**CLASE 1: CONSEJOS**

**1. Dax opera con 3 objetos:**

1. Columna
2. Medidas
3. Tablas

**2. No es recomendable usar columnas calculadas**

* Ocupan demasiado espacio
* Son redundantes
* Si eliminamos tablas se pierden columnas
* Es preferible usar Power Query

**3. Las medidas son el objeto más importante de DAX, úsalas siempre.**

* Es una agregación o un calculo
* Es como una función en programación
* No ocupan espacios
* Se reutilizan

**4. Es recomendable siempre usar medidas explicitas en los reportes**

* Implícitas: Se generan automáticamente por PBI
* Explicitas: Las generamos con DAX

**5. Intentar no usar tablas calculadas**

* No crear tablas calculadas
* Usar forma virtual (con fórmulas) o usar duplicar de Power Query

**6. Usar nombres explicativos en objetos e identar código**

* Para identar código Shift + Enter
* Usar página web dax formatter.com para indentar

**7. Crear siempre tabla de medidas**

* Pueden crear varias tablas para clasificar
* Se pueden generar subcarpetas

**8. Las funciones agregativas tienen su versión avanzada**

* Son funciones de iteración
* Ej. Suma tiene sumx

**9. Puedes definir variantes de medidas con CALCULATE**

* CALCULATE es la función más importante de dax
* Permite crear variantes
* Permite modificar el contexto de filtro

**10. DAX tiene muchas funciones, pero solo nos interesan unas pocas**

***Para entender formulas hay que entender y dominar los contextos.***

**CLASE 2: MODELADO DE DATOS EN PBI**

El Modelo de datos es la parte base. Power BI usa un modelo analítico: No es necesario que esté muy atomizado en varias tablas como en SQL. Para eso usa el **modelo estrella** o dimensional.

1. **Del modelo de una tabla al modelo estrella.**

Con una tabla no dimensionada los cálculos pueden tener problemas y las fórmulas se hacen complicadas:

***Problema 1***: Incorrecta asignación del nivel de granularidad. Asignaciones y agrupaciones malas.

***Problema 2***: Mucha información repetida (redundante). Se deben hacer muchas actualizaciones en una sola tabla.

***Problema 3***: Alto riesgo de alterar la integridad de los datos al hacer actualizaciones o modificaciones manuales.

Solventar y crear maestro:

* + Debemos encontrar entidades de un proceso de negocio.
  + Las entidades suelen ser columnas de la tabla relacionadas a algo (ej. el maestro de clientes, maestro de productos).
  + Creamos una tabla con ellas
  + Quitamos sus duplicados

Solventar y crear modelo de datos:

* + Relacionar maestros con tabla original.

1. **Elementos de un modelo multidimensional**

Tablas Maestras o dimensiones:

* Suelen ir con dimNombre
* Deme tener una primary key
* Para identificar debemos hacernos grandes preguntas. Cada tabla responde a una pregunta. Por ejemplo, tabla cliente es ¿A quién?, la tabla producto responde ¿qué vendo?, la tabla tienda responde ¿a dónde?
* Existe una ***dimensión calendario que siempre se debe crear***. Ella responde a ¿Cuándo?
* Contienen los filtros o segmentos del reporte

Tabla medio o tabla de hechos (Fact table):

* Cada fila representa un hecho o evento
* Tiene llaves una columnade llave foráneas (destino de relación) y las métricas (lo que cuantifico).
* A partir de las métricas se calcula DAX.
* Tienen el mayor nivel de granularidad o detalle.

Todo esto se denomina modelo estrella. No debe haber relación entre dimensiones, solo con tabla de hechos. Luego se pasa a Power Bi.

1. **Modelo estrella con varias tablas de hechos**

Se puede tener más de una tabla de hechos. Pero no necesariamente tiene las mismas dimensiones.

1. **Modelo copo de nieve**

Es una variante del modelo estrella. Ocurre cuando una dimensión no tiene relación directa con la tabla de hechos, sino que es indirecta a través de la relación con otra dimensión. Es decir, relación de dimensión 🡪 dimensión 🡪 hechos.

No hay ninguna ventaja usar copo de nieve. No ahorra espacio. Lo ideal es mantener modelo estrella estándar.

1. **Cardinalidad de una relación**

El símbolo 1: de uno

El símbolo \*: varios

* ***La relación esperada en modelado de PBI es la relación de uno a varios***.
* Cuando hay relación de uno a uno mejor es combinar las tablas de manera horizontal, una a la par de otra. No importa si hay nulos en la nueva tabla.
* Relación de varios a varios: Lo mejor es tener una tabla intermedia de dimensión. Crear modelo estrella con dos hechos.

1. **Estatus de una relación**

Una llave primaria no se puede relacionar con más de una llave foránea en la tabla hechos. Porque la segunda queda como relación inactiva (líneas puntuadas en PBI). Pero se puede activar con DAX.

1. **Dirección de filtro cruzada en una relación**

* La flecha de las relaciones es la dirección de filtro cruzado.
* Se puede cambiar en propiedades, ambas. Para que cruce desde una dim hacia otra pasando por tabla de hechos. Dim 🡪 hechos 🡪 dim.
* Pero ***Nunca activar manualmente relaciones bidireccionales***. Porque se hace ambiguos el modelo, porque dax elegirá arbitrariamente el camino más rápido.
* Mejor hacerlo directamente con CROSSFILTER de DAX.

1. **Mas allá del modelo estrella**

Variantes del modelo estrella:

* + Modelo encabezado-detalle (Header/detail)
  + Modelo con intervalos de tiempo
  + Modelo con relaciones de varios a varios
  + Modelo con tablas snapshot
  + Modelo con dimensiones cambiantes (SCD, RCD)
  + Modelo padre-hijo, etc.

**CLASE 3: PRÁCTICA**

**La tabla de hechos solo debe tener:**

* Llaves foráneas
* Métricas
* Código de transacción (opcional)

**Proceso ETL**

* Desactivar relaciones automáticas: Ir a configuración opciones y carga de datos, desmarcar detectar automáticamente nuevas relaciones.
* La llave primaria debe tener el mismo número de valores distintos que valores únicos.
* Ir a parte inferior y elegir perfil de fuente de datos completo. Para que no muestre las primeras 1000, sino todas.
* Para llaves primarias en PBI no importar si es texto o número porque ya lo optimiza.
* Una dimensión calendario debe tener todas las fechas completas y todos los años cerrados.

Modelo de varios hechos

* Sacar todas las entidades de las tablas disponibles
* Separa por dimensiones
* Generar dimensiones para cada tabla

Modelo copo de nieve:

* Si existe hay que desnormalizar el modelo
* Se debe integrar las tablas relacionadas a una dimensión a esa dimensión.
* En PQ se usa combinar consultas
* En PBI las relaciones significan un filtro, por eso no se debe relacionar una dimensión a dos dimensiones o más, sino directamente a la tabla de hechos. Eso funcionaria en modelos SQL, pero no en PBI.

**CLASE 4: CONTEXTO DE FILA EN DAX**

**1. Definición del ROW CONTEXT**

Representa la *fila actual* en la que nos encontramos durante ***el proceso de iteración*** *en una tabla*. *Solo existe cuando estemos en una iteración*.

**a) Objetos de DAX con contexto de fila:**

* El ***único objeto donde hay iteración son*** ***las columnas calculadas.*** Siempre tienen contexto de fila.
* Las ***medidas y tablas no usan contexto de fila*** por defecto, sino ***contexto de filtro***.

**b) Funciones X:**

Las iteraciones solo existen en las funciones x, es decir, en toda función que tenga X al final. Esas siempre tendrán contexto de fila, Por ejemplo:

* SUMX
* MINX
* MEDIANX
* RANKX

**c) Funciones de tabla:**

* FILTER
* ADDCOLUMNS
* SELECTCOLUMNS
* TOPN
* GENERATE

**d) Funciones int. De tiempo:**

* LASTNONBLANK
* FIRTNONBLANK

Todas las columnas calculadas que tengan estas funciones generan ***contexto de fila***.

**e) Funciones de relación**

Estas funciones no generan contexto de fila, pero si utilizan un contexto de fila.

* RELATED: Requiere que me ubique en el ***lado múltiple de una relación***
* RELATEDTABLE: Requiere que me ubique en ***el lado individual de una relación***. Ya no se utiliza mucho porque existe CALCULATE.

**2. Contexto de fila en columnas calculadas**

Las columnas calculadas son columnas extras en la tabla, pero no es recomendable trabajar con ellas. Nuevo contexto de fila y toma los valores de las celdas correspondientes: ej. Fila 1 celda de columnas 2 por celdas de columnas 5. Ósea, precio \* cantidad de la fila 1, este proceso se denomina iteración.

**3. Contexto de fila en medidas X**

Ejemplo: generar medida explicita y crear medida monto de ventas:

# contexto de fila

***Una medida no tiene contexto de fila de forma automática.***

**4. Contexto de fila en funciones de tabla**

Tabla con Filter para encontrar precios mayores a 50:

**5. Funciones Related y Relatedtable**

No generan contexto de fila, la primera se usa cuando estoy en el lado múltiple de una relación. La segunda es cuando estoy en el lado individual. Precio de lista:

Monto de venta a precio de lista:

Medida:

#trae valor de la fila relacionada

RELATEDTABLE: Ej si queremos a la dimensión productos agregarle una columna con las ventas por productos. Pero hay mejor maneras de hacerlo.

**CLASE 5: CONTEXTO DE FILTRO**

**1. Definición del contexto de filtro**

Es el conjunto de filtros que actúan sobre modelos de datos durante la evaluación de una expresión de DAX. Es un conjunto de valores que toma una columna. Un filtro es un valor que toma una columna y en dax se expresa en forma de tabla:

|  |
| --- |
| Categoría (columna) |
| Peluche (filtro) |

***Características***

* *El contexto de filtro siempre va a existir*

***Objetos que usan contexto de filtro o los afecta***

* Medidas
* Tablas

***Funciones DAX***

* CALCULATE
* CALCULATETABLE

***Funciones de tabla***

* FILTER no usa contexto de filtro, lo hace por iteración (contexto de fila).

**2. Contexto de filtro en medidas**

El contexto de filtro esta conformado por uno o más filtros de objetos, medidas o filtros de páginas. Si no hay filtros, el contexto de filtro es vacío. *Cada celda tiene su propio contexto de filtro, por ejemplo, Categoría Cameras y Genero*. Siempre debemos identificar primero el contexto de filtro para entender.

El filtro fluye desde las tablas dimensiones hasta la tabla de hechos. De arriba abajo. ***Ósea que el contexto de filtro se aplica a las dimensiones***. Luego de eso calcula la formula en la tabla hechos y por último se lo lleva hacia una nueva tabla o matriz, que es la que presentamos en el dashboard.

*Función COUNTROWS: Cuenta el número de filas de una tabla de hechos.*

**3. Contexto de filtro en tablas**

FILTER no modifica el contexto de filtro de una tabla, sino que realiza una iteración y evalúa una expresión. Solo se hace por iteración:

***La función VALUES no genera un contexto de filtro al usar una tabla calculad***a. ***Pero al usarlo en una medida de DAX si genera un contexto de filtro***. VALUES retorna una tabla virtual con los valores distintos.

***Sin contexto de filtro (Tabla nueva):***

***Con contexto de filtro (Medida DAX): Retorna valores distintos***

La función ***DISTINtCount es la combinación de COUNTROWS y VALUES*** como la fórmula anterior.

**4. Contexto de filtro y contexto de fila**

Medida para saber cuántos registros están teniendo más de 2 unidades vendidas.

Los pasos son:

* Generar contexto de filtro
* Aplicar contexto de filtro
* Aplicar la fórmula a la medida

**5. Introducción a CALCULATE**

Monto de venta de los clientes que son del continente asiático. ***La doble iteración se debe evitar en DAX***:

Se itera en SUMX y en FILTER y no es recomendable. Es más eficiente usar CALCULATE porque permite modificar o alterar el contexto de filtro.

Monto de venta Asia =

# medida original

# filtro adicional

Pasos del proceso:

* Contexto de filtro a las dimensiones
* CALCULATE agrega un nuevo contexto de filtro que es continente Asia
* Aplicamos contexto de filtro en la tabla hechos
* Aplicamos la fórmula
* Llevamos a la nueva matriz del Dashboard.

**CLASE 6: TRANSICIÓN DE CONTEXTO EN DAX**

La transición de contexto es la capacidad de las funciones CALCULATE y CALCULATETABLE de ***convertir un contexto de fila en un contexto de filtro equivalente***, que se aplicará al modelo durante la evaluación de una expresión de DAX.

**Características de la transición de contexto:**

* *Es una propiedad de la función* CALCULATE. Permite modificar el contexto de filtro y también genera una **transición de contexto**.
* El contexto de fila y de filtro no son lo mismo. El contexto de filtro es un conjunto de filtros que se aplican al modelo y el contexto de fila es la fila actual en una iteración.
* Para que exista una ***transición de contexto*** hay **2 requisitos**:
  + - *Estar en una iteración (Contexto de fila)*
    - *Utilizar la función CALCULATE o CALCULATETABLE: fuera de CALCULATE no hay transición de contexto.*

**Funciones que generan transición de contexto**

* CALCULATE
* CALCULTETABLE
* Funciones de inteligencia de tiempo

**Objetos de DAX que generan transición de contexto**

* Columnas calculadas
* Medidas 🡪

**2. Transición de contexto en columnas calculadas**

El objetivo es Crear una columna calculada que me diga cuál es el monto de venta por producto. *Pero hay que recordar que una columna calculada no tiene contexto de filtro.*

**Columna calculada incorrectamente**

El cálculo se hace sobre la tabla producto.

Es incorrecta porque el proceso es:

1. **Contexto de filtro**: Cómo las columnas calculadas no tienen es vacío.
2. **Aplicación de filtro**: No se aplica ningún filtro.
3. **Aplicar la fórmula**: Cómo no se filtra se suma toda la columna monto de venta.
4. **Llevar al dashboard**: Cómo sumo sin filtros siempre aparecerá el mismo resultado en todas las filas.

**Columna calculada correctamente**

Es correcta porque se usa CALCULATE:

1. **Contexto de filtro**: Toda las columnas en cuestión se agrega al contexto de filtro por CALCULATE.
2. **Aplicación de filtro**: Se aplica el filtro en la tabla hechos. Por el id o FK correspondiente.
3. **Aplicar formula**: Se aplica la fórmula en la tabla de hechos filtrada.
4. **Llevar a dashboard**: Se llevan los valores correctos por cada fila.

**Columna calculada usando una medida existente**

Donde la Medida existente Monto de Ventas es:

Esto es así porque ***toda medida en DAX tiene un CALCULATE implícito***.

**Ejercicio: Crear una columna calculada que me diga cuál es la primera fecha de venta de un producto**.

**3. Transición de contexto en tablas**

Queremos crear una con los clientes que compraron más de 7 unidades en total.

**Tabla calculada incorrectamente**

Es incorrecta por el proceso:

1. Contexto de filtro: No hay contexto de filtro y retorna todo

**Tabla calculada correctamente**

Los pasos correctos son:

1. **Contexto de filtro**: Por CALCULATE toda la fila es el contexto de filtro (convertir fila a filtro)
2. **Aplicar filtro**: Se aplica filtro a tabla hechos
3. **Aplicar fórmula**: Aplicar fórmula a la tabla hechos filtrada
4. **Presentar Dashboard**: Se llevan a la nueva tabla

Todo filtro se puede expresar como una tabla, entonces pódemos crear una medida

**4. Transición de contexto en medidas**

Sabemos y no olvidar que ***Toda medida en DAX tiene un CALCULATE implícito.***

En el caso de la tabla anterior la podemos arreglar y reemplazar el CALCULATE:

Entendiendo el concepto de transición de contexto ***podemos calcular clientes por mes***:

**Formula con suma incorrecta a nivel total**

Los pasos de PBI son:

1. **Contexto de filtro**: No hay filtro, es un contexto vacío.
2. **Aplicar filtro**: No se filtra
3. **Evaluar fórmula**: Cuenta número de valores distintos en la tabla hechos por id\_clientes

**Fórmula con suma correcta a nivel total:**

# tabla virtual retornada con valores distintos

# transición de contexto por CALCULATE implícito (nuevo filtro)

Los pasos en PBI son:

1. **SUMX recibe una tabla**: Una tabla nueva o virtual.
2. **VALUES**: muestra los valores distintos de la columna indicada
3. **Se evalúa el segundo argumento de SUMX**: Número de clientes. Cómo es una medida tiene CALCULATE implícito, por tanto, existe una transición de contexto.
4. **Contexto de filtro**: La fila se convierte en el filtro. El contexto de fila se convierte en contexto de filtro.
5. **Aplicamos filtro**: Aplicamos filtro a la tabla hechos.
6. **Aplicamos medida interna**: Se aplica formula (suma distintos) a la tabla hechos
7. **Llevar a dashboard**: Llevamos los cálculos a la tabla virtual.
8. **Calcular la medida externa**: Suma de la tabla virtual usando SUMX

**EJERCICIO FINAL: Medida que diga el mejor día de ventas**

# tabla virtual que recbibe MAXX

# Transición de contexto por CALCULATE (nuevo filtro la fila)

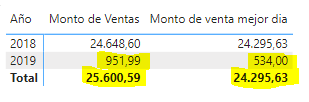
***Pasos en PBI con fórmula base:***

1. **Contexto de filtro**: Se aplica el año en la tabla (contexto externo)
2. **Aplicar filtro**: Se aplica filtro a la tabla hechos
3. **Aplicamos la fórmula**: Suma de la tabla hechos filtrada
4. **Llevar a dashboard**: llevar a la tabla de presentación

***Pasos en PBI con nueva medida****:*

1. **MAXX**: es un iterador y recibe una tabla VALUES
2. **VALUES**: Retorna una tabla virtual con valores distintos de fecha, o tabla de dimensión filtrada.
3. **Evaluar segundo argumento**: Toda medida tiene CALCULATE implícito, se convierte en transición.
4. **Contexto de filtro**: La fila de la tabla virtual se convierte en filtro.
5. **Aplicar filtro**: Aplicar filtro a la tabla hechos.
6. **Aplicamos la medida interna**: Sumamos elementos de columna montos de venta
7. **Rellenar la tabla virtual**: el resultado se guarda en la tabla virtual.
8. **Aplicar la medida externa**: de todos los valores guardados elije el máximo con MAXX.
9. **Llevar a dashboard**: Llevamos a la tabla de presentación.

Por eso los valores y las sumas son diferente cuándo se muestran en una misma tabla:



Esta técnica de iterador (MAXX, SUMX, ETC) más VALUE es una técnica ***llamada reducción de granularidad***.

**CLASE 6: SUMMARIZE Y SUMMARIZECOLUMNS**

1. **Introducción a SUMMARIZE y sintaxis**

GROUPBY es más lenta que SUMARIZE, pero a veces es útil. SUMMARIZE es una función de tabla y realiza tres operaciones diferentes:

* *Agrupar una tabla en base a una o más columnas del modelo*. ***Solo para esta operación usar SUMMARIZE.***
* NO USAR PARA ESTO: Crear columnas nuevas en la tabla evaluando una expresión
* (NO USAR PARA ESTO): Permitir generar subtotales

**Características de SUMMARIZE**

* Es una función que genera doble contexto a la vez: de filtro y de fila.
* Realiza una operación clustering, lo que hace que agregue columnas no deseadas al contexto de filtro***. Por eso no usar para traer valores de otras tablas***.

**Ejercicio**: Muestra una tabla cuáles son las marcas diferentes

***Forma incorrecta porque trae valores y puede tener columnas no deseadas:***

***Forma correcta de cálculo: agregar columna extra fuera de SUMMARIZE***

*# Argumento es una tabla*

*# Expresión de la otra columna extra*

Al agregar ADDCOLUMN evitamos problemas con la función SUMMARIZE.

1. **SUMMARIZE en medidas**

**Ejercicio**: Cuál fue el último mes donde tengo data o última fecha de orden

# MAXX recibe una tabla creada por SUMMARIZE

# tabla de hechos

# periodos en los que hay ventas registrado en tabla hechos (agrupa por)

# argumento a evaluar, ósea el que será el resultado.

Cuando coloco la tabla de hechos solo me muestra solo los periodos donde hubo ventas***. Y esto sale debido a la tabla expandida***. Una tabla expandida es una tabla que tiene una tabla original más las columnas que vienen de la parte individual de la relación ósea se agrega toda la tabla de dimensión (hechos + dimensión = tabla expandida). De esa tabla expandida sale una tabla virtual solo con los periodos distintos donde hubo ventas a la que se aplica la agregación MAXX.

1. **SUMMARIZE en medidas continuación**

**Ejercicio**: Encontrar el monto promedio diario por cliente

# tabla virtual recibida agrupada por fechas y clientes diferentes

# tabla original

# columna por la que se agrupara la tabla original

# columna por la que se agrupará la tabla original

# saca el monto de ventas de la tabla virtual de SUMMARIZE agrupados

# Calcula el promedio con los montos agrupados.

Los pasos en PBI son:

1. **Contexto de filtro**: Viene de la matriz y es mes igual a enero o el mes
2. **Aplicar contexto de filtro**: Aplicar filtro
3. **Evaluar fórmula**: a excepción de CALCULATE todas las funciones se evalúan de adentro hacia afuera.
4. **Summarize**: recibe tabla virtual expandida
5. **Agrupación**: Summarize genera una agrupación por columna brindada
6. **Evaluar expresión**: Se evalúa la expresión en contexto de fila (hay transición) y la fila de la tabla virtual se convierte en filtro.
7. **Lleno de tabla virtual**: se llena la tabla virtual con los valores (suma por summarize)
8. **AERAGE**: Recibe la tabla virtual llena
9. **Evaluación de expresión**: Se evalúa expresión en contexto de fila
10. **Se evalúa AVERAGE**: Calcula el promedio de los valores de la tabla virtual
11. **Llevar a Dashboard**: Muestra los valores deseados.

**Ejemplo**: Monto de venta máximo diario por cliente

1. **SUMMARIZE MEDIDAS**

Ejercicio: Obtener el mejor cliente

# retorna tabla virtual con clientes diferentes que han comprado, agrupado.

# expresión condicional de tabla anterior.

# recibe tabla en este caso mejor cliente

# concatena expresión por coma.

Esta técnica es muy utilizada con el uso de variables. Se usa para obtener:

* Mejor cliente
* Mejor producto
* Mejor ciudad
* Top 5 Categorías
* Clientes recurrentes
* Clientes nuevos
* Etc

1. **SUMMARIZECOLUMNS**

Esta pensado ser el reemplazo de SUMMARIZE. Es una función de tabla y realiza las mismas 3 funciones de SUMMARIZE. Pero con esta función si es posible crear expresiones como columnas adicionales sin ningún riesgo. Está más optimizada que SUMMARIZE.

***Pero tiene un problema: Todavía no es posible usarlo en medidas. Porque no soporta transición de contexto.***

En tablas calculadas siempre es mejor usar SUMMARIZECOLUMNS

**Ejercicio**: Obtener las marcas y el monto de ventas:

**CLASE 7: ALL, ALLEXCEPT Y ALLSELECTED**

Estas expresiones nos permiten ignorar o remover el filtro de una expresión. Tiene naturaleza dual: Como tabla y como modificar de CALCULATE.

1. **Sintaxis de ALL, ALLEXCEPT Y ALLSELECTED: USO EN TABLA**

Sintaxis básicas:

ALL (  
 Tabla o columna,

Columna agrupada 1,

Columna agrupada 2,

Etc.

)

**a.** **Cómo función de tabla** ***retorna todas las filas de una tabla***, o todos ***los valores distintos de una columna ignorando los filtros*** que le hayan sido aplicados.

**b. Cómo modificador de CALCULATE** ***remueve los filtros correspondientes del contexto filtro***, sin materializar directamente la tabla resultante.

**Ejercicio**: Crear una tabla con los valores distintos de la columna color de la tabla productos.

#ignora los filtros

La diferencia con VALUES es que **VALUES si considera el contexto de filtro, mientras que ALL no**. Pero en tablas calculadas ambas retornan lo mismo. Además, ***ALL puede tener más de una columna.***

Con la función ALLEXCETP puedo tomar una dimensión o tabla, pero sin las columnas que le indique:

#Tabla

#Columna no retornada

#Columna no retornada

ALLSELECTED crea una tabla que ignora los filtros internos que se le apliquen, los externos si los considera.

1. **ALL, ALLEXCEPT y ALLSELECTED en medidas de porcentaje**

***Ejemplo****:* ***Crear medida que muestre la participación porcentual por subcategoría.***

Primero calculamos el total de las columnas:

# devuelve la suma sin filtro, lo mismo para todas las filas.

# recibe toda la tabla copiada ignorando filtros

# operación en tabla completa sin filtro

Luego dividimos por cada valor y la suma del total de la columna:

#medida uno

#medida dos

El proceso en Power BI es:

1. Contexto de filtro: Es la subcategoría o filas de la matriz
2. Filtro del modelo: se filtra la tabla de hecho
3. Aplicar la fórmula: Se aplica la formula en la tabla hechos filtrada.
4. SUMX recibe una tabla: Recibe tabla sin filtro por ALL
5. Evaluar segundo argumento: Evaluar en contexto de fila
6. Se ejecuta la agregación: se suma todos los resultados.
7. Se lleva a la columna: lo guarda en la columna fantasma o dashboard.
8. Medida original: se aplica la división en la medida participación.

Pero este tiene un problema porque no toma en cuenta los filtros debido a ALL, entonces si aplicamos filtros externos no tomara el total resultante, sino el total sin filtros. Esto se arregla usando ALLSELECTED:

#Tabla completa afectada por filtros externos (más contextos fe filtro)

#Operacion con filtros externos

En Esta forma el filtro son los filtros externos de la vista informe. Porque ALLSELECTED ignora cualquier filtro interno solo considera los que son externos a la visual.

1. **ALL, ALLEXCEPT Y ALLSELECTED dentro de CALCULATE: como modificador**

La forma de table de ALL se usa bastante con funciones como RANK y TOPN, así como en construcciones de FILTER+ALL. Pero no es manera más común, la forma más recomendada es usar dentro de CALCULATE. ES MEJOR USAR CALCULATE PARA PROCENTAJES QUE LAS MEDIDAS ANTERIORES.

# Aplica la formula de suma de columna monto a tabla sin filtro (2do)

# Remueve o elimina el filtro del contexto y evalúa tabla completa (1ro)

En power BI funciona así:

1. **Contexto de filtro**: Es la subcategoría de la matriz
2. **ALL elimina filtros que afectan a la tabla hechos**: ALL en CALCULATE no solo ignora los filtros si no que los remueve del contexto.
3. **Contexto de filtro:** El contexto de filtro se vuelve vacío por ALL.
4. **Aplicamos contexto de filtro**: No se aplica ninguno.
5. **Aplicamos la formula**: sumar la columna no filtrada
6. **Llevar a dashboard**.

Pero para que tome los filtros externos tomamos la función ALLSELECTED:

# aplica formula sin filtros internos (2do)

# remueve filtros internos, externos los deja. (1ro)

Peso relativo = CALCULATE([Peso de pedidos], ALLSELECTED (OrderList)) (POR PLANTA)

Ahora los pasos son:

1. **Contexto de filtro**: tengo dos filtros de subcategoría y externo
2. **ALLSELECTED remueve** todos los filtros internos y considera filtro externo
3. **Aplicamos contexto de filtro**: solo del filtro externo
4. **Aplicamos la formula**: a la columna filtrada
5. **Llevar a dashboard**.

En CALCULATE ***los modificadores se evalúan primero*** y después los demás argumentos.

1. **ALL + VALUES para restaurar filtros**

Ejercicio: que las columnas de la participación sumen 100%

# aplica formula suma con el nuevo filtro de VALUES

# es un modificador, no una tabla (1ro): elimina filtros y usa tabla completa

# esto es una tabla (2do): retorna contexto de filtro nuevo

Retorna lo correcto porque ALL es un modificador y VALUE es una tabla y cómo es ***modificador se evalúa primero*** y luego el filtro de la tabla.

Los pasos en Power BI son:

1. Contexto de filtro: Es de la columna y filas
2. Aplica ALL: ALL elimina los filtros que afectan a la tabla hechos
3. Contexto de filtro: Es ahora vacio
4. Aplicar VALUES: Considera el contexto de filtro original no el que se borra, en este caso es año.
5. Contexto de filtro: se agrega la tabla de VALUES como nuevo filtro, VALUES restaura el filtro.
6. Aplicar nuevo filtro: Se filtra en este caso por año
7. Evalúa la formula: se aplica la suma
8. Aplica la división
9. Mostrar resultado
10. **ALLEXCEPT EN COLUMNAS CALCULADAS**

**Ejercicio**: Numero de registros por cada cliente

Los pasos en PBI:

1. Contexto de filtro: vacío por ser columna calculada, pero hay transición de contexto por CALCULATE, eso quiere decir que toda la fila se agrega al contexto de filtro.
2. Aplica ALLEXCEPT: Elimina los filtros que afecta a una tabla, excepto los especificados en los siguientes argumentos.
3. Aplicar filtro: Solo se queda con filtro de cliente
4. Evaluar la formula: cuenta cuantos registros hay.

**CLASE 8: FILTROS COMPLEJOS EN CALCULATE**

La Sintaxis de CALCULATE es:

# Obligatoria

CALCULATE ***evalúa una expresión en un contexto de filtro modificado***. Es decir, agregar filtro, quitar filtro o reemplazar filtro. También permite generar una transición de contexto. La familia de funciones ALL son modificadores de filtro. La función CALCULATE se evalúa en 5 pasos.

1. **Ejemplos de CALCULATE. Filtros como tablas, todo filtro se expresa como tabla.**

Primero creamos una medida sobre **el monto de ventas de los clientes hombres que tienen un hijo**. Eso lo hacemos con CALCULATE. De esa manera el contexto de filtro será categoría de productos, género y cantidad de hijos. ***Todos los filtros son tablas.***

#una tabla que se agrega al filtro

# Es una tabla

**Nota: Siempre entender la segunda fila de CALCULATE como si fuera una tabla.**

***Ejercicio dos es encontrar el monto de venta de clientes que sean del continente de Asia y de sexo masculino***.

*Monto de venta continente Asia o masculino* =

#Tabla

#Tabla

1. **Construcción de filtros compuestos con SUMMARIZE**

***El primer ejemplo es obtener el monto de ventas de clientes de Asia o tienda en Asia.***

1. **Construcción de Filtros con FILTER + VALUES**

***Medida monto de venta en primeros 60 días de venta.***

*#fecha mínima de acuerdo a filtro matriz*

*# tabla con distintos de fecha de orden*

*# evalua fila a fila*

*# evalua fila a fila*

*# calcula la suma con nuevos filtros*

*#tabla con filtros y es nuevo contexto de filtro*

I*dealmente en la variable Tabla Filtro se usa mejor y más eficientemente la dimensión calendario y fecha*, en vez de fecha de orden de la tabla de hechos de venta.

1. **Construcción de filtros con Transición de contexto**

***Queremos saber el número de clientes que compro más de un producto por fecha. (avanzado)***

*# recibe tabla*

*# avalua condición*

1. **Construcción de filtros con tablas de hechos (AVANZADO)**

Son útiles cuando el modelo no tiene dimensiones. ***Medida que muestre cuantas ordenes tienen un registros.***

**CLASE 9: REAL VS PRESUPUESTO**

1. Introducción a KEEPFILTER

Cuando en una matriz o tabla visual hay un contexto de filtro (ej. Color amarillo) y en CALCULATE se le coloca otro filtro de color, pero diferente (Ej color Blanco) siempre predomina el filtro de CALCULATE y se sobrescribe el resultado. Es decir, CALCULATE tiene más predominancia.

# predomina el filtro y se sobrescribe en objeto visual

Para modificar ese comportamiento se usa la función KEEPFILTER. Esta función evita que CALCULATE reemplace al filtro original. Y solo retorna las filas donde el filtro cumple la condición. De esa manera coexisten ampos filtros.

:

Ejercicio: Calcular en una misma gráfica el monto real de ventas con el presupuesto.

