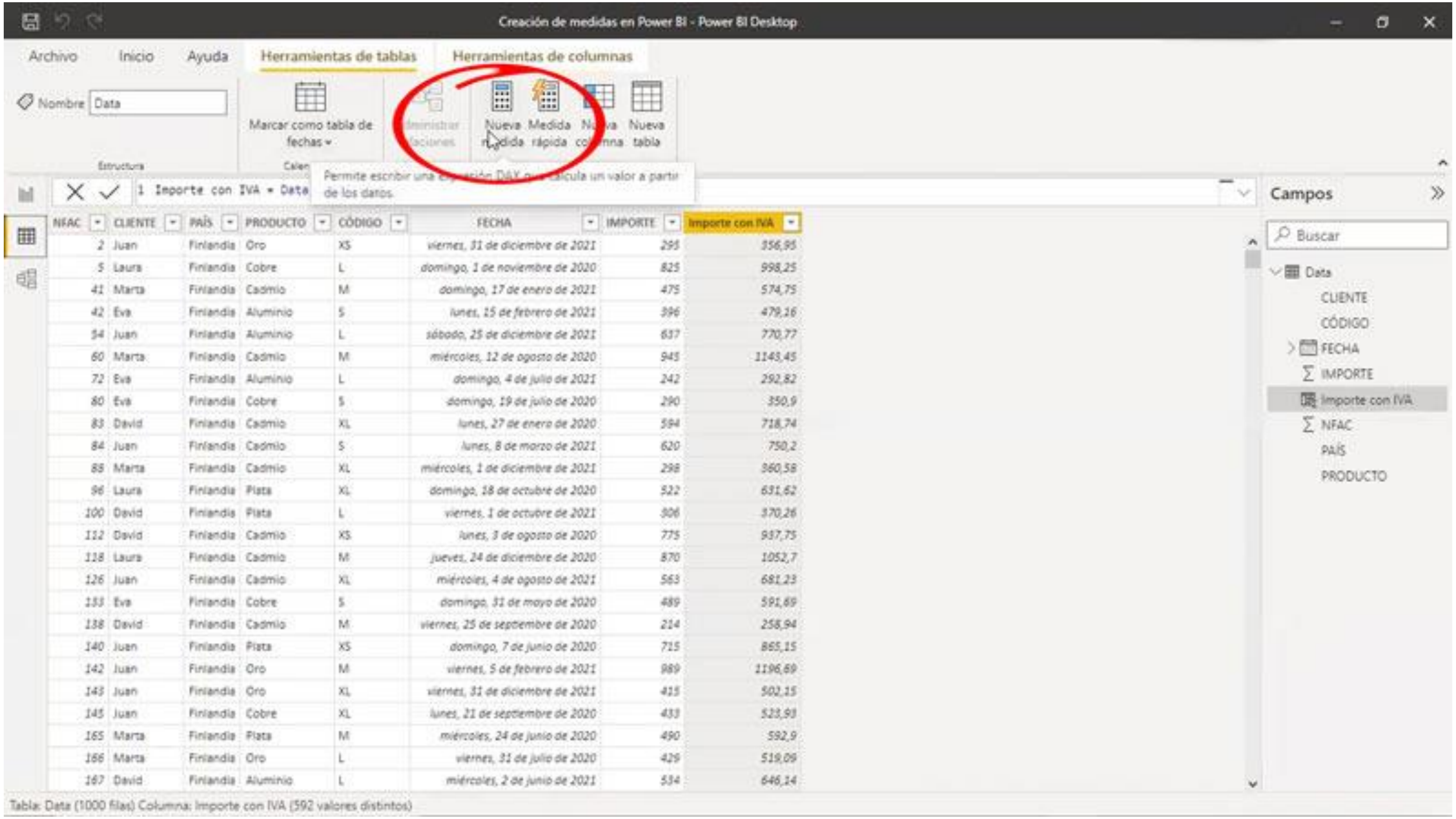
CONTENIDOS

https://github.com/jorloicono/AF-DAX-AVANZADO

PRESENTACIONES

INTRODUCCIÓN A DAX

- Conceptos fundamentales de modelado de datos
- Utilizar columnas calculadas y métricas
- Explorar funciones DAX comunes: agregar lógicas, matemáticas, texto, fecha información, conversión y relacionales
- Usar funciones de tabla y comprender FILTER, ALL, Y ALLEXCEPT
- Realizar análisis temporal de tendencias
- Calcular proyecciones de días
- Aplicar cálculos de medias móviles



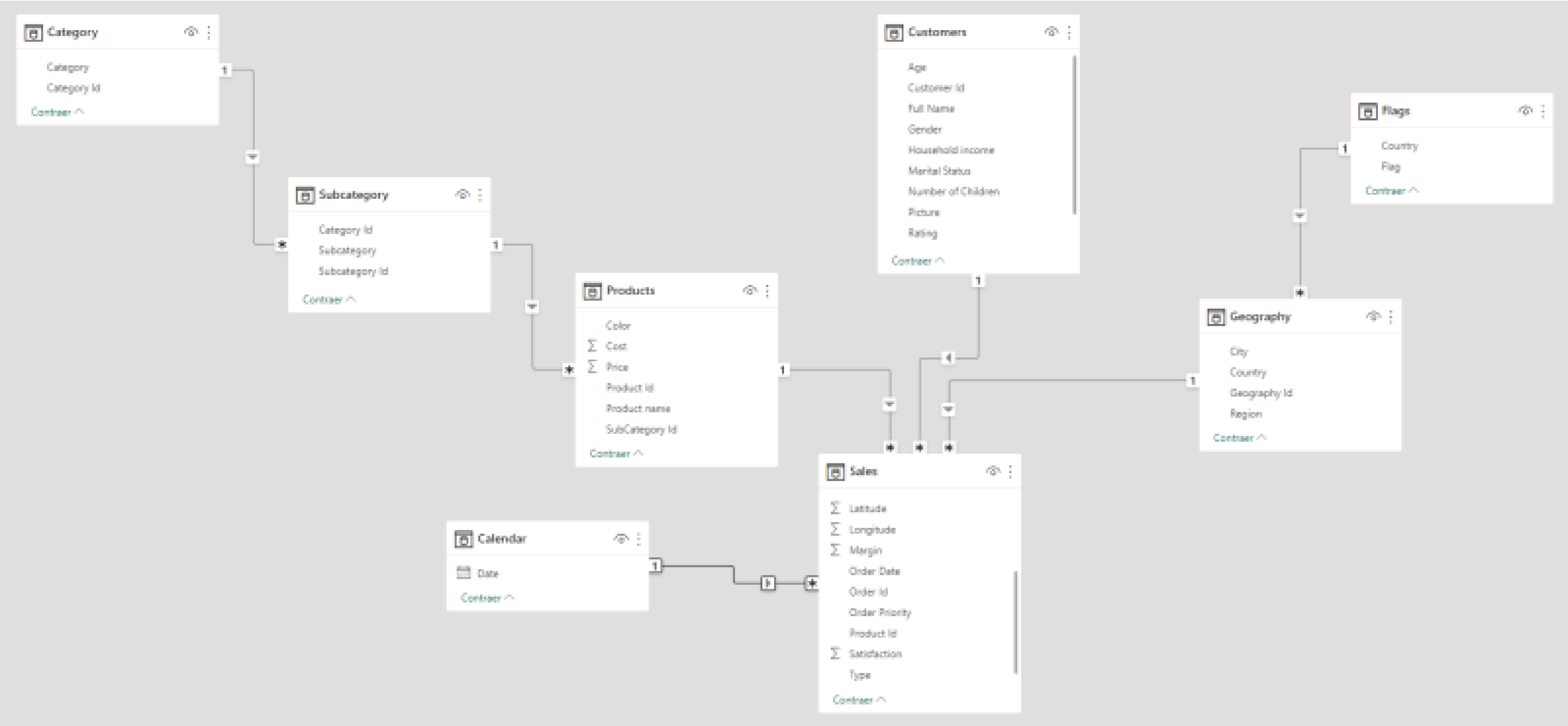
Nombre: Data

Permite escribir una expresión DAX que calcula un valor a partir de los datos.

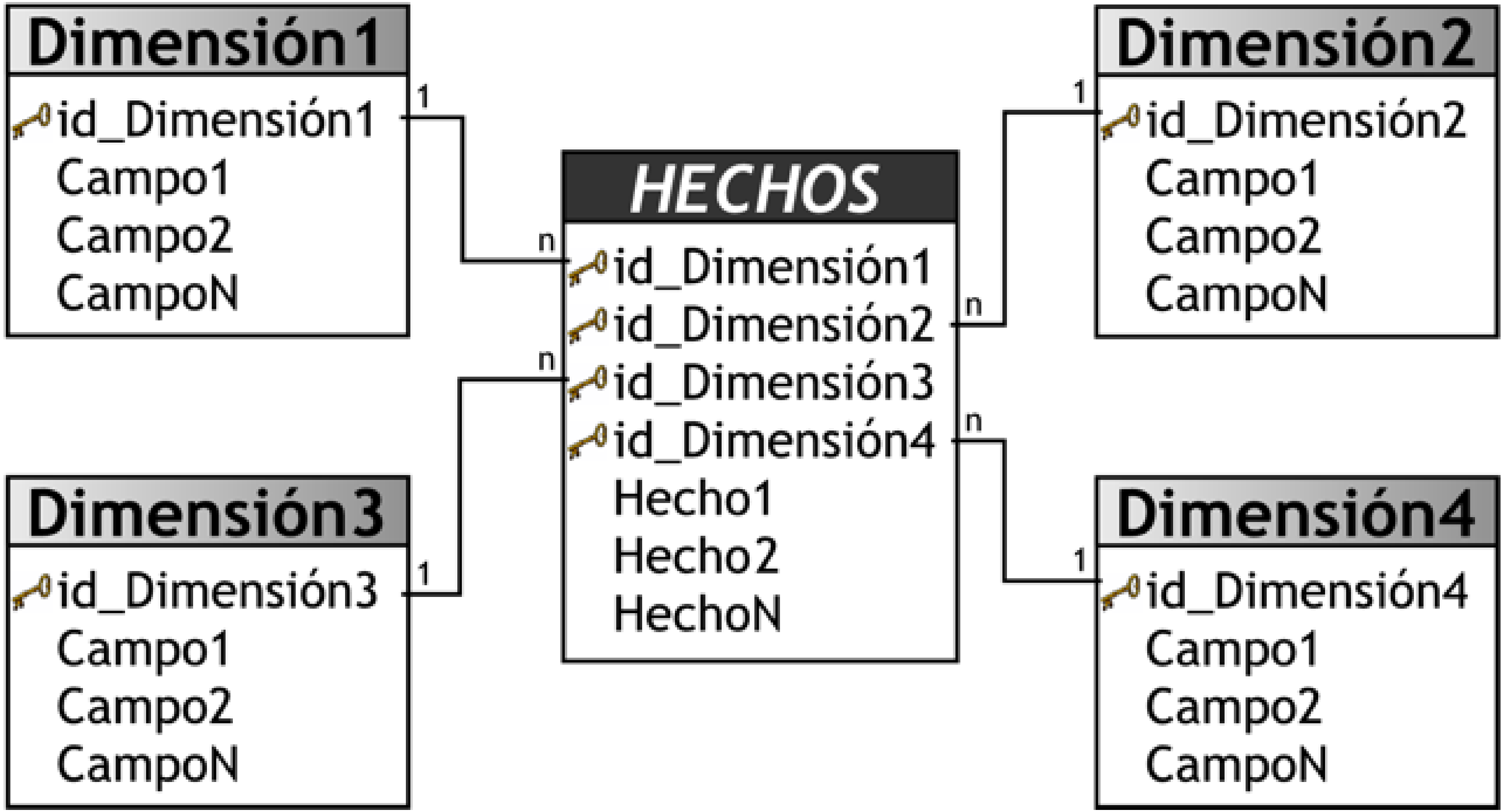
```
1 Importe con IVA = Data
```

NFAC	CLIENTE	PAÍS	PRODUCTO	CÓDIGO	FECHA	IMPORTE	Importe con IVA
2	Juan	Finlandia	Oro	XS	viernes, 31 de diciembre de 2021	295	356,95
5	Laura	Finlandia	Cobre	L	domingo, 1 de noviembre de 2020	825	998,25
41	Marta	Finlandia	Cadmio	M	domingo, 17 de enero de 2021	475	574,75
42	Eva	Finlandia	Aluminio	S	lunes, 15 de febrero de 2021	396	479,16
54	Juan	Finlandia	Aluminio	L	sábado, 25 de diciembre de 2021	637	770,77
60	Marta	Finlandia	Cadmio	M	miércoles, 12 de agosto de 2020	945	1143,45
72	Eva	Finlandia	Aluminio	L	domingo, 4 de julio de 2021	242	292,82
80	Eva	Finlandia	Cobre	S	domingo, 19 de julio de 2020	290	350,9
83	David	Finlandia	Cadmio	XL	lunes, 27 de enero de 2020	594	718,74
84	Juan	Finlandia	Cadmio	S	lunes, 8 de marzo de 2021	620	750,2
88	Marta	Finlandia	Cadmio	XL	miércoles, 1 de diciembre de 2021	298	360,58
96	Laura	Finlandia	Plata	XL	domingo, 18 de octubre de 2020	522	631,62
100	David	Finlandia	Plata	L	viernes, 1 de octubre de 2021	306	370,26
112	David	Finlandia	Cadmio	XS	lunes, 3 de agosto de 2020	775	937,75
118	Laura	Finlandia	Cadmio	M	jueves, 24 de diciembre de 2020	870	1052,7
126	Juan	Finlandia	Cadmio	XL	miércoles, 4 de agosto de 2021	563	681,23
133	Eva	Finlandia	Cobre	S	domingo, 31 de mayo de 2020	489	591,69
138	David	Finlandia	Cadmio	M	viernes, 25 de septiembre de 2020	214	258,94
140	Juan	Finlandia	Plata	XS	domingo, 7 de junio de 2020	715	865,15
142	Juan	Finlandia	Oro	M	viernes, 5 de febrero de 2021	989	1196,69
143	Juan	Finlandia	Oro	XL	viernes, 31 de diciembre de 2021	415	502,15
145	Juan	Finlandia	Cobre	XL	lunes, 21 de septiembre de 2020	433	523,93
165	Marta	Finlandia	Plata	M	miércoles, 24 de junio de 2020	490	592,9
166	Marta	Finlandia	Oro	L	viernes, 31 de julio de 2020	429	519,09
167	David	Finlandia	Aluminio	L	miércoles, 2 de junio de 2021	534	646,14

Tabla: Data (1000 filas) Columna: Importe con IVA (592 valores distintos)



Category
Category
Category Id
Contraer ^
Subcategory
Category Id
Subcategory
Subcategory Id
Contraer ^
Products
Color
Cost
Price
Product Id
Product name
SubCategory Id
Contraer ^
Customers
Age
Customer Id
Full Name
Gender
Household income
Marital Status
Number of Children
Picture
Rating
Contraer ^
Flags
Country
Flag
Contraer ^
Geography
City
Country
Geography Id
Region
Contraer ^
Calendar
Date
Contraer ^
Sales
Latitude
Longitude
Margin
Order Date
Order Id
Order Priority
Product Id
Satisfaction
Type
Contraer ^
1
*
1
*
1
*
1
*
1
*
1
*
1
*



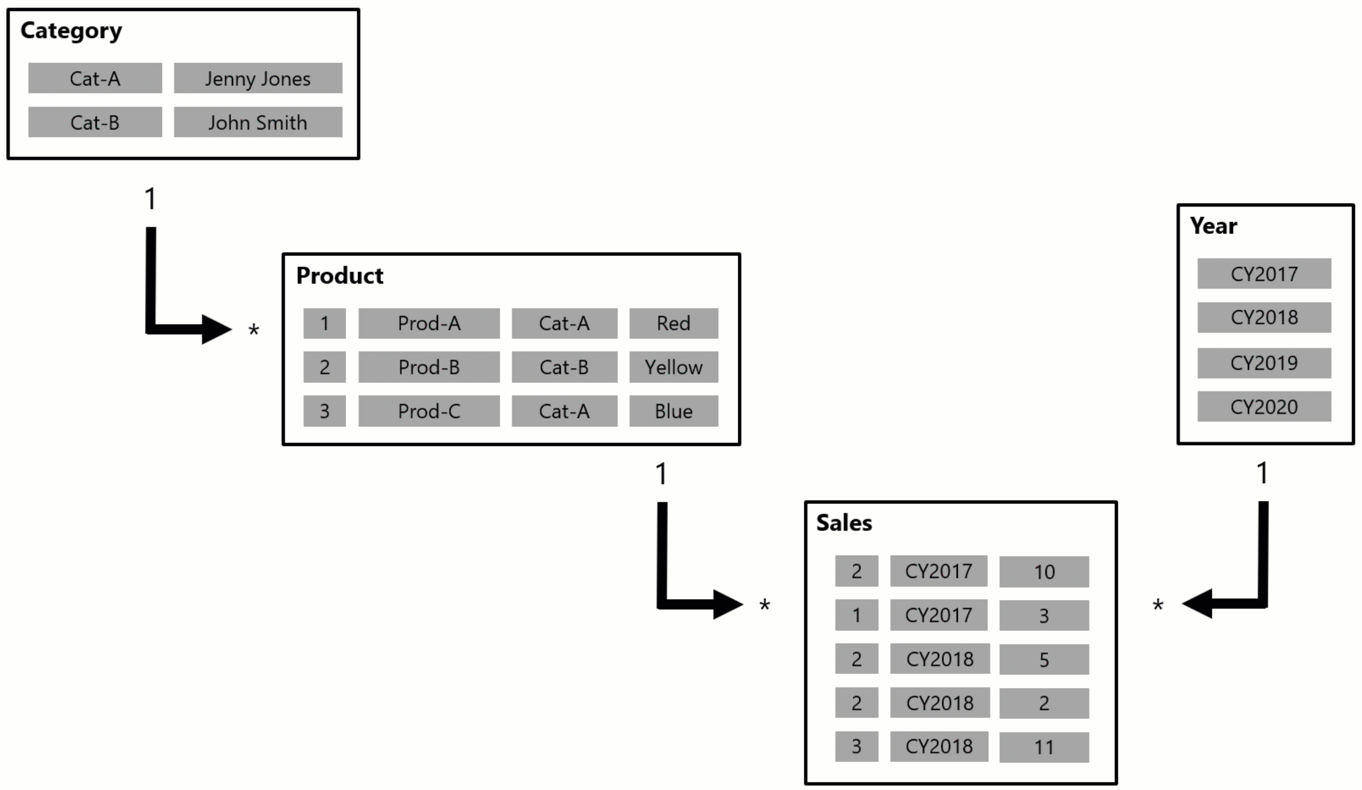
Dimensión1
id_Dimensión1
Campo1
Campo2
CampoN

Dimensión2
id_Dimensión2
Campo1
Campo2
CampoN

HECHOS
id_Dimensión1
id_Dimensión2
id_Dimensión3
id_Dimensión4
Hecho1
Hecho2
HechoN

Dimensión3
id_Dimensión3
Campo1
Campo2
CampoN

Dimensión4
id_Dimensión4
Campo1
Campo2
CampoN



Category

Cat-A	Jenny Jones
Cat-B	John Smith

Product

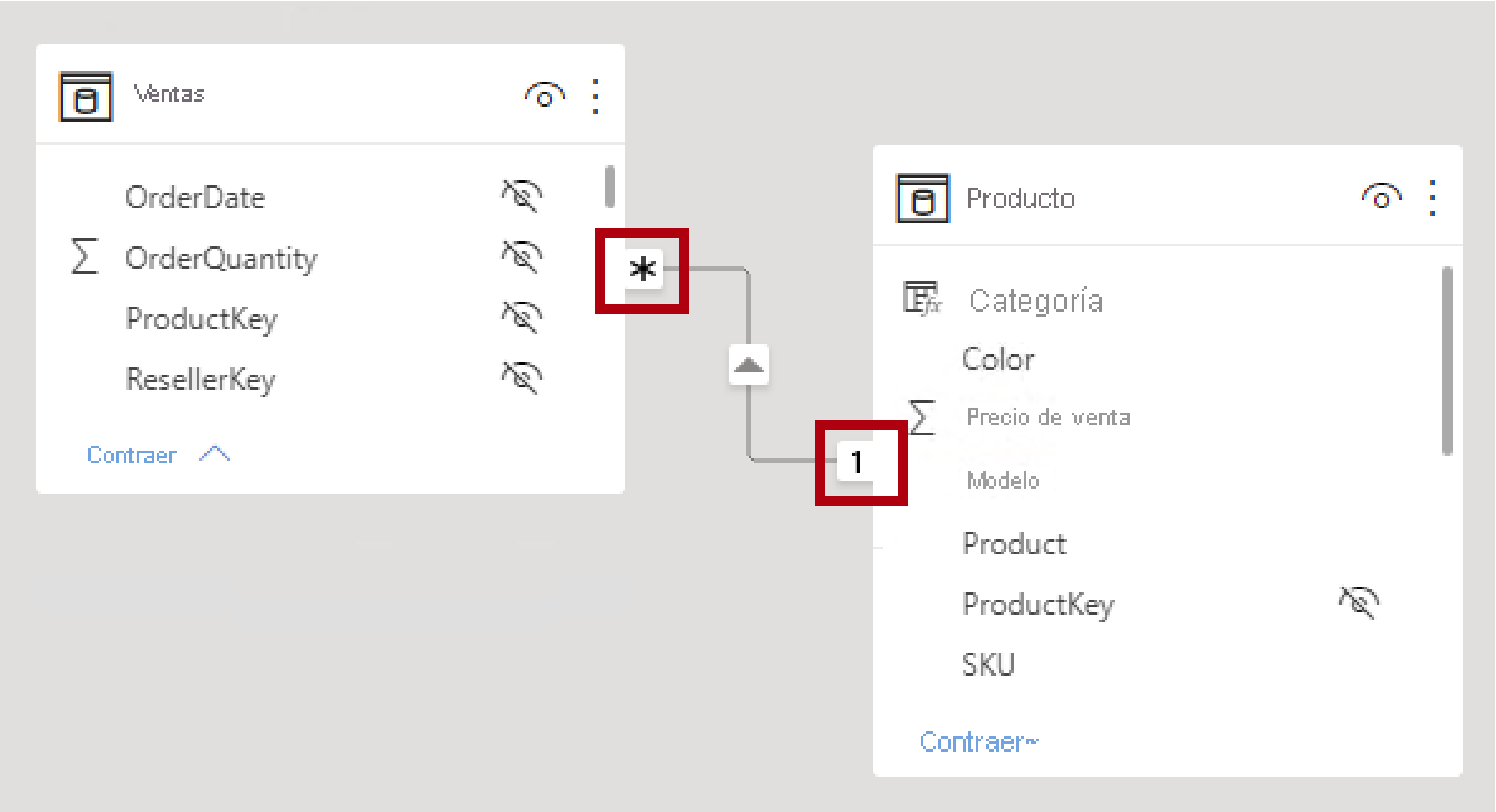
1	Prod-A	Cat-A	Red
2	Prod-B	Cat-B	Yellow
3	Prod-C	Cat-A	Blue

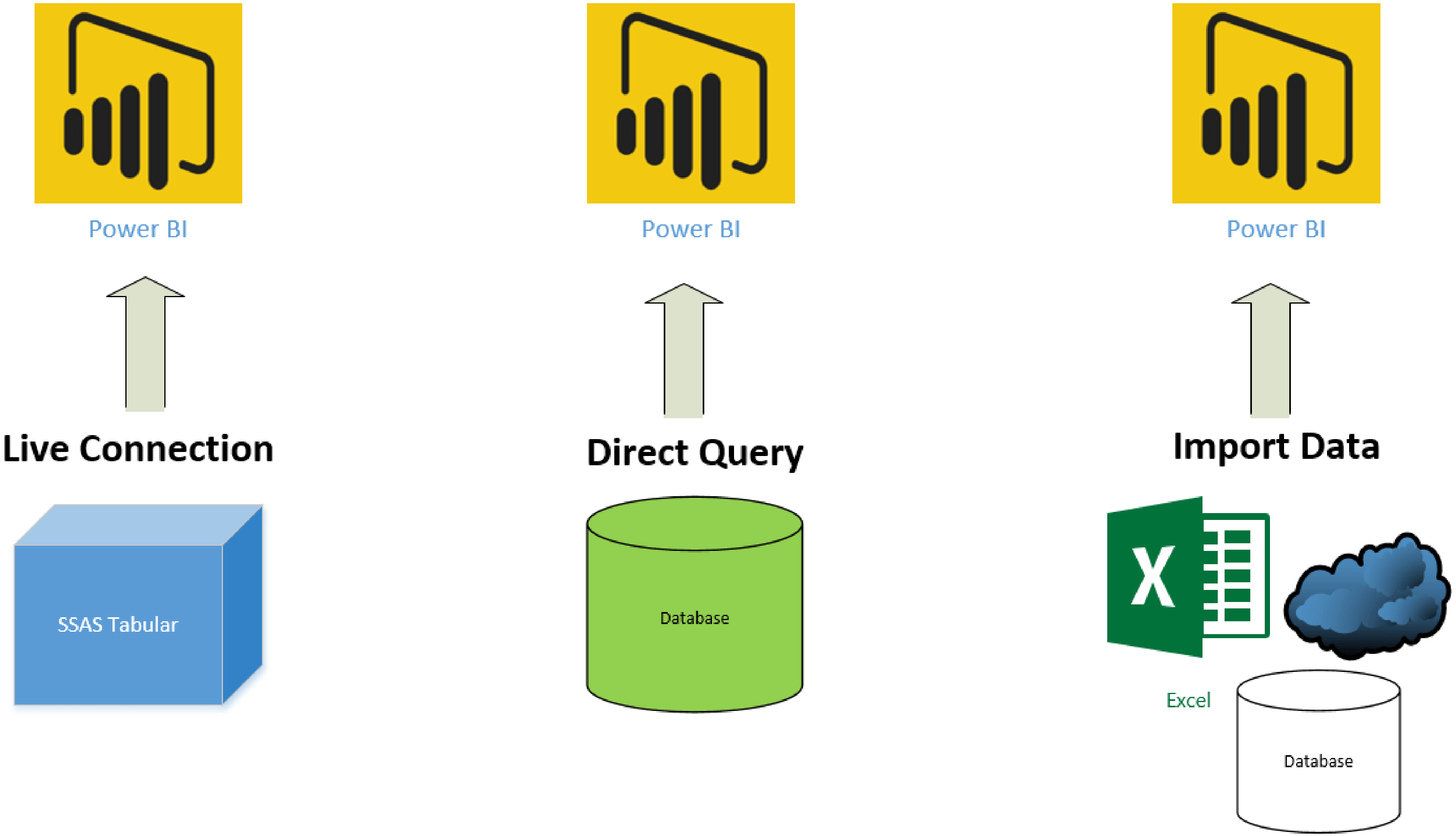
Year

CY2017
CY2018
CY2019
CY2020

Sales

2	CY2017	10
1	CY2017	3
2	CY2018	5
2	CY2018	2
3	CY2018	11

Ventas
OrderDate
OrderQuantity
ProductKey
ResellerKey
Contraer
*
1
Producto
Categoría
Color
Precio de venta
Modelo
Product
ProductKey
SKU
Contraer



Power BI

Live Connection

SSAS Tabular

Power BI

Direct Query

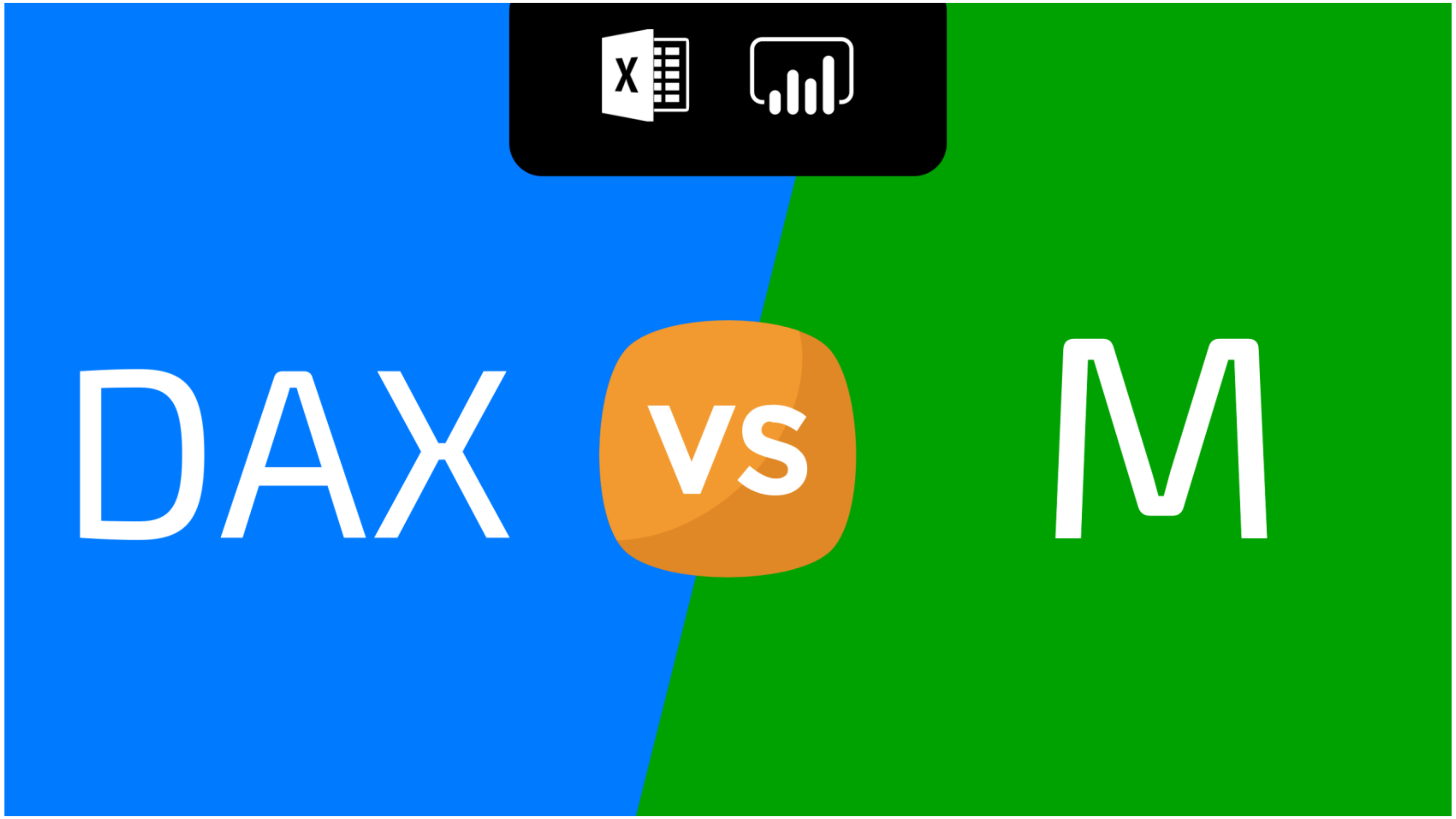
Database

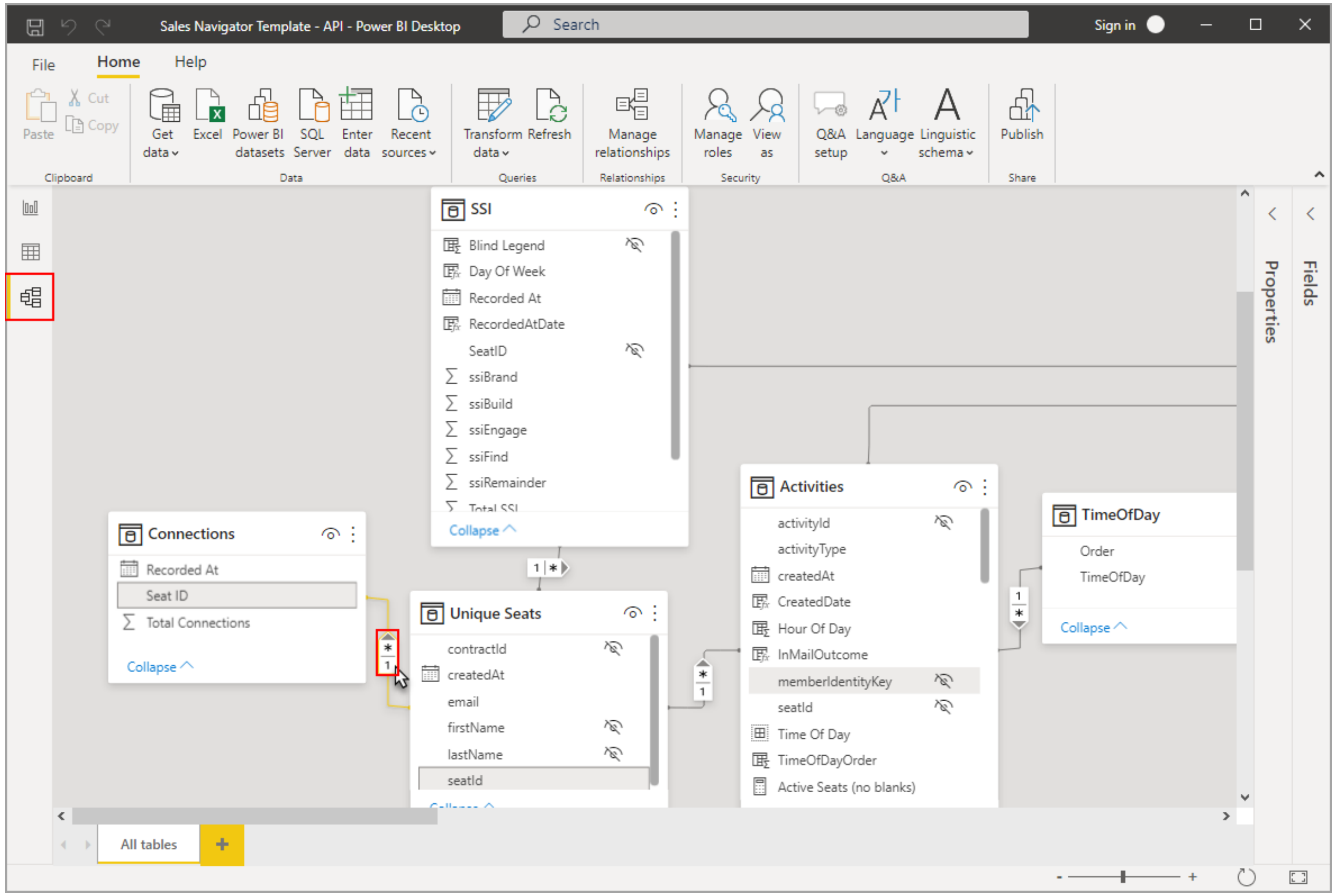
Power BI

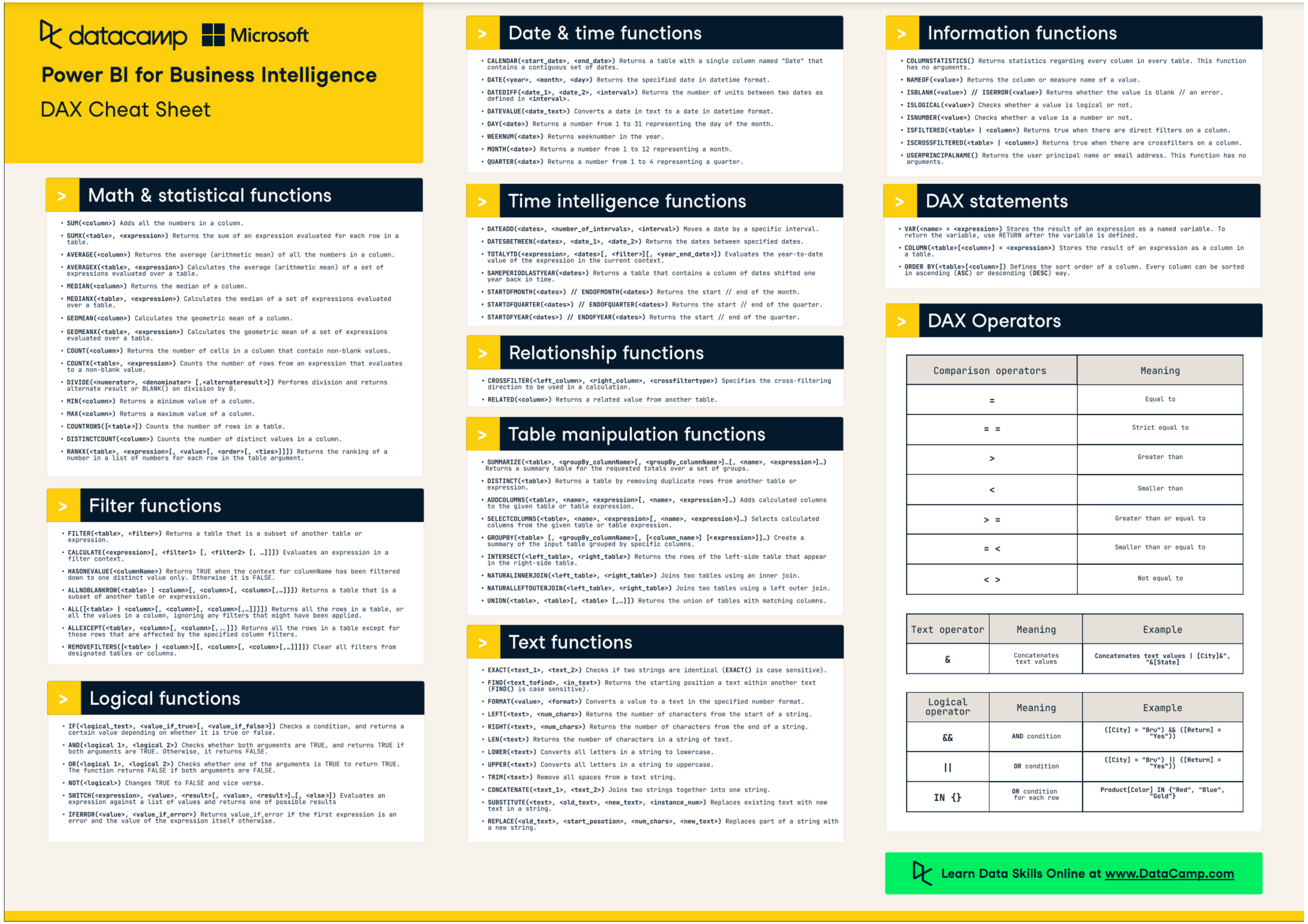
Import Data

Excel

Database


DAX
vs
M


Sales Navigator Template - API - Power BI Desktop
Search
Sign in
File
Home
Help
Paste
Cut
Copy
Clipboard
Get data
Excel
Power BI datasets
SQL Server
Enter data
Recent sources
Data
Transform data
Refresh
Queries
Manage relationships
Relationships
Manage roles
View as
Security
Q&A setup
Language
Linguistic schema
Q&A
Publish
Share
SSI
Blind Legend
Day Of Week
Recorded At
RecordedAtDate
SeatID
ssiBrand
ssiBuild
ssiEngage
ssiFind
ssiRemainder
Total SSI
Collapse
Connections
Recorded At
Seat ID
Total Connections
Collapse
Unique Seats
contractId
createdAt
email
firstName
lastName
seatId
Collapse
Activities
activityId
activityType
createdAt
CreatedDate
Hour Of Day
InMailOutcome
memberIdentityKey
seatId
Time Of Day
TimeOfDayOrder
Active Seats (no blanks)
TimeOfDay
Order
TimeOfDay
Collapse
1
*
1
*
*
1
1
*
Properties
Fields
All tables
+



datacamp Microsoft

Power BI for Business Intelligence

DAX Cheat Sheet

> Math & statistical functions

- **SUM(<column>)** Adds all the numbers in a column.
- **SUMX(<table>, <expression>)** Returns the sum of an expression evaluated for each row in a table.
- **AVERAGE(<column>)** Returns the average (arithmetic mean) of all the numbers in a column.
- **AVERAGEX(<table>, <expression>)** Calculates the average (arithmetic mean) of a set of expressions evaluated over a table.
- **MEDIAN(<column>)** Returns the median of a column.
- **MEDIANX(<table>, <expression>)** Calculates the median of a set of expressions evaluated over a table.
- **GEOMEAN(<column>)** Calculates the geometric mean of a column.
- **GEOMEANX(<table>, <expression>)** Calculates the geometric mean of a set of expressions evaluated over a table.
- **COUNT(<column>)** Returns the number of cells in a column that contain non-blank values.
- **COUNTX(<table>, <expression>)** Counts the number of rows from an expression that evaluates to a non-blank value.
- **DIVIDE(<numerator>, <denominator> [,<alternateresult>])** Performs division and returns alternate result or BLANK() on division by 0.
- **MIN(<column>)** Returns a minimum value of a column.
- **MAX(<column>)** Returns a maximum value of a column.
- **COUNTROWS([<table>])** Counts the number of rows in a table.
- **DISTINCTCOUNT(<column>)** Counts the number of distinct values in a column.
- **RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])** Returns the ranking of a number in a list of numbers for each row in the table argument.

> Filter functions

- **FILTER(<table>, <filter>)** Returns a table that is a subset of another table or expression.
- **CALCULATE(<expression>[, <filter1> [, <filter2> [, …]]])** Evaluates an expression in a filter context.
- **HASONEVALUE(<columnName>)** Returns TRUE when the context for columnName has been filtered down to one distinct value only. Otherwise it is FALSE.
- **ALLNOBLANKROW(<table> | <column>[, <column>[, <column>[,…]]])** Returns a table that is a subset of another table or expression.
- **ALL([<table> | <column>[, <column>[, <column>[,…]]]])** Returns all the rows in a table, or all the values in a column, ignoring any filters that might have been applied.
- **ALLEXCEPT(<table>, <column>[, <column>[,..]])** Returns all the rows in a table except for those rows that are affected by the specified column filters.
- **REMOVEFILTERS([<table> | <column>][, <column>[, <column>[,…]]]])** Clear all filters from designated tables or columns.

> Logical functions

- **IF(<logical_test>, <value_if_true>[, <value_if_false>])** Checks a condition, and returns a certain value depending on whether it is true or false.
- **AND(<logical 1>, <logical 2>)** Checks whether both arguments are TRUE, and returns TRUE if both arguments are TRUE. Otherwise, it returns FALSE.
- **OR(<logical 1>, <logical 2>)** Checks whether one of the arguments is TRUE to return TRUE. The function returns FALSE if both arguments are FALSE.
- **NOT(<logical>)** Changes TRUE to FALSE and vice versa.
- **SWITCH(<expression>, <value>, <result>[, <value>, <result>]…[, <else>])** Evaluates an expression against a list of values and returns one of possible results
- **IFERROR(<value>, <value_if_error>)** Returns value_if_error if the first expression is an error and the value of the expression itself otherwise.

> Date & time functions

- **CALENDAR(<start_date>, <end_date>)** Returns a table with a single column named "Date" that contains a contiguous set of dates.
- **DATE(<year>, <month>, <day>)** Returns the specified date in datetime format.
- **DATEDIFF(<date_1>, <date_2>, <interval>)** Returns the number of units between two dates as defined in **<interval>**.
- **DATEVALUE(<date_text>)** Converts a date in text to a date in datetime format.
- **DAY(<date>)** Returns a number from 1 to 31 representing the day of the month.
- **WEEKNUM(<date>)** Returns weeknumber in the year.
- **MONTH(<date>)** Returns a number from 1 to 12 representing a month.
- **QUARTER(<date>)** Returns a number from 1 to 4 representing a quarter.

> Time intelligence functions

- **DATEADD(<dates>, <number_of_intervals>, <interval>)** Moves a date by a specific interval.
- **DATESBETWEEN(<dates>, <date_1>, <date_2>)** Returns the dates between specified dates.
- **TOTALYTD(<expression>, <dates>[, <filter>][, <year_end_date>])** Evaluates the year-to-date value of the expression in the current context.
- **SAMEPERIODLASTYEAR(<dates>)** Returns a table that contains a column of dates shifted one year back in time.
- **STARTOFMONTH(<dates>) // ENDOFMONTH(<dates>)** Returns the start // end of the month.
- **STARTOFQUARTER(<dates>) // ENDOFQUARTER(<dates>)** Returns the start // end of the quarter.
- **STARTOFYEAR(<dates>) // ENDOFYEAR(<dates>)** Returns the start // end of the quarter.

> Relationship functions

- **CROSSFILTER(<left_column>, <right_column>, <crossfiltertype>)** Specifies the cross-filtering direction to be used in a calculation.
- **RELATED(<column>)** Returns a related value from another table.

> Table manipulation functions

- **SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]…[, <name>, <expression>]…)** Returns a summary table for the requested totals over a set of groups.
- **DISTINCT(<table>)** Returns a table by removing duplicate rows from another table or expression.
- **ADDCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]…)** Adds calculated columns to the given table or table expression.
- **SELECTCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]…)** Selects calculated columns from the given table or table expression.
- **GROUPBY(<table> [, <groupBy_columnName>[, [<column_name>] [<expression>]]…)** Create a summary of the input table grouped by specific columns.
- **INTERSECT(<left_table>, <right_table>)** Returns the rows of the left-side table that appear in the right-side table.
- **NATURALINNERJOIN(<left_table>, <right_table>)** Joins two tables using an inner join.
- **NATURALLEFTOUTERJOIN(<left_table>, <right_table>)** Joins two tables using a left outer join.
- **UNION(<table>, <table>[, <table> [,…]])** Returns the union of tables with matching columns.

> Text functions

- **EXACT(<text_1>, <text_2>)** Checks if two strings are identical (**EXACT()** is case sensitive).
- **FIND(<text_tofind>, <in_text>)** Returns the starting position a text within another text (**FIND()** is case sensitive).
- **FORMAT(<value>, <format>)** Converts a value to a text in the specified number format.
- **LEFT(<text>, <num_chars>)** Returns the number of characters from the start of a string.
- **RIGHT(<text>, <num_chars>)** Returns the number of characters from the end of a string.
- **LEN(<text>)** Returns the number of characters in a string of text.
- **LOWER(<text>)** Converts all letters in a string to lowercase.
- **UPPER(<text>)** Converts all letters in a string to uppercase.
- **TRIM(<text>)** Remove all spaces from a text string.
- **CONCATENATE(<text_1>, <text_2>)** Joins two strings together into one string.
- **SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)** Replaces existing text with new text in a string.
- **REPLACE(<old_text>, <start_posotion>, <num_chars>, <new_text>)** Replaces part of a string with a new string.

> Information functions

- **COLUMNSTATISTICS()** Returns statistics regarding every column in every table. This function has no arguments.
- **NAMEOF(<value>)** Returns the column or measure name of a value.
- **ISBLANK(<value>) // ISERROR(<value>)** Returns whether the value is blank // an error.
- **ISLOGICAL(<value>)** Checks whether a value is logical or not.
- **ISNUMBER(<value>)** Checks whether a value is a number or not.
- **ISFILTERED(<table> | <column>)** Returns true when there are direct filters on a column.
- **ISCROSSFILTERED(<table> | <column>)** Returns true when there are crossfilters on a column.
- **USERPRINCIPALNAME()** Returns the user principal name or email address. This function has no arguments.

> DAX statements

- **VAR(<name> = <expression>)** Stores the result of an expression as a named variable. To return the variable, use RETURN after the variable is defined.
- **COLUMN(<table>[<column>] = <expression>)** Stores the result of an expression as a column in a table.
- **ORDER BY(<table>[<column>])** Defines the sort order of a column. Every column can be sorted in ascending (**ASC**) or descending (**DESC**) way.

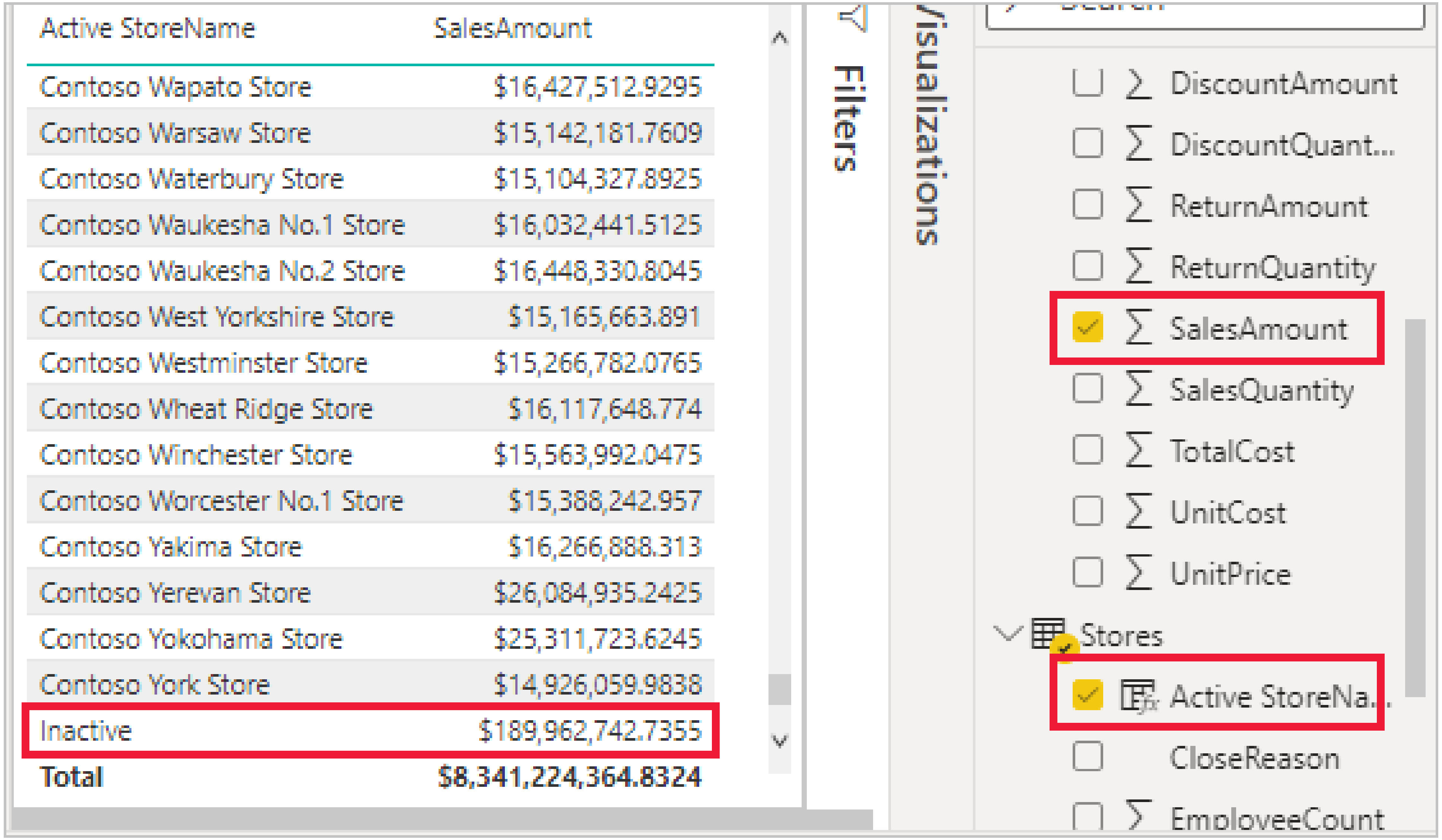
> DAX Operators

Comparison operators	Meaning
=	Equal to
= =	Strict equal to
>	Greater than
<	Smaller than
> =	Greater than or equal to
= <	Smaller than or equal to
< >	Not equal to

Text operator	Meaning	Example
&	Concatenates text values	Concatenates text values | [City]&", "&[State]

Logical operator	Meaning	Example
&&	AND condition	([City] = "Bru") && ([Return] = "Yes"))
||	OR condition	([City] = "Bru") || ([Return] = "Yes"))
IN {}	OR condition for each row	Product[Color] IN {"Red", "Blue", "Gold"}

Learn Data Skills Online at www.DataCamp.com

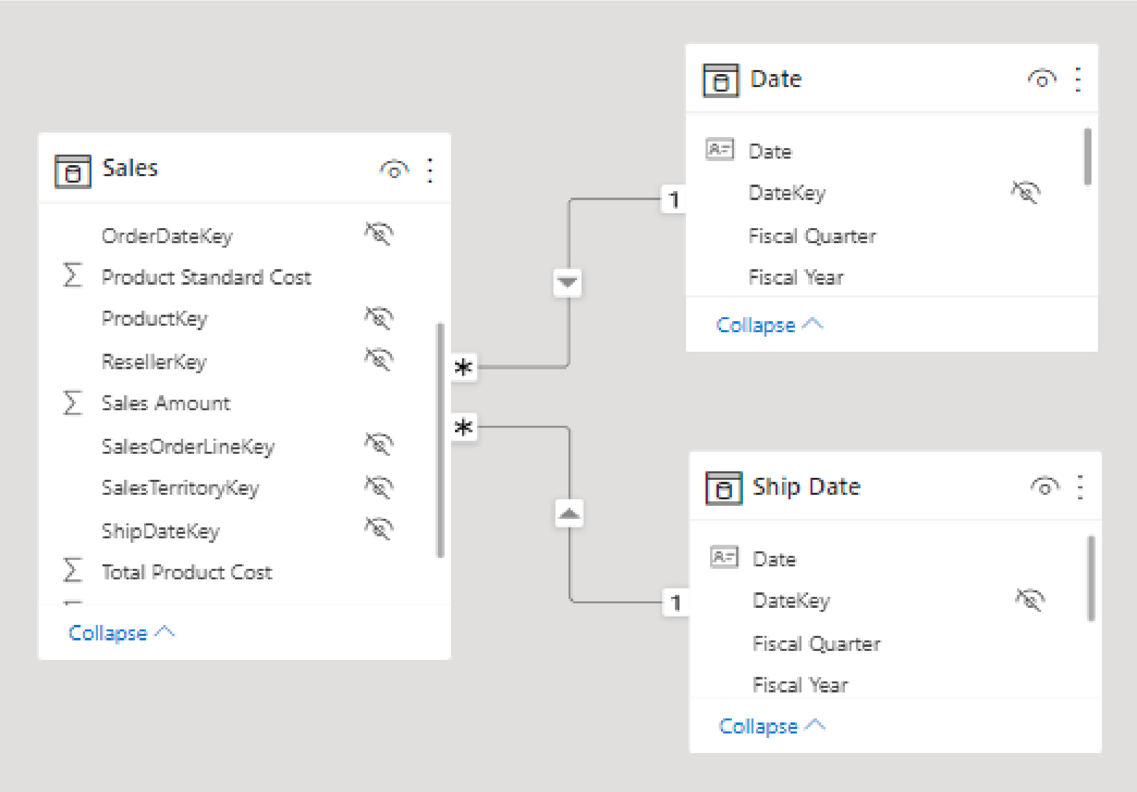


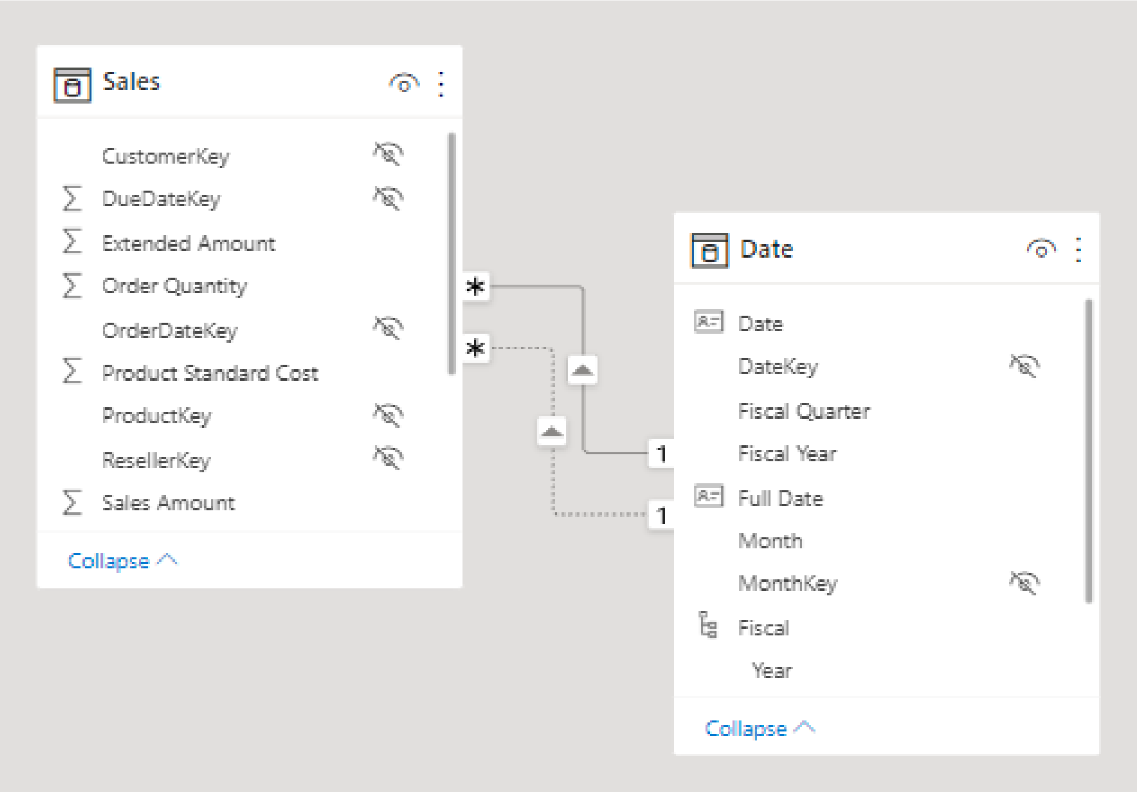
Active StoreName	SalesAmount
Contoso Wapato Store	$16,427,512.9295
Contoso Warsaw Store	$15,142,181.7609
Contoso Waterbury Store	$15,104,327.8925
Contoso Waukesha No.1 Store	$16,032,441.5125
Contoso Waukesha No.2 Store	$16,448,330.8045
Contoso West Yorkshire Store	$15,165,663.891
Contoso Westminster Store	$15,266,782.0765
Contoso Wheat Ridge Store	$16,117,648.774
Contoso Winchester Store	$15,563,992.0475
Contoso Worcester No.1 Store	$15,388,242.957
Contoso Yakima Store	$16,266,888.313
Contoso Yerevan Store	$26,084,935.2425
Contoso Yokohama Store	$25,311,723.6245
Contoso York Store	$14,926,059.9838
Inactive	$189,962,742.7355
Total	**$8,341,224,364.8324**

□ ∑ DiscountAmount
□ ∑ DiscountQuant...
□ ∑ ReturnAmount
□ ∑ ReturnQuantity
☑ ∑ SalesAmount
□ ∑ SalesQuantity
□ ∑ TotalCost
□ ∑ UnitCost
□ ∑ UnitPrice

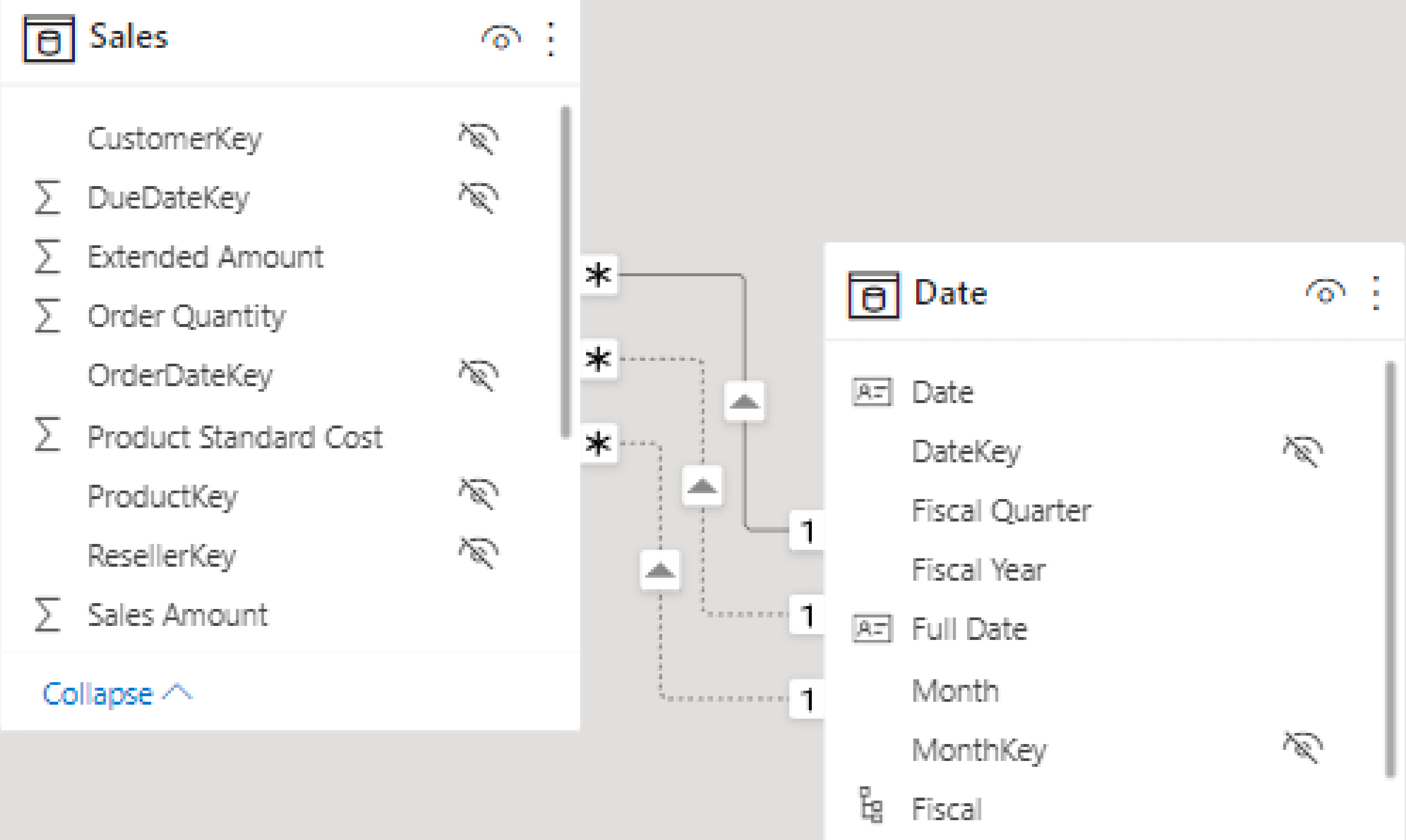
Stores

☑ Active StoreNa...
□ CloseReason
□ ∑ EmployeeCount

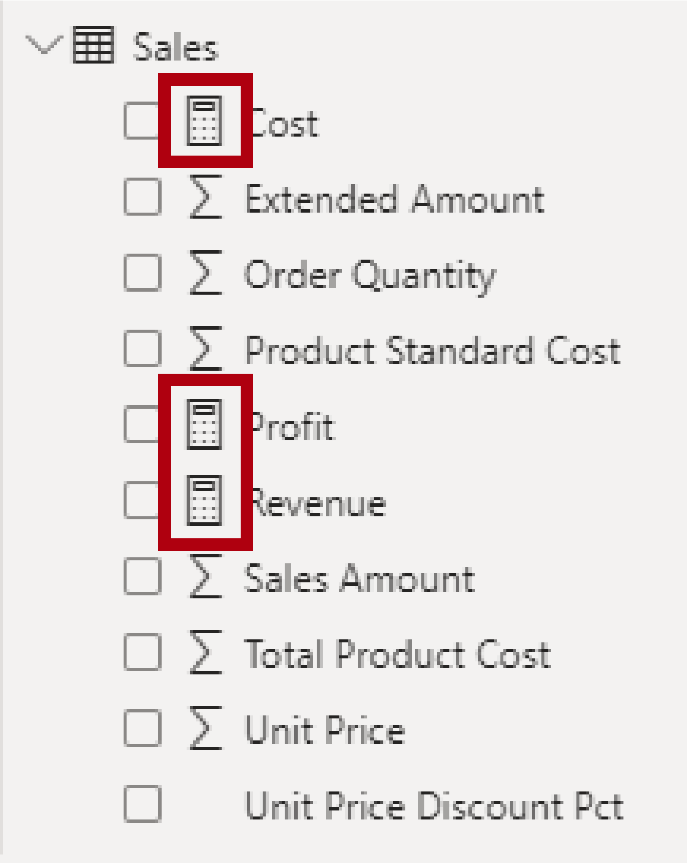




DIMENSIONES REALIZADORAS DE ROLES

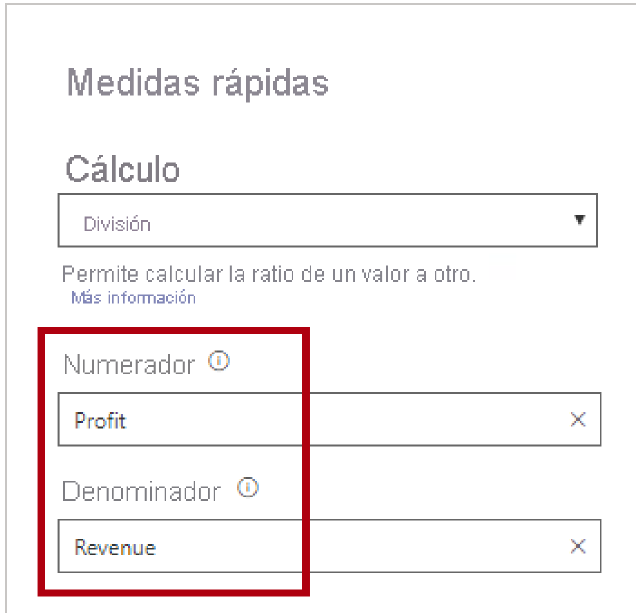

Sales
CustomerKey
DueDateKey
Extended Amount
Order Quantity
OrderDateKey
Product Standard Cost
ProductKey
ResellerKey
Sales Amount
Collapse
*
*
*
1
1
1
Date
Date
DateKey
Fiscal Quarter
Fiscal Year
Full Date
Month
MonthKey
Fiscal

LABORATORIO 1



Sales

- □ Cost
- □ ∑ Extended Amount
- □ ∑ Order Quantity
- □ ∑ Product Standard Cost
- □ Profit
- □ Revenue
- □ ∑ Sales Amount
- □ ∑ Total Product Cost
- □ ∑ Unit Price
- □ Unit Price Discount Pct



Medidas rápidas

Cálculo

División

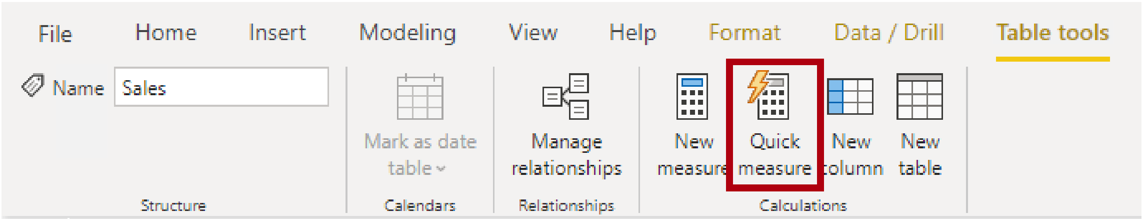
Permite calcular la ratio de un valor a otro.
Más información

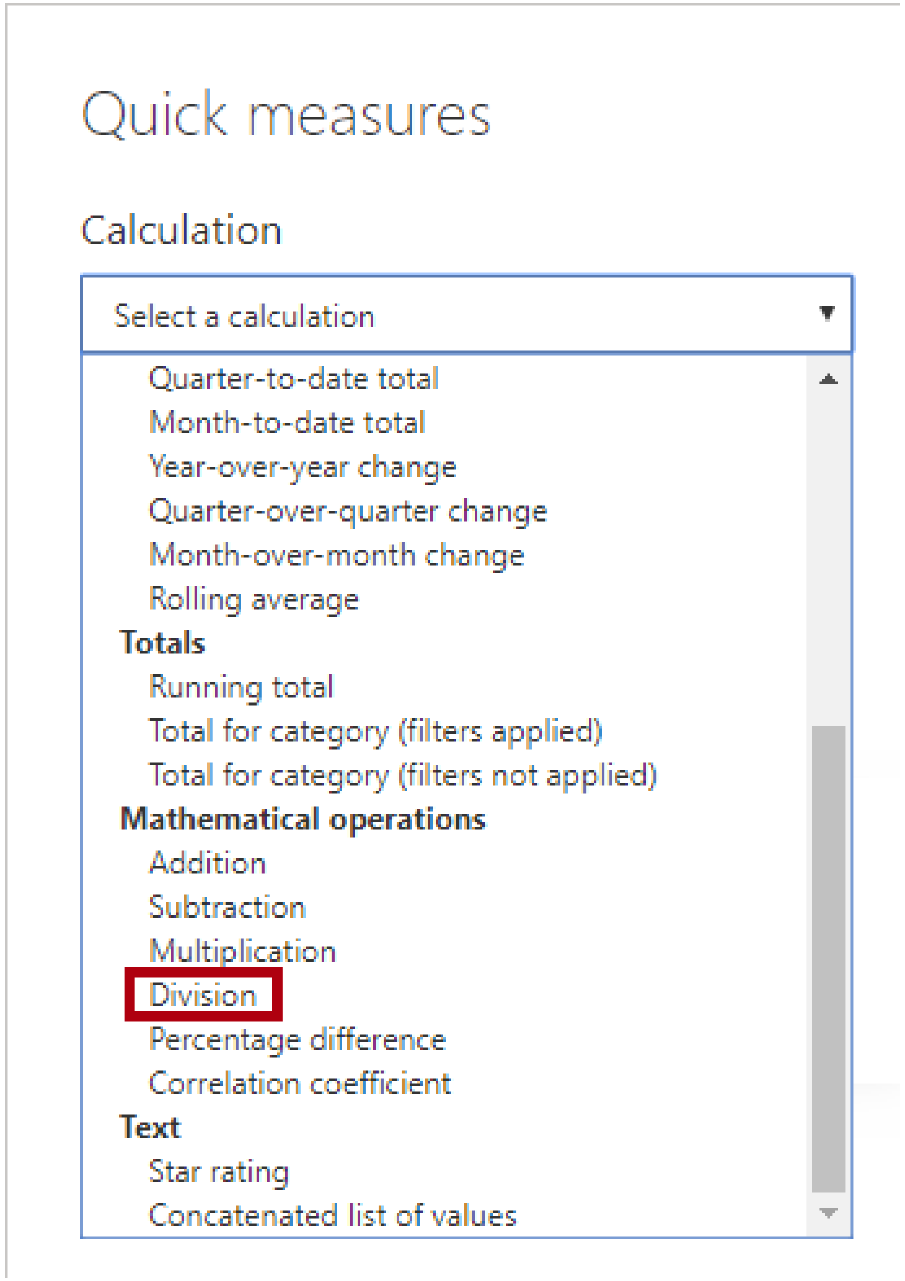
Numerador ⓘ

Profit

Denominador ⓘ

Revenue





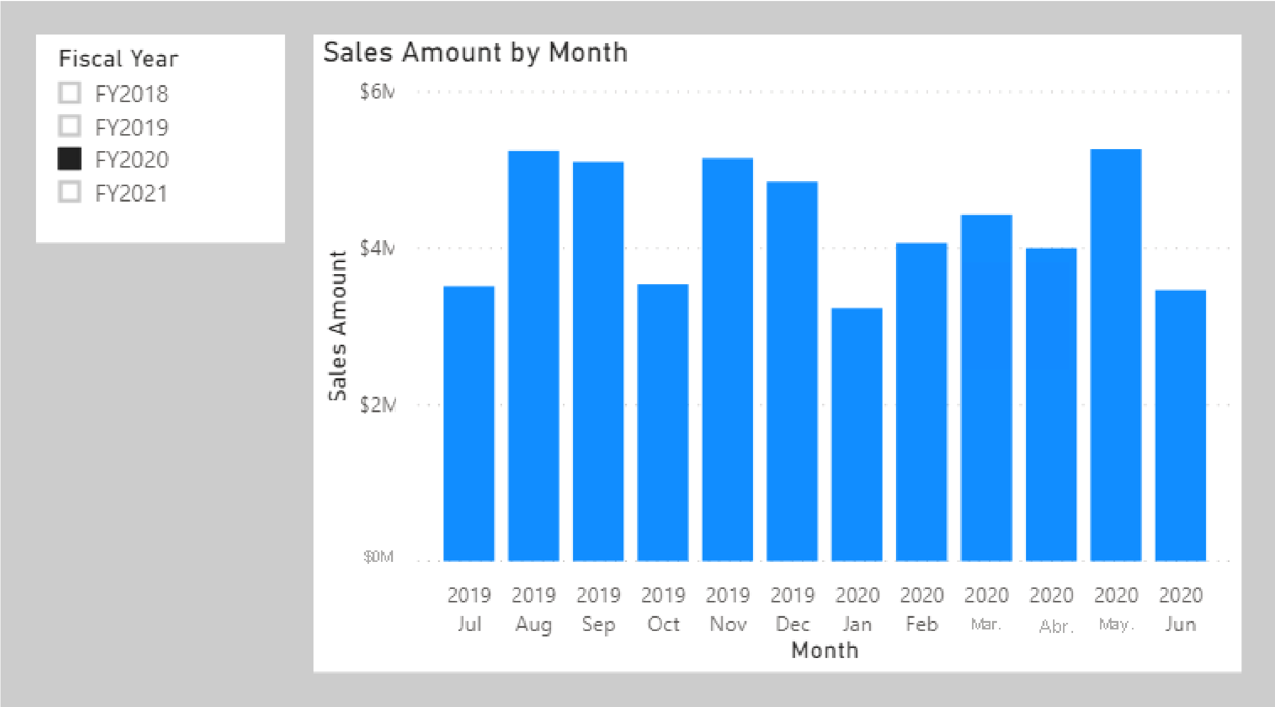
Quick measures

Calculation

Select a calculation

LABORATORIO 2

CONTEXTO



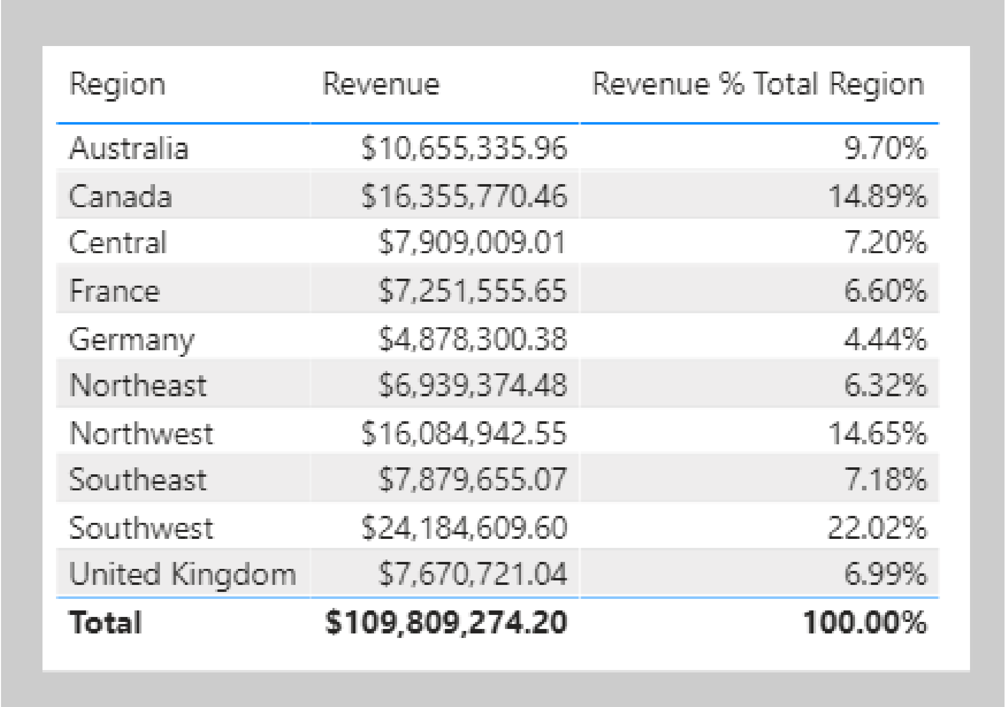
Fiscal Year

□ FY2018

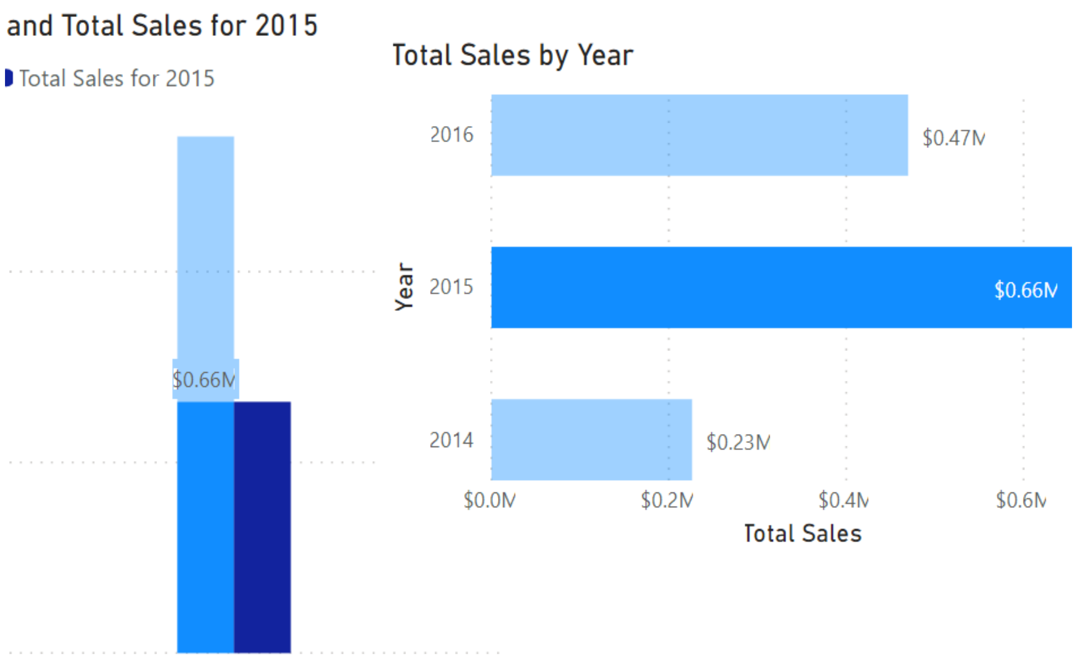
□ FY2019

■ FY2020

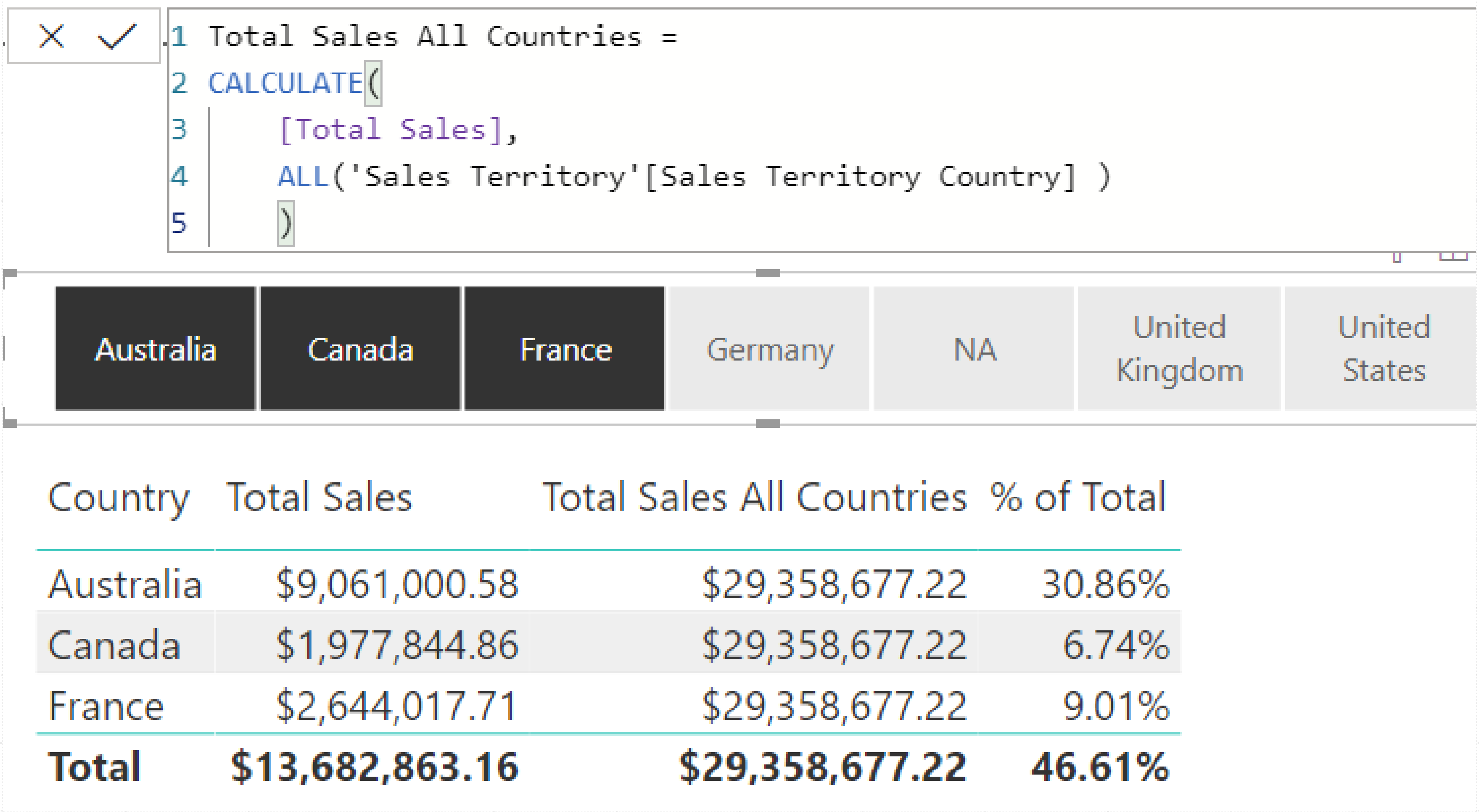
□ FY2021



Region	Revenue	Revenue % Total Region
Australia	$10,655,335.96	9.70%
Canada	$16,355,770.46	14.89%
Central	$7,909,009.01	7.20%
France	$7,251,555.65	6.60%
Germany	$4,878,300.38	4.44%
Northeast	$6,939,374.48	6.32%
Northwest	$16,084,942.55	14.65%
Southeast	$7,879,655.07	7.18%
Southwest	$24,184,609.60	22.02%
United Kingdom	$7,670,721.04	6.99%
Total	**$109,809,274.20**	**100.00%**

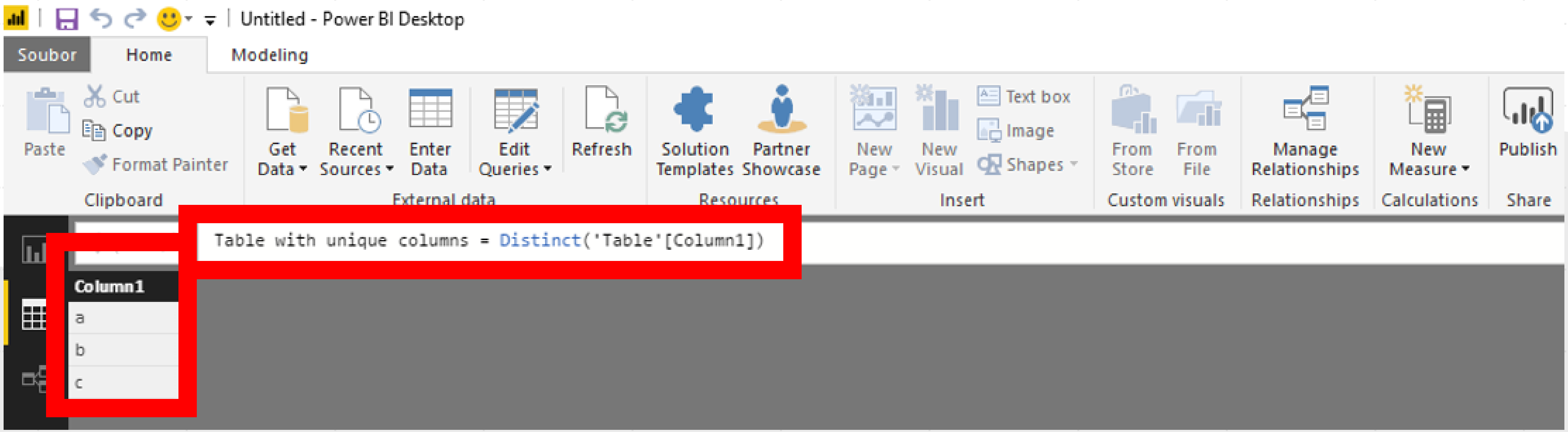

and Total Sales for 2015
Total Sales for 2015
$0.66M
Total Sales by Year
Year
2016
2015
2014
$0.47M
$0.66M
$0.23M
$0.0M
$0.2M
$0.4M
$0.6M
Total Sales

CALCULATE



```
Total Sales All Countries =
CALCULATE(
    [Total Sales],
    ALL('Sales Territory'[Sales Territory Country] )
    )
```

Country	Total Sales	Total Sales All Countries	% of Total
Australia	$9,061,000.58	$29,358,677.22	30.86%
Canada	$1,977,844.86	$29,358,677.22	6.74%
France	$2,644,017.71	$29,358,677.22	9.01%
Total	**$13,682,863.16**	**$29,358,677.22**	**46.61%**

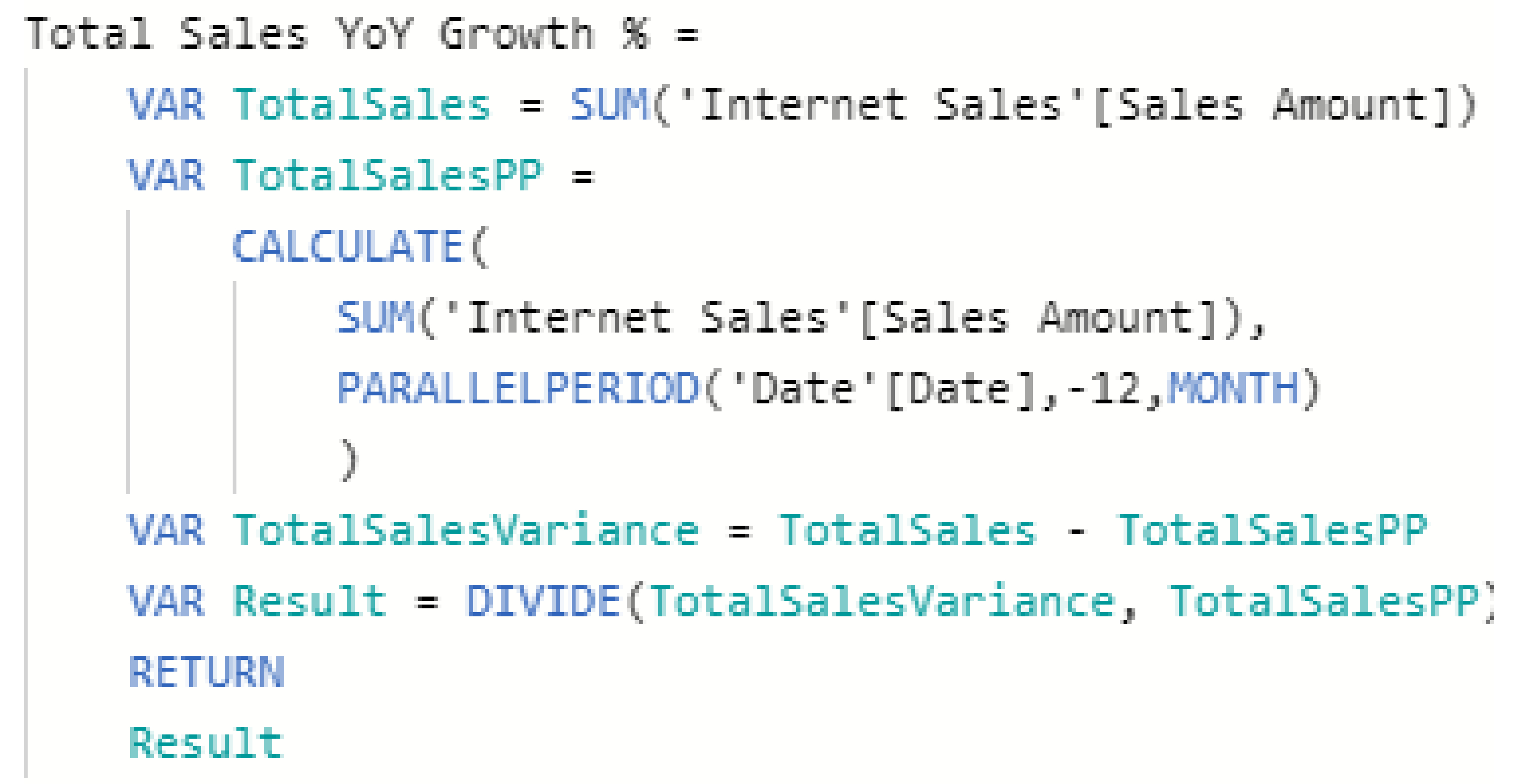


```
Table with unique columns = Distinct('Table'[Column1])
```

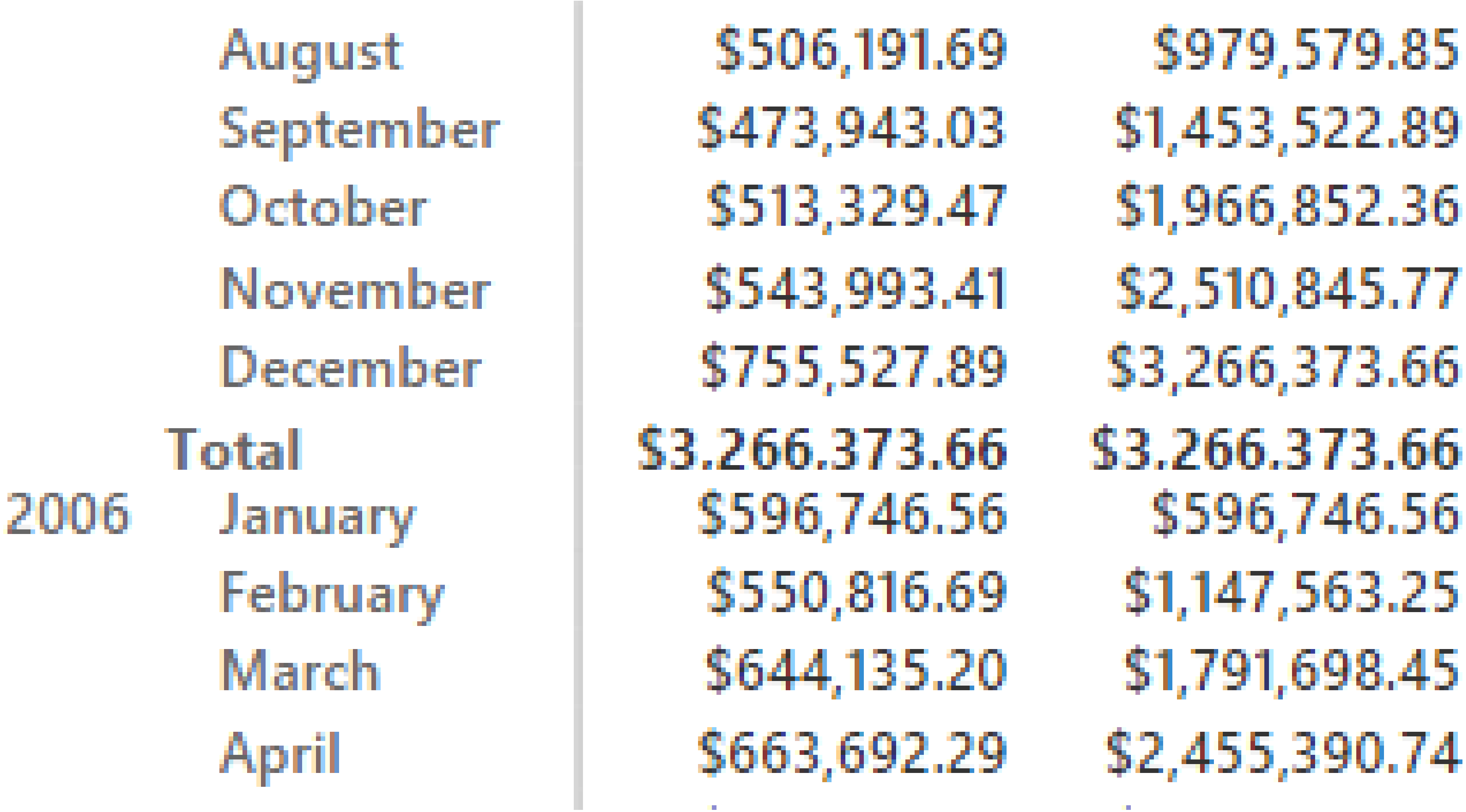
Column1
a
b
c

LABORATORIO 3

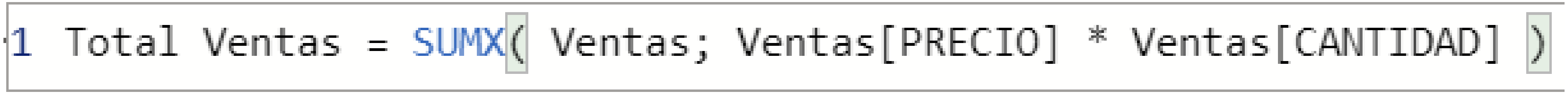
EJERCICIO 1



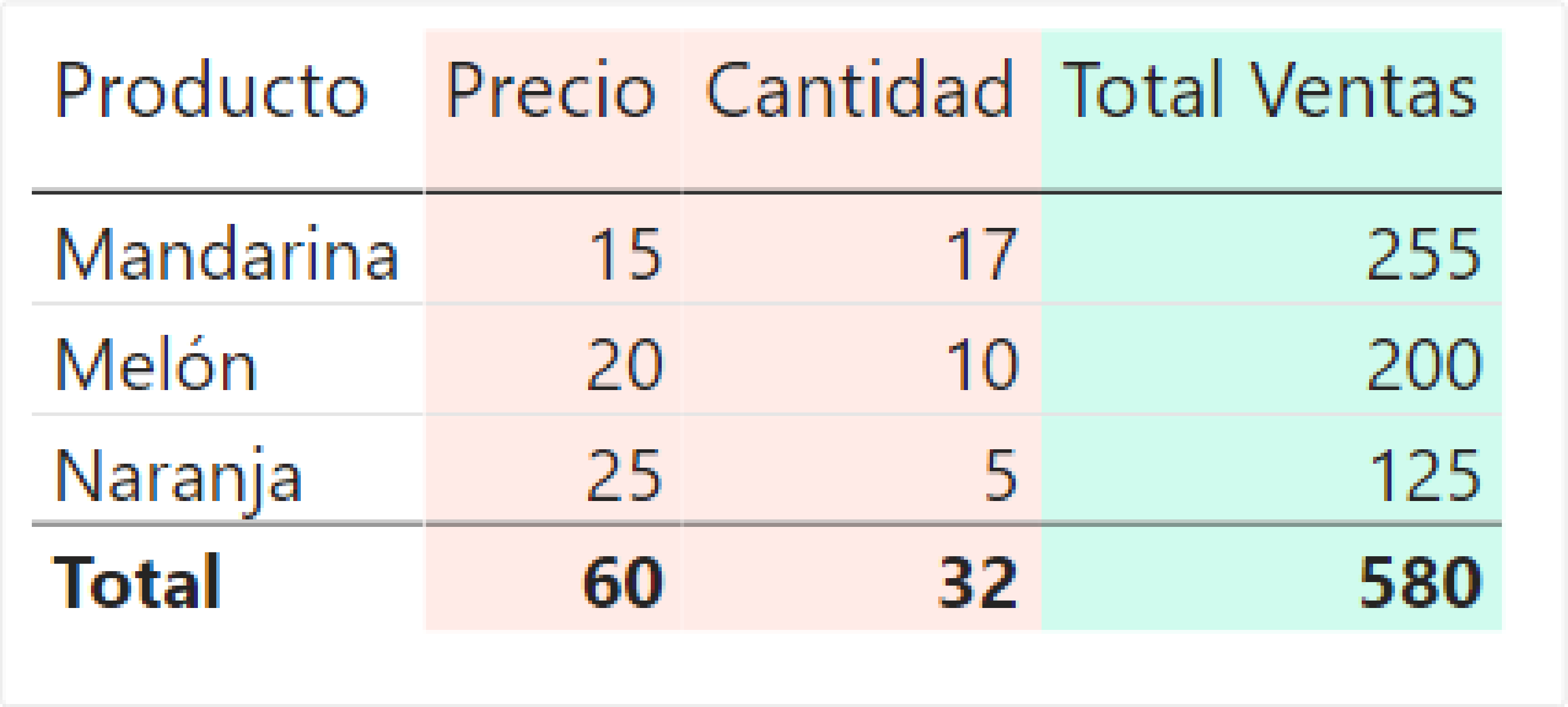
```
Total Sales YoY Growth % =
    VAR TotalSales = SUM('Internet Sales'[Sales Amount])
    VAR TotalSalesPP =
        CALCULATE(
            SUM('Internet Sales'[Sales Amount]),
            PARALLELPERIOD('Date'[Date],-12,MONTH)
            )
    VAR TotalSalesVariance = TotalSales - TotalSalesPP
    VAR Result = DIVIDE(TotalSalesVariance, TotalSalesPP)
    RETURN
    Result
```



	August	$506,191.69	$979,579.85
	September	$473,943.03	$1,453,522.89
	October	$513,329.47	$1,966,852.36
	November	$543,993.41	$2,510,845.77
	December	$755,527.89	$3,266,373.66
	Total	**$3,266,373.66**	**$3,266,373.66**
2006	January	$596,746.56	$596,746.56
	February	$550,816.69	$1,147,563.25
	March	$644,135.20	$1,791,698.45
	April	$663,692.29	$2,455,390.74



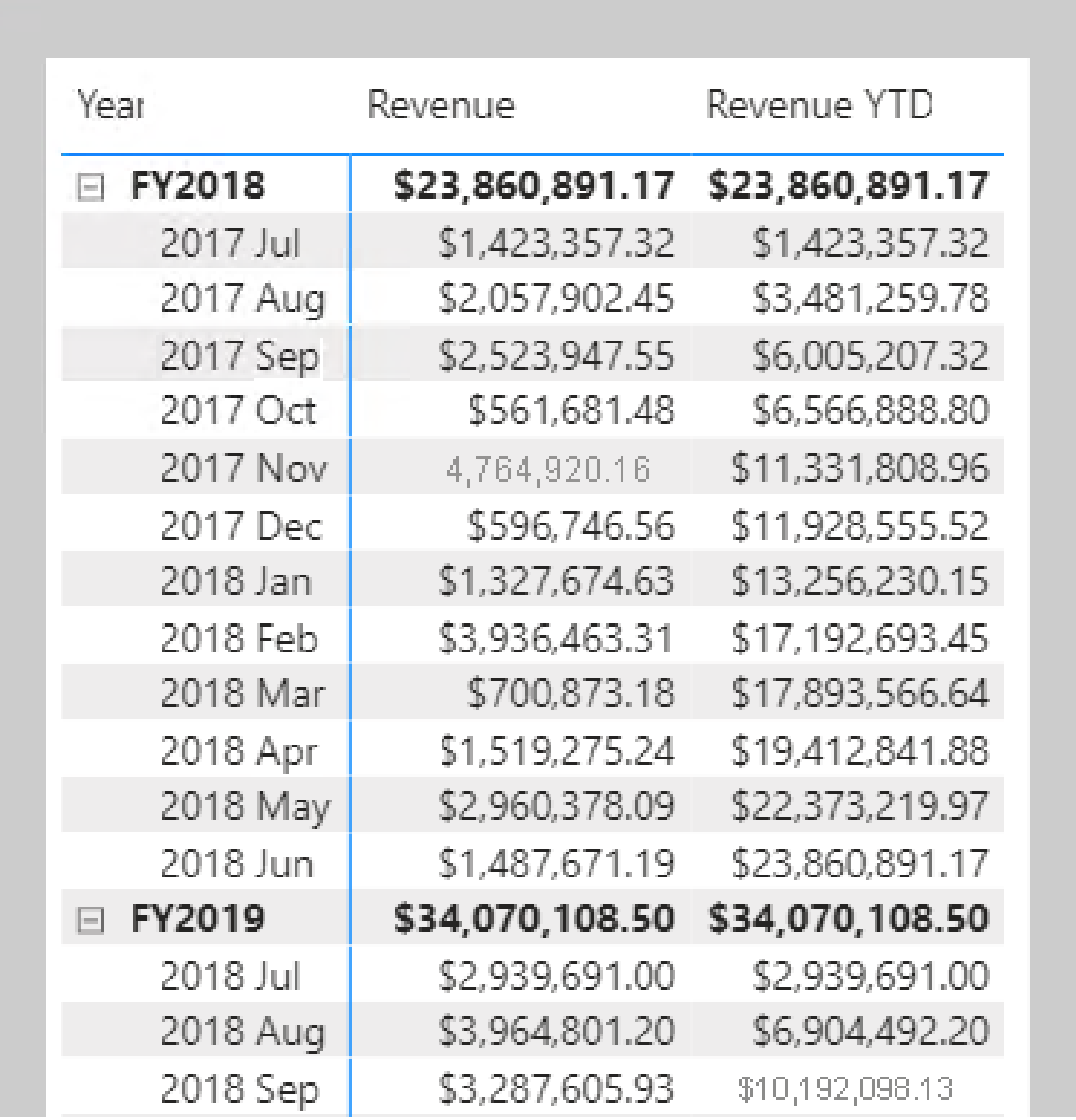
```
Total Ventas = SUMX( Ventas; Ventas[PRECIO] * Ventas[CANTIDAD] )
```



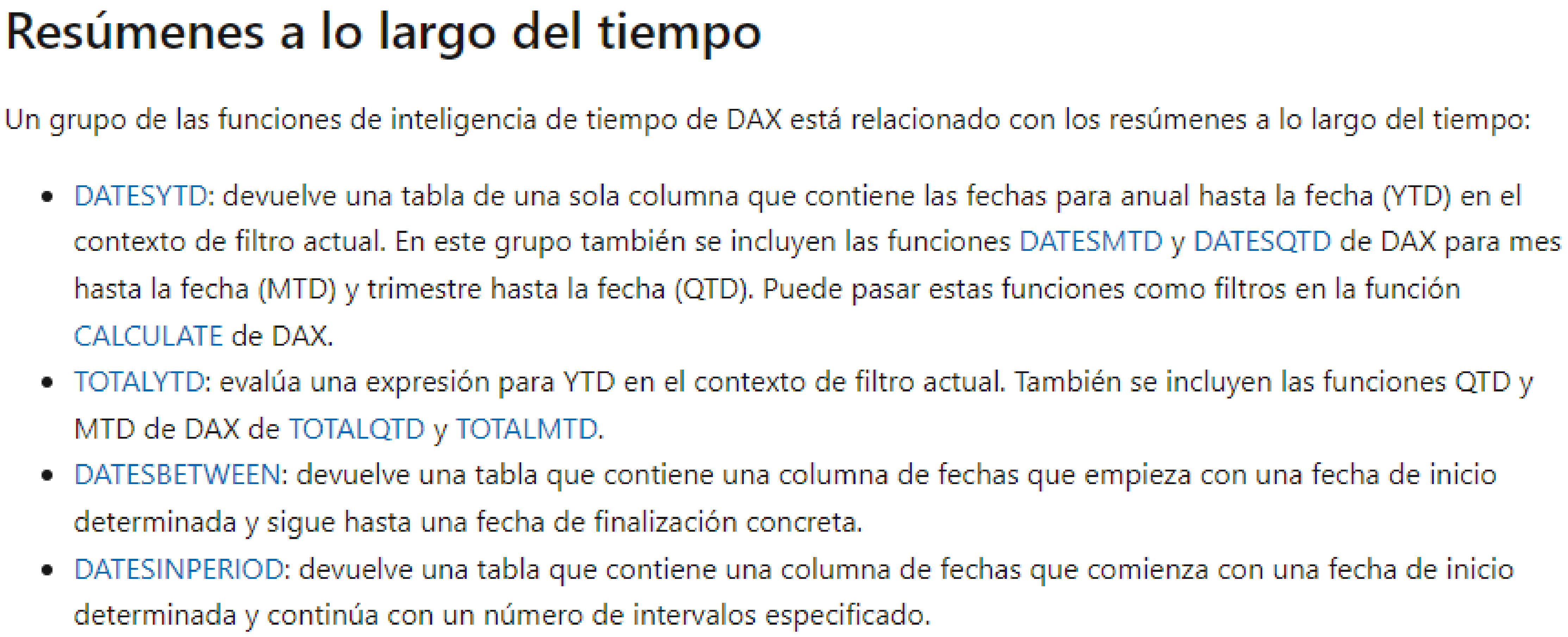
Producto	Precio	Cantidad	Total Ventas
Mandarina	15	17	255
Melón	20	10	200
Naranja	25	5	125
Total	**60**	**32**	**580**

LABORATORIO 4

TIME INTELLIGENCE



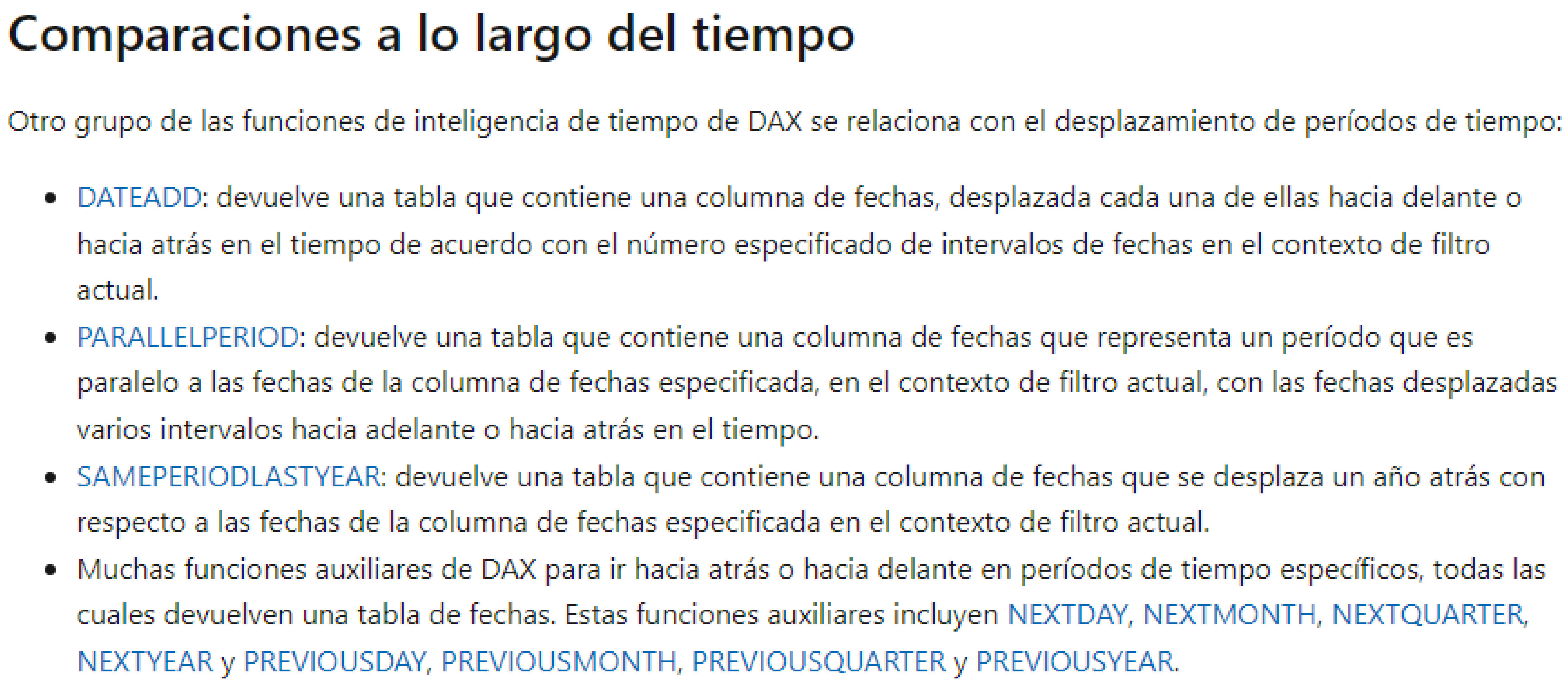
Year	Revenue	Revenue YTD
⊟ **FY2018**	**$23,860,891.17**	**$23,860,891.17**
2017 Jul	$1,423,357.32	$1,423,357.32
2017 Aug	$2,057,902.45	$3,481,259.78
2017 Sep	$2,523,947.55	$6,005,207.32
2017 Oct	$561,681.48	$6,566,888.80
2017 Nov	4,764,920.16	$11,331,808.96
2017 Dec	$596,746.56	$11,928,555.52
2018 Jan	$1,327,674.63	$13,256,230.15
2018 Feb	$3,936,463.31	$17,192,693.45
2018 Mar	$700,873.18	$17,893,566.64
2018 Apr	$1,519,275.24	$19,412,841.88
2018 May	$2,960,378.09	$22,373,219.97
2018 Jun	$1,487,671.19	$23,860,891.17
⊟ **FY2019**	**$34,070,108.50**	**$34,070,108.50**
2018 Jul	$2,939,691.00	$2,939,691.00
2018 Aug	$3,964,801.20	$6,904,492.20
2018 Sep	$3,287,605.93	$10,192,098.13



Resúmenes a lo largo del tiempo

Un grupo de las funciones de inteligencia de tiempo de DAX está relacionado con los resúmenes a lo largo del tiempo:

- DATESYTD: devuelve una tabla de una sola columna que contiene las fechas para anual hasta la fecha (YTD) en el contexto de filtro actual. En este grupo también se incluyen las funciones DATESMTD y DATESQTD de DAX para mes hasta la fecha (MTD) y trimestre hasta la fecha (QTD). Puede pasar estas funciones como filtros en la función CALCULATE de DAX.
- TOTALYTD: evalúa una expresión para YTD en el contexto de filtro actual. También se incluyen las funciones QTD y MTD de DAX de TOTALQTD y TOTALMTD.
- DATESBETWEEN: devuelve una tabla que contiene una columna de fechas que empieza con una fecha de inicio determinada y sigue hasta una fecha de finalización concreta.
- DATESINPERIOD: devuelve una tabla que contiene una columna de fechas que comienza con una fecha de inicio determinada y continúa con un número de intervalos especificado.



Comparaciones a lo largo del tiempo

Otro grupo de las funciones de inteligencia de tiempo de DAX se relaciona con el desplazamiento de períodos de tiempo:

- DATEADD: devuelve una tabla que contiene una columna de fechas, desplazada cada una de ellas hacia delante o hacia atrás en el tiempo de acuerdo con el número especificado de intervalos de fechas en el contexto de filtro actual.
- PARALLELPERIOD: devuelve una tabla que contiene una columna de fechas que representa un período que es paralelo a las fechas de la columna de fechas especificada, en el contexto de filtro actual, con las fechas desplazadas varios intervalos hacia adelante o hacia atrás en el tiempo.
- SAMEPERIODLASTYEAR: devuelve una tabla que contiene una columna de fechas que se desplaza un año atrás con respecto a las fechas de la columna de fechas especificada en el contexto de filtro actual.
- Muchas funciones auxiliares de DAX para ir hacia atrás o hacia delante en períodos de tiempo específicos, todas las cuales devuelven una tabla de fechas. Estas funciones auxiliares incluyen NEXTDAY, NEXTMONTH, NEXTQUARTER, NEXTYEAR y PREVIOUSDAY, PREVIOUSMONTH, PREVIOUSQUARTER y PREVIOUSYEAR.

LABORATORIO 5