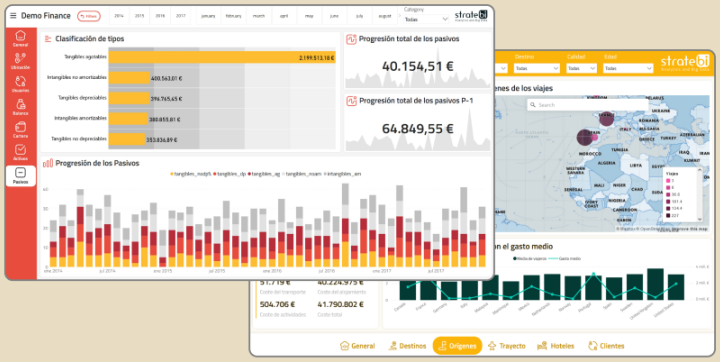
PowerBI Tips Vol. 5

NUEVO 2025

PowerBI

Más de 20 nuevos trucos, explicados paso a paso

1. CONEXIÓN y transformación de DATOS | 1.1. Convertir lista de Strings en una tabla con DAX – 1.2. Reordenar dinámicamente las columnas en Power Query – 1.3. Excluir las N primeras filas – 1.4. Solucionar errores al pivotar una tabla – 1.5. Importar varios ficheros Excel ubicados en Sharepoint – 1.6. Obtener datos de Dynamics a través de API – **2. MODELADO |** 2.1. Calcular la media de N meses anteriores – 2.2. Utilizar una medida como segmentación en Power BI – 2.3. Aproximación a la funcionalidad Seguridad por página – 2.4. Localizar problemas de integridad referencial – 2.5. No mostrar registros sin datos en una tabla – 2.6. Localizar duplicados desde la vista de consultas DAX – **3. VISUALIZACIÓN |** 3.1. Mostrar una medida en unidades o en miles con un slicer – 3.2. Limitar una tabla según el valor seleccionado – 3.3. Resaltar valores de una tabla según el valor seleccionado – 3.4. Añadir formato condicional a un gráfico de líneas – 3.5. Crear una jerarquía en un slicer con field parameters – **4. OTROS |** 4.1. Ordenar un objeto visual tipo tabla por dos columnas – 4.2. Formatear 0 y 1 – 4.3. Utilizar los campos y los valores de los campos en Field parameters – 4.4. Funcionalidades de la vista DAX Query – 4.5. Cómo utilizar Copilot en Power BI – **5. Más trucos, videotutoriales y papers**



stratebi
Analytics and Big Data

CONTENIDO

1. CONEXIÓN Y TRANSFORMACIÓN DE DATOS 3

1. CONVERTIR LISTA DE STRINGS EN UNA TABLA CON DAX 3
2. REORDENAR DINÁMICAMENTE LAS COLUMNAS EN POWER QUERY 7
3. EXCLUIR LAS N PRIMERAS FILAS 11
4. SOLUCIONAR ERRORES AL PIVOTAR UNA TABLA 20
5. IMPORTAR VARIOS FICHEROS EXCEL UBICADOS EN SHAREPOINT 23
6. OBTENER DATOS DE DYNAMICS A TRAVÉS DE API 25

2. MODELADO 28

7. CALCULAR LA MEDIA DE N MESES ANTERIORES 28
8. UTILIZAR UNA MEDIDA COMO SEGMENTACIÓN EN POWER BI 35
9. APROXIMACIÓN A LA FUNCIONALIDAD SEGURIDAD POR PÁGINA 38
10. LOCALIZAR PROBLEMAS DE INTEGRIDAD REFERENCIAL 43
11. NO MOSTRAR REGISTROS SIN DATOS EN UNA TABLA 46
12. LOCALIZAR DUPLICADOS DESDE LA VISTA DE CONSULTAS DAX 49

3. VISUALIZACIÓN 51

13. MOSTRAR UNA MEDIDA EN UNIDADES O EN MILES CON UN SLICER 51
14. LIMITAR UNA TABLA SEGÚN EL VALOR SELECCIONADO 54
15. RESALTAR VALORES DE UNA TABLA SEGÚN EL VALOR SELECCIONADO 60
16. AÑADIR FORMATO CONDICIONAL A UN GRÁFICO DE LÍNEAS 63
17. CREAR UNA JERARQUÍA EN UN SLICER CON FIELD PARAMETERS 67

4. OTROS 71

18. ORDENAR UN OBJETO VISUAL TIPO TABLA POR DOS COLUMNAS 71
19. FORMATEAR 0 Y 1 73
20. UTILIZAR LOS CAMPOS Y LOS VALORES DE LOS CAMPOS EN FIELD PARAMETERS 74
21. FUNCIONALIDADES DE LA VISTA DAX QUERY 77
22. CÓMO UTILIZAR COPILOT EN POWER BI 81

5. MAS TRUCOS, VIDEOTUTORIALES Y PAPERS 84

6. CURSOS 86

1. CONEXIÓN Y TRANSFORMACIÓN DE DATOS

1. Convertir lista de Strings en una tabla con DAX

El propósito de este tip es mostrar cómo se puede convertir una lista de Strings en tabla utilizando DAX.



Desarrollo:

Con este ejemplo lo que se pretende es simular la funcionalidad de la función **Table.FromList** de M, pero con lenguaje DAX. Además, veremos cómo, a partir de esta solución, se pueden filtrar los valores de un campo de otra tabla.

Partiendo de una lista de valores en un string, para obtener una tabla con una fila por cada elemento de la lista, se podrían hacer lo siguiente:

- Utilizar "|" como separador de los elementos del string.
- Utilizar la función PATHLENGHT para determinar el número de elementos del string.
- Utilizar la función GENERATESERIES iterando sobre la cadena para crear una fila para cada elemento
- Utilizar la función PATHITEM para obtener el elemento que se está utilizando y usar SELECTCOLUMNS para devolver el resultado.

Con estas indicaciones, la tabla a crear sería de la siguiente forma:

```
String into table =

VAR list = "1234|4567|7890|9876|6543"

VAR _length =
    PATHLENGTH ( list )

VAR Result =
    SELECTCOLUMNS (
        GENERATESERIES ( 1, _length ),
        "Values", PATHITEM ( list, [value], INTEGER )
    )

RETURN
    Result
```

Obteniendo como resultado la siguiente tabla:

Values
1234
4567
7890
9876
6543

Para utilizar esta tabla **como filtro** de un campo de otra, se crea una nueva tabla similar a la anterior, pero con una variable añadida (_filter):

```
String into table filter =

VAR list = "4567|9876"

VAR _length =
    PATHLENGTH ( list )

VAR _pivot =
    SELECTCOLUMNS (
        GENERATESERIES ( 1, _length ),
        "Value", PATHITEM ( list, [value] )
    )

VAR _filter =
    TREATAS ( _pivot, 'String into table'[Values] )

RETURN
    _filter
```

Como se puede observar, se está filtrando la tabla **String into table** por los dos valores que contiene la tabla de filtro, obteniendo como resultado:

Values
4567
9876

Una posible **mejora** sería utilizar el **constructor** de tabla para crear la tabla de filtro (en formato string o en formato número).

Constructor en formato **string**:

```
String into table filter constructor =

VAR _filter =
    -- strings
TREATAS (
    { "1234", "7890" },
    'String into table'[Values]
)

-- numbers
/*TREATAS (
    { 1234, 7890 },
   'String into table'[Values]
)*/

RETURN
    _filter
```

Values
1234
7890

Constructor en formato **numérico**:

```
String into table filter constructor =

VAR _filter =
    -- strings
/*TREATAS (
    { "1234", "7890" },
    'String into table'[Values]
)*/

-- numbers
TREATAS (
    { 1234, 7890 },
   'String into table'[Values]
)

RETURN
    _filter
```

Values
1234
7890

2. Reordenar dinámicamente las columnas en Power Query

El propósito de este tip es mostrar cómo se pueden reordenar dinámicamente las columnas en Power Query, manteniendo un orden fijo de las columnas que sea necesario.



Desarrollo:

Puede haber algunos escenarios en los que se necesite un orden concreto de las columnas de una tabla. Aunque este orden no afecte al modelo en Power BI, puede darse el caso de que se necesite mantener un orden concreto en otros procesos de automatización donde el orden sí que sea relevante.

En primer lugar, vamos a crear una nueva tabla con el nombre de los campos de la tabla que queremos reordenar y un campo para asignar la posición de estos.

Para ello, desde el editor de **Power Query en Excel**, creamos una consulta en y vamos a utilizar como origen un **paso personalizado** en el que vamos a utilizar la función **Table.ColumnNames** para obtener el nombre de los campos de la tabla DimGeography.

```
= Table.ColumnNames(DimGeography)
```

	Lista
1	GeographyKey
2	CountryRegionCode
3	EnglishCountryRegionName
4	StateProvinceCode
5	StateProvinceName
6	City
7	SpanishCountryRegionName
8	FrenchCountryRegionName
9	PostalCode
10	SalesTerritoryKey
11	DimCustomer
12	DimReseller
13	DimSalesTerritory

A continuación, seleccionamos la opción **Cerrar y cargar**. En este momento, desde Excel, vamos a añadir un nuevo campo en la tabla creada llamado "**Position**" con el número de orden que queremos asignar a cada campo.

Asignamos el orden a los campos y, si hay algún campo al que no queramos asignarle una posición en concreto, simplemente seguimos la secuencia:

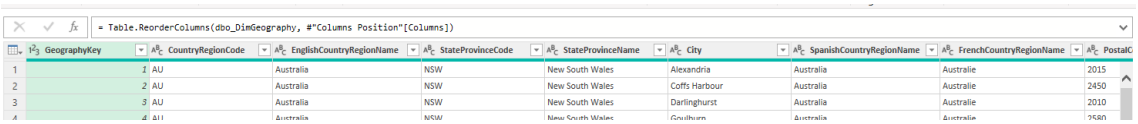
Columns	Position
GeographyKey	1
City	6
StateProvinceCode	4
StateProvinceName	5
CountryRegionCode	2
EnglishCountryRegionName	3
SpanishCountryRegionName	7
FrenchCountryRegionName	8
PostalCode	9
SalesTerritoryKey	10
DimCustomer	11
DimReseller	12
DimSalesTerritory	13

Ahora, volvemos a abrir **Power Query desde Excel** (Pestaña Datos -> De una tabla o rango) y, como se puede observar, tenemos una consulta nueva creada que incluye el nuevo campo con el orden. Vamos a renombrar esta nueva query como **Columns Position**.

A continuación, ordenamos el campo **Position** de forma ascendente.

	Columns	Position
1	GeographyKey	1
2	CountryRegionCode	2
3	EnglishCountryRegionName	3
4	StateProvinceCode	4
5	StateProvinceName	5
6	City	6
7	SpanishCountryRegionName	7
8	FrenchCountryRegionName	8
9	PostalCode	9
10	SalesTerritoryKey	10
11	DimCustomer	11
12	DimReseller	12
13	DimSalesTerritory	13

Para reordenar los campos de la tabla **DimGeography**, añadimos un paso personalizado que haga uso de la función **Table.ReorderColumns** de la siguiente manera:



Como podemos ver, ya hemos conseguido el resultado deseado, establecer el orden de las columnas. Para que este orden se pueda editar dinámicamente, habría que editar desde **Excel** el campo **Position** y volver a cargar la tabla en Power Query siguiendo los pasos explicados.

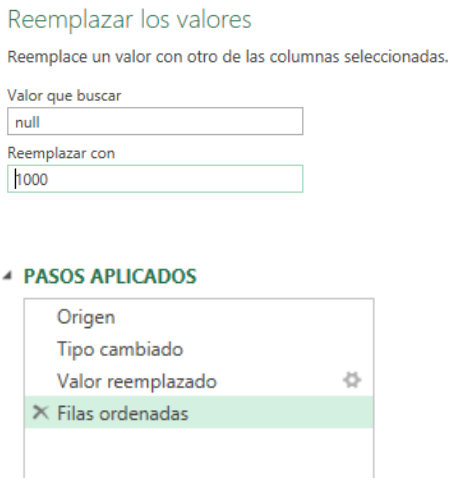
Otro caso posible es que se necesite un orden concreto de algunos campos pero que del resto no sea relevante. En este caso, se establecería el orden de aquellos campos que lo requieran y, en los demás, se deja el registro vacío.

Columns	Position
GeographyKey	1
City	6
StateProvinceCode	4
StateProvinceName	5
CountryRegionCode	2
EnglishCountryRegionName	3
SpanishCountryRegionName	
FrenchCountryRegionName	
PostalCode	
SalesTerritoryKey	
DimCustomer	
DimReseller	
DimSalesTerritory	

Si abrimos Power Query, veremos que estos registros tienen asignado el valor null en el campo **Position**.

	Columns	Position
1	DimReseller	null
2	SalesTerritoryKey	null
3	DimCustomer	null
4	SpanishCountryRegionName	null
5	PostalCode	null
6	DimSalesTerritory	null
7	FrenchCountryRegionName	null
8	GeographyKey	1
9	CountryRegionCode	2
10	EnglishCountryRegionName	3
11	StateProvinceCode	4
12	StateProvinceName	5
13	City	6

A continuación, reemplazamos estos valores null por un valor muy alto (en este ejemplo, 1000) y, desde el panel de la derecha de **Pasos aplicados**, movemos el paso "**Valor reemplazado**" justo encima del de "**Filas ordenadas**".



	Columns	1.2 Position
1	GeographyKey	1
2	CountryRegionCode	2
3	EnglishCountryRegionName	3
4	StateProvinceCode	4
5	StateProvinceName	5
6	City	6
7	DimSalesTerritory	1000
8	DimReseller	1000
9	DimCustomer	1000
10	SalesTerritoryKey	1000
11	PostalCode	1000
12	FrenchCountryRegionName	1000
13	SpanishCountryRegionName	1000

De esta forma, tenemos los 6 primeros campos con el orden requerido y los demás, con el orden predeterminado al importarlos.

3. Excluir las N primeras filas

El propósito de este tip es mostrar cómo se pueden excluir las N primeras filas de una tabla en Power BI con DAX.



Desarrollo:

Puede suceder que se quieran excluir las N primeras filas de una tabla para centrar el análisis en el resto. Por ejemplo, excluir los 5 modelos que tengan más ventas.

En primer lugar, vamos a añadir una visualización de tipo tabla con el nombre del modelo del producto y el importe de las ventas, ordenando el importe de mayor a menor, para identificar cuáles son los modelos que vamos a excluir.

ModelName	Sales Amount
Mountain-200	7.929.475,24
Road-150	5.549.896,77
Road-250	4.451.260,13
Touring-1000	2.992.007,85
Road-350-W	1.580.219,71
Road-550-W	1.514.622,36
Mountain-100	1.341.121,04
Road-750	779.205,57
Road-650	645.379,50
Touring-2000	451.924,20
Mountain-400-W	417.833,07
Touring-3000	400.869,00
Mountain-500	264.330,21
Sport-100	225.335,60
Long-Sleeve Logo Jersey	86.782,64
Short-Sleeve Classic Jersey	86.168,04
Women's Mountain Shorts	71.319,81
HL Mountain Tire	48.860,00
Fender Set - Mountain	46.619,58
Hydration Pack	40.307,67

Desde la vista de consultas DAX podemos ejecutar una consulta para ver el número de modelos diferentes que hay, en este caso, 120.

```
1    EVALUATE
2        VALUES(DimProduct[ModelName])
3
4
```

Results | Result 1 of 1 | Copy

	DimProduct[ModelName]
1	
2	HL Road Frame
3	Sport-100
4	Mountain Bike Socks
5	Cycling Cap
6	Long-Sleeve Logo Jersey
7	LL Road Frame
8	ML Road Frame
9	HL Mountain Frame
10	Road-150
11	Road-450
12	Road-650
13	Mountain-100
14	Mountain-200

Query 1 +

cess (71.4 ms) Query 1 of 1 Result 1 of 1 1 column, 120 rows

A continuación, en otra nueva consulta, vamos a calcular los 5 modelos que tienen mayor importe de ventas:

```
DEFINE
var _top5model =
TOPN(
    5,
    ALLSELECTED(DimProduct[ModelName]),
    [Sales Amount]
    )
EVALUATE
FILTER(
    VALUES(DimProduct[ModelName]),
    DimProduct[ModelName] IN (_top5model)
)
EVALUATE
_top5model
```

Results | Result 1 of 2 | Copy

	DimProduct[ModelName]
1	Road-150
2	Mountain-200
3	Road-250
4	Touring-1000
5	Road-350-W

El siguiente paso es calcular el importe de **Sales Amount** para cada modelo:

```
DEFINE
var _top5model =
TOPN(
    5,
    ALLSELECTED(DimProduct[ModelName]),
    [Sales Amount]
    )
EVALUATE
SUMMARIZECOLUMNS(
    DimProduct[ModelName],
    _top5model,
    "Sales", [Sales Amount]
)
ORDER BY [Sales Amount] DESC
```

Results | Result 1 of 1 | Copy

	DimProduct[ModelName]	[Sales]
1	Mountain-200	7929475.24
2	Road-150	5549896.77
3	Road-250	4451260.13
4	Touring-1000	2992007.85
5	Road-350-W	1580219.71

Para quitar estos registros de la tabla, modificamos la consulta anterior creando un nuevo campo (**SalesTop5**):

```
DEFINE
var _top5model =
TOPN(
    5,
    ALLSELECTED(DimProduct[ModelName]),
    [Sales Amount]
    )
EVALUATE
SUMMARIZECOLUMNS(
    DimProduct[ModelName],
    //_top5model,
    "Sales", [Sales Amount],
    "SalesTop5", CALCULATE([Sales Amount], KEEPFILTERS(NOT(DimProduct[ModelName] IN (_top5model))))
)
ORDER BY [Sales Amount] DESC
```

Results | Result 1 of 1 | Copy

	DimProduct[ModelName]	[Sales]	[SalesTop5]
1	Mountain-200	7929475.24	
2	Road-150	5549896.77	
3	Road-250	4451260.13	
4	Touring-1000	2992007.85	
5	Road-350-W	1580219.71	
6	Road-550-W	1514622.36	1514622.36
7	Mountain-100	1341121.04	1341121.04
8	Road-750	779205.57	779205.57
9	Road-650	645379.5	645379.5
10	Touring-2000	451924.2	451924.2
11	Mountain-400-W	417833.07	417833.07
12	Touring-3000	400869	400869
13	Mountain-500	264330.21	264330.21
14	Sport-100	225335.6	225335.6

Con esto, no estamos eliminando los registros como tal, sino que estamos haciendo que no se muestre ningún valor. Sin embargo, si comentamos el campo **Sales** en la consulta, ahora sí que no se muestran los 5 primeros valores.

```
DEFINE
var _top5model =
TOPN(
    5,
    ALLSELECTED(DimProduct[ModelName]),
    [Sales Amount]
    )
EVALUATE
SUMMARIZECOLUMNS(
    DimProduct[ModelName],
    //_top5model,
    //"Sales", [Sales Amount],
    "SalesTop5", CALCULATE([Sales Amount], KEEPFILTERS(NOT(DimProduct[ModelName] IN (_top5model))))
)
ORDER BY [Sales Amount] DESC
```

Results | Result 1 of 1 | Copy

	DimProduct[ModelName]	[SalesTop5]
1	Road-550-W	1514622.36
2	Mountain-100	1341121.04
3	Road-750	779205.57
4	Road-650	645379.5
5	Touring-2000	451924.2
6	Mountain-400-W	417833.07
7	Touring-3000	400869
8	Mountain-500	264330.21
9	Sport-100	225335.6
10	Long-Sleeve Logo Jersey	86782.64
11	Short-Sleeve Classic Jer...	86168.04
12	Women's Mountain Sho...	71319.81
13	HL Mountain Tire	48860
14	Fender Set - Mountain	46619.58

Por último, para convertir esta consulta en una medida, añadimos **DEFINE MEASURE** en la primera línea y hacemos la siguiente modificación:

```
Update model: Add new measure
DEFINE MEASURE'FactInternetSales'[Top5] =
var _top5model =
TOPN(
    5,
    ALLSELECTED(DimProduct[ModelName]),
    [Sales Amount]
    )
VAR _result = CALCULATE([Sales Amount], KEEPFILTERS(NOT(DimProduct[ModelName] IN (_top5model))))
RETURN _result
EVALUATE
SUMMARIZECOLUMNS(
    DimProduct[ModelName],
    //_top5model,
    //"Sales", [Sales Amount],
    "SalesTop5", [Top5]
    )
ORDER BY [Sales Amount] DESC
```

Results | Result 1 of 1 | Copy

	DimProduct[ModelName]	[SalesTop5]
1	Road-550-W	1514622.36
2	Mountain-100	1341121.04
3	Road-750	779205.57
4	Road-650	645379.5
5	Touring-2000	451924.2
6	Mountain-400-W	417833.07
7	Touring-3000	400869
8	Mountain-500	264330.21
9	Sport-100	225335.6
10	Long-Sleeve Logo Jersey	86782.64
11	Short-Sleeve Classic Jer...	86168.04
12	Women's Mountain Sho...	71319.81

Seleccionamos la opción **Añadir nueva medida** y ya podemos utilizarla en nuestra visualización de tabla.

ModelName	Sales Amount	Top5
Mountain-200	7.929.475,24	
Road-150	5.549.896,77	
Road-250	4.451.260,13	
Touring-1000	2.992.007,85	
Road-350-W	1.580.219,71	
Road-550-W	1.514.622,36	1.514.622,3575 €
Mountain-100	1.341.121,04	1.341.121,04 €
Road-750	779.205,57	779.205,5699999999 €
Road-650	645.379,50	645.379,5037999999 €
Touring-2000	451.924,20	451.924,2 €
Mountain-400-W	417.833,07	417.833,07 €
Touring-3000	400.869,00	400.869 €
Mountain-500	264.330,21	264.330,21 €
Sport-100	225.335,60	225.335,6 €
Long-Sleeve Logo Jersey	86.782,64	86.782,64 €
Short-Sleeve Classic Jersey	86.168,04	86.168,03999999999 €
Women's Mountain Shorts	71.319,81	71.319,81 €
HL Mountain Tire	48.860,00	48.860 €
Fender Set - Mountain	46.619,58	46.619,58 €
Hydration Pack	40.307,67	40.307,67 €
All-Purpose Bike Stand	39.591,00	39.591 €
Hitch Rack - 4-Bike	39.360,00	39.360 €
Classic Vest	35.687,00	35.687 €
Half-Finger Gloves	35.020,70	35.020,7 €
ML Mountain Tire	34.818,39	34.818,39 €
HL Road Tire	27.970,80	27.970,8 €
Touring Tire	27.105,65	27.105,65 €
ML Road Tire	23.140,74	23.140,74 €
LL Road Tire	22.435,56	22.435,56 €
LL Mountain Tire	21.541,38	21.541,38 €
Water Bottle	21.177,56	21.177,56 €
Mountain Bottle Cage	20.229,75	20.229,75 €
Cycling Cap	19.688,10	19.688,1 €
Total	**29.358.677,22**	**6.855.817,5213 €**

Como se puede observar en la imagen anterior, siguen apareciendo los 5 primeros modelos, pero esto es porque está el campo de **Sales Amount** en la tabla, pero si lo quitamos, ya obtenemos el resultado deseado:

ModelName	Top5
Road-550-W	1.514.622,3575 €
Mountain-100	1.341.121,04 €
Road-750	779.205,5699999999 €
Road-650	645.379,5037999999 €
Touring-2000	451.924,2 €
Mountain-400-W	417.833,07 €
Touring-3000	400.869 €
Mountain-500	264.330,21 €
Sport-100	225.335,6 €
Long-Sleeve Logo Jersey	86.782,64 €
Short-Sleeve Classic Jersey	86.168,03999999999 €
Women's Mountain Shorts	71.319,81 €
HL Mountain Tire	48.860 €
Fender Set - Mountain	46.619,58 €
Hydration Pack	40.307,67 €
All-Purpose Bike Stand	39.591 €
Hitch Rack - 4-Bike	39.360 €
Classic Vest	35.687 €
Half-Finger Gloves	35.020,7 €
ML Mountain Tire	34.818,39 €
HL Road Tire	27.970,8 €
Touring Tire	27.105,65 €
ML Road Tire	23.140,74 €
LL Road Tire	22.435,56 €
LL Mountain Tire	21.541,38 €
Water Bottle	21.177,56 €
Mountain Bottle Cage	20.229,75 €
Cycling Cap	19.688,1 €
Mountain Tire Tube	15.444,05 €
Road Bottle Cage	15.390,88 €
Road Tire Tube	9.480,24 €
Touring Tire Tube	7.425,12 €
Patch kit	7.307,39 €
Total	**6.855.817,5213 €**

4. Solucionar errores al pivotar una tabla

El propósito de este tip es mostrar cómo se pueden solucionar los errores al pivotar una tabla en Power Query.

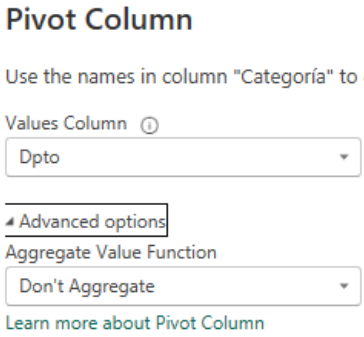


Desarrollo:

Partiendo de la siguiente tabla, podemos observar que hay algunas fechas y categorías repetidas, pero, sin embargo, tienen diferente valor en el campo Dpto.

Date	Dpto	Categoría
11/05/2024	Ventas	A
28/02/2024	Marketing	B
13/06/2024	RRHH	B
30/01/2024	RRHH	C
16/09/2024	Ventas	A
16/09/2024	Marketing	A
13/06/2024	Ventas	B

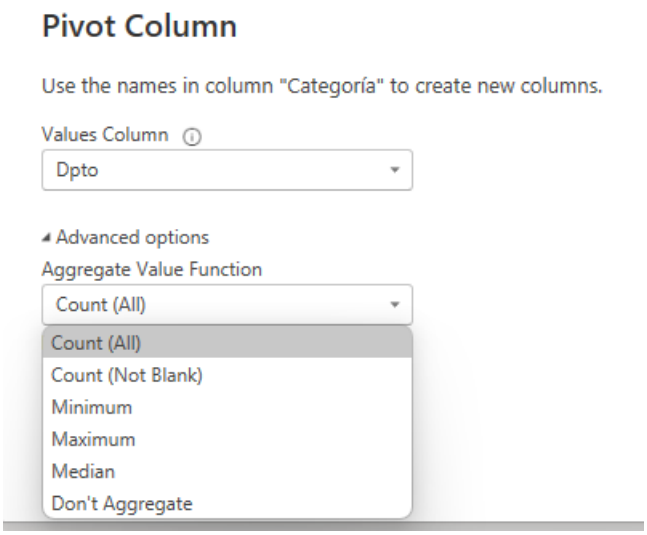
En el caso de que necesitemos pivotar la tabla por el campo **Dpto**, vamos a obtener un error si no se aplica ninguna operación de agregación, ya que existen varios registros para la misma categoría y fecha que tienen un valor diferente de Dpto.



	Date	A	B	C
1	30/01/2024	null	null	RRHH
2	28/02/2024	null	Marketing	null
3	11/05/2024	Ventas	null	null
4	13/06/2024	null	Error	null
5	16/09/2024	Error	null	null

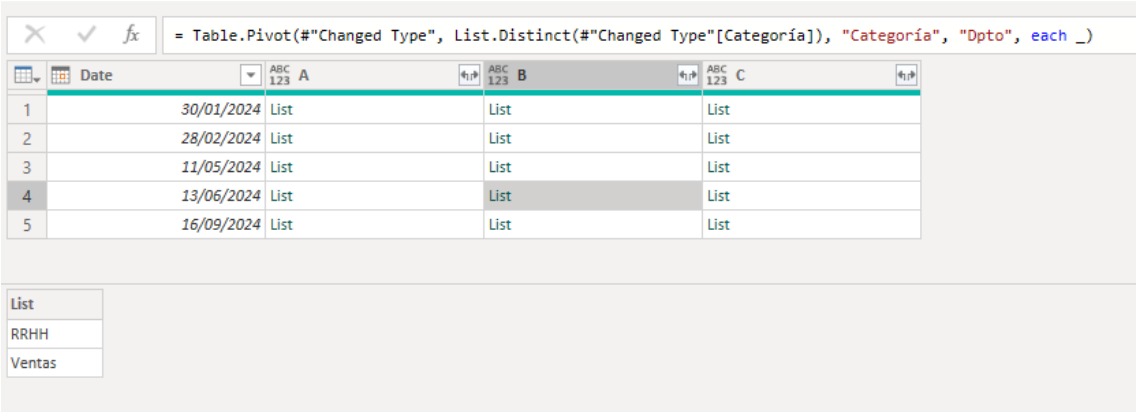
Expression.Error: There were too many elements in the enumeration to complete the operation.
Details:
[List]

Sin embargo, si se aplicara una de las agregaciones disponibles (por ejemplo, Count), no daría error.



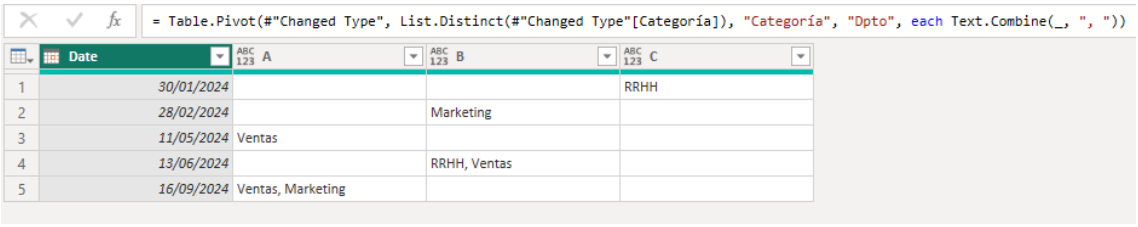
Date	A	B	C
30/01/2024	0	0	1
28/02/2024	0	1	0
11/05/2024	1	0	0
13/06/2024	0	2	0
16/09/2024	2	0	0

Pero en este ejemplo no necesitamos aplicar una agregación, sino que se necesita ver todos los valores. Para ello, en primer lugar, vamos a convertir los registros de la tabla en una **lista**, añadiendo "**each _**" en el paso **pivotar la tabla**:



Como se puede ver en la imagen, la tabla ahora está formada por listas.

Para convertir la lista en valores separados por comas, se utiliza la función **Text.Combine** de la siguiente manera:



Ahora se muestran todos los valores y, en aquellos registros que tengan más de un valor, se muestran todos separados por comas y sin errores.

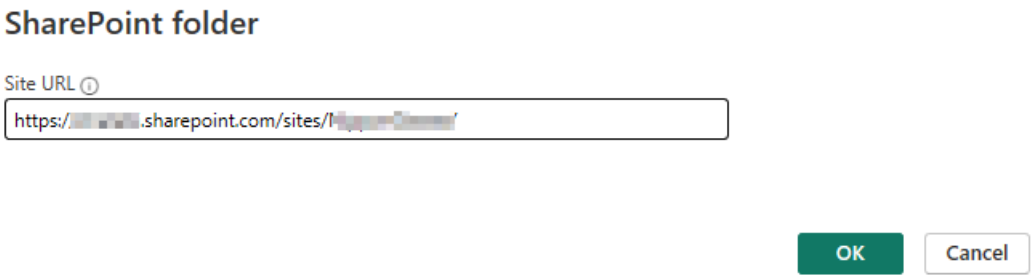
5. Importar varios ficheros Excel ubicados en Sharepoint

El propósito de este tip es mostrar cómo se puede importar varios ficheros de Excel ubicados en una carpeta de Sharepoint sin utilizar las estructuras que se crean por defecto en Power Query al combinar los archivos.

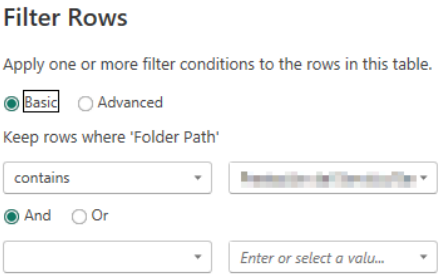


Desarrollo:

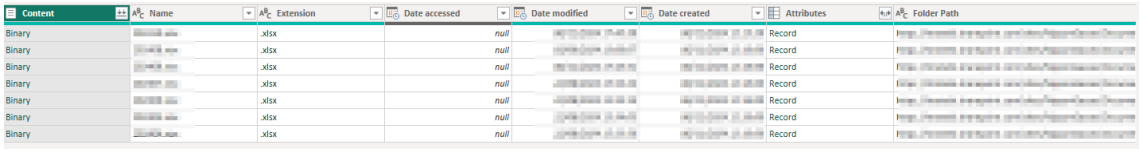
El primer paso, es conectar la carpeta de Sharepoint a Power BI. Para ello, se necesita utilizar el conector **Sharepoint folder** y escribir la URL del sitio de Sharepoint (no la URL entera, solo hasta el nombre del sitio).



Si los ficheros no están ubicados en la raíz del sitio sino en una subcarpeta, el siguiente paso es filtrar el campo **Folder Path** por el nombre de la carpeta donde estén ubicados.



Como se puede observar en la siguiente imagen, en esta subcarpeta hay 7 ficheros .xlsx, que son los que vamos a combinar.

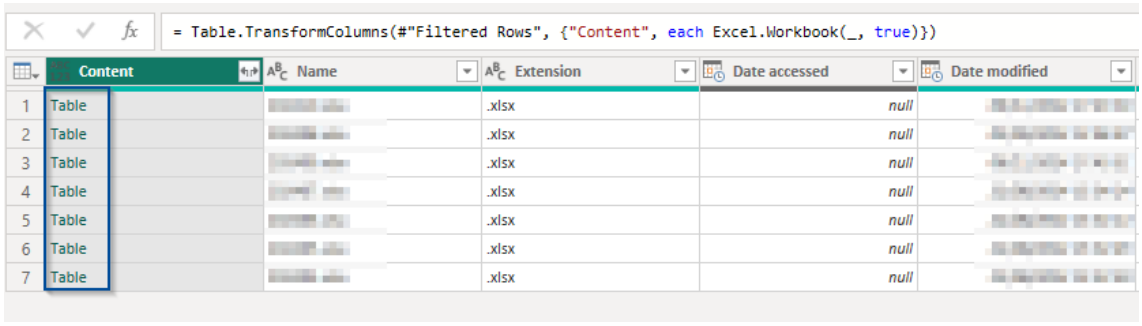


Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
Binary		.xlsx	null			Record	
Binary		.xlsx	null			Record	
Binary		.xlsx	null			Record	
Binary		.xlsx	null			Record	
Binary		.xlsx	null			Record	
Binary		.xlsx	null			Record	
Binary		.xlsx	null			Record	

Si seleccionáramos el botón de **Combinar ficheros** que aparece en el campo **Content**, se combinarían los ficheros, pero se crearían todas las siguientes estructuras auxiliares:

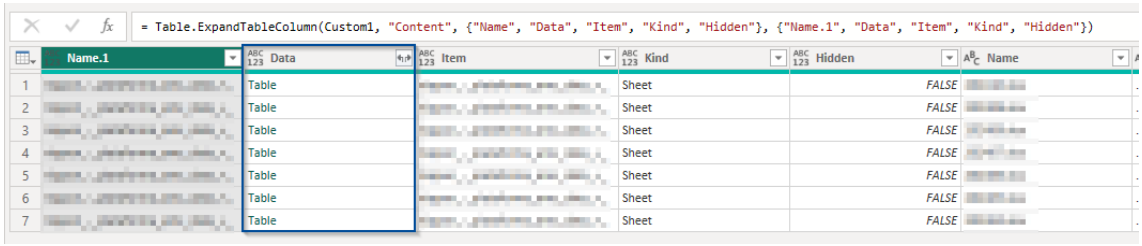


En el caso de tener varias consultas, se crearía una carpeta **Helper Queries** por cada una de ellas, llegando a dificultar la legibilidad de la ubicación de las tablas del modelo. Precisamente este es el objetivo de este tip, combinar estos ficheros de forma sencilla sin hacer uso de estos elementos, obteniendo como resultado el listado de consultas de Power Query mucho más claro y ordenado. Volviendo al punto donde tenemos el listado de los ficheros de la carpeta, el siguiente paso es crear un paso personalizado para transformar en tablas los registros de tipo binario:



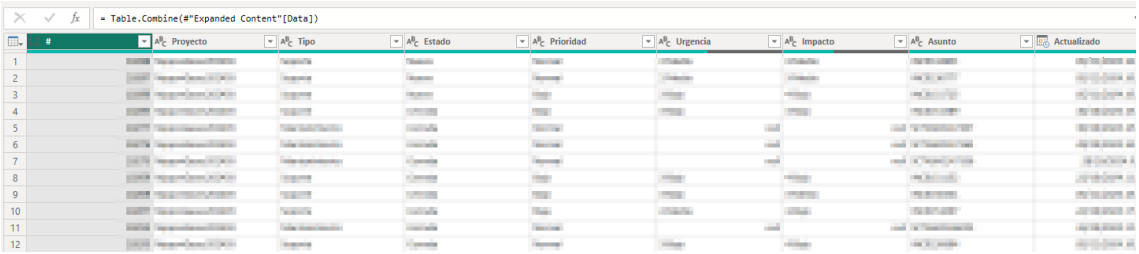
	Content	Name	Extension	Date accessed	Date modified
1	Table		.xlsx	null	
2	Table		.xlsx	null	
3	Table		.xlsx	null	
4	Table		.xlsx	null	
5	Table		.xlsx	null	
6	Table		.xlsx	null	
7	Table		.xlsx	null	

A continuación, se expande la columna **Content**, seleccionando los campos que sean necesarios:



	Name.1	Data	Item	Kind	Hidden	Name
1		Table		Sheet	FALSE	
2		Table		Sheet	FALSE	
3		Table		Sheet	FALSE	
4		Table		Sheet	FALSE	
5		Table		Sheet	FALSE	
6		Table		Sheet	FALSE	
7		Table		Sheet	FALSE	

Como se puede observar en la imagen, el campo **Data** contiene registros de tipo tabla, por lo que el siguiente paso es crear otro **paso personalizado** para combinar estas tablas con la función **Table.Combine**:



De esta forma, con solo 2 pasos, conseguimos combinar todos los ficheros .xlsx de una carpeta de Sharepoint sin utilizar elementos auxiliares.

6. Obtener datos de Dynamics a través de API

El propósito de este tip es mostrar cómo se puede obtener datos de Dynamics a través de la API.



Desarrollo:

Con el conector de Power BI para Dynamics pueden surgir problemas de rendimiento si en el modelo existen relaciones muchos a muchos, es decir, si hay tablas que contienen algún campo que contenga otra tabla. Este caso puede incluso a dar problemas de timeout al refrescar el modelo semántico en el servicio de Power BI.

Como alternativa a utilizar el conector de Dynamics, se puede hacer uso de su API paginada, obteniendo una lista de registros.

Para esto, se crea una función desde Power Query con la siguiente estructura:

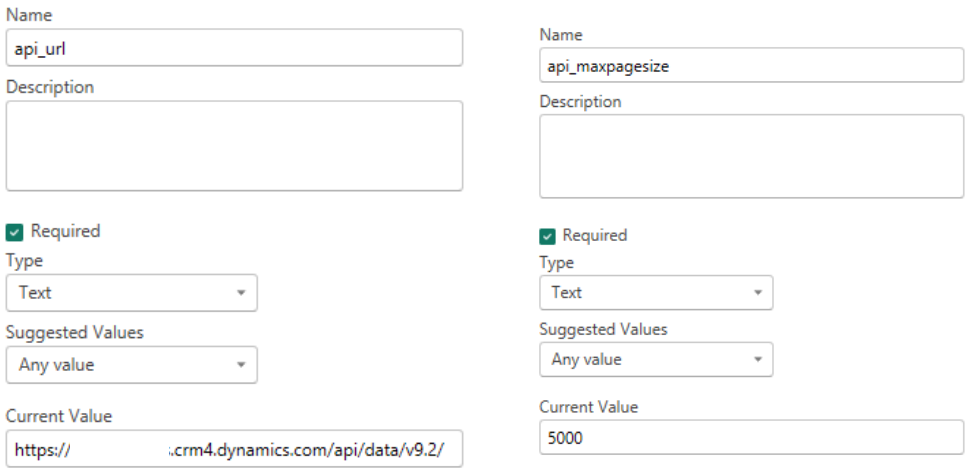
```
(dyn365Entity as text, optional columns as text) =>
let
    includeAnnotations = "odata.maxpagesize=" & api_maxpagesize & ", odata.include-annotations=OData.Community.Display.V1.FormattedValue",
    selectColumns = if columns = null then ""  else "?$select=" & columns,
    options = [
        Headers = [Prefer = includeAnnotations],
        RelativePath = dyn365Entity & selectColumns
    ],

    initRequest = Json.Document(Web.Contents(api_url, options)),
    initData = Table.FromRecords({initRequest}),
    initNextUrl = try initRequest[#"@odata.nextLink"] otherwise null,

    getData = (data as table, url as text) =>
        let
            firstOcurrence = Text.PositionOf(url, dyn365Entity & "?"),
            urlSubstring = Text.Range(url, firstOcurrence),
            options = [
                Headers = [Prefer = includeAnnotations],
                RelativePath = urlSubstring
            ],
            newRequest = Json.Document(Web.Contents(api_url, options)),
            newData = Table.FromRecords({newRequest}),
            newNextUrl = try newRequest[#"@odata.nextLink"] otherwise null,
            data = Table.Combine({data, newData}),
            check = if newNextUrl = null then data else @getData(data, newNextUrl)
        in
            check,

    outputTable = if initNextUrl = null then initData else @getData(initData, initNextUrl)
in
    outputTable
```

Esta función, básicamente pagina los datos cada 5000 registros. Para utilizar esta función, es necesario crear previamente dos parámetros: **api_url** (representa el valor de la url base) y **api_maxpagesize** (representa el tamaño máximo por página, en este caso, 5000). Además, se deberá crear un parámetro por cada entidad con su nombre.



Para hacer uso de función, por cada entidad de Dynamics que se vaya a importar en el modelo, en el paso de origen se referencia la función **get_data_from_dyn_365** con el parámetro del nombre de la entidad y, de forma opcional y entre comillas dobles, el/los nombres de los campos a importar.



2. MODELADO

7. Calcular la media de N meses anteriores

El propósito de este tip es mostrar cómo calcular la media de los últimos N meses de forma dinámica.



Desarrollo:

En primer lugar, vamos a crear un periodo dinámico, en este caso de 3 meses, obteniendo el número de días de los últimos tres meses. Para conseguir esto, creamos una medida que utilice la función **DATESINPERIOD** de la siguiente manera:

```
1  Moving AVG =
2  var period =
3  DATESINPERIOD(DimDate[FullDateAlternateKey],
4  MAX(DimDate[FullDateAlternateKey]),
5  -3,
6  MONTH
7  )
8
9  RETURN
10     COUNTROWS(period)
```

Year	Total Purchases Amount	Moving AVG
2011		
enero		31
febrero		59
marzo		90
abril	9.491,52	89
mayo	94.404,30	92
junio		91
julio		92
agosto		92
septiembre		92
octubre		92
noviembre		91
diciembre	299.239,98	92
2012		
enero	354.968,23	92
febrero	622.885,33	91
marzo	649.928,94	91
abril	346.917,55	90
mayo	106.491,00	92
junio	171.578,43	91
julio	483.826,64	92
agosto	291.046,13	92
septiembre	333.986,04	92
octubre	565.264,76	92
noviembre		91
diciembre		92
2013		
enero		92
febrero	131.485,79	90
marzo		90
abril	123.167,77	89
mayo	749.096,55	92
junio	719.047,52	91

A continuación, vamos a modificar la medida para calcular el **numerador** de la media, es decir, la suma de **Total Purchases Amount** de los últimos 3 meses.

```
Moving AVG =
VAR period =
DATESINPERIOD(
    Dim_time[DateAlternateKey],
    MAX(Dim_time[DateAlternateKey]),
    -3,
    MONTH
)

VAR TotalPurchases=
    CALCULATE(
        [Total Purchases Amount], period
    )

RETURN
  TotalPurchases
```

Year	Total Purchases Amount	Moving AVG
2011		
abril	9.491,52	9.491,52
mayo	94.404,30	103.895,82
junio		103.895,82
julio		94.404,30
diciembre	299.239,98	299.239,98
2012		
enero	354.968,23	654.208,21
febrero	622.885,33	1.277.093,54
marzo	649.928,94	1.627.782,49
abril	346.917,55	1.619.731,82
mayo	106.491,00	1.103.337,49
junio	171.578,43	624.986,99
julio	483.826,64	761.896,08
agosto	291.046,13	946.451,21
septiembre	333.986,04	1.108.858,81
octubre	565.264,76	1.190.296,93
noviembre		899.250,80
diciembre		565.264,76
2013		
febrero	131.485,79	131.485,79
marzo		131.485,79
abril	123.167,77	254.653,56
mayo	749.096,55	872.264,33
junio	719.047,52	1.591.311,84
julio	117.331,99	1.585.476,06
agosto	2.786.955,42	3.623.334,93
septiembre	4.926.761,06	7.831.048,47
octubre	2.211.744,23	9.925.460,72
noviembre	1.925.623,83	9.064.129,13
diciembre	4.367.817,60	8.505.185,67
2014		
enero	4.479.190,76	10.772.632,20

En el siguiente paso, vamos a calcular el número de meses, es decir, el **denominador** de la media. Para ello, vamos a crear una nueva variable.

```
Moving AVG =
VAR period =
DATESINPERIOD(
    Dim_time[DateAlternateKey],
    MAX(Dim_time[DateAlternateKey]),
    -3,
    MONTH
)

VAR TotalPurchases=
    CALCULATE(
        [Total Purchases Amount], period
    )

VAR mon =
CALCULATE(
    DISTINCTCOUNT(Dim_time[Month]),
    period)

RETURN
   mon
```

Year	Total Purchases Amount	Moving AVG
2011		
enero		1
febrero		2
marzo		3
abril	9.491,52	3
mayo	94.404,30	3
junio		3
julio		3
agosto		3
septiembre		3
octubre		3
noviembre		3
diciembre	299.239,98	3
2012		
enero	354.968,23	3
febrero	622.885,33	3
marzo	649.928,94	3
abril	346.917,55	3
mayo	106.491,00	3
junio	171.578,43	3
julio	483.826,64	3
agosto	291.046,13	3
septiembre	333.986,04	3
octubre	565.264,76	3
noviembre		3
diciembre		3
2013		
enero		3
febrero	131.485,79	3
marzo		3
abril	123.167,77	3
mayo	749.096,55	3
junio	719.047,52	3

Una vez calculados el numerador y el denominador en sus correspondientes variables, vamos a calcular la **media**, obteniendo el resultado final:

```
Moving AVG =
VAR period =
DATESINPERIOD(
    Dim_time[DateAlternateKey],
    MAX(Dim_time[DateAlternateKey]),
    -3,
    MONTH
)

VAR TotalPurchases=
    CALCULATE(
        [Total Purchases Amount], period
    )

VAR mon =
CALCULATE(
    DISTINCTCOUNT(Dim_time[Month]),
    period)

RETURN
   DIVIDE(TotalPurchases, mon)
```

Year	Total Purchases Amount	Moving AVG
2011		
abril	9.491,52	3.163,84
mayo	94.404,30	34.631,94
junio		34.631,94
julio		31.468,10
diciembre	299.239,98	99.746,66
2012		
enero	354.968,23	218.069,40
febrero	622.885,33	425.697,85
marzo	649.928,94	542.594,16
abril	346.917,55	539.910,61
mayo	106.491,00	367.779,16
junio	171.578,43	208.329,00
julio	483.826,64	253.965,36
agosto	291.046,13	315.483,74
septiembre	333.986,04	369.619,60
octubre	565.264,76	396.765,64
noviembre		299.750,27
diciembre		188.421,59
2013		
febrero	131.485,79	43.828,60
marzo		43.828,60
abril	123.167,77	84.884,52
mayo	749.096,55	290.754,78
junio	719.047,52	530.437,28
julio	117.331,99	528.492,02
agosto	2.786.955,42	1.207.778,31
septiembre	4.926.761,06	2.610.349,49
octubre	2.211.744,23	3.308.486,91
noviembre	1.925.623,83	3.021.376,38
diciembre	4.367.817,60	2.835.061,89
2014		
enero	4.479.190,76	3.590.877,40

Como hemos explicado durante el desarrollo del ejemplo, se está calculando la media de los últimos 3 meses, pero puede darse la necesidad de que estos N meses sean dinámicos, es decir, que el usuario pueda elegir la cantidad de meses sobre los que calcular la media.

Para esto, en primer lugar vamos a crear una **tabla nueva** con valores del 1 al 12 (representando los 12 meses del año) de la siguiente forma:

```
1 Selected_Months = GENERATESERIES(1,12)
```

Value
1
2
3
4
5
6
7
8
9
10
11
12

A continuación, añadimos un objeto visual de tipo segmentación al lienzo con el campo **Value** de esta nueva tabla. En este punto, la segmentación no tiene ningún efecto sobre los valores que se muestran en el objeto de tipo tabla.

Para hacer que se calcule de forma dinámica la media dependiendo del número que se seleccione en la segmentación, tenemos que crear una nueva medida:

```
1 Selected month = SELECTEDVALUE(Selected_Months[Value], 3)
```

En el caso de que no se seleccione ningún valor de la segmentación o de que se seleccione más de uno, el número de meses por defecto va a ser 3.

Ahora, con esta medida creada, vamos a modificar **Moving AVG** sustituyendo el -3 por la medida **Selected month.**

```
Moving AVG =
VAR period =
DATESINPERIOD(
    Dim_time[DateAlternateKey],
    MAX(Dim_time[DateAlternateKey]),
    -[Selected month],
    MONTH
)

VAR TotalPurchases=
    CALCULATE(
        [Total Purchases Amount], period
    )

VAR mon =
CALCULATE(
    DISTINCTCOUNT(Dim_time[Month]),
    period)

RETURN
   DIVIDE(TotalPurchases, mon)
```

Al ir seleccionando distintos valores de la segmentación podemos comprobar que, efectivamente, la media se va recalculando según el número de meses seleccionado.



Year	Total Purchases Amount	Moving AVG
2011		
abril	9.491,52	2.372,88
mayo	94.404,30	20.779,16
junio		20.779,16
julio		20.779,16
agosto		20.779,16
septiembre		18.880,86
diciembre	299.239,98	59.848,00
2012		
enero	354.968,23	130.841,64
febrero	622.885,33	255.418,71
marzo	649.928,94	385.404,50
abril	346.917,55	454.788,01
mayo	106.491,00	416.238,21
junio	171.578,43	379.560,25
julio	483.826,64	351.748,51
agosto	291.046,13	279.971,95
septiembre	333.986,04	277.385,65
octubre	565.264,76	369.140,40
noviembre		334.824,71
diciembre		238.059,39
2013		
enero		179.850,16
febrero	131.485,79	139.350,11
marzo		26.297,16
abril	123.167,77	50.930,71
mayo	749.096,55	200.750,02
junio	719.047,52	344.559,53
julio	117.331,99	341.728,77
agosto	2.786.955,42	899.119,85
septiembre	4.926.761,06	1.859.838,51
octubre	2.211.744,23	2.152.368,04
noviembre	1.925.623,83	2.393.683,31

8. Utilizar una medida como segmentación en Power BI

El propósito de este tip es mostrar cómo usar una métrica en un filtro, agrupando los valores por rangos.



Desarrollo:

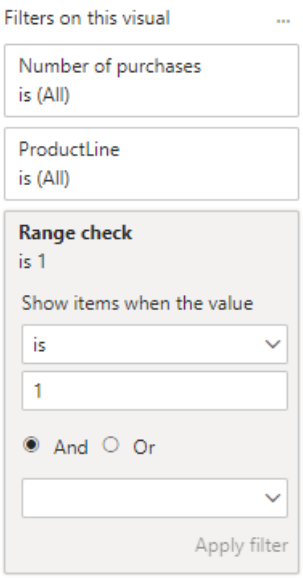
En este caso, se necesita poder segmentar los datos por el campo del rango de fechas. Para esto, en primer lugar, se debe crear una tabla con los campos min, max y el nombre del grupo. Esta tabla no la vamos a relacionar con ninguna otra, quedando **aislada** en el modelo.

Min	Max	Purchases range
0	100	(0,100]
101	500	(100,500]
501	1000	(501,1000]
1001	9999999	(>1000)

A continuación, se debe crear una nueva medida en la tabla para calcular el rango de cada valor:

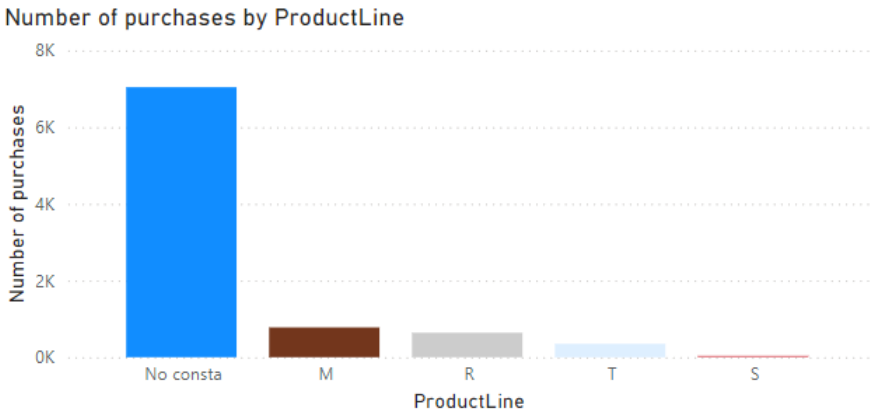
```
1 Range check =
2 VAR selectedMin = MIN('Purchases Range'[Min])
3 VAR selectedMax = MAX('Purchases Range'[Max])
4 VAR originalMeasure = [Number of purchases]
5 RETURN
6 IF(AND(originalMeasure>=selectedMin, originalMeasure<=selectedMax), 1, 0)
```

El siguiente paso es utilizar esta medida como filtro en las visualizaciones en las que se necesite:



Ahora, se añade un objeto visual de tipo segmentación al lienzo con el campo **Purchases range** y, como se puede observar, al seleccionar cada uno de los rangos de la segmentación, el objeto visual queda filtrado por este.






Purchases range
(0,100]
(100,500]
(501,1000]
(>1000)
Number of purchases by ProductLine
800
600
400
200
0
Number of purchases
M
R
ProductLine


Purchases range
(0,100]
(100,500]
(501,1000]
(>1000)
Number of purchases by ProductLine
40
30
20
10
0
Number of purchases
S
ProductLine

9. Aproximación a la funcionalidad Seguridad por página

El propósito de este tip es mostrar una posible aproximación a la funcionalidad de Seguridad por página.

Nota: Esta funcionalidad no está implementada de forma oficial en Power BI y, por lo tanto, cualquier aproximación que se desarrolle debe utilizarse con precaución, sobre todo si el informe incluye datos confidenciales o sensibles.



Desarrollo:

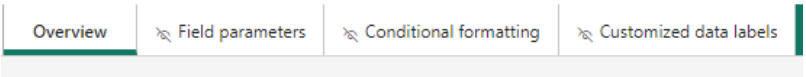
Como se indica más arriba, esta funcionalidad no está oficialmente implementada, sino que es una aproximación para imitar la seguridad a nivel de página.

El primer paso es crear una tabla con el nombre de las páginas del informe y los diferentes roles que va a haber. En este caso, el informe tiene 4 páginas y vamos a crear 2 roles: **Developer** y **Manager**.

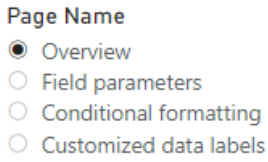
Page	Page Name	Developer	Manager
1	Overview	True	True
2	Field parameters	True	True
3	Conditional formatting	True	False
4	Customized data labels	True	False

Los valores True/False indican la visibilidad de cada rol sobre las diferentes páginas del informe.

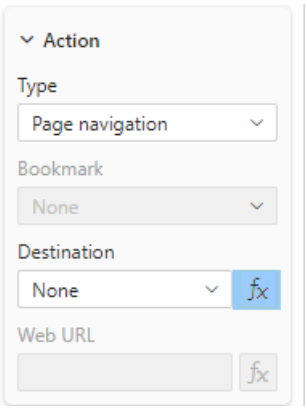
Antes de continuar, nos debemos asegurar de que solo la primera página está visible y las demás, ocultas.



A continuación, añadimos una segmentación en todas las páginas con el campo **Page Name** de la nueva tabla que acabamos de crear.

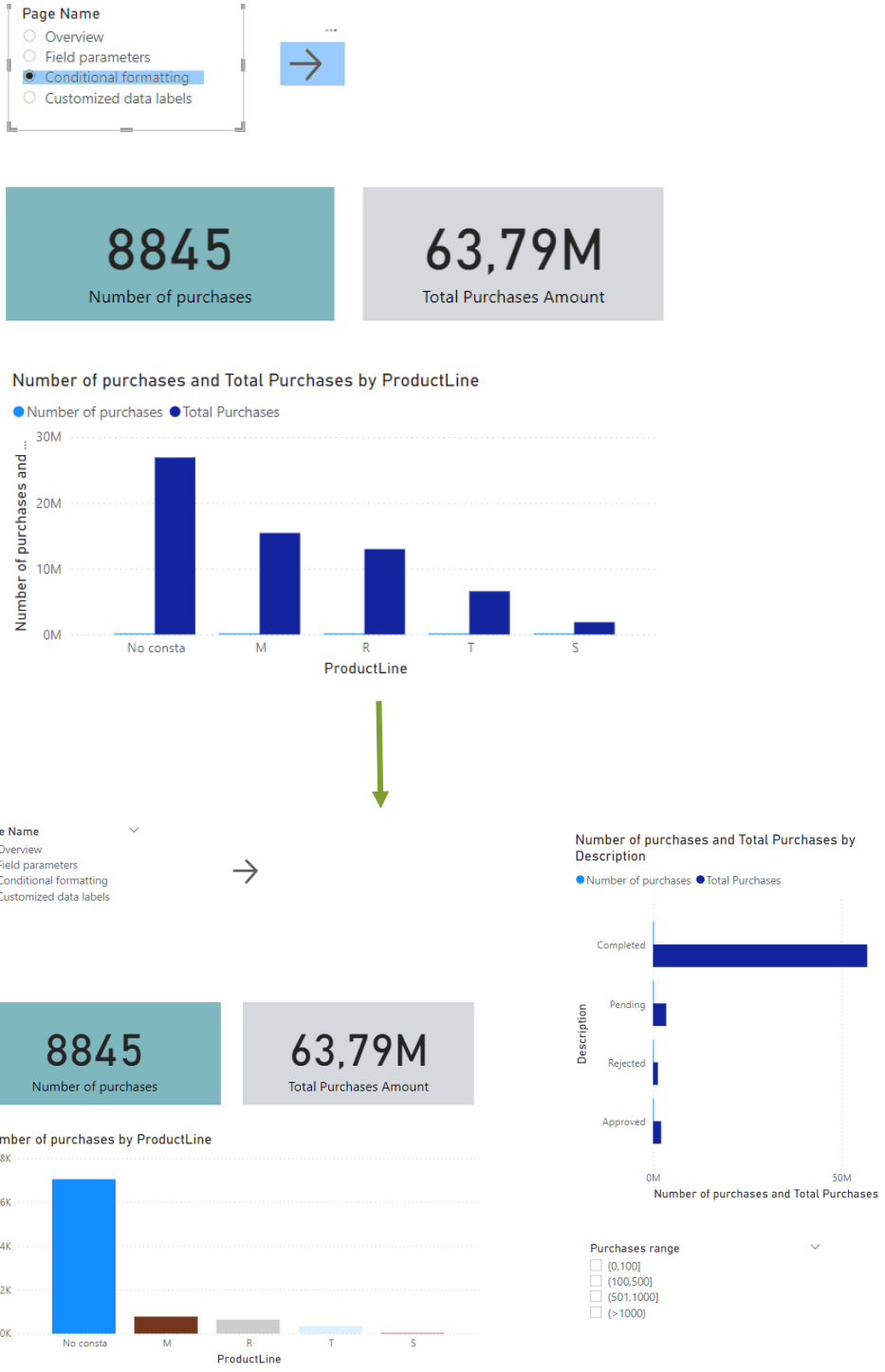


Además, vamos a añadir en todas las páginas un botón con el tipo de Acción **Navegación entre páginas** y, como destino, vamos a seleccionar la opción de **Formato condicional** y configurándolo de la siguiente forma:

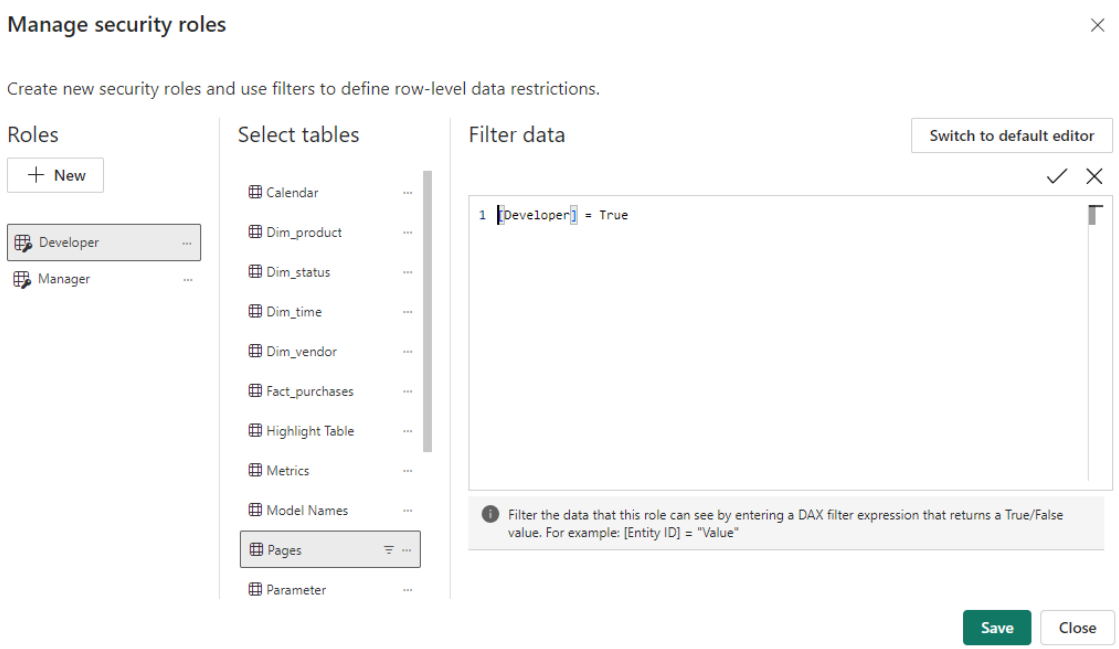


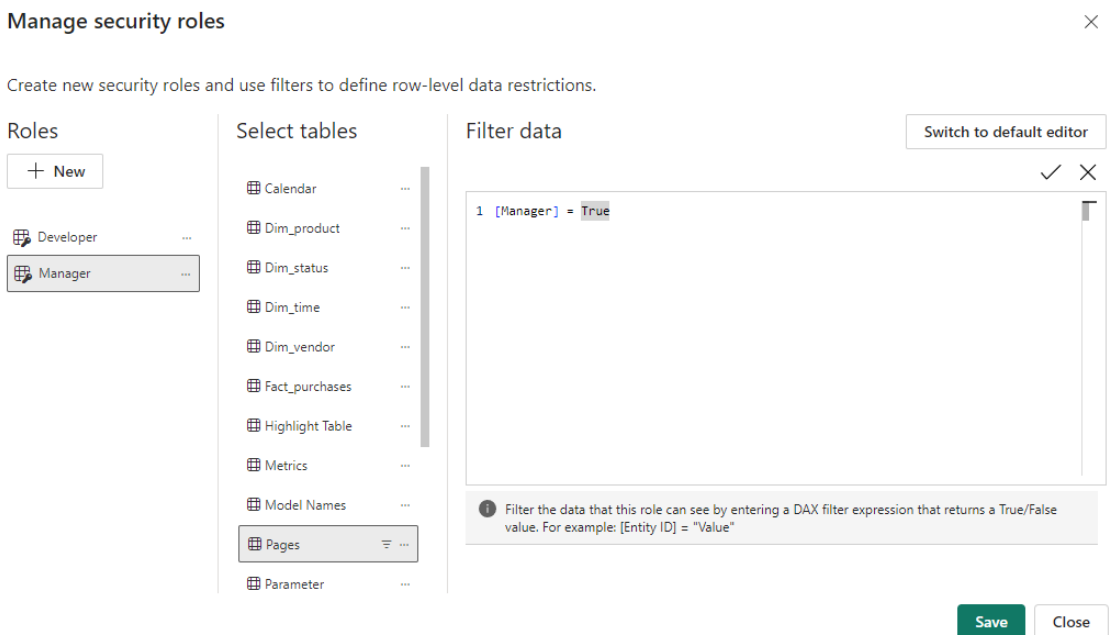


De esta forma, si seleccionamos una de las opciones de la segmentación y, a continuación, el botón, nos llevará a la página seleccionada.



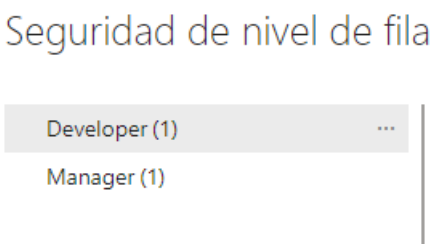
A continuación, se crean los dos **roles** como se indica en las capturas:



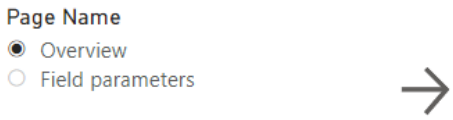


El siguiente paso es guardar y **publicar** el informe en el servicio de Power BI.

Una vez publicado, configuramos la seguridad del modelo para ambos roles.



Si seleccionamos la opción **probar el rol** de **Manager**, vemos que, efectivamente, solo se pueden ver las dos primeras páginas (Overview y Field parameters), que era lo establecido.





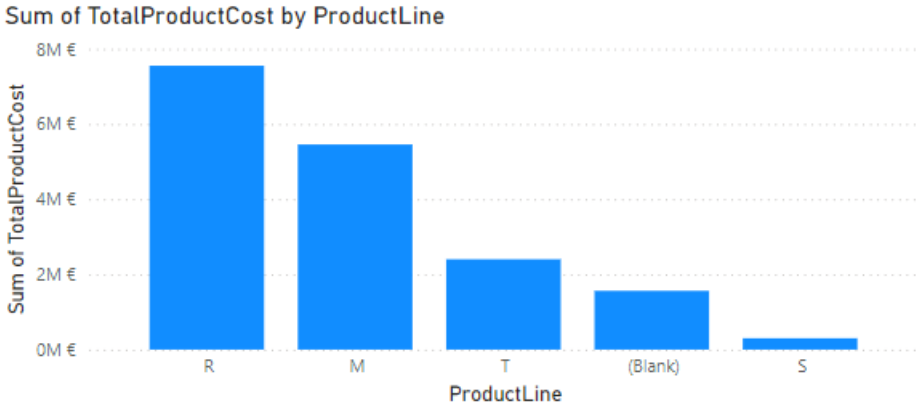
10. Localizar problemas de integridad referencial

El propósito de este tip es explicar a qué se debe que aparezca el valor “En blanco” en las visualizaciones y cómo solventar esta situación.



Desarrollo:

En ocasiones puede suceder que, en alguna de las visualizaciones, veamos que hay un valor “**En blanco**”.



Esto se debe a un problema de **integridad referencial**, es decir, hay valores de un campo en la tabla de hechos que no tienen su correspondencia en la tabla de dimensión. La aplicación de este tip puede servir como **control de calidad de los datos** del modelo semántico en Power BI.
En este ejemplo, hay valores de códigos de producto en la tabla de hechos que no existen en la tabla de dimensión de producto.
Para solucionar este problema, primero tenemos que identificar cuáles son los valores que no tienen correspondencia entre la tabla de hechos y la de dimensión.

Para esto, hay varias implementaciones posibles, pero en este caso, lo que vamos a hacer es crear una tabla auxiliar para identificar esos códigos de producto que no existen en la dimensión:

```
Ref integrity =
ADDCOLUMNS(
    EXCEPT(
        VALUES(FactInternetSales[ProductKey]),
        VALUES(DimProduct[ProductKey])),
    "Table", "Product")
```

Básicamente, se están calculando los **códigos** de productos que están en la tabla de hechos, pero no en la dimensión de producto. Además, añadimos una columna con el nombre de la dimensión donde hay este problema de integridad referencial. El resultado de esta tabla es el siguiente:

ProductKey	Table
310	Product
346	Product
336	Product
312	Product

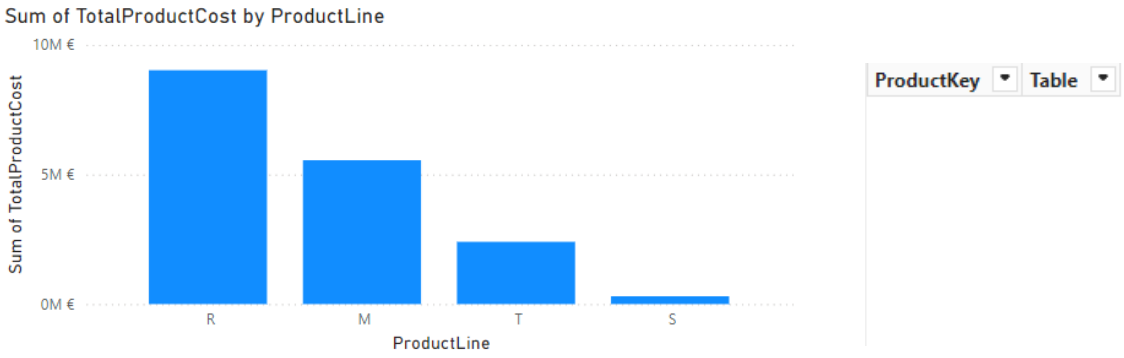
Una vez que sabemos qué códigos de productos son los que están originando el problema, podemos **subsanar** estos problemas en la dimensión correspondiente.

En el caso de que tengamos este tipo de problema en **varias dimensiones**, podríamos realizar la siguiente modificación sobre la definición de la tabla:

```
Ref integrity =
VAR Prod=
ADDCOLUMNS(
    EXCEPT(
        VALUES(FactInternetSales[ProductKey]),
        VALUES(DimProduct[ProductKey])),
    "Table", "Product")
VAR Geo =
ADDCOLUMNS(
    EXCEPT(
        VALUES(FactInternetSales[SalesTerritoryKey]),
        VALUES(DimGeography[SalesTerritoryKey])),
    "Table", "Geography")
RETURN UNION(Prod, Geo)
```

ProductKey	Table
310	Product
346	Product
336	Product
312	Product
4	Geography
9	Geography

En este caso, se debería de renombrar el campo **ProductKey** por **Id** o alguno similar.
Mediante el uso de variables, creamos una por cada dimensión que sea necesaria y la lógica es la misma que en el anterior caso. De esta forma, podemos añadir **todas** las **dimensiones** que sean **necesarias** y, en una misma tabla, tendríamos la **referencia** de todas aquellas claves que debemos añadir en su correspondiente dimensión.
Una vez **subsanados** todos los **errores**, no aparecerá el valor "**En blanco**" en la visualización y esta tabla aparecerá vacía.



11. No mostrar registros sin datos en una tabla

El propósito de este tip es explicar qué se puede hacer para que no se muestren registros en una visualización de tabla si el dato no existe, pero, sin embargo, existe el registro porque se está calculando un acumulado.



Desarrollo:

Se puede dar el escenario en el que se esté calculando el **importe total** de compras, por ejemplo, y el **acumulado** de este importe durante el año. Si hay algún mes que no tiene importe, esta fila sigue apareciendo en la tabla, **replicándose** el último valor calculado. Este comportamiento es correcto, pero puede ser que solo interese ver aquellos meses en los que exista un valor para el importe.

Year	MonthName	Total Purchases Amount	Purchases YTD
2014	enero	4.479.190,76	4.479.190,76
	febrero	3.990.455,44	8.469.646,20
	marzo	5.390.859,62	13.860.505,82
	abril	4.812.622,59	18.673.128,41
	mayo	5.459.623,06	24.132.751,47
	junio	5.790.441,34	29.923.192,82
	julio	7.400.998,57	37.324.191,39
	agosto	4.077.395,67	41.401.587,06
	septiembre	327,16	41.401.914,21
	octubre	1.020,00	41.402.934,21
	noviembre		41.402.934,21
	diciembre		41.402.934,21

Para esto, en la tabla de calendario, vamos a crear una nueva columna:

```
1 Check = 'Calendar'[Date] <= EOMONTH(MAX(Fact_purchases[DueDate]), 0)
```

Date	Year	Quarter	Month	Day	Month name	Check
17/10/2014	2014	4	10	17	octubre	True
18/10/2014	2014	4	10	18	octubre	True
19/10/2014	2014	4	10	19	octubre	True
20/10/2014	2014	4	10	20	octubre	True
21/10/2014	2014	4	10	21	octubre	True
22/10/2014	2014	4	10	22	octubre	True
23/10/2014	2014	4	10	23	octubre	True
24/10/2014	2014	4	10	24	octubre	True
25/10/2014	2014	4	10	25	octubre	True
26/10/2014	2014	4	10	26	octubre	True
27/10/2014	2014	4	10	27	octubre	True
28/10/2014	2014	4	10	28	octubre	True
29/10/2014	2014	4	10	29	octubre	True
30/10/2014	2014	4	10	30	octubre	True
31/10/2014	2014	4	10	31	octubre	True
01/11/2014	2014	4	11	1	noviembre	False
02/11/2014	2014	4	11	2	noviembre	False
03/11/2014	2014	4	11	3	noviembre	False
04/11/2014	2014	4	11	4	noviembre	False

El cálculo que se está realizando es comparar la fecha de la dimensión con el último día del mes de la fecha máxima de la tabla de hechos y, si es menor o igual, devuelve verdadero, en caso contrario, devuelve falso.

El siguiente paso es modificar la medida que tenemos creada, **Purchases YTD,** para que tenga en cuenta el resultado de este nuevo campo que acabamos de crear.

```
Purchases YTD =
CALCULATE(
    TOTALYTD([Total Purchases Amount],'Calendar'[Date]),
    'Calendar'[Check]
)
```

Year	MonthName	Total Purchases Amount	Purchases YTD
2014	enero	4.479.190,76	4.479.190,76
	febrero	3.990.455,44	8.469.646,20
	marzo	5.390.859,62	13.860.505,82
	abril	4.812.622,59	18.673.128,41
	mayo	5.459.623,06	24.132.751,47
	junio	5.790.441,34	29.923.192,82
	julio	7.400.998,57	37.324.191,39
	agosto	4.077.395,67	41.401.587,06
	septiembre	327,16	41.401.914,21
	octubre	1.020,00	41.402.934,21
2013	febrero	131.485,79	131.485,79

Como se puede observar en la captura, para el año 2014 ya no aparecen los últimos meses que no tenían valor de **Total Purchases Amount**.

12. Localizar duplicados desde la vista de consultas DAX

El propósito de este tip es mostrar cómo se pueden identificar duplicados sin necesidad de crear columnas adicionales, únicamente realizando una consulta sencilla desde la vista de consultas DAX.



Desarrollo:

En este caso, el objetivo es encontrar **registros duplicados** de productos para la misma fecha, es decir, aquellos productos que se han comprado varias veces en la misma fecha.

Para esto, abrimos la vista de consultas y escribimos los siguiente:

```
EVALUATE
    GROUPBY(Fact_purchases, Fact_purchases[DueDate], Fact_purchases[ProductKey],
    "Check", COUNTAX(CURRENTGROUP(), 1)
    )
    ORDER BY [Check] DESC

```

En la consulta, se está contabilizando el **número de registros** que tienen el mismo valor de fecha y clave de producto.

Obteniendo como resultado de la consulta:

Results | Result 1 of 1 | Copy

	Fact_purchases[DueDate]	Fact_purchases[Product...	[Check]
1	3/12/2014 12:00:00 AM	8	3
2	3/18/2014 12:00:00 AM	8	3
3	8/18/2013 12:00:00 AM	8	3
4	2/23/2012 12:00:00 AM	8	3
5	1/20/2014 12:00:00 AM	8	3
6	4/3/2014 12:00:00 AM	8	3
7	5/22/2014 12:00:00 AM	370	3
8	2/26/2014 12:00:00 AM	10	2
9	9/4/2013 12:00:00 AM	444	2
10	3/6/2014 12:00:00 AM	10	2
11	3/14/2014 12:00:00 AM	440	2
12	3/20/2014 12:00:00 AM	440	2
13	3/28/2014 12:00:00 AM	440	2
14	4/7/2014 12:00:00 AM	440	2
15	12/6/2013 12:00:00 AM	30	2
16	1/17/2014 12:00:00 AM	437	2
17	3/19/2014 12:00:00 AM	209	2
18	4/25/2012 12:00:00 AM	26	2
19	3/21/2014 12:00:00 AM	444	2
20	10/6/2013 12:00:00 AM	441	2

Aquellas filas cuyo valor del campo **Check** sea mayor que 1 significa que están **duplicadas**, que era el objetivo.

3. VISUALIZACIÓN

13. Mostrar una medida en unidades o en miles con un slicer

El objetivo de este apartado es mostrar cómo se puede mostrar el resultado de una medida en un objeto visual en unidades o en miles según la opción que se seleccione en el slicer.



Desarrollo:

Antes de crear el slicer, el primer paso a realizar es crear una **tabla** en el modelo (esta tabla va a quedar **aislada**, sin relacionar con ninguna otra) de la siguiente forma:

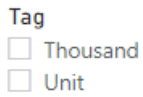
```
Units =
DATATABLE(
    "Tag", STRING, "Value", INTEGER,
    {
        {"Thousand", 1000},
        {"Units", 1}
    }
)
```

Esta tabla también se puede crear manualmente desde la opción **Enter Data** en vez de usar DAX.

	Tag	Value	+
1	Thousand	1000	
2	Unit	1	
+			

Esta tabla contiene el campo de etiqueta que vamos a utilizar en el slicer y el valor divisor que se utiliza, dependiendo de la etiqueta que se seleccione en la segmentación.

A continuación, se añade un slicer al lienzo con el campo "**Tag**" de la tabla que acabamos de crear.

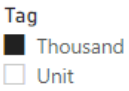


Ahora, para que el resultado de la medida se vea afectado por la opción que seleccionemos en la segmentación, vamos a crear una **nueva medida** de forma que se divida la medida por el denominador correspondiente.

```
Total Purchases Amount units = DIVIDE([Total Purchases Amount], SELECTEDVALUE(Units[Value], 1))
```

En el caso de que no haya ningún valor seleccionado en la segmentación, el denominador es 1 y, por tanto, se mostraría el valor original (en formato de unidades).

CategoryName	Total Purchases Amount	Total Purchases Amount units
Accessories		
Bike Racks	8.470,00	8,47
Bike Stands	23.784,00	23,78
Bottles and Cages	8.035,00	8,04
Cleaners	445,50	0,45
Fenders	1.233,00	1,23
Helmets	4.578,32	4,58
Hydration Packs	20.560,00	20,56
Lights	4.458,25	4,46
Locks	773,25	0,77
Panniers	11.601,00	11,60
Pumps	5.050,00	5,05
Tires and Tubes	12.582.947,35	12.582,95
Clothing		
Bib-Shorts	113.155,00	113,16
Caps	4.359,60	4,36
Gloves	61.524,00	61,52
Jerseys	1.067.157,00	1.067,16
Shorts	535.755,00	535,76
Socks	38.780,00	38,78
Tights	69.615,00	69,62
Vests	178.125,00	178,13
Components		
Brakes	4.555.897,50	4.555,90
Chains	47.218,50	47,22
Pedals	13.413.430,80	13.413,43
Saddles	8.725.476,37	8.725,48



CategoryName	Total Purchases Amount	Total Purchases Amount units
Accessories		
Bike Racks	8.470,00	8.470,00
Bike Stands	23.784,00	23.784,00
Bottles and Cages	8.035,00	8.035,00
Cleaners	445,50	445,50
Fenders	1.233,00	1.233,00
Helmets	4.578,32	4.578,32
Hydration Packs	20.560,00	20.560,00
Lights	4.458,25	4.458,25
Locks	773,25	773,25
Panniers	11.601,00	11.601,00
Pumps	5.050,00	5.050,00
Tires and Tubes	12.582.947,35	12.582.947,35
Clothing		
Bib-Shorts	113.155,00	113.155,00
Caps	4.359,60	4.359,60
Gloves	61.524,00	61.524,00
Jerseys	1.067.157,00	1.067.157,00
Shorts	535.755,00	535.755,00
Socks	38.780,00	38.780,00
Tights	69.615,00	69.615,00
Vests	178.125,00	178.125,00
Components		
Brakes	4.555.897,50	4.555.897,50
Chains	47.218,50	47.218,50
Pedals	13.413.430,80	13.413.430,80
Saddles	8.725.476,37	8.725.476,37

Tag
Thousand
Unit

Como variación de este ejemplo, para valores más altos, se podrían añadir otras unidades como millones, billones, etc.

14. Limitar una tabla según el valor seleccionado

El objetivo de este apartado es mostrar cómo se puede limitar el número de filas mostradas en un objeto de tipo tabla o matriz según la opción que se seleccione en el slicer.



Desarrollo:

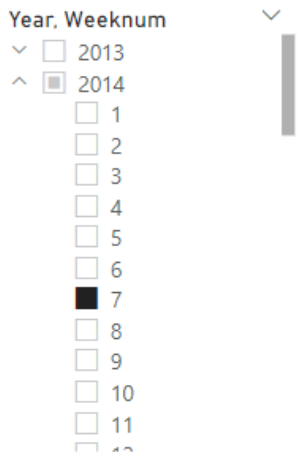
Partiendo de la siguiente tabla, lo que queremos es que se limite el número de filas mostradas según el valor de la **semana** seleccionado en el slicer.

Year	Week num	Total Purchases Amount
2013	8	85.070,30
2013	9	46.415,49
2013	16	43.600,17
2013	17	79.567,60
2013	19	517.929,63
2013	21	152.589,35
2013	22	78.577,58
2013	23	388.799,87
2013	24	304.161,00
2013	26	26.086,65
2013	28	78.927,86
2013	30	38.404,13
2013	32	14.069,07
2013	33	45.831,66
2013	34	1.425.564,77
2013	35	1.301.489,93
2013	36	1.212.082,16
2013	37	1.216.137,36
2013	38	1.351.625,99
2013	39	813.130,31
2013	40	635.272,53
2013	41	436.981,40
2013	42	244.075,63
2013	43	565.700,88
2013	44	663.499,03
Total		**59.461.965,99**

Para eso, en primer lugar, se debe crear una **nueva tabla** de semanas con los campos de número de semana, año y la fecha máxima. Esta tabla debe quedar **aislada** en el modelo (no la relacionamos con ninguna otra).

```
Weeks =
ADDCOLUMNS(
    SUMMARIZE(Dim_time, Dim_time[Weeknum], Dim_time[Year]
    ),
    "Max date", CALCULATE(MAX(Dim_time[DateAlternateKey])
    ))
```

A continuación, se añade un slicer en el lienzo con los campos de **año** y **semana** de la tabla nueva:



Después, se crea una nueva **medida** para limitar los valores mostrados en la tabla:

```
Purchases filtered =
VAR MaxWeekDate = SELECTEDVALUE(Weeks[Max date])
RETURN
    CALCULATE([Total Purchases Amount],
    KEEPFILTERS(
        FILTER(
        ALL(Dim_time[DateAlternateKey]),
        Dim_time[DateAlternateKey] <= MaxWeekDate
    )))
```

En primer lugar, se crea una **variable** que almacena la fecha máxima del valor de la semana seleccionado en la segmentación.

En la función **FILTER**, se eliminan los posibles filtros que puedan afectar a la tabla de fechas y se filtran aquellos valores que tengan una fecha anterior o igual a la fecha máxima almacenada en la variable creada anteriormente.

Si se dejara la medida así, sin la función **KEEPFILTERS**, el resultado sería el siguiente:

Year	Week num	Total Purchases Amount	Purchases filtered
2014	1	730.720,36	24.074.373,53
2014	2	954.592,55	24.074.373,53
2014	3	1.118.970,81	24.074.373,53
2014	4	925.520,93	24.074.373,53
2014	5	749.386,11	24.074.373,53
2014	6	1.139.442,23	24.074.373,53
2014	7	556.425,81	24.074.373,53
2014	8	1.086.945,31	24.074.373,53
2014	9	1.207.642,08	24.074.373,53
2014	10	988.358,00	24.074.373,53
2014	11	1.501.618,46	24.074.373,53
2014	12	1.312.703,75	24.074.373,53
2014	13	1.027.221,53	24.074.373,53
2014	14	1.355.514,70	24.074.373,53
2014	15	1.051.629,25	24.074.373,53
2014	16	1.021.185,13	24.074.373,53
2014	17	1.295.688,69	24.074.373,53
2014	18	1.055.187,59	24.074.373,53
2014	19	1.255.655,19	24.074.373,53
2014	20	1.450.563,29	24.074.373,53
Total		**59.461.965,99**	**24.074.373,53**



Year	Week num	Total Purchases Amount	Purchases filtered
2014	1	730.720,36	25.193.344,35
2014	2	954.592,55	25.193.344,35
2014	3	1.118.970,81	25.193.344,35
2014	4	925.520,93	25.193.344,35
2014	5	749.386,11	25.193.344,35
2014	6	1.139.442,23	25.193.344,35
2014	7	556.425,81	25.193.344,35
2014	8	1.086.945,31	25.193.344,35
2014	9	1.207.642,08	25.193.344,35
2014	10	988.358,00	25.193.344,35
2014	11	1.501.618,46	25.193.344,35
2014	12	1.312.703,75	25.193.344,35
2014	13	1.027.221,53	25.193.344,35
2014	14	1.355.514,70	25.193.344,35
2014	15	1.051.629,25	25.193.344,35
2014	16	1.021.185,13	25.193.344,35
2014	17	1.295.688,69	25.193.344,35
2014	18	1.055.187,59	25.193.344,35
2014	19	1.255.655,19	25.193.344,35
2014	20	1.450.563,29	25.193.344,35
Total		**59.461.965,99**	**25.193.344,35**



Como se puede deducir de las imágenes anteriores, el valor que devuelve la métrica es la suma de **Total Purchases Amount** hasta la semana seleccionada del slicer, repitiendo el mismo valor en todos los registros mostrados en la tabla.

Pero este no es el comportamiento deseado, sino que lo que se quiere es que se muestre el resultado únicamente hasta la semana seleccionada.

Para conseguir este comportamiento, se debe añadir la función **KEEPFILTERS** como se ha mostrado, obteniendo el resultado deseado:

Year	Week num	Total Purchases Amount	Purchases filtered
2013	45	459.715,81	459.715,81
2013	46	46.105,52	46.105,52
2013	47	576.587,41	576.587,41
2013	48	843.215,09	843.215,09
2013	49	981.375,24	981.375,24
2013	50	1.000.097,84	1.000.097,84
2013	51	1.075.098,09	1.075.098,09
2013	52	1.009.589,68	1.009.589,68
2013	53	301.656,76	301.656,76
2014	1	730.720,36	730.720,36
2014	2	954.592,55	954.592,55
2014	3	1.118.970,81	1.118.970,81
2014	4	925.520,93	
2014	5	749.386,11	
2014	6	1.139.442,23	
2014	7	556.425,81	
2014	8	1.086.945,31	
2014	9	1.207.642,08	
2014	10	988.358,00	
2014	11	1.501.618,46	
2014	12	1.312.703,75	
2014	13	1.027.221,53	
2014	14	1.355.514,70	
2014	15	1.051.629,25	
2014	16	1.021.185,13	
Total		**59.461.965,99**	**20.863.315,50**



Ahora se muestra el valor de **Total Purchases Amount** correctamente de todos los registros hasta la semana seleccionada en la segmentación.

Si quitamos la métrica **Total Purchases Amount** de la tabla, se muestran los datos hasta la semana seleccionada, sin valores en blanco, que es el resultado buscado:

Year	Week num	Purchases filtered
2013	9	46.415,49
2013	16	43.600,17
2013	17	79.567,60
2013	19	517.929,63
2013	21	152.589,35
2013	22	78.577,58
2013	23	388.799,87
2013	24	304.161,00
2013	26	26.086,65
2013	28	78.927,86
2013	30	38.404,13
2013	32	14.069,07
2013	33	45.831,66
2013	34	1.425.564,77
2013	35	1.301.489,93
2013	36	1.212.082,16
2013	37	1.216.137,36
2013	38	1.351.625,99
2013	39	813.130,31
2013	40	635.272,53
2013	41	436.981,40
2013	42	244.075,63
2013	43	565.700,88
2013	44	663.499,03
2013	45	459.715,81
2013	46	46.105,52
2013	47	576.587,41
2013	48	843.215,09
2013	49	981.375,24
2013	50	1.000.097,84
2013	51	1.075.098,09
2013	52	1.009.589,68
2013	53	301.656,76
2014	1	730.720,36
2014	2	954.592,55
2014	3	1.118.970,81
Total		**20.863.315,50**

15. Resaltar valores de una tabla según el valor seleccionado

El objetivo de este tip es mostrar cómo se puede resaltar un valor mostrado en una visualización de tabla o matriz según la opción que se seleccione en el slicer.



Desarrollo:

Partiendo de la tabla creada en el apartado anterior, lo que queremos es que, en vez de que se filtren los valores de la tabla, se **resalte** el valor de la tabla según la semana que se seleccione en la segmentación.

Para ello, vamos a crear otra **medida** de la siguiente forma:

```
Highlighted value =
VAR currentYear = SELECTEDVALUE(Weeks[Year])
VAR currentWeek = SELECTEDVALUE(Weeks[Weeknum])

RETURN
SELECTEDVALUE(Dim_time[Year])=currentYear && SELECTEDVALUE(Dim_time[Weeknum])=currentWeek
```

Esta medida devuelve verdadero o falso si el año y semana de cada fila son iguales al año y semana seleccionados en la segmentación.

Year	Week num	Total Purchases Amount	Highlighted value
2014	1	730.720,36	False
2014	2	954.592,55	False
2014	3	1.118.970,81	False
2014	4	925.520,93	True
2014	5	749.386,11	False
2014	6	1.139.442,23	False
2014	7	556.425,81	False
2014	8	1.086.945,31	False
2014	9	1.207.642,08	False
2014	10	988.358,00	False
2014	11	1.501.618,46	False
2014	12	1.312.703,75	False
2014	13	1.027.221,53	False



A continuación, para hacer que la medida devuelva el valor de un color en vez de verdadero o falso, hay que modificar la medida añadiendo la función **IF**. En el caso de que la expresión sea verdadera, la medida va a devolver el valor del color, "**Light green**" en este caso.

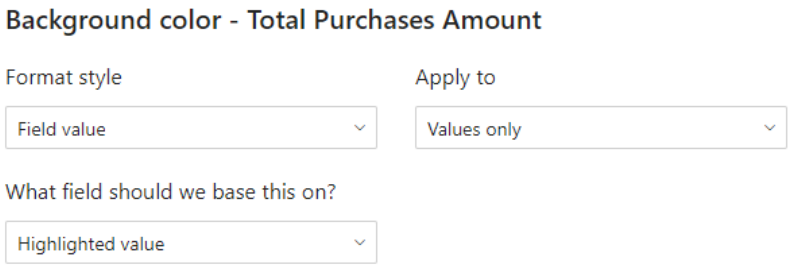
```
Highlighted value =
VAR currentYear = SELECTEDVALUE(Weeks[Year])
VAR currentWeek = SELECTEDVALUE(Weeks[Weeknum])

RETURN
IF(
    SELECTEDVALUE(Dim_time[Year])=currentYear && SELECTEDVALUE(Dim_time[Weeknum])=currentWeek,
    "light green"
)
```

Year	Week num	Total Purchases Amount	Highlighted value
2014	1	730.720,36	
2014	2	954.592,55	
2014	3	1.118.970,81	
2014	4	925.520,93	light green
2014	5	749.386,11	
2014	6	1.139.442,23	
2014	7	556.425,81	
2014	8	1.086.945,31	
2014	9	1.207.642,08	
2014	10	988.358,00	
2014	11	1.501.618,46	
2014	12	1.312.703,75	
2014	13	1.027.221,53	



Una vez que tenemos este resultado, solo falta aplicar el **formato condicional**. En este caso, vamos a aplicarlo sobre el color de fondo de **Total Purchases Amount** de la siguiente manera:



Como último paso, quitamos en la tabla la medida creada para obtener el color, obteniendo el resultado buscado:

Year	Week num	Total Purchases Amount
2014	1	730.720,36
2014	2	954.592,55
2014	3	1.118.970,81
2014	4	925.520,93
2014	5	749.386,11
2014	6	1.139.442,23
2014	7	556.425,81
2014	8	1.086.945,31
2014	9	1.207.642,08
2014	10	988.358,00
2014	11	1.501.618,46
2014	12	1.312.703,75



Year	Week num	Total Purchases Amount
2014	1	730.720,36
2014	2	954.592,55
2014	3	1.118.970,81
2014	4	925.520,93
2014	5	749.386,11
2014	6	1.139.442,23
2014	7	556.425,81
2014	8	1.086.945,31
2014	9	1.207.642,08
2014	10	988.358,00
2014	11	1.501.618,46
2014	12	1.312.703,75



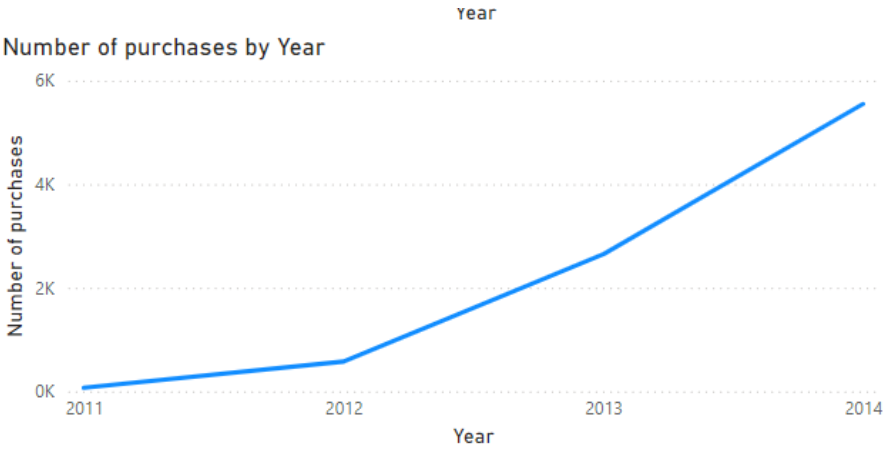
16. Añadir formato condicional a un gráfico de líneas

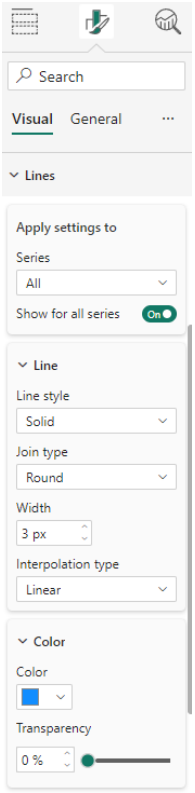
El objetivo de este tip es mostrar cómo se puede añadir el formato condicional a un gráfico de líneas, ya que, actualmente, no se puede aplicar esta funcionalidad a este tipo de visualización.



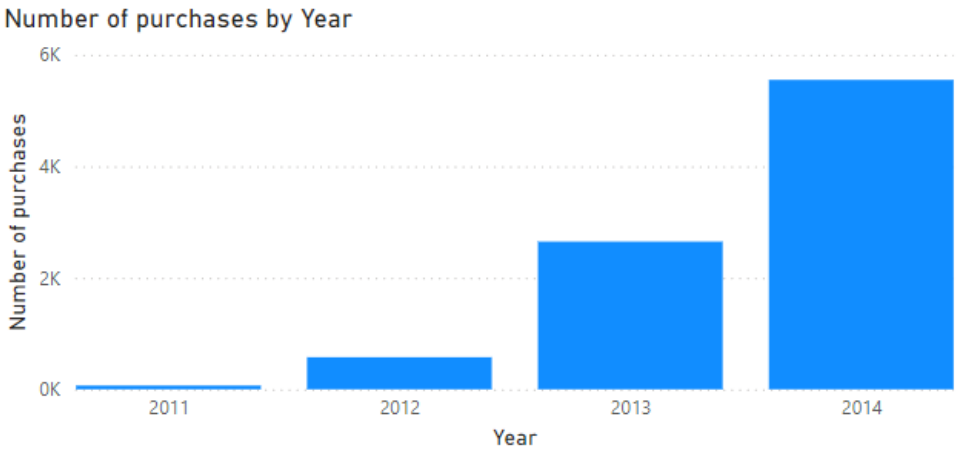
Desarrollo:

Por defecto, si se necesita añadir formato condicional a un gráfico de líneas, actualmente no está esa opción habilitada, mientras que en otros tipos de visualizaciones sí:

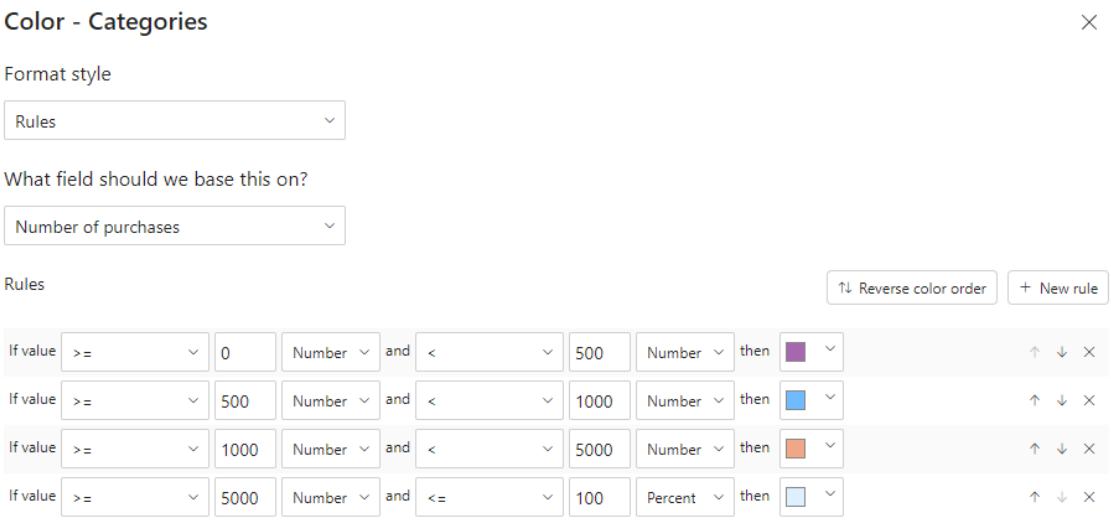


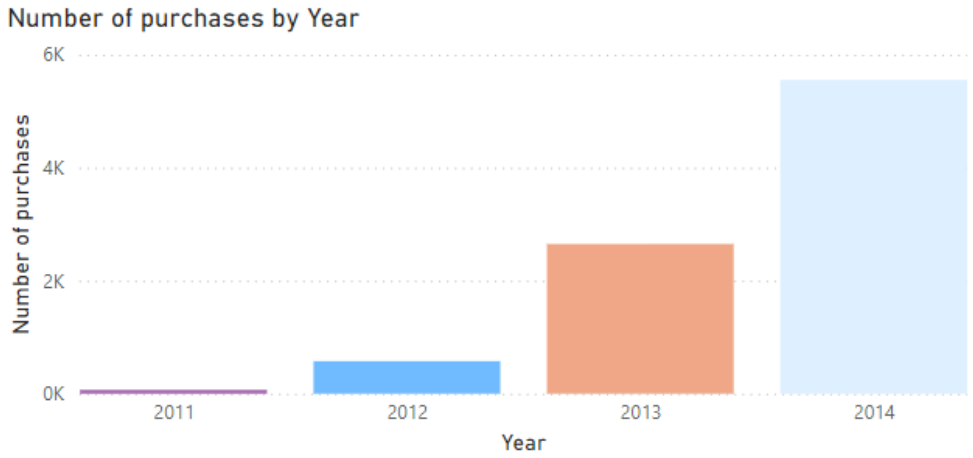


Pero existe una alternativa para poder aplicar el formato condicional en este tipo de gráfico. Lo primero que se debe hacer es convertir el **gráfico de líneas** en otro tipo de visualización que sí admita formato condicional, por ejemplo, un **gráfico de columnas**.

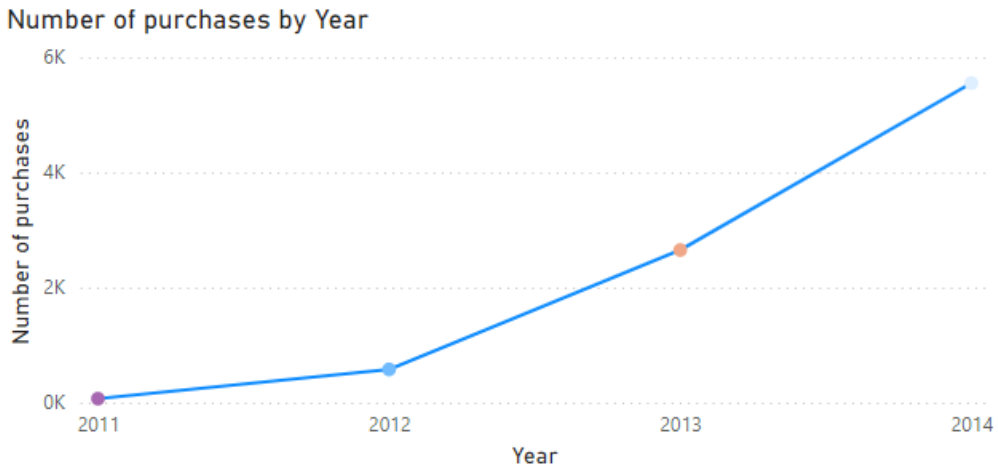


A continuación, en las opciones de columna, configuramos el formato condicional que necesitemos, en este caso:



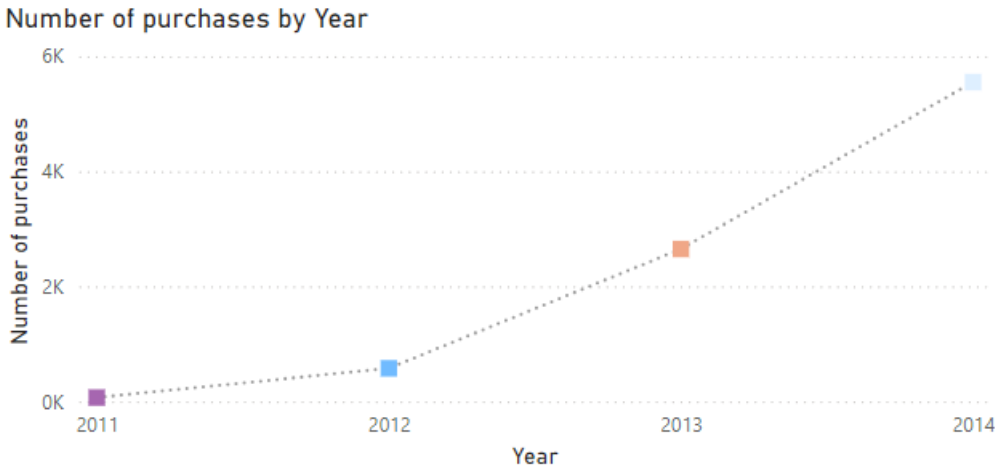


Como último paso, quedaría **volver a convertir** el gráfico en un gráfico de líneas.



Como se puede ver, los marcadores de cada uno de los valores ahora tienen los colores establecidos en el formato condicional.

Para hacer que sea más visible el formato condicional, se puede cambiar el color y tipo de la línea, la forma y tamaño del marcador, etc.



17. Crear una jerarquía en un slicer con field parameters

El objetivo de este tip es mostrar cómo se puede crear una segmentación con una jerarquía de Field parameters.



Desarrollo:

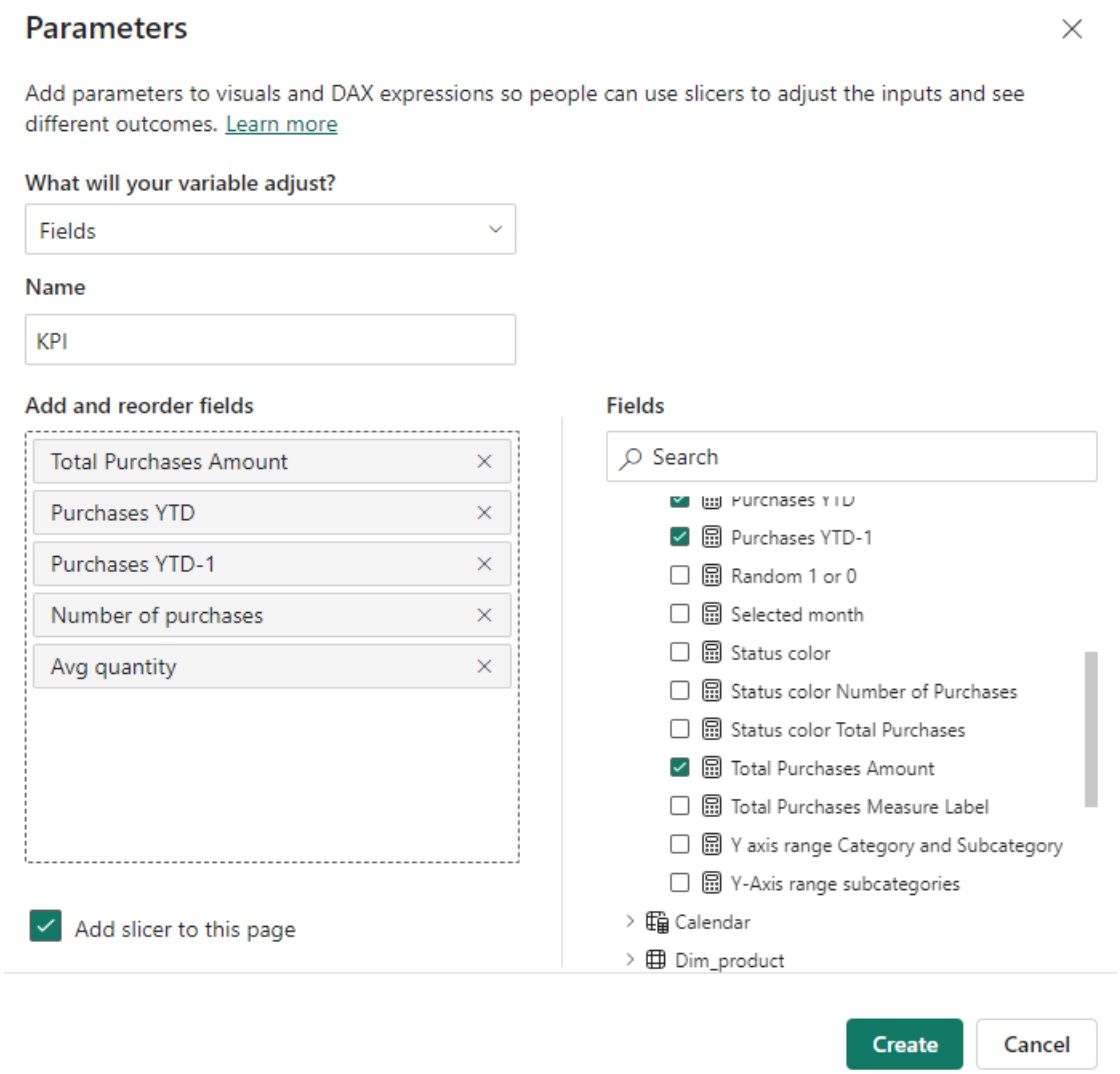
Partiendo de la siguiente matriz que contiene 5 medidas, se requiere categorizar las 3 primeras como **Amount** y las otras 2, como **Quantity**.

Year	MonthName	Total Purchases Amount	Purchases YTD	Purchases YTD-1	Number of purchases	Avg quantity
2011	abril					
	mayo					
	junio		103.895,82			
	julio		103.895,82			
	agosto		103.895,82			
	septiembre		103.895,82			
	octubre		103.895,82			
	noviembre		103.895,82			
	diciembre	299.239,98	403.135,80		53	238,66
2012	enero	354.968,23	354.968,23		43	304,86
	febrero	622.885,33	977.853,56		92	211,58
	marzo	649.928,94	1.627.782,49		92	273,28
	abril	346.917,55	1.974.700,05	9.492	70	159,91
	mayo	106.491,00	2.081.191,05	94.404	7	550,00
	junio	171.578,43	2.252.769,48		47	115,02
	julio	483.826,64	2.736.596,12		61	355,52
	agosto	291.046,13	3.027.642,25		56	156,64

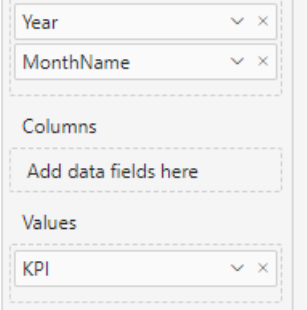
AMOUNT

QUANTITY

Para esto, en primer lugar, vamos a crear los field parameters necesarios:



El siguiente paso es sustituir las medidas de la matriz por las creadas en la tabla KPI para que la segmentación creada afecte a la matriz:



El siguiente paso, para crear la jerarquía, tenemos que modificar la definición de la tabla KPI de la siguiente manera:

```
KPI = {
    ("Total Purchases Amount", NAMEOF('Metrics'[Total Purchases Amount]), 0),
    ("Purchases YTD", NAMEOF('Metrics'[Purchases YTD]), 1),
    ("Purchases YTD-1", NAMEOF('Metrics'[Purchases YTD-1]), 2),
    ("Number of purchases", NAMEOF('Metrics'[Number of purchases]), 3),
    ("Avg quantity", NAMEOF('Metrics'[Avg quantity]), 4)
}
```

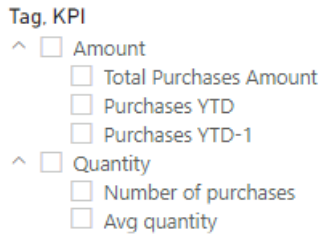
KPI	KPI Fields	KPI Order
Total Purchases Amount	'Metrics'[Total Purchases Amount]	0
Purchases YTD	'Metrics'[Purchases YTD]	1
Purchases YTD-1	'Metrics'[Purchases YTD-1]	2
Number of purchases	'Metrics'[Number of purchases]	3
Avg quantity	'Metrics'[Avg quantity]	4



```
KPI = {
    ("Total Purchases Amount", NAMEOF('Metrics'[Total Purchases Amount]), 0, "Amount"),
    ("Purchases YTD", NAMEOF('Metrics'[Purchases YTD]), 1, "Amount"),
    ("Purchases YTD-1", NAMEOF('Metrics'[Purchases YTD-1]), 2, "Amount"),
    ("Number of purchases", NAMEOF('Metrics'[Number of purchases]), 3, "Quantity"),
    ("Avg quantity", NAMEOF('Metrics'[Avg quantity]), 4, "Quantity")
}
```

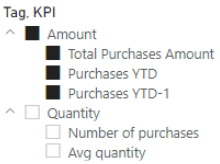
KPI	KPI Fields	KPI Order	Value4
Total Purchases Amount	'Metrics'[Total Purchases Amount]	0	Amount
Purchases YTD	'Metrics'[Purchases YTD]	1	Amount
Purchases YTD-1	'Metrics'[Purchases YTD-1]	2	Amount
Number of purchases	'Metrics'[Number of purchases]	3	Quantity
Avg quantity	'Metrics'[Avg quantity]	4	Quantity

Por último, añadimos este campo nuevo en la segmentación:



De esta forma, conseguimos tener la jerarquía buscada, pudiendo filtrar la matriz por la categoría correspondiente.

Year	MonthName	Total Purchases Amount	Purchases YTD	Purchases YTD-1
2011	abril	9.491,52	9.491,52	
	mayo	94.404,30	103.895,82	
	junio		103.895,82	
	julio		103.895,82	
	agosto		103.895,82	
	septiembre		103.895,82	
	octubre		103.895,82	
	noviembre		103.895,82	
	diciembre	299.239,98	403.135,80	
2012	enero	354.968,23	354.968,23	
	febrero	622.885,33	977.853,56	
	marzo	649.928,94	1.627.782,49	



4. OTROS

18. Ordenar un objeto visual tipo tabla por dos columnas

El propósito de este tip es mostrar cómo se puede ordenar una tabla por dos columnas.



Desarrollo:

Partiendo de un objeto visual de tipo tabla, puede darse el caso de que se necesite ordenar la tabla por más de una columna. Si no se hace nada especial, solo se puede ordenar por una columna, ascendente o descendentemente, dependiendo del número de veces que seleccionemos la columna.

CategoryName	SubcategoryName	Class	Number of purchases
Accessories	Bike Racks	No consta	1
Accessories	Bike Stands	No consta	1
Accessories	Bottles and Cages	No consta	3
Accessories	Cleaners	No consta	1
Accessories	Fenders	No consta	1
Accessories	Helmets	No consta	3
Accessories	Hydration Packs	No consta	1
Accessories	Lights	No consta	3
Accessories	Locks	No consta	1
Accessories	Panniers	No consta	1
Accessories	Pumps	No consta	2
Accessories	Tires and Tubes	H	160
Accessories	Tires and Tubes	L	175
Accessories	Tires and Tubes	M	175
Accessories	Tires and Tubes	No consta	194
Clothing	Bib-Shorts	No consta	3
Clothing	Caps	No consta	1
Clothing	Gloves	No consta	6
Clothing	Jerseys	No consta	8
Clothing	Shorts	No consta	7
Clothing	Socks	No consta	4
Clothing	Tights	No consta	3
Clothing	Vests	No consta	6
Components	Brakes	No consta	100
Components	Chains	No consta	50
Components	Pedals	H	93

Sin embargo, si se mantiene pulsada la tecla **SHIFT**, se pueden seleccionar varias columnas, quedando la tabla ordenada por ellas.

CategoryName	SubcategoryName	Class	Number of purchases
Accessories	Tires and Tubes	H	160
Accessories	Tires and Tubes	L	175
Accessories	Tires and Tubes	M	175
Accessories	Bike Racks	No consta	1
Accessories	Bike Stands	No consta	1
Accessories	Bottles and Cages	No consta	3
Accessories	Cleaners	No consta	1
Accessories	Fenders	No consta	1
Accessories	Helmets	No consta	3
Accessories	Hydration Packs	No consta	1
Accessories	Lights	No consta	3
Accessories	Locks	No consta	1
Accessories	Panniers	No consta	1
Accessories	Pumps	No consta	2
Accessories	Tires and Tubes	No consta	194
Clothing	Bib-Shorts	No consta	3
Clothing	Caps	No consta	1
Clothing	Gloves	No consta	6
Clothing	Jerseys	No consta	8
Clothing	Shorts	No consta	7
Clothing	Socks	No consta	4
Clothing	Tights	No consta	3
Clothing	Vests	No consta	6
Components	Pedals	H	93
Components	Saddles	H	171
Components	Pedals	L	192
Components	Saddles	L	172
Components	Pedals	M	192
Components	Saddles	M	172

19. Formatear 0 y 1

El propósito de este tip es mostrar cómo se puede formatear los valores 0 y 1 para que se muestren como True/False u On/Off.



Desarrollo:

Partiendo de una medida que devuelva los valores **0 o 1**, podemos formatearla para convertir este valor en **True/false**. En este caso, vamos a partir de una medida que devuelve 0 o 1 de forma aleatoria.

```
1  Random 1 or 0 = RANDBETWEEN(0,1)
```

CategoryName	Random 1 or 0
Accessories	0
Bikes	1
Clothing	1
Components	0
No consta	1
Total	**0**

Si utilizamos la función **FORMAT** en esta medida, veremos cómo se sustituyen los valores 0 por False y 1 por True.

```
1  Random 1 or 0 = FORMAT(RANDBETWEEN(0,1), "True/false")
```

CategoryName	Random 1 or 0
Accessories	Verdadero
Bikes	Verdadero
Clothing	Verdadero
Components	Falso
No consta	Verdadero
Total	**Verdadero**

También podríamos sustituirlos por On/Off.

```
Random 1 or 0 = FORMAT(RANDBETWEEN(0,1), "On/Off")
```

20. Utilizar los campos y los valores de los campos en Field parameters

El propósito de este tip es mostrar cómo se puede mostrar tanto el nombre de los campos como el valor de estos y utilizarlos como segmentación en el informe.

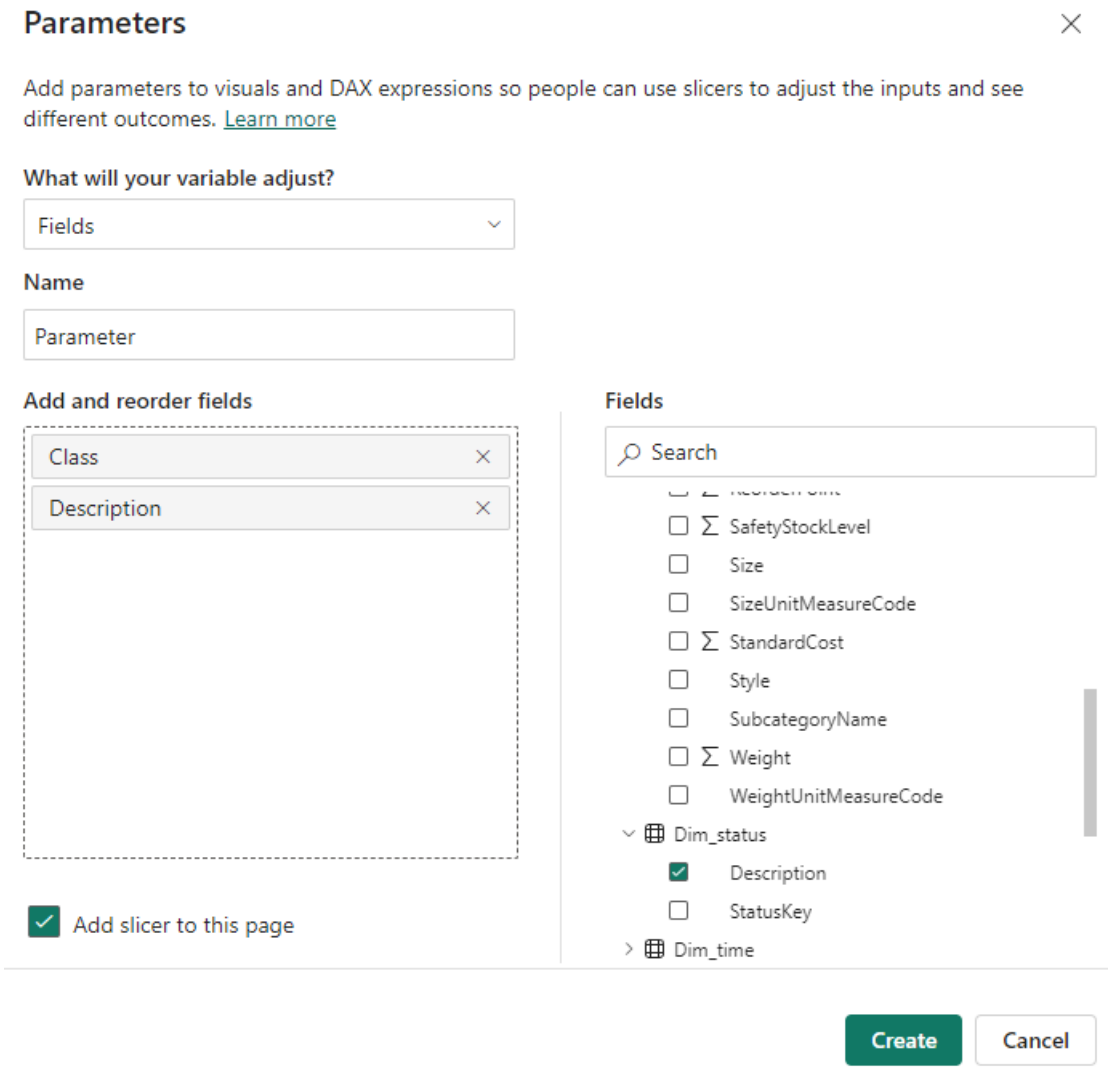


Desarrollo:

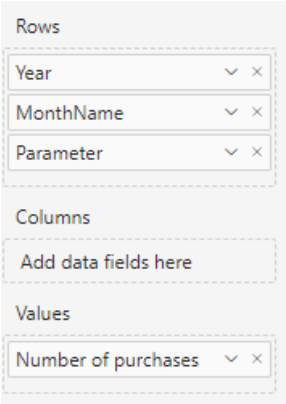
Partiendo de un objeto visual de tipo matriz como la siguiente:

Year	Number of purchases
2011	**68**
abril	5
mayo	10
diciembre	53
2012	**576**
enero	43
febrero	92
marzo	92
abril	70
mayo	7
junio	47
julio	61
agosto	56
septiembre	32
octubre	76
2013	**2650**
febrero	32
abril	19
mayo	102
junio	86
julio	25
agosto	409
septiembre	748
octubre	336
Total	**8845**

Creamos un nuevo **Field parameter**, en este caso, con el campo de **estado** y de **clase** de producto, seleccionando la opción de añadir la segmentación a la página.


Parameters
Add parameters to visuals and DAX expressions so people can use slicers to adjust the inputs and see different outcomes. Learn more
What will your variable adjust?
Fields
Name
Parameter
Add and reorder fields
Class
Description
Add slicer to this page
Fields
Search
SafetyStockLevel
Size
SizeUnitMeasureCode
StandardCost
Style
SubcategoryName
Weight
WeightUnitMeasureCode
Dim_status
Description
StatusKey
Dim_time
Create
Cancel

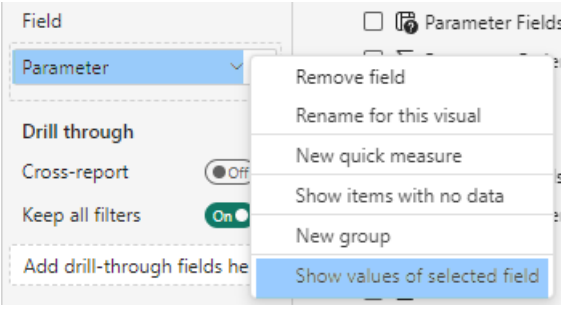
A continuación, se debe añadir el campo del **field parameter** en la matriz.



Ahora, si se selecciona cualquiera de los valores de la segmentación, se puede ver que sí se actualiza la matriz según la selección.

Pero además de esta segmentación, vamos a añadir una **segunda segmentación** de forma que se pueda ver los valores del campo seleccionado en la segmentación.

Para esto, copiamos y pegamos la segmentación, seleccionamos el nombre del campo y la opción de Mostrar valores del campo seleccionado.



De esta forma, podemos ver los valores de cada una de las opciones de la primera segmentación.



21. Funcionalidades de la vista DAX Query

El propósito de este tip es mostrar diferentes funcionalidades de la pestaña de DAX View.

Desarrollo:

En este apartado se van a explicar varias funcionalidades de la vista de consultas DAX de Power BI.

1. Crear varias medidas simultáneamente

Partiendo de una query en blanco, se pueden crear varias métricas simultáneamente escribiendo las fórmulas DAX de todas ellas utilizando la función **DEFINE**.

Una vez escritas, aparece la opción **Update model: add new measure**.



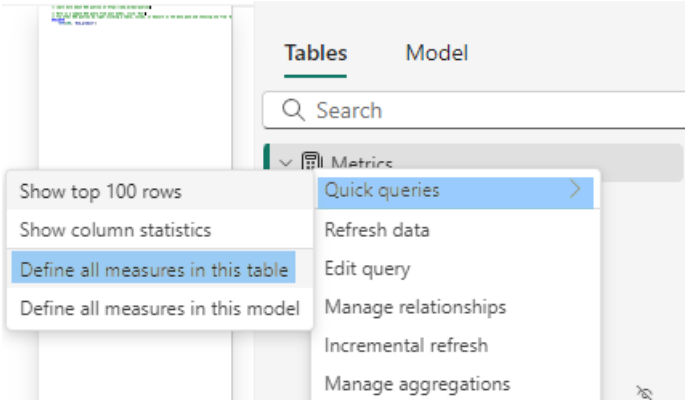
Se puede ir seleccionando esta opción por cada una de las medidas o se puede seleccionar el botón **Update model with changes** que aparece arriba para que se añadan al modelo todas las medidas a la vez.



Esto evita tener que estar creando las medidas una por una haciendo click derecho sobre la tabla de medidas, nueva medida.

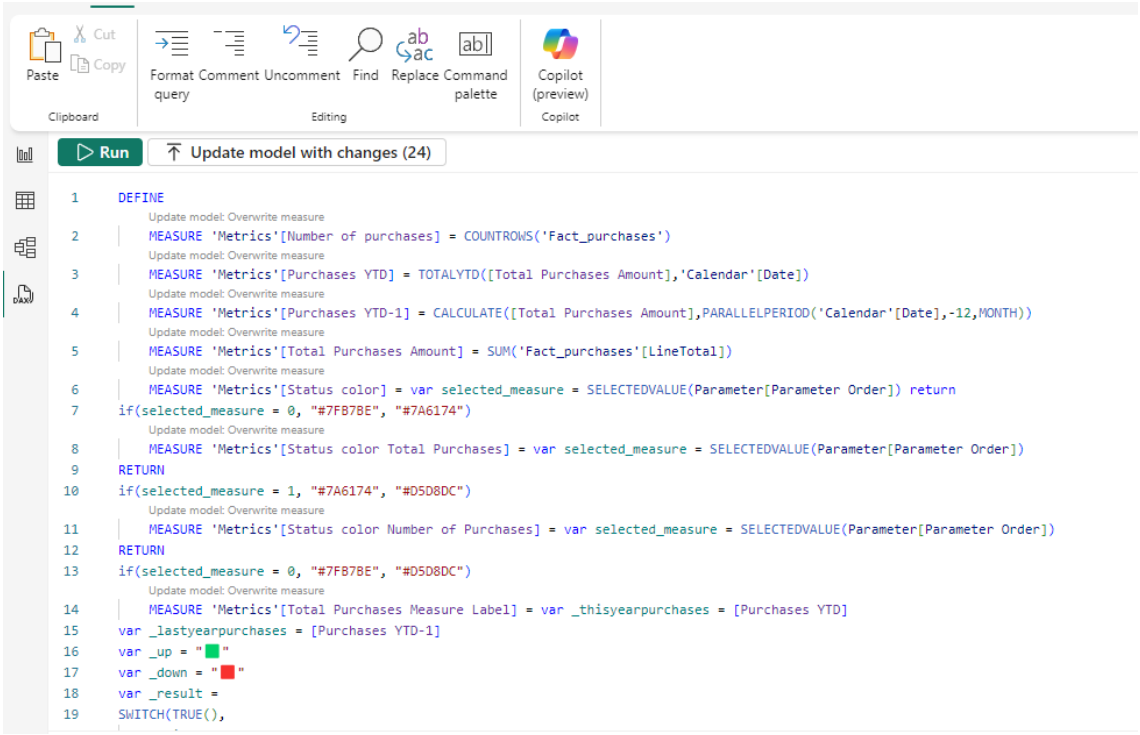
2. Obtener la definición de las medidas del modelo.

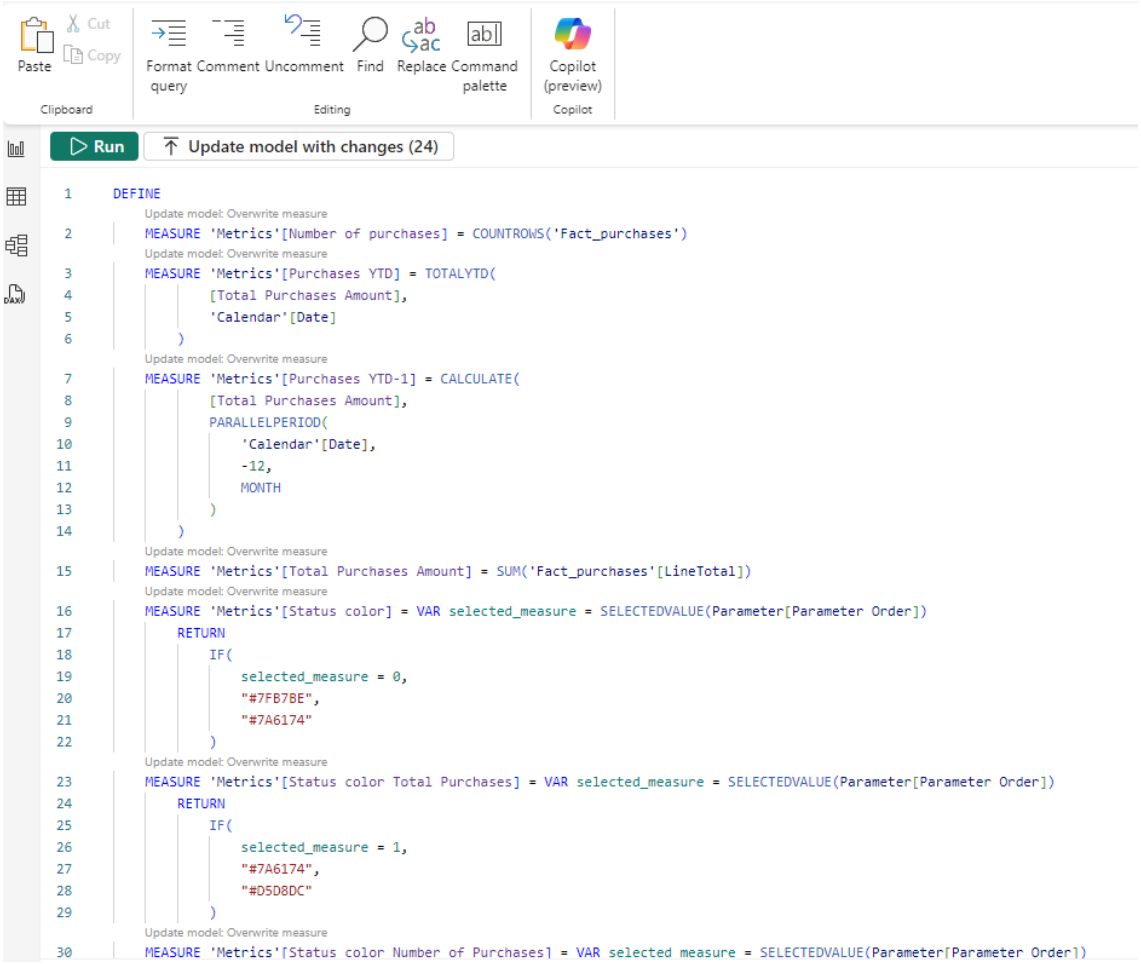
Si se tienen medidas creadas en el modelo, se puede obtener la definición de todas ellas haciendo clic derecho sobre una tabla que contenga medidas y elegir entre definir las medidas de esa tabla o del modelo.



Teniendo la definición de todas las medidas, se pueden copiar y pegar en otro archivo .pbix de forma rápida y sencilla y crear las medidas en el modelo como se ha visto en el punto 1.

Una vez obtenida la definición de las medidas, se pueden formatear desde el botón **Format Query**:

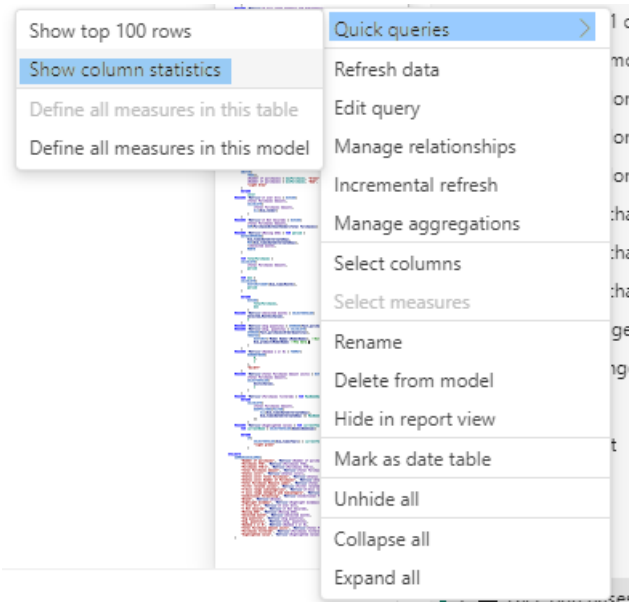




Además, se puede añadir comentarios, buscar o reemplazar.

3. Obtener estadísticas de columnas

Mediante la opción Mostrar estadísticas de columnas, se pueden obtener diferentes estadísticas de la tabla como el recuento, valores distintos, recuento de nulos, valores mínimo, máximo, mediana, media, desviación estándar, etc.



Results | Result 1 of 1 | Copy

	[Table]	[Column]	[Count]	[Distinct Values]	[Null Count]	[Min]	[Max]	[Median]	[Mean]	[Standard Deviation]
1	Fact_purchases	PurchaseKey	8845	8845	0	1	8845	4423	4423	2553,33154917257
2	Fact_purchases	ProductKey	8845	265	0	1	457	135	172,47687959299	149,503520747408
3	Fact_purchases	VendorKey	8845	86	0	1	104	51	53,2169587337479	30,5479571928979
4	Fact_purchases	StatusKey	8845	4	0	1	4	4	3,80927077444884	0,700580377708193
5	Fact_purchases	OrderQuantity	8845	28	0	3	8000	60	265,532730356133	355,906468517934
6	Fact_purchases	ReceivedQuantity	8845	34	0	2	8000	60	263,120293951385	354,021057906419
7	Fact_purchases	RejectedQuantity	8845	13	0	0	1250	0	8,21933295647258	58,3122881573739
8	Fact_purchases	UnitPrice	8845	177	0	0,21	82,8345	39,2805	34,7429641153194	16,3145169865 49
9	Fact_purchases	LineTotal	8845	183	0	37,0755	249420	262,5	7212,20970469192	11542,1516830202
10	Fact_purchases	EmployeeTitle	8845	3	0	Buyer	Purchasing Manager	N/A	N/A	N/A
11	Fact_purchases	OrderDateKey	8845	300	0	20110416	20140922	20140212	20135785,5083098	6378,87672811601
12	Fact_purchases	ShipDateKey	8845	297	0	20110425	20141017	20140221	20135991,2703222	6373,18991459536
13	Fact_purchases	DueDateKey	8845	299	0	20110430	20141022	20140226	20136103,7984172	6368,79204598486
14	Fact_purchases	OrderDate	8845	300	0	16/04/2011	22/09/2014	N/A	N/A	N/A
15	Fact_purchases	ShipDate	8845	297	0	25/04/2011	17/10/2014	N/A	N/A	N/A
16	Fact_purchases	DueDate	8845	299	0	30/04/2011	22/10/2014	N/A	N/A	N/A

22. Cómo utilizar Copilot en Power BI

El propósito de este tip es mostrar diferentes funcionalidades de Copilot tanto en Power BI Desktop como en el servicio de Power BI.



Desarrollo:

En este apartado se van a explicar diferentes funcionalidades de Copilot en la herramienta de escritorio de Power BI y en el servicio.

1. **Crear medidas a partir de lenguaje natural y editarlas**. Uno de los usos que tiene Copilot en Power BI Desktop es obtener el código DAX de medidas partiendo del lenguaje natural. Así, desde la vista de consultas DAX, podríamos escribir:

```
Write a measure that returns Order Amount for the employee Gabi and
show that measure in a query broken down by product
```

Obteniendo el código DAX, así como el resultado de la consulta:

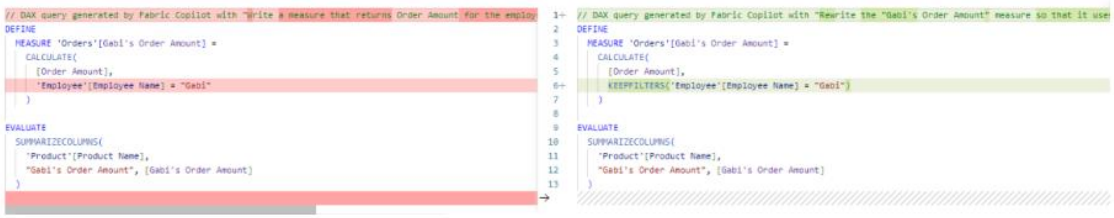
```
// DAX query generated by Fabric Copilot with "Write a measure
DEFINE
    Update model: Add new measure
    MEASURE 'Orders'[Gabi's Order Amount] =
        CALCULATE(
            [Order Amount],
            'Employee'[Employee Name] = "Gabi"
        )

EVALUATE
    SUMMARIZECOLUMNS(
        'Product'[Product Name],
        "Gabi's Order Amount", [Gabi's Order Amount]
    )
```

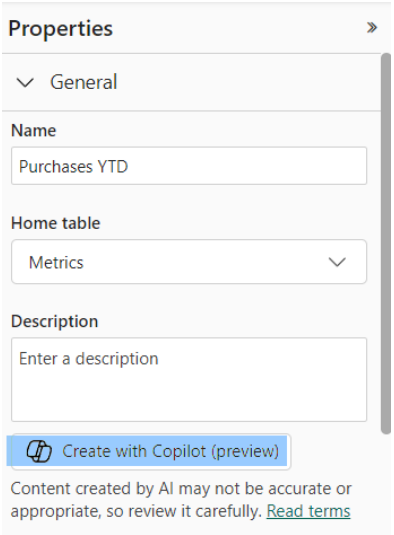
Results | Result 1 of 1 | Copy

	Product[Product Name]	[Gabi's Order Amount]
1	Apples	2600
2	Lemons	800
3	Limes	810
4	Bananas	930

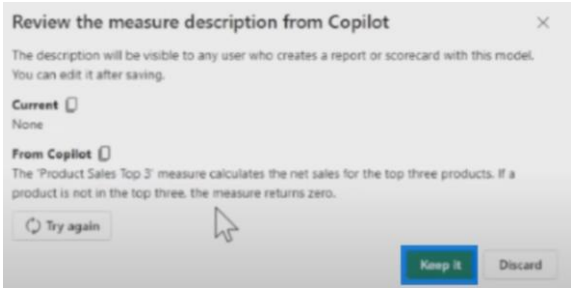
También se puede pedir a Copilot que **modifique** una medida creada desde la vista de consultas DAX. En este caso, mostrará una comparativa de la medida actual con la nueva sugerencia señalando los cambios:



2. **Crear descripciones de medidas del modelo**. Desde la vista de modelo, en el panel de propiedades, hay disponible una casilla para agregar una descripción a las medidas o a los campos. Justo debajo de esta casilla, al seleccionar una medida, aparece una opción para generar esta descripción con Copilot. Esta funcionalidad no está disponible para los campos del modelo, solo para las medidas.

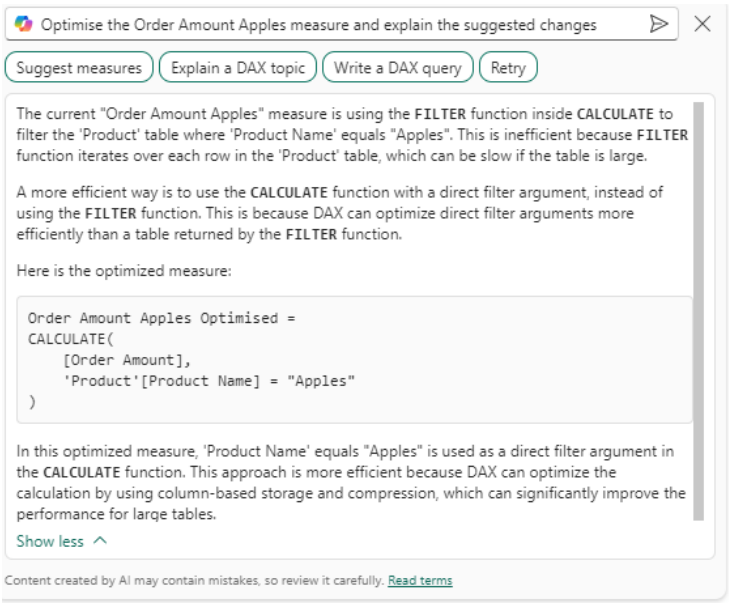


Al seleccionar esta opción, aparecerá una ventana flotante en la que se muestra la descripción actual, la descripción propuesta por Copilot y botones para regenerar la descripción, añadirla o descartar la sugerencia.

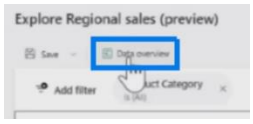


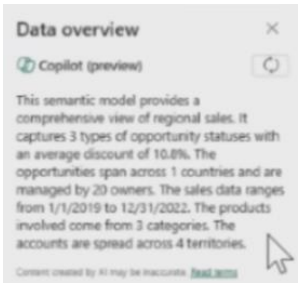
3. **Proporcionar explicaciones**. Además de crear medidas, puedes pedir a Copilot que te explique el funcionamiento de una medida del modelo o que explique cualquier concepto general como, por ejemplo, el uso de una función DAX o cuáles son las principales funciones DAX.

4. **Sugerir mejoras de las medidas.** También se puede pedir a Copilot que muestre sugerencias de mejora de una medida que se haya creado previamente en el modelo semántico.



5. **Explicar el modelo semántico**. Desde la funcionalidad de Explorar datos del servicio de Power BI está disponible la opción de obtener una explicación del modelo semántico para poder conocer mejor el modelo y realizar un mejor análisis. Esta opción está disponible desde el botón de **Data Overview** y se generará un cuadro como el siguiente:





5. MAS TRUCOS, VIDEOTUTORIALES Y PAPERS

1. Integracion SAP - PowerBI
2. PowerBI Trucos (Vol I)
3. PowerBI Trucos (Vol II)
4. Power BI Trucos (Vol III)
5. Power BI Trucos (Vol IV)
6. PowerBI + Synapse Analytics (paso a paso)
7. 30 Consejos y Buenas Prácticas para hacer un proyecto de Power BI con éxito
8. Cómo crear diseños de Dashboards espectaculares con PowerBI
9. Videotutorial: Trabajando con Python en Power BI
10. Aplicación PowerBi Turismo
11. Aplicación PowerBI Financiera I
12. Aplicación PowerBI Financiera II
13. Aplicación PowerBI eCommerce
14. Aplicación PowerBI Salud
15. Aplicación PowerBi Smart City
16. Aplicación PowerBI Energía
17. Aplicación PowerBI Sports Analytcis
18. Power BI Premium Utilization and Metrics
19. PowerBI Embedded: Funcionamiento y costes
20. Bravo para PowerBI
21. Como integrar Power BI con Microsoft Dynamicssalesfo
22. SQL Server Profiler para Power BI
23. Como usar Report Analyzer en PowerBI, para mejorar el rendimiento
24. Power BI embebido en Jupyter Notebook
25. Tabular Editor para Power BI: Videotutorial y manual en español
26. Personaliza tus gráficas en Power BI con Charticulator y Deneb
27. Comparativa PowerBI vs Amazon QuickSight
28. Como usar emoticonos en PowerBI
29. Buenas prácticas con Dataflows en Power BI
30. Power Automate para Power BI: Cómo funciona
31. ALM Toolkit para Power BI
32. Os presentamos Goals in Power BI para hacer Scorecards
33. Tutorial gratuito en español sobre Power BI Report Builder
34. Conoce PowerBI Diagram View (Visual Data Prep). Paso a paso
35. Futbol Analytics, lo que hay que saber
36. Dashboard de medicion de la calidad del aire en Madrid
37. Como funciona Microsoft Power BI? Videoturial de Introducción
38. Big Data para PowerBI
39. Quieres crear aplicaciones empresariales usando PowerBI, PowerApps y Power Automate de forma conjunta?
40. Power BI tip: Uso de parámetros what-if
41. Como integrar Salesforce y PowerBI
42. Videotutorial: Usando R para Machine Learning con PowerBI
43. Las 50 claves para aprender y conocer PowerBI

44. PowerBI: Arquitectura End to End
45. Usando Python con PowerBI
46. PowerBI + Open Source = Sports Analytics
47. Comparativa de herramientas Business Intelligence
48. Use Case Big Data “Dashboards with Hadoop and Power BI”
49. Todas las presentaciones del Workshop ‘El Business Intelligence del Futuro’
50. Descarga Paper gratuito: Zero to beautiful (Data visualization)
51. SAP connection tools for process automation: Microsoft, Pentaho, Talend (User Guide)

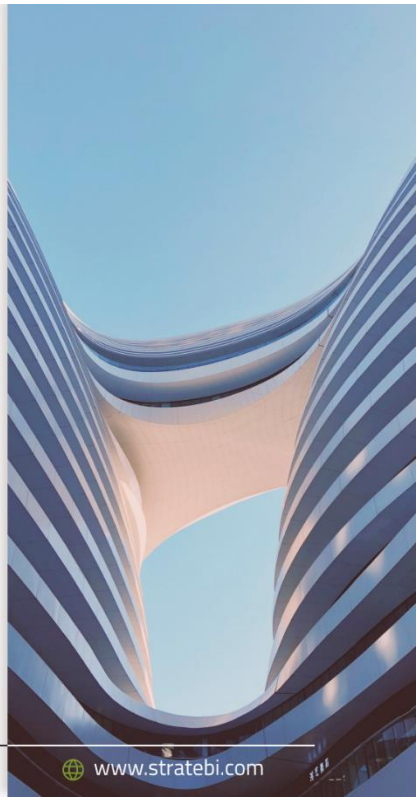
stratebi Analytics and Big Data

Microsoft Gold Partner

stratebi Analytics and Big Data

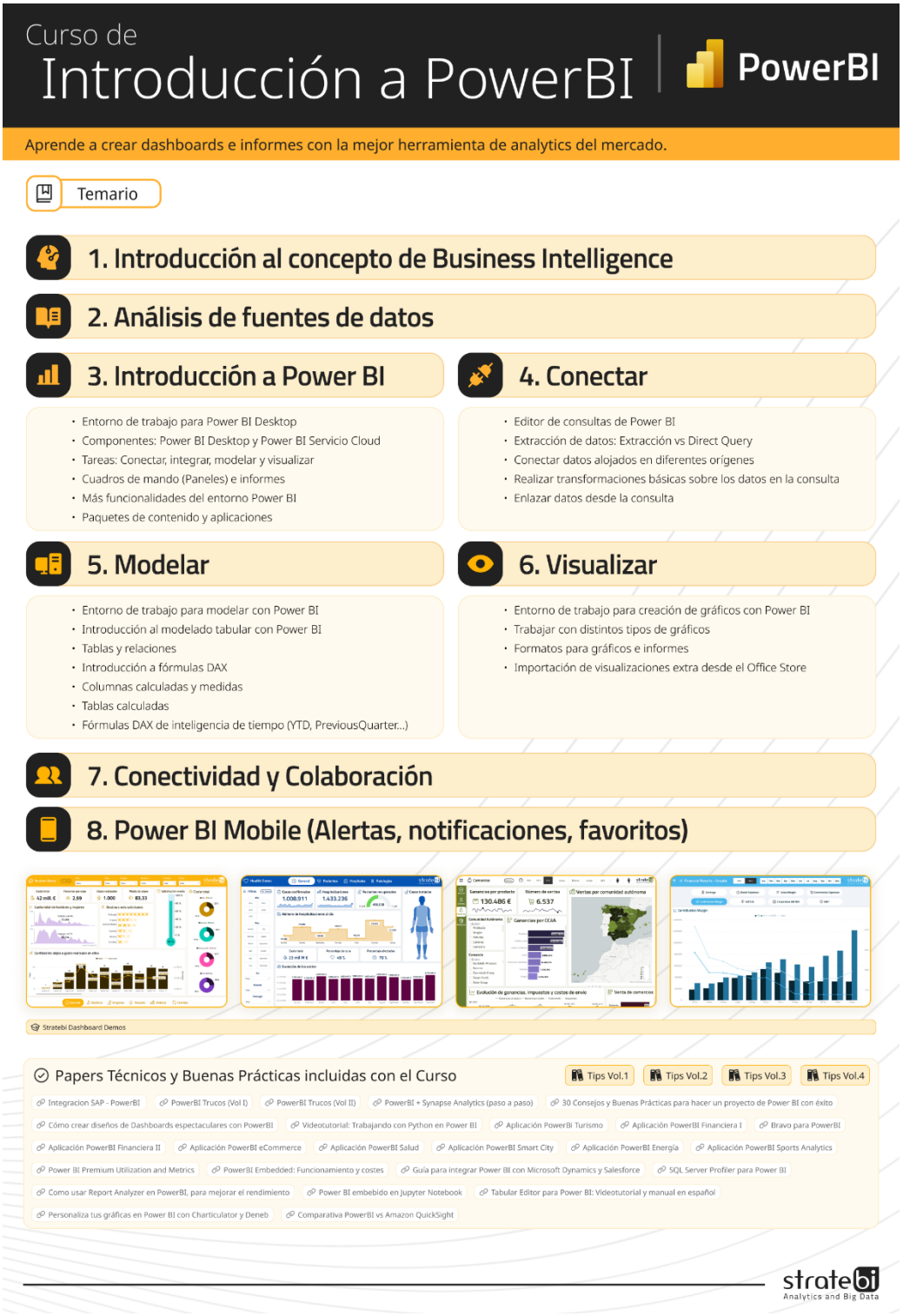
Tecnología avanzada al servicio de tus datos.
Partners certificados:

Azure aws snowflake PowerBI
databricks LinceBI talend
Qlik VERTICA ClickHouse Google Cloud
Streamlit KYLIGENCE DAMA International
Microsoft Fabric dbt



Avda. de Brasil, 17 - Madrid (+34) 917 88 34 10 www.stratebi.com

6. CURSOS


Curso de
Introducción a PowerBI
PowerBI
Aprende a crear dashboards e informes con la mejor herramienta de analytics del mercado.
Temario
1. Introducción al concepto de Business Intelligence
2. Análisis de fuentes de datos
3. Introducción a Power BI
Entorno de trabajo para Power BI Desktop
Componentes: Power BI Desktop y Power BI Servicio Cloud
Tareas: Conectar, integrar, modelar y visualizar
Cuadros de mando (Paneles) e informes
Más funcionalidades del entorno Power BI
Paquetes de contenido y aplicaciones
4. Conectar
Editor de consultas de Power BI
Extracción de datos: Extracción vs Direct Query
Conectar datos alojados en diferentes orígenes
Realizar transformaciones básicas sobre los datos en la consulta
Enlazar datos desde la consulta
5. Modelar
Entorno de trabajo para modelar con Power BI
Introducción al modelado tabular con Power BI
Tablas y relaciones
Introducción a fórmulas DAX
Columnas calculadas y medidas
Tablas calculadas
Fórmulas DAX de inteligencia de tiempo (YTD, PreviousQuarter...)
6. Visualizar
Entorno de trabajo para creación de gráficos con Power BI
Trabajar con distintos tipos de gráficos
Formatos para gráficos e informes
Importación de visualizaciones extra desde el Office Store
7. Conectividad y Colaboración
8. Power BI Mobile (Alertas, notificaciones, favoritos)
Stratebi Dashboard Demos
Papers Técnicos y Buenas Prácticas incluidas con el Curso
Tips Vol.1
Tips Vol.2
Tips Vol.3
Tips Vol.4
Integracion SAP - PowerBI
PowerBI Trucos (Vol I)
PowerBI Trucos (Vol II)
PowerBI + Synapse Analytics (paso a paso)
30 Consejos y Buenas Prácticas para hacer un proyecto de Power BI con éxito
Cómo crear diseños de Dashboards espectaculares con PowerBI
Videotutorial: Trabajando con Python en Power BI
Aplicación PowerBi Turismo
Aplicación PowerBI Financiera I
Bravo para PowerBI
Aplicación PowerBI Financiera II
Aplicación PowerBI eCommerce
Aplicación PowerBI Salud
Aplicación PowerBI Smart City
Aplicación PowerBI Energía
Aplicación PowerBI Sports Analytics
Power BI Premium Utilization and Metrics
PowerBI Embedded: Funcionamiento y costes
Guía para integrar Power BI con Microsoft Dynamics y Salesforce
SQL Server Profiler para Power BI
Como usar Report Analyzer en PowerBI, para mejorar el rendimiento
Power BI embebido en Jupyter Notebook
Tabular Editor para Power BI: Videotutorial y manual en español
Personaliza tus gráficas en Power BI con Charticulator y Deneb
Comparativa PowerBI vs Amazon QuickSight
stratebi
Analytics and Big Data

Curso de

PowerBI avanzado | PowerBI

Herramientas de análisis empresarial.

Temario

1. Lenguaje DAX

- Contextos de evaluación en DAX
- Medidas y columnas calculadas
- Creación de tablas
- Operaciones lógicas
- Operaciones matemáticas y estadísticas
- Operaciones de inteligencia de tiempo
- Operaciones con cadenas de texto
- Otras funciones DAX
- Seguridad a nivel de fila (RLS)
- Dax Quiz
- Ejercicios

2. Dataflows

- Introducción a Dataflows
- Conceptos de Dataflows
- ETL con Dataflows
- Buenas prácticas
- Lenguaje M
- Ejercicios

3. External Tools

- Tabular Editor
  - Introducción a Tabular Editor
  - Grupos calculados
  - Perspectivas
  - Avance Scripting
  - Ejercicios
- DAX Studio
- ALM Toolkit

4. Conceptos avanzados

- Data Gateways
- Agregaciones
- Modelos duales
- Machine Learning en Power BI
- Python en Power BI
- R en Power BI
- Microsoft Teams con Power BI
- Power BI Report Builder

5. Buenas prácticas Microsoft Power BI

6. Ejercicios (Opcionales)

TABULAR EDITOR

python



Stratebi Dashboard Demos

Papers Técnicos y Buenas Prácticas incluidas con el Curso

Tips Vol.1 | Tips Vol.2 | Tips Vol.3 | Tips Vol.4

- Como usar emoticonos en PowerBI
- Buenas prácticas con Dataflows en Power BI
- Power Automate para Power BI: Cómo funciona
- ALM Toolkit para Power BI
- PowerBI Trucos (III)
- Os presentamos Goals in Power BI para hacer Scorecards
- Tutorial gratuito en español sobre Power BI Report Builder
- Conoce PowerBI Diagram View (Visual Data Prep): paso a paso
- Big Data para PowerBI
- Fútbol Analytics, lo que hay que saber
- Dashboard de medición de la calidad del aire en Madrid
- ¿Cómo funciona Microsoft Power BI? Videotutorial de introducción
- Cómo integrar Salesforce y PowerBI
- ¿Quieres crear aplicaciones empresariales usando PowerBI, PowerApps y Power Automate de forma conjunta?
- Power BI tip: Uso de parámetros what-if
- Videotutorial: Usando R para Machine Learning con PowerBI
- Las 50 claves para aprender y conocer PowerBI
- PowerBI: Arquitectura End to End
- Usando Python con PowerBI
- Todas las presentaciones del Workshop 'El Business Intelligence del Futuro'
- PowerBI + Open Source = Sports Analytics
- Comparativa de herramientas Business Intelligence
- Use Case Big Data "Dashboards with Hadoop and Power BI"
- Descarga Paper gratuito: Zero to beautiful (Data visualization)
- SAP connection tools for process automation: Microsoft, Pentaho, Talend (User Guide)

stratebi Analytics and Big Data