En Power BI, tanto las \*\*medidas\*\* como las \*\*columnas calculadas\*\* se utilizan para realizar cálculos, pero tienen propósitos diferentes y se comportan de maneras distintas. Aquí están las diferencias clave:

### 1. \*\*Contexto de Cálculo\*\*

- \*\*Medidas\*\*: Las medidas se calculan en el \*\*contexto de filtro\*\*. Esto significa que el resultado de una medida puede cambiar dependiendo de los filtros aplicados en un informe, como los slicers, filtros visuales o el contexto de fila en una tabla dinámica. Por ejemplo, una medida para calcular el total de ventas puede mostrar un resultado diferente según el año o el producto seleccionado.

- \*\*Columnas Calculadas\*\*: Las columnas calculadas se calculan en el \*\*contexto de fila\*\*. Esto significa que el cálculo se realiza una vez por cada fila en la tabla, y el resultado se almacena como parte de los datos. Una columna calculada no cambia con el contexto de filtro en los informes, ya que el valor es estático una vez calculado.

### 2. \*\*Almacenamiento\*\*

- \*\*Medidas\*\*: No ocupan espacio adicional en el modelo de datos porque no se almacenan; solo se calculan cuando se necesitan en un informe.

- \*\*Columnas Calculadas\*\*: Ocupan espacio en el modelo de datos porque los valores calculados se almacenan físicamente en la tabla. Esto puede aumentar el tamaño del archivo del modelo de datos.

### 3. \*\*Uso\*\*

- \*\*Medidas\*\*: Son ideales para cálculos agregados como sumas, promedios, mínimos, máximos, conteos, y otros cálculos que dependen del contexto de filtro.

- \*\*Columnas Calculadas\*\*: Son útiles para cálculos que deben ser parte de la tabla de datos y que no dependen del contexto de filtro. Por ejemplo, si necesitas crear una nueva columna que clasifica los productos según su categoría, una columna calculada sería la opción correcta.

### 4. \*\*Flexibilidad\*\*

- \*\*Medidas\*\*: Son más flexibles porque se pueden utilizar en cualquier parte del informe y se recalculan dinámicamente según los filtros y las interacciones en el informe.

- \*\*Columnas Calculadas\*\*: Son menos flexibles en comparación con las medidas, ya que el cálculo se realiza una vez y se almacena.

### 5. \*\*Ejemplo\*\*

- \*\*Medida\*\*:

```DAX

TotalVentas = SUM(Ventas[Cantidad])

```

Aquí, `TotalVentas` cambiará dependiendo del contexto de filtro (por ejemplo, año, región, producto).

- \*\*Columna Calculada\*\*:

```DAX

Margen = Ventas[Precio] - Ventas[Costo]

```

Esta columna calculada `Margen` se evaluará para cada fila individual en la tabla `Ventas` y el valor será constante, independientemente del contexto de filtro en los informes.

### Resumen:

- \*\*Medidas\*\*: Dinámicas, calculadas en tiempo de ejecución, dependen del contexto de filtro, no ocupan espacio adicional.

- \*\*Columnas Calculadas\*\*: Estáticas, calculadas en el contexto de fila, ocupan espacio en el modelo, útiles para cálculos que necesitan ser parte de los datos originales.

La función SELECTEDVALUE en DAX se utiliza para devolver el valor seleccionado de una columna en un contexto de filtro. Es especialmente útil cuando solo hay un valor en el contexto y quieres capturar ese valor, pero también permite manejar casos en los que hay múltiples valores seleccionados.

### Uso Común

* **Capturar un valor único en contexto**: Ideal para situaciones donde esperas que solo haya un valor en el contexto de filtro y quieres utilizar ese valor en cálculos o visualizaciones.
* **Mostrar mensajes personalizados**: Puedes proporcionar una alternativa clara cuando el contexto de filtro no cumple con las expectativas, como cuando hay múltiples valores seleccionados.

### Resumen:

SELECTEDVALUE es una función muy útil en DAX para manejar contextos de filtro donde esperas que solo haya un valor seleccionado. Te permite capturar ese valor y también manejar casos en los que no se selecciona ningún valor o se seleccionan múltiples valores.

All selected

La función ALLSELECTED en DAX se utiliza para devolver todos los valores de una columna o tabla, pero tiene en cuenta los filtros aplicados en el nivel visual (como en una tabla o gráfico) y en el nivel de informe (como slicers o filtros de página). Es útil cuando necesitas calcular medidas que respeten ciertos filtros pero ignoren otros, particularmente en escenarios donde se trabaja con totales parciales o se desea comparar valores filtrados con totales.

### Ejemplos

1. **Calcular un porcentaje del total visible en un visual:**

Supongamos que tienes un gráfico de barras que muestra las ventas por categoría de producto. Quieres calcular el porcentaje de ventas de cada categoría en relación con el total de ventas visible en el gráfico, considerando los filtros aplicados en el nivel de visualización.

DAX

Copiar código

PorcentajeVentasVisible =

DIVIDE(

SUM(Ventas[Cantidad]),

CALCULATE(SUM(Ventas[Cantidad]), ALLSELECTED(Ventas[Categoría]))

)

Aquí, ALLSELECTED elimina los filtros solo en la columna Categoría para obtener el total de ventas visible en el gráfico. El resultado es el porcentaje de ventas de cada categoría respecto al total visible.

### Uso Común

* **Porcentaje del total visible**: Es útil para calcular porcentajes o proporciones que deben tener en cuenta los filtros aplicados en el visual, pero no necesariamente los filtros más específicos.
* **Escenarios de total acumulado**: ALLSELECTED es ideal para comparar datos filtrados con el total visible o para hacer cálculos que respeten solo ciertos filtros.

### Resumen:

ALLSELECTED es una función clave para manejar contextos de filtro en DAX, especialmente cuando necesitas trabajar con totales visibles que respetan ciertos filtros, como los aplicados a un visual o a un informe en Power BI.

Si tienes algún ejemplo específico en mente o necesitas más detalles sobre cómo aplicar ALLSELECTED, estaré encantado de ayudarte.

La función COALESCE en DAX es útil para manejar valores nulos o faltantes en tus datos. Devuelve el primer valor no nulo de una lista de expresiones. Es especialmente útil cuando trabajas con datos que pueden tener valores faltantes o nulos, y quieres asegurar que siempre se devuelva un valor válido.

La función COALESCE en Power BI es una herramienta poderosa para manejar datos faltantes y asegurar que tus cálculos siempre devuelvan un valor válido, evitando errores o resultados inesperados.

Si tienes algún escenario específico donde te gustaría aplicar COALESCE, puedo ayudarte a implementarlo.

La función TREATAS en DAX es una herramienta avanzada que te permite aplicar un conjunto de valores como filtros en una o más columnas de una tabla, aunque esas columnas no estén directamente relacionadas. Esto es útil en escenarios donde deseas simular una relación entre tablas que no están relacionadas en tu modelo de datos.

**Cómo Funciona**

TREATAS toma los valores de la tabla proporcionada en el primer argumento y los trata como si fueran valores de las columnas especificadas en el resto de los argumentos. Luego aplica estos valores como filtros en el contexto de la consulta.

### Uso Común

* **Simular relaciones**: TREATAS es ideal para situaciones donde no existe una relación física entre tablas en tu modelo de datos, pero necesitas aplicar un filtro basado en valores de otra tabla.
* **Cálculos avanzados**: Permite realizar cálculos complejos y específicos donde se necesita filtrar datos basados en columnas que no están directamente relacionadas.

### Resumen:

La función TREATAS es una herramienta poderosa para aplicar filtros basados en valores de otras tablas en Power BI, lo que te permite realizar cálculos avanzados y específicos en situaciones donde las tablas no están directamente relacionadas.

La función `RANKX` en DAX se utiliza para calcular el rango (posición) de un valor en una tabla o conjunto de datos, basándose en una expresión o columna específica. Es útil para crear rankings, como clasificar productos por ventas, clientes por ingresos, entre otros.

### Sintaxis

```DAX

RANKX(<table>, <expression>, [<value>], [<order>], [<ties>])

```

- \*\*table\*\*: Es la tabla o conjunto de datos en el que se calculará el rango.

- \*\*expression\*\*: La expresión o columna que se utiliza para calcular el rango.

- \*\*value\*\* (opcional): Especifica el valor en la fila actual (normalmente se deja vacío para el contexto de la fila).

- \*\*order\*\* (opcional): Define el orden del ranking: `DESC` (descendente, predeterminado) o `ASC` (ascendente).

- \*\*ties\*\* (opcional): Especifica cómo manejar empates: `SKIP` (predeterminado, salta posiciones) o `DENSE` (no salta posiciones en caso de empate).

### Ejemplos

1. \*\*Clasificar productos por ventas:\*\*

Si deseas clasificar productos según las ventas totales:

```DAX

RankingVentas = RANKX(

ALL(Productos),

SUM(Ventas[Cantidad]),

,

DESC,

SKIP

)

```

En este ejemplo:

- `ALL(Productos)` asegura que el ranking se haga entre todos los productos, ignorando los filtros aplicados.

- `SUM(Ventas[Cantidad])` es la expresión que calcula las ventas totales por producto.

- `DESC` organiza el ranking de mayor a menor.

- `SKIP` maneja los empates saltando posiciones.

### Uso Común

- \*\*Clasificaciones dinámicas\*\*: `RANKX` es ideal para crear rankings que se actualizan automáticamente según los filtros aplicados en el informe.

- \*\*Comparaciones entre grupos\*\*: Permite crear rankings dentro de subgrupos, como productos dentro de categorías, empleados dentro de departamentos, etc.

### Resumen:

La función `RANKX` es una herramienta poderosa en DAX para calcular posiciones o rankings en cualquier conjunto de datos. Ofrece flexibilidad en la forma de ordenar los valores y cómo manejar los empates.

Top N en Power BI se refiere a la creación de visualizaciones o cálculos que muestren solo los primeros N elementos de un conjunto de datos según algún criterio, como las ventas más altas, los clientes más rentables, etc. Esto se puede lograr usando la función DAX TOPN o mediante filtros en visuales.

**Uso de la función TOPN en DAX**

La función TOPN devuelve las primeras filas de una tabla en función de un ordenamiento especificado.

**Sintaxis**

DAX

Copiar código

TOPN(<n\_value>, <table>, <orderBy\_expression>, [<order>], ...)

* **n\_value**: El número de filas que deseas devolver.
* **table**: La tabla de la que se extraerán las filas.
* **orderBy\_expression**: La expresión que se utiliza para ordenar las filas.
* **order** (opcional): El orden de la clasificación (DESC o ASC). El valor predeterminado es DESC (descendente).

**Ejemplos**

1. **Obtener los 5 productos más vendidos:**

Si deseas obtener los 5 productos con mayor cantidad de ventas:

DAX

Copiar código

Top5Productos =

TOPN(

5,

Productos,

SUM(Ventas[Cantidad]),

DESC

)

En este ejemplo:

* + 5 es el número de productos que quieres.
  + Productos es la tabla de la cual obtendrás los productos.
  + SUM(Ventas[Cantidad]) es la expresión que calcula las ventas totales por producto.
  + DESC asegura que se ordenen de mayor a menor cantidad de ventas

Resumen:

Top N es una técnica poderosa en Power BI para enfocarse en los elementos más importantes de un conjunto de datos, ya sea utilizando la función TOPN en DAX o aplicando filtros directamente en visualizaciones.

------------------------------------------------------------------

1. **CALCULATE(Medidas[Total Venta], DATESYTD('Dim FechaPedido'[Fecha]))**:
   * CALCULATE se utiliza para modificar el contexto de filtro que se aplica a una expresión. En este caso, la expresión es Medidas[Total Venta], que representa el total de ventas.
   * DATESYTD('Dim FechaPedido'[Fecha]) es una función de inteligencia de tiempo que devuelve un conjunto de fechas desde el inicio del año hasta la fecha actual en el contexto del filtro. Este conjunto de fechas se basa en la columna Fecha de la tabla Dim FechaPedido.

Es importante mencionar que:

* + **Dim FechaPedido** debe ser una tabla de fechas (un calendario) que contenga una columna Fecha.
  + La función DATESYTD asume que el año fiscal comienza el 1 de enero a menos que se especifique de otra manera.

En resumen, la medida Acumulado Ventas anuales (DATESYMD) calcula la suma acumulada de Medidas[Total Venta] desde el comienzo del año hasta la fecha seleccionada en el contexto actual, utilizando la columna de fechas Fecha en la tabla Dim FechaPedido. Esto te permite ver cómo se han acumulado las ventas a lo largo del año hasta la fecha seleccionada en tus visualizaciones.