



Power BI Desktop - *Deep dive Query Editor*

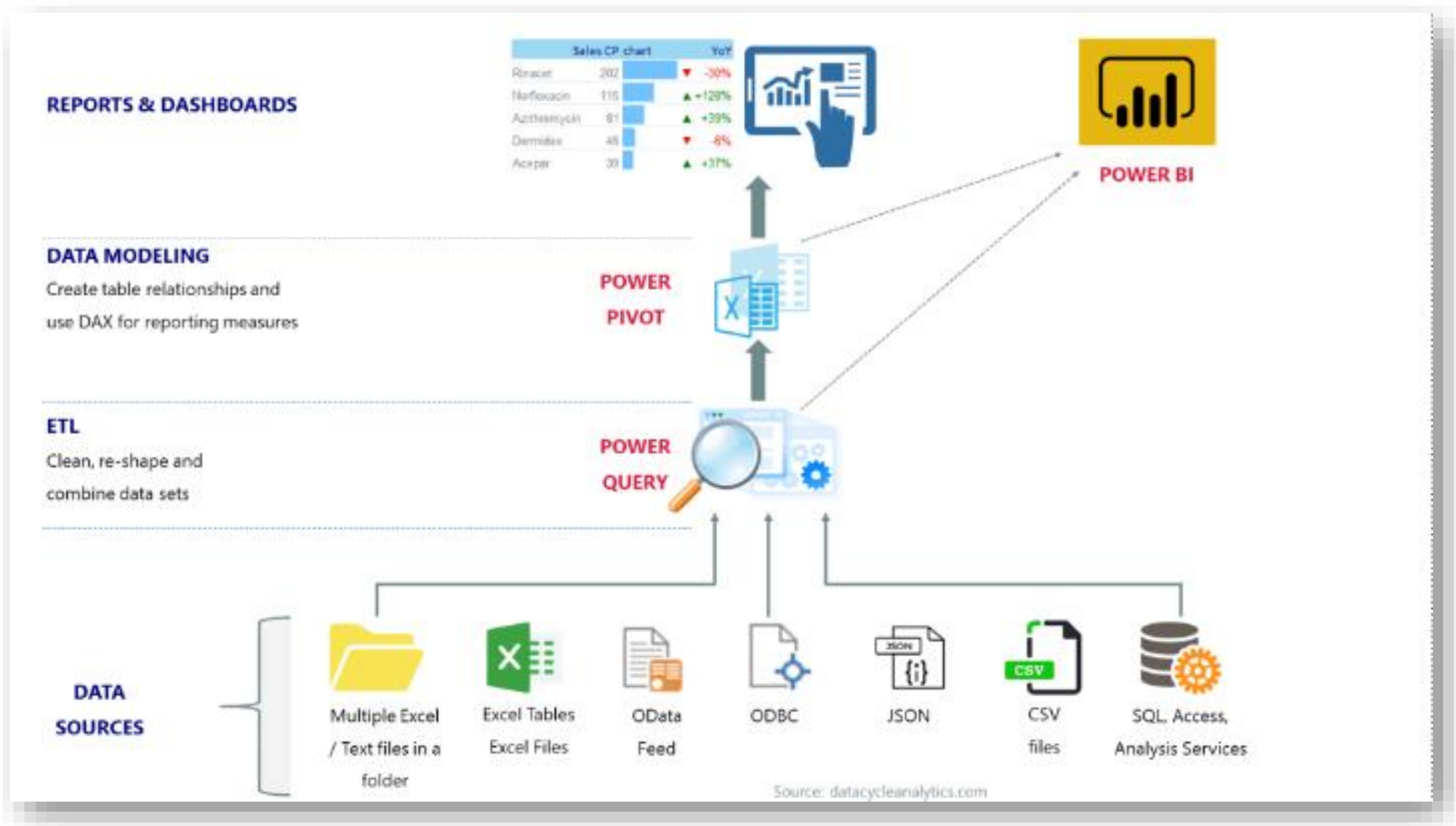
Barbara Berghaus IT1.4.2.1

Last updated: 10.09.2019

Agenda

1	Working with the Query Editor – Technical Deep dive	4
1.1	Basic Principles	6
2	Example – Data Preparation from PDF	20
2.1	Creating the table structure.....	26
2.2	Creating the necessary columns.....	42
2.3	Appending Queries & Testing Template	47

Overview



1 Working with the Query Editor – Technical Deep dive



Why should you deal with the Query Editor?



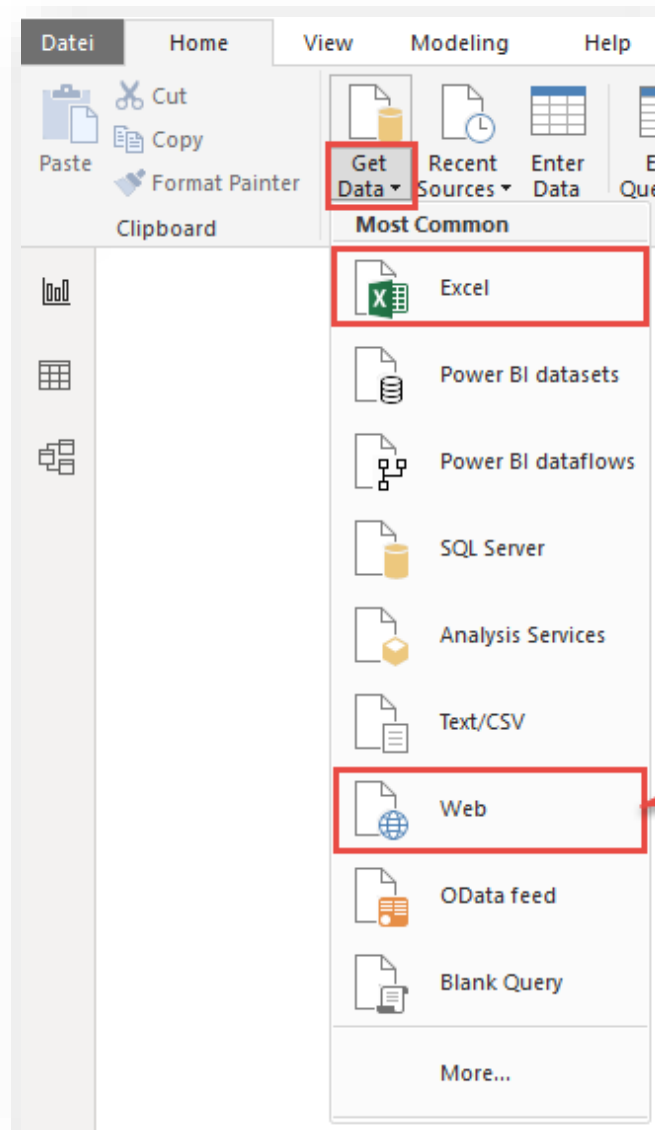
1.1 Basic Principles



Exercise 1

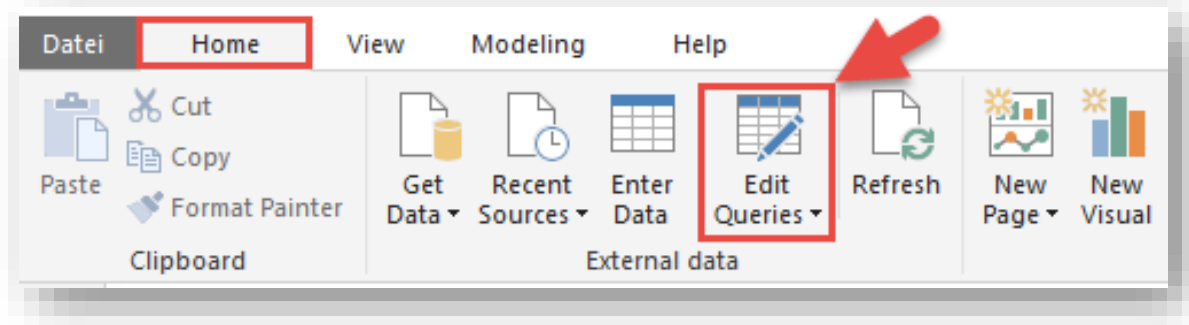
Get Data

- 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**




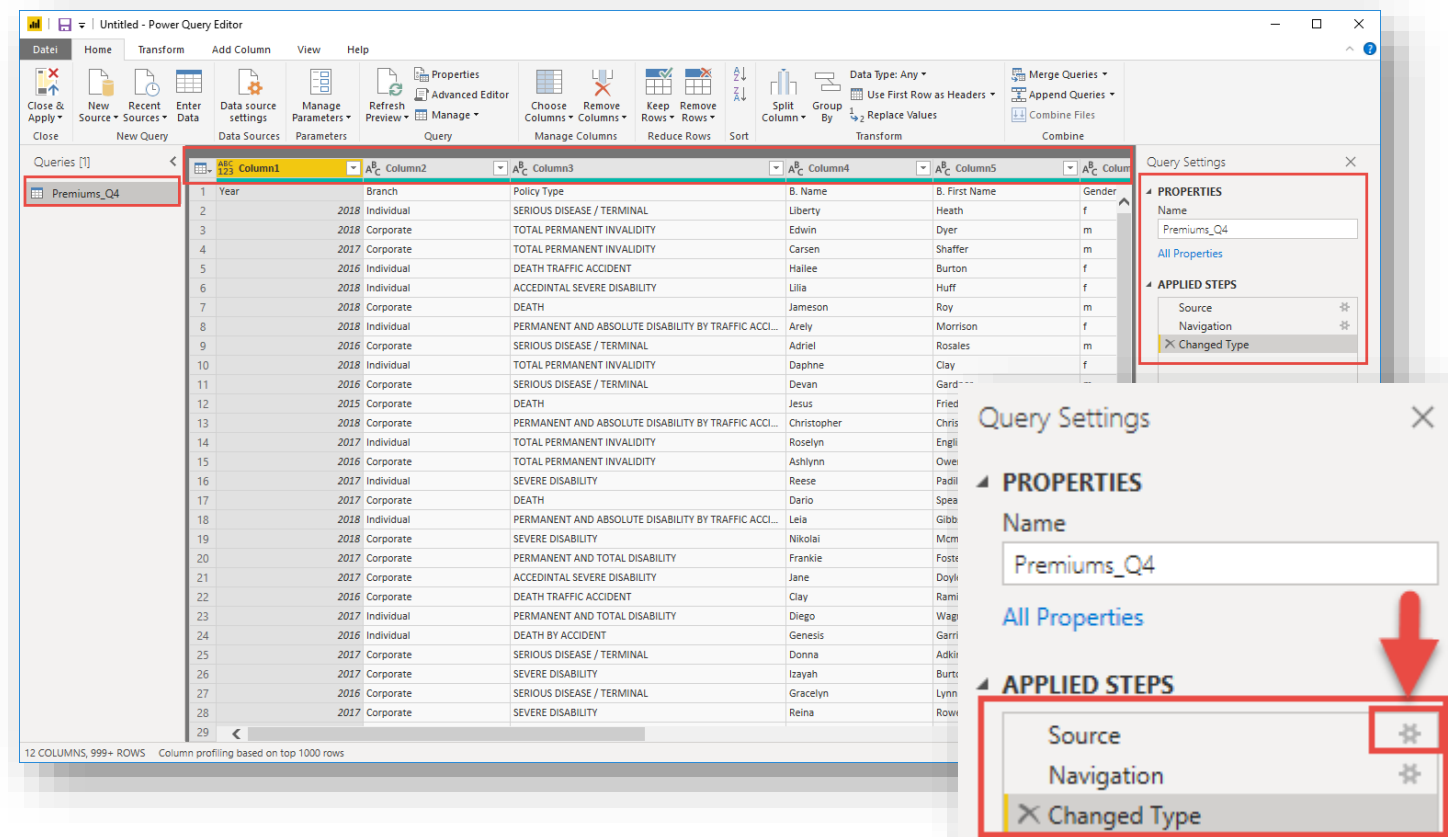
When the data is stored in Sharepoint select **Web** & paste the URL

- 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

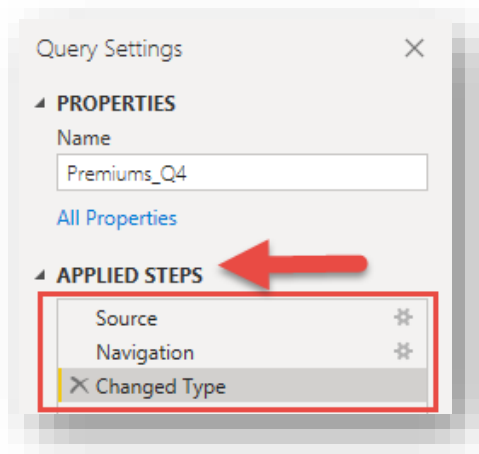
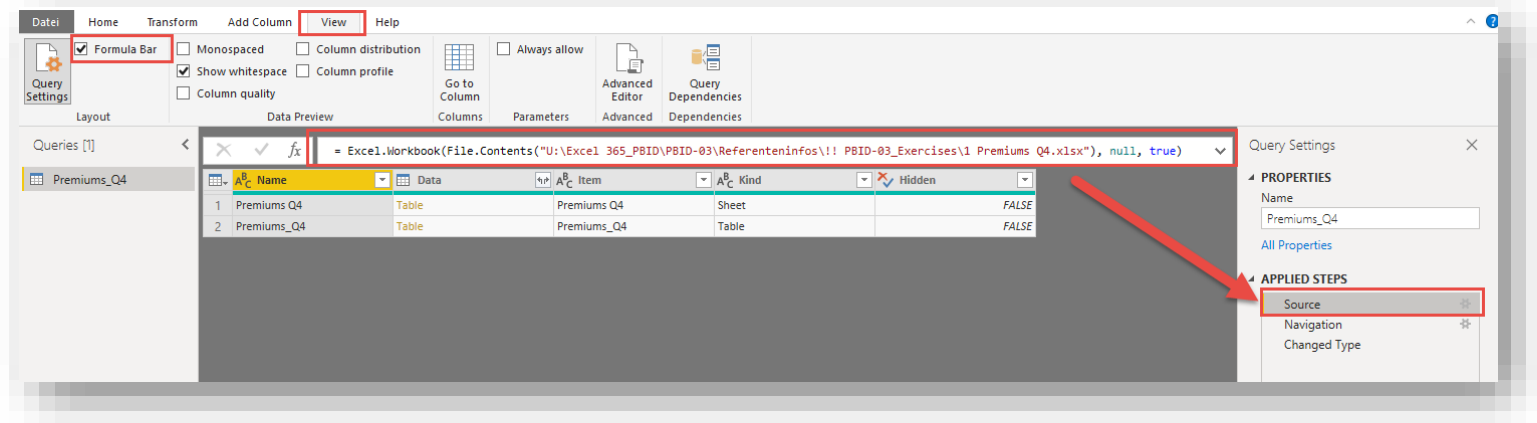
- Each modification to the file is recorded under the **Applied Steps** pane on the right
- It is possible (but not always recommended) to reorder steps by dragging them
- It is possible to edit each step, by clicking the 
- It is possible to see how the data looks in each step just by clicking it



- 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:

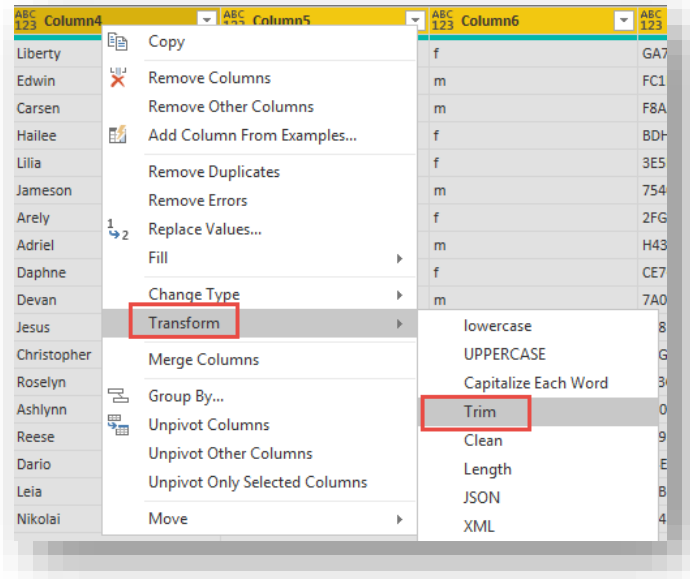
- **Source** – file path / URL to source, type of file (Excel, PDF, CSV etc.)
- **Navigation** – which page or excel sheet?
- **Changed Type** – automatically detects the column data type (date, decimal number, text etc.)





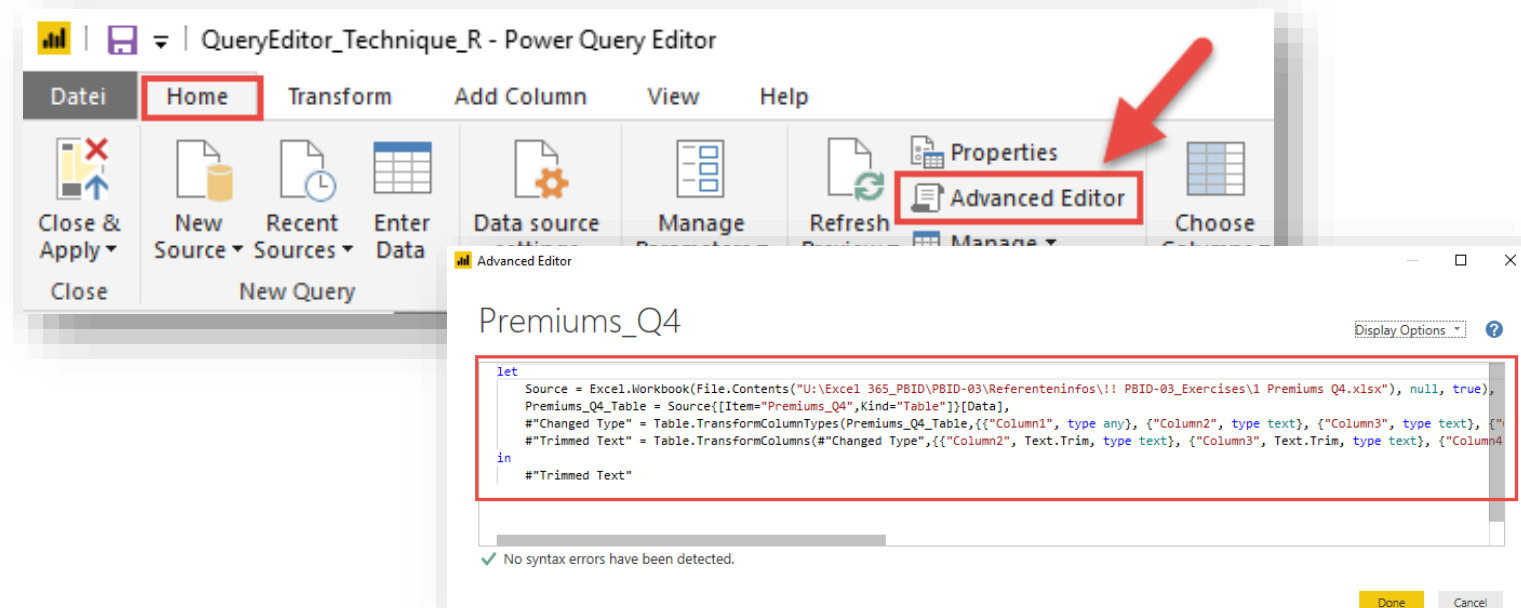
Exercise 2

- 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

Premiums Q4

```
let
    Source = Excel.Workbook(File.Contents("\\MUCE019334\MUC-Home10$\ny21942\desktop\TrainingExample\Premiums Q4.xlsx"),
        #"Premiums Q4_Sheet" = Source{[Item="Premiums Q4",Kind="Sheet"]}[Data],
        #"Promoted Headers" = Table.PromoteHeaders(#"Premiums Q4_Sheet", [PromoteAllScalars=true]),
        #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"Year", Int64.Type}, {"Branch", type text}, {"Pol
in
    #"Changed Type"
```

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

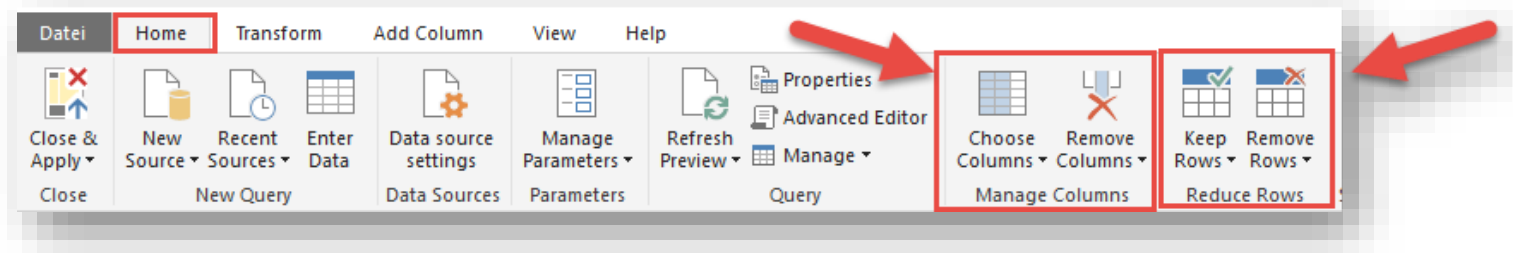
- The name of the variable of the predecessor line has to be repeated in the following line
- Each line is closed with a comma, only the last line in the **"let"**-block has no comma
- The name of the last variable in the **"let"**-block is the name of the step & thus has to be repeated in the **"in"**-block

Premiums Q4

```
let
    Source = Excel.Workbook(File.Contents("\\MUCE019334\MUC-Home10$\ny21942\desktop\TrainingExample\Premiums Q4.xlsx"),
        #"Premiums Q4_Sheet" = Source{[Item="Premiums Q4",Kind="Sheet"]}[Data],
        #"Removed Blank Rows" = Table.SelectRows(#"Premiums Q4_Sheet", each not List.IsEmpty(List.RemoveMatchingItems(Record
        #"Removed Top Rows" = Table.Skip(#"Removed Blank Rows",1),
        #"Added Custom" = Table.AddColumn(#"Removed Top Rows", "SourceFileIdentifier", each "MUC_" & Cedant & File_Name1)
in
    #"Added Custom"
```

Common functions

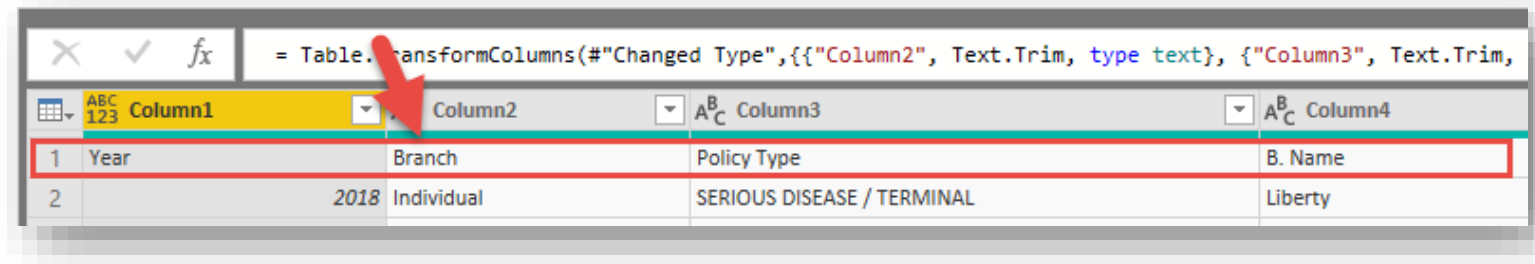
- **Remove / Choose Columns** – select which columns to include or exclude from the table
- **Remove Rows** – remove x top or bottom rows, remove empty rows



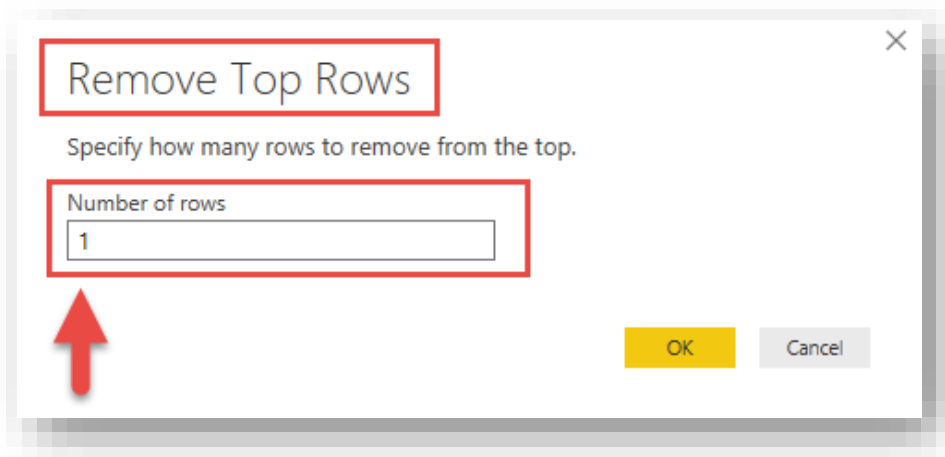
Exercise 4

Delete the first row

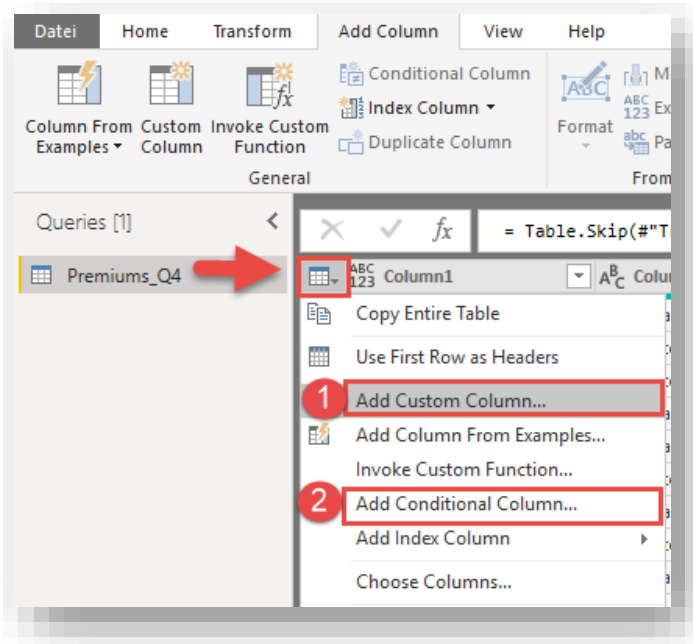
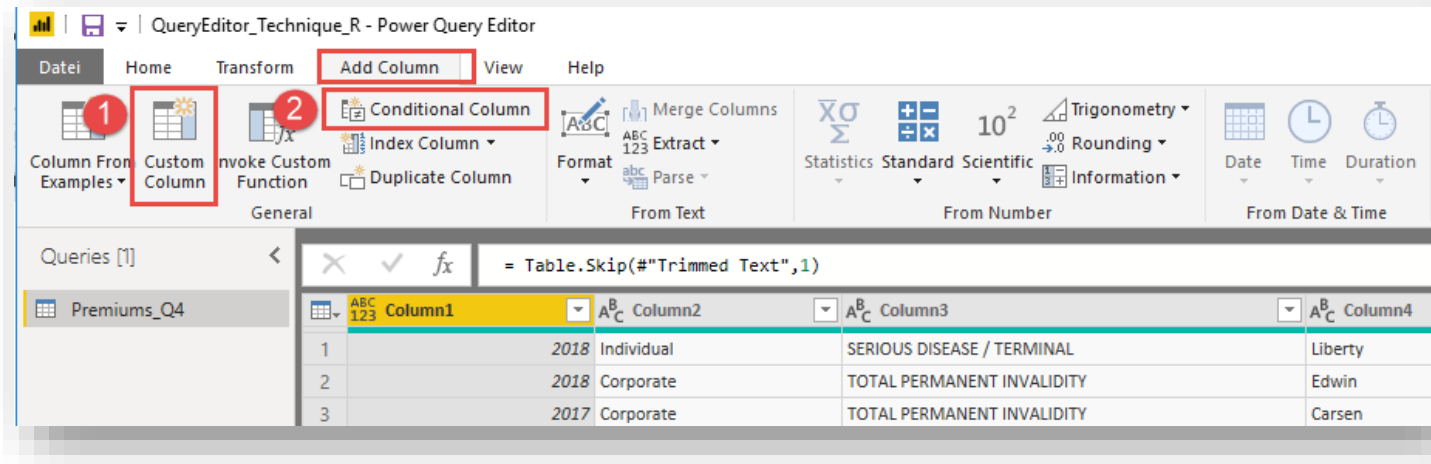
- Titles of the imported files might change
- Therefore, you can avoid problems with column titles using the standard titles like *Column 1*, *Column 2*, ...
- In our example please delete the first row with the original column titles



	Column1	Column2	Column3	Column4
1	Year	Branch	Policy Type	B. Name
2	2018	Individual	SERIOUS DISEASE / TERMINAL	Liberty



- **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

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

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

Add Conditional Column

Add a conditional column that is computed from the other columns or values.

New column name
Treaty Identifier

	Column Name	Operator	Value ①	Output ②
If	Column2	equals	ABC 123 Individual	Treaty 1
Else If	Column2	equals	ABC 123 Corporate	Treaty 2
Else ①	ABC 123 null			

Buttons: Add rule, OK, Cancel

APPLIED STEPS

- Source
- Navigation
- Changed Type
- Trimmed Text
- Removed Top Rows
- Added Conditional Column**

Premiums_Q4

```

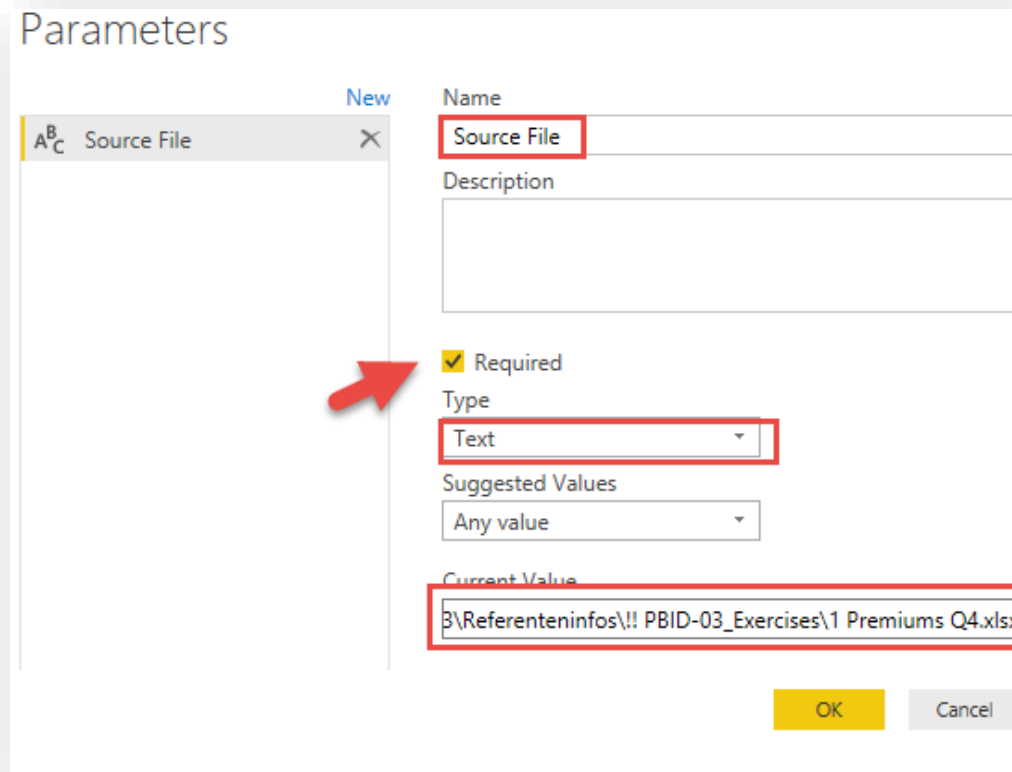
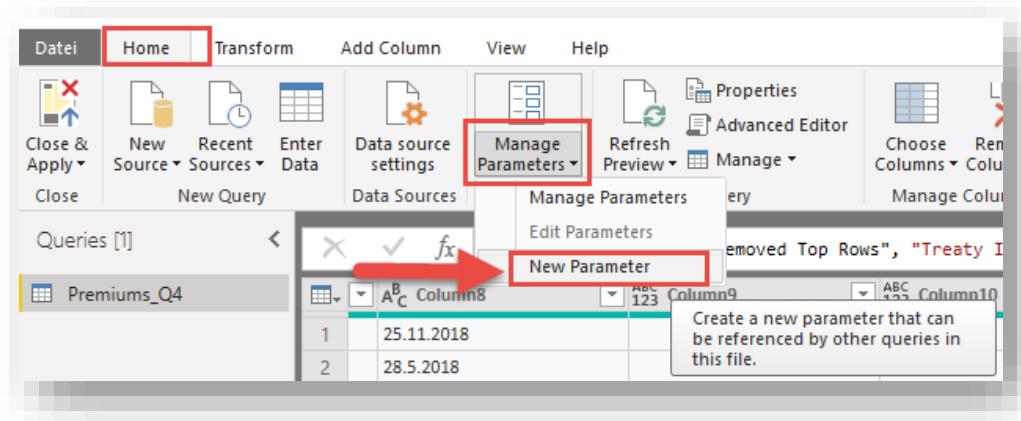
let
    Source = Excel.Workbook(File.Contents("U:\Excel 365_PBID\PBID-03\Referenteninfos\!! PBID-03_Exercises\1 Premiums Q4.xlsx"), null, true),
    Premiums_Q4_Table = Source[[Item="Premiums_Q4",Kind="Table"]][Data],
    #"Changed Type" = Table.TransformColumnTypes(Premiums_Q4_Table,{{"Column1", type any}, {"Column2", type text}, {"Column3", type text}, {"Column4", type text}}, {"Column1", type any}, {"Column2", type text}, {"Column3", type text}, {"Column4", type text}),
    #"Trimmed Text" = Table.TransformColumns(#"Changed Type",{{"Column2", Text.Trim, type text}, {"Column3", Text.Trim, type text}, {"Column4", Text.Trim, type text}}, {"Column2", Text.Trim, type text}, {"Column3", Text.Trim, type text}, {"Column4", Text.Trim, type text}),
    #"Removed Top Rows" = Table.Skip(#"Trimmed Text",1),
    #"Added Conditional Column" = Table.AddColumn(#"Removed Top Rows", "Treaty Identifier", each if [Column2] = "Individual" then "Treaty 1" else if [Column2] = "Corporate" then "Treaty 2" else null, type text),
in
    #"Added Conditional Column"
  
```



Exercise 6

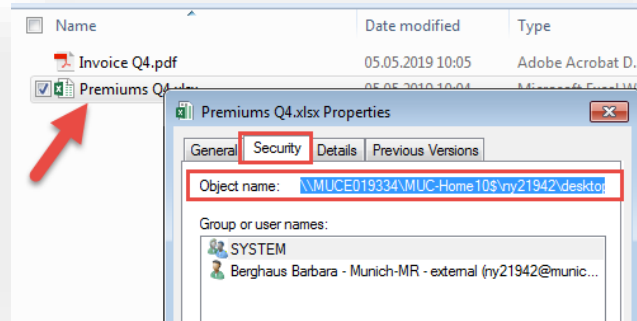
Using Parameters

- 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

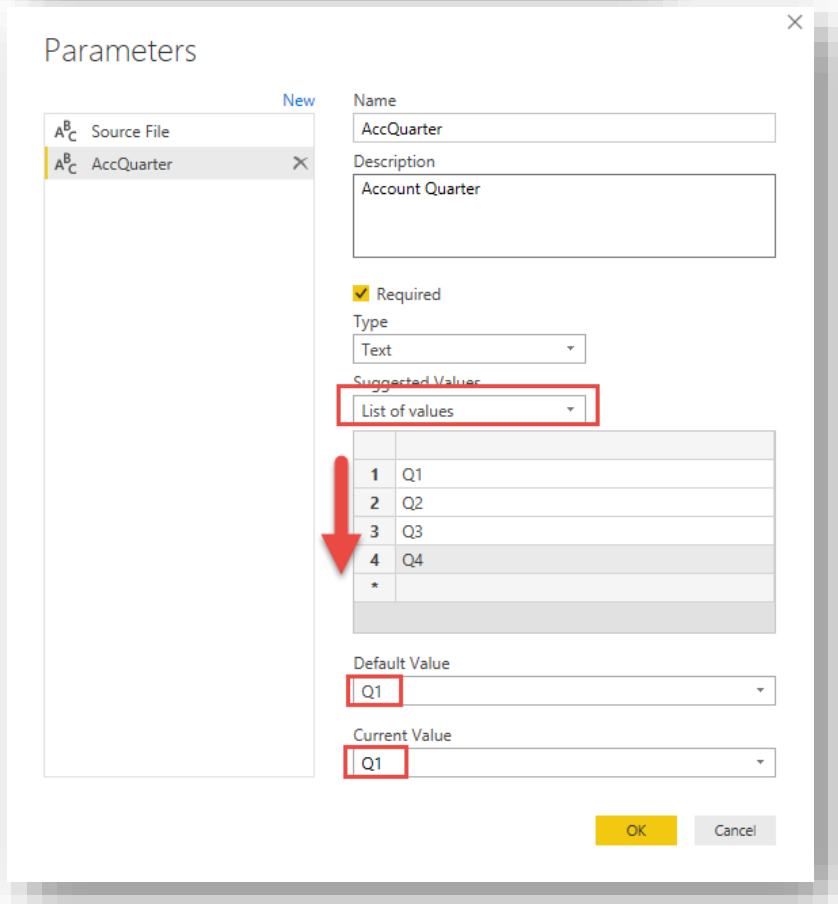


The file path can be copied:

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

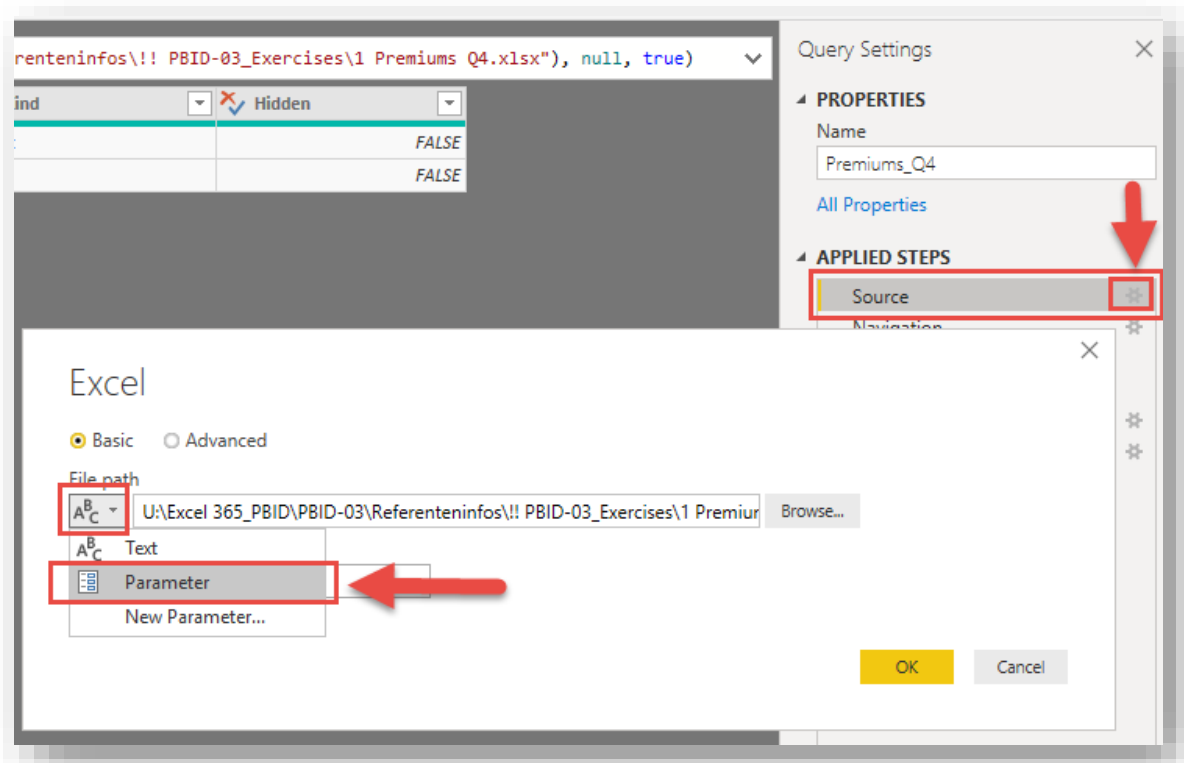
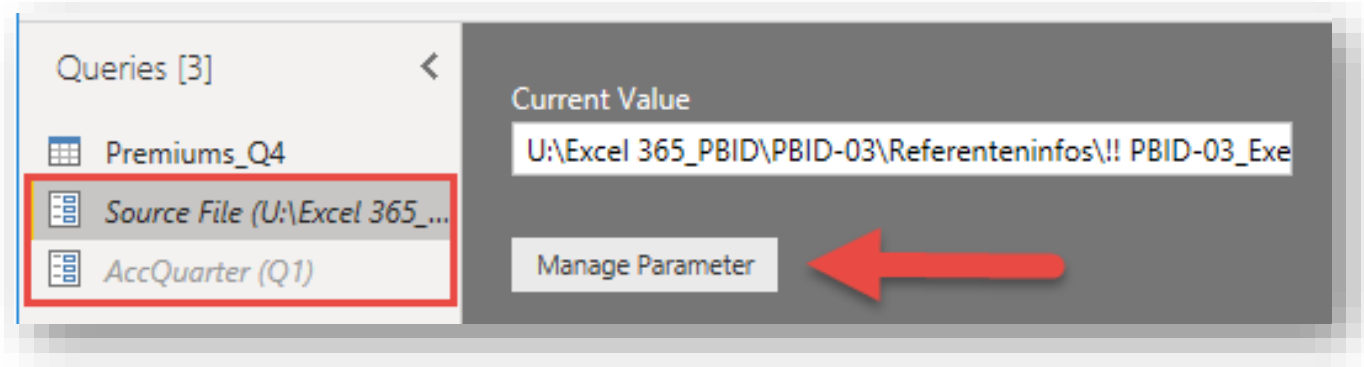


- It is also possible to provide a list of values, when you want to make sure, that the entries from the user are always correct
- For example the entry for the column **Acct Quarter** should always be **Q4** – therefore, the option **List of values** should be chosen here

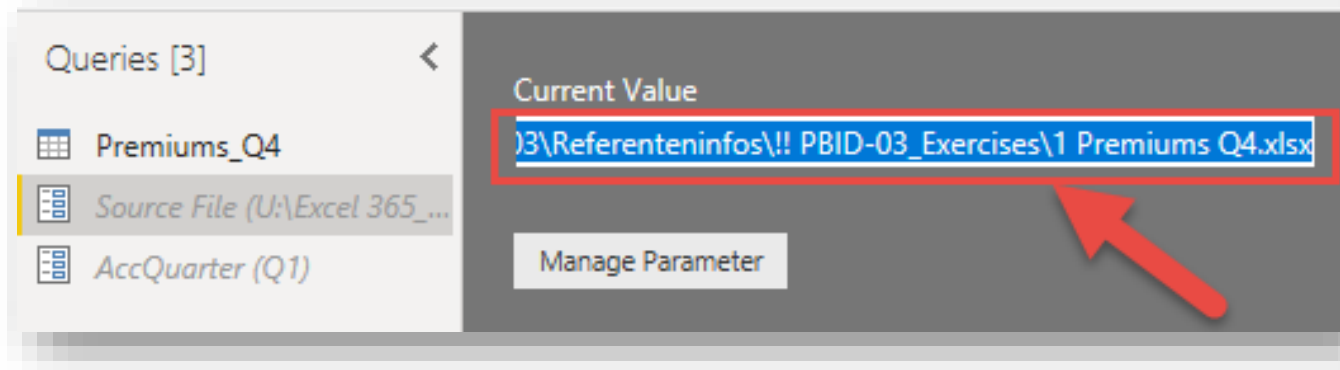


1	Q1
2	Q2
3	Q3
4	Q4
*	

- 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**

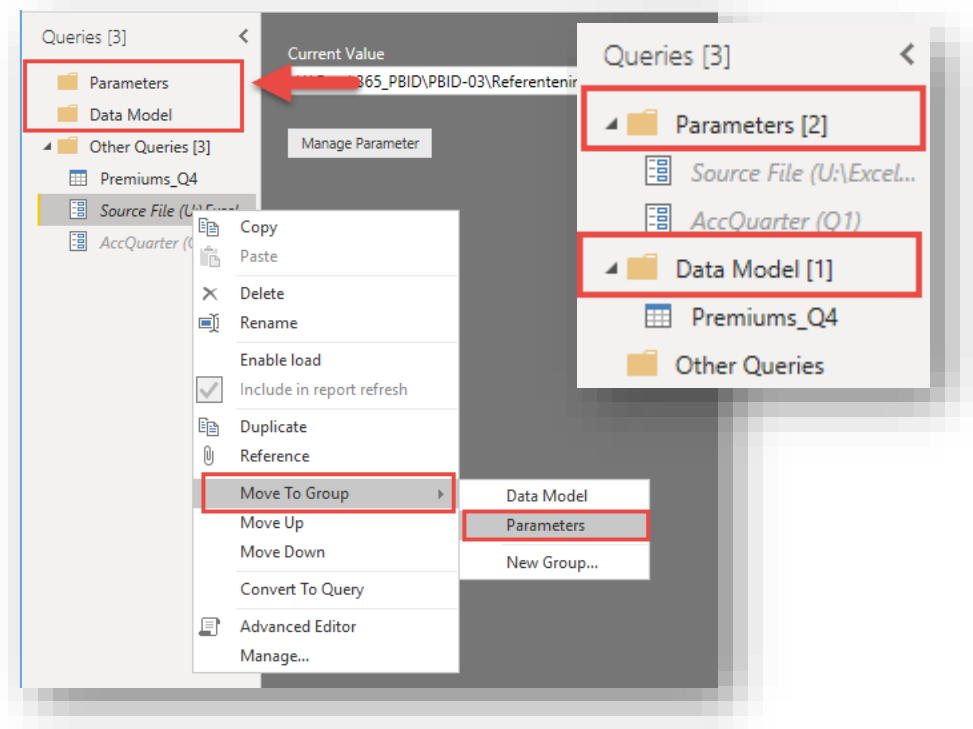
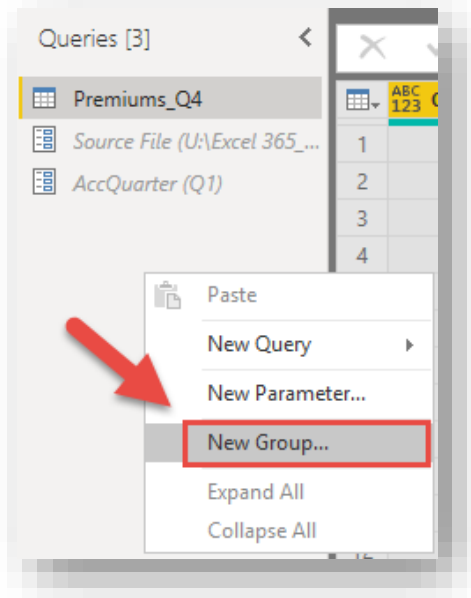


- When using a new source file, instead of editing the code, just change the parameter value
- or use a template (see next exercise)



Organize the entries

- With **Groups** you can organize your queries and parameters
- Save the **PBIX**-file



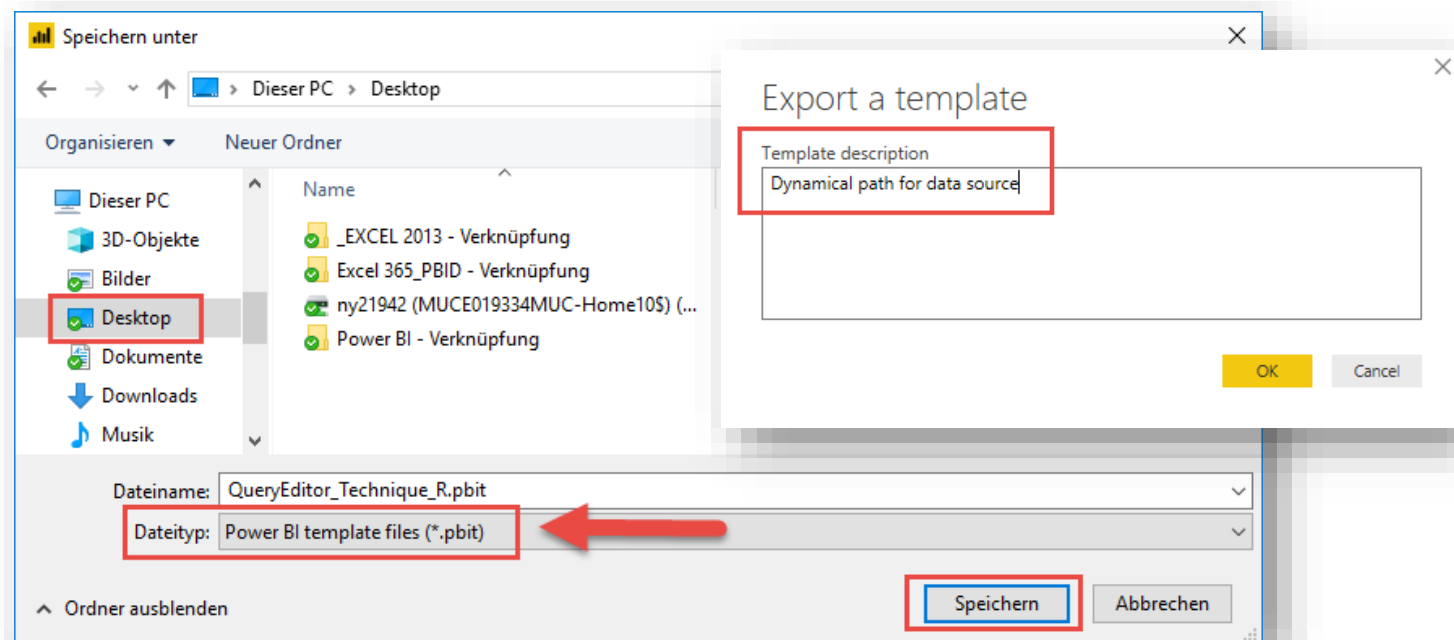
Using templates

- 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

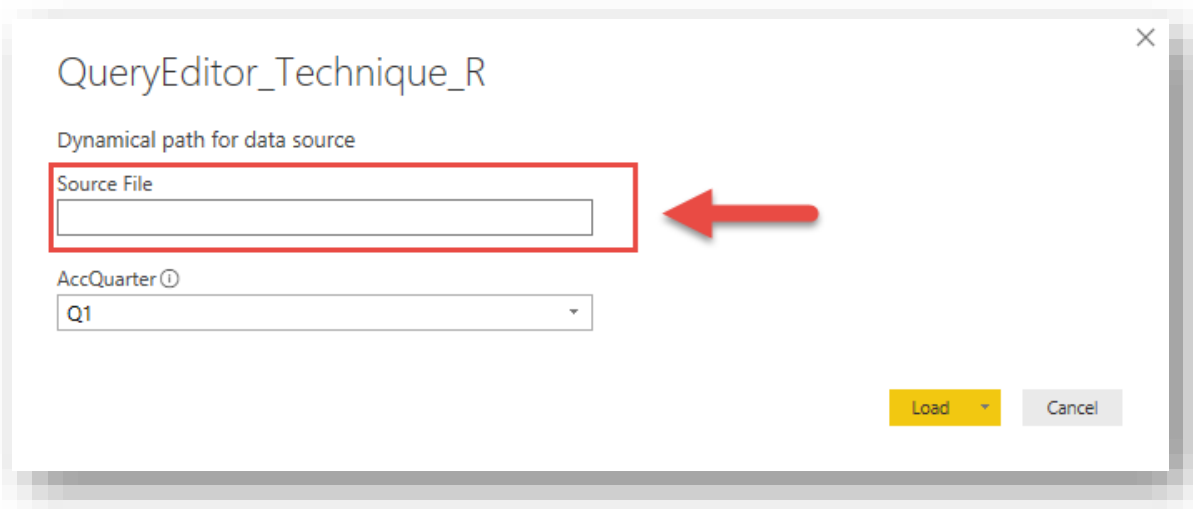
- Save the **PBIX**-file as a template (**PBIT**) on the desktop
- Enter a Template description



- 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

	A	B	C	D	E	F	G	H	I	J	K
1	SourceFileIdentifier	TreatyIdentifier	AUIIdentifier	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 PERM			LH_2018Q4_C	2018	Q4			2018 EUR
5	MUC_COMPANY_ABC_PREMIUMS	Corporate	TOTAL PERM			LH_2018Q4_C	2018	Q4			2018 EUR
6	MUC_COMPANY_ABC_PREMIUMS	Corporate	TOTAL PERM			LH_2018Q4_C	2018	Q4			2017 EUR
7	MUC_COMPANY_ABC_PREMIUMS	Corporate	TOTAL PERMANENT	10099	204.43	LH_2018Q4_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 ABC Ltd. 12/03/2018	
Branch	INDIVIDUAL
Model	SEVERE DISABILITY
Period	4TH QUARTER 2018
PDF-File	
	DEBIT CREDIT
<u>Current Account Operations</u>	
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
<u>Deposit Operations</u>	
Funds held – retained	6.497,84
Fund Held – released	9.371,38
TOTAL	0,00 2.873,54

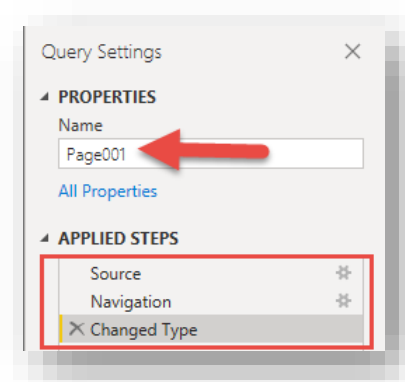
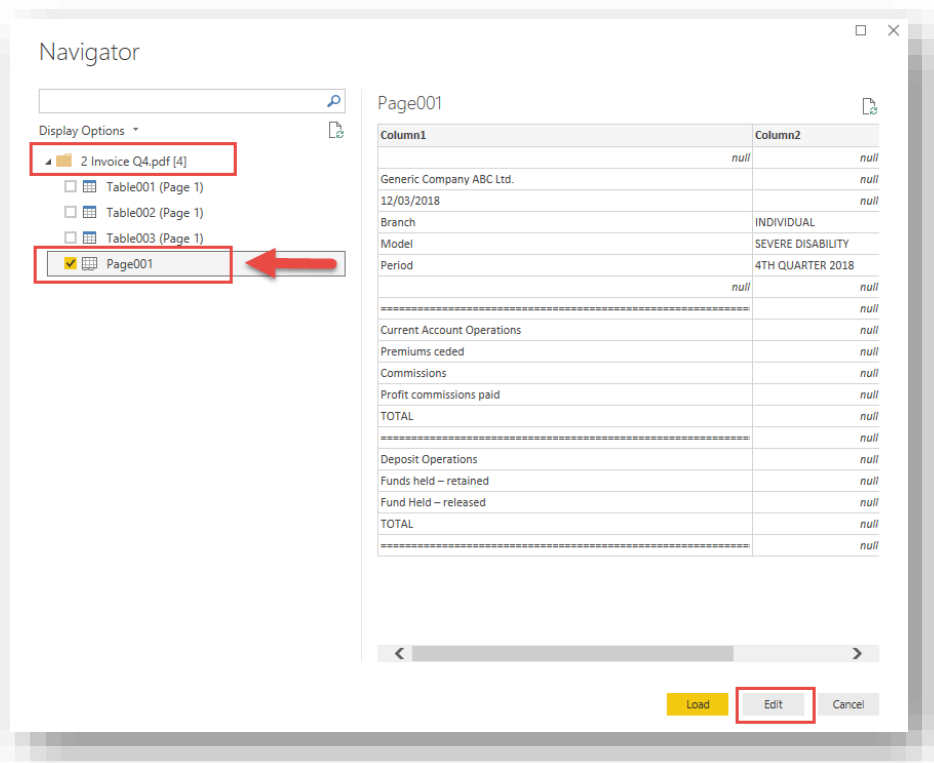
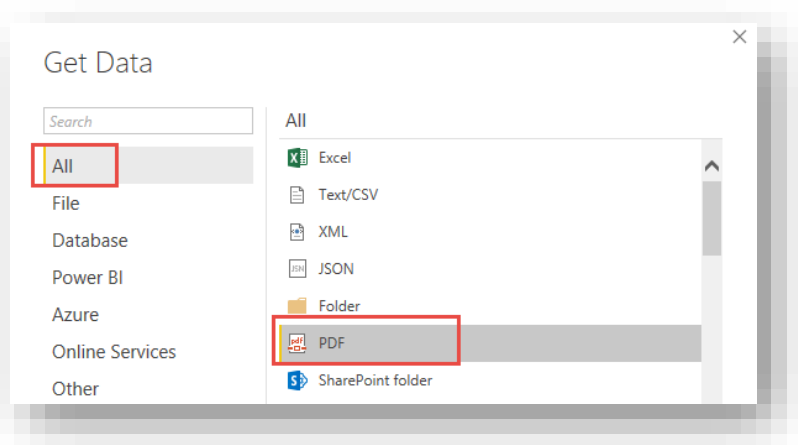
The task

- The data from both files have to be appended – therefore, the PDF file has to be shaped accordingly
- With a template it should be possible to enter a file path & other basic information for the PDF file

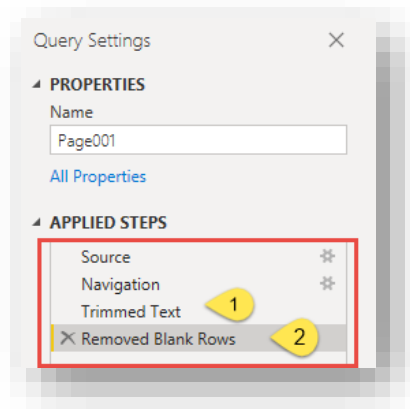


Exercise 1

- 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

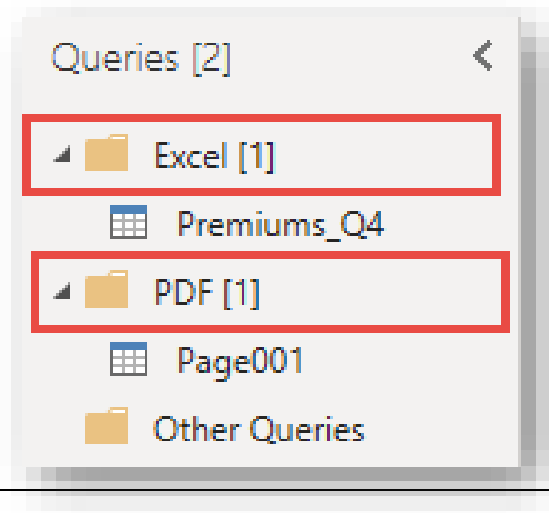


- **Delete step** Changed Type
- **Trim** text columns
- **Remove** blank rows

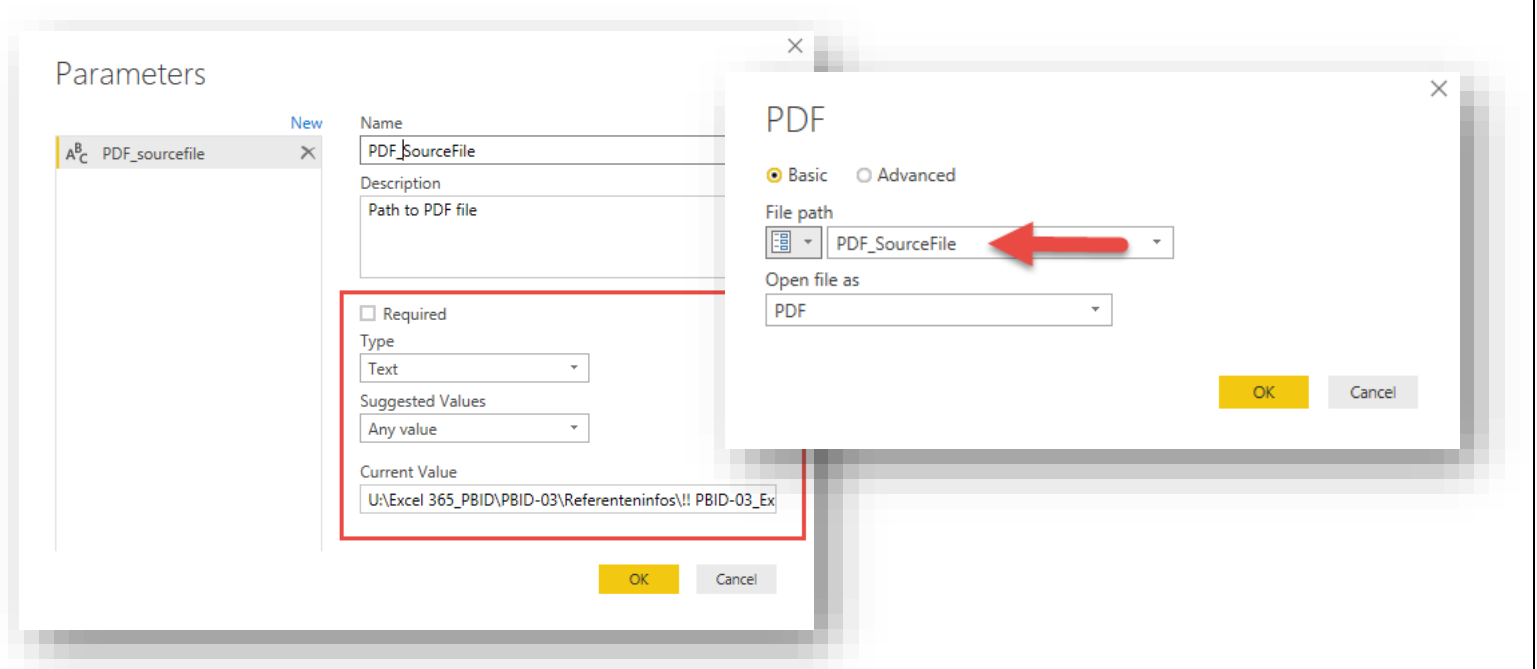


Organizing queries & parameters

- 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

	A	B	C	D	E	F	G	H	I
1	PDF:								
2									
3									
4	Code for Training Example								
5	Source = try Pdf.Tables(File.Contents(PDF_SourceFile)) otherwise if PDF_SourceFile = null then null else error Error								
6	Page1 = try Source[{"Page001"}][Data] otherwise #table(100, {}),								
7									

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

Page001

Display Options ?

```
let
    Source = Pdf.Tables(File.Contents(PDF_SourceFile)),
    Page1 = Source[[Id="Page001"]][Data],
    #"Trimmed Text" = Table.TransformColumns(Page1,{{"Column2", Text.Trim, type text}, {"Column3", Text.Trim, type text}, {"Column4", Text.Tr
    #"Removed Blank Rows" = Table.SelectRows(#"Trimmed Text", each not List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(_), {"", null
in
    #"Removed Blank Rows"
```

- 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

Page001

Display Options ?

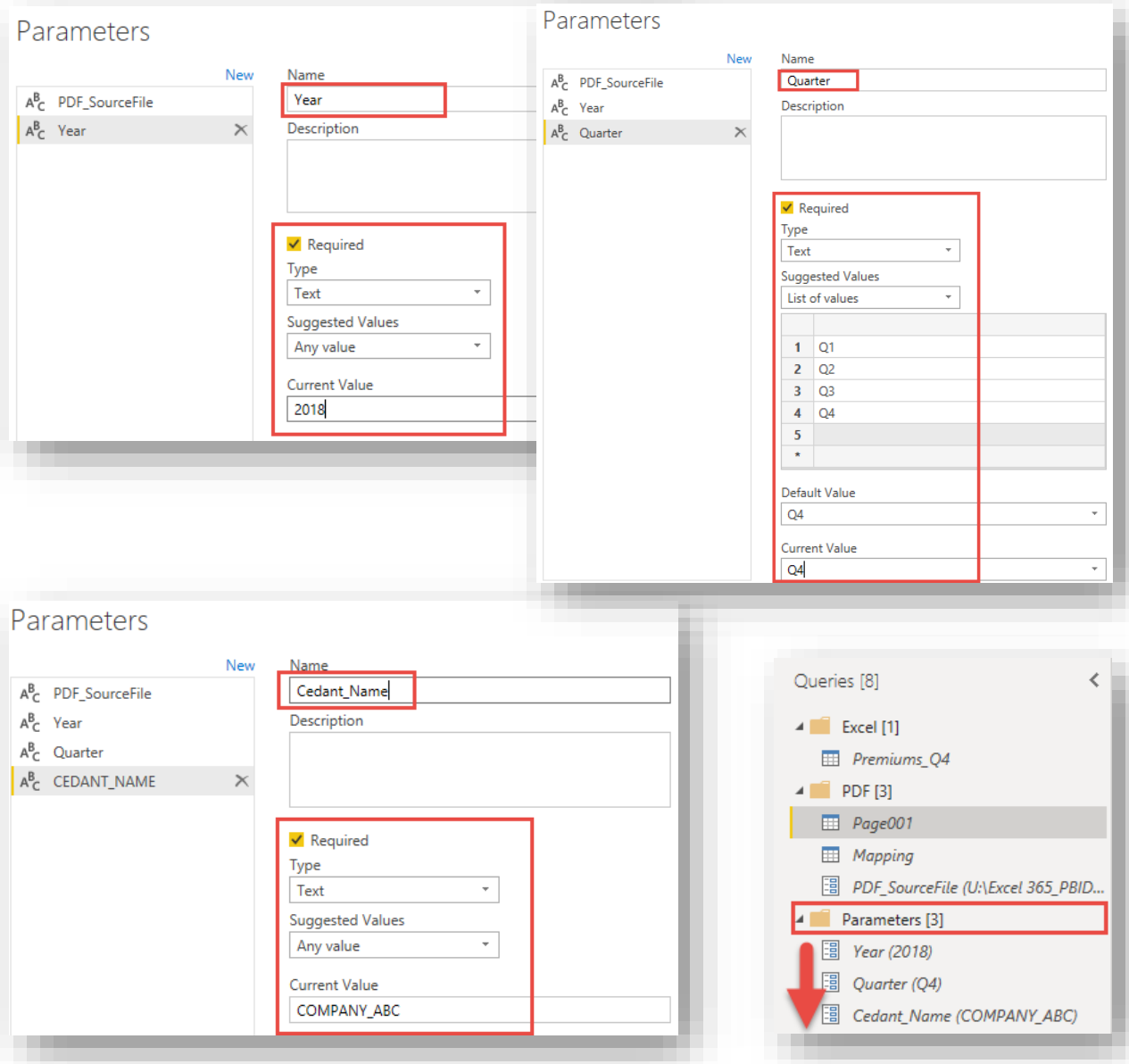
```
let
    Source = try Pdf.Tables(File.Contents(PDF_SourceFile)) otherwise if PDF_SourceFile = null then null else error Error.Record("Invalid input"),
    Page1 = try Source[[Id="Page001"]][Data] otherwise #table{100, {}},
    #"Trimmed Text" = Table.TransformColumns(Page1,{{"Column2", Text.Trim, type text}, {"Column3", Text.Trim, type text}, {"Column4", Text.Tr
    #"Removed Blank Rows" = Table.SelectRows(#"Trimmed Text", each not List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(_), {"", null
in
    #"Removed Blank Rows"
```




Exercise 2

More parameters are needed

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



The first screenshot shows the 'Parameters' dialog box with the 'Year' parameter selected. The 'Name' field is 'Year', and the 'Current Value' is '2018'. The 'Required' checkbox is checked, and the 'Type' is 'Text'.

The second screenshot shows the 'Parameters' dialog box with the 'Quarter' parameter selected. The 'Name' field is 'Quarter', and the 'Current Value' is 'Q4'. The 'Required' checkbox is checked, and the 'Type' is 'Text'. The 'Suggested Values' list contains 'Q1', 'Q2', 'Q3', and 'Q4'.

The third screenshot shows the 'Parameters' dialog box with the 'Cedant_Name' parameter selected. The 'Name' field is 'Cedant_Name', and the 'Current Value' is 'COMPANY_ABC'. The 'Required' checkbox is checked, and the 'Type' is 'Text'.

The fourth screenshot shows the 'Queries' list with the 'Parameters' group highlighted. The group contains three parameters: 'Year (2018)', 'Quarter (Q4)', and 'Cedant_Name (COMPANY_ABC)'. A red arrow points to the 'Parameters' group.

2.1 Creating the table structure

TreatyIdentifier^{xx}

Treaty-Identifier as
originally listed in invoice^{xx}

Individual^{xx}

AUIdentifier^{xx}

AU-Identifier as originally
listed in invoice^{xx}

Individual^{xx}

	A ^B _C Column1	A ^B _C Column2	A ^B _C 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”

Add Conditional Column

Add a conditional column that is computed from the other columns or values.

New column name

TreatyIdentifier

	Column Name	Operator	Value	Output
If	Column1	contains	ABC 123	Branch
Then				Column2

Add rule

Otherwise

ABC 123

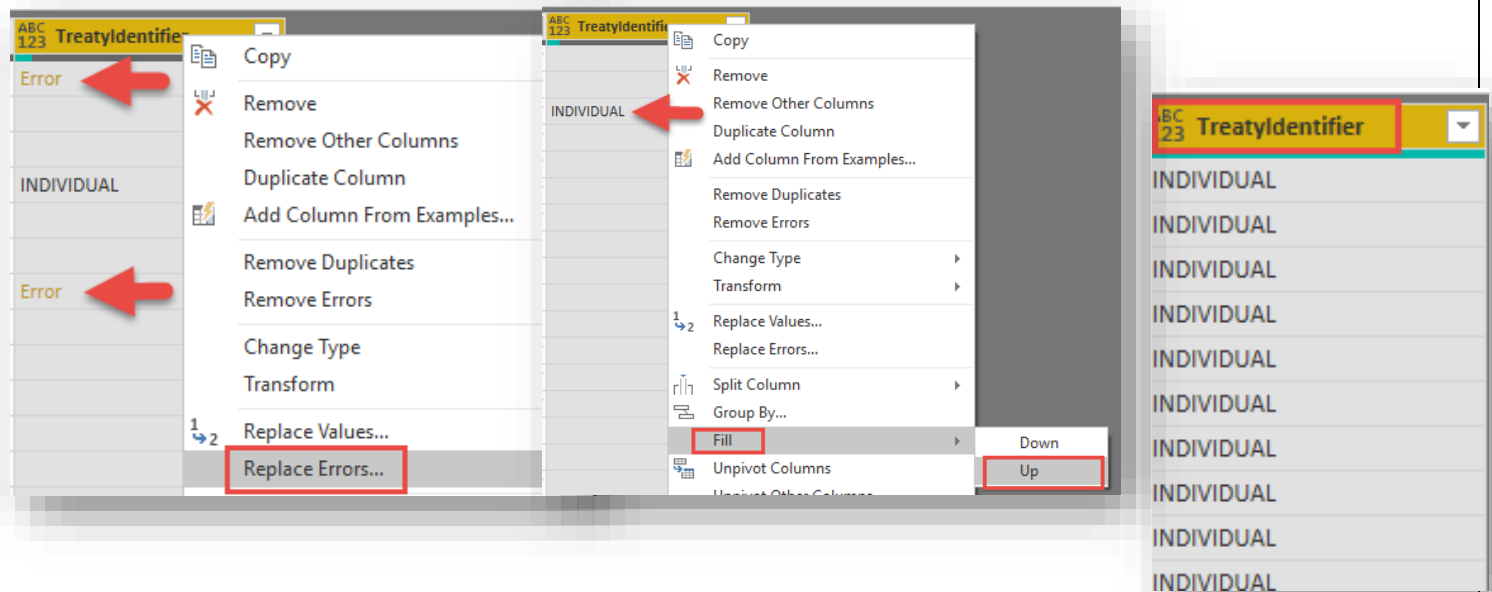
null

OK

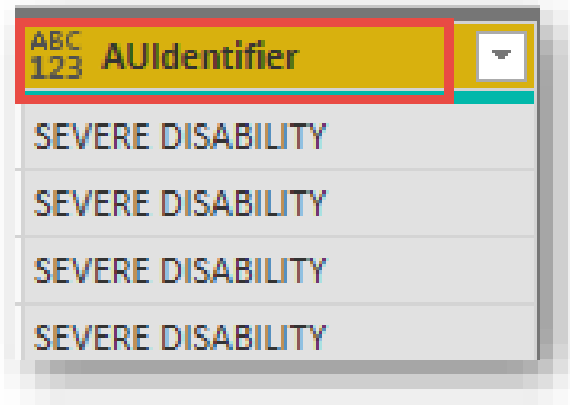
Cancel

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

Result



- In the same way create **AUIdentifier**



Amount

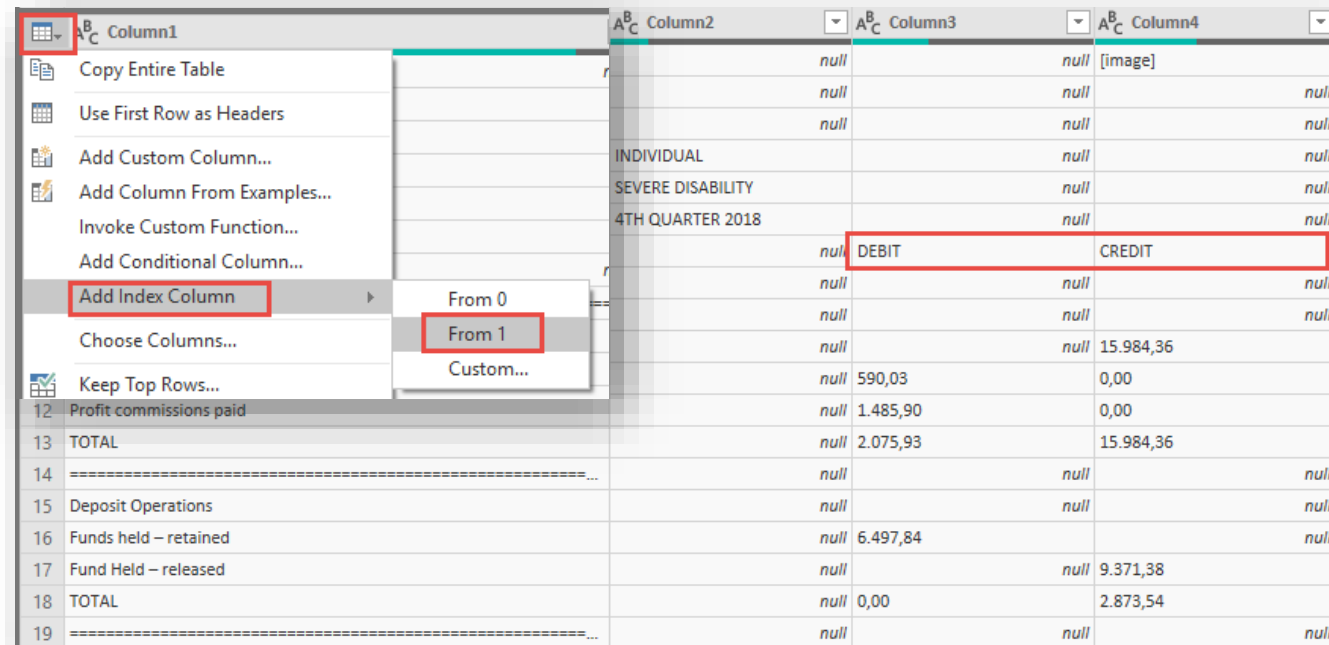
Amount as originally listed in invoice or calculated percentage

99,9

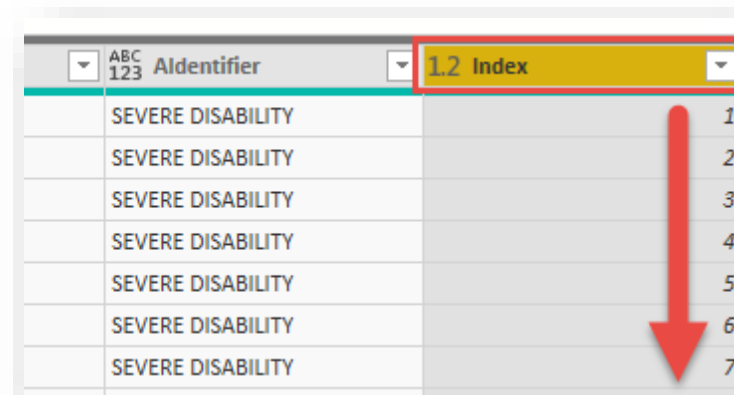


Exercise 4

- 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**



Column1	Column2	Column3	Column4
	null	null	[image]
	null	null	null
	null	null	null
INDIVIDUAL		null	null
SEVERE DISABILITY		null	null
4TH QUARTER 2018		null	null
	null	DEBIT	CREDIT
	null	null	null
	null	null	null
	null	null	15.984,36
Profit commissions paid	590,03	0,00	
	1.485,90	0,00	
TOTAL	2.075,93	15.984,36	
=====	null	null	null
Deposit Operations	null	null	null
Funds held - retained	6.497,84		null
Fund Held - released	null	9.371,38	
TOTAL	0,00	2.873,54	
=====	null	null	null



AIdentifier	Index
SEVERE DISABILITY	1
SEVERE DISABILITY	2
SEVERE DISABILITY	3
SEVERE DISABILITY	4
SEVERE DISABILITY	5
SEVERE DISABILITY	6
SEVERE DISABILITY	7

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

Add Conditional Column

Add a conditional column that is computed from the other columns or values.

New column name

Custom

Column Name	Operator	Value ①	Output ①
If	Column3	equals	DEBIT
Then			Index

Add rule

Otherwise ①

ABC 123 null

OK Cancel

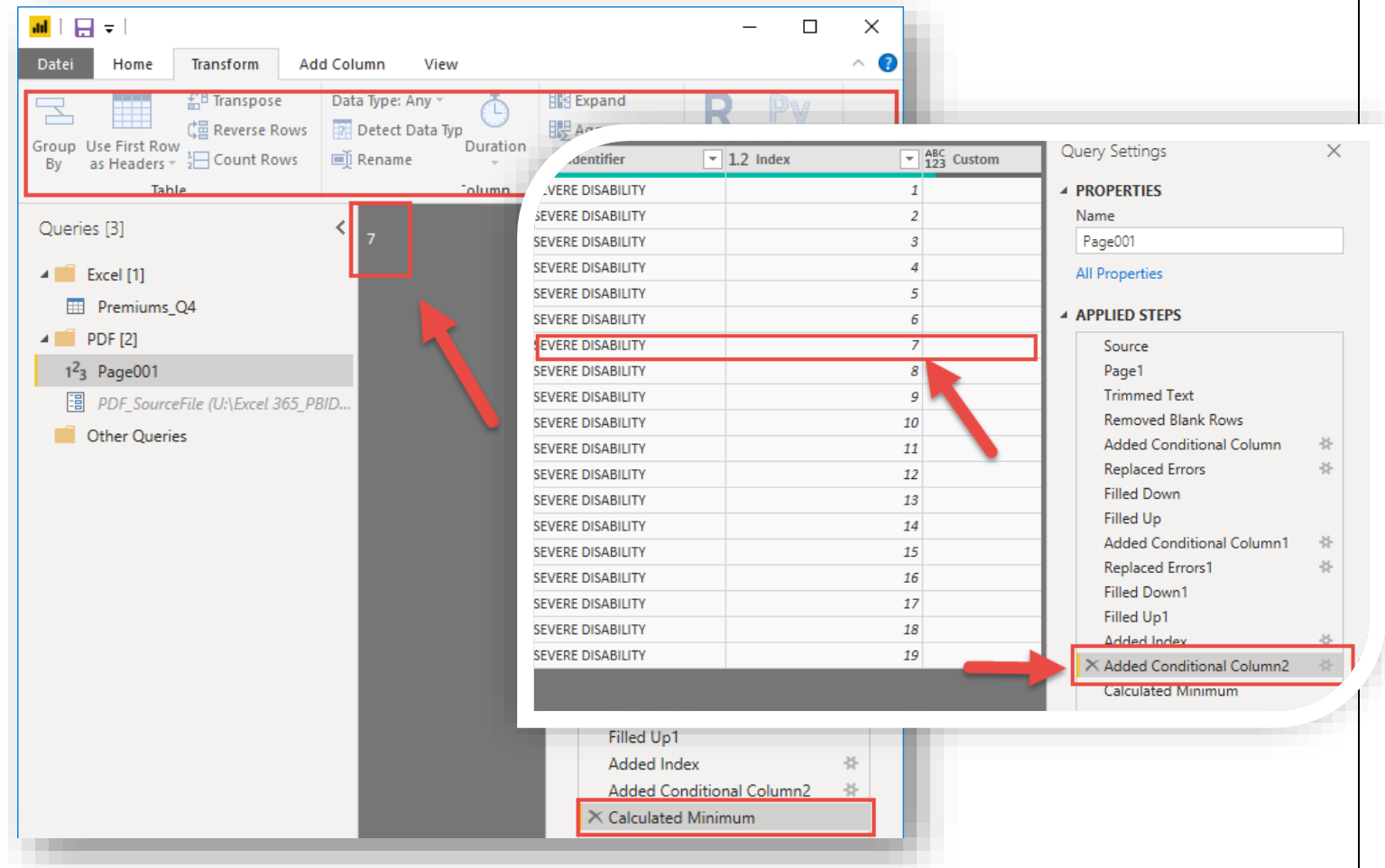
- In order to determine which is the top row (in case the header name exists more then once), a variable is needed to store the minimum value

1.2 Index	ABC 123 Custom
1	null
2	null
3	null
4	null
5	null
6	null
7	7
8	null
9	null
10	null

Statistics	Standard	Scientific
Sum		
Minimum		
Maximum		
Median		
Average		
Standard Deviation		
Count Values		
Count Distinct Values		

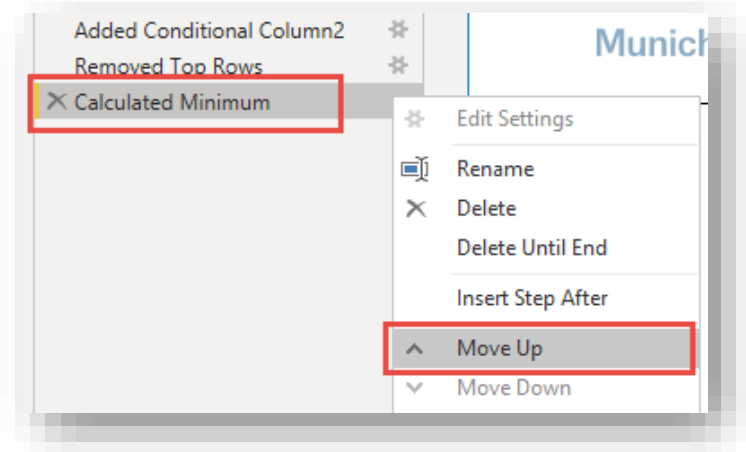
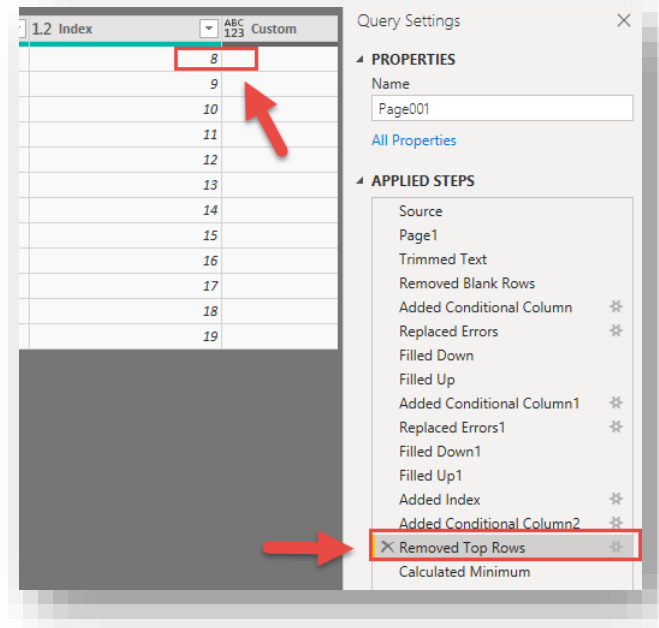
->

- 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



The screenshot displays the Power Query Editor interface. The main area shows a table with 19 rows, all containing the text 'SEVERE DISABILITY'. The 'Index' column shows row numbers from 1 to 19. A red box highlights the number '7' in the 'Index' column of the 7th row. Another red box highlights the 7th row of data. A third red box highlights the 'Added Conditional Column2' step in the 'APPLIED STEPS' list on the right, which is set to 'Calculated Minimum'. The 'Query Settings' pane on the right shows the query name 'Page001' and a list of applied steps including 'Source', 'Page1', 'Trimmed Text', 'Removed Blank Rows', 'Added Conditional Column', 'Replaced Errors', 'Filled Down', 'Filled Up', 'Added Conditional Column1', 'Replaced Errors1', 'Filled Down1', 'Filled Up1', 'Added Index', and 'Added Conditional Column2'.

- 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

```

1  #"Added Conditional Column2" = Table.AddColumn(#"Added Index", "Custom", each if [Column3]
2  #"Calculated Minimum" = List.Min(#"Added Conditional Column2"[Custom]),
   #"Removed Top Rows" = Table.Skip(#"Calculated Minimum", 7)
in
   #"Removed Top Rows"

```

Replace with
#"Added Conditional
Column 2"

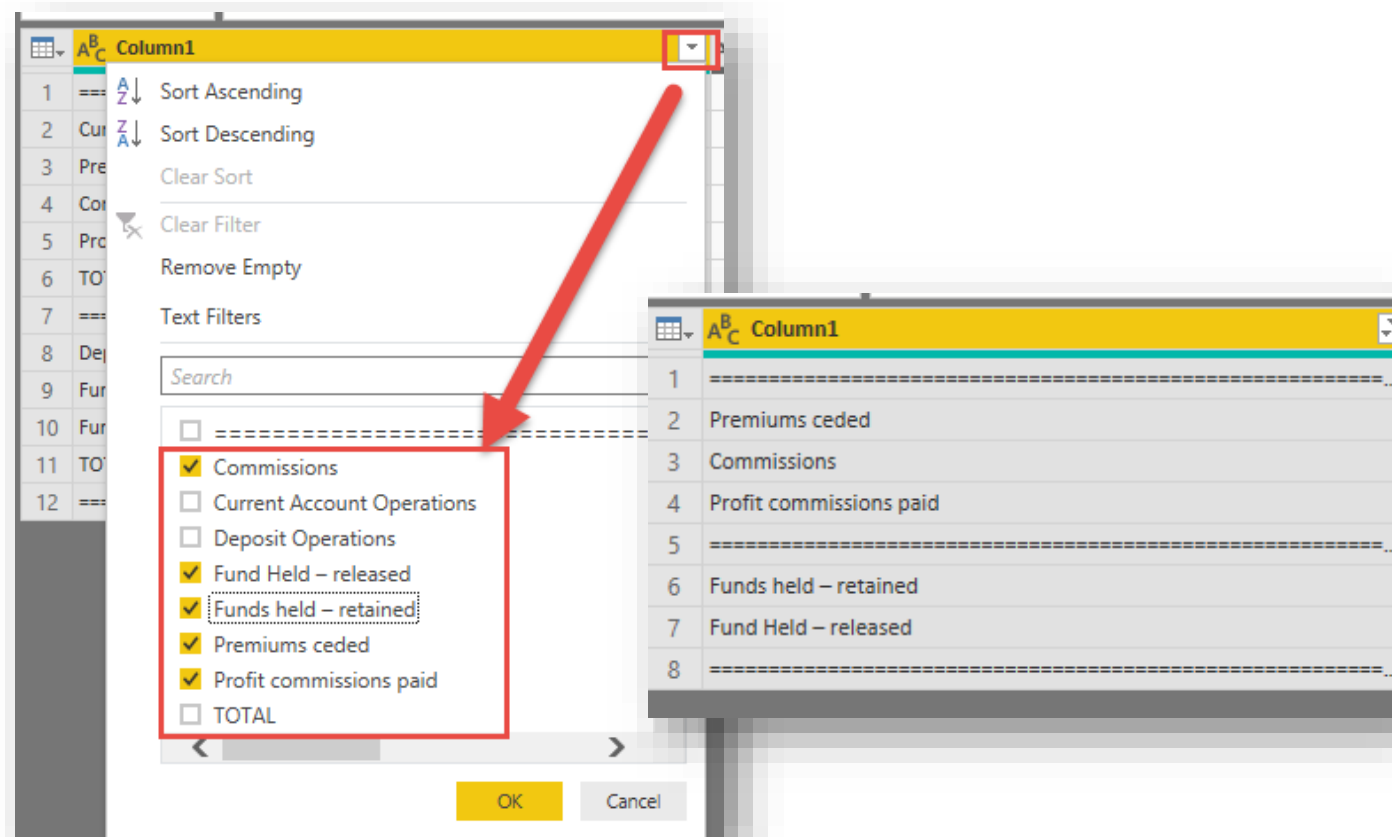
Replace with
#"Calculated
Minimum"!

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

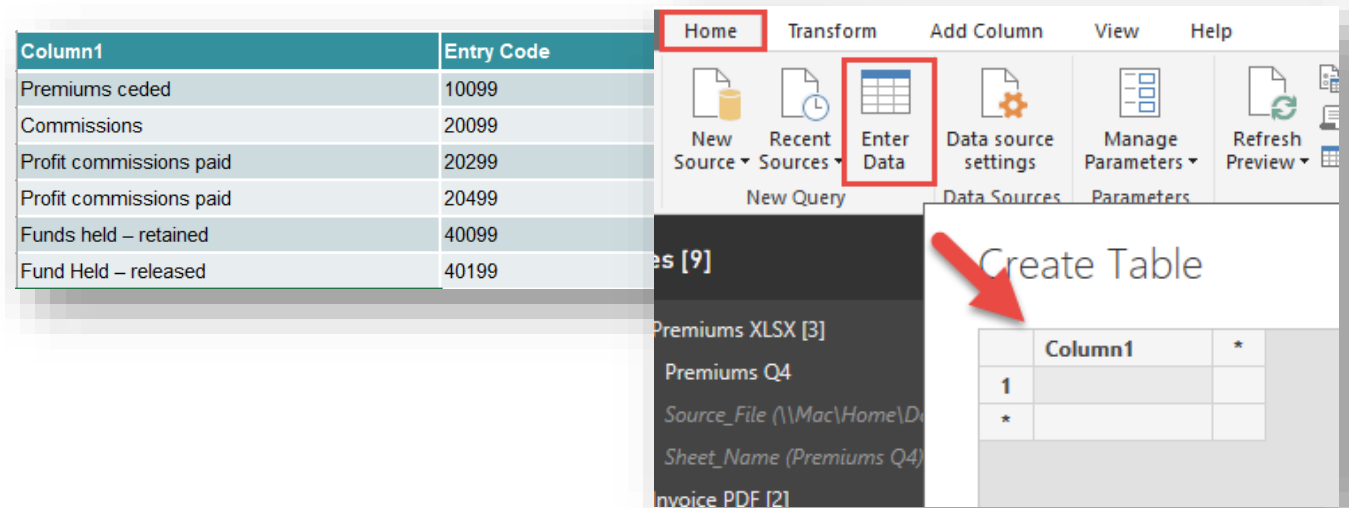
Calculated in model
(FSRI-Entry-Code)

12345

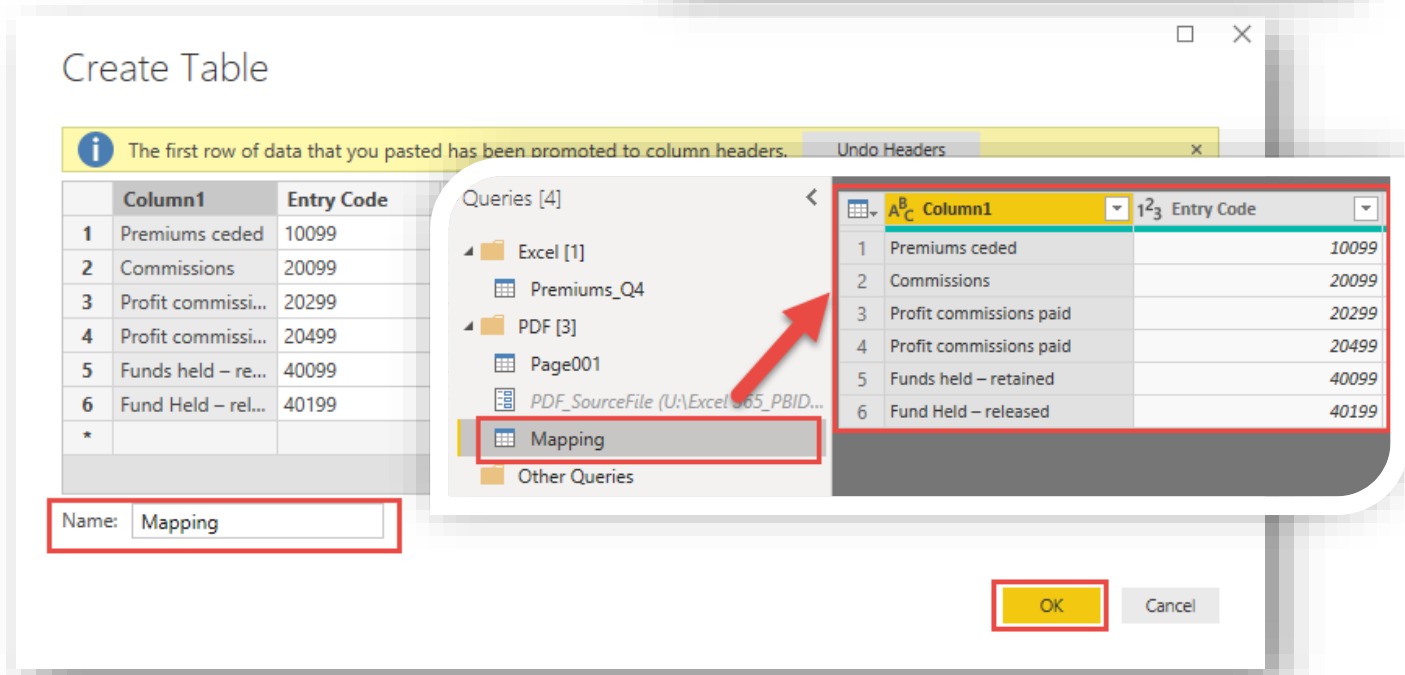


Exercise 5

- 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



Column1	Entry Code
1 Premiums ceded	10099
2 Commissions	20099
3 Profit commissions paid	20299
4 Profit commissions paid	20499
5 Funds held – retained	40099
6 Fund Held – released	40199

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

Add Conditional Column

Add a conditional column that is computed from the other columns or values.

New column name
Custom.1

	Column Name	Operator	Value ①		Output ①
If	Column1	equals	ABC 123	Premium ceded	Then ABC 123 10099
Else If	Column1	equals	ABC 123	Commissions	Then ABC 123 20099
Else If	Column1	equals	ABC 123	Profit commissions paid	Then ABC 123 20299 / 20499
Else If	Column1	equals	ABC 123	Funds held – retained	Then ABC 123 40099
Else If	Column1	equals	ABC 123	Fund Held – released	Then ABC 123 40199

Otherwise ①
ABC 123 null

Split Column by Delimiter

Specify the delimiter used to split the text column.

Select or enter delimiter
--Custom--
/

Split at
☐ Left-most delimiter
☐ Right-most delimiter
☒ Each occurrence of the delimiter

Advanced options
 Split into
☐ Columns
☒ Rows

Quote Character
"

☐ Split using special characters
Insert special character

ABC 123 Custom.1

null

null

20099

20299 / 20499

null

40099

40199

null

ABC Custom.1

null

null

20099

20299

20499

null

40099

40199

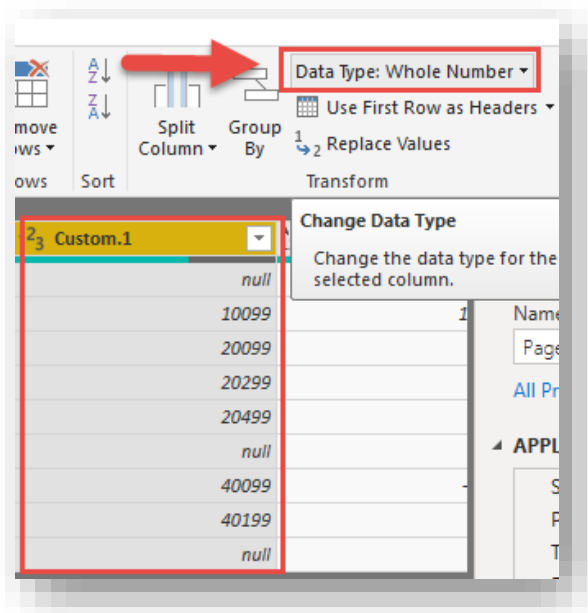
null

When you split rows, the rows are doubled in this case

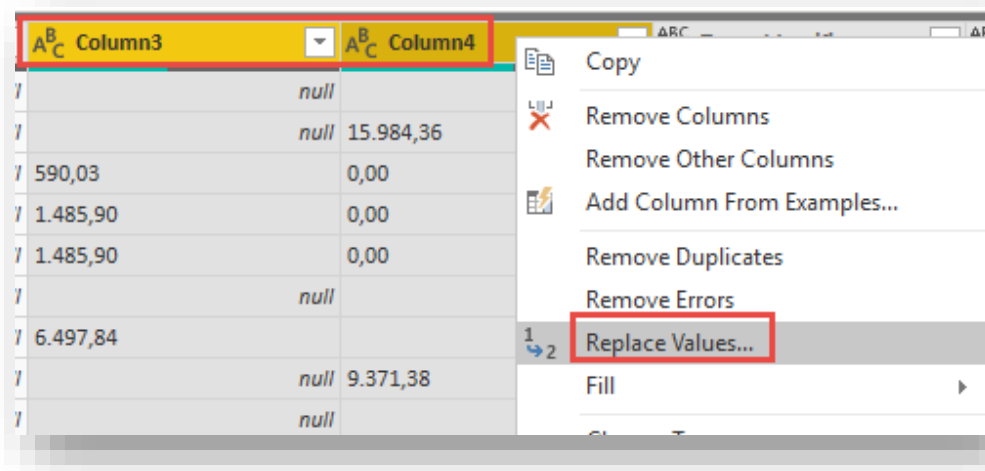
OK Cancel

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

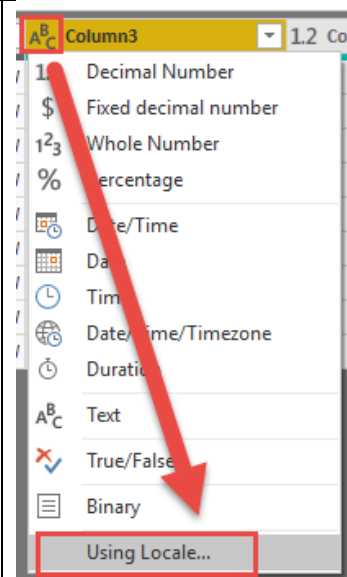
- As a last step switch the data type to **Whole number**



- 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)



Change Type with Locale

Change the data type and select the locale of origin.

Data Type

Decimal Number

Locale

German (Germany)

Sample input values:

2.100,50

-1,50

1.2 Column3	1.2 Column4
0	0
0	15984,36
590,03	0
1485,9	0
1485,9	0
0	0
6497,84	0
0	9371,38
0	0

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

Custom Column

Add a column that is computed from the other columns.

New column name

Amount_TEMP

Custom column formula ①

= [Column4] - [Column3]

[Learn about Power BI Desktop formulas](#)

✓ No syntax errors have been detected.

Available columns

Column1
Column2
Column3
Column4
TreatyIdentifier
AUIdentifier
Index

<< Insert

OK

Cancel

ABC 123	Custom.2
	0
	15984,36
	-590,03
	-1485,9
	-1485,9
	0
	-6497,84
	9371,38
	0

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

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

- You can use the **Mapping table** to correct the calculations

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

×

Merge

Select a table and matching columns to create a merged table.

Page001

	Column3	Column4	TreatyIdentifier	AUIdentifier	Index	Custom	Custom.1	Amount.Temp
null	0	0	INDIVIDUAL	SEVERE DISABILITY	8	null	null	
null	0	15984,36	INDIVIDUAL	SEVERE DISABILITY	10	null	null	-15984,3
null	590,03	0	INDIVIDUAL	SEVERE DISABILITY	11	null	20099	590,0
null	1485,9	0	INDIVIDUAL	SEVERE DISABILITY	12	null	20299	1485,
null	1485,9	0	INDIVIDUAL	SEVERE DISABILITY	12	null	20499	1485,

Mapping

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

Join Kind

Left Outer (all from first, matching from second)

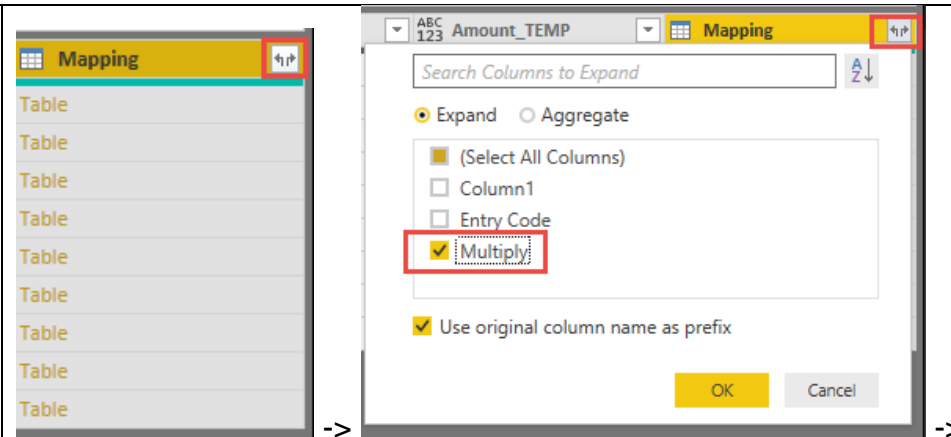
☐ Use fuzzy matching to perform the merge

> Fuzzy merge options

OK

Cancel

- 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

ABC 123	Amount_TEMP	Mapping.Multiply
	0	null
	0	null
	15984,36	1
	-590,03	1
	-1485,9	-1
	-1485,9	1
	-6497,84	
	9371,38	
	0	

Custom Column

Add a column that is computed from the other columns.

New column name

Amount

Custom column formula ⓘ

= [Amount_TEMP]*[Mapping.Multiply]

- Result



ABC 123 Amount
null
-15984,36
null
590,03
-1485,9
1485,9
-6497,84
-9371,38
null

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

Custom Column

Add a column that is computed from the other columns.

New column name

SourceFileIdentifier

Custom column formula ⓘ

= "MUC_COMPANY_ABC_PDF"

Available columns

Column1
Column2
Column3
Column4
TreatyIdentifier
AUIdentifier
Index

<< Insert

[Learn about Power BI Desktop formulas](#)

✓ No syntax errors have been detected.

OK

Cancel

->

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

<div><div>TreatyIdentifier^α</div><div>Treaty-Identifier-as- originally-listed-in-invoice^α</div><div>Individual^α</div></div>	<div>ABC 123 TreatyIdentifier</div> <div>INDIVIDUAL</div> <div>INDIVIDUAL</div> <div>INDIVIDUAL</div> <div>INDIVIDUAL</div> <div>INDIVIDUAL</div> <div>INDIVIDUAL</div> <div>INDIVIDUAL</div> <div>INDIVIDUAL</div> <div>INDIVIDUAL</div> <div>INDIVIDUAL</div> <div>INDIVIDUAL</div>	- already created
<div><div>AUIdentifier^α</div><div>AU-Identifier-as-originally- listed-in-invoice^α</div><div>Individual^α</div></div>	<div>ABC 123 AUIdentifier</div> <div>SEVERE DISABILITY</div> <div>SEVERE DISABILITY</div> <div>SEVERE DISABILITY</div> <div>SEVERE DISABILITY</div> <div>SEVERE DISABILITY</div> <div>SEVERE DISABILITY</div> <div>SEVERE DISABILITY</div> <div>SEVERE DISABILITY</div> <div>SEVERE DISABILITY</div> <div>SEVERE DISABILITY</div>	- already created
<div><div>Entry^αCode^α</div><div>Calculated-in-model[↔] (FSRI-Entry-Code)^α</div><div>12345^α</div></div>	<div>1²3 Custom.1</div> <div>null</div> <div>null</div> <div>null</div> <div>20099</div> <div>20299</div> <div>20499</div> <div>40099</div> <div>40199</div> <div>null</div>	<div>1²3 Entry Code</div> <div>null</div> <div>null</div> <div>null</div> <div>20099</div> <div>20299</div> <div>20499</div> <div>40099</div> <div>40199</div> <div>null</div> <div>-></div>

<div data-bbox="138 204 436 459"> <p>Amount^α</p> <p>Amount as originally listed in invoice or calculated percentage^α</p> <p>99,9^α</p> </div>	<div data-bbox="631 172 891 523"> <p>ABC 123 Amount</p> <table> <tr><td></td><td>null</td></tr> <tr><td></td><td>null</td></tr> <tr><td></td><td>15984,36</td></tr> <tr><td></td><td>-590,03</td></tr> <tr><td></td><td>1485,9</td></tr> <tr><td></td><td>-1485,9</td></tr> <tr><td></td><td>6497,84</td></tr> <tr><td></td><td>9371,38</td></tr> <tr><td></td><td>null</td></tr> </table> </div> <p>- already created</p>		null		null		15984,36		-590,03		1485,9		-1485,9		6497,84		9371,38		null
	null																		
	null																		
	15984,36																		
	-590,03																		
	1485,9																		
	-1485,9																		
	6497,84																		
	9371,38																		
	null																		
<div data-bbox="138 561 353 778"> <p>Acct Text</p> <p>LH_[Year&Quarter]_[Cedant Name]</p> <p>LH_2018Q4_MRE</p> </div>	<div data-bbox="660 555 1579 981"> <h3>Custom Column</h3> <p>Add a column that is computed from the other columns.</p> <p>New column name</p> <p>Acct Text</p> <p>Custom column formula ①</p> <p>= "LH_" & Year & Quarter & "_" & Cedant_Name</p> </div> <div data-bbox="1765 587 2033 954"> <p>ABC 123 Acct Text</p> <table> <tr><td></td><td>LH_2018Q4_COMPANY_ABC</td></tr> <tr><td></td><td>LH_2018Q4_COMPANY_ABC</td></tr> <tr><td></td><td>LH_2018Q4_COMPANY_ABC</td></tr> <tr><td></td><td>LH_2018Q4_COMPANY_ABC</td></tr> <tr><td></td><td>LH_2018Q4_COMPANY_ABC</td></tr> <tr><td></td><td>LH_2018Q4_COMPANY_ABC</td></tr> <tr><td></td><td>LH_2018Q4_COMPANY_ABC</td></tr> <tr><td></td><td>LH_2018Q4_COMPANY_ABC</td></tr> <tr><td></td><td>LH_2018Q4_COMPANY_ABC</td></tr> </table> </div> <p>-></p>		LH_2018Q4_COMPANY_ABC		LH_2018Q4_COMPANY_ABC		LH_2018Q4_COMPANY_ABC		LH_2018Q4_COMPANY_ABC		LH_2018Q4_COMPANY_ABC		LH_2018Q4_COMPANY_ABC		LH_2018Q4_COMPANY_ABC		LH_2018Q4_COMPANY_ABC		LH_2018Q4_COMPANY_ABC
	LH_2018Q4_COMPANY_ABC																		
	LH_2018Q4_COMPANY_ABC																		
	LH_2018Q4_COMPANY_ABC																		
	LH_2018Q4_COMPANY_ABC																		
	LH_2018Q4_COMPANY_ABC																		
	LH_2018Q4_COMPANY_ABC																		
	LH_2018Q4_COMPANY_ABC																		
	LH_2018Q4_COMPANY_ABC																		
	LH_2018Q4_COMPANY_ABC																		
<div data-bbox="123 1045 324 1327"> <p>Acct-Year^α</p> <p>Client Accounting Year^α</p> <p>2018^α</p> </div>	<div data-bbox="660 1061 1232 1396"> <h3>Custom Column</h3> <p>Add a column that is computed from the other columns.</p> <p>New column name</p> <p>Acct Year</p> <p>Custom column formula ①</p> <p>= Year</p> </div> <div data-bbox="1406 1072 1624 1359"> <p>ABC 123 Acct Year</p> <table> <tr><td></td><td>2018</td></tr> <tr><td></td><td>2018</td></tr> <tr><td></td><td>2018</td></tr> <tr><td></td><td>2018</td></tr> <tr><td></td><td>2018</td></tr> <tr><td></td><td>2018</td></tr> <tr><td></td><td>2018</td></tr> <tr><td></td><td>2018</td></tr> <tr><td></td><td>2018</td></tr> </table> </div> <p>-></p>		2018		2018		2018		2018		2018		2018		2018		2018		2018
	2018																		
	2018																		
	2018																		
	2018																		
	2018																		
	2018																		
	2018																		
	2018																		
	2018																		

<div data-bbox="129 204 344 459"> <p>Acct-Quarter^{xx}</p> <p>Client- Accounting- Quarter^{xx}</p> <p>Q4^{xx}</p> </div>	<div data-bbox="667 245 1294 549"> <h3>Custom Column</h3> <p>Add a column that is computed from the other columns.</p> <p>New column name</p> <p>Acct Quarter</p> <p>Custom column formula ⓘ</p> <p>= Quarter</p> </div> <div data-bbox="1451 204 1814 549"> <div>ABC 123 Acct Quarter</div> <div>Q4</div> <div>Q4</div> <div>Q4</div> <div>Q4</div> <div>Q4</div> <div>Q4</div> </div> <p>-></p>
<div data-bbox="129 663 322 868"> <p>UY^{xx}</p> <p>UY-as-in- original-file^{xx}</p> </div>	<div data-bbox="667 692 1249 979"> <h3>Custom Column</h3> <p>Add a column that is computed from the other columns.</p> <p>New column name</p> <p>UY</p> <p>Custom column formula ⓘ</p> <p>= null</p> </div> <div data-bbox="1406 663 1787 979"> <div>ABC 123 UY</div> <div>null</div> <div>null</div> <div>null</div> <div>null</div> <div>null</div> </div> <p>-></p>
<div data-bbox="129 1094 322 1378"> <p>OY^{xx}</p> <p>OY-as-in- original-file^{xx}</p> <p>2017^{xx}</p> </div>	<div data-bbox="667 1123 1249 1410"> <h3>Custom Column</h3> <p>Add a column that is computed from the other columns.</p> <p>New column name</p> <p>OY</p> <p>Custom column formula ⓘ</p> <p>= Null</p> </div> <div data-bbox="1415 1094 1747 1410"> <div>ABC 123 OY</div> <div>null</div> <div>null</div> <div>null</div> <div>null</div> <div>null</div> </div> <p>-></p>

Currency
Currency-of-amount-
(ISO3-Code)
EUR

Custom Column

Add a column that is computed from the other columns.

New column name

Currency

Custom column formula ①

= "EUR"

ABC 123 Currency
EUR
EUR
EUR
EUR
EUR

->

- 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**!

ABC 123 SourceFileIdentifier	ABC 123 TreatyIdentifier	ABC 123 AllIdentifier	123 Entry Code	ABC 123 Amount
1 MUC_COMPANY_ABC_PDF	INDIVIDUAL			null
2 MUC_COMPANY_ABC_PDF	INDIVIDUAL			null
3 MUC_COMPANY_ABC_PDF	INDIVIDUAL		10099	-15984,36
4 MUC_COMPANY_ABC_PDF	INDIVIDUAL		20099	590,03
5 MUC_COMPANY_ABC_PDF	INDIVIDUAL		20299	-1485,9
6 MUC_COMPANY_ABC_PDF	INDIVIDUAL		20499	1485,9
7 MUC_COMPANY_ABC_PDF	INDIVIDUAL		40099	-6497,84
8 MUC_COMPANY_ABC_PDF	INDIVIDUAL		40199	-9371,38
9 MUC_COMPANY_ABC_PDF	INDIVIDUAL		null	null

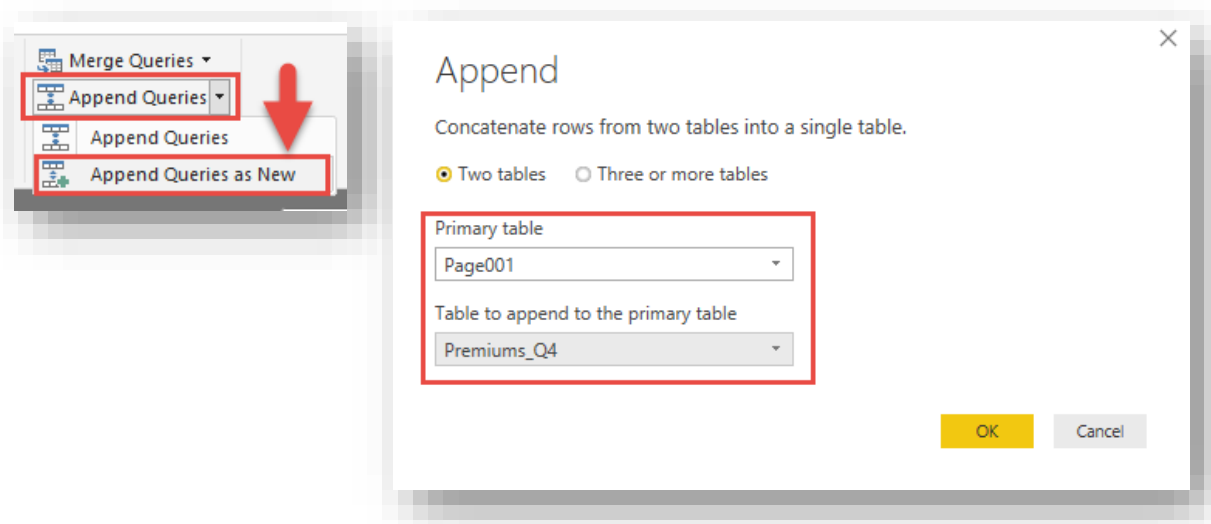
ABC 123 Acct Text	ABC 123 Acct Year	ABC 123 Acct Quarter	ABC 123 UY	ABC 123 OY	ABC 123 Currency
LH_2018Q4_COMPANY_ABC	2018	Q4		null	null EUR
LH_2018Q4_COMPANY_ABC	2018	Q4		null	null EUR
LH_2018Q4_COMPANY_ABC	2018	Q4		null	null EUR
LH_2018Q4_COMPANY_ABC	2018	Q4		null	null EUR
LH_2018Q4_COMPANY_ABC	2018	Q4		null	null EUR
LH_2018Q4_COMPANY_ABC	2018	Q4		null	null EUR
LH_2018Q4_COMPANY_ABC	2018	Q4		null	null EUR
LH_2018Q4_COMPANY_ABC	2018	Q4		null	null EUR

2.3 Appending Queries & Testing Template



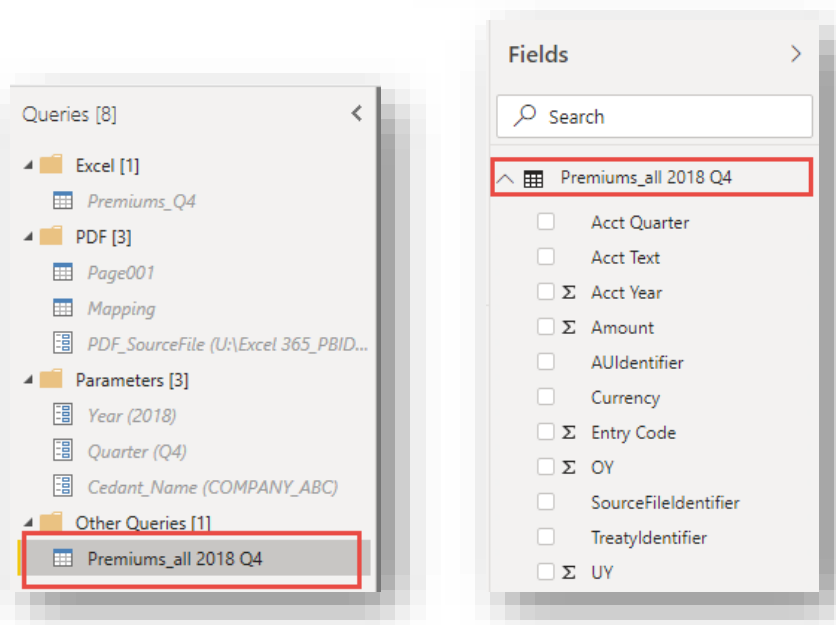
Exercise 7

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



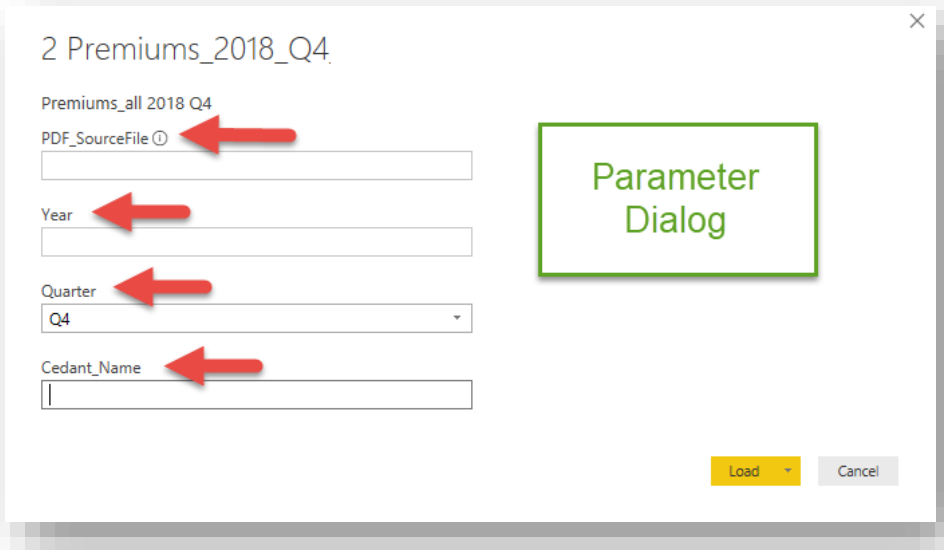
Change the setting *Enable Load*

- 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





2 Premiums_2018_Q4

Premiums_all 2018 Q4

PDF_SourceFile 

Year 

Quarter  Q4

Cedant_Name 

Parameter Dialog

Load Cancel

Stay tuned ...

