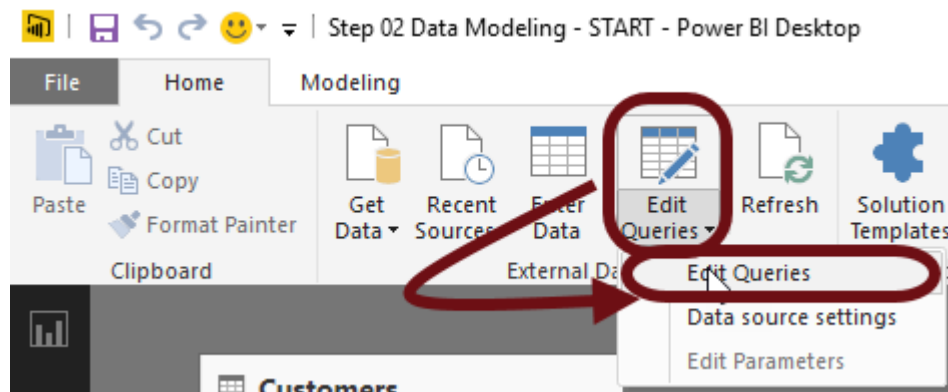# Data Modeling

Steven Renders

Table of Contents

Step 02 Data Modeling

You can continue to work on your previous Power BI Desktop file, or you can start with this file: Step 02
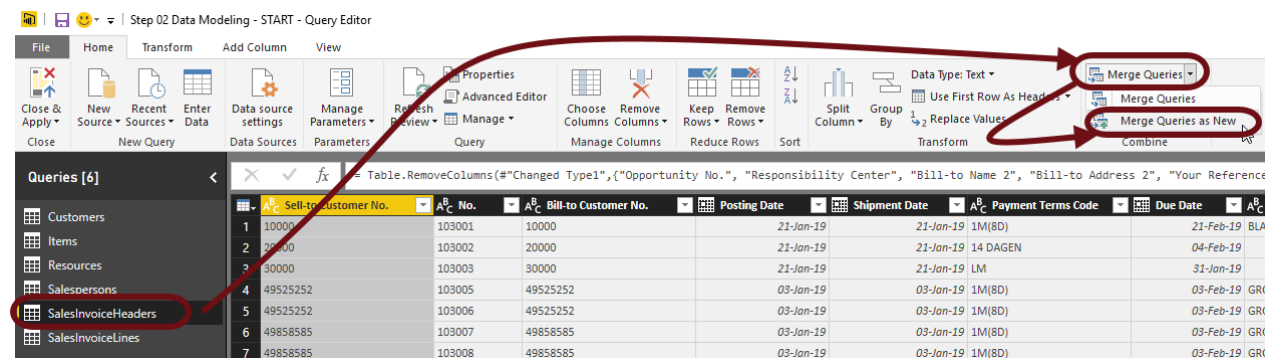Step 02 Data Modeling - START.pbix

Merge Tables

In this step you merge the SalesInvoiceHeaders and SalesInvoiceLines queries, into a new query.

In Power BI Desktop click Edit Queries:

Select the SalesInvoiceHeaders query and select Merge Queries, Merge Queries as New:

In the popup that opens, select SalesInvoiceLines in the dropdown below the first table. Then select the
No. in the first table and the Document No. in the second table. These columns are used as the link
between the tables:

Merge

Select tables and matching columns to create a merged table.

SalesInvoiceHeaders

Sell-to Customer No.	No.	Bill-to Customer No.	Posting Date	Shipment Date	Payment Terms Code	Du
10000	103001	0000	21-Jan-19	21-Jan-19	1M(8D)	
20000	103002	0000	21-Jan-19	21-Jan-19	14 DAGEN	
30000	103003	0000	21-Jan-19	21-Jan-19	LM	
49525252	103005	9525252	03-Jan-19	03-Jan-19	1M(8D)	
49525252	103006	9525252	03-Jan-19	03-Jan-19	1M(8D)	

SalesInvoiceLines

Sell-to Customer No.	Document No.	Line No.	Type	No.	Posting Group	Quantity	Amount	Am
10000	103001	0000	Resource	DICK		25	1971.25	
10000	103001	0000	Resource	DICK		120	9462	
20000	103002	0000	Resource	DICK		25	2012.75	
20000	103002	0000	Resource	DICK		96	7728.96	
30000	103003	0000	Resource	DICK		25	2075	

Join Kind

Left Outer (all from first, matching from second)

✔ The selection has matched 30 out of the first 30 rows.

OK   Cancel

Click the Ok button.

At the right of the window, select the icon to expand the table:

nd.LeftOuter)

ent Terms Code	Due Date	Aᴮ꜀ Location Code	Aᴮ꜀ Currency Code	Aᴮ꜀ Cust. Ledger Entry No.	NewColumn
	21-Feb-19	BLAUW		2795	Table
	04-Feb-19			2798	Table
	31-Jan-19			2801	Table
	03-Feb-19	GROEN		2599	Table
	03-Feb-19	GROEN		2601	Table
	03-Feb-19	GROEN		2603	Table
	03-Feb-19	GROEN		2605	Table
	03-Feb-19	GROEN		2607	Table

Deselect the option: Use original column name as prefix.

Deleselect the Sell-to Customer No. field, as we already have it in the table.
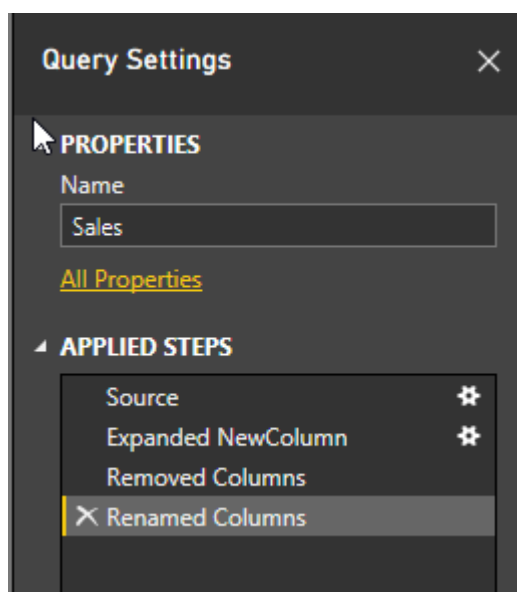
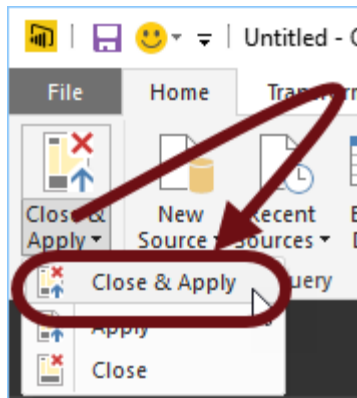In the resulting columns, remove the No. column, because we have it twice:

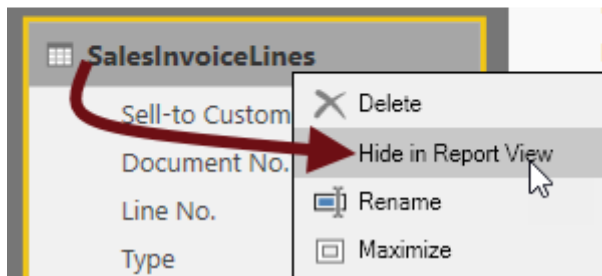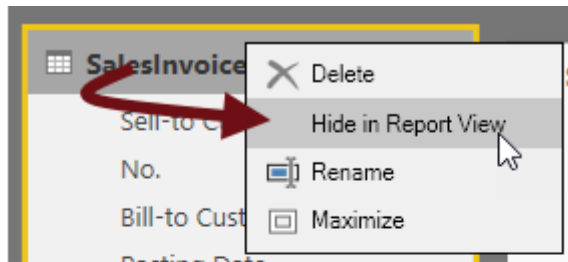Now rename the No..1 column to: No:
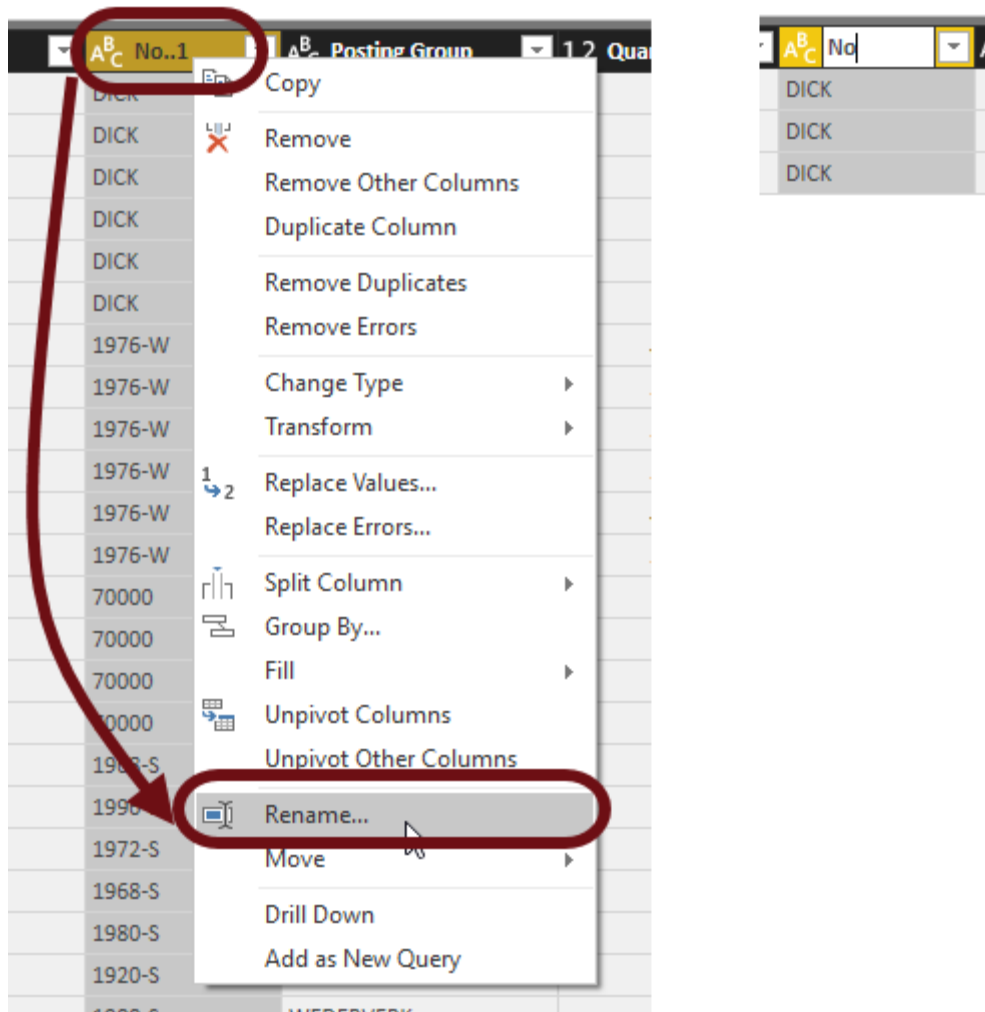
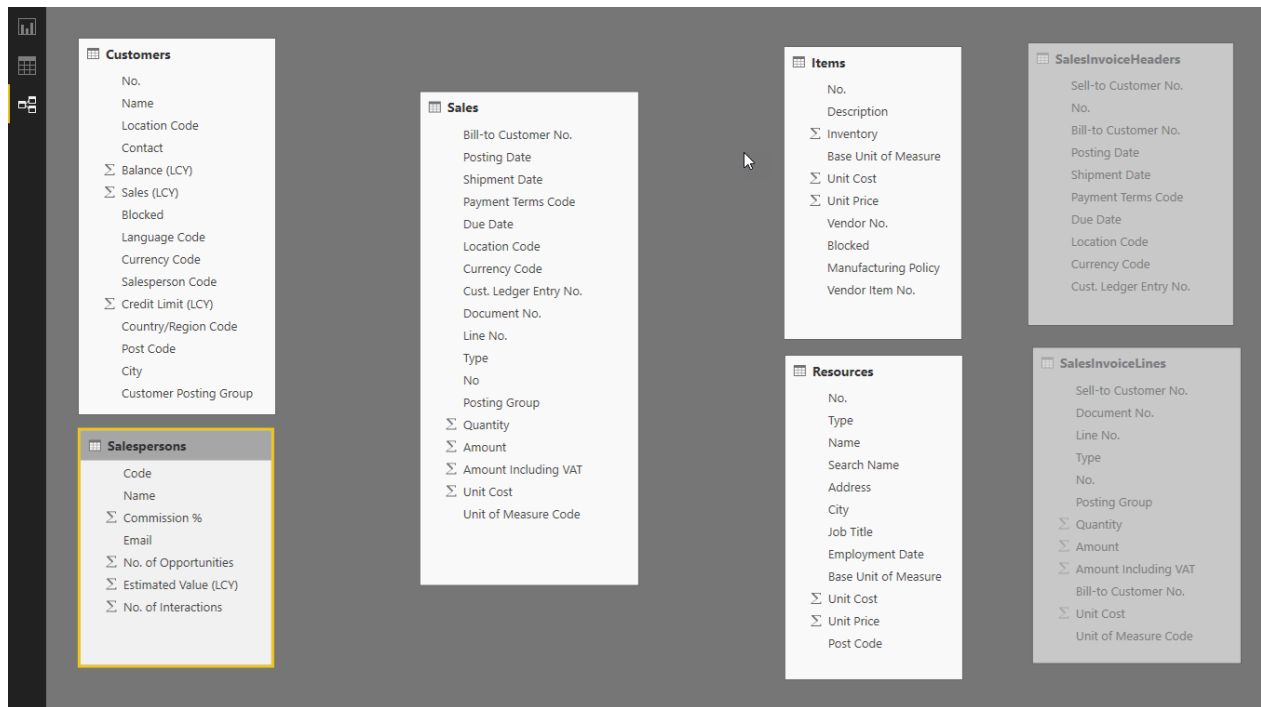Rename the new query to: Sales:

Select Close and apply:

In the relationships window, hide the SalesInvoiceHeaders and SalesInvoiceLines queries:

The resulting data model should now resemble the following:

Create Relationships

In this step you create relationships, links, between the tables in order to have a star/snowflake data model.

In the relationships window, drag the Bill-to Customer No. field of the Sales table onto the No. field of the Customer table:
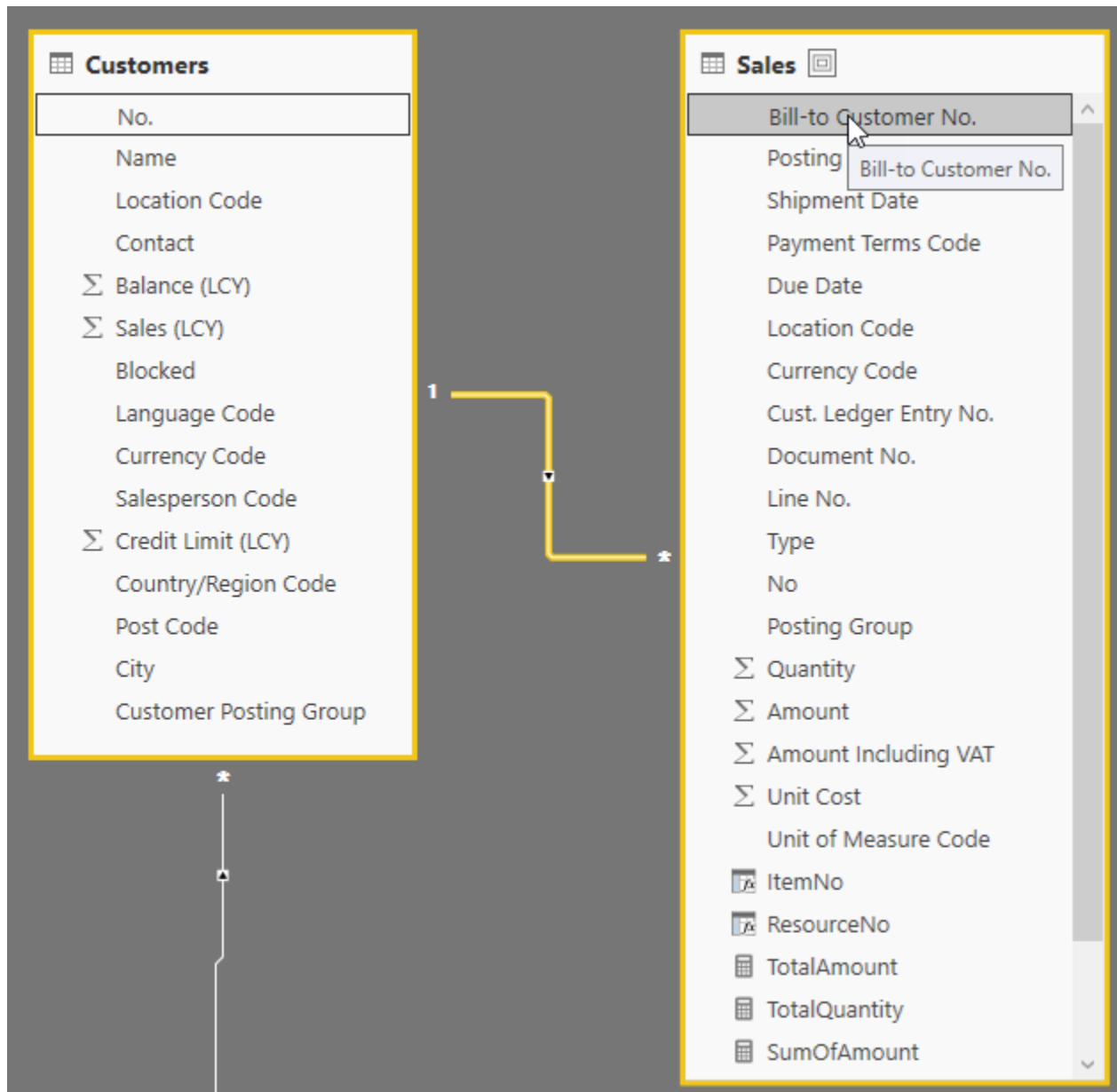
Now create a similar relationship between the Customers and Salespersons table:
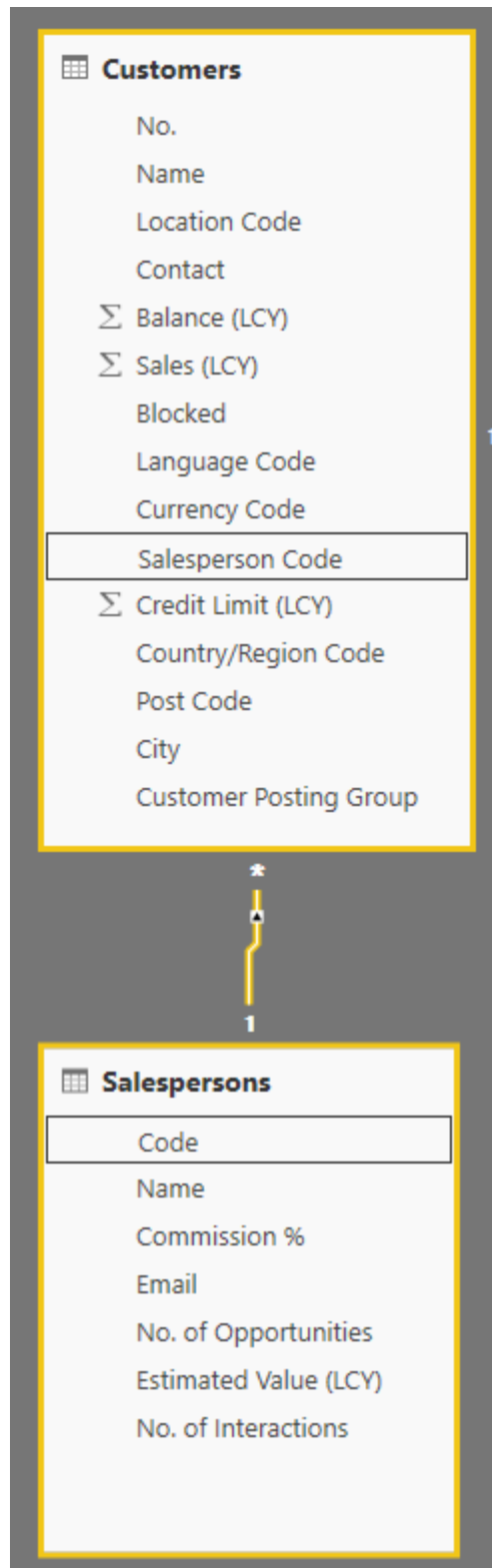
Now, in order to complete our data model, create a relationship between the Sales table and the Items and Resources table:
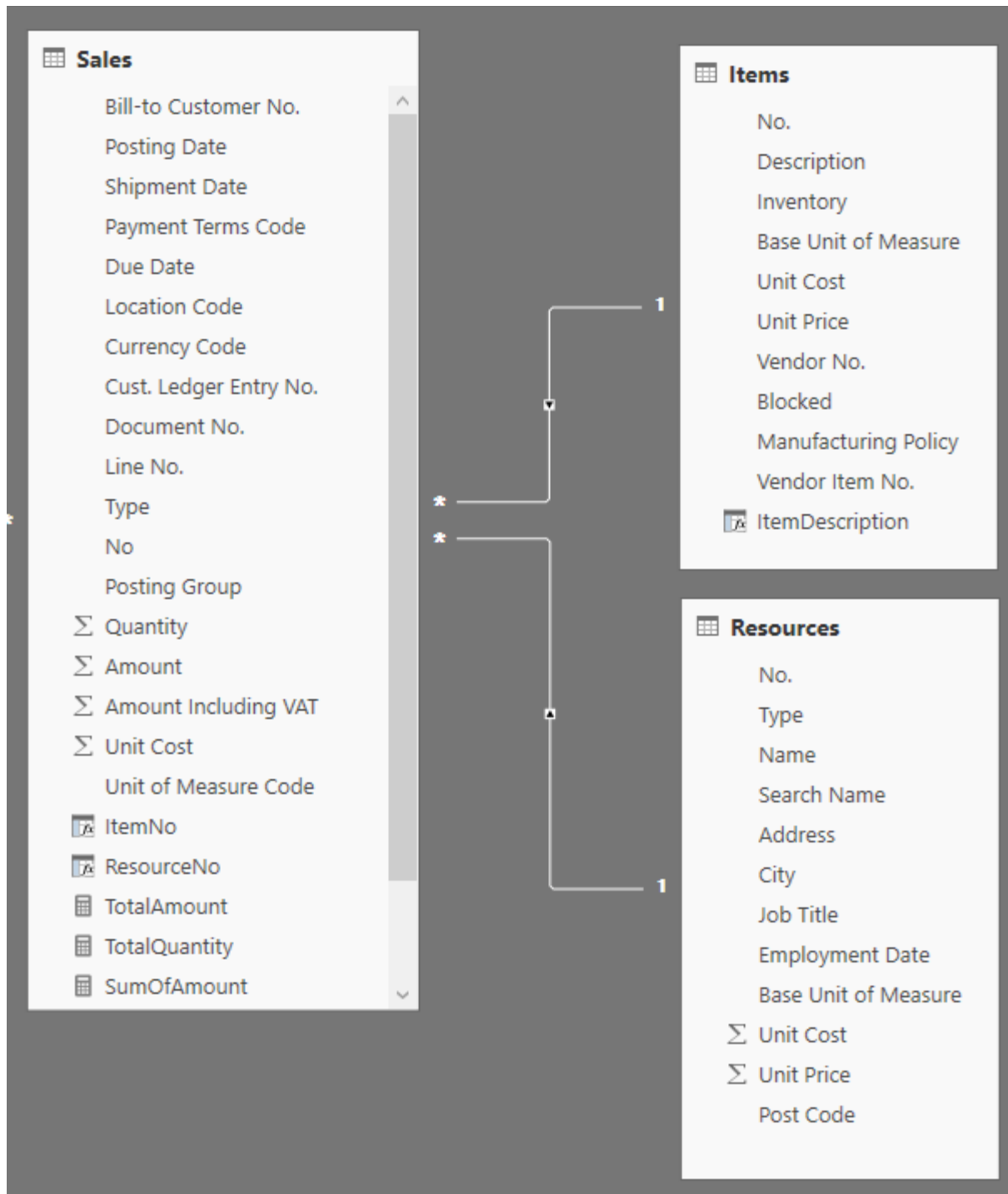
When you try this, you will notice a problem. You cannot use the No field from the Sales table to create this relationship, because it contains both values:

Type	No
Resource	DICK
Resource	DICK
Resource	DICK
Resource	DICK
Resource	DICK
Resource	DICK
Item	1976-W
Item	1976-W
Item	1976-W
Item	1976-W
Item	1976-W
Item	1976-W
Item	70000
Item	70000

If you would create a relationship between the Sales and Items table, using the No field from Sales and No. field from Items, the relationship would make no sense. The same is valid for the No field from the Sales table and the No. field from the Resources table.

In order to create these two relationships, we need two new fields in the Sales table: ItemNo and ResourceNo. You will create these fields as calculated columns in the Sales table.

Create Calculated Columns

In the data tab in Power BI Desktop, select the Sales table and then in the Modeling tab in the ribbon, click New Column:
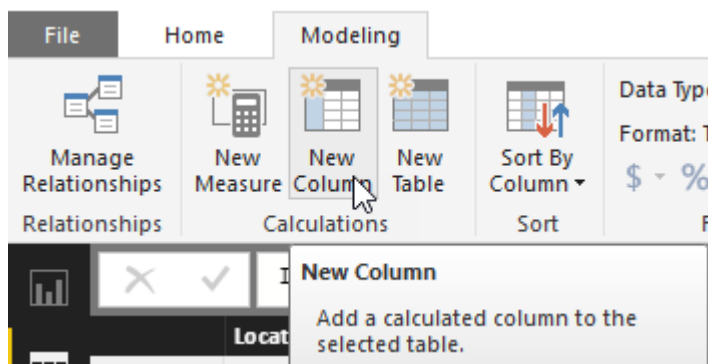
In the formula enter: **ItemNo = IF(Sales[Type] = "Item",Sales[No])**
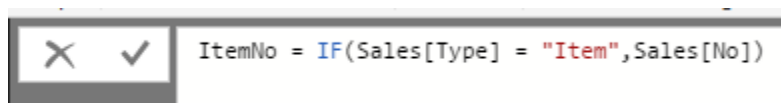
Then click the V button, and the column gets created:

Now repeat these steps to create a new calculated column: ResourceNo:

Now that you have the two calculated columns in the Sales table, you can use them to create relationships to the Items and Resources tables:
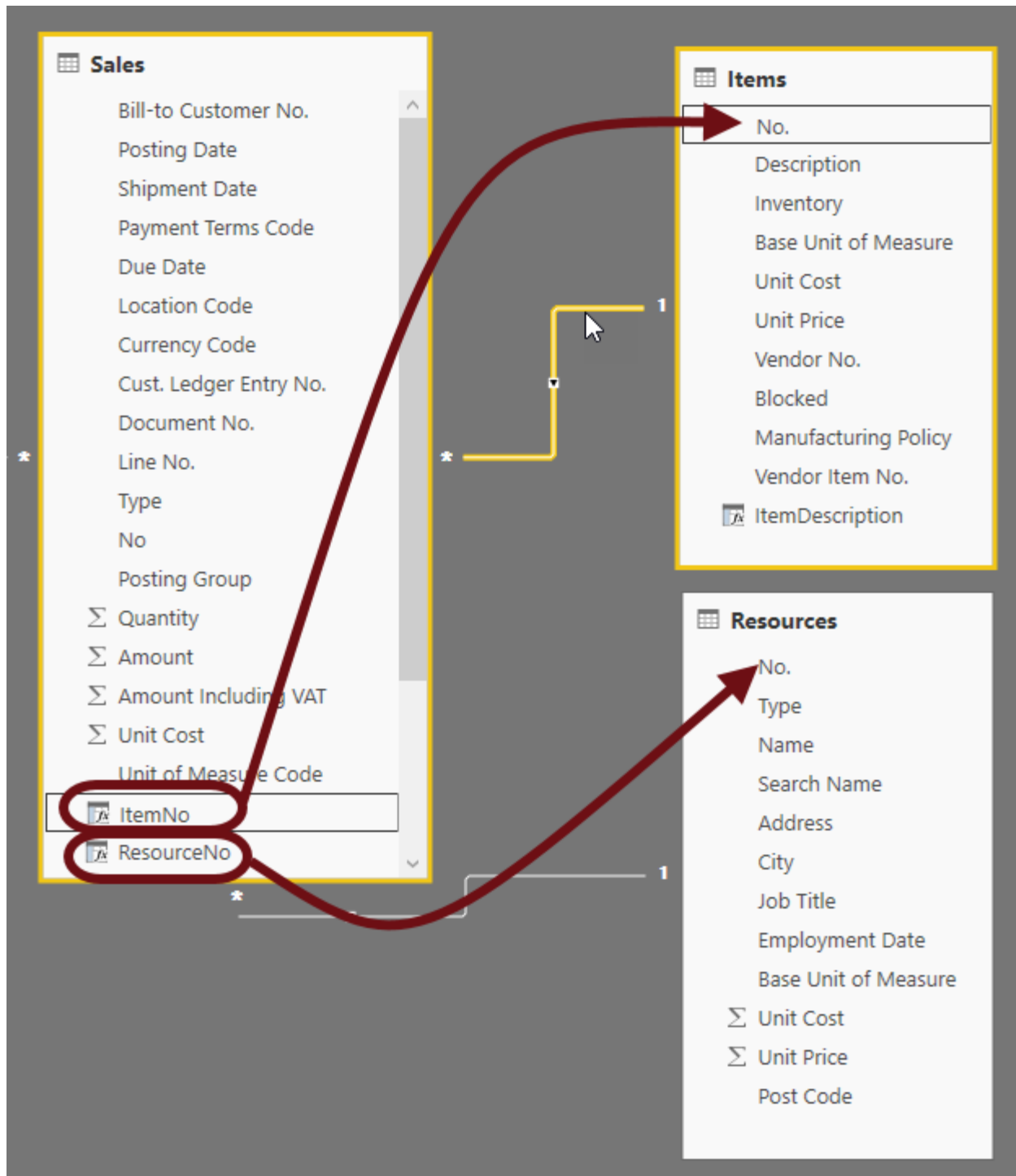
Now create a new column in the Items table that concatenates the Description and the number:

ItemDescription
Tourfiets (1001)
Bevestiging voor LS-100 (SPK-100)
Subwoofer 8" 100W (LSU-8)
Tweeter 4" 100W (LSU-4)
Woofer 15" 100W (LSU-15)
Standaard voor speakers LS-150 (LS-S15)
Handleiding voor speakers (LS-MAN-10)
Speaker, notenhout, 80W (LS-81)
Speakers, Cherry, 75W (LS-75)
Kabels voor speakers (LS-2)
Speakers, Cherry, 150W (LS-150)
Speakers, Zwart, 120W (LS-120)
Speakers, Wit voor pc (LS-10PC)
Speaker 100W OakwoodDeluxe (LS-100)
Behuizing LS-100,Eiken 120 lts (HS-100)

Try it yourself, and if you can find the correct DAX expression, use this as inspiration:

ItemDescription = Items[Description] & " (" & Items[No.] & ")"

Create Measures

One of the requirements of our report (and dashboard), is to show the total amount, by customer, and also compare it with the grand total, to see the percentage of total sales:

| Bill-to Customer No.	SumOfAmount▼	TotalAmount	% of TotalAmount
35963852	203,408.16	590,994.47	34.42 %
47563218	114,729.33	590,994.47	19.41 %
35451236	88,160.64	590,994.47	14.92 %
42147258	60,228.99	590,994.47	10.19 %
10000	26,762.09	590,994.47	4.53 %
49858585	22,298.71	590,994.47	3.77 %
43687129	21,264.96	590,994.47	3.60 %
20000	12,404.27	590,994.47	2.10 %
30000	9,449.70	590,994.47	1.60 %
49525252	9,294.03	590,994.47	1.57 %
49633663	7,401.40	590,994.47	1.25 %
46897889	6,808.24	590,994.47	1.15 %
32656565	4,000.25	590,994.47	0.68 %
01445544	2,311.30	590,994.47	0.39 %
40000	1,646.30	590,994.47	0.28 %
50000	826.10	590,994.47	0.14 %
Total	**590.994.47**	**590.994.47**	**100.00 %**

To do that, you will need to add the following Measures to the Sales table:

- SumOfAmount
- TotalAmount
- % of TotalAmount

To create the measure SumOfAmount in the Sales table, goto the data tab, select the Modeling pane in the ribbon and click New Measure:
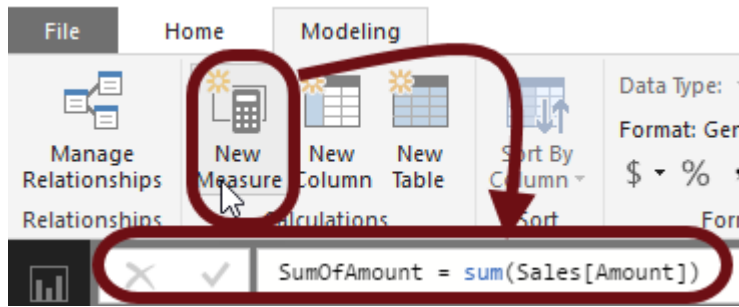
When you do this, the measure will become visible in the list of fields of the Sales table:
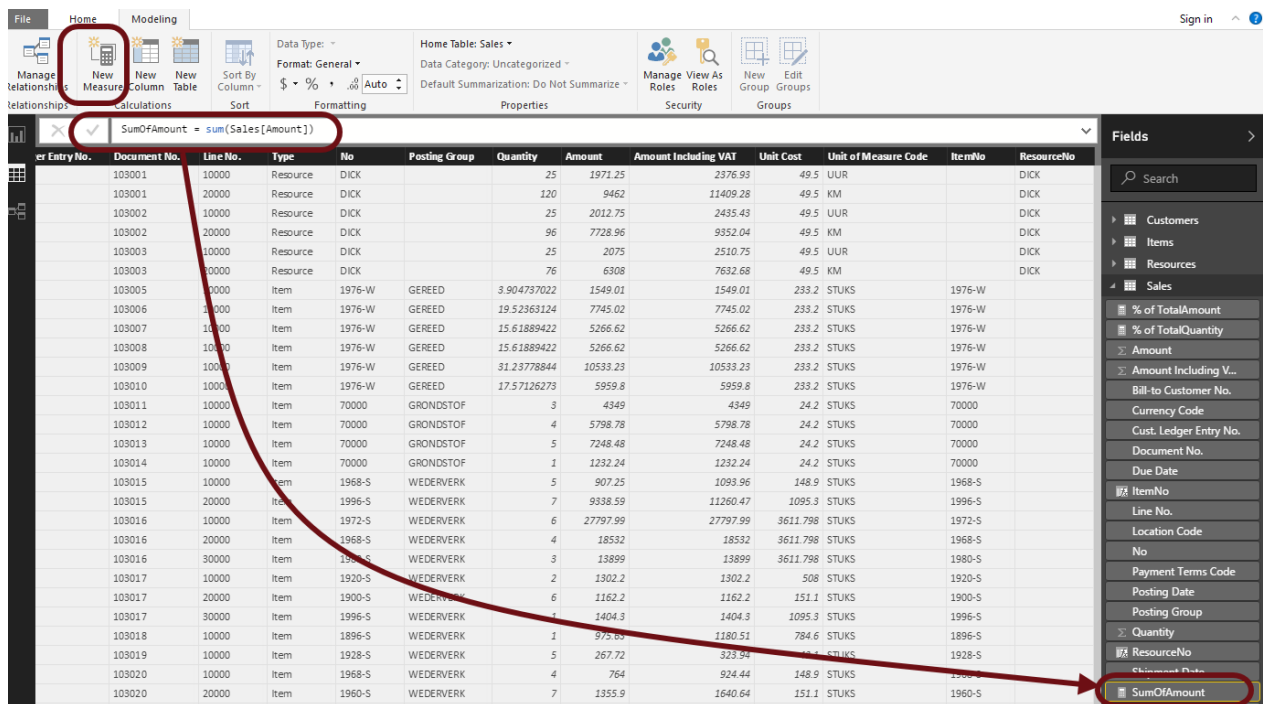
But, it is not added as a column, so you will not see it in the data.

The expression **SumOfAmount = sum(Sales[Amount])** is a DAX expression, that calculates the Sum of the column Amount in the Sales table.

When you go to the report tab and add this measure in a table, the following is the result:

Now, if you change the visual to a Matrix, and also add the Bill-to Customer No you will see the following:

As you can see in the table (matrix), there's the total amount per customer and at the bottom the grand total. Now you need to calculate the % of the Total Amount, per customer. The problem is, how to divide each Sum Of Sales by the Total shown?

To do this, you need a new measure, with a similar calculation, being the sum of amount, but it should always be the grand total, not a subtotal per customer.

To do this, create a new measure TotalAmount with the following formula:

TotalAmount = CALCULATE(Sales[SumOfAmount],ALL(Sales))

This formula tells DAX to ignore the filter context imposed by each subcategory row field using a combination of CALCULATE and ALL functions.

Add this field to the matrix:
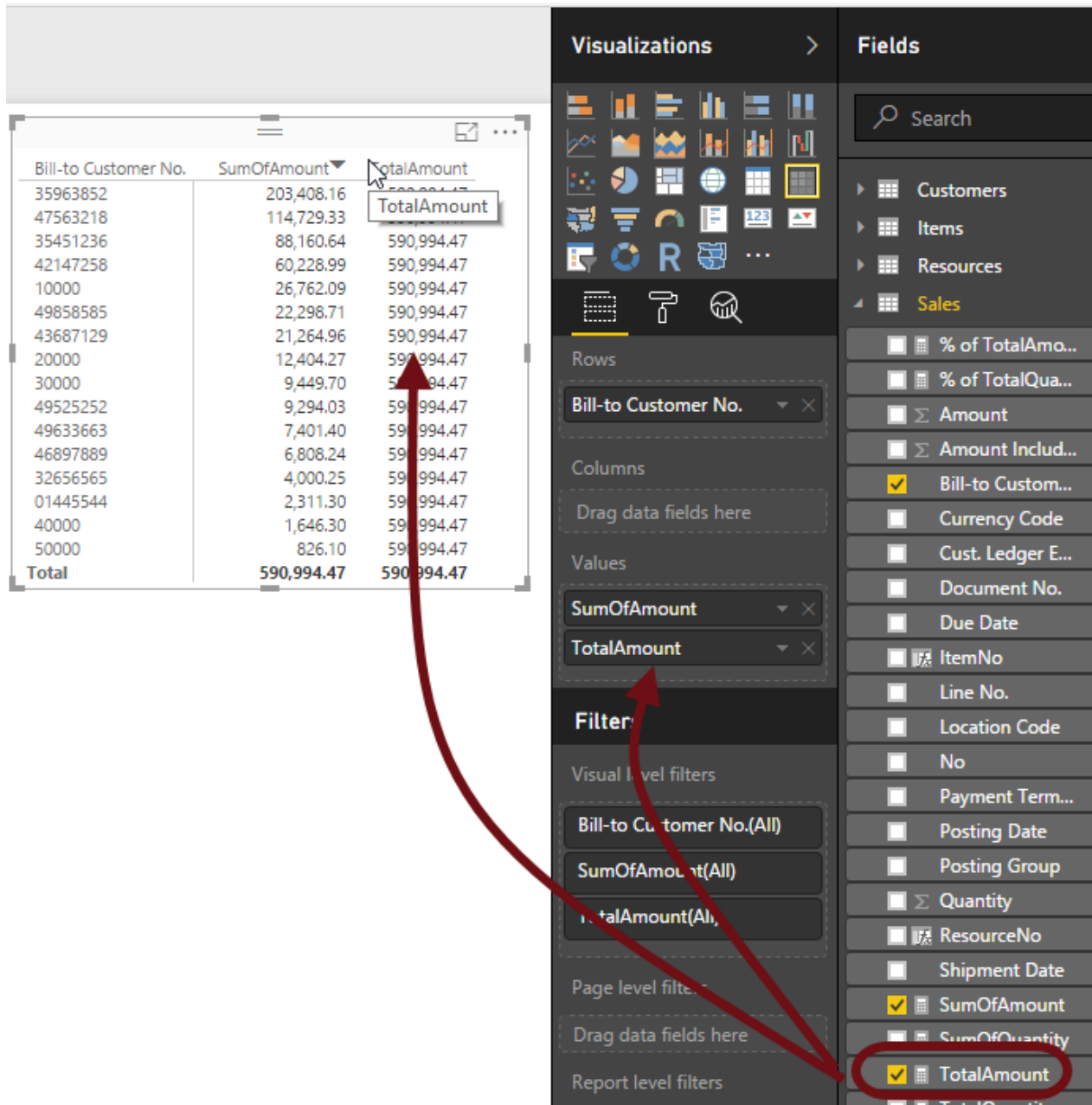
Now you can see the SumOfAmount, per customer, and compare it with the TotalAmount.

The measure always shows the Total sales in each row, though there is a filter context indicated by each subcategory. What happened is that the CALCULATE function modified the existing filter context in each row by overriding it with its own: ALL(Sales).

For practical purposes, this ALL function makes the measure ignore the existing filter context.

So what does the CALCULATE function do exactly?

The CALCULATE function is useful because it can modify the filter context of the Expression inside, as indicated by the n number of filter conditions specified by the user, according to these steps:

- If the filter context specified by a filter condition already exists, it will override such already existing filter context with the new one specified in the CALCULATE expression,
- If the filter context does not exist at all, it will add a new one according to the filter condition specified.

Now create another new measure with the following formula:
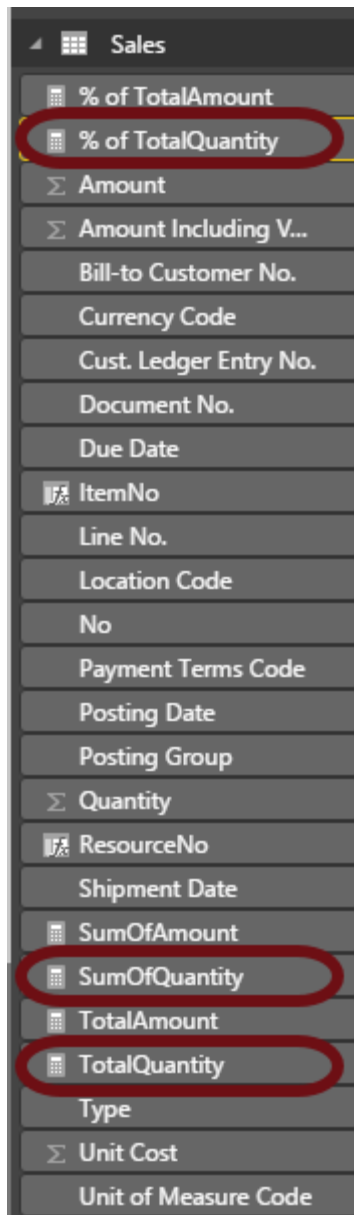
% of TotalAmount = [SumOfAmount] / [TotalAmount]

And the add it to the Matrix:

Now that we have three new measures with the Amounts, repeat the same steps to create the following measures:

- SumOfQuantity
- TotalQuantity
- % of TotalQuantity

Although superior, the calculate function is very similar to the Excel Sumif function.

Understanding the filter context is essential.

As we saw from our examples, the CALCULATE function modifies or adds filter contexts to a calculation.

Add a Date Dimension

There are many ways to build a date table in Power BI Desktop. We will do it in a way most people are familiar with, using DAX Expression: the 'CALENDARAUTO' function

It will automatically create the table with a field 'Date' based on the Min and Max values in your existing dataset.

Goto Modeling and click New Table:
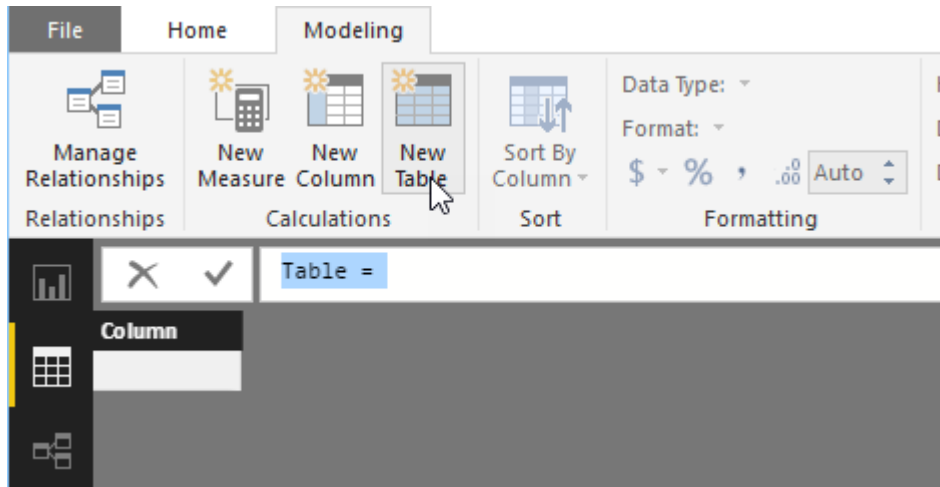
Enter the following formula:

DateTable = CALENDARAUTO()

When you apply the function, the DateTable gets created:

Next, you need to build the other required date fields. For each required field, use the 'New Column' button, this is found next to the 'New Table' button we just used. Then use the DAX formulas as illustrated below:

Formulas:

- Year = YEAR(DateTable[Date])
- MonthNo =MONTH(DateTable[Date])
- QuarterNo = int((DateTable[MonthNo] + 2)/3)
- Quarter = "Qtr" & DateTable[QuarterNo]
- Month =FORMAT(DateTable[Date],"MMMM")
- MMM-YYYY=FORMAT(DateTable[Date],"MMM-YYYY")
- DayOfWeekNo = WEEKDAY(DateTable[Date])
- DayOfWeek =format(DateTable[Date],"DDDD")
- Day =day(DateTable[Date])

Then for the Quarter set the Sort By Column to QuarterNo

Then for the Month set the Sort By Column to MonthNo

Then for the DayOfWeek set the Sort By Column to DayOfWeekNo
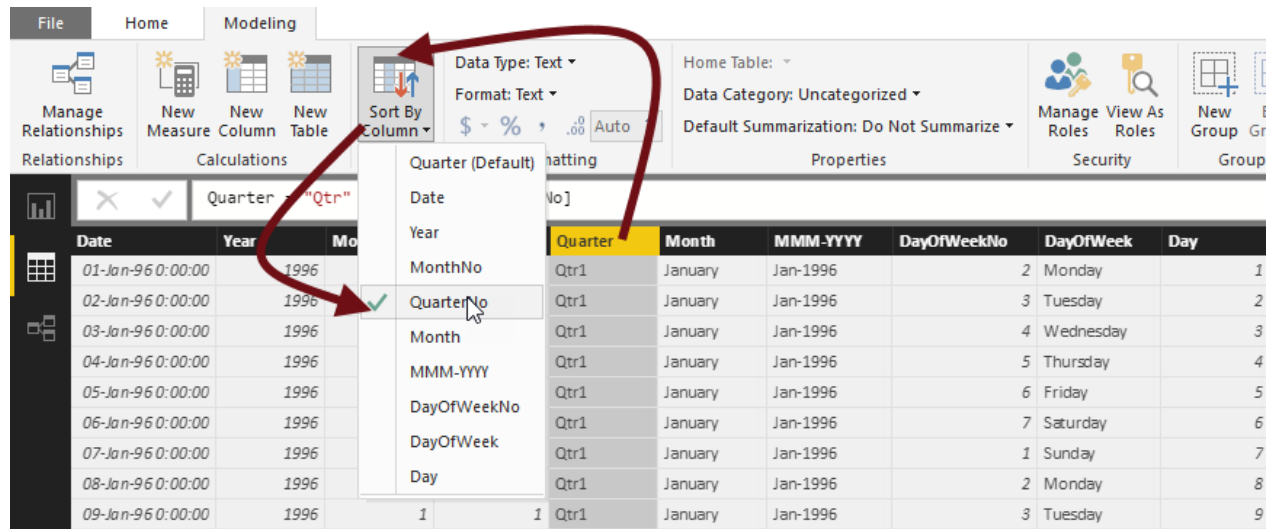
Now we need to create a relationship between our 'Date table and the 'Sales' table.

Use a many-to-one relationship from 'Sales' to 'Calendar', which can be created in the Relationships view by clicking and dragging the 'Posting Date' column label in 'Sales' table onto the 'Date' column in the 'Calendar' table:
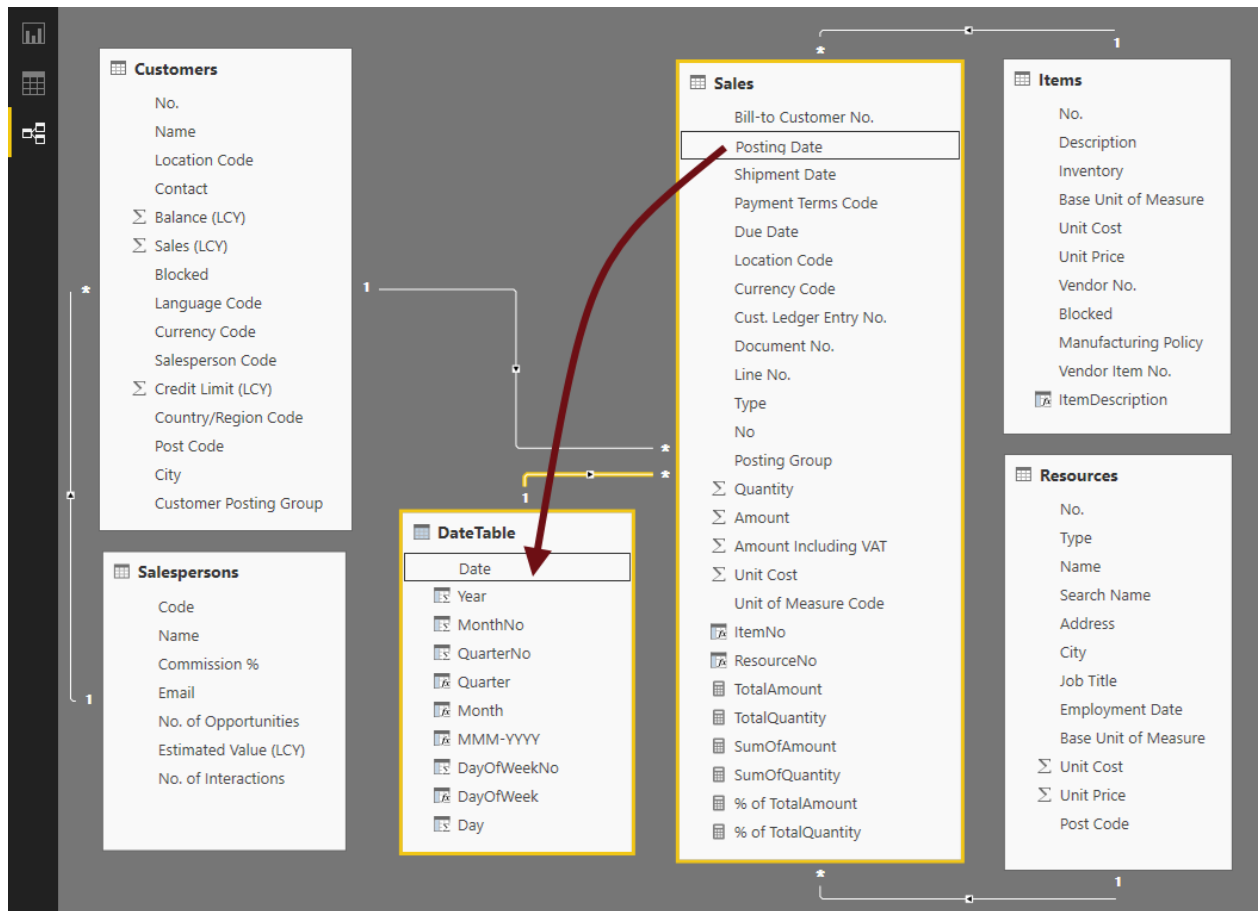
Add Extra Measures

Now that we have a DateTable dimension, we can use it to create some extra measures:

Cumulative Sales (Selected)

- Cumulative Sales (Selected) =
  CALCULATE([SumOfAmount],FILTER(ALLSELECTED(DateTable),DateTable[Date] <=
  MAX(DateTable[Date])))

Cumulative Sales LY

- Cumulative Sales LY = CALCULATE([Sales LY],FILTER(ALLSELECTED(DateTable),DateTable[Date] <=
  MAX(DateTable[Date])))

Cumulative Sales 2 Years Prior

- Cumulative Sales 2 Years Prior = CALCULATE([Sales 2 Yrs
  Prior],FILTER(ALLSELECTED(DateTable),DateTable[Date] <= MAX(DateTable[Date])))

Sales LY

- Sales LY = CALCULATE([TotalAmount],SAMEPERIODLASTYEAR(DateTable[Date]))

Sales 2 Yrs Prior

- Sales 2 Yrs Prior =
  CALCULATE([TotalAmount],SAMEPERIODLASTYEAR(SAMEPERIODLASTYEAR(DateTable[Date])))

Total Transactions

- Total Transactions = COUNTROWS (Sales)

Last Sale Amount

- Last Sale Amount = CALCULATE([TotalAmount] , LASTNONBLANK(DateTable[Date] , [TotalAmount]))

Last Sales Date

- Last Sales Date = LASTNONBLANK(DateTable[Date] , [TotalAmount])

Provide Metadata

In order to help us automate some visualisations when creating a report, you will now add meta data to the data model. You do this by selecting a Data Category for a field.

For example, go into the Customers table, select the Post Code column and then select Data Category and put it on Postal Code:
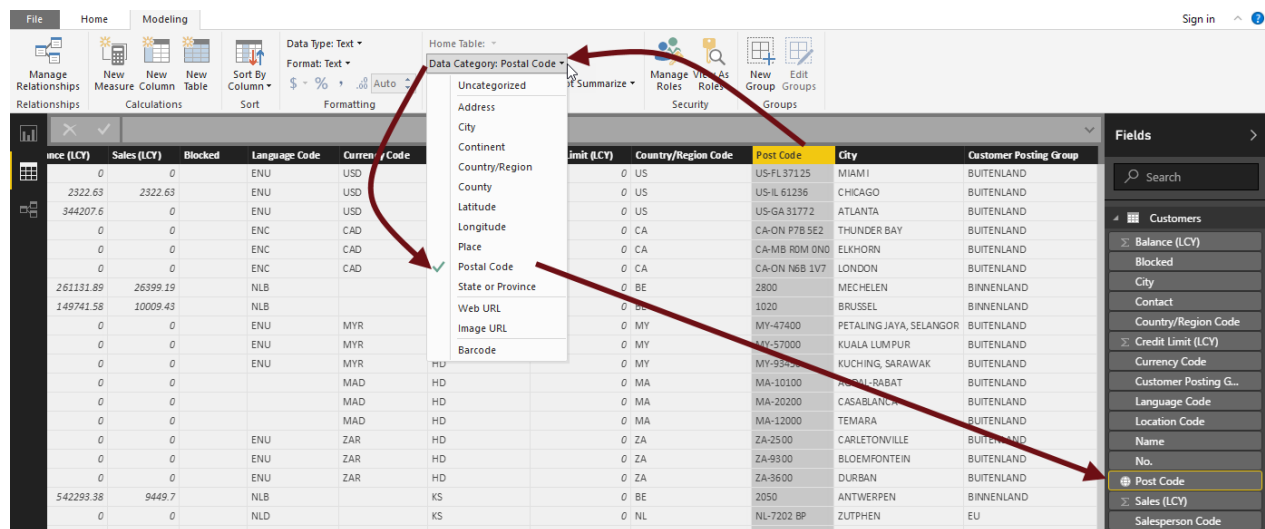
Similarly, select the City category for the City field.

Then for fields that do not need to be summarized, for example the Customers Credit Limit, select Do Not Summarize:

Save Power BI Desktop file

Save your .pbix file.