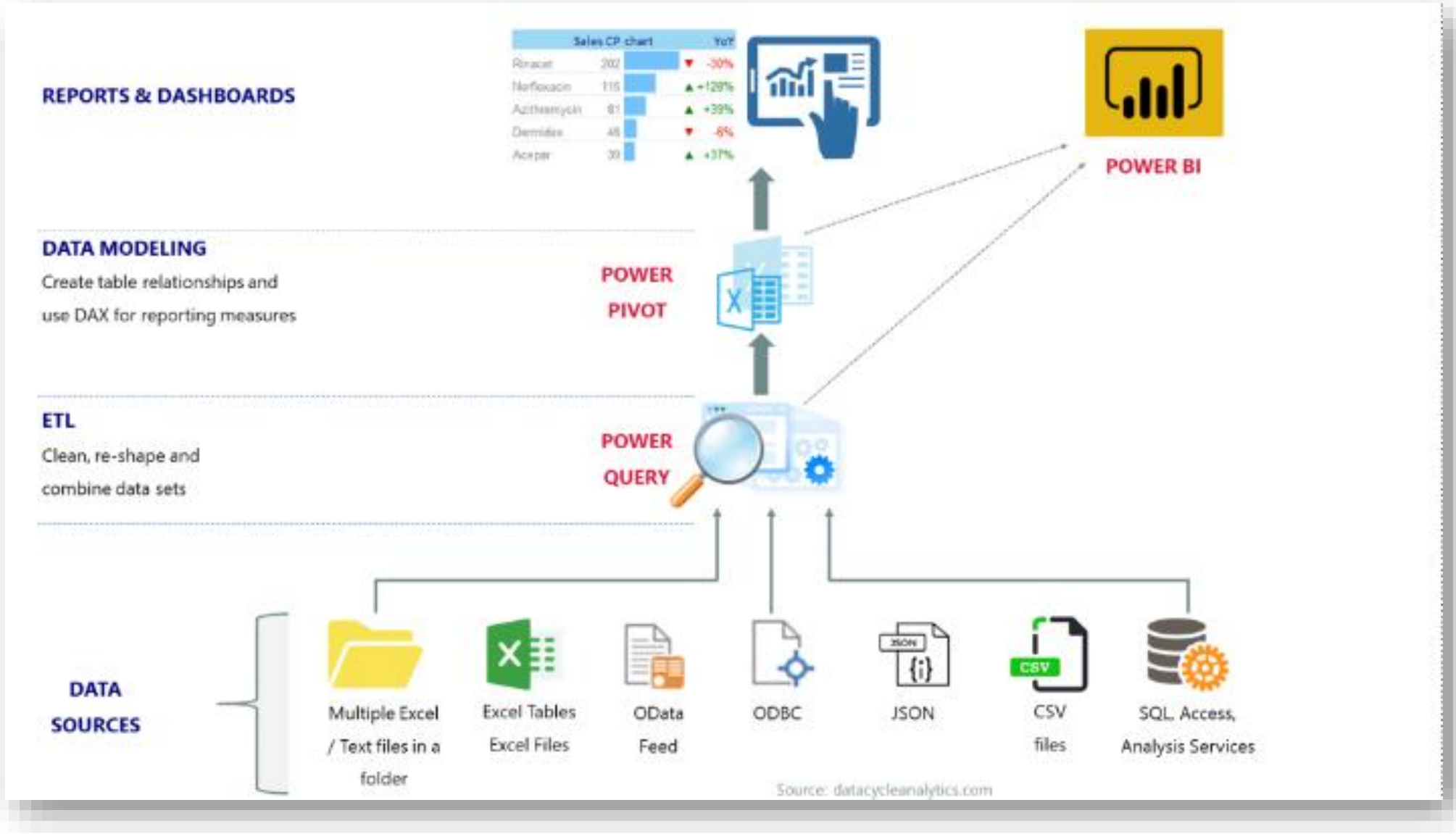
Power BI Desktop - *Deep dive Query Editor*

Barbara Berghaus IT1.4.2.1

Last updated: 10.09.2019

Agenda

Overview


REPORTS & DASHBOARDS
Sales CP chart
YoY
Rinacet 202 -30%
Norfloxacin 115 +128%
Azithromycin 81 +39%
Dermidex 48 -6%
Acepar 39 +37%
POWER BI
DATA MODELING
Create table relationships and
use DAX for reporting measures
POWER
PIVOT
ETL
Clean, re-shape and
combine data sets
POWER
QUERY
DATA
SOURCES
Multiple Excel
/ Text files in a
folder
Excel Tables
Excel Files
OData
Feed
ODBC
JSON
CSV
files
SQL, Access,
Analysis Services
Source: datacycleanalytics.com

1 Working with the Query Editor – Technical Deep dive

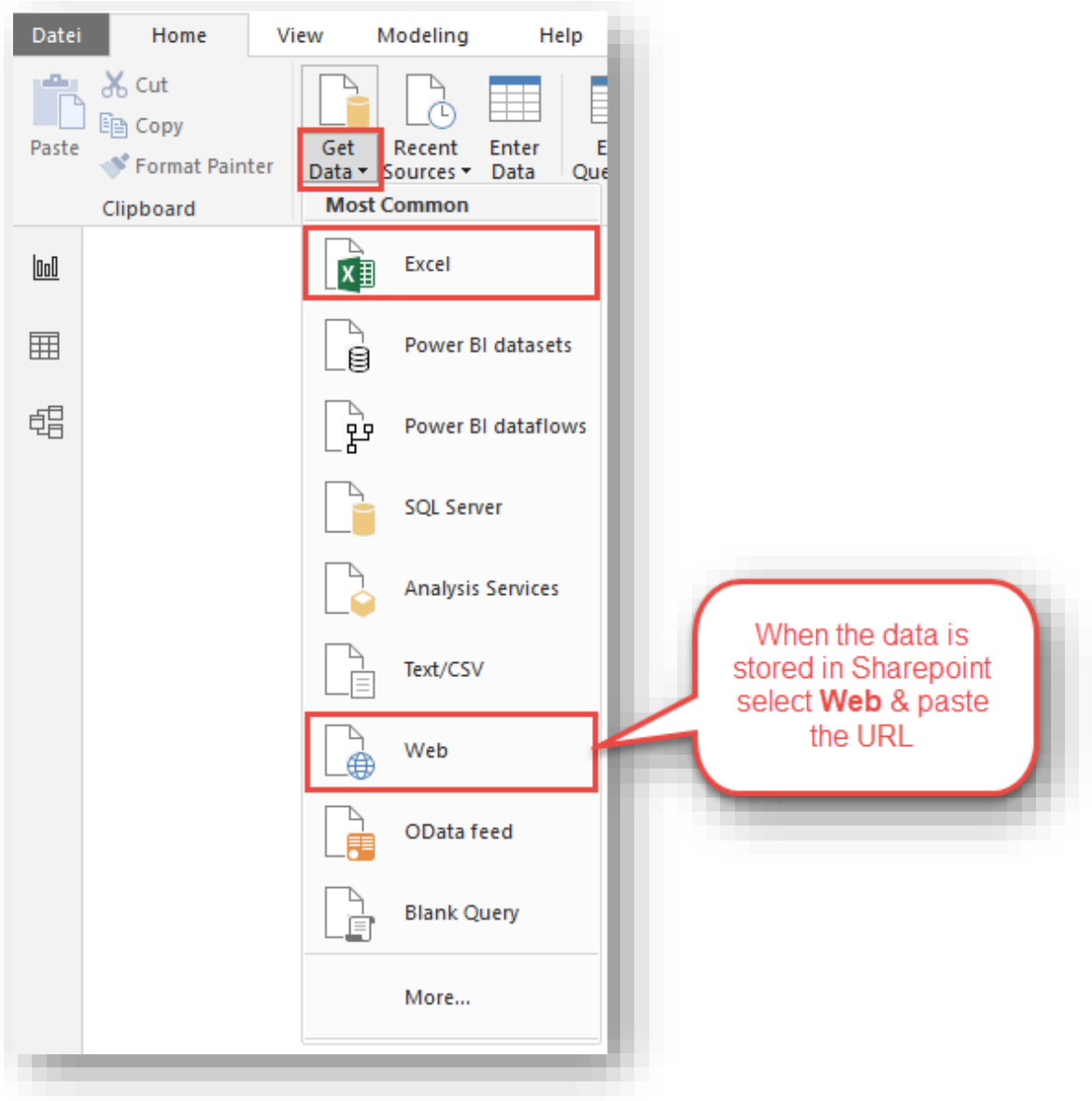
Why should you deal with the Query Editor?

1.1 Basic Principles

Exercise 1

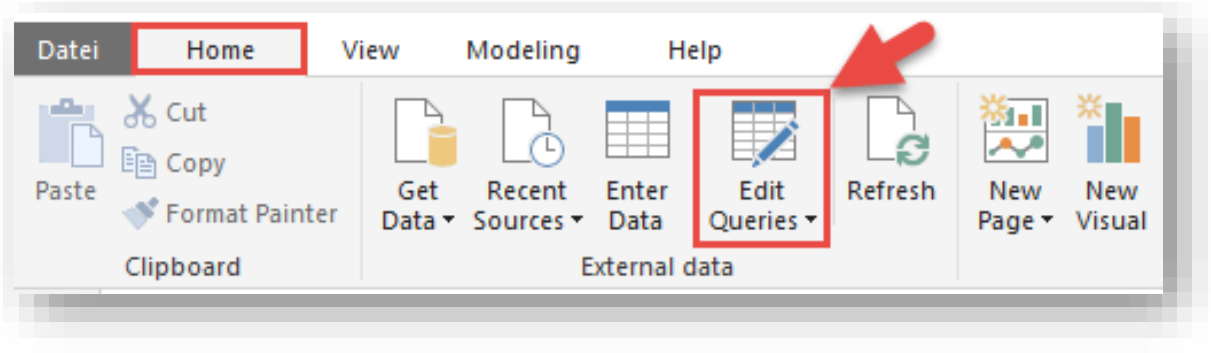
Get Data

- First step for a new model is getting data. There are many possible file types which Power BI can detect and standardize.
- Most useful connectors are: XLSX / CSV / TXT / PDF
- Unfortunately, XLS files are not supported. (please convert to XLSX!)
- In order to create a new source, click on the ***Get Data*** on the top toolbar
- Choose ***Excel*** …
- … & open Premiums Q4.xlsx
- Select the table name Premiums_Q4 & proceed with **Edit**



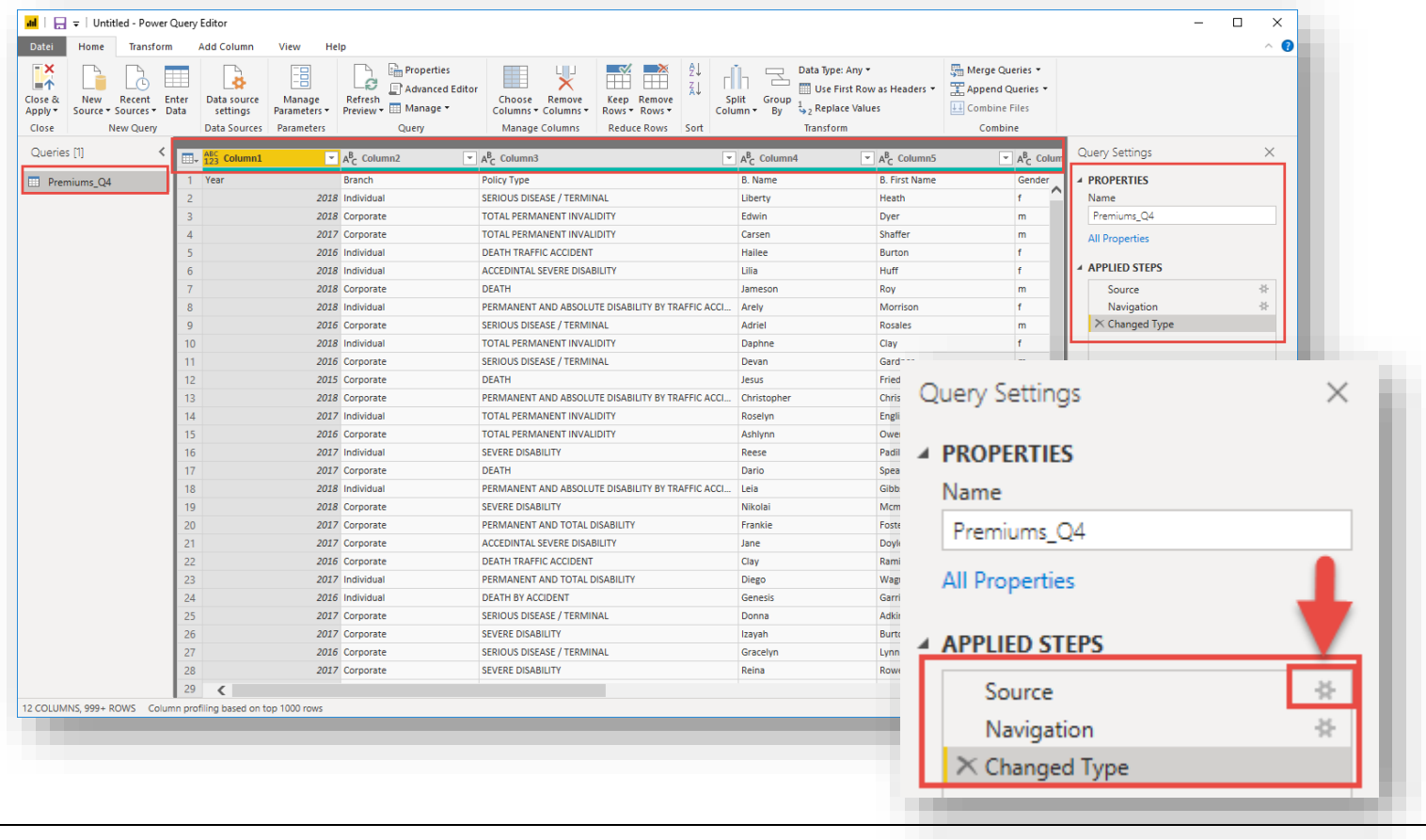
- The ***Query Editor*** is automatically opened when data is loaded, but it is also possible to access through the toolbar

- The ***Query Editor*** allows modeling, transforming and enriching data prior to loading it into the file



Query Editor: Applied steps

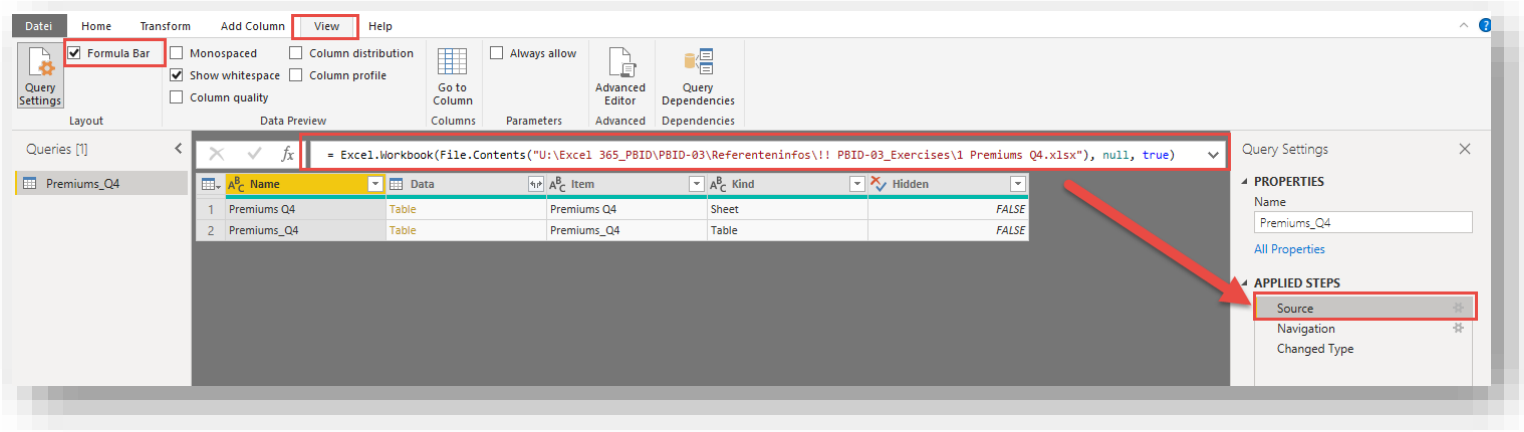
- Each modification to the file is recorded under the ***Applied Steps*** pane on the right
- It is possible (but not always recommended) to reorder steps by dragging them
- It is possible to edit each step, by clicking the ⚙ icon
- It is possible to see how the data looks in each step just by clicking it

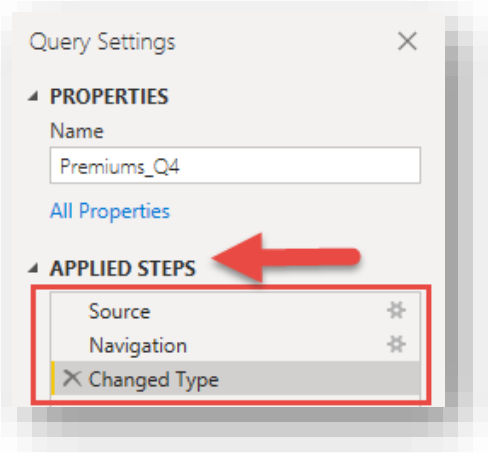


- Each step translates into a formula in the language **Power Query** (or **M**), and is visible under the toolbar
- Please note:
  Power Query (M) is case-sensitive!

When importing most file types, by default 3 steps are created:

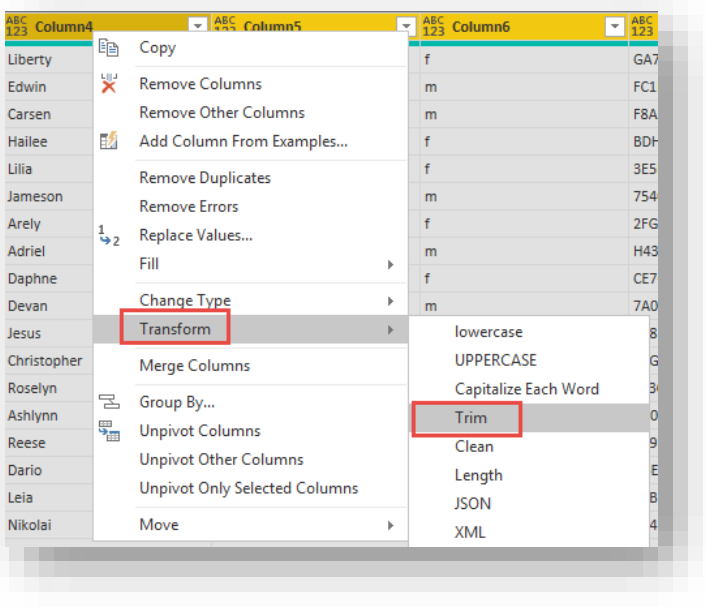
- ***Source*** – file path / URL to source, type of file (Excel, PDF, CSV etc.)
- ***Navigation*** – which page or excel sheet?
- ***Changed Type*** – automatically detects the column data type (date, decimal number, text etc.)





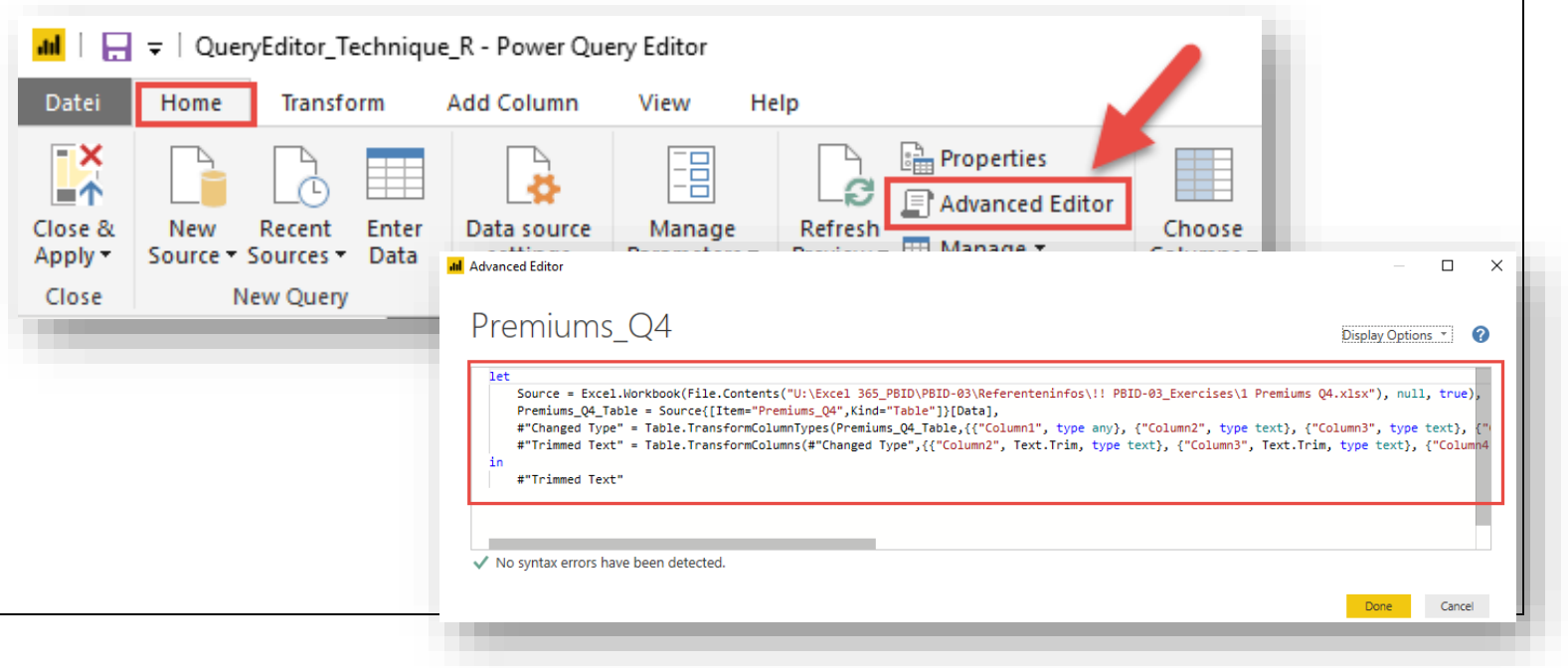
Exercise 2

- It is recommended to always **trim text columns** to avoid problems with spaces
- Additionally: when deleting duplicates: convert everything to ***lowercase / UPPERCASE*** so that there are no problems when deleting duplicates due to the distinction between ***UPPERCASE*** & ***lowercase***.



Exercise 3

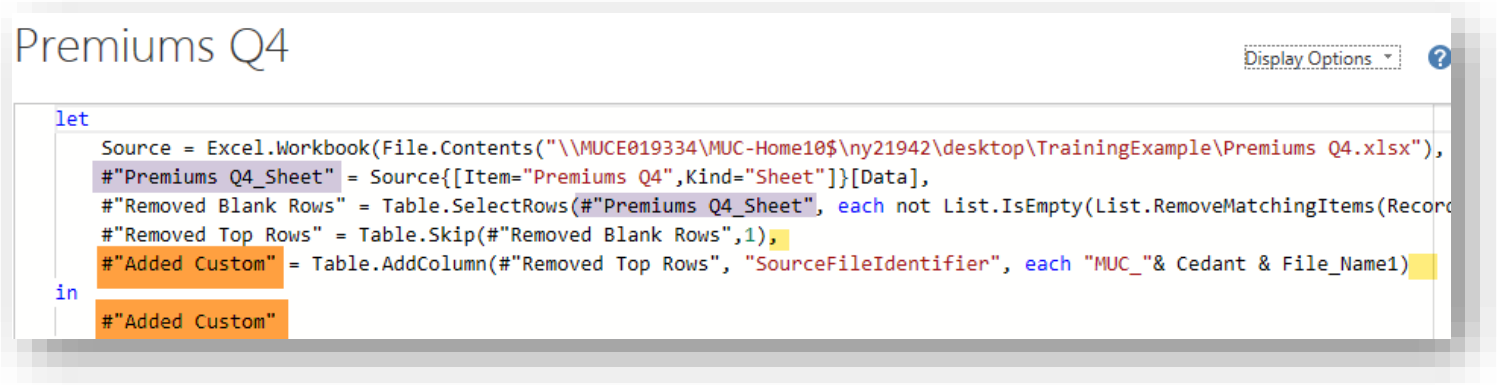
- Working with the Advanced Editor
- Each modification adds a step to the *Applied Steps* pane on the right, which translates into a new line of code in the *Advanced Editor*
- Each line of code represents a step



- Conceptually, each step contains a version or state of the table
- Each version is saved into a variable: e.g. Source, #"Changed Type"
- Variables usually are named with quotes and a hashtag, but not always
- Previous versions of the table (which are saved in variables) can be referenced & used in each step

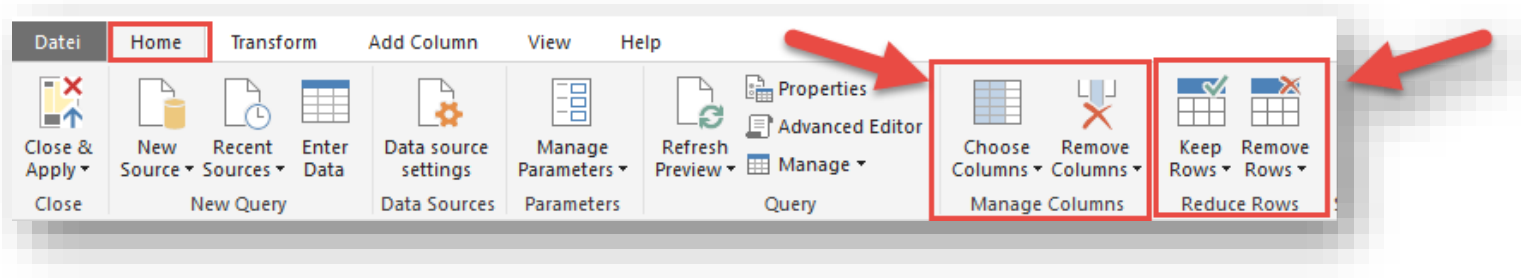
The final version of table is passed in the "**in**" section

- The name of the variable of the predecessor line has to be repeated in the following line
- Each line is closed with a comma, only the last line in the "*let*"-block has no comma
- The name of the last variable in the "*let*"-block is the name of the step & thus has to be repeated in the "*in*"-block



Common functions

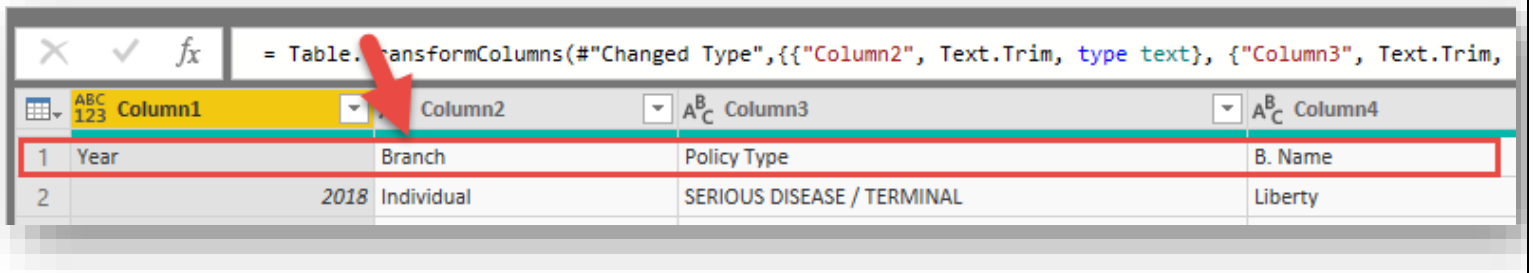
- **Remove / Choose Columns** – select which columns to include or exclude from the table
- **Remove Rows** – remove x top or bottom rows, remove empty rows

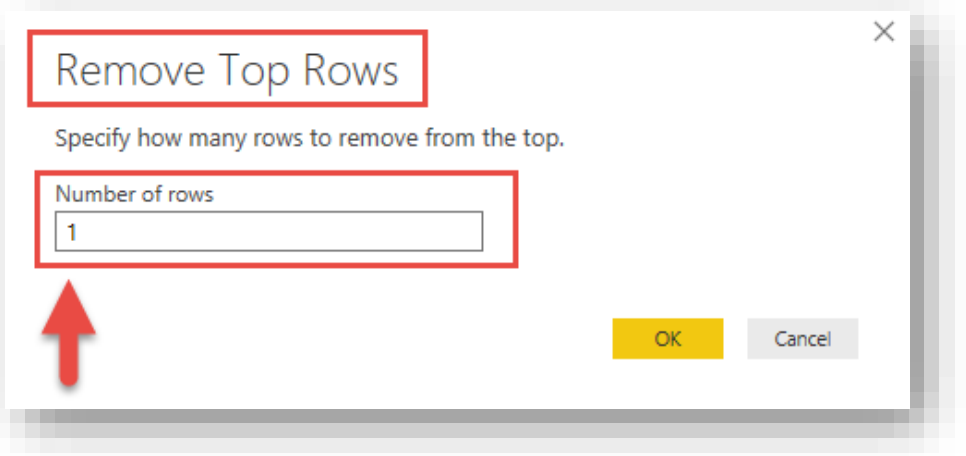


Exercise 4

Delete the first row

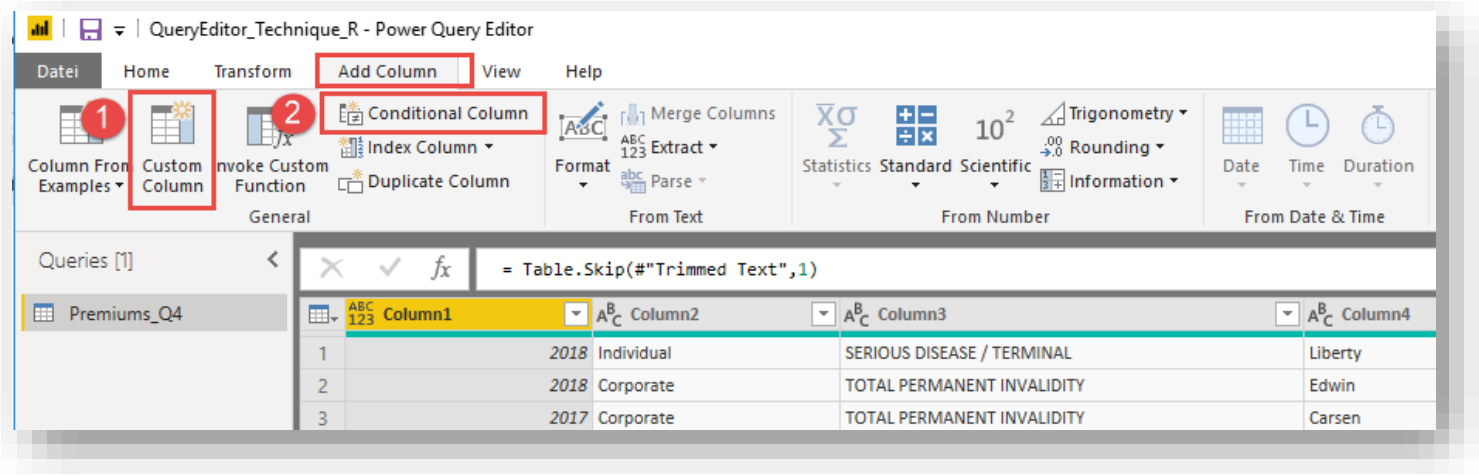
- Titles of the imported files might change
- Therefore, you can avoid problems with column titles using the standard titles like *Column 1, Column 2*, …
- In our example please delete the first row with the original column titles

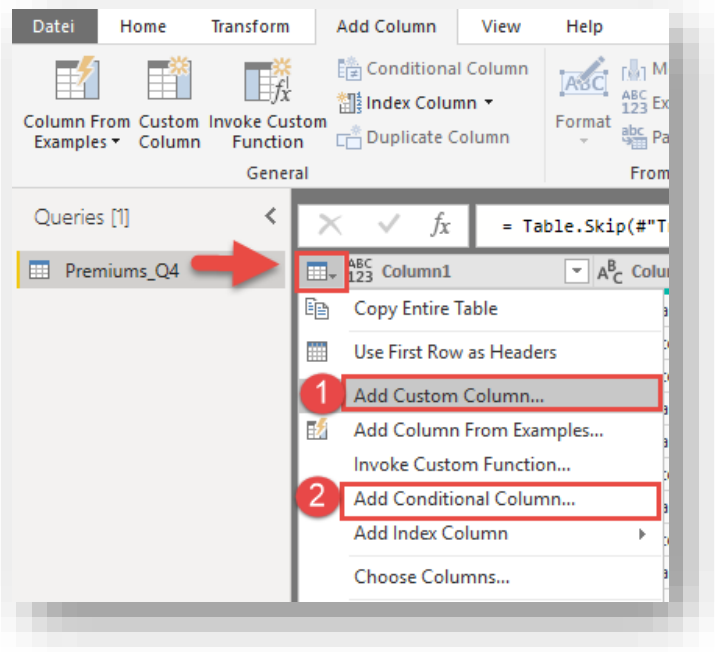




- **1 Custom Column** – creates a new column with data from an expression

- **2 Conditional Columns** – creates a new column with data which follow the inserted criteria



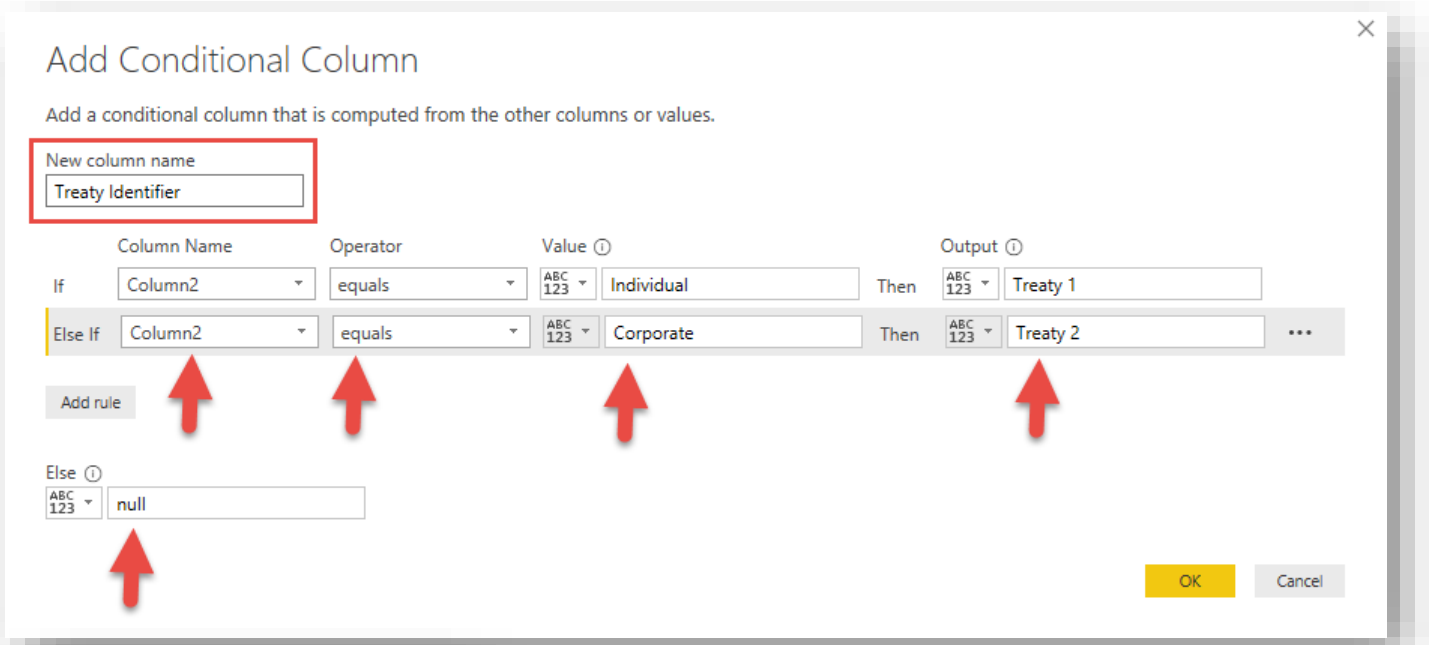


Exercise 5

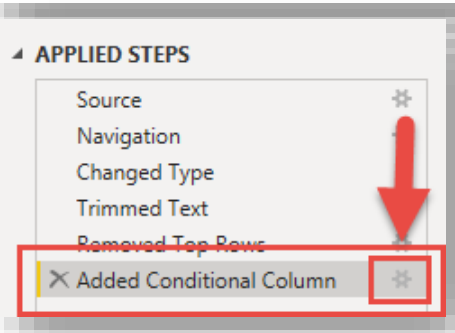
Create a conditional column

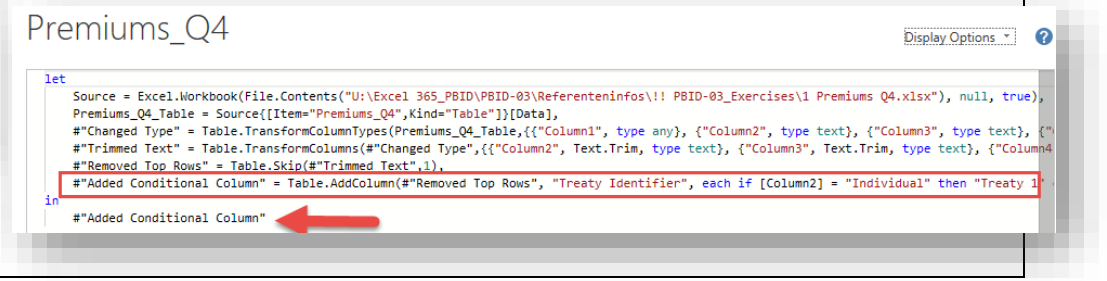
- We assume that Treaty ID is determined by the Branch column
- Please add a ***conditional column*** accordingly

Cedant Treaty Id	MR Treaty Id
Individual	Treaty 1
Corporate	Treaty 2



- A ***new step*** will be added, and a ***new line*** of code in the *Advanced Editor*

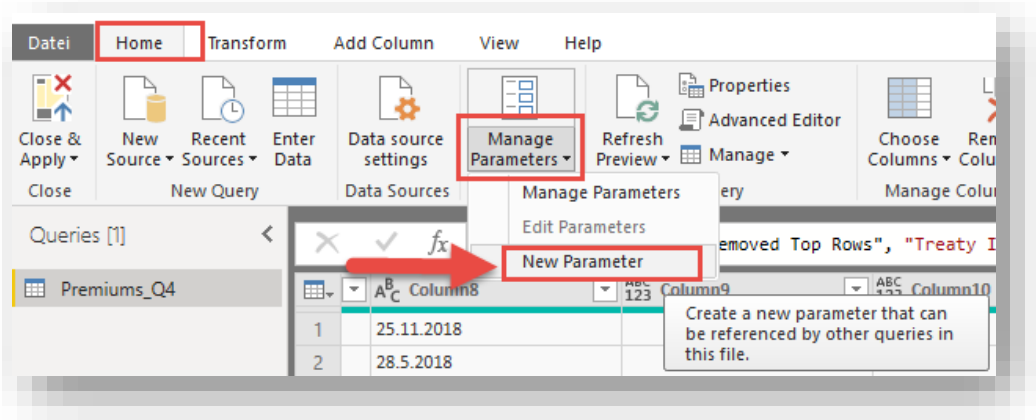


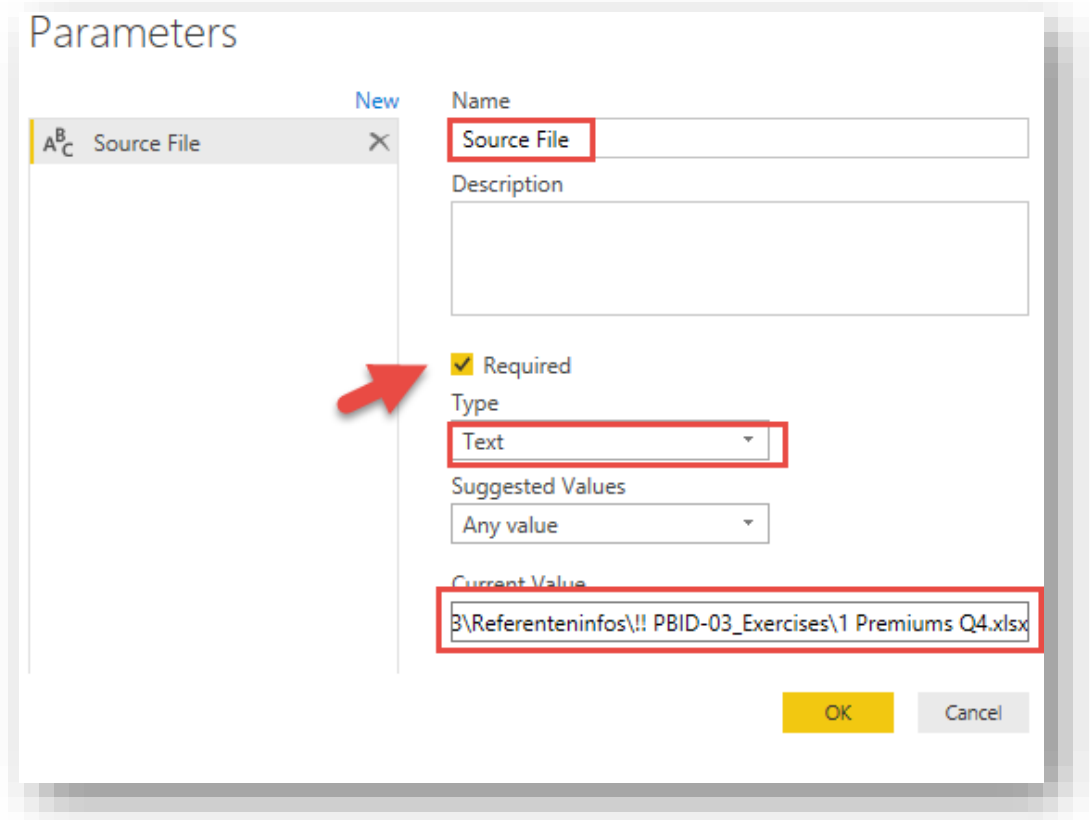


Exercise 6

Using Parameters

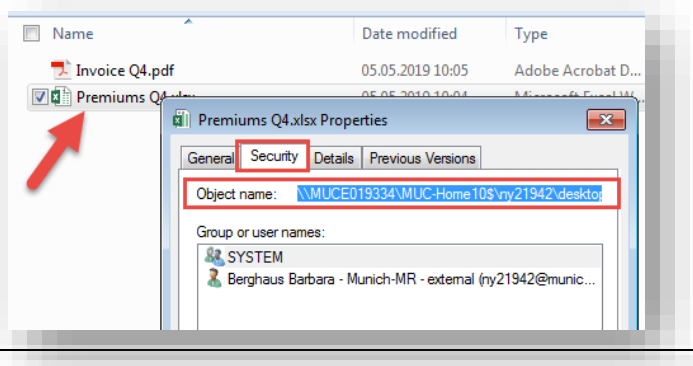
- Each part of the code in the ***Advanced Editor*** can be replaced with a parameter - for example, changing the source file path, so the same model can use dynamically different files
- A new parameter can be defined under ***Manage Parameters***
- Usually ***Type***" should be Text
- The option ***Required*** is activated by default – it should be deselected for parameters when an information is not always available



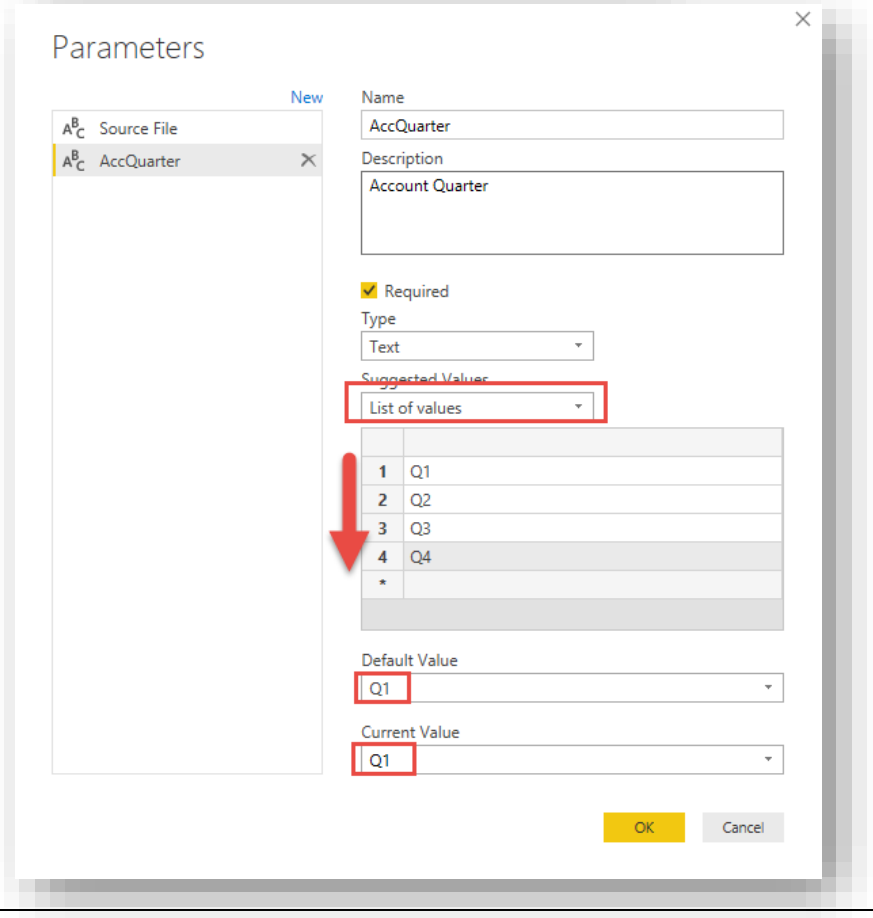


The file path can be copied:

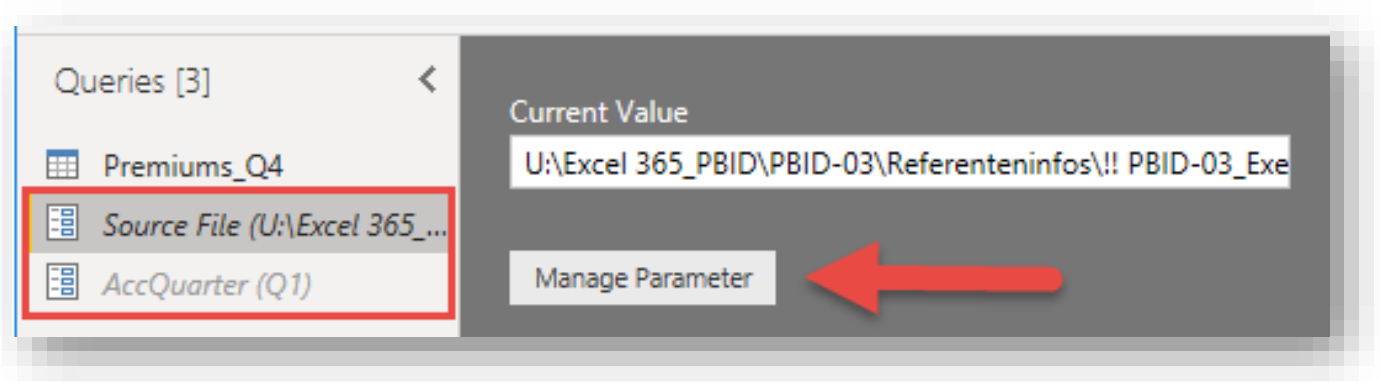
- Rightcklick Premiums Q4.xlsx & open ***Properties***
- Switch to ***Security*** & copy the file path

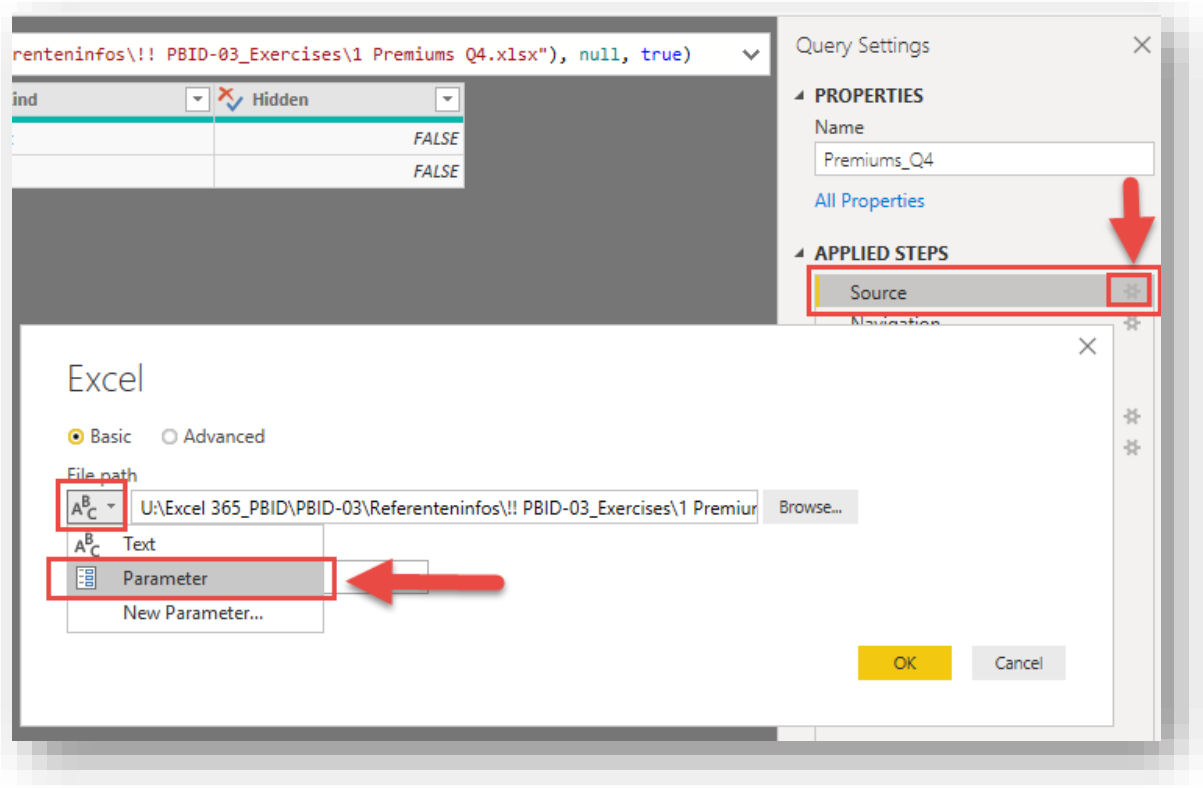


- It is also possible to provide a list of values, when you want to make sure, that the entries from the user are always correct
- For example the entry for the column **Acct Quarter** should always be **Q4** – therefore, the option ***List of values*** should be choosen here

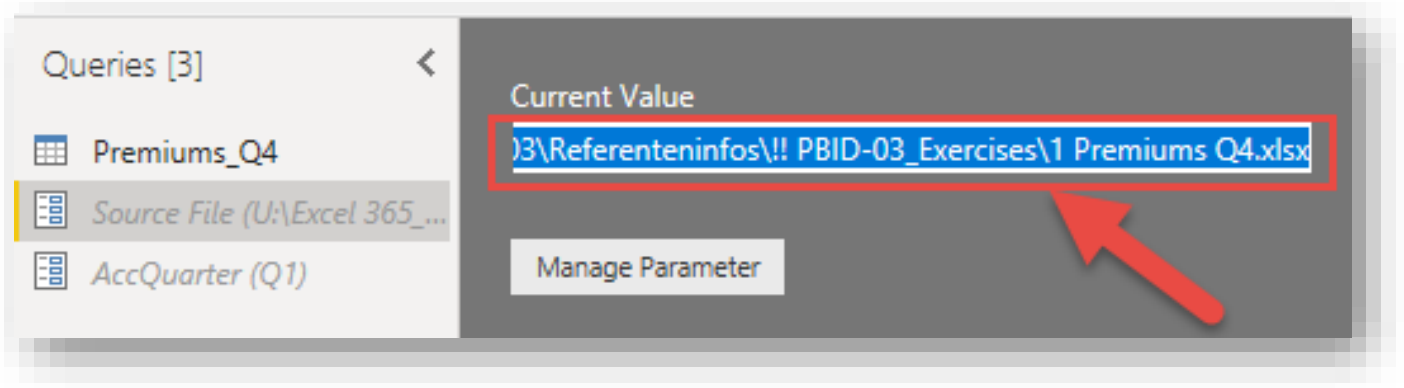


- Creating a parameter will automatically add it under the queries pane
- Note that Parameters are just like any other query in Power BI Desktop; they can be referenced from other queries, loaded to the Data Model and referenced in DAX expressions, etc.
- By default though, Parameters are not loaded to the Data Model
- To incorporate the parameter in the query, click on the cogwheel in the step Source





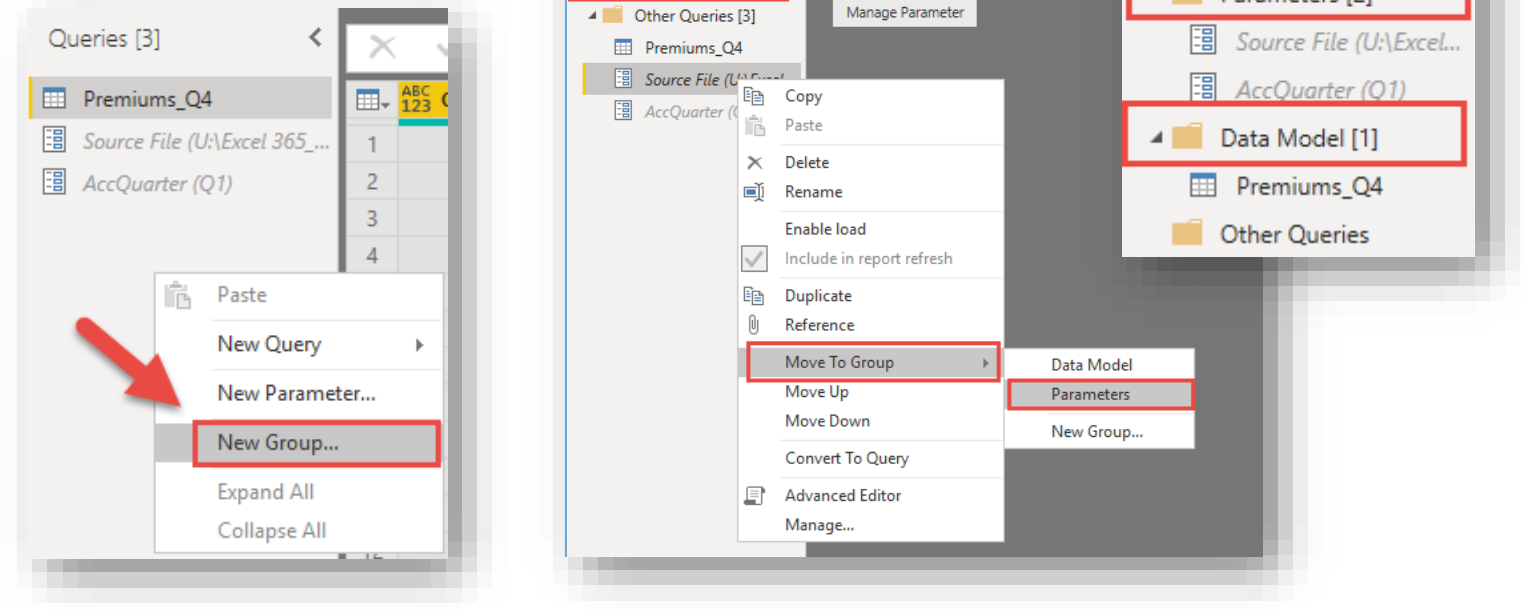
- When using a new source file, instead of editing the code, just change the parameter value
  - or use a template (see next exercise)

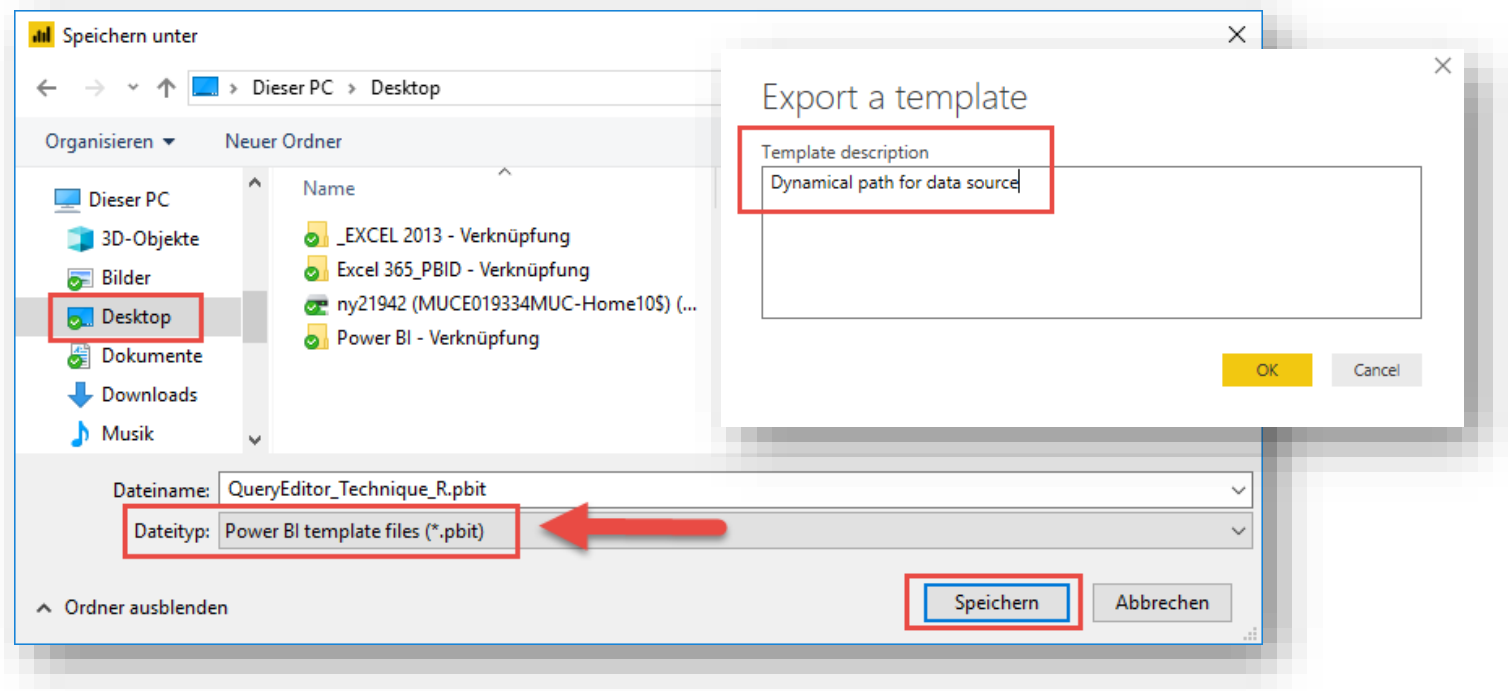


Organize the entries

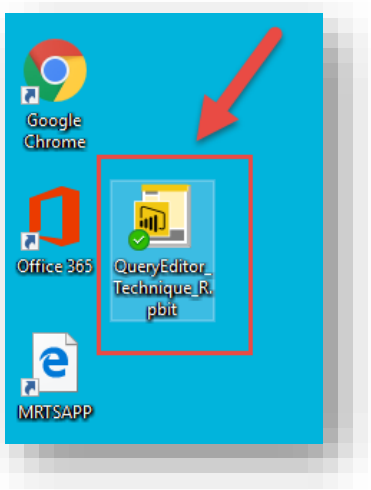
- With ***Groups*** you can organize your queries and parameters

- Save the **PBIX**-file



Using templates	• A Power BI Report Template contains the definition of the Report (pages, visuals, etc.), Data Model definition (schema, relationships, measures, etc.) and Queries definition (collection of queries, including Query Parameters, etc.) • In other words, a Power BI Report template includes pretty much everything that a Report file includes, with the exception of the data itself • When a user tries to instantiate a template, either via double-click on the PBIT file, or by using the File / Import / Power BI Template path, a new Power BI Desktop file will be created, containing the actual data based on current user's credentials for data sources, etc.
Exercise 7 • Save the **PBIX**-file as a template (**PBIT)** on the desktop • Enter a Template description	

- Close the PBIX-File & open the **Template**



- A parameter dialog is shown where the user can enter the new path for the data source & other information …
- Then a **new PBIX-file** with the selected data source is opened
- All ***Applied steps*** from the ***Query Editor*** are then processed on this new data

2 Example – Data Preparation from PDF

- Example data

2 Invoice Q4.pdf
2 CEDANT_2018_Q4.pbix

The starting point

- You receive new quarterly figures from your cedant
- The data arrive as **XLSX-files** & as **PDF-files**
- The XLSX file is already shaped & transformed to the needed format
- Now open the PDF file

The task

1 The data from both files have to be appended – therefore, the PDF file has to be shaped accordingly

2 With a template it should be possible to enter a file path & other basic information for the PDF file

	A	B	C	D	E	F	G	H	I	J	K
1	SourceFileIdentifier	TreatyIdentifier	AUIdentifier	Entry Code	Amount	Acct Text	Acct Year	Acct Quarter	UY	OY	Currency
2	MUC_COMPANY_ABC_PREMIUMS	Individual	SERIOUS DISEASE	10199	178.91	LH_2018Q4_C	2018	Q4		2018	EUR
3	MUC_COMPANY_ABC_PREMIUMS	Individual	SERIOUS DISEASE	10099	35.78	LH_2018Q4_C	2018	Q4		2018	EUR
4	MUC_COMPANY_ABC_PREMIUMS	Corporate	TOTAL PERI			C	2018	Q4		2018	EUR
5	MUC_COMPANY_ABC_PREMIUMS	Corporate	TOTAL PERI			C	2018	Q4		2018	EUR
6	MUC_COMPANY_ABC_PREMIUMS	Corporate	TOTAL PERI			C	2018	Q4		2017	EUR
7	MUC_COMPANY_ABC_PREMIUMS	Corporate	TOTAL PERMANENT			C	2018	Q4		2017	EUR
8	MUC_COMPANY_ABC_PREMIUMS	Individual	DEATH TRAFFIC AC	10199	100.57	LH_2018Q4_C	2018	Q4		2016	EUR

Shaped XLSX-File

GENERIC COMPANY

Generic Company ABC Ltd.

12/03/2018

Branch INDIVIDUAL
Model SEVERE DISABILITY
Period 4TH QUARTER 2018

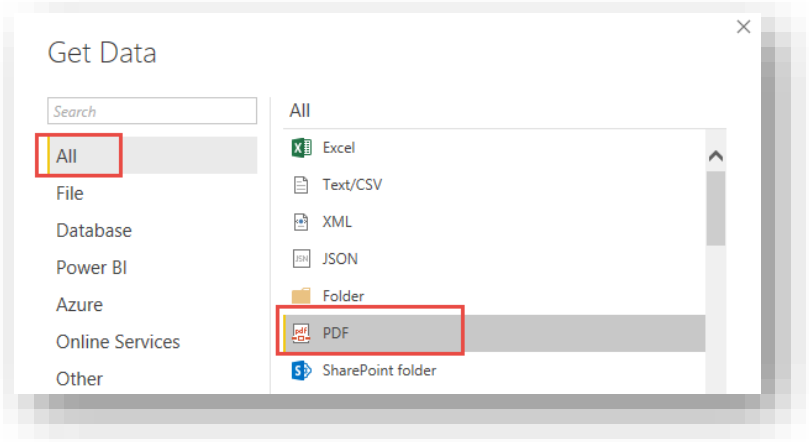
PDF-File

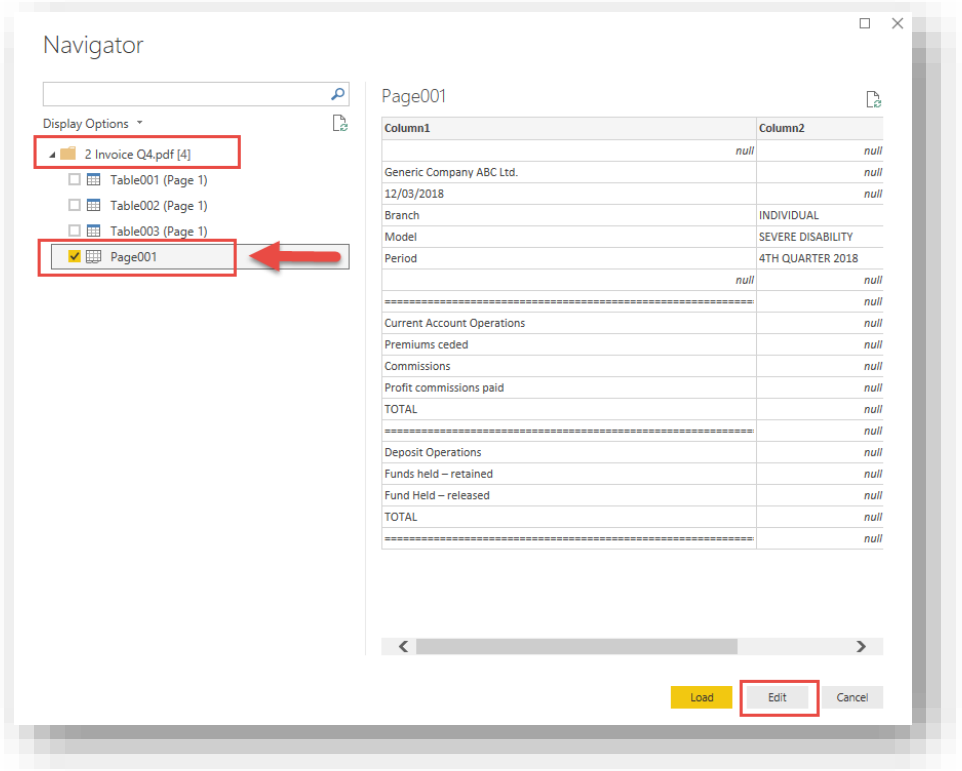
	DEBIT	CREDIT
Current Account Operations		
Premiums ceded		15.984,36
Commissions	590,03	0,00
Profit commissions paid	1.485,90	0,00
TOTAL	**2.075,93**	**15.984,36**
Deposit Operations		
Funds held – retained	6.497,84	
Fund Held – released		9.371,38
TOTAL	**0,00**	**2.873,54**

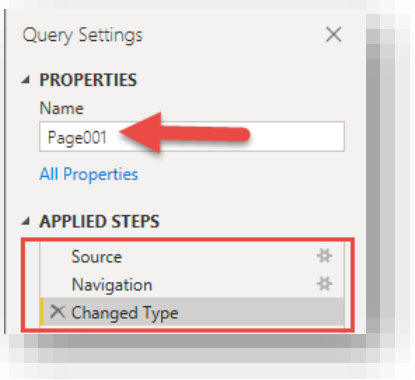
Exercise 1

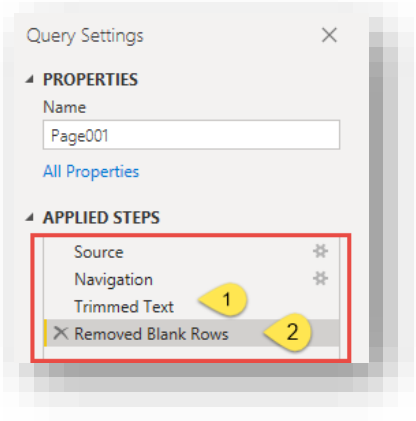
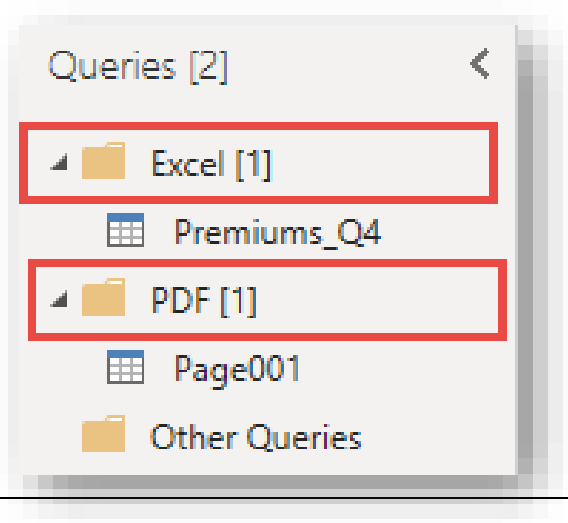
- Open 2 CEDANT_2018_Q4.pbix
- The data of the XLSX file is already imported
- Import 2 Invoice Q4.pdf with the ***PDF connector***
- Similar to importing .XLSX files, it is required to select a page to import

- The PDF connector auto-detects tables within the page itself
- It is not recommended using tables, rather selecting an entire page

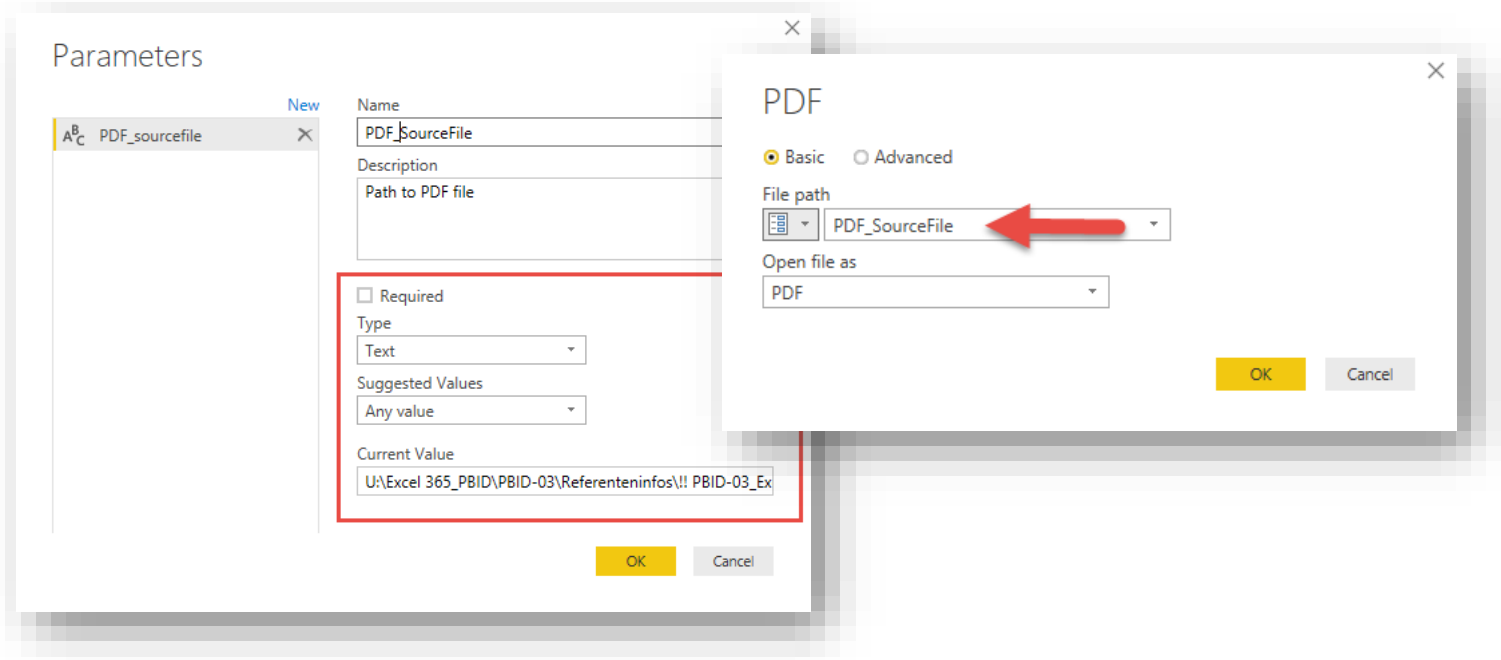






• **Delete step** Changed Type • **Trim** text columns • **Remove** blank rows	
Organizing queries & parameters • As there are two different files in the model, you should create ***Groups*** for each file & parameters • Right click on the ***Queries left pane / New Group*** • Create the following groups … • … & move the queries (& later the parameters) as shown in the screenshot	

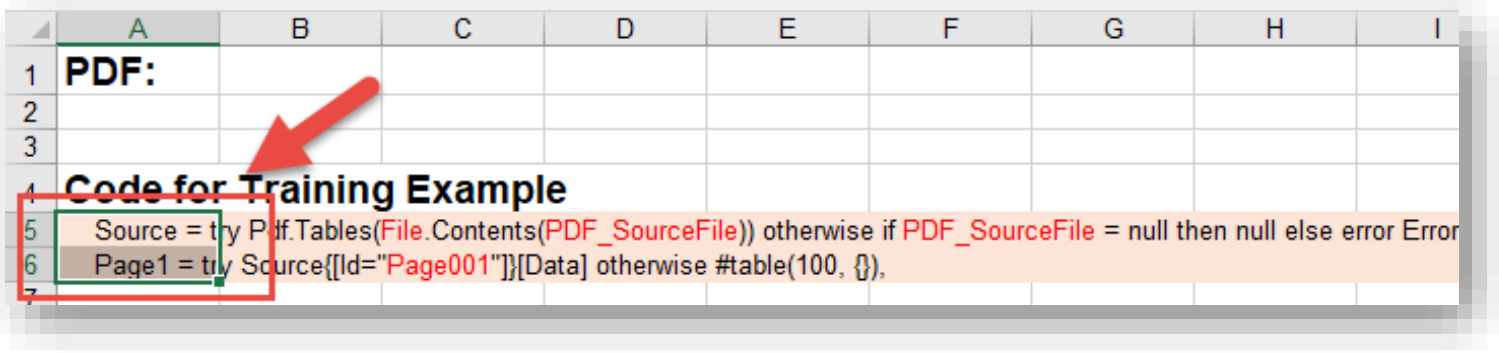
- Create a parameter for the source file: PDF_SourceFile
- The option ***Required*** is not selected!
- Connect the parameter to query Page001



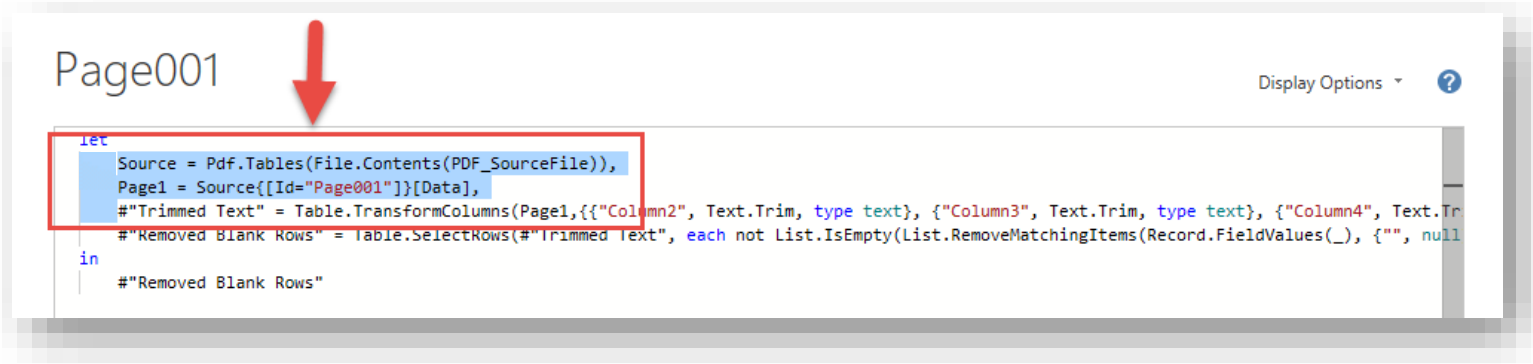
Flexible Parameter Code for source data

- Open the Excel file **Flexible Parameter_Code for PBI.xlsx**
- Highlight the first two cells of code as shown in the screenshot
- Copy the necessary code
- The name of the **parameter** (PDF_SourceFile used for the data source) is already inserted here

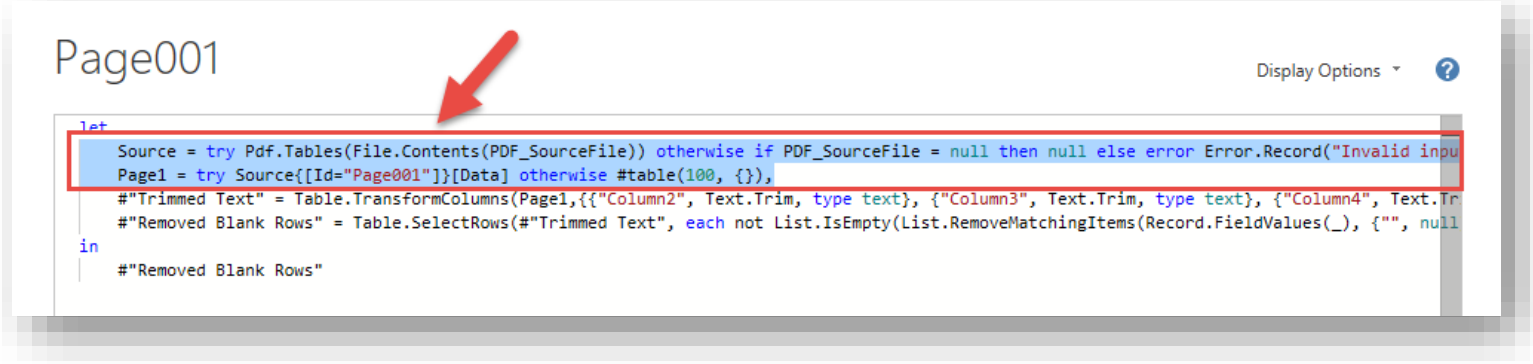
- It is necessary, that the parameters you created don't require an input mandatorily – that is because in some cases there is no PDF file
- Therefore, you have to adapt the code in the Advanced Editor



- Open the ***Advanced Editor*** & highlight the code as shown



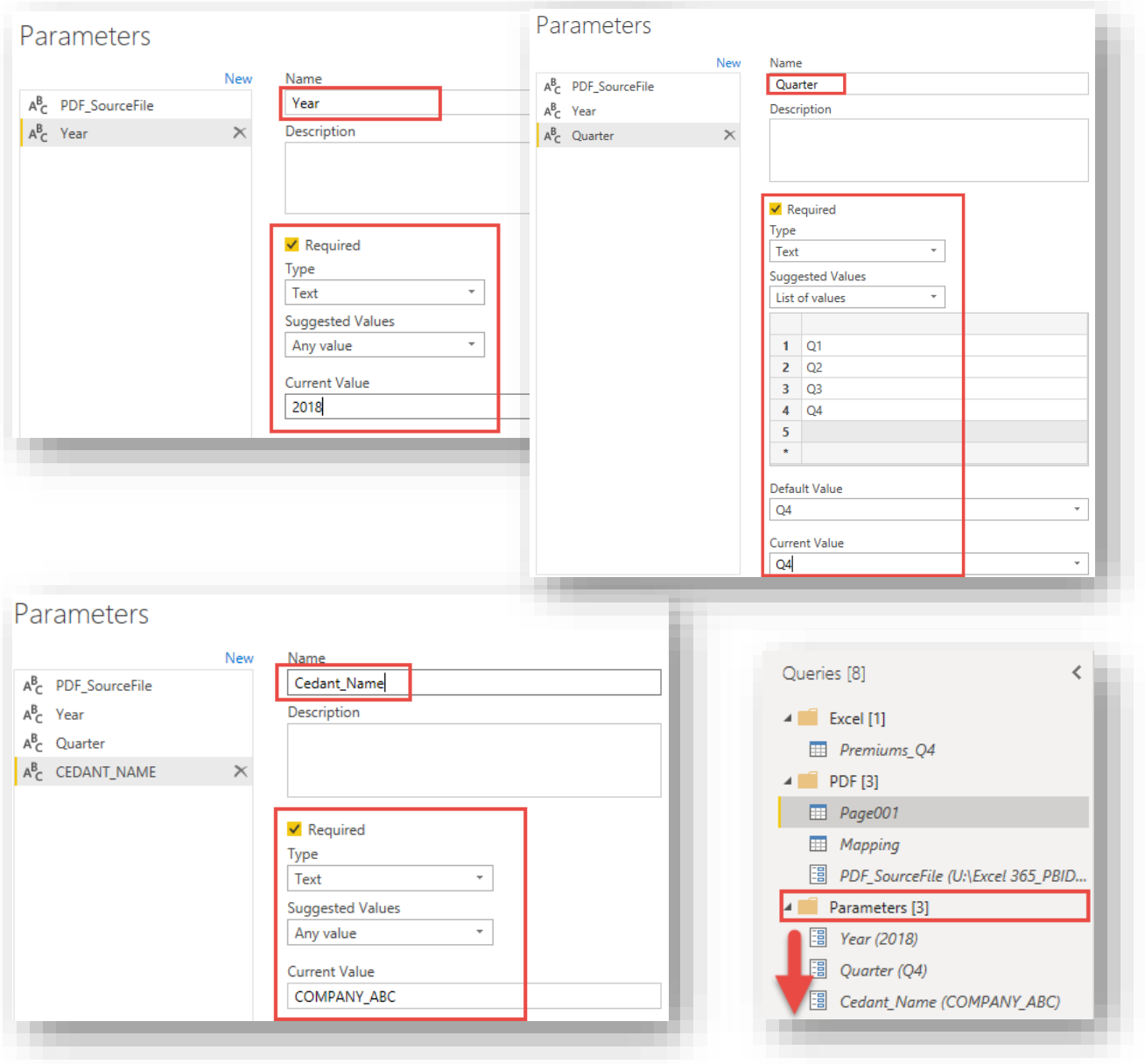
- Replace the two code lines with the new code for the lines Source & Page1
- Close the ***Advanced Editor*** – the query result should be shown as before
- If you now enter e. g. a wrong file path or leave the parameter empty these cases are handled by the changed code



Exercise 2

More parameters are needed

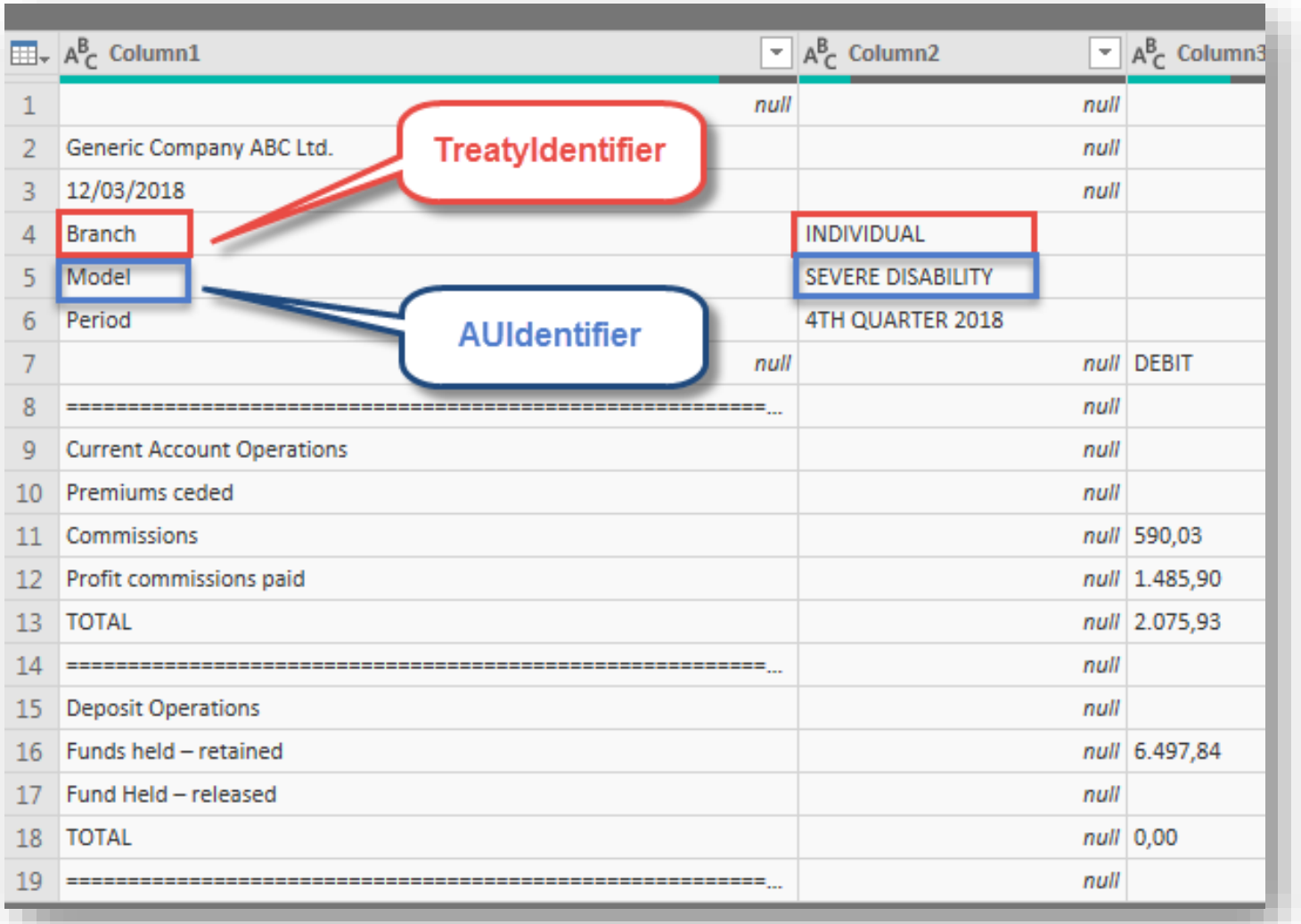
- Year
- Quarter
- Cedant_Name
- Move them to a new group **Parameters**



2.1 Creating the table structure

TreatyIdentifier¤
Treaty·Identifier·as·originally·listed·in·invoice¤
Individual¤

AUIdentifier¤
AU·Identifier·as·originally·listed·in·invoice¤
Individual¤



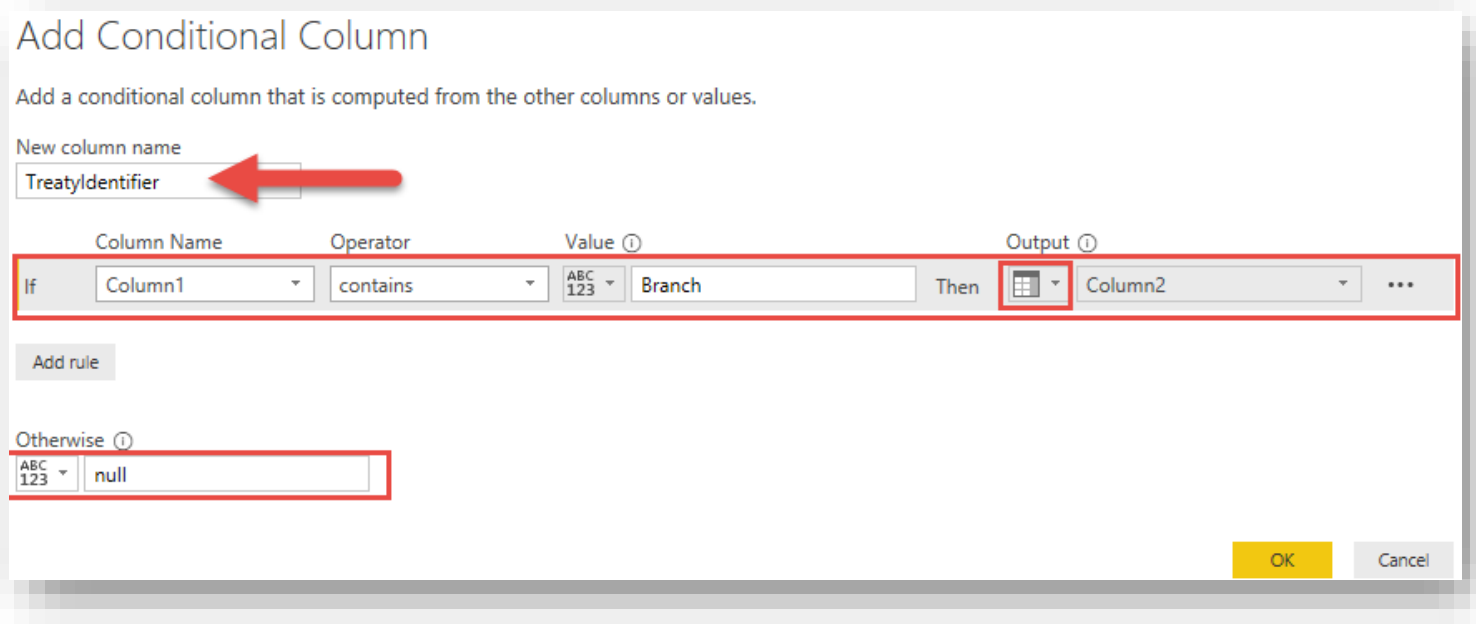
	Column1	Column2	Column3
1	null	null	
2	Generic Company ABC Ltd.	null	
3	12/03/2018	null	
4	Branch	INDIVIDUAL	
5	Model	SEVERE DISABILITY	
6	Period	4TH QUARTER 2018	
7	null	null	DEBIT
8	==========	null	
9	Current Account Operations	null	
10	Premiums ceded	null	
11	Commissions	null	590,03
12	Profit commissions paid	null	1.485,90
13	TOTAL	null	2.075,93
14	==========	null	
15	Deposit Operations	null	
16	Funds held – retained	null	6.497,84
17	Fund Held – released	null	
18	TOTAL	null	0,00
19	==========	null	

- **TreatyIdentifier** (Branch) & **AUIdentifier** (Model) are created from columns **1** & **2**

Exercise 3

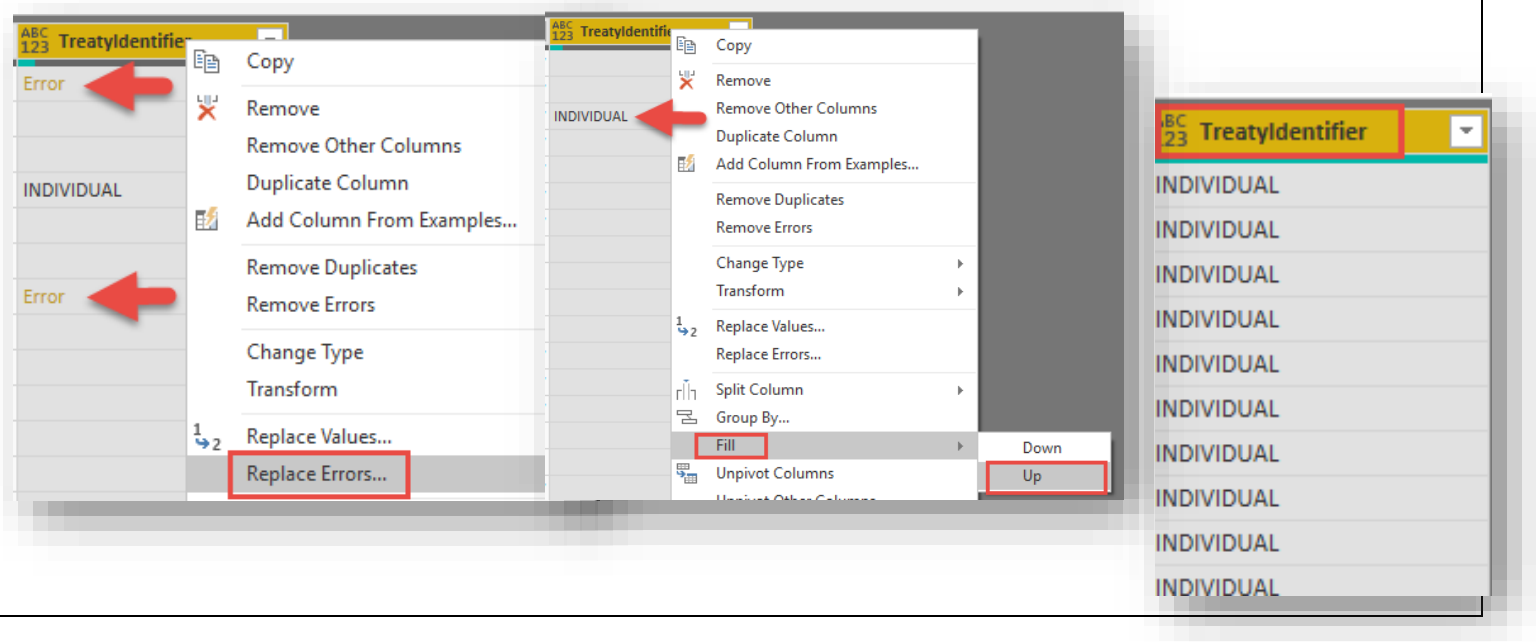
- Create a conditional column

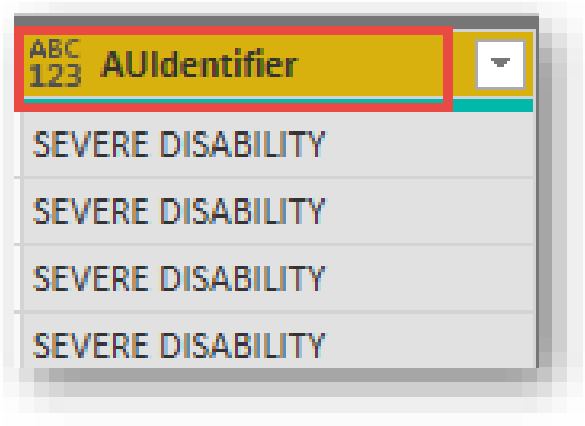
- Tip:
  In this case it is saver to use "*contains*" than "*equals*"



- Right click / ***Replace errors*** with Null
- Right click / ***FillUp / Down***

Result

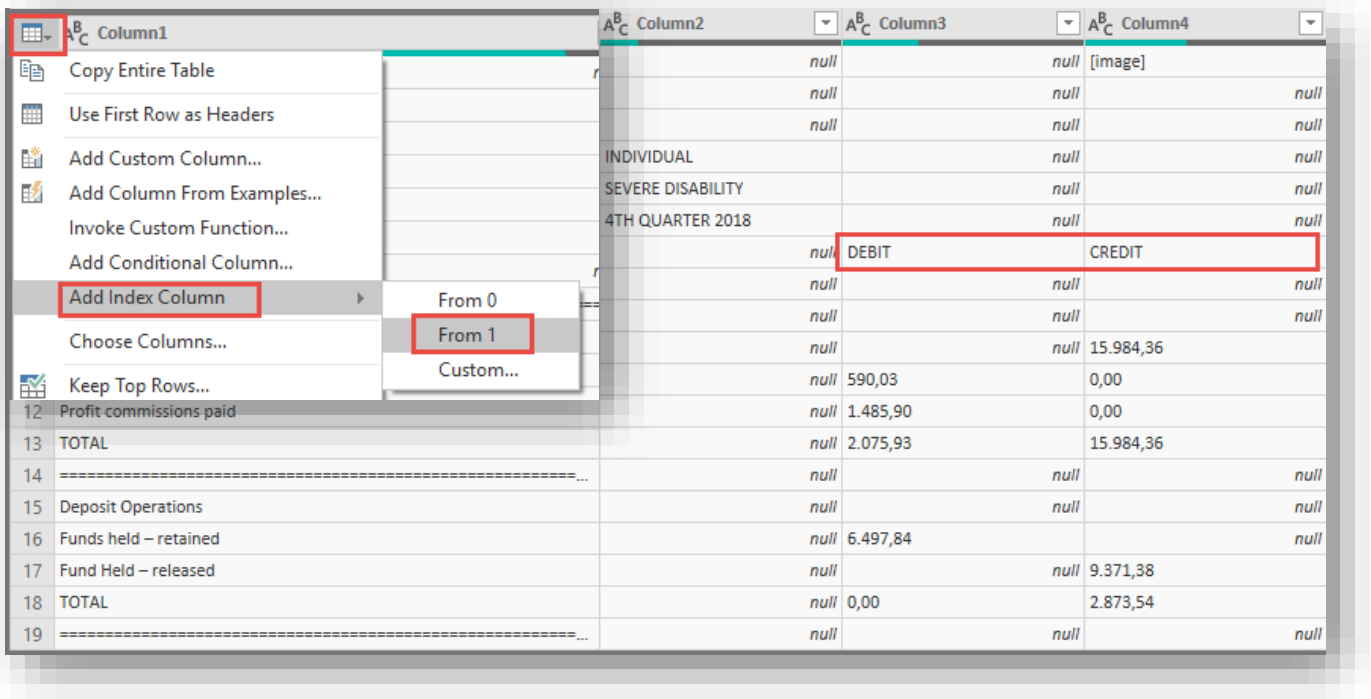


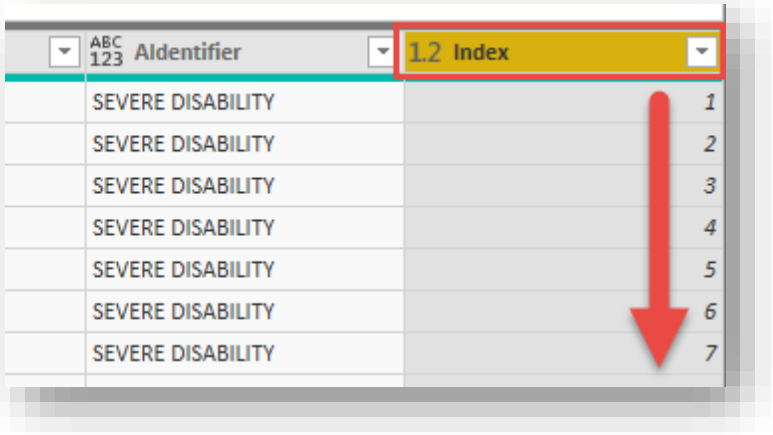
• In the same way create **AUIdentifier**	

Amount¤
Amount·as·originally·listed·in·invoice·or·calculated·percentage·¤
99,9¤

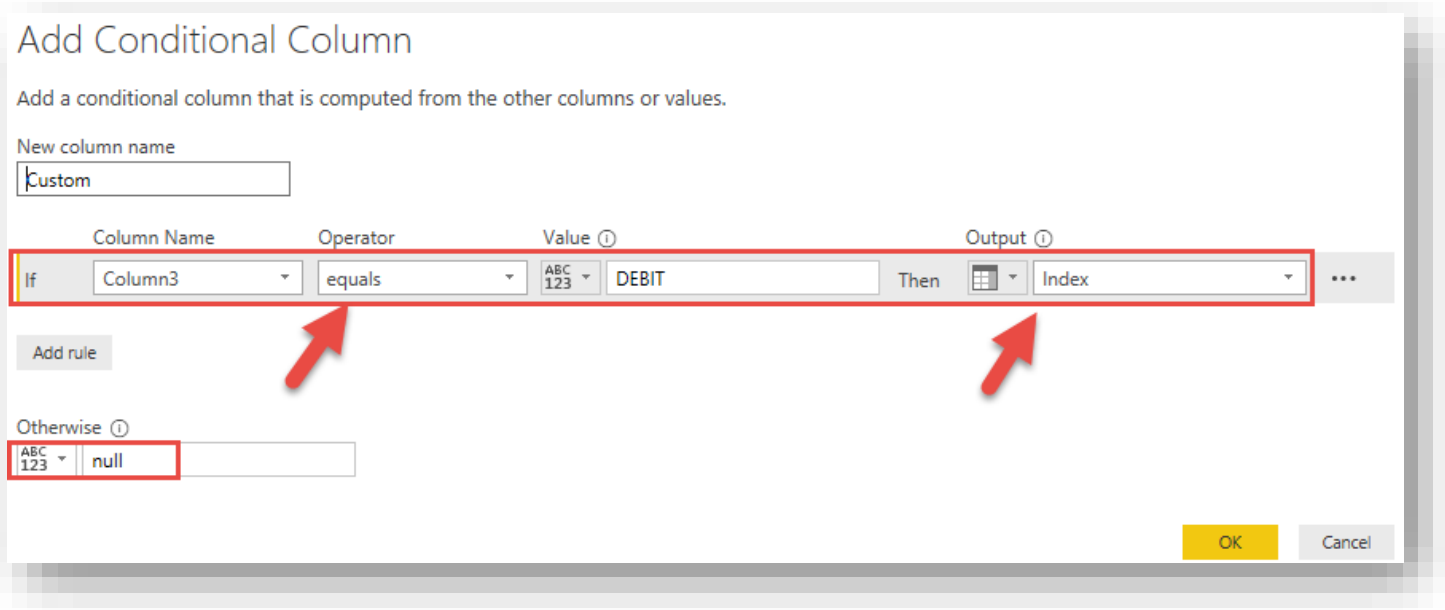
Exercise 4

- In order to prepare the query to create the **Entry Code** & **Amount**, there are several steps to perform
- First thing is to define the range of the table which contains the headers & the values
- Therefore, the next step is to **dynamically remove top rows**.
- In this case the headers of the actual table are in row **7** (**Credit** and **Debit**)
- To retrieve that number dynamically you need to create an **Index column**

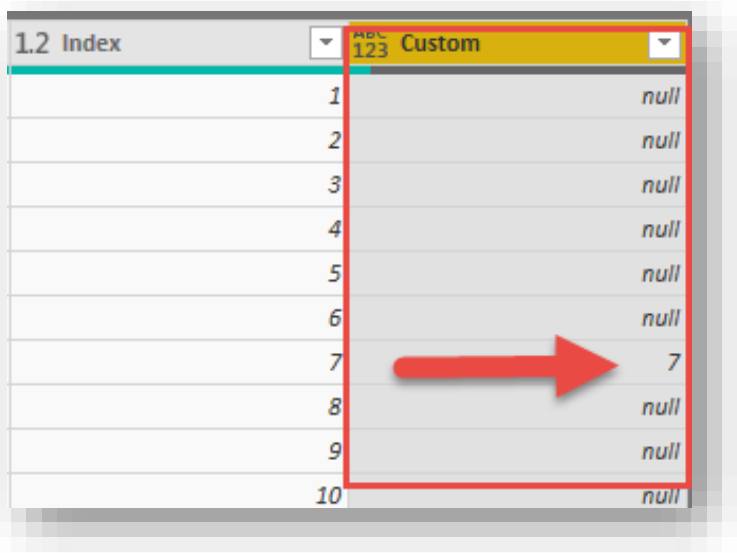




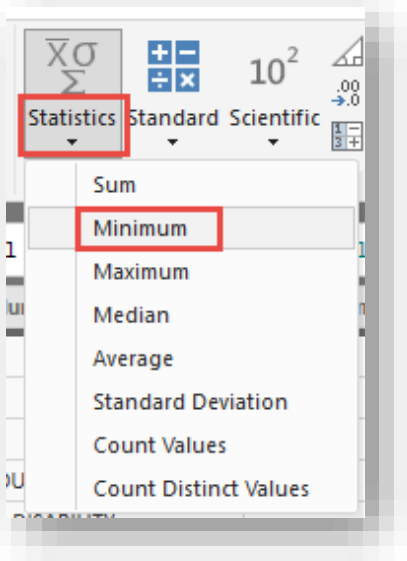
- With a conditional column you can then retrieve the number of the row, where the table starts



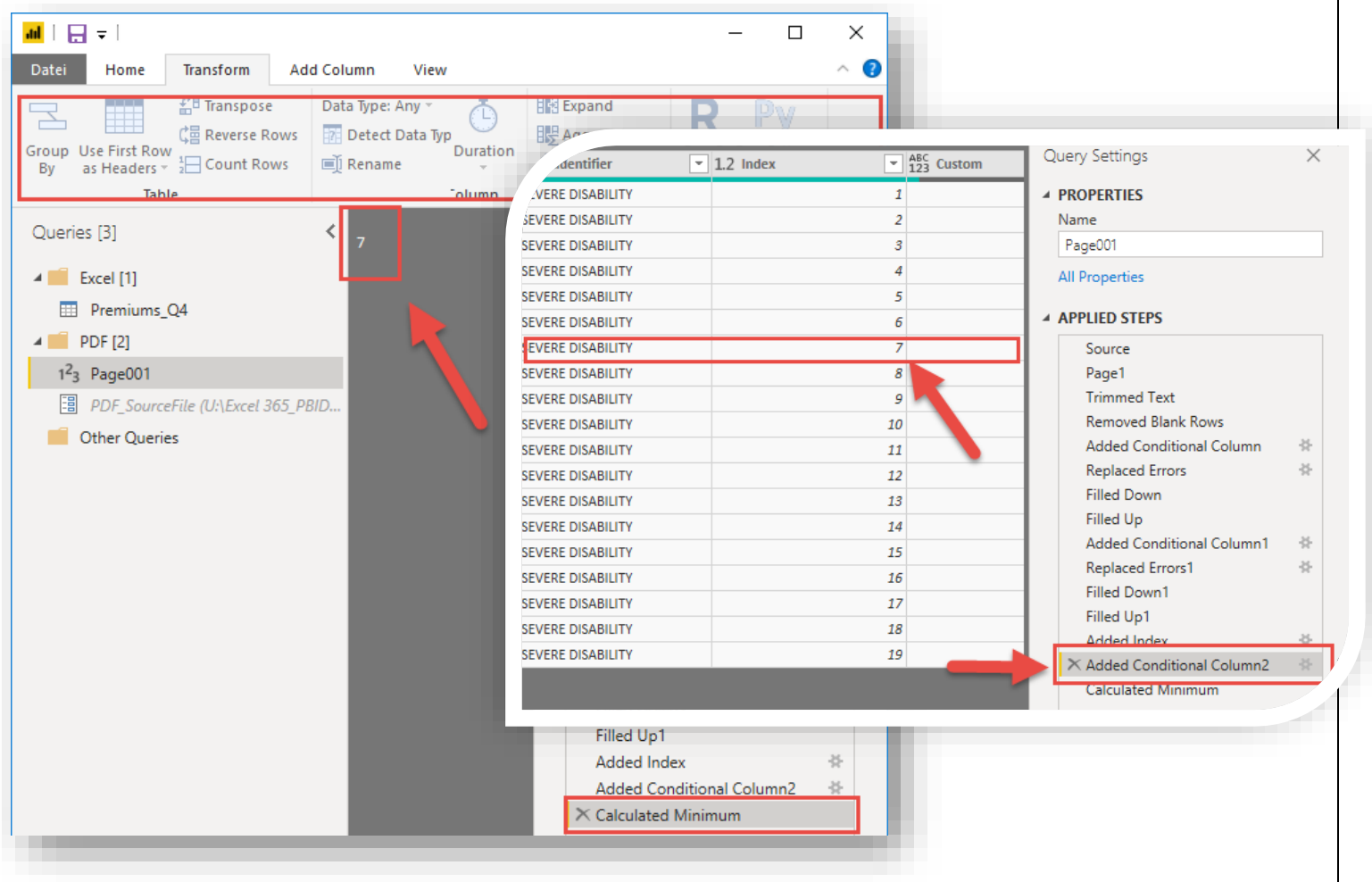
- In order to determine which is the top row (in case the header name exists more then once), a variable is needed to store the minimum value



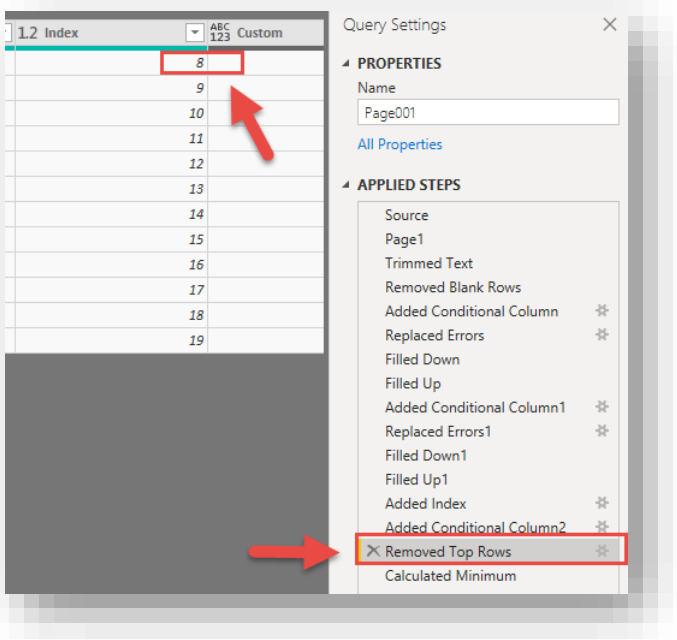
->

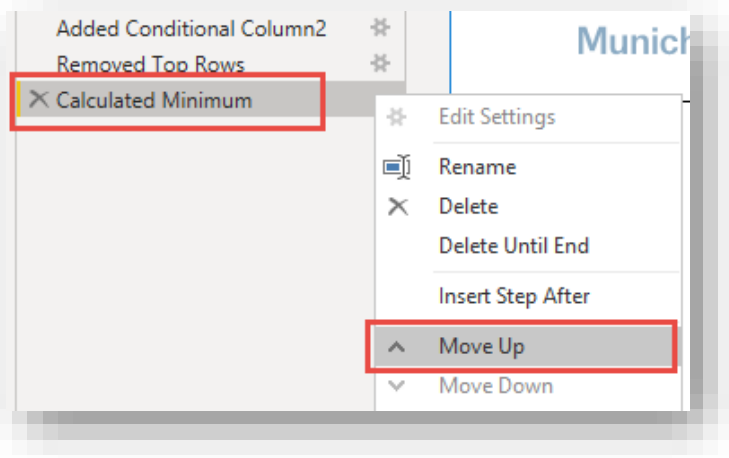


- As you cannot work with further commands here, you have to move up one step
- On the step above you add the step deleting the **first 7 rows** – as from here the table with the figures for Amount starts

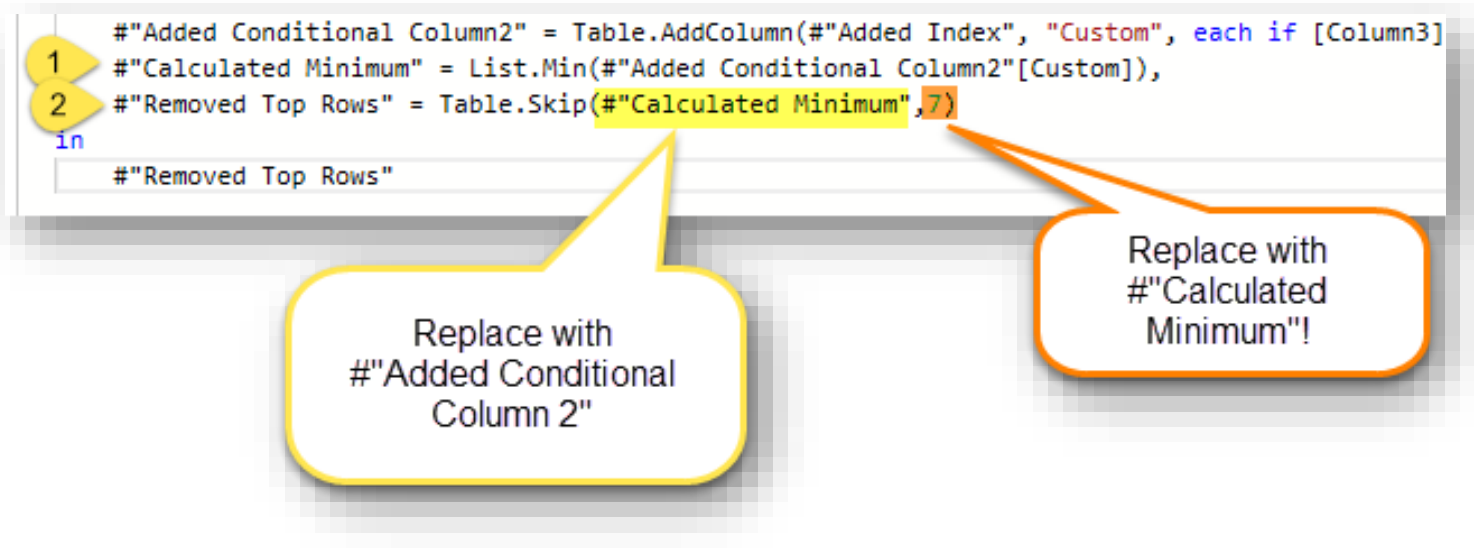


- You receive a table which starts with Index-number **8**
- Then: to keep the number of deleted rows dynamic, move up step “Calculated Minimum”





- Open the ***Advanced Editor***
- Replace the contents of the last line as shown in the screenshot

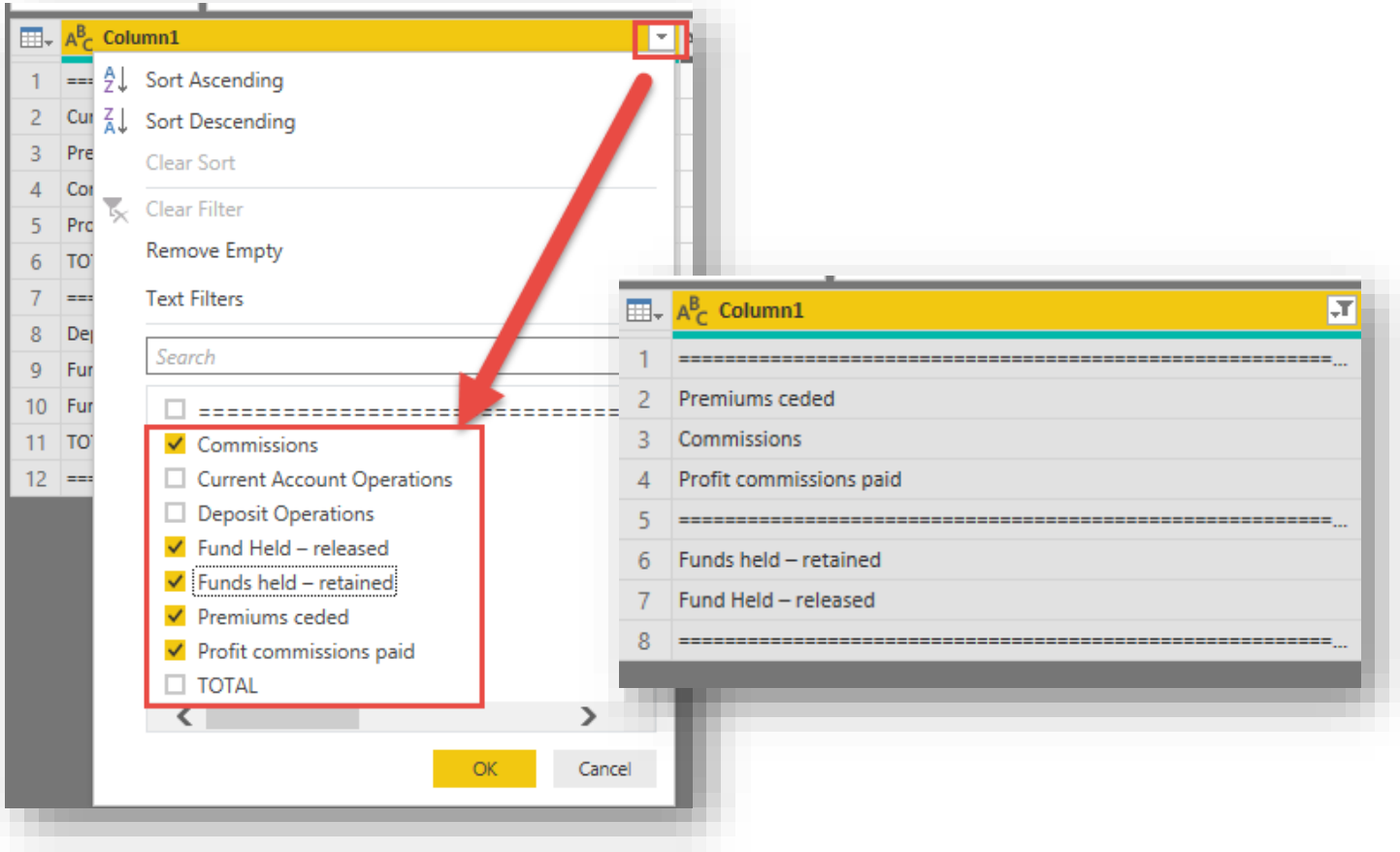


Result

- The query should work as before

```
    #"Added Index" = Table.AddIndexColumn(#"Filled Up1", "Index", 1, 1),
    #"Added Conditional Column2" = Table.AddColumn(#"Added Index", "Custom", each if [Column3] = "DEBIT" then [Index] else null),
    #"Calculated Minimum" = List.Min(#"Added Conditional Column2"[Custom]),
    #"Removed Top Rows" = Table.Skip( #"Added Conditional Column2", #"Calculated Minimum")
in
    #"Removed Top Rows"
```

- Filter the **Column1** to exclude nulls, totals and sub-headers



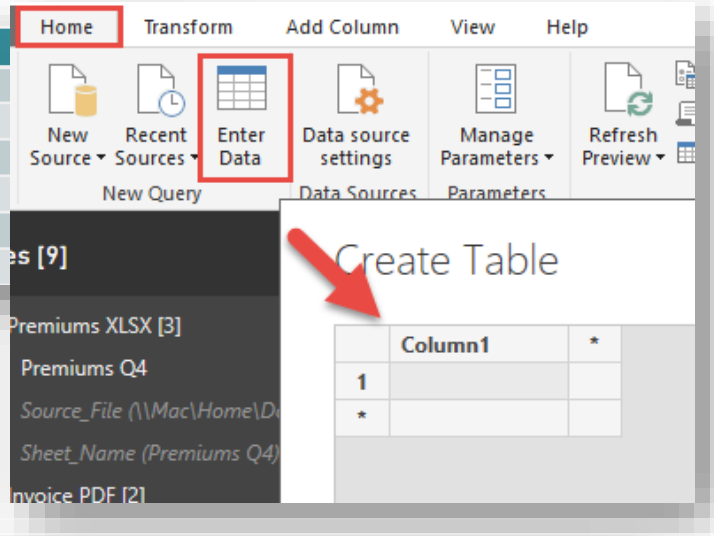
- Now you have to create the column Entry Code

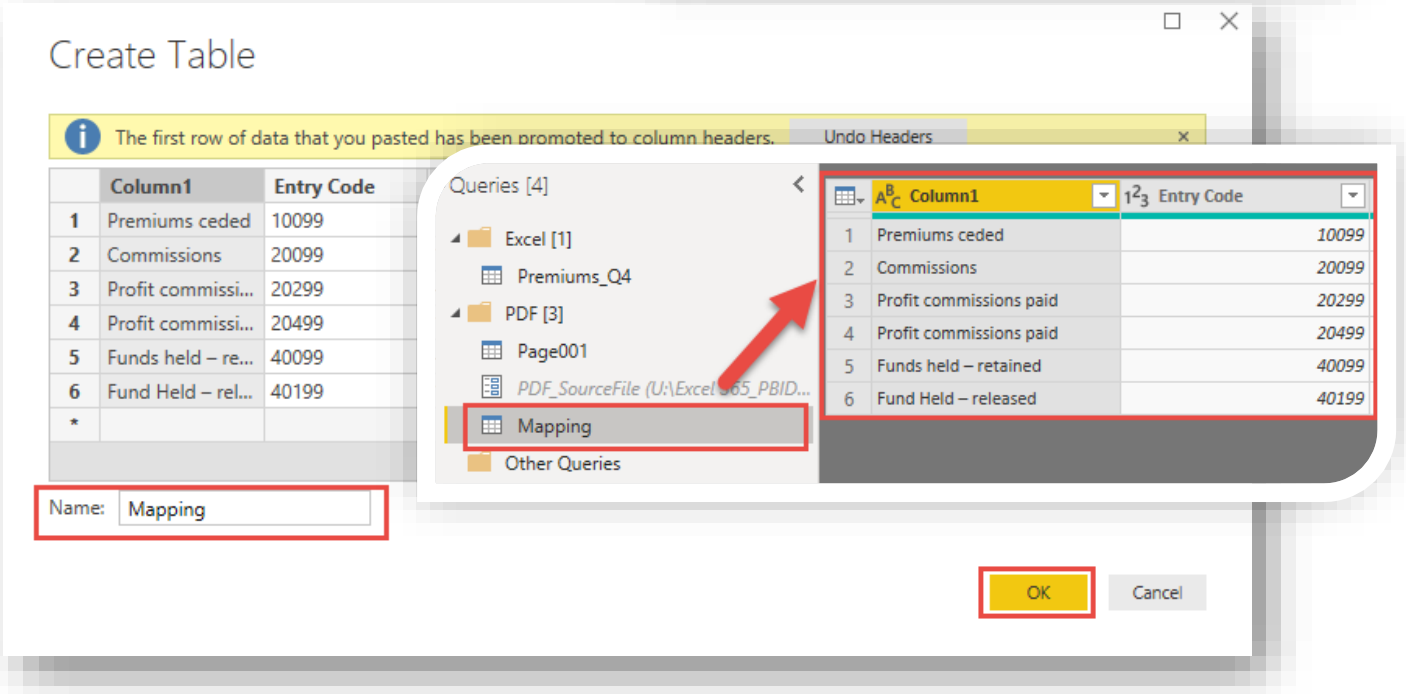
Entry Code
Calculated in model (FSRI Entry Code)
12345

Exercise 5

- Considering the following mapping of entry codes, create a mapping table
- You can copy that table from the Excel file Flexible Parameter_Code for PBI.xlsx / Sheet Mapping
- Name the table **Mapping**

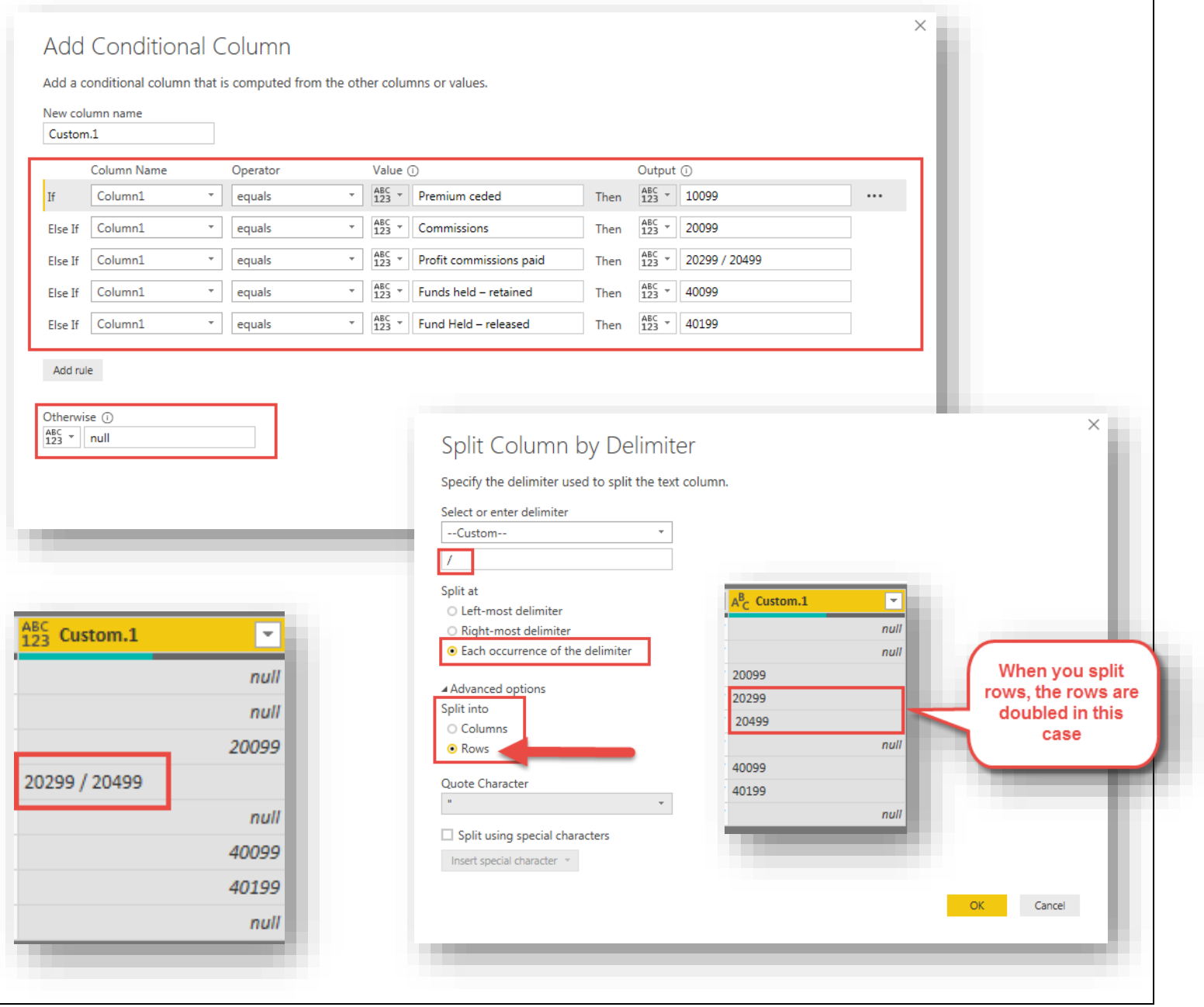
Column1	Entry Code
Premiums ceded	10099
Commissions	20099
Profit commissions paid	20299
Profit commissions paid	20499
Funds held – retained	40099
Fund Held – released	40199

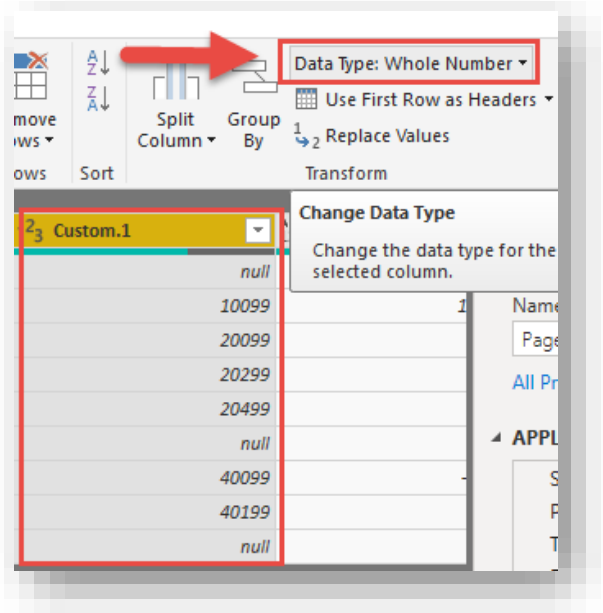
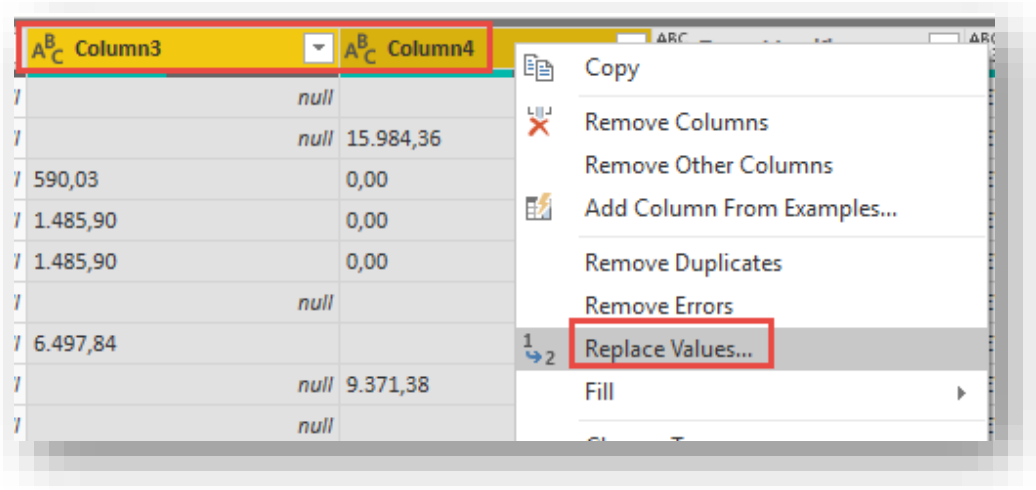




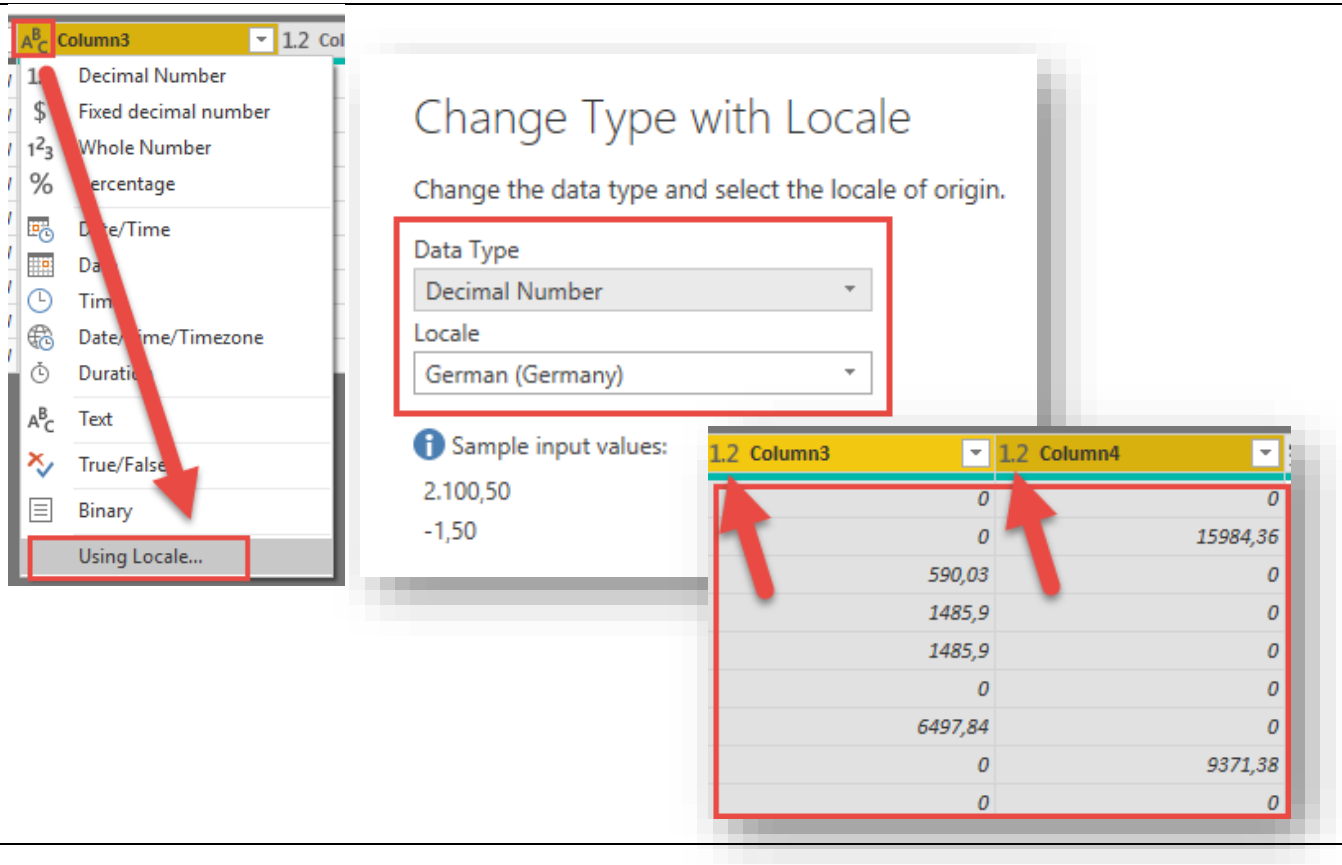
- Create a ***Conditional Column***: here you can define which **Entry code** belongs to which text

- Profit commissions paid has two different entry codes, thus needed to be duplicated

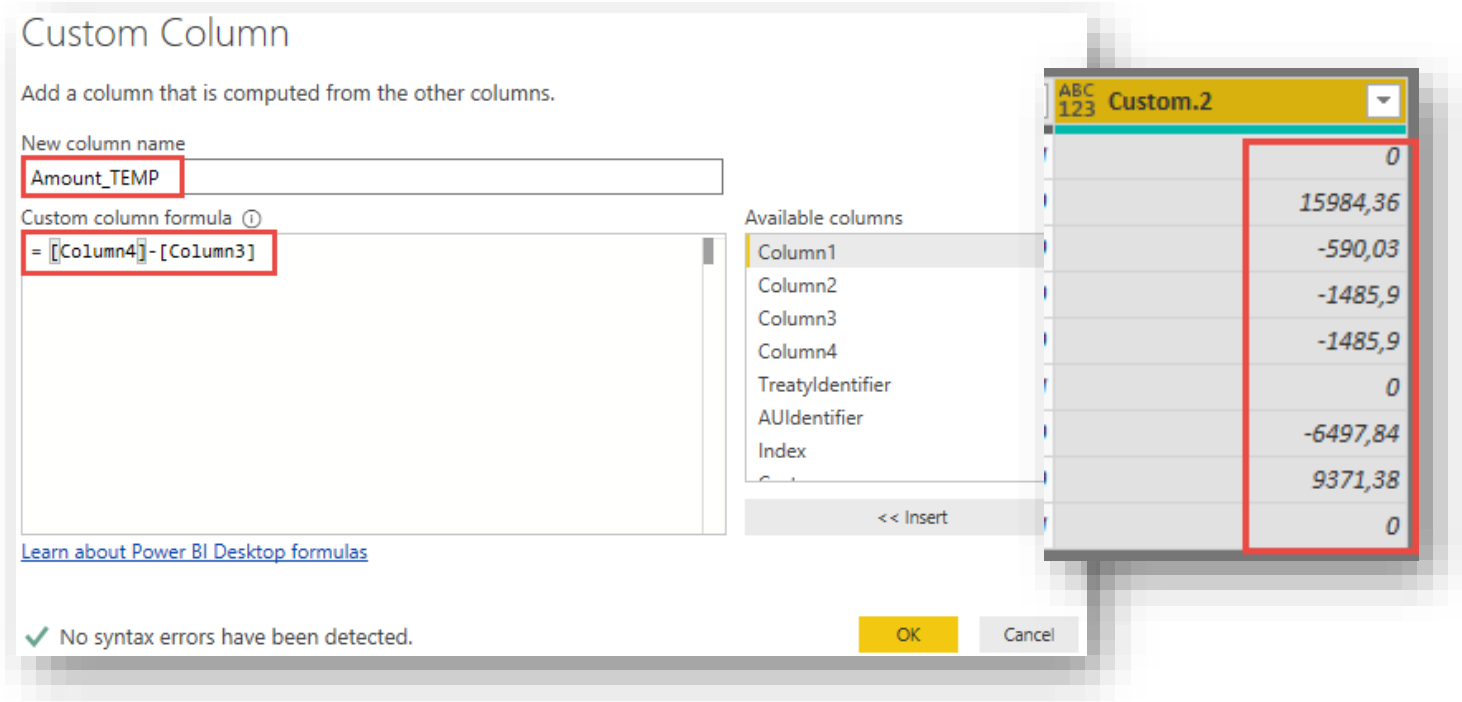


• As a last step switch the data type to ***Whole number***	
• In order to create the amount column, the debit should be deducted from credit • Nulls are not deductible, thus all **nulls** should be replaced with **0** • Highlight **column3** & **4**, then replace null to 0 • Make sure column type is **decimal**	

- If numbers are written with a comma as a decimal (8,6 and not 8.6), use convert with ***Using locale …*** (in A / APAC)



- After preparing the columns for amount, create a new ***Custom Column*** where you calculate a temporary amount



- Some amounts are wrong, because they should be multiplied by **-1**

- You can use the Mapping table to correct the calculations

There are 2 solutions:

Conditional column – in case there are not many values to check, it is possible to manually create a column with the formula:

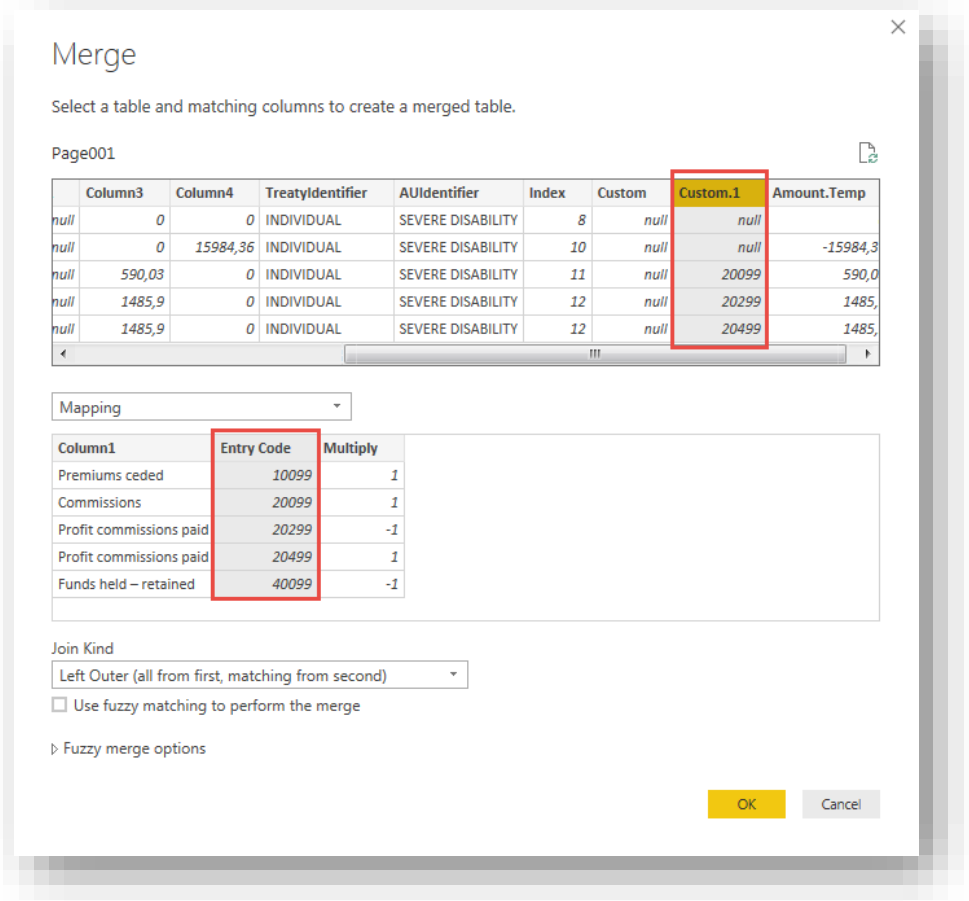
When [Entry Code] = 100 then [amount]*-1 else [amount]

Import Lookup – if there are many complicated conditions, a lookup table offers an easy solution which automatically updates the values

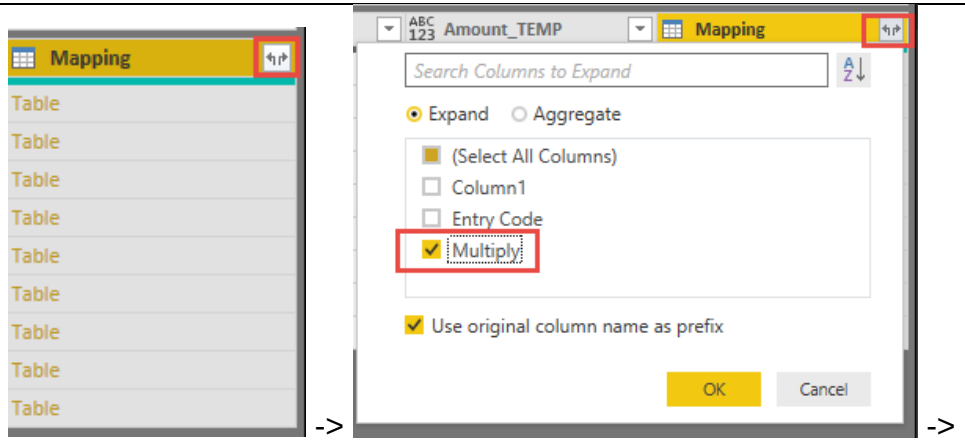
Column1	Entry Code	Multiply
Premiums ceded	10099	1
Commissions	20099	1
Profit commissions paid	20299	-1
Profit commissions paid	20499	1
Funds held – retained	40099	-1
Fund Held – released	40199	1

Merge tables

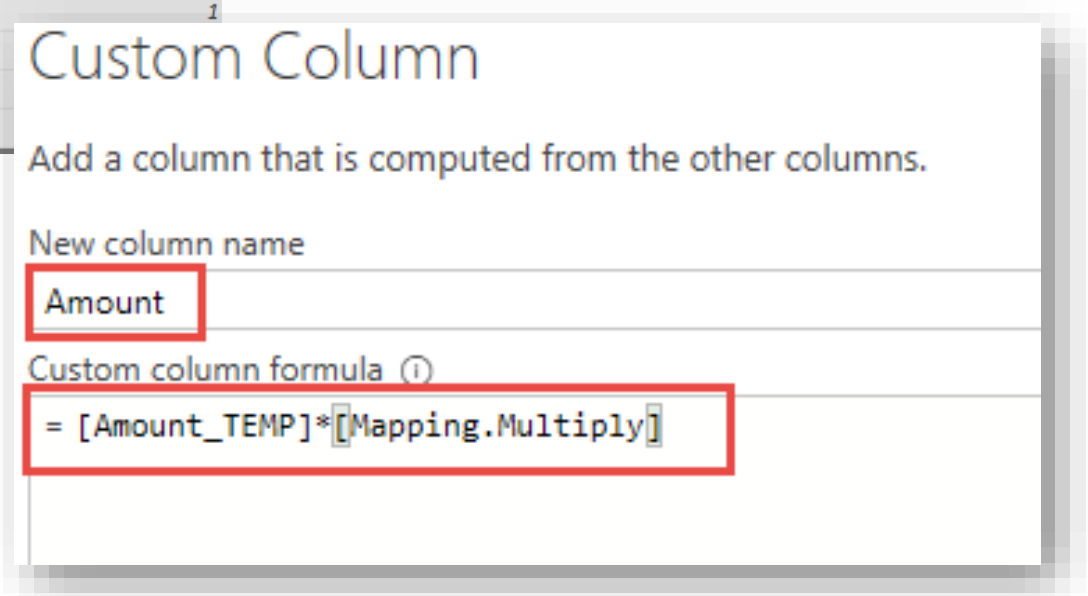
- In the original query, select the merge queries option - this function works like the VLOOKUP function in Excel
- Select the mapping table and chose in both
  tables which column should be merged on

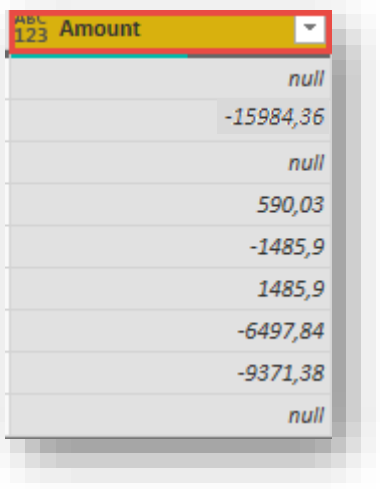


- A new column has been added and it contains only the value “Table”.
- In order to import the value, select the **expand icon**, and select which columns should be imported



- Create a ***Custom Column*** that multiplies both columns now

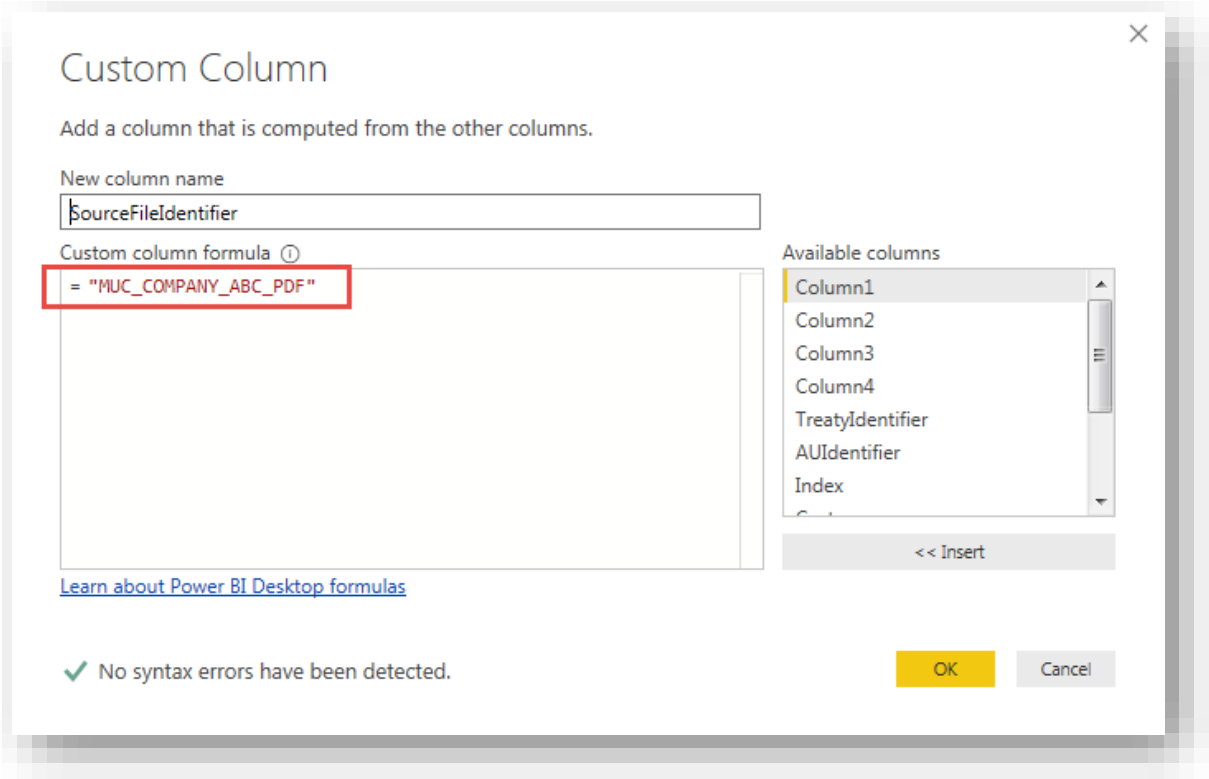


• Result	
After data is prepared, complete the model by adding necessary columns as in the Excel file	

2.2 Creating the necessary columns

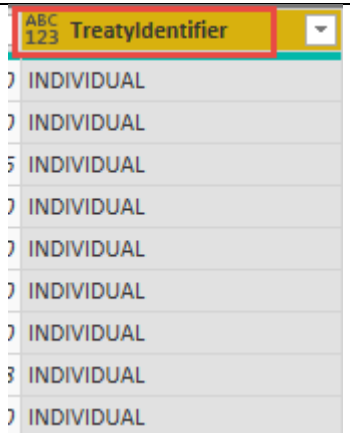
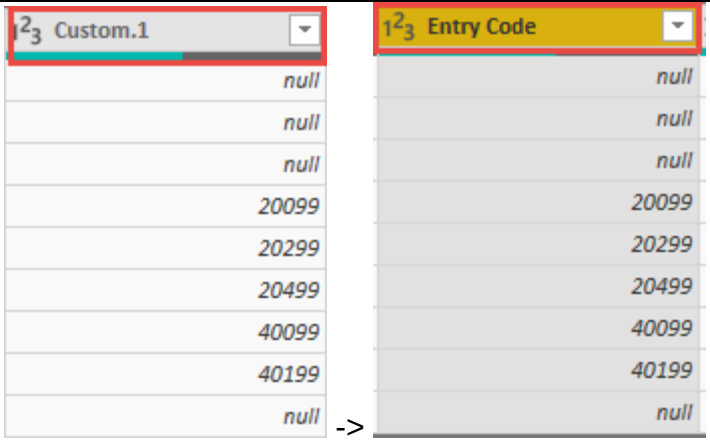
Exercise 6

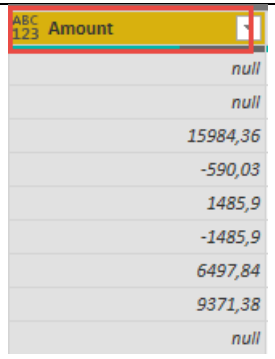
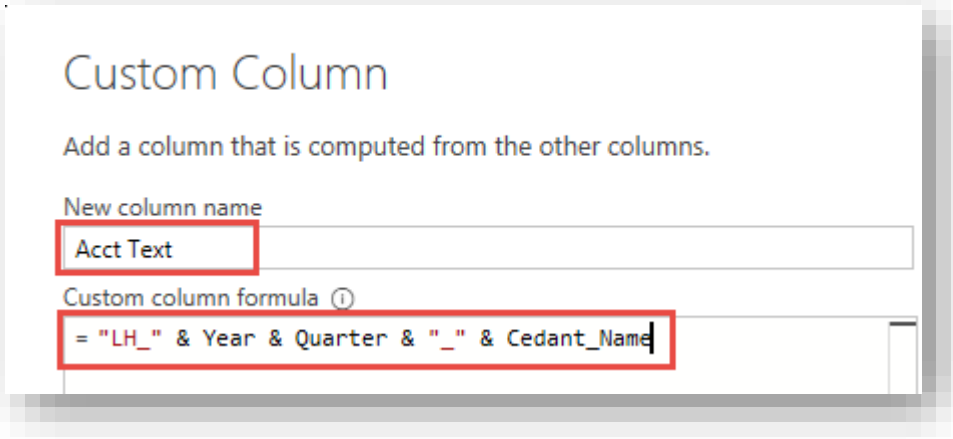
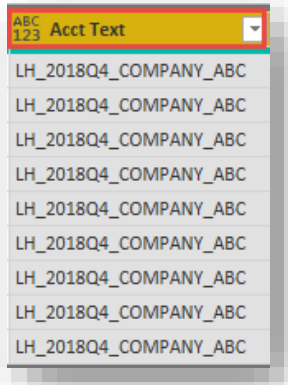
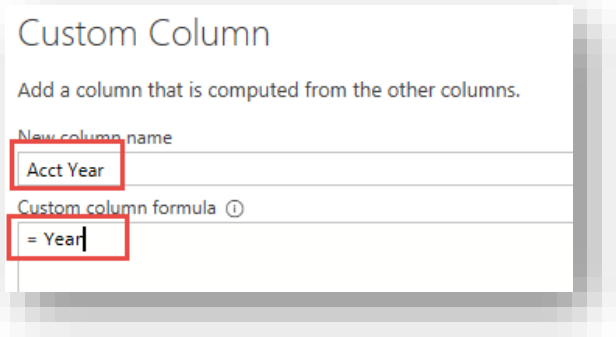
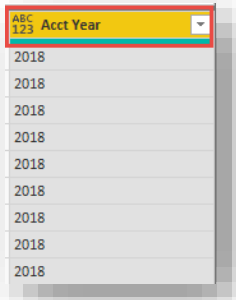
SourceFileIdentifier
[Country Code]_[Cedant Name]_[File Name]
ESP_MRE_PREMIUM



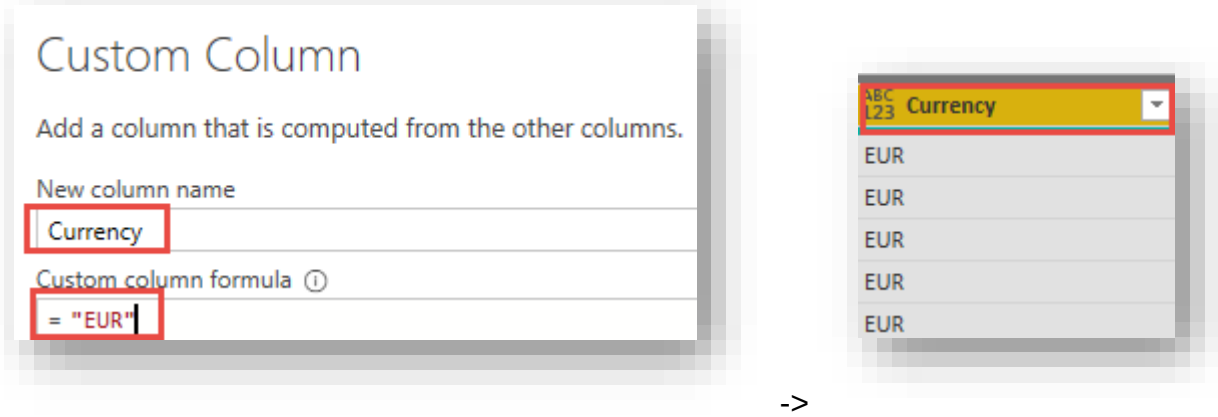
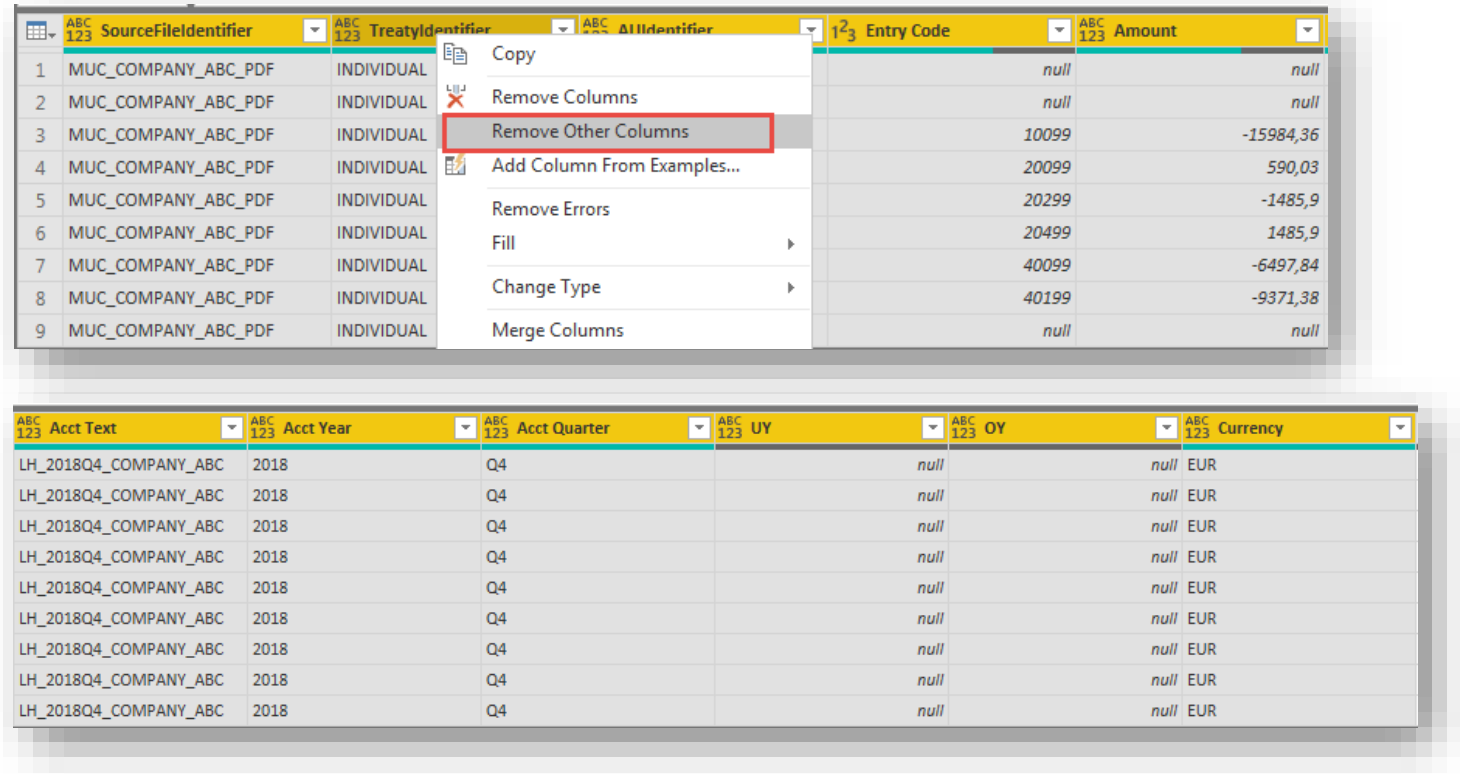
->

ABC 123 SourceFileIdentifier
MUC_COMPANY_ABC_PDF
MUC_COMPANY_ABC_PDF
MUC_COMPANY_ABC_PDF
MUC_COMPANY_ABC_PDF
MUC_COMPANY_ABC_PDF
MUC_COMPANY_ABC_PDF
MUC_COMPANY_ABC_PDF

TreatyIdentifier¤ Treaty·Identifier·as·originally·listed·in·invoice¤ Individual¤	 - already created
AUIdentifier¤ AU·Identifier·as·originally·listed·in·invoice¤ Individual¤	 - already created
Entry°Code¤ Calculated·in·model↵(FSRI·Entry·Code)¤ 12345¤	 ->

Amount Amount as originally listed in invoice or calculated percentage 99,9	 - already created
Acct Text LH_[Year&Quarter]_[Cendant Name] LH_2018Q4_MRE	 -> 
Acct Year Client Accounting Year 2018	 -> 

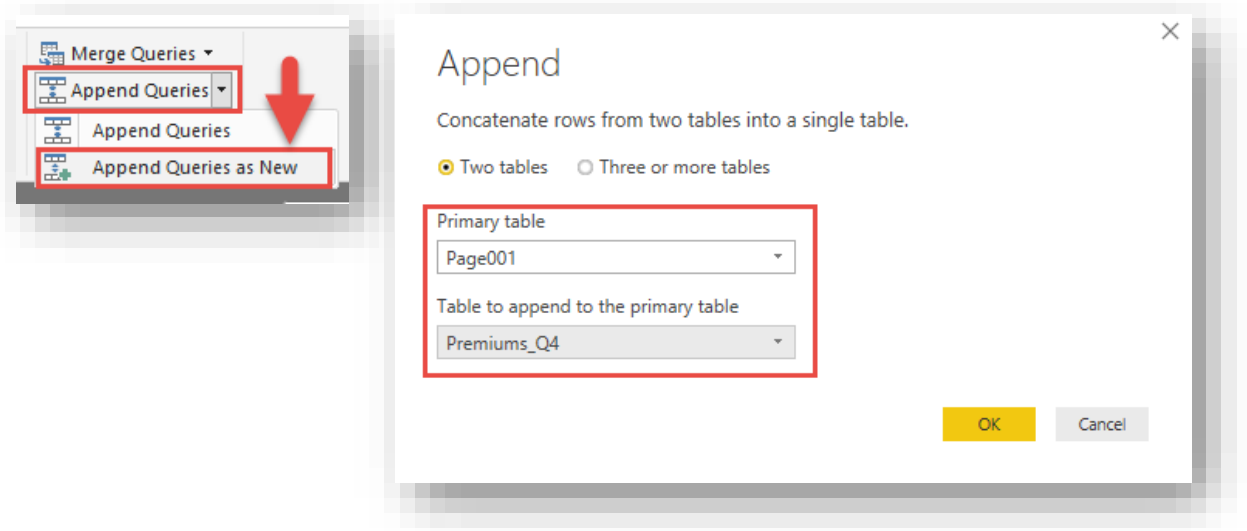
Acct Quarter Client Accounting Quarter Q4	Custom Column Add a column that is computed from the other columns. New column name: Acct Quarter Custom column formula: = Quarter -> Acct Quarter: Q4, Q4, Q4, Q4, Q4, Q4
UY UY as in original file	Custom Column Add a column that is computed from the other columns. New column name: UY Custom column formula: = null -> UY: null, null, null, null, null
OY OY as in original file 2017	Custom Column Add a column that is computed from the other columns. New column name: OY Custom column formula: = Null -> OY: null, null, null, null, null, null

	
• Reorder Columns • Highlight all columns you want to keep – then rightclick & choose ***Remove other columns*** • Check & correct the **data type** of each column – avoid the data type ***Any***!	

2.3 Appending Queries & Testing Template

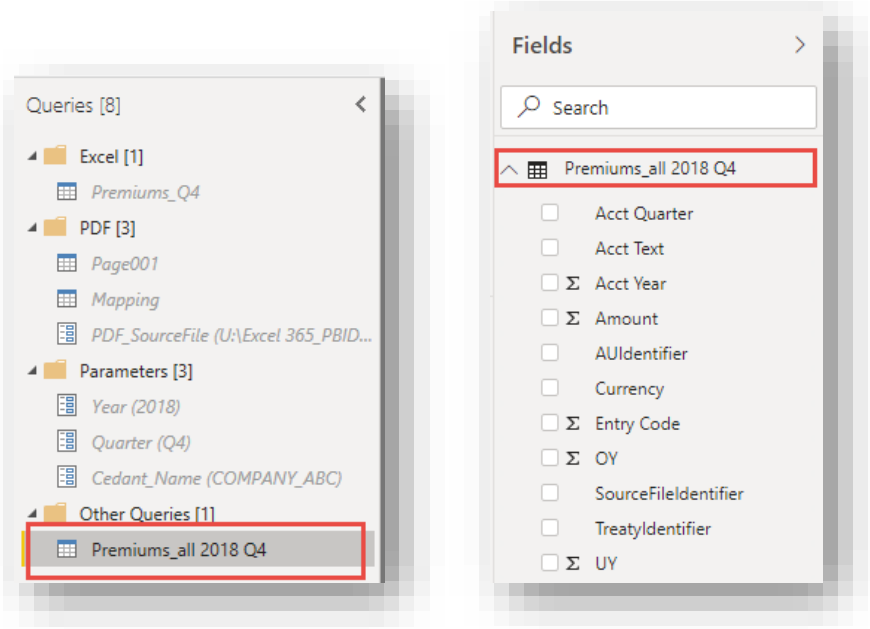
Exercise 7

- **Append** the 2 queries now
- Rename the new query: **Premiums_all 2018 Q4**



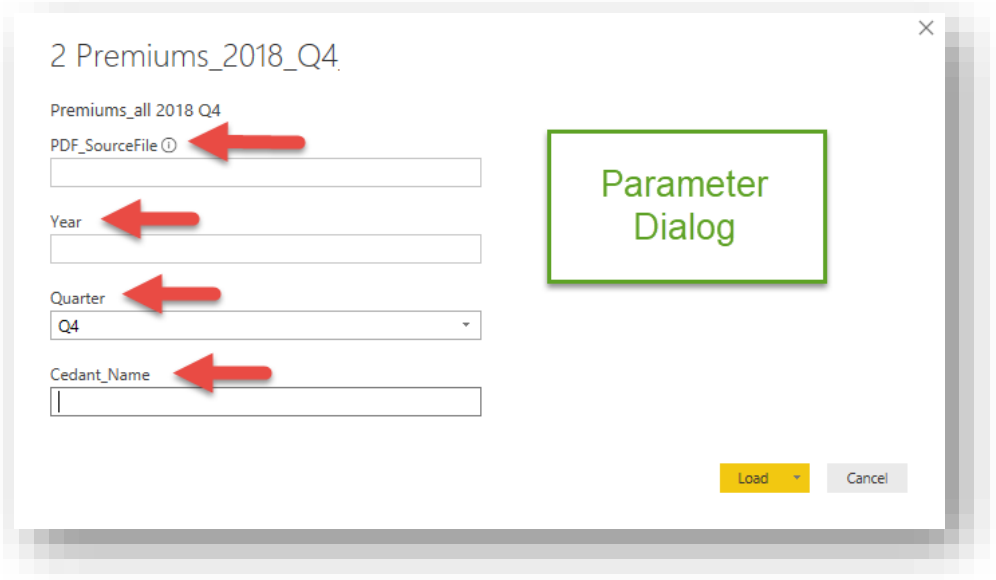
Change the setting *Enable Load*

- Only the new query should be loaded to the data model
- Save with ***Close & Apply***
- Save the **PBIX** file



Test the Template

- Save the PBIX-file as a ***template***
- Copy the PDF file to the desktop
- Open the template & test the parameter dialog with the new path for the PDF file



Stay tuned ... 

Munich RE