medida que utiliza la medida [Aumentar valor] que se muestra arriba.

Total Margen con incremento selector = [Margen Total] \* (100 + [Incremento Value])/100

7. Agregue la medida anterior en la matriz de antes, como se muestra a continuación. Ahora puede usar el Slicer (# 1) para variar el aumento de ventas y ver qué impacto tiene ese aumento en la rentabilidad del negocio [Total Margen con incremento selector]. Para el ejemplo utilice un incremento del 5%.

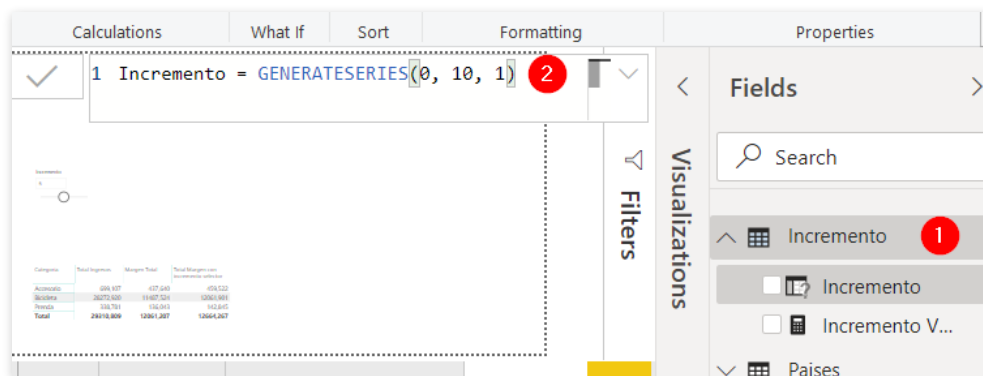
The screenshot shows the Power BI interface with a matrix and a slicer. The slicer, titled 'Incremento', has a value of 5 selected. The matrix displays the following data:

Categoría	Total Ingresos	Margen Total	Total Margen con incremento selector
Accesorio	699,107	437,640	459,522
Bicicleta	28272,920	11487,524	12061,901
Prenda	338,781	136,043	142,845
<b>Total</b>	<b>29310,809</b>	<b>12061,207</b>	<b>12664,267</b>

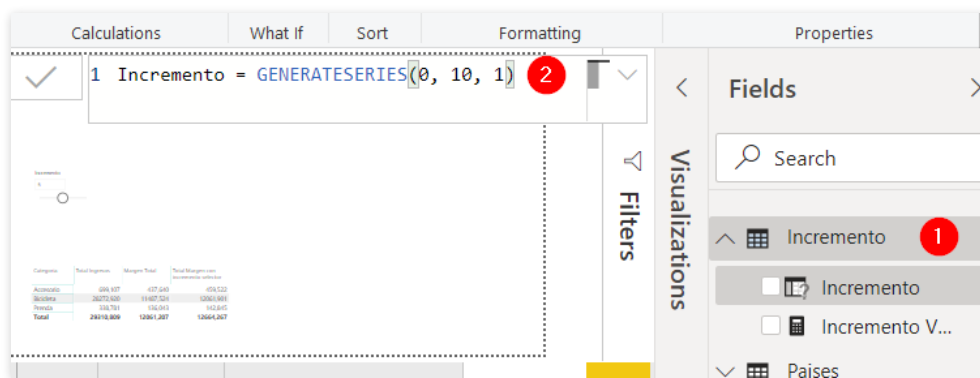
A red box highlights the 'Incremento' slicer, and another red box highlights the 'Total Margen con incremento selector' column in the matrix. A red circle with the number 1 is placed over the 'Incremento' slicer, and a red circle with the number 2 is placed over the 'Total Margen con incremento selector' column.

**Cómo funciona:** Cuando hace clic en el botón Nuevo parámetro en Power BI, Power BI crea automáticamente tres cosas. Una nueva tabla de valores, una nueva medida y un nuevo Slicer o segmentador de datos (opcional pero creado por defecto).

**La nueva tabla:** Seleccione la nueva tabla creada por el botón Nuevo parámetro en la lista Campos a la derecha (# 1 a continuación). Vas a ver que esta nueva tabla de aumento es una tabla calculada que utiliza la fórmula que se muestra como # 2 a continuación.

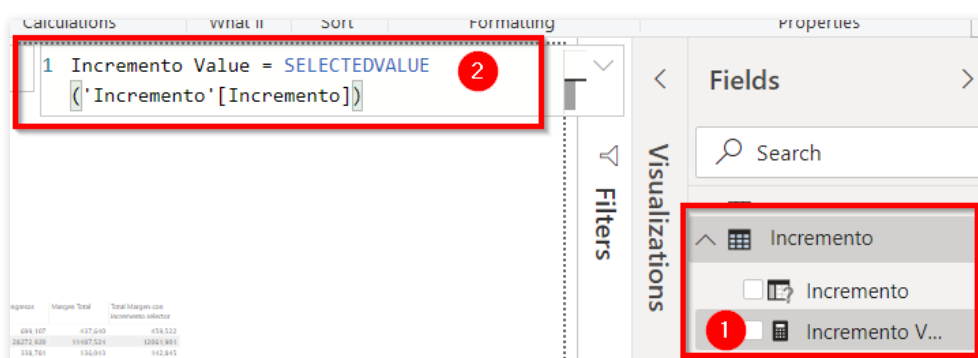


Como puede ver la función generada automáticamente Power Bi, es la GENERATESERIES (), sus sintaxis es simplemente, por ende mirando la fórmula, GENERATESERIES () es una función en el escritorio de Power BI que puede usar en cualquier momento que desee crear una nueva tabla de valores con un incremento constante entre los valores.



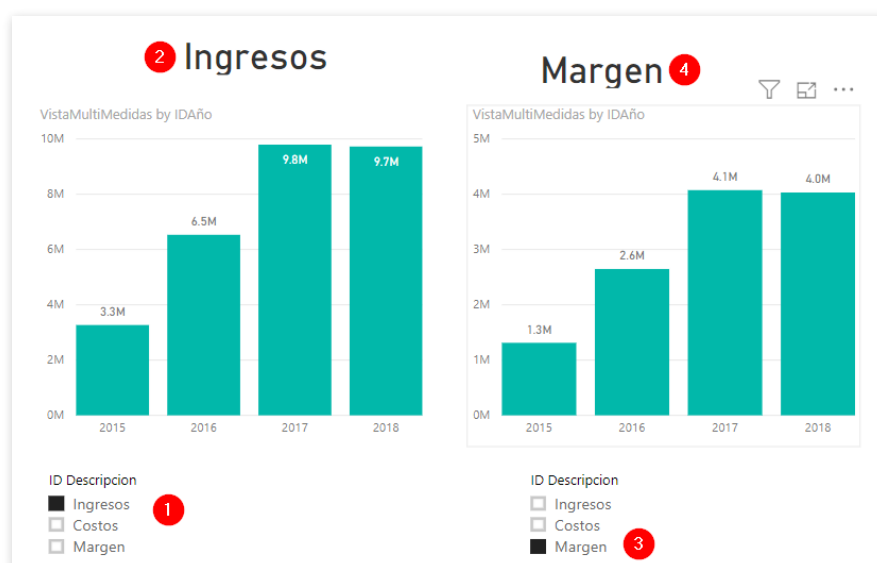
**El nuevo Slicer:** Vuelva a la vista Informe. El nuevo segmentador que se creó automáticamente utiliza la única columna de la tabla Incremento [Incremento] como campo de entrada. La segmentadora permite al usuario seleccionar un valor único para representar el aumento de ventas requerido para el análisis.

**La nueva medida:** Si ahora hace clic en la medida [Incremento Value], verá que la fórmula de la medida es la siguiente:  
Incremento Value = SELECTEDVALUE (Incremento [Incremento])



## La función SWITCH recargada

En capítulos anteriores, le presenté la función SWITCH. Una característica realmente genial de SWITCH es que puede crear una medida de cambio que le permite alternar entre varias otras medidas. Hecha un vistazo a la imagen de abajo.



La primera imagen de la izquierda tiene Ingresos Seleccionado en la segmentación. Cuando se selecciona Ingresos, el gráfico se actualiza para mostrar Ingresos (# 2). Cuando el usuario selecciona un valor diferente en la segmentación, como Margen (# 3), el gráfico cambia para mostrar el Margen (# 4). Este efecto de alternar medidas es realmente atractivo para el usuario del informe y se puede utilizar para crear reportes interactivos muy complejos y útiles para analizar.

## Cómo crear esta medida con SWITCH

Debe crear una tabla desconectada y una medida casi virtual para poder completar esta técnica:

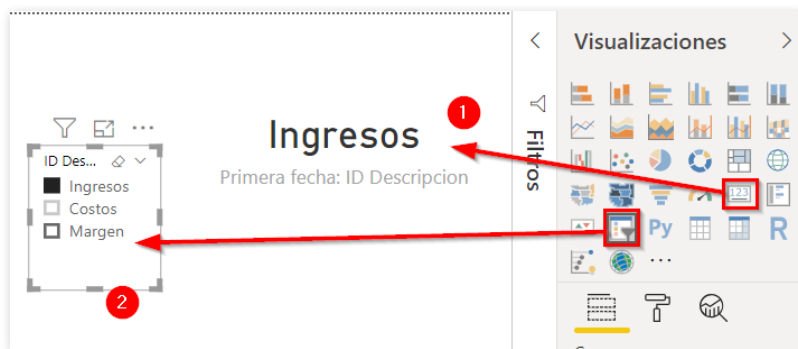
1. En el menú Inicio, haga clic en Ingresar datos.
2. Ingrese 3 filas de datos como se muestra a continuación. Llame a la tabla MultiMedidas, luego haga clic en Cargar.

Create Table

	ID	ID Descripción	*
1	1	Ingresos	
2	2	Costos	
3	3	Margen	
*			

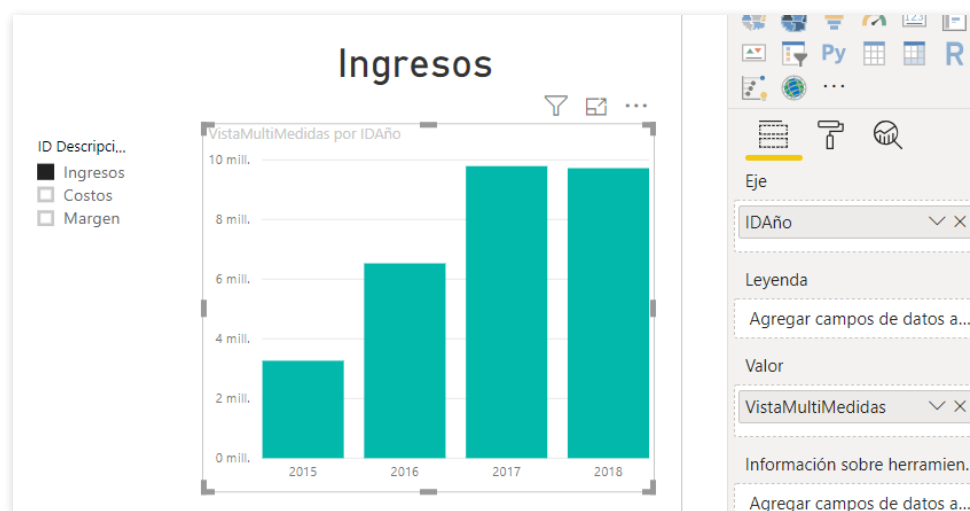
Name:

3. Navegue a la vista Datos, haga clic en la nueva tabla MultiMedidas, luego vaya al menú Modelado y cambie el orden de clasificación de la columna IDDescripcion para que se clasifique por columna de IDMedida. ¿Recuerdas cómo? Lo hiciste en capítulos anteriores.
4. Haga clic derecho en la tabla MultiMedidas y seleccione nueva medida, para agregar una nueva medida que se llamará de la siguiente manera.  
 SelectorMedidas =SELECTEDVALUE (MultiMedidas [IDMedida])
5. Si inserta una "Tarjeta" podrá validar que al seleccionar cualquier valor, devuelve ese mismo valor, de lo contrario devuelve EN BLANCO. Todo esto cuando usa una lista de segmentación para filtrar.
6. Cree una nueva página. Coloque una Tarjeta (# 1) en el informe y agregue a la vista la columna MultiMedidas[IDDescripcion] a los Campos de la Tarjeta. Luego coloque una segmentacion o Slicer (# 2) en el informe y coloque MultiMedidas [IDDescripcion] en la segmentacion también.



7. Ahora, cuando hace clic en la segmtación, la Tarjeta se actualizará para mostrarle qué medida ha seleccionado:  
 ejemplo. Esta tarjeta será el título de la tabla que se agregará a continuación.
8. Haga clic derecho en la MultiMedidas y escriba la siguiente medida.  
 VistaMultiMedidas = SWITCH ([SelectorMedidas], 1, [Total Ingresos],  
 2, [Total Costos],  
 3, [Margen Total ])
9. Agregue un nuevo Gráfico de columnas al informe (gráfico de columnas agrupadas o apiladas, ambas funcionarán). Coloque Calendario [IDAño] en el Eje y la nueva medida [VistaMultiMedidas] en el campo Valor como se muestra a continuación.





10. Ahora puede interactuar en un gráfico interactivo de tres medidas que muestra la medida seleccionada por el usuario en la segmentación.

## 17. Uso de variables

Las variables a diferencia de las medidas permite almacenar el resultado de las expresiones o medidas en la memoria cache. La variable tiene un funcionamiento de dos palabras clave en DAX que se usan juntas para permitir que las formulas que hagamos sean legibles. Las palabras claves son **VAR** y **RETURN**.

Esta es la Sintaxis:

Mi columna (o medida) =

VAR <primerNombreVariable> = <expresión>

VAR <segundoNombreVariable> = <otra expresión>

RETURN

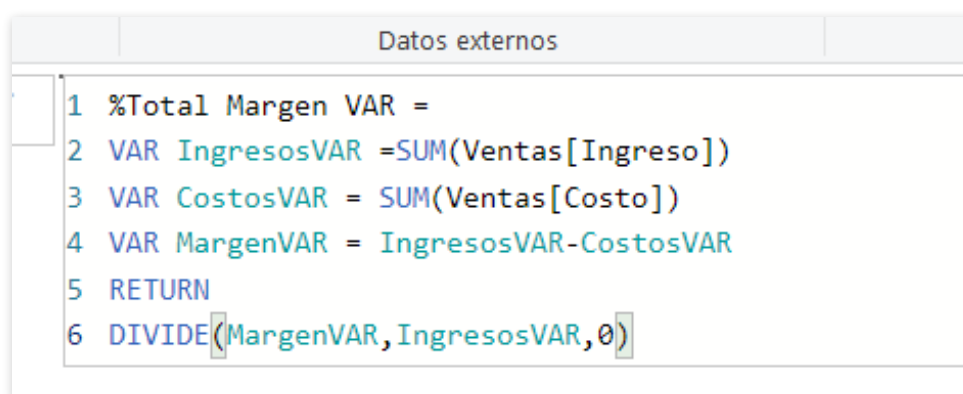
<otra expresión DAX que puede hacer referencia a las variables>

Las variables son útiles para simplificar medidas, es decir en vez de crear tres medidas para crear otra medida, como es el caso cuando deseamos por ejemplo calcular el % Margen Total de los ingresos menos los costos. Tendría que calcular una medida para los ingresos, otra medida para los costos, luego una medida que reste los ingresos de los costos y finalmente una medida que divida el Margen sobre los ingresos.

Aquí es donde entra en funcionamiento las variables si deseo simplificar el proceso y en vez de crear dos o tres medidas creo una sola. Veamos un ejemplo.

Cree la siguiente medida utilizando la función de variables VAR.

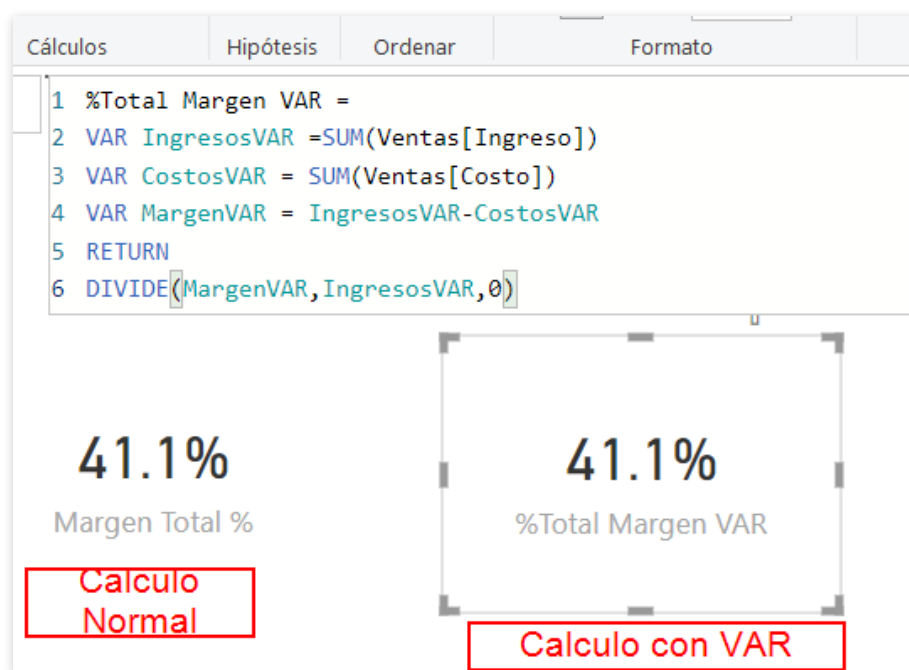
```
%Total Margen VAR  
VAR IngresosVAR =SUM(Ventas[Ingreso])  
VAR CostosVAR = SUM(Ventas[Costo])  
VAR MargenVAR = IngresosVAR-CostosVAR  
RETURN  
DIVIDE(MargenVAR,IngresosVAR,0)
```



Como puedes apreciar en la imagen de arriba, hemos simplificado en una sola medida, tres medidas, lo que permite mejorar el rendimiento de nuestros modelos. Ahora bien el uso del CÓMO, CUÁNDO y DÓNDE utilizar las variables dependerá de nuestro criterio. No dejan de ser una expresión, por lo tanto, si estas expresiones solo se utilizaran en el código que tratamos, su uso se nos hace útil, sin embargo, si tenemos distintas medidas que la utilizaran, quizás sea más útil incluirlas como medidas que puedan ser llamadas posteriormente, en cada uno de los algoritmos o medidas que vayamos a utilizar.

Como siempre es importante tener un esquema claro de cual será nuestro reporte y cuáles serán nuestras necesidades de cálculo para su mejor representación, optimizando los recursos en el desarrollo de nuestro modelo.

Tenemos que tener en cuenta que al introducir una variable en DAX, ésta se calcula al encontrar RETURN, y la almacena en la memoria cache como un valor constante para el resto de nuestra medida, pudiendo ser llamada cuantas veces requiramos en la misma expresión. No es necesario el recalcularse la expresión cada vez que es llamada en DAX, por lo que como comentábamos mejora el rendimiento de ésta.



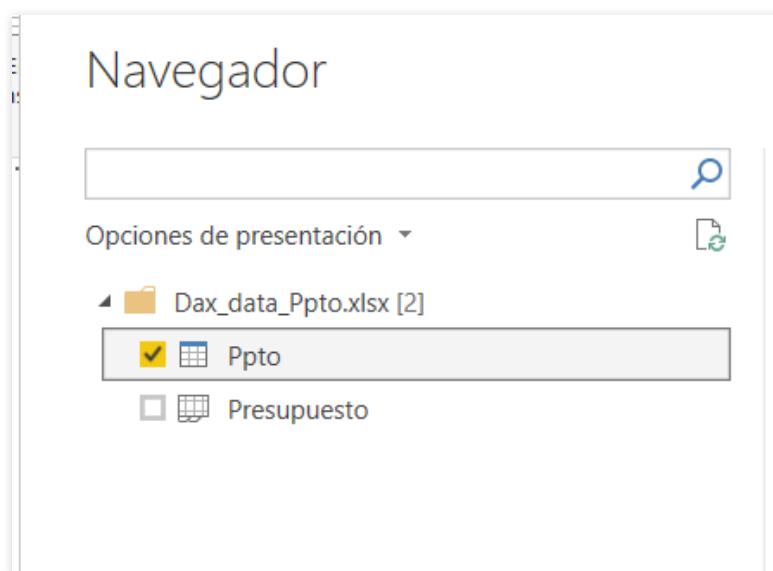
## 18. Múltiples Tablas

Hasta ahora en este libro, hemos utilizado solo una tabla de datos, la tabla de ventas. Es muy probable que desee o necesite usar varias tablas de datos en sus modelos de datos. Cuando trae una segunda tabla de datos a Power BI, es común que las personas piensen que deberían unir la nueva tabla de datos a la tabla de datos original, pero esto es incorrecto. La forma correcta de unir una segunda tabla de datos a un modelo de datos es tratar la nueva tabla de datos exactamente igual que la primera tabla de datos.

Para ayudarlo a comprender cómo hacerlo, veamos un escenario empresarial común en el que una empresa desea cargar una tabla de ventas y una tabla de presupuesto. Uno de los desafíos de este escenario es que el presupuesto a menudo tiene un nivel de granularidad diferente al de las ventas reales. Por ejemplo, las ventas tienen un desempeño diario y se pueden informar todos los días para cada producto individual, pero el presupuesto no maneja ese nivel de detalle diario, sino que se establece de manera mensual para cada categoría de producto.

Los siguientes pasos lo guían a través del proceso de importación de una tabla de Presupuesto, creando una nueva tabla de Presupuesto, y luego creando una medida para el presupuesto:

1. En Power BI, haga clic en Obtener datos, más, acceder a la base de datos y navegue a la base de datos que esta en el archivo de Excel, **Dax\_data\_Ppto.**
2. Seleccione las tablas **Ppto** de la lista, como se muestra a continuación.



3. Clic en Cargar. Ahora verá que la tabla Presupuesto tiene un presupuesto de ventas mensual para cada categoría. Una columna para el año, mes número y otra con la fecha que tiene el formato M/D/AAA, como se muestra en la imagen de abajo.

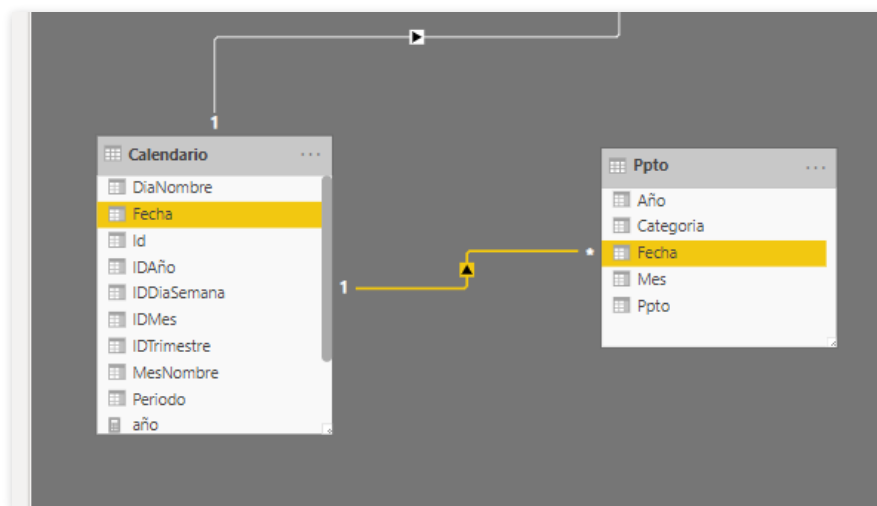
Categoría	Ppto	Año	Mes	Fecha
Accesorio	25400	2017	1	1/1/2017
Accesorio	16500	2017	2	2/1/2017
Accesorio	36700	2017	3	3/1/2017
Accesorio	51000	2017	4	4/1/2017
Accesorio	23000	2017	5	5/1/2017
Accesorio	27000	2017	6	6/1/2017
Accesorio	16000	2017	7	7/1/2017
Accesorio	53000	2017	8	8/1/2017
Accesorio	56000	2017	9	9/1/2017
Accesorio	56000	2017	10	10/1/2017
Accesorio	54000	2017	11	11/1/2017
Accesorio	72000	2017	12	12/1/2017
Accesorio	61000	2018	1	1/1/2018

#### 4. Ahora puede empezar a crear las relaciones.

Si pudo observar en la hoja de Presupuesto en Excel tuvimos que crear una columna concatenada de la columna Año y Mes, pero como el Presupuesto es mensual (Un escenario empresarial muy común), le agregamos en la opción día el número 1 para que pueda cumplir con la regla de la tabla Calendario que es diaria. Mi recomendación para la fecha presupuesto que utilices sea en formato igual a la tabla Calendario, es decir M/D/AAA, que para el día utilices siempre 1 y así la unión y relación de ambas tablas funciones correctamente.

Como ya sabemos Power BI solo admite relaciones uno a mucho. Es decir que la tabla de búsqueda calendario debe tener simplemente un valor único, es decir cada tabla de relación siempre debe contener un valor único para que la unión a una tabla de muchos valores funcione bien.

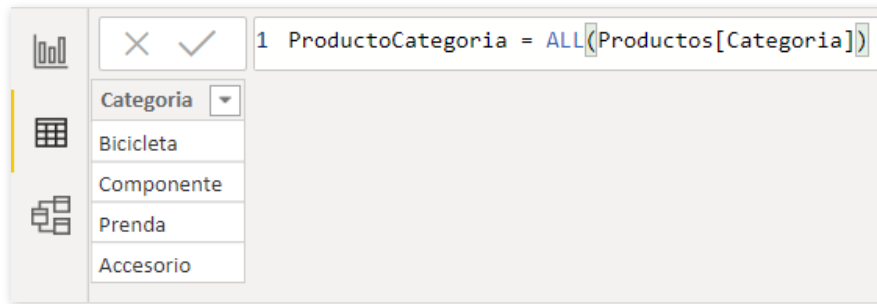
#### 5. Cree la relación de la columna Fecha de la tabla Ppto con la columna Fecha de la tabla Calendario.



6. Ahora bien necesitamos unir la tabla Productos [Categoria] con la tabla Ppto [Categoria], sin embargo si intenta unir ambas columna de estas tablas obtendrá un error. Por qué, porque Power BI no permite unir dos tablas con multiples valor de forma correcta, para que la unión sea efectiva debe ser de uno a muchos o de muchos a uno.

7. Para unir la tabla Ppto [Categoria] a Tabla Producto será necesario crear una nueva tabla que se llame ProductoCategoria con los valores unicos de cada categoria para realizar una relación efectiva.

8. Para construir la nueva tabla ProductoCategoria, haga clic en el botón Nueva tabla y escriba la fórmula que se muestra abajo.

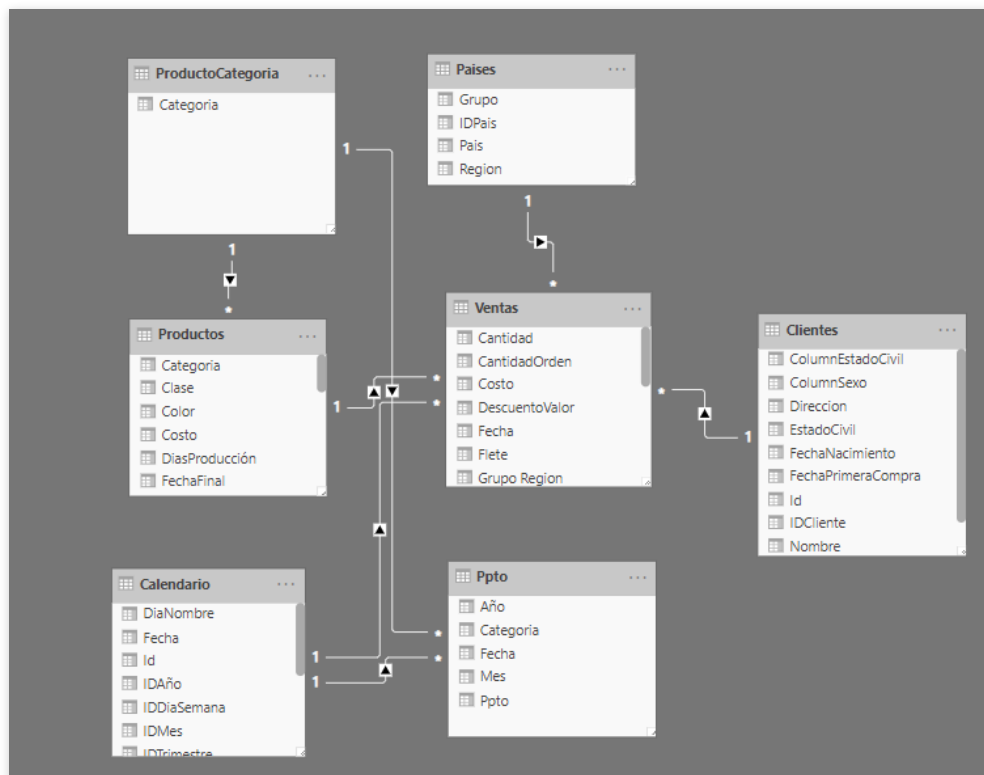


9. La nueva tabla ProductoCategoria se ha creado con una lista resumida de las cuatro categorías de productos.

10. Para unir la tabla Ppto a la tabla ProductoCategoria, haga clic y arrastre la columna Ppto [Categoría] a la columna Producto [Categoría].

11. Para unir la tabla Productos a la tabla ProductoCategoria, haga clic y arrastre la columna Productos [Categoría] a la columna ProductoCategoria [Categoría].

Cuando haya terminado, debería tener algo como el diseño que se muestra a continuación. Tenga en cuenta que resulta difícil realizar un seguimiento de todas las relaciones cuando tiene muchas tablas en un modelo de datos. Esta es una razón por la cual se recomienda organizar las tablas usando la metodología de diseño estrella.



## Tiempo de practicar

Iniciemos este ejemplo primero, creando la medida [Total Ppto] , luego active una nueva matriz, después ponga en Filas Calendario [Periodo] ,y [Total Ingresos] en Valores, al igual que [Total Ppto]. Inserte también una segmentación e incluya la columna [Categoria] de la tabla ProductoCategoria. Asegúrese de seleccionar las columnas correctas de cada tabla.

Categoria	Periodo	Total Ingresos	Total Ppto
<input type="checkbox"/> Accesorio	201707	865,693	909,000
<input checked="" type="checkbox"/> Bicicleta	201708	772,022	826,000
<input type="checkbox"/> Componente	201709	931,964	1007,000
<input type="checkbox"/> Prenda	201710	999,929	1000,000
	201711	1116,621	1195,000
	201712	1635,373	1750,000
	201801	1254,568	1242,000
	201802	1332,195	1322,000
	201803	1392,038	1364,000
	201804	1514,049	1620,000
	201805	1775,297	1704,000
	201806	1848,954	1775,000
	201807		155,200
	201808		150,100
	201809		148,500
	201810		150,500
	201811		155,000
	201812		175,000
	<b>Total</b>	<b>28272,920</b>	<b>19691,300</b>

Ahora cree las siguientes medidas:

**[Var. Ingresos vs Ppto]**

**[%Var. Ingresos vs Ppto]**

Categoria	Periodo	Total Ingresos	Total Ppto	Var. Ingresos vs Ppto	%Var. Ingresos vs Ppto
<input type="checkbox"/> Accesorio	201706	554,799	527,000	27,799	5.0%
<input checked="" type="checkbox"/> Bicicleta	201707	865,693	909,000	-43,307	-5.0%
<input type="checkbox"/> Componente	201708	772,022	826,000	-53,978	-7.0%
<input type="checkbox"/> Prenda	201709	931,964	1007,000	-75,036	-8.1%
	201710	999,929	1000,000	-71	-0.0%
	201711	1116,621	1195,000	-78,379	-7.0%
	201712	1635,373	1750,000	-114,627	-7.0%
	201801	1254,568	1242,000	12,568	1.0%
	201802	1332,195	1322,000	10,195	0.8%
	201803	1392,038	1364,000	28,038	2.0%
	201804	1514,049	1620,000	-105,951	-7.0%
	201805	1775,297	1704,000	71,297	4.0%
	201806	1848,954	1775,000	73,954	4.0%
	201807	155,200	155,200	-155,200	0.0%
	201808		150,100	-150,100	0.0%
	201809		148,500	-148,500	0.0%
	201810		150,500	-150,500	0.0%
	201811		155,000	-155,000	0.0%
	201812		175,000	-175,000	0.0%
	<b>Total</b>	<b>28272,920</b>	<b>19691,300</b>	<b>8581,620</b>	<b>30.4%</b>



**Muchas Gracias  
Exitos y bendiciones**

Email: [info@wallbi.com](mailto:info@wallbi.com)

[www.wallbi.com](http://www.wallbi.com)