

Indhold

**** Introduction to modeling your data.....	2
How to manage your data relationships	3
Sales-tabel.....	6
sentinet	7
Product 2010-01-01 til 2015-06-30	7
Date: dagligt: 1999 til 2015	8
Manufacturer	8
Create calculated columns: realtime beregnede værdier, ikke-persistente.....	8
Next unit: Optimize data models.....	15
Optimize data models by hiding fields.....	16
Hide fields.....	16
Sort visualization data by another field end default.....	18
Next unit: Create measures	21
Create measures: calculations on your data that are performed realtime, ikke gemt i DB.....	22
Create a measure	22
Apply a measure.....	24
*** Create your own measures in Power BI Desktop Tutorial.....	24
In this article.....	24
Prerequisites	25
Traditionel start-DB-model i Contoso Sales Sample.....	25
Automatic measures created by the system, vist med Σ i Data-tabel-oversigten.....	26
Create and use your own measures.....	29
Data Analysis Expressions (DAX) formula language.	29
Quick measures.....	30
Create a measure	30
Net Sales=netto salg=salg-discount-returnValues, omkostninger er ikke trukket fra	32
Use your measure in the report.....	35
SalesCnt = COUNTROWS(Sales) antal rækker i konteksten	36
Use your measure with a slicer, Udsnit, som filtrer udtrækket.....	36
Use your measure in another measure.....	39
What you've learned.....	43
Next steps.....	43
Feedback	43
Next unit: Create calculated tables	44
*** Create calculated tables, use to cross join two tables	44
Create calculated tables in Power BI Desktop.....	45

In this article.....	45
Create a calculated table	46
Functions for calculated tables	48
Next unit: Explore time-based data.....	48
**** Explore time-based data	49
Calendar-tabellen i Contoso har DateKey-felt(date.time) skjult default, men det kan unhides,	51
Et stablet søjlediagram, hvis flere y-værdier, ville de være lagt ovenpå hinanden.....	53
vise NetSales for alle måneder i valgt år	53
Next unit: Check your knowledge	53
**** Create measures for data analysis in Power BI Desktop.....	54
In this article.....	54
Understanding measures.....	54
Data Analysis Expressions.....	56
Let's look at an example	56
Data categories for measures	57
Organizing your measures	58
Dynamic format strings.....	59
Learn more.....	59

**** Introduction to modeling your data

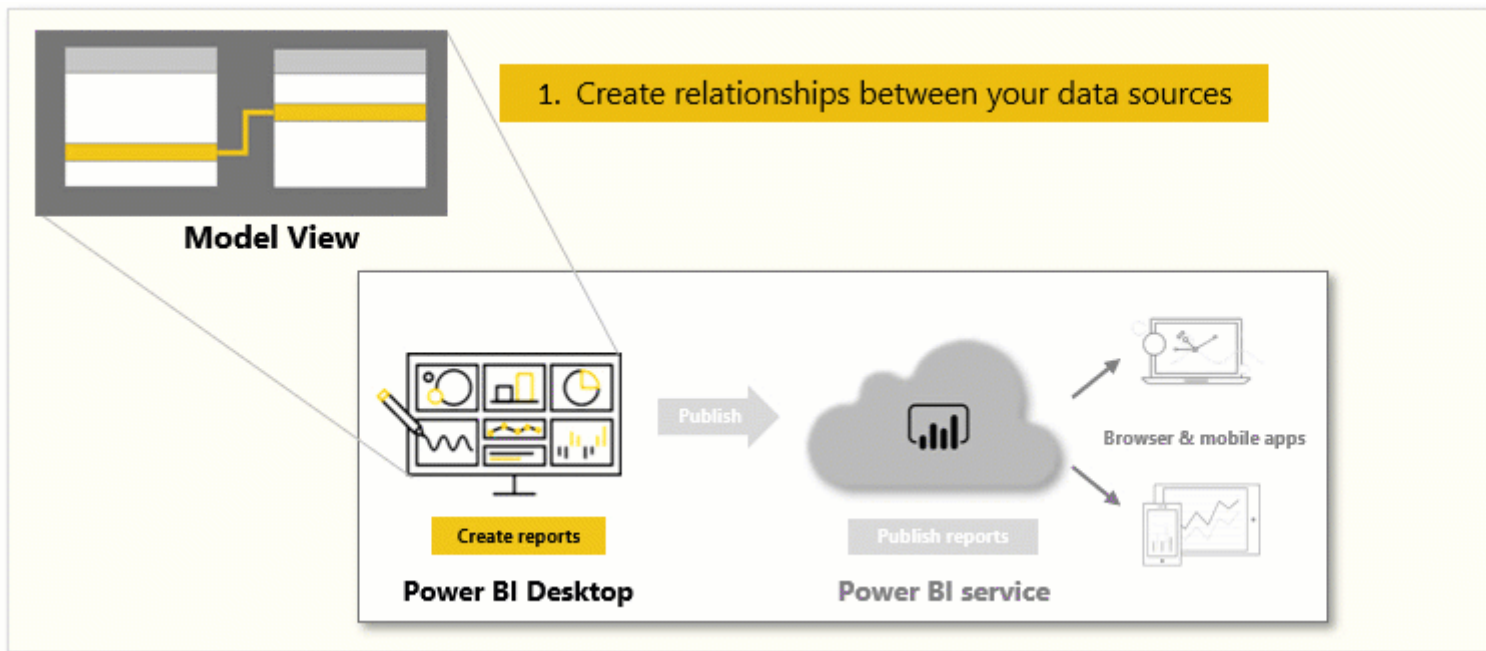
Completed 100 XP

- 5 minutes

Often, you'll connect to multiple data sources to create your reports. All that data needs to work together to create a cohesive report. *Modeling* is how to get your connected data ready for use.

Tasks in this module:

- Create relationships between your data sources
- Create a new field with calculated columns
- Optimize data by hiding fields and sorting visualization data
- Create a measure to perform calculations on your data
- Use a calculated table to create a relationship between two tables
- Format time-based data so that you can drill down for more details



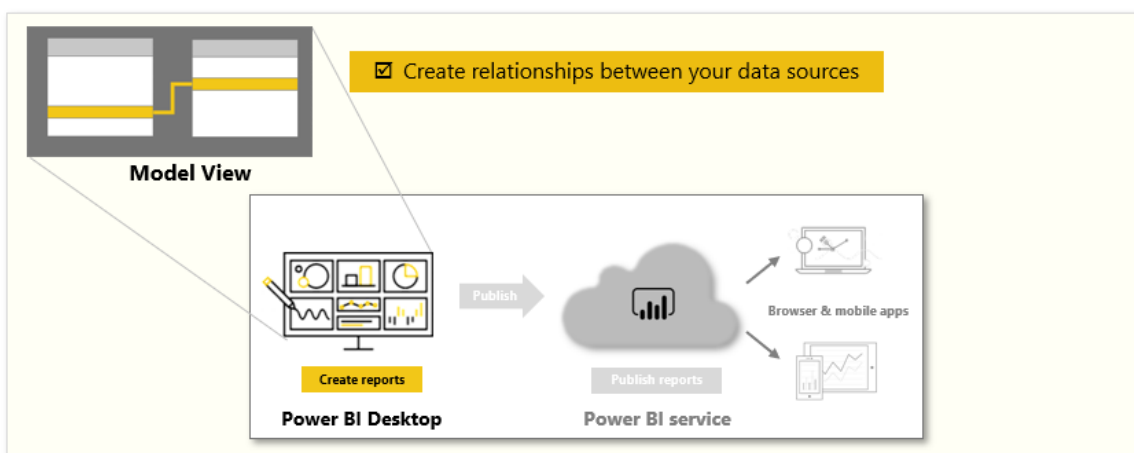
How to manage your data relationships

Completed 100 XP

- 8 minutes

The **Model view** in Power BI Desktop allows you to visually set the relationship between tables or elements. A relationship is where two or more tables are linked together because they contain related data. This enables users to run queries for related data across multiple tables. Use the Model view to see a diagrammatic view of your data.

Tasks in this unit



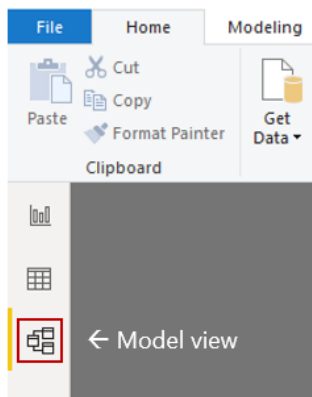
include:

Note

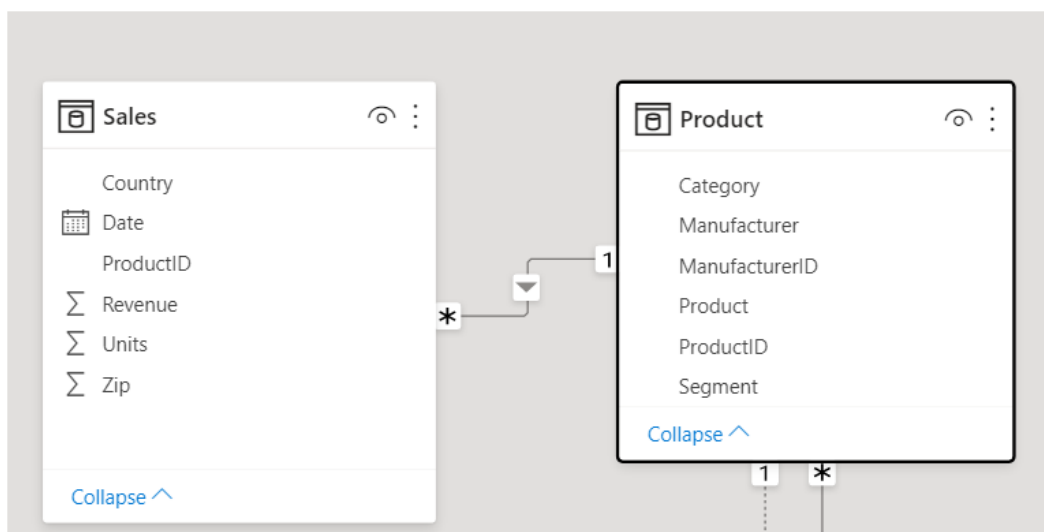
To follow along with the examples on this page, download the sample Access database [here](#) and import all the tables into Power BI Desktop (**Get Data > Database > Access database**). If you have any issues loading the Access database, please read this [article](#).

C:\Users\bh\Documents\programming\Power BI Desktop\PowerBI_sample

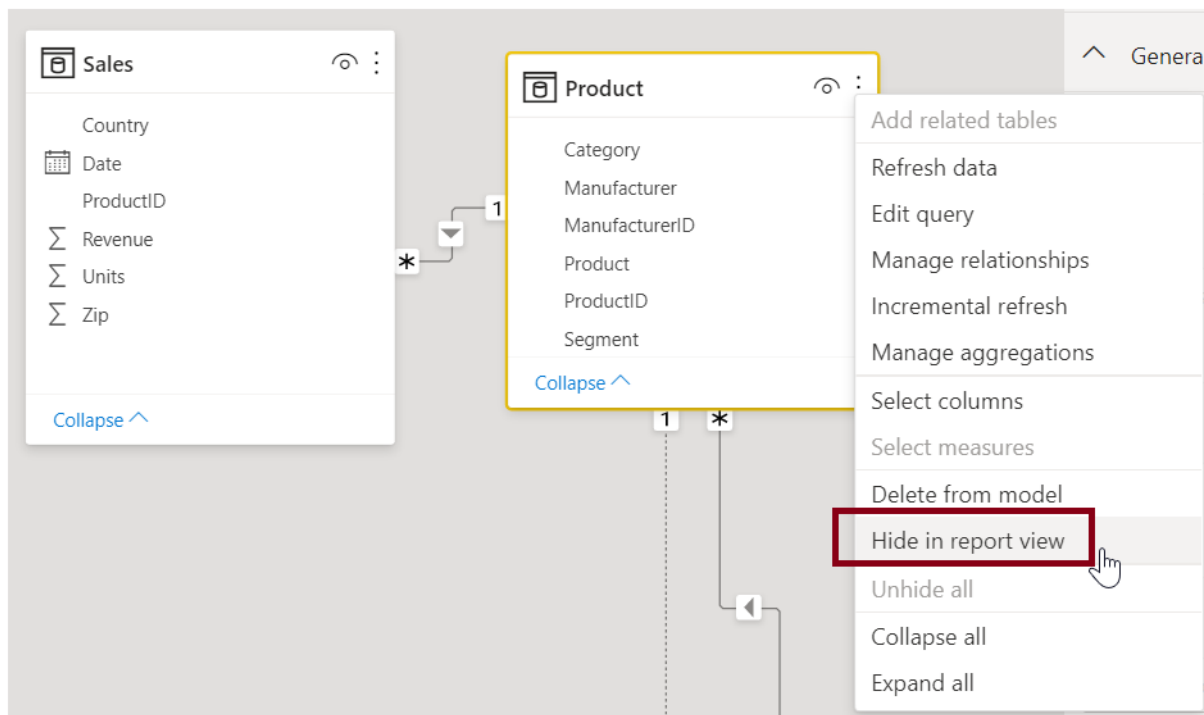
In the Model view, notice that a block represents each table and the lines between them represent relationships.



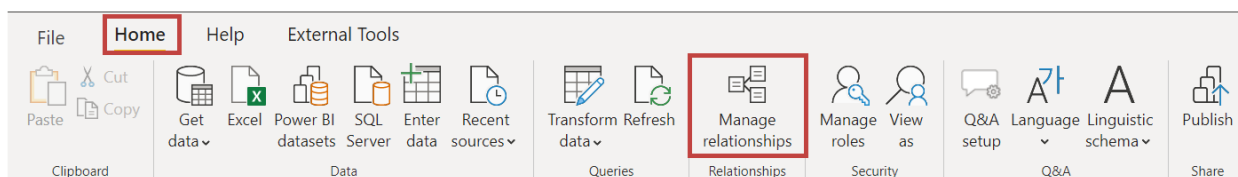
Adding and removing relationships is straightforward. To remove a relationship, right-click the relationship and select **Delete**. To create a relationship, drag the field from one table and drop the field on the field of the other table that you want to link.



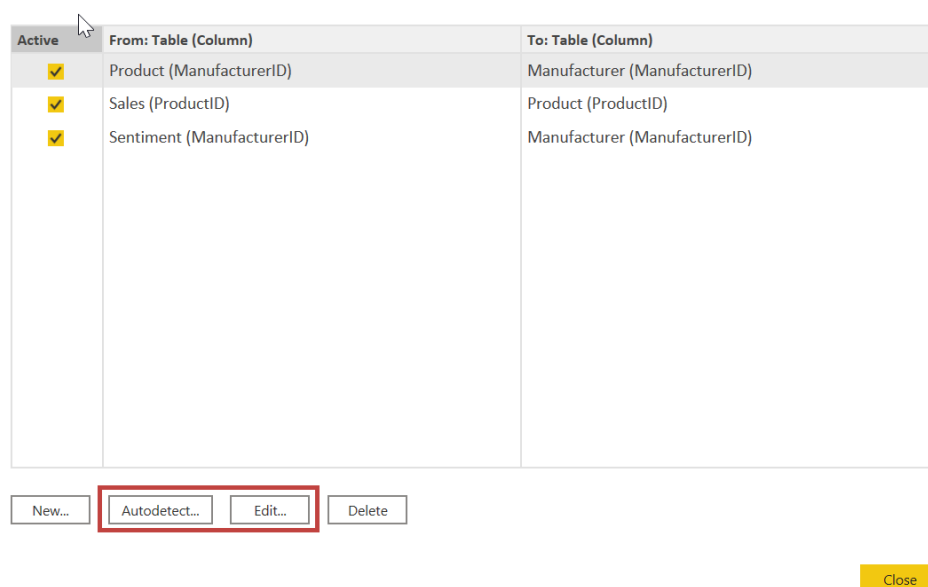
To hide a table or individual column from your report, right-click the table or column in the Model view and select **Hide in report view**.



For a more detailed view of your data relationships, on the **Home** tab, select **Manage Relationships**. The **Manage Relationships** dialog box displays your relationships as a list instead of as a visual diagram. From the dialog box, you can select **Autodetect** to find relationships in new or updated data. Select **Edit** to manually edit your relationships. You'll find advanced options in the Edit section to set the *Cardinality* and *Cross-filter* direction of your relationships.



Manage relationships



Your options for Cardinality are explained in the following table.

Cardinality options

Example

- Many to One** The most common default relationship. The column in one table can have more than one instance of a value (or lookup table) has only one instance of a value.
- One to One** The column in one table has only one instance of a particular value, and the other related table has only one particular value.

Generally, we recommend minimizing the use of bi-directional relationships. They can negatively impact model query performance, and possibly deliver confusing experiences for your report users.

Setting accurate relationships between your data allows you to create complex calculations across multiple data elements.

For more information, see: [Create and manage relationships in Power BI Desktop](#) and [Bi-directional relationship guidance](#).

Next unit: Create calculated columns

Sales-tabel

Sales.revenue =antal*ProductPris

ProductID	Date	Zip	Units	Revenue	Country
1009	23-02-2011 00:00:00	64735	2	262,395	USA
1009	10-03-2011 00:00:00	78704	2	262,395	USA
1009	11-02-2011 00:00:00	28078	2	262,395	USA
1009	11-03-2011 00:00:00	32082	2	262,395	USA
1009	31-01-2011 00:00:00	70433	2	262,395	USA
1009	31-01-2011 00:00:00	58237	2	262,395	USA
1009	08-01-2011 00:00:00	74052	2	262,395	USA
1009	30-11-2010 00:00:00	32080	2	262,395	USA
1009	09-11-2010 00:00:00	15215	2	262,395	USA
1009	29-09-2010 00:00:00	17522	2	262,395	USA
1009	20-09-2010 00:00:00	77706	2	262,395	USA
1009	31-08-2010 00:00:00	20854	2	262,395	USA
1009	15-09-2010 00:00:00	46168	2	262,395	USA
1009	20-08-2010 00:00:00	32837	2	262,395	USA
506	29-06-2014 00:00:00	59330	1	1359,6975	France
506	18-06-2014 00:00:00	31620	1	1359,6975	France
506	18-06-2014 00:00:00	77590	1	1359,6975	France
506	17-06-2014 00:00:00	80132	1	1359,6975	France
506	17-06-2014 00:00:00	72130	1	1359,6975	France
506	25-06-2014 00:00:00	80132	1	1359,6975	France
506	23-06-2014 00:00:00	85120	1	1359,6975	France
506	23-06-2014 00:00:00	80136	1	1359,6975	France
506	16-06-2014 00:00:00	42240	1	1359,6975	France
506	16-06-2014 00:00:00	27850	1	1359,6975	France
506	16-06-2014 00:00:00	78660	1	1359,6975	France
506	16-06-2014 00:00:00	53140	1	1359,6975	France
506	16-06-2014 00:00:00	47150	1	1359,6975	France

2.1M , sum= 1323652888.1098685, avg=630.310

Sales starter i 2010-01-01, slut 2015-06-30, dates dækker 2009 t

Sentiment=user-score									
DateID	StateID	ManufacturerID	Score	Manufacturer	Date	State	zip	ProductID	
8	38	8	80	Natura	01-02-2014 00:00:00	RI	02801	694	
8	11	8	65	Natura	01-02-2014 00:00:00	GA	39813	694	
8	23	8	68	Natura	01-02-2014 00:00:00	MO	63005	694	
8	40	8	81	Natura	01-02-2014 00:00:00	SD	57001	694	
8	36	8	73	Natura	01-02-2014 00:00:00	OR	00068	694	
8	46	8	68	Natura	01-02-2014 00:00:00	WA	00072	694	
8	14	8	62	Natura	01-02-2014 00:00:00	IL	62356	694	
8	21	8	83	Natura	01-02-2014 00:00:00	MI	48001	694	
8	33	8	85	Natura	01-02-2014 00:00:00	NY	00501	694	
8	39	8	64	Natura	01-02-2014 00:00:00	SC	29001	694	
8	12	8	69	Natura	01-02-2014 00:00:00	IA	50001	694	
8	10	8	65	Natura	01-02-2014 00:00:00	FL	00041	694	
8	9	8	69	Natura	01-02-2014 00:00:00	DE	19701	694	
8	15	8	74	Natura	01-02-2014 00:00:00	IN	46001	694	
8	39	8	83	Natura	01-02-2014 00:00:00	SC	29001	694	
8	4	8	82	Natura	01-02-2014 00:00:00	AZ	00014	694	
8	33	8	81	Natura	01-02-2014 00:00:00	NY	00501	694	
8	36	8	60	Natura	01-02-2014 00:00:00	OR	00068	694	
8	41	8	81	Natura	01-02-2014 00:00:00	TN	42223	694	
8	5	8	81	Natura	01-02-2014 00:00:00	CA	00012	694	
8	16	8	84	Natura	01-02-2014 00:00:00	KS	66002	694	
8	20	8	66	Natura	01-02-2014 00:00:00	MD	20058	694	
8	2	8	78	Natura	01-02-2014 00:00:00	AL	35004	694	
8	47	8	71	Natura	01-02-2014 00:00:00	WI	53001	694	
8	37	8	73	Natura	01-02-2014 00:00:00	PA	15937	694	
8	13	8	70	Natura	01-02-2014 00:00:00	ID	00056	694	
8	19	8	79	Natura	01-02-2014 00:00:00	MA	00158	694	
8	33	8	89	Natura	01-02-2014 00:00:00	NY	00501	694	

ManufacturerID egentlig overflødig i sentiment, kan jo hentes via ProductID, men det går hurtigere at udtrække data

Product 2010-01-01 til 2015-06-30

Product					
ProductID	Product	Category	Segment	ManufacturerID	Manufact
536	Maximus UC-01	Urban	Convenience	7	VanArsdel
537	Maximus UC-02	Urban	Convenience	7	VanArsdel
538	Maximus UC-03	Urban	Convenience	7	VanArsdel
539	Maximus UC-04	Urban	Convenience	7	VanArsdel
540	Maximus UC-05	Urban	Convenience	7	VanArsdel
541	Maximus UC-06	Urban	Convenience	7	VanArsdel
542	Maximus UC-07	Urban	Convenience	7	VanArsdel
543	Maximus UC-08	Urban	Convenience	7	VanArsdel
544	Maximus UC-09	Urban	Convenience	7	VanArsdel
545	Maximus UC-10	Urban	Convenience	7	VanArsdel
546	Maximus UC-11	Urban	Convenience	7	VanArsdel
547	Maximus UC-12	Urban	Convenience	7	VanArsdel
548	Maximus UC-13	Urban	Convenience	7	VanArsdel
549	Maximus UC-14	Urban	Convenience	7	VanArsdel
550	Maximus UC-15	Urban	Convenience	7	VanArsdel
551	Maximus UC-16	Urban	Convenience	7	VanArsdel
552	Maximus UC-17	Urban	Convenience	7	VanArsdel
553	Maximus UC-18	Urban	Convenience	7	VanArsdel
554	Maximus UC-19	Urban	Convenience	7	VanArsdel
555	Maximus UC-20	Urban	Convenience	7	VanArsdel
556	Maximus UC-21	Urban	Convenience	7	VanArsdel
557	Maximus UC-22	Urban	Convenience	7	VanArsdel
558	Maximus UC-23	Urban	Convenience	7	VanArsdel
559	Maximus UC-24	Urban	Convenience	7	VanArsdel
560	Maximus UC-25	Urban	Convenience	7	VanArsdel
561	Maximus UC-26	Urban	Convenience	7	VanArsdel

2412, Manufacturer-name bør ikke stå i Product, gør

nok tabellen udnødigt 15% større, men et join undgås

Categories: Urban, rural, youth, mix (-lifestyle ?!)

Segment: Convenience, Extreme, Moderation, Regular, All Season, Youth, Select, Productivity

Date: dagligt: 1999 til 2015

X ✓		Date					
Date	MonthNo	MonthName	MonthID	Month	Quarter	Year	
2015-01-05	1	Jan	201501	Jan-15	Q1	2015	
1999-01-02	1	Jan	199901	Jan-99	Q1	1999	
1999-01-03	1	Jan	199901	Jan-99	Q1	1999	
1999-01-04	1	Jan	199901	Jan-99	Q1	1999	
1999-01-05	1	Jan	199901	Jan-99	Q1	1999	
1999-01-06	1	Jan	199901	Jan-99	Q1	1999	
1999-01-07	1	Jan	199901	Jan-99	Q1	1999	
1999-01-08	1	Jan	199901	Jan-99	Q1	1999	
1999-01-09	1	Jan	199901	Jan-99	Q1	1999	
1999-01-10	1	Jan	199901	Jan-99	Q1	1999	
1999-01-11	1	Jan	199901	Jan-99	Q1	1999	
1999-01-12	1	Jan	199901	Jan-99	Q1	1999	
1999-01-13	1	Jan	199901	Jan-99	Q1	1999	
1999-01-14	1	Jan	199901	Jan-99	Q1	1999	
1999-01-15	1	Jan	199901	Jan-99	Q1	1999	
1999-01-16	1	Jan	199901	Jan-99	Q1	1999	

6209, en række per d,

altså $6209/365 \approx 17$ år, 1999 til 2015

Manufacturer

Manufacturer	
ManufacturerID	Manufacturer
1	Abbas
2	Aliqui
3	Barba
4	Currus
5	Fama
6	Leo
7	VanArsdel
8	Natura
9	Palma
10	Pirum
11	Pomum
12	Quibus
13	Salvus
14	Victoria

kun 14 forskellige, så hver har i snit $2412/14 \approx 172$ produkter

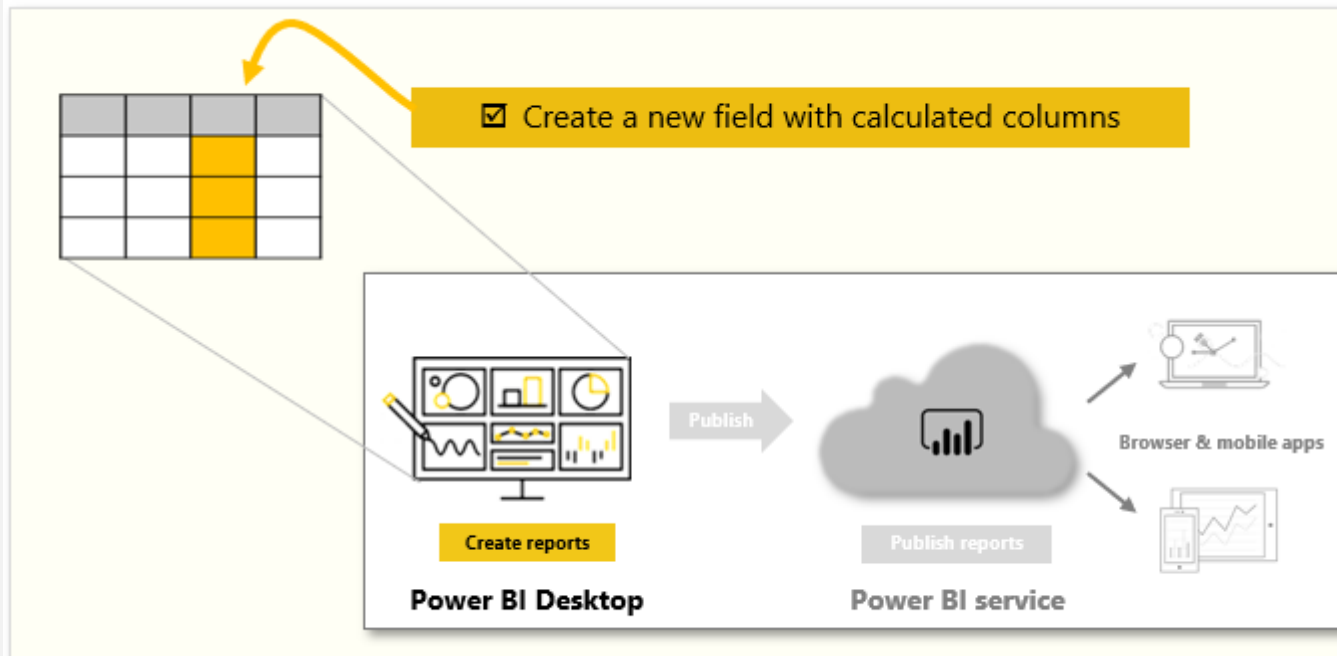
Create calculated columns: realtime beregnede værdier, ikke-persistente

Completed 100 XP

- 7 minutes

Sometimes, the data that you're analyzing doesn't contain a field that you need. The answer might be *calculated columns*. You can create a new calculated column by transforming two or more elements of existing data. For example, you can create a new column by combining two columns into one.

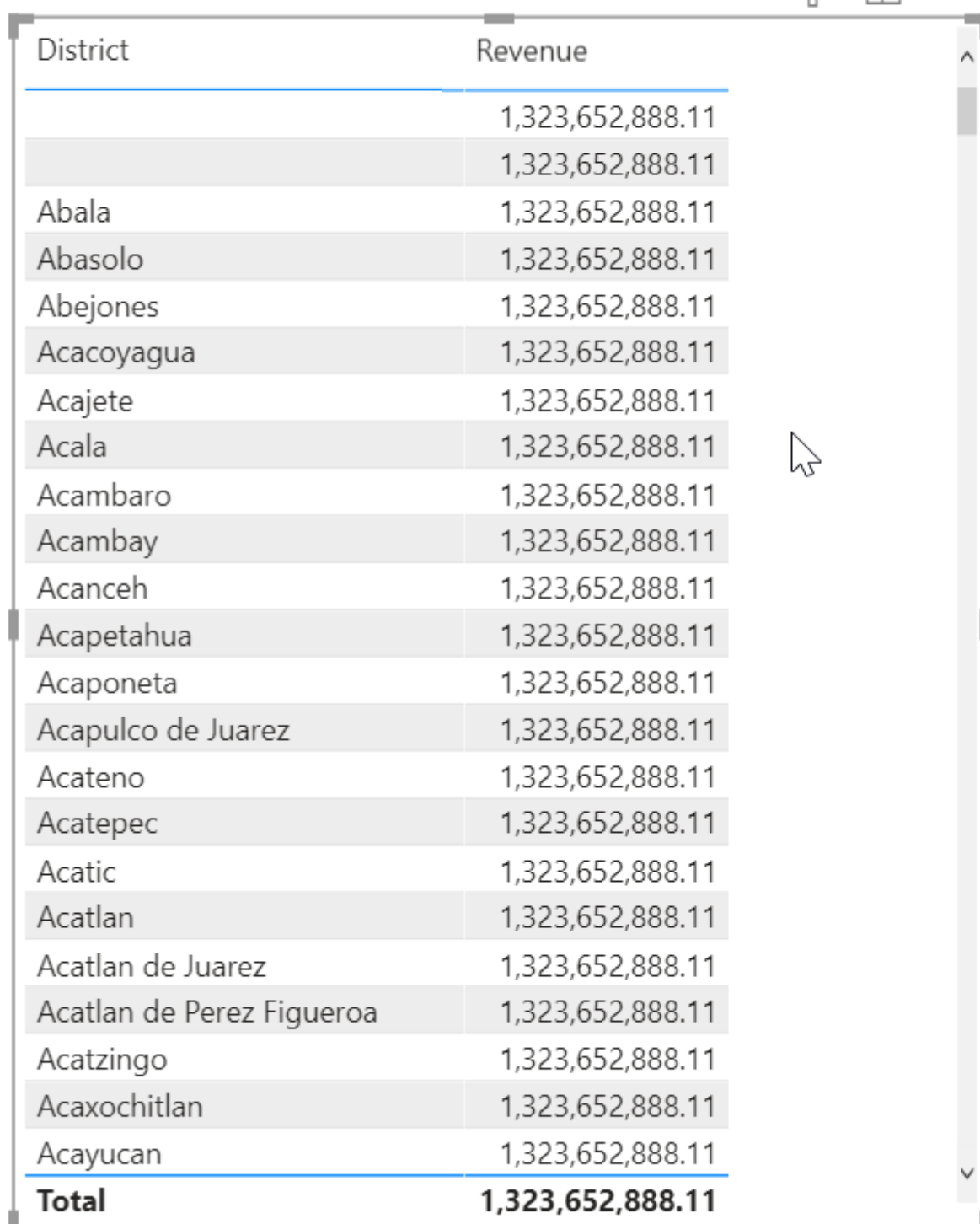
Tasks in this unit include:



Note

To follow along with the examples on this page, download the sample Access database [here](#) and import into Power BI Desktop (**Get Data > Database > Access database**). If you have any issues loading the Access database, please read this [article](#).

One reason for creating a calculated column is to establish a relationship between tables when no unique fields exist. **The lack of a relationship becomes obvious when you create a simple table visual in Power BI Desktop and get the same value for all entries.**

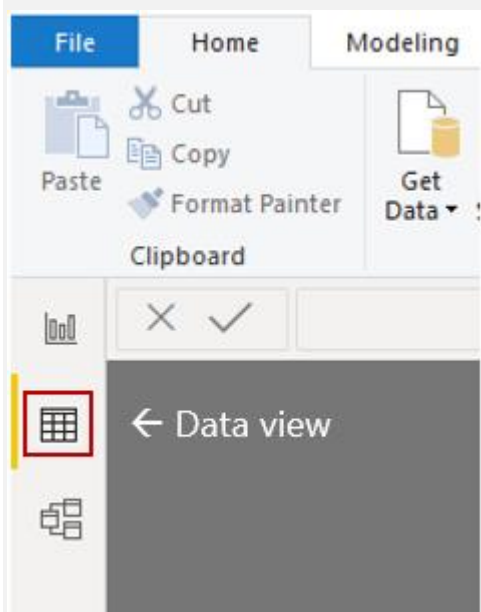


District	Revenue
	1,323,652,888.11
	1,323,652,888.11
Abala	1,323,652,888.11
Abasolo	1,323,652,888.11
Abejones	1,323,652,888.11
Acacoyagua	1,323,652,888.11
Acajete	1,323,652,888.11
Acala	1,323,652,888.11
Acambaro	1,323,652,888.11
Acambay	1,323,652,888.11
Acanceh	1,323,652,888.11
Acapetahua	1,323,652,888.11
Acaponeta	1,323,652,888.11
Acapulco de Juarez	1,323,652,888.11
Acateno	1,323,652,888.11
Acatepec	1,323,652,888.11
Acatic	1,323,652,888.11
Acatlan	1,323,652,888.11
Acatlan de Juarez	1,323,652,888.11
Acatlan de Perez Figueroa	1,323,652,888.11
Acatzingo	1,323,652,888.11
Acaxochitlan	1,323,652,888.11
Acayucan	1,323,652,888.11
Total	1,323,652,888.11

1323 mio

For example, to create a relationship with unique fields in data, you can create a new calculated column for "CountryZip" by combining the values from the Country and the Zip columns.

To create a calculated column, select the **Data view** in Power BI Desktop from the left side of the report canvas.



vælg Geography table

From the **Table tools** tab, select **New Column** to enable the formula bar. You can enter calculations by using Data Analysis Expressions (**DAX**) language. DAX is a powerful formula language that lets you build robust calculations. As you type a formula, Power BI Desktop displays matching formulas or data elements to assist and accelerate the creation of your formula.

The Power BI formula bar will suggest specific DAX functions and related data columns as you enter your expression.

Skriver 'country' så foreslås Geography[Country]

```
CountryZip = Geography[Country]&"", "&Geography[Zip]
```

File Home Help External Tools **Table tools** Column

Name Geography

Mark as date table ▾

Manage relationships

New measure Quick measure

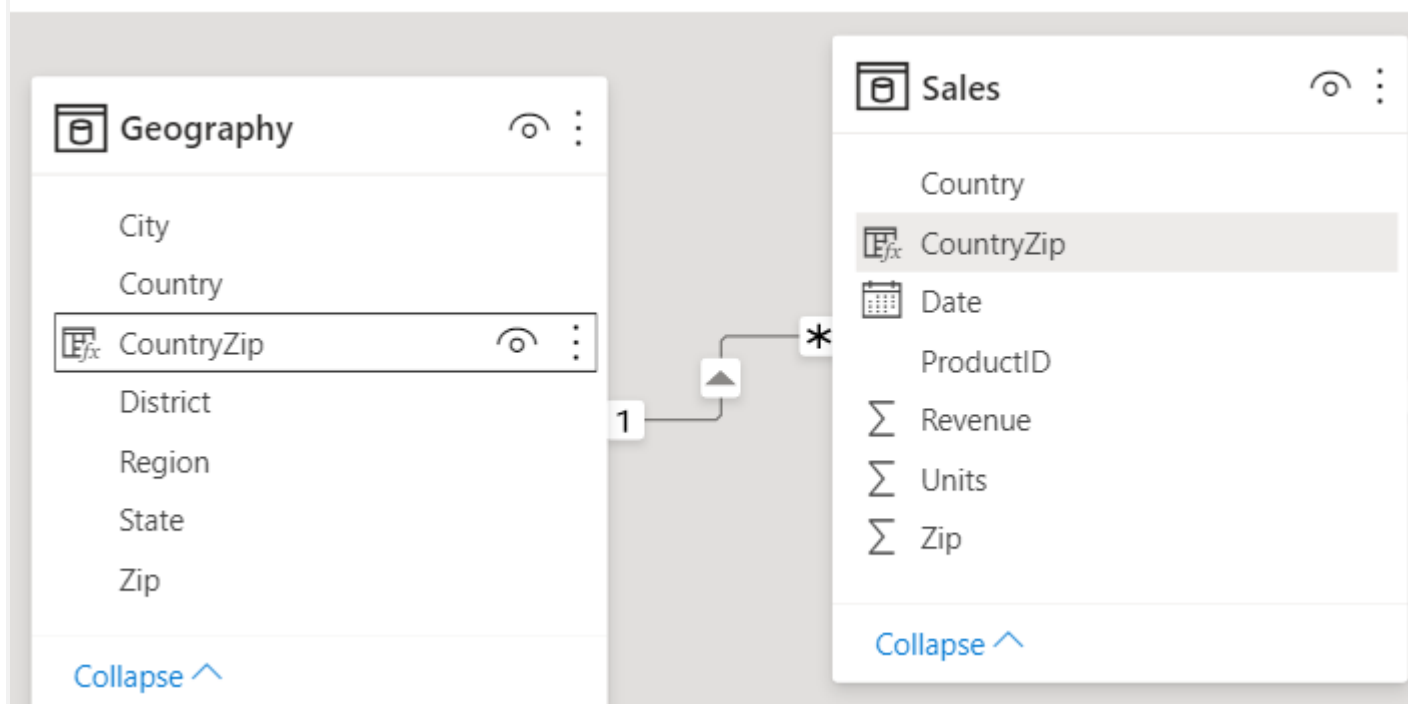
Structure Calendars Relationships Calculations

CountryZip = Geography[Country] & ", " & Geography[Zip]

Zip	City	State	Region	District	Country
68274		Oaxaca		Oaxaca de Juarez	Mexico
68275		Oaxaca		Oaxaca de Juarez	Mexico
68276		Oaxaca		Oaxaca de Juarez	Mexico
71512		Oaxaca		Ocotlan de Morelos	Mexico
71513		Oaxaca		Ocotlan de Morelos	Mexico
71515		Oaxaca		Ocotlan de Morelos	Mexico
71565		Oaxaca		La Pe	Mexico
71566		Oaxaca		La Pe	Mexico
71750		Oaxaca		Pinotepa de Don Luis	Mexico

CountryZip skal selvfølgelig kompeneres præcist på samme måde i de to tabeller, lave jo string sammenligning

After you have created the new CountryZip calculated column in the Geography table and the Sales table, they can be used as a unique key to establish a relationship between the two tables. By going to the **Relationship** view, you can then drag the CountryZip field from the Sales table to the Geography table to create the relationship. Fra een CountryZip er der mange tilhørende salg



When you return to the **Report** view, notice that a different value for each district shows. For hver CountryZip finds tilhørende Sales[], og hver Sales brugs aggregated felt sum(Revenue)

District	Revenue
	63,578,104.59
Abala	24,200.77
Acajete	103,786.04
Acanceh	8,505.53
Acayucan	187,536.40
Ahome	206,798.02
Alaquines	17,712.92
Almoleya Del Rio	14,755.97
Altotonga	415,359.00
Alvarado	16,364.83
Amatepec	422,143.84
Amatitlan	17,252.34
Amatlan de los Reyes	378,426.35
Angel R Cabada	239,830.92
Angostura	504,536.81
Aquismon	533,129.52
Atenco	332,793.88
Atoyac	93,379.28
Aude	655,627.19
Aveyron	22,605.98
Axtla de Terrazas	969,827.41
Baca	5,339.46
Badiraguato	819,829.29
Total	1,323,652,888.11

For more information on calculated columns, including the use of IF statements, see [Tutorial: Create calculated columns in Power BI Desktop](#).

Next unit: Optimize data models

[Continue](#)

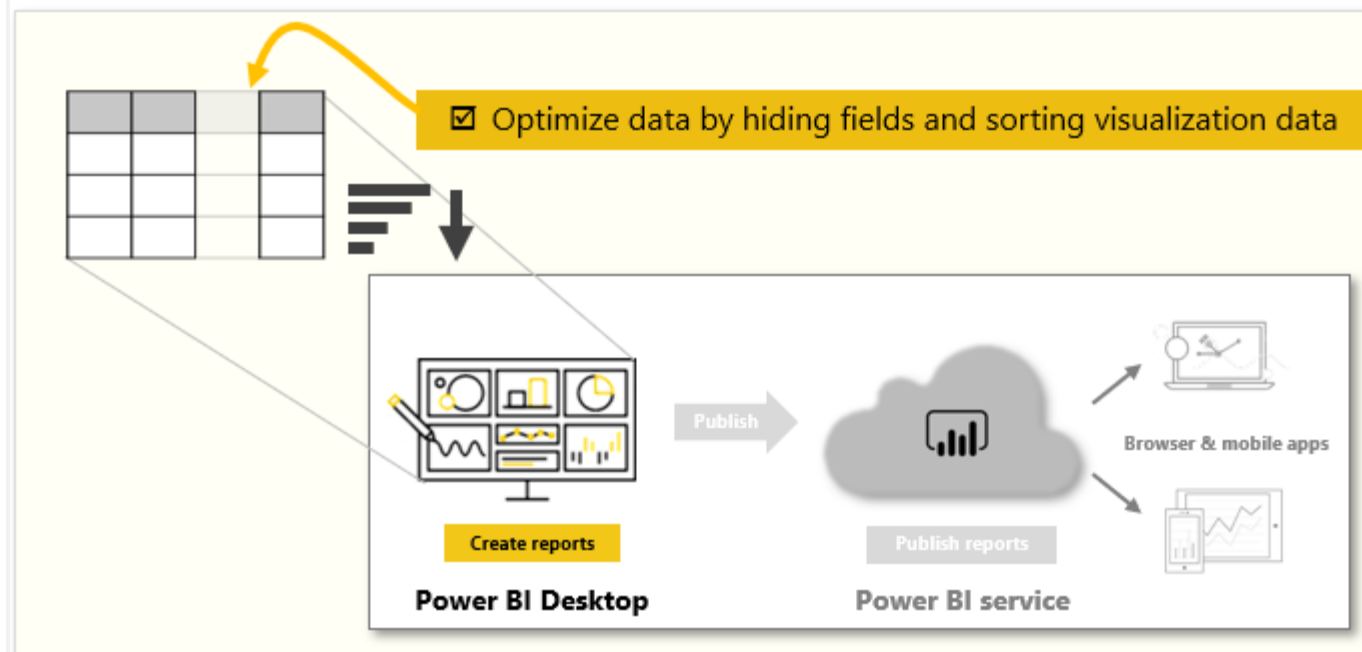
Optimize data models by hiding fields

Completed 100 XP

- 8 minutes

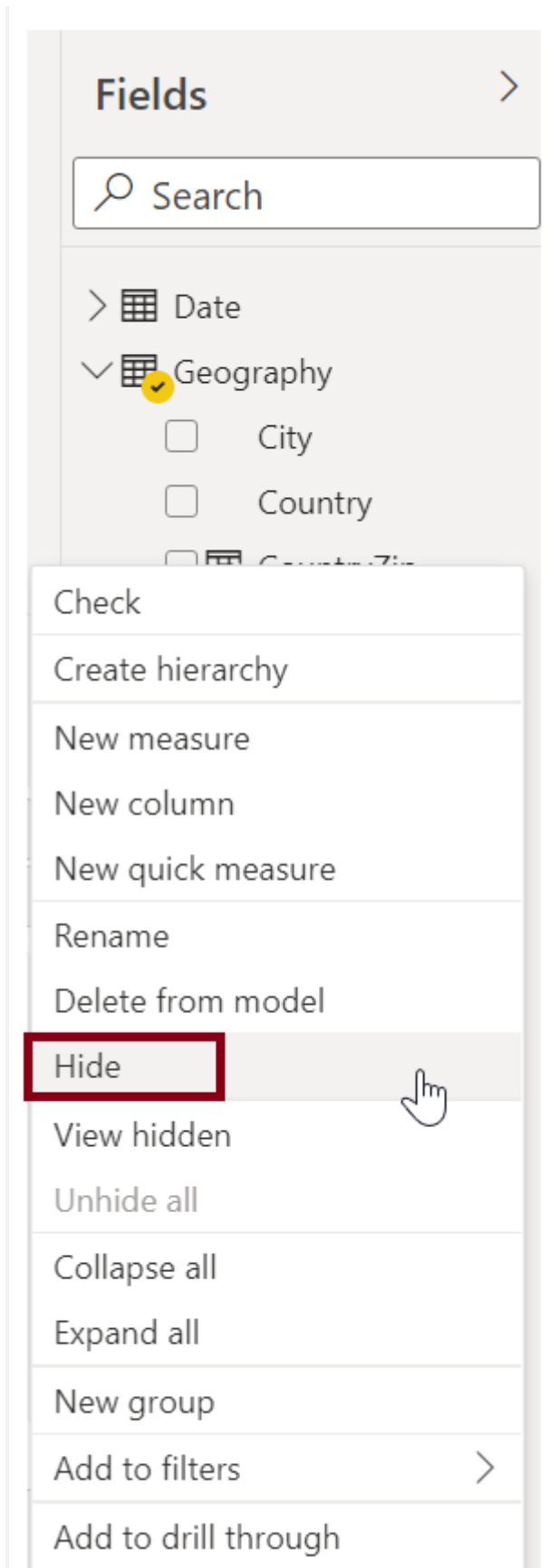
Imported data often contains fields that you don't need for your reporting and visualization tasks. Data might contain unnecessary information or it might be available in another column. Power BI Desktop has tools to optimize your data and make it more usable for building reports and visuals.

Tasks in this module include:



Hide fields

To hide a field in the Fields, Data, pane of Power BI Desktop, right-click the column and select **Hide**. Your hidden fields aren't deleted. If you've used a hidden field in existing visuals, the data is still there; the hidden field just isn't displayed on the Fields pane.

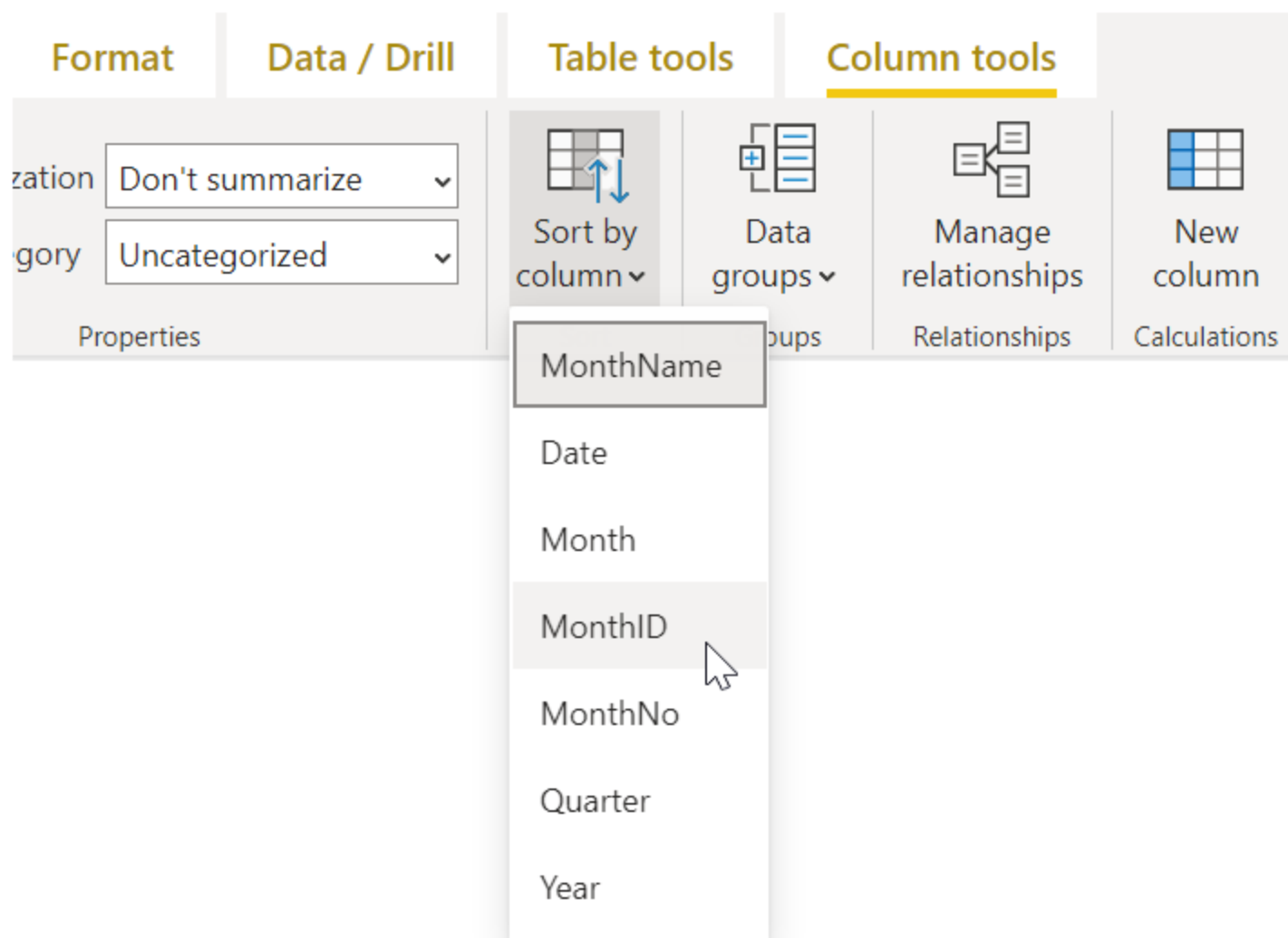


If you view tables in the **Model** view, hidden fields appear dimmed. The data in these tables is still available and is still part of the model. You can unhide any field that has been hidden by right-clicking the field and then selecting **unhide**.

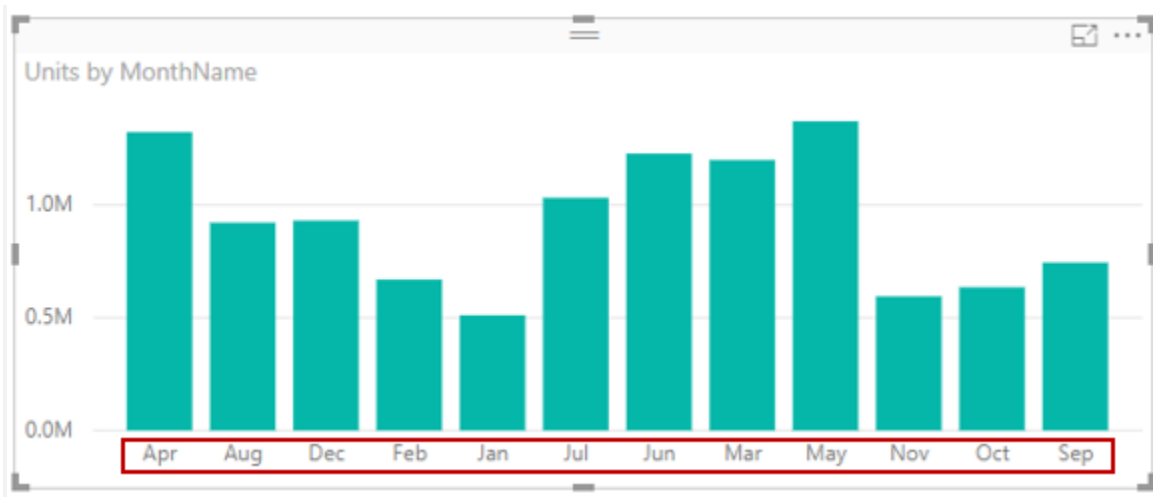
Sort visualization data by another field end default

The **Sort by Column** tool, available on the **Column tools** tab, is useful to help ensure that your data is displayed in the order that you intended.

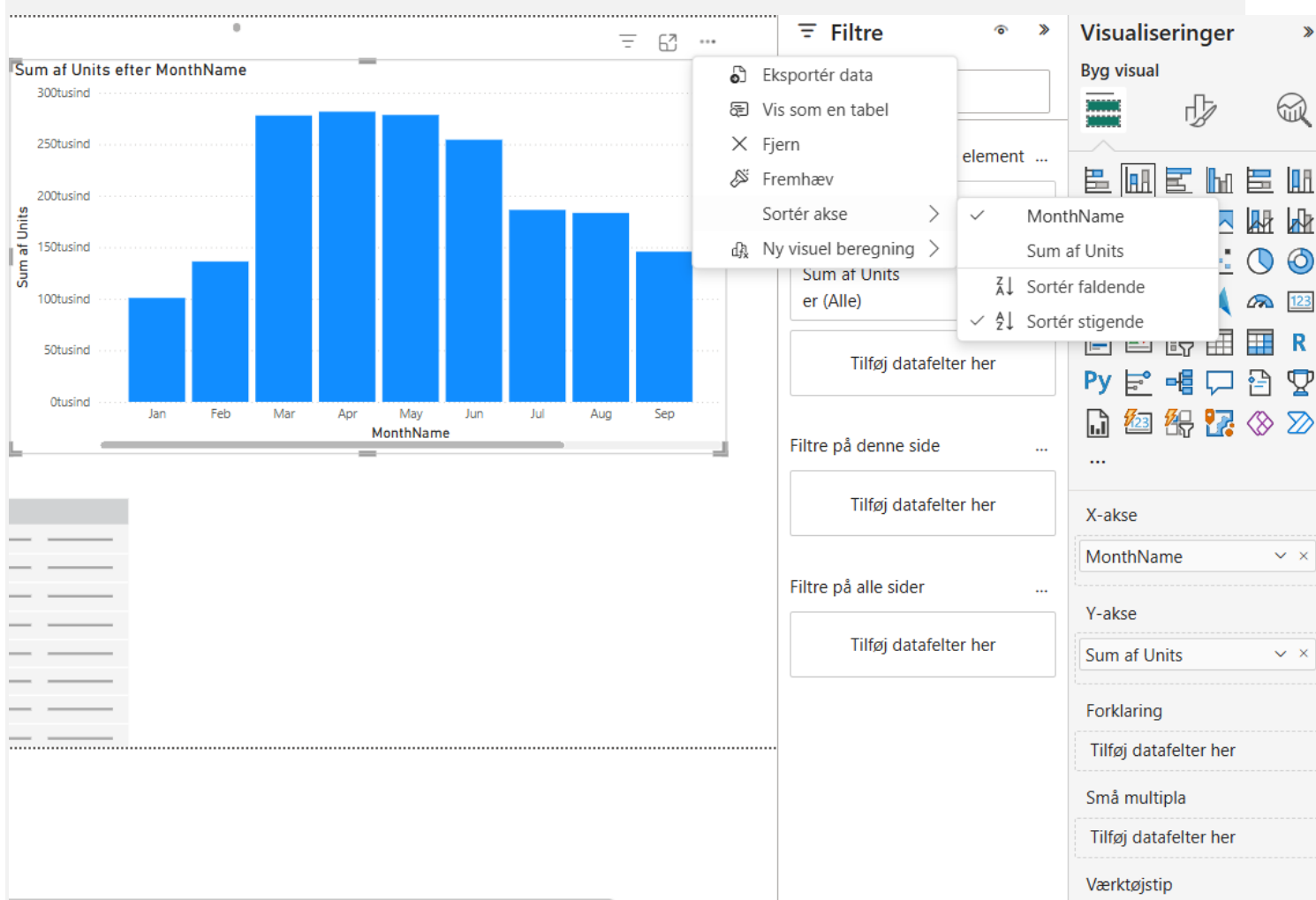
Først skal kolonnen markeres, MonthNo (ikke sikkert nødvendigt, selv om MonthName er valgt, kan andre kolonner vælges):



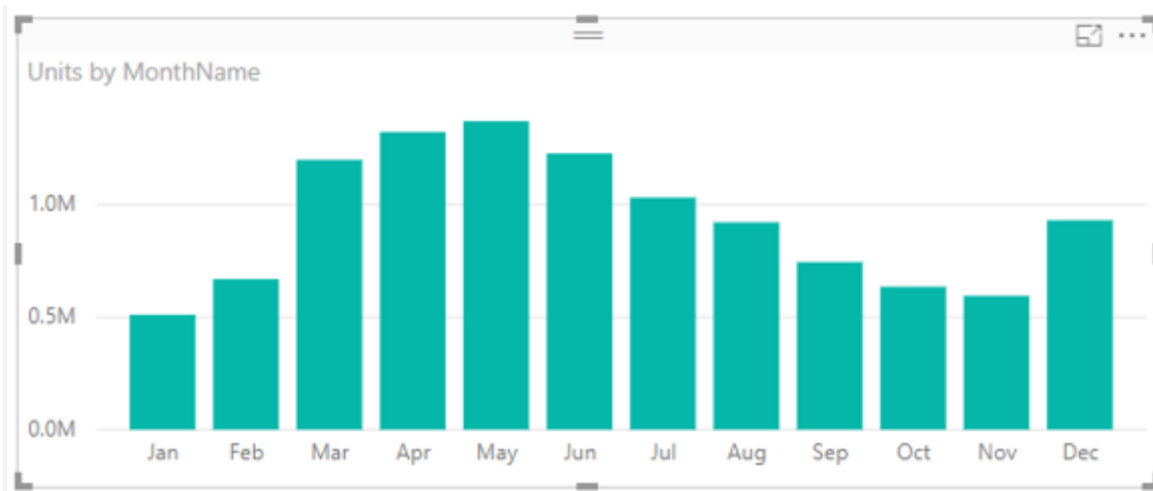
As a common example, data that includes the name of the month is sorted alphabetically by default, for example, August would appear before February.



In this case, selecting the **MonthName** field in the **Fields** list, selecting **Sort by Column** from the **Column tools** tab, and then choosing a field to sort by can remedy the problem. The **MonthNo** category sort option will order the months as intended , **det sker i Tabelvisning, når MonthNo er valgt;**

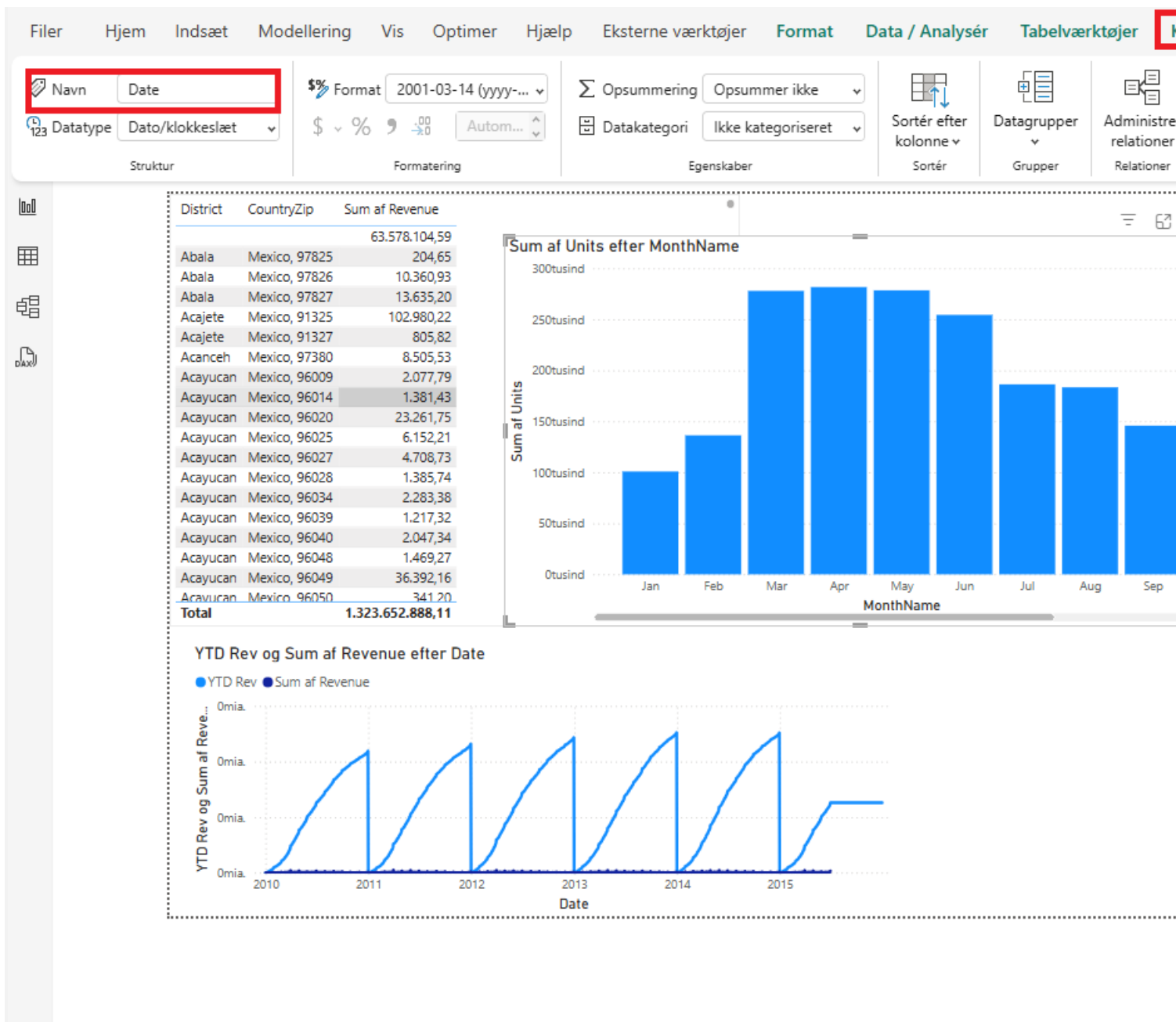


Selvom der vises, at der er sorteret efter MonthName er det øjensynligt ikke sket, men altså efter MonthNo ?!



Datefeltet i Sales er DateTime, skal omgøres til Date for at kunne bruges som key op mod Date
Sales har Date, men for at kunne anvende avanceret date info skal relation til Date bruges.

Setting the data type for a field is another way to optimize your information so that it's handled correctly. To change a data type from the report canvas, select the column in the Fields, Data, pane, and then use the **Format** drop-down menu on the **Column tools** tab to select one of the formatting options. Any visuals you've created that display that field are updated automatically.



Først ingen forbindelse mellem Sales og Date, relationen var aktiv, men virkede ikke, den virkede efter at date format var helt ens i de to tabeller!

For more information, see [Sort by column in Power BI Desktop](#).

Next unit: Create measures

[Continue](#)

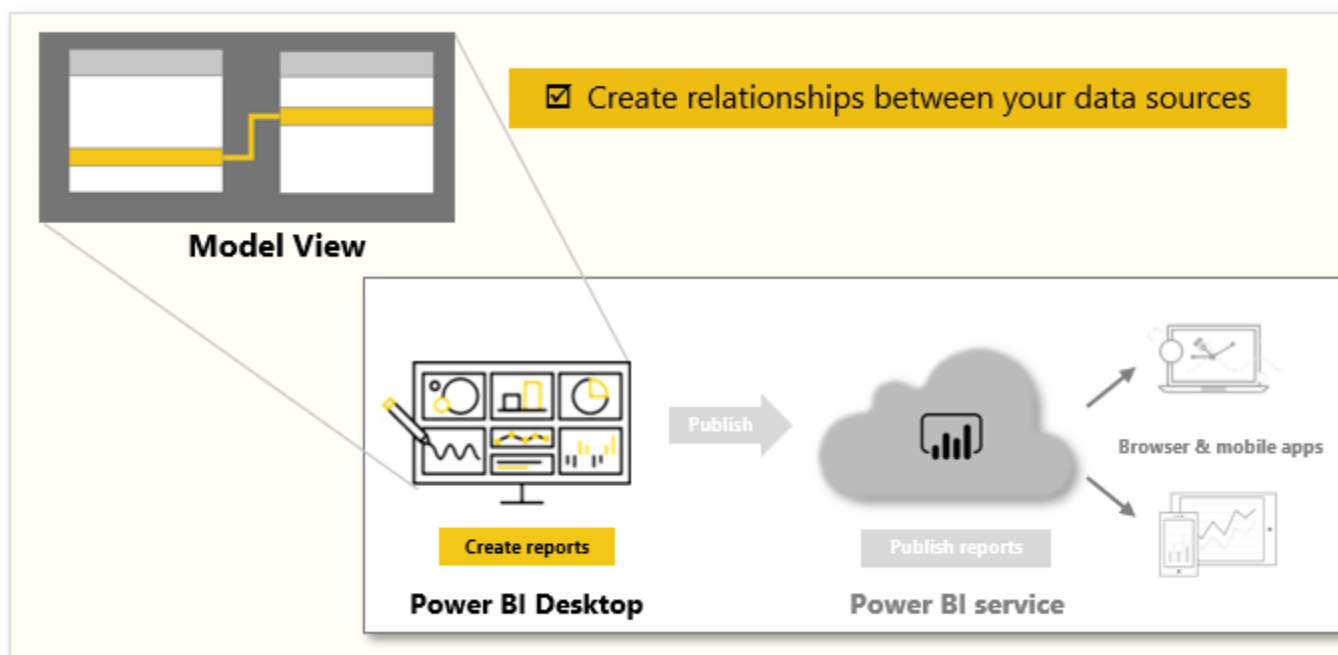
Create measures: calculations on your data that are performed realtime, ikke gemt i DB

Completed 100 XP, *måltal*

- 15 minutes

In Power BI, *measures* are defined calculations on your data that are performed at the time of your query. Measures are calculated as you interact with your reports and aren't stored in your database.

Tasks in this unit include:

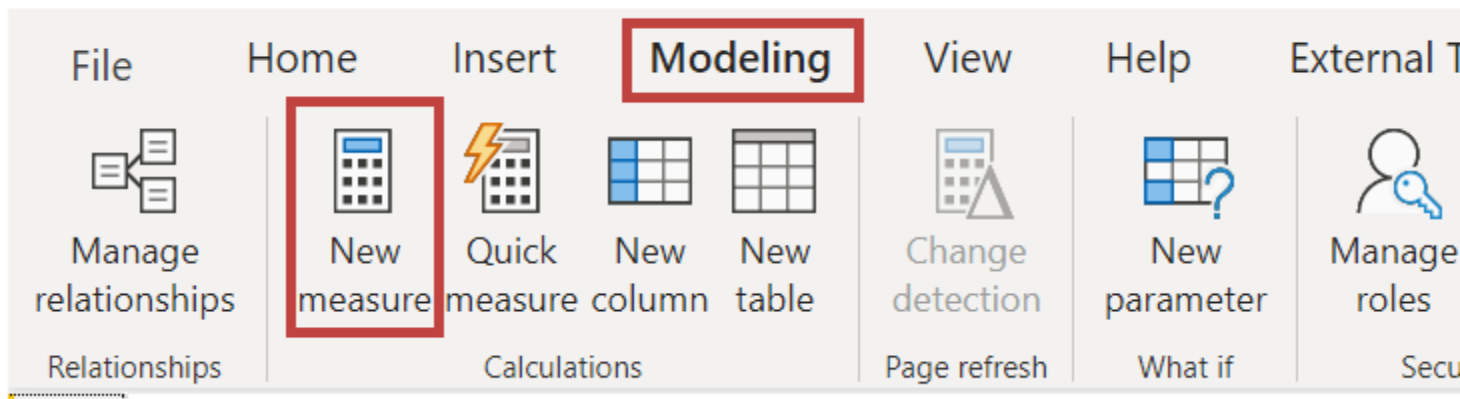


Note

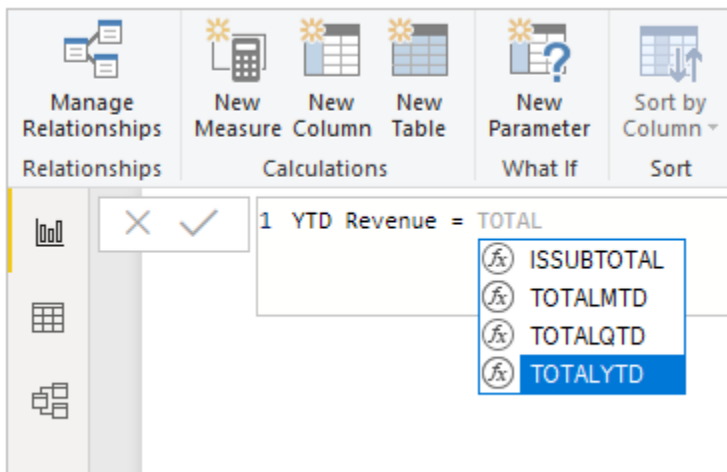
To follow along with the examples on this page, download the sample Access database [here](#) and import into Power BI Desktop (**Get Data > Database > Access database**). If you have any issues loading the Access database, please read this [article](#).

Create a measure

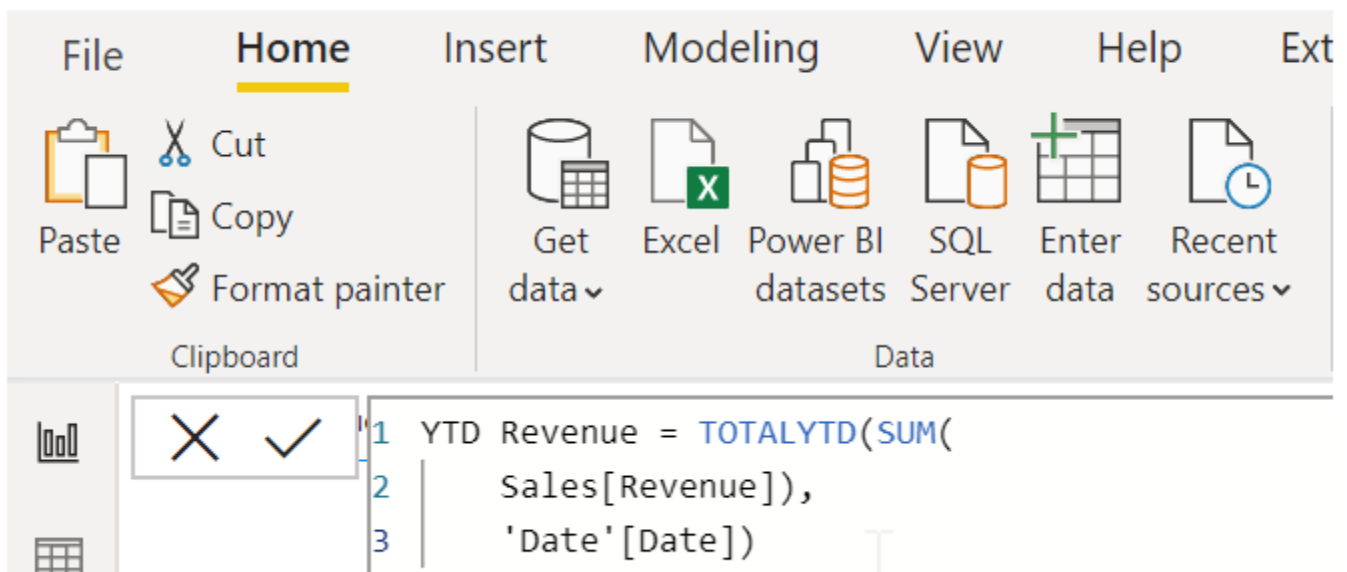
To create a measure, in **Report** view, med søjle-graph symbol, select **New Measure** from the **Modeling, Modelling** tab.



From the Formula bar, you can enter the DAX expression that defines your measure. As you enter your calculation, Power BI suggests relevant DAX functions and data fields. You'll also receive a tooltip that explains some of the syntax and function parameters.



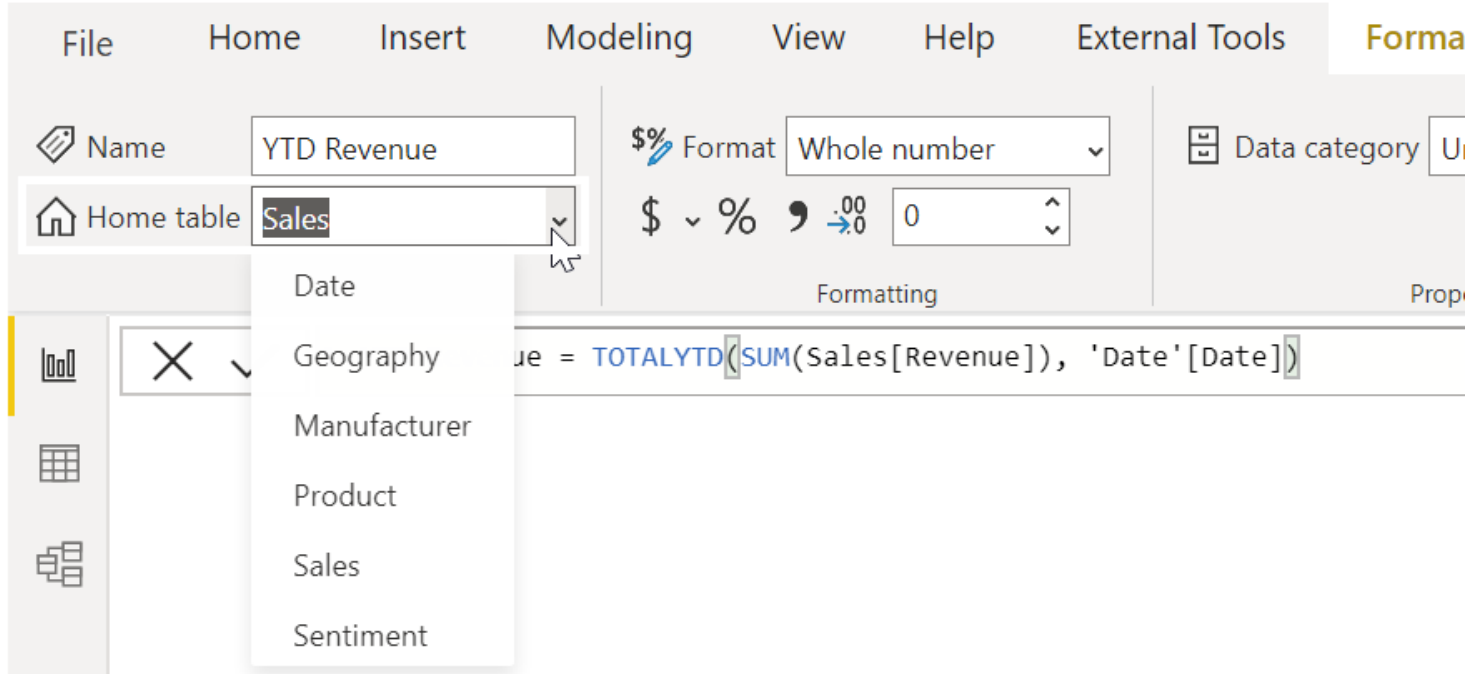
To ensure the new measure works make sure you have set the **Date** field in the **Date** table to type **Date**. If your calculation is long, you can add extra line breaks in the Expression Editor by typing **ALT-Enter**.



for hvert år, opsummeres værdi fra årets start til dato i visualen

Apply a measure

After you've created a new measure, it will appear in one of the tables on the Fields pane, which is found on the right side of the screen. Power BI inserts the new measure into whichever table you have currently selected. **While it doesn't matter, exactly, where the measure is located in your data, you can easily move it by selecting the measure and using the **Home Table** drop-down menu on the **Measure tools** tab.**



You can use a measure like any other table column: just drag and drop it onto the report canvas or visualization fields. **Measures also integrate seamlessly with slicers, segmenting your data on the fly, which means that you can define a measure once and then use it in many different visualizations.**

For more information, see [Tutorial: Create your own measures in Power BI Desktop](#).

*** Create your own measures in Power BI Desktop Tutorial

- Article
- 01/08/2023
- 6 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Automatic measures](#)
3. [Create and use your own measures](#)
4. [What you've learned](#)
5. [Next steps](#)

By using measures, you can create some of the most powerful data analysis solutions in Power BI Desktop. Measures help you by performing calculations on your data as you interact with your reports. This tutorial will guide you through understanding measures and creating your own basic measures in Power BI Desktop.

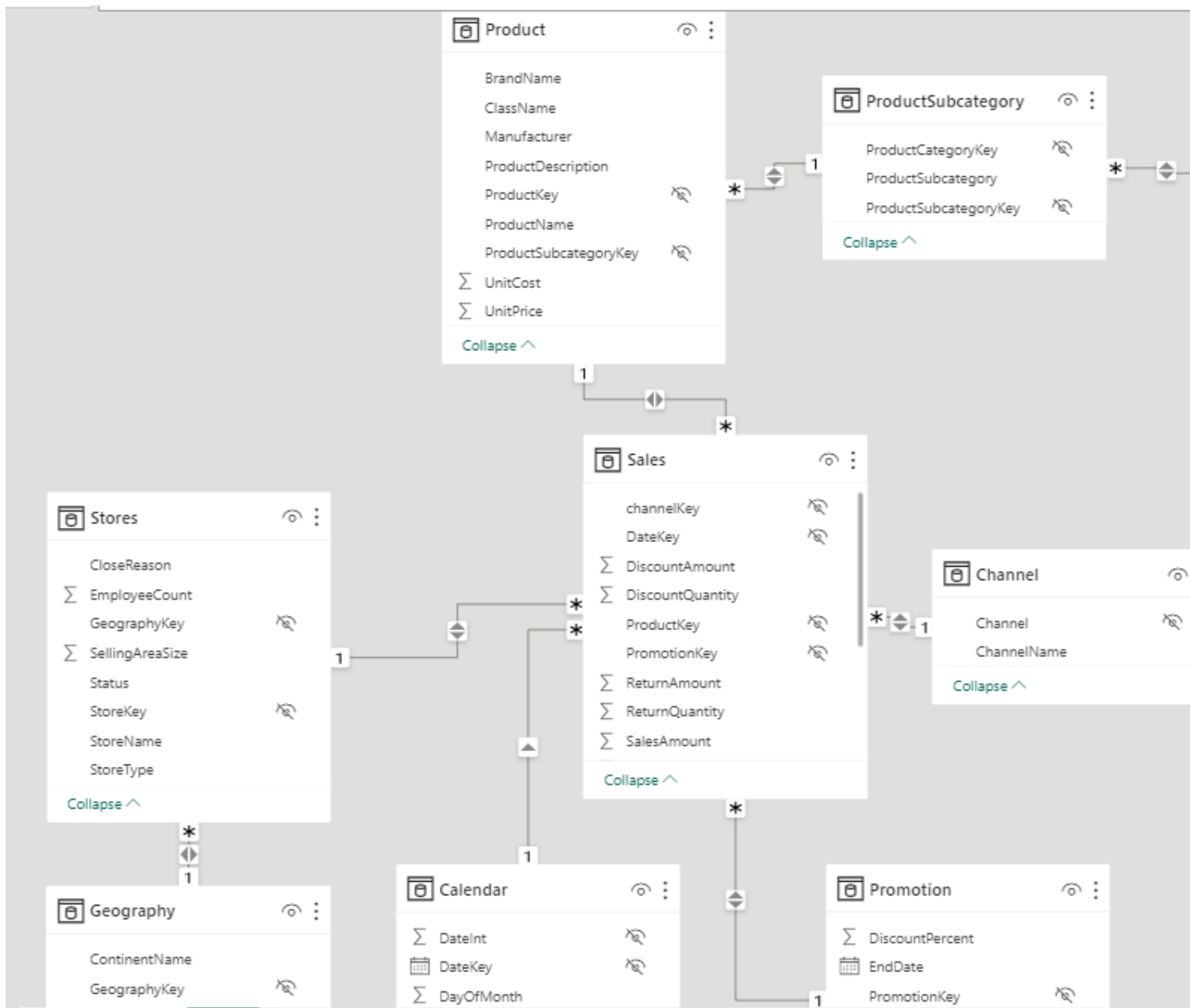
Prerequisites

- This tutorial is intended for Power BI users already familiar with using Power BI Desktop to create more advanced models. You should already be familiar with using Get Data and Power Query Editor to import data, work with multiple related tables, and add fields to the report canvas. If you're new to Power BI Desktop, be sure to check out [Get Started with Power BI Desktop](#).
- This tutorial uses the [Contoso Sales Sample for Power BI Desktop](#) file, which includes online sales data from the fictitious company, Contoso. Because this data is imported from a database, you can't connect to the datasource or view it in Power Query Editor. Download and extract the file on your computer.

Traditionel start-DB-model i Contoso Sales Sample

Sales indeholder del felter om hvert slag, men dertil keys over mod Dimensions-tabeller

X ✓		Sales							
Key	PromotionKey	UnitCost	UnitPrice	SalesQuantity	ReturnQuantity	ReturnAmount	DiscountQuantity	DiscountAmount	
375	1	\$321,44	\$699	10	0	\$0	0	\$0	
375	1	\$321,44	\$699	10	0	\$0	0	\$0	
375	1	\$321,44	\$699	10	0	\$0	0	\$0	
375	1	\$321,44	\$699	10	0	\$0	0	\$0	



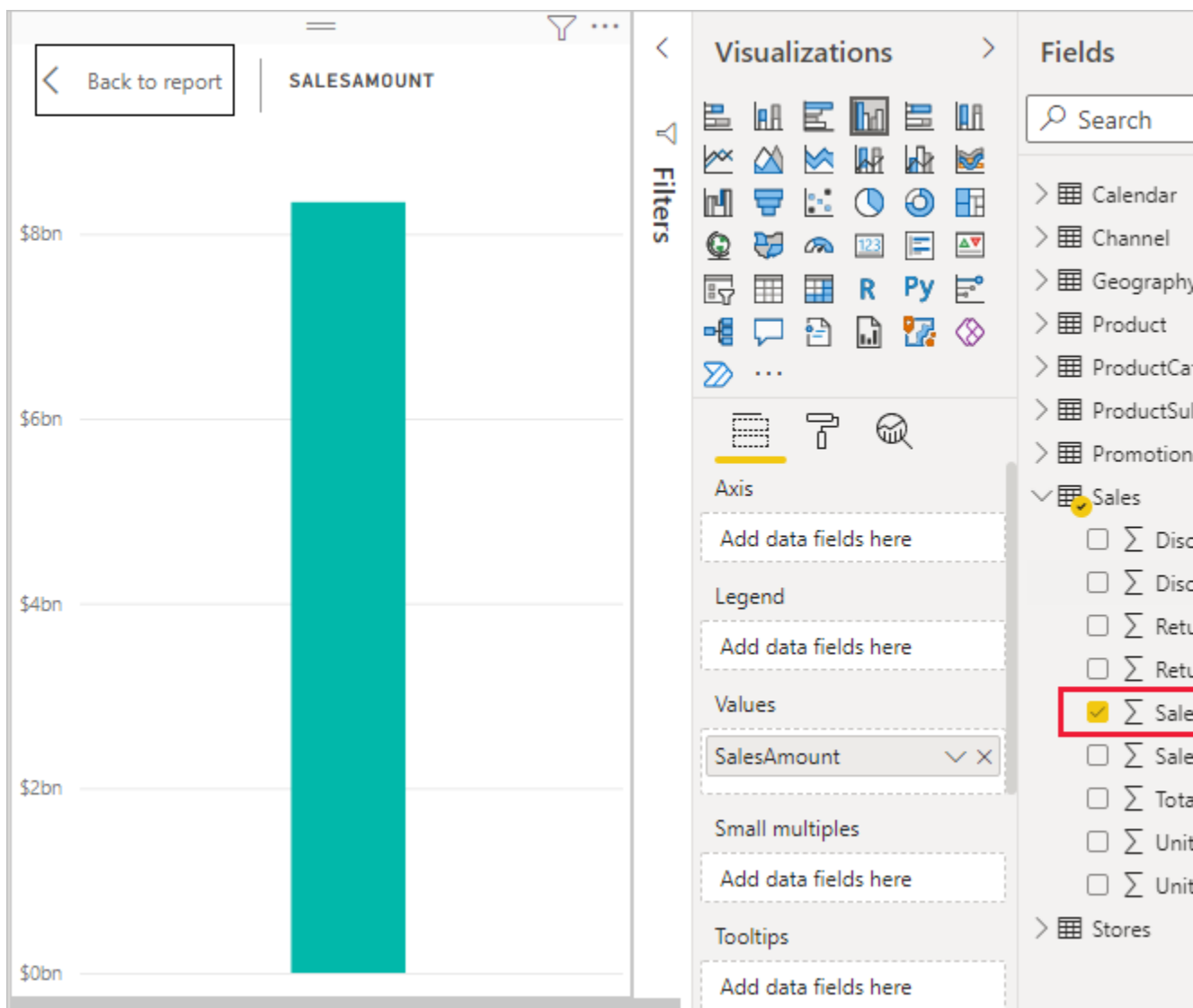
Automatic measures created by the system, vist med Σ i *Data*-tabel-oversigten

Alle numeriske værdier får automatisk tilknyttet en measure, som laver aggregering over konteksten

When Power BI Desktop creates a measure, it's most often created for you automatically. To see how Power BI Desktop creates a measure, follow these steps:

1. In Power BI Desktop, select **File > Open**, browse to the *Contoso Sales Sample for Power BI Desktop.pbix* file, and then choose **Open**.
2. In the **Fields, Data(dk)** pane, expand the **Sales** table. Then, either select the check box next to the **SalesAmount** field or drag **SalesAmount** onto the report canvas (med søjle-diagram-knap).

A new column chart visualization appears, showing the sum total of all values in the **SalesAmount** column of the **Sales** table.



bn for billion, angivet som mia i min dk udgave, 8.1 mia, passer med figuren

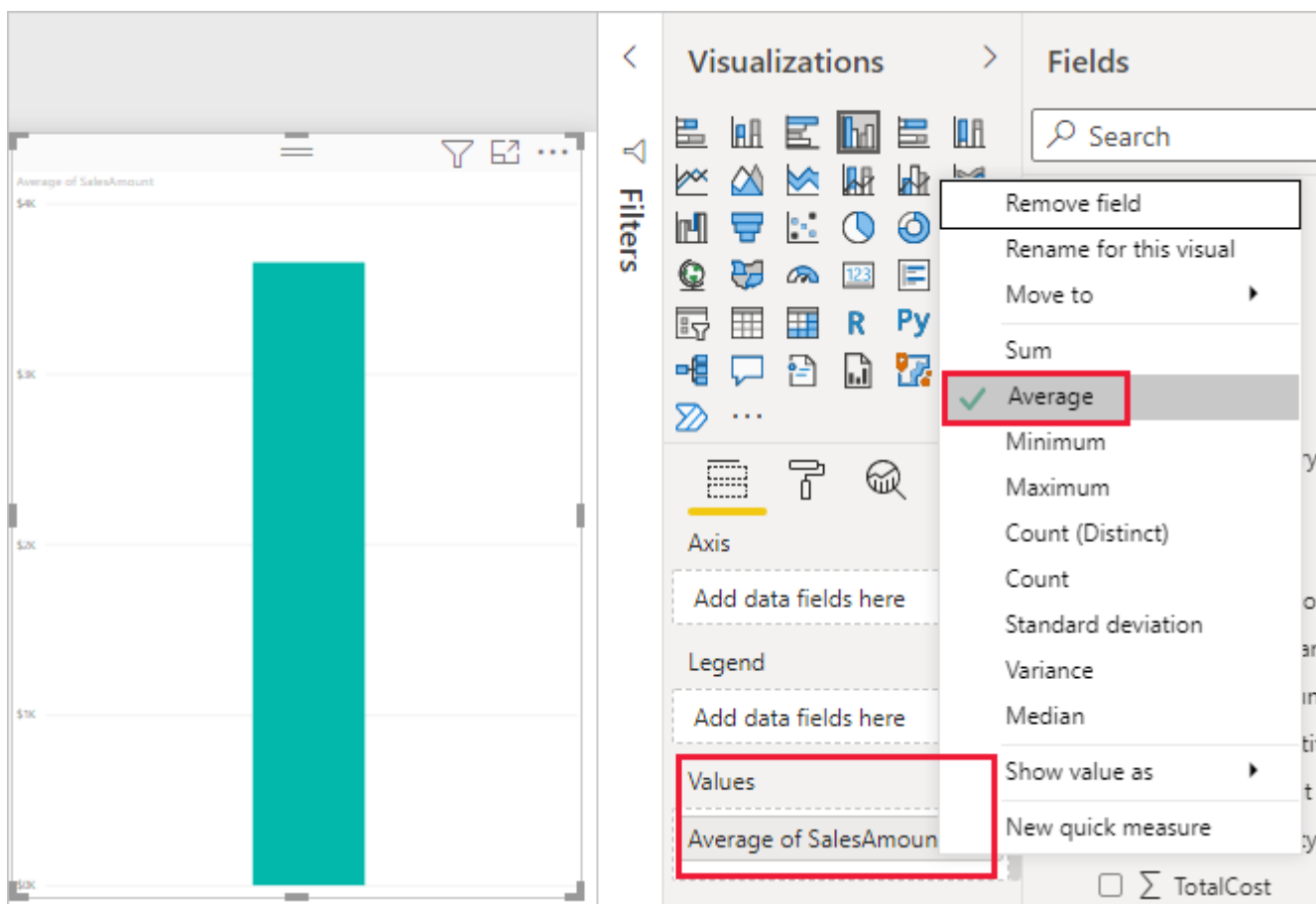
Any field (column) in the **Fields** pane with a sigma icon Σ is **numeric** =et measure, beregnes realtime, and its values can be aggregated. Rather than display a table with many values (2,000,000 rows for **SalesAmount**), Power BI Desktop automatically **creates and calculates a measure to aggregate the data if it detects a numeric datatype**. Sum is the default aggregation for a numeric datatype, but you can easily apply different aggregations like average or count. Understanding aggregations is fundamental to understanding measures, **because every measure performs some type of aggregation**.

Vælges tidsaktie som x, vil aggregering ske over x-værdierne, fx år; aggregering sker i kontekst!

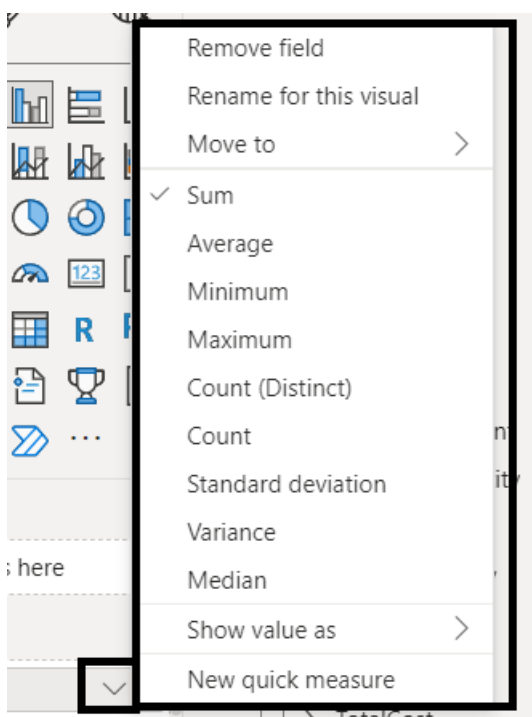
To change the chart aggregation, follow these steps:

1. Select the **SalesAmount** visualization in the report canvas.
2. In the **Values** area of the **Visualizations** pane, select the down arrow to the right of **SalesAmount**.
3. From the menu that appears, select **Average**.

The visualization changes to an average of all sales values in the **SalesAmount** field.

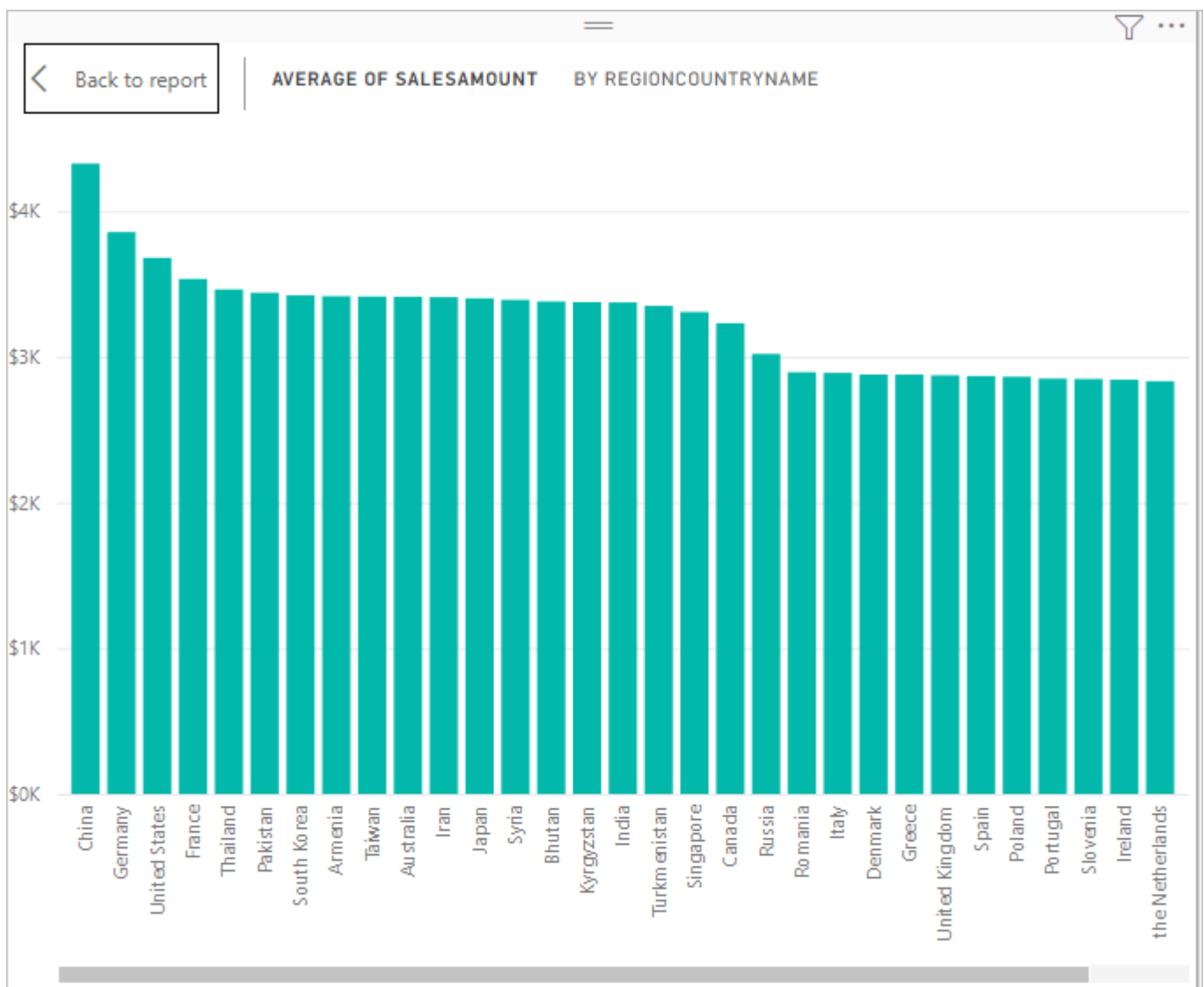


Man kan ændre til gennemsnit, øverst ses kolonne-værktøjer hvor man kan se at opsummering bygger på sum, vælges sum vises netop SalesAmount, som jo er lig 'sum' anvendt.



Depending on the result you want, you can change the type of aggregation. However, not all types of aggregation apply to every numeric datatype. For example, for the **SalesAmount** field, Sum and Average are useful, and Minimum and Maximum have their place as well. However, Count doesn't make sense for the **SalesAmount** field, because while its values are numeric, they're really currency.

Values calculated from measures change in response to your interactions with your report. For example, if you drag the **RegionCountryName** field from the **Geography** table onto your existing **SalesAmount** chart, it changes to show the average sales amounts for each country/region.



United States mangler på figuren

When the result of a measure changes because of an interaction with your report, you've affected your measure's *context*. Every time you interact with your report visualizations, you're changing the context in which a measure calculates and displays its results.

Create and use your own measures

In most cases, Power BI Desktop automatically calculates and returns values according to the types of fields and aggregations you choose. However, in some cases **you might want to create your own measures to perform more complex, unique calculations**. With Power BI Desktop, you can create your own measures with the

Data Analysis Expressions (DAX) formula language.

DAX formulas use many of the same functions, operators, and syntax as Excel formulas. However, DAX functions are designed to work with relational data and perform more dynamic calculations as you interact with your reports. There are over 200 DAX functions that do everything from simple aggregations like sum

and average to more complex statistical and filtering functions. There are many resources to help you learn more about DAX. After you've finished this tutorial, see [DAX basics in Power BI Desktop](#).

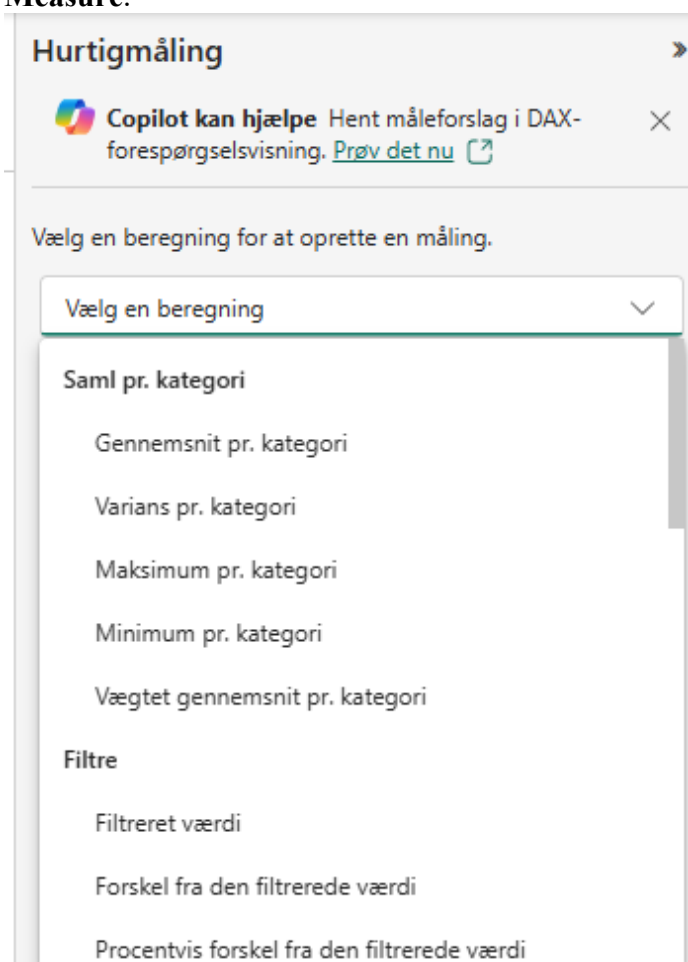
When you create your own measure, it's called a *model* measure, and it's added to the **Fields** list for the table you select. Some advantages of model measures are that you can name them whatever you want, making them more identifiable. You can use them as arguments in other DAX expressions, and you can make them perform complex calculations quickly.

Quick measures

Many common calculations are available as *quick measures*, which write the DAX formulas for you based on your inputs in a window. These quick, powerful calculations are also great for learning DAX or seeding your own customized measures.

Create a quick measure using one of these methods:

- From a table in the **Fields(dk/Data)** pane, right-click or select **More options (...)**, and then choose **New quick measure, Ny hurtigmåling** from the list.
- Under **Calculations** in the **Home** tab of the Power BI Desktop ribbon, select **New Quick Measure**.



For more information about creating and using quick measures, see [Use quick measures](#).

Create a measure

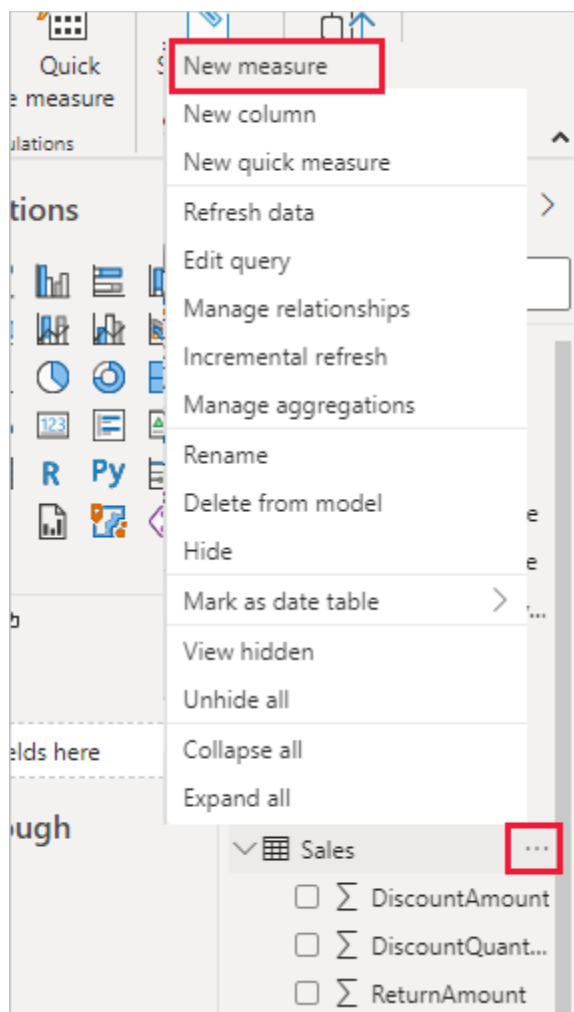
Suppose you want to analyze your **net sales by subtracting discounts and returns**(må være salg hvor kunden får penge tilbage) **from total sales amounts**. For the context that exists in your visualization, you need a

measure that subtracts the sum of DiscountAmount and ReturnAmount from the sum of SalesAmount. There's no field for Net Sales in the **Fields** list, but you have the building blocks to create your own measure to calculate net sales.

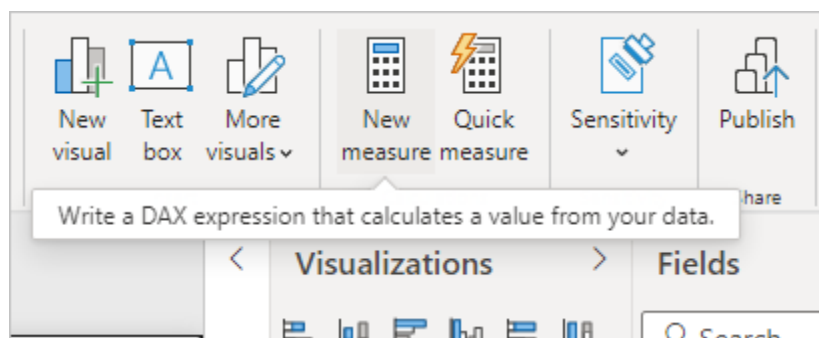
To create a measure, follow these steps:

1. In the **Fields** pane, right-click the **Sales** table, or hover over the table and select **More options (...)**.
2. From the menu that appears, choose **New measure**=en målt værdi, en *måling*

This action saves your new measure in the **Sales** table, where it's easy to find.



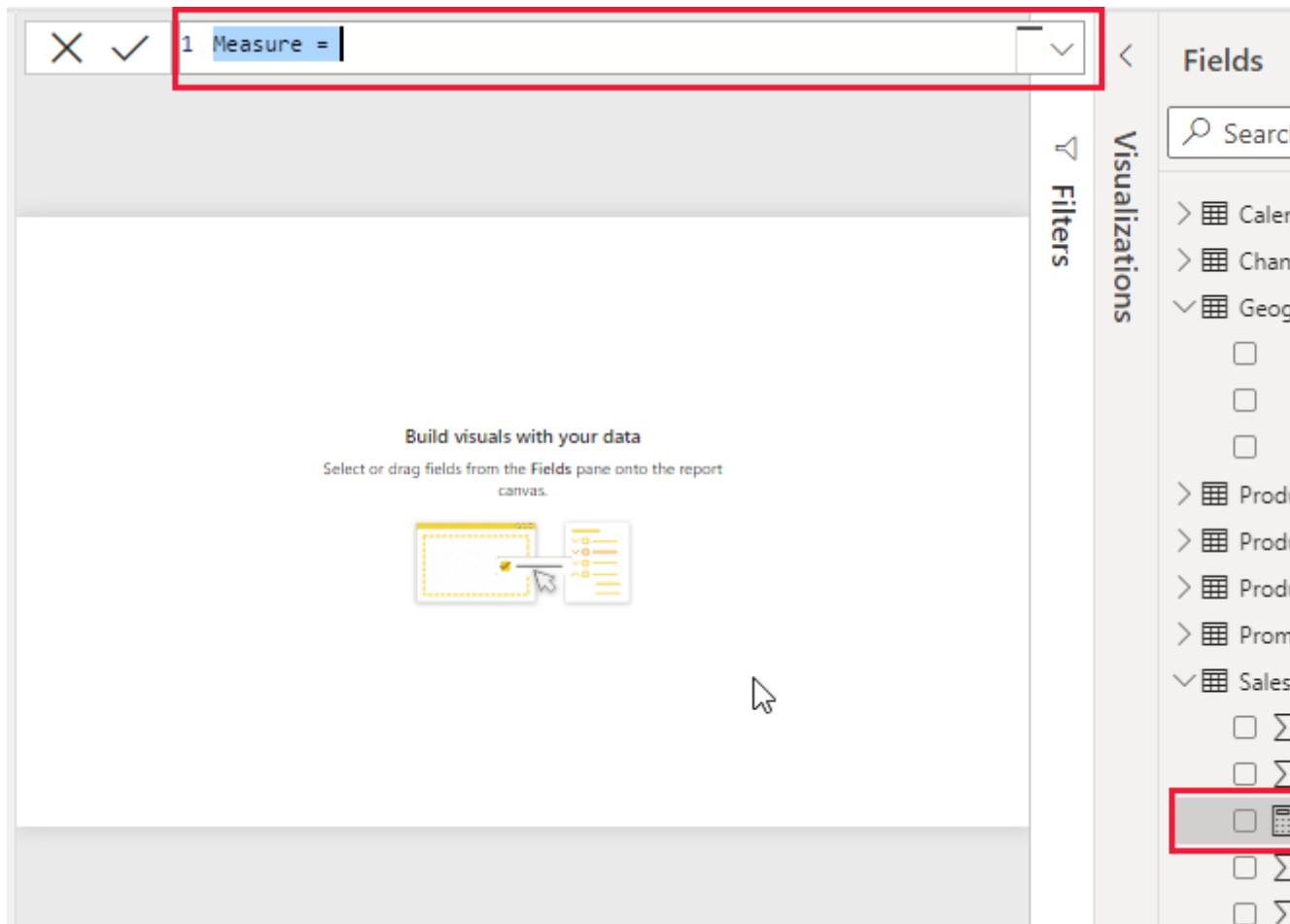
You can also create a new measure by selecting **New Measure** in the **Calculations** group on the **Home** tab of the Power BI Desktop ribbon.



Tip

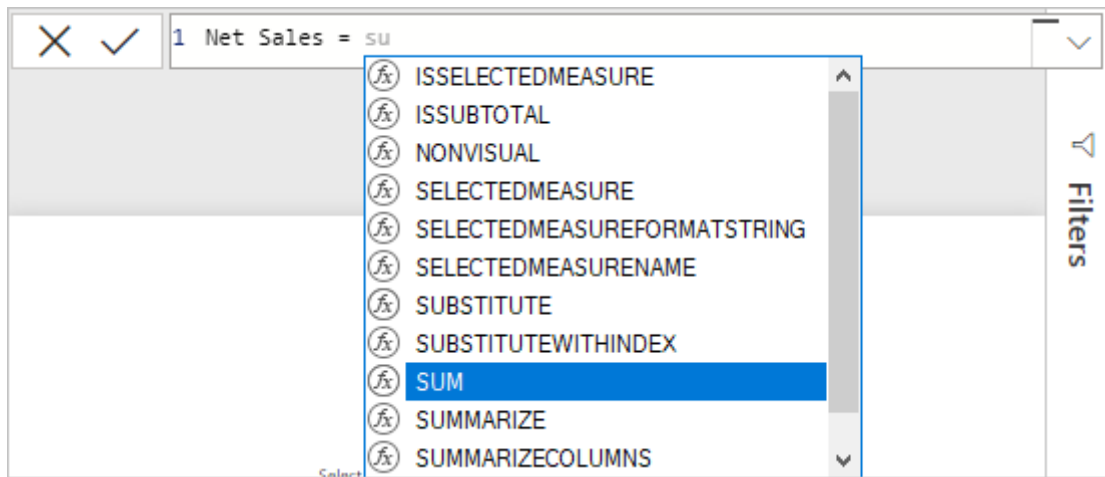
When you create a measure from the ribbon, you can create it in any of your tables, but it's easier to find if you create it where you plan to use it. In this case, select the **Sales** table first to make it active, and then choose **New measure**.

The formula bar appears along the top of the report canvas, where you can rename your measure and enter a DAX formula.

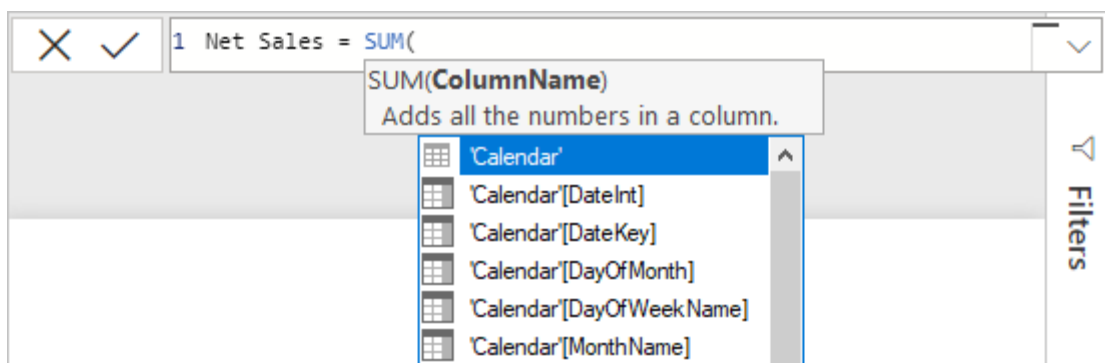


3. By default, each new measure is named *Measure*. If you don't rename it, new measures are named *Measure 2*, *Measure 3*, and so on. Because we want this measure to be more identifiable, highlight *Measure* in the formula bar, and then change it to *Net Sales*.
4. Begin entering your formula. After the equals sign, start to type *Sum*. As you type, a drop-down suggestion list appears, showing all the DAX functions, beginning with the letters you type. Scroll down, if necessary, to select **SUM** from the list, and then press **Enter**.

Net Sales=netto salg=salg-discount-returnValues, omkostninger er ikke trukket fra

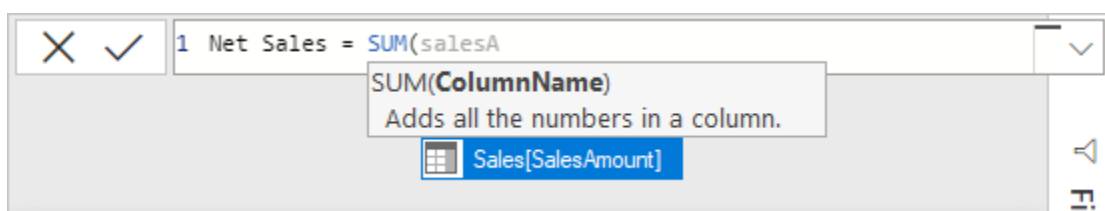


An opening parenthesis appears, along with a drop-down suggestion list of the available columns you can pass to the SUM function.



5. Expressions always appear between opening and closing parentheses. For this example, your expression contains a single argument to pass to the SUM function: the **SalesAmount** column. Begin typing *SalesAmount* until **Sales(SalesAmount)** is the only value left in the list.

The column name preceded by the table name is called the fully qualified name of the column. Fully qualified column names make your formulas easier to read.



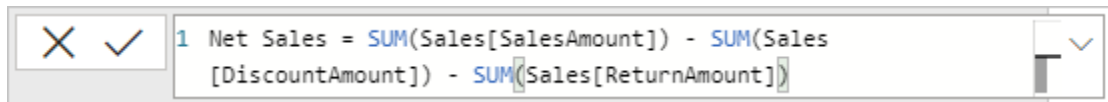
6. Select **Sales[SalesAmount]** from the list, and then enter a closing parenthesis.

Tip

Syntax errors are most often caused by a missing or misplaced closing parenthesis.

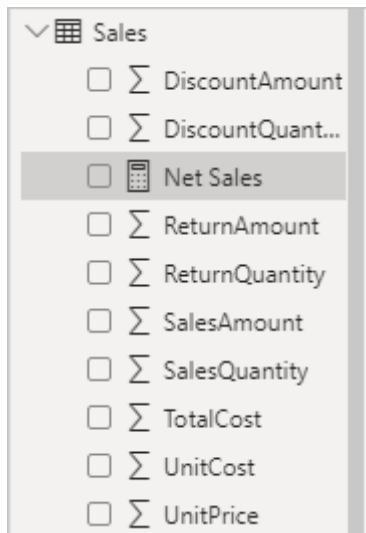
7. Subtract the other two columns inside the formula:
 - a. After the closing parenthesis for the first expression, type a space, a minus operator (-), and then another space.
 - b. Enter another SUM function, and start typing *DiscountAmount* until you can choose the **Sales[DiscountAmount]** column as the argument. Add a closing parenthesis.

c. Type a space, a minus operator, a space, another SUM function with **Sales[ReturnAmount]** as the argument, and then a closing parenthesis.



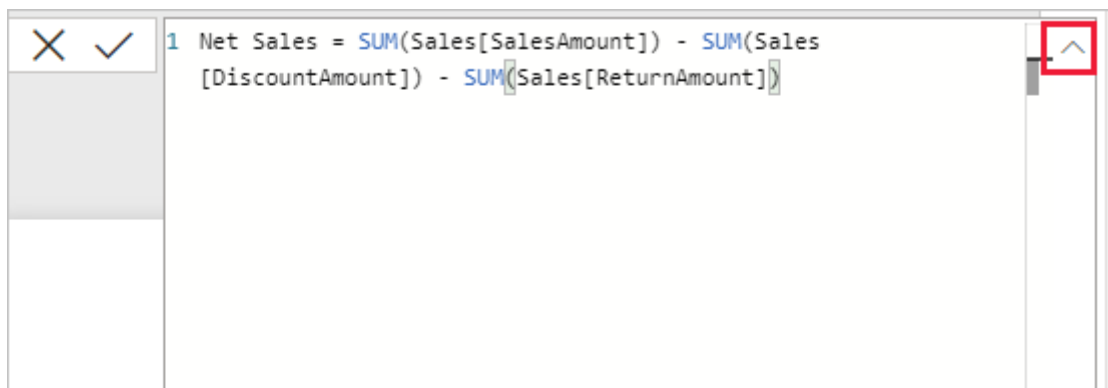
8. Press **Enter** or select **Commit** (checkmark icon) in the formula bar to complete and validate the formula.

The validated **Net Sales** measure is now ready to use in the **Sales** table in the **Fields** pane.

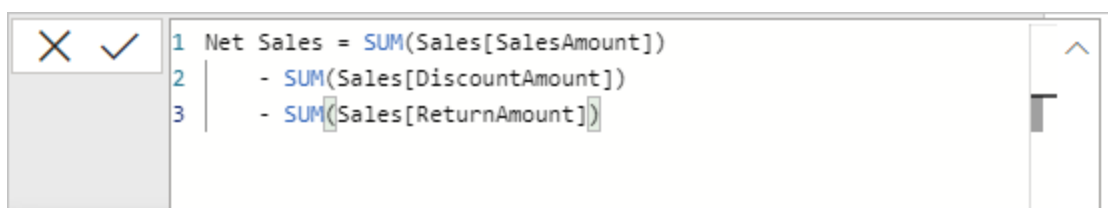


9. If you run out of room for entering a formula or want it on separate lines, select the down arrow on the right side of the formula bar to provide more space.

The down arrow turns into an up arrow and a large box appears.



10. Separate parts of your formula by pressing **Alt + Enter** for separate lines, or pressing **Tab** to add tab spacing.

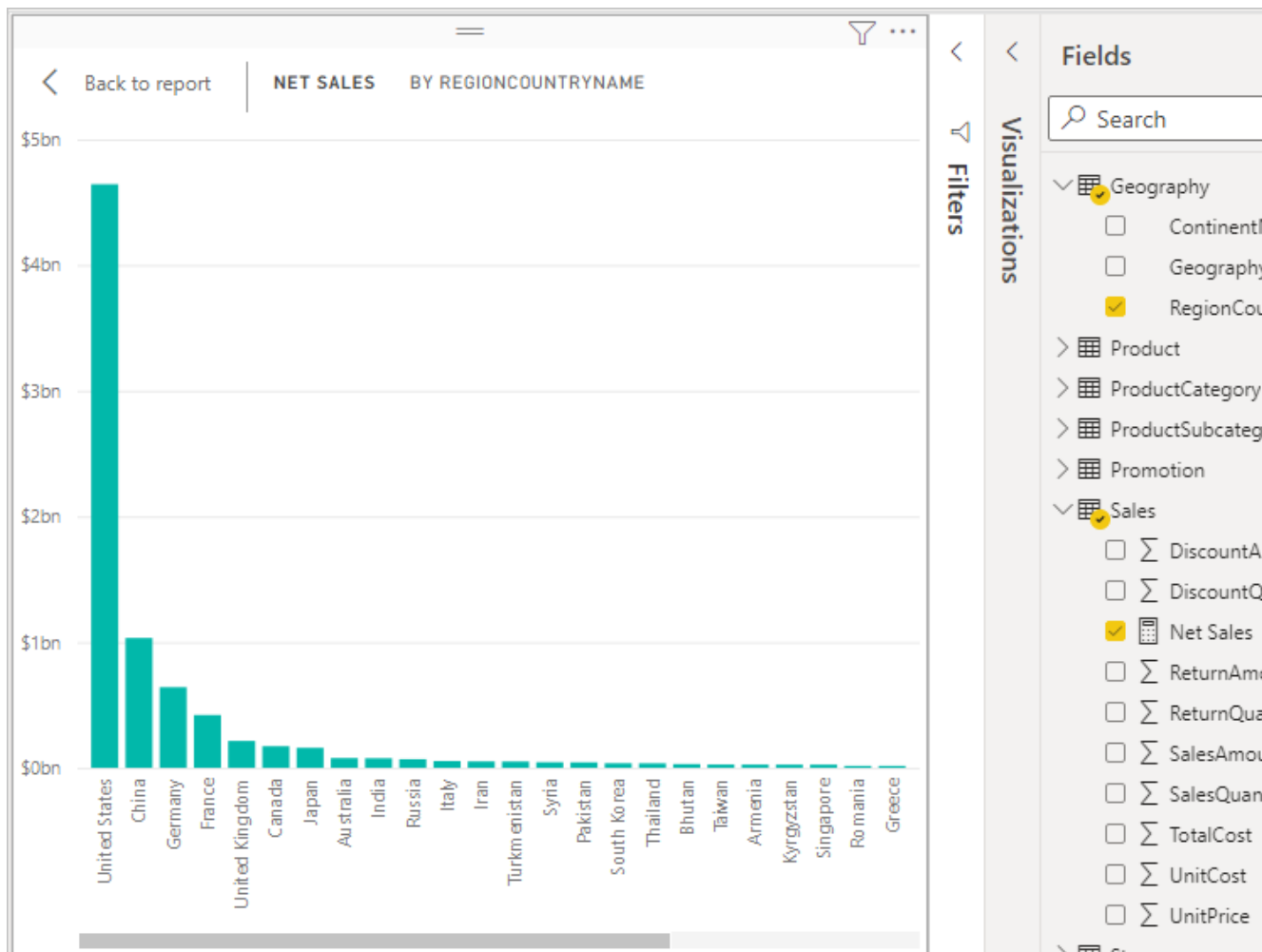


Use your measure in the report

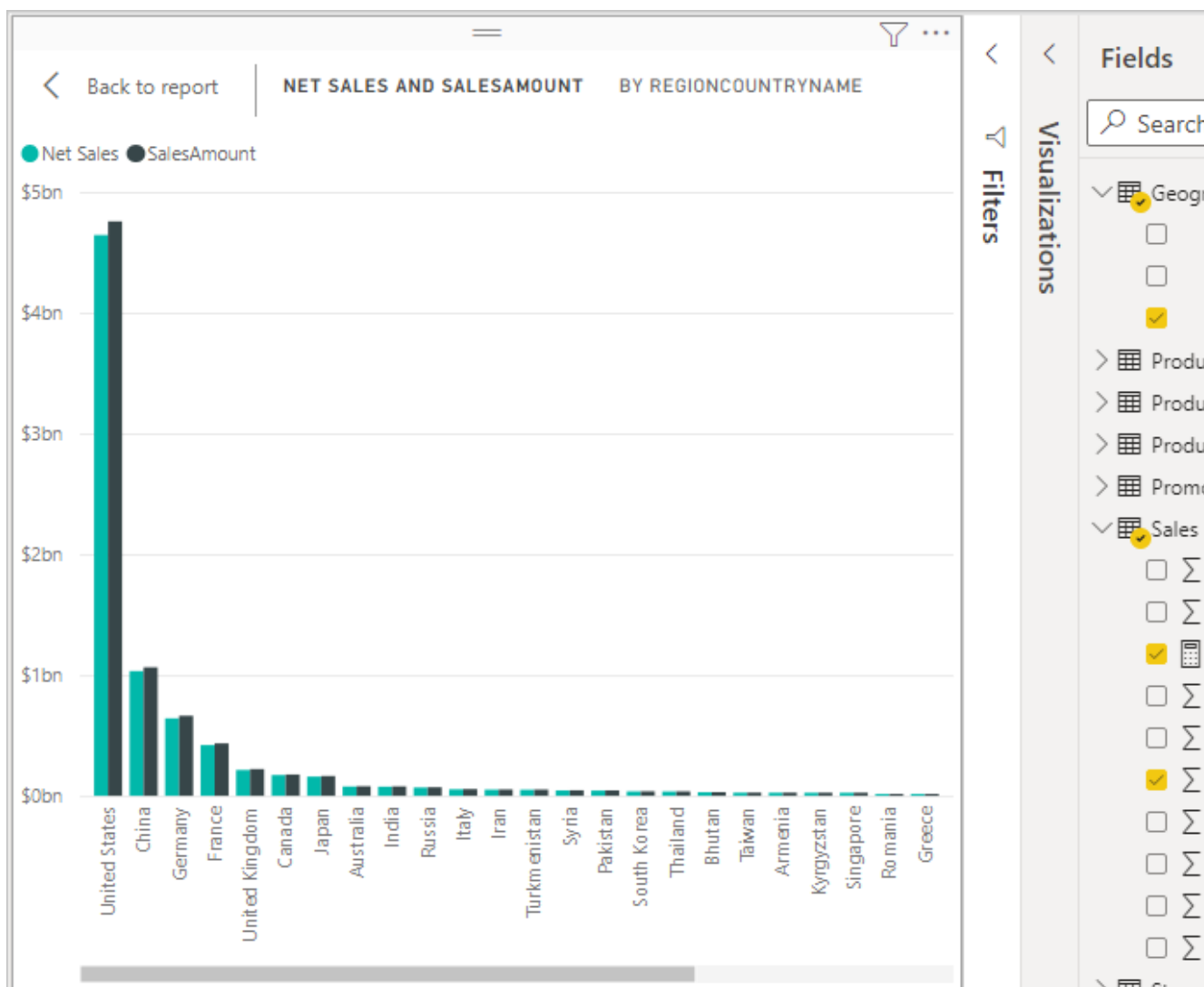
Add your new **Net Sales** measure to the report canvas, and calculate net sales for whatever other fields you add to the report.

To look at net sales by country/region:

1. Select the **Net Sales** measure from the **Sales** table, or drag it onto the report canvas.
2. Select the **RegionCountryName** field from the **Geography** table, or drag it onto the **Net Sales** chart.



3. To see the difference between net sales and total sales by country/region, select the **SalesAmount** field or drag it onto the chart.



The chart now uses two measures: **SalesAmount**, which Power BI summed automatically, and the **Net Sales** measure, which you manually created. Each measure was calculated in the context of another field, **RegionCountryName**.

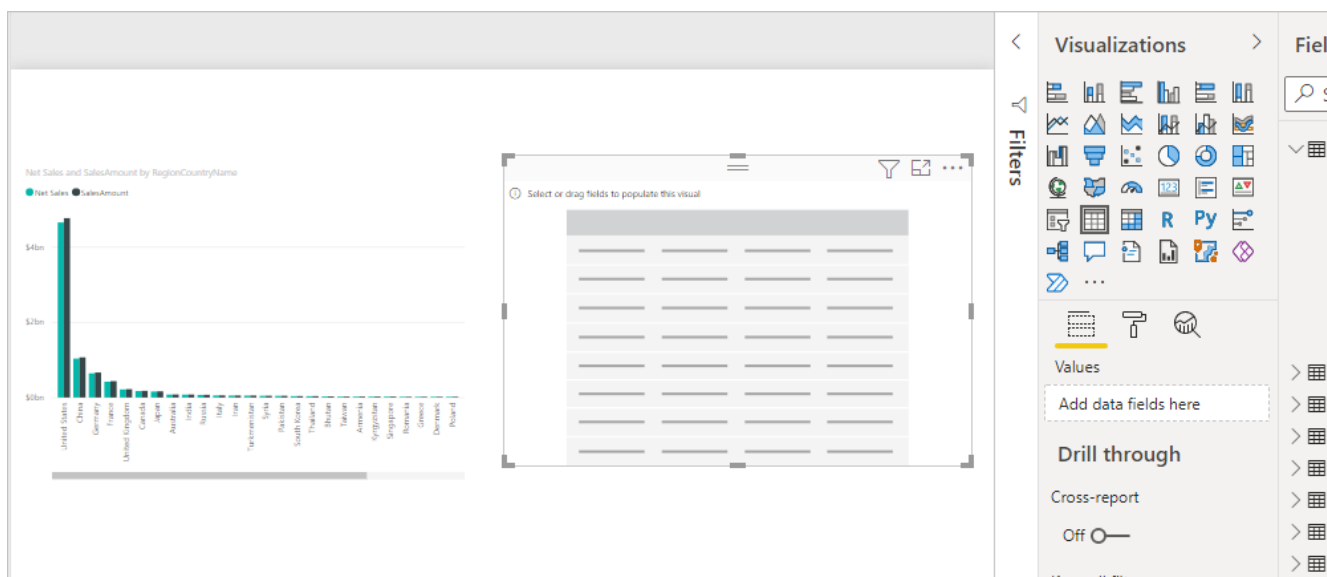
SalesCnt = COUNTROWS(Sales) antal rækker i konteksten

Use your measure with a slicer, *Udsnit*, som filtrer udtrækket

Add a slicer to further filter net sales and sales amounts by calendar year:

1. Select a blank area next to the chart. In the **Visualizations** pane, select the **Table** visualization.

This action creates a blank table visualization on the report canvas.

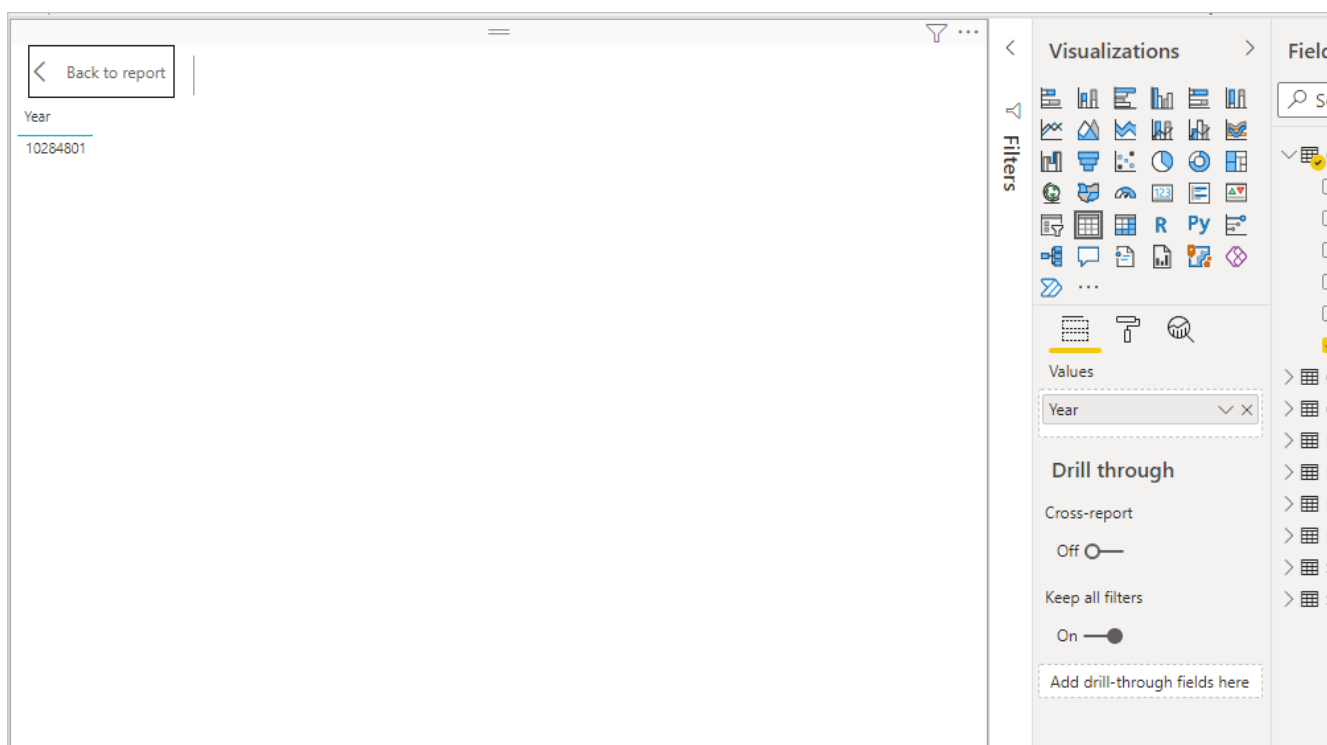


2. Drag the **Year** field from the **Calendar** table onto the new blank table visualization.

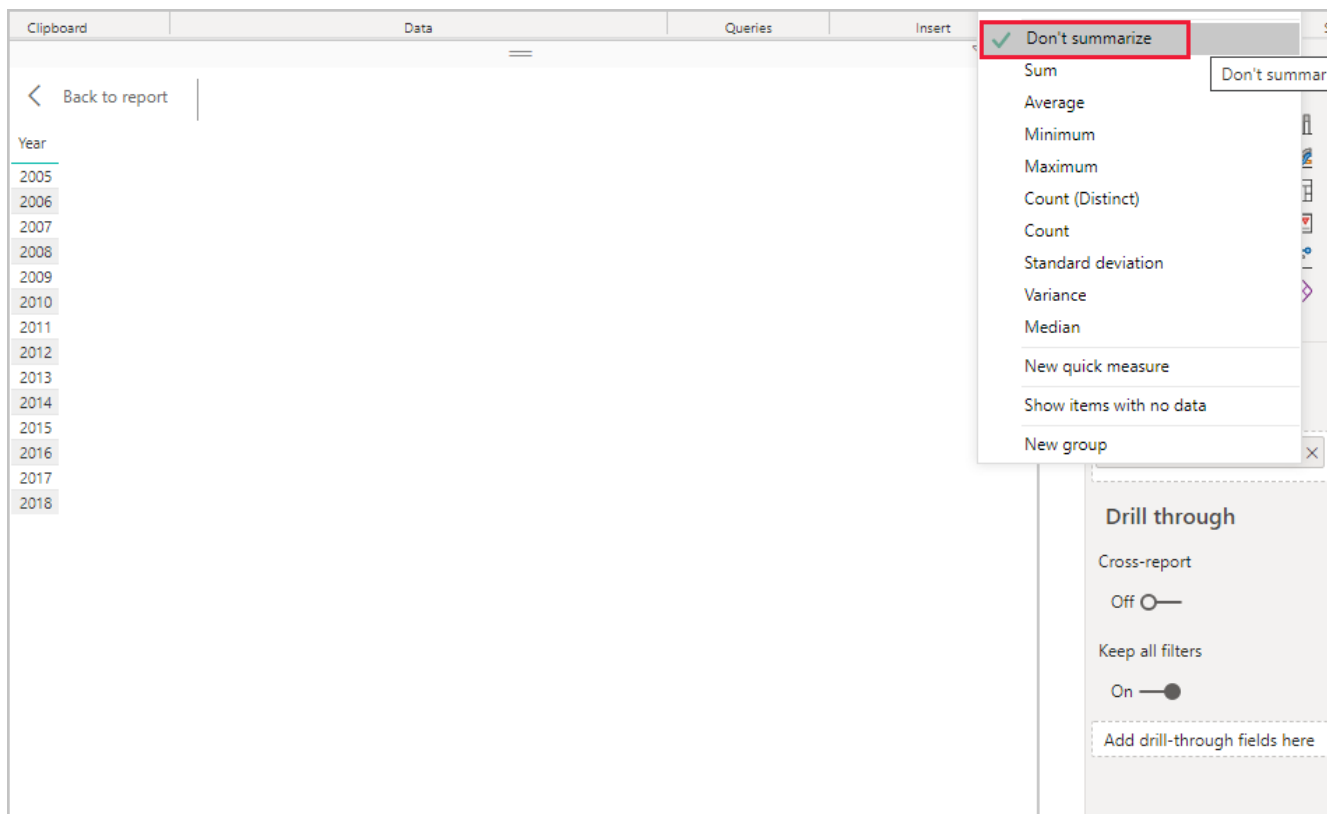
Alle numeriske værdier får automatisk tilknyttet en measure, som laver aggregering over konteksten

Because **Year** is a numeric field, Power BI Desktop sums up its values. This summation doesn't work well as an aggregation; we'll address that in the next step.

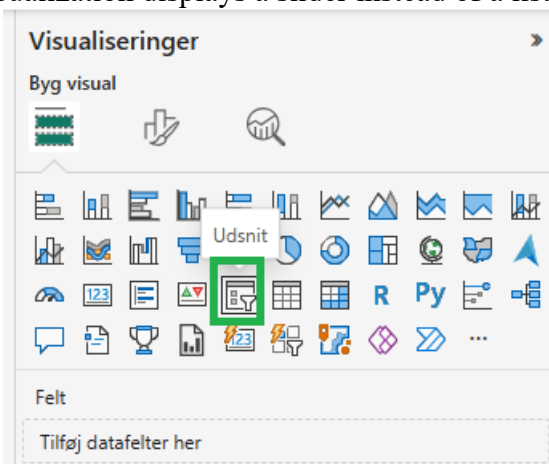
Date har 5113 rows, en per dag, svarende til 14 år, sum af years ca: $5113 \times 2010 = 10'277'130$, marker felt i tabel, så højreklik kopi: 10'284'801, passer ok!



3. In the **Values** box in the **Visualizations** pane, select the down arrow next to Columns/Year, and then choose **Don't summarize** from the list. The table now lists individual years.



4. Select the **Slicer/Filtre** icon, *Udsnit*, in the **Visualizations** pane, se næste figure, to convert the table to a slicer. If the visualization displays a slider instead of a list, choose **List** from the

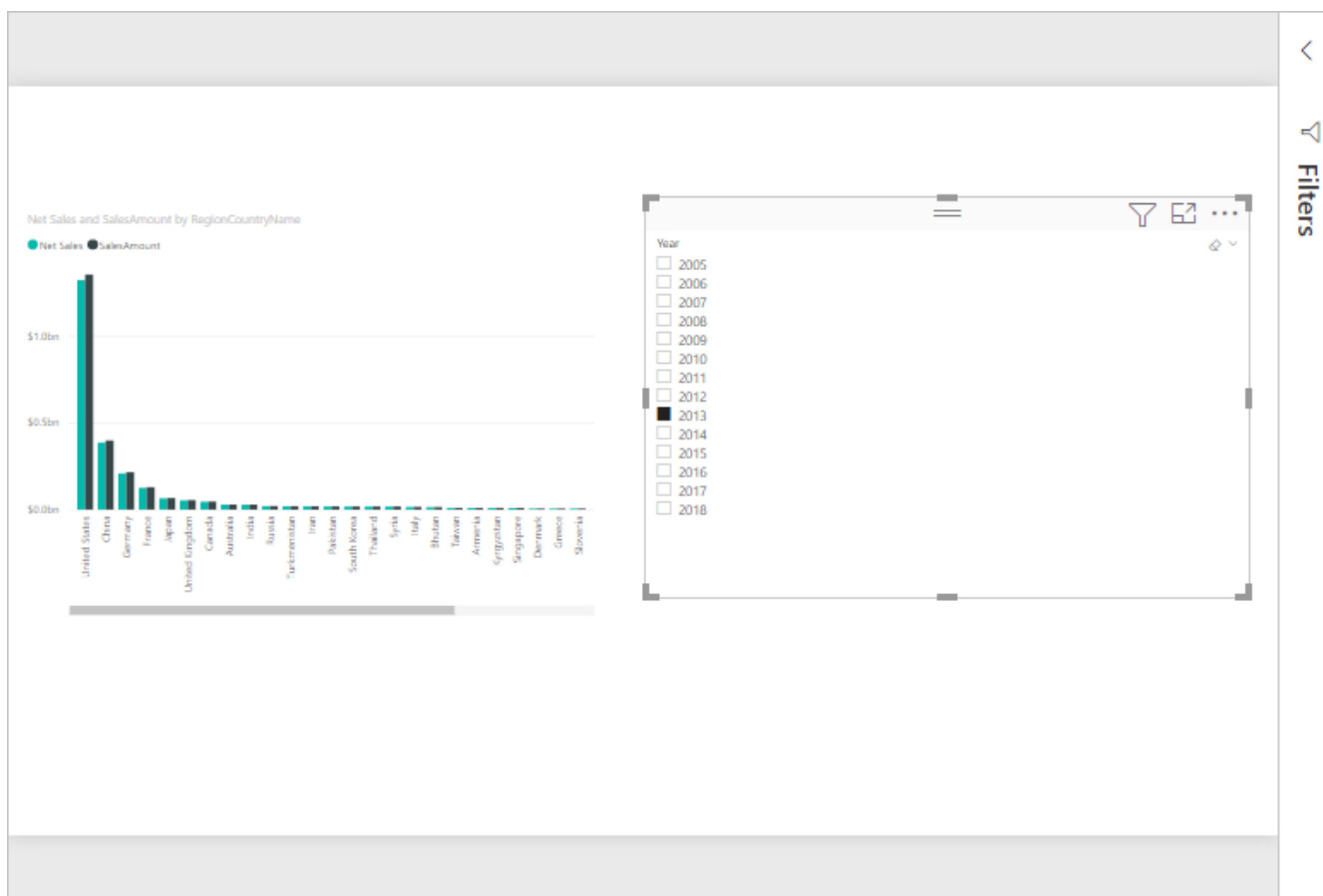


down arrow in the slider.
denne visual

trækker så Year-felt over I

5.

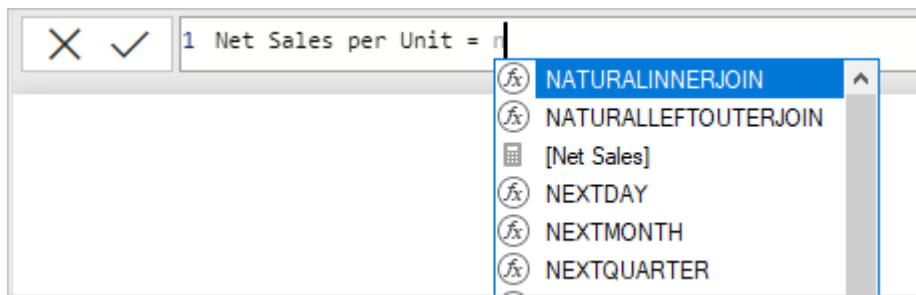
6. Select any value in the **Year** slicer to filter the **Net Sales and Sales Amount by RegionCountryName** chart accordingly. The **Net Sales** and **SalesAmount** measures recalculate and display results in the context of the selected **Year** field.



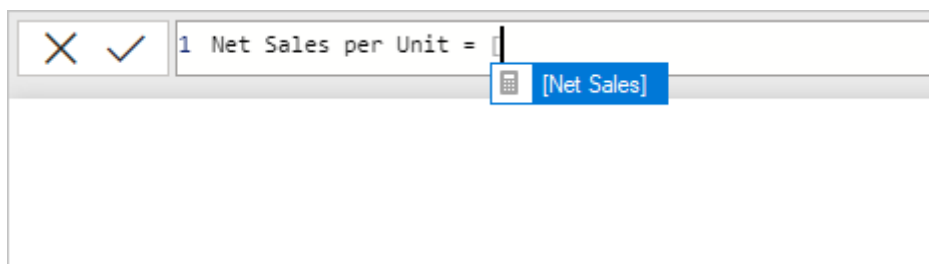
Use your measure in another measure

Suppose you want to find out which products have the highest net sales amount per unit sold. You'll need a measure that divides net sales by the quantity of units sold. Create a new measure that divides the result of your **Net Sales** measure by the sum of **Sales[SalesQuantity]**.

1. In the **Fields** pane, create a new measure named **Net Sales per Unit** in the **Sales** table.
2. In the formula bar, begin typing *Net Sales*. The suggestion list shows what you can add. Select **[Net Sales]**.



3. You can also reference measures by just typing an opening bracket ([). The suggestion list shows only measures to add to your formula.



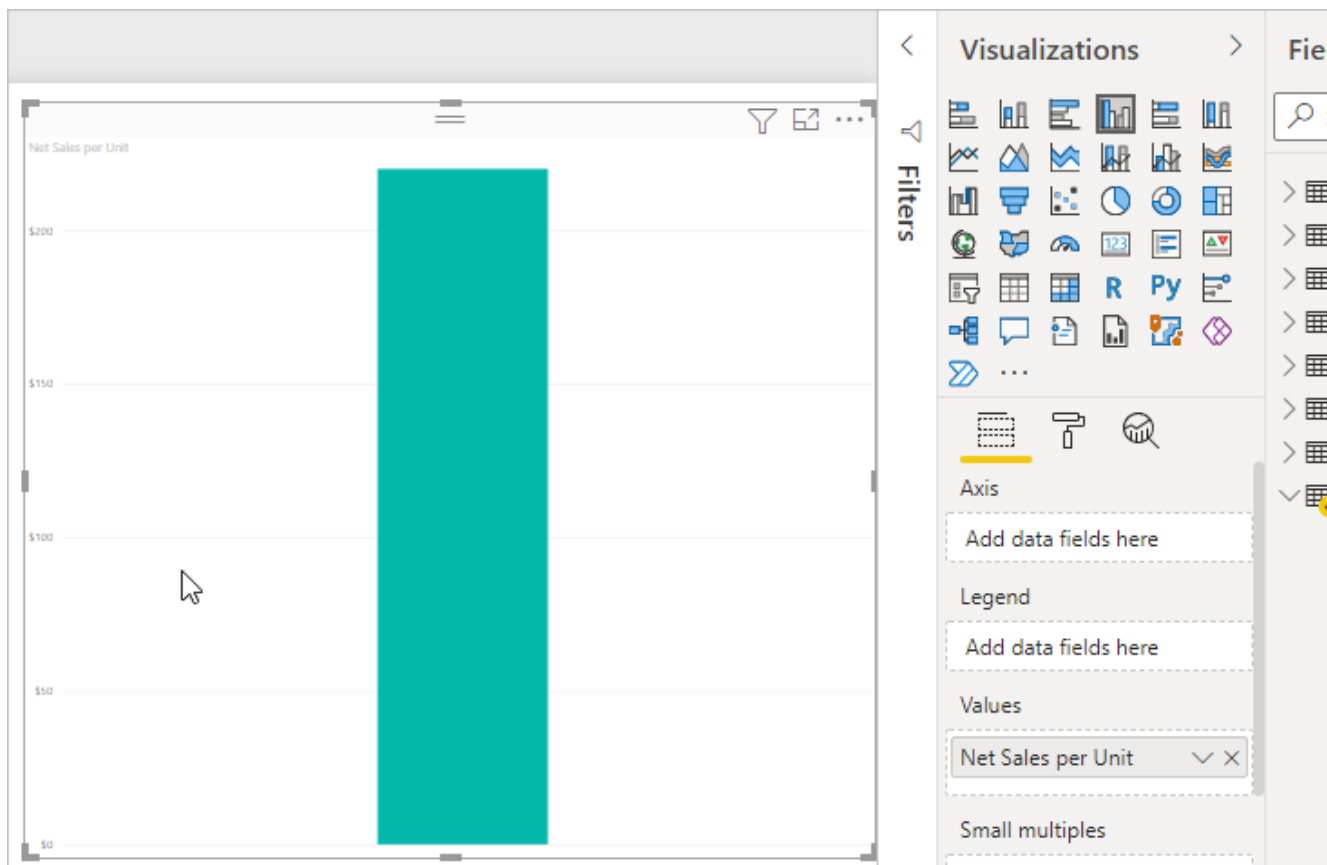
4. Enter a space, a divide operator (/), another space, a SUM function, and then type *Quantity*. The suggestion list shows all the columns with *Quantity* in the name. Select **Sales[SalesQuantity]**, type the closing parenthesis, and press **ENTER** or choose **Commit** (checkmark icon) to validate your formula.

The resulting formula should appear as:

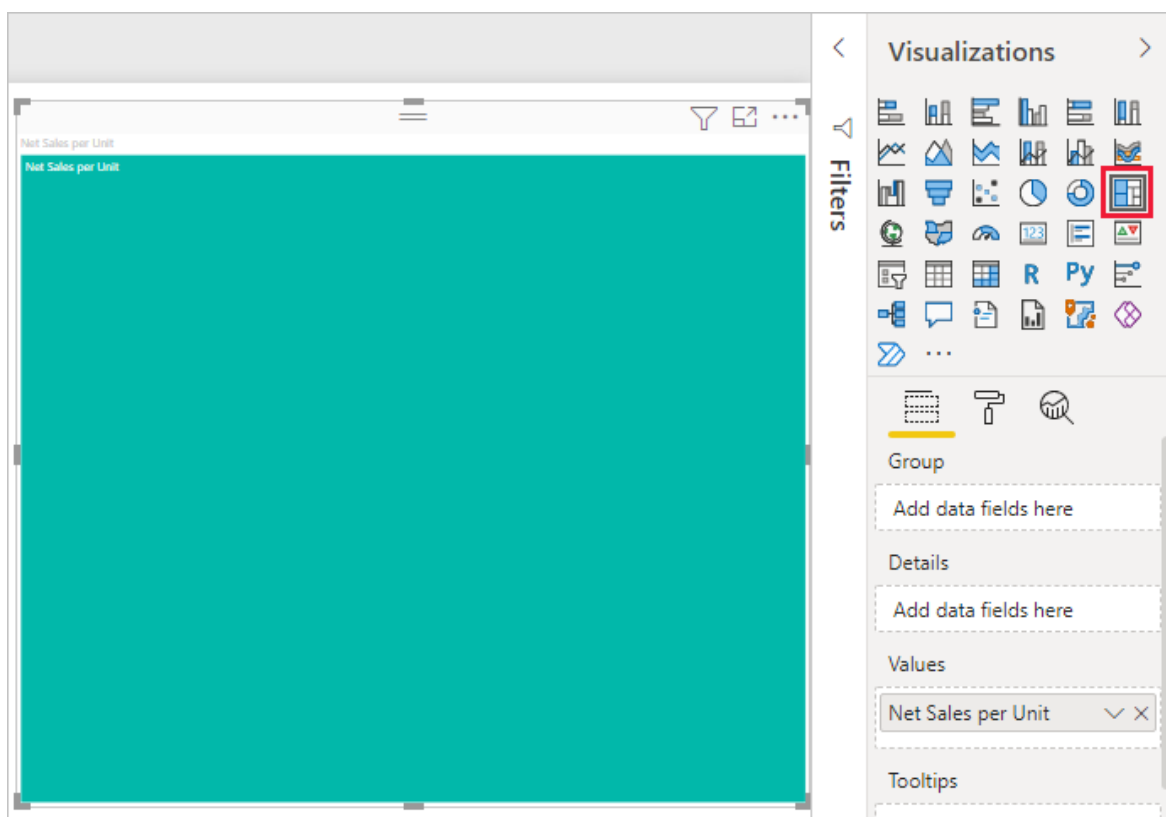
Net Sales per Unit = [Net Sales] / SUM(Sales[SalesQuantity])

5. Select the **Net Sales per Unit** measure from the **Sales** table, or drag it onto a blank area in the report canvas.

The chart shows the net sales amount per unit over all products sold. This chart isn't informative; we'll address it in the next step.

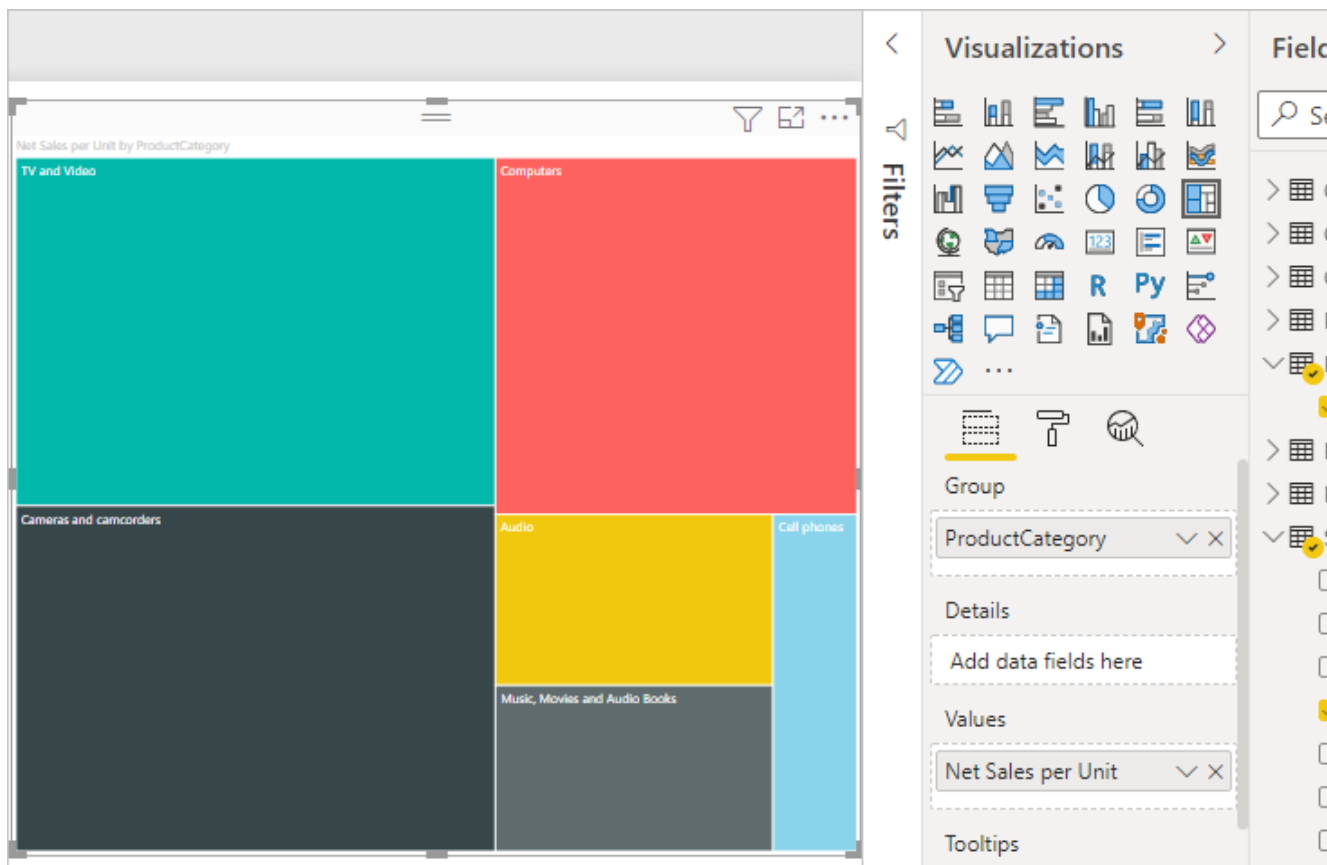


6. For a different look, change the chart visualization type to **Treemap**.

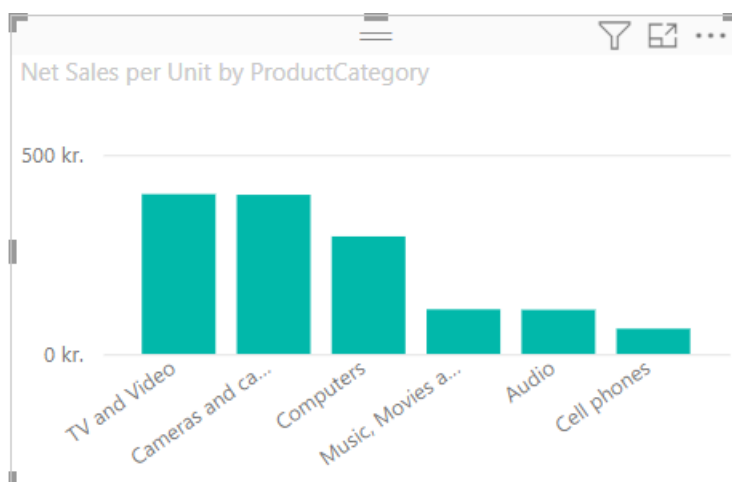


7. Select the **Product Category** field, or drag it onto the treemap or the **Group** field of the **Visualizations** pane. Now you have some good info!

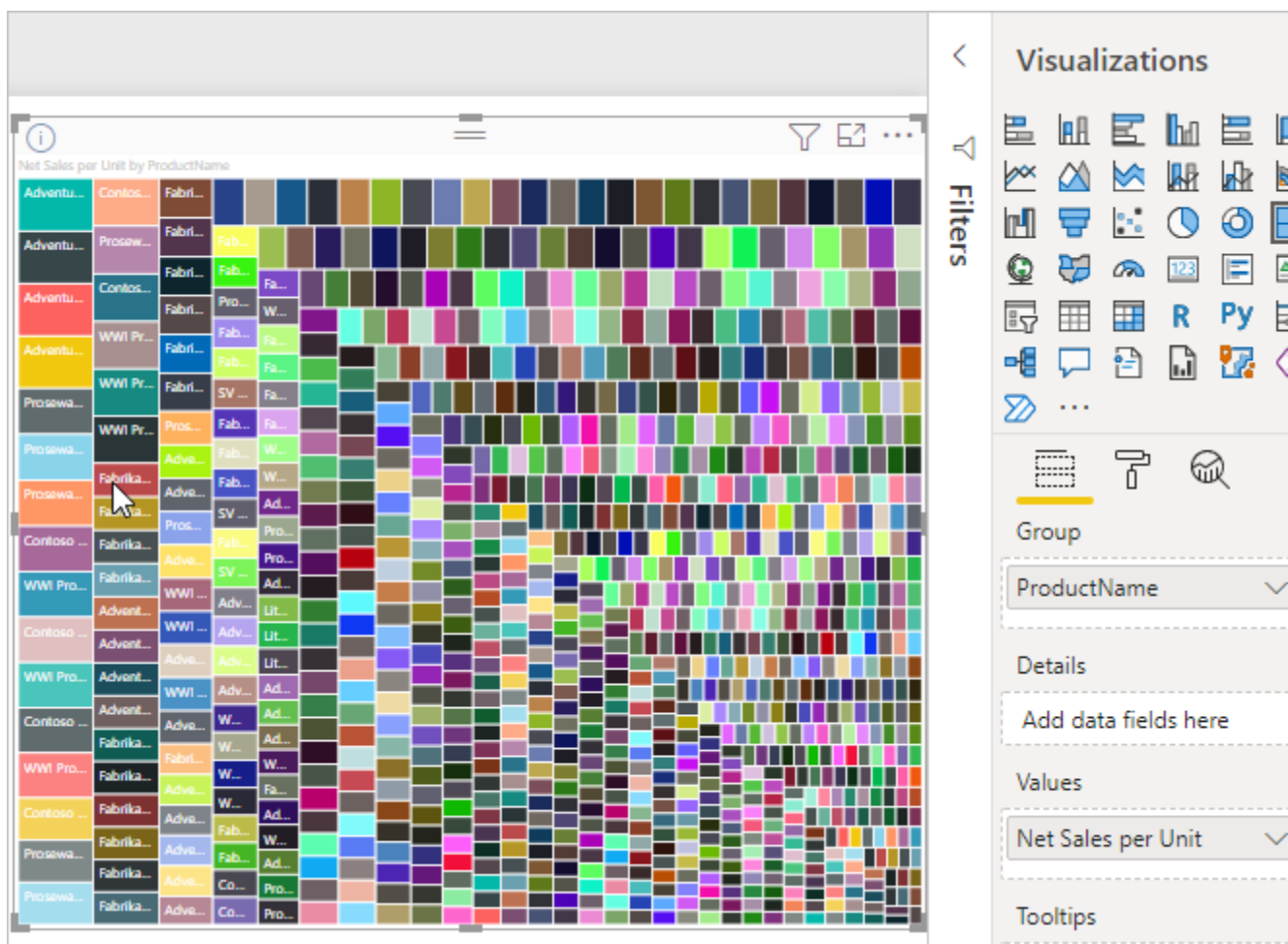
Så grupperes efter værdierne, og measusrens aggregering sker indenfor hver gruppe



viser at grønne, TV og video er dyrest produkter med med net sales /unit er ca 380



- Try removing the **ProductCategory** field, and dragging the **ProductName** field onto the chart instead.



Ok, now we're just playing, but you have to admit that's cool! Experiment with other ways to filter and format the visualization.

What you've learned

Measures give you the power to get the insights you want from your data. You've learned how to create measures by using the formula bar, name them whatever makes most sense, and find and select the right formula elements by using the DAX suggestion lists. You've also been introduced to context, where the results of calculations in measures change according to other fields or other expressions in your formula.

Next steps

- To learn more about Power BI Desktop quick measures, which provide many common measure calculations for you, see [Use quick measures for common calculations](#).
- If you want to take a deeper dive into DAX formulas and create some more advanced measures, see [Learn DAX basics in Power BI Desktop](#). This article focuses on fundamental concepts in DAX, such as syntax, functions, and a more thorough understanding of context.
- Be sure to add the [Data Analysis Expressions \(DAX\) Reference](#) to your favorites. This reference is where you'll find detailed info on DAX syntax, operators, and over 200 DAX functions.

Feedback

Submit and view feedback for

Next unit: Create calculated tables

[Continue](#)

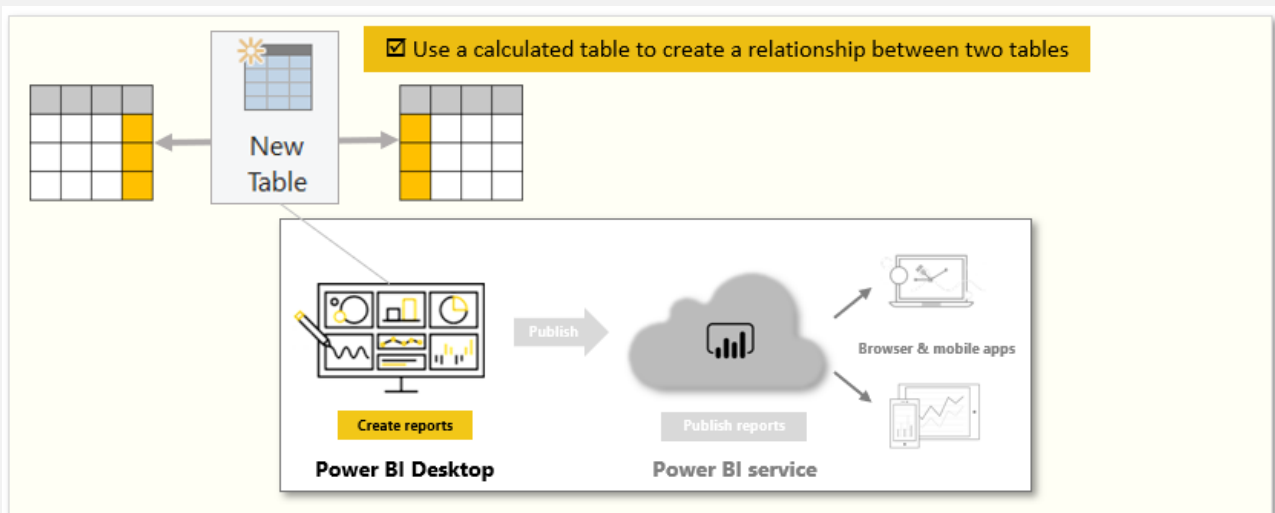
*** Create calculated tables, use to cross join two tables

Completed 100 XP

- 4 minutes

Calculated tables is a function within DAX. Most of the time, you can import data into your model from an external data source. c than as part of a query. You can use calculated tables, for example, to cross join two tables.

Tasks in this unit include:



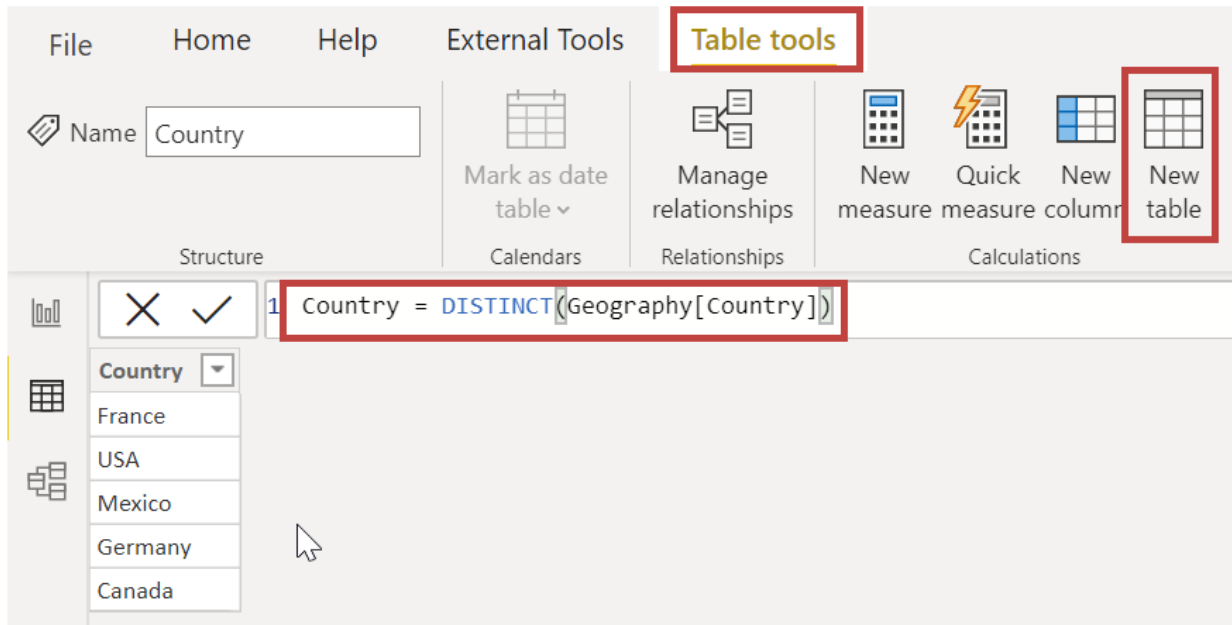
Note

To follow along with the examples on this page, download the sample Access database [here](#) and import into Power BI Desktop (**Get Data > Database > Access database**). If you have any issues loading the Access database, please read this [article](#).

AccessDB PowerBI findes allerede

To create a calculated table, go to **Data (table view)** in Power BI Desktop, on the left side of the report canvas. Select **New Table** from the **Table tools** tab to open the formula bar.

Type the name of your new table, the equal sign, and the calculation that you want to use to form the table. Your new table will appear on the Fields pane in your model.



<enter> tabellen udregnes vises i panelet

After the new table has been created, you can use your calculated table as you would any other table in relationships, formulas, and reports.

For more information, see: [Using calculated tables in Power BI Desktop](#).

Create calculated tables in Power BI Desktop

- Article
- 01/12/2023
- 6 contributors

Feedback

In this article

1. [Create a calculated table](#)
2. [Functions for calculated tables](#)

Most of the time, you create tables by importing data into your model from an external data source. But *calculated tables* let you add new tables based on data you've already loaded into the model. Instead of querying and loading values into your new table's columns from a data source, you create a [Data Analysis Expressions \(DAX\) formula](#) to define the table's values.

DAX is a formula language for working with relational data, like in Power BI Desktop. DAX includes a library of over 200 functions, operators, and constructs, providing immense flexibility in creating formulas to calculate results for just about any data analysis need. Calculated tables are best for intermediate calculations and data you want to store as part of the model, rather than calculating on the fly or as query results. For example, you might choose to *union* or *cross join* two existing tables.

Just like other Power BI Desktop tables, calculated tables can have relationships with other tables. Calculated table columns have data types, formatting, and can belong to a data category. You can name your columns whatever you want, and add them to report visualizations just like other fields.

Calculated tables are recalculated if any of the tables they pull data from are refreshed or updated. If the table uses data from DirectQuery, calculated tables aren't refreshed. In the case with DirectQuery, the table will only reflect the changes after the dataset has been refreshed. If a table needs to use DirectQuery, it's best to have the calculated table in DirectQuery as well.

Create a calculated table

You create calculated tables by using the **New table** feature in Report View, Data View, or Model View of Power BI Desktop.

For example, imagine you're a personnel manager who has a table of **Northwest Employees** and another table of **Southwest Employees**. You want to combine the two tables into a single table called **Western Region Employees**.

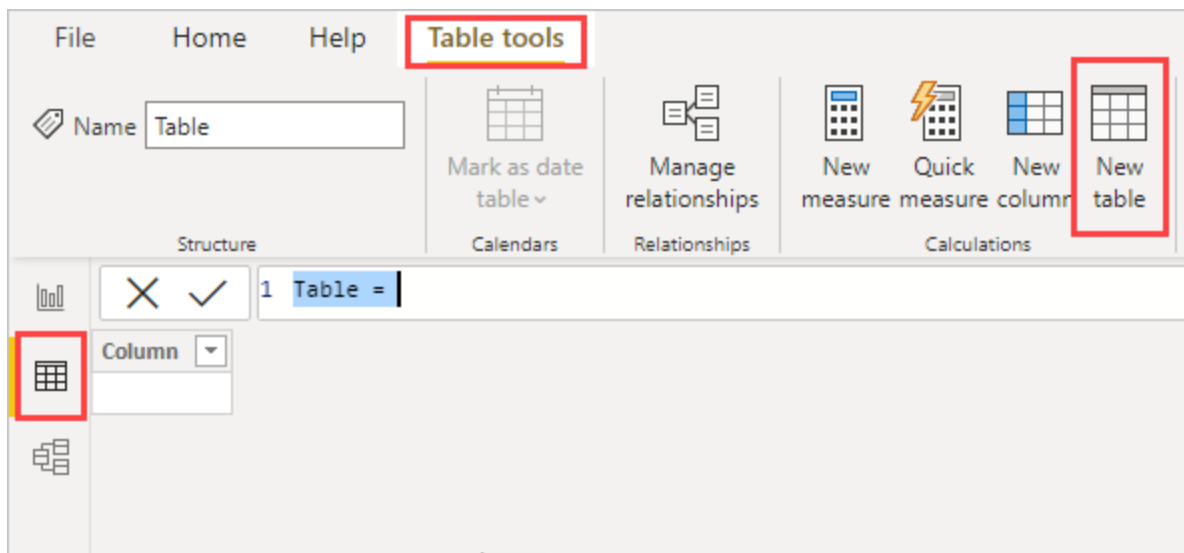
Northwest Employees

Employee	City	State	Tenure
Allen, Kerry	Eugene	OR	1
Baker, Cameron	Portland	OR	15
Morin, Max	Redmond	WA	10
Ramirez, Riley	Portland	OR	3
Rocha, Kim	Redmond	WA	15
Smith, Avery	Redmond	WA	15

Southwest Employees

Employee	City	State	Tenure
Connors, Morgan	San Diego	CA	10
Irwin, Jesse	Phoenix	AZ	3
Nguyen, Rory	Los Angeles	CA	3
Torres, Devon	Los Angeles	CA	2

In Report View, Data View, or Model View of Power BI Desktop, in the **Calculations** group select **New table**. It's a bit easier to do in **Table tools** in the Data View, because then you can immediately see your new calculated table.



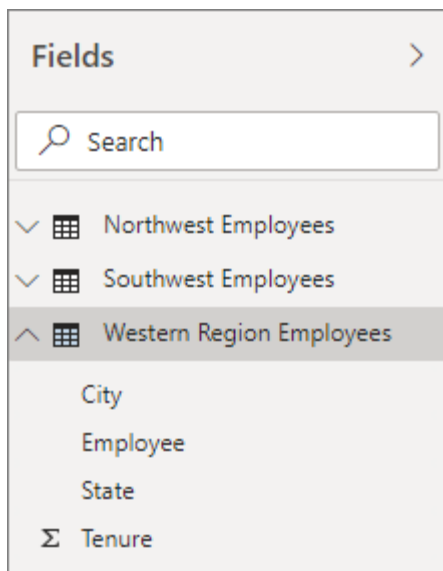
Enter the following formula in the formula bar:

DAXCopy

Western Region Employees = `UNION('Northwest Employees', 'Southwest Employees')`

A new table named **Western Region Employees** is created, and appears just like any other table in the **Fields** pane. You can create relationships to other tables, add measures and calculated columns, and add the fields to reports just like with any other table.

	1	Western Region Employees = <code>UNION('Northwest Employees', 'Southwest Employees')</code>		
	Employee	City	State	Tenure
	Allen, Kerry	Eugene	OR	1
	Baker, Cameron	Portland	OR	15
	Morin, Max	Redmond	WA	10
	Ramirez, Riley	Portland	OR	3
	Rocha, Kim	Redmond	WA	15
	Smith, Avery	Redmond	WA	15
	Connors, Morgan	San Diego	CA	10
	Irwin, Jesse	Phoenix	AZ	3
	Nguyen, Rory	Los Angeles	CA	3
	Torres, Devon	Los Angeles	CA	2



Functions for calculated tables

You can define a calculated table by any DAX expression that returns a table, including a simple reference to another table. For example:

DAXCopy

```
New Western Region Employees = 'Western Region Employees'
```

This article provides only a quick introduction to calculated tables. You can use calculated tables with DAX to solve many analytical problems. Here are some of the more common DAX table functions you might use:

- DISTINCT
- VALUES
- CROSSJOIN
- UNION
- NATURALINNERJOIN
- NATURALLEFTOUTERJOIN
- INTERSECT
- CALENDAR
- CALENDARAUTO

See the [DAX Function Reference](#) for these and other DAX functions that return tables.

Next unit: Explore time-based data

[Continue](#)

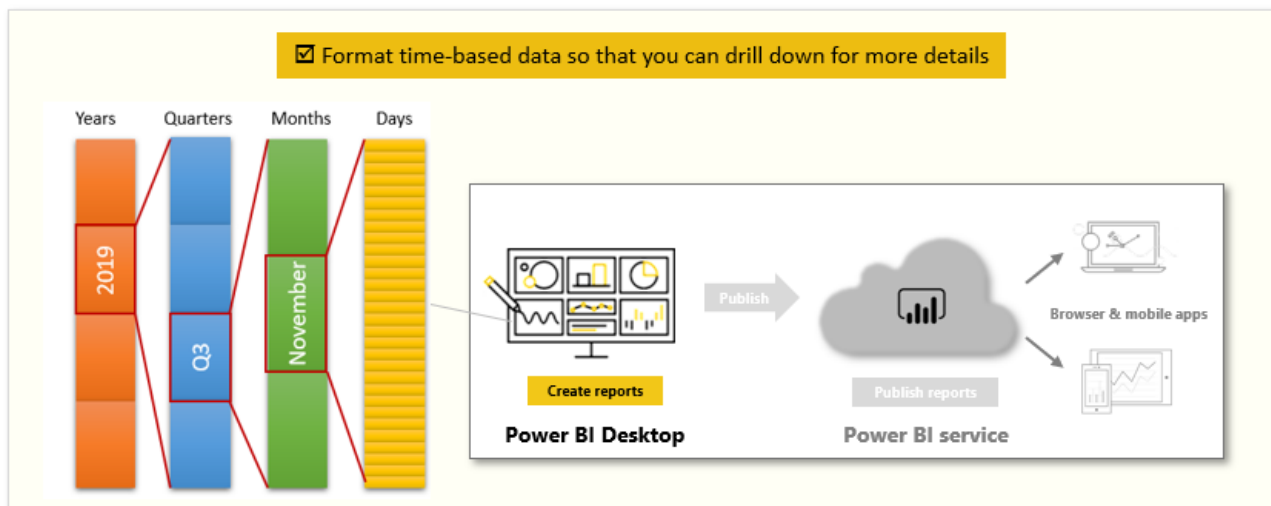
**** Explore time-based data

Completed 100 XP

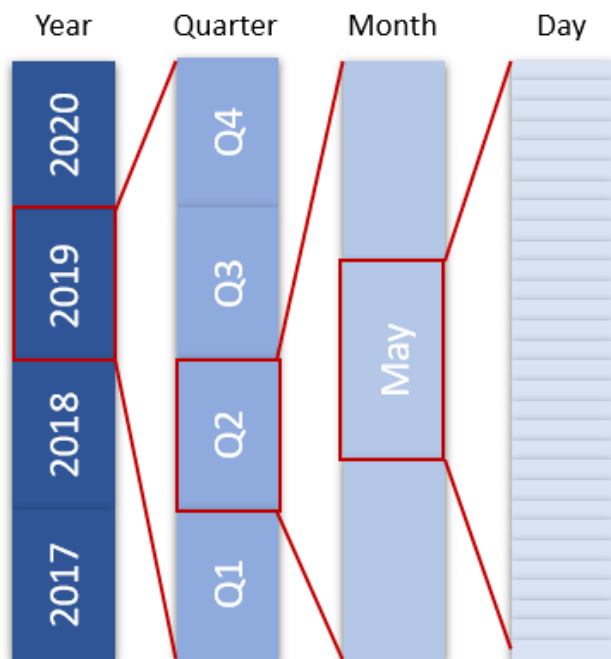
- 5 minutes

Analyzing time-based data with Power BI is a simple process. The modeling tools in Power BI Desktop automatically generate fields that let you drill down through time periods.

Tasks in this unit include:



When you create a table visualization in your report by using a date field, Power BI Desktop automatically includes breakdowns by time period. For example, the single date field in the **Date** table was automatically separated into Year, Quarter, Month, and Day by Power BI.



Year	Quarter	Month	Day	Reven
2000	Qtr 4	December	29	
2000	Qtr 4	December	30	
2000	Qtr 4	December	31	
2001	Qtr 1	January	1	
2001	Qtr 1	January	2	
2001	Qtr 1	January	3	
2001	Qtr 1	January	4	
2001	Qtr 1	January	5	
2001	Qtr 1	January	6	
Total				

Visuals display data at the **year level by default**, but you can change that by turning on **Drill Down** in the top, right-hand corner of the visual.



Søg efter

Filtre på dette visuelle ele...

- DateKey - Day er (Alle)
- DateKey - Month er (Alle)
- DateKey - Quarter er (Alle)
- DateKey - Year er (Alle)
- SalesAmount er (Alle)

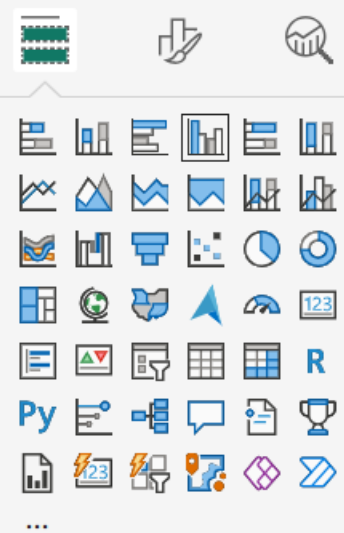
Tilføj datafelter her

Filtre på denne side

Tilføj datafelter her

Hurtigmåling

Byg visual



X-akse

- DateKey
- Year
- Quarter
- Month
- Day

Y-akse

- SalesAmount

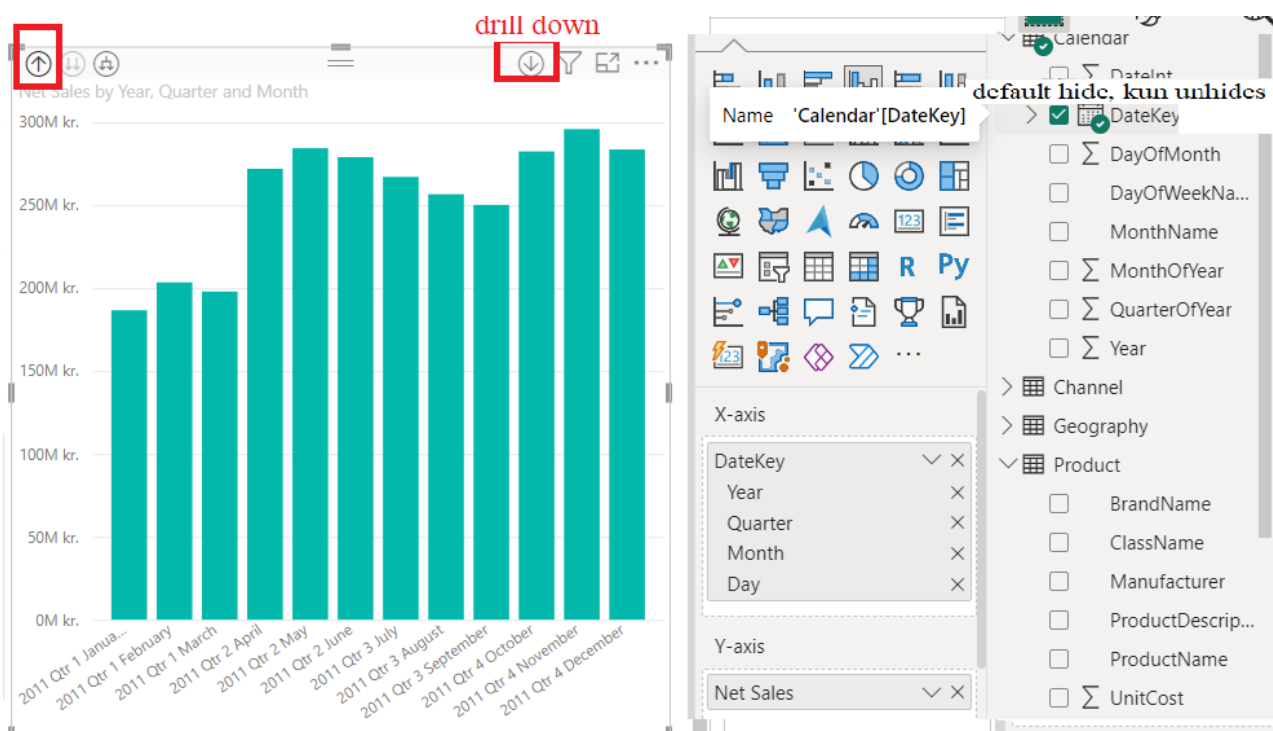
Sales over kun over de tre år 2011 til 2013



drill down knap vist

When you select the bars or lines in your chart, the system will drill down to the next level of time hierarchy, for example, from *years* to *quarters*. You can continue to drill down until you reach the most

granular level of the hierarchy: *days*. To move back up through the time hierarchy, select **Drill Up** in the top, left-hand corner of the visual.

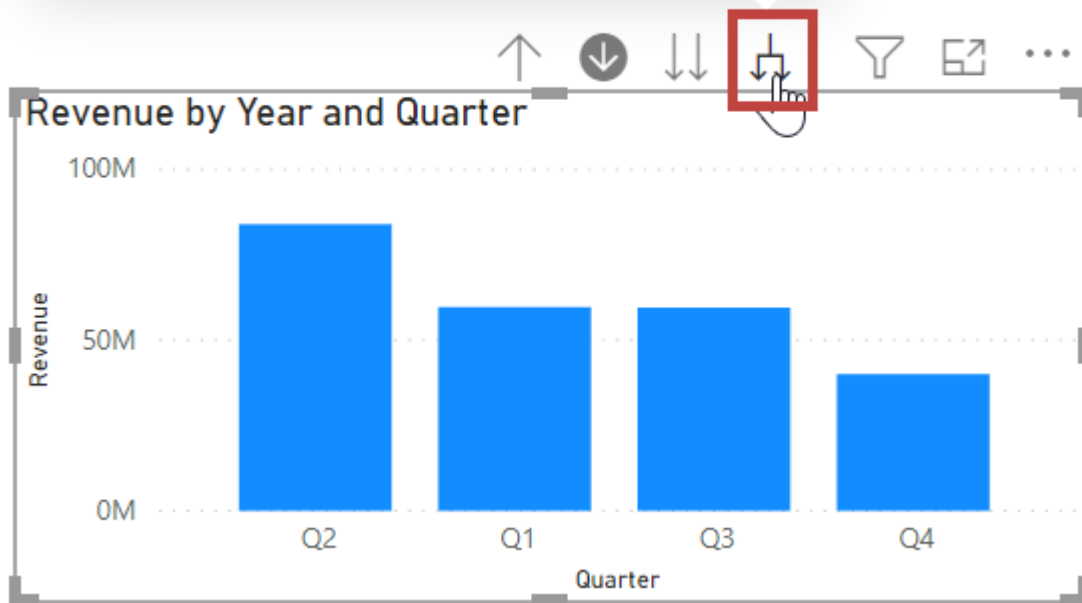


Calendar-tabellen i Contoso har DateKey-felt(date.time) skjult default, men det kan unhides,

og kan så bruges som tidsakse, som ovenfor viser, kun 2011 er valgt med slicer.

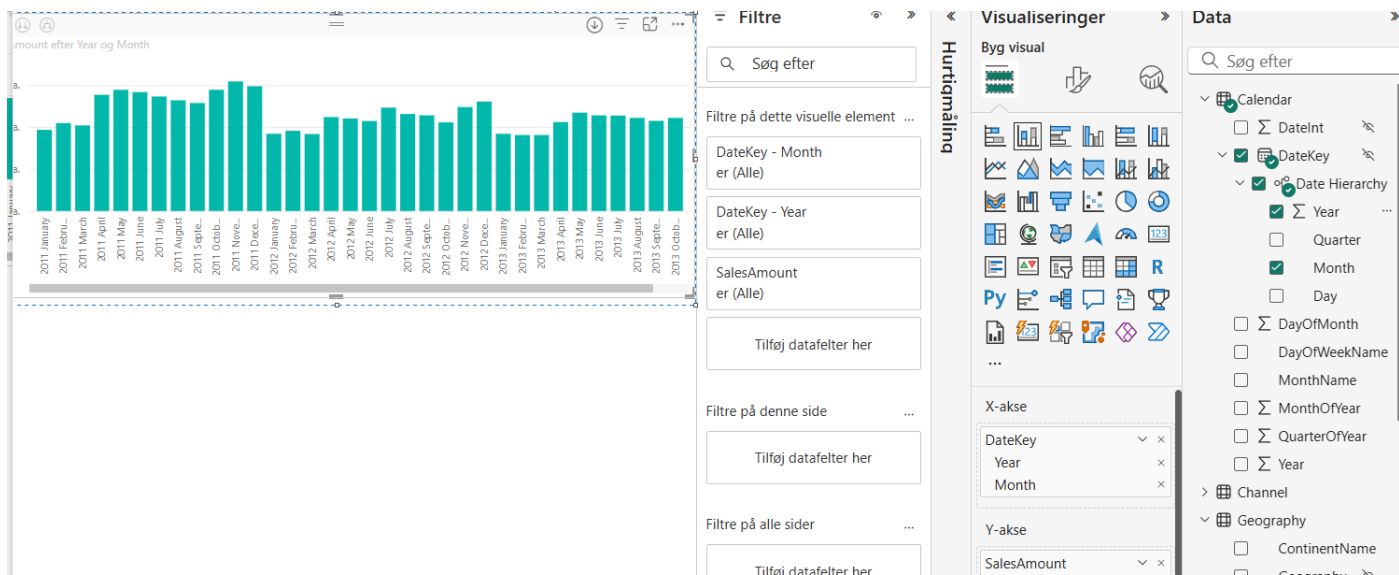
You can also drill down through all the data that is shown on the visual instead of through one selected period. To do so, use the **Go to the next level in the hierarchy** double-arrow icon.

Expand all down one level in the hierarchy



As long as your model has a date field, Power BI will automatically generate different views for different time hierarchies.





Et stablet søjlediagram, hvis flere y-værdier, ville de være lagt ovenpå hinanden.

viser NetSales for alle måneder i valgt år

Next unit: Check your knowledge

[Continue](#)

Need help? See our [troubleshooting guide](#)

4. How do you create a calculation?

☒ Use a calculation

✓ **Calculate**

☐ Select the

☐ Select the

☐ Select the

5. Why would a table be used?

☐ Because it

☐ Because it

Congratulations!

You earned experience points for completing the knowledge check.

Continue

[Review answers](#)

**** Create measures for data analysis in Power BI Desktop

- Article
- 09/18/2023
- 9 contributors

Feedback

In this article

1. [Understanding measures](#)
2. [Data Analysis Expressions](#)
3. [Let's look at an example](#)
4. [Data categories for measures](#)

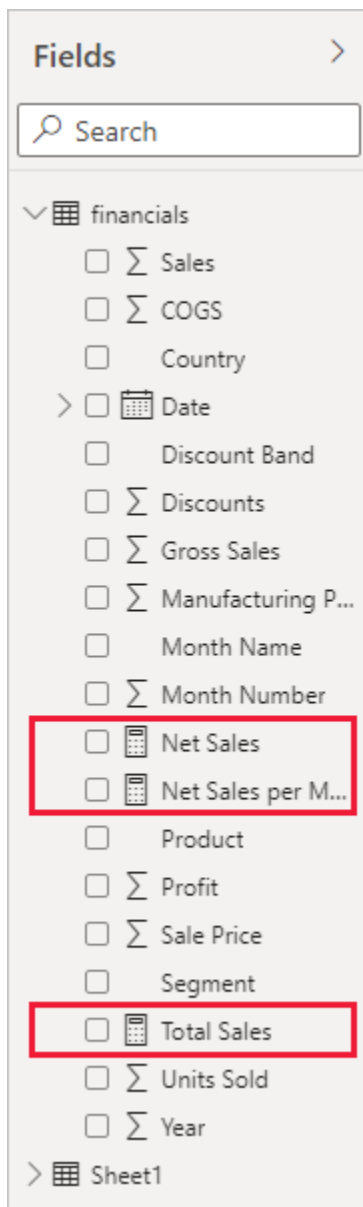
Show 3 more

Power BI Desktop helps you create insights into your data with just a few actions. But sometimes that data just doesn't include everything you need to answer some of your most important questions. Measures can help you get there.

Measures are used in some of the most common data analyses. Simple summarizations such as sums, averages, minimum, maximum and counts can be set through the **Fields** well. The calculated results of measures are always changing in response to your interaction with your reports, allowing for fast and dynamic ad-hoc data exploration. Let's take a closer look. For more information, see [Create measures](#).

Understanding measures

In Power BI Desktop, measures are created and displayed in *Report View*, *Data View*, or *Model View*. Measures you create yourself appear in the **Fields** list with a calculator icon. You can name measures whatever you want, and add them to a new or existing visualization just like any other field.



Note

You might also be interested in *quick measures*, which are ready-made measures you can select from dialog boxes. They're a good way to quickly create measures, and also a good way to learn Data Analysis Expressions (DAX) syntax, since their automatically created DAX formulas are available to review. For more information, see [quick measures](#).

```

1 Gennemsnitligt SalesAmount pr. ProductName =
2 AVERAGEX(
3     KEEPFILTERS(VALUE('Product'[ProductName]));
4     CALCULATE(SUM('Sales'[SalesAmount]))
5 )

```

Hurtigmåling



Copilot kan hjælpe Hent måleforslag i DAX-forespørgselsvisning. [Prøv det nu](#)

Vælg en beregning for at oprette en måling.

Gennemsnit pr. kategori

Beregn gennemsnittet af basisværdien inden for kategori

Basisværdi ⓘ

SalesAmount

Kategori ⓘ

ProductName

For hver kategori, productname laves calculation af sum af SalesAmount

Data Analysis Expressions

Measures calculate a result from an expression formula. When you create your own measures, you'll use the [Data Analysis Expressions](#) (DAX) formula language. DAX includes a library of over 200 functions, operators, and constructs. Its library provides immense flexibility in creating measures to calculate results for just about any data analysis need.

DAX formulas are a lot like Excel formulas. DAX even has many of the same functions as Excel, such like DATE, SUM, and LEFT. But the DAX functions are meant to work with relational data like we have in Power BI Desktop.

Let's look at an example

Janice is a sales manager at Contoso. Janice has been asked to provide reseller sales projections over the next fiscal year. Janice decides to base the estimates on last year's sales amounts, with a six percent annual increase resulting from various promotions that are scheduled over the next six months.

To report the estimates, Janice imports last year's sales data into Power BI Desktop. Janice finds the **SalesAmount** field in the **Reseller Sales** table. Because the imported data only contains sales amounts for last year, Janice renames the **SalesAmount** field to *Last Years Sales*. Janice then drags **Last Years Sales** onto the report canvas. It appears in a chart visualization as a single value that is the sum of all reseller sales from last year.

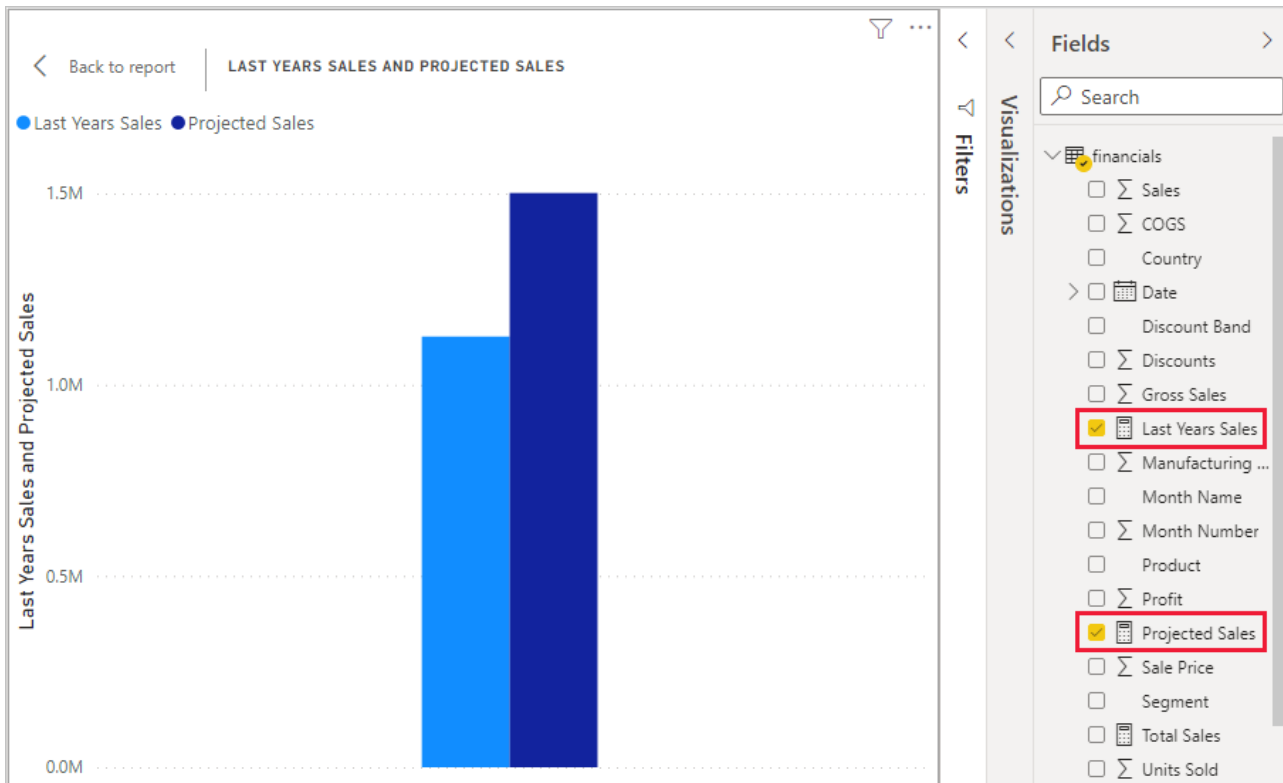
Janice notices that even without specifying a calculation, one has been provided automatically. **Power BI Desktop created its own measure by summing up all of the values in Last Years Sales.**

But Janice needs a measure to calculate sales projections for the coming year, which will be based on last year's sales multiplied by 1.06 to account for the expected 6 percent increase in business. For this calculation, Janice will create a measure. Janice creates a new measure by using the *New Measure* feature, then enters the following DAX formula:

DAXCopy

```
Projected Sales = SUM('Sales'[Last Years Sales])*1.06
```

Janice then drags the new Projected Sales measure into the chart.



Quickly and with minimal effort, Janice now has a measure to calculate projected sales. Janice can further analyze the projections by filtering on specific resellers or by adding other fields to the report.

Data categories for measures

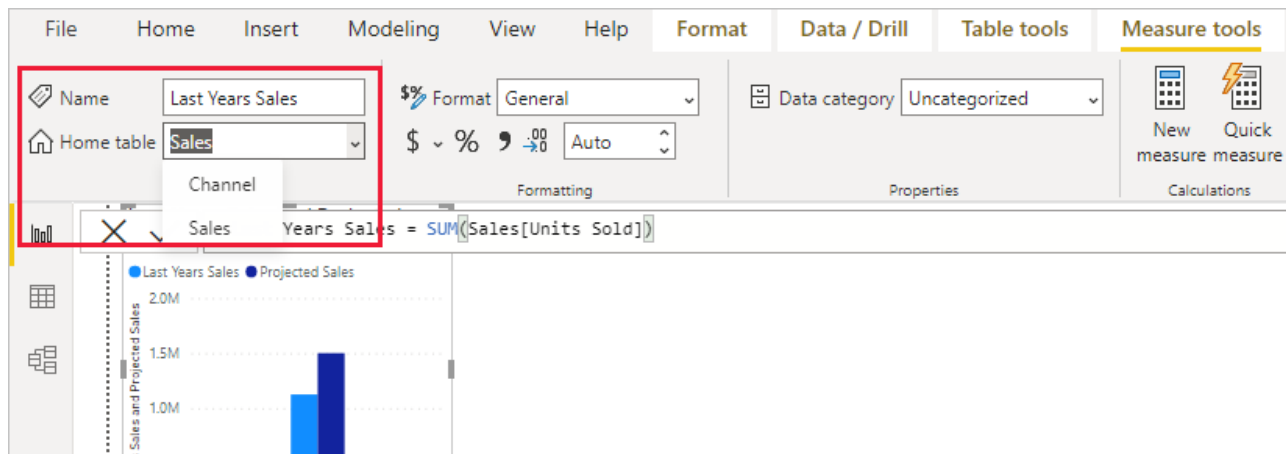
You can also pick data categories for measures.

Among other things, data categories allow you to use measures to dynamically create URLs, and mark the data category as a Web URL.

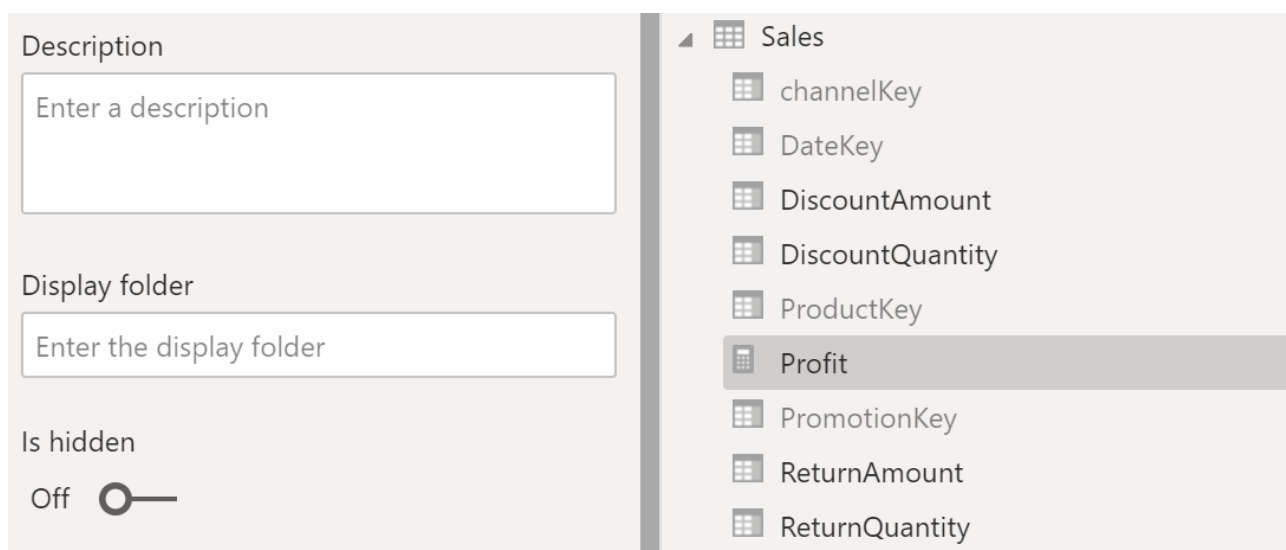
You could create tables that display the measures as Web URLs, and be able to select on the URL that's created based on your selection. This approach is especially useful when you want to link to other Power BI reports with [URL filter parameters](#).

Organizing your measures

Measures have a *Home* table that defines where they're found in the field list. You can change their location by choosing a location from the tables in your model.



You can also organize fields in a table into *Display Folders*. Select **Model** from the left edge of the Power BI Desktop. In the **Properties** pane, select the field you want to move from the list of available fields. Enter a name for a new folder in **Display folder** to create a folder. Creating a folder moves the selected field into that folder.



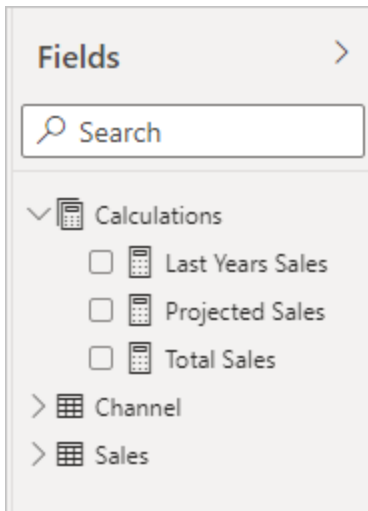
You can create subfolders by using a backslash character. For example, *Finance\Currencies* creates a *Finance* folder and within it, a *Currencies* folder.

You can make a field appear in multiple folders by using a semicolon to separate the folder names. For example, *Products\Names;Departments* results in the field appearing in a *Departments* folder and a *Names* folder inside a *Products* folder.

You can create a special table that contains only measures.

That table always appears at the top of the **Fields–Data**. To do so, create a table with just one column. You can use **Enter data** to create that table. Then move your measures to that table.

Finally, hide the column, but not the table, that you created. Select the arrow at the top of **Fields** to close and reopen the fields list to see your changes.



Tip

Hidden measures are displayed and accessible in Power BI Desktop, however, you won't see hidden measures in Excel or the Power BI services, since Excel and the Power BI service are considered client tools.

Dynamic format strings

With *dynamic format strings*, you can customize how measures appear in visuals by conditionally applying a format string with a separate DAX expression. To learn more, see [Dynamic format strings](#).

Learn more

We've only provided you with a quick introduction to measures here. There's a lot more to help you learn how to create your own. For more information, see [Tutorial: Create your own measures in Power BI Desktop](#). You can download a sample file and get step-by-step lessons on how to create more measures.

To dive a little deeper into DAX, see [Learn DAX basics in Power BI Desktop](#). The [Data Analysis Expressions Reference](#) provides detailed articles on each of the functions, syntax, operators, and naming conventions. DAX has been around for several years in Power Pivot in Excel and SQL Server Analysis Services. There are many other great resources available, too. Be sure to check out the [DAX Resource Center Wiki](#), where influential members of the BI community share their knowledge of DAX.