

PowerQuery

Indhold

PowerQuery	1
Man kan importere flere filer af samme type fra mappe. Tilføjes ny fil senere udføres samme beregninger automatisk	3
Kombinerer række-værdier fra 3 kolonner, fx ved at gange værdierne sammen	5
Lav ny kolonne, med custom column from examples: udfyldes første felt, så kan Excel se mønsteret	6
**** Strings-manipulation	6
=TRIM(text) , =FJERN.OVERFLØDIGE.BLANKE(celle_ref) i tekst-kolonne i Excel	6
Search for string	7
Sammensætning af string og tal: "Q" & Number.ToText([Kvartal])&"-" & Number.ToText([År])	8
Autodetect datatyper:	8
Advanced editor	8
Ændringer på hele tabelniveau, vælg PQ editor, tabelfelt	9
Ændringer på kolonne niveau: marker kolonne, højreklik	10
**** Date format transformations	11
Fik ikke nemt oversat fra eng format mm/dd/yyyy til dansk,	11
Problem fikset ved at få udskrevet mm,dd,yyyy(udtrukket med strings fkt) i 3 kolonner og så konstruere korrekt data	11
Man kan fjerne kolonner uden af ødelægge forudgående beregninger.	11
Til sidst kan man altså godt fjerne hjælpekolonner, før den endelig fejl 'udgives'	11
Date data type	11
M code for date transformations	13
Locale settings	15
Pivot-tabeller med /måneder/kvartal/år : opdelingen, sker bedste direkte i Excel	23
Kolonne med formel, der inddrager værdier fra andre kolonner, i new custom column	24
Merge, join to tabeller , 'vandret'	24
Append: rækkerne i tabel2 tilføjes rækkerne i 1.tabel, 'lodret'	25
**** Tabeller med gruppering, aggregeret værdier	26
Info om product-salg, for alle customers:	28

**** Filtrering af felt kan starte med UI, kan evt. finjusteres med Power Query- M; filtrering af rækker:	29
Visuel hjælp til hver M-kommando.....	29
Pivot Tabel-form : hver række indeholder flere rekords	30
Unpivot data, data på normaliseret form: har kun een rekord per række	30
Vigtige transformationstyper: fremgår af Query Settings-listen t.h.	31
Man kan nemt afprøve og fortryde transformation ved at slette proces-trinnet fra listen.....	31
-Remove columns, rows, blanks.....	31
-Convert data types – text, numbers, dates.....	31
-Split or merge columns: nemt fx at lave nye kolonner med fornavn,efternavn	31
Sort & filter columns	32
Add calculated columns	32
Aggregate or summarize data kan laves med PowerQuery og lidt besværligt med pivot-tabel i Excel	32
'Advanced Group By' for grupper med flere felter, med flere aggregeringer	32
Find & replace text/value	33
Man kan søg/erstat i rå-excel-data, og vælge 'hele projektmappen', så slår det igennem i alle afledte tabeller, ellers blot selv arket; det er faktisk nemmere end i PQ	33
I PowerQuery: Right-click on a column -> Select Replace Values	33
*** Nested Calculations in PQ: lave grupper i een kolonne, så ranking indenfor hver gruppe, så expand grupperne i kolonnen	33
Function I M:	36
Here's the full M code:	38
**** Bedste praksis -tips.....	39
Vælg den rigtige connector til data-source	39
Filtr tidligt	40
Filtrering med M i editor:	41
Lav dyre handlinger, som skal gennemgå hele tabellen, sidst	42
Arbejd midlertidigt mod et undersæt af dine data, med fx 'bevar første 100 rækker'	43
Der arbejdes med max 1000 rækker i PQ, hele tabellen generes først efter save tabel	43
Brug de korrekte datatyper	43
Udforsk dine data, lav fx Deskriptiv statistik over tal-kolonner	44
TÆL.HVIS(txt-kolonne;"") for tekst-kolonner kan finde antal tomme felter	44

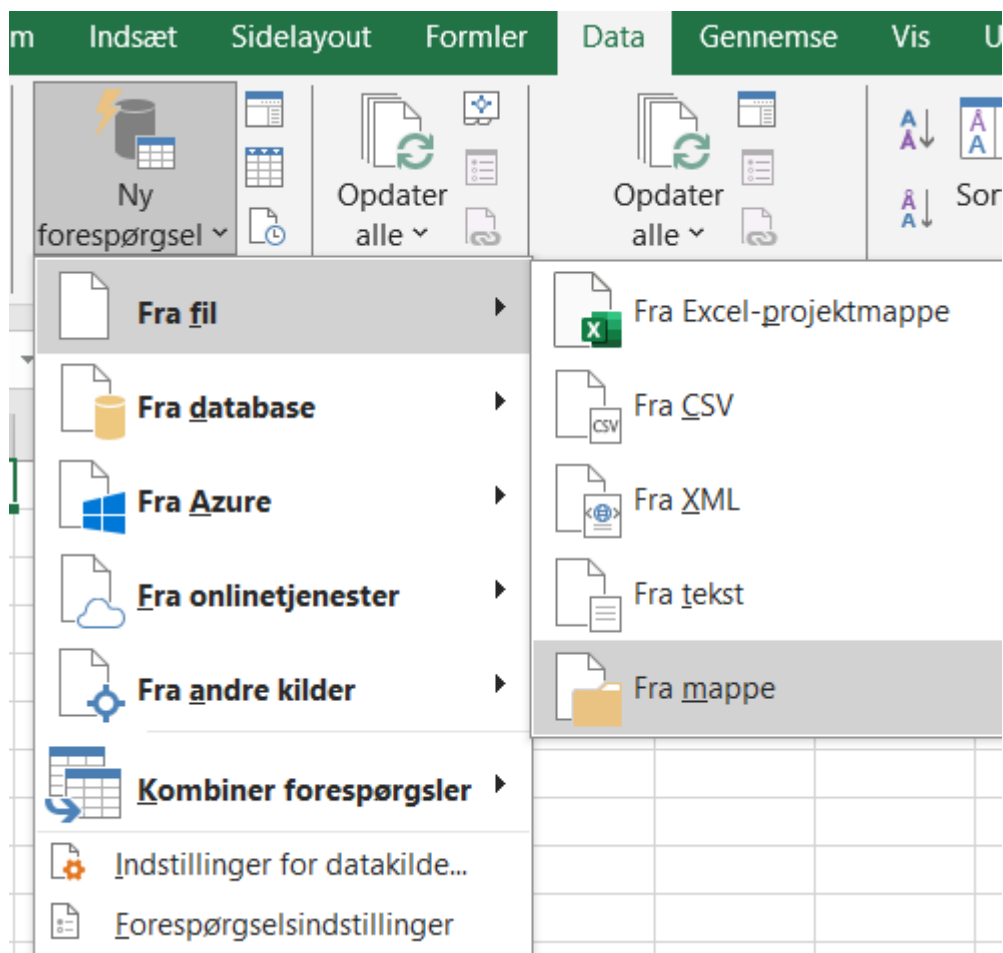
R og Python har dog bedre værktøjer til problemdata:.....	45
Data-Profileringsværktøj, ikke i min Excelversion	45
Tag en modulopbygget tilgang: opdel kompleks query i mindre dele	45
Opdel Queries i grupper, især ved store projekter	48
Fremtidssikrede forespørgsler	49
Brug parametre	50
Opret funktioner, der kan genbruges.....	54
**** Datamodellering.....	57
Modelværktøjet	59
Modeller kan bruges til at lave pivottabeller i Excel, ligesom i PowerBI.....	59
PQ in PowerBI.....	62

Man kan importere flere filer af samme type fra mappe. Tilføjes ny fil senere udføres samme beregninger automatisk

Se mere avanceret eksempel i *Create a custom function, der behandler hver fil i filsæt.docx*

How to easily automate boring Excel tasks with Power Query.mp4 viser hvordan man kan importere flere ens filer fra mappe. Senere når ny årgang tilføjes, sker automatisk samme manipulationer for nye fil, data tilføjes grafen.

følger **How to easily automate boring Excel tasks with Power Query.mp4** , her bruges folder-connector



Kombiner og transformer data/filer, der tilføjes kolonne hvor man ser source-filen

Første fil åbnes som sample file, manipulationer heri gennemgår de øvrige filer også

Sales (2) - Power Query-editor

Fil | Hjem | Transformér | Tilføj kolonne | Vis

Luk og indlæs | Opdater eksempel | Avanceret editor | Administrer | Vælg kolonner | Fjern kolonner | Bevar rækker | Fjern rækker | Sortér | Opdel kolonne | Gruppér efter | Datatype: Tekst | Brug den første række som overskrifter | Erstat værdier | Flet forespørgsler | Tilføj forespørgsler | Kombiner filer | Administrer parametre

= Table.TransformColumnTypes(#"Udvidet tabelkolonne1",{{"Source.Name", type text}, {"Date", type date}},

	Source.Name	Date	Product A	Product B	Product C	Product D
1	201601.csv	01-01-2016	1403	1274	1151	621
2	201601.csv	02-01-2016	716	764	606	495
3	201601.csv	03-01-2016	1299	717	1123	346
4	201601.csv	04-01-2016	1241	58	6	269
5	201601.csv	05-01-2016	355	976	913	436
6	201601.csv	06-01-2016	696	16	783	965
7	201601.csv	07-01-2016	691	562	198	126
8	201601.csv	08-01-2016	626	530	1367	1094
9	201601.csv	09-01-2016	701	303	258	1249
10	201601.csv	10-01-2016	631	423	75	244
11	201601.csv	11-01-2016	230	218	263	1117
12	201601.csv	12-01-2016	743	977	1428	1249
13	201601.csv	13-01-2016	1356	671	554	21
14	201601.csv	14-01-2016	1203	1308	1052	278

Kombinerer række-værdier fra 3 kolonner, fx ved at gange værdierne sammen

Sample File - Power Query Editor

Form | Add Column | View

Conditional Column | Index Column | Duplicate Column | Merge Columns | Extract | Parse | Statistics | Standard | Scientific

= Table.RenameColumns(#"Changed Type1",{{"Ship Mode Contain", type text}}

Order Quantity	Unit Sell Price	Discount Per
9	95.99	
32	5.98	
3	40.98	
4	291.73	
43	100.98	0.07
32	135.06	0.05
36	9.21	0.09
43	15.56	0.06

trykkes pil op i

formel-linjen udvides det til formeeditor-vinduet

Lav ny kolonne, med custom column from examples: udfyldes første felt, så kan Excel se mønsteret.

Sales Person	Order Date	Text Between Delimiters
Mr Robert Carlsson	2/01/20	Robert
Miss Susanne Mann	2/01/20	
Mr Robert Carlsson	2/01/20	Robert

3.kolonne er afledt af 1.kolonne, 1. kolonne er valgt, så 'kolonne fra eksempel', der skrives fornavn Robert, så gætter E at jeg ønsker substring nr 2.

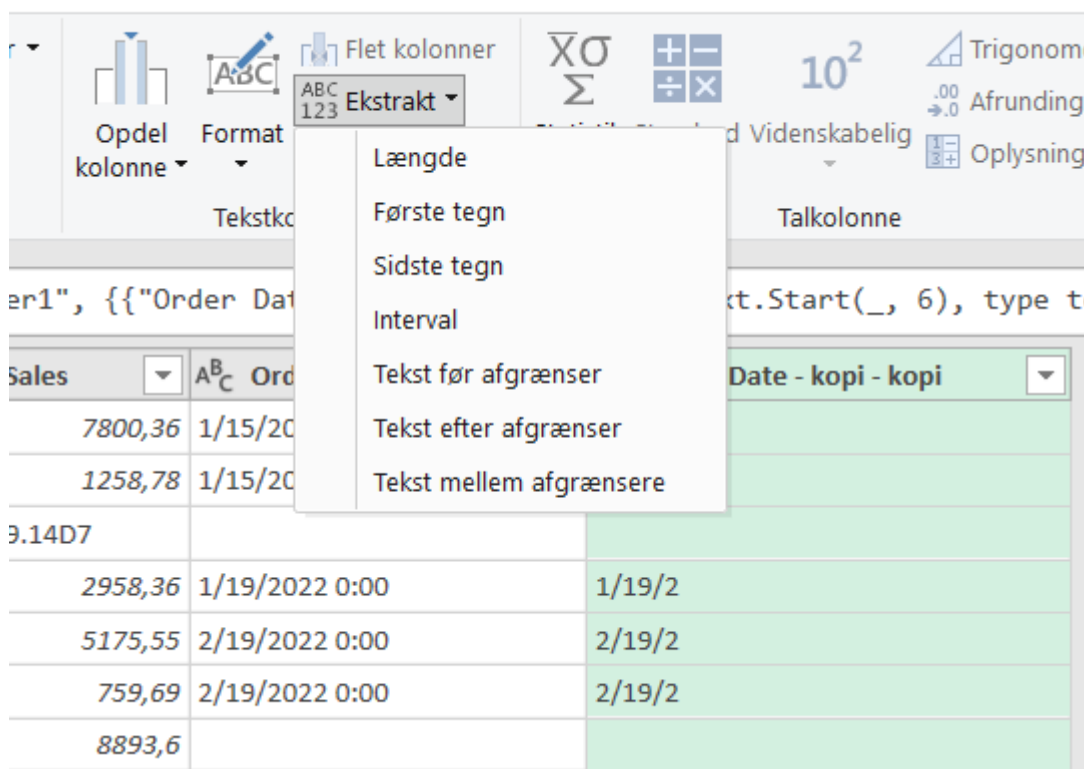
	person	frugt	quantity	repetition	Tekst mellem afgrænsere
1	hr Peter Hansen	apple	10	5	Peter
2	fru pia møller olsen	mango	2	4	pia

= Table.AddColumn("#Ændret type", "Tekst mellem afgrænsere", each Text.BetweenDelimiters([person], " ", " "), type text)

= Table.TransformColumns("#Omdøbte kolonner", {"efternavn", each Text.AfterDelimiter(_, " ", {0, RelativePosition.FromEnd}), type text})) ->efternavne

**** Strings-manipulation

=TRIM(text) , =FJERN.OVERFLØDIGE.BLANKE(celle_ref) i tekst-kolonne i Excel



= Table.TransformColumns("#Ekstraherede første tegn", {"Order Date - kopi - kopi", each Text.BeforeDelimiter(_, "/", type text)}} : månedsnr , mm

=Table.AddColumn("#Tilføjet brugerdefineret", "Tekst mellem afgrænsere", each Text.BetweenDelimiters(Text.From([#"Order Date - kopi - kopi.2"], "da-DK"), "/", " ", 1, 0), type text) ", "da-DK"), "/", " ", 1, 0), type text) finder årstallet, udtrækker fra 'Order Date - kopi - kopi.2'

= Table.AddColumn("#Fjernede kolonner", "dato.1", each [dd] & "-" & [mm] & "-" & [yyyy]) // konstruktion af dato dd-mm-yyyy

= Table.AddColumn("#Duplikerede kolonner3", "Tekst før afgrænser", each Text.BeforeDelimiter([Sales Rep], " "), type text) : ny kolonne, med første del af Sales rep-string, fornavnet

[Search for string](#)

Søger efter u i grønne kolonne, findes på plads 2 i Laura, findes ikke i række2, derfor værdi -1:

```
= Table.AddColumn(#"Removed Columns1", "FindText", each Text.PositionOf([Sales Rep], "w"))
```

	Item	Prod M	Sales Rep	Quantity	Price	Commission	Sales Rep - Copy	FindText
2020-01-01	Printer	KL001	Laura Smith	20	80	0.08	Laura Smith	2
2020-01-01	White Board	KL002	Stacey Johnson	28	40	0.06	Stacey Johnson	-2
2020-01-01	Office Chair	KL003	Mark Booth	7	230	0.08	Mark Booth	-2
2020-01-01	White Board	KL002	Stacey Johnson	23	40	0.06	Stacey Johnson	-2
2020-01-01	Printer	KL001	Laura Smith	8	80	0.08	Laura Smith	2

Sammensætning af string og tal: "Q" & Number.ToText([Kvartal])&"-" & Number.ToText([År])

-sådan kan kolonne med kvartal-år laves, fx 'Q1-2024'

Autodetect datatyper:

vælg alle kolonner med <shift> og henover alle kolonne-headers, så transform 'detect/identificer data type'

Advanced editor

let

```
Kilde = Excel.CurrentWorkbook(){[Name="Tabel2"]}[Content],
```

```
#"Ændret type" = Table.TransformColumnTypes(Kilde,{{"Date", type date}, {"Open", type number}, {"High", type number}, {"Low", type number}, {"Close ", type number}, {"Adj Close ", type number}, {"Volume", Int64.Type}}),
```

```
#"Dupliserede kolonner" = Table.DuplicateColumn("#"Ændret type", "Date", "Date - kopi"),
```

//forrige trin angives, data fra "Date", der laves ny kolonne "Date - kopi"

```
#"Omrokkede kolonner" = Table.ReorderColumns("#"Dupliserede kolonner",{"Date", "Date - kopi", "Open", "High", "Low", "Close ", "Adj Close ", "Volume"}),
```

//kolonne med uge nr finde i næste 4 trin:

```
#"Omdøbte kolonner" = Table.RenameColumns("#"Omrokkede kolonner",{"Date - kopi", "uge"}),
```

```
#"Ændret type1" = Table.TransformColumnTypes("#"Omdøbte kolonner",{"uge", type datetimezone}}),
```

```
#"Beregnet uge i året" = Table.TransformColumns("#"Ændret type1",{"uge", Date.WeekOfYear, Int64.Type}}),
```

```
#"Sorterede rækker" = Table.Sort("#"Beregnet uge i året",{"Date", Order.Ascending}}),
```

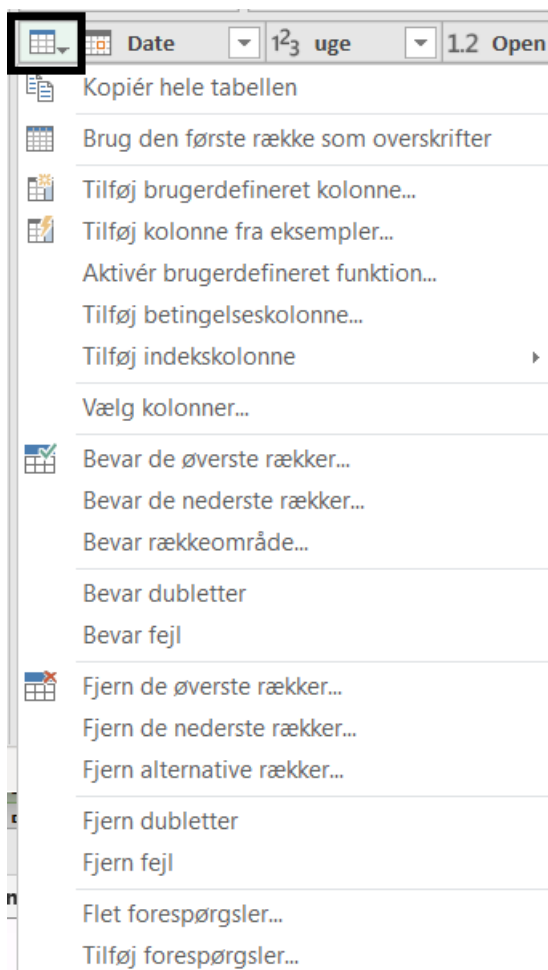
```
#"Fjernede fejl1" = Table.RemoveRowsWithErrors("#"Sorterede rækker", {"Date", "uge", "Open", "High", "Low", "Close ", "Adj Close ", "Volume"})
```

in

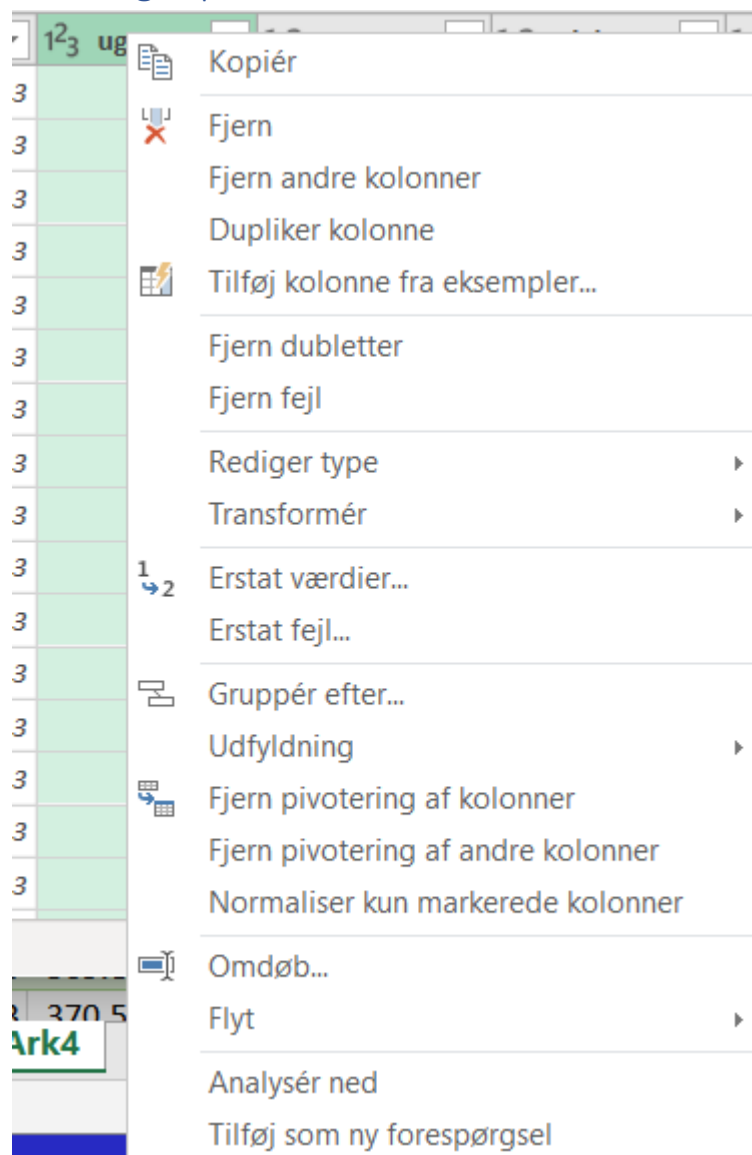
#"Fjernede fejl1"

I let delen defineres variable, og beregninger, in-delen er output til videre trin

Ændringer på hele tabelniveau, vælg PQ editor, tabelfelt



Ændringer på kolonne niveau: marker kolonne, højreklik



**** Date format transformations

Fik ikke nemt oversat fra eng format mm/dd/yyyy til dansk,

Problem fikset ved at få udskrevet mm,dd,yyyy(udtrukket med strings fkt) i 3 kolonner og så konstruere korrekt data

i filen: **power-query-introduction2.xlsx** arket, sheet **dato**

Man kan fjerne kolonner uden af ødelægge forudgående beregninger.

Til sidst kan man altså godt fjerne hjælpekolonner, før den endelig fejl 'udgives'

Date data type

When we get our data into Power Query, we first need to ensure dates have the correct data type. Sometimes Power Query's automatic detection of data types gets it wrong, so we need to check that all columns are correct.

To change a specific column into a date format, we have lots of options:

- Click the data type **icon** in the column header and select **Date**
- Select the column, then click **Transform > Data Type > Date** from the Ribbon
- Right-click on the column header, then click **Change Type > Date**
- Change the applied data type in the M code to **type date**

Changing the data type is the most important step. Until the columns are recognized as dates, Power Query cannot use them as dates.

One confusing aspect is that **Power Query data types exist in two ways**:

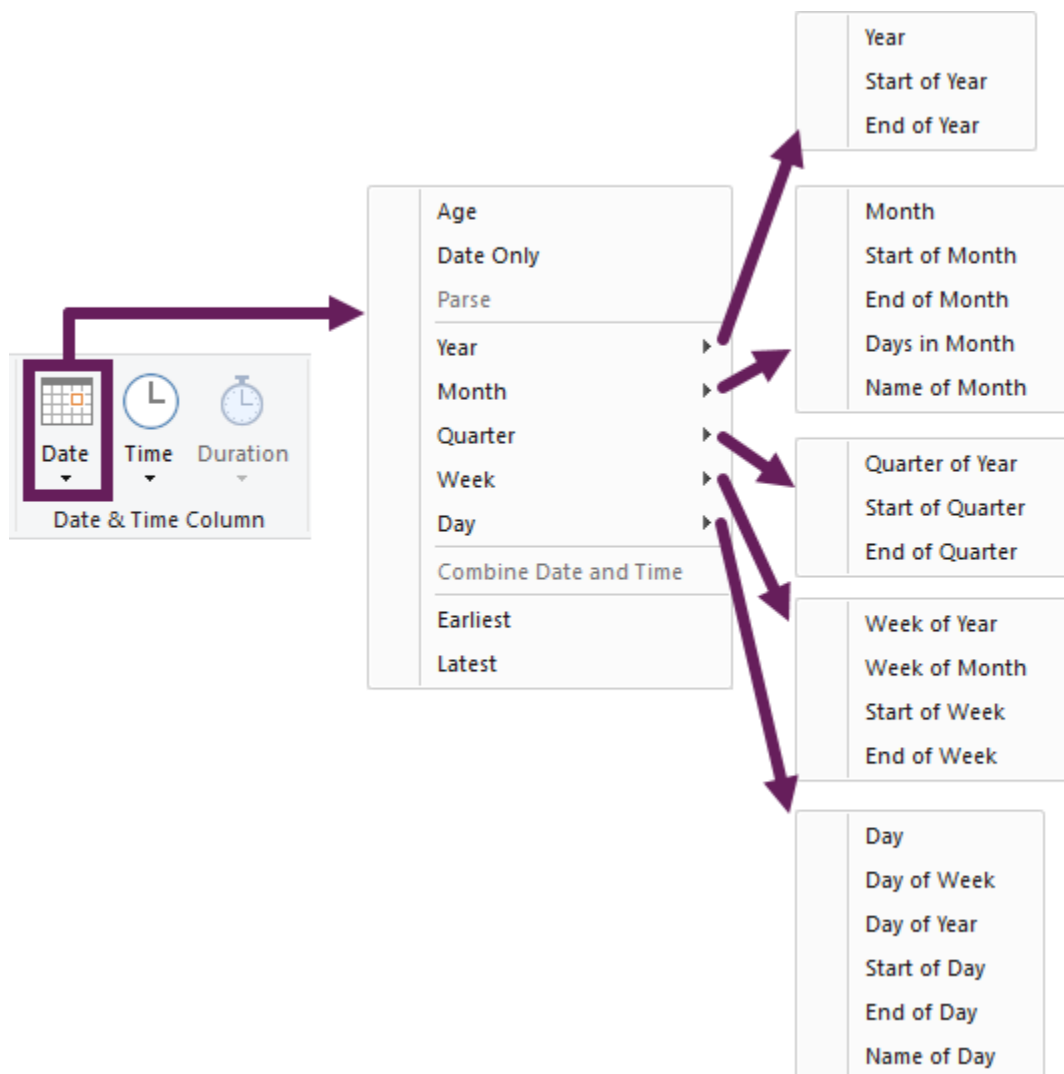
Value level: Each value in Power Query has a data type. If the column data type shows **abc123 (which is the any data type)**, each value in that column still has its own data type (text, number, date etc). The value type determines the type of transformations that can be performed on the value.

Column level: The column data type determines how the contents of a column is understood. The column data type declares that the values within that column should conform to that data type; therefore the actions available are based on that type.

If we write the M code ourselves, it is possible to have a date column that does not contain all date values.

Extracting additional information from a date

From a date column, we can extract other information. The Power Query date formats available from the ribbon are shown below.



Disse felter oprettes automatisk i pivottabel, til rollup/drill down

These are available in Power Query's Transform and Add Column ribbons. The Transform ribbon converts the selected column while Add Column creates a new column based on the selection.

The following are examples of the types of information we can extract from the date column (Year, Days in Month, Week of Year, Day Name, Day of Year)

	Date (YYYY-MM-DD)	¹ ₂ Year	¹ ₂ Days in Month	¹ ₂ Week of Year	^A ₈ Day Name	¹ ₂ Day of Year
1	23/07/2017	2017	31	30	Sunday	204
2	07/09/2017	2017	30	37	Thursday	250
3	02/03/2012	2012	31	10	Friday	62
4	10/03/2014	2014	31	11	Monday	69
5	26/08/2016	2016	31	35	Friday	239
6	05/10/2012	2012	31	41	Friday	279
7	13/03/2010	2010	31	11	Saturday	72
8	06/03/2009	2009	31	10	Friday	65
9	06/09/2007	2007	30	36	Thursday	249
10	10/05/2007	2007	31	19	Thursday	130

M code for date transformations

Each of the transformations above, apply a Power Query function to extract the information from the date. For example, the M code to add the Year and Day of Year columns are:

Year

```
= Table.AddColumn("#Changed Type", "Year", each Date.Year([#"Date (YYYY-MM-DD)"]), Int64.Type)
```

The Date.Year function extracts the Year from a date.

Day of Year

```
= Table.AddColumn("#Inserted Day Name", "Day of Year", each Date.DayOfYear([#"Date (YYYY-MM-DD)"]), Int64.Type)
```

The Date.DayOfYear function calculates the year day number from a date.

Additional information:

- Find out more about Power Query functions here: [Power Query formulas \(how to use them and pitfalls to avoid\)](#).
- Details of all the date functions are here: <https://learn.microsoft.com/en-us/powerquery-m/date-functions>

Tricky transformations

The biggest issue we typically face is getting values into a date type initially. Often this occurs when working with different country date formats.

The date format used in the US is **month – day – year**. Yet, it is **day – month – year** in the UK. Then there is the ISO 8601 format, which is **year – month – day**.

Personally, I prefer the **year – month – day** format as it can be sorted correctly when it is text or numeric. But I can't imagine the UK government is about to change anything on my account.

If using data containing dates in a format different from our local settings, Power Query can get confused. For example, if the cell value is 01/02/03, it could be:

- 2nd January 2003 – the US format
- 1st February 2003 – the UK format
- 3rd February 2001 – the ISO 8601 format

In this section, we look at how we force Power Query to understand dates, even if different country date formats are used.

Examples

Look at the screenshot below. It shows dates in five different formats (without any data types applied). I have included the format in the header so you can see what the date format is meant to be.

	ABC 123 dd.mm.yyyy	ABC 123 mm.dd.yyyy	ABC 123 yyyy.mm.dd	ABC 123 yy.mm.dd	ABC 123 yyyymmdd
1	23.07.2017	07.23.2017	2017.07.23	17.07.23	20170723
2	07.09.2017	09.07.2017	2017.09.07	17.09.07	20170907
3	02.03.2012	03.02.2012	2012.03.02	12.03.02	20120302
4	10.03.2014	03.10.2014	2014.03.10	14.03.10	20140310
5	26.08.2016	08.26.2016	2016.08.26	16.08.26	20160826
6	05.10.2012	10.05.2012	2012.10.05	12.10.05	20121005
7	13.03.2010	03.13.2010	2010.03.13	10.03.13	20100313
8	06.03.2009	03.06.2009	2009.03.06	09.03.06	20090306
9	06.09.2007	09.06.2007	2007.09.06	07.09.06	20070906
10	10.05.2007	05.10.2007	2007.05.10	07.05.10	20070510

NOTE: I am based in the UK. My regional settings are set to English (United Kingdom), so the dd.mm.yyyy is the expected format for dates. Depending on your location settings, you may get different results to those shown below.

If I use this data and select Date as the column data type, this is the result:

	dd.mm.yyyy	mm.dd.yyyy	yyyy.mm.dd	yy.mm.dd	yyyymmdd
1	23/07/2017	Error	23/07/2017	17/07/2023	Error
2	07/09/2017	09/07/2017	07/09/2017	17/09/2007	Error
3	02/03/2012	03/02/2012	02/03/2012	12/03/2002	Error
4	10/03/2014	03/10/2014	10/03/2014	14/03/2010	Error
5	26/08/2016	Error	26/08/2016	16/08/2026	Error
6	05/10/2012	10/05/2012	05/10/2012	12/10/2005	Error
7	13/03/2010	Error	13/03/2010	10/03/2013	Error
8	06/03/2009	03/06/2009	06/03/2009	09/03/2006	Error
9	06/09/2007	09/06/2007	06/09/2007	07/09/2006	Error
10	10/05/2007	05/10/2007	10/05/2007	07/05/2010	Error

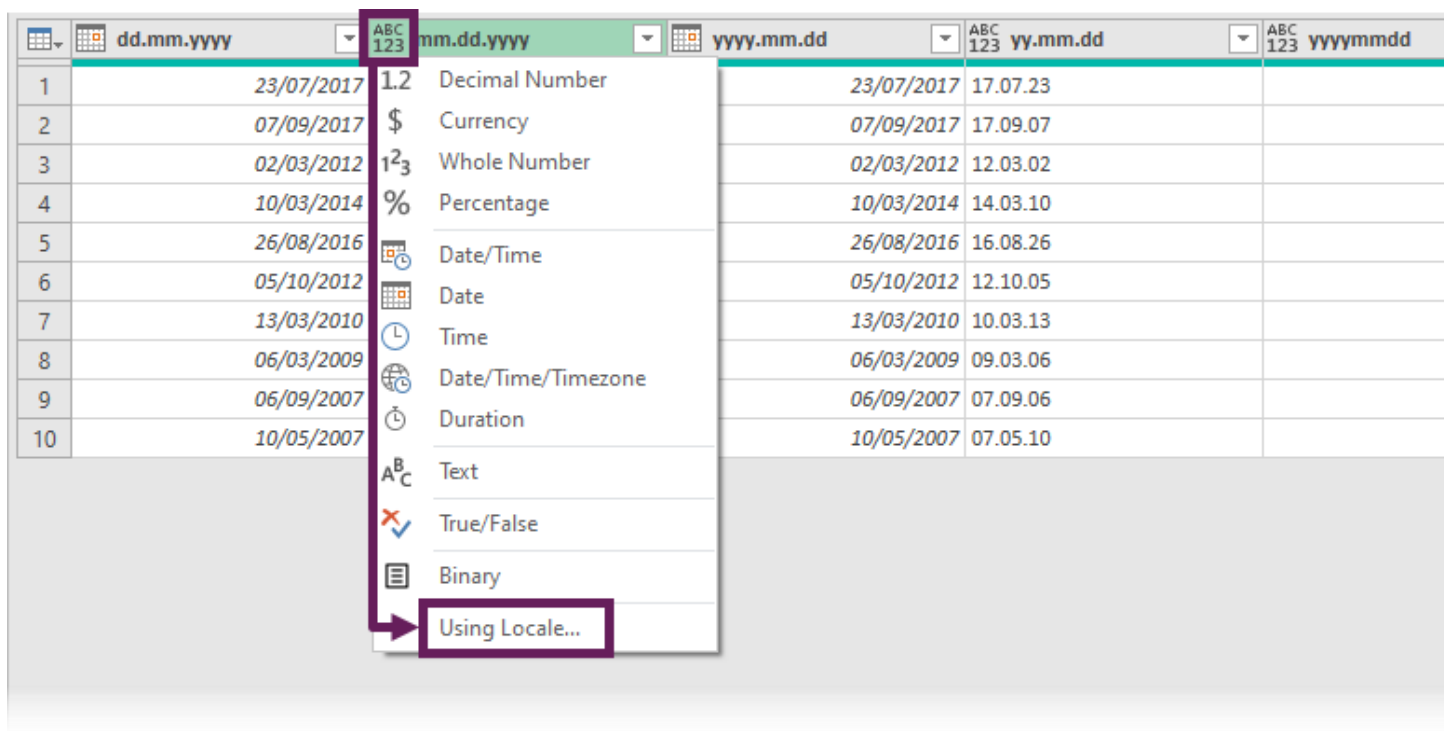
- **dd.mm.yyyy** – as expected, Power Query is happy to translate a date in my local region format.
- **mm.dd.yyyy** – the first row is an error because the value 07.23.2017 as a UK date would be the 7th day of the 23rd month. This is not a valid date in my region. In the second row, the value is 09/07/2017. As a UK format, this would be the 9th day of the 7th month, which is a valid date. However, even though the format is valid, the result is incorrect. The original date is in a mm.dd.yyyy format, so the correct value should be 7th day of the 9th month.
- **yyyy.mm.dd** – due to the year having a 4 character format, Power Query has correctly guessed that the remaining digits are month followed by day.
- **yy.mm.dd** – the data for this example includes the year with just two digits. Power Query can't work out which digits relate to the year, month or day. It has applied my local format. However, this has mixed up the day and year, meaning every value is incorrect. This is the most dangerous result, as if we don't notice and keep going, it could cause havoc later on.
- **yyyymmdd** – there are no separators in this example. Power Query has not recognized this as any date, so each value displays as an error.

Power Query gives us lots of tools to solve each of these date formats.

Locale settings

If the date is a valid country format, we can explicitly tell Power Query which country to use.

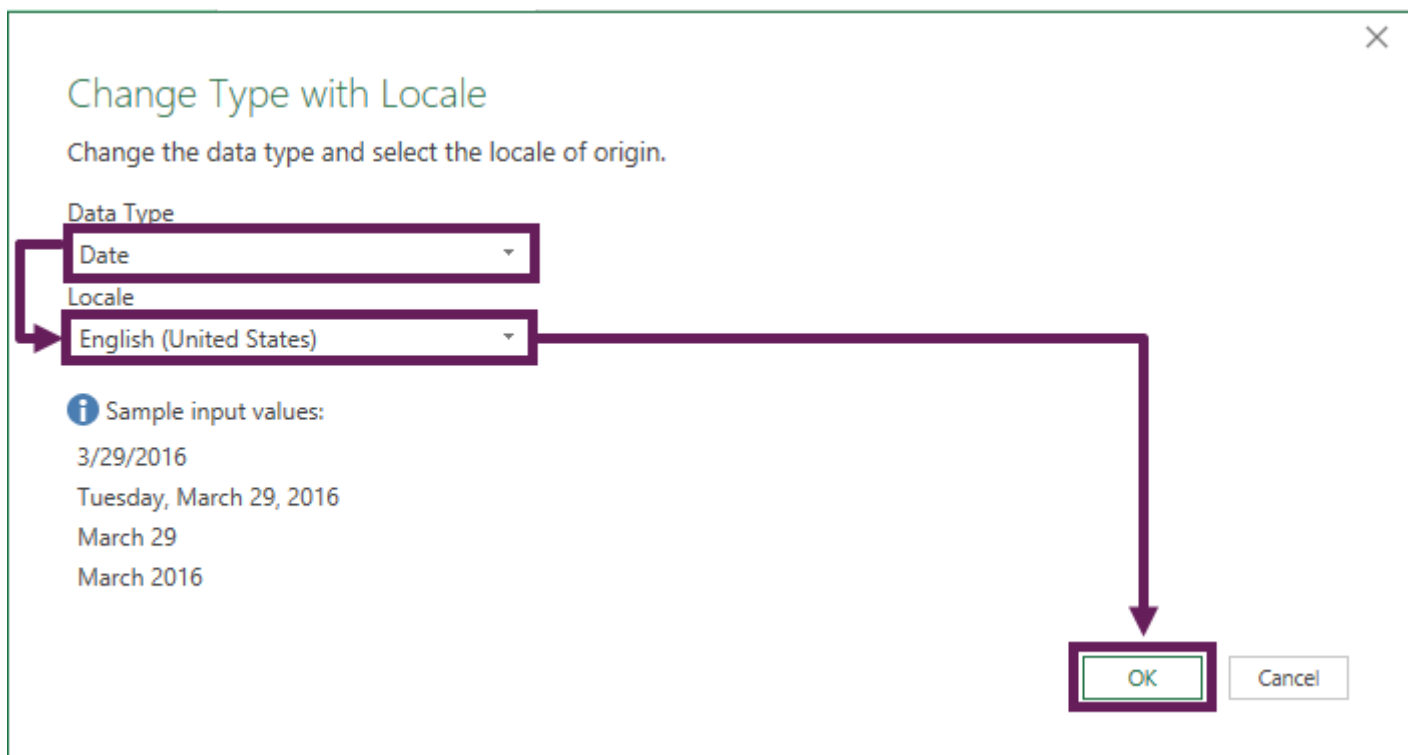
Click the **data type** icon and select **Using Locale...** from the menu.



From the **Change Type with Locale** dialog box, select the following options:

- **Data Type:** Date
- **Locale:** The region for the date format (for **mm.dd.yyyy**, I've selected **English (United States)**)

Then click **OK** to close the window.



Power Query now shows the correct date.

	dd.mm.yyyy	mm.dd.yyyy	yyyy.mm.dd	ABC 123 yy.mm.dd	ABC 123 YYYY
1	23/07/2017	23/07/2017	23/07/2017	17.07.23	
2	07/09/2017	07/09/2017	07/09/2017	17.09.07	
3	02/03/2012	02/03/2012	02/03/2012	12.03.02	
4	10/03/2014	10/03/2014	10/03/2014	14.03.10	
5	26/08/2016	26/08/2016	26/08/2016	16.08.26	
6	05/10/2012	05/10/2012	05/10/2012	12.10.05	
7	13/03/2010	13/03/2010	13/03/2010	10.03.13	
8	06/03/2009	06/03/2009	06/03/2009	09.03.06	
9	06/09/2007	06/09/2007	06/09/2007	07.09.06	
10	10/05/2007	10/05/2007	10/05/2007	07.05.10	

In the previous section, Power Query already got the correct result for the yyyy.mm.dd format, so we can move on to the next column.

To change the yy.mm.dd format, we can select English (Canada) as the locale date format. At the bottom of the window, it displays a sample of formats that are correctly recognized.

×

Change Type with Locale

Change the data type and select the locale of origin.

Data Type

Date

Locale

English (Canada)

Sample input values:

2016-03-29

March 29, 2016

29 March

March, 2016

OK

Cancel

The yyyy-mm-dd format is the first item in the list of sample inputs (shown in the purple box above). This order matches our example data. Selecting this option generates the correct values for this column. Power Query is intelligent enough to recognize the separators and create the correct date.

	dd.mm.yyyy	mm.dd.yyyy	yyyy.mm.dd	yy.mm.dd	ABC 23 YYYYY
1	23/07/2017	23/07/2017	23/07/2017	23/07/2017	
2	07/09/2017	07/09/2017	07/09/2017	07/09/2017	
3	02/03/2012	02/03/2012	02/03/2012	02/03/2012	
4	10/03/2014	10/03/2014	10/03/2014	10/03/2014	
5	26/08/2016	26/08/2016	26/08/2016	26/08/2016	
6	05/10/2012	05/10/2012	05/10/2012	05/10/2012	
7	13/03/2010	13/03/2010	13/03/2010	13/03/2010	
8	06/03/2009	06/03/2009	06/03/2009	06/03/2009	
9	06/09/2007	06/09/2007	06/09/2007	06/09/2007	
10	10/05/2007	10/05/2007	10/05/2007	10/05/2007	

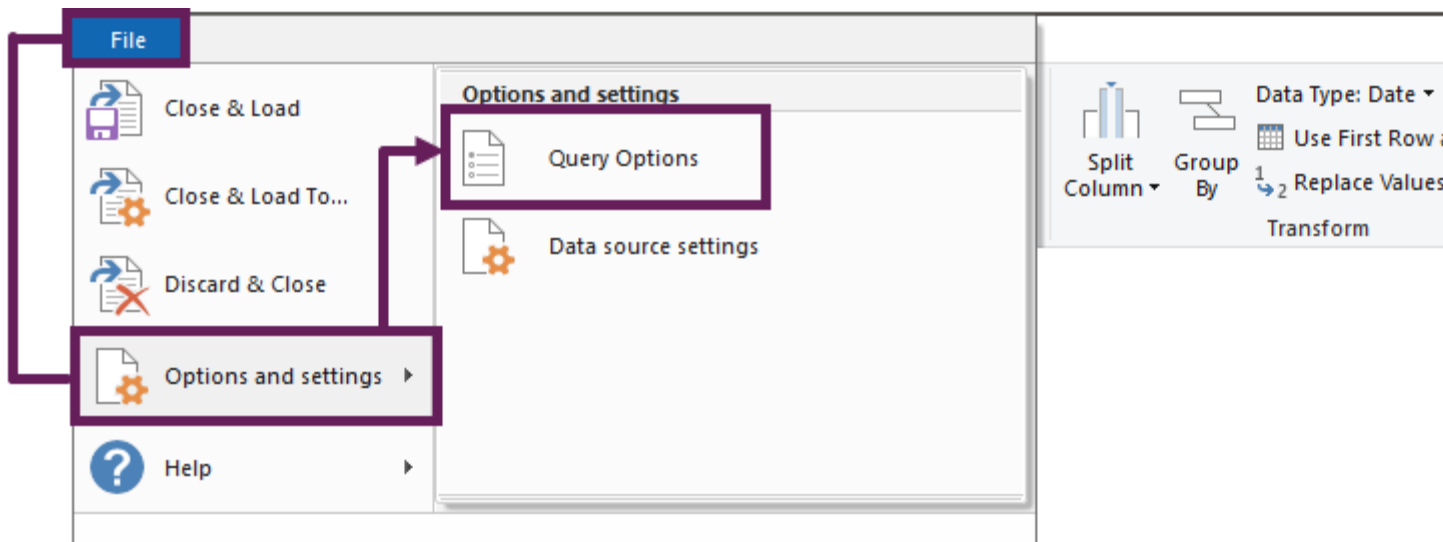
It may take a bit of exploration to find out which locale (if any) match the format we've got.

While the yyyymmdd format is in the same order as English (Canada), due to the lack of separators, Power Query doesn't recognize this as a date. We will come back to this example in a moment.

Changing the workbook Locale setting

If a workbook is going to have a lot of dates in a specific region format, we can change the Locale settings for the whole workbook.

Click **File > Options and settings > Query Options**



This setting applies to the current workbook only.

Once the locale is set, we only need to use the standard date data type and Power Query will apply that locale setting to each field.

Dealing with unknown date formats

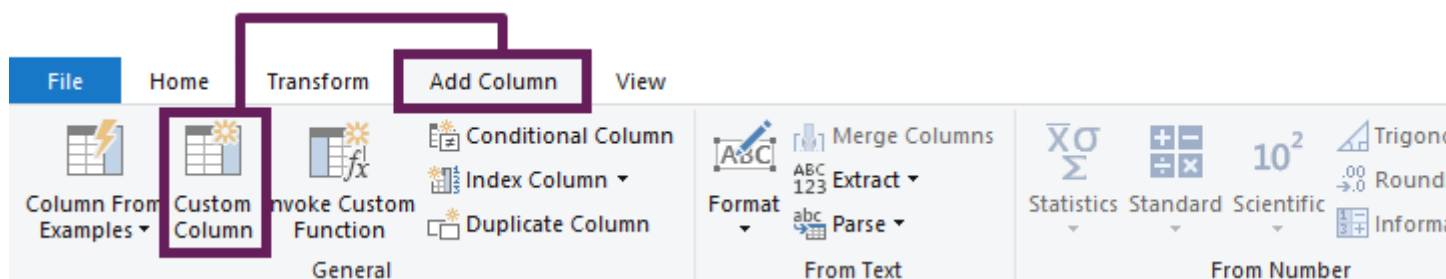
We were left with the tricky yyyyymmdd format. There are a few options we can try here. The two we are going to look at are:

- Power Query Formulas
- Column From Examples

Using Power Query formulas

Our date is in a standard format, so we can apply basic M code formulas to calculate the result.

Click **Add Column > Custom Column**



In the **Custom Column** dialog box, I have entered the following formula.

Custom Column

Add a column that is computed from the other columns.

New column name

Custom column formula ⓘ

Available columns

dd.mm.yyyy
mm.dd.yyyy
yyyy.mm.dd
yy.mm.dd
yyyyymmdd

[Learn about Power Query formulas](#)

✓ No syntax errors have been detected.

```
=Text.End(Text.From([yyyyymmdd]),2) & "/"
& Text.Middle(Text.From([yyyyymmdd]),4,2)
& "/" & Text.Start(Text.From([yyyyymmdd]),4)
```

This is constructed using 4 functions:

- Text.From() – converts the value in the field to text
- Text.Start() – similar to Excel's LEFT function to return the first characters from a text string
- Text.End() – similar to Excel's RIGHT function to return the last characters from a text string
- Text.Middle() – similar to Excel's MID function to extract characters from the middle of a text string

With this formula, we've constructed the date in a UK format using text.

If changing the data type for this new column, Power Query can now easily recognize this as a date.

This formula is specific for this scenario. You will need to change the formula to match your requirements.

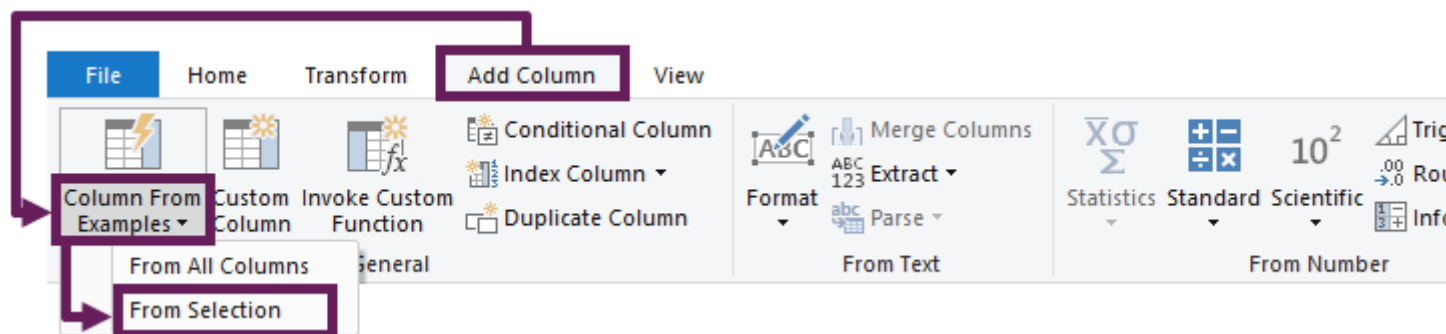
Other useful functions for creating date values are:

- **#date:** Creates a date based on year, month and day numbers.
Example: `=#date(2023,02,28)` is 28 February 2023.
- **Date.From:** Creates a date based on a value and a culture code.
Example: `=Date.From("03/15/2023","en-US")` is understood to be a US date for the 15 March 2023. The value can be text, or numbers.
- **Date.FromText:** Creates a date based on a value and options.
Example: `=Date.FromText("März 2023, 15", [Format="MMMM yyyy, dd", Culture="de-DE"])` is recognized as a date of 15 March 2023, presented is presented in the MMMM yyyy, dd format and using German spellings.

Using Column From Examples

Column From Examples is a feature that helps to standardize formats with complex structures.

Select the column, then click **Add Column > Column From Examples (drop-down) > From Selection**



In the column on the far right, we can **start typing the date** we want to see, then **press Enter**.

For example, for the value **20170723**, I have entered **23/07/2017** as shown by the screenshot below.

Add Column From Examples
 Enter sample values to create a new column (Ctrl+Enter to apply).
 Transform: `Date.From(Text.From([yyyymmdd], "en-GB"))`

	dd.mm.yyyy	mm.dd.yyyy	yyyy-mm-dd	yy-mm-dd	ABC 123 yyyymmdd	Date
1	23/07/2017	23/07/2017	23/07/2017	23/07/2017	20170723	23/07/2017
2	07/09/2017	07/09/2017	07/09/2017	07/09/2017	20170907	07/09/2017
3	02/03/2012	02/03/2012	02/03/2012	02/03/2012	20120302	02/03/2012
4	10/03/2014	10/03/2014	10/03/2014	10/03/2014	20140310	10/03/2014
5	26/08/2016	26/08/2016	26/08/2016	26/08/2016	20160826	26/08/2016
6	05/10/2012	05/10/2012	05/10/2012	05/10/2012	20121005	05/10/2012
7	13/03/2010	13/03/2010	13/03/2010	13/03/2010	20100313	13/03/2010
8	06/03/2009	06/03/2009	06/03/2009	06/03/2009	20090306	06/03/2009
9	06/09/2007	06/09/2007	06/09/2007	06/09/2007	20070906	06/09/2007
10	10/05/2007	10/05/2007	10/05/2007	10/05/2007	20070510	10/05/2007

Power Query does its magic and creates values for all our rows. Check that these are right. If not, correct any that are wrong; each time we provide more detail, Power Query adjusts its suggested transformation.

This is an advanced tool. If there isn't enough standardization in the value, Power Query can easily calculate the incorrect results. But for our scenario, it is reasonably straightforward.

Once happy with the format, click **OK**.

	dd.mm.yyyy	mm.dd.yyyy	yyyy-mm-dd	yy-mm-dd	ABC 123 yyyymmdd	Custom Column
1	23/07/2017	23/07/2017	23/07/2017	23/07/2017	20170723	23/07/2017
2	07/09/2017	07/09/2017	07/09/2017	07/09/2017	20170907	07/09/2017
3	02/03/2012	02/03/2012	02/03/2012	02/03/2012	20120302	02/03/2012
4	10/03/2014	10/03/2014	10/03/2014	10/03/2014	20140310	10/03/2014
5	26/08/2016	26/08/2016	26/08/2016	26/08/2016	20160826	26/08/2016
6	05/10/2012	05/10/2012	05/10/2012	05/10/2012	20121005	05/10/2012
7	13/03/2010	13/03/2010	13/03/2010	13/03/2010	20100313	13/03/2010
8	06/03/2009	06/03/2009	06/03/2009	06/03/2009	20090306	06/03/2009
9	06/09/2007	06/09/2007	06/09/2007	06/09/2007	20070906	06/09/2007
10	10/05/2007	10/05/2007	10/05/2007	10/05/2007	20070510	10/05/2007

Conclusion

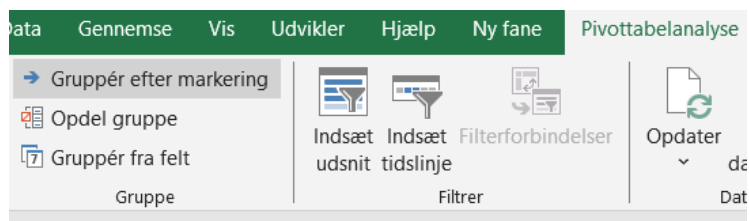
We have seen how to use the Power Query date format to extract additional information from a date. These are quick and easy transformations. However, before we are able to make those transformations, we need the column to be a date data type.

Using different date formats, we were able to get standardized dates from all of them. Therefore, no matter where the data comes from, we can convert it and use it in our local region.

Related Posts:

- [How to add fiscal Month, Quarter or Year Column in Power Query](#)
- [Calculate quarter from dates in Excel \(inc non-standard calendars\)](#)
- [EOMONTH Function in Excel](#)

Pivot-tabeller med /måneder/kvartal/år : opdelingen, sker bedste direkte i Excel



I	J	K	L	M
	Sum af Revenue	Kolonnemærkater		
	Summen af Revenue	Kolonnemærkater		
	Rækkemærkater	Brazil	Canada	Colombia
	2019			
	Kvt1	84501548	83804235	82931624
	Kvt2			
	apr			
	01-04-2019			1539709
	02-04-2019	2191591	1700838	755715
	03-04-2019	226869	565827	4527887

Marker et datofelt i pivottabellen, vælg så Pivottabelanalyse, og gruppér efter markering

Kolonne med formel, der inddrager værdier fra andre kolonner, i new custom column

New column name

Bonus

Custom column formula ⓘ

```
= if[Total Sales]>5000 then[Total Sales]*.10 else 0
```

Available columns

- Shoe Type
- Sales Rep First Name
- Sales Rep Last Name
- Order Date
- Total Sales
- Month Name
- Day Name

<< Insert

[The Complete Guide to Power Query | How To Excel](#)

Merge, join to tabeller , 'vandret'

Fra power-query-introduction2.xlsx

PQ Sales Data

Shoe Type	Sales Rep First Name	Sales Rep Last Name	Order Date	Total Sales	Bonus	Month Name
Oxford	Rachel	Green	2022-01-15	7,800.36	780.04	January
Sneakers	Rachel	Green	2022-01-15	1,258.78	0.00	January
Boat	Chandler	Bing	2022-01-19	2,958.36	0.00	January
Sandals	Chandler	Bing	2022-02-19	5,175.55	517.56	February

PQ Category

Shoe Type	Category
Oxford	Dress
Sneakers	Casual
Boat	Casual
Sandals	Casual
Espadrilles	Casual

Join Kind

Left Outer (all from first, matching from second)

☐ Use fuzzy matching to perform the merge

> Fuzzy matching options

✓ The selection matches 15 of 15 rows from the first table.

OK

Cancel

<https://www.youtube.com/watch?v=inNE4dZ1RVI> med datasæt:/Sales, learnPowerQuery filen

<https://www.dropbox.com/scl/fi/8744j8z865ysu0wklf9g6/PowerQuery-Practice-Data.xlsx?rlkey=07hcdhbqv8xfecacp55twxmta&e=2&dl=0>

https://www.boxofficemojo.com/chart/top_lifetime_gross_adjusted/?adjust_gross_to=2019

Append: rækkerne i tabel2 tilføjes rækkerne i 1.tabel, 'Iodret'

Creates a new query that contains all rows from a first query followed by all rows from a second query. You can create an Intermediate Append, which creates a new query for each append operation. Or you can create an Inline Append, which appends data to a query until you reach a final result.

For more information, see [Append queries](#).

See Also

[Power Query for Excel Help](#)

[Learn to combine multiple data sources \(Power Query\)](#)

About append queries

The append operation is based on the names of the column headers in both tables, and not their relative column position. The final table has all matching columns from all tables appended. If the tables don't have matching columns, null values are added to the unmatched column. The tables will be appended in the order in which they're selected, starting with the Primary table.

You can perform two types of append operations. With an inline append, you append data to your existing query until you reach a final result. The result is a new step at the end of the current query. With an intermediate append, you create a new query for each append operation.

Security Note: Privacy Levels prevent a user from inadvertently combining data from multiple data sources, which may be private or organizational. Depending on the query, a user could inadvertently send data from the private data source to another data source that might be malicious. Power Query analyzes each data source and classifies it into the defined level of privacy: Public, Organizational, and Private. For more information, see [Set privacy levels \(Power Query\)](#).

**** [Tabeller med gruppering, aggregeret værdier](#)

Bedst, mest fleksibelt med PQ, kan til dels også laves med pivot-tabeller, **her metoden med PQ:**

	A	B	C	D	E
1	Sales Data				
2	January 2019				
3					
4	Date	Customer	Product	Sold By	Value
5	09/01/2019	ABC Discounts	Farmhouse Bloomer	Sally	76.00
6	02/01/2019	Wilson's of Greenvale	Tiger	Sally	40.00
7	19/01/2019	Summer Bay Stores	Gluten Free	Sally	66.00
8	19/01/2019	Wilson's of Greenvale	Tiger	David	51.00
9	12/01/2019	Wilson's of Greenvale	Farmhouse Bloomer	David	75.00
10	22/01/2019	ABC Discounts	Farmhouse Bloomer	Mark	46.00
11	23/01/2019	Mega Mart	Tiger	Lucy	40.00
12	25/01/2019	Mega Mart	White Sliced	David	57.00
13	23/01/2019	D&T's	White Sliced	John	50.00

Salgsdata, skal grupperes efter Product og Customer. /Home/Group by/Advanced vælges

Fra 0116-Power-Query-Group-By.xlsx

For hver ny gruppering konstrueres en ny tabel -bedst at starte med kopi af salgsdata-tabellen, og så gruppere denne

Group By

Specify the columns to group by and one or more outputs.

☐ Basic
 ☒ Advanced

Product

Customer

Add grouping

New column name

Operation

Column

Total Sale

Sum

Value

Transactions

Count Rows

Add aggregation

OK

Cancel

med tre felter:

```
= Table.Group( Source, {"Product", "Customer"}, { {"Antal", each Table.RowCount(_), Int64.Type}, {"TotalValue", each List.Sum([Value]), type number}, {"GennemsnitsValue", each List.Average([Value]), type number} } ), dernæst sortering efter Produkt
```

	ABC 123	Product	ABC 123	Customer	1.3	Antal	1.2	TotalValue	1.2	GennemsnitsValue
1		Brown Sliced		D&T's		3		174		58
2		Brown Sliced		Tom's Superstore		1		55		55
3		Brown Sliced		Super Saver		1		67		67
4		Brown Sliced		Summer Bay Stores		1		54		54
5		Brown Sliced		Dollar Stretcher		5		328		65,6
6		Brown Sliced		Mega Mart		1		49		49
7		Brown Sliced		Uni-shopper		2		155		77,5
8		Farmhouse Bloo...		Uni-shopper		1		65		65
9		Farmhouse Bloo...		AB Discounts		2		122		61
10		Farmhouse Bloo...		Summer Bay Stores		2		124		62
11		Farmhouse Bloo...		Mega Mart		3		137		45,66666667
12		Farmhouse Bloo...		Wilson's of Green...		2		114		57
13		Gluten Free		Summer Bay Stores		1	salg: 4	66		66
14		Gluten Free		Tom's Superstore		2		113		56,5
15		Gluten Free		Super Saver		1		70		70
16		Tiger		Super Saver		1		76		76

Forespørgselsind

EGENSKABER

Navn
SalesData

Alle egenskaber

ANVENDTE TRIN

Source
Grouped Rows
Sorted Rows

Proces-Trin ses t.h.: source=indlæs af original salgsdata, så gruppering, til sidst sortering

Info om product-salg, for alle customers:

Product

Tilføj gruppering

Nyt kolonnenavn

Antal

TotalValue

Gennemsnit

Handling

Tæl rækker

Sum

Gennemsnit

Kolonne

Value

Value

	ABC 123	Product	1.3	Antal	1.2	TotalValue	1.2	Gennemsnit
1		Farmhouse Bloo...		10		562		56,2
2		Tiger		15		853		56,86666667
3		Gluten Free		4	salg: 4	249		62,25
4		White Sliced		7		382		54,57142857
5		Brown Sliced		14		882		63

**** Filtrering af felt kan starte med UI, kan evt. finjusteres med Power Query- M; filtrering af rækker:

let

```
Source = Exchange.Contents("xyz@contoso.com"),  
Mail1 = Source{[Name="Mail"]}[Data],  
#"Expanded Sender" = Table.ExpandRecordColumn(Mail1, "Sender", {"Name"}, {"Name"}),  
#"Filtered Rows" = Table.SelectRows("#Expanded Sender", each ([HasAttachments] = true)),  
//kun emails med Attachments passerer filteret, andre fjernes
```

..

Visuel hjælp til hver M-kommando

Filtrer rækker

Anvend en eller flere filterbetingelser på rækkerne i denne tabel.

☒ Grundlæggende ☐ Avanceret

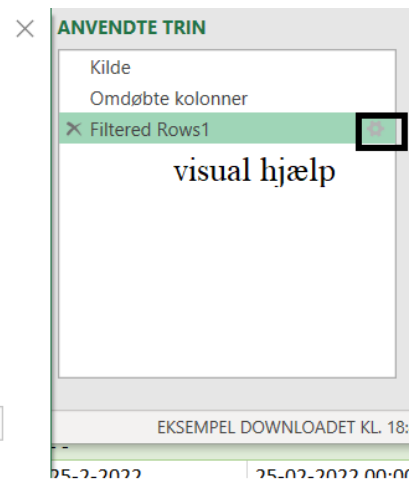
Behold rækker, hvor 'Shoe Type'

er lig med

☒ Og ☐ Eller

OK

Annuller



Pivot Tabel-form : hver række indeholder flere rekords

Before: Data Exported to Excel from Reporting System

	A	B	C	D	E	F	G	H
1	Sales Report 2014							
2	Sales Rep: Jeff Smith							
3								
4	Product Name	Jan-14	Feb-14	Mar-14	Apr-14	May-14	Jun-14	Jul-14
5	Product A	192.38	285.75	38.25	200.25	221.63	349.88	165.38
6	Product B	280.13	435.38	382.50	191.25	410.63	119.25	424.13
7	Product C	498.38	151.88	424.13	376.88	301.50	38.25	268.88
8	Product D	214.88	356.63	562.50	156.38	250.88	199.13	74.25
9	Product E	383.63	500.63	312.75	397.13	360.00	142.88	500.63
10	Product F	158.63	87.75	60.75	526.50	50.63	113.63	213.75
11	Product G	506.25	104.63	138.38	90.00	173.25	362.25	357.75
12	Product H	45.00	505.13	396.00	276.75	153.00	346.50	151.88
13	Product I	222.75	60.75	492.75	327.38	313.88	508.50	502.88
14	Product J	456.75	400.50	72.00	39.38	131.63	513.00	279.00
15	Product K	409.50	81.00	169.88	559.13	315.00	390.38	150.75
16	Product L	92.25	110.25	129.38	117.00	82.13	99.00	136.13
17	Total	3,460.50	3,080.25	3,179.25	3,258.00	2,764.13	3,182.63	3,225.38
18								
19	Export Date: 2/1/2015							
20								

her ses

mange rekords I hver række, en pivot-tabel-format: laves en pivot-tabel vil datoer kunne RollUP[til kvartaler,...]/drilledDown igen(finere dato-opdeling)

And you want the end result to look like this:

Unpivot data, data på normaliseret form: har kun een rekord per række

After: Use Power Query to Unpivot Data

	A	B	C
1	Product Name	Month	Sales Amount
2	Product A	Jan-14	192.375
3	Product A	Feb-14	285.75
4	Product A	Mar-14	38.25
5	Product A	Apr-14	200.25
6	Product A	May-14	221.625
7	Product A	Jun-14	349.875
8	Product A	Jul-14	165.375
9	Product A	Aug-14	499.5
10	Product A	Sep-14	405

Can be used for the source of a Pivot Table.

Here is an article and video on exactly [How to Unpivot Your Data with Power Query](#). Marker kolonner med multi-rekords, [shift]+[Apples] -[Tea]

atype: Heltal ▾
 Erstat værdier ▾
 Fjern pivotering af kolonner ▾
 Identificer datatype ▾
 Udfyldning ▾
 Flyt ▾
 Omdøb ▾
 Pivoter kolonne ▾
 Konverter til liste ▾
 En hvilken som helst kolonne
 Opdel kolonne ▾
 Format ▾
 Flet kolonner ▾
 Ekstrakt ▾
 Fortolkning ▾
 Statistik ▾
 Standard ▾
 Videnskabelig ▾
 10² ▾
 Trigonometri ▾
 Afrunding ▾
 Oplysninger ▾
 Dato- og klokkeslætskolonne ▾
 Dato ▾
 Klokkeslæt ▾
 Variabel ▾

fx = Table.TransformColumnTypes(Source,{{"Order Date", type datetime}, {"Received Date", type datetime}, {"PO#", type text}, {"Vendor", type text}, {"Attribut", type text}, {"Værdi", type text}})

	date	Received Date	PO#	Vendor	Apples	Pears	Oranges	Coffee	Tea
1	14 00:00...	27-01-2014 00:00:...	100203	Vend A	10	94	49	null	null
2	14 00:00...	25-01-2014 00:00:...	100204	Vend A	49	100	60	null	null
3	14 00:00...	26-01-2014 00:00:...	100205	Vend B	null	null	null	37	null
4	14 00:00...	01-02-2014 00:00:...	100206	Vend C	null	null	null	null	58
5	14 00:00...	28-01-2014 00:00:...	100207	Vend B	null	null	null	69	null
6	14 00:00...	30-01-2014 00:00:...	100208	Vend B	null	null	null	77	null
7	14 00:00...	29-01-2014 00:00:...	100209	Vend C	null	null	null	null	84

= Table.UnpivotOtherColumns("#Changed Type", {"Order Date", "Received Date", "PO#", "Vendor"}, "Attribut", "Værdi") ->

	Order Date	Received Date	PO#	Vendor	Attribut	Værdi
1	15-01-2014 00:00:...	27-01-2014 00:00:...	100203	Vend A	Apples	
2	15-01-2014 00:00:...	27-01-2014 00:00:...	100203	Vend A	Pears	
3	15-01-2014 00:00:...	27-01-2014 00:00:...	100203	Vend A	Oranges	
4	17-01-2014 00:00:...	25-01-2014 00:00:...	100204	Vend A	Apples	
5	17-01-2014 00:00:...	25-01-2014 00:00:...	100204	Vend A	Pears	
6	17-01-2014 00:00:...	25-01-2014 00:00:...	100204	Vend A	Oranges	
7	18-01-2014 00:00:...	26-01-2014 00:00:...	100205	Vend B	Coffee	
8	19-01-2014 00:00:...	01-02-2014 00:00:...	100206	Vend C	Tea	

Vigtige transformationstyper: fremgår af Query Settings-listen t.h.

Man kan nemt afprøve og fortryde transformation ved at slette proces-trinnet fra listen

-Remove columns, rows, blanks

-Convert data types – text, numbers, dates

-Split or merge columns: nemt fx at lave nye kolonner med fornavn, efternavn

Marker kolonnen med **personnavn**, højreklik 'opdel kolonne efter afgrænser', i power-query-introduction2/NameSplitted.xlsx

```

#"Split Column by Delimiter" = Table.SplitColumn("#previous ", "Sales Rep", Splitter.SplitTextByDelimiter(" ",
QuoteStyle.Csv), {"Sales Rep.1", "Sales Rep.2"}),
#"Changed Type1" = Table.TransformColumnTypes("#Split Column by Delimiter",{{"Sales Rep.1", type text}, {"Sales
Rep.2", type text}})

```

Sort & filter columns

Add calculated columns

Aggregate or summarize data kan laves med PowerQuery og lidt besværligt med pivot-tabel i Excel

'Advanced Group By' for grupper med flere felter, med flere aggregeringer

Click the **Advanced** button in the dialog box to show the advanced options.

Advanced allows us to summarize and calculate based on as many columns as we need.

For this example, let's say we want the total and count of sales transactions by product and customer.

Set the first section to include the **Product** and **Customer** columns. Click the **Add Grouping** button to add columns, or the three dots (...) to remove columns.

In the **new column name section**, apply the following settings:

-**New column name:** Total Sales

-**Operation:** Sum

-**Column:** Value

Click the **add aggregation** button, then set the new column as follows

-**New column name:** Transactions, Antal,

-**Operation:** Count Rows

-**Column:** [not required for Count Rows], resultatet ses t.v. ; t.h. ses 2 andre grupperinger

	A	B	C	D	E		A	B	C
1	Product	Customer	Antal	TotalValue	GennemsnitsValue	1	Product	Antal	TotalValue
2	Brown Sliced	Dollar Stretcher	5	328	65.6	2	White Sliced	7	382
3	Brown Sliced	kombinationen forekommer 5x	2	14	77.5	3	Tiger	15	853
4	Brown Sliced	Tom's Superstore	1	55	55	4	Gluten Free	4	249
5	Brown Sliced	D&T's	3	174	58	5	Farmhouse Bloomer	10	562
6	Brown Sliced	Mega Mart	1	49	49	6	Brown Sliced	14	882
7	Brown Sliced	Summer Bay Stores	1	54	54				
8	Brown Sliced	Super Saver	1	67	67				
9	Farmhouse Bloomer	ABC Discounts	2	122	61				
10	Farmhouse Bloomer	Wilson's of Greenvale	2	114	57				
11	Farmhouse Bloomer	Mega Mart	3	137	45.66666667				
12	Farmhouse Bloomer	Summer Bay Stores	2	124	62				
13	Farmhouse Bloomer	Uni-shopper	1	65	65				
14	Gluten Free	Summer Bay Stores	1	66	66				
15	Gluten Free	Tom's Superstore	2	113	56.5				
16	Gluten Free	Super Saver	1	70	70				
17	Tiger	Wilson's of Greenvale	4	181	45.25				
18	Tiger	Mega Mart	2	78	39				
19	Tiger	Tom's Superstore	3	167	55.66666667				
20	Tiger	Summer Bay Stores	1	64	64				

	A	B
1	Customer	Sales By
2	ABC Discounts	Sally,Mark
3	Wilson's of Greenvale	Sally,David,Jack,John
4	Summer Bay Stores	Sally,Mark,Jack
5	Mega Mart	Lucy,David,John,Sally
6	D&T's	John,David,Sally,Jack
7	Dollar Stretcher	Sally,John,Lucy
8	Tom's Superstore	Lucy,John,Sally,Jack
9	Uni-shopper	David,Lucy
10	Super Saver	David,Jack,Sally

Find & replace text/value

Man kan søg/erstat i rå-excel-data, og vælge 'hele projektmappen', så slår det igennem i alle afledte tabeller, ellers blot selv arket; det er faktisk nemmere end i PQ

I PowerQuery: Right-click on a column-> Select **Replace Values**

```
Table.ReplaceValue( #"Changed Type", "Together", "Living Together", Replacer.ReplaceText {"Marital_Status"}) : M-code
```

The 5 arguments within the function are:

Argument	Description
1. Table as table	This is the table where values are replaced
2. Old Value as any	The value that you want to replace by a new value
3. New Value as any	The value you want to replace the old value by
4. Replacer as function	One of replacer functions: – Replacer.ReplaceValue – Replacer.ReplaceText
5. Columns to search as list	Column to replace values in

*** Nested Calculations in PQ: lave grupper i een kolonne, så ranking indenfor hver gruppe, så expand grupperne i kolonnen

[Chris Webb](#) [Power BI](#), [Power Query](#) May 11, 2015 2 Minutes

[Chris Webb's BI Blog: Nested Calculations In Power Query](#)

Quite often, in Power Query, you want to 'nest' calculations and transformations – apply them not across the whole table, but repeat the same calculation or transformation across multiple groups of rows within that table. Let me give you an example...

NestedCalculationsPQ.xlsx filen

Take the following input table: **først oprettes kolonne med grupper-table-værdier, indenfor hver måned, så findes rank efter Sale indenfor hver de tre grupper, til sidst ekspanderes tabellerne til den overordnede, store tabel.**

Month	Product	Sales
January	Apples	11
January	Oranges	22
January	Grapes	33
February	Apples	55
February	Oranges	44
February	Grapes	77
March	Apples	88
March	Oranges	99
March	Grapes	66

Imagine you wanted to add a column showing the rank of each row by Sales. In Power Query you would just need to:

1. Load the data
2. Sort the table in descending order by Sales
3. Add an index column starting at 1, which is the rank

You would end up with the following output:

	Month	Product	Sales	Rank
1	March	Oranges	99	1
2	March	Apples	88	2
3	February	Grapes	77	3
4	March	Grapes	66	4
5	February	Apples	55	5
6	February	Oranges	44	6
7	January	Grapes	33	7
8	January	Oranges	22	8
9	January	Apples	11	9

...and here's the M code, all of which is generated by the UI:

let

```
Source = Excel.CurrentWorkbook(){[Name="Sales"]}[Content],
#"Sorted Rows" = Table.Sort(Source,{{"Sales", Order.Descending}}),
#"Added Index" = Table.AddIndexColumn("#Sorted Rows", "Rank", 1, 1)
```

in
#"Added Index"

Now imagine you wanted to rank Sales within each month. That's to say you want to apply the same steps as shown in the previous query but for each month individually to get an output like this:

	Month	Product	Sales	Rank
1	January	Grapes	33	1
2	January	Oranges	22	2
3	January	Apples	11	3
4	February	Grapes	77	1
5	February	Apples	55	2
6	February	Oranges	44	3
7	March	Oranges	99	1
8	March	Apples	88	2
9	March	Grapes	66	3

One way to tackle this, and problems like it, is to do the following. First, do a Group By on the column you want to repeat the calculation over, in this case Month, and use the All Rows aggregation operation. This will result in a table with one row for each month and a column containing nested tables, as shown below:

fx = Table.Group(Source, {"Month"}, {"AllRows", each _ type table}))

Month	AllRows
1 January	Table
2 February	Table
3 March	Table

Group By...

Specify the columns to group by.

Group by
Month

New column name
AllRows

Operation
All Rows
Sum
Average
Median
Min
Max
Count Rows
Count Distinct Rows

Column

OK Cancel

Each one of these tables contains the rows from the original table for the month.

You can then take your original transformation and turn it into a function, either in a separate query or as a step in your current query. Here's an example of how the query above can be turned into a function that takes a table and returns a table with a rank column added:

Function I M:

(tabletorank as table) as table =>

let

SortRows = Table.Sort(tabletorank,{{"Sales", Order.Descending}}),

AddIndex = Table.AddIndexColumn(SortRows, "Rank", 1, 1)

in

AddIndex // =return value ; tabletorank as table er input-værdi

Next, you need to pass each nested table to this function. You could do that in a calculated column, but the most elegant way I think is by using the [Table.TransformColumns\(\)](#) function which takes a function and applies it to every value in a column (see [here](#) for another example of how to use it).

Finally, you get the final output by clicking on the Expand icon in the AllRows column and then choosing to expand all the columns in the nested table except the ones you originally grouped on:

	Month	AllRows
1	January	Table
2	February	Table
3	March	Table

les Rank

Month AllRows

Search Columns to Expand

☒ Expand ☐ Aggregate

☒ (Select All Columns)

☐ Month

☒ Product

☒ Sales

☒ Rank

☐ Use original column name as prefix

OK Cancel

Man kan få expanded
tabellens kolonner til arket med [ok]

ABC
123

month

AllRows

1	jan	Table
2	feb	Table
3	mar	Table

month	product	sales
jan	æble	10
jan	pære	15
jan	banan	5

Forespørgselsindstillinger x

EGENSKABER

Navn

Tabel1

Alle egenskaber

ANVENDTE TRIN

Source

①

X Grouped

① ⚙️

RankFunction

①

AddedRank

①

ExpandAgain

① ⚙️

EGENSKABER

Navn
Tabel1

Alle egenskaber

ANVENDTE TRIN

Source	(i)
X Grouped	(i) ⚙️
RankFunction	(i)
AddedRank	(i)
ExpandAgain	(i) ⚙️

Here's the full M code:

```
let
//Get data from Excel
Source = Excel.CurrentWorkbook(){[Name="Sales"]}[Content],
//Group by Month, der laves ny kolonne AllRows med tabeltype(af grupper), behøver altså ikke
skrive #"Gruoped", der defineres variabel ikke et procestrin:
Grouped = Table.Group(Source, {"Month"}, {"AllRows", each _, type table}),
//Declare a function that adds a Rank column to a table, input table, der så sorters efter Sales
RankFunction = (tabletorank as table) as table =>
let
SortRows = Table.Sort(tabletorank,{"Sales", Order.Descending}),
AddIndex = Table.AddIndexColumn(SortRows, "Rank", 1, 1)
in
AddIndex,

//Apply that function to the AllRows column, laver her kolonnen AddedRank:
AddedRank = Table.TransformColumns(Grouped, {"AllRows", each RankFunction(_)}),
//Expand the tables in the AllRows column again:
ExpandAgain = Table.ExpandTableColumn(AddedRank, "AllRows",
{"Product", "Sales", "Rank"}, {"Product", "Sales", "Rank"})
```

//ovenfor: tabellen inddrager kolonner {"Product", "Sales", "Rank"}, de **nye navne** er de samme, kolonnen med tabellerne i AllRows, ekspanderes
in
ExpandAgain

You can download the example workbook [here](#).

**** Bedste praksis-tips

<https://learn.microsoft.com/da-dk/power-query/best-practices>

Vælg den rigtige connector til data-source

Power Query indeholder et stort antal dataconnectors. Disse connectorer spænder fra datakilder som TXT-, CSV- og Excel-filer til databaser som Microsoft SQL Server og populære SaaS-tjenester, f.eks. Microsoft Dynamics 365 og Salesforce. **Hvis du ikke kan se din datakilde i vinduet Hent data , kan du altid bruge ODBC- eller OLEDB-connectoren til at oprette forbindelse til datakilden.**

Hvis du bruger den bedste connector til opgaven, får du den bedste oplevelse og ydeevne. Hvis du f.eks. bruger SQL Server-connectoren i stedet for ODBC-connectoren, når du opretter forbindelse til en SQL Server-database, får du ikke kun en meget bedre **Hent data-oplevelse** , men SQL Server-connectoren indeholder også funktioner, der kan forbedre din oplevelse og ydeevne, f.eks. forespørgselsdelegering. Hvis du vil læse mere om forespørgselsdelegering, skal du gå til [Oversigt over evaluering af forespørgsler og forespørgselsdelegering i Power Query](#).

Hver dataconnector følger en standardoplevelse, som forklaret i [Hent data](#). Denne standardiserede oplevelse har en fase med navnet **Dataeksempel**. I denne fase får du et brugervenligt vindue til at vælge de data, du vil hente fra din datakilde, hvis connectoren tillader det, og en simpel dataeksempel på disse data. Du kan endda vælge flere datasæt fra din datakilde via vinduet **Navigatør** , som vist på følgende billede.

Navigator

Display Options

AdventureWorks2012 [93]

☐ HumanResources.vEmployee
☐ HumanResources.vEmployeeDepartment
☐ HumanResources.vEmployeeDepartmentHistory
☐ HumanResources.vJobCandidate
☐ HumanResources.vJobCandidateEducation
☐ HumanResources.vJobCandidateEmployment
☐ Person.vAdditionalContactInfo
☐ Person.vStateProvinceCountryRegion
☐ Production.vProductAndDescription
☐ Production.vProductModelCatalogDescription
☐ Production.vProductModelInstructions
☐ Purchasing.vVendorWithAddresses
☐ Purchasing.vVendorWithContacts
☐ Sales.vIndividualCustomer
☐ Sales.vPersonDemographics
☐ Sales.vSalesPerson
☐ Sales.vSalesPersonSalesByFiscalYears
☐ Sales.vStoreWithAddresses
☐ Sales.vStoreWithContacts

Select Related Tables

HumanResources.vEmployee

Preview downloaded on Wednesday, April 29, 2020

BusinessEntityID	Title	FirstName	MiddleName
1	null	Ken	J
2	null	Terri	Lee
3	null	Roberto	
4	null	Rob	
5	Ms.	Gail	A
6	Mr.	Jossef	H
7	null	Dylan	A
8	null	Diane	L
9	null	Gigi	N
10	null	Michael	
11	null	Ovidiu	V
12	null	Thierry	B
13	Ms.	Janice	M
14	null	Michael	I
15	null	Sharon	B
16	null	David	M
17	null	Kevin	F
18	null	John	L
19	null	Mary	A
20	null	Wanida	M
21	null	Terry	J
22	null	Sariya	E

Load

Transform Data

Cancel

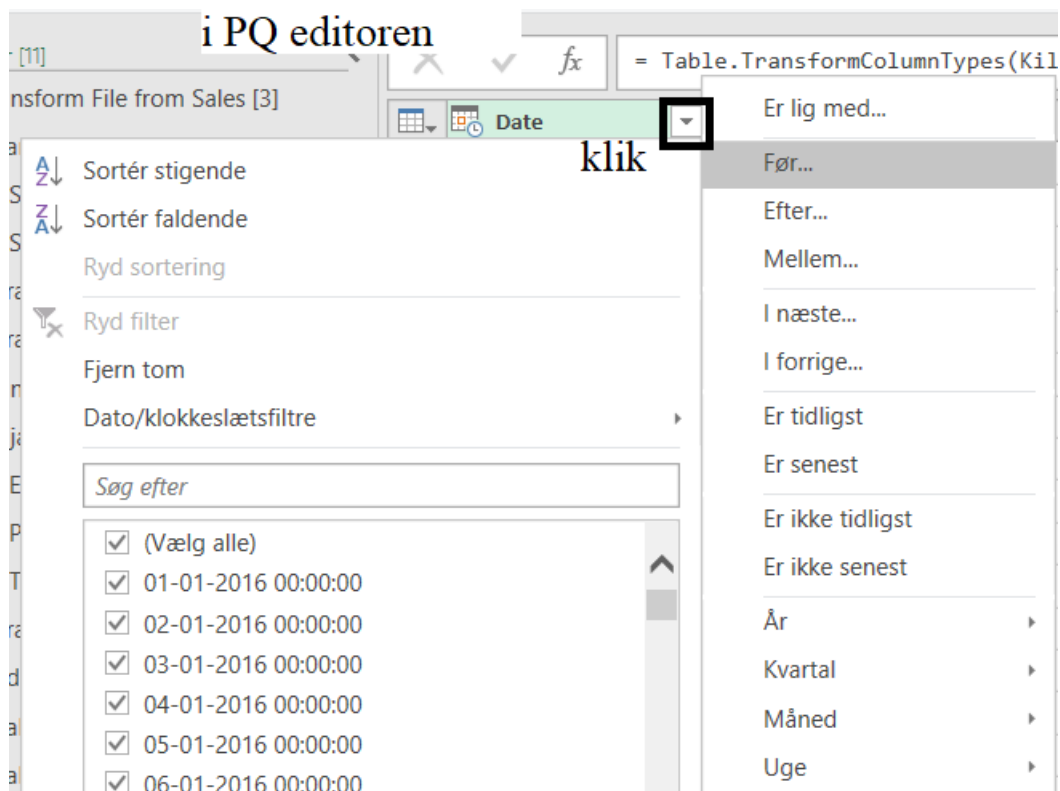
Bemærk

Hvis du vil se en komplet liste over tilgængelige connectors i Power Query, skal du gå til [Forbind orer i Power Query](#).

Filtr tidligt

Det anbefales altid at filtrere dine data i de tidlige faser af din forespørgsel eller så tidligt som muligt. Nogle connectors vil drage fordel af dine filtre via forespørgselsdelegering, som beskrevet i [Oversigt over forespørgselsevaluering og forespørgselsdelegering i Power Query](#). **Det er også bedste praksis at bortfiltrere alle data, der ikke er relevante for projektet.** Dette giver dig bedre mulighed for at fokusere på din opgave ved kun at vise data, der er relevante i afsnittet med forhåndsvisning af data, og **datamængder bliver selvfølgelig mindre**

Du kan bruge menuen til automatisk filtrering, der viser en særskilt liste over de værdier, der findes i kolonnen, til at vælge de værdier, du vil beholde eller filtrere ud. Du kan også bruge søgelinjen til at hjælpe dig med at finde værdierne i kolonnen.



Filtrering med M i editor:

```
"Filtered Rows" = Table.SelectRows("#before ", each ([Sales]>5000))
```

Du kan også drage fordel af de typespecifikke filtre, f.eks. **I forrige** forrige for en dato, datetime eller endda datotidszonekolonne.

Vælges fx før 01-02-2016, fås i editoren bagefter

```
"Filtrerede rækker" = Table.SelectRows("#Ændret type", each [Date] < #datetime(2016, 2, 1, 0, 0, 0))
```

Disse typespecifikke filtre kan hjælpe dig med at oprette et dynamisk filter, der altid henter data i det forrige x antal sekunder, minutter, timer, dage, uger, måneder, kvartaler eller år, som vist på følgende billede.

Filter rows

Apply one or more filter conditions to the rows in this table.

☒ Basic ☐ Advanced

Keep rows where "Date"

is in the previous



90

☒ and ☐ or



Enter a value



Bemærk

Hvis du vil vide mere om filtrering af dine data baseret på værdier fra en kolonne, skal du gå til [Filtrer efter værdier](#).

Lav dyre handlinger, som skal gennemgå hele tabellen, sidst

Visse handlinger kræver læsning af hele datakilden for at returnere *resultater* og vil derfor være langsomme at få vist i Power Query-editor. Hvis du f.eks. udfører en sortering, er det muligt, at de første par sorterede rækker er i slutningen af kildedataene. Hvis du vil returnere resultater, skal sorteringshandlingen først læse *alle* rækkerne.

Andre handlinger (f.eks. filtre) behøver ikke at læse alle dataene, før der returneres nogen resultater. I stedet arbejder de med dataene på en såkaldt "streaming"-måde. Dataene "streams" efter, og resultaterne returneres undervejs. I Power Query-editor behøver sådanne handlinger kun at læse nok af kildedataene for at udfylde eksemplet.

Når det er muligt, skal du udføre sådanne streaminghandlinger først og udføre dyrere handlinger sidst. Dette hjælper med at minimere den tid, du bruger på at vente på, at prøveversionen gengives, hver gang du følger et nyt trin til forespørgslen.

Arbejd midlertidigt mod et undersæt af dine data, med fx 'bevar første 100 rækker'

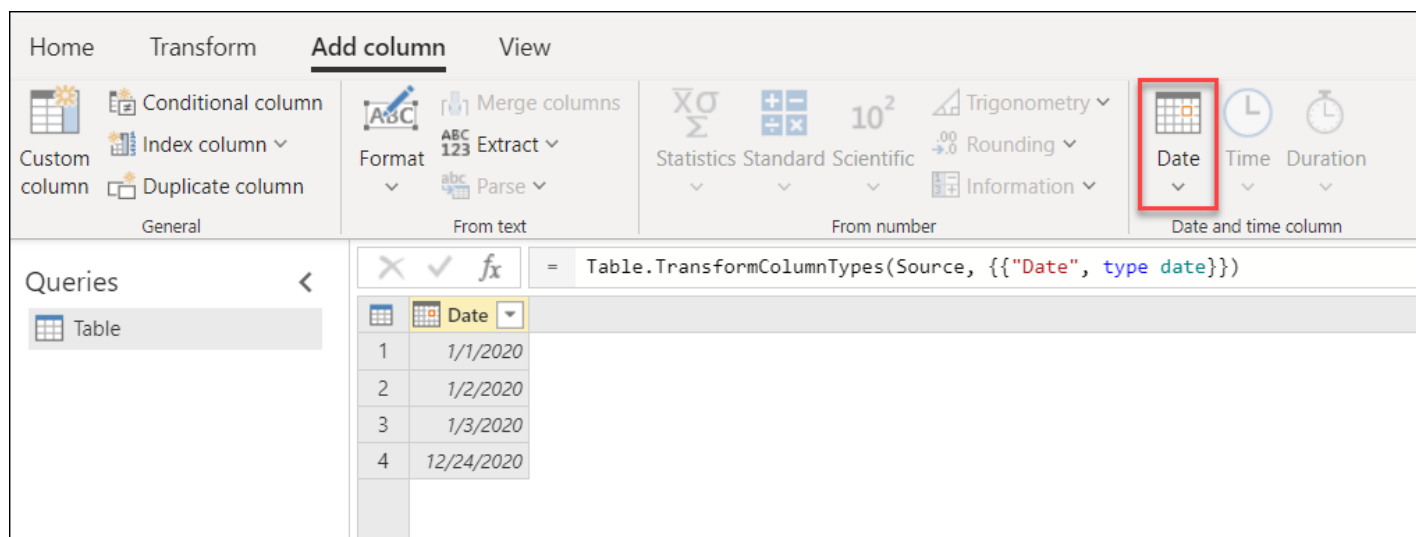
Hvis det tager lang tid at føje nye trin til forespørgslen i Power Query-editor, kan du overveje først at udføre handlingen "Bevar de første rækker" og begrænse antallet af rækker, du arbejder imod.

Når du derefter har tilføjet alle de trin, du har brug for, skal du fjerne trinnet "Bevar de første rækker".

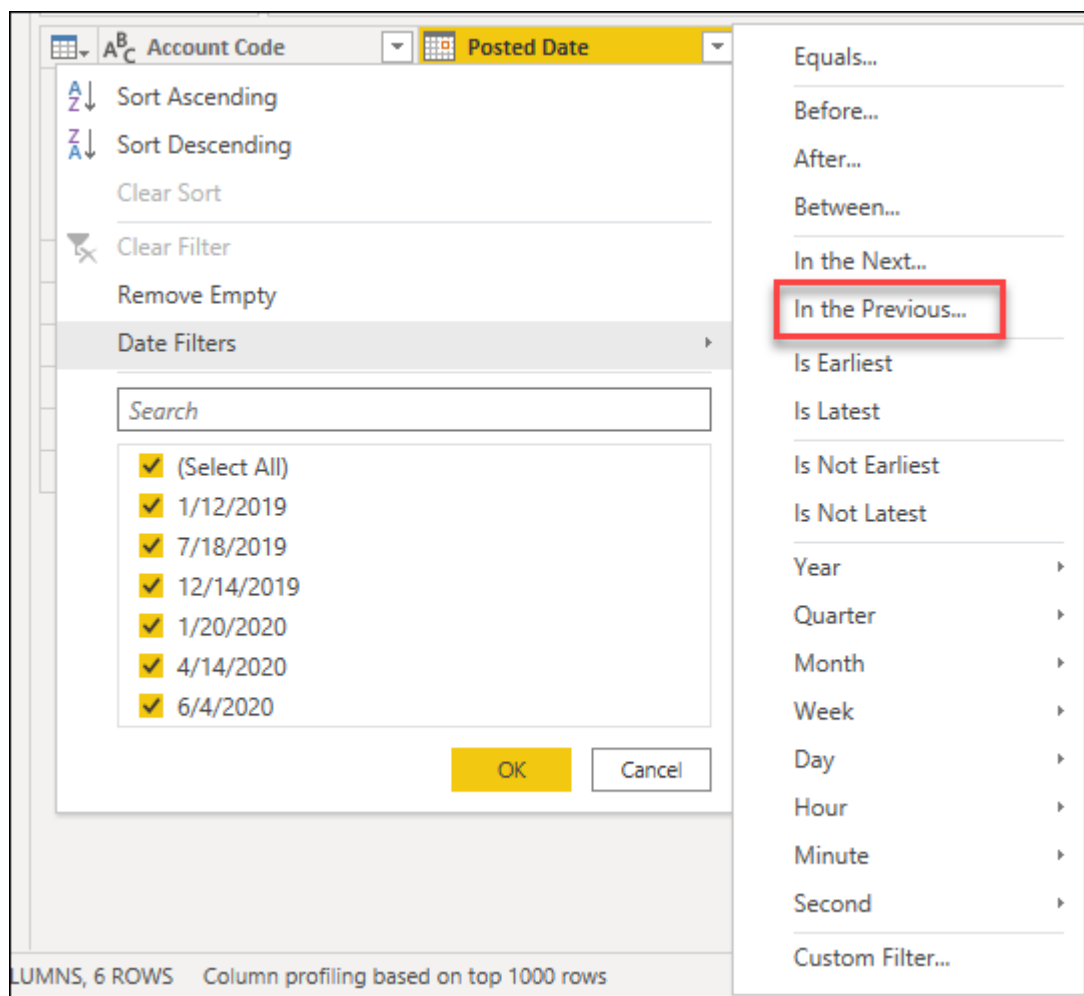
Der arbejdes med max 1000 rækker i PQ, hele tabellen generes først efter save tabel

Brug de korrekte datatyper

Nogle funktioner i Power Query er kontekstafhængige for datatypen for den valgte kolonne. Når du f.eks. vælger en datokolonne, vil de tilgængelige indstillinger under **kolonnegruppen** Dato og klokkeslæt i **menuen Tilføj kolonne** være tilgængelige. Men hvis kolonnen ikke har et datatypesæt, nedtones disse indstillinger.



Der opstår en lignende situation for de typespecifikke filtre, da de er specifikke for visse datatyper. Hvis kolonnen ikke har defineret den korrekte datatype, vil disse typespecifikke filtre ikke være tilgængelige.



Det er afgørende, at du altid arbejder med de korrekte datatyper for dine kolonner. Når du arbejder med strukturerede datakilder, f.eks. databaser, hentes oplysningerne om datatypen fra tabelskemaet, der findes i databasen. Men for ustrukturerede datakilder, f.eks. TXT- og CSV-filer, er det vigtigt, at du angiver de korrekte datatyper for de kolonner, der kommer fra den pågældende datakilde. Power Query tilbyder som standard automatisk registrering af datatyper for ustrukturerede datakilder. Du kan læse mere om denne funktion, og hvordan den kan hjælpe dig i [Datatyper](#).

Bemærk

Hvis du vil vide mere om vigtigheden af datatyper, og hvordan du arbejder med dem, skal du se [Datatyper](#).

Udforsk dine data, lav fx Deskriptiv statistik over tal-kolonner

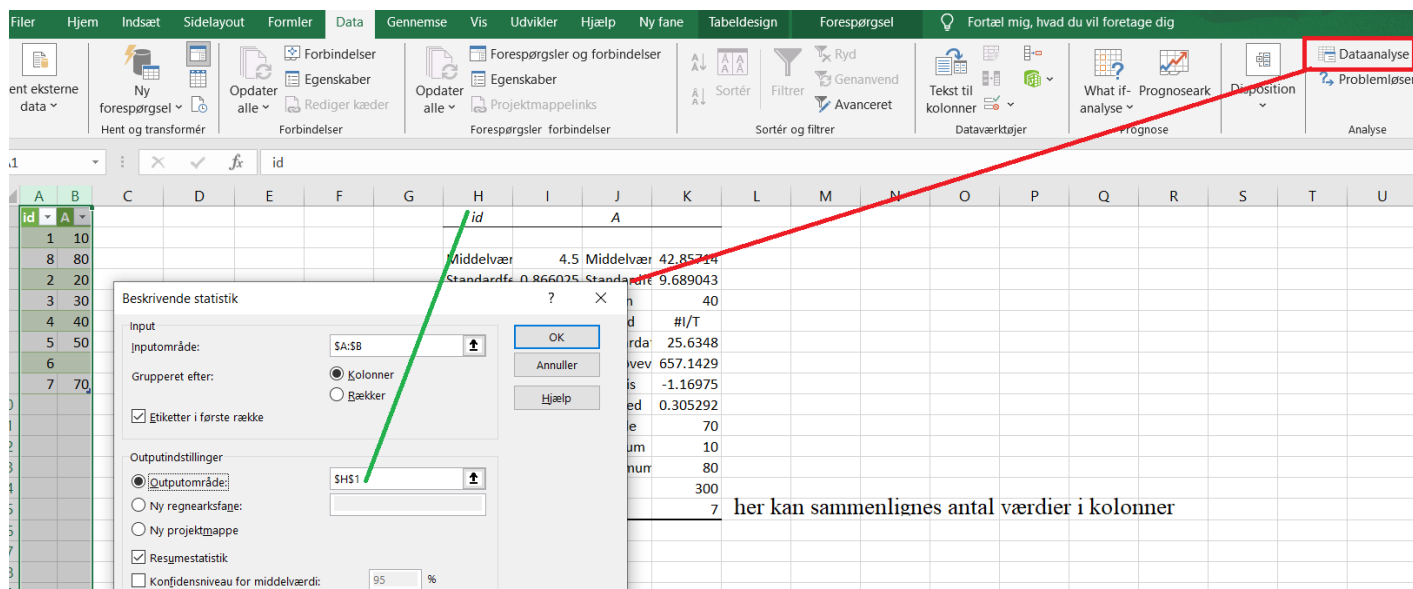
TÆL.HVIS(txt-kolonne;"") for tekst-kolonner kan finde antal tomme felter

Det er nemt at fjerne rækker med Null-værdier i PQ

R og Python har dog bedre værktøjer til problemdata:

Som i Excel kan man hurtigt fjerne rækker med Null, men man kan også få udfyldt Null-værdier med fx medianen for kolonnen, så hele rækkens data ikke går tabt.

Og i R kan jeg få udskrevet rækker og kolonner med Null-værdier, i stedet for blot at slette rækker.

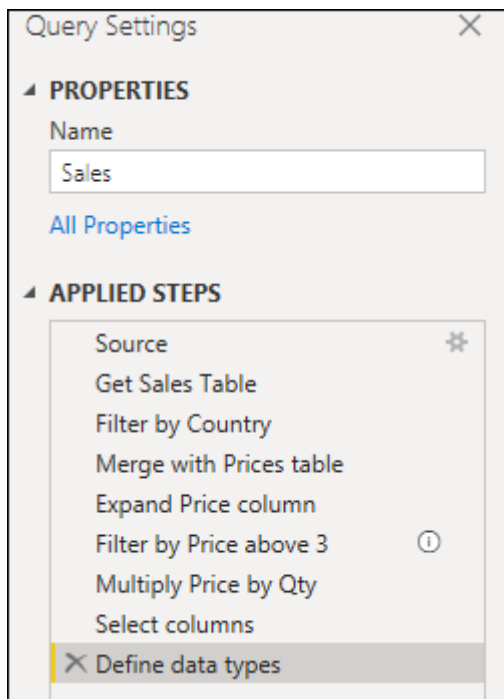


Data-Profileringsværktøj, ikke i min Excelversion

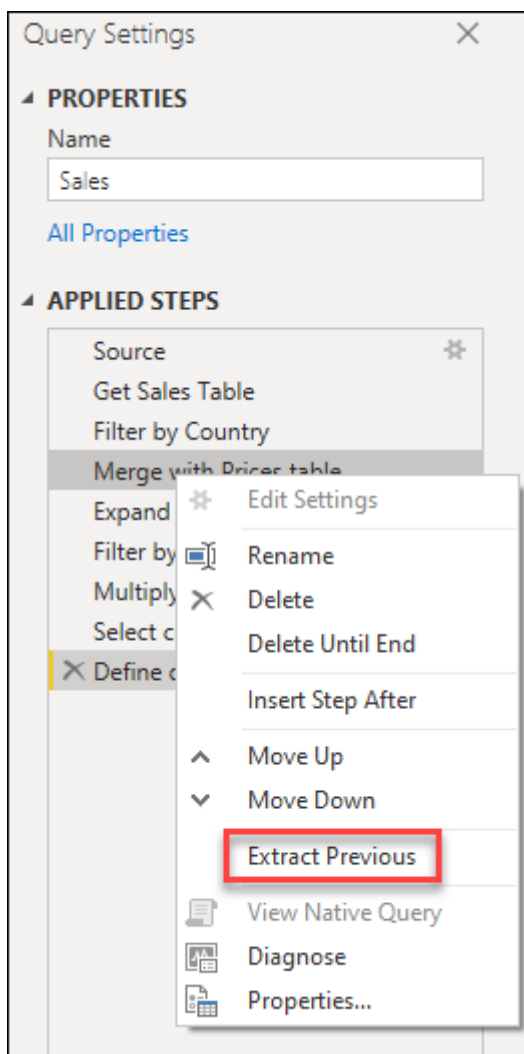
Tag en modulopbygget tilgang: opdel kompleks query i mindre dele

Det er helt muligt at oprette en enkelt forespørgsel, der indeholder alle de transformationer og beregninger, du har brug for. Men hvis forespørgslen indeholder et stort antal trin, kan det være en god idé at opdele forespørgslen i flere forespørgsler, hvor den ene forespørgsel refererer til den næste. Målet med denne tilgang er at forenkle og afkoble transformationsfaser i mindre dele, så de er nemmere at forstå.

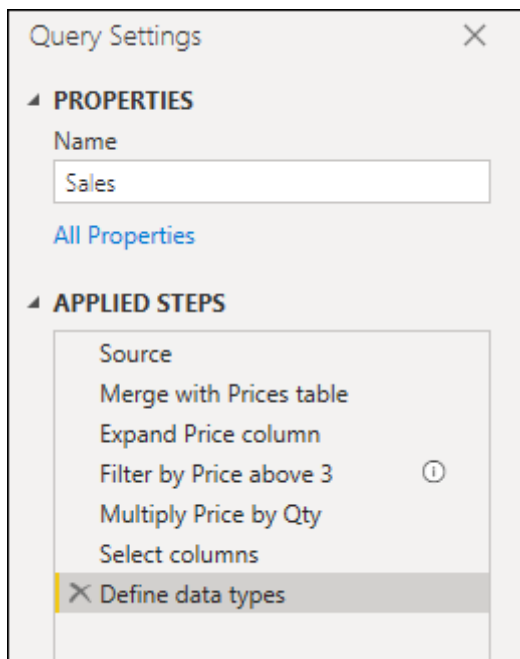
Lad os f.eks. sige, at du har en forespørgsel med de ni trin, der vises på følgende billede.



Du kan opdele denne forespørgsel i to på tabeltrinnet Flet med priser. På den måde er det nemmere at forstå de trin, der blev anvendt på salgsforespørgslen før fletningen. Hvis du vil udføre denne handling, skal du højreklikke på **trinnet Flet med priser** og vælge indstillingen **Udtræk forrige** .



Du bliver derefter bedt om at angive et navn til den nye forespørgsel med en dialogboks. Dette opdeler effektivt din forespørgsel i to forespørgsler. Én forespørgsel har alle forespørgslerne før fletningen. Den anden forespørgsel har et indledende trin, der refererer til den nye forespørgsel og resten af de trin, du havde i den oprindelige forespørgsel, fra **trinnet Flet med priser** nedad.



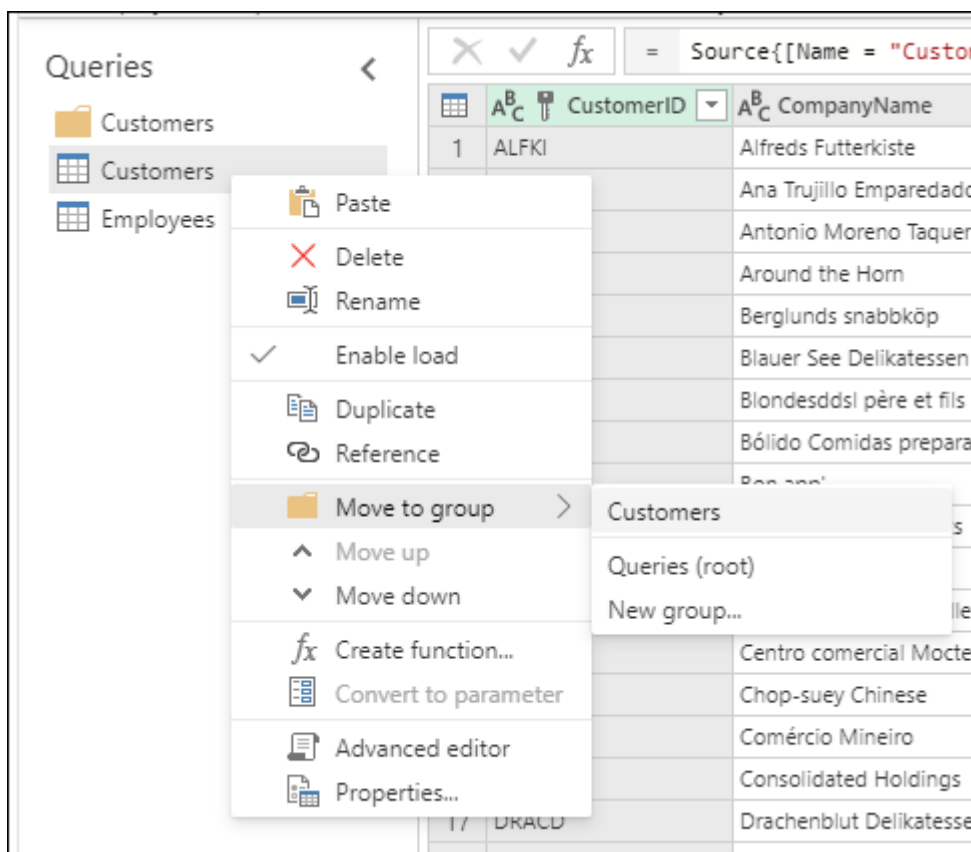
Du kan også udnytte brugen af forespørgselsreferencer, som du vil. Men det er en god idé at holde dine forespørgsler på et niveau, der ikke virker skræmmende ved første øjekast med så mange trin.

Bemærk

Hvis du vil vide mere om forespørgselsreferencer, skal du gå til [Om ruden](#) Forespørgsler.

[Opdel Queries i grupper, især ved store projekter](#)

En god måde at holde dit arbejde organiseret på er ved at udnytte brugen af grupper i forespørgselsruden.



Det eneste formål med grupper er at hjælpe dig med at holde dit arbejde organiseret ved at fungere som mapper til dine forespørgsler. Du kan oprette grupper i grupper, hvis du får brug for det. Det er lige så nemt at flytte forespørgsler på tværs af grupper som at trække og slippe.

Prøv at give dine grupper et meningsfuldt navn, der giver mening for dig og din sag.

Bemærk

Hvis du vil vide mere om alle de tilgængelige funktioner og komponenter, der findes i ruden Forespørgsler, skal du gå til [Om ruden](#) Forespørgsler.

Fremtidssikrede forespørgsler

Det er en topprioritet at sikre, at du opretter en forespørgsel, der ikke har problemer under en fremtidig opdatering. Der er flere funktioner i Power Query, der gør din forespørgsel modstandsdygtig over for ændringer og kan opdateres, selvom nogle komponenter i datakilden ændres.

Det er bedste praksis at definere omfanget af din forespørgsel for, hvad den skal gøre, og hvad den skal tage højde for med hensyn til struktur, layout, kolonnenavne, datatyper og enhver anden komponent, som du anser for relevant for området.

Nogle eksempler på transformationer, der kan hjælpe dig med at gøre din forespørgsel modstandsdygtig over for ændringer, er:

- Hvis din forespørgsel har et dynamisk antal rækker med data, men et fast antal rækker, der fungerer som den sidefod, der skal fjernes, kan du bruge funktionen **Fjern nederste rækker** .

Bemærk

Hvis du vil vide mere om filtrering af dine data efter rækkeposition, skal du gå til [Filtrer en tabel efter rækkeplacering](#).

- Hvis din forespørgsel har et dynamisk antal kolonner, men du kun skal vælge bestemte kolonner fra dit datasæt, kan du bruge funktionen **Vælg kolonner** .

Bemærk

Hvis du vil vide mere om at vælge eller fjerne kolonner, skal du gå til [Vælg eller fjern kolonner](#).

- Hvis din forespørgsel har et dynamisk antal kolonner, og du kun skal frigøre et undersæt af dine kolonner, kan du bruge **funktionen Unpivot only selected columns** .

Bemærk

Hvis du vil vide mere om indstillingerne for at frigøre dine kolonner, skal du gå til [Unpivot-kolonner](#).

- Hvis din forespørgsel indeholder et trin, der ændrer datatypen for en kolonne, men nogle celler returnerer fejl, da værdierne ikke er i overensstemmelse med den ønskede datatype, kan du fjerne de rækker, der gav fejlværdier.

Bemærk

Hvis du vil have mere at vide om at arbejde og håndtere fejl, skal du gå til [Håndtering af fejl](#).

Brug parametre

Oprettelse af dynamiske og fleksible forespørgsler er bedste praksis. Parametre i Power Query hjælper dig med at gøre dine forespørgsler mere dynamiske og fleksible. En parameter fungerer som en metode til nemt at gemme og administrere en værdi, der kan genbruges på mange forskellige måder. Men det bruges mere almindeligt i to scenarier:

- **Argument for trin** – Du kan bruge en parameter som argument for flere transformationer, der er baseret på brugergrænsefladen.

Filter rows ?

Apply one or more filter conditions to the rows in this table.

☒ Basic ☐ Advanced

Keep rows where "Margin"

is greater than ▼ 1.2 Minimum Margin ▼

☒ and ☐ or

▼ ▼

OK Cancel

Luk og indlæs

Opdater eksempel

Administrer

Vælg kolonner

Fjern kolonner

Bevar rækker

Forminds

Luk

Forespørgsel

Administrer kolonner

Forminds

Forespørgsler [6]

- TabA
- TabB
- TabC
- MergedTables
- TableCreatedRowAdded
- highestID (3)

12 id

1	1
2	2

Filtrer rækker

Anvend en eller flere filterbetingelser på rækkerne i denne tabel.

☒ Grundlæggende ☐ Avanceret

Behold rækker, hvor 'id'

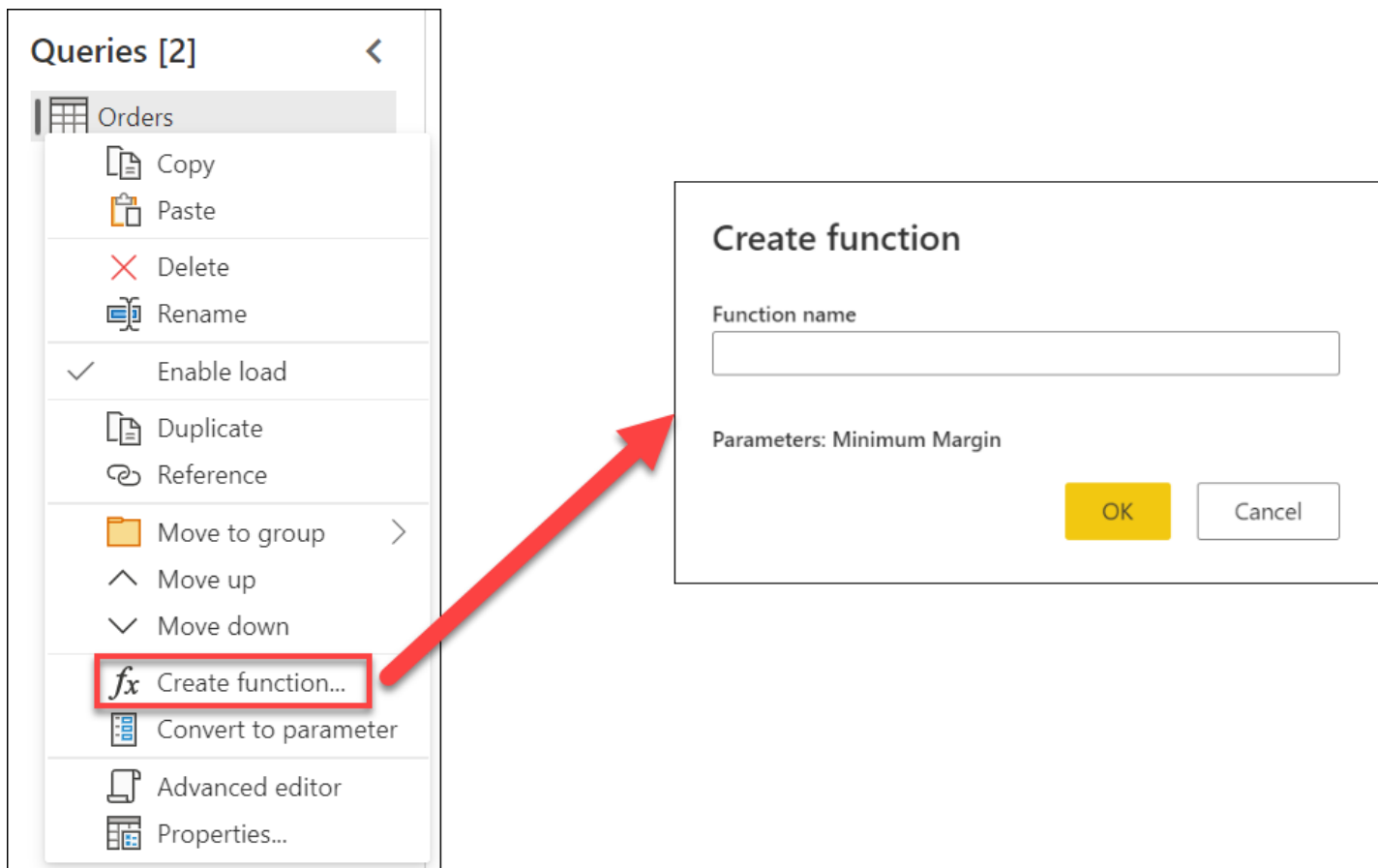
er mindre end ▼ highestID ▼

☒ Og ☐ Eller

▼ 1.2 Angiv eller vælg en vær... ▼

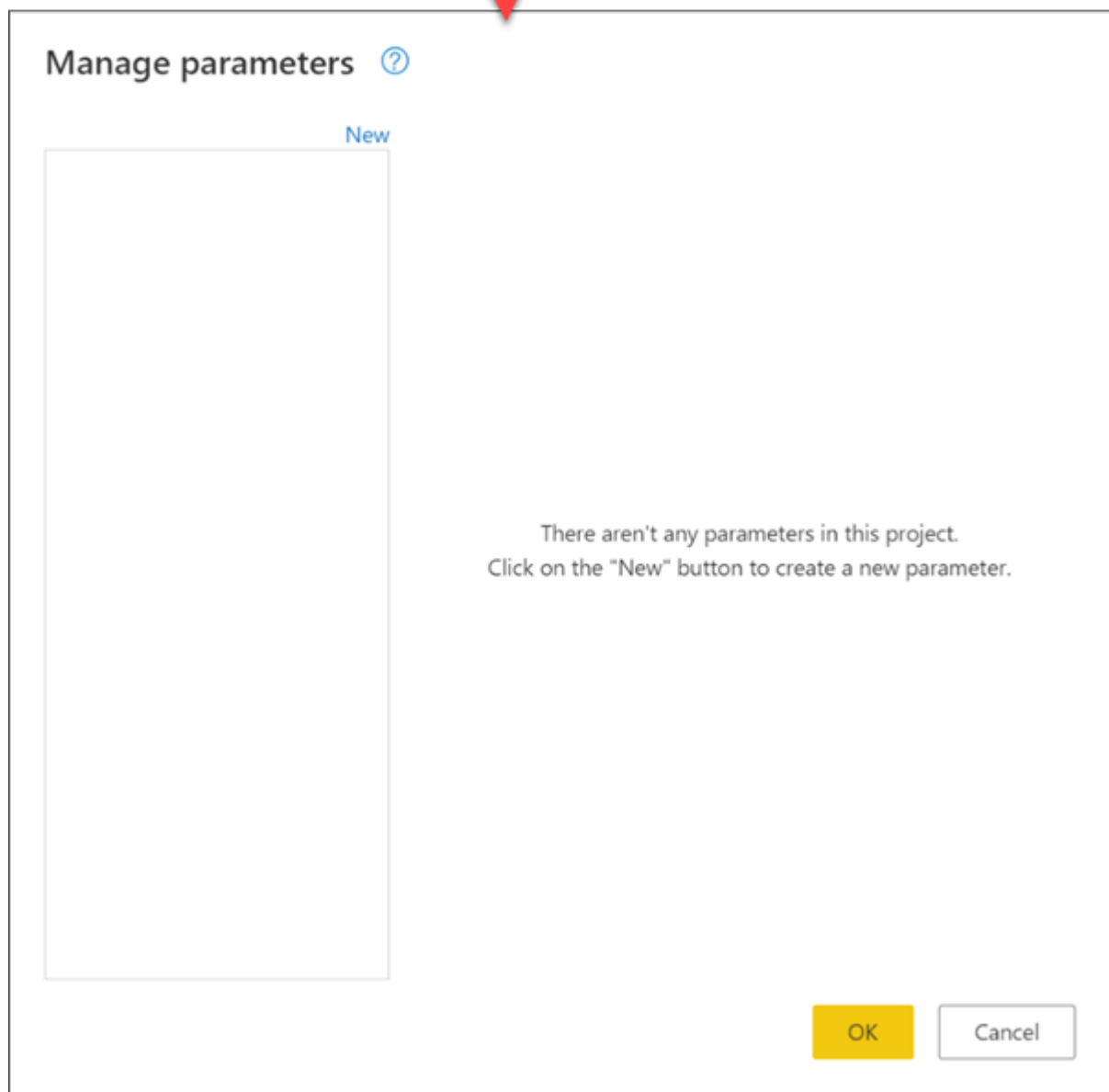
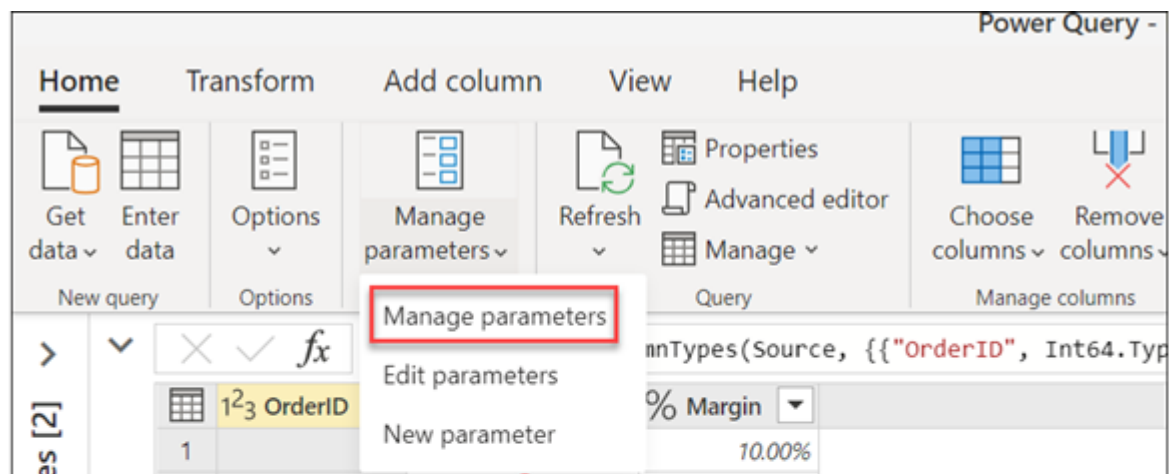
= Table.SelectRows("#Udvidet TabC", each [id] < highestID)

- **Brugerdefineret funktionsargument** – Du kan oprette en ny funktion ud fra en forespørgsel og referere til parametre som argumenterne for din brugerdefinerede funktion.



De vigtigste fordele ved at oprette og bruge parametre er:

- Centraliseret visning af alle dine parametre via vinduet **Administrer parametre** .



- Genbrug af parameteren i flere trin eller forespørgsler.
- Gør det nemt og enkelt at oprette brugerdefinerede funktioner.

Du kan endda bruge parametre i nogle af argumenterne i dataconnectorerne. Du kan f.eks. oprette en parameter for servernavnet, når du opretter forbindelse til sql Server-databasen. Derefter kan du bruge denne parameter i dialogboksen SQL Server-database.

The screenshot shows a dialog box titled "SQL Server database". It contains several input fields and options. The "Server" field, which includes a dropdown icon and the text "Server name", is highlighted with a red rectangular box. Below this is the "Database (optional)" field, which has a dropdown icon and the text "AdventureWorks". Further down, there is a section for "Data Connectivity mode" with two radio buttons: "Import" (which is selected) and "DirectQuery". At the bottom left, there is a link for "Advanced options". At the bottom right, there are two buttons: "OK" (in a yellow box) and "Cancel" (in a white box with a grey border).

Hvis du ændrer serverplaceringen, skal du bare opdatere parameteren for servernavnet, så opdateres dine forespørgsler.

Bemærk

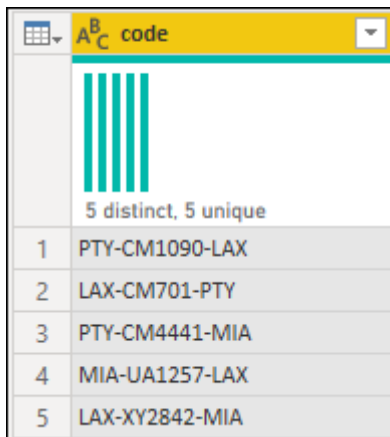
Hvis du vil vide mere om oprettelse og brug af parametre, skal du gå til [Brug af parametre](#).

Opret funktioner, der kan genbruges

Hvis du befinder dig i en situation, hvor du har brug for at anvende det samme sæt transformationer på forskellige forespørgsler eller værdier, kan det være en fordel at oprette en brugerdefineret funktion i Power Query, der kan genbruges så mange gange, du har brug for. En brugerdefineret funktion i Power Query er en tilknytning fra et sæt inputværdier til en enkelt outputværdi og oprettes ud fra oprindelige M-funktioner og -operatorer.

Lad os f.eks. sige, at du har flere forespørgsler eller værdier, der kræver det samme sæt transformationer. Du kan oprette en brugerdefineret funktion, der senere kan aktiveres i forhold til de forespørgsler eller værdier, du vælger. Denne brugerdefinerede funktion sparer dig tid og hjælper dig med at administrere dine transformationer på en central placering, som du kan ændre når som helst.

Brugerdefinerede funktioner i Power Query kan oprettes ud fra eksisterende forespørgsler og parametre. Forestil dig f.eks. en forespørgsel, der har flere koder som en tekststreng, og du vil oprette en funktion, der afkode disse værdier.



code	
1	PTY-CM1090-LAX
2	LAX-CM701-PTY
3	PTY-CM4441-MIA
4	MIA-UA1257-LAX
5	LAX-XY2842-MIA

Du starter med at have en parameter, der har en værdi, der fungerer som et eksempel.

Manage Parameters

New

A^BC code

Name

code

Description

Sample code

☒ Required

Type

Text

Suggested Values

Any value

Current Value

PTY-CM1090-LAX

OK

Cancel

Fra denne parameter opretter du en ny forespørgsel, hvor du anvender de transformationer, du har brug for. I dette tilfælde vil du opdele koden **PTY-CM1090-LAX** i flere komponenter:

- **Oprindelse** = PTY
- **Destination** = LAX
- **Flyselskab** = CM
- **FlightID** = 1090

✕

✓

f_x

= Table.TransformColumnTypes(RowToTable,{{"FlightID", type text}, {"Airline", type

▼

	A_C^B Origin	A_C^B Destination	A_C^B Airline	A_C^B FlightID
	1 distinct, 1 unique	1 distinct, 1 unique	1 distinct, 1 unique	1 distinct, 1 unique
1	PTY	LAX	CM	1090

Payments

Player	Payments
Abraham	50
Alea	50
Ali	50
Armando	15
Armando	15
Athena	50
Aurelia	20
Aurelia	20
Aurelia	10
Axel	40
Axel	10
Belle	25
Berk	50
Blair	50
Breanna	50
Brendan	50

Import Data

Select how you want to view this data in your workbook:

- ☒ Table
- ☐ PivotTable Report
- ☐ PivotChart
- ☐ Only Create Connection

Where do you want to put the data?

- ☐ Existing worksheet: =SC511
- ☒ New worksheet

☒ Add this data to the Data Model

OK Cancel

Players

Athlete	Season	Club
Abraham	4	Pirates
Alea	3	Kangaroos
Ali	6	Mythics
Armando	6	Lightning
Athena	2	Icebergs
Aurelia	5	Kangaroos
Axel	1	Angels
Belle	3	Tumbleweeds
Berk	3	Kangaroos
Blair	3	The Calm

Teams

City	Team	Division	Coaches
Crimson Park	Tumbleweeds	East	Alice
Grayhurst	Angels	West	Marshall
	Mythics	West	Franklin
	Orange Devils	West	Jimmy
Hellim	Blue Raiders	Central	Ursula
	The Calm	Central	Andy
Lemon Grove	Dragons	Central	Lois
	Kangaroos	East	Rita-Jean
	Pirates	West	Kate
	Tigers	Central	Don

A total for Armando instead of multiple entries

Modelværktøjet

Power Pivot for Excel - Data Models - Group and Teams.xlsx

File Home Design Advanced

Paste From Database From Data Service From Other Sources Existing Connections Refresh PivotTable

Clipboard Get External Data Formatting Sort and Filter Find Calculations Data View Diagram View Show Hidden Calculation Area

Players

- Athlete
- Season
- Club

Teams

- City
- Team
- Division
- Coaches

Payments

- Player
- TotalPD

automatisk opdateres

Modeller kan bruges til at lave pivottabeller i Excel, ligesom i PowerBI

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									

Count of Athlete	Column Labels								
Row Labels	Crimson Park	Grayhurst	Hellim	Lemon Grove	St. Lucia	Stone Beach	Grand Total		
Angels		10					10		
Blue Raiders			9				9		
Dragons					8		8		
Icebergs						10	10		
Kangaroos					11		11		
Lightning						10	10		
Mythics		9					9		
Noise					9		9		
Orange Devils		9					9		
Pirates					10		10		
Ranchers						8	8		
The Calm			11				11		
Tigers					9		9		
Tumbleweeds	9						9		
Grand Total	9	28	20		38	18	132		

Grayhurst har 3 klubber: Angels, Mythics, Orange Devils; Angels har åbenbart 10 players registreret; i alt findes 132 players

13 spiller har kun spillet 1 sæson, er 'first year players

Count of Player	Column Labels						
Row Labels	1	2	3	4	5	6	Grand Total
Central	6	12	13	4	10	2	47
East	3	6	11	12	13	2	47
West	4	8	6	10	6	4	38
Grand Total	13	26	30	26	29	8	132

Players

Athlete

Season

Club

Teams

City

Team

Division

Coaches

Payments

Players

Division 2

Drag fields between areas below:

Filters

Columns

Season

Rows

Division

Σ Values

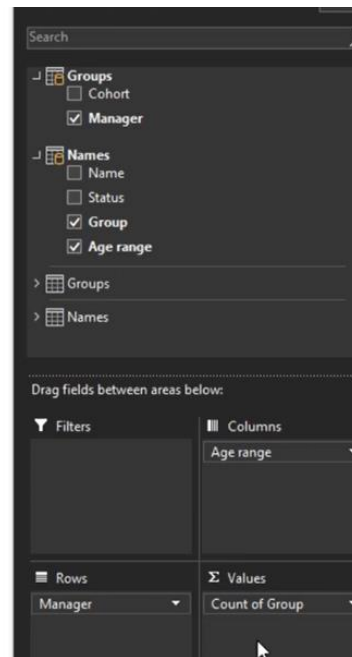
Count of Player

Name	Status	Group	Age range
Alexi	Inactive	Crimson	C
Armine	Active	Crimson	A
Bobbi	Active	Orange	A
Candide	Active	Purple	C
Cornelius	Active	Orange	A
Davi	Inactive	Purple	C
Debbie	Active	Yellow	B
Dory	Inactive	Yellow	B
Drew	Active	Crimson	C
Finch	Active	Purple	A
Manny	Active	Purple	C
Regina	Inactive	Yellow	B
Robin	Active	Crimson	B
Sugar	Active	Purple	A
Ted	Active	Orange	C

Cohort	Manager
Crimson	Evi
Orange	Nathalie
Purple	Zoraya
Yellow	Evi

t.v. nok en række personer i forskellige (spejder?)grupper, t.h. lederen af grupperne.

Count of Group	Column Labels			
Row Labels	A	B	C	Grand Total
Evi	1	4	2	7
Nathalie	2	1		3
Zoraya	2	3		5
Grand Total	5	4	6	15



PQ in PowerBI

Importerer tabeller ligesom i Excel

	A	B	C	D	E	F	G	H
1								
2								
3	Apocalypse Food Prep Overview							
4	Location	Products	January-22	February-22	March-22	April-22	Grand Total	
5	Costco	Bottled Water	\$4.25	\$4.49	\$4.80	\$4.89	\$18.43	
6	Costco	Canned Vegetables	\$0.85	\$0.85	\$0.85	\$0.85	\$3.40	
7	Costco	Dried Beans	\$20.99	\$22.99	\$22.99	\$23.99	\$90.96	
8	Costco	Duct Tape	\$5.00		\$5.25		\$10.25	
9	Costco	Flashlight	\$13.00		\$13.00		\$26.00	
10	Costco	Milk	\$2.50	\$2.40	\$2.50	\$2.70	\$9.90	
11	Costco	Rice	\$23.00	\$23.99	\$24.99	\$25.25	\$97.23	
12	Costco	Rope	\$13.50		\$14.00		\$27.50	
13	Costco	Water Filter	\$29.00		\$31.00		\$60.00	
14		Costco Total	\$111.89	\$54.72	\$119.38	\$57.68	\$343.67	
15	Target	Bottled Water	\$4.99	\$5.15	\$5.25	\$5.35	\$20.74	
16	Target	Canned Vegetables	\$0.65	\$0.69	\$0.74	\$0.79	\$2.87	
17	Target	Dried Beans	\$23.99	\$24.99	\$25.45	\$25.79	\$100.22	
18	Target	Duct Tape	\$6.00		\$6.25		\$12.25	
19	Target	Flashlight	\$12.50		\$12.75		\$25.25	
20	Target	Milk	\$2.20	\$2.45	\$2.70	\$3.00	\$10.35	
21	Target	Rice	\$22.99	\$23.25	\$24.25	\$24.75	\$95.24	
22	Target	Rope	\$15.00		\$15.25		\$30.25	
23	Target	Water Filter	\$38.00		\$39.00		\$77.00	
24		Target Total	\$126.32	\$56.53	\$131.64	\$59.68	\$374.17	
25	Walmart	Bottled Water	\$5.00	\$5.25	\$5.45	\$5.75	\$21.45	
26	Walmart	Canned Vegetables	\$0.65	\$0.75	\$0.85	\$1.00	\$3.25	
27	Walmart	Dried Beans	\$22.99	\$23.99	\$24.25	\$24.49	\$95.72	
28	Walmart	Duct Tape	\$5.00		\$5.50		\$10.50	
29	Walmart	Flashlight	\$10.00		\$11.00		\$21.00	
30	Walmart	Milk	\$2.00	\$2.10	\$2.40	\$2.90	\$9.40	
31	Walmart	Rice	\$25.00	\$25.75	\$26.99	\$26.99	\$104.73	
32	Walmart	Rope	\$14.00		\$14.50		\$28.50	
33	Walmart	Water Filter	\$39.00		\$39.00		\$78.00	
34		Walmart Total	\$123.64	\$57.84	\$129.94	\$61.13	\$372.55	
35		Grand Total	\$561.85	\$169.09	\$580.96	\$178.49	\$1,090.39	
36								

typisk

pivottabel, som skal bearbejdes. Tabellerne importeres, og bearbejdes med PQ i PBI, på samme måde som i ren Excel

Navigator

Display Options ▾

- Apocalypse Food Prep.xlsx [3]
 - ☒ Pivot Table
 - ☒ Purchase Overview
 - ☐ Purchase Tracker
- Suggested Tables [1]
 - ☐ Table 1 (Purchase Overview)

Purchase Overview

Preview downloaded on Friday, May 13, 2022

Apocalypse Food Prep Overview	Column2	Column3	Column4
	null	null	null
	null	null	null
Location	Products	1/1/2022	
Costco	Bottled Water	4.25	
Costco	Canned Vegetables	0.85	
Costco	Dried Beans	20.99	
Costco	Duct Tape	5	
Costco	Flashlight	23	
Costco	Milk	2.3	
Costco	Rice	23	
Costco	Rope	13.5	
Costco	Water Filter	29	
	null Costco Total	111.89	
Target	Bottled Water	4.99	
Target	Canned Vegetables	0.65	
Target	Dried Beans	23.99	
Target	Duct Tape	6	
Target	Flashlight	12.5	
Target	Milk	2.2	
Target	Rice	22.99	
Target	Rope	15	
Target	Water Filter	38	

Load

Transform Data

Når der trykkes Transform Data, åbnes PQ:

Queries [2] < This preview may be up to 18 days old. Refresh

Pivot Table

Purchase Overview

Table.TransformColumnTypes(#"Promoted Headers",{"Apocalypse Food Prep Overview", type text}, {"Column2", type text}, {"Column3", type any

	Apocalypse Food Prep Overview	Column2	Column3	Column4	Column5	Column6
1		null	null	null	null	null
2		null	null	null	null	null
3	Location	Products	1/1/2022	2/1/2022	3/1/2022	4/1/2022
4	Costco	Bottled Water	4.25	4.49	4.8	4.89
5	Costco	Canned Vegetables	0.85	0.85	0.85	0.85
6	Costco	Dried Beans	20.99	22.99	22.99	23.99
7	Costco	Duct Tape	5	null	5.25	null
8	Costco	Flashlight	13	null	13	null
9	Costco	Milk	2.3	2.4	2.5	2.7
10	Costco	Rice	23	23.99	24.99	25.25
11	Costco	Rope	13.5	null	14	null
12	Costco	Water Filter	29	null	31	null
13		Costco Total	111.89	54.72	119.38	57.68
14	Target	Bottled Water	4.99	5.15	5.25	5.35
15	Target	Canned Vegetables	0.65	0.69	0.74	0.79
16	Target	Dried Beans	23.99	24.99	25.45	25.79
17	Target	Duct Tape	6	null	6.25	null
18	Target	Flashlight	12.5	null	12.75	null
19	Target	Milk	2.2	2.45	2.7	3
20	Target	Rice	22.99	23.25	24.25	24.75
21	Target	Rope	15	null	15.25	null
22	Target	Water Filter	38	null	39	null

Query Editor

Add Column View Tools Help

Data Type: Fixed decimal number 1,2 Replace Values Unpivot Columns

Rows Detect Data Type Fill Move

Columns Rename Pivot Column Convert to List

Any Column Text Column Number Column Date & Time Column

Table.RemoveColumns(#"Flattened")

Unpivot Columns

Translate all but the currently unselected columns into attribute-value pairs.

	Location	Products	2/1/2022	3/1/2022	4/1/2022
1	Costco	Bottled Water	4.25	4.49	4.80
2	Costco	Canned Vegetables	0.85	0.85	0.85
3	Costco	Dried Beans	20.99	22.99	22.99
4	Costco	Duct Tape	5.00	null	5.25
5	Costco	Flashlight	13.00	null	13.00
6	Costco	Milk	2.30	2.40	2.50
7	Costco	Rice	23.00	23.99	24.99

Unpivot tabel, så hver række kun har én rekord: