

webbojo

# Power Query



Bo Jönsson

# Power Query

**Bo Jönsson**

UDGIVET AF

WEBBOJO

© 2021 af Bo Jönsson

Alle rettigheder forbeholdes. Ingen del af indholdet i denne bog må gengives eller transmitteres i nogen form eller på nogen måde uden skriftlig tilladelse fra udgiveren.

ISBN: "kommer snarest"

Denne bog udgives "som den er" og udtrykker forfatterens synspunkter og meninger. De synspunkter, meninger og oplysninger, der udtrykkes i denne bog, inklusive URL og andre internetreferencer på webstedet, kan ændre sig uden varsel.

Nogle eksempler, der er afbildet heri, er kun til illustration og er funktionelle. Ingen reel tilknytning eller forbindelsen er tiltænkt eller skal udledes.

# Indhold

---

<b>Kapitel 1 Hvad er Power Query?</b> .....	<b>1</b>
Nogle eksempler på, hvad Power Query kan gøre .....	1
Hvad er forskellen mellem Power Query og Get & Transform .....	2
Hvor vanskeligt er Power Query at lære? .....	2
Hvordan installeres Power Query? .....	2
Tid til at komme i gang .....	2
<b>Kapitel 2 Importer data</b> .....	<b>3</b>
Brug af data i den samme projektmappe .....	3
Importer data fra tabeller .....	3
Forstå de anvendte trin .....	4
M kode .....	5
Tilføjelse af trin .....	6
Redigering af trinnene .....	6
Indlæs dataene i Excel .....	7
Importerer data fra standard områder .....	8
Importerer data fra et navngivet område .....	8
Tandhjulsikonet .....	9
Vis formellinjen .....	9
Er det bedst at bruge navngivne områder eller tabeller? .....	10
Brug af data fra eksterne filer og projektmapper .....	10
CSV-fil .....	10
Tekstfil .....	12
Projektmappe .....	13
<b>Kapitel 3 Opdatering af data</b> .....	<b>17</b>
Opdater alle data .....	17
Opret en standard forespørgsel .....	17
Opdater kildedataene .....	20
Opdater dataene .....	21
Opdater kun specifikke forespørgsler .....	21
Opdater knap .....	21
Menuen Vis forespørgsler .....	22
VBA-opdatering .....	23
Avancerede opdateringsindstillinger .....	23
Vær opmærksom .....	25
Lukket eller Åbne filer .....	25
Forespørgsler baseret på andre forespørgsler .....	25
Opdatering i baggrunden .....	25

## Indhold

<b>Kapitel 4 Rediger forespørgsler</b> .....	<b>26</b>
Rediger en eksisterende forespørgsel.....	26
Åben Power Query.....	26
Vælg en forespørgsel der skal redigeres.....	28
Luk og indlæs.....	29
Luk uden at gemme ændringer.....	30
Rediger trinnene.....	30
Grundlæggende indstillinger.....	30
Tilføj, indsæt, slet og flyt trin.....	31
Flyt trin.....	35
Hvor forsvandt mine trin hen?.....	35
M koder i formellinje og avanceret editor.....	36
Indlæs i Excel.....	37
<b>Kapitel 5 Indstillinger for Luk og indlæs</b> .....	<b>38</b>
Luk og indlæs til... indstillinger.....	38
Sådan ses forespørgselsoutput.....	39
Hvor placeres forespørgslens output.....	39
Føj disse data til datamodellen.....	39
Knappen Egenskaber.....	39
Ret indstilling for Luk og indlæs for en eksisterende forespørgsel.....	39
Fjern og luk.....	41
<b>Kapitel 6 Brug af parametre</b> .....	<b>42</b>
Opret forespørgslen.....	42
Kolonnen Dato.....	42
Kolonnen Solgt af.....	44
Opret parametrene.....	45
Oprettelse af en tekstparameter.....	45
Oprettelse af datoparameteren.....	47
Indsæt parametrene i forespørgslen.....	48
Brug af parameteren.....	49
<b>Kapitel 7 Grundlæggende transformationer</b> .....	<b>51</b>
Fane Hjem.....	51
Administrer kolonner.....	51
Fjern eller fjern andre - betyder det noget?.....	52
Bevar rækker.....	52
Sorter.....	54
Andre sektioner under fanen Hjem.....	55
Transformer.....	55
Tabel.....	55

## Indhold

En hvilken som helst kolonne .....	56
Tekstkolonne.....	57
Tal kolonne .....	59
Dato og klokkeslæt .....	60
Tilføj kolonne.....	61
Fanerne Transformer kontra Tilføj kolonne .....	61
Generelt .....	62
Filtrering .....	63
Andre menuer.....	63
Højre klik på kolonneoverskrift .....	63
Tabelikon .....	64
Skift datatype.....	64
Næste.....	65
<b>Kapitel 8 Kombiner og tilføj forespørgsler .....</b>	<b>66</b>
Opret den første forespørgsel .....	66
Kopier og rediger forespørgslen .....	69
Kombiner forespørgsler .....	71
Når kolonneoverskrifterne er forskellige.....	73
Kombination af filer med forskellige typer/datastrukturer .....	74
<b>Kapitel 9 Power Query – Importer alle filer i en mappe .....</b>	<b>75</b>
Opsætning af filerne til eksemplerne .....	75
Importer alle CSV-filer fra en mappe .....	75
Metode 1: Brug en brugerdefineret kolonne.....	78
Metode 2: Kombiner filerne .....	80
Tilføjelse af flere filer til mappen .....	81
Filens ensartethed er nøglen til succes.....	82
Importer Excel projektmappe fra en mappe .....	82
Metode 1: Brug en brugerdefineret kolonne.....	83
Metode 2: Kombiner filerne .....	84
Afslut forespørgslen.....	85
Bemærkninger og tips.....	86
<b>Kapitel 10 Power Query – Vis alle filer i en mappe og filattributter .....</b>	<b>87</b>
Vis en liste over alle filerne i en mappe (og undermapper).....	87
Filattributter .....	89
Projektmappe attributter .....	90
Afslutningsvis.....	91
<b>Kapitel 11 Power Query – Importer data fra den aktuelle projektmappe .....</b>	<b>92</b>
Importer tabeller fra den aktuelle projektmappe .....	92
Tabeller eller områder .....	96

## Indhold

<b>Kapitel 12 Power Query – data fra internettet.....</b>	<b>97</b>
Grundlæggende internet import .....	97
Rediger forespørgslen.....	100
Brug af parametre med webforespørgsler .....	103
Brug brugerdefinerede formler .....	103
Oprettelse af funktionen.....	103
Brug af funktionen .....	105
Importerer flere websider .....	106
<b>Kapitel 13 Power Query – Fjern pivotering .....</b>	<b>108</b>
Grundlæggende Fjern pivotering af kolonner .....	108
Fjern pivotering af flere kolonner .....	111
Fjern pivotering med flere overskriftsrækker .....	113
<b>Kapitel 14 Power Query – udpakning af data i en kolonne .....</b>	<b>116</b>
Udpakning af en kolonne med data .....	116
Rens importen.....	117
Tilføj en indeks- og modulus kolonne .....	118
Efterbehandling.....	121
Hvad har vi lært?.....	121
<b>Kapitel 15 Power Query - Opslagsværdier ved hjælp af fletning .....</b>	<b>122</b>
Indlæs dataene i Power Query.....	123
Opslag med en nøjagtig match .....	125
Join typer .....	128
Yderforbindelser .....	128
Indre del.....	128
Anti sammenføjringer .....	128
Opslag med en omtrentlig match .....	129
Flere match .....	134
Opslag med flere kriterier .....	134
Fuzzy match opslag.....	135
<b>Kapitel 16 Power Query – Skift placering for kildedata .....</b>	<b>136</b>
Ret kildeplacering i forespørgslen .....	136
Brug indstillingerne for datakilde.....	137
Brug af parametre .....	138
<b>Kapitel 17 Power Query – formler .....</b>	<b>139</b>
Opret en formel .....	139
Simple formelfunktioner .....	140
Formeleksempel nr. 1 .....	141
Formeleksempel nr. 2 .....	141
Rækkekontekst .....	142

## Indhold

Datatyper .....	142
Formeleksempel nr. 3 .....	142
Find funktioner .....	143
IntelliSense .....	143
Se på eksisterende transformationstrin .....	144
Essentielle formeltips .....	145
M kode er følsom overfor store og små bogstaver .....	145
M begynder at tælle ved 0 .....	145
Øve, øve, øve .....	145
<b>Kapitel 18 Power Query – IF-udtryk til betinget logik .....</b>	<b>146</b>
Eksempel .....	146
IF-logik ved hjælp af en betinget kolonne .....	146
Scenarie 1 - Søndagsbonus .....	146
Scenarie 2 - Søndagsbonus og produktrabat .....	148
IF-logik ved at skrive M koden .....	148
Scenarie 1 - Søndagsbonus .....	149
Scenario 2 - Søndag bonus og produktrabat .....	150
Konklusion .....	151
<b>Kapitel 19 Power Query – Gruppering og aggregering af data .....</b>	<b>152</b>
Eksempel på gruppering og aggregering .....	152
Grundlæggende Grupper efter .....	152
Avanceret Grupper efter .....	153
Konklusion .....	155
<b>Kapitel 20 Power Query – Brugerdefinerede funktioner .....</b>	<b>156</b>
Hvad er brugerdefinerede funktioner? .....	156
En nem brugerdefineret funktion .....	156
Opret M koden igen .....	159
Angiv argumenterne .....	159
Flere argumenter og typer .....	160
Erklæring af inputtype .....	160
Erklæring af outputtype .....	160
Erklæring af flere argumenter .....	160
Brugerdefinerede funktioner der indeholder transformationer .....	160
Dupliker forespørgslen .....	161
Omdøb forespørgslen .....	161
Transformationer .....	161
Ændre forespørgslen til en funktion .....	162
Test funktionen .....	164
Brug funktionen .....	165



## Indhold

Næste skridt .....	166
Konklusioner .....	166
<b>Kapitel 21 Power Query – Almindelige fejl og hvordan man retter dem .....</b>	<b>167</b>
Fejl typer .....	167
Procesoprettelsesfejl .....	167
M kodefejl .....	167
Brugerdefinerede kolonner .....	167
Avanceret editor og formellinje .....	169
Formel.Firewall fejl .....	169
Ignorer personlige indstillinger .....	169
Fejl ved databehandling .....	170
Forkert kilde placering .....	171
Kolonnenavne .....	171
Datatyper .....	172
Softwarebugs .....	173
<b>Kapitel 22 Power Query – Tips og tricks .....</b>	<b>174</b>
Skift standardindstillingerne for Luk og indlæs .....	174
Giv trinene et meningsfuldt navn .....	175
Sørg altid for at formellinjen er synlig .....	175
Forhindre automatisk registrering af datatype .....	176
Brug Gå til kolonne .....	177
Opret mapper til gruppering af forespørgsler .....	177
Forespørgselsliste (i Power Query editoren) .....	178
Panelet Projektmappe-forespørgsler (i Excel) .....	178
Slet trin indtil slutningen .....	178
Opdel en forespørgsel i to .....	179
Kopier og indsæt forespørgsler i en ny projektmappe .....	180
Brug af kommentarer .....	180
Indsættelse af kommentarer i den Avancerede editor / Formellinjen .....	180
Indsættelse af kommentarer i trinegenskaberne .....	180
Tak for denne gang .....	181

# Introduktion

Min første introduktion til Excel sket for velover 25 år siden. Før da klarede jeg opgaverne ved hjælp af programmering og matematikprogrammet Maple. I takt med at jeg begyndte at undervise blev behovet i erhvervslivet for at undervisning i f.eks. Excel større og min karriere indenfor Excel var startet.

Det er for mange år siden gået op for mig at man aldrig bliver færdig med Excel. Der er altid nye emner man kan kaste sig over. Hver gang der kommer en ny tilføjelse til bogreolen, er der nye tips, tricks og sågar nye områder der kan udforskes. Indimellem får jeg også opfordringer og det er her Nils Egebjerg kommer ind i billedet. En dag spurgte han "Hvorfor har du ikke kurser i Power Query?" At undervise i et emne kræver noget undervisningsmateriale og der er en hel del om Power Query. Dog er det alt sammen på Engelsk! Intet på Dansk. Så efter en del research på internettet og i bøgerne på reolen var et udkast til en bog klar. Efter et års arbejde er den klar!

## Sådan bruges denne bog

Jeg foreslår du åbner den opgavefilen, der svarer til det eksempel, du arbejder med, og gennemføre hvert trin i eksemplet, mens du læser bogen. Du vil blive overrasket over, hvor let denne proces er, og forbløffet over, hvor meget du lærer og kan huske. Dette er den tilgang, jeg bruger i min undervisning.

## Downloads

Denne bog har en tilhørende hjemmeside, der stiller alle opgavefilerne til rådighed, som du kan bruge i bogens eksempler.

For at gennemgå eksemplerne i denne bog skal du kopiere bogens opgavefiler til din computer. Disse opgavefiler kan downloades fra bogens hjemmeside, der findes på:

<https://webbojo.dk/power-query-filer/>

## Anerkendelser

Jeg skylder en stor tak til Nils Egebjerg og Pauli Andersen. De har ofret mange søndagsformiddage sammen med mig på at arbejde sig igennem bogens kapitler. Der er blevet stillet et væld af spørgsmål og budt ind med problemstillinger fra den virkelige verden.

## Kapitel 1 Hvad er Power Query?

Power Query er et ETL-værktøj. ETL står for Extract, Transform og Load. Lad os se på hvert af disse ord for sig for at få en bedre forståelse.

- **Extract** - Data kan udvindes fra forskellige kilder; databaser, CSV-filer, tekstfiler, Excel projektmapper, specifikke celler i det samme regneark, websteder og endda nogle PDF-filer. Grundlæggende, hvis der er data, der er gemt et eller andet sted i et struktureret eller semistruktureret format, kan Power Query få adgang til det og udtrække det.
- **Transform** - Når dataene er blevet udtrukket i det forrige trin, kan de renses (dvs. fjerne mellemrum, opdele kolonner, ændre dato formater, udfylde tomme felter, finde og erstatte osv.) og omformes (dvs. fjern pivotering, fjerne kolonner osv.). Når data udtrækkes fra forskellige kilder, er det ikke sandsynligt, at de er konsistente, transformeringsprocessen bruges til at gøre dem klar til brug.
- **Load** - Når dataene er blevet udtrukket og transformeret, skal de placeres et sted, så du kan bruge dem. Fra et Excel perspektiv kan det indsættes i et regneark, en datamodel eller en anden forespørgsel.

For at opsummere tager Power Query data fra forskellige kilder og forvandler dem til noget, der kan bruges.

Som værktøj er dette allerede ret nyttigt. Men her er den bedste del. Når ETL-processen er oprettet, kan den afvikles igen og igen med et enkelt klik. Hvilket kan spare arbejdstid hver uge.

## Nogle eksempler på, hvad Power Query kan gøre

---

Lyder det hele lidt abstrakt og forvirrende indtil videre? Lad mig vise dig nogle eksempler for at give dig en reel ide om hvad der er muligt.

### Eksempel 1:

Lad os sige at du hver dag modtager en CSV-fil med en opdateret prislister. Normalt ville du åbne CSV filen, kopiere de relevante kolonner i dit regneark og derefter bruge en masse VENSTRE og HØJRE formler til at opdele nogle tekstfelter. Til sidst bruger du en pivottabel til at præsentere informationen i det rigtige format.

Du kan gøre alt dette med Power Query. Gem blot CSV filen det rigtige sted, og opdater derefter alt med et enkelt klik.

### Eksempel 2:

Eller hvad med dette scenarie; Hvis du hver måned modtager 30 Excel projektmapper fra forskellige afdelinger i virksomheden, alle projektmapper er i samme format. Du åbner hvert regneark og kopierer dataene til et hovedregneark for at konsolidere til en hovedrapport.

Med Power Query kan du konsolidere alle data på én gang med en enkelt opdatering. Du behøver ikke at åbne hvert regneark, du behøver ikke at kopiere og indsætte. Gem blot alle filerne i den samme mappe, og opdater derefter konsolideringen med et enkelt klik.

### Eksempel 3:

Din virksomhed har netop erhvervet en ny virksomhed. Som det ofte er tilfældet, er deres softwaresystem anderledes end resten af virksomheden. Ledelsen har i øjeblikket ingen intentioner om at ændre it-systemer. Dit job er at oprette et opsummeret dashboard for hele virksomheden hver uge. Det lyder som om det kunne være en masse arbejde.

Med Power Query kan du muligvis hente data direkte fra hvert system og ved hjælp af en kortlægningstabel opdatere dit dashboard med en enkelt opdatering.

## Hvad er forskellen mellem Power Query og Get & Transform

---

Gennemsnits brugeren benytter ikke ordet "Query" særlig ofte til at beskrive de opgaver de udfører. Sæt ordet "Power" foran, og det bliver ikke klarere. Måske er det derfor, Microsoft ændrede navnet til Get & Transform<sup>1</sup> i Excel 2016 og placerede det i båndet under fanen Data. Jeg er sikker på, at de forsøgte at gøre det mere klart for brugeren. Jeg er også sikker på, at hvis du bad de fleste om at gætte på, hvad værktøjet gjorde, bare baseret på navnet, vil mange give det rigtige svar. Selvom det officielt kaldes Get and Transform i Excel 2016 og senere, ser navnet Power Query ud til at hænge fast i Excel samfundet.

## Hvor vanskeligt er Power Query at lære?

---

Hvis du tror du skal være programmør? Eller i det mindste en Excel ekspert? Så tager du fejl. Power Query har et brugervenligt interface, der er designet til almindelige brugere.

Der er lidt af en læringskurve, men det meste består i at lære, hvad de enkelte knapper gør.

Hvis du vil gå dybden med Power Query, er der et programmeringssprog kaldet "M", som du kan lære. Men du kan udnytte 99,9% af Power Query uden overhovedet at skulle bruge det.

## Hvordan installeres Power Query?

På grund af udviklingen af Power Query i Excel i de senere år betyder det, at der er flere centrale spørgsmål:

- Behøver du at downloade Power Query?
- Hvilken version af Power Query skal du downloade?
- Er din version af Excel kompatibel med Power Query?

## Tid til at komme i gang

---

I en undersøgelse blandt Power Query brugere viste det sig at den gennemsnitlige tidsbesparelse var på 22% Det er meget! Mere end en arbejdsdag om ugen med irriterende gentaget arbejde er elimineret for hver bruger af Power Query.

Med denne statistik i tankerne, så lad os komme i gang med at lære Power Query, så du selv kan begynde at spare tid.

---

<sup>1</sup> På dansk benyttes også udtrykket forespørgsler/forbindelser.

## Kapitel 2 Importer data

OK, lad os komme i gang med Power Query. Det første, vi skal gøre, er at importere nogle data. Da dette er vores første rigtige arbejde med Power Query, vil vi bruge dette som en mulighed for at udforske brugergrænsefladen.

Dette kapitel er opdelt i to:

- Brug af data i den samme projektmappe.
- Brug af data fra eksterne filer og eksterne projektmapper.

Når vi arbejder os gennem eksemplerne, håber jeg, at du vil se, at det er let at få data importeret til Power Query.

Bemærk, at Excel og Power Query konstant opdateres af Microsoft. Afhængigt af din specifikke version af Excel, kan navnene, placeringerne og indstillingerne for forskellige handlinger afvige lidt fra eksemplerne, der er vist i dette kapitel.

### Brug af data i den samme projektmappe

For at bruge data fra den samme projektmappe skal Power Query kende det område, som kildedataene befinder sig i.

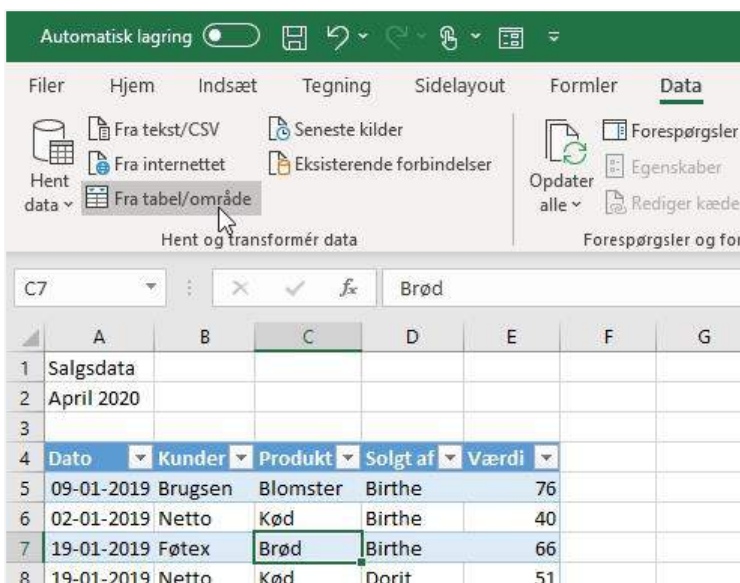
Der er to formater af dataområder som Power Query håndterer godt, Excel tabeller og navngivne områder. Eksemplerne nedenfor viser, hvordan man bruger begge disse metoder.

#### Importer data fra tabeller

For det første er det vigtigt at bemærke, at jeg bruger "Tabeller" til at henvise specifikt til Excel Tabel funktionen. Tabeller er et vigtigt element i, hvordan Excel kan gemme data. Outputtet fra Power Query vil typisk være i et tabelformat. Så tabeller er grundlaget for en fornuftig brug af Power Query. Hvis du ikke er bekendt med tabeller, anbefaler jeg, at du bruger lidt tid på at lære dem og bruge dem.

I det følgende eksempel arbejder vi med filen **Tablel.xlsx**.

Marker en hvilken som helst celle i tabellen. Klik derefter på **Data → Fra tabel/område**.



Power Query vinduet åbnes. Da dette er vores første kig på brugergrænsefladen, bruger vi lidt tid på at forstå, hvad du ser.

## Kapitel 2 Power Query – Importer data

	Dato	Kunder	Produkt	Solgt af	Værdi
1	09-01-2019 00:00:00	Brugsen	Blomster	Birthe	76
2	02-01-2019 00:00:00	Netto	Kød	Birthe	40
3	19-01-2019 00:00:00	Føtex	Brød	Birthe	66
4	19-01-2019 00:00:00	Netto	Kød	Dorit	51
5	12-01-2019 00:00:00	Netto	Blomster	Dorit	75
6	22-01-2019 00:00:00	Brugsen	Blomster	Mogens	46
7	23-01-2019 00:00:00	Bilka	Kød	Lene	40
8	25-01-2019 00:00:00	Bilka	Frugt	Dorit	57
9	23-01-2019 00:00:00	Rema 1000	Frugt	Jørgen	50
10	10-01-2019 00:00:00	Lidl	Frugt	Birthe	75
11	06-01-2019 00:00:00	Spar	Kød	Lene	49
12	10-01-2019 00:00:00	Meny	Frugt	Dorit	60
13	11-01-2019 00:00:00	Spar	Brød	Lene	38
14	31-01-2019 00:00:00	Bilka	Blomster	Lene	43
15	06-01-2019 00:00:00	Spar	Kød	Jørgen	45
16	18-01-2019 00:00:00	Føtex	Kød	Birthe	64
17	20-01-2019 00:00:00	Lidl	Grøntsager	Jørgen	57
18	05-01-2019 00:00:00	Bilka	Kød	Dorit	38
19	29-01-2019 00:00:00	Lidl	Grøntsager	Lene	49
20	13-01-2019 00:00:00	Meny	Grøntsager	Dorit	73
21	16-01-2019 00:00:00	Spar	Grøntsager	Birthe	55

**Båndet** - Ligesom med Excel indeholder båndet de vigtigste kommandoer grupperet i separate faner. Vi bruger båndet i denne bog, så du får vist de mest almindelige kommandoer i aktion.

**Egenskaber** - Det er her vi navngiver forespørgslen. Det er vigtigt at give din forespørgsel et meningsfuldt navn, så du ved, hvad forespørgslen gør uden at skulle se på trinnene.

**Anvendte trin** - Ligesom med makrooptageren, vil hver handling, du foretager, blive optaget i de anvendte trin. Det er muligt at tilføje, fjerne og redigere trin.

**Forhåndsvisning af data** - Dette område viser en forhåndsvisning af vores data, baseret på det anvendte trin, der er markeret. Ved at højreklikke på kolonneoverskrifterne kan vi få adgang til mange af datatransformationsværktøjerne som også findes på båndet.

### Forstå de anvendte trin

Vinduet Anvendte trin indeholder allerede en lille liste over emner. Disse trin er Excels forsøg på automatisk at transformere dataene baseret på hvad den mener, vi vil gøre.

**Kilde** – Identificerer kildedataene (f.eks. Tabellen i regnearket)

**Ændret type** – Excel har analyseret dataene og forsøgt at anvende den rigtige datatype på hver kolonne.

	Dato	Kunder	Produkt	Solgt af	Værdi
1	09-01-2019 00:00:00	Brugsen	Blomster	Birthe	76
2	02-01-2019 00:00:00	Netto	Kød	Birthe	40
3	19-01-2019 00:00:00	Føtex	Brød	Birthe	66
4	19-01-2019 00:00:00	Netto	Kød	Dorit	51
5	12-01-2019 00:00:00	Netto	Blomster	Dorit	75

Disse trin er blevet anvendt automatisk af Excel. Vær opmærksom på at de automatiske trin er ikke altid e rigtige, så du bliver muligvis nødt til at lave dem igen. Jeg har fundet ud af at Ændret type ofte kan forårsage frustrerende fejl senere, som er utroligt vanskelige at finde. Nogle gange er det lettere at slette trinnet Ændret type og lave det manuelt.

## Kapitel 2 Power Query – Importer data

Når vi foretager flere handlinger for at transformere dataene, føjes disse trin til listen over anvendte trin.

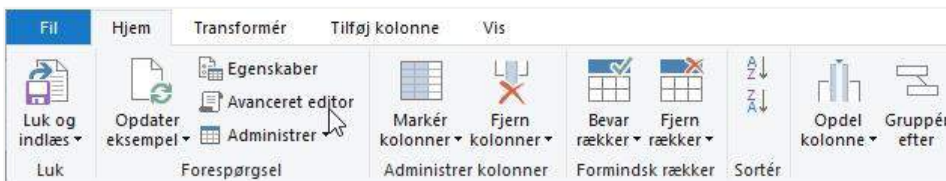
Klik på hvert trin for at se de datatransformationer, der anvendes én for én.

### M kode

M kode er det sprog, som Power Query bruger til at registrere de anvendte trin.

Lad os tage et kort kig på M koden for at se, hvordan Power Query har registreret trinnene. Bare rolig, vi vil ikke bruge meget tid på at se på M kode, men det vil hjælpe med at forstå, hvordan Power Query fungerer, og hvordan man kan foretage mere avancerede redigeringer af forespørgsler på et senere tidspunkt.

Klik på: **Hjem** → **Avanceret editor**.



Dialogboksen Avanceret editor vises.



Lad os kort se på koden og sammenligne den med de anvendte trin, vi allerede har.

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

Linjen ovenfor identificerer hvilken Excel tabel, der skal bruges som kilde. Hvis vi ville ændre forespørgslen for at se på en anden tabel, kunne vi ændre tblSalesData til navnet på den nye tabel.

```
#"Ændret type" = Table.TransformColumnTypes(Kilde,{{"Dato", type datetime}, {"Kunder", type text}, {"Produkt", type text}, {"Solgt af", type text}, {"Værdi", Int64.Type}})
```

Linjen ovenfor ændrer datatyperne. For eksempel er kolonnen "Dato" ændret til en dato/klokkeslæt. Denne ændring er baseret på Power Querys gennemgang af dataene i kolonnen.

## Kapitel 2 Power Query – Importer data

Luk den avancerede editor, det er nok M kode for i dag.

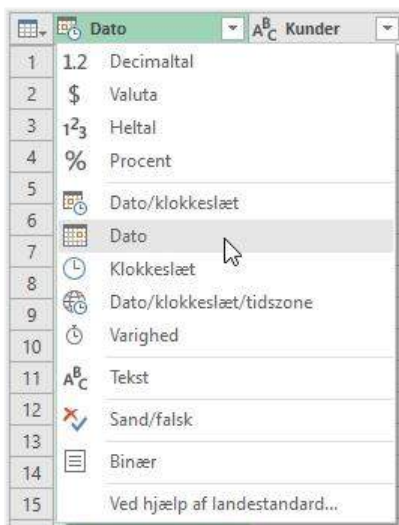
### Tilføjelse af trin

Nu vil vi gennemføre nogle grundlæggende transformationer og se, hvordan de føjes til listen over anvendte trin.

Dobbelklik på kolonnen "Værdi", og ret navnet til "Beløb".

	Dato	Kunder	Produkt	Solgt af	Værdi	Beløb
1	09-01-2019 00:00:00	Brugsen	Blomster	Birthe		76
2	02-01-2019 00:00:00	Netto	Kød	Birthe		40

Klik på ikonet Dato/Klokkeslæt og ændre det til Dato.



Kolonnen Dato viser nu kun datoen og ikke datoen og klokkeslættet.

Se nu på vinduet Anvendte trin, der er tilføjet to nye trin til de handlinger, vi har udført.



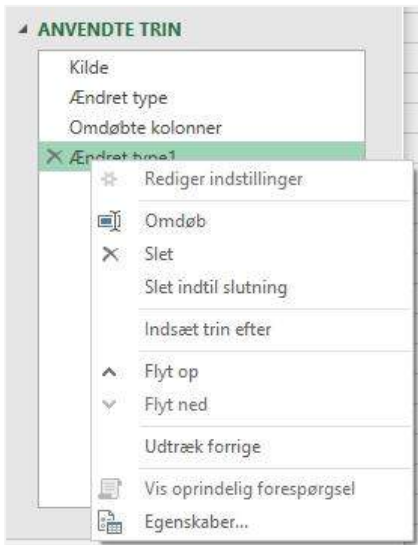
### Redigering af trinnene

For at slette et trin kan vi klikke på X symbolet ved siden af trinnet.

For at se flere ændringer kan du højre klikke for at se alle indstillingerne.



## Kapitel 2 Power Query – Importer data



Disse indstillinger er generelt selvforklarende. Hvis der er nogen, du ikke forstår, så prøv dem bare på noget prøvedata ... hvad er det værste, der kan ske!

Omdøb er sandsynligvis den mest nyttige kommando her og det er den der bliver brugt mindst. Når der oprettes flere trin, kan det være vanskeligt at huske, hvad hvert trin gør. Det er god praksis at omdøbe trin, når forespørgsler strækker sig over mere end et par trin (se skærmbillede nedenfor).



### Indlæs dataene i Excel

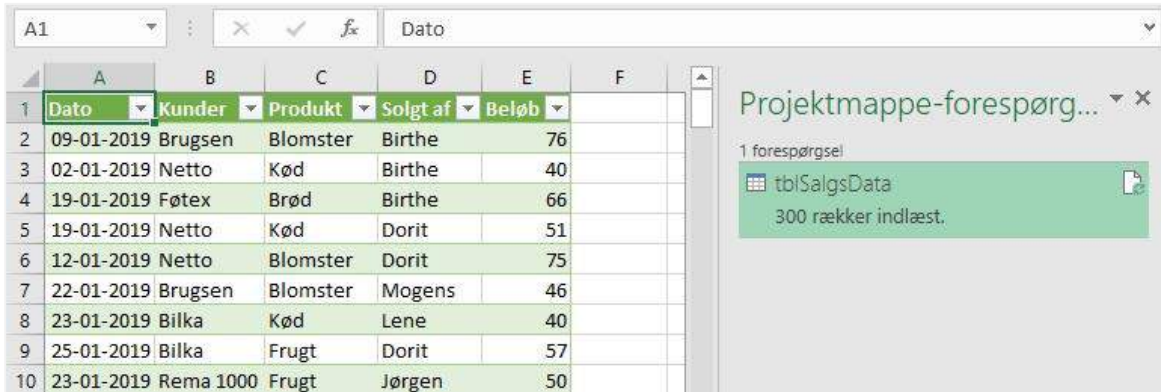
For at afslutte processen indlæser vi dataene i Excel projektmappen.

Klik på **Hjem** → **Luk og indlæs**.



Et nyt regneark oprettes med en tabel, der viser de transformererede data.

## Kapitel 2 Power Query – Importer data



Dato	Kunder	Produkt	Solgt af	Beløb
09-01-2019	Brugsen	Blomster	Birthe	76
02-01-2019	Netto	Kød	Birthe	40
19-01-2019	Føtex	Brød	Birthe	66
19-01-2019	Netto	Kød	Dorit	51
12-01-2019	Netto	Blomster	Dorit	75
22-01-2019	Brugsen	Blomster	Mogens	46
23-01-2019	Bilka	Kød	Lene	40
25-01-2019	Bilka	Frugt	Dorit	57
23-01-2019	Rema 1000	Frugt	Jørgen	50

Du er sandsynligvis ikke blæst bagover endnu. Alt, hvad vi har gjort, er at indlæse en tabel fra Excel i Power Query, foretaget et par mindre ændringer og derefter indlæse disse data i Excel. Husk, at dette er et simpelt eksempel for at vise dig de grundlæggende værktøjsoperationer. Når vi bevæger os gennem kapitlerne, vil vi forvandle disse enkle trin til noget fabelagtigt....

### Importerer data fra standard områder

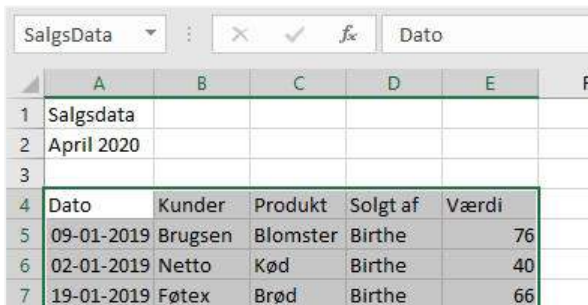
Hvis du markerer et celleområde (f.eks. cellerne B5:F55) snarere end en tabel, vil Excel automatisk ændre disse celler til en tabel, før de importeres. Alt andet efter dette punkt er identisk med ovenstående trin.

### Importerer data fra et navngivet område

Der er en anden måde at fortælle Power Query, hvor mange celler der skal importeres; og det er ved at bruge et navngivet område.

I det følgende eksempel arbejder vi med filen **Navngivenområde.xlsx**.

Den nemmeste måde at oprette et navngivet område er at **markere alle celler, skriv derefter navnet i Navnefeltet og tryk på Enter**.



Dato	Kunder	Produkt	Solgt af	Værdi
09-01-2019	Brugsen	Blomster	Birthe	76
02-01-2019	Netto	Kød	Birthe	40
19-01-2019	Føtex	Brød	Birthe	66

I eksempel filen er der allerede oprettet et navngivet område, kaldet SalgsData.

Lad os importere dataene i det navngivet område til Power Query. Trinene ligner meget det at importere data fra en tabel (derfor gentager jeg ikke alle oplysningerne ovenfra, men jeg vil fremhæve forskellene).

Marker det navngivne område fra listefeltet Navnboks.



Dato	Kunder	Produkt	Solgt af	Værdi
09-01-2019	Brugsen	Blomster	Birthe	76
02-01-2019	Netto	Kød	Birthe	40
19-01-2019	Føtex	Brød	Birthe	66

Celleområdet markeres i regnearket.

Vælg **Data** → **Fra tabel/område**.

## Kapitel 2 Power Query – Importer data

De importerede data vil se ud på samme måde som når du bruger en tabel. Men der kan være nogle ekstra trin, der anvendes.



I dette eksempel er der tilføjet et ekstra trin Hævede overskrifter – vi vil se på dette senere i dette kapitel. Klik gennem de anvendte trin for at se, hvilke ændringer der er blevet anvendt.

### Tandhjulsikonet

Har du lagt mærke til det lille tandhjulsikon ved siden af trinnet Hævede overskrifter? Dette er for at indikere, hvor Power Query giver os mulighed for at ændre indstillinger.

Klik på tandhjulsikon.



Dialogboksen Brug første række som overskrifter åbnes. Vinduet, der åbnes, vil variere afhængigt af trintypen.

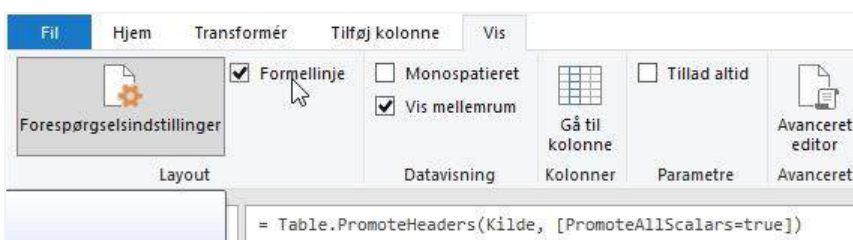


I vores eksempel ønsker vi stadig indstillingerne Fremhæv alle skalar typer, så vi kan klikke på OK uden at foretage nogen ændringer.

### Vis formellinjen

I stedet for at se på alle M koder i den avancerede editor, som vi gjorde før, kan vi vise formellinjen.

Klik på: **Vis** → **Formellinjen**.



Når vi klikker gennem trinnene, ændres formellinjen for at vise den M kode, der bruges til hvert trin. Du behøver ikke at ændre dette, men vær opmærksom på, at den er der til fremtidig brug.

### Er det bedst at bruge navngivne områder eller tabeller?

Efter at have set på de to muligheder, hvilken er så bedst? Der er ikke ligetil at svare på, da det afhænger af dine kildedata.

#### Tabeller

- Kan automatisk udvides
- Skal være i et struktureret dataformat
- Skal have en defineret overskriftsrække
- Kan ikke have beregninger i overskriftsrækken

#### Navngivne områder

- Udvider sig som standard ikke automatisk
- Kan indeholde et vilkårligt antal overskriftsrækker (vil dog kræve flere transformationer i Power Query)
- Kan have beregninger i overskriftsrækken

Dynamiske navngivne områder kan også importeres til Power Query, men det er for avanceret til denne introduktionsbog.

Mit råd er altid at vælge en tabel, medmindre der er en bestemt grund til ikke at gøre det.

## Brug af data fra eksterne filer og projektmapper

---

Når vi har kigget på data, der er gemt i den samme projektmappe som forespørgslen, skal vi vende opmærksom til data, der er gemt uden for projektmappen.

Tit åbner vi CSV-filer, tekstfiler eller Excel projektmapper og indsætter dataene ind i en anden projektmappe. Ved at importere data direkte til projektmapper ved hjælp af Power Query, betyder det, at vi aldrig behøver at kopiere og indsætte igen. Linket til kildedataene opretholdes, og det kan opdateres med et klik på en knap (mere om det i et senere kapitel).

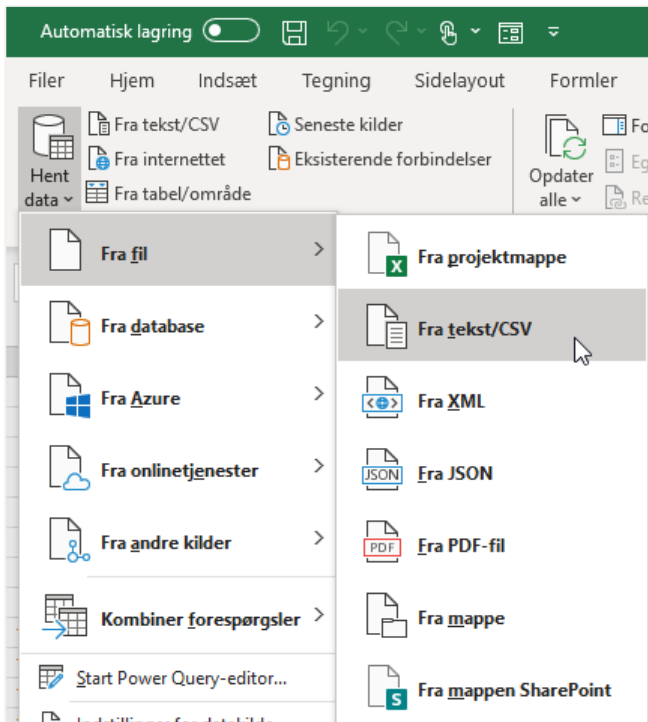
### CSV-fil

CSV er et meget almindeligt filformat til eksport fra andre systemer. Den gode nyhed er, at Power Query elsker CSV filer.

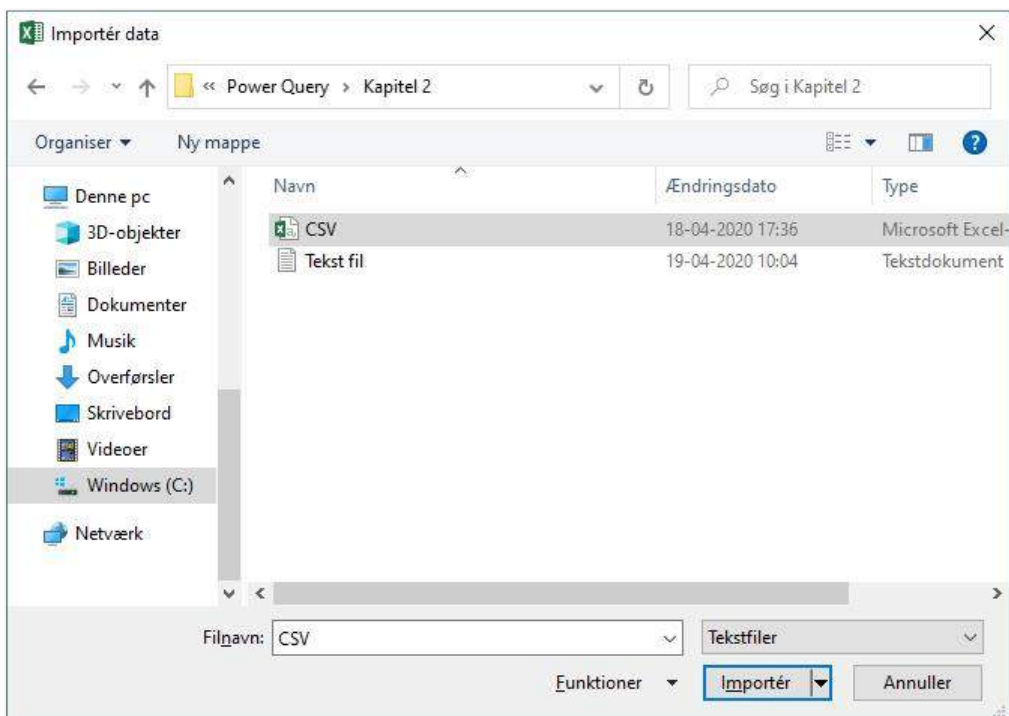
I det følgende eksempel arbejder vi med filen **CSV.csv**.

Vælg **Data** → **Hent data** → **Fra fil** → **Fra tekst/CSV**.

## Kapitel 2 Power Query – Importer data

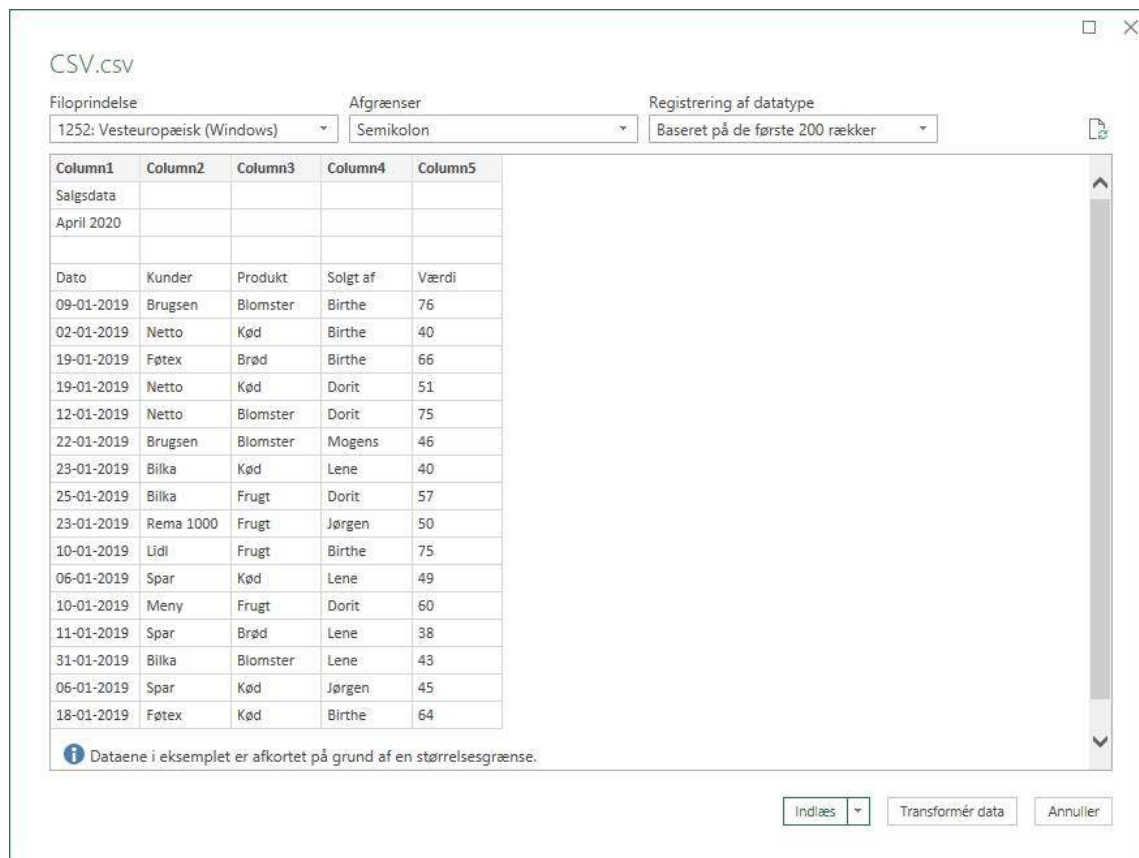


Dialogboksen Importer data åbnes. Naviger frem til CSV filen, marker den, og klik på Importer.



Power Query åbner et nyt vindue og viser en prøve af dataene.

## Kapitel 2 Power Query – Importer data



CSV-filer har pr. definition en semikolonafrænsler, så der skulle ikke være behov for at ændre indstillingerne.

Hvis du vil indlæse dataene direkte i Excel uden ændringer, kan du klikke på Indlæs, eller for at transformere dataene, klikkes på Transformer data. Alle trin fra nu af er de samme, som vi allerede har set i ovenstående afsnit.

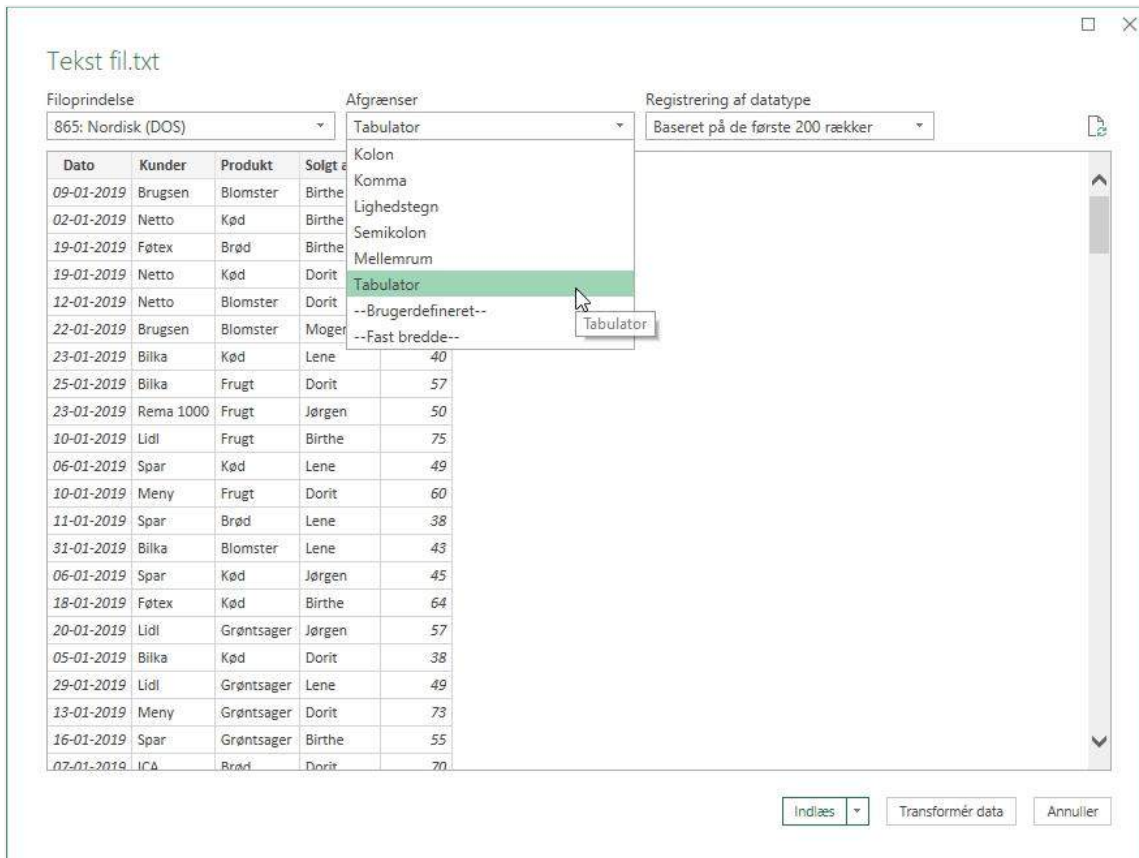
### Tekstfil

Tekstfiler ligner CSV, men kan have en række dataformater. Lad os se på en tabulatorsepareret tekstfil for at vise, hvordan den adskiller sig.

I det følgende eksempel arbejder vi med filen **Text.txt**.

Følg de samme trin som for en CSV-fil. I stedet for at vælge komma som afgrænsler, skal du vælge tabulator.

## Kapitel 2 Power Query – Importer data



Hvis de data, du importerer, er i et andet tekstformat, skal du vælge en passende indstilling i listefeltet.

Klik derefter på enten Indlæs eller Transformer data efter behov.

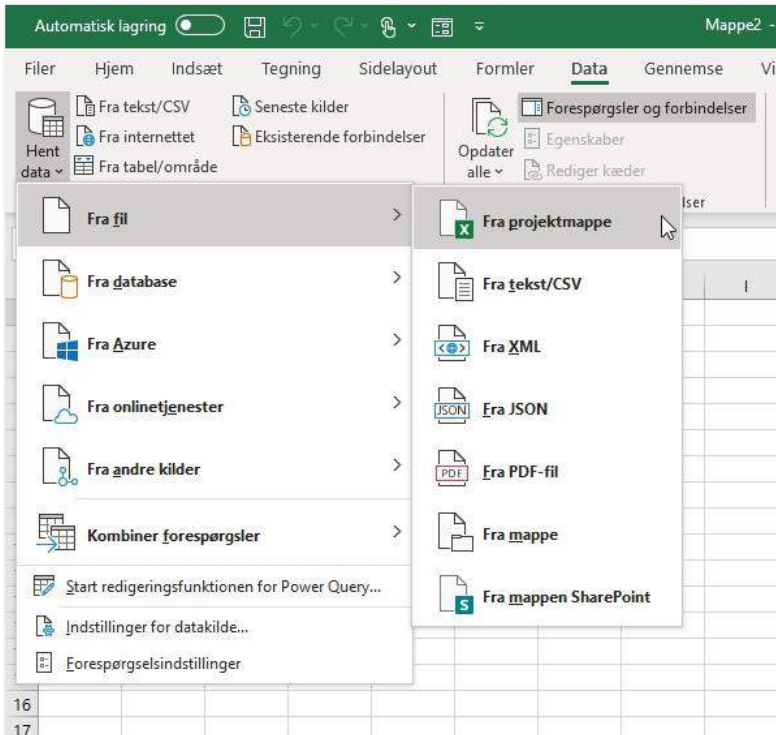
### Projektmappe

Til sidst vil vi i dette kapitel importere indholdet af en Excel fil.

Følgende eksempel bruger **Projektmappe.xlsx**.

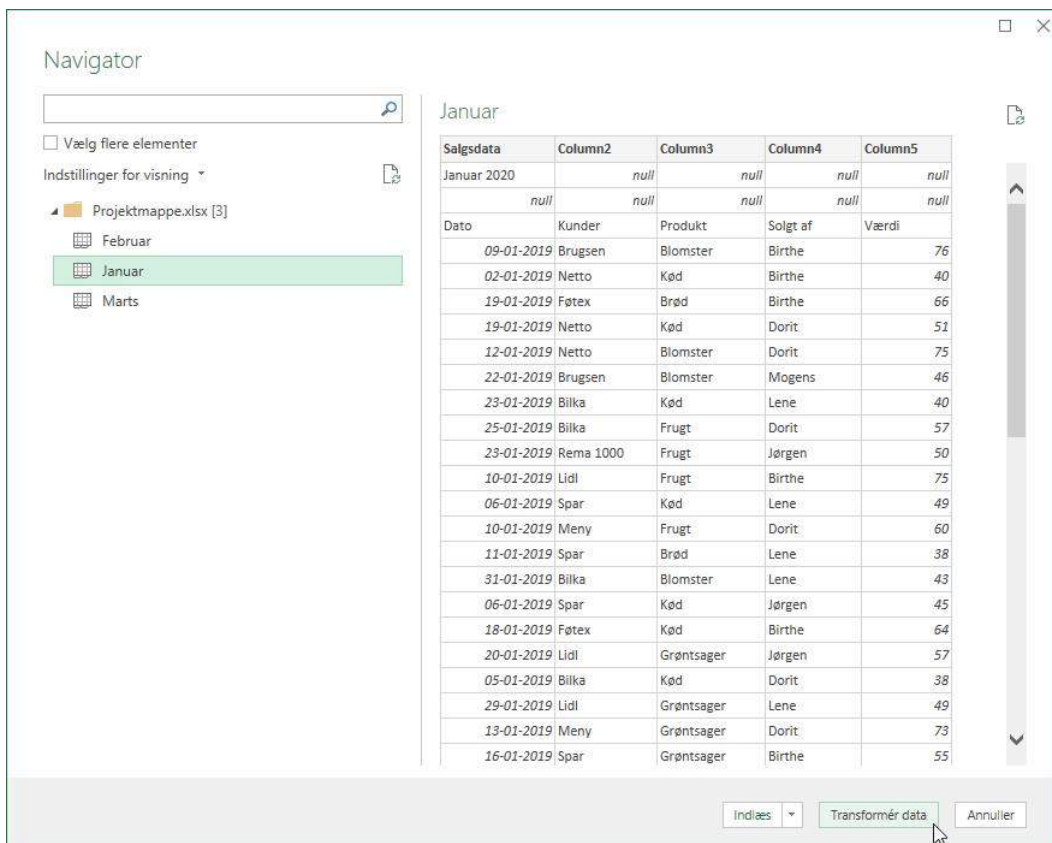
Vælg **Data** → **Hent data** → **Fra fil** → **Fra projektmappe**.

## Kapitel 2 Power Query – Importer data



Vælg Excel projektmappen i dialogboksen Importer data, og klik på Importer.

Power Query Navigator dialogboksen åbnes. I dette eksempel skal du blot vælge et enkelt regneark (for eksempel Januar som vist på skærbilledet nedenfor) og klikke på Rediger. Vi vil se på det at kombinere flere filer i et senere kapitel.





## Kapitel 2 Power Query – Importer data

Vi præsenteres nu for et scenarie, vi ikke har stødt på før. Men det er et scenarie, der kan forekomme i mange tilfælde.... Excel ved ikke, hvilken række der indeholder overskrifterne. Se på skærmbilledet nedenfor, Excel har antaget at række 1 i regnearket indeholder overskrifterne, men ifølge de foregående eksempler skal overskrifterne være Dato, Kunder, Produkt, Solgt af og Værdi, som er i række 3.

ABC 123	Salgsdata	A <sup>B</sup> C Column2	A <sup>B</sup> C Column3	A <sup>B</sup> C Column4	ABC 123	Column5
1	Januar 2020		null	null	null	null
2		null	null	null	null	null
3	Dato	Kunder	Produkt	Solgt af		Værdi
4	09-01-2019	Brugsen	Blomster	Birthe		76
5	02-01-2019	Netto	Kød	Birthe		40
6	19-01-2019	Føtex	Brød	Birthe		66
7	19-01-2019	Netto	Kød	Dorit		51
8	12-01-2019	Netto	Blomster	Dorit		75

Men bare rolig, vi kan let løse dette med nogle enkle transformationer.

Lad os fjerne de tre øverste rækker. Klik på: Hjem → Fjern rækker → Fjern øverste rækker.

ABC 123	Salgsdata	A <sup>B</sup> C Column2	A <sup>B</sup> C Column3	A <sup>B</sup> C Column4	ABC 123	Column5
1	Januar 2020					
2		null				
3	Dato	Kunder	Produkt	Solgt af		Værdi
4	09-01-2019	Brugsen	Blomster	Birthe		76
5	02-01-2019	Netto	Kød	Birthe		40
6	19-01-2019	Føtex	Brød	Birthe		66
7	19-01-2019	Netto	Kød	Dorit		51

Dialogboksen Fjern øverste rækker åbnes. I vores eksempel ønsker vi at fjerne 2 rækker, og klik derefter på OK.

Fjern de øverste rækker

Angiv, hvor mange rækker du vil fjerne fra toppen.

Antal rækker

Dernæst kan vi fremhæve den nye øverste række til at være overskriftsrække. Klik på: Hjem → Brug den første række som overskrifter.

## Kapitel 2 Power Query – Importer data

The screenshot shows the Power Query editor interface. The title bar reads "Januar - Power Query-editor". The ribbon includes tabs for "Fil", "Hjem", "Transformér", "Tilføj kolonne", and "Vis". The "Hjem" tab is active, showing options like "Luk og indlæs", "Opdater eksempel", "Egenskaber", "Avanceret editor", "Administrer", "Mærkér kolonner", "Fjern kolonner", "Bevar rækker", "Fjern rækker", "Sortér", "Opdel kolonne", "Grupper", and "Datatype: Enhver". The formula bar contains the expression: `= Table.Skip("#Ændret type",2)`. Below the formula bar, a table is displayed with the following data:

	ABC 123 Salgsdata	A <sup>B</sup> C Column2	A <sup>B</sup> C Column3	A <sup>B</sup> C Column4	ABC 123 Column5
1	Dato	Kunder	Produkt	Solgt af	Værdi
2	09-01-2019	Brugsen	Blomster	Birthe	76
3	02-01-2019	Netto	Kød	Birthe	40
4	19-01-2019	Føtex	Brød	Birthe	66
5	19-01-2019	Netto	Kød	Dorit	51

Det er det. Vi har nu det samme dataformat som i de foregående eksempler.

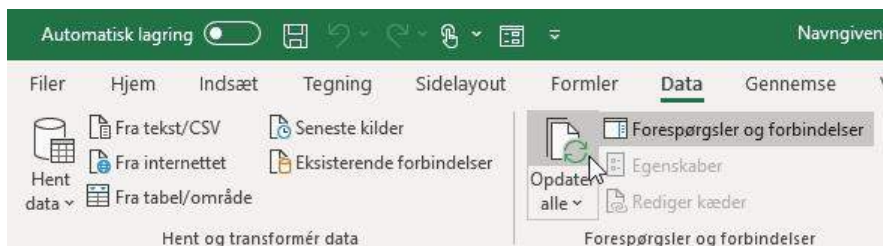
## Kapitel 3 Opdatering af data

I forrige kapitel kiggede vi på, hvordan vi importerer data fra forskellige filformater og indlæser dem i Excel. I dette kapitel går vi videre og ser på, hvordan vi kan opdatere dataene, hvilket gør det muligt for os at opbygge en forespørgsel én gang og bruge den igen og igen.

### Opdater alle data

I modsætning til Excels beregning, der som standard genberegner ved hver ændring, genberegner Power Query kun, med en specifik kommando.

Opdateringsprocessen er enkel, klik på **Data** → **Opdater alle**.

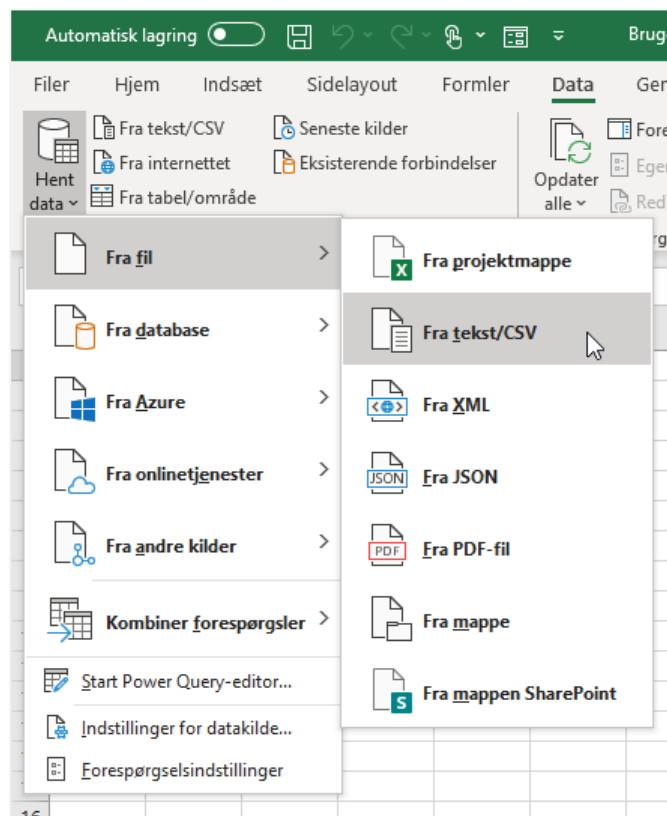


Lad os se denne opdateringsproces i funktion.

I dette kapitel arbejder vi med filen **Opdater data 1.csv**.

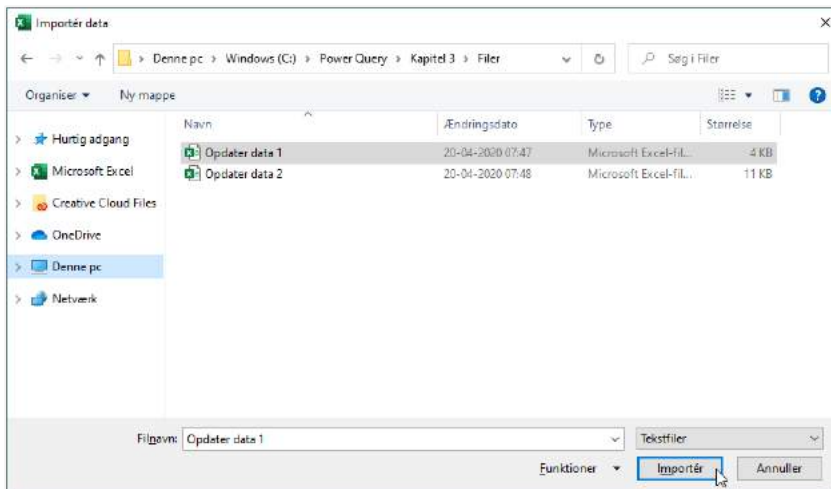
### Opret en standard forespørgsel

Åbn en ny projektmappe og opret en ny forespørgsel ved hjælp af en CSV-fil; klik på **Data** → **Hent data** → **Fra tekst/CSV**.

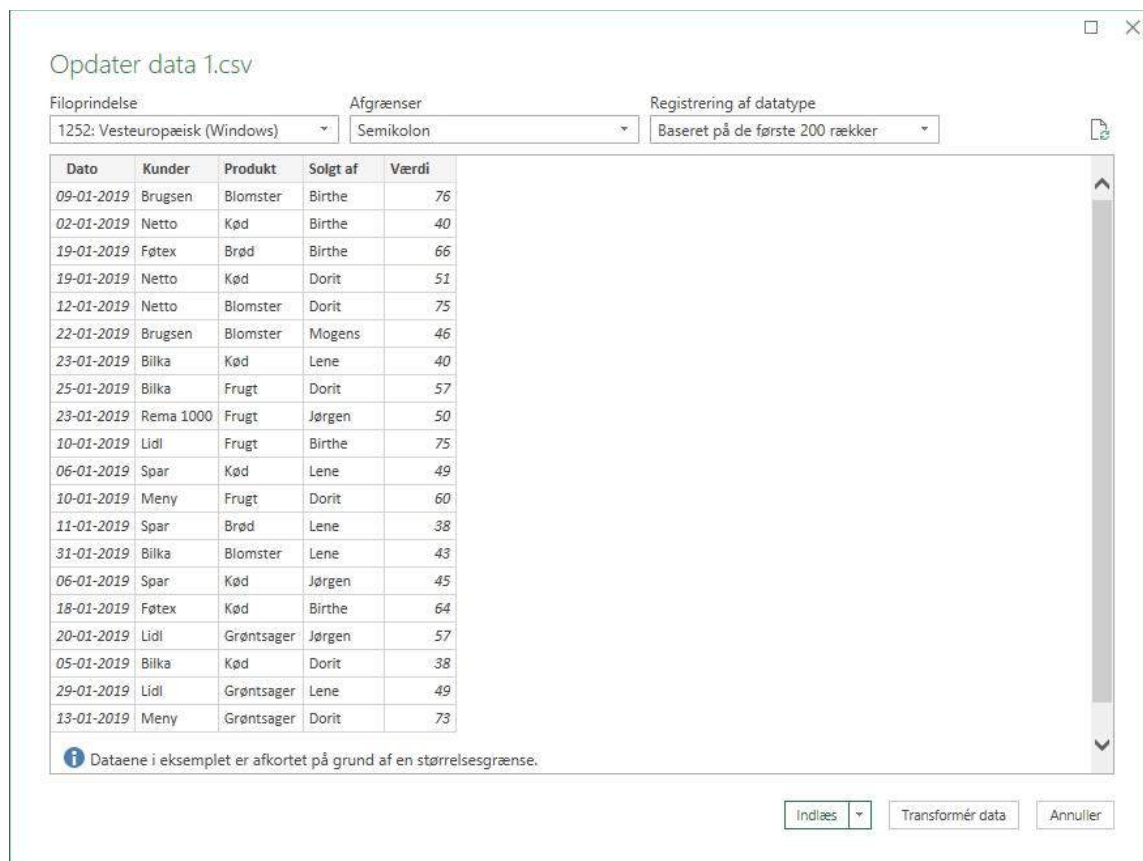


## Kapitel 3 Power Query – Opdatering af data

Dialogboksen **Importer data** åbnes. Naviger frem til filen **Opdater data 1.csv**, vælg den, og klik på **Importer**.



Power Query åbner en ny dialogboks og viser en prøve på dataene. Klik på **Transformer data**.



Vi vil nu foretage nogle grundlæggende datatransformationer for at omforme CSV filen til mere nyttig information. Det er første gang i denne bog, vi udfører nogle af disse transformationer, men bekymre dig ikke for meget om dem nu, vi vil se dem igen i senere kapitler.

Klik på kolonnen **Dato**. Klik på **Transformerer** → **Dato** → **Måned** → **Månedens slutning** via båndet (dette ændrer kolonnen **Dato** til den sidste dag i kalendermåneden).

## Kapitel 3 Power Query – Opdatering af data

The screenshot shows the Power Query Editor interface. The 'Transform' ribbon is active, and the 'Data' dropdown menu is open, showing options for date formatting. The 'Månedens slutning' option is selected. The data table below shows a list of transactions with columns for Date, Customer, Product, Sold by, and Sales. The formula bar shows the transformation: `= Table.TransformColumnTypes(#"Hævede overskrifter",{{"Dato", type date}}`

	Dato	Kunder	Produkt	Solgt af	
1	09-01-2019	Brugsen	Blomster	Birthe	
2	02-01-2019	Netto	Kød	Birthe	
3	19-01-2019	Føtex	Brød	Birthe	
4	19-01-2019	Netto	Kød	Dorit	
5	12-01-2019	Netto	Blomster	Dorit	75
6	22-01-2019	Brugsen	Blomster	Mogens	46
7	23-01-2019	Bilka	Kød	Lene	40
8	25-01-2019	Bilka	Frugt	Dorit	57
9	23-01-2019	Rema 1000	Frugt	Jørgen	50
10	10-01-2019	Lidl	Frugt	Birthe	75
11	06-01-2019	Spar	Kød	Lene	49
12	10-01-2019	Meny	Frugt	Dorit	60
13	11-01-2019	Spar	Brød	Lene	38
14	31-01-2019	Bilka	Blomster	Lene	43
15	06-01-2019	Spar	Kød	Jørgen	45
16	18-01-2019	Føtex	Kød	Birthe	64
17	20-01-2019	Lidl	Grøntsager	Jørgen	57

Klik på kolonneoverskriften Produkt, **hold derefter Shift tasten nede**, og klik på kolonneoverskriften Solgt af. Begge kolonner er nu markeret.

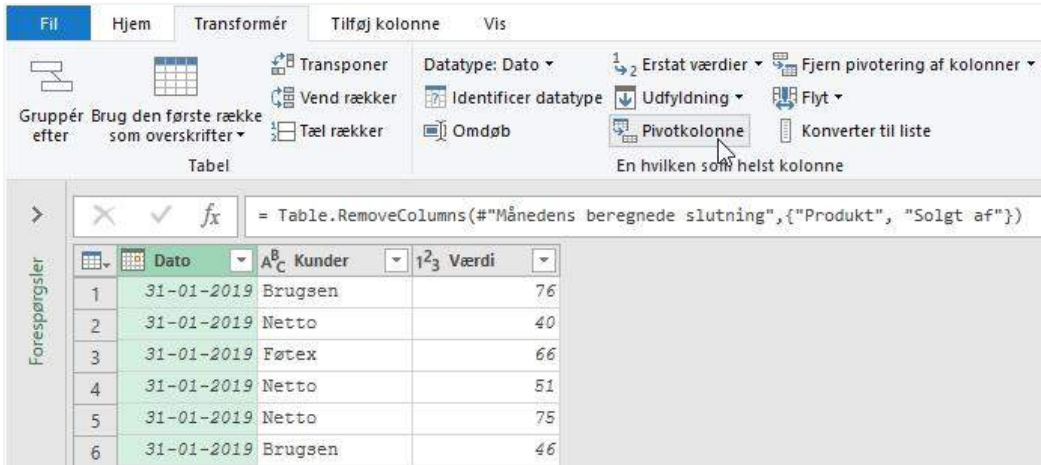
På båndet skal du klikke på **Hjem** → **Fjern kolonner** (dette fjerner de valgte kolonner).

The screenshot shows the Power Query Editor interface. The 'Home' ribbon is active, and the 'Fjern kolonner' option is selected. The data table below shows the same list of transactions, but with the 'Produkt' and 'Solgt af' columns highlighted in green. The formula bar shows the transformation: `= Table.TransformColumns(#"Ændret type",{{"Dato", type date, Date.EndOfMonth, type date}})`

	Dato	Kunder	Produkt	Solgt af	Værdi
1	31-01-2019	Brugsen	Blomster	Birthe	76
2	31-01-2019	Netto	Kød	Birthe	40
3	31-01-2019	Føtex	Brød	Birthe	66
4	31-01-2019	Netto	Kød	Dorit	51
5	31-01-2019	Netto	Blomster	Dorit	75
6	31-01-2019	Brugsen	Blomster	Mogens	46
7	31-01-2019	Bilka	Kød	Lene	40
8	31-01-2019	Bilka	Frugt	Dorit	57
9	31-01-2019	Rema 1000	Frugt	Jørgen	50
10	31-01-2019	Lidl	Frugt	Birthe	75
11	31-01-2019	Spar	Kød	Lene	49
12	31-01-2019	Meny	Frugt	Dorit	60
13	31-01-2019	Spar	Brød	Lene	38
14	31-01-2019	Bilka	Blomster	Lene	43
15	31-01-2019	Spar	Kød	Jørgen	45
16	31-01-2019	Føtex	Kød	Birthe	64
17	31-01-2019	Lidl	Grøntsager	Jørgen	57
18	31-01-2019	Bilka	Kød	Dorit	38

Lad os nu pivotere dataene i kolonnen Dato. Vælg kolonneoverskriften Dato, og klik derefter på **Transformér** → **Pivotkolonne**.

## Kapitel 3 Power Query – Opdatering af data



Dialogboksen Pivotkolonne åbnes. Ret **Værdikolonnen** til Værdi. Klik på de avancerede indstillinger, og skift **Funktion til samlet værdi** til Sum. Klik derefter på **OK**.



Klik nu på **Hjem** → **Luk og indlæs**. Excel opretter et nyt regneark med en tabel. Denne tabel indeholder dataene fra CSV-filen, som er blevet transformeret. Det skal se ud som skærbillede nedenfor.

	A	B	C	D
1	Kunder	31-01-2019	28-02-2019	
2	Bilka	420	375	
3	Brugsen	122	165	
4	Føtex	308	276	
5	ICA	254	415	
6	Lidl	486	244	
7	Meny	355	353	
8	Netto	295	476	
9	Rema 1000	353	557	
10	Spar	335	74	
11				
12				

### Opdater kildedataene

For at demonstrere opdateringsprocessen simulerer vi det scenarie, hvor en bruger for eksempel modtager en ny fil dagligt, ugentligt eller månedligt. For at demonstrere dette er vi nødt til at omdøbe begge projektmapper eksempel.

- Opdater data 1.csv kan omdøbes til, hvad du ønsker

## Kapitel 3 Power Query – Opdatering af data

- Opdater data 2.csv skal omdøbes til Opdater data 1.csv

Power Query peger stadig på en fil kaldet Opdater data 1.csv, som indeholder de nye data.

### Opdater dataene

Klik på **Data** → **Opdater alle**.

I baggrunden vil Excel nu importere dataene fra filen til Power Query, anvende de samme transformationer og indlæse dataene i regnearket. De nye data skal vises automatisk på regnearket.

	A	B	C	D	E	F	G	H
1	Kunder	31-01-2019	28-02-2019	31-03-2019	30-04-2019	31-05-2019	30-06-2019	
2	Bilka	420	375	313	447	388	433	
3	Brugsen	122	165	104	72	214	84	
4	Føtex	308	276	333	415	440	415	
5	ICA	254	415	607	88	99	431	
6	Lidl	486	244	364	408	138	215	
7	Meny	355	353	351	172	420	395	
8	Netto	295	476	176	497	309	71	
9	Rema 1000	353	557	493	527	604	393	
10	Spar	335	74	220	314	367	652	
11								

Bemærkede du det? Data fra marts til juni dukkede op med et enkelt klik - det er smart!

Hver gang der sker en ændring af de eksisterende data, eller hvis der modtages en ny fil, er det kun nødvendigt at gemme filen med den samme fil sti, og derefter klikke på en knap. Det er meget effektivt.

I et kommende kapitel vil vi se på det at knytte filstien til en celle, så du kan importere uden at skulle overskrive den forrige fil.

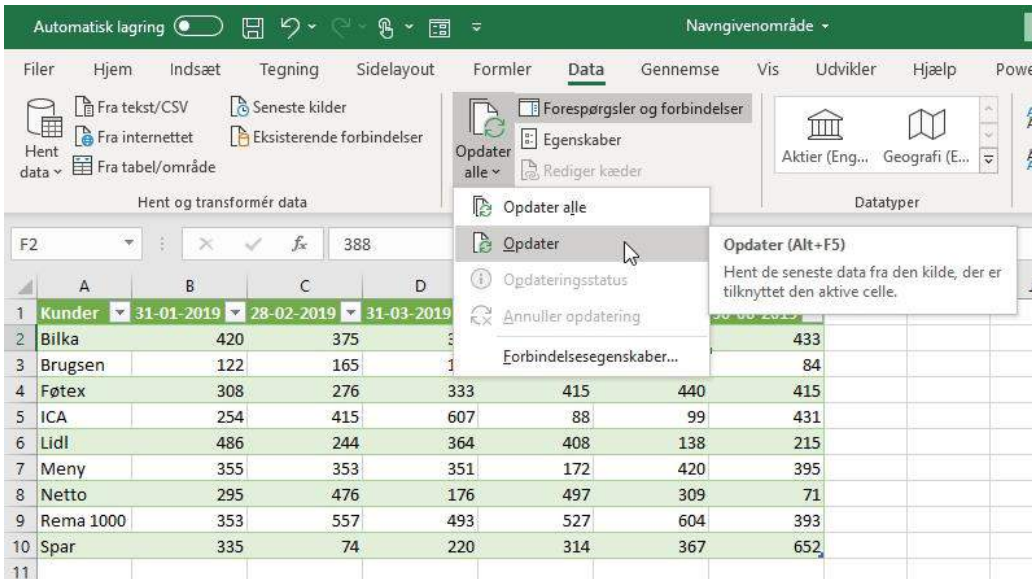
### Opdater kun specifikke forespørgsler

Når vi har mange forespørgsler i en projektmappe, kan det tage et stykke tid at opdatere dem alle. Under disse omstændigheder er det fornuftigt kun at opdatere de forespørgsler, vi har brug for. Der er nogle få måder at gøre dette på.

### Opdater knap

Hvis du klikker på en tabel, der kommer fra en forespørgsel, indeholder listefeltet **Data** → **Opdater en ekstra mulighed**.

## Kapitel 3 Power Query – Opdatering af data

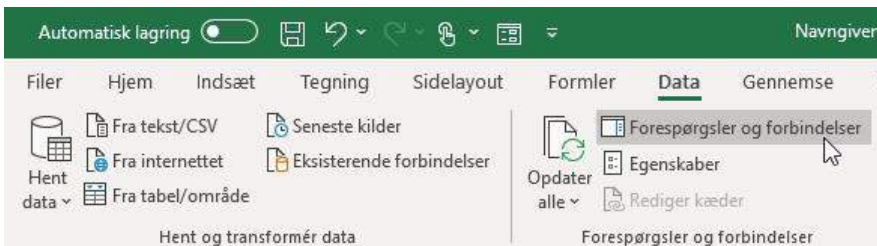


Klik på **Opdater** vil kun opdatere den valgte forespørgsel.

### Menuen Vis forespørgsler

Da vi indlæste dataene fra Power Query i Excel, åbnede panelet **Projektmappe-forespørgsler**. Dette panel viser alle de forespørgsler, der er oprettet i projektmappen.

Hvis dette panel ikke vises, skal du klikke på **Data** → **Forespørgsler og forbindelser**.

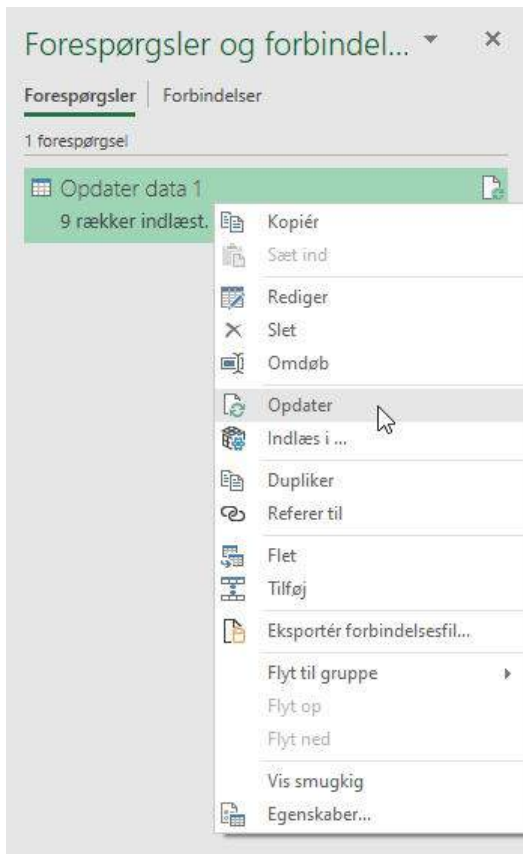


Hver forespørgsel i panelet Projektmappe-forespørgsler har et opdateringsikon. Klik på ikonet for at opdatere dataene.



Alternativt kan vi højre klikke på forespørgslen og vælge Opdater fra menuen.





## VBA-opdatering

Jeg vil ikke afdække dette i denne bog, men du kan bruge VBA til at opdatere individuelle forespørgsler. Dette kan være en nyttig mulighed, når:

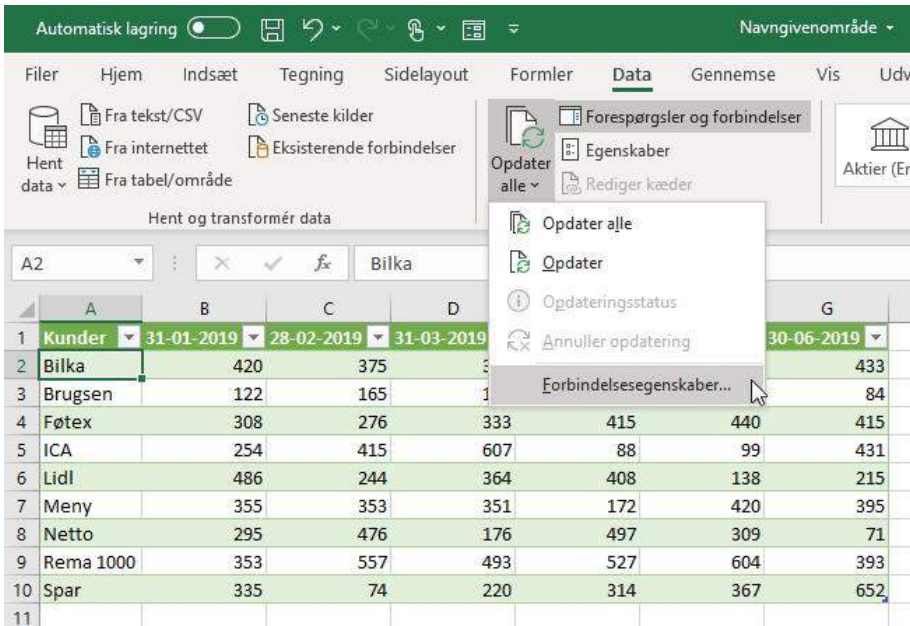
- Du ønsker at lave et brugervenligt interface til brugeren
- Du vil kontrollere rækkefølgen, hvormed forespørgsler opdateres

## Avancerede opdateringsindstillinger

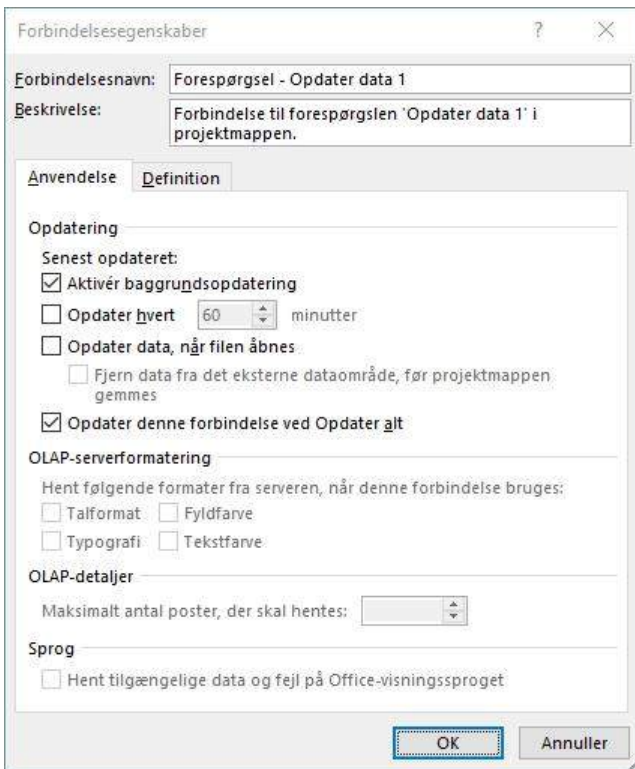
Efterhånden som flere forespørgsler tilføjes til en projektmappe, kan det hurtigt blive tidskrævende at opdatere individuelle forespørgsler eller det bliver for langsomt at opdatere dem alle. Den gode nyhed er, at Excel allerede har indstillinger, der gør det muligt at opdatere data automatisk.

Vælg en celle i en forespørgselstabel, og klik derefter på **Data** → **Opdater alle (listefeltet)** → **Forbindelsesegenskaber**.

### Kapitel 3 Power Query – Opdatering af data



Dialogboksen **Forbindelsesegenskaber** åbnes.



Opdateringsindstillingerne, der er tilgængelige i denne dialogboks, er:

**Baggrundsopdatering** - Opdatering i baggrunden giver os mulighed for at fortsætte med at arbejde, mens dataene opdateres i baggrunden. Det er som standard aktiveret. Hvis vi fjerner denne indstilling, kan vi ikke bruge Excel, før opdateringsprocessen er afsluttet. Når der anvendes baggrundsopdatering, tager selve opdateringsprocessen længere tid, men det er ikke nødvendigvis et problem, da vi kan fortsætte med at arbejde.

**Opdater hvert x minutter** - Projektmappen skal være åben for at muligheden “Opdater hvert x minutter” skal fungere. Det kan være nyttigt, hvor kildedataene regelmæssigt ændres. Det er sandsynligvis kun en god ide at bruge denne indstilling, når baggrundsopdateringen er aktiveret, ellers kan det blive meget irriterende for brugerne.

**Opdater, når du åbner filen** - Opdatering af data automatisk når du åbner filen, er en nyttig funktion, da du ved, at dataene er opdaterede når du åbner den.

**Opdater denne forbindelse ved Opdater alt** - Hvis en forespørgsel indeholder statiske data, er det ikke nødvendigt at opdatere den hver gang. Ved at fjerne checkmærket ved siden af “Opdater denne forbindelse ved Opdater alt”, fjernes disse forespørgsler fra Opdater alle som igen reducerer opdateringstiden.

## Vær opmærksom

---

Det er tid til at overveje de ting, der muligvis vil drille os.

### Lukket eller Åbne filer

Det sted, hvor kildedataene er gemt, er vigtigt for dataopdateringen.

Når dataene er gemt i en ekstern fil, det kan være som CSV-fil, tekstfil eller endda en Excel projektmappe, er det den sidst gemte version af filen, der indlæses i Power Query. Hvis den eksterne fil er åben, vil ændringerne ikke blive inkluderet i opdateringen, før den er gemt.

Hvis dataene og forespørgslen er i den samme projektmappe, er alle ændringer inkorporeret, selvom filen endnu ikke er gemt.

### Forespørgsler baseret på andre forespørgsler

I et senere kapitel skal vi overveje, hvordan man bruger en forespørgsel i en anden forespørgsel. Der er ikke noget galt med denne praksis, men vi skal bare være opmærksomme på konsekvenserne hvis vi kun opdaterer en enkelt forespørgsel, da eventuelle efterfølgende forespørgsler i kæden ikke opdateres.

Ved at bruge Opdater alle opdateres forespørgslerne i den rigtige rækkefølge. Dette er ikke tilfældet, når man opdaterer individuelle forespørgsler, det er op til brugeren at opdatere de rigtige forespørgsler på det rigtige tidspunkt og i den rigtige rækkefølge.

### Opdatering i baggrunden

Opdatering i baggrunden er en smart funktion, da vi kan fortsætte med at arbejde i Excel. Imidlertid vil alle pivottabeller eller formler, der opdateres eller beregnes inden baggrundsopdateringen er afsluttet, ikke nødvendigvis have de korrekte data. Derfor er der nogle tilfælde, hvor du måske fortrækker at fjerne baggrundsopdateringen.

## Kapitel 4 Rediger forespørgsler

I dette kapitel vil vi se på, hvordan du redigerer forespørgsler. Vi har set lidt på dette i et tidligere kapitel, men nu går vi mere i dybde med emnet.

I dette kapitel arbejder vi med filen **Rediger forespørgsler.xlsx**.

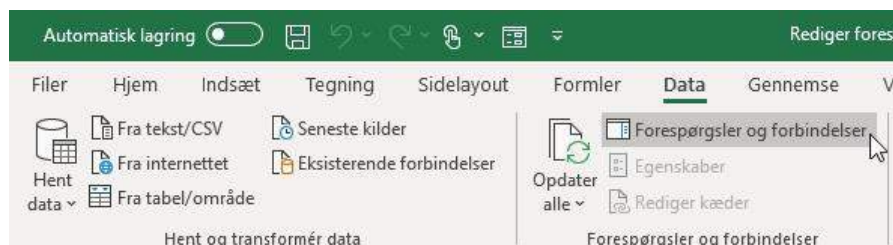
### Rediger en eksisterende forespørgsel

Som med mange funktioner i Power Query, er der flere måder at få adgang til den samme handling på. Åbning af Power Query for at redigere en eksisterende forespørgsel er ikke anderledes, der er mange måder at gøre det på. Jeg vil kort afdække det mest almindelige nedenfor, skønt vi kun behøver at kende en eller to.

### Åben Power Query

*Projektmappe-forespørgsler - Dobbeltklik*

Den nemmeste metode er at vælge **Data** → **Forespørgsler og forbindelser** via båndet.



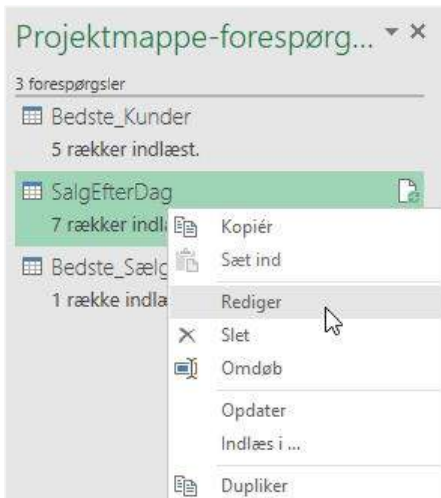
Panelet Projektmappe-forespørgsler åbnes og dobbeltklik på navnet på en forespørgsel.



## Kapitel 4 Power Query – Rediger forespørgsler

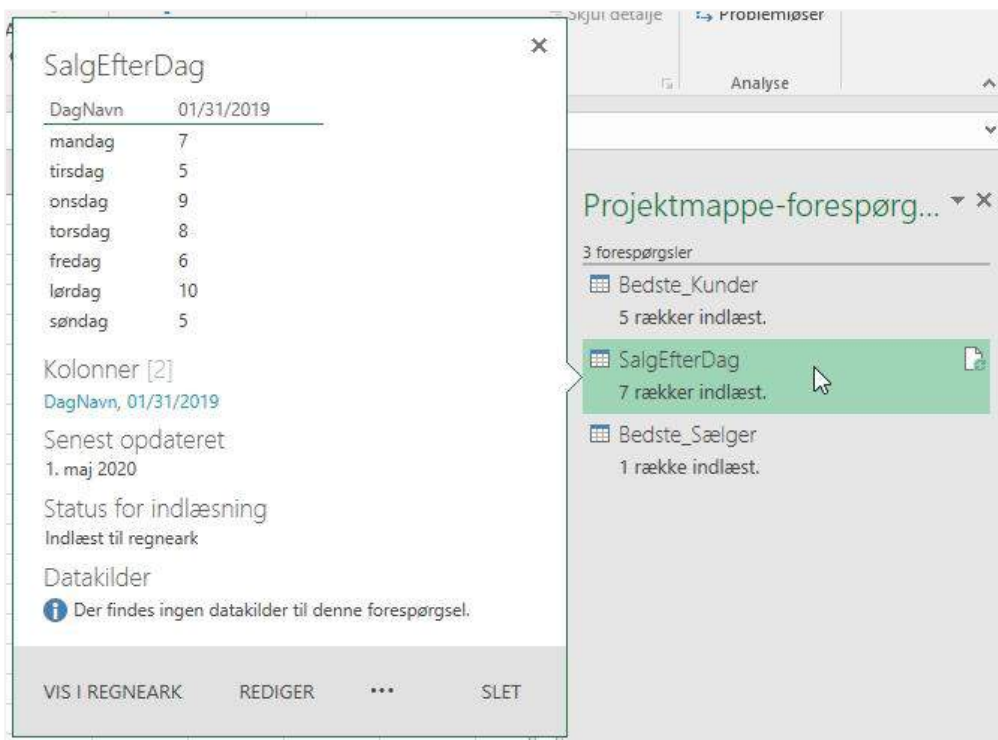
*Projektmappe-forespørgsler – Højre klik Rediger*

I panelet Projektmappe-forespørgsler kan vi højre klikke på forespørgslen og vælge Rediger.



*Projektmappe-forespørgsler – Før musen henover*

Når du holder musen hen over en forespørgsel (du skal ikke klikke, bare føre musemarkøren over) viser Excel forespørgselsoversigten. Klik på Rediger i bunden.

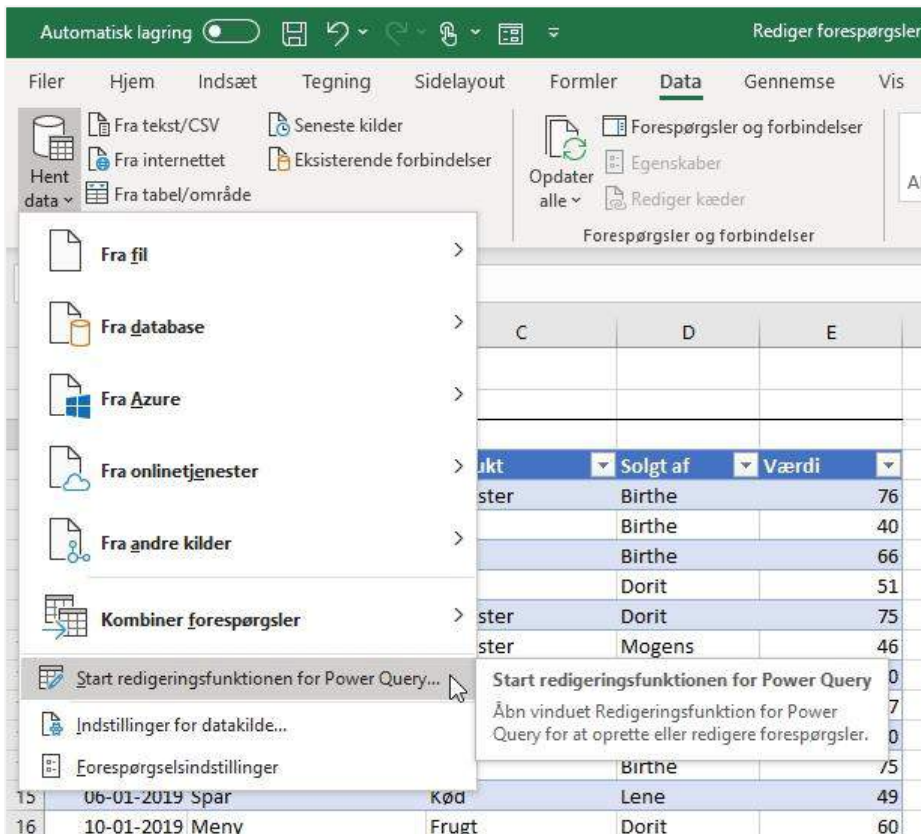


*Start Power Query editoren*

En anden metode er at åbne Power Query editoren uden at vælge en bestemt forespørgsel.

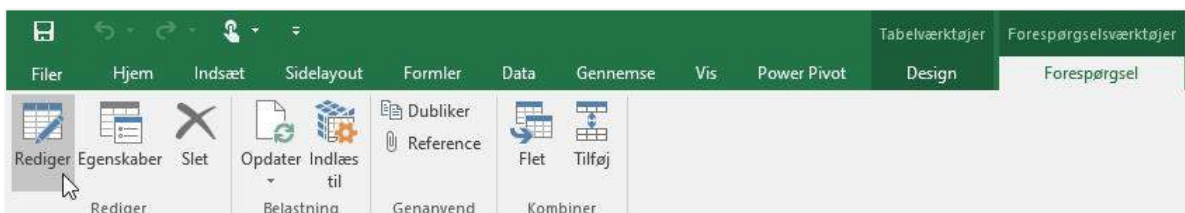
## Kapitel 4 Power Query – Rediger forespørgsler

Vælg på **Data** → **Hent data** → **Start redigeringsfunktionen for Power Query**.



### Forespørgselsværktøjer på båndet

For brugere af Excel 2016! Lige en metode til du kan prøve. Vælg en hvilken som helst celle i forespørgselstabelen. Et nyt forespørgselsbånd vises. Klik på **Forespørgsel** → **Rediger**.



### Vælg en forespørgsel der skal redigeres

Når Power Query editoren åbnes, kan vi redigere enhver forespørgsel, der findes i projektmappen. Udvid forespørgselslisten ved at klikke på pilen øverst til venstre i datavisningen.

## Kapitel 4 Power Query – Rediger forespørgsler

The screenshot shows the Power Query editor interface. On the left, a table with 7 rows and 3 columns is displayed. The columns are 'Dag navn', '1.2', and '01/31/2019'. The rows represent days of the week and their corresponding counts.

	Dag navn	1.2	01/31/2019
1	mandag		7
2	tirsdag		5
3	onsdag		9
4	torsdag		8
5	fredag		6
6	lørdag		10
7	søndag		5

The formula bar at the top shows: `= Table.RemoveColumns(#"Sorterede rækker1",{"Ugedag"}))`

The right-hand pane, 'Forespørgselsindstillinger', shows the query name 'SalgEfterDag'. Under 'ANVENDTE TRIN', 'Fjernede kolonner1' is selected.

Forespørgselslisten åbnes.

The screenshot shows the Power Query editor interface. On the left, a list of queries is displayed: 'Bedste\_Kunder', 'SalgEfterDag', and 'Bedste\_Sælger'. 'SalgEfterDag' is selected and highlighted in green.

The main editor area shows the same table as in the previous screenshot.

The right-hand pane, 'Forespørgselsindstillinger', shows the query name 'SalgEfterDag'. Under 'ANVENDTE TRIN', 'Fjernede kolonner1' is selected.

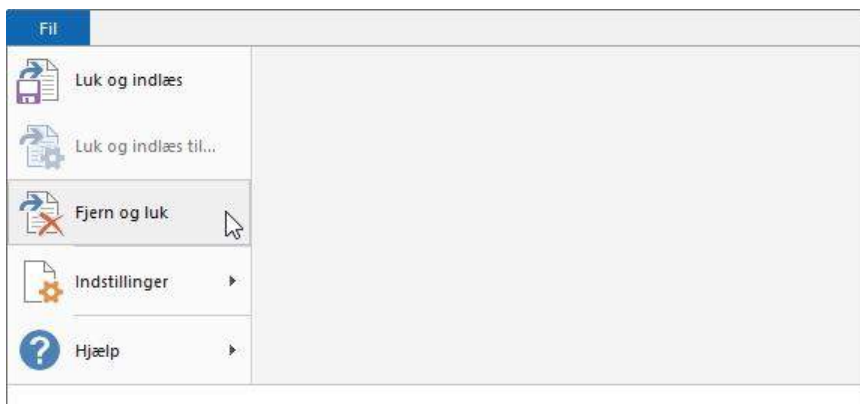
Klik på en forespørgsel for at aktivere den i visningen og for at redigere den.

### Luk og indlæs

I denne bog har vi hidtil kun brugt Luk og indlæs når vi opretter forespørgsler. Vi har kun brug for at indlæse enkelte forespørgsler i Excel. Men vær opmærksom på at Luk og indlæs anvendes til alle forespørgsler i projektmappen. Så hvis du ændrer i flere forespørgsler og derefter opdaterer med Luk og indlæs, opdateres alle forespørgsler, ikke kun den aktive forespørgsel.

### Luk uden at gemme ændringer

Efter redigering af en forespørgsel kan vi beslutte, at vi ikke ønsker at indlæse ændringerne i Excel. For at kassere ændringerne skal du klikke på **Fil** → **Fjern og luk**.



Alternativt kan du klikke på standard [X] knappen (øverst til højre i vinduet). Der vises en bekræftelses dialogboks for at sikre, at vi ikke klikker ved en fejl.



### Rediger trinnene

Nu hvor vi er i Power Query og kan bevæge os frit mellem forespørgsler, er vi klar til at begynde at redigere individuelle trin. I dette kapitel vil vi arbejde med eksempel filen til at ændre de eksisterende transformationer.

Brug en af metoderne ovenfor til at åbne Power Query editoren, og lad os komme i gang med at redigere nogle transformationer.

### Grundlæggende indstillinger

Som vi allerede har set i et tidligere kapitel, findes tandhjulsikonet ved siden af mange af trinnene. Hvis du klikker på det, vises indstillingsvinduet for det specifikke trin.

Marker forespørgslen **Bedste\_Kunder**, og klik derefter på tandhjulsikonet ved siden af trinnet **Bevarede første rækker**.



## Kapitel 4 Power Query – Rediger forespørgsler

Kunder	Mængde
1 Rema 1000	2927
2 Bilka	2376
3 Føtex	2187
4 Meny	2046
5 Spar	1962

Dialogboksen **Bevar de øverste rækker** åbnes. Forespørgslen er i øjeblikket indstillet til at vise 5 kunder, ret dette til 3 og klik på **OK**.

Bevar de øverste rækker

Angiv, hvor mange rækker du vil bevare.

Antal rækker

3

OK Annuller

### Tilføj, indsæt, slet og flyt trin

Vælg forespørgslen **Bedste\_Kunder**, og klik på trinnet kaldet **Månedens beregnede slutning**. Datavisningen viser alle trin op til slutningen af det aktuelle valg.

For at indsætte et trin skal du vælge et eksisterende trin og derefter udføre de transformationer, du vil anvende herefter.

Klik på menuen i kolonnen Dato og vælg 31. januar 2019 (afhængigt af dit datoformat kan det se lidt anderledes ud hos dig).

## Kapitel 4 Power Query – Rediger forespørgsler

The screenshot shows the Power Query Editor interface. A dialog box for filtering dates is open. The 'Dato' column is selected. The 'Søg efter' section shows a list of dates with '31-01-2019' selected. The background table has columns 'Kunder' and 'Værdi'.

	Dato	Kunder	Værdi
		Brugsen	76
		Netto	40
		Føtex	66
		Netto	51
		Netto	75
		Brugsen	46
		Bilka	40
		Bilka	57
		Rema 1000	50
		Lidl	75
		Spar	49
		Meny	60
		Spar	38
		Bilka	43
		Spar	45
		Føtex	64
		Lidl	57
		Bilka	38
19	31-01-2019	Lidl	49
20	31-01-2019	Meny	73

Et nyt trin er tilføjet i midten af trinnene.

The screenshot shows the 'ANVENDE TRIN' pane in Power Query. The 'Filtrerede rækker' step is highlighted, and a tooltip shows the filter criteria.

- Source
- Ændret type
- Fjernede kolonner
- Månedens beregnede slutning
- Filtrerede rækker** ✖
- Grupperede rækker ✖
- Sorterede rækker
- Bevarede første rækker ✖

Fortsæt med at klikke gennem trinnene for at se transformationerne; der skulle ikke være problemer med dette eksempel.

Når du ændrer, flytter eller indsætter trin, kan det forårsage problemer senere i forespørgslen, da de ændringer, du foretager, ikke vil indgå korrekt i de efterfølgende trin. Lad mig vise dig et eksempel.

Klik på trinnet **Fjernede kolonner**. Ret overskriften på kolonnen **Dato** til **Månedens slut**.

## Kapitel 4 Power Query – Rediger forespørgsler

The screenshot shows the Power Query Editor interface. On the left, a list of queries includes 'Bedste\_Kunder', 'SalgEfterDag', and 'Bedste\_Sælger'. The main area displays a table with columns: 'Måned slut', 'Kunder', and 'Værdi'. The table contains 14 rows of data. On the right, the 'Forespørgselsindstillinger' pane is open, showing the 'ANVENDTE TRIN' (Applied Steps) list. The steps are: 'Source', 'Ændret type', 'Fjernede kolonner', 'Månedens beregnede slutning', 'Filtrerede rækker', 'Grupperede rækker', 'Sorterede rækker', and 'Bevarede første rækker'. The 'Fjernede kolonner' step is currently selected.

	Måned slut	Kunder	Værdi
1	09-01-2019	Brugsen	76
2	02-01-2019	Netto	40
3	19-01-2019	Føtex	66
4	19-01-2019	Netto	51
5	12-01-2019	Netto	75
6	22-01-2019	Brugsen	46
7	23-01-2019	Bilka	40
8	25-01-2019	Bilka	57
9	23-01-2019	Rema 1000	50
10	10-01-2019	Lidl	75
11	06-01-2019	Spar	49
12	10-01-2019	Meny	60
13	11-01-2019	Spar	38
14	31-01-2019	Bilka	43

Der vises en advarsel. Ikke alle advarselsmeddelelser er dårlige; du skal vælge, om det er relevant for dit scenarie eller ej. I dette eksempel skal vi forsøge at identificere fejlen, så klik på **Indsæt**.

The dialog box titled 'Indsæt trin' (Insert Step) contains the following text: 'Er du sikker på, at du vil indsætte et trin? Hvis du indsætter ekstra trin, kan det påvirke efterfølgende trin, som kan medføre, at din forespørgsel ikke kan køres.' At the bottom, there are two buttons: 'Indsæt' (Insert) and 'Annuller' (Cancel).

Det nye trin er tilføjet (se skærbillede nedenfor).

The 'ANVENDTE TRIN' (Applied Steps) pane now shows an updated list of steps: 'Source', 'Ændret type', 'Fjernede kolonner', 'Omdøbte kolonner', 'Månedens beregnede slutning', 'Filtrerede rækker', 'Grupperede rækker', 'Sorterede rækker', and 'Bevarede første rækker'. The 'Omdøbte kolonner' (Rename Columns) step is highlighted, indicating it has been added to the sequence.

Klik nu på trinnet **Månedens beregnede slutning**. Åh nej ... der er en fejl. Dette skyldes, at dette trin forsøger at finde en kolonne kaldet Dato, som ikke længere er der.

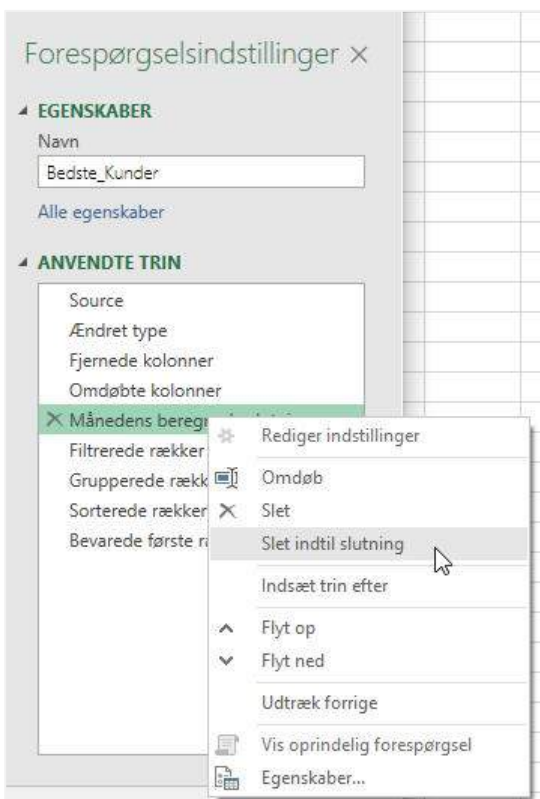
## Kapitel 4 Power Query – Rediger forespørgsler



Hvis du stadig har formellinjen åben, kan du se henvisningen til kolonnen Dato, som ikke længere er tilgængelig (se skærbilledet ovenfor).

Nu bliver du præsenteret for nogle valg. Har du nu også brug for at indsætte dette trin på dette punkt? Bør du slette og udføre de berørte trin igen? Skal du benytte M koden til de resterende trin for at gentage ændringen? Kun du kan tage denne beslutning. Når du har været i denne situation et par gange, vil du hurtigt finde ud af, hvad der er bedst.

Hvis du vælger at slette alle de resterende trin, skal du **højre klikke** på trinnet og vælge **Slet indtil slut**.

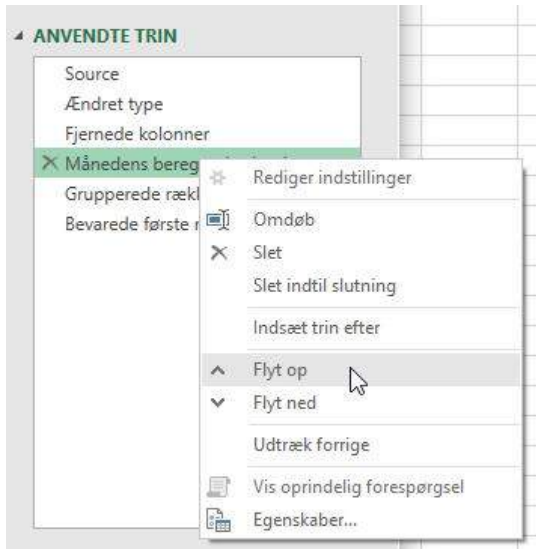


Da dette kun var en illustration af de potentielle problemer med ændring af trin, lad os da slette det omdøbte kolonne trin, som vi tilføjede, og klikke på X ved siden af det tilføjede trin.



## Flyt trin

For at flytte et trin op eller ned er der flere muligheder. Vi kan enten **højre klikke** på trinnet og vælge **Flyt op** eller **Flyt ned**.



Den anden mulighed er at klikke på et enkelt trin, holde museknappen nede og trække trinnet til et nyt sted.

Som du kan forestille dig, har flytningen af trin de samme problemer som indsættelse af trin.

## Hvor forsvandt mine trin hen?

Power Query er smart, den prøver at gruppere lignende handlinger sammen.

Vælg forespørgslen **SalgEfterDag**, og sørg for, at du har valgt det sidste trin, der er **Fjernede kolonner**. Vi ændrer nu overskrifterne for hver kolonne:

- **DagNavn** ændres til **Dag** - et nyt trin er tilføjet.
- **01-31-2019** (eller din lokale dato) ændres til **Antal salg** - et nyt trin tilføjes ikke.

Se på de anvendte trin igen ... der tilføjes kun et trin, hvor blev det andet trin af?

## Kapitel 4 Power Query – Rediger forespørgsler

The screenshot shows the Power Query Editor interface. On the left, a table is displayed with the following data:

	Dag	Antal salg
1	mandag	7
2	tirsdag	5
3	onsdag	9
4	torsdag	8
5	fredag	6
6	lørdag	10
7	søndag	5

On the right, the 'Forespørgselsindstillinger' (Query Settings) pane is open, showing the 'ANVENDETE TRIN' (Applied Steps) list. The steps are:

- Source
- Ændret type
- Indsatte navn på dag
- Omrøkerede kolonner
- Beregnet dag i ugen
- Fjernede kolonner
- Månedens beregnede slutning
- Ændret type1
- Filtrerede rækker
- Omrøkerede kolonner1
- Pivotkolonne
- Sorterede rækker
- Fjernede kolonner1
- Omdøbte kolonner

Se på formellinjen (hvis du har den åben), vil du se, at begge trin er blevet kombineret til et.

Dette sker kun, hvor lignende trin udføres efter hinanden. Det hjælper bestemt med at holde antallet af anvendte trin mindre, end de ellers kunne være.

### M koder i formellinje og avanceret editor

OK, tingene er ved at blive alvorlige - vi skal ændre M koden! Vi kan redigere koden direkte på to måder (1) Formellinjen (2) Avanceret editor. Da jeg foretrækker formellinje metoden, starter vi med den.

#### Formellinjen

Hvis formellinjen ikke er synlig, kan du slå den til ved at klikke på **Vis** → **Formellinje**.

Vælg forespørgslen **SalgEfterDag**, og klik på trinnet **Sorterede rækker**.

Formellinjen viser følgende:

```
= Table.Sort(Pivotkolonne,{{"Dag i uge", Order.Ascending}})
```

Vi kan muligvis ikke skrive M koden fra bunden af, men det betyder ikke, at vi ikke kan redigere den. I formellinjen skal du ændre "Dag i uge" til "Dag Navn". Se igen på data forhåndsvisning; den er nu sorteret alfabetisk efter Dag navn.

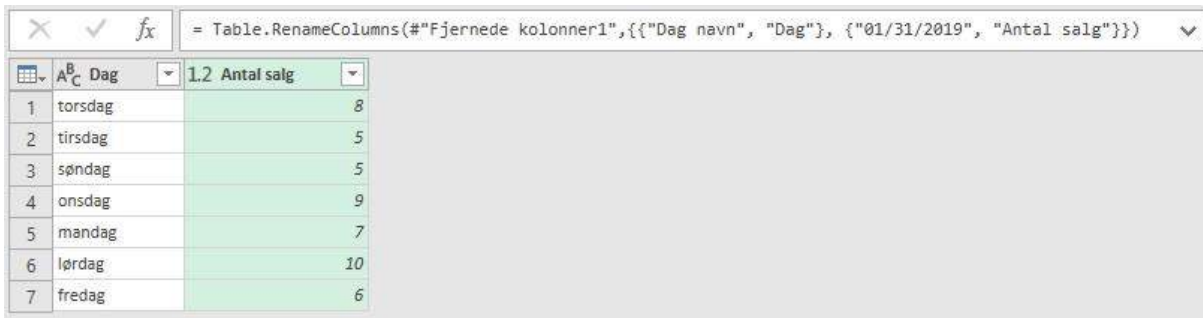
Skift derefter teksten, hvor der står Order.Acending til Order.Descending. Kontroller eksempelvisningsvinduet igen, og bemærk, at rækkefølgen er ændret.

Den endelige formel skal se sådan ud:

```
= Table.Sort(Pivotkolonne,{{"DagNavn", Order.Descending}})
```

Klik på det sidste trin i forespørgslen. Dataforhåndsvisning ser nu sådan ud.

## Kapitel 4 Power Query – Rediger forespørgsler



The screenshot shows the Power Query Advanced Editor interface. The formula bar at the top contains the M code: `= Table.RenameColumns(#"Fjernede kolonner1",{"Dag navn", "Dag"}, {"01/31/2019", "Antal salg"})`. Below the formula bar, a table is displayed with the following data:

	Dag	Antal salg
1	torsdag	8
2	tirsdag	5
3	søndag	5
4	onsdag	9
5	mandag	7
6	lørdag	10
7	fredag	6

De ændringer, vi har foretaget i formellinjen ovenfor, er afspejlet i vores forespørgsel.

### Avanceret editor

Vi kunne foretage de samme ændringer i Advanced Editor. Klik på **Hjem** → **Avanceret editor**.

Den avancerede editor viser hvert trin, vi skal identificere den relevante linje, så kan vi foretage de samme ændringer.

### Indlæs i Excel

Til sidst skal vi lukke ned ved at klikke på Luk og indlæs. De ændringer, vi har foretaget, indlæses i tabellerne på regnearket.

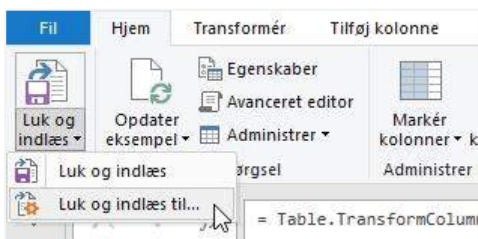
## Kapitel 5 Indstillinger for Luk og indlæs

I dette kapitel af Power Query bogen vil vi kort se på indstillingerne Luk og indlæs. Indtil videre har vi kun brugt standardindstillingen, der indlæser output fra forespørgslen i et nyt regneark. Men der er mange andre muligheder, vi kunne bruge i stedet.

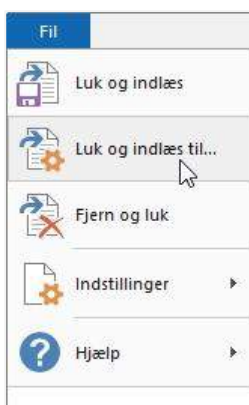
Vi kommer ikke til at bruge alle Luk og Indlæs indstillinger i denne bog. Men du skal være opmærksom på, at de er der, så du kan gøre brug af dem, når du har brug for det.

### Luk og indlæs til... indstillinger

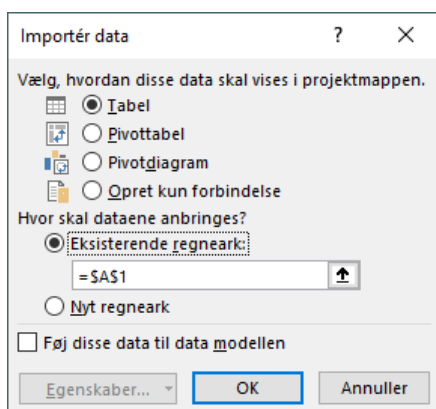
Når vi har importeret dataene til Power Query og foretaget vores transformationer, skal vi derefter beslutte, hvad vi skal gøre med forespørgslen. Hvis du klikker på Luk og indlæs, indlæses det direkte i Excel som et nyt regneark. Men for at se alle de andre indstillinger skal du vælge Luk og indlæs til... . Vi finder denne mulighed i listefeltet **Luk og indlæs til** (se skærbilledet nedenfor).



Eller vi kan finde det ved at klikke på **Fil** → **Luk og indlæs til...** (se skærbillede nedenfor).



Dialogboksen Importer data åbnes.





Denne dialogboks indeholder tre forskellige valg (1) Vælg, hvordan dataene skal vises i projektmappen (2) Vælg, hvor skal dataene indlæses (3) Hvorvidt dataene skal tilføjes til datamodellen. Lad os se på hver af disse.

### Sådan ses forespørgselsoutput

De fire øverste indstillinger i dialogboksen Indlæs til vedrører visning af output:

**Tabel** - Indlæs dataene i Excel som en tabel, der vises i regnearket.

**Pivottabel** - Kolonnerne i de transformerede data er tilgængelige til brug som dimensioner i en pivottabel.

**Pivotdiagram** - Kolonnerne i de transformerede data er tilgængelige til brug i et pivotdiagram.

**Opret kun forbindelse** - Outputtet er ikke synligt på noget regneark i nogen form. Det kan dog bruges i en anden forespørgsel eller som en parameter (mere om det i et senere kapitel).

### Hvor placeres forespørgslens output

Den anden gruppe af indstillinger bestemmer, hvor forespørgslens output skal placeres. Dette er kun relevant, hvis vi valgte Tabel i den første gruppe.

Valgmulighederne er enten:

- Nyt regneark (output indsættes ved celle A1)
- Eksisterende regneark (hvorefter vi kan definere den nøjagtige celle).

### Føj disse data til datamodellen

Datamodellen er en moderne måde for Excel at håndtere data på. Hvis vi bruger Power Pivot eller opretter relationer, er en datamodel vigtig.

Hvad hvis vi kun ønsker en enkelt tabel til at oprette en Pivottabel?

Pivothukommelsen er Excels måde at gemme data, der bruges i en pivottabel. Vi kan ikke se hukommelse; den gemmes ikke synligt i Excel filformatet. Men vi kan helt sikkert se virkningen af Pivothukommelsen, da filstørrelser for projektmapper, der indeholder Pivottabeller, kan blive meget store.

Ved at indlæse dataene i en datamodel snarere end en pivottabel, er filstørrelserne markant mindre.

Som en illustration kan man indlæses 1 million rækker med data (der kun indeholder to kolonner) i datamodellen. Derefter kan man indlæse datamodellen for at oprette en pivottabel; filstørrelsen er omkring 800 KB. Derimod kan dataene indlæses direkte i en pivottabel uden en datamodel, og filstørrelsen er ca. 3.500 KB. Over 4 gange større! I dette eksempel er filstørrelserne små, men når man arbejder med flere data, er forskellen endnu mere markant.

Derfor, selvom man ikke ønsker at bruge Power Pivot eller oprette relationer, er datamodellen stadig en bedre mulighed.

### Knappen Egenskaber

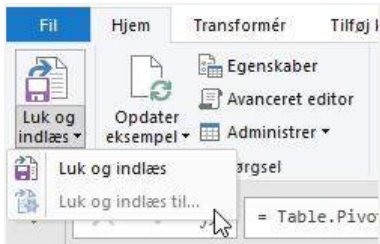
Hvis du klikker på knappen Egenskaber, åbnes dialogboksen Forespørgselsegenskaber, som vi allerede har set i kapitlet Opdater data.

## Ret indstilling for Luk og indlæs for en eksisterende forespørgsel

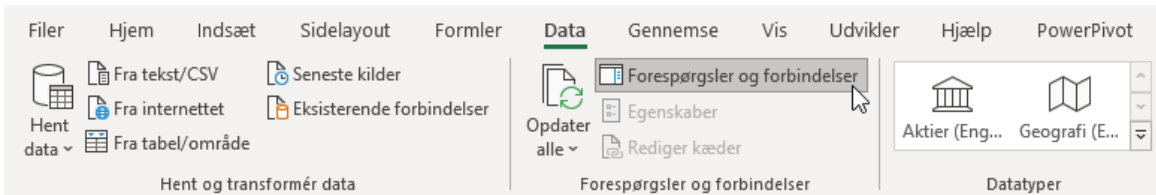
---

Hvis vi går tilbage til en eksisterende forespørgsel, bliver indstillingen Luk og indlæst til... grå.

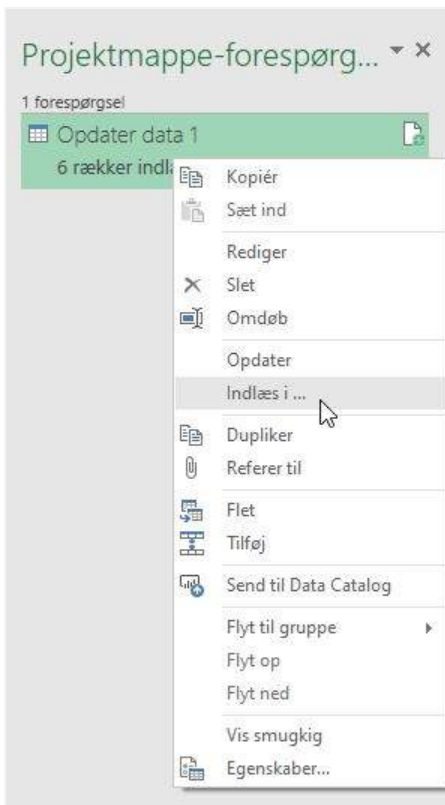
## Kapitel 5 Power Query – Indstillinger for Luk og indlæs



I stedet for at ændre indstillingerne for Luk og indlæs i Power Query, kan vi se dem i vinduet Projektmappe-forespørgsler. I Excel skal du klikke på **Data → Vis forespørgsler og forbindelser**.

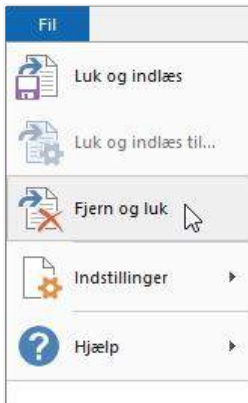


Højre klik på forespørgslen på listen over forespørgsler og vælg Indlæs i ...



## Fjern og luk

Hvis vi har foretaget nogle ændringer i Power Query editoren, som vi ikke længere ønsker, kan vi kassere eventuelle ændringer ved at klikke på **Fil** → **Fjern og luk**.



## Kapitel 6 Brug af parametre

Idet Power Query registrerer transformationstrinnene, indeholder det mange hardkodede værdier i M koden. Hvis vi for eksempel filtrerer en kolonne for at vælge alle værdier over 50, vil 50 være en hardkodet værdi i M koden. Eller hvis vi importerer en CSV-fil, er filstien hardkodet i forespørgslen.

Hvad hvis vi vil ændre disse værdier eller filstier? Naturligvis kunne vi **redigere forespørgslen hver gang**, hvilket ville være meget tidskrævende. Eller vi kunne oprette parametre.

Udtrykket "Parameter" kan være lidt forvirrende. Vi bruger udtrykket variabel, når vi programmerer i VBA, eller måske betingelser, kriterier eller argumenter, når vi laver Excel formler, dette er alle de samme ting. Parametre, variabler, betingelser, kriterier og argumenter er alle værdier, vi kan ændre for at få et andet output. Microsoft besluttede at bruge udtrykket parameter da de designede Power Query.

I dette kapitel bruger jeg celleværdier som parametre; Derfor kan vi ved at ændre celleværdien ændre resultatet af en forespørgsel. Parameteren kan være indeholdt i en CSV-fil eller som en databaseindstilling. Men vi fokuserer på den situation, du mest sandsynligt vil støde på, som er en celleværdi.

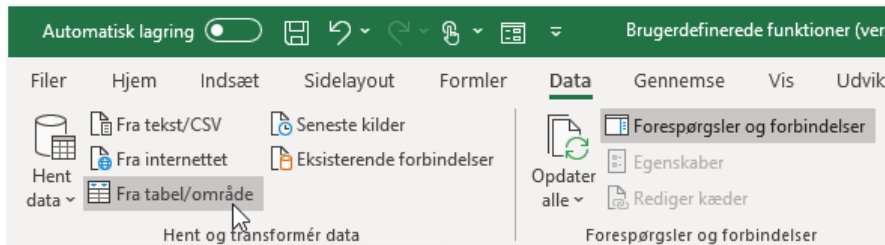
Da Power Query blev udviklet, introduceres nye funktioner der var lettere at bruge. Parametre er en af disse funktioner. Metoden, jeg vil vise dig, tror jeg, i øjeblikket er den nemmeste og bedste at implementere.

I dette kapitel arbejder vi med filen **Brug af parametre.xlsx**.

### Opret forespørgslen

Det første trin er at oprette en forespørgsel som normalt. Som bemærket ovenfor, vil alle trinene være hardkodet.

Vælg en hvilken som helst celle i kildetabellen, og vælg **Data** → **Fra tabel/område**.



Power Query editoren åbnes. Foretag følgende transformationer.

### Kolonnen Dato

Klik på ikonet **Dato og tid** ved siden af overskriften **Dato**, vælg **Dato** i menuen.

## Kapitel 6 Power Query - Brug af parametre

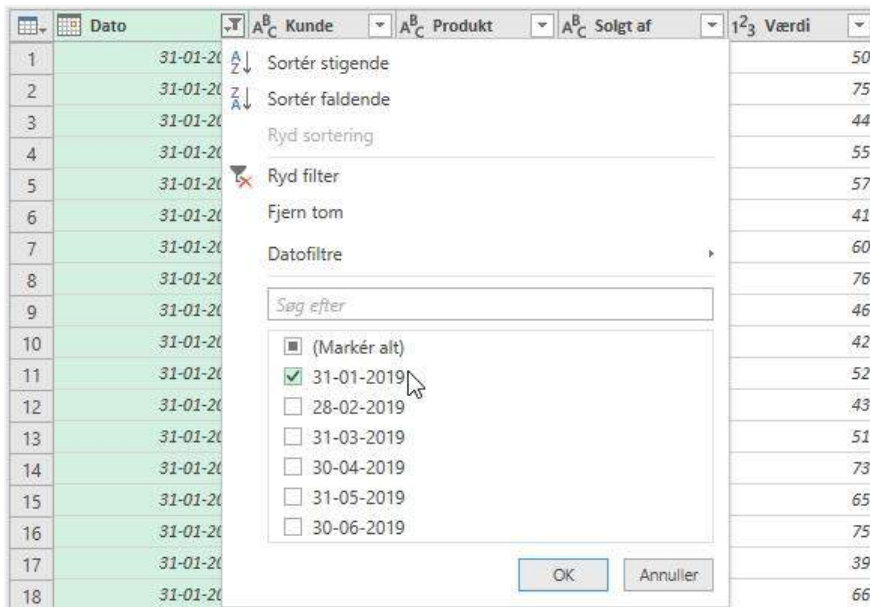
	Dato	Kunde	Produkt	Solgt af	Værdi
1	1.2	Decimaltal	Chokolade	Dorthe	64
2	\$	Valuta	Chokolade	Lars	46
3	1 2 3	Heltal	Chokolade	Søren	58
4	%	Procent	Chokolade	Dorthe	72
5			Chokolade	Jette	50
6		Dato/klokkeslæt	Chokolade	Dorthe	76
7		Dato	Chokolade	Mette	70
8		Klokkeslæt	Chokolade	Søren	72
9		Dato/klokkeslæt/tidszone	Chokolade	Lars	46
10		Varighed	Chokolade	Lars	71
11		Tekst	Chokolade	Jørgen	84
12		Sand/falsk	Chokolade	Jørgen	57
13			Chokolade	Søren	60
14		Binær	Chokolade	Jørgen	50
15		Ved hjælp af landestandard...	Chokolade	Lars	52
16	18-06-2019 00:00:00	Fakta	Chokolade	Dorthe	49
17	10-01-2019 00:00:00	Brugsen	Chokolade	Søren	75

Vælg kolonneoverskriften Dato, og klik derefter på **Transformerer** → **Dato** → **Måned** → **Månedens slutning**.

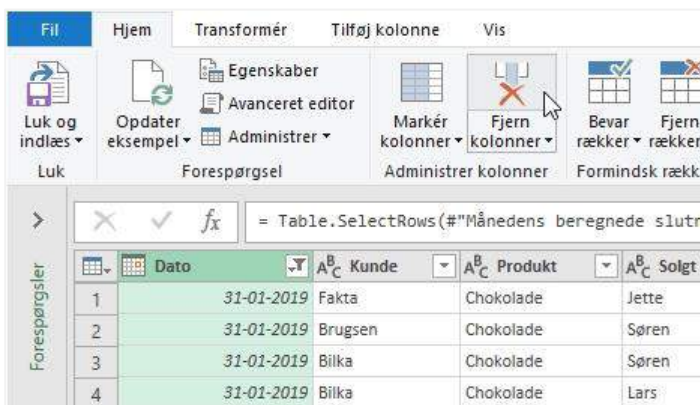
The screenshot shows the Power Query Desktop interface. The 'Transform' ribbon is active, and the 'Dato' column is selected. The 'Dato' menu is open, showing options like 'Måned', 'Månedens start', 'Månedens slutning', 'Dage i måned', and 'Navn på måned'. The 'Månedens slutning' option is highlighted. The background shows a table with columns 'Dato', 'Kunde', 'Produkt', 'Solgt af', and 'Værdi'.

Filtrer kolonnen Dato til kun at omfatte 31. januar 2019. Datoformatet kan se forskellig ud, afhængigt af dine indstillinger.

## Kapitel 6 Power Query - Brug af parametre



Sørg for, at kolonnen Dato stadig er markeret, og klik derefter på **Hjem** → **Fjern kolonner**



### Kolonnen Solgt af

Klik på filterikonet i kolonnen Solgt af og sikre dig, at kun **Jørgen** er valgt.

Sørg for kolonnen Solgt af stadig er markeret, vælg **Hjem** → **Fjern kolonner**.

Det er nok transformationer for nu. Klik på **Hjem** → **Luk og indlæs**.

Tabellen skal se sådan ud:

	A	B	C
1	Kunde	Produkt	Værdi
2	Føtex	Kylling	51
3	ICA	Kylling	39
4	Fakta	Rugbrød	80
5	Rema 1000	Rugbrød	67
6	Spar	Æbler	73

Fra kildedataene oprettede vi en tabel, der viser de produkter, der er solgt af Jørgen i januar 2019. Men hvad nu hvis vi vil have de produkter, der sælges af Mette i marts 2019, eller Lars i marts 2019? Det er her parametrene kommer ind. I det næste afsnit opretter vi nogle parametre til dynamisk ændring af navn og dato.

## Opret parametrene

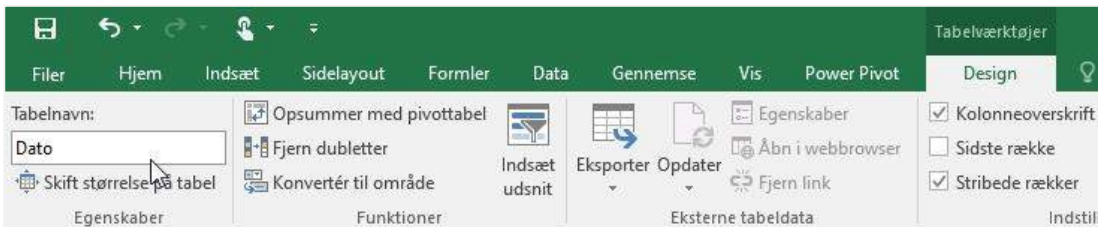
Den korte forklaring er at en parameter blot er en normal forespørgsel, hvor vi borer ned i selve værdien og indlæser den som en forbindelse.

I dette eksempel bruger vi en Excel tabel som kilde, men den kan ligeledes være i det navngivne område, CSV-fil eller en hvilken som helst anden datakilde, vi kan få ind i Power Query.

I regnearket der indeholder forespørgselsstabellen oprettes to tabeller med enkeltværdier i:

F	G	H
	Dato	
	31-03-2019	
	Solgt af	
	Søren	

Efter oprettelse af hver tabel har jeg omdøbt dem.



Den første tabel har jeg navngivet **Dato** og den anden **SolgtAf**.

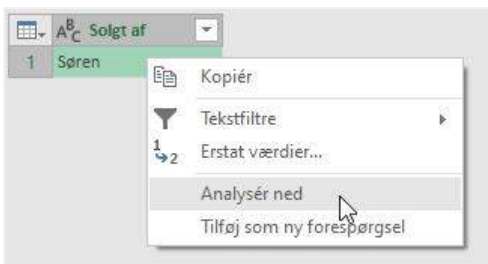
## Oprettelse af en tekstparameter

Først opretter vi parameteren til ændring af navn.

Vælg cellen i tabellen SolgtAf og opret en forespørgsel ved at klikke på **Data** → **Fra tabel/område**.

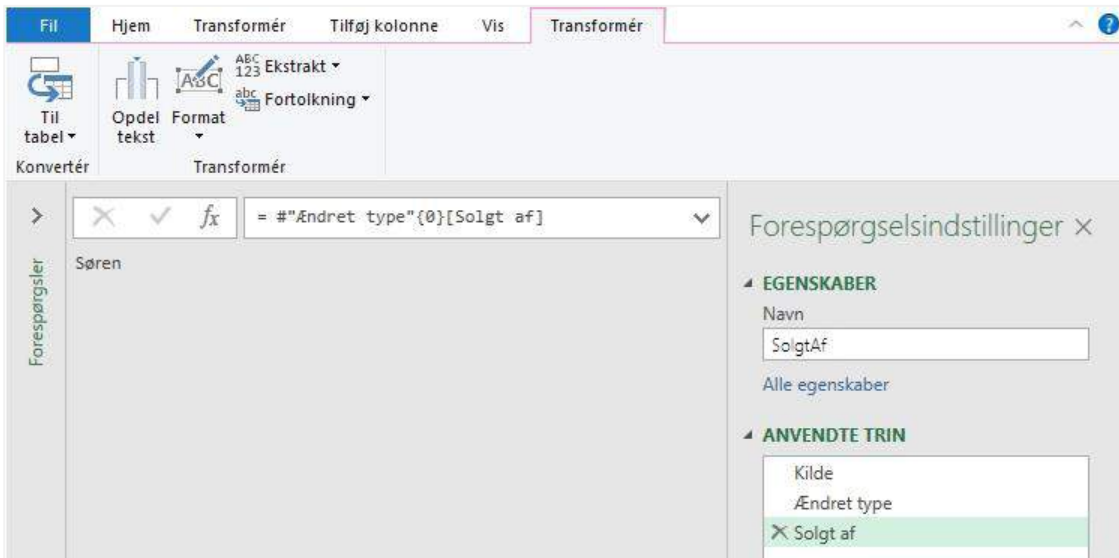
Vær opmærksom på datatypen. Kolonnen Solgt af i den originale forespørgsel ovenfor er en tekst datatype, og datatypen i denne forespørgsel er også tekst. Vi har brug for, at disse er identiske.

**Højre klik på værdien** i Power Query og vælg **Analyser ned** i menuen.



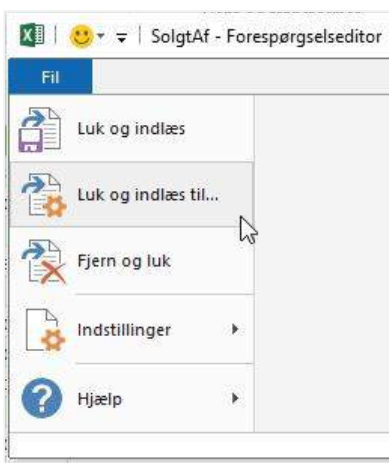
Vinduet ændres til en visning, vi ikke har set før, vinduet **Tekstværktøjer**:

## Kapitel 6 Power Query - Brug af parametre

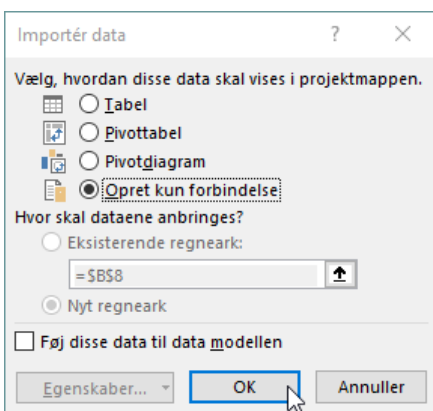


Skriv en note om forespørgselsnavnet, som er **SolgtAf** (ingen mellemrum), som vist på skærbilledet ovenfor.

Klik på **Fil** → **Luk og indlæs til**.



Vælg **Opret kun forbindelse** i dialogboksen **Importer data**, og klik derefter på **OK**.



Panelet **Projektmappe-forespørgsler** viser nu to forespørgsler, de originale data, kaldet **Salgsdata**, og tekstparameteren kaldet **SolgtAf**.

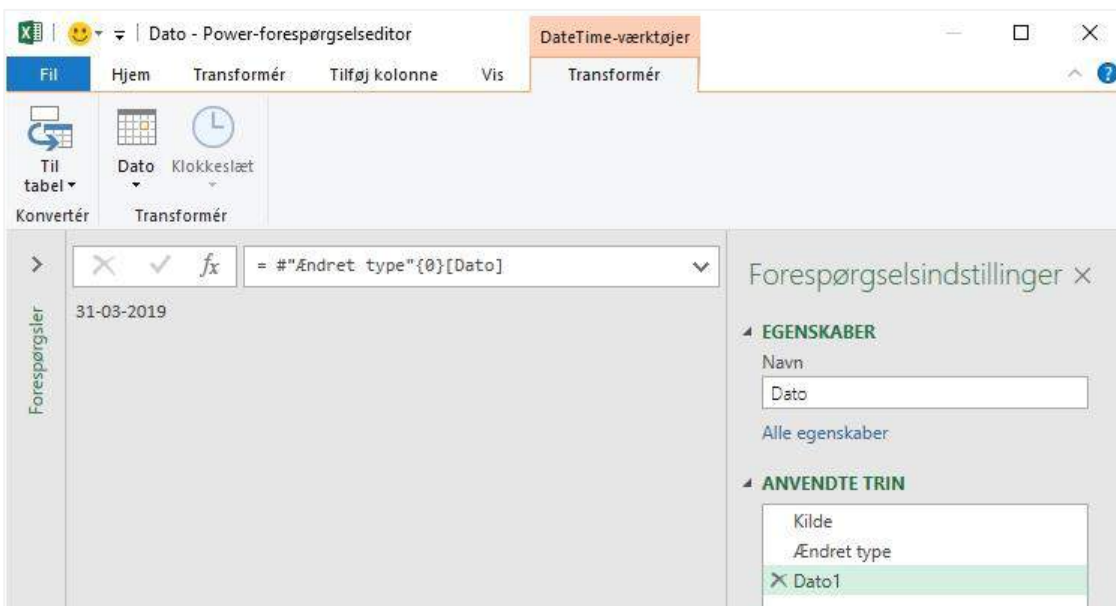




### Oprettelse af datoparameteren

OK, lad os gennemgå de samme trin igen for parameteren Dato. Sammenlignet med den tekstparameter, vi oprettede ovenfor, er der yderligere et trin. I den originale forespørgsel havde kolonnen Dato en dato datatype på det tidspunkt, hvor den blev filtreret, derfor har vi brug for en datatype for parameteren.

Efter at have ændret kolonnen **Dato** til en **datatype**, skal du **højre klikke** på værdien og klikke på **Analyser ned**.



Gør et notat af forespørgselsnavnet, som er Dato i dette scenarie.

Som tidligere skal du klikke på **Luk og indlæs til ...** vælg derefter **Opret kun forbindelse**, og klik på **OK**.

Vi skal nu oprette to parametre, SolgtAf som en teksttype og Dato som en datatype.



## Indsæt parametrene i forespørgslen

Efter at have oprettet parametrene, lad os da bruge dem. For at gøre dette vil vi foretage nogle grundlæggende ændringer af M koden. Vi kunne bruge Avanceret editor eller formellinjen. For at holde tingene enkle bruger jeg formellinjen til dette eksempel.

**Vigtig information: M kode er følsom overfor store og små bogstaver (SolgtAf og solgtaf er ikke ens).**

Åbn den originale forespørgsel (forespørgslen Salgsdata i vores eksempel).

Hvis formellinjen ikke er synlig, skal du klikke på **Vis** → **Formellinje**.

Find det trin, hvor vi hardkodede værdien **Jørgen**.

	A <sup>B</sup> C <sup>B</sup> Kunde	A <sup>B</sup> C <sup>B</sup> Produkt	A <sup>B</sup> C <sup>B</sup> Solgt af	V <sup>2</sup> Værdi
1	Føtex	Kylling	Jørgen	51
2	ICA	Kylling	Jørgen	39
3	Fakta	Rugbrød	Jørgen	80
4	Rema 1000	Rugbrød	Jørgen	67
5	Spar	Æbler	Jørgen	73

Udskift "**Jørgen**" med parameteren **SolgtAf**.

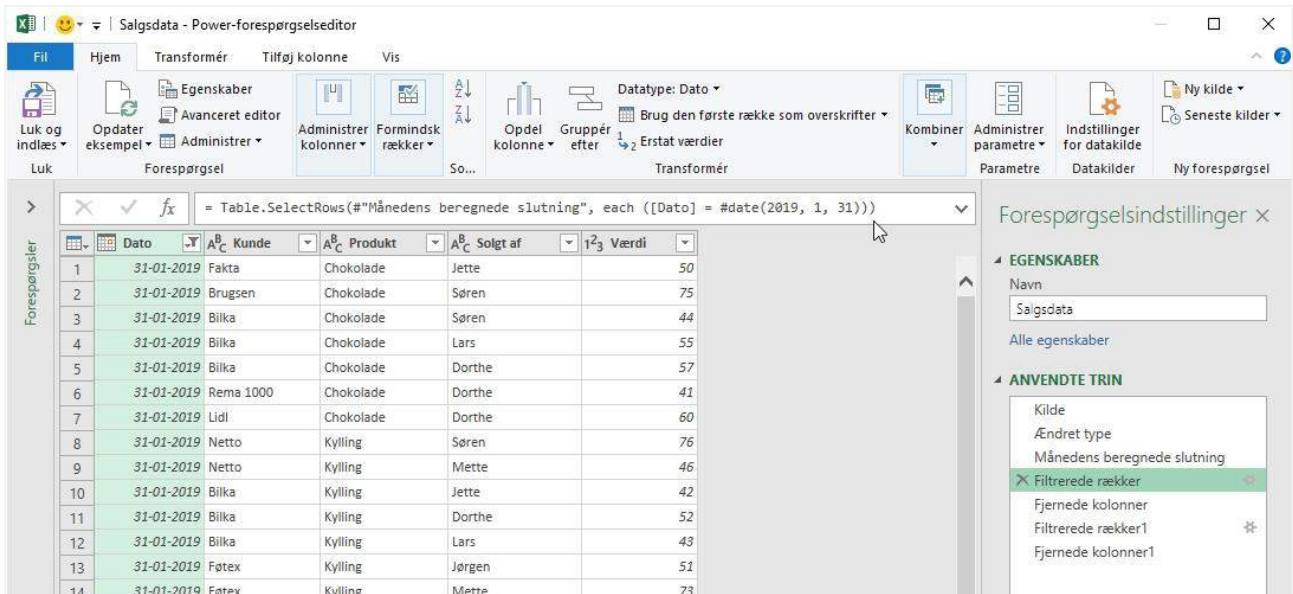
```
= Table.SelectRows(#"Fjernede kolonner", each ([Solgt af] = "Jørgen"))
```

bliver til:

```
= Table.SelectRows(#"Fjernede kolonner", each ([Solgt af] = SolgtAf))
```

Dernæst anvender vi parameteren Dato. Find det trin, hvor vi hardkodede 31. januar 2019 som dato.

## Kapitel 6 Power Query - Brug af parametre



Udskift **#date(2019, 1, 31)** med parameteren **Dato**.

```
= Table.SelectRows(#"Månedens beregnede slutning", each ([Dato] = #date(2019, 1, 31)))
```

bliver til

```
= Table.SelectRows(#"Månedens beregnede slutning", each ([Dato] = Dato))
```

Det er alt, hvad der er, vi har nu anvendt parametrene. Klik på **Hjem** → **Luk og indlæs** for at indlæse ændringerne i Excel.

## Brug af parameteren

Nu, hvor vi er tilbage i Excel, kan vi ændre celleværdierne dato og solgt af og derefter klikke på **Data** → **Opdater alle**.

	A	B	C	D	E	F	G
1	Kunde	Produkt	Værdi				
2	ICA	Rugbrød	59			Dato	
3	Rema 1000	Æbler	72			31-03-2019	
4	Spar	Chokolade	74				
5	Bilka	Rugbrød	73			Solgt af	
6	Bilka	Rugbrød	44			Dorthe	
7	Bilka	Rugbrød	69				
8	Rema 1000	Kylling	79				
9	Lidl	Kylling	36				
10	Fakta	Æbler	79				
11	Rema 1000	Rugbrød	80				
12	Rema 1000	Æbler	70				
13	Bilka	Æbler	59				

Forespørgslen opdateres for at vise værdierne for de parametre, vi har valgt.

## Kapitel 6 Power Query - Brug af parametre

Du kan nu opsætte en hvilken som helst Power Query hardkodet værdi som en parameter. Her er nogle forslag over de mest nyttige ting man kan konfigurere som parametre:

- Filstier til import af eksterne datafiler.
- Perioders slutdatoer til finansielle rapporter.
- Navne på afdelinger eller omkostninger for at oprette rapporter til specifikke områder.
- Eventuelle indstillinger, som en anden bruger sandsynligvis skal ændre.

## Kapitel 7 Grundlæggende transformationer

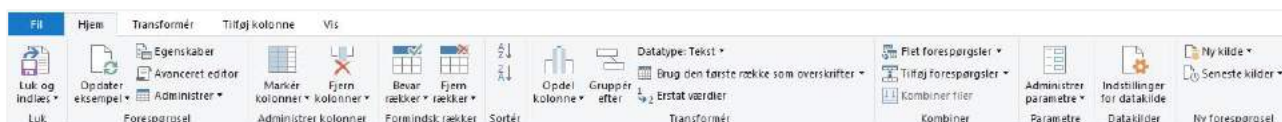
Vi har allerede set på nogle få transformationer i denne Power Query bog, men nu går vi meget længere og ser på de tilgængelige transformationer i de vigtigste Power Query menuer. Vi kan ikke dække alt i dette kapitel; der er bare for mange muligheder. Nogle af de mere detaljerede transformationer vil også blive inkluderet i de næste kapitler i denne bog.

Microsoft har forsøgt at gøre menuerne kontekstuelle; derfor kan de samme handlinger ses flere steder i menuerne. Det er muligt, at du finder andre menuer til anvendelse af de samme indstillinger. Da Power Query kontinuerligt opdateres, er det muligt, at skærbilleder kan forekomme lidt anderledes end din version af Excel. Du har muligvis også ikoner, som jeg ikke har, og omvendt.

Lad os komme i gang med at se, hvad der er tilgængeligt for os.

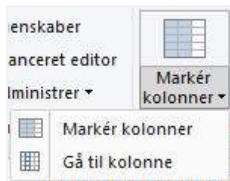
### Fane Hjem

Lad os først se fanen **Hjem**.



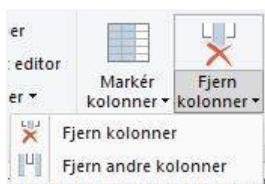
### Administrer kolonner

Afsnittet **Administrer kolonner** indeholder to ikoner, der hver indeholder to yderligere indstillinger.



**Marker kolonner** – markerer kolonnerne, der skal gemmes i eksempelvisningsvinduet.

**Gå til kolonne** – markerer en bestemt kolonne. Dette er ikke rigtig en transformation, da den markerer en kolonne uden at tilføje trin til forespørgslen. Det samme kunne opnås ved at klikke på musen på en kolonneoverskrift.



**Fjern kolonner** - fjerner de valgte kolonner.

**Fjern andre kolonner** - fjerner alle kolonner undtagen de valgte.

Hvis du har læst omhyggeligt, vil du måske undre dig over, hvad der er forskellen mellem Marker kolonner og Fjern andre kolonner. Begge kommandoer beholder de markerede kolonner, så hvad er forskellen? Kort sagt, ingen! Der er ingen forskel, M koden er nøjagtig den samme.

Fra en brugertankeproces er der imidlertid en forskel. Se på disse to sætninger

- "Dette er de kolonner, jeg vil beholde."
- "Jeg vil fjerne alle kolonner undtagen disse."

De er subtilt forskellige i tilgangen, men fører til det samme resultat i Power Query. Så jeg kan forstå, hvorfor begge findes, selvom de giver det samme resultat. Dette illustrerer også, hvordan mange af Power Query menuerne fungerer. De er baseret på, hvad en bruger muligvis prøver at opnå.

### Fjern eller fjern andre - betyder det noget?

Nu hvor jeg har fået dig til at tænke lidt. Her er et spørgsmål til dig... er følgende det samme?

- Marker den første kolonne, og klik på **Fjern andre kolonner**
- Marker alle kolonner undtagen den første og klik på **Fjern kolonner**

Hvad synes du? Er de ens?

Svaret er ... Nej, de er ikke ens. Lad mig forklare hvorfor.

Lad os antage, at vi har en tabel med tre kolonner mærket kolonne1, kolonne2 og kolonne3.

Hvis vi skulle markere kolonne1 og fjerne de andre kolonner. Da ville M koden ser sådan ud:

```
=Table.SelectColumns(Source, {"Column1"})
```

Hvis vi imidlertid skulle vælge kolonne 2 og kolonne 3 og fjerne disse kolonner (dvs. at lade kolonne1 være tilbage), ser M koden sådan ud:

```
=Table.RemoveColumns(Source, {"Column2", "Column3"})
```

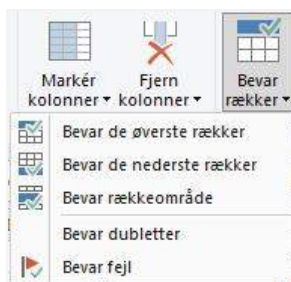
Dette er ikke kun en anden linje med M kode, der opnår det samme resultat. Forestil dig nu, at en ny kolonne føjes til kildedataene. Den nye kolonne kaldes kolonne4. Vi opdaterer forespørgslen som sædvanligt. Hvad sker der?

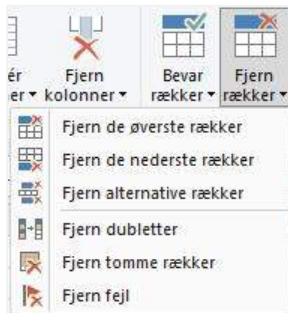
- **Fjern andre kolonner** vil sikre, at kolonne1 er den eneste resterende kolonne (dvs. kolonne4 er også blevet fjernet).
- **Fjern kolonner** fjerner kolonne2 og kolonne3, hvilket betyder, at kolonne1 og kolonne4 er tilbage.

Dette koncept er vigtigt og findes igen og igen i Power Query-indstillingerne. Når du anvender trin, der har både Fjern/Fjern andre versioner, skal du tænke over, hvad der vil ske, hvis nogen tilføjer flere kolonner, dette sikrer, at du vælger den bedste mulighed til dit scenarie.

### Bevar rækker

Valgmulighederne Bevar rækker fjerner et antal datarække ved at bevare eller fjerne rækker, der opfylder specifikke kriterier.





De første tre elementer i begge menuer fører til et andet vindue, som giver os mulighed for at indstille antallet af rækker, der skal holdes eller fjernes.

- Bevar de øverste rækker
- Bevar de nederste rækker
- Bevar rækkeområde
- Fjern de øverste rækker
- Fjern de nederste rækker
- Fjern alternative rækker

Som et eksempel ser dialogboksen til **Bevar rækkeområde** således ud:

The dialog box is titled 'Bevar rækkeområde'. It contains the instruction 'Angiv det rækkeinterval, der skal bevares.' Below this are two input fields: 'Første række' and 'Antal rækker'. At the bottom right, there are 'OK' and 'Annuller' buttons.

Eller **Fjern alternative rækker** ser sådan ud:

The dialog box is titled 'Fjern alternative rækker'. It contains the instruction 'Angiv mønsteret for rækker, der skal fjernes og bevares.' Below this are three input fields: 'Første række, der skal fjernes', 'Antal rækker, der skal fjernes', and 'Antal rækker, der skal bevares'. At the bottom right, there are 'OK' and 'Annuller' buttons.

## Kapitel 7 Power Query – Grundlæggende transformationer

De resterende menupunkter fører ikke til en anden dialogboks:

- Bevar dubletter
- Bevar fejl
- Fjern dubletter
- Fjern tomme rækker
- Fjern fejl

Trinnene for disse tilføjes automatisk uden behov for yderligere input.

### Sorter

**Sorteringsindstillingerne** ligner Excel, men der er en subtil forskel ... hvordan sorteringsrækkefølgen anvendes.



Lad os antage, at vi har nogle data.

	A <sup>B</sup> C Kolonne1	1 <sup>2</sup> 3 Kolonne2	ABC 123 Kolonne3
1	B		1
2	B		2
3	C		3
4	C		4
5	B		5
6	B		6
7	C		7
8	C		8

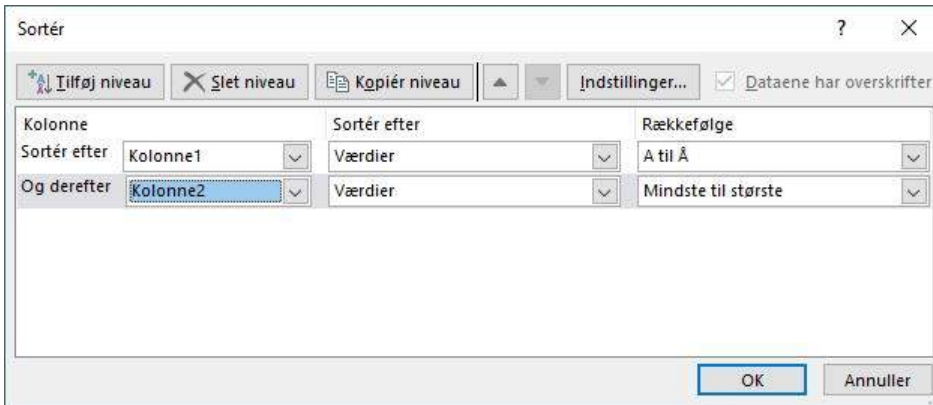
Hvis vi sorterer efter Kolonne1, derefter separat sorterer efter Kolonne2, hvad vil der ske? I Excel, hvis vi anvendte to separate sorteringer, ville det ende med at der bliver sorteret efter Kolonne2, da det var den sidste kolonne, der blev sorteret. Men sådan er det ikke med Power Query.

Hvis du husker det, reducerer Power Query antallet af trin for at være så effektiv som muligt. Hvis du sorterer efter Kolonne1, så efter Kolonne2, vil dette være et enkelt trin, og resultatet vil se sådan ud.

	A <sup>B</sup> C Kolonne1	1 <sup>2</sup> 3 Kolonne2	ABC 123 Kolonne3
1	B		1
2	B		2
3	B		5
4	B		6
5	C		3
6	C		4
7	C		7
8	C		8

Bemærk, at resultatet først sorteres efter Kolonne1, derefter efter Kolonne2. Derfor er Power Query sorterings funktionalitet, den samme som vi finder i Excels dialogboks Sorter.





Hvis der er mindst et trin mellem den første sortering og den anden sortering, kombinerer Power Query dem ikke, da de er separate trin.

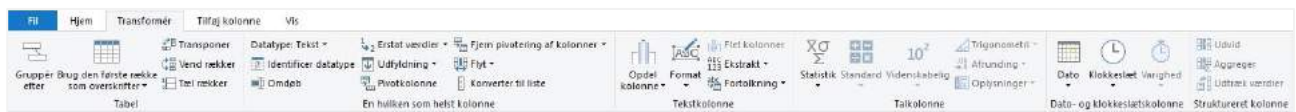
## Andre sektioner under fanen Hjem

Afsnittet **Transformer** under fanen Hjem indeholder dubletter af transformationer, der findes på fanen Transformer. Lidt længere nede kan du læse om sektionen Transformer for at få alle detaljer om disse.

Vi dækker ikke de andre ikoner i fanen Hjem i dette kapitel. Mange af disse vil blive inkluderet detaljeret i de næste kapitler.

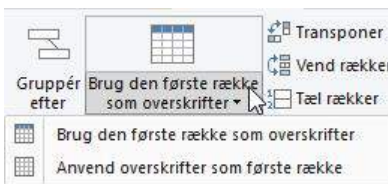
## Transformer

Lad os derefter vende opmærksomheden på fanen **Transformer**.

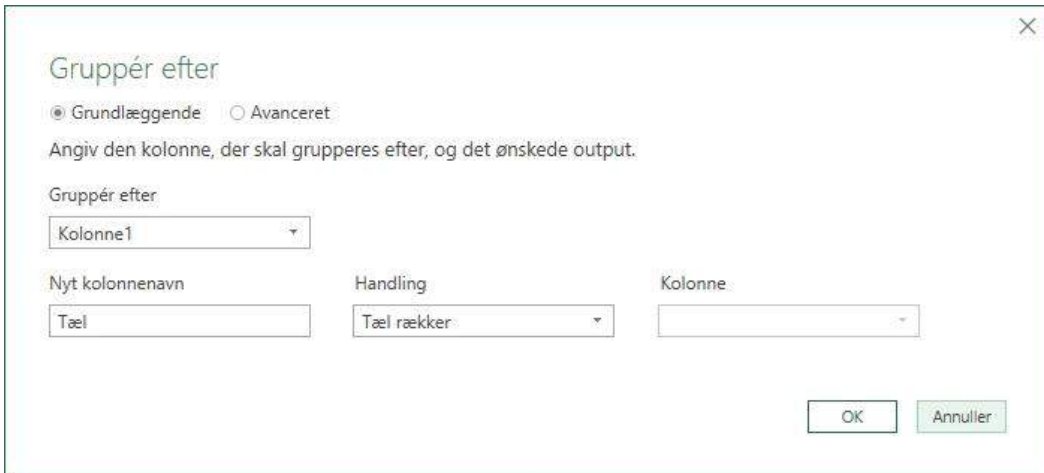


## Tabel

Tabelsektionen indeholder transformationer til hele tabellen, ikke kun for de markerede kolonner.



**Grupper efter** er en måde at opsummere dataene på forskellige måder. Klik på ikonet for at åbne dialogboksen Gruppér efter.



**Brug første række som overskrifter** gør den første datarække til kolonneoverskrifter. Du bruger ofte dette, når du importerer data fra et navneområde, eller når som helst Power Query ikke kan gætte, hvad kolonneoverskrifterne skal være.

**Brug overskrifter som første række** bruges, når Excel gætter forkert, at den første række i en import skal være kolonneoverskrift.

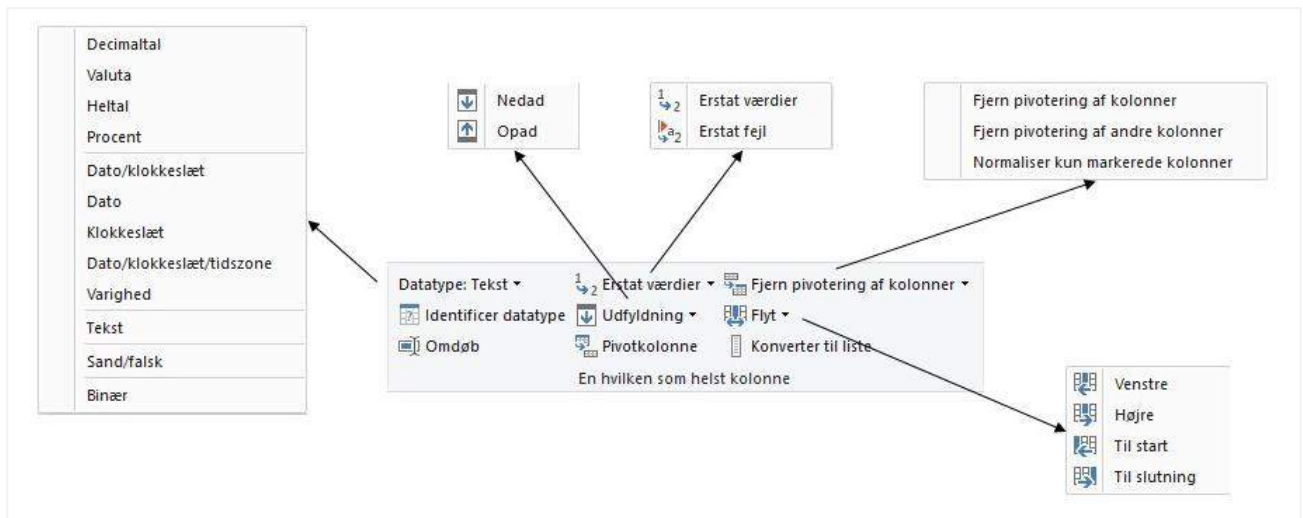
**Transponer** fungerer på lignende måde som Excel-funktionen, det skifter kolonner til rækker og vice versa.

**Vend rækker** vender rækkerne i den modsatte rækkefølge.

**Tæl rækker** er efter min mening ikke en transformation, den graver ned for at finde antallet af rækker.

## En hvilken som helst kolonne

Sektionen **En hvilken som helst kolonne** indeholder masser af nyttige transformationer (de vil blive gennemgået mere detaljeret i senere kapitler).



**Datatype** - brug denne indstilling for at ændre datatypen. Det har det samme resultat som at klikke på datatypeikonet i kolonneoverskriften.

**Identificer datatype** - Power Query prøver at gætte datatypen ved at analysere dataene i kolonnen. Husk, at det ikke altid er det rigtige, så det er ofte bedre at eksplicit vælge den datatype, du ønsker.

**Omdøb** sætter den valgte kolonneoverskriftstekst i redigeringsstilstand, hvilket er det samme som at dobbeltklikke på kolonneoverskriften.

**Erstat værdier** er meget nyttigt; det fungerer som Excels Find og erstat funktion. Klik på ikonet for at åbne dialogboksen Erstat værdier.



**Erstat fejl** fungerer på samme måde som Erstat værdier, men gælder kun for data, der indeholder fejl.

**Udfyld opad/nedad** udfylder alle tomme celle i kolonnen med værdien ovenfor eller nedenfor for.

**Pivotkolonne** vil pivotere dataene baseret på den markerede kolonne. Mere om dette i et senere kapitel.

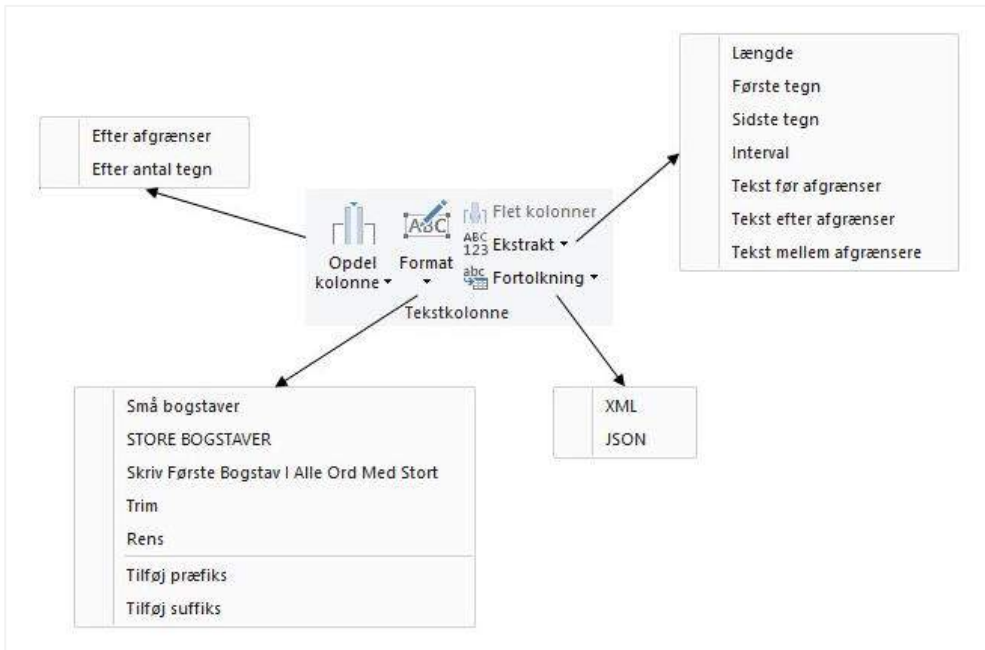
**Fjern pivotering af kolonner** kan bruges til at konvertere information til en flad fil tabel. Mere om dette i et senere kapitel.

**Flyt** bruges til at flytte kolonner til venstre eller højre hen over skærmen, selvom man ofte vil vælge at trækker dem med musen.

**Konverter til liste** konvertere de markerede kolonner til en liste over værdier; dette svarer til at bore ned i en tabel (som vi allerede har set).

## Tekstkolonne

Værktøjerne til at transformere tekst findes i sektionen **Tekstkolonne**.

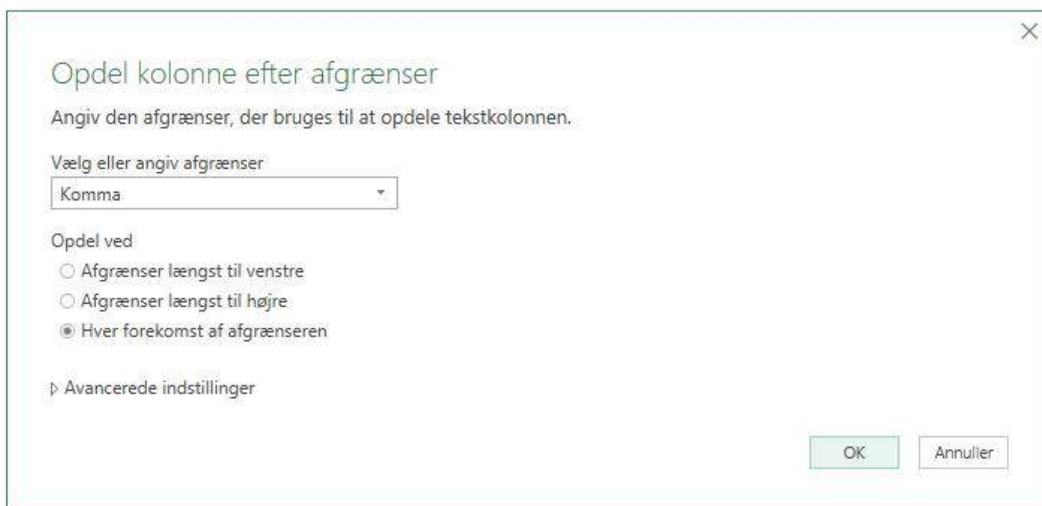


**Opdel kolonne** er nyttigt, når vi vil opdele en del af en streng fra en anden streng. Opdeling skaber yderligere kolonner for hver opdeling af strengen.

Der er mange muligheder for opdeling af tekst. De tre øverste muligheder åbner alle en dialogboks.

- Efter afgrænser
- Efter antal tegn

Som et eksempel ser dialogboksen Opdel kolonne efter afgrænser således ud:



**Format** har indstillinger til:

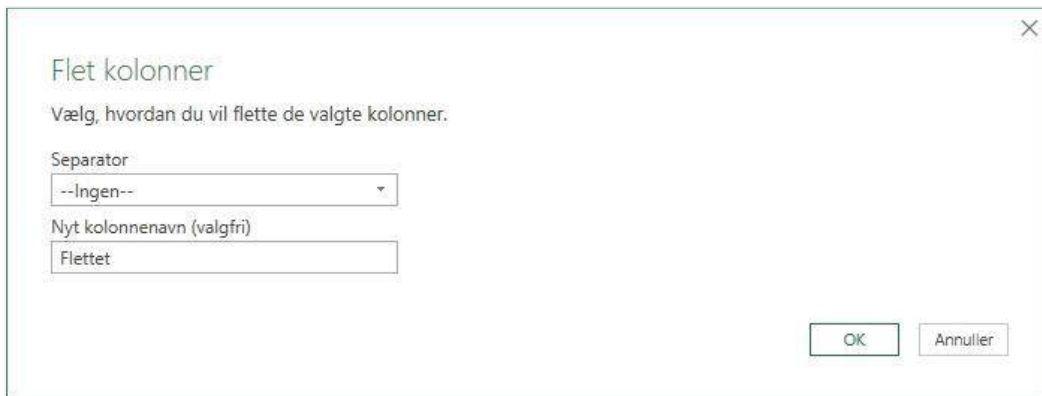
- Ændre hvordan teksten ser ud (f.eks. Store og små bogstaver)
- Trim og Rens for at fjerne mellemrum og tegn, der ikke kan udskrives
- Tilføj præfiks og suffiks tilføjer en fast streng til den eksisterende tekst.

**Ekstrakt** indeholder muligheder for at bevare en del af en streng og kassere andet. Valgmulighederne inkluderer:

- Længde (svarer til funktionen LÆNGDE i Excel)
- Første tegn (svarer til funktionen VENSTRE i Excel)
- Sidste tegn (svarer til funktionen HØJRE i Excel)
- Interval
- Tekst før afgrænsning
- Tekst efter afgrænsning
- Tekst mellem afgrænsere

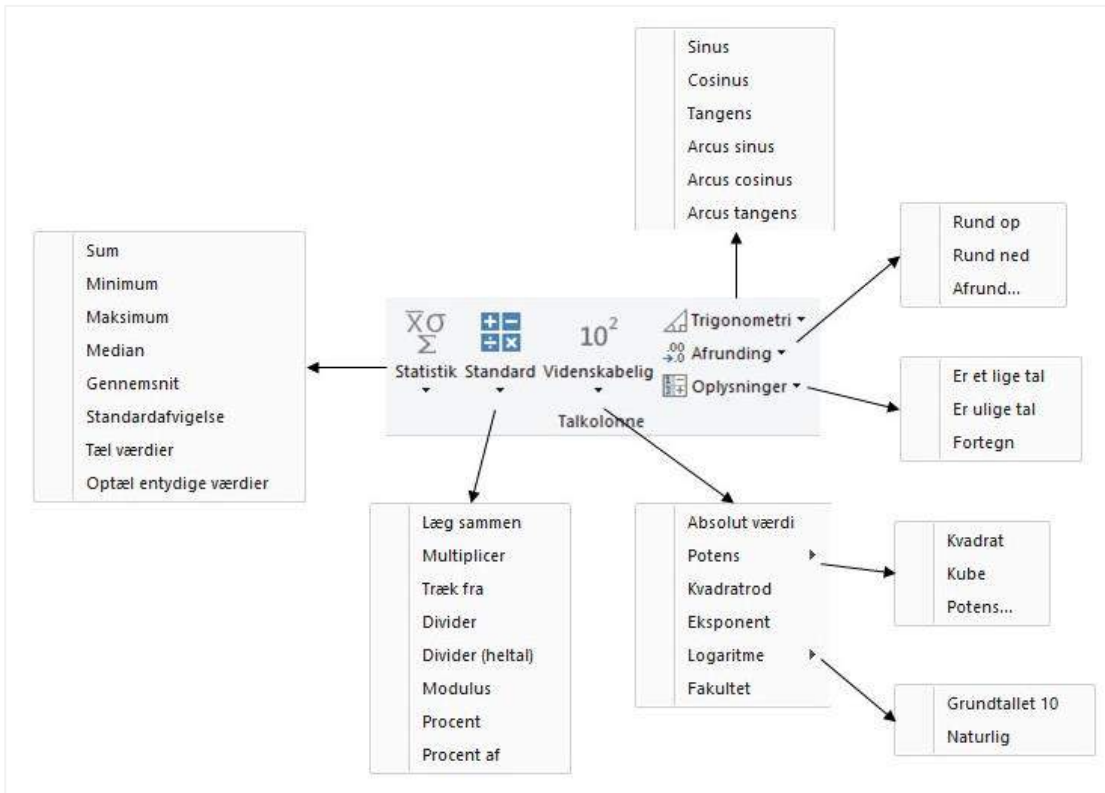
Fortolkning er en avanceret tekstfiltrering til XML og JSON kodet tekst.

**Flet kolonner** kræver, at der markeres to eller flere kolonner. Klik på ikonet for at åbne dialogboksen Flet kolonner, der indeholder yderligere indstillinger.



### Tal kolonne

Hvis du ikke er matematikker, får du sikkert ikke brug for denne sektion.



**Statistik** arbejder med værdierne for at få statistiske oplysninger om dataene, sikkert ikke noget, du får brug for.

**Standard** udfører beregninger på tallene. Som et eksempel, vælges Læg sammen for at åbne dialogboksen Adder. Hvis du indtaster 5 som værdi og klikker på OK, lægges 5 til hvert tal i kolonnen.



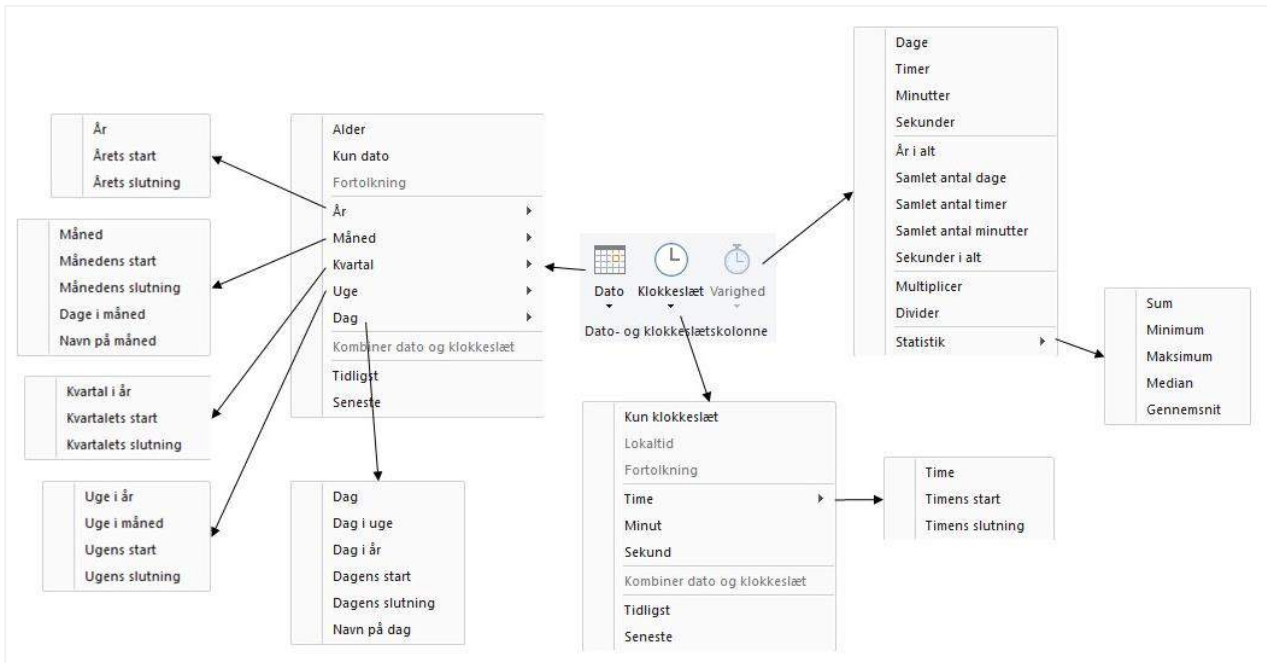
**Afrunding** giver muligheder for Runde op, ned eller til et specifikt antal decimaler.

**Oplysninger** kan bruges til at kontrollere, om værdien er ulige, lige, positiv, negativ eller nul.

**Videnskabelig og trigonometri** - beklager, men dem kan du se på selv. Jeg kan huske, at jeg lærte disse i skolen, men de har længe forladt min hukommelse. Du er velkommen til at eksperimentere, hvad der er det værste, der kan ske.

### Dato og klokkeslæt

Sektionen **Dato og klokkeslætskolonne** indeholder adskillige transformationer til håndtering af datoer, klokkeslæt og varigheder. I Excel er datoer vanskeligt at arbejde med, men i Power Query har vi nu masser af værktøjer til at omdanne vores data til noget mere anvendeligt.



**Dato** indeholder masser af muligheder til at transformere datoer. Nogle af mine favorittransformationer i dette afsnit er:

- Årets start (dato)
- Årets slutning (dato)
- Slut på måned (måned)
- Månedens navn (måned)
- Årets uge (uge)
- Ugedag (dag)

Grundlæggende er det fra enhver dato muligt at generere enhver kategorisering, der kan tænkes.

**Klokkeslæt** indeholder mulighederne for at udtrække timer, minutter, sekund osv. fra en kolonne, der er formateret som klokkeslæt eller dato.

**Varighed** indeholder muligheder for at konvertere varigheder til dage, år, minutter, sekunder osv.

## Tilføj kolonne

### Fanerne Transformer kontra Tilføj kolonne

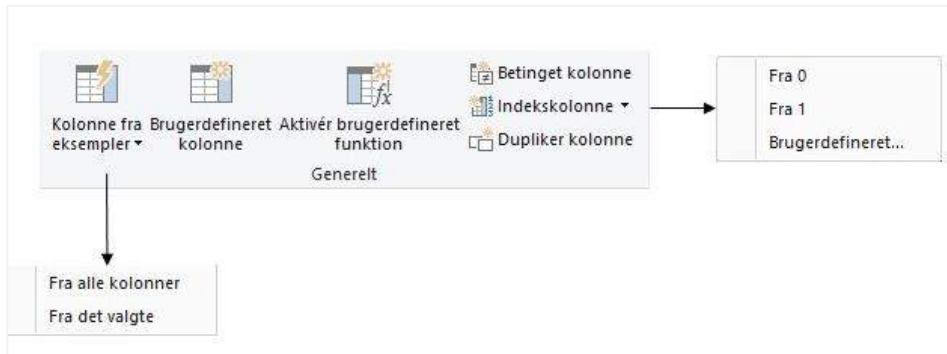
Sammenlign fanerne **Transformer** og **Tilføj kolonne**; der er mange ikoner der lignende og lydende som det samme. Så hvad er forskellen mellem dem?



Valg af et ikon fra fanen **Tilføj kolonne** tilføjer (som du sikkert har gættet) en kolonne. Ikoner på fanen **Transformer** transformerer de valgte kolonner uden at tilføje nye kolonner. For at finde ud af, hvad et ikon gør, skal du læse om det ækvivalente ikon i sektionen **Transformer** ovenfor.

### Generelt

Afsnittet **Generelt** tilbyder mere avanceret funktionalitet.



**Kolonne fra eksempler** bruger Power Querys logik til at identificere mønstre i dataene og anvende dem på de resterende rækker. Eksemplet neden for viser, at ved at vise det ønskede resultat for de første to rækker gætter Power Query mønsteret og anvender dette på de resterende rækker.

	Eksempel	Brugerdefineret
1	LD94 LSK	L 94
2	FR34 LSE	F 34
3	PS78 MMN	P 78
4	DF45 LAE	D 45
5	BN89 ERT	B 89
6	ZS09 DLT	Z 09

**Tilpasset kolonne** gør det muligt for os at bruge M kodeformler på dataene. Vi vil dække nogle grundlæggende versioner af disse i senere kapitel.

**Aktiver brugerdefineret funktion** giver os mulighed for at oprette vores egne Power Query funktioner ud fra eksisterende transformationer.

**Indekskolonne** tilføjer en kolonne med et indeksnummer.

**Duplikatkolonne** opretter en dublet version af den markerede kolonne.

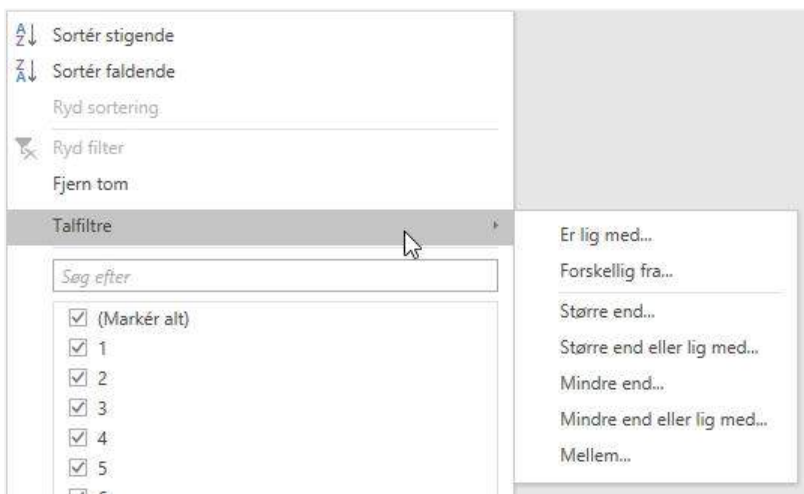
**Betinget kolonne** bruger IF-typelogik til at generere værdier, der opfylder specifikke kriterier.





## Filtrering

En af de mest almindelige transformationstyper er at filtrere en kolonne. Filtermenuen er kontekstuel, så emnerne afhænger af kolonneypen. Skærmbillede herunder viser menuen til en talkolonne.



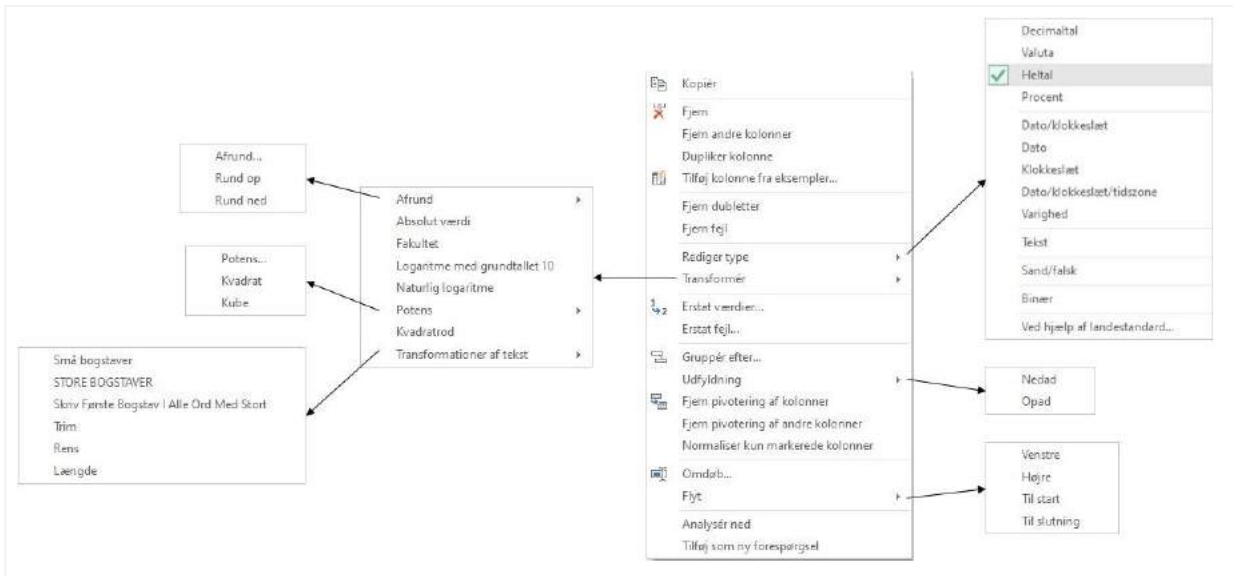
Indstillingerne ligner, men er ikke helt de samme som i Excel. Dette er ikke fordi Power Query mangler funktionalitet, men fordi der er forskellige måder at opnå det samme resultat ved at sortere og formindske rækker.

## Andre menuer

Almindelige transformationer kan også findes via andre menuer.

### Højre klik på kolonneoverskrift

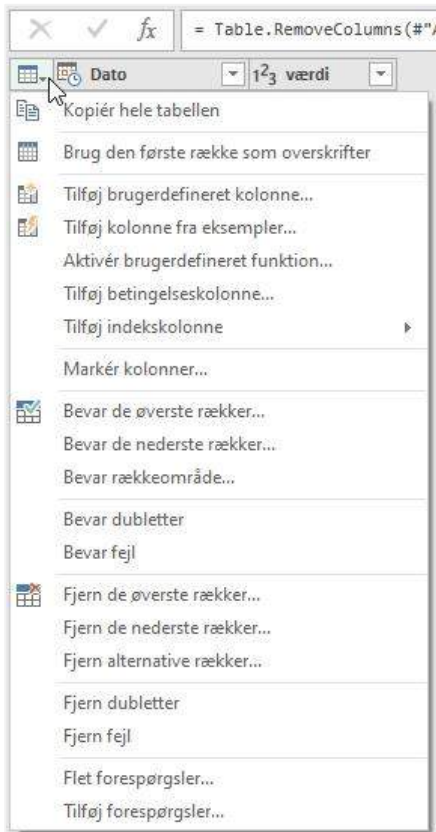
Højre klik på kolonneoverskriften viser en kontekstuel menu (vist nedenfor).



Alle disse er tilgængelige i båndet, men højre klik menuen giver en hurtigere måde at få adgang til mange af transformationerne.

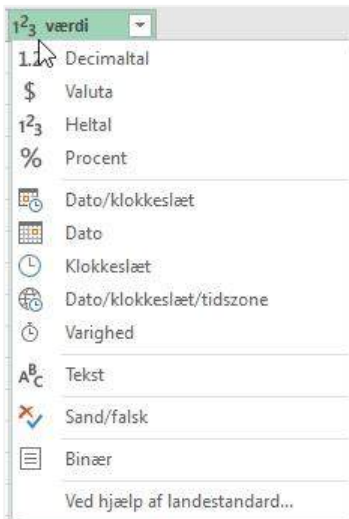
## Tabelikon

Øverst til venstre i tabellen er et lille tabelikon. Klik på det ikon for at se de mest populære tabeltransformationer.



## Skift datatype

Som vi har set før, vises menuen for at ændre datatypen ved at klikke på ikonet til venstre for kolonneoverskriften.



## Næste

Det var ganske meget information, men bare rolig, du behøver ikke at huske det hele. Du skal bare vide, hvad Power Query er i stand til. Når du har brug for det, kan du vende tilbage til dette kapitel.

## Kapitel 8 Kombiner og tilføj forespørgsler

Data kommer ikke altid i en enkelt fil eller fra et download. Vi er ofte nødt til at finde en måde at kombinere flere filer sammen på, så vi kan bruge dem, som om det var en enkelt datakilde. Dette kan være udgiftsrapporter, månedlige dataudtræk, produktprofiler, undersøgelsesdata osv. Grundlæggende gælder det for alle data, der kommer fra flere kilder.

Power Query giver os muligheden for at kombinere forespørgsler eller tilføje en forespørgsel til en anden. Og glem ikke, at vi kan opdatere alle datakilderne med et enkelt klik på Opdater alle. Dette betyder, at vi kan bringe en stopper for kedelige kopier og indsæt rutiner til at kombinere flere filer.

I dette kapitel vil vi kombinere specifikke Excel filer, men fremgangsmåden er den samme for CSV-filer eller andre filtyper. Faktisk kan vi kombinere filer af forskellige typer fra forskellige kilder til en enkelt tabel. Nedenstående eksempel er ganske grundlæggende; det skal vise principperne. Men når du først har forstået principperne, kan du tilpasse dem til dine scenarier.

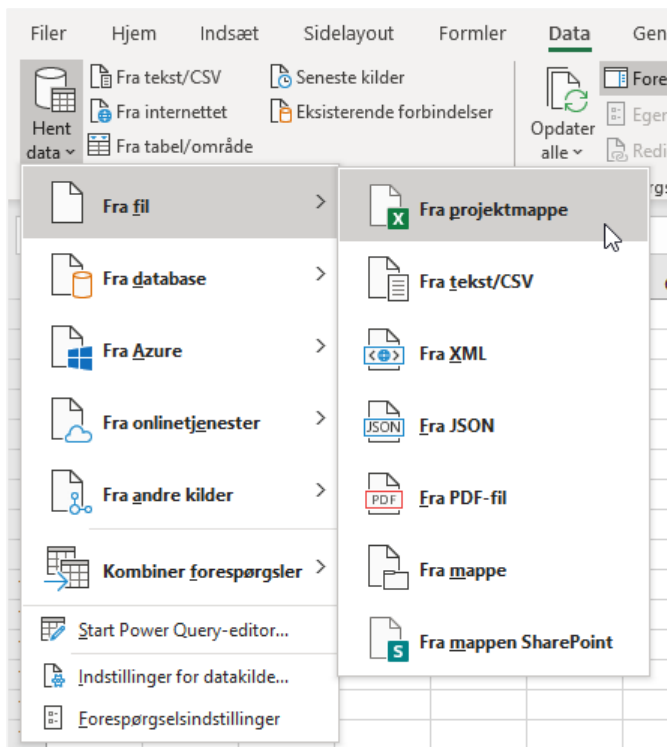
**Eksemplerne i dette kapitel bruger følgende filer Januar 2020.xlsx, Februar 2020.xlsx, Marts 2020.xlsx og April 2020.xlsx.**

Vi vil arbejde igennem alle de vigtige trin fra start til slut, så få Excel startet op, og lad os komme i gang.

### Opret den første forespørgsel

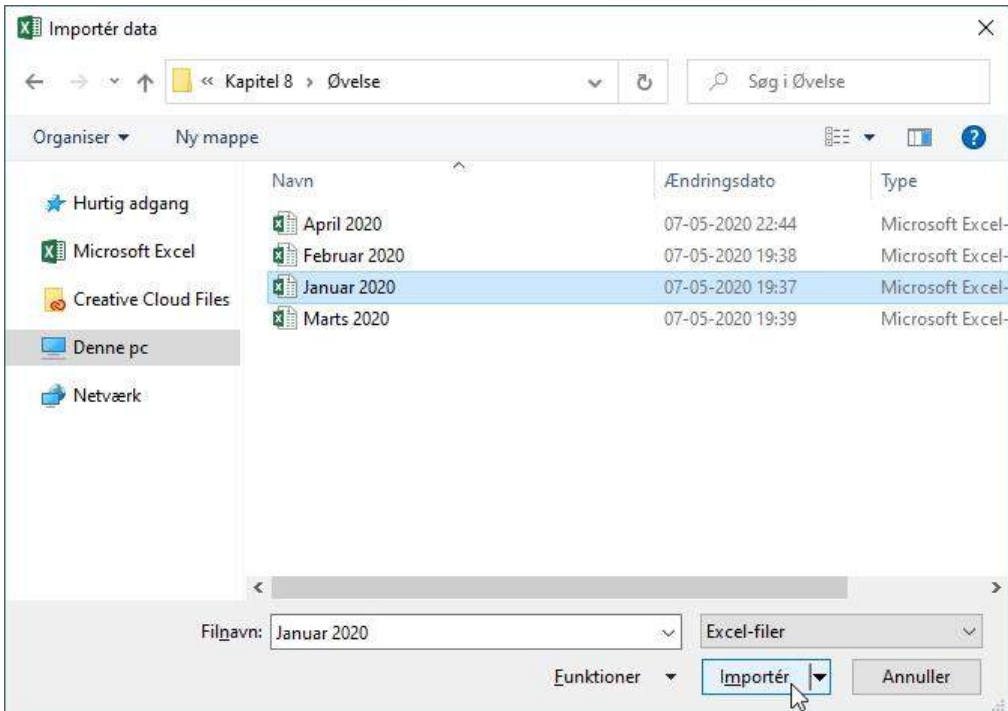
Åbn en ny projektmappe i Excel; dette vil være det sted, hvor de kombinerede data indlæses.

Klik på **Data** → **Hent data** → **Fra fil** → **Fra projektmappe**. Hvis dine data er gemt i et andet filformat, skal du vælge en passende indstilling.

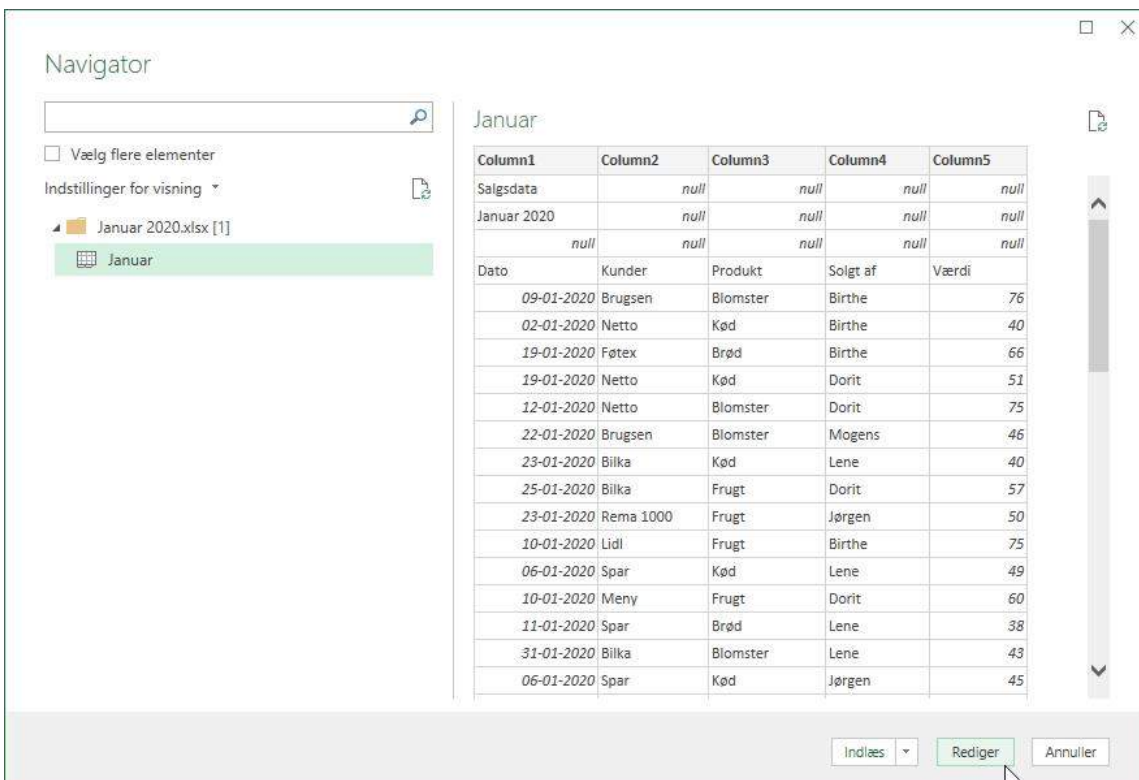


Find filen Januar 2020.xlsx, og klik på **Importer**.

## Kapitel 8 Power Query – Kombiner og tilføj forespørgsler



I dialogboksen **Navigator** skal du vælge det regneark, der indeholder dataene (som er **Januar** regnearket i vores eksempel), og klik derefter på **Rediger**.



Power Query editoren åbnes og viser en forhåndsvisning af dataene.

## Kapitel 8 Power Query – Kombiner og tilføj forespørgsler

	ABC 123 Column1	A <sup>B</sup> <sub>C</sub> Column2	A <sup>B</sup> <sub>C</sub> Column3	A <sup>B</sup> <sub>C</sub> Column4	ABC 123 Column5
1	Salgsdata	null	null	null	null
2	Januar 2020	null	null	null	null
3	null	null	null	null	null
4	Dato	Kunder	Produkt	Solgt af	Værdi
5	09-01-2020	Brugsen	Blomster	Birthe	76
6	02-01-2020	Netto	Kød	Birthe	40
7	19-01-2020	Føtex	Brød	Birthe	66
8	19-01-2020	Netto	Kød	Dorit	51
9	12-01-2020	Netto	Blomster	Dorit	75

Der er et par enkelte transformationer, vi er nødt til for at få det omdannet til et anvendeligt format.

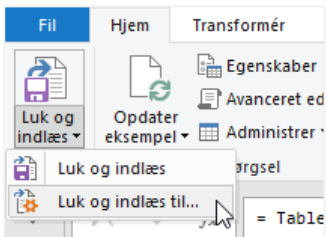
1. Fjern de tre øverste rækker ved at klikke på **Hjem** → **Fjern rækker** → **Fjern de øverste rækker**. Indtast 3 i dialogboksen Fjern de øverste rækker, og klik på **OK**.
2. Fremhæv den første række med data til overskrifter ved at klikke på **Transformer** → **Brug første række som overskrifter**.
3. Tilføj en kolonne med måneds slutdato ved at klikke på **Tilføj kolonne** → **Dato** → **Måned** → **Månedens slutning**. Du skal sikre dig at det er Dato kolonnen du bruger, når du tilføjer denne kolonne.
4. Tilføj en kolonne, der indeholder ugenummeret. Klik på **Tilføj kolonne** → **Dato** → **Uge** → **Uge i år**. Du skal sikre dig at det er Dato kolonnen du bruger, når du tilføjer denne kolonne.
5. Omdøb kolonnen Uge i år til Uge.
6. Flyt kolonnen Månedens slutning til start ved at **højre klikke** på kolonneoverskriften på Månedens slutning, vælg **Flyt** → **Til start** i menuen.
7. Kontroller, at følgende datatyper er anvendt, hvis ikke, gør det da manuelt.
  - Månedens slutning = Dato
  - Dato = Dato
  - Kunder = Tekst
  - Produkt = tekst
  - Solgt af = Tekst
  - Værdi = Heltal
  - Uge = Heltal

Eksempelvinduet skal se sådan ud:

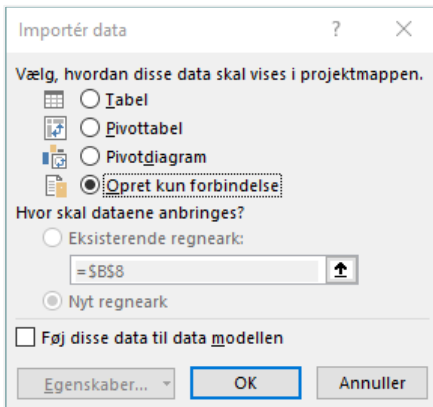
	Månedens slutning	Dato	A <sup>B</sup> <sub>C</sub> Kunder	A <sup>B</sup> <sub>C</sub> Produkt	A <sup>B</sup> <sub>C</sub> Solgt af	1 <sup>2</sup> <sub>3</sub> Værdi	1 <sup>2</sup> <sub>3</sub> Uge i år
1	31-01-2020	09-01-2020	Brugsen	Blomster	Birthe	76	2
2	31-01-2020	02-01-2020	Netto	Kød	Birthe	40	1
3	31-01-2020	19-01-2020	Føtex	Brød	Birthe	66	3
4	31-01-2020	19-01-2020	Netto	Kød	Dorit	51	3
5	31-01-2020	12-01-2020	Netto	Blomster	Dorit	75	2
6	31-01-2020	22-01-2020	Brugsen	Blomster	Mogens	46	4
7	31-01-2020	23-01-2020	Bilka	Kød	Lene	40	4
8	31-01-2020	25-01-2020	Bilka	Frugt	Dorit	57	4
9	31-01-2020	23-01-2020	Rema 1000	Frugt	Jørgen	50	4

## Kapitel 8 Power Query – Kombiner og tilføj forespørgsler

Det er alle de transformationer, vi har brug for. Klik på **Hjem** → **Luk og indlæs til ...**



Vælg **Opret kun forbindelse** i dialogboksen **Importer data**, og klik derefter på **OK**.



Panelet **Projektmappe-forespørgsler** viser forespørgslen Januar 2020 (klik på **Data** → **Forespørgsler og forbindelser**, hvis dette vindue ikke er synligt).

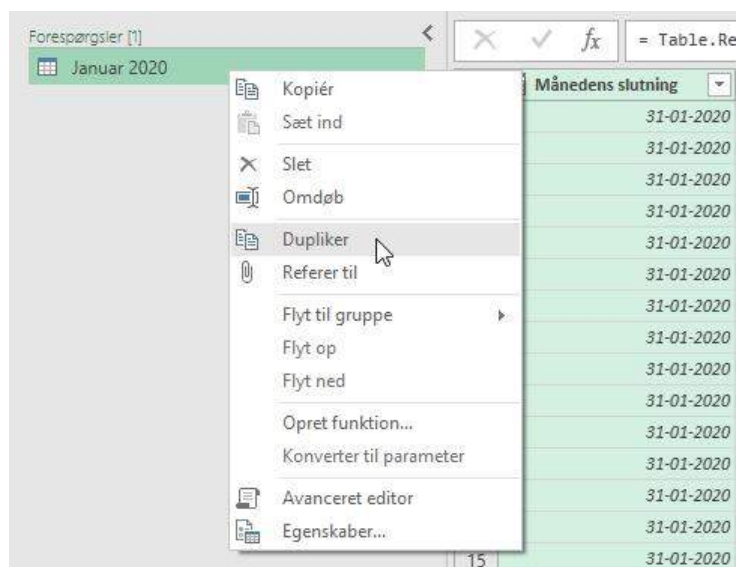


## Kopier og rediger forespørgslen

I vores eksempel ønsker vi ikke at gennemgå de samme transformationstrin igen. I stedet kopierer vi forespørgslen for januar og ændrer derefter trinnene, så den arbejder sammen med filen Februar 2020.xlsx.

Åbn Power Query editoren ved at **dobbelklikke** på forespørgslen januar i panelet **Projektmappe-forespørgsler**. Udvid forespørgselsnavigationsafsnittet (til venstre for eksempelvisningsvinduet), **højre klik** på forespørgslen januar, og vælg **Dupliker**.

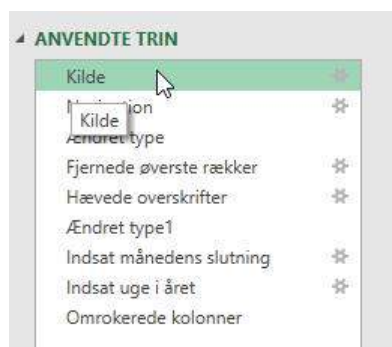
## Kapitel 8 Power Query – Kombiner og tilføj forespørgsler



Omdøb den nye forespørgsel til Februar.



De første to trin i vores eksempel har januar som hardkodede værdier i M koden, så vi skal redigere disse trin. Vi er nødt til at ændre referencer fra januar til februar. Hvordan du redigerer disse er op til dig. Klik enten på indstillingsikonet ved siden af trinnet, eller rediger M koden direkte i formellinjen eller den avancerede editor.



Med hensyn til M koden er ændringerne som følger:

Kilde:

```
= Excel.Workbook(File.Contents("C:\Power Query\Januar 2020.xlsx"), null, true)
```

Bliver til:

```
= Excel.Workbook(File.Contents("C:\Power Query\Februar 2020.xlsx"), null, true)
```



## Kapitel 8 Power Query – Kombiner og tilføj forespørgsler

Navigation:

```
= Kilde{[Item="Januar 2020",Kind="Sheet"]} [Data]
```

Bliver til:

```
= Kilde{[Item="Februar 2020",Kind="Sheet"]} [Data]
```

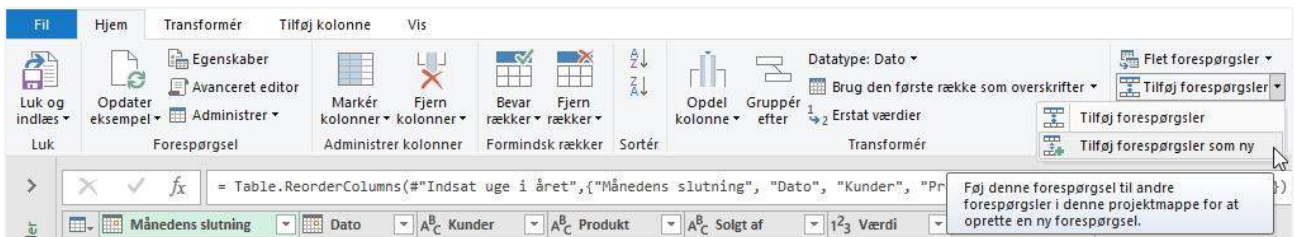
Det var det. Klik på **Luk og indlæs til ...** for at indlæse forespørgslen februar som **Opret kun forbindelse**.

Projektmappe-forespørgsler viser nu begge forespørgslerne januar og februar.



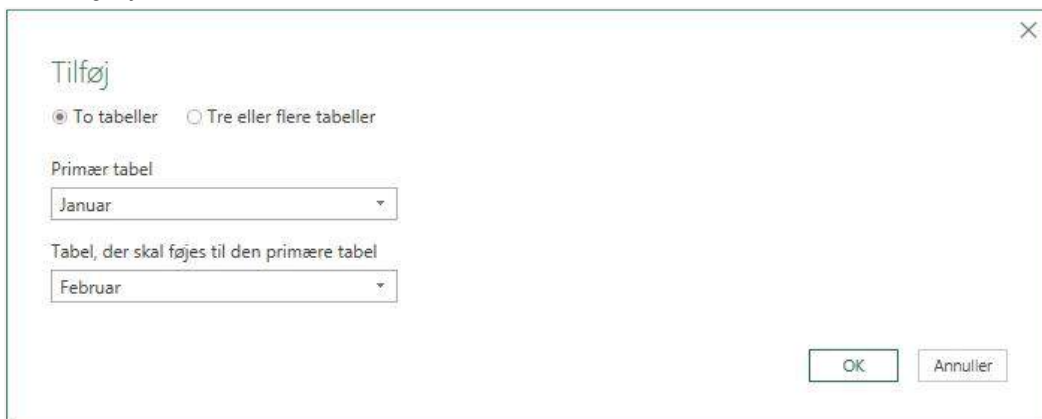
## Kombiner forespørgsler

Kombination eller tilføjelse af forespørgsler er rimeligt ligetil. Åbn Power Query editoren. Klik på **Hjem** → **Tilføj forespørgsler** → **Tilføj forespørgsler som ny**.



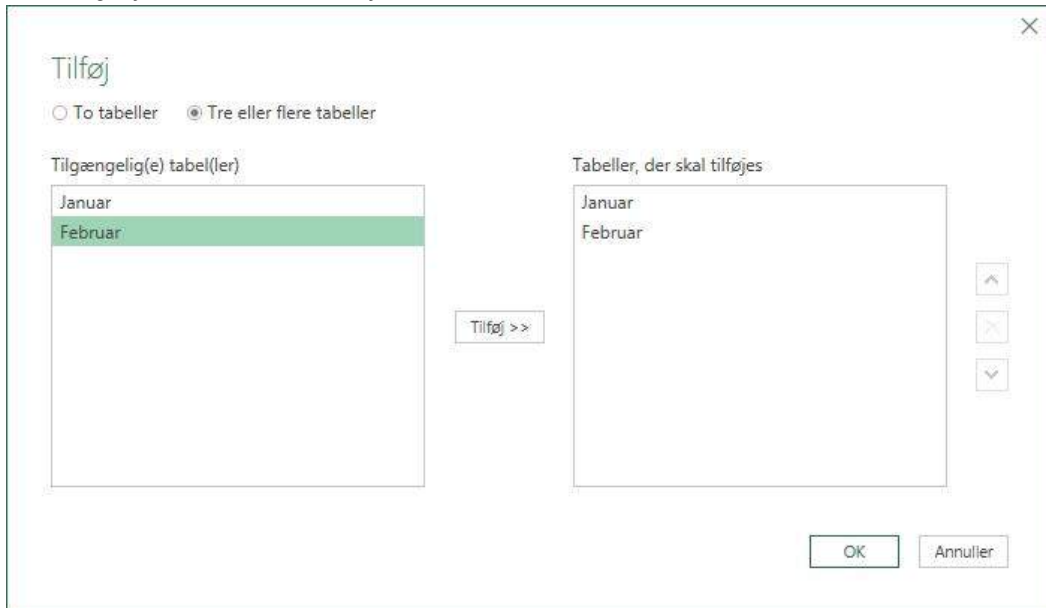
Dialogboksen **Tilføj** åbnes. Der er to visninger, visningen for kombination af to forespørgsler eller visningen for kombination af tre eller flere forespørgsler. Begge er forholdsvis nemme at bruge.

*Indstillinger for kombiner To tabeller*



## Kapitel 8 Power Query – Kombiner og tilføj forespørgsler

Indstillinger for kombiner Tre eller flere tabeller



Når du har klikket på **OK**, oprettes en ny forespørgsel, der sandsynligvis får navnet Append1. Omdøb den kombinerede forespørgsel til noget meningsfuldt.

Klik på **Luk og indlæs** for at indlæse forespørgslen i et nyt regneark.

Se på den nye tabel i Excel. Den indeholder data for både januar og februar. Jeg har tilføjet en pivottabel, der viser værdien efter uge.

	A	B	C	D	E	F	G	H	I	J
1	Månedens slutning	Dato	Kunder	Produkt	Solgt af	Værdi	Uge i år		Rækkenavn	Sum af Værdi
2	31-01-2020	09-01-2020	Brugsen	Blomster	Birthe	76	2		1	308
3	31-01-2020	02-01-2020	Netto	Kød	Birthe	40	1		2	885
4	31-01-2020	19-01-2020	Føtex	Brød	Birthe	66	3		3	807
5	31-01-2020	19-01-2020	Netto	Kød	Dorit	51	3		4	701
6	31-01-2020	12-01-2020	Netto	Blomster	Dorit	75	2		5	435
7	31-01-2020	22-01-2020	Brugsen	Blomster	Mogens	46	4		6	872
8	31-01-2020	23-01-2020	Bilka	Kød	Lene	40	4		7	878
9	31-01-2020	25-01-2020	Bilka	Frugt	Dorit	57	4		8	472
10	31-01-2020	23-01-2020	Rema 1000	Frugt	Jørgen	50	4		9	505
11	31-01-2020	10-01-2020	Lidl	Frugt	Birthe	75	2		<b>Hovedtotal</b>	<b>5863</b>
12	31-01-2020	06-01-2020	Spar	Kød	Lene	49	2			
13	31-01-2020	10-01-2020	Meny	Frugt	Dorit	60	2			
14	31-01-2020	11-01-2020	Spar	Brød	Lene	38	2			

Der er nu tre forespørgsler i projektmappen: Januar, Februar og Kombinerede salgsdata.



## Kapitel 8 Power Query – Kombiner og tilføj forespørgsler

Prøv derefter at tilføje filen Marts 2020.xlsx på egen hånd. Følg de samme trin som ovenfor for at oprette forbindelsen, rediger derefter den kombinerede forespørgsel for at inkludere og data for marts. Du kan bruge indstillingen tre eller flere tabeller i vinduet Tilføj.

Data og pivottabel ser nu sådan ud med når de ekstra uger fra filen Marts 2020 er tilføjet.

	A	B	C	D	E	F	G	H	I	J
1	Månedens slutning	Dato	Kunder	Produkt	Solgt af	Værdi	Uge i år		Rækkenavn	Sum af Værdi
2	31-01-2020	09-01-2020	Brugsen	Blomster	Birthe	76	2		1	308
3	31-01-2020	02-01-2020	Netto	Kød	Birthe	40	1		2	885
4	31-01-2020	19-01-2020	Føtex	Brød	Birthe	66	3		3	807
5	31-01-2020	19-01-2020	Netto	Kød	Dorit	51	3		4	701
6	31-01-2020	12-01-2020	Netto	Blomster	Dorit	75	2		5	435
7	31-01-2020	22-01-2020	Brugsen	Blomster	Mogens	46	4		6	872
8	31-01-2020	23-01-2020	Bilka	Kød	Lene	40	4		7	878
9	31-01-2020	25-01-2020	Bilka	Frugt	Dorit	57	4		8	472
10	31-01-2020	23-01-2020	Rema 1000	Frugt	Jørgen	50	4		9	668
11	31-01-2020	10-01-2020	Lidl	Frugt	Birthe	75	2		13	583
12	31-01-2020	06-01-2020	Spar	Kød	Lene	49	2		12	575
13	31-01-2020	10-01-2020	Meny	Frugt	Dorit	60	2		10	866
14	31-01-2020	11-01-2020	Spar	Brød	Lene	38	2		11	748
15	31-01-2020	31-01-2020	Bilka	Blomster	Lene	43	5		<b>Hovedtotal</b>	<b>8798</b>
16	31-01-2020	06-01-2020	Spar	Kød	Jørgen	45	2			

## Når kolonneoverskrifterne er forskellige

Filen April.xlsx er lidt anderledes end de andre. Kolonneoverskriften Kunder er ændret til Solgt til.

Hvis vi følger de samme trin som ovenfor, går forespørgslen i stykker i alle trin, hvor kolonneoverskriften bruges i transformationerne. I vores eksempel går forespørgslen først i stykker, når du ændrer datatypen.



Power Query kan ikke finde en overskrift der hedder Kunder, da den ikke kaldes Solgt til i kildefilen. Der er to trin, der refererer til den specifikke kolonneoverskrift. Uden disse trin ville vi ikke have vidst, at der var et problem.



For at illustrere det skal du ændre koden for hvert trin, så Kunder bliver til Solgt til. Luk og indlæs derefter forespørgslen April som kun en forbindelse.

Følg derefter de samme instruktioner som ovenfor for at tilføje forespørgslen april til den kombinerede forespørgsel.

## Kapitel 8 Power Query – Kombiner og tilføj forespørgsler

Se på skærbilledet nedenfor. Kolonnerne Kunder og Solgt til var begge den tredje kolonne i forespørgslen, men alligevel er de ikke blevet kombineret. Dette er vigtigt at bemærke. Power Query bruger kolonneoverskrifterne til at bestemme, hvilke kolonner der skal kombineres, ikke placeringen i forespørgslen.

	Månedens slutning	Dato	A <sub>C</sub> <sup>B</sup> Kunder	A <sub>C</sub> <sup>B</sup> Produkt	A <sub>C</sub> <sup>B</sup> Solgt af	1 <sub>2</sub> <sup>3</sup> Værdi	1 <sub>2</sub> <sup>3</sup> Uge i år	A <sub>C</sub> <sup>B</sup> Solgt til	
141	31-03-2020	27-03-2020	Lidl	Grøntsager	Dorit		38	13	null
142	31-03-2020	14-03-2020	Rema 1000	Brød	Lene		70	11	null
143	31-03-2020	11-03-2020	Netto	Blomster	Jørgen		73	11	null
144	31-03-2020	22-03-2020	ICA	Frugt	Jens		53	12	null
145	31-03-2020	27-03-2020	Brugsen	Frugt	Lene		46	13	null
146	31-03-2020	26-03-2020	ICA	Kød	Birthe		49	13	null
147	31-03-2020	16-03-2020	Netto	Frugt	Mogens		80	12	null
148	31-03-2020	16-03-2020	Føtex	Grøntsager	Birthe		54	12	null
149	31-03-2020	08-03-2020	Rema 1000	Kød	Dorit		57	10	null
150	31-03-2020	11-03-2020	Rema 1000	Kød	Birthe		46	11	null
151	30-04-2020	26-04-2020	null	Frugt	Dorit		76	17	Rema 1000
152	30-04-2020	18-04-2020	null	Frugt	Jens		74	16	Spar
153	30-04-2020	05-04-2020	null	Kød	Birthe		47	14	ICA
154	30-04-2020	11-04-2020	null	Blomster	Dorit		78	15	Netto
155	30-04-2020	23-04-2020	null	Frugt	Dorit		38	17	Føtex

For at få denne kombination til at fungere korrekt, kunne vi tilføje et trin i forespørgslen April for at ændre kolonneoverskriften Solgt til til kunder. Dette ville give ensartede overskrifter og give kolonnerne mulighed for at blive kombineret korrekt.

## Kombination af filer med forskellige typer/datastrukturer

Vi så ovenfor, at når man kombinerer forespørgsler, kombineres kolonner med den samme overskrift til den samme kolonne. Derfor kan vi kombinere filer fra mange forskellige kilder i en forespørgsel. Forestil dig som eksempel, at du har data fra et regnskabssystem og ti filer fra forskellige områder, der skitserer deres budget for det næste år. Dette er ikke noget problem for Power Query. Opret forskellige forespørgsler til hver kildefil, som alle skal ende med de samme kolonner, og kombiner derefter forespørgsler til en enkelt tabel.

Dernæst skal vi se på, hvordan vi kan automatisere endnu mere af dette ved at kombinere alle filerne i en mappe.

## Kapitel 9 Power Query – Importer alle filer i en mappe

I dette kapitel ser vi på hvordan vi kan bruge Power Query til at importere filerne i en mappe. Vi giver Power Query en mappesti, klikker på et par knapper, og den vil importere og kombinere alle filerne i en enkelt tabel. Det er imponerende, ikke sandt! Hvis vi vil tilføje en anden fil til outputtabellen, behøver vi kun at gemme en kopi af filen i mappen og klikke på Opdater, og den nye fil vil også blive importeret. Dette kan spare dig meget tid i løbet af dagen.

Inden vi ser nærmere denne teknik, er der et punkt, som jeg er nødt til at gøre dig opmærksom på. Alle de filer, der skal importeres, skal have den samme struktur og kolonneopstilling. Power Query er fantastisk, men du skal give den en chance.

### Opsætning af filerne til eksemplerne

For at arbejde med eksemplerne i denne opgave skal du starte med at flytte januar og februar filerne til en separat mappe. Det er de filer, du skal importere.



Tag ikke marts og april filerne med, vi tilføjer dem senere som en del af eksemplerne.

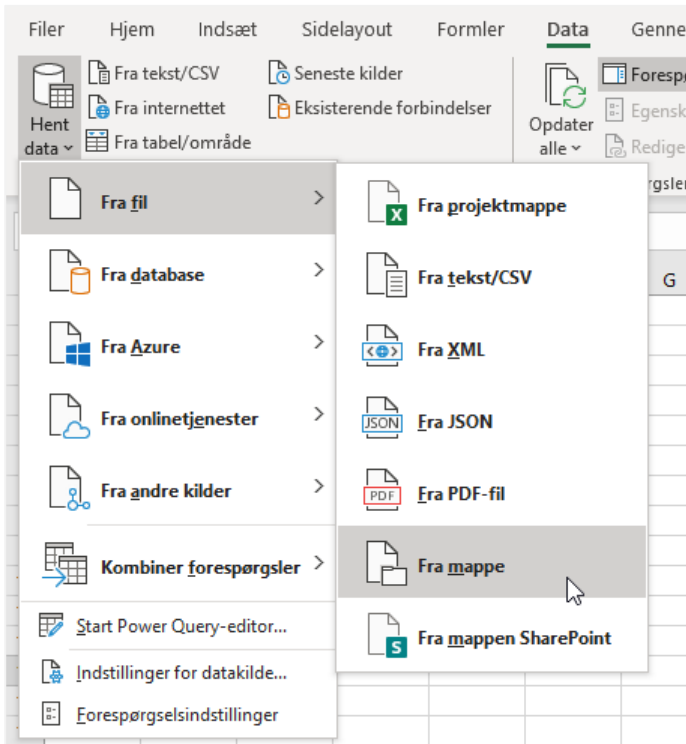
### Importer alle CSV-filer fra en mappe

De nemmeste filer at arbejde med er CSV eller tekstfiler, så det er et godt sted at starte. Hvad er det der gør dem nemme at arbejde med?

- De indeholder altid et enkelt regneark.
- Den første række behandles som overskriftsrække.
- Flettede celler er ikke tilladt.

I Excel skal du vælge **Data** → **Hent data** → **Fra fil** → **Fra mappe**.

## Kapitel 9 Power Query - Importer alle filer i en mappe

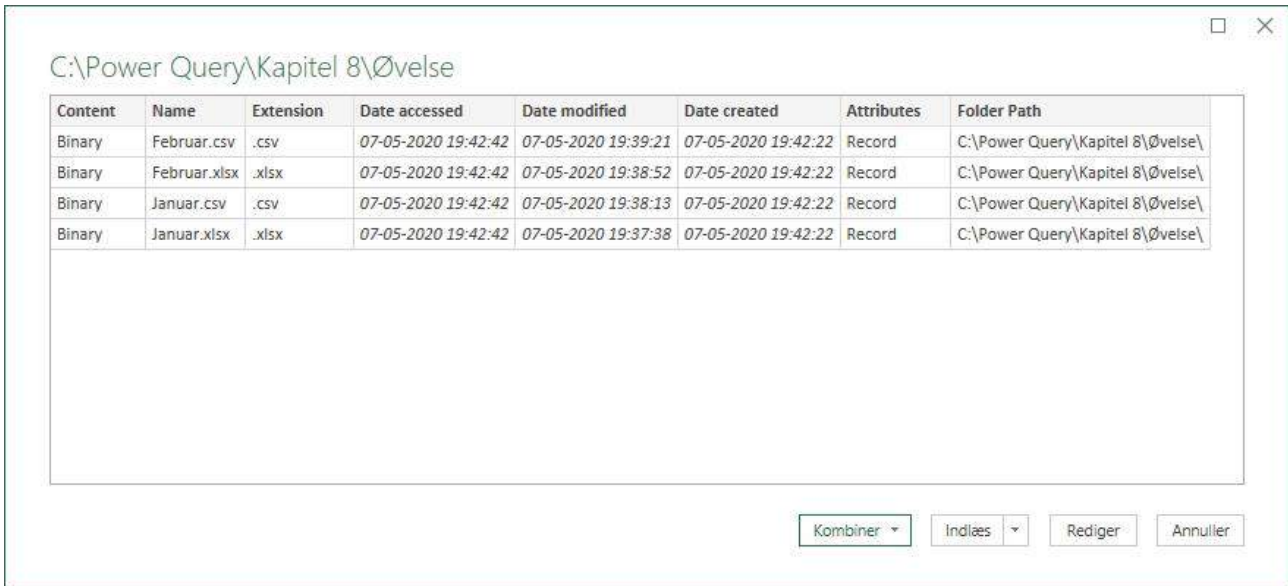


Dialogboksen Mappe åbnes. Klik på **Gennemse**, og naviger frem til den mappe, der skal bruges, og klik derefter på **OK**.



Der vises en forhåndsvisning af mappen og filattributterne. Hvis filerne er helt ens, og der ikke er nogen redigeringer, der kan udføres, kan du bare klikke på **Kombiner (listefeltet) → Kombiner Luk og indlæs**. Du vil dog bemærke, at vi har en blanding af .csv og .xlsx filer i mappen; vi kan derfor ikke kombinere disse med det samme. I stedet vil vi klikke på **Rediger** og se nærmere på hvad der sker.

## Kapitel 9 Power Query - Importer alle filer i en mappe

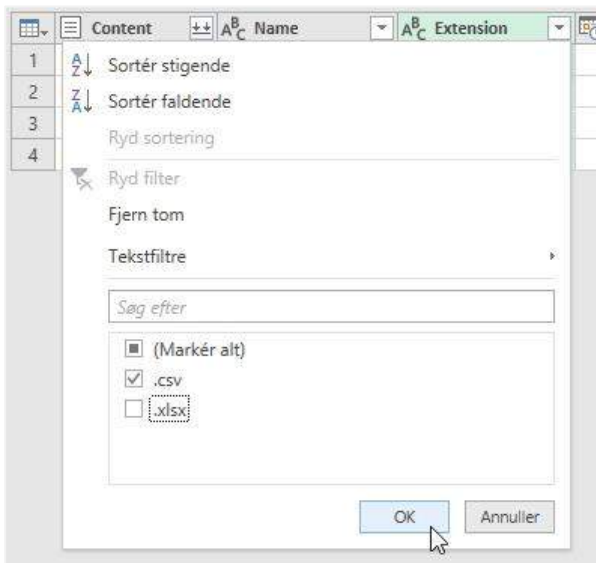


Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
Binary	Februar.csv	.csv	07-05-2020 19:42:42	07-05-2020 19:39:21	07-05-2020 19:42:22	Record	C:\Power Query\Kapitel 8\Øvelse\
Binary	Februar.xlsx	.xlsx	07-05-2020 19:42:42	07-05-2020 19:38:52	07-05-2020 19:42:22	Record	C:\Power Query\Kapitel 8\Øvelse\
Binary	Januar.csv	.csv	07-05-2020 19:42:42	07-05-2020 19:38:13	07-05-2020 19:42:22	Record	C:\Power Query\Kapitel 8\Øvelse\
Binary	Januar.xlsx	.xlsx	07-05-2020 19:42:42	07-05-2020 19:37:38	07-05-2020 19:42:22	Record	C:\Power Query\Kapitel 8\Øvelse\

Forespørgselseditoren åbnes. Den viser ikke dataene fra filen, som den tidligere har gjort, i stedet viser den data om filerne.

Vores første transformation vil være kun at beholde de filer, vi ønsker at kombinere (CSV-filer i vores nuværende eksempel). Det er altid en god ide at filtrere for kun at inkludere de filer, vi har brug for; du ved aldrig, hvornår en anden bruger måske beslutter sig for at gemme en tilfældig fil i mappen.

Filtrer kolonnen Extension så den kun at omfatte .csv.



Herfra har vi to muligheder:

- Brug en brugerdefineret kolonne
- Kombiner filerne

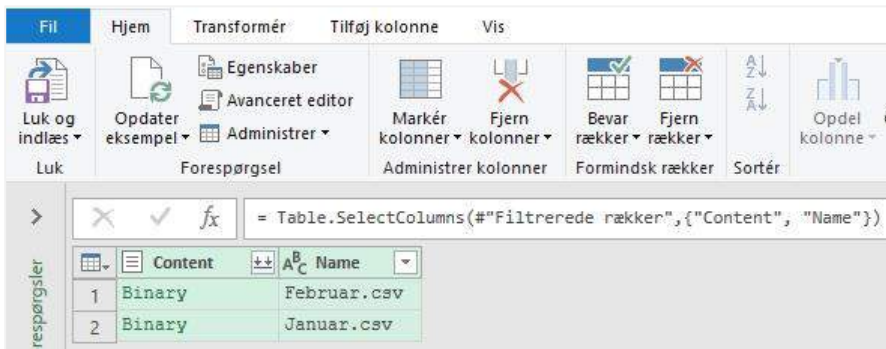
Hvilken metode du vælger, er op til dig, og du kan endda beslutte at bruge forskellige metoder under forskellige omstændigheder.

## Metode 1: Brug en brugerdefineret kolonne

Den første metode benytter en brugerdefineret kolonne, der indeholder en Power Query funktion.

Start med at vælge kolonnerne Content og Name, og klik derefter på **Hjem** → **Fjern kolonner** → **Fjern andre kolonner**. Dette fjerner filattribut kolonnerne, som vi ikke har brug for. Vi beholder kolonnen *Name* for at illustrere, at du ved hjælp af denne metode kan bevare filattributter i det endelige output.

Klik på **Tilføj kolonne** → **Brugerdefineret kolonne**.



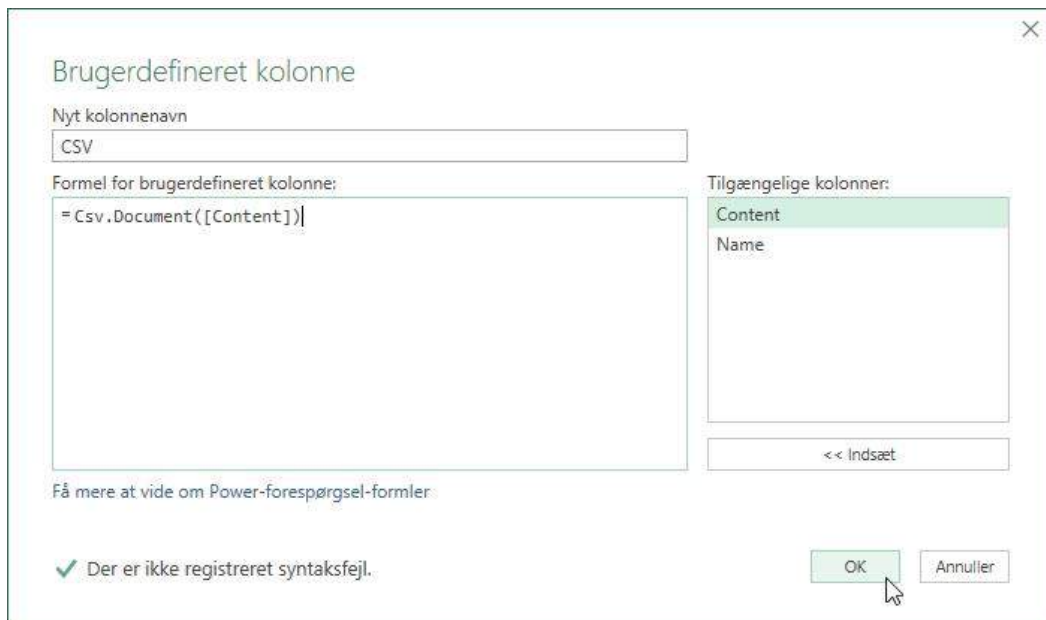
Indtast følgende oplysninger i vinduet **Brugerdefineret kolonne**:

Nyt kolonnenavn: CSV

Tilpasset kolonneformel:

```
= Csv.Document ([Content])
```

Klik på **OK**.



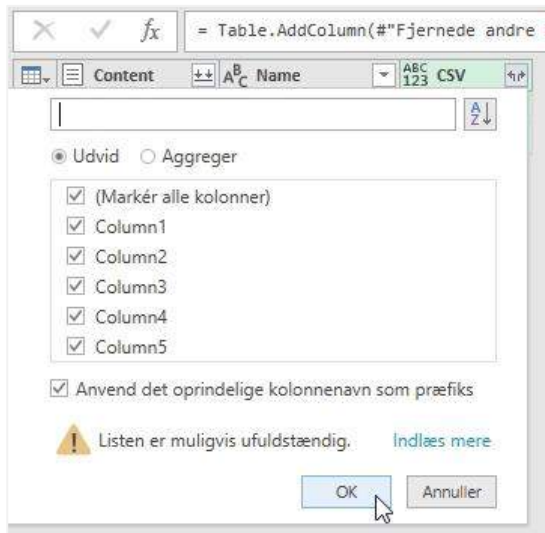
En ny kolonne kaldet CSV føjes til eksempelvinduet.

Du vil se et nyt ikon i CSV kolonneoverskriften. Dette er ikonet Udvid og angiver, at kolonnen indeholder flere detaljer, som kan udvides.

Klik på ikonet Udvid, og klik derefter på **OK**, da vi vil beholde alle kolonner med data.



## Kapitel 9 Power Query - Importer alle filer i en mappe



Vi har nu udvidet alle data fra begge filer, og januar og februar filerne er kombineret.

	Content	Name	CSV.Column1	CSV.Column2	CSV.Column3	CSV.Column4	CSV.Column5
1	Binary	Februar.csv	Dato	Kunder	Produkt	Solgt af	Værdi
2	Binary	Februar.csv	26-02-2020	Rema 1000	Frugt	Dorit	76
3	Binary	Februar.csv	18-02-2020	Spar	Frugt	Jens	74
4	Binary	Februar.csv	05-02-2020	ICA	Kød	Birthe	47
5	Binary	Februar.csv	11-02-2020	Netto	Blomster	Dorit	78
6	Binary	Februar.csv	23-02-2020	Føtex	Frugt	Dorit	38
7	Binary	Februar.csv	07-02-2020	Føtex	Grøntsager	Dorit	79
8	Binary	Februar.csv	16-02-2020	Netto	Blomster	Dorit	47
9	Binary	Februar.csv	02-02-2020	Bilka	Frugt	Jens	45
10	Binary	Februar.csv	08-02-2020	Meny	Kød	Dorit	52
11	Binary	Februar.csv	10-02-2020	Meny	Grøntsager	Birthe	83
12	Binary	Februar.csv	05-02-2020	Lidl	Frugt	Jens	39
13	Binary	Februar.csv	04-02-2020	ICA	Frugt	Birthe	37
14	Binary	Februar.csv	06-02-2020	Bilka	Blomster	Lene	72

Gør følgende for at rydde op i tabellen:

- Fjern kolonnen Content
- Tilføj kolonneoverskrifterne, det skal være Kildefil, Dato, Kunder, Produkt, Solgt af og Værdi.
- I filteret for kolonnen Værdi fjernes teksten "Værdi"
- Angiv datatyperne for hver kolonne

## Kapitel 9 Power Query - Importer alle filer i en mappe

Vores eksempelvindue skal nu se sådan ud:

	A <sup>B</sup> <sub>C</sub> Kildefil	Dato	A <sup>B</sup> <sub>C</sub> Kunder	A <sup>B</sup> <sub>C</sub> Produkt	A <sup>B</sup> <sub>C</sub> Solgt af	1 <sup>2</sup> <sub>3</sub> Værdi
1	Februar.csv	26-02-2020	Rema 1000	Frugt	Dorit	76
2	Februar.csv	18-02-2020	Spar	Frugt	Jens	74
3	Februar.csv	05-02-2020	ICA	Kød	Birthe	47
4	Februar.csv	11-02-2020	Netto	Blomster	Dorit	78
5	Februar.csv	23-02-2020	Føtex	Frugt	Dorit	38
6	Februar.csv	07-02-2020	Føtex	Grøntsager	Dorit	79
7	Februar.csv	16-02-2020	Netto	Blomster	Dorit	47
8	Februar.csv	02-02-2020	Bilka	Frugt	Jens	45
9	Februar.csv	08-02-2020	Meny	Kød	Dorit	52

Det er det hele, klik på **Luk og indlæs** for at indlæse tabellen i Excel.

### Metode 2: Kombiner filerne

Nu er det tid til metode 2. Denne metode bruger Power Query's smarte indbyggede kombineringsproces.

Hvis vi antager, at vi ikke benyttede metode 1, skal forhåndsvisningsvinduet se sådan ud:

	Content	A <sup>B</sup> <sub>C</sub> Name	A <sup>B</sup> <sub>C</sub> Extension	Date accessed	Date modified
1	Binary	Februar.csv	.csv	07-05-2020 19:59:...	07-05-2020 19:58:...
2	Binary	Januar.csv	.csv	07-05-2020 19:59:...	07-05-2020 19:57:...

Har du bemærket ikonet med de to nedadventede pile øverst i kolonnen Content? Det er ikonet **Kombiner filer**, det analyser fra filniveau til indholds niveau og kombinerer hver fil. Fortsæt og klik på **Kombiner filer**.

Dialogboksen **Kombiner filer** åbnes og viser en prøve af dataene ved hjælp af den første fil i forespørgslen som et eksempel. Alt ser godt ud, så klik på **OK**.

### Kombiner filer

Angiv indstillingerne for hver fil. Få mere at vide

Eksempelfil:  
Første fil

Filoprindelse: 1252: Vesteuropæisk (Windows) | Afgrænsning: Komma | Registrering af datatype: Baseret på de første 200 rækker

Dato	Kunder	Produkt	Solgt af	Værdi
26-02-2020	Rema 1000	Frugt	Dorit	76
18-02-2020	Spar	Frugt	Jens	74
05-02-2020	ICA	Kød	Birthe	47
11-02-2020	Netto	Blomster	Dorit	78
23-02-2020	Føtex	Frugt	Dorit	38
07-02-2020	Føtex	Grøntsager	Dorit	79
16-02-2020	Netto	Blomster	Dorit	47
02-02-2020	Bilka	Frugt	Jens	45
08-02-2020	Meny	Kød	Dorit	52
10-02-2020	Meny	Grøntsager	Birthe	83
05-02-2020	Lidl	Frugt	Jens	39
04-02-2020	ICA	Frugt	Birthe	37
06-02-2020	Bilka	Blomster	Lene	72

Spring filer med fejl over

OK Annuller

## Kapitel 9 Power Query - Importer alle filer i en mappe

Power Query vil nu udføre sin magi. Når den er færdig, kan vi se filerne er blevet kombineret, det var alt, hvad det krævede, en knap! Som vist på skærbilledet nedenfor er der foretaget en masse transformationer og trin for at opnå dette.

	Source.Name	Dato	Kunder	Produkt	Solgt af	Værdi
1	Februar.csv	26-02-2020	Rema 1000	Frugt	Dorit	
2	Februar.csv	18-02-2020	Spar	Frugt	Jens	
3	Februar.csv	05-02-2020	ICA	Kød	Birthe	
4	Februar.csv	11-02-2020	Netto	Blomster	Dorit	
5	Februar.csv	23-02-2020	Føtex	Frugt	Dorit	
6	Februar.csv	07-02-2020	Føtex	Grøntsager	Dorit	
7	Februar.csv	16-02-2020	Netto	Blomster	Dorit	
8	Februar.csv	02-02-2020	Bilka	Frugt	Jens	
9	Februar.csv	08-02-2020	Meny	Kød	Dorit	
10	Februar.csv	10-02-2020	Meny	Grøntsager	Birthe	
11	Februar.csv	05-02-2020	Lidl	Frugt	Jens	
12	Februar.csv	04-02-2020	ICA	Frugt	Birthe	
13	Februar.csv	06-02-2020	Bilka	Blomster	Lene	
14	Februar.csv	05-02-2020	Meny	Grøntsager	Dorit	

Du kan se, at dataene for både januar og februar er blevet kombineret i denne tabel.

Klik på **Hjem** → **Luk og indlæs** for at indlæse dataene i Excel.

### Tilføjelse af flere filer til mappen

Jeg har oprettet en Pivottabel for at vise, at filerne for januar og februar er blevet kombineret.

	A	B	C	D	E	F	G	H	I
2	Februar.csv	26-02-2020	Rema 1000	Frugt	Dorit	76		Rækkenavne	Sum af Værdi
3	Februar.csv	18-02-2020	Spar	Frugt	Jens	74		jan	2928
4	Februar.csv	05-02-2020	ICA	Kød	Birthe	47		feb	2935
5	Februar.csv	11-02-2020	Netto	Blomster	Dorit	78		Hovedtotal	5863
6	Februar.csv	23-02-2020	Føtex	Frugt	Dorit	38			

Kopier filen Marts.csv til mappen.



Og nu til det spændende, klik på **Data** → **Opdater alle**, dataene bliver opdateret.

"Hvis den er opdateret, hvorfor blev pivottabellen så ikke opdateret?" - Godt spørgsmål. Pivottabellen opdateres, inden dataene blev indlæst fra forespørgslen, så vi er nødt til at klikke på **Data** → **Opdater alle** igen for at opdatere pivottabellen.

	A	B	C	D	E	F	G	H	I
1	Source.Name	Dato	Kunder	Produkt	Solgt af	Værdi			
2	Februar.csv	26-02-2020	Rema 1000	Frugt	Dorit	76		Rækkenavne	Sum af Værdi
3	Februar.csv	18-02-2020	Spar	Frugt	Jens	74		jan	2928
4	Februar.csv	05-02-2020	ICA	Kød	Birthe	47		feb	2935
5	Februar.csv	11-02-2020	Netto	Blomster	Dorit	78		mar	2935
6	Februar.csv	23-02-2020	Føtex	Frugt	Dorit	38		Hovedtotal	8798
7	Februar.csv	07-02-2020	Føtex	Grøntsager	Dorit	79			

Læs eventuelt kapitlet Opdater data for at deaktivere behovet for dobbelt opdatering.

### Filens ensartethed er nøglen til succes

Tilføj nu filen April.csv til mappen. Klik på **Data** → **Opdater alle**.

Metode 1 opdaterer uden problemer. Metode 2 vil dog vise en fejl.



Hvorfor skete dette? Hvad gik galt? Åbn April.csv og kig nærmere på den første række. I denne fil er kolonnen der kaldes Kunder, der findes i de andre filer, ændret til Solgt til.

	A	B	C	D	E
1	Dato	Solgt til	Produkt	Solgt af	Værdi
2	26-04-2020	Rema 1000	Frugt	Dorit	76
3	18-04-2020	Spar	Frugt	Jens	74
4	05-04-2020	ICA	Kød	Birthe	47
5	11-04-2020	Netto	Blomster	Dorit	78
6	23-04-2020	Føtex	Frugt	Dorit	38

Som en del af de automatiske trin promoverer Power Query den første række i en CSV-fil som kolonneoverskrifter. Da vi oprindeligt kombinerede filerne ved hjælp af metode 2, blev Februar brugt som eksempelfil, derfor blev kolonneoverskrifterne fra Februar benyttet som kolonneoverskrifter for hele forespørgslen. Power Query kan på ingen måde vide, at April er forskellig fra de andre filer.

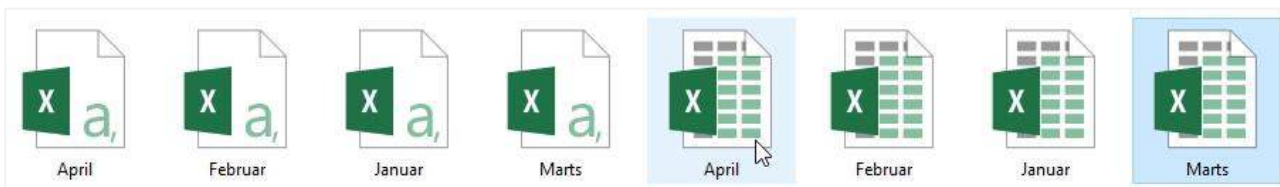
Metode 1 bruger ikke kolonneoverskrifterne fra CSV-filerne, men bruger de relative kolonnepositioner. Hvilket er fint, medmindre kolonnens rækkefølge ændres i filer, i hvilket tilfælde du kunne kombinere forskellige datatyper uden at vide det.

Dette er grunden til, at filens ensartethed er nøglen til succes med denne teknik.

### Importer Excel projektmappe fra en mappe

Metoden til at importere Excel projektmapper er næsten identisk med CSV-filer. Men det indebærer forskellige udfordringer. Vi har allerede set i tidligere kapitler, at Power Query godt kan lide at hardkode værdier i forespørgslen. Dette bliver et problem, når alle vores Excel projektmapper har regneark med forskellige navne. For eksempel har alle vores eksempelfiler forskellige regnearksnavne.

Kopier *Marts.xlsx* og *April.xlsx* ind i mappen.



## Kapitel 9 Power Query - Importer alle filer i en mappe

Følg disse trin i Excel:

- Vælg **Data** → **Hent data** → **Fra fil** → **Fra mappe**.
- Klik på **Gennemse** i dialogboksen Mappe. Find din mappe, og klik derefter på **OK**.
- Når listen over filer vises, skal du klikke på knappen **Rediger**.
- Forespørgsel editor vinduet åbnes.
- I kolonnen Extension filtreres så den kun omfatte .xlsx filer.

Lad os se på de to metoder som før.

### Metode 1: Brug en brugerdefineret kolonne

Metode 1 benytter en brugerdefineret kolonneformel som før.

Vælg kolonnerne Content og Name, og **fjern derefter andre kolonner**.

Lad os nu tilføje den nye kolonne, klik på **Tilføj kolonne** → **Brugerdefineret kolonne**.

Indtast følgende oplysninger i dialogboksen Brugerdefineret kolonne.

Nyt kolonnenavn: ExcelFiler

Tilpasset kolonneformel:

```
=Excel.Workbook([Content])
```

Klik på OK.

Brugerdefineret kolonne

Nyt kolonnenavn  
ExcelFiler

Formel for brugerdefineret kolonne:  
=Excel.Workbook([Content])

Tilgængelige kolonner:  
Content  
Name

<< Indsæt

Få mere at vide om Power-forespørgsel-formler

✓ Der er ikke registreret syntaksfejl.

OK Annuller

Den nye kolonne ExcelFiler tilføjes. Klik på ikonet Udvid, og klik derefter på **OK** for at inkludere alle kolonner.

	Content	Name	ExcelFiler
1	Binary	April.xlsx	Table
2	Binary	Februar.xlsx	Table
3	Binary	Januar.xlsx	Table
4	Binary	Marts.xlsx	Table

## Kapitel 9 Power Query - Importer alle filer i en mappe

Hvert regneark og tabel fra projektmappen er vist. Filen April indeholder et regneark og en tabel, filtrer kolonnen ExcelFiler.Kind for at fjerne tabellerne og kun beholde regneark (Sheet).

A <sup>B</sup> C	Name	ABC 123	ExcelFiler.Name	ABC 123	ExcelFiler.Data	ABC 123	ExcelFiler.Item	ABC 123	ExcelFiler.Kind
	April.xlsx		April		Table		April		Sheet
	April.xlsx		tblApril		Table		tblApril		Table
	Februar.xlsx		Februar		Table		Februar		Sheet
	Januar.xlsx		Januar		Table		Januar		Sheet
	Marts.xlsx		Marts		Table		Marts		Sheet

Vælg kolonnen ExcelFiler.Data, og fjern derefter de andre kolonner. Klik på ikonet Udvid i kolonnen ExcelFiler.Data.

Se på forhåndsvisningsvinduet alle regneark er nu kombineret. Vælg **Luk og indlæs** for at indlæse dataene i Excel.

### Metode 2: Kombiner filerne

Nu er det tid til metode 2. Dette bruger Power Querys indbyggede kombineringsproces igen. Hvis vi antager, at vi ikke benyttede metode 1, skal forhåndsvisningsvinduet se sådan ud:

	Content	A <sup>B</sup> C	Name	A <sup>B</sup> C	Extension	Date accessed	Date modified
1	Binary		April.xlsx		.xlsx	08-05-2020 09:45:...	07-05-2020 22:44:...
2	Binary		Februar.xlsx		.xlsx	07-05-2020 20:35:...	07-05-2020 19:38:...
3	Binary		Januar.xlsx		.xlsx	07-05-2020 20:35:...	07-05-2020 19:37:...
4	Binary		Marts.xlsx		.xlsx	08-05-2020 09:45:...	07-05-2020 19:39:...

Klik på kombinationsikonet i kolonnen Content, som vi gjorde, da vi brugte metode 2 til CSV-filer.

Da April.xlsx er først på listen over filer, vil Power Query bruge denne, som en prøve fil. I dialogboksen Kombiner filer skal du vælge regnearket April og klikke på **OK**.

Kombiner filer

Vælg det objekt, der skal pakkes ud fra hver fil. Få mere at vide

Eksempelfil: Første fil

Indstillinger for visning

- Parameter for eksempelfil [2]
  - tblApril
  - April

April

Column1	Column2	Column3	Column4	Column5
Salgsdata		null	null	null
April 2020		null	null	null
	null	null	null	null
Dato	Solgt til	Produkt	Solgt af	Værdi
26-04-2020	Rema 1000	Frugt	Dorit	76
18-04-2020	Spar	Frugt	Jens	74
05-04-2020	ICA	Kød	Birthe	47
11-04-2020	Netto	Blomster	Dorit	78
23-04-2020	Føtex	Frugt	Dorit	38
07-04-2020	Føtex	Grøntsager	Dorit	79
16-04-2020	Netto	Blomster	Dorit	47
02-04-2020	Bilka	Frugt	Jens	45

Spring filer med fejl over

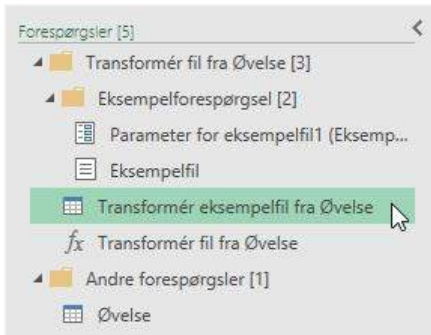
OK Annuller

Hvis alle dine regneark har det samme navn, er du heldig. Alle data kombineres da til en enkelt tabel.

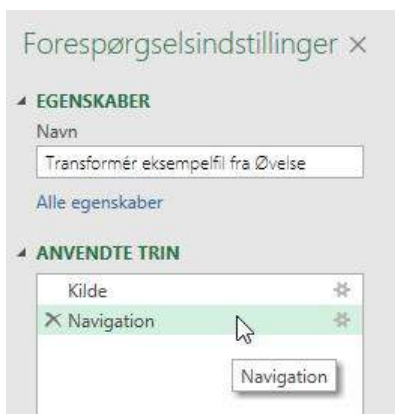
## Kapitel 9 Power Query - Importer alle filer i en mappe

Men i vores scenarie, som du måske har regnet med, indlæses kun dataene April i forhåndsvisningsvinduet. Der er kun en projektmappe med et regneark kaldet April, så kun et regneark indlæses. Vi er nødt til at finde placeringen i M koden, hvor regnearks navnet er blevet kodet og justere det.

Åbn forespørgslen Transformer eksempel fil fra Øvelse.



Klik på Navigations trinnet i panelet Anvendte trin.



M koden til dette trin ser sådan ud:

```
= Kilde{ [Item="April", Kind="Sheet"] } [Data]
```

Slet den fremhævede del. M koden skal nu se sådan ud:

```
= Source{ [Kind="Sheet"] } [Data]
```

Vend tilbage til hovedforespørgslen og se på forhåndsvisningsvinduet. Alle regneark er nu kombineret.

### Afslut forespørgslen

Vi bliver nødt til at foretage nogle ændringer for at rydde op i tingene.

- Fjern de 3 øverste rækker.
- Fremhæv den første række som overskrift ved at klikke på **Transformerer** → **Brug første række som overskrifter**.
- Vælg den sidste kolonne og filter for at fjerne "null" og "Værdi".
- Anvend de korrekte datatyper i hver kolonne.

Til sidst skal du **lukke og indlæse** forespørgslen i Excel.

## Bemærkninger og tips

---

Her er et par bonustips:

- Når du leder efter filer, ser Power Query også i undermapper. Dette er smart, når filer er organiseret i en defineret struktur. Men hvis mapperne er rodede, kan det være nødvendigt med en smule oprydning for at få denne metode til at fungere.
- Planlæg, hvad der kan gå galt, tænk over, hvad der kan ske, hvis en anden skulle gemme en anden filtype i mappen. Når du opretter forespørgslen, skal du medtage ekstra filtertrin for at sikre kun, at de ønskede filer importeres.
- Data, der kommer fra et it-system, fungerer normalt godt sammen med denne teknik. Jeg har opdaget, at al data, der kræver brugerinput, har tendens til at skabe problemer ... hvorfor? Fordi folk ændrer ting, tilføjer kolonner, omdøber ark, overskriver statiske værdier. Hvis dine filer kommer fra brugerinput, skal du overveje at bruge kombinerings- og tilføjelsesmetoden.



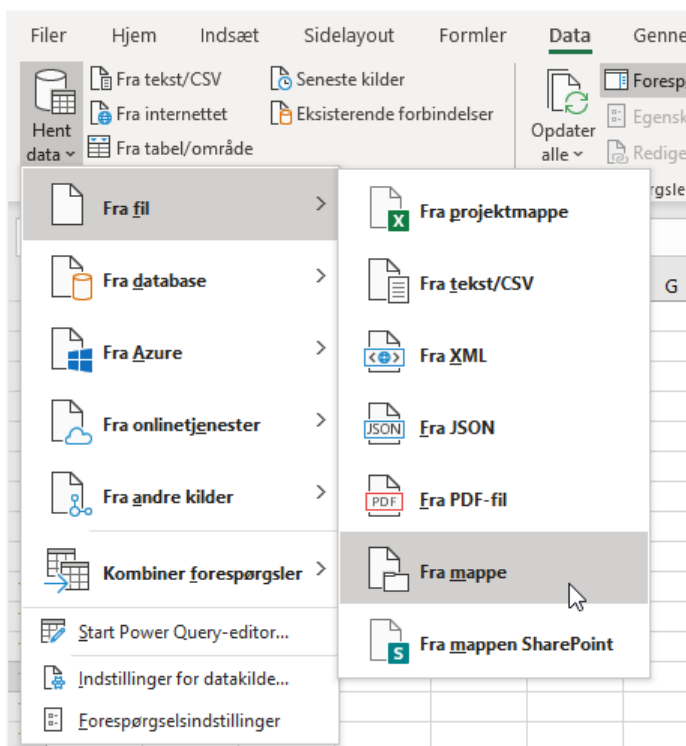
## Kapitel 10 Power Query – Vis alle filer i en mappe og filattributter

I **sidste kapitel** så vi, hvordan du **importerer alle filerne fra en mappe**, hvilket er en smart tidsbesparende teknik. Mens vi fokuserede på dette aspekt, sprang vi over en anden nyttig del af Power Query; som er muligheden for at hente en liste over alle filerne i en mappe og undermapper sammen med filattributterne.

Jeg har tidligere skrevet artikler om, hvordan man viser alle filerne i en mappe ved hjælp af VBA og også ved hjælp af en ældre Excel funktioner. Disse artikler er der stadig mange der læser, så jeg ved, at dette er et område, som brugerne er interesseret i. Men i Power Query findes nu den nemmeste metode af alle.

### Vis en liste over alle filerne i en mappe (og undermapper)

I Excel skal du klikke på **Data** → **Hent data** → **Fra fil** → **Fra mappe**.



Klik på **Gennemse** i dialogboksen Mappe, og naviger frem til mappen, og klik derefter på **OK**.



Når listen over filer åbnes, skal du klikke på **Rediger**.

## Kapitel 10 – Vis alle filer i en mappe og filattributter

C:\Power Query

Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
Binary	CSV.csv	.csv	24-04-2020 12:41:36	18-04-2020 17:36:31	24-04-2020 12:41:36	Record	C:\Power Q
Binary	Navngivnområde.xlsx	.xlsx	24-04-2020 12:41:36	18-04-2020 17:28:29	24-04-2020 12:41:36	Record	C:\Power Q
Binary	Projektmappe.xlsx	.xlsx	24-04-2020 12:41:36	19-04-2020 10:36:01	24-04-2020 12:41:36	Record	C:\Power Q
Binary	Tablel.xlsx	.xlsx	24-04-2020 12:41:36	17-04-2020 22:06:21	24-04-2020 12:41:36	Record	C:\Power Q
Binary	Tekst fil.txt	.txt	24-04-2020 12:41:36	19-04-2020 10:04:33	24-04-2020 12:41:36	Record	C:\Power Q
Binary	Opdater data 1.csv	.csv	24-04-2020 12:41:08	20-04-2020 07:48:11	24-04-2020 12:41:08	Record	C:\Power Q
Binary	Opdater data 1x.csv	.csv	24-04-2020 12:41:08	20-04-2020 07:47:56	24-04-2020 12:41:08	Record	C:\Power Q
Binary	Power Query - Rediger forespørgsler.xlsx	.xlsx	24-04-2020 12:40:34	14-04-2020 22:30:14	24-04-2020 12:40:34	Record	C:\Power Q
Binary	Power Query - Brug af parametre.xlsx	.xlsx	24-04-2020 12:39:58	13-04-2020 09:01:04	24-04-2020 12:39:58	Record	C:\Power Q
Binary	April 2020.xlsx	.xlsx	24-04-2020 12:39:29	23-04-2020 21:27:13	24-04-2020 12:39:29	Record	C:\Power Q
Binary	Februar 2020.xlsx	.xlsx	24-04-2020 12:39:29	21-04-2020 22:26:08	24-04-2020 12:39:29	Record	C:\Power Q
Binary	Januar 2020.xlsx	.xlsx	24-04-2020 12:39:29	22-04-2020 09:28:26	24-04-2020 12:39:29	Record	C:\Power Q
Binary	Marts 2020.xlsx	.xlsx	24-04-2020 12:39:29	21-04-2020 23:04:53	24-04-2020 12:39:29	Record	C:\Power Q

Kombiner ▾ Indlæs ▾ Rediger Annuller

Power Query editoren åbnes og viser nogle nyttige attributter:

- Name
- Extension
- Data accessed
- Date modified
- Date created
- Folder Path

Flyt kolonnen Name til slutningen, marker derefter kolonnerne Folder Path og Name, klik på **Tilføj kolonne** → **Flet kolonner**.

Table.ReorderColumns(Kilde, {"Content", "Extension", "Date accessed", "Date modified", "Date created", "Attributes", "Folder Path", "Name"})

Content	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path	Name
1	Binary	.csv	24-04-2020 12:41:36	18-04-2020 17:36:31	24-04-2020 12:41:36	Record	C:\Power Query\Kapitel 2\ CSV.csv
2	Binary	.xlsx	24-04-2020 12:41:36	18-04-2020 17:28:29	24-04-2020 12:41:36	Record	C:\Power Query\Kapitel 2\ Navngivnområde.xlsx
3	Binary	.xlsx	24-04-2020 12:41:36	19-04-2020 10:36:01	24-04-2020 12:41:36	Record	C:\Power Query\Kapitel 2\ Projektmappe.xlsx
4	Binary	.xlsx	24-04-2020 12:41:36	17-04-2020 22:06:21	24-04-2020 12:41:36	Record	C:\Power Query\Kapitel 2\ Tablel.xlsx
5	Binary	.txt	24-04-2020 12:41:36	19-04-2020 10:04:33	24-04-2020 12:41:36	Record	C:\Power Query\Kapitel 2\ Tekst fil.txt
6	Binary	.csv	24-04-2020 12:41:08	20-04-2020 07:48:11	24-04-2020 12:41:08	Record	C:\Power Query\Kapitel 3\ Opdater data 1.csv
7	Binary	.csv	24-04-2020 12:41:08	20-04-2020 07:47:56	24-04-2020 12:41:08	Record	C:\Power Query\Kapitel 3\ Opdater data 1x.csv
8	Binary	.xlsx	24-04-2020 12:40:34	14-04-2020 22:30:14	24-04-2020 12:40:34	Record	C:\Power Query\Kapitel 4\ Power Query - Rediger forespørgsler.xlsx
9	Binary	.xlsx	24-04-2020 12:39:58	13-04-2020 09:01:04	24-04-2020 12:39:58	Record	C:\Power Query\Kapitel 7\ Power Query - Brug af parametre.xlsx
10	Binary	.xlsx	24-04-2020 12:39:29	23-04-2020 21:27:13	24-04-2020 12:39:29	Record	C:\Power Query\Kapitel 8\ April 2020.xlsx
11	Binary	.xlsx	24-04-2020 12:39:29	21-04-2020 22:26:08	24-04-2020 12:39:29	Record	C:\Power Query\Kapitel 8\ Februar 2020.xlsx
12	Binary	.xlsx	24-04-2020 12:39:29	22-04-2020 09:28:26	24-04-2020 12:39:29	Record	C:\Power Query\Kapitel 8\ Januar 2020.xlsx
13	Binary	.xlsx	24-04-2020 12:39:29	21-04-2020 23:04:53	24-04-2020 12:39:29	Record	C:\Power Query\Kapitel 8\ Marts 2020.xlsx

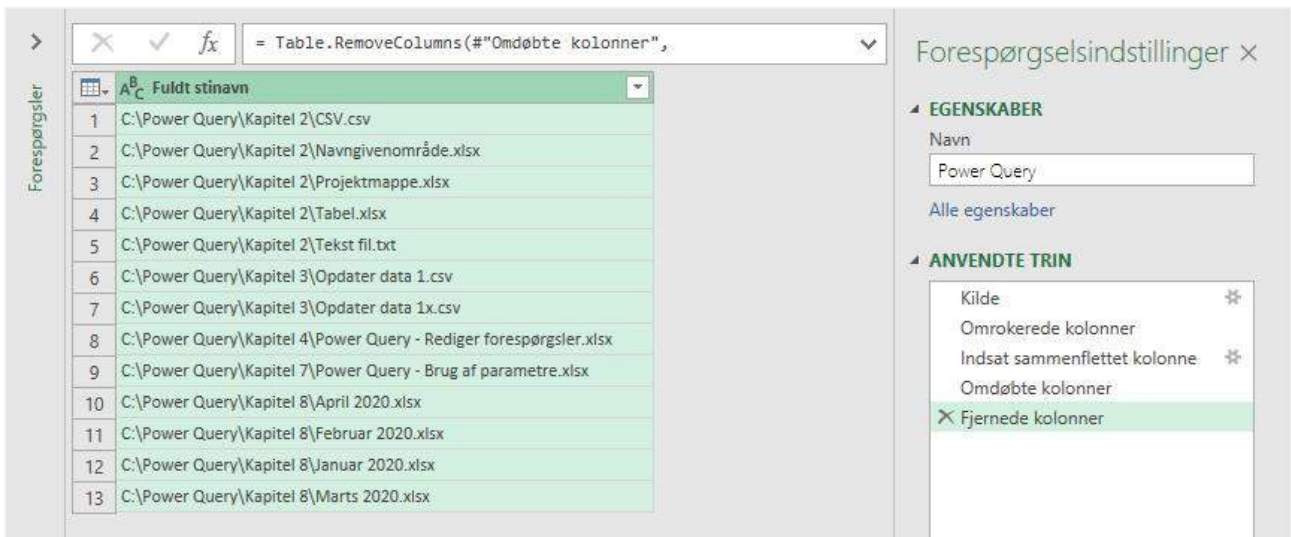
Vælg ingen separator i dialogboksen Flet kolonner, og klik på OK.



En ny kolonne oprettes med den komplette fil sti. Foretag følgende transformationer:

- Omdøb den nye kolonne.
- Fjern alt undtagen den nye kolonne.

Eksempelvinduet viser en liste over alle filer og mapper.



Klik på **Hjem** → **Luk og indlæs** for at indlæse dataene til Excel.

Det var nemt!

## Filattributter

I afsnittet ovenfor så vi nogle filattributter, for det meste baseret på datoer, men dette er bare starten. Kolonnen **Attributes** i eksempelvisningsvinduet giver adgang til masser af anden information.

## Kapitel 10 – Vis alle filer i en mappe og filattributter

	Date accessed	Date modified	Date created	Attributes	Folder
1	24-04-2020 12:41:36	18-04-2020 17:36:31	24-04-2020 12:41:36	Record	C:\Power Q
2	24-04-2020 12:41:36	18-04-2020 17:28:29	24-04-2020 12:41:36	Record	C:\Power Q
3	24-04-2020 12:41:36	19-04-2020 10:36:01	24-04-2020 12:41:36	Record	C:\Power Q
4	24-04-2020 12:41:36	17-04-2020 22:06:21	24-04-2020 12:41:36	Record	C:\Power Q
5	24-04-2020 12:41:36	19-04-2020 10:04:33	24-04-2020 12:41:36	Record	C:\Power Q
6	24-04-2020 12:41:08	20-04-2020 07:48:11	24-04-2020 12:41:08	Record	C:\Power Q
7	24-04-2020 12:41:08	20-04-2020 07:47:56	24-04-2020 12:41:08	Record	C:\Power Q
8	24-04-2020 12:40:34	14-04-2020 22:30:14	24-04-2020 12:40:34	Record	C:\Power Q

Klik på udvidelsesikonet i overskriften i kolonnen Attributes, og klik derefter på OK for at se filattributterne. Jeg vil ikke lyve, men Jeg har ingen idé om, hvad nogle af disse attributter er, men andre er meget nyttige, såsom:

- Kind - filtypen
- Size - filstørrelsen
- Read Only - om filen i øjeblikket er indstillet til Read Only
- Hidden - om filen er synlig

Forestil dig, at lagerpladser på din harddisk er lille. Ved hjælp af denne metode kan du hurtigt identificere de største filstørrelser eller de filer, der ikke er åbnet for nylig, eller dupliserede filnavne. Når du har denne liste, kan du beslutte, hvad du skal beholde, og hvad du skal slette.

## Projektmappe attributter

Attributter ender ikke på filniveau. Følg trinnene ovenfor for at navigere frem til en mappe, der indeholder Excel filer. Filtrer for kun at inkludere de Excel filer, du er interesseret i. Dit eksempelvindue kan se sådan ud.

	Content	Name	Extension	Date accessed	Date modified
1	Binary	Navngivenområde.xlsx	.xlsx	04-05-2020 20:43:...	04-05-2020 20:
2	Binary	Projektmappe.xlsx	.xlsx	24-04-2020 12:41:...	19-04-2020 10:
3	Binary	Tabel.xlsx	.xlsx	24-04-2020 12:41:...	17-04-2020 22:
4	Binary	Example 7 - Edit Queries.xlsx	.xlsx	01-05-2020 20:16:...	01-05-2020 20:
5	Binary	Power Query - Rediger forespørgsle...	.xlsx	03-05-2020 10:09:...	01-05-2020 20:

Klik på **Tilføj kolonne** → **Brugerdefineret kolonne**.

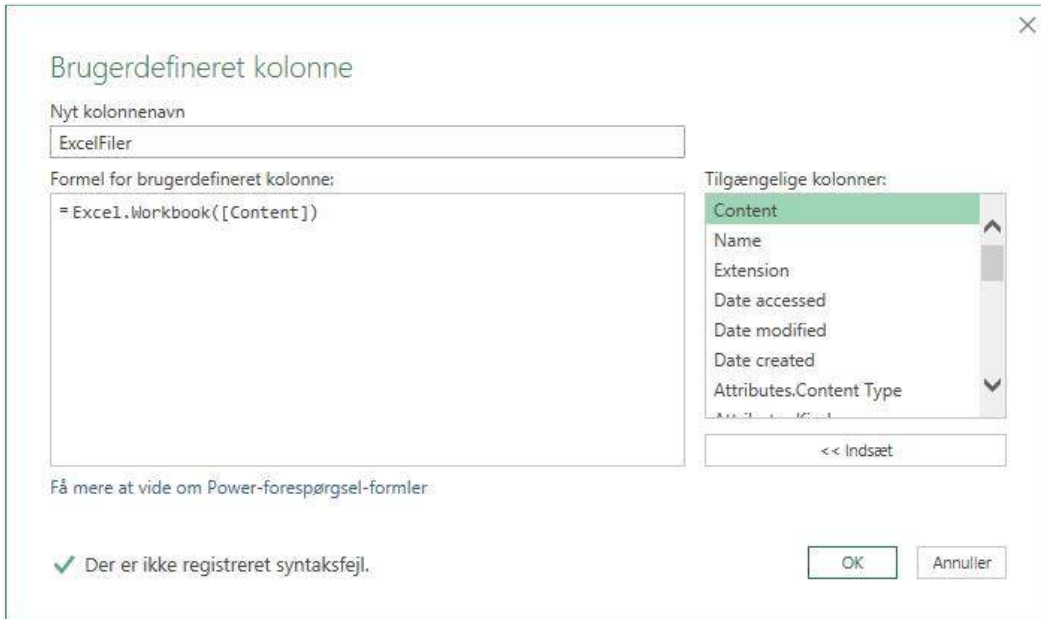
I dialogboksen Brugerdefineret kolonne indtastes følgende indstillinger:

Kolonnenavn: ExcelFil

Tilpasset kolonneformel:

```
=Excel.Workbook([Content])
```

Klik på **OK**.



Klik på Udvid ikonet i kolonnen ExcelFiler for at se nærmere på hver fil.

	Content	Name	ExcelFiler	Extension
1	Binary	Navngivenområde.xlsx	Table	.xlsx
2	Binary	Projektmappe.xlsx	Table	.xlsx
3	Binary	Tabel.xlsx	Table	.xlsx
4	Binary	Example 7 - Edit Queries.xlsx	Table	.xlsx

Attributterne her indeholder:

- Navn på regneark
- Kind - regneark eller tabel
- Hidden - om regnearket er skjult

## Afslutningsvis

Det var så det. Et smart trick til at få oplysninger om dine filer. Jeg bruger selv denne teknik til at kontrollere, hvilke filer jeg har i mine mapper eller drev. Nu kan jeg altså klikke på Opdater, og så er denne kontrol afsluttet på få sekunder.

## Kapitel 11 Power Query – Importer data fra den aktuelle projektmappe

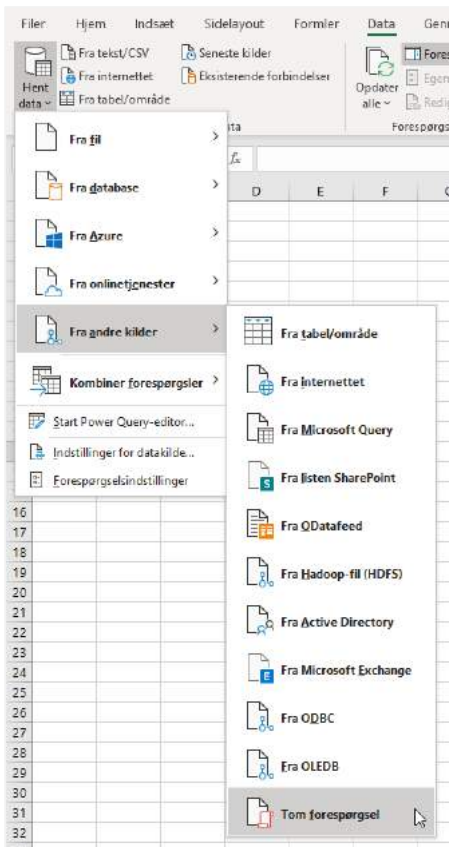
Vi har set, hvordan man importerer eksterne data fra en enkelt fil, hvordan man importerer alle filerne i en mappe, og hvordan man importerer data fra en tabel eller navneområde inden for den samme projektmappe. Men hvad nu hvis vi vil importere ALLE data inden for den samme projektmappe? Det er det vi skal se på i dette kapitel.

Hvis der er en fornuftig Excel vane, som brugerne ignorerer hele tiden, er det at opbevare data af samme type på en fane. Som du ser i eksempelfiler, indeholder januar, februar og marts den samme type data i den samme struktur, derfor bør være i en tabel på den samme fane. De fleste nye brugere af Excel tænker ikke sådan, så der er en god chance for at du støder på disse typer projektmapper og bliver nødt til at bruge denne teknik på et tidspunkt.

Eksemplerne i dette kapitel bruger filen **Importer data fra den aktuelle projektmappe.xlsx**.

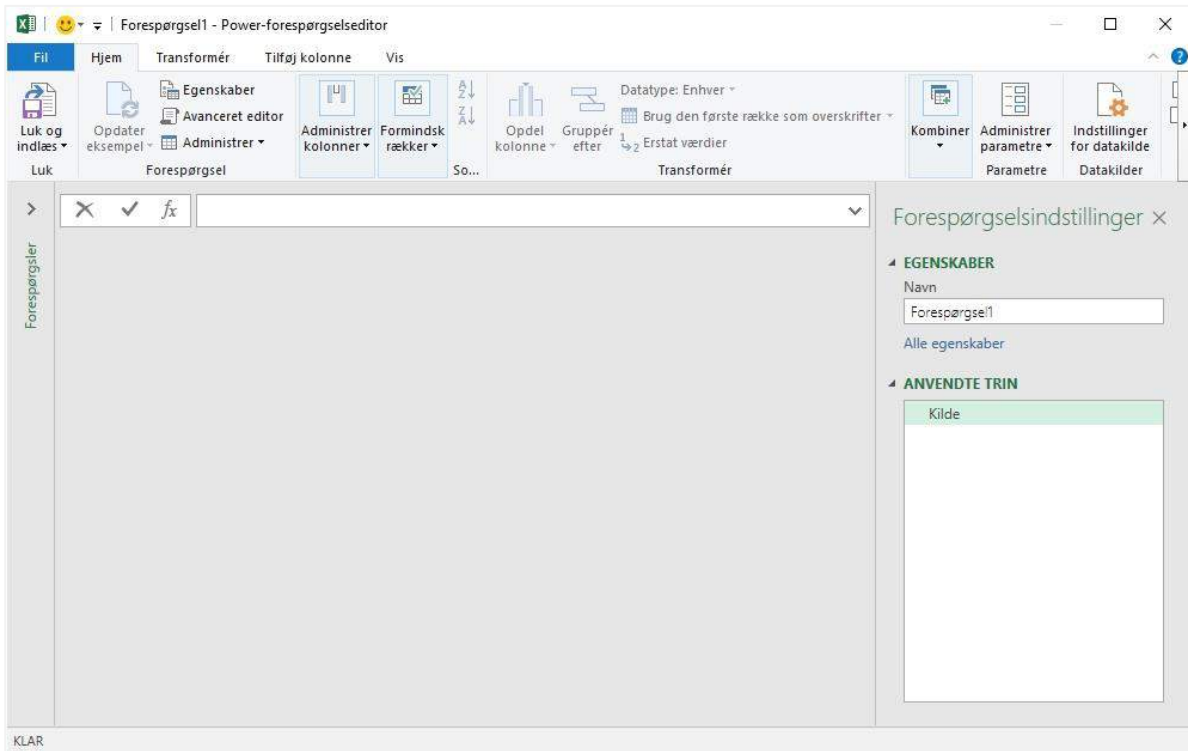
### Importer tabeller fra den aktuelle projektmappe

Åbn filen Importer data fra den aktuelle projektmappe.xlsx. Dernæst opretter vi med en tom forespørgsel ved at klikke på **Hent data** → **Fra andre kilder** → **Tom forespørgsel**.



Power Query editoren åbnes. Der er et trin i panelet Anvendte trin, intet i vinduet Eksempel og de fleste af transformationerne er gråtonede. Selvom panelet Anvendte trin viser trinnet Kilde, er der faktisk intet i dette trin i øjeblikket. Dette er i sandhed en tom forespørgsel.

## Kapitel 11 – Importer data fra den aktuelle projektmappe



Vi skal skrive noget M kode for at informere Power Query om kilden.

Hvis du bruger formellinjen, kunne vi skrive følgende (Klik på **Vis** → **Formellinje**, hvis formellinjen ikke er synlig).

```
= Excel.CurrentWorkbook()
```

Eller hvis vi bruger den avancerede editor (**Hjem** → **Avanceret editor**), kunne vi have følgende:

```
let
kilde = Excel.CurrentWorkbook()
in
kilde
```

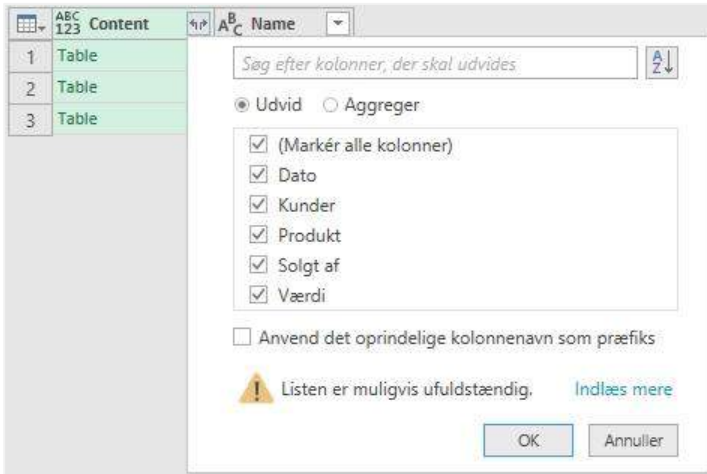
Husk at M kode er følsomt overfor store og små bogstaver, så du bliver nødt til at indtaste teksten nøjagtigt, som den er vist ovenfor.

Eksempelvinduet viser tabellerne i regnearket.

	ABC 123	Content		ABC Name
1		Table		tblJanuar
2		Table		tblFebruar
3		Table		tblMarts

Klik på ikonet **Udvid** for at åbne strukturen i projektmappen. Fjern markeringen i **Anvend det oprindelige kolonnenavn som præfiks**, og klik derefter på **OK**.

## Kapitel 11 – Importer data fra den aktuelle projektmappe



Eksempelvinduet viser nu de kombinerede data.

	ABC 123 Dato	ABC 123 Kunder	ABC 123 Produkt	ABC 123 Solgt af	ABC 123 Værdi	ABC Name
1	09-01-2020 00:00:00	Brugsen	Blomster	Birthe	76	tblJanuar
2	02-01-2020 00:00:00	Netto	Kød	Birthe	40	tblJanuar
3	19-01-2020 00:00:00	Føtex	Brød	Birthe	66	tblJanuar
4	19-01-2020 00:00:00	Netto	Kød	Dorit	51	tblJanuar
5	12-01-2020 00:00:00	Netto	Blomster	Dorit	75	tblJanuar
6	22-01-2020 00:00:00	Brugsen	Blomster	Mogens	46	tblJanuar
7	23-01-2020 00:00:00	Bilka	Kød	Lene	40	tblJanuar
8	25-01-2020 00:00:00	Bilka	Frugt	Dorit	57	tblJanuar
9	23-01-2020 00:00:00	Rema 1000	Frugt	Jørgen	50	tblJanuar
10	10-01-2020 00:00:00	Lidl	Frugt	Birthe	75	tblJanuar

færdiggør forespørgslen med følgende transformationer:

- Fjern kolonnen Name
- Skift datatype på hver kolonne
- Giv forespørgslen et passende navn (jeg har valgt KombineredeTabel).

Klik på **Luk og indlæs** for at indlæse dataene til et nyt regneark. Du ved det ikke endnu, men du har et problem, som jeg vil vise dig.

Panelet Projektmappe-forespørgsler viser at der er indlæst 251 rækker.



Foretag nogle ændringer i tabellerne januar, februar eller marts, og klik på **Data** → **Opdater alle**.

Hvad skete der lige her. Vi har nu fået 401 rækker, men vi har ikke tilføjet flere rækker.



## Kapitel 11 – Importer data fra den aktuelle projektmappe



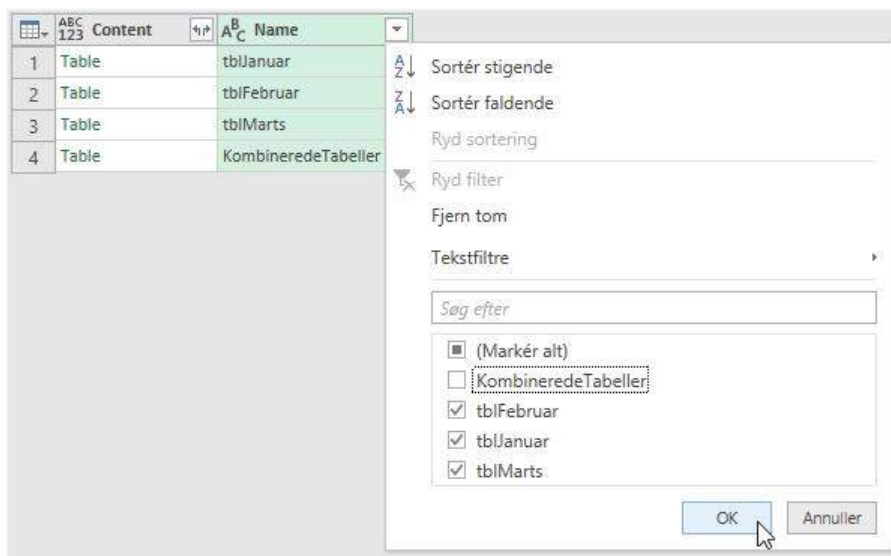
Lad os gå tilbage til Power Query og se, hvad der går galt. I panelet **Projektmappe-forespørgsler** dobbeltklikkes på forespørgslen for at åbne Power Query editoren.

Klik på trinnet kilde i vinduet Anvendte trin, og klik derefter på **Hjem** → **Opdater eksempel**. Eksempelvinduet viser nu dette:

	ABC 123 Content	A <sup>B</sup> C Name
1	Table	tblJanuar
2	Table	tblFebruar
3	Table	tblMarts
4	Table	KombineredeTabeller

Forhåbentlig kan du se problemet nu. Forespørgslen, vi oprettede, indlæser dataene i Excel som en tabel, derfor er de nu inkluderet som en kildetabel, hver gang forespørgslen opdateres og kombineres med de andre tabeller, inden den indlæses i Excel igen. Hver gang vi klikker på Opdater, bliver tabellen længere og længere og længere. Det ligner Excel versionen af filmen Inception.

Lad os løse dette problem med det samme. Tilføj et trin efter kildetrinnet for at filtrere den kombinerede forespørgsel fra.



Vil denne filtrering fjerne KombineredeTabeller, eller vil filtreringen omfatte tblJanuar, tblFebruar og tblMarts? Dette er vigtigt, da vi ønsker, at forespørgslen skal udvides til at omfatte nye tabeller, når de føjes til projektmappen.

Se på formellinjen; M koden ser sådan ud:

```
= Table.SelectRows(Kilde, each ([Name] <> "KombineredeTabeller"))
```

Den gode nyhed er, at denne kode fjerner tabellen KombineredeTabeller, men tillader, at alle andre tabeller, der er tilføjet til projektmappen, automatisk inkluderes i forespørgslen rækkevidde.

Det er smart at have en standard navnekonvention for tabeller og forespørgsler. For eksempel kan du beslutte, at alle kildetabeller skal have tbl i starten, dette tillader os at filtrere så vi kun inkluderer tabeller, der starter med tbl. Ved at gøre dette betyder det ikke længere noget, hvor mange tabeller eller forespørgsler vi har, løkkeeffekten kan kontrolleres.

### Tabeller eller områder

---

Power Query importerer med glæde tabeller og navngivne områder. Hvis de begge findes i regnearket, vises begge på listen.

Udskriftsområder er en speciel type navngivet område, så de er også med på listen. Korrekte navnekonventioner og filtreringer er påkrævet for at sikre, at forespørgslen bruger den kilde, vi ønsker.

## Kapitel 12 Power Query – data fra internettet

Internettet er et stort sted med utrolig meget information. Men bare fordi dataene er der, er det ikke nødvendigvis let at importere dem, vi skal se på, hvordan vi kan indlæse webdata i Power Query. Mens du måske bliver begejstret ved tanken om at absorbere data overalt med et klik på en knap, er der ikke uanede muligheder! Power Query kan gøre meget, men webstedet skal være i et rimeligt anvendeligt format til at begynde med.

Nok snak, det er tid til at importere nogle data fra nettet.

### Grundlæggende internet import

I denne øvelse vil vi bruge <https://www.xe.com> til at importere valutakurser til Power Query.

Webstedet indeholder informationer, der viser valutakurserne for enhver valuta på en hvilken som helst dato. Her er DKK kurserne for 19. marts 2020.

**Current and Historical Rate Tables**

Build current and historic rate tables with your chosen base currency with XE Currency Tables. For commercial purposes, get an automated currency feed through the XE Currency Data API.

DKK - Danish Krone    2020-03-19

**XE Currency Table: DKK - Danish Krone**

2020-03-19 14:29 UTC

All figures are live mid-market rates, which are not available to consumers and are for informational purposes only. To see the rates we quote for money transfer, please use our money transfer service.

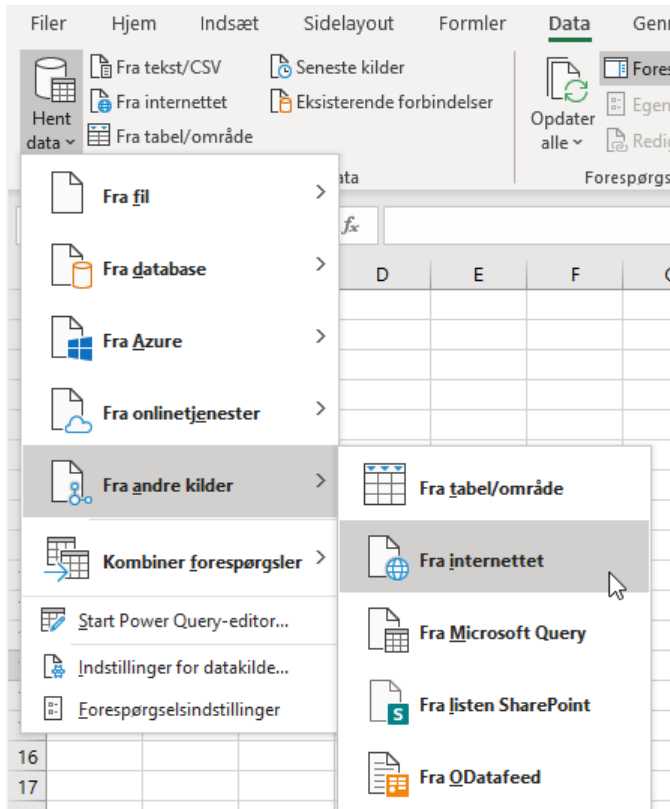
Currency code ▲▼	Currency name ▲▼	Units per DKK	DKK per Unit
USD	US Dollar	0.1446239781	6.9144827365
EUR	Euro	0.1338289556	7.4722244932
GBP	British Pound	0.1240259773	8.0628270134
INR	Indian Rupee	10.8098204998	0.0925084741
AUD	Australian Dollar	0.2465581991	4.0558375420
CAD	Canadian Dollar	0.2090524536	4.7834884640
SGD	Singapore Dollar	0.2091417855	4.7814452651
CHF	Swiss Franc	0.1409611444	7.0941535303
MYR	Malaysian Ringgit	0.6383800658	1.5664649534

Den mest gavnlige del af skærmbillede ovenfor er URL adressen. Hvis du ser nøje efter, indeholder den valutakoden og datoen i URL'en. Jeg spekulerer på, om vi kunne bruge det på en mere avanceret måde?

<https://www.xe.com/currencytables/?from=DKK&date=2020-03-19>

Lad os starte med det grundlæggende, inden vi går til det mere avancerede. Kopier URL'en, og vælg derefter i Excel på **Hent data → Fra andre kilder → Fra internettet**.

## Kapitel 12 Power Query - Webdata

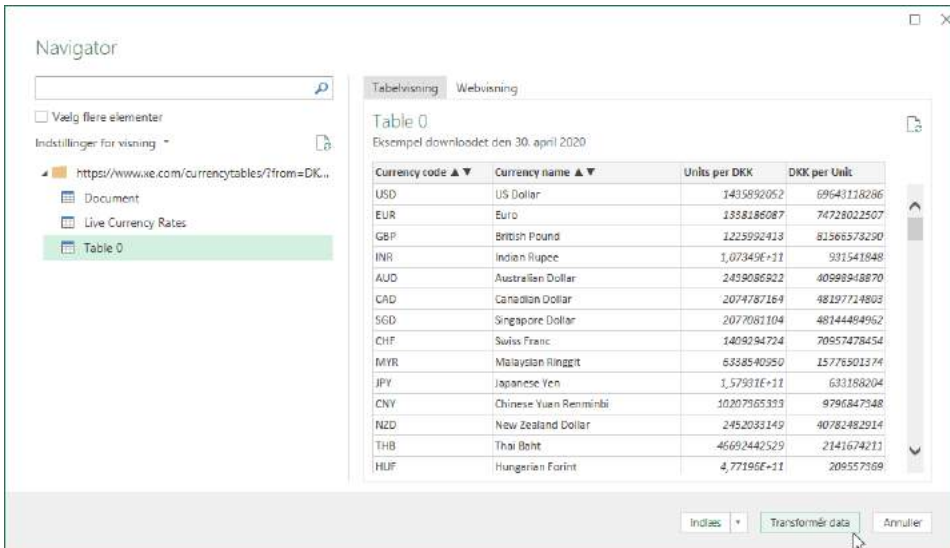


Dialogboksen Fra web åbnes. **Indsæt web-adressen**, vi kopierede tidligere, i URL-adressen, og klik på **OK**.

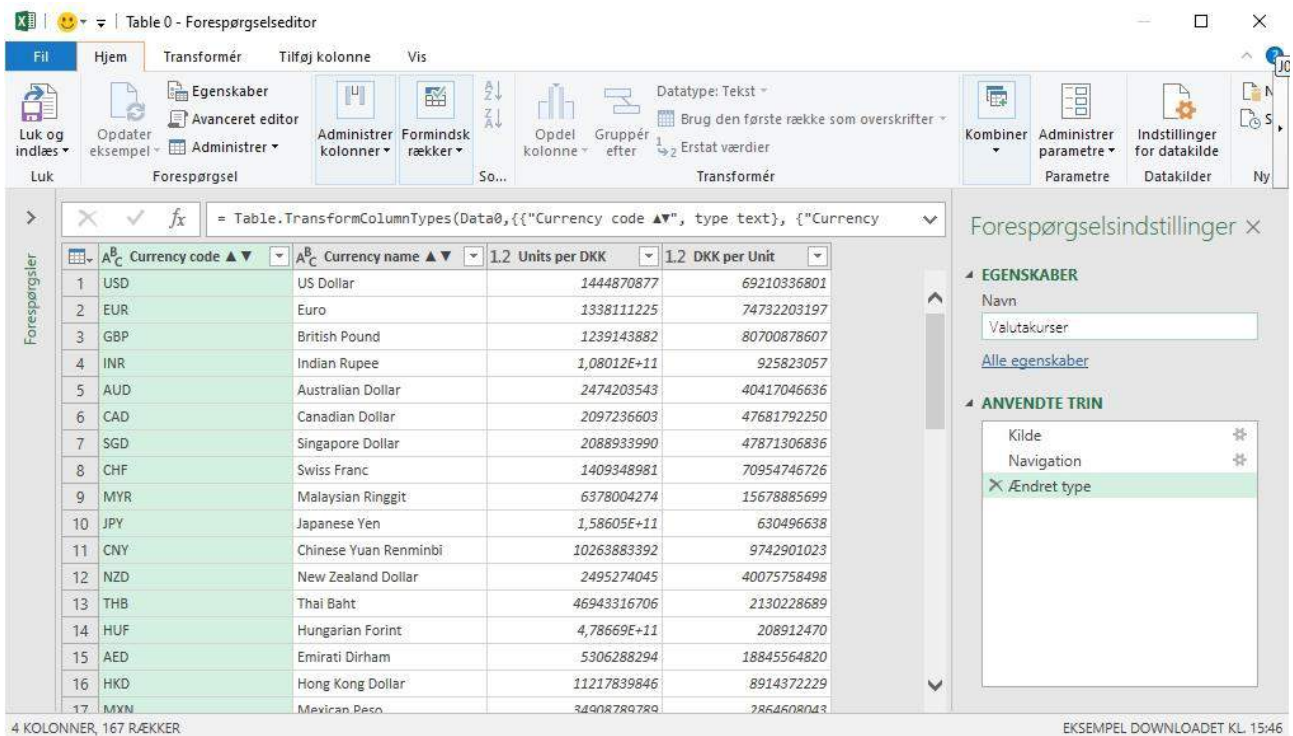


Den næste dialogboks, der vises, er dialogboksen **Navigator**. Da webtabeller fra websteder kan have dårlige navnekonventioner, er det ikke altid klart, hvilken tabel vi har brug for. Vi skal klikke gennem hver, indtil vi finder den, vi har brug for. Når du har fundet den rigtige tabel, skal du klikke på **Transformer data**.

## Kapitel 12 Power Query - Webdata



Når Forespørgselseditoren åbnes, indeholder den alle valutadata.



Giv forespørgslen et fornuftigt navn, f.eks. Valutakurser.

Klik på **Hjem** → **Luk og indlæs** for at indlæse dataene i Excel. Hvor nemt var det lige!

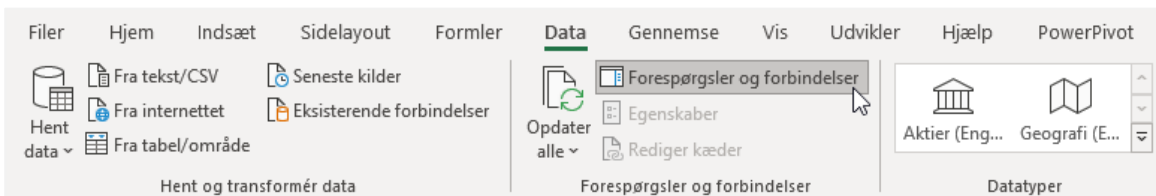
	A	B	C	D
1	Currency code ▲ ▼	Currency name ▲ ▼	Units per DKK ▼	DKK per Unit ▼
2	USD	US Dollar	0,14440822	6,92481358
3	EUR	Euro	0,133807234	7,473437487
4	GBP	British Pound	0,12379629	8,077786479
5	INR	Indian Rupee	10,79937712	0,092597933
6	AUD	Australian Dollar	0,246905634	4,050130348
7	CAD	Canadian Dollar	0,209741576	4,767771928
8	SGD	Singapore Dollar	0,208908881	4,786775925
9	CHF	Swiss Franc	0,140906209	7,096919346
10	MYR	Malaysian Ringgit	0,637428649	1,568803036
11	JPY	Japanese Yen	15,84873461	0,06309652
12	CNY	Chinese Yuan Renminbi	1,025822696	0,97482733
13	NZD	New Zealand Dollar	0,249042222	4,015383384
14	THB	Thai Baht	4,691701333	0,213142297
15	HUF	Hungarian Forint	47,75557752	0,020939962

Hvis websiden indeholder live-data, opdateres forespørgslen med disse live-data, hver gang du klikker på **Opdater**. Det er temmelig effektivt.

## Rediger forespørgslen

Lad os nu redigere forespørgslen, så vi kan bruge den med forskellige datoer og valutaer (undervejs vil vi støde på en fejl, som vi er nødt til at løse; bare rolig vi løser det).

I Excel vælger du **Data** → **Forespørgsler og forbindelser**.



Dobbeltklik på forespørgslen i panelet Projektmappe-forespørgsler for at åbne Forespørgselseditoren.



Forespørgselseditoren åbnes igen, klik på trinnet Kilde under de anvendte trin. URL-adressen er hardkodet i forespørgslen. Rediger M koden i formellinjen for at ændre forespørgslen så den indeholder en anden valuta og en anden dato. I vores eksempel har jeg valgt EUR og 1. januar 2020.

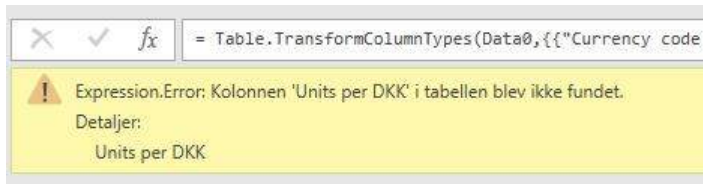
Følgende M-kode:

```
= Web.Page(Web.Contents("https://www.xe.com/currencytables/?from=DKK&date=2020-03-19"))
```

bliver til

```
= Web.Page(Web.Contents("https://www.xe.com/currencytables/?from=EUR&date=2020-01-01"))
```

Klik gennem de resterende trin i forespørgslen. Trinnet **Ændret type** afslører følgende fejl.



Når man ser på M koden i formellinjen, afsløres problemet; der er specifikke referencer til DKK-valutaen i overskriften (se de fede fremhævede afsnit nedenfor).

```
= Table.TransformColumnTypes(Data0,{{"Currency code ▲▼", type text}, {"Currency name ▲▼", type text}, {"Units per DKK", type number}, {"DKK per Unit", type number}})
```

Vores valgte valuta er nu EUR; kolonneoverskriften bliver **Units per EUR** og **EUR per Unit**.

Ændringen i kolonneoverskrifterne kommer direkte fra kildesiden; derfor er der intet, vi kan gøre for at stoppe det. Men der er noget, vi kan gøre for at omgå det.

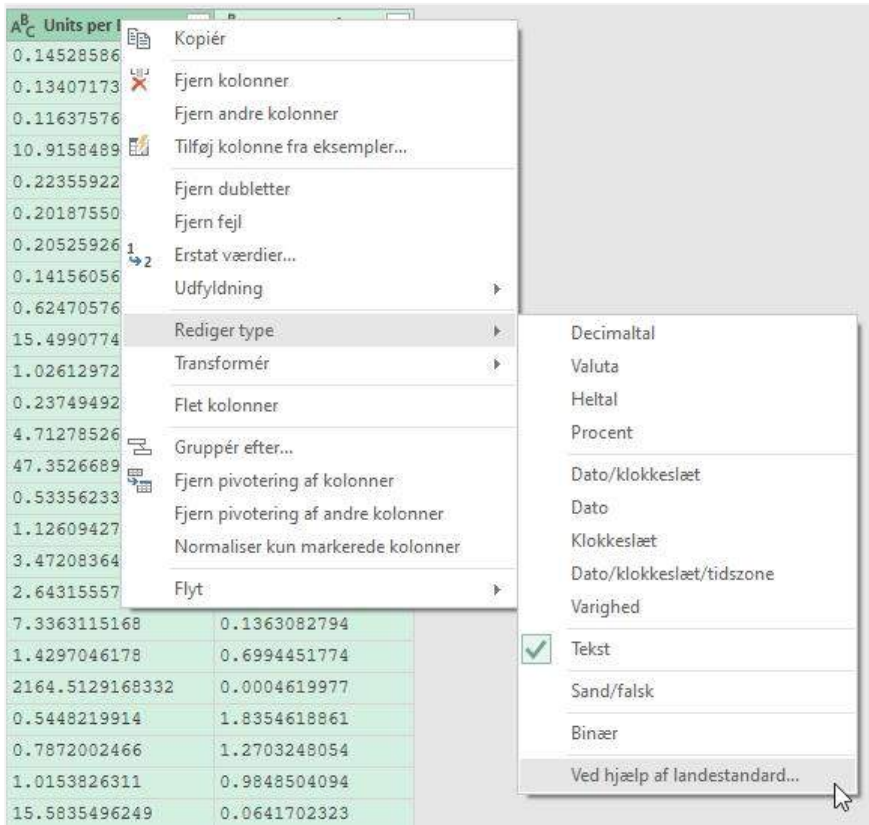
- Slet trinnet Ændret type, der forårsager fejlen.
- Degrader overskriften til den første række ved at klikke på **Transformer** → **Brug den første række som overskrifter** → **Anvend overskrifter som første række**.
- Slet den øverste række ved at klikke på **Hjem** → **Fjern rækker** → **Fjern øverste rækker**. Indtast 1 i feltet Antal rækker i dialogboksen Fjern de øverste rækker, klik på **OK**.
- Indtast egnede kolonnenavne manuelt (jeg har valgt Kode, Valuta, Kurs og Omvendt kurs).
- Anvend passende datatyper på hver kolonne (jeg har valgt Tekst, Tekst, Decimal, Decimal).

Bemærk at der her godt kan opstå problemer med datatyperne. I det her tilfælde er det hvordan Power Query opfatter værdierne. Vi har hentet værdier fra en amerikansk hjemmeside. Der bliver brugt punktum og ikke komma, som vi normalt plejer at bruge på dansk<sup>2</sup>.

Marker de to kolonner med værdier. Højre klik på en af kolonneoverskrifterne i markeringen og vælg **Rediger type** → **Ved hjælp af landestandarder**.

---

<sup>2</sup> Hvis du arbejder med engelsk/amerikansk valutakodning skal du bare fortsætte.



I dialogboksen **Rediger type med landstandard** vælger du i listefeltet **Datatype** **Decimaltal** og i listefeltet **Landstandard** vælger du **Engelsk (USA)**. Klik **OK**.





Eksempelvinduet ser nu sådan ud:

	A <sub>C</sub> Kode	A <sub>C</sub> Valuta	1.2 Kurs	1.2 Omvendt kurs
1	USD	US Dollar	1,070902492	0,933791832
2	EUR	Euro	1	1
3	GBP	British Pound	0,91682644	1,09071898
4	INR	Indian Rupee	80,11234589	0,012482471
5	AUD	Australian Dollar	1,823263481	0,548467081
6	CAD	Canadian Dollar	1,549557409	0,645345564
7	SGD	Singapore Dollar	1,552032349	0,644316467
8	CHF	Swiss Franc	1,053656035	0,949076328
9	MYR	Malaysian Ringgit	4,727005247	0,211550432
10	JPY	Japanese Yen	117,9569171	0,008477672
11	CNY	Chinese Yuan Renminbi	7,612533459	0,131362313
12	NZD	New Zealand Dollar	1,838599669	0,54389219
13	THB	Thai Baht	34,87489435	0,028673922

Klik på **Luk og indlæs** for at indlæse forespørgslen i Excel. Vi kan nu ændre valuta og datoer i kilde URL, hvilket betyder, at vi kan få kursen for enhver dato, vi ønsker.

### Brug af parametre med webforespørgsler

Det er ikke ideelt at redigere M koden, hver gang vi vil importere forskellige datoer og priser. Det er tidskrævende, og kan vi stole på andre brugere, der ikke ved noget om Power Query, om at opdatere trinnet korrekt?

Hvis du har fulgt med i denne Power Query bog, vil du huske, at vi tidligere har **oprettet parametre** til styring af variabler.

### Brug brugerdefinerede formler

I et tidligere kapitel **kombinerede vi alle filer fra en mappe**. For at opnå dette brugte vi funktionen Excel.Workbook() til at kombinere alle filerne, der var i mappen. Vi kan oprette vores egen funktion, som i stedet for at kombinere filer returnerer webforespørgsler baseret på de web-adresser, vi giver den.

Lad os starte med en ny tabel (jeg har kaldt tabellen for tblURLer). Den indeholder følgende værdier:

- Kode
- Dato
- URL

	A	B	C
1	Kode	Dato	URL
2	DKK	20-03-2020	https://www.xe.com/currencytables/?from=DKK&date=2020-03-20

Formlen i celle C2 konstrueres ved at sammenkæde valutakoden og datoen i URL-strukturen.

```
= "https://www.xe.com/currencytables/?from=" & [Kode] & "&date=" & TEKST ([Dato] ; "åååå-mm-dd")
```

Tilføj derefter denne tabel til Power Query. Marker en celle inde i tabellen og vælg derefter **Data → Fra tabel/område**.

Vi er nu klar til at oprette en brugerdefineret funktion.

### Oprettelse af funktionen

Lad os vende tilbage til den originale forespørgsel ved at klikke på forespørgslen Valutakurser fra forespørgselslisten.

## Kapitel 12 Power Query - Webdata

Vi skal ændre denne forespørgsel til en brugerdefineret funktion. Klik på **Hjem** → **Avanceret editor** for at få vist M koden.

Ret M koden som følgende:

Tilføj følgende kode øverst i forespørgslen:

```
(URL) as table =>
```

Skift kilde URL i koden til bogstaverne URL, som er den variabel, vi oprettede ovenfor.

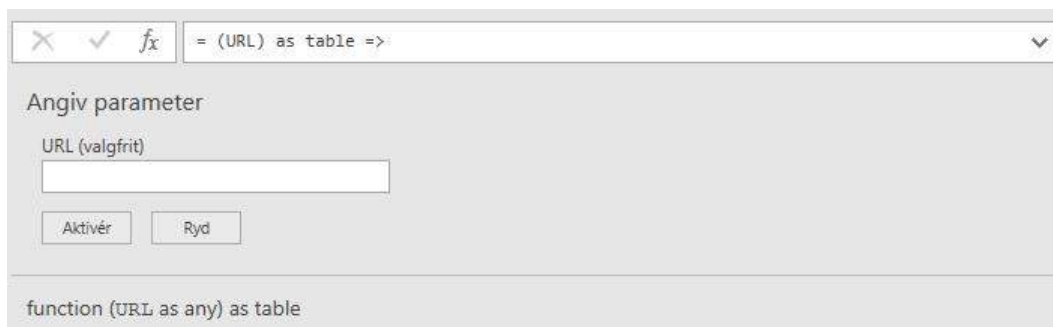
M-koden vil nu være som følger (de fremhævede sektioner er de dele, vi har ændret).

```
(URL) as table =>
```

```
let
    Kilde = Web.Page(Web.Contents(URL)),
    Data0 = Kilde{0}[Data],
    #Sænkede overskrifter = Table.DemoteHeaders(Data0),
    #Ændret type = Table.TransformColumnTypes(#Sænkede overskrifter,{{"Column1", type text}, {"Column2", type text}, {"Column3", type text}, {"Column4", type text}}),
    #Fjernede øverste rækker = Table.Skip(#Ændret type,1),
    #Omdøbte kolonner = Table.RenameColumns(#Fjernede øverste rækker,{{"Column1", "Kode"}, {"Column2", "Valuta"}, {"Column3", "Kurs"}, {"Column4", "Omvendt kurs"}}),
    #Ændret type med landestandard = Table.TransformColumnTypes(#Omdøbte kolonner,{{"Kurs", type number}, {"Omvendt kurs", type number}}, "en-US")
in
    #Ændret type med landestandard"
```

Klik på **Udført** i vinduet Advanced Editor for at acceptere ændringerne.

Forespørgselseditoren vil nu se sådan ud:



The screenshot shows the Power Query Advanced Editor interface. At the top, there is a title bar with a close button (X), a checkmark, and a function icon (fx). Below the title bar, there is a text box containing the function definition: `function (URL as any) as table`. The main area is titled "Angiv parameter" and contains a text box for "URL (valgfrit)" with "Aktivér" and "Ryd" buttons below it.

I forespørgseloversigten vises forespørgslen Valutakurser, vi netop har ændret, med et **fx** ved siden af, hvilket betyder, at det er en funktion.



## Brug af funktionen

Åbn forespørgslen tblURLer.

Klik på **Tilføj kolonne** → **Brugerdefineret kolonne**.

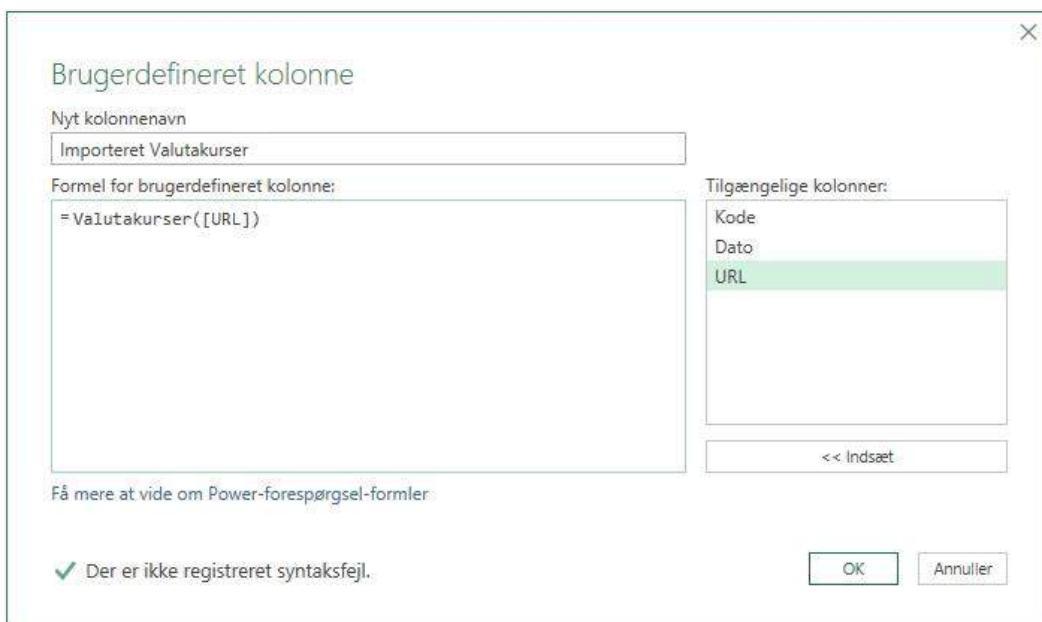
Indtast følgende i vinduet Brugerdefineret kolonne:

Nyt kolonnenavn: Importerede valutakurser

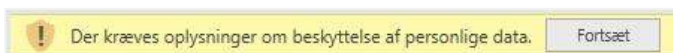
Tilpas kolonneformel:

```
=Valutakurser([URL])
```

Klik derefter på **OK**.



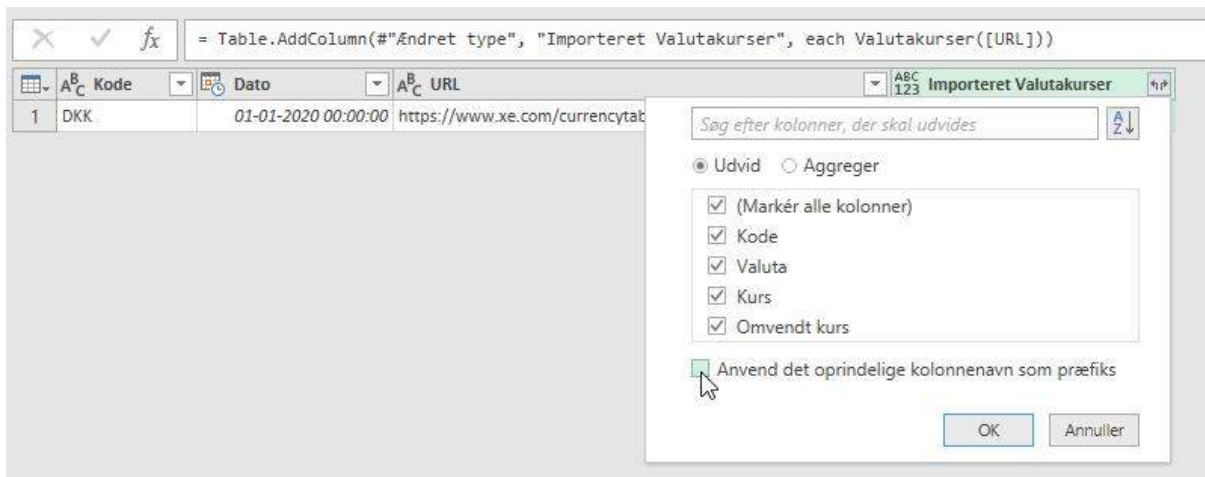
Klik på **Fortsæt** ved advarslen om beskyttelse af personlige oplysninger.



I dialogboksen Niveauer for beskyttelse af personlige oplysninger markeres afkrydsningsfeltet Ignorer personlige niveauer, og klik derefter på Gem.



Klik på **udvidelsesikonet** øverst i kolonnen Importeret valutakurser (den nye kolonne, vi lige har tilføjet). Fjern markeringen i afkrydsningsfeltet **Anvend det oprindelige kolonnenavn som præfiks**, og klik derefter på **OK**.



Dataene fra XE.com er importeret til forespørgslen.

Ryd op i forespørgslen ved at:

- Skift datatype for hver kolonne.
- Fjern de kolonner, der ikke er brug for.

Klik på **Luk og indlæs** for at indlæse data i Excel.

Det er tid til at teste dette for at bevise, at vi kan få valutakurser for enhver valuta og enhver dato. I Excel skal du ændre dato eller valutakode i tblURL tabellen. Klik på **Data** → **Opdater alle**, valutakurserne fra forespørgslen opdateres nu. Hvor smart er det lige?

## Importerer flere websider

Indtil videre har vi importeret en enkelt webside. Men hvad nu hvis vi ville importere flere websider? Hvis den brugerdefinerede funktion fungerer, vil dette være let.

Tilføj nogle flere rækker med data på tblURL tabellen.

## Kapitel 12 Power Query - Webdata

	A	B	C
1	Kode	Dato	URL
2	DKK	20-03-2020	<a href="https://www.xe.com/currencytables/?from=DKK&amp;date=2020-03-20">https://www.xe.com/currencytables/?from=DKK&amp;date=2020-03-20</a>
3	SEK	20-03-2020	<a href="https://www.xe.com/currencytables/?from=SEK&amp;date=2020-03-20">https://www.xe.com/currencytables/?from=SEK&amp;date=2020-03-20</a>
4	NOK	20-03-2020	<a href="https://www.xe.com/currencytables/?from=NOK&amp;date=2020-03-20">https://www.xe.com/currencytables/?from=NOK&amp;date=2020-03-20</a>

Klik på **Data** → **Opdater alle**.

Se tabellen over valutakurser. Power Query har importeret alle datoer og kurser inkluderet i tabellen tbiURLer. Vi kan få alle valutakurser for enhver dato for enhver basisvaluta.

Et par advarsler

Dette er effektivt ting. Men der er et par ting jeg vil gøre dig opmærksom på:

- Hvis du kalder for mange web-adresser, kan forespørgslen muligvis blive langsom til at opdateres.
- Hvis xe.com ændrer deres websted, kan forespørgslen ophøre med at fungere.
- Vi kan ikke hente data fra alle websteder, da nogle ikke er struktureret på den rigtige måde.

## Kapitel 13 Power Query – Fjern pivotering

Forberedelse af data er sandsynligvis den mest kritiske faktor i dataanalyse for at opnå succes. Hvis data layoutet er forkert, kommer du til at kæmpe med Excel; med at bruge avancerede formler, tilføje unødvendige kolonner eller endda kopiere data. Men hvis dataene er i den rigtige struktur, er det let at arbejde med Excel. Jeg plejede at blive overrasket over mennesker, der kunne bruge komplekse formler. Jeg noterede en masse tips og tricks for at prøve at lære de samme teknikker. Men jeg har fundet ud af, at det at få data i det rigtige format med disse avancerede færdigheder sjældent er nødvendigt.

Så hvad er det korrekte datalayout du skal bruge i Excel?

Før Power Pivot, ville det tage os lang tid at transformere data; vi måtte ty til kompleksitet eller lange komplicerede VBA-koder. Men Power Query har mange funktioner der gør denne transformation let. En af de mest nyttige funktioner er Fjern pivotering af kolonner, hvilket er det vi dækker i dette kapitel. Igennem tre eksempler vil vi bevæge os fra det lette til det vanskelige kilde dataformater. Men det bliver klart, at der er nogle få grundlæggende transformationer, der hvis de bruges på den rigtige måde, er alt, hvad det kræver.

Eksemplerne i dette kapitel bruger filen kaldet **Fjern pivotering.xlsx**.

Nu skal du fyre op under Excel, åbn eksempel filen. Vi er klar til at starte!

### Grundlæggende Fjern pivotering af kolonner

I vores første eksempel med Fjern pivotering af kolonner bruger vi regnearket Eksempel 1.

	A	B	C	D	E
1	Kunder	Produkt	Januar	Februar	Marts
2	Bilka	Produkt 1	96	120	0
3	Bilka	Produkt 2	98	199	66
4	Bilka	Produkt 3	0	103	191
5	Bilka	Produkt 4	141	78	138
6	Bilka	Produkt 5	159	167	65
7	Bilka	Produkt 6	175	50	115
8	Bilka	Produkt 7	185	50	100
9	Bilka	Produkt 8	0	199	0
10	Bilka	Produkt 9	0	89	0

Kildedataene viser en separat kolonne for hver måned. I stedet for at have en kolonne for hver måned er det bedre at have en kolonne, der indeholder månedsnavnet som en attribut sammen med en enkelt værdikolonne. Dette er det datalayout, vi arbejder mod i slutningen af dette eksempel.

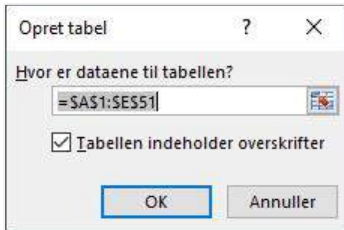
Vælg en hvilken som helst celle i dataområdet, og ændr det til en Excel tabel ved at klikke på **Indsæt** → **Tabel** (eller trykke på Ctrl + T).



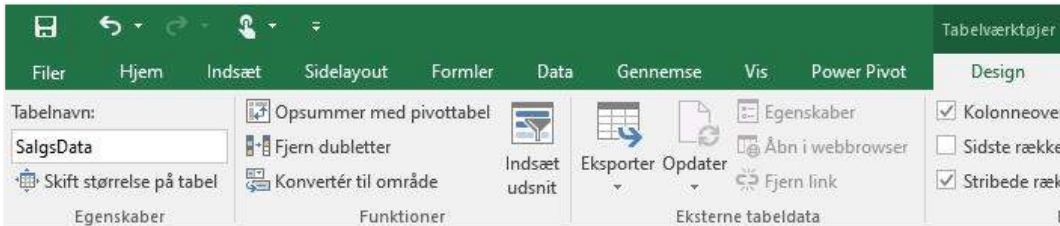
Dialogboksen **Opret tabel** vil automatisk finde dataområdet, og hvis der er en overskriftsrække. Kilde informationerne inkluderer et total i række 52, som vi kan udelukke fra dette dataområde.

Da vores data har en enkelt overskriftsrække, kan vi vælge indstillingen **Tabellen indeholder overskrifter** og derefter klikke på OK.

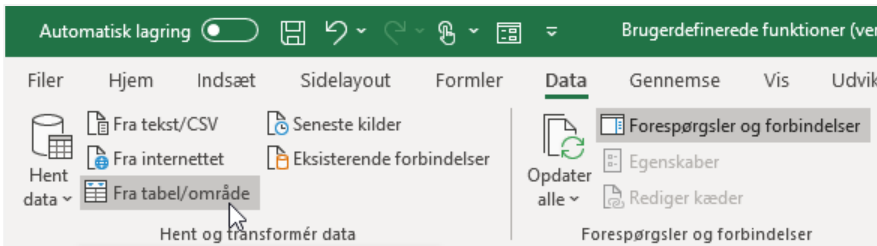
## Kapitel 13 Power Query – Fjern pivotering



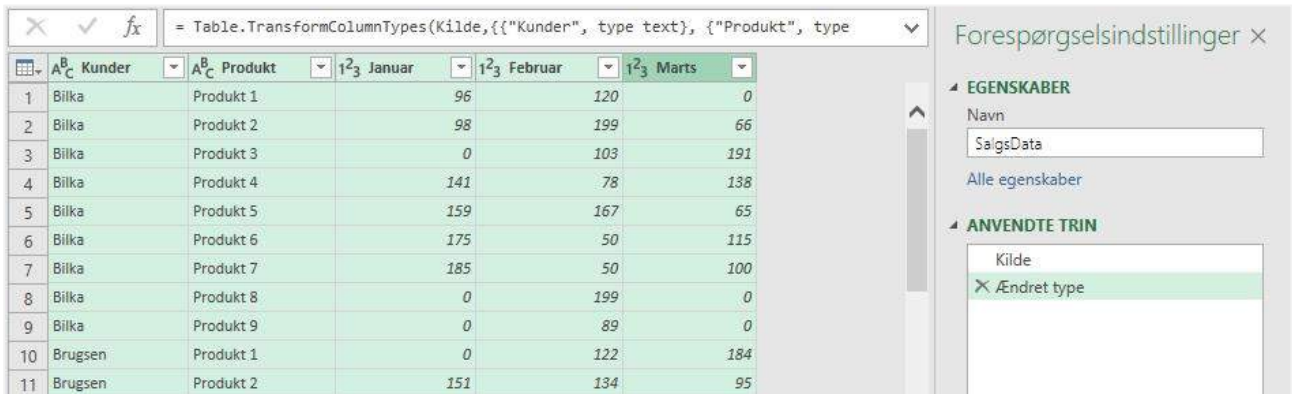
Klik på en hvilken som helst celle i tabellen, fanen Tabelværktøjer Design vises. Giv tabellen et fornuftigt navn (jeg har valgt SalgsData).



Nu er vi klar til at indlæse tabellen SalgsData i Power Query. Klik på **Data** → **Fra tabel/område**.



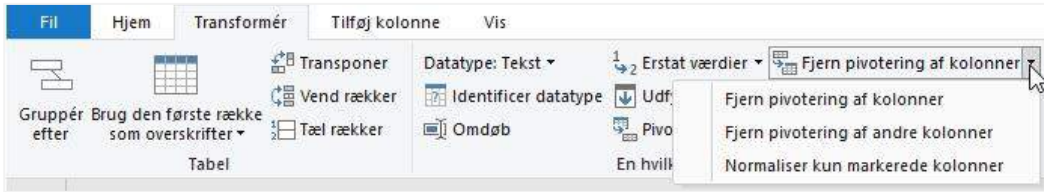
Power Query editoren åbnes og viser en forhåndsvisning af dataene.



Hvis der på et senere tidspunkt vil være flere kolonner eller forskellige kolonner, skal du slette trinnet **Ændret type**. Dette trin hardkoder kolonneoverskrifterne i M koden, så de ikke automatisk udvides eller formindskes efter forskellige kolonnelay-out.

Nu skal vi fjerne pivoteringen. Under **Transformer** → **Fjern pivotering** af kolonner er der tre indstillinger, den indstilling, vi vælger, afhænger af det resultat, vi ønsker.

## Kapitel 13 Power Query – Fjern pivotering



Fra disse tre indstillinger er der dog kun to resultater; vi kan fjerne pivoteringen på markerede eller ikke markerede kolonner. Hvad er forskellen, og er den vigtig?

Hvis vi skulle markere kolonnerne januar, februar og marts og anvende **Normaliser kun markerede kolonner**, vil pivoteringen på disse kolonner fjernes. Power Query opretter en ny kolonne med titlen **Attribut** (jeg har omdøbt den til **Måneder**), der indeholder månederne.

	A <sub>C</sub> Kunder	A <sub>C</sub> Produkt	A <sub>C</sub> Måneder	1.2 Værdi
1	Bilka	Produkt 1	Januar	96
2	Bilka	Produkt 1	Februar	120
3	Bilka	Produkt 1	Marts	0
4	Bilka	Produkt 2	Januar	98
5	Bilka	Produkt 2	Februar	199
6	Bilka	Produkt 2	Marts	66
7	Bilka	Produkt 3	Januar	0
8	Bilka	Produkt 3	Februar	103

M koden for denne transformation ville være:

```
= Table.Unpivot("#Ændret type", {"Januar", "Februar", "Marts"}, "Attribut", "Værdi")
```

Kolonneoverskrifterne fra januar, februar og marts er hardkodet i M koden. Så hvad sker der, hvis vi tilføjer yderligere data for april? Power Query fjerner ikke pivoteringen af kolonnen April, da den ikke er inkluderet i koden. Men hvis du sandsynligvis vil tilføje en ny attribut kolonne, fungerer det helt fint.

I stedet kunne vi markere kolonnerne Kunde og Produkt og bruge indstillingen **Fjern pivotering af andre kolonner**. For vores nuværende datasæt opnår vi det samme visuelle resultat i eksempelvisningsvinduet, men M koden er anderledes.

```
= Table.UnpivotOtherColumns("#Ændret type", {"Kunder", "Produkt"}, "Attribut", "Værdi")
```

Koden henviser ikke til kolonnerne for månederne: Derfor vil måneder der tilføjes, også få fjernet pivoteringen. Dette er smart til datakolonner, men ikke til attribut/dimensionskolonner.

Ryd nu op i forespørgslen med følgende transformationer:

- Skift overskriften på kolonnen **Attribut** til **Måneder**.
- Skift datatype for hver kolonne, så den passer til dataene i kolonnen (jeg har valgt tekst, tekst, tekst, decimaltal).
- Filtrer kolonnen **Værdi** for at fjerne nuller.



Klik på **Luk og indlæs** for at indlæse dataene i Excel. Regnearket ser sådan ud:

	A	B	C	D
1	Kunder	Produkt	Måneder	Værdi
2	Bilka	Produkt 1	Januar	96
3	Bilka	Produkt 1	Februar	120
4	Bilka	Produkt 2	Januar	98
5	Bilka	Produkt 2	Februar	199
6	Bilka	Produkt 2	Marts	66
7	Bilka	Produkt 3	Februar	103
8	Bilka	Produkt 3	Marts	191
9	Bilka	Produkt 4	Januar	141
10	Bilka	Produkt 4	Februar	78
11	Bilka	Produkt 4	Marts	138

Herefter kan vi bruge formler eller pivottabeller til at oprette en hvilken som helst oversigt. Dette er det vi får ud af dataforberedelser som denne.

## Fjern pivotering af flere kolonner

Nu er det tid på Eksempel 2. Kildedataene ser sådan ud:

I stedet for en kolonne for hver måned, har vi nu to, der viser salgsværdier og salgsenheder. Det mest nyttige format er at have månedsnavne i en kolonne med værdien og enhederne som separate kolonner.

	A	B	C	D	E	F	G	H
1	Kunder	Produkt	Januar	Januar	Februar	Februar	Marts	Marts
2	Bilka	Produkt 1	Værdi	Enhed	Værdi	Enhed	Værdi	Enhed
3	Bilka	Produkt 1	159	80	167	63	65	70
4	Bilka	Produkt 2	98	106	199	61	66	81
5	Bilka	Produkt 3	185	60	50	72	100	105
6	Bilka	Produkt 4	175	94	50	82	115	94
7	Bilka	Produkt 5	141	98	78	105	138	97
8	Bilka	Produkt 6	0	0	103	68	191	110
9	Bilka	Produkt 7	0	0	89	50	0	0
10	Bilka	Produkt 8	96	97	120	119	0	0
11	Bilka	Produkt 9	0	0	199	108	0	0
12	Brugsen	Produkt 1	151	71	134	108	95	68

De indledende trin er de samme som eksempel 1:

- Konverter dataene til en tabel (brug den første række som overskrift og ekskluder det samlede antal fra området).
- Giv tabellen et nyttigt navn.
- Klik på **Data** → **Fra tabel/område** for at indlæse dataene i Power Query.
- Bare for at være sikker, skal du slette trinnet **Ændret type**, der automatisk blev oprettet af Power Query.

## Kapitel 13 Power Query – Fjern pivotering

Eksempelvinduet skal se ud som følger:

	A <sup>B</sup> <sub>C</sub> Kunder	A <sup>B</sup> <sub>C</sub> Produkt	1 <sup>2</sup> <sub>3</sub> Januar V...	1 <sup>2</sup> <sub>3</sub> Januar En...	1 <sup>2</sup> <sub>3</sub> Februar V...	1 <sup>2</sup> <sub>3</sub> Februar En...	1 <sup>2</sup> <sub>3</sub> Marts V...	1 <sup>2</sup> <sub>3</sub> Marts E...
1	Bilka	Produkt 1	159	80	167	63	65	70
2	Bilka	Produkt 2	98	106	199	61	66	81
3	Bilka	Produkt 3	185	60	50	72	100	105
4	Bilka	Produkt 4	175	94	50	82	115	94
5	Bilka	Produkt 5	141	98	78	105	138	97
6	Bilka	Produkt 6	0	0	103	68	191	110
7	Bilka	Produkt 7	0	0	89	50	0	0
8	Bilka	Produkt 8	96	97	120	119	0	0
9	Bilka	Produkt 9	0	0	199	108	0	0
10	Brugsen	Produkt 1	151	71	134	108	95	68

Nu skal vi foretage en hel del transformationer.

- Marker kolonner kunde og produkt, og klik på **Transformer** → **Fjern pivotering af kolonner (listefeltet)** → **Fjern pivotering af andre kolonner**.
- Marker kolonnen **Attribut**, og opdel kolonnen i de dele, vi har brug for.

I vores eksempel er vi nødt til at opdele efter Linjeskift. Klik på **Transformer** → **Opdel kolonne** → **Opdel efter afgrænsere**. Anvend derefter indstillingerne vist nedenfor. Power Query vil sandsynligvis gætte rigtigt, og har udfyldt dialogboksen for dig med de rigtige indstillinger.

### Opdel kolonne efter afgrænsere

Angiv den afgrænser, der bruges til at opdele tekstkolonnen.

Vælg eller angiv afgrænser

--Brugerdefineret--

#{lf}

Opdel ved

Afgrænser længst til venstre

Afgrænser længst til højre

Hver forekomst af afgrænseren

Avancerede indstillinger

Opdel i

Kolonner

Rækker

Antallet af kolonner, der skal opdeles i

2

Anførselstegn

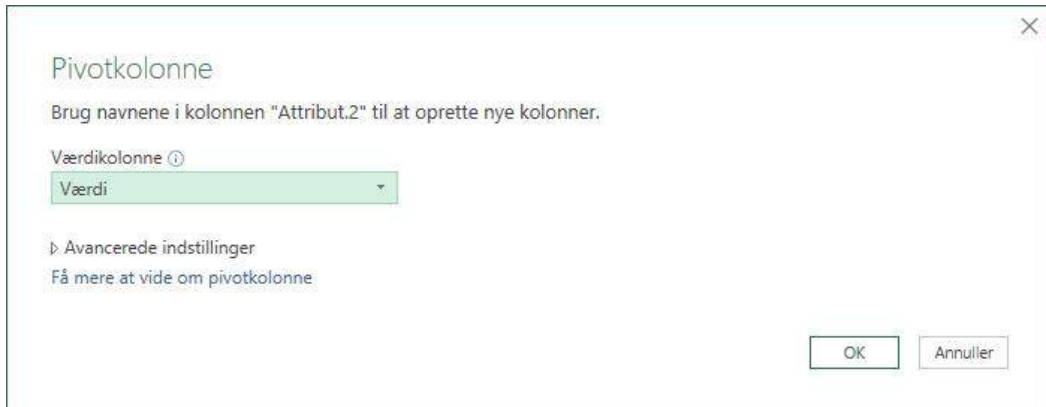
"

Opdel med specialtegn

Indsæt specialtegn

OK Annuller

- Kolonnen Attribut bliver nu opdelt i to separate kolonner.
- Marker kolonne, der viser Værdi og Enhed, og klik derefter på **Transformerer** → **Pivotkolonne**.
- I dialogboksen **Pivotkolonne**, vælges kolonne der indeholder tallene, og klik derefter på **OK**.



- Ret navnet på kolonnen Attribut til Måneder.
- Ret datatypen for alle kolonnerne.

Klik til sidst på **Luk og Indlæs**.

De ikke pivoterede data i Excel ser nu sådan ud ... perfekt!

	A	B	C	D	E
1	Kunder	Produkt	Måneder	Værdi	Enhed
2	Bilka	Produkt 1	Februar	167	63
3	Bilka	Produkt 1	Januar	159	80
4	Bilka	Produkt 1	Marts	65	70
5	Bilka	Produkt 2	Februar	199	61
6	Bilka	Produkt 2	Januar	98	106
7	Bilka	Produkt 2	Marts	66	81
8	Bilka	Produkt 3	Februar	50	72
9	Bilka	Produkt 3	Januar	185	60
10	Bilka	Produkt 3	Marts	100	105

## Fjern pivotering med flere overskriftsrækker

Nu til vores sidste eksempel. Alt er det samme som eksempel 2 bortset fra tabeloverskrifterne. Vi har nu to overskriftsrækker, den første viser måneder og den anden viser Værdi eller Enhed.

	A	B	C	D	E	F	G	H
1			Januar		Februar		Marts	
2	Kunder	Produkt	Værdi	Enhed	Værdi	Enhed	Værdi	Enhed
3	Bilka	Produkt 1	159	80	167	63	65	70
4	Bilka	Produkt 2	98	106	199	61	66	81
5	Bilka	Produkt 3	185	60	50	72	100	105
6	Bilka	Produkt 4	175	94	50	82	115	94
7	Bilka	Produkt 5	141	98	78	105	138	97
8	Bilka	Produkt 6	0	0	103	68	191	110
9	Bilka	Produkt 7	0	0	89	50	0	0
10	Bilka	Produkt 8	96	97	120	119	0	0
11	Bilka	Produkt 9	0	0	199	108	0	0

## Kapitel 13 Power Query – Fjern pivotering

Excel tabeller og data i Power Query kan kun have en enkelt overskriftsrække. Dette er god datahåndteringspraksis. Ved at opdele overskrifterne i to rækker bliver det et præsenteringsformat, som er meget vanskeligt at arbejde med i Excel.

Lad os starte med at omdanne disse data til en Excel tabel. Der er en afgørende forskel i forhold til de foregående eksempler. Vælg ikke indstillingen Tabellen indeholder overskrifter.



Når du har klikket på **OK**, er der foretaget nogle væsentlige ændringer i kildedataene.

- En overskriftsrække, der viser Kolonne1, Kolonne2, Kolonne3 osv. er tilføjet.
- Alle rækker er blevet flyttet en række ned for at give plads til den nye overskriftsrække.
- De flettede celler er ikke længere flettet med indholdet placeret i den første celle i hvert flettet område.

	A	B	C	D	E	F	G	H
1	Kolonne1	Kolonne2	Kolonne3	Kolonne4	Kolonne5	Kolonne6	Kolonne7	Kolonne8
2	Januar		Februar		Marts			
3	Kunder	Produkt	Værdi	Enhed	Værdi	Enhed	Værdi	Enhed
4	Bilka	Produkt 1	159	80	167	63	65	70
5	Bilka	Produkt 2	98	106	199	61	66	81
6	Bilka	Produkt 3	185	60	50	72	100	105
7	Bilka	Produkt 4	175	94	50	82	115	94
8	Bilka	Produkt 5	141	98	78	105	138	97
9	Bilka	Produkt 6	0	0	103	68	191	110
10	Bilka	Produkt 7	0	0	89	50	0	0
11	Bilka	Produkt 8	96	97	120	119	0	0
12	Bilka	Produkt 9	0	0	199	108	0	0

Importer dataene til Power Query ved at vælge en hvilken som helst celle i tabellen og derefter klikke på **Data → Fra tabel/område**.

Eksempelvinduet viser følgende:

	A <sub>C</sub> Kolonne1	A <sub>C</sub> Kolonne2	ABC 123 Kolonne3	ABC 123 Kolonne4	ABC 123 Kolonne5	ABC 123 Kolonne6	ABC 123 Kolonne7	ABC 123 Kolonne8
1	null	null	Januar	null	Februar	null	Marts	null
2	Kunder	Produkt	Værdi	Enhed	Værdi	Enhed	Værdi	Enhed
3	Bilka	Produkt 1	159	80	167	63	65	70
4	Bilka	Produkt 2	98	106	199	61	66	81
5	Bilka	Produkt 3	185	60	50	72	100	105
6	Bilka	Produkt 4	175	94	50	82	115	94
7	Bilka	Produkt 5	141	98	78	105	138	97

Denne visning giver os et problem: den viser ikke måneden i hver kolonne. Hvis vi forsøger at fjerne pivoteringen, vil den fjerne pivoteringen af null værdien. Vi skal først bruge transformationen Udfyld henover... men der er ikke en! I stedet er vi nødt til at transponere dataene, udfylde, foretage nogle transformationer og derefter transponere tilbage igen.

- Klik på **Transformer → Transponer**.

## Kapitel 13 Power Query – Fjern pivotering

- Marker Kolonne1, klik på **Transformerer** → **Udfyldning** → **Nedad**.
- Marker Kolonne1 og Kolonne2.
- Klik på **Transformerer** → **Flet kolonner**.
- I dialogboksen Flet kolonner skal du vælge en separator, der ikke findes i teksten i de to kolonner (jeg har valgt et mellemrum, men dette er muligvis ikke den bedste mulighed i dit tilfælde).
- Klik på **Transformerer** → **Transponer** for at returnere dataene til det forrige format.
- Marker Kolonne1 og Kolonne2.
- Klik på **Transformerer** → **Format** → **Trim** for at fjerne mellemrumstegnet fra kolonnerne kunde og produkt, vi tilføjede i et tidligere trin.
- Fremhæv nu den første række som overskrift ved at klikke på **Transformerer** → **Brug første række som overskrifter**.

Efter alle disse transformationer vil forhåndsvisningsvinduet se sådan ud:

	A <sup>B</sup> <sub>C</sub> Kunder	A <sup>B</sup> <sub>C</sub> Produkt	1 <sup>2</sup> <sub>3</sub> Januar Værdi	1 <sup>2</sup> <sub>3</sub> Januar Enhed	1 <sup>2</sup> <sub>3</sub> Februar V...	1 <sup>2</sup> <sub>3</sub> Februar V...	1 <sup>2</sup> <sub>3</sub> Marts V...	1 <sup>2</sup> <sub>3</sub> Marts V...
1	Bilka	Produkt 1	159	80	167	63	65	70
2	Bilka	Produkt 2	98	106	199	61	66	81
3	Bilka	Produkt 3	185	60	50	72	100	105
4	Bilka	Produkt 4	175	94	50	82	115	94
5	Bilka	Produkt 5	141	98	78	105	138	97
6	Bilka	Produkt 6	0	0	103	68	191	110
7	Bilka	Produkt 7	0	0	89	50	0	0
8	Bilka	Produkt 8	96	97	120	119	0	0
9	Bilka	Produkt 9	0	0	199	108	0	0
10	Brugsen	Produkt 1	151	71	134	108	95	68

Det er nu det samme format som vi havde i eksempel 2. For at gennemføre dette eksempel skal du følge alle transformationstrinnene i dette eksempel.

## Kapitel 14 Power Query – udpakning af data i en kolonne

I vores færden gennem Power Query bogen, er vi nået til et almindeligt og frustrerende dataformat; data stablet i en kolonne. I dette kapitel undersøger vi en grundlæggende metode til udpakning af data.

Eksemplet vi ser nedenfor, kunne meget vel være oprettet af en person, der ikke vidste, hvordan man strukturerer data korrekt, men dette format kan ligeledes findes i data, der eksporteres, eller kopieres fra et it-system (skøn eksport eller kopiering af data har en tendens at være en enkelt kolonne, som er lidt lettere at håndtere).

Eksemplerne i dette kapitel bruger filen kaldet **Udpakning af data i en kolonne.xlsx**.

Skærmbillede herunder viser kolonnen Adresse har data stablet i en kolonne.

	A	B	C
1	Kunder	Adresse	Telefon
2	Brugsen	Valbyvej 5	56845712
3		Valby	
4		2500	
5			
6	Irma	Søvej 44	69547123
7		Helsingør	
8		3000	
9			
10	Rema 1000	Blomstergade 112	45871255
11		Århus	
12		8000	
13			
14	Bilka	Ådalen 7	46845236
15		Vejle	
16		7000	
17			
18			

Som du kan forestille dig, er det meget vanskeligt at manipulere data i dette layout. For at gøre det brugbart skal hver række i adressen være en separat kolonne. Power Query redder os endnu en gang.

### Udpakning af en kolonne med data

Åbn regnearket der følger med denne opgave. Dataene ligner det skærmbillede, vi så ovenfor.

Dataene er allerede i et tabelformat, som jeg har navngivet Kunder. Hvis du vil importere dataene til Power Query, skal du vælge en hvilken som helst celle i tabellen og klikke på **Data → Fra tabel/område**.

Power Query vinduet åbner og viser dataene, der ser ud som skærmbilledet nedenfor:

## Kapitel 14 Power Query – Udpakning af data i en kolonne

	Kunder	Adresse	Telefon
1	Brugsen	Valbyvej 5	56845712
2	null	Valby	null
3	null	2500	null
4	null	null	null
5	Irma	Søvej 44	69547123
6	null	Helsingør	null
7	null	3000	null
8	null	null	null
9	Rema 1000	Blomstergade 112	45871255
10	null	Århus	null
11	null	8000	null
12	null	null	null
13	Bilka	Ådalen 7	46845236
14	null	Vejle	null
15	null	7000	null
16	null	null	null

### Rens importen

Lad os nu gennemgå trinnene for at få de separate adresselinjer omdannet til kolonner. Lad os først fjerne alle tomme rækker ved at filtrere adressekolonnen for at fjerne nulværdier.

INNER: 16 RÆKKER

Klik på kolonnen Kunder, og klik derefter på **Transformerer** → **Udfyldning** → **Nedad**. Dette sikrer, at kundenavnet står i hver række; der skal ikke være nulværdier i kolonnen Kunder.

## Kapitel 14 Power Query – Udpakning af data i en kolonne

The screenshot shows the Power Query interface with the 'Tilføj kolonne' tab selected. The 'Udfyldning' dropdown menu is open, showing options like 'Nedad' and 'Opad'. Below the ribbon, a table with columns 'Kunder', 'Adresse', and 'Telefon' is visible. The 'Adresse' column contains three rows of data for each customer, with some cells containing 'null'.

Kunder	Adresse	Telefon
1 Brugsen	Valbyvej 5	56845712
2 null	Valby	null
3 null	2500	null
4 Irma	Søvej 44	69547123
5 null	Helsingør	null
6 null	3000	null
7 Rema 1000	Blomstergade 112	45871255
8 null	Århus	null
9 null	8000	null
10 Bilka	Ådalen 7	46845236
11 null	Vejle	null
12 null	7000	null

### Tilføj en indeks- og modulus kolonne

I de næste par transformationer begynder ting at virke lidt underligt. Når alle trin er afsluttet, giver det mening, men indtil da er du bare nødt til at stole på mig.

Der er 3 rækker for hver kundeadresse. Vores næste transformation går ud på at inkludere en kolonne med tal, der viser tallene 0, 1 og 2 gentagne gange. 0 repræsenterer den første række af adressen, 1 den anden række af adressen og 2 den sidste linje i adressen.

Klik på **Tilføj kolonne** → **Indekskolonne**.

The screenshot shows the Power Query interface with the 'Tilføj kolonne' tab selected. The 'Indekskolonne' dropdown menu is open, showing options like 'Betinget kolonne', 'Indekskolonne', and 'Dupliker kolonne'.

En indeksskolonne er tilføjet, der starter ved nul.

Marker den nye indeksskolonne, og klik derefter på **Tilføj kolonne** → **Standard** → **Modulus**.



## Kapitel 14 Power Query – Udpakning af data i en kolonne

Generelt

Fra tekst

Standard

Læg sammen  
Multipliser  
Træk fra  
Divider  
Divider (heltal)  
Modulus

Opret en ny kolonne, der indeholder værdierne i den første markerede kolonne divideret (modulus) med værdierne i den anden markerede kolonne. Hvis en enkelt kolonne er markeret, indeholder den nye kolonne værdierne i den markerede kolonne divideret (modulus) med en angivet værdi.

	A <sup>B</sup> <sub>C</sub> Kunder	ABC <sub>123</sub> Adresse	1 <sup>2</sup> <sub>3</sub> Telefon	1.2 Indeks
1	Brugsen	Valbyvej 5	56845712	0
2	Brugsen	Valby	null	1
3	Brugsen	2500	null	2
4	Irma	Søvej 44	6954	
5	Irma	Helsingør		
6	Irma	3000		
7	Rema 1000	Blomstergade 112	4587	
8	Rema 1000	Århus		
9	Rema 1000	8000	null	8
10	Bilka	Ådalen 7	46845236	9
11	Bilka	Vejle	null	10
12	Bilka	7000	null	11

Indtast 3 i dialogboksen **Modulus**, da der er 3 rækker på hver adresse. Klik på **OK**.

### Modulus

Angiv et tal, du vil finde restværdien fra for de enkelte værdier i kolonnen.

Værdi

ABC<sub>123</sub>

Hvis du har fulgt alle trinene over eksempelvisningsvinduet, skal din opstilling se ud som skærbillede nedenfor:

	A <sup>B</sup> <sub>C</sub> Kunder	ABC <sub>123</sub> Adresse	1 <sup>2</sup> <sub>3</sub> Telefon	1.2 Indeks	1.2 Modulus
1	Brugsen	Valbyvej 5	56845712	0	0
2	Brugsen	Valby	null	1	1
3	Brugsen	2500	null	2	2
4	Irma	Søvej 44	69547123	3	0
5	Irma	Helsingør	null	4	1
6	Irma	3000	null	5	2
7	Rema 1000	Blomstergade 112	45871255	6	0
8	Rema 1000	Århus	null	7	1
9	Rema 1000	8000	null	8	2
10	Bilka	Ådalen 7	46845236	9	0
11	Bilka	Vejle	null	10	1
12	Bilka	7000	null	11	2

Marker kolonnen Modulus, vi oprettede. Klik på **Transformerer** → **Pivotkolonne**. I vinduet **Pivotkolonne** skal du vælge kolonnen **Adresse**, udvide de avancerede indstillinger og vælg **Aggreger ikke**, og klik derefter på **OK**.

## Kapitel 14 Power Query – Udpakning af data i en kolonne

Pivotkolonne

Brug navnene i kolonnen "Modulus" til at oprette nye kolonner.

Værdikolonne ⓘ

Adresse ▼

Avancerede indstillinger

Funktion til samlet værdi ⓘ

Aggreger ikke ▼

Få mere at vide om pivotkolonne

OK Annuller

Se forhåndsvisningsvinduet, du tror sandsynligvis, at vi har ødelagt det hele, men det er alt sammen en del af processen. Vi er ved at bringe orden til opstillingen.

	A <sup>B</sup> <sub>C</sub> Kunder	1 <sup>2</sup> <sub>3</sub> Telefon	1.2 Indeks	ABC 123 0	ABC 123 1	ABC 123 2
1	Bilka	null	10	null	Vejle	null
2	Bilka	null	11	null	null	7000
3	Bilka	46845236	9	Ådalen 7	null	null
4	Brugsen	null	1	null	Valby	null
5	Brugsen	null	2	null	null	2500
6	Brugsen	56845712	0	Valbyvej 5	null	null
7	Irma	null	4	null	Helsingør	null
8	Irma	null	5	null	null	3000
9	Irma	69547123	3	Søvej 44	null	null
10	Rema 1000	null	7	null	Århus	null
11	Rema 1000	null	8	null	null	8000
12	Rema 1000	45871255	6	Blomstergade 112	null	null

Marker kolonnerne med overskrifterne 1 og 2, og anvend derefter **Transformerer** → **Udfyldning** → **Nedad**.

Marker kolonnen 0, og filtrer for at fjerne nulværdierne. Eksempelvinduet viser nu dataene i følgende format:

	A <sup>B</sup> <sub>C</sub> Kunder	1 <sup>2</sup> <sub>3</sub> Telefon	1.2 Indeks	ABC 123 0	ABC 123 1	ABC 123 2
1	Bilka	46845236	9	Ådalen 7	Vejle	7000
2	Brugsen	56845712	0	Valbyvej 5	Valby	2500
3	Irma	69547123	3	Søvej 44	Helsingør	3000
4	Rema 1000	45871255	6	Blomstergade 112	Århus	8000

Smart ikke? Vi gik fra at tro, at alt var ødelagt, til et pænt og nydeligt datasæt ved hjælp af et par grundlæggende transformationer.

En af de kritiske færdigheder i Power Query er evnen til at foretage masser af små transformationer, som til sidst tilføjer dataene i det rigtige format. Ofte kan noget, der kan virke som en lille transformation, såsom udpakning af data, kræve en masse mindre trin.

## Efterbehandling

Nogle få enkle transformationer, der renser alt sammen:

- Fjern indeksskolonnen (marker kolonnen, og vælg **Hjem** → **Fjern kolonner**).
- Omdøb kolonner 0, 1 og 2 (dobbeltklik på hver kolonneoverskrift og tilføj et meningsfuldt navn).

Det var det. Dataene er nu klar. Klik på **Luk og indlæs** for at overføre dataene til Excel.

	A	B	C	D	E
1	Kunder	Telefon	Adresse	By	Postnr
2	Bilka	46845236	Ådalen 7	Vejle	7000
3	Brugsen	56845712	Valbyvej 5	Valby	2500
4	Irma	69547123	Søvej 44	Helsingør	3000
5	Rema 1000	45871255	Blomstergade 112	Århus	8000

## Hvad har vi lært?

I dette kapitel har vi set, hvordan man udpakker en datakolonne. Men endnu vigtigere er, at resultater, der kan synes elementære, kan være ret komplicerede. At forstå enkle transformationer og hvordan man kombinerer dem sammen, er noget, der tager tid og træning at lære.

Der er andre metoder til at pakke data ud, som kræver færre trin, men de er mere komplekse at forstå for en nybegynder, så de er uden for vores område i denne begynderbog.

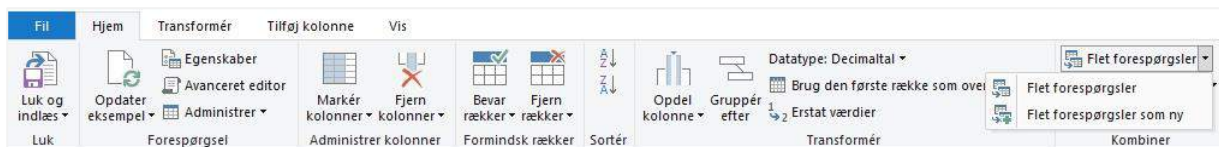
## Kapitel 15 Power Query- Opslagsværdier ved hjælp af fletning

At slå data op fra en tabel eller et regneark er sandsynligvis den mest almindelige opgave, der udføres af Excel brugere når der skal oprettes rapporter. At lære at bruge LOPSLAG er for mange deres første indblik i Excels rækkevidde. Men hvad sker der når du bruger Power Query? Der findes ikke en LOPSLAG funktion, så hvordan kan vi søge efter data i en anden tabel?

Der er tre typer opslag, en nøjagtig match, en omtrentlig match og et uklart match.

- Et eksakt match er det mest almindelige opslag og kræver, at opslagsværdierne er identiske.
- Omtrentlige match finder værdien over (eller nedenfor) den søgte værdi.
- Uklart match finder værdier baseret på, hvordan de ligner andre værdier ved hjælp af en slags algoritme.

Power Query kan udføre alle disse typer, selvom vi i denne lektion primært vil fokusere på de to første. For at udføre et opslag i Power Query bruger vi transformationen Flet, der findes under fanen Hjem.



Eksempler i denne lektion bruger vi filen Opslag Data.xlsx

I filen er der tre tabeller:

- Salg - Indeholder månedlige salgsdata
- Kunder - Indeholder kundekontaktoplysninger
- Rabatter - Rabatten, der gives til kunderne ved køb af specifikke beløb.

	A	B	C	D	E	F	G	H	I	J
1	Kunder	Produkt	Måned	Værd		Kunder	Telefon	Adresse	By	Postnr
2	Netto	Kylling	Januar	122		Netto	613780627	Alfavej 5	Valby	2500
3	Netto	Kylling	Marts	46		Irma	448107275	Betagde 6	Valby	2500
4	Netto	Mel	Februar	55		Bilka	490391249	Gamma strædet 47	Roskilde	4000
5	Netto	Rugbrød	Februar	110		Fakta	714880427	Epsilon plads 12	Valby	2500
6	Netto	Rugbrød	Marts	58		Rema 1000	412996856	Myvej 7	Valby	2500
7	Irma	Æbler	Januar	174		Føtex	395735980	Sigma plads 8	Roskilde	4000
8	Irma	Æbler	Marts	115		Lidl	681215925	Rhovej 66	København	1100
9	Irma	Kylling	Marts	56		Spar	216801627	Deltastrædet 89	Roskilde	4000
10	Irma	Mel	Februar	131						
11	Irma	Mel	Marts	46						
12	Irma	Fisk	Januar	129						
13	Irma	Fisk	Februar	350		Rabatgrænse	Rabat værdi			
14	Irma	Fisk	Marts	206		500	2%			
15	Irma	Rugbrød	Januar	50		750	5%			
16	Irma	Rugbrød	Februar	76		1.000	10%			

For at illustrere, hvordan vi kan slå data op i Power Query, opretter vi to rapporter.

*Samlet salg efter by (opslag med et nøjagtig match)*

Salgsværdierne og byen for hver kunde er i separate tabeller; derfor er vi nødt til at bruge flettefunktionen til at udføre opslaget.

## Kapitel 15 Power Query - Opslagsværdier ved hjælp af fletning

	A <sup>B</sup> C By	1.2 Samlet salg
1	Valby	2711
2	Roskilde	3429
3	København	629

Samlet rabat efter kunde (opslag med en omtrentlig match)

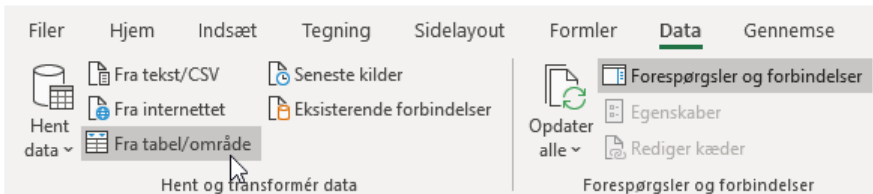
For at finde rabatten for hver kunde skal vi udføre et omtrentligt match. Vi ønsker ikke nødvendigvis at matche den nøjagtige værdi, men finde det interval, hvor salgsværdien falder. Vi vil stadig bruge fletfunktionen, men vi har brug for et par ekstra transformationer for at få det rigtige slutresultat.

	A <sup>B</sup> C Kunder	1.2 Samlet salg	1.2 Rabat værdi	1.2 Multiplikation
1	Netto	391	null	null
2	Lidl	629	0,02	12,58
3	Rema 1000	917	0,05	45,85
4	ICA	947	0,05	47,35
5	Spar	1059	0,1	105,9
6	Bilka	1094	0,1	109,4
7	Mega Mart	1108	0,1	110,8
8	Føtex	1276	0,1	127,6
9	Irma	1403	0,1	140,3

### Indlæs dataene i Power Query

Start med at indlæse de tre tabeller i Power Query.

Klik på en hvilken som helst celle i salgstabellen, og vælg derefter **Data** → **Fra tabel/område** i Excel menuen.



Power Query editoren åbnes. Jeg har forsøgt at opsætte tabellerne, så de kræver så få transformationer som muligt, men i den virkelige verden er det sjældent så enkelt.

## Kapitel 15 Power Query - Opslagsværdier ved hjælp af fletning

The screenshot shows the Power Query Editor window titled 'Salg - Forespørgselseditor'. The ribbon includes 'Fil', 'Hjem', 'Transformér', 'Tilføj kolonne', and 'Vis'. The 'Transformér' tab is active, showing options like 'Opdel kolonne', 'Gruppér efter', and 'Erstat værdier'. The data table has the following columns: 'Kunder', 'Produkt', 'Måned', and 'Værdi'. The table contains 23 rows of data. The right-hand pane shows 'Forespørgselsindstillinger' with sections for 'EGENSKABER' and 'ANVENDTE TRIN'. The formula bar shows the query definition: `= Table.TransformColumnTypes(Kilde,{{"Kunder", type text}},`

	Kunder	Produkt	Måned	Værdi
1	Netto	Kylling	Januar	122
2	Netto	Kylling	Marts	46
3	Netto	Mel	Februar	55
4	Netto	Rugbrød	Februar	110
5	Netto	Rugbrød	Marts	58
6	Irma	Æbler	Januar	174
7	Irma	Æbler	Marts	115
8	Irma	Kylling	Marts	56
9	Irma	Mel	Februar	131
10	Irma	Mel	Marts	46
11	Irma	Fisk	Januar	129
12	Irma	Fisk	Februar	350
13	Irma	Fisk	Marts	206
14	Irma	Rugbrød	Januar	50
15	Irma	Rugbrød	Februar	76
16	Irma	Rugbrød	Marts	70
17	Bilka	Mælk	Januar	328
18	Bilka	Mælk	Februar	38
19	Bilka	Kylling	Februar	113
20	Bilka	Kylling	Marts	141
21	Bilka	Mel	Marts	50
22	Bilka	Fisk	Januar	83
23	Bilka	Fisk	Februar	54

4 KOLONNER, 85 RÆKKER

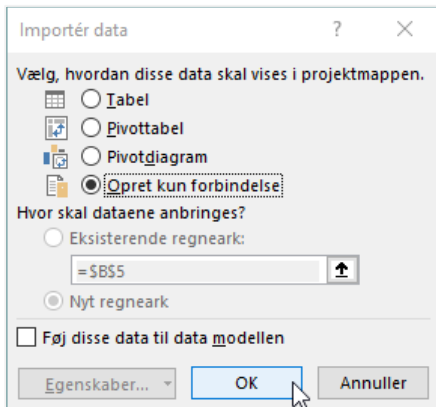
EKSEMPEL DOWNLOADET KL. 11:03

I vores eksempel behøver vi ikke at indlæse dataene i regnearket, i stedet kan vi bare oprette en forbindelse. Fra Power Query skal du klikke på Hjem → Luk og indlæs til ...

The screenshot shows a close-up of the 'Luk og indlæs' dropdown menu in the Power Query ribbon. The menu is open, showing options: 'Luk og indlæs', 'Luk og indlæs til...', and 'Luk og indlæs til...'. The 'Luk og indlæs til...' option is highlighted by the mouse cursor.

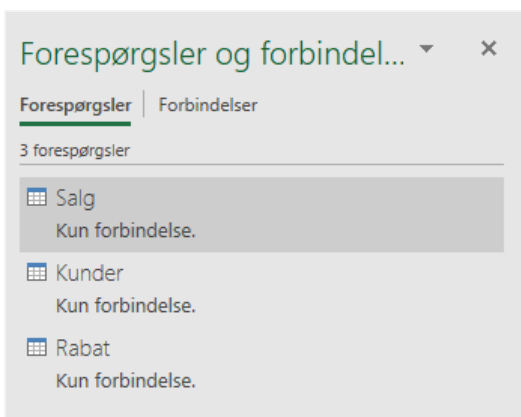
Vælg **Opret kun forbindelse** i dialogboksen **Importer data**, og klik derefter på **OK**.

## Kapitel 15 Power Query - Opslagsværdier ved hjælp af fletning



Gentag denne handling med tabellerne Kunder og Rabatter.

I Excel skal du åbne panelet Forespørgsler og forbindelser (Klik på **Data** → **Vis forespørgsler og forbindelser**, hvis det ikke er synligt). De tre forespørgsler vises.



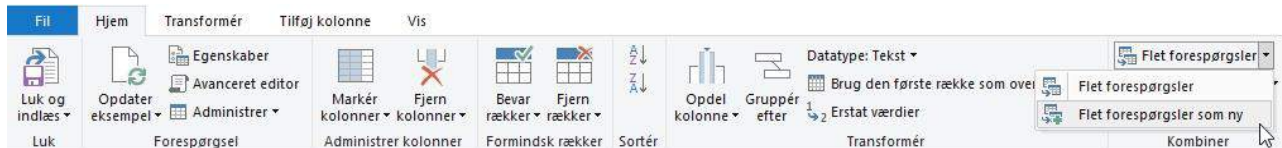
Nu er vi klar til at begynde at bruge flettefunktionen.

### Opslag med en nøjagtig match

For at illustrere et nøjagtigt match, opretter vi en rapport over det samlede salg efter by.

Lad os vende tilbage til Power Query editoren ved at dobbeltklikke på forespørgslen Salg i panelet Projektmappen forespørgsler.

Vælg **Hjem** → **Flet forespørgsler** i Power Query-editoren.



Der er to muligheder her: Flet forespørgsler og Flet forespørgsler som nye. Forskellen mellem dem er, om fletningen skal oprette en ny forespørgsel eller skal tilføjes som et transformationstrin i en eksisterende forespørgsel. For nemheds skyld bruger vi en ny forespørgsel. Vælg muligheden **Flet forespørgsler som ny**.

## Kapitel 15 Power Query - Opslagsværdier ved hjælp af fletning

Dialogboksen Flet åbnes.

Flet

Vælg tabeller og matchende kolonner for at oprette en flettet tabel.

Salg

Kunder	Produkt	Måned	Værdi
Netto	Kylling	Januar	122
Netto	Kylling	Marts	46
Netto	Mel	Februar	55
Netto	Rugbrød	Februar	110
Netto	Rugbrød	Marts	58

Kunder

Kunder	Telefon	Adresse	By	Postnr
Netto	613780627	Alfavej 5	Valby	2500
Irma	448107275	Betagde 6	Valby	2500
Bilka	490391249	Gamma strædet 47	Roskilde	4000
Fakta	714880427	Epsilon plads 12	Valby	2500
Rema 1000	412996856	Myvej 7	Valby	2500

Jointype

Venstre ydre (alle fra den første, matchende fra den a...)

Brug fuzzymatching til at udføre fletningen

► Fuzzy match-indstillinger

✓ Markeringen matcher 85 af 85 rækker fra den første tabel.

OK Annuller

Der sker meget her:

1. Vælg den første forespørgsel, der skal bruges – i vores eksempel er det forespørgslen Salg.
2. Vælg den anden forespørgsel, der skal bruges - i vores eksempel er det forespørgslen Kunder.
3. Vælg kolonnen fra de to tabeller, der skal matches.
4. Jointypen tilbyder seks forskellige typer af joins - Vælg Venstre ydre, hvis du arbejder med eksemplet.
5. Klik på OK.

Der er 6 sammenføjningsstyper. Se afsnittet nedenfor for at se, hvad de forskellige typer gør.

Der oprettes en ny forespørgsel. Den første forespørgsel, der er valgt i vinduet Flet, vises med en ekstra kolonne, der indeholder tabellen for den anden forespørgsel.

	A <sup>B</sup> C Kunder	A <sup>B</sup> C Produkt	A <sup>B</sup> C Måned	1 <sup>2</sup> 3 Værdi	Kunder.1
1	Netto	Kylling	Januar	122	Table
2	Netto	Kylling	Marts	46	Table
3	Netto	Mel	Februar	55	Table
4	Netto	Rugbrød	Februar	110	Table
5	Netto	Rugbrød	Marts	58	Table
6	Irma	Æbler	Januar	174	Table
7	Irma	Æbler	Marts	115	Table
8	Irma	Kylling	Marts	56	Table

Klik på ikonet Udvid tabel i overskriften i kolonnen Kunder. Vi har kun brug af kolonnen Kunde og By, så vælg disse, og klik derefter på **OK**.



## Kapitel 15 Power Query - Opslagsværdier ved hjælp af fletning

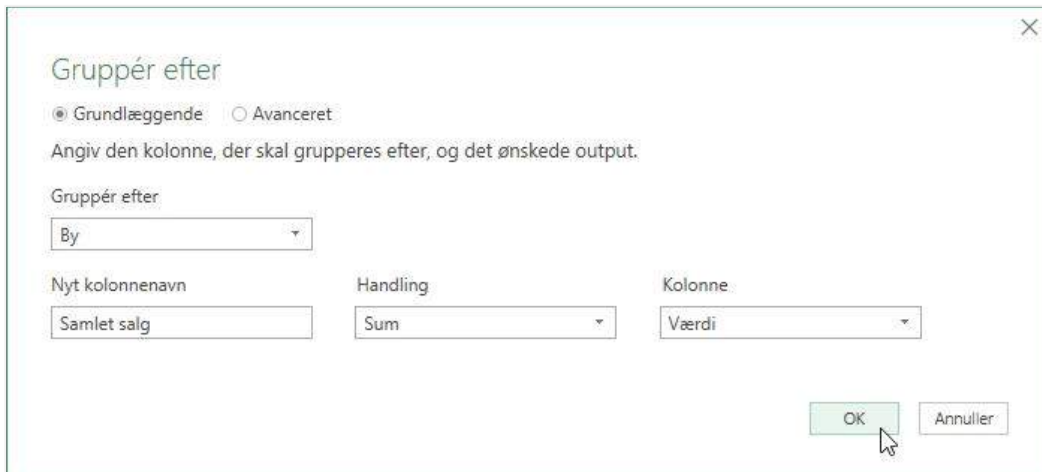


Power Query vil flette forespørgslerne ved at slå op fra den første tabel til den anden tabel.

Den valgte sammenføjning inkluderede alle elementer fra salgstabellen og de matchende poster fra kundetabellen. Eventuelle elementer uden et match vil vise null, som vist i skærbilledet nedenfor. Kunden ICA findes i forespørgslen Salg, men ikke i forespørgslen Kunder, derfor vises nulværdien.

	A <sup>B</sup> <sub>C</sub> Kunder	A <sup>B</sup> <sub>C</sub> Produkt	A <sup>B</sup> <sub>C</sub> Måned	1 <sup>2</sup> <sub>3</sub> Værdi	A <sup>B</sup> <sub>C</sub> Kunder.1.Kunder	A <sup>B</sup> <sub>C</sub> Kunder.1.By	
18	Bilka	Mælk	Februar		38	Bilka	Roskilde
19	Bilka	Kylling	Februar		113	Bilka	Roskilde
20	Bilka	Kylling	Marts		141	Bilka	Roskilde
21	Bilka	Mel	Marts		50	Bilka	Roskilde
22	Bilka	Fisk	Januar		83	Bilka	Roskilde
23	Bilka	Fisk	Februar		54	Bilka	Roskilde
24	Bilka	Fisk	Marts		123	Bilka	Roskilde
25	Bilka	Rugbrød	Januar		75	Bilka	Roskilde
26	Bilka	Rugbrød	Februar		39	Bilka	Roskilde
27	Bilka	Rugbrød	Marts		50	Bilka	Roskilde
28	ICA	Mælk	Januar		49	null	null
29	ICA	Mælk	Februar		53	null	null
30	ICA	Mælk	Marts		186	null	null
31	ICA	Kylling	Januar		137	null	null
32	ICA	Kylling	Februar		217	null	null
33	ICA	Fisk	Januar		78	null	null
34	ICA	Fisk	Marts		127	null	null
35	ICA	Rugbrød	Januar		156	null	null
36	ICA	Rugbrød	Februar		105	null	null
37	Rema 1000	Mælk	Januar		54	Rema 1000	Valby
38	Rema 1000	Mælk	Februar		133	Rema 1000	Valby

For at færdiggøre vores eksempel vil vi bruge denne kolonne til at oprette en opsummeringsrapport. Marker kolonnen Kunder.1.By, og klik derefter på **Transformerer** → **Grupper efter**.



I skærbilledet ovenfor har jeg valgt at opsummere kolonnen værdien i en kolonne kaldet Samlet salg. Når jeg har klikket på OK, oprettes der en opsummerende rapport af salg efter by (se skærbillede nedenfor).

	A <sup>B</sup> C Kunder.1.By	1.2 Samlet salg
1	Valby	2711
2	Roskilde	3429
3	null	1108
4	København	1576

### Join typer

Før vi ser på endnu et eksempel, så lad os bruge et par minutter at se på de seks forskellige join typer.

Joins gør det muligt for os at sammenligne lister og derefter returnere værdier i overensstemmelse hermed. Heldigvis er beskrivelserne, der er leveret for hver join type, en god oversigt over, hvad den foretager sig.

### Yderforbindelser

Yderforbindelser returnerer alle rækker fra en eller begge lister. Vi kan vælge enten Venstre, Højre eller Fuld, afhængigt af hvilken liste der skal returnere alle dens rækker.

- Venstre yder - alle poster fra den første liste returneres sammen med de matchende elementer fra den anden liste
- Højre ydre - alle elementer fra den anden liste returneres sammen med de matchende elementer fra den første liste
- Fuld ydre - alle elementer fra begge lister

### Indre del

En indre sammenknytning returnerer kun de værdier, der findes på begge lister. Hvis den første liste eller den anden liste har poster, der ikke er på den anden liste, er disse udelukket fra slutresultatet.

### Anti sammenføjninger

Anti-sammenføjninger returnerer de poster, der ikke matcher nogen værdier på den anden liste

- Venstre Anti - returnerer alle poster på den første liste, der ikke svarer til den anden liste
- Højre Anti - alle elementer på den anden liste, der ikke svarer til den første liste

Dette er alt sammen effektive ting og bevis på, at Power Query kan opnå mere end LOPSLAG.

## Opslag med en omtrentlig match

Nu er det tid til at se på et omtrentlig match.

I dette eksempel beregner vi værdien af en rabat for en kunde baseret på salgsværdien. Kunder med større salg end:

- kr. 500 modtager en rabat på 2%
- Kr. 750 modtager en rabat på 5%
- kr. 1.000 modtager en rabat på 10%

Disse tærskler vises i tabellen i regnearket.

Rabatgrænse	Rabat værdi
500	2%
750	5%
1.000	10%

Først skal vi først beregne det samlede salg for kunden, og derefter foretage en mere kompliceret fletning for at beregne rabatten baseret på det samlede salg.

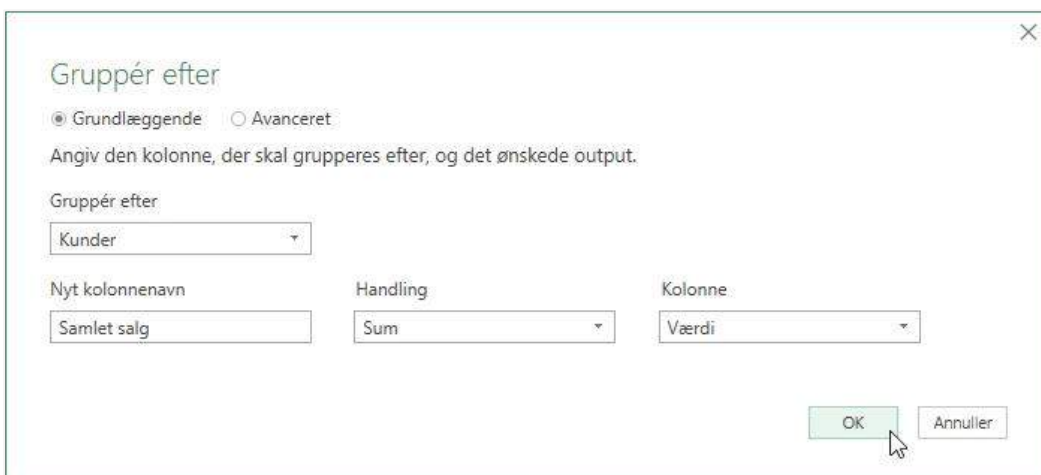
Tilføj en ny version af Salgstabellen til Power Query ved at følge de samme trin ovenfor.

I Power Query-editoren skal du klikke på **Transformerer** → **Grupper efter**.



Dialogboksen Gruppere efter åbnes, foretag følgende valg:

- Gruppe efter: Kunder
- Nyt kolonnenavn: Samlet salg
- Betjening: Sum
- Kolonne: Værdi



Klik på **OK**. Eksempelvinduet skal vise følgende tabel med det samlede salg af Kunder.

## Kapitel 15 Power Query - Opslagsværdier ved hjælp af fletning

	A B C Kunder	1.2 Samlet salg
1	Netto	391
2	Irma	1403
3	Bilka	1094
4	Mega Mart	1108
5	Rema 1000	917
6	Føtex	1276
7	Lidl	629
8	Spar	1059
9	ICA	947

I det sidste eksempel oprettede vi Fletningen som en ny forespørgsel; denne gang kan vi tilføje fletningen som endnu et trin. Klik på **Hjem** → **Flet forespørgsel**.

I dialogboksen Flet vil vi bruge en fuld ydre sammenføjning mellem kolonnen Samlet salg i Salgstabellen og Rabatbåndet til Rabat tabellen.

Flet

Vælg en tabel og matchende kolonner for at oprette en flettet tabel.

Salg (2)

Kunder	Samlet salg
Netto	391
Irma	1403
Bilka	1094
ICA	1108
Rema 1000	917

Rabat

Rabatgrænse	Rabat værdi
500	0,02
750	0,05
1000	0,1

Jointype

Komplet ydre (alle rækker fra begge)

Brug fuzzymatching til at udføre fletningen

▸ Fuzzy match-indstillinger

✓ Markeringen svarer til 0 af 9 rækker fra den første tabel og 0 af 3 rækker f...

OK Annuller

Klik på **OK** for at lukke og vende tilbage til eksempelvinduet.

Udvid kolonnen Rabatter, inkluder begge kolonner, og klik derefter på **OK**.

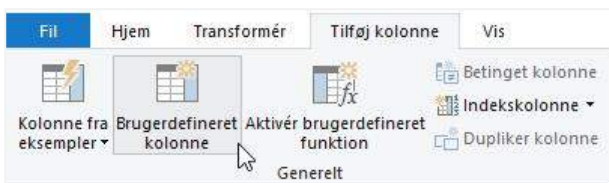
## Kapitel 15 Power Query - Opslagsværdier ved hjælp af fletning



Eksempelvinduet ser nu sådan ud:

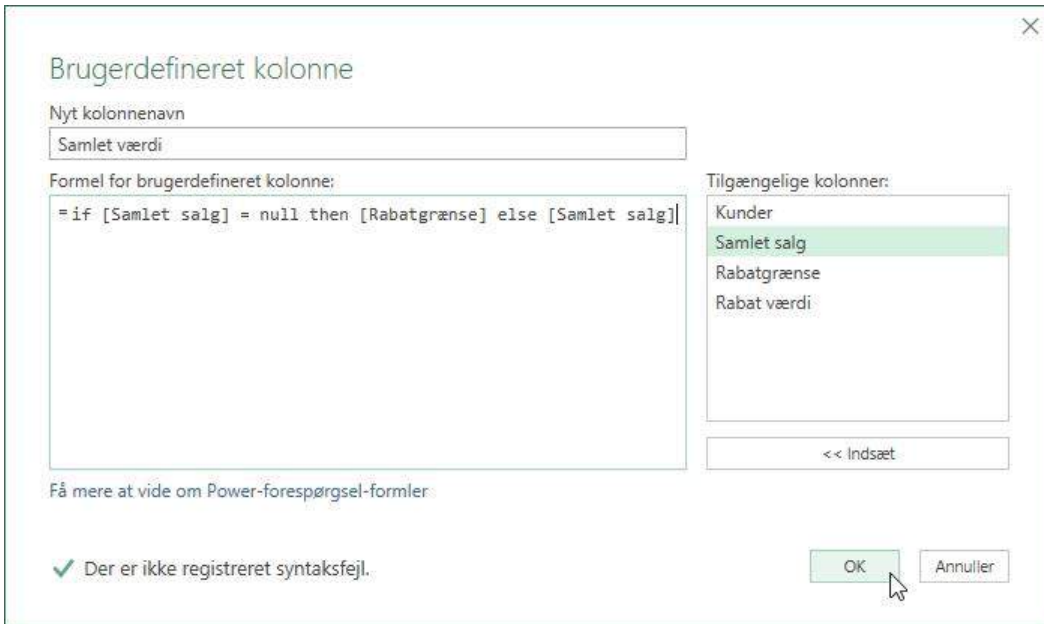
	Kunder	1.2 Samlet salg	1.2.3 Rabatgrænse	1.2 Rabat værdi
1	Netto	391	null	null
2	Irma	1403	null	null
3	Bilka	1094	null	null
4	Mega Mart	1108	null	null
5	Rema 1000	917	null	null
6	Føtex	1276	null	null
7	Lidl	629	null	null
8	Spar	1059	null	null
9	ICA	947	null	null
10	null	null	500	0,02
11	null	null	750	0,05
12	null	null	1000	0,1

Vi skal nu skrive en if-erklæring. Vi kunne bruge funktionen Betinget kolonne, men jeg synes, det er lettere at skrive det som en formel. Klik på **Tilføj kolonne** → **Brugerdefineret kolonne**.



Dialogboksen Brugerdefineret kolonne åbnes. Giv kolonnen et nyt navn, og indtast følgende tekst i formelfeltet:

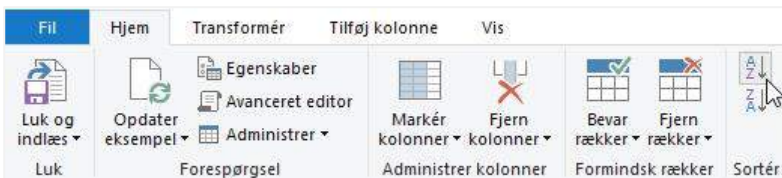
```
if [Samlet salg] = null then [Rabatgrænse] else [Samlet salg]
```



Jeg vil gennemgå if erklæringen mere detaljeret i et senere kapitel. I øjeblikket skal du bare tænke på det som standard Excel HVIS funktionen med følgende ændringer:

- Fjern parenteserne
- udskift det første semikolon med ordet **then**
- udskift det sidste semikolon med ordet **else**

Sorter den nye kolonne ved at vælge kolonneoverskriften, og klik derefter på **Hjem** → **A-Z**.



Vælg derefter kolonnen Rabatværdi og klik på **Transformer** → **Udfyldning** → **Nedad**.



## Kapitel 15 Power Query - Opslagsværdier ved hjælp af fletning

Eksempelvinduet skal se ud som følger:

	A <sup>B</sup> C Kunder	1.2 Samlet salg	1 <sup>2</sup> 3 Rabatgrænse	1.2 Rabat værdi	ABC 123 Samlet værdi
1	Netto	391	null	null	391
2	null	null	500	0,02	500
3	Lidl	629	null	0,02	629
4	null	null	750	0,05	750
5	Rema 1000	917	null	0,05	917
6	ICA	947	null	0,05	947
7	null	null	1000	0,1	1000
8	Spar	1059	null	0,1	1059
9	Bilka	1094	null	0,1	1094
10	Mega Mart	1108	null	0,1	1108
11	Føtex	1276	null	0,1	1276
12	Irma	1403	null	0,1	1403

Nu er der kun et par trin tilbage for at rydde op i tabellen:

- Filtrer null værdierne fra kolonnen Kunder
- Fjern alle kolonner undtagen **Kunder**, **Samlet salg** og **Rabat værdi**.

Lad os nu afslutte med at beregne rabatværdien. Vælg kolonnerne Samlet salg og Rabat værdi, klik på **Tilføj kolonne** → **Standard** → **Multipliser**.

The screenshot shows the Power Query ribbon with the 'Tilføj kolonne' (Add Column) tab selected. A dropdown menu is open, showing options like 'Læg sammen', 'Multipliser', 'Træk fra', 'Divider', etc. The 'Multipliser' option is highlighted. In the background, a table is visible with columns 'Kunder', 'Samlet salg', and 'Rabat værdi'. The formula bar shows a partial formula: '= Table.RemoveColumns(#"Filtrerede rækker", {"Rabatgrænse", ...})'.

Forespørgslen viser nu det endelige resultat.

	A <sup>B</sup> C Kunder	1.2 Samlet salg	1.2 Rabat værdi	1.2 Multiplikation
1	Netto	391	null	null
2	Lidl	629	0,02	12,58
3	Rema 1000	917	0,05	45,85
4	ICA	947	0,05	47,35
5	Spar	1059	0,1	105,9
6	Bilka	1094	0,1	109,4
7	Mega Mart	1108	0,1	110,8
8	Føtex	1276	0,1	127,6
9	Irma	1403	0,1	140,3

## Flere match

Hvad sker der, hvis der er flere elementer, der kunne matches? I dette scenarie returnerer et nøjagtigt match i LOPSLAG i Excel det første element, den finder på listen. Flet håndterer det anderledes.

Fletning returnerer hver forekomst af et element der matcher. Lad os for eksempel antage, at vi har to tabeller, en med produktinformation og en anden med lageroplysninger om disse produkter.

Produkter				Lager			
ProduktID	Størrelse	Farve	Placering	Størrelse	Farve	Placering	Enheder
ProduktID 1	S	Sort	Lokale 1	M	Sort	Lokale 2	5
ProduktID 2	M	Brun	Lokale 2	Størrelse	Sort	Lokale 1	2
ProduktID 3	M	Sort	Lokale 3	XL	Brun	Lokale 3	7
ProduktID 4	L	Blå	Lokale 2	L	Sort	Lokale 1	8
ProduktID 5	XL	Brun	Lokale 2	S	Blå	Lokale 2	2
ProduktID 6	S	Blå	Lokale 3	M	Blå	Lokale 3	4
ProduktID 7	L	Sort	Lokale 2	S	Sort	Lokale 2	6
ProduktID 8	XL	Blå	Lokale 2	L	Brun	Lokale 2	5

Hvis vi kun skulle flette kolonnen Størrelse, ville elementerne med M størrelse duplikeres, da der er to M'er i tabellen, der bliver slået op i.

	A <sup>B</sup> <sub>C</sub> ProduktID	A <sup>B</sup> <sub>C</sub> Størrelse	A <sup>B</sup> <sub>C</sub> Farve	A <sup>B</sup> <sub>C</sub> Placering	1 <sup>2</sup> <sub>3</sub> Lager.Enheder
1	ProduktID 1	S	Sort	Lokale 1	2
2	ProduktID 1	S	Sort	Lokale 1	2
3	ProduktID 1	S	Sort	Lokale 1	6
4	ProduktID 6	S	Blå	Lokale 3	2
5	ProduktID 6	S	Blå	Lokale 3	2
6	ProduktID 6	S	Blå	Lokale 3	6
7	ProduktID 2	M	Brun	Lokale 2	5
8	ProduktID 2	M	Brun	Lokale 2	4
9	ProduktID 3	M	Sort	Lokale 3	5
10	ProduktID 3	M	Sort	Lokale 3	4
11	ProduktID 5	XL	Brun	Lokale 2	7
12	ProduktID 8	XL	Blå	Lokale 2	7
13	ProduktID 4	L	Blå	Lokale 2	8
14	ProduktID 4	L	Blå	Lokale 2	5
15	ProduktID 7	L	Sort	Lokale 2	8
16	ProduktID 7	L	Sort	Lokale 2	5

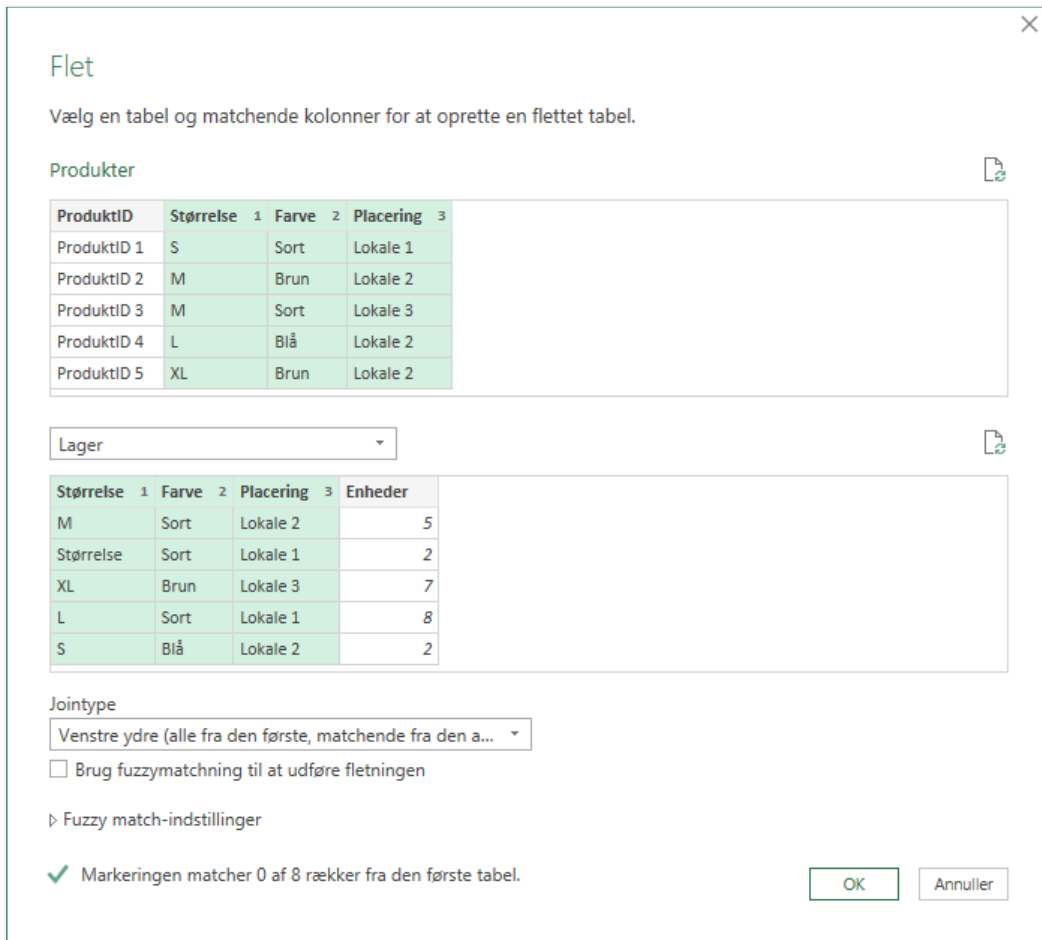
Hvis du af en eller anden grund kun ønskede at matche et emne, skal du fjerne duplikaterne fra en af tabellerne, før du udfører fletningen.

## Opslag med flere kriterier

Den gode nyhed er, at Power Query ikke begrænser dig til en liste. Lad os sige, at du vil matche tre kolonner? Det er ikke noget problem.

Den rækkefølge, du vælger kolonnerne, bestemmer, hvilke kolonner der matches. Se på skærmbilledet nedenfor. De kolonner, der blev valgt i den første tabel, var Farve, Størrelse og derefter Placering (i den rækkefølge). Tallene i kolonneoverskriften identificerer rækkefølgen, hvormed elementerne blev valgt. De kolonner, der blev valgt i den anden tabel, var i samme rækkefølge, Farve, Størrelse og Placering.





## Fuzzy match opslag

Når du kigger på dialogboksen Flet i vores eksempler, bemærkede du så muligheden for at "Fuzzy match-indstilling"? Dette er en forholdsvis ny mulighed, den matcher værdier, der er ens. For eksempel vil det matche "Power Query" med "power-query". Du kan ikke gøre det med LOPSLAG! Vi kan endda ændre tærsklerne for, hvordan ens værdier skal være, før de matcher.

## Kapitel 16 Power Query – Skift placering for kildedata

Du har opbygget nogle forespørgsler i et testmiljø, og nu er det tid til at begynde at bruge dem rigtigt. Hvordan kan du ændre forespørgsler til at pege på den rigtige kilde i stedet for testen? I dette kapitel besvarer vi dette spørgsmål.

Ligesom som de fleste ting i Excel er der flere måder at gøre det på:

- Ret placering i forespørgslens M kode.
- Brug datakildeindstillingerne.
- Brug parametre

Lad os komme i gang uden yderligere snak.

### Ret kildeplacering i forespørgslen

Det første anvendte trin i de fleste forespørgsler er identificeringen af kildedataene. Teknisk behøver det ikke være sådan, og det behøver heller ikke at blive kaldt kilde. I de fleste tilfælde vil det dog være sådan. Det at være den første gør det nemt at finde M koden; som det altid er i starten!

Skærbillede herunder viser det første trin i forespørgslen med M koden vist i formellinjen. Hvis formellinjen ikke er vist, kan du klikke på **Vis** → **Formellinje** i Power Query menuen.

The screenshot shows the Power Query interface. At the top, the M code editor displays the formula: `= Excel.Workbook(File.Contents("C:\Power Query\Kapitel 4\Power Query - Rediger forespørgsler.xlsx"), null, true)`. Below the editor is a table with columns: Name, Data, Item, Kind, and Hidden. The table lists several queries, with 'Query Data' selected. To the right, the 'Forespørgselsindstillinger' (Query Properties) pane is open, showing the 'EGENSKABER' (Properties) section with the name 'Query Data' and the 'ANVENDTE TRIN' (Applied Steps) section with 'Kilde' (Source) selected.

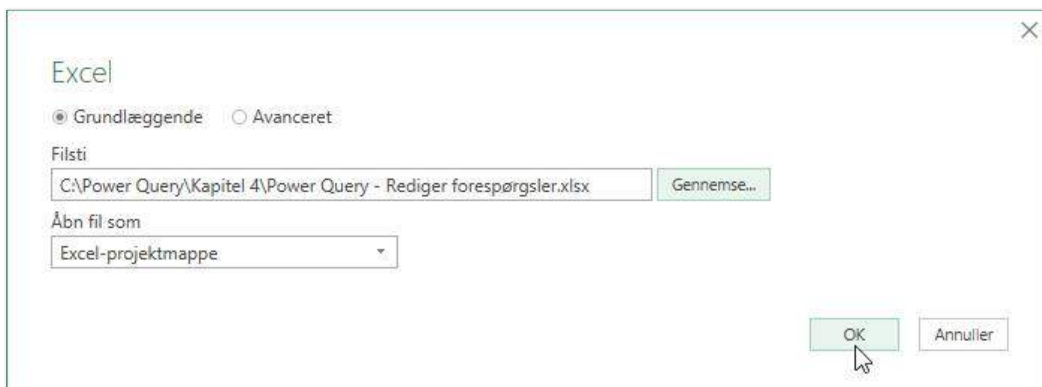
	Name	Data	Item	Kind	Hidden
1	Query Data	Table	Query Data	Sheet	FALSE
2	Forespørgsel Output	Table	Forespørgsel Output	Sheet	FALSE
3	tblSalesData	Table	tblSalesData	Table	FALSE
4	Bedste_sælger	Table	Bedste_sælger	Table	FALSE
5	SalgEfterDag	Table	SalgEfterDag	Table	FALSE
6	Bedste_kunder	Table	Bedste_kunder	Table	FALSE

Vi kan redigere koden direkte i formellinjen, vi skal bare angive den nye filplacering som en tekststreng.

Vi har også muligheden for at redigere koden i den avancerede editor, klik på **Hjem** → **Avanceret editor** eller **Vis** → **Avanceret editor** (se skærbilledet nedenfor).



Og en tredje mulighed er at klikke på tandhjulsikonet ved siden af kildetrinnet i vinduet Anvendte trin. I dette eksempel bruges en Excel projektmappe som kilde, så klik på **Gennemse** for at navigere til den nye filplacering, og klik derefter på **OK**.



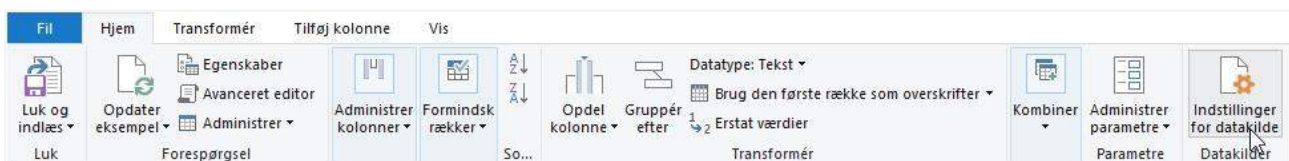
Afhængig af kildetyper ændres dialogboksen, men principperne er de samme.

Vi har allerede set teknikken til at ændre kilden i denne Power Query bog, da vi importerede fra nettet.

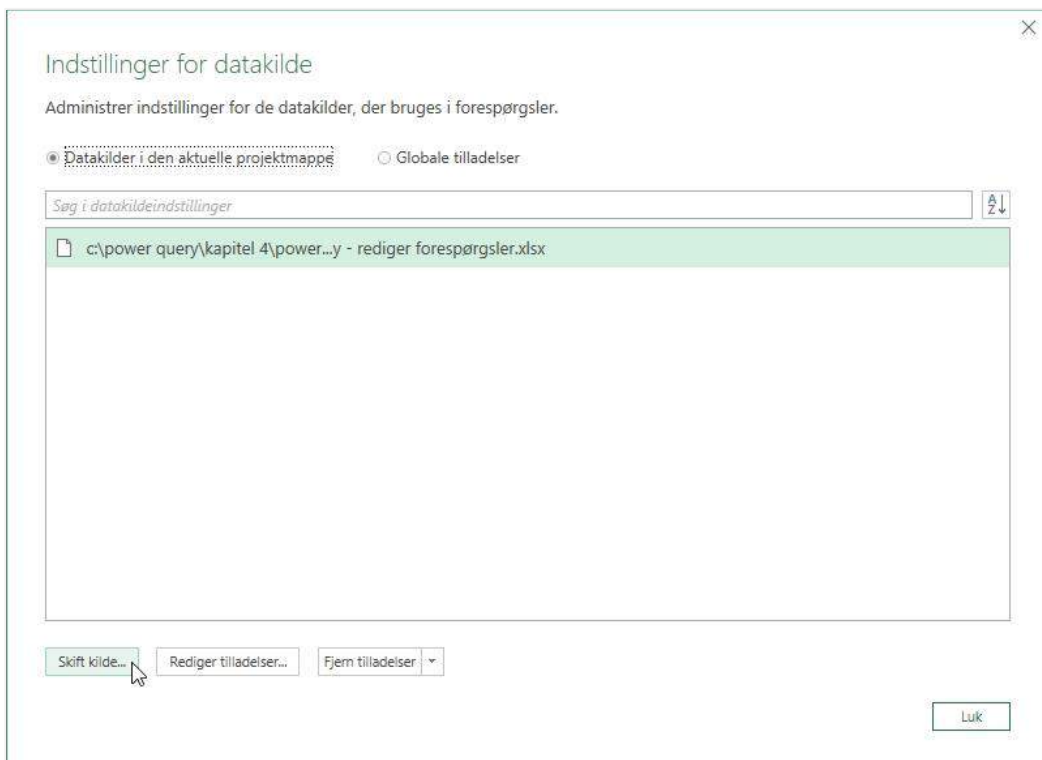
## Brug indstillingerne for datakilde

Hvor der er flere forespørgsler, der alle kommer fra den samme kilde, er der en bedre måde at pege på kildedataene; Indstillinger for datakilde. Faktisk kan dette generelt være en bedre måde.

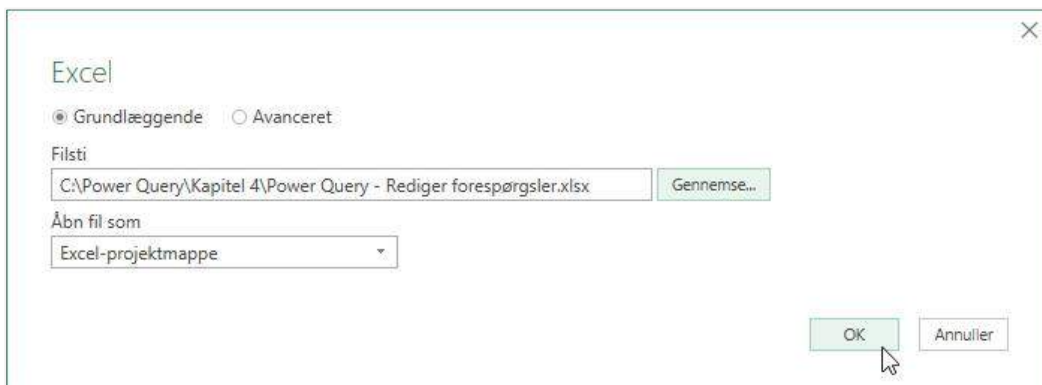
I Power Query editoren klikker du på **Hjem** → **Indstillinger for datakilde**.



I dialogboksen Indstillinger for datakilde skal du vælge den kilde, du vil ændre, og klik derefter på knappen **Skift kilde**.



Dette åbner den samme dialogboks som hvis du havde klikket på tandhjulskonet (som vi så tidligere).



Der er dog en afgørende forskel; snarere end at ændre kilden til en enkelt forespørgsel, ændrer den kilden for alle forespørgsler, der bruger den som kilde.

Indstillingerne for datakilde er tilgængelige andre steder i Excel og Power Query, men jeg synes denne måde, er den nemmeste at huske.

## Brug af parametre

Hvis du ved, at kildedataplaceringen vil ændre sig, er det en god ide at oprette en parameter. En celle i regnearket, der kan indeholde en tekststreng med datakildens placering. Denne tekststreng kan være mappestien, filnavnet, tabelnavnet eller en SharePoint-web-adresse osv. For at opdatere forespørgslen er det bare at ændre tekststrengen.

Vi har set mange gange, at Power Query hardkoder tekstværdier i M koden, så brug af parametre er en god fleksibel tilgang. Da vi allerede har dækket Power Query parametre i et andet kapitel, gentager jeg ikke processen her, du kan eventuelt læse kapitlet om parametre igen!

## Kapitel 17 Power Query – formler

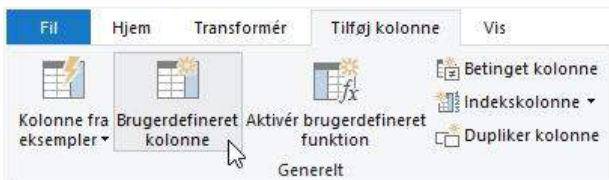
Formler er livsnerven i Excel; de er vigtige for at opnå selv basale opgaver. Power Query er derimod designet, så de fleste transformationer fås gennem den intuitive brugergrænseflade. Men Power Query har et formelsprog med over 700 funktioner. Det er så der, vi kan tackle nogle af de mest komplicerede situationer, vi støder på.

Jeg er inkarneret Excel bruger. Da jeg begyndte at bruge Power Query, benyttede jeg derfor masser af formler, fordi det var hvad jeg plejede i Excel. Da jeg først forstod brugerfladen, blev det imidlertid klart, at de fleste transformationer kan opnås uden overhovedet at bruge formler. Derfor anbefaler jeg, at du kigger efter transformationer, der er tilgængelige i standardmenuerne, før du går i dybden med specifikke funktioner.

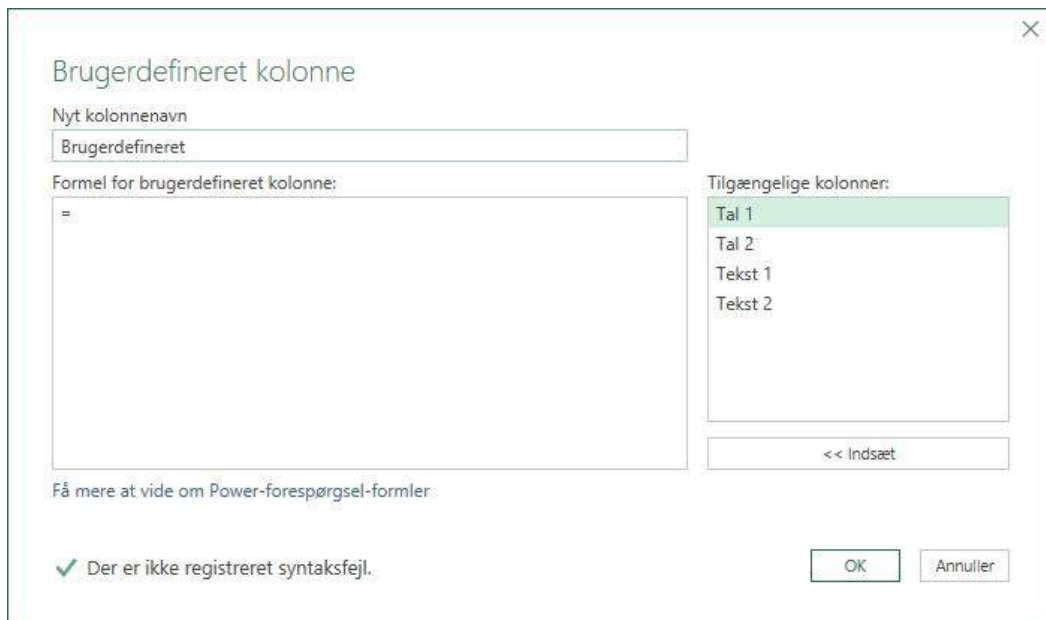
Noget der kan være forvirrende er, at funktionerne ikke er de samme som Excel; de skrives ved hjælp af M kode. I starten er det frustrerende, at vi ikke kan overføre vores eksisterende viden til Power Query. Når du har fundet ud af, hvordan formlerne er konstrueret, vil du hurtigt indse, at det er let at finde det, du har brug for. I denne Power Query bog har vi allerede brugt formler, hver gang vi oprettede en brugerdefineret kolonne. Dette kapitel giver en bedre forståelse af, hvordan man bruger formler.

### Opret en formel

Du opretter en brugerdefineret kolonne ved at klikke på **Tilføj kolonne** → **Brugerdefineret kolonne**.



Dialogboksen Brugerdefineret kolonne åbnes.



De vigtigste områder i denne dialogboks er:

### 1. Nyt kolonnenavn

Den indtastede tekst bruges som kolonnenavn i datatabellen.

### 2. Tilgængelige kolonner

Denne liste indeholder alle kolonnerne i forespørgslen. Hvis du vælger en kolonne og klikker på knappen Indsæt (eller dobbeltklikker på navnet), indsættes den i formlen. Dette er rigtigt nyttigt, da det reducerer tastefejl, der ellers vil få vores kode til at fejle.

### 3. Tilpasset kolonneformel

Det er her vi indtaster teksten, der udgør formlen.

### 4. Fejlkontrol

Fejlkontrollen hjælper med at fortælle os, om syntaksen er korrekt. Den vil kontrollere, om der er matchende start og slutparenteser osv. Det vil ikke fortælle dig, om formlen returnerer det rigtige resultat, kun at syntaks er korrekt. Faktisk er det meget vanskeligt at fortolke, hvad fejlene betyder, derfor arbejder jeg ud fra, at grønt flueben = godt, og alt andet = dårligt.

### 5. Lær mere om Power Query-formler

Dette link fører os til Microsofts websteder med Power Query formler. Hvis du sidder fast, er det et fornuftigt sted at besøge.

## Simple formelfunktioner

Simple formelfunktioner er de samme som i Excel:

#### Addition

```
= [Kolonne 1] + [Kolonne 2]
```

#### Subtraktion

```
= [Kolonne 1] - [Kolonne 2]
```

#### Multiplikation

```
= [Kolonne 1] * [Kolonne 2]
```

#### Division

```
= [Kolonne 1] / [Kolonne 2]
```

#### Sammenkædning

```
= [Kolonne 1] & [Kolonne 2]
```

#### Parenteser

Parenteser fungerer på samme måde som i Excel.

```
= ([Kolonne 1] + [Kolonne 2]) / [Kolonne 3]
```

#### Lighedstegn

```
= [Kolonne 1] = [Kolonne 2]
```

Formlen ovenfor giver et Sandt eller Falskt resultat.

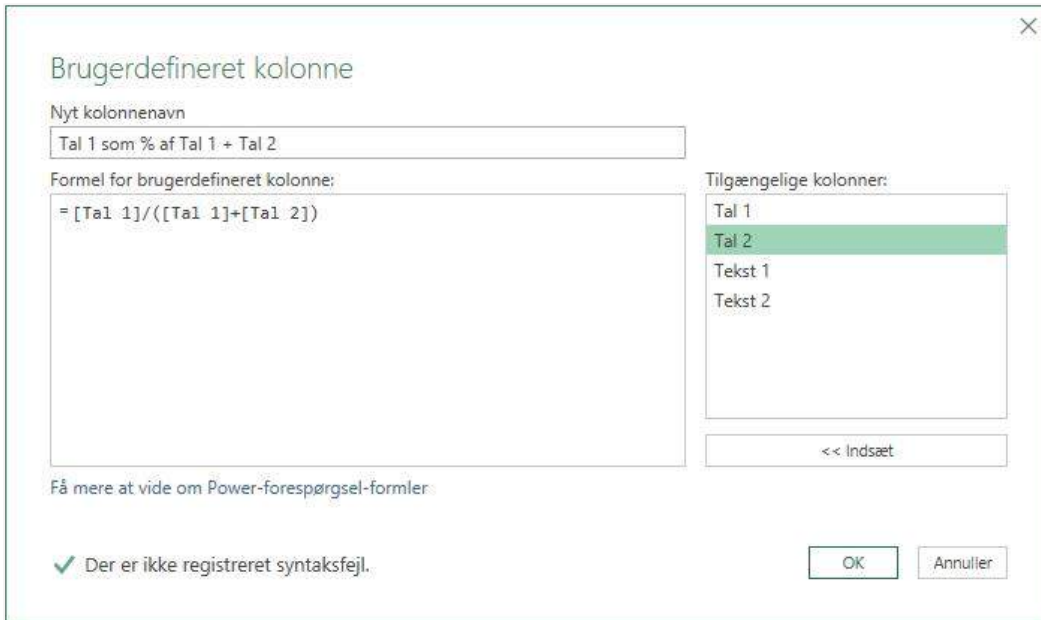
Du behøver ikke at indtaste lighedstegnet (=) i starten; det vil automatisk være i formelboksen.

*Potens*

Potens operatoren (^), som vi bruger i Excel, fungerer ikke i Power Query formler. I stedet skal du bruge formlen Number.Power.

**Formeleksempel nr. 1**

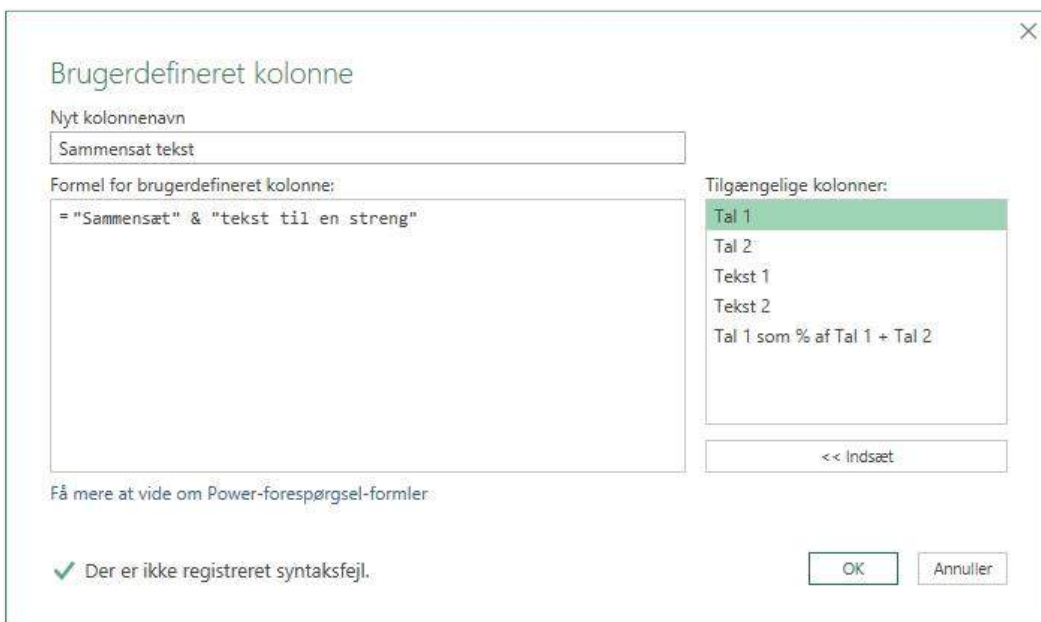
Skærbilledet nedenfor viser nogle af disse grundlæggende formeloperationer i aktion.



Du kan se de kolonnenavne, der bruges; Tal 1 divideres med summen af Tal 1 + Tal 2.

**Formeleksempel nr. 2**

Skærbillede herunder viser, at formler kan konstrueres ud fra statiske værdier. En formel kan også omfatte en blanding af kolonner og statiske værdier.



## Rækkekontekst

Medmindre du konstruerer avanceret formelmagi, har formlerne i Power Query det, der er kendt som "Rækkekontekst". Dette betyder, at formlen anvendes på hver række en efter en. Du kan vise kolonnenavne i formlen, men du anvender ikke denne formel til at udfylde kolonner, men på hver række.

## Datatyper

Power Query formler er pirrelige omkring datatyper. Funktionerne forventer, at visse datatyper fungerer korrekt. F.eks. Returnerer funktionen Date.Year året ud fra en dato. Hvis kolonnen, der bruges i funktionen, ikke er en dato type, beregnes formlen som en fejl. Så hvis datatypen ikke svarer til formelkravene, er vi nødt til at finde en måde at ændre den på. Vi kan enten tilføje et trin, før vi opretter den brugerdefinerede kolonne, eller vi kan bruge en konverteringsformel.

Nogle af de mest almindelige konverteringsformler er:

- Text.From - Konvertere noget til tekst
- Date.ToText - Konverterer en dato til tekst
- Date.From - Konverterer et tal til en dato
- Date.FromText - Konverterer tekst til en dato
- Number.ToText - Konverterer et tal til tekst
- Number.From - Konvertere eventuelle input til et tal
- Number.FromText - Konverterer tekst til et tal
- Logical.From - Konvertere tal til deres en Sand eller Falsk værdi
- Logical.FromText - Tekststrengene "sandt" eller "falskt" konverteres til booleske værdier for sandt eller falsk
- Logical.ToText - Konverterer booleske Sand- eller Falsk værdier til "sand" eller "falsk" tekststreng

## Formeleksempel nr. 3

Følgende tabel indeholder tal og tekst.

	Husnummer	Vejnavn
1	5	Havnevej
2	36	Havnevej
3	65	Havnevej
4	27	Havnevej
5	30	Havnevej

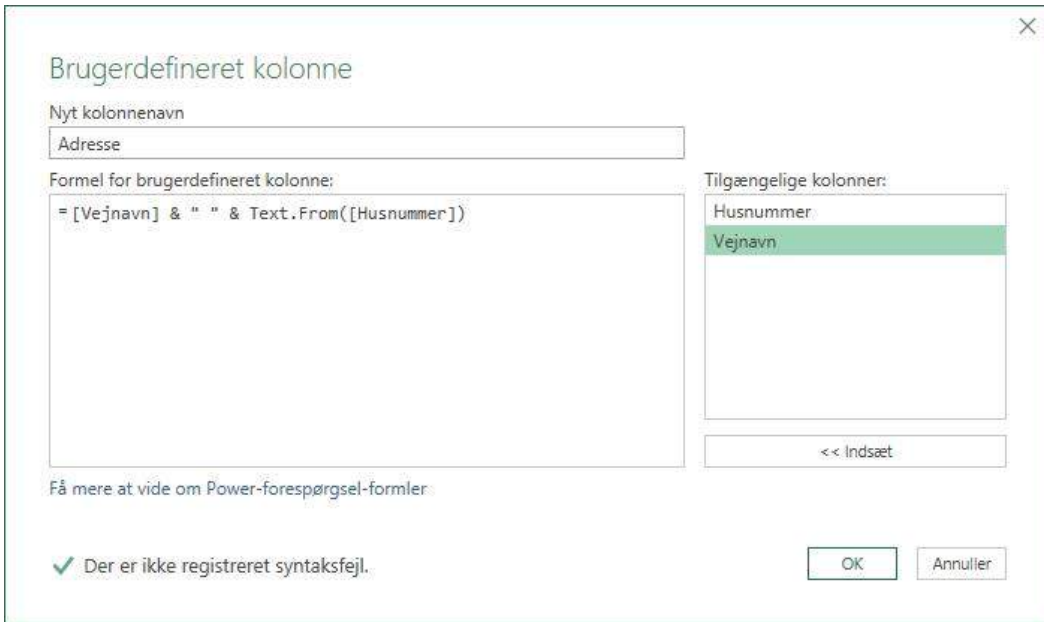
Da Husnummeret er et tal, er vi nødt til at konvertere det til tekst, før vi kan sammenkæde det med vejnavnet i en enkelt adressestreng.

Formelteksten ville være:

```
= [Vejnavn] & " " & Text.From([Husnummer])
```

Teksten fremhævet med fed bruger funktionen Text.From til at konvertere tallet til en tekstdatatype, før det kombineres med kolonnen Vejnavn.





## Find funktioner

Som jeg nævnte tidligere, kan vi ikke bruge Excel formler. Den nemmeste måde at finde de tilgængelige formler er at besøge Microsofts hjemmeside om Power Query-formler.

Funktionerne er grupperet efter, hvad de gør. Hvis vi ønsker en formel, der leverer et tal, ser vi nærmere på talgruppen. De fleste af formlerne i dette afsnit starter med ordet Number. Her er nogle eksempler:

- *Number.Abs* - Returnerer den absolutte værdi af et tal
- *Number.FromText* - Returnerer et tal fra en tekststreng
- *Number.IsOdd* - Returnerer sandt, hvis værdien er ulige, eller falsk, hvis den er lige.

Specifikke tal formler kan også starter med andre ord, såsom:

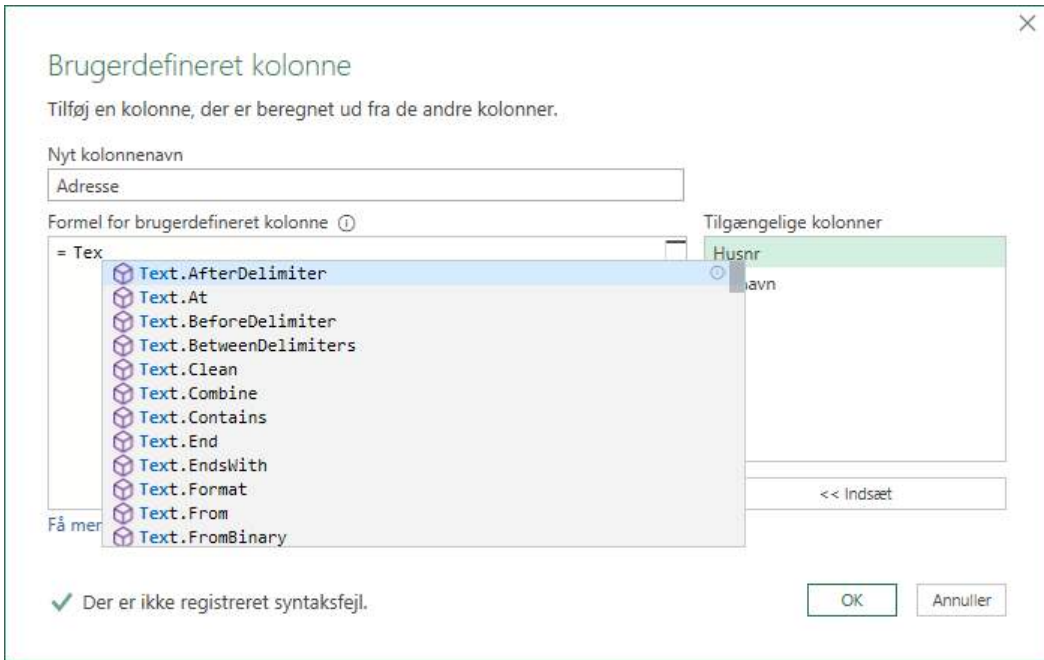
- *Decimal.From* - Returnerer en decimal fra en given værdi
- *Percentage.From* - Returnerer en procentdel fra en given værdi

Generelt starter formler, der leverer en dato, med Date i formelnavnet (f.eks. Date.Year), og formler, der leverer tekst, starter med Text i formelnavnet (f.eks. Text.Start).

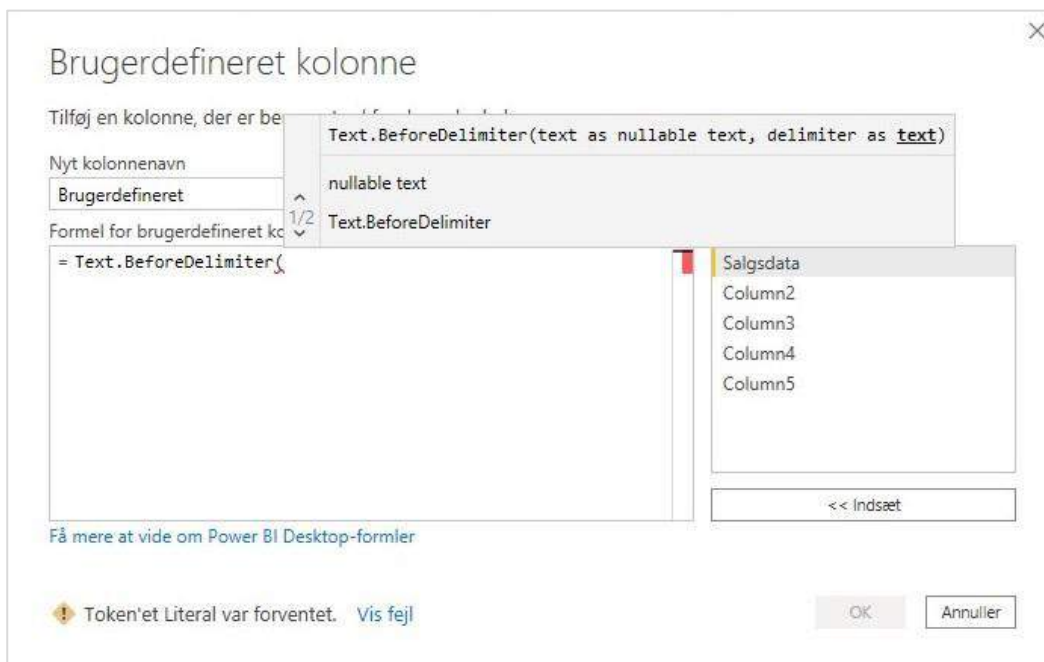
Tænk på Excel funktioner, hvordan lærte du dem? Du har sandsynligvis fået nogen til at vise dig, eller du var nødt til at gætte med en masse "trail and error". På grund af den måde Power Query funktioner er navngivet, er det meget lettere at finde den funktion, vi ønsker.

## IntelliSense

I skrivende stund har Microsoft for nylig frigivet IntelliSense til Power Query. Dette betyder, at vi kan begynde at skrive navnet på en funktion, og der vises en liste af muligheder. Skærmbillede herunder viser listen over funktioner bare ved at skrive "Add" i formelboksen.



IntelliSense hjælper os med at finde funktioner, men gør dem også betydeligt mere anvendelige. Når du har fundet navnet på funktionen, skal du skrive start parentes, og der vises de nødvendige argumenter og datatyper. Det tager et stykke tid for Power Query at få indbygget denne funktionalitet, men nu hvor den er her, kan jeg ikke leve uden.



## Se på eksisterende transformationstrin

De transformationer, vi bruger via brugergrænsefladen, bruger de samme funktioner, som vi har adgang til.

Lad os sige, at vi ønskede at udtrække de første 4 bogstaver fra en tekststreng. Vi kunne bruge **Tilføj kolonne** → **Uddrag** → **Første tegn** fra båndet og derefter indsætte 4 i dialogboksen Indsæt første tegn. Vi kunne derefter tage et kig på M koden i formel for det anvendte trin, den ville vise følgende:

```
= Table.AddColumn("#Tilføjet brugerdefineret", "Første tegn", each Text.Start([Vejnavn], 4), type text)
```

Sektionen med fed er funktionen *Text.Start*. Denne funktion udtrækker de første tegn fra en tekststreng (den ligner Excels VENSTRE funktion).

Dette illustrerer, at når du sidder fast, kan kigge på M koden for en lignende transformation, som kan give dig en ide til hvilken type funktion, du muligvis har brug for.

## Essentielle formeltips

---

Forskellene mellem M kode og Excel funktioner kan få os til at fejle. Nogle af de vigtigste forskelle er:

### M kode er følsom overfor store og små bogstaver

I Excel kan vi skrive funktioner med store og små bogstaver, Excel forstår, hvad vi vil, og konverterer funktioner til store bogstaver. Power Query har dog ikke været så tilgivende; *Text.Start* er navnet på en funktion, mens *text.start* er meningsløs, da det bruger den forkerte bogstaver i hvert ord. I nyere versioner er Power Query begyndt at konvertere funktioner til det rigtige. Da dette er en ny funktion, virker det muligvis ikke endnu.

### M begynder at tælle ved 0

*Text.PositionOf* er en funktion, der finder placeringen af en streng i en anden streng. Hvis vi bruger følgende M kode som eksempel:

```
=Text.PositionOf("Excel er smart", "Excel")
```

Dette modsvarer Excel funktionen *FIND* med argumenterne på den modsatte rækkefølge.

```
=FIND("Excel"; "Excel er smart")
```

Excel funktionen returnerer værdien 1, mens Power Query funktionen returnerer værdien 0. Dette skyldes, at Power Query begynder at tælle fra 0. Det kan virke underligt for de fleste Excel brugere, men det er almindeligt i en verden af programmeringssprog.

## Øve, øve, øve

---

I dette kapitel har vi gennemgået det grundlæggende i Power Query formler. Jeg garanterer, at medmindre du øver dette de næste par uger eller måneder, at du glemmer alt, hvad du har læst. Øvelse er vigtig. M ligner Excel, derfor skal du regelmæssigt henviser til M funktionsbiblioteket for at finde det, du har brug for. Der er ingen skam i det ... faktisk er det hele pointen med det hele.

Det er nok snak, det er tid til at selv prøver nogle af disse emner.

## Kapitel 18 Power Query – IF-udtryk til betinget logik

I et tidligere kapitel kiggede vi kort på if-udtrykket i Power Query, nu vil vi grave lidt dybere og prøve at forstå, hvordan det fungerer.

I Excel er HVIS en nøglefunktion, det er en af de første, vi lærer. I sidste kapitel så vi på funktioner i Power Query, men vi snakkede ikke om en Power Query versionen af HVIS funktionen. Der er vel en ikke? ... jo en slags er der.

For at bruge if logik i Power Query, har vi brug for en mere programmeringsbaseret metodologi snarere end en funktionsbaseret metodik, som vi finder i Excel. Hvis du nogensinde har programmeret VBA eller andre sprog, vil du være bekendt med denne tilgang. Hvis du kommer fra en ren Excel verden, kan dette dog være nyt for dig.

Der er to muligheder:

1. Brug af betinget kolonnefunktion til grundlæggende if logik.
2. Skriv M kode til mere avanceret if logik.

Vi afdækker begge i dette kapitel.

### Eksemplet

Filen indeholder kun en tabel. Marker en hvilken som helst celle i tabellen og klik på **Data** → Fra tabel/område for at indlæse dataene i Power Query.

I dette eksempel beregner vi værdien ud fra følgende:

- **Scenarie 1:** Om søndagen er der en bonus på yderligere 10% (kræver grundlæggende logik)
- **Scenarie 2:** Om søndagen er der en bonus på yderligere 10% og for to produkter (Frugt og Kød), er der en 5% rabat denne dag (kræver avanceret logik).

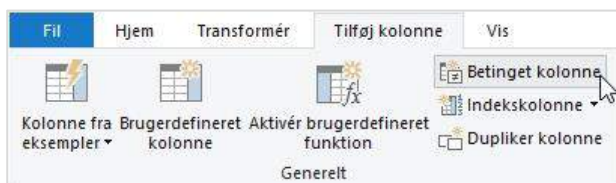
### IF-logik ved hjælp af en betinget kolonne

Lad os starte med scenarie 1 og bruge funktionen Betinget kolonne.

#### Scenarie 1 - Søndagsbonus

Vælg kolonnen Dato, og tilføj derefter en kolonne, der indeholder ugedagen ved at klikke på **Tilføj kolonne** → **Dato** → **Dag** → **Navn på dag**.

Klik derefter på **Tilføj kolonne** → **Betinget kolonne**.



Indtast følgende indstillinger i dialogboksen Tilføj betinget kolonne:

Nyt kolonnenavn: **Søndagsbonus**.

Hvis Navn på dag er lig med søndag, så er 1.1 ellers 1.



Klik på **OK** for at acceptere formelen.

I scenariet ovenfor har vi kun brugt et par af de tilgængelige indstillinger i dialogboksen Tilføj Betingelseskolonne.

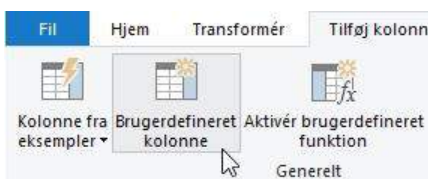
Andre operatører der kan bruges er:

- Er lig med
- Er forskellig fra
- Begynder med
- Begynder ikke med
- Slutter med
- Slutter ikke med
- Indeholder
- Indeholder ikke

Vi kan klikke på knappen 123ABC for at indsætte kolonneværdier eller parametre i stedet for hardkodede værdier.

Vi kan også oprette lige så mange, if udtryk, som vi ønsker, ved at klikke på knappen **Tilføj regel**. If udtrykkene udføres efter sin tur, så hvis det første if udtryk ikke udløses, vil den gå videre til den næste, derefter den tredje osv. Men hvis det første if udtryk er sandt, springes den resterende logik imidlertid over.

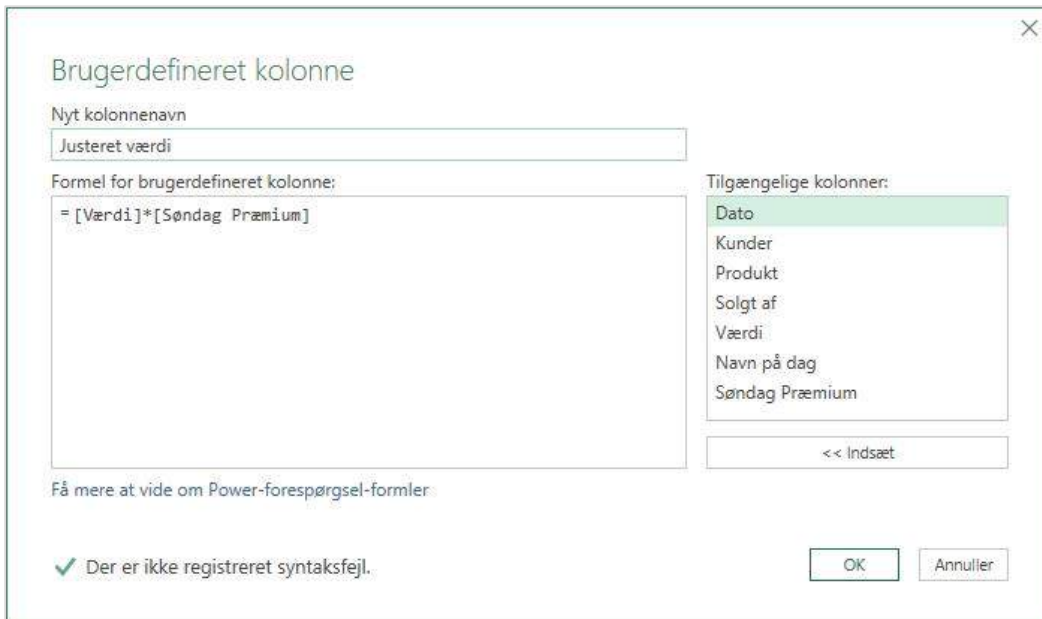
Lad os nu tilføje en formel. Klik på **Tilføj kolonne** → **Brugerdefineret kolonne** for at få vist dialogboksen Tilpasset kolonne.



Indtast følgende formel:

```
= [Værdi] * [Søndagsbonus]
```

Navngiv formelen Justeret værdi, og klik derefter på **OK**.



Alternativt kunne du i det sidste trin have brugt en multiplikations kolonne. **Tilføj kolonne** → **Standard** → **Multiplificér**, men jeg foretrækker fremgangsmåden Brugerdefineret kolonne.

Betinget logik er temmelig let ... ikke?

## Scenarie 2 - Søndagsbonus og produktrabat

I Scenarie 2 bliver tingene vanskelige... meget vanskelige.

Jeg kunne gennemgå en lang liste med transformationer for at illustrere, hvordan man gør det, men det vil jeg ikke. Funktionen Betinget kolonne er til grundlæggende opgaver; det tillader os ikke at bruge AND eller OR logik.

I stedet vil vi bruge mulighed 2, som betyder vi selv skal skrive M koden.

## IF-logik ved at skrive M koden

Før vi begynder med eksemplerne, skal vi bruge et par minutter på at forstå, hvordan if fungerer.

I en Excel verden vil funktionen HVIS blive skrevet som:

```
= HVIS([noget] = "noget 2"; [gør dette hvis det er sandt]; [gør dette hvis det er falsk])
```

Så hvis du kommer fra en Excel verden, kan det være lettere at tænke på det sådan her:

- Fjern start og slutparenteser
- Ret det første semikolon til then
- Ret det sidste semikolon til else
- Ret HVIS til if

Dette giver følgende M kode

## Kapitel 18 Power Query – IF-udtryk til betinget logik

```
= if([noget] = "noget 2" then [gør dette hvis det er sandt]
else [gør dette hvis det er falsk])
```

Når du har været igennem det et par gange, så vænner du dig til det. Det har den ekstra fordel, at det lyder som en almindelig simpel engelsk sætning.

For at indlejre en if i en anden if, bruger vi den samme if – then - else - metodik, som vist nedenfor

```
= if([det der skal testes #1] = "noget andet" then [gør dette hvis det er sandt]
else if [det der skal testes #2] = "noget helt andet"
then [gør dette hvis det er sandt]
else [gør dette hvis det er falsk])
```

Der kan være så mange if indlejringer, som vi ønsker. Dette fungerer på samme måde som at tilføje nye regler i en betinget kolonne.

Brug af AND eller OR logik er også ret enkel, vi bruger bare ordet and/or i koden. Koden herunder viser brugen af and logikken.

```
= if [det der skal testes #1] = "noget andet"
and [det der skal testes #2] = "noget andet #2"
then [gør dette hvis det er sandt] else [gør dette hvis det er falsk]
```

Hvor der er en blanding af både **and/or** logik, kræves parenteser for at sikre, at logikken evalueres i den rigtige rækkefølge.

Det var en gennemgang af det grundlæggende, nu er det tid til eksemplerne.

Start fra begyndelsen med at indlæse tabellen i Power Query igen. Derefter tilføjes kolonnen Ugedagsnavn, som vi gjorde i eksemplet ovenfor.

### Scenarie 1 - Søndagsbonus

Tilføj en brugerdefineret kolonne til tabellen ved at klikke på **Tilføj kolonne** → **Brugerdefineret kolonne**.

Brugerdefineret kolonne

Nyt kolonnenavn  
Søndag Premium

Formel for brugerdefineret kolonne:  
= if [Navn på dag] = "søndag" then [Værdi] \* 1.1 else [Værdi]

Tilgængelige kolonner:  
Dato  
Kunder  
Produkt  
Solgt af  
Værdi  
Navn på dag

<< Indsæt

Få mere at vide om Power-forespørgsel-formler

✓ Der er ikke registreret syntaksfejl.

OK Annuller

## Kapitel 18 Power Query – IF-udtryk til betinget logik

Brug følgende formel:

```
if [Navn på dag] = "søndag" then [Værdi] * 1.1 else [Værdi]
```

Vær opmærksom på ordene **if**, **then** og **else**, de alle er med små bogstaver. Power Query er følsom over for store og små bogstaver, så hvis det bliver stavet forkert, fungerer formelen ikke.

Det er det. Klik på **OK** for at tilføje formelen.

### Scenario 2 - Søndag bonus og produktrabat

Der er nogle få måder at skrive denne formel på. I sidste ende er der tre mulige resultater for dette scenarie:

- Søndagsbonus og produktrabat
- Kun søndagsbonus
- Ingen bonus eller rabat

Løsning nr. 1:

```
= if [Navn på dag] = "søndag" and ([Produkt] = "Frugt" or [Produkt] = "Kød") then [Værdi] * 1.1 * 0.95  
else if [Navn på dag] = "søndag"  
then [Værdi] * 1.1  
else [Værdi]
```

Brugerdefineret kolonne

Nyt kolonnenavn  
Bonus og rabat

Formel for brugerdefineret kolonne:  
=if [Navn på dag] = "søndag" and ([Produkt] = "Frugt" or [Produkt] = "Kød") then [Værdi] \* 1.1 \* 0.95 else if [Navn på dag] = "søndag" then [Værdi] \* 1.1 else [Værdi]

Tilgængelige kolonner:  
Dato  
Kunder  
Produkt  
Solgt af  
Værdi  
Navn på dag  
Søndag Premium

<< Indsæt

Der er ikke registreret syntaksfejl.

OK Annuller

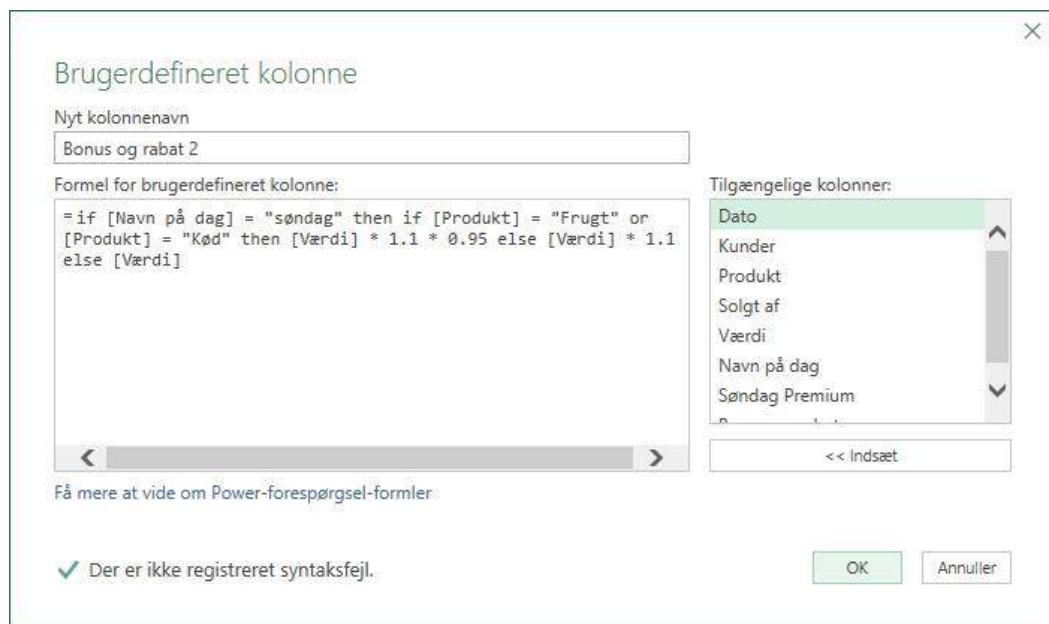
Vi har brug for at tilføje nogle parenteser for at sikre, at formelen udføres i den rigtige rækkefølge. Vi vil gerne have at or beregnes først, så vi sætter denne sektion i parenteser.



### Løsning 2:

Her er en anden løsning, du kan prøve.

```
if [Navn på dag] = "søndag" then  
if [Produkt] = "Frukt" or [Produkt] = "Kød"  
then [Værdi] * 1.1 * 0.95  
else [Værdi] * 1.1 else [Værdi]
```



## Konklusion

Betinget logik i Power Query ved hjælp af en if sætning er forskellig fra Excel. Det er dog ikke svært at forstå, så alt der er brug for er øvelse.

## Kapitel 19 Power Query – Gruppering og aggregering af data

Når jeg ser tilbage på denne Power Query bog, kan jeg se, at jeg har udeladt at skrive om en af de mere almindelige og ganske effektive funktioner; som er Grupper efter. Vi har allerede brugt funktionen et par gange, men i dette kapitel vil jeg at give det en mere detaljeret gennemgang.

I Excel er der mange måder at gruppere og opsummere data på. Vi bruger muligvis teknikker som os:

- Funktioner – SUM.HVIS(ER), SUMPRODUKT eller de nye **dynamiske arrayfunktioner**.
- Pivottabel.
- Funktionen Subtotal (Data → Subtotal).

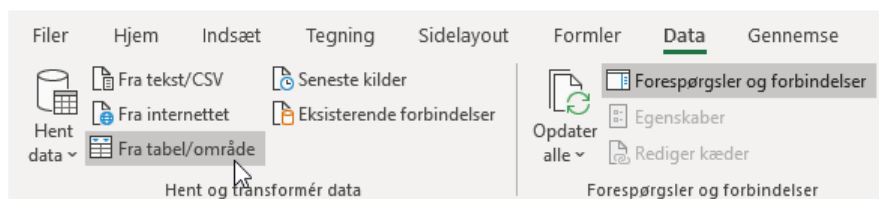
Nu er der en ny metode... funktionen Grupper efter i Power Query. Lad os se nærmere på denne.

For at gøre tingene lettere skal vi bruge den samme fil som i det forrige kapitel.

Eksemplerne i dette indlæg bruger filen kaldet **Eksempel 15 - Hvis Funktion.xlsx**.

### Eksempel på gruppering og aggregering

Eksempelfilen indeholder kun en tabel. Vælg en celle i tabellen, og klik på **Data → Fra tabel/område** for at indlæse dataene i Power Query.



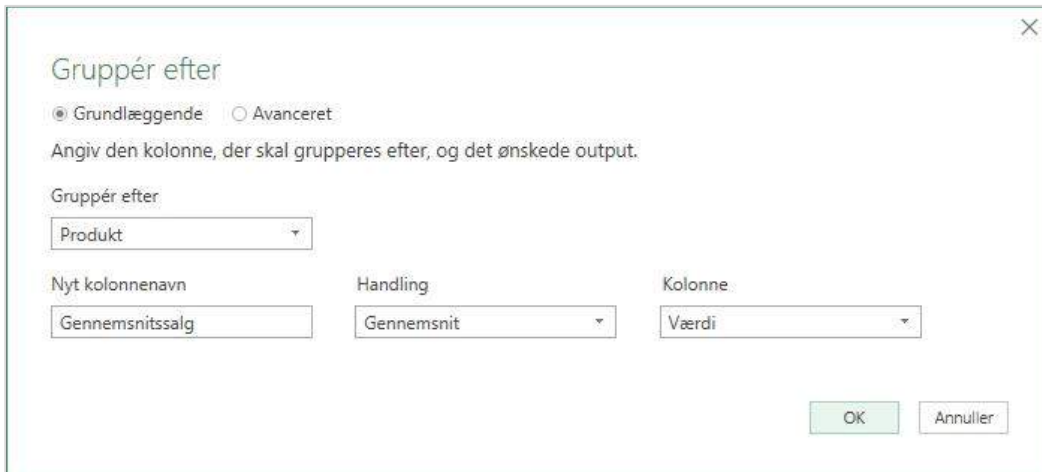
Når du først er i Power Query editoren er funktionen Grupper efter det første ikon på fanen Transformér: **Transformér → Grupper efter**. Ikonet vises andetsteds i brugergrænsefladen, men det er nemmeste at finde den her.



Grupper efter findes i en grundlæggende og en avanceret form, dog er de avancerede indstillinger temmelig lette at gå til.

### Grundlæggende Grupper efter

Den grundlæggende form giver en mulighed for at gruppere en enkelt kolonne og en enkelt handling. For at vise de **grundlæggende** indstillinger skal du sikre dig, at radioknappen Grundlæggende er valgt (som vist på skærbilledet nedenfor).



Hvis du arbejder med eksemplet, skal du vælge følgende indstillinger (disse vises også i skærbilledet ovenfor).

- Kolonne der skal bruges: Produkt
- Nyt kolonnenavn: Gennemsnitssalg
- Handling: Gennemsnit
- Kolonne: Værdi

Denne handling beregner gennemsnitsværdien af hver unik vare i kolonnen **Produkt**. Klik på **OK**.

Eksempelvinduet ser sådan ud:

	A <sup>B</sup> C Produkt	1.2 Gennemsnitssalg
1	Blomster	56,2
2	Kød	56,86666667
3	Brød	62,25
4	Frugt	54,57142857
5	Grøntsager	63

Der er mange operatører, som vi kan bruge:

- Sum
- Gennemsnit
- Median
- Min
- Maks
- Tæl rækker
- Tæl entydige rækker
- Alle rækker (som opretter en tabel for hvert grupperet element)

### Avanceret Grupper efter

De avancerede indstillinger giver os muligheden for at bruge lige så mange kolonner, som vi ønsker. Klik på radioknappen Avanceret for at få vist indstillingerne.

## Kapitel 19 Power Query – Gruppering og aggregering af data

Lad os sige at vi i dette eksempel, ønsker en total og antallet af salgstransaktioner for Produkt og Kunder.

I det første felt vælges Produkt. Klik på knappen **Tilføj gruppering**, og i det andet felt vælges Kunder.



Gruppér efter

Grundlæggende  Avanceret

Angiv de kolonner, der skal grupperes efter, og ét eller flere output.

Gruppér efter

Produkt

Kunder

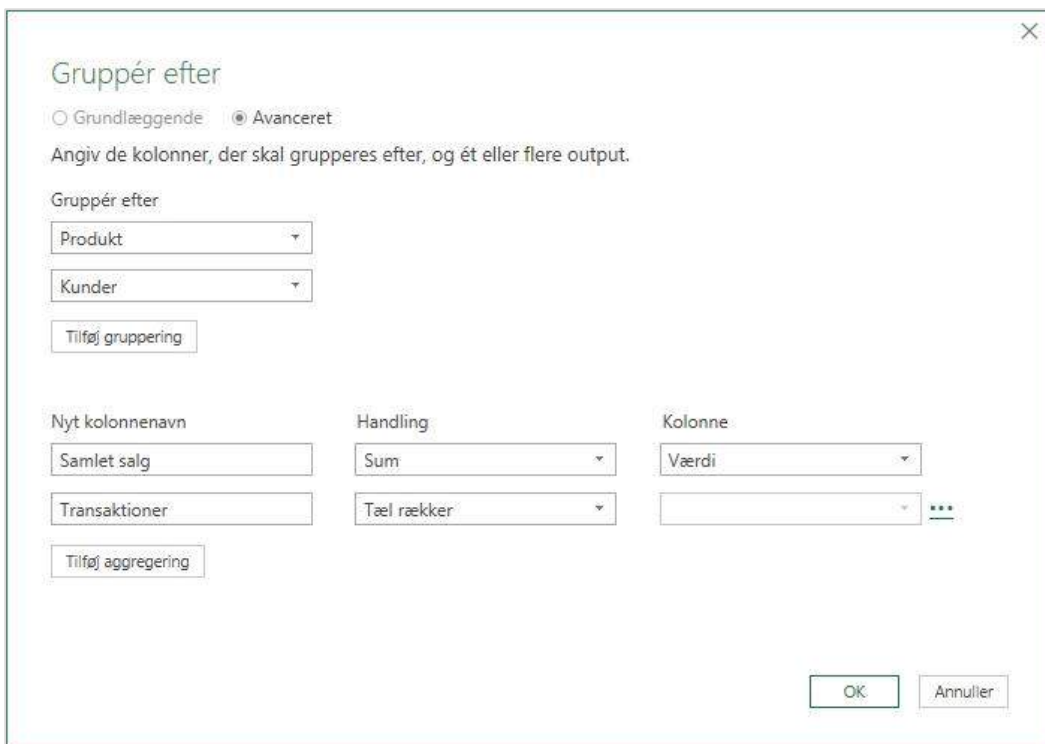
Tilføj gruppering

Anvend følgende indstillinger i den **nye kolonnenavn**sektion:

- Nyt kolonnenavn: Samlet salg
- Handling: Sum
- Kolonne: Værdi

Klik på knappen **Tilføj aggregering**, og indstil derefter den nye kolonne som følgende:

- Nyt kolonnenavn: Transaktioner
- Handling: Tæl rækker
- Kolonne: (ikke nødvendigt for Tæl rækker)



Gruppér efter

Grundlæggende  Avanceret

Angiv de kolonner, der skal grupperes efter, og ét eller flere output.

Gruppér efter

Produkt

Kunder

Tilføj gruppering

Nyt kolonnenavn

Samlet salg

Transaktioner

Tilføj aggregering

Handling

Sum

Tæl rækker

Kolonne

Værdi

OK Annuller

Klik **OK**.

## Kapitel 19 Power Query – Gruppering og aggregering af data

Eksempelvinduet ser nu sådan ud:

	A <sup>B</sup> <sub>C</sub> Produkt	A <sup>B</sup> <sub>C</sub> Kunder	1.2 Samlet salg	1.2 Transaktioner
1	Blomster	Brugsen	122	2
2	Kød	Netto	181	4
3	Brød	Føtex	66	1
4	Blomster	Netto	114	2
5	Kød	Bilka	78	2
6	Frugt	Bilka	156	3
7	Frugt	Rema 1000	50	1
8	Frugt	Lidl	75	1
9	Kød	Spar	167	3
10	Frugt	Meny	60	1

Dette viser de data, der aggregeres af de kombinerede kolonner Produkt og Kunder. Mens hver kolonne kan indeholde gentagelser, er hvert par unikt.

## Konklusion

---

Jeg håber du giver mig ret i at Grupper efter er nemt at arbejde med, men alligevel en effektiv funktion.

## Kapitel 20 Power Query – Brugerdefinerede funktioner

I dette kapitel skal vi udforske emnet brugerdefinerede funktioner. Hvad er det? Hvornår skal vi bruge dem? Og hvordan kan man oprette dem? I skrivende stund er der allerede over 600 standardfunktioner i Power Query. Alligevel gør disse funktioner ikke altid det, vi har brug for, eller på den mest effektive måde, derfor er vi indimellem nødt til at benytte brugerdefinerede funktioner for at opnå det ønskede slutresultat.

### Hvad er brugerdefinerede funktioner?

Funktioner er processer eller procedurer, der konsekvent udfører en opgave. For eksempel vil Excels SUM-funktion konsekvent summere værdier, eller LOPSLAG returnerer konsekvent en ækvivalent matchende værdi fra en liste.

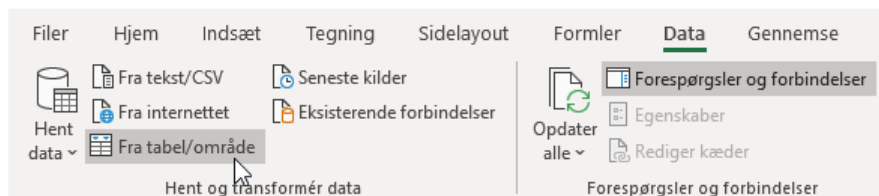
Bare fordi funktioner i Excel normalt beregner numeriske og tekst resultater, betyder det ikke, at dette skal være tilfældet. Faktisk er dette meget snæversynet. En funktion kan bruges til at udføre en række forskellige opgaver, og det er især tilfældet med Power Query. Vi kan bruge en brugerdefineret funktion til at hente data fra en webside eller til at læse dataene i en fil. Fra et Power Query perspektiv er det bedre at tænke på funktioner som en rutinemæssig opgave, der kan anvendes igen og igen.

Jeg tænkte i noget tid over, om jeg skulle inkludere brugerdefinerede funktioner i denne bog, da det kan være et temmelig vanskeligt område. Men jeg besluttede, at det er for værdifuldt at udelukke. Dette kapitel involverer en smule kodning og ændring af den automatisk genererede M kode, så følg nøje med.

Vi vil tackle dette område ved at arbejde os gennem to eksempler (1) en nem tilpasset funktion (2) en transformationsbaseret brugerdefineret funktion.

Vi bruger den samme fil som i forrige kapitel. Eksemplerne nedenfor bruger den kaldet Brugerdefinerede funktioner.xlsx

Eksempelfilen indeholder kun en tabel. For at indlæse dataene i Power Query skal du markere en hvilken som helst celle i tabellen og klikke på **Data** → **Fra tabel/område**.

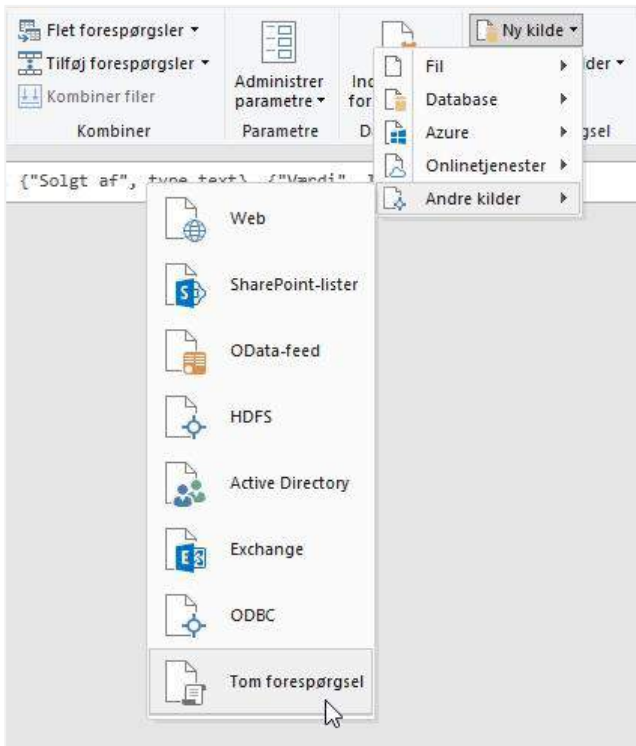


Giv forespørgslen et fornuftigt navn - Jeg har kaldt den **Data**, og det er den måde, jeg vil henvise til forespørgslen senere i kapitlet.

### En nem brugerdefineret funktion

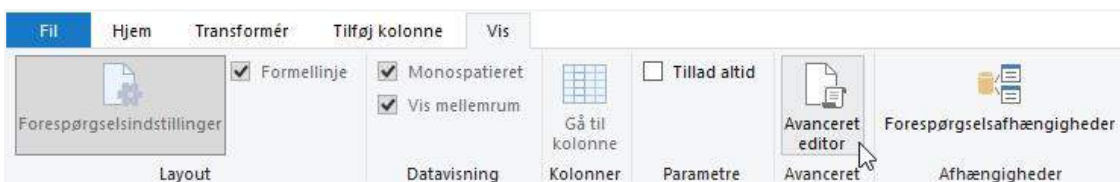
I dette første eksempel vil vi oprette en meget let brugerdefineret funktion. Det er udelukkende for at illustrere processen, så når vi kommer til sidste eksempel, vil der ikke være en overvældende læringskurve.

Opret en tom forespørgsel ved at klikke på **Hjem** → **Ny forespørgsel** → **Fra andre kilder** → **Tom forespørgsel**.



Giv forespørgslen et fornuftigt navn, til illustrationsformål har jeg kaldt den **minFunktion1** (vi ville ikke bruger dette navn i den virkelige verden, der vælger vi noget mere beskrivende, men det fungerer til vores illustration).

Åbn den avancerede editor for denne forespørgsel ved at klikke på **Vis** → **Avanceret editor** (eller **Hjem** → **Avanceret editor**).



Dialogboksen Avanceret Editor ser sådan ud:



Selvom det er en tom forespørgsel, bliver vi udstyret med en grundlæggende kode. Udtrykkene **let** og **in** forbliver i den endelige kode, men vi vil ændre resten af den.

I vores første eksempel er vores mål at oprette en brugerdefineret funktion, der multiplicerer værdien med 10.

Indtast følgende tekst i Advanced Editor.

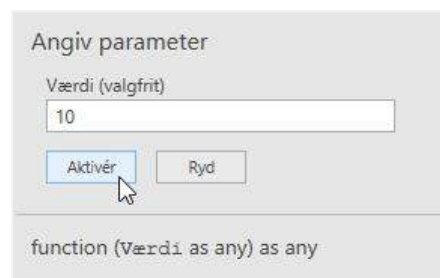
**HUSK: M kode er følsom over for store og små bogstaver, så skriv omhyggeligt.**

(Værdi) =>

```
let
    navnTrin = Værdi * 10
in
    navnTrin
```

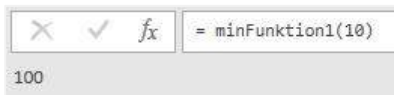
Vi vil oprette denne kode igen lidt senere i kapitlet, men her og nu skal du bare klikke på **Udført** for at lukke dialogboksen Avanceret editor.

Vi vil nu se følgende. Det er her vi kan teste funktionen, vi har oprettet. Indtast et tal i feltet **Værdi**, og klik på **Aktiver**.



Hvis vi har gjort alt korrekt, beregnes den korrekte værdi og den vise for os.





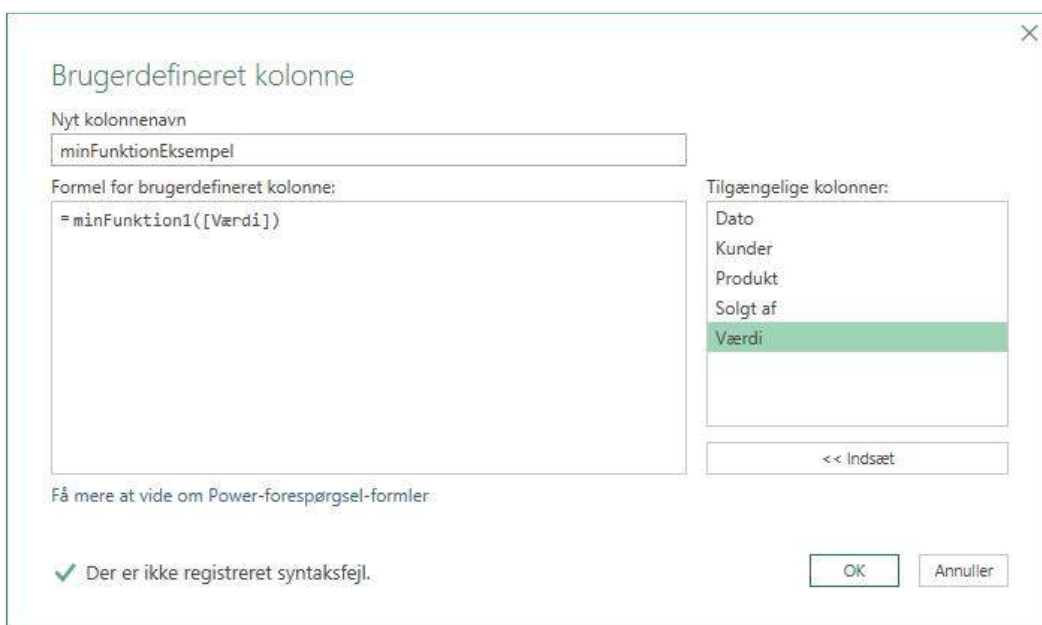
Ved at aktivere funktionen vil Power Query oprettet en forespørgsel kaldet **Kaldet funktion**, som vi dog ikke har brug for, så den kan vi bare slette. I den samme forespørgselsliste har du den forespørgsel, vi oprettede i starten (som vi kaldte Data) og den funktion, vi oprettede (kaldet minFunktion1).

Åbn derefter forespørgslen Data, vi oprettede i starten. Opret en brugerdefineret kolonne (**Tilføj kolonne** → **Brugerdefineret kolonne**) og brug den brugerdefinerede funktion, vi oprettede.

Koden, der skal bruges i dialogboksen for brugerdefinerede funktion er:

```
=minFunktion1([Værdi])
```

Bemærk: Ordet **[Værdi]** i funktionen er navnet på den kolonne, der skal bruges som argument.



Brug af en brugerdefineret funktion er ligesom at bruge en standard funktion i Power Query.

### Opret M koden igen

Lad os oprette koden igen, vi oprettede tidligere, for at forstå, hvad den gør.

### Angiv argumenterne

Den første linje erklærer de argumenter (eller input), der skal bruges i funktionen. I vores eksempel har det kun et input, som jeg har kaldt Værdi.

```
(Værdi) =>
```

Den anden linje identificerer starten på transformationstrinnene; der er ikke behov for at ændre noget i denne linje.

```
let
```

Derefter oprettede vi et trin, der kaldes **navnTrin**. Der er nogle regler om navnetrin, især med hensyn til brug af mellemrum. Men i stedet for at snakke om det nu er det lettere at altid bare at udelukke mellemrum, når man opretter manuelle trin.

navnTrin vil antage værdien (argumentet oprettet i den første linje) og multiplicere den med 10.

```
navnTrin = Value * 10
```

Endelig returnerer vi resultatet af funktionen. **in** betegner slutningen af transformationerne. Efter **in** udtrykket erklærer vi navnet på det trin, der skal returneres. I vores tilfælde har vi kun et trin, så vi returnerer `navnTrin`.

```
in
    navnTrin
```

## Flere argumenter og typer

Når du bruger standard Power Query funktioner, vil du bemærke, at de kan have flere argumenter og kræve specifikke datatyper. Vi kan gøre det samme i vores egne brugerdefinerede funktioner, der alt sammen håndteres i argumenterklæringen i starten af forespørgslen:

### Erklæring af inputtype

For at erklære en tekstdatatype ændrer vi koden på følgende måde:

```
(minTekst as text) =>
```

For kun at tillade en tabel som argument, vil vi bruge:

```
(minTabel as table) =>
```

Temmelig let, ikke sandt. Som du ved er der masser af datatyper. Hvis du ikke er i tvivl om koden for en datatype, skal du derefter bruge en anden forespørgsel, foretage en datatypeændring gennem via den normale brugergrænseflade, og gennemgå derefter koden, der er oprettet for at identificere ændringen.

### Erklæring af outputtype

Power Query har også en datatype til outputtet af funktionen; dette er angivet uden for parenteserne.

```
(minTekst as text) as table =>
```

Eksemplet ovenfor viser, at funktionen antager en tekstværdi som argument, og resultatet, der returneres, er en tabel datatype.

### Erklæring af flere argumenter

For at erklære flere argumenter i funktionen angiver vi dem imellem parenteserne.

```
(minTekst as text, minTabel as table) as number =>
```

Ovennævnte erklæring har:

- To argumenter kaldet (1) `minText` (med datatypen `text`) og (2) `minTabel` (med datatypen `table`).
- Output er blevet erklæret som datatypen `number`.

## Brugerdefinerede funktioner der indeholder transformationer

Da vi **importerede data fra internettet** i et tidligere kapitel, brugte vi den samme metode. Læs eventuelt dette kapitel igen for at få lidt mere øvelse. Men denne gang går vi efter noget lidt vanskeligere.

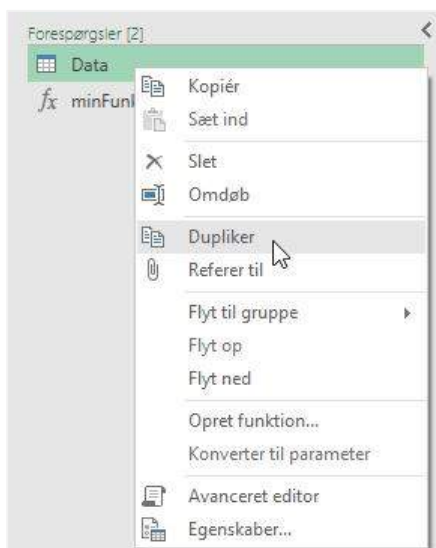
I dette eksempel beregner vi den procentsats, der udgør transaktionen, af det samlede køb for en kunde. Så hvis Bilka har købt for kr. 420, hvilken procentdel af det samlede beløb udgjorde hver transaktion. Billedet nedenfor er det endelige resultat, vi prøver at beregne. Bilka har ni transaktioner, den første er 13,1% og den anden er 11,67%.

	Dato	Kunder	Produkt	Værdi	Produkt % af Kunder
1	19-01-2020	Bilka	Frugt	55	13,10 %
2	14-01-2020	Bilka	Grøntsager	49	11,67 %
3	23-01-2020	Bilka	Kød	40	9,52 %
4	07-01-2020	Bilka	Frugt	44	10,48 %
5	31-01-2020	Bilka	Blomster	43	10,24 %
6	05-01-2020	Bilka	Kød	38	9,05 %
7	06-01-2020	Bilka	Blomster	52	12,38 %
8	04-01-2020	Bilka	Blomster	42	10,00 %
9	25-01-2020	Bilka	Frugt	57	13,57 %
10	09-01-2020	Brugsen	Blomster	76	62,30 %
11	22-01-2020	Brugsen	Blomster	46	37,70 %

I Excel termer er denne transformation som at dividere værdien i hver række med SUMHVISER for Kunder.

## Dupliker forespørgslen

Dupliker forespørgslen Data, vi importerede i starten. Dette skyldes, at vi har brug for (1) en forespørgsel som en funktion (2) en forespørgsel, som funktionen skal bruges til. For at gøre dette skal du **højre klikke** på forespørgslen i forespørgselslisten og klikke på **Dupliker**.



## Omdøb forespørgslen

Som vi så ovenfor, bruges det navn, vi giver forespørgslen, til at kalde funktionen.

Jeg har givet forespørgslen navnet **fnKunderTotal**. Placering af **fn** i starten er en almindelig praksis for brugerdefinerede funktioner; det hjælper med at visuelt identificerer, at det er en funktion.

## Transformationer

Slet alle trin undtagen kilde trinnet.

Klik derefter på **Transformer** → **Grupper efter**. Vi så, hvordan du bruger funktionen **Grupper efter** i det forrige kapitel. Indtast følgende indstillinger i vinduet **Gruppere efter**:

- Kolonne: Kunder
- Nyt kolonnenavn: Kunder total
- Handling: Sum

- Kolonne: Værdi

Gruppér efter

Grundlæggende  Avanceret

Angiv den kolonne, der skal grupperes efter, og det ønskede output.

Gruppér efter

Kunder

Nyt kolonnenavn Handling Kolonne

Kunder total Sum Værdi

OK Annuller

Klik på **OK**

Filtrer kolonnen Kunder for at vælge en enkelt kunde (Bilka i dette eksempel).

	Kunder	1.2 Kunder total
1	Brugsen	Sortér stigende
2	Netto	Sortér faldende
3	Føtex	Ryd sortering
4	Bilka	Ryd filter
5	Rema 1000	Fjern tom
6	Lidl	Tekstfiltre
7	Spar	Søg efter
8	Meny	(Markér alt)
9	ICA	<input checked="" type="checkbox"/> Bilka
		<input type="checkbox"/> Brugsen
		<input type="checkbox"/> Føtex
		<input type="checkbox"/> ICA
		<input type="checkbox"/> Lidl
		<input type="checkbox"/> Meny
		<input type="checkbox"/> Netto
		<input type="checkbox"/> Rema 1000
		<input type="checkbox"/> Spar

OK Annuller

Ændre forespørgslen til en funktion

Åbn den avancerede editor:



Her kommer den vanskelige del ... at foretage de korrekte ændringer i M koden.

Angiv argumenterne som den første linje i koden. Vi har kun brug for to argumenter (1), tabellen der skal bruges, og (2) kundenavnet, der skal filtreres på.

```
(kildeTabel as table, kundeNavn as text) =>
```

Da vi bruger en deklareret tabel som kilde, behøver vi ikke kildetrinnet, der allerede findes i koden. Slet den følgende linje.

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

I den næste linje er der stadig en henvisning til kildetrinnet, som vi lige har slettet. Vi er nødt til at pege denne linje til vores erklærede argument.

Skift dette:

```
#"Grupperede rækker1" = Table.Group(Kilde, {"Kunder"}, {"Kunder total", each List.Sum([Værdi]), type number})),
```

Til dette:

```
#"Grupperede rækker1" = Table.Group(kildeTabel, {"Kunder"}, {"Kunder total", each List.Sum([Værdi]), type number})),
```

Skift nu det hard kodede navn på kunden til det erklærede kundenavn argument: Ændr dette:

```
#"Filtrerede rækker" = Table.SelectRows(#"Grupperede rækker", each ([Kunder] = "Bilka"))
```

Til dette:

```
#"Filtrerede rækker" = Table.SelectRows(#"Grupperede rækker", each ([Kunder] = kundeNavn))
```

Vi kunne bruge dette i dets nuværende form. Hvis vi skulle, ville det returnere en tabel, der indeholder en enkelt værdi, snarere end en enkelt værdi i sig selv. Men vi kan gå et ekstra skridt videre så vi bare får returneret værdien.

Foretag følgende ændringer af M koden. Skift dette:

```
#"Filtrerede rækker" = Table.SelectRows("#Grupperede rækker", each ([Kunder] = kundeNavn))  
  
in  
  
    #"Filtrerede rækker1"
```

Til dette:

```
#"Filtrerede rækker" = Table.SelectRows("#Grupperede rækker", each ([Kunder] = kundeNavn)),  
  
    KunderTotal = Record.Field("#Filtrerede rækker"{0}, "Kunder total")  
  
in  
  
    KunderTotal
```

Record.Field er en Power Query funktion, der returnerer enkeltværdier fra en kolonne:

- **"#Filtrerede rækker" {0}** repræsenterer den første række fra tabellen, som den eksisterede efter trinnet, Filtrerede rækker
- **"Kundens total"** er navnet på kolonnen

**Advarsel: bemærkede du det ekstra komma i slutningen af linjen "#Fjernede kolonner"? Det skal være der, så tilføj det, hvis du mangler det.**

Den komplette M kode skal se sådan ud:

```
(kildeTabel as table, kundeNavn as text) =>  
  
let  
  
    #"Grupperede rækker" = Table.Group(kildeTabel, {"Kunder"}, {"Kunder total", each List.Sum([Værdi]), type number}),  
  
    #"Filtrerede rækker" = Table.SelectRows("#Grupperede rækker", each ([Kunder] = kundeNavn)),  
  
    KunderTotal = Record.Field("#Filtrerede rækker"{0}, "Kunder total")  
  
in  
  
    KunderTotal
```

Klik på **Udført** for at lukke dialogboksen Avanceret editor.

### Test funktionen

Lad os teste funktionen. Vælg forespørgselstabellen Data, der skal bruges som kildetabel, og indtast et kundenavn. Klik på Aktiver.



Angiv parametre

kildeTabel  
Data

kundeNavn  
Bilka

Aktiver Ryd

function (kildeTabel as table, kundeNavn as text) as any

Det burde returnere en værdi:

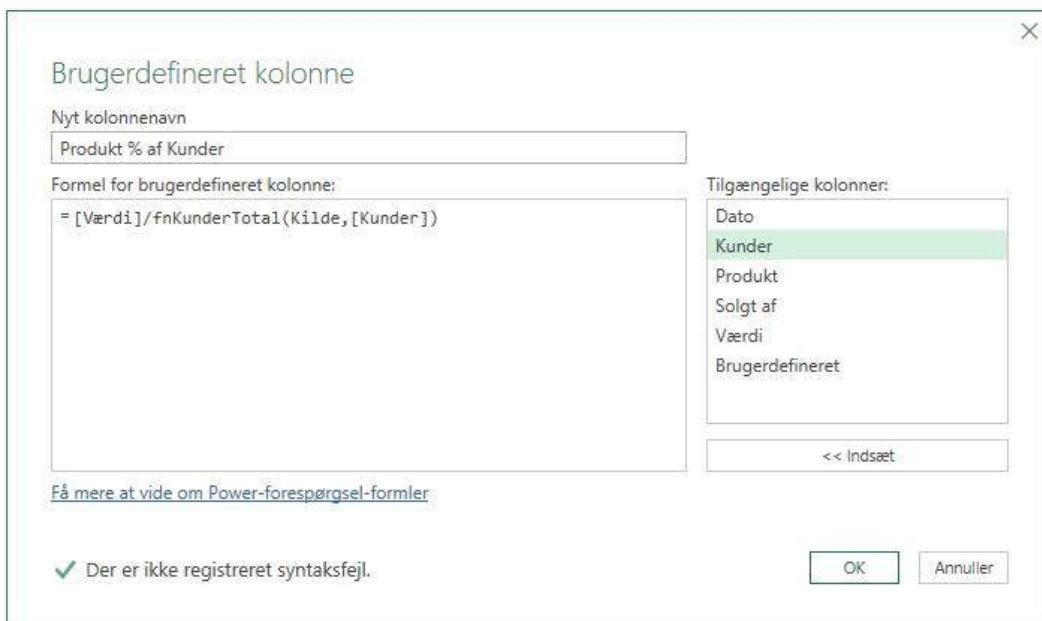


## Brug funktionen

Alt, hvad der er tilbage nu, er at bruge den brugerdefinerede funktion. Åbn dataforespørgslen, og tilføj en brugerdefineret kolonne. Indtast følgende som formelen:

```
= [Værdi] / fnKunderTotal(kilde, [Kunder])
```

- **Kilde** er trinnets navn (dvs. brug tabellen, som den eksisterede efter kildetrinnet, er afsluttet)
- **[Kunde]** er kundekolonnen.



Klik på **OK** for at acceptere funktionen.

Til sidst skal du udføre følgende transformationer:

- Ret den nye kolonne til datatypen %.
- Sorter efter kolonnen Kunder.
- Fjern kolonnen Solgt af.

Eksempelvinduet vil nu se sådan ud og viser % fordelingen for hver transaktion for kunden.

	Dato	A <sup>B</sup> <sub>C</sub> Kunder	A <sup>B</sup> <sub>C</sub> Produkt	1 <sup>2</sup> <sub>3</sub> Værdi	% Produkt % af Kunder
1	19-01-2020	Bilka	Frugt	55	13,10 %
2	14-01-2020	Bilka	Grøntsager	49	11,67 %
3	23-01-2020	Bilka	Kød	40	9,52 %
4	07-01-2020	Bilka	Frugt	44	10,48 %
5	31-01-2020	Bilka	Blomster	43	10,24 %
6	05-01-2020	Bilka	Kød	38	9,05 %
7	06-01-2020	Bilka	Blomster	52	12,38 %
8	04-01-2020	Bilka	Blomster	42	10,00 %
9	25-01-2020	Bilka	Frugt	57	13,57 %
10	09-01-2020	Brugsen	Blomster	76	62,30 %
11	22-01-2020	Brugsen	Blomster	46	37,70 %

### Næste skridt

Hvis du vil tage dette til næste niveau, hvorfor så ikke prøve at ændre funktionen, så den fungerer på en hvilken som helst kolonne i stedet for kun en kolonne kaldet "Kunder".

### Konklusioner

I dette kapitel har vi set, hvordan man opretter grundlæggende og transformationsbaserede brugerdefinerede funktioner. Denne metode kan bruges, når vi har brug for en værdi eller værdier, der ikke er i samme række. Nogle eksempler er:

- Løbende totaler
- % af en helhed
- Værdier i forrige eller næste række

Manipulering af M koden er vanskelig og tilbøjelig til fejle, især når man endnu ikke har nogen erfaring. Derfor, som med så mange andre funktioner i Power Query, er det bedste råd at øve, øve, øve!



## Kapitel 21 Power Query – Almindelige fejl og hvordan man retter dem

Jeg er sikker på, at du ikke er kommet så langt uden at støde på din del af Power Query fejl. Ligesom Excel og andre programmer har Power Query dens egne unikke fejlmeddelelser. Du har sandsynligvis glemte dem første gang, du stødte på fejl som #NAVN? eller #VÆRDI! i Excel, men over tid har du forhåbentlig fundet ud af, hvad du skal gøre, når de dukker op. Nu, hvor du arbejder med Power Query, virker fejlmeddelelserne underlige og ukendte og kan være skræmmende i starten. Med tiden vil du begynde at forstå, hvad fejlene består i og hvad der forårsager dem.

Selvom vi ikke kan gennemgå alle fejl, er formålet med dette kapitel at hjælpe med at afmystificere nogle af de mere almindelige fejl, du sandsynligvis vil støde på.

### Fejl typer

---

Fejlmeddelelser kan vises forskellige steder, f.eks. i panelet Projektmappe-forespørgsler, i Power Query editoren, eller måske bare som en værdi i et felt.

Jeg har grupperet fejlene i tre typer:

- Procesoprettelsesfejl.
- Fejl ved databehandling.
- Softwarebugs.

Vi vil se på hver af disse efter tur og finde ud af, hvordan vi løser almindelige problemer.

### Procesoprettelsesfejl

---

Procesoprettelsesfejl opstår, når vi laver vores forespørgsel. Disse er drevet af enten fejl i M koden, eller brugere, der ikke forstår, hvordan Power Query fungerer.

#### M kodefejl

M kodefejl kan være udfordrende at finde, især hvis du er ny i M sproget. Et komma eller et indtastet ord er nok til at få processen til at mislykkes. De tre vigtigste steder, hvor vi kan redigere M koden:

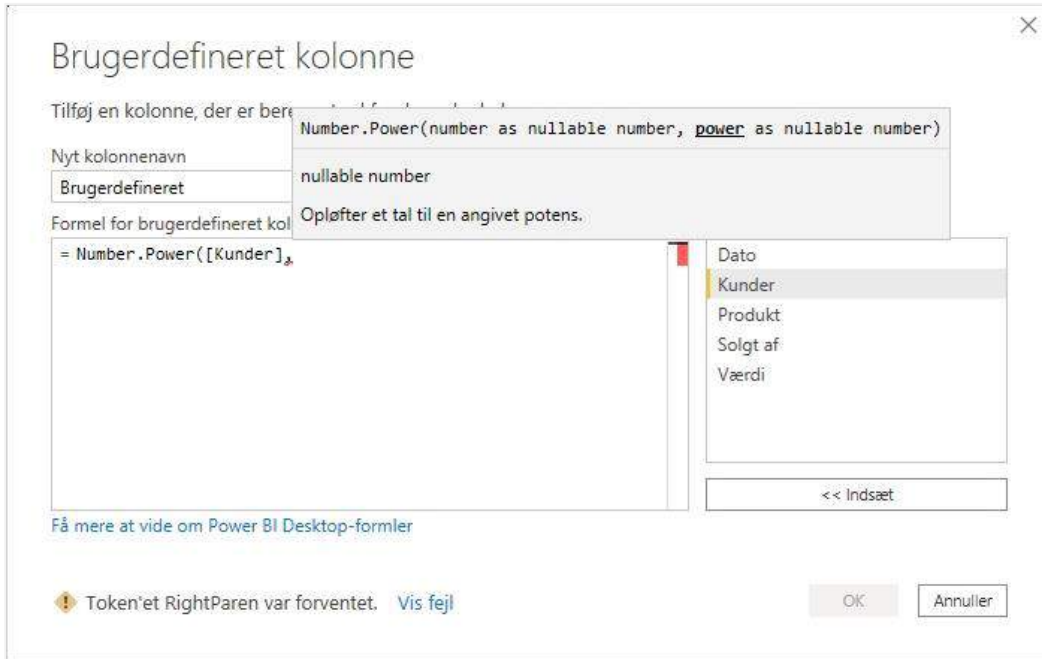
- Brugerdefinerede kolonner.
- Avanceret editor.
- Formellinjen.

Lad os starte med at se på brugerdefinerede kolonner, og derefter går vi videre til at se på Avanceret editor og formellinjen sammen.

#### Brugerdefinerede kolonner

Af M kode indstillingerne er funktionen Brugerdefineret kolonne den mest tilgængelige og den, vi er mest tilbøjelige til at bruge

Brugerdefinerede kolonner indeholder en advarsel nederst i dialogboksen der hjælper os med vores formler. Medmindre man har arbejdet med Power Query i et stykke tid, forstår man ikke, hvad mange af disse meddelelser betyder, og selv da kan disse fejlmeddelelser være forvirrende.



Skærbillede ovenfor viser meddelelsen **Token'et RightParen var forventet**. Dette er kun en af mange potentielle meddelelser. Når vi skriver i formelboksen, ændres meddelelsen. Derfor kan det ikke betale sig at se på denne meddelelse, før vi beslutter os for, at formelen er færdig. Hvis indstillingen Vis fejl er synlig, kan vi klikke på den og blive ført hen til det sted, hvor problemet er.

Når du først ved, hvad meddelelserne betyder, er de ikke så vanskelige, som det kan synes. De mest almindelige advarsler, du vil støde på, er:

- **Token'et Literal forventet** betyder, at det næste element i formlen forventes at være en værdi, kolonnenavn eller en funktion.
- **Token Then forventet**, eller **Token Else** forventet betyder ordene **then** eller **else** at blive indtastet herefter. Disse vises, når du skriver en if-erklæring.
- **Token RightParen forventet** betyder, at en afsluttende parentes, forventes for at lukke en formel.
- **Et komma kan ikke foranstilles RightBrace** betyder, hvad det siger; et komma kan ikke være direkte foran et lukkebeslag, der er ingen omstændigheder i M, hvor dette skulle være nødvendigt.
- **Ugyldig id** indikerer, at der er et problem med den indtastede værdi som bogstav. Dette opstår ofte, når en teststreng ikke er blevet lukket ved hjælp af det dobbelte citationstegn.
- **Token EoF forventet**, forekommer normalt, når der bruges et ugyldigt funktionsnavn, eller det bruges forkerte store og små bogstaver (for eksempel er **if** en gyldig kommando, mens **If** med stort forbogstav er ugyldig).
- **Formlen er ufuldstændig** betyder som regel, at der ikke er indtastet nogen formel (f.eks. at kun lighedstegnet er i formelboksen).

De nyere versioner af Power Query inkluderer IntelliSense, hvilket reducerer risikoen for fejl markant. Det hjælper med at identificere de argumenter og datatyper, der kræves til hver funktion, men hjælper også med at minimere tastefejl.

Når vi får den meddelelse, der siger, at der **ikke er fundet nogen syntaksfejl**, kan vi klikke på **OK**-knappen for at lukke vinduet. Dette betyder ikke, at formelen eller datatyperne er korrekte, men at syntaksen er indtastet korrekt.

## Avanceret editor og formellinje

Den avancerede editor og formellinjen har ikke det samme niveau af kontrol som en brugerdefineret kolonne; Du kan acceptere ændringer, selvom det forårsager en fejl. Dette betyder, at antallet af forskellige fejlmeddelelser øges, når du bruger disse funktioner:

- Den avancerede editor har den samme advarselsmeddelelse i bunden som en brugerdefineret kolonne, men gør det muligt at klikke på Udført, selvom der er en fejl i koden.
- Formellinjen har ingen fejlkontrol, så du kan ændre i koden og trykke på Enter tasten for at acceptere disse ændringer.

I betragtning af de mange mulige fejl, du kan oprette, kan jeg ikke gennemgå dem alle. Når du først ved, hvordan du læser fejlmeddelelsen, er det imidlertid meget lettere at fejlfinde.

Skærmbillede herunder viser en Expression.SyntaxError ...og hvad betyder det? Hvis du kigger længere nede i koden, har Power Query så nydeligt vist os, hvor fejlen er. Tallene til venstre viser linjenumrene i koden. Hvis du bemærker, er der et brud på numrene med en pil ---->, dette er for at indikere, at denne linje fremhæver, hvor fejlen er. Ved at kigge langs den linje finder vi en sæt af ^^, der peger på det nøjagtige sted, hvor fejlen befinder sig.

```
0001: let
0002:     Kilde = Csv.Document(File.Contents("C:\Power Query\Kapitel 3\Opdater data 1.csv"), [Delimiter=";"]);
0003:     #"Hævede overskrifter" = Table.PromoteHeaders(Kilde, [PromoteAllScalars=true]);
0004:     #"Endret type" = Table.TransformColumnTypes(#"Hævede overskrifter", {"Dato", type dat}, {"Kunde";
0005: in
0006:     #"Endret type"
```

I vores eksempel ovenfor er fejlen, at vi har brugt datatypen **dat**, som ikke er gyldig.

Hvor der er flere fejl i koden, er vi muligvis nødt til at gennemgå flere omgange med fejlrettelse, da fejlmeddelelsen kun viser en fejl ad gangen.

## Formel.Firewall fejl

Der er en meget frustrerende fejl, der vil dukke op fra tid til anden: **Formula.Firewall: Query '[Name of query]' (name of step) references other queries or steps, so it may not directly access a data source. Please rebuild this data combination.**

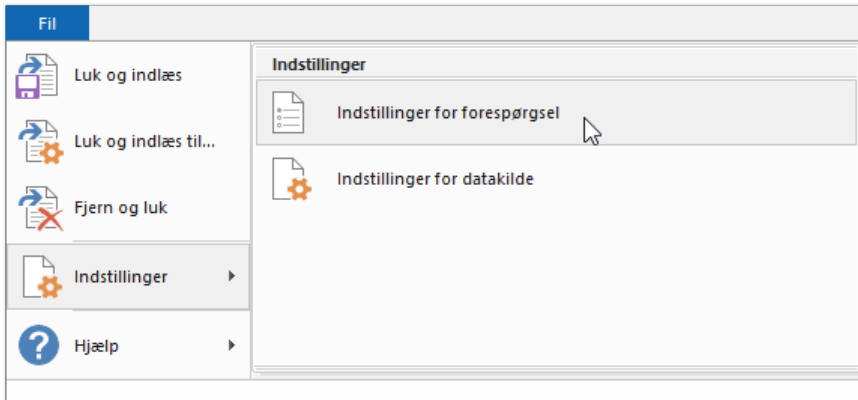
Hvad betyder det? Og hvordan kan du ordne det?

Power Query kan ikke lide at bruge to datakilder sammen, hvor privatlivets indstillinger er forskellige. Dette opstår normalt, når der er eksterne og interne datakilder, der bruges i den samme forespørgsel.

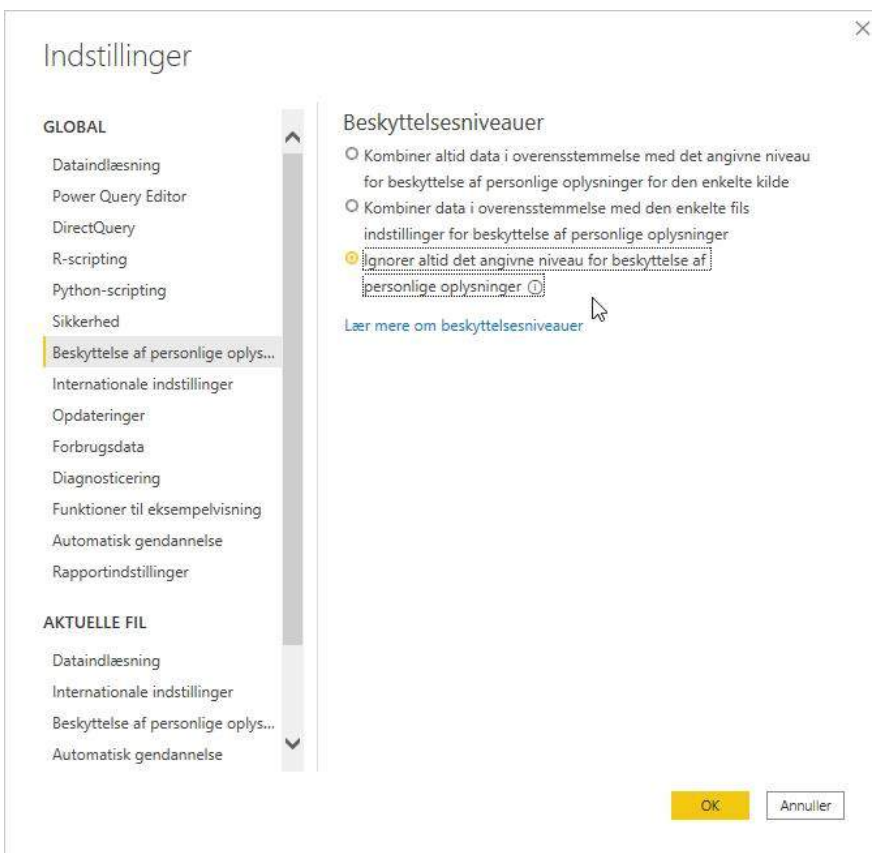
## Ignorer personlige indstillinger

Denne mulighed er ikke ideel, da den ignorerer indstillingerne for databeskyttelse. Hvis du imidlertid er den eneste person, der får adgang til dataene, er det en nyttig løsning.

Klik på **Fil** → **Indstillinger** → **Indstillinger for forespørgsel**.



Dialogboksen Indstillinger åbnes. Vælg **Beskyttelse af personlige oplysninger** → **Ignorer altid det angivne niveau for personlige oplysninger**, og klik derefter på **OK**.



Opdater dataene for at opdatere forespørgslen. Data indlæses nu korrekt.

Der er også andre metoder til at komme omkring dette problem, men de er for avanceret til denne bog.

## Fejl ved databehandling

Fejl ved databehandling opstår, når dataene føres igennem transformationsprocessen. Der er muligvis ikke noget specifikt galt med dataene eller processen, men alligevel fungerer de to ikke godt sammen. Det kan være noget så simpelt i transformationstrinnene, hvor der forventes at der er en kolonne kaldet "Produkt", men en "Produkt" kolonne findes ikke i datasættet. Ingen af dem er forkerte, de passer bare ikke sammen.

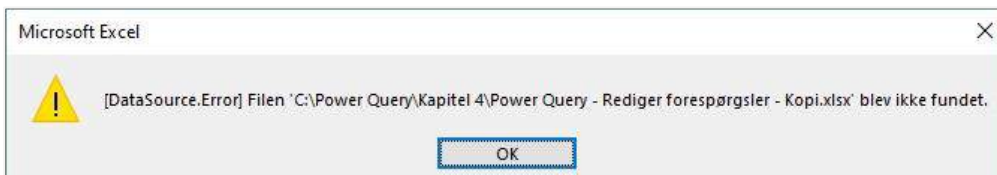
De mest almindelige fejl på dette område er:

- Forkert kilde placering.
- Kolonnenavn ændres.
- Forkerte datatyper.

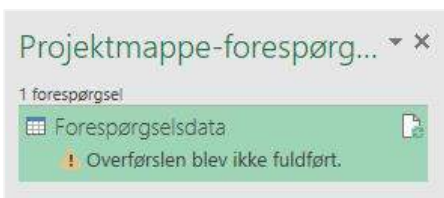
Lad os se på hver af dem lidt mere detaljeret.

### Forkert kilde placering

Den forkerte kilde placeringsfejl opstår, når enten en fil eller database har ændret placering, eller hvor en server er stoppet, og der derfor ikke kan fås adgang til kilden. Uanset hvad kan Power Query ikke finde kildedataene. Efter opdatering vil en fejlmeddelelse som følgende vise en detaljerede filplacering, som den ikke kan finde.



Vi vil også se en fejl i panelet Projektmappe-forespørgsler. Hvis vi dobbeltklikker på forespørgslen, vil vi få en mere detaljeret beskrivelse af fejlen.



Power Query editoren åbner og viser følgende meddelelse. Klik på **Gå til fejl** for at gå til det nøjagtige trin.



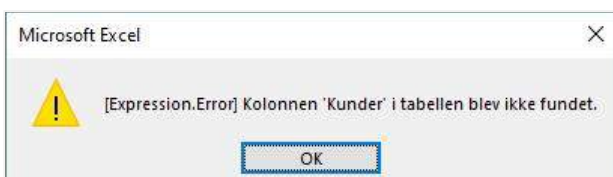
Endelig kan du klikke på **Rediger indstillinger** for at ændre kildeplaceringen i vinduet.



Der er andre og måske bedre muligheder for at ændre kildedataplacering, jeg har skrevet om dette i et tidligere kapitel: Skift placering for kildedata, så tjek det for flere detaljer.

### Kolonnenavn

Generelt er kolonneoverskriftsnavne hardkodet et sted i M koden. Derfor kan ændringer i kildedatasstrukturen udløse følgende fejl.



Panelet Projektmappe-forespørgsler viser den samme fejl **download afsluttede ikke**, vi så tidligere. Åbning af forespørgslen vil afsløre yderligere detaljer om fejlen.



For at rette fejlen kan vi enten:

- Skift overskriftens navn i kildedataene.
- Ret den hardkodede værdi i M koden via Avanceret editor eller Formellinjen.
- Slet det gamle trin, og indsæt et nyt, der fanger det nye kolonnenavn.

Uanset hvilken mulighed du vælger, skal du være forsigtig. Dårligt implementerede ændringer kan forårsage andre problemer længere nede i forespørgslen. Den gode nyhed er, at du altid kan kassere ændringer og starte igen.

## Datatyper

Fejl i datatypen forhindrer ikke indlæsning af data i forespørgslen; i stedet indlæses disse celler som tomme. Panelet Projektmappe-forespørgsler viser fejlen og angiver antallet af linjer med fejl.



Skærbillede ovenfor viser 300 fejl, men det kan også bare være 1 eller 2, afhængigt af strukturen på dataene.

Fejl i datatypen opstår, når:

- Data konverteres fra en type til en anden - for eksempel i forsøget på at ændre en tekststreng til en decimal datatype.
- Forkerte datatyper, der bruges i funktioner - for eksempel i forsøget på at bruge en talfunktion på en tekstdatatype eller forsøge at multiplicere ikke-numeriske datatyper.

Excel er meget tilgivende og skifter gerne mellem datatyper, hvor det kan lade sig gøre. Power Query er ikke så tilgivende, derfor er det vigtigt at få den rigtige datatype.

Klik på linket "Fejl" i panelet Projektmappe-forespørgsler for at åbner forespørgslen, der viser alle fejlene.



1.2 Række nummer	1.2 Solgt af	Dato	A <sup>B</sup> C Kunder	A <sup>B</sup> C Produkt	1 <sup>2</sup> 3 Værdi
1	Error	09-01-2019 00:00...	Brugsen	Blomster	76
2	Error	02-01-2019 00:00...	Netto	Kød	40
3	Error	19-01-2019 00:00...	Føtex	Brød	66
4	Error	19-01-2019 00:00...	Netto	Kød	51
5	Error	12-01-2019 00:00...	Netto	Blomster	75
6	Error	22-01-2019 00:00...	Brugsen	Blomster	46
7	Error	23-01-2019 00:00...	Bilka	Kød	40
8	Error	25-01-2019 00:00...	Bilka	Frugt	57

Som det kan ses på skærmbilledet ovenfor, viser Power Query hvor fejlene er og tilbyder endda Rækkenummer, så du kan identificere problemet med kildedataene. Efter at have klikket på ordet "Error" i eksempelvisningsvinduet vises der detaljer om det specifikke problem.



I skærmbilledet kan vi se, at Power Query forsøgte at konvertere tekstværdien "Lene" til et tal, hvilket ikke er muligt.

Selvom der kan være flere linjer med fejl, betyder det ikke, at du skal rette hver række individuelt. Ændring af et trin kan være nok til at rette alle fejl på samme tid.

## Softwarebugs

---

Da jeg begyndte at bruge Power Query, stødte jeg på to softwarebugs. I begge tilfælde konkluderede jeg, at det var min fejl, at jeg ikke forstod værktøjet korrekt. Det var dog ikke mig, men softwaren, der ikke fungerede korrekt.

Idet Power Query løbende opdateres, kan fejl komme og gå hurtigt, når nyere versioner frigives. I løbet af de sidste par år vil jeg sige, at softwaren er blevet robust og nu sjældent lider under problemer.

Forhåbentlig vil du ikke støde på nogen af de problemer, jeg havde; de er allerede løst. Derfor, hvis du støder på et problem, hvor softwaren ikke opfører sig som dokumenteret, bør opdatering til den nyeste version løse problemet.

## Kapitel 22 Power Query – Tips og tricks

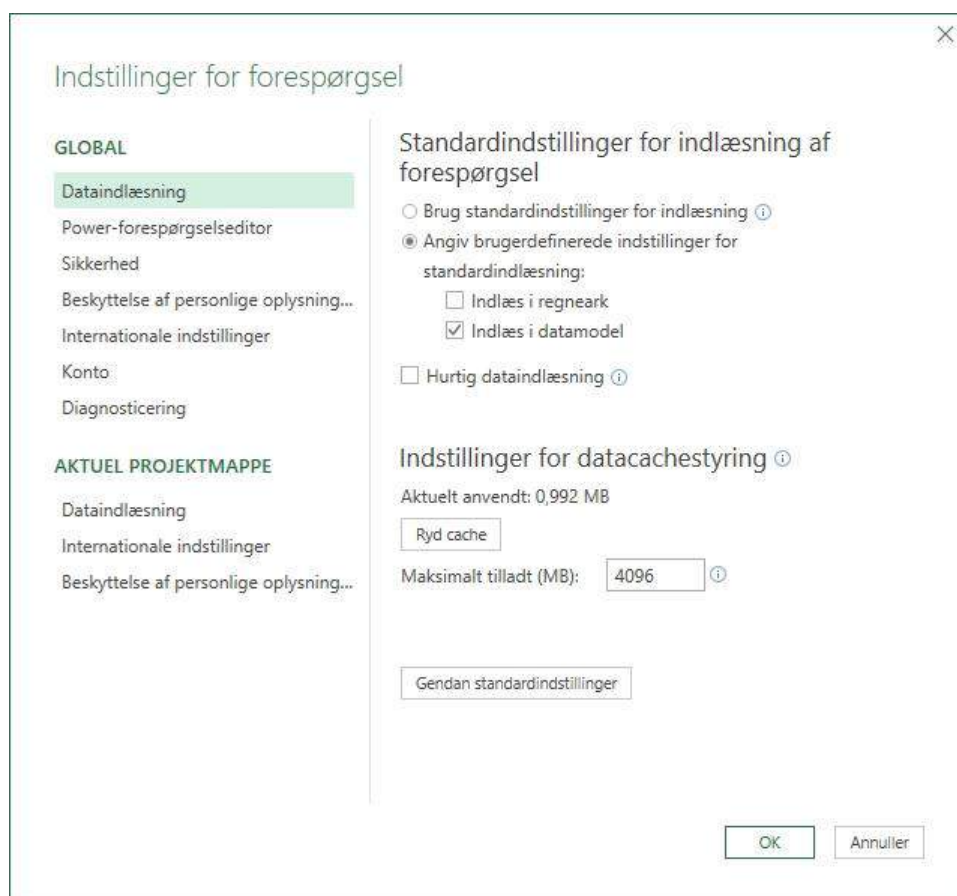
Det er nu tid til at afslutte denne Power Query bog. Og hvad er bedre at slutte end med end nogle tip og tricks, der hjælper dig med at få succes.

### Skift standardindstillingerne for Luk og indlæs

Power Query's standardindstillinger for Luk og indlæs antager, at vi vil indlæse dataene i en Excel tabel. Når du arbejder videre med Power Query, vil dette blive mindre og mindre ønskeligt. Når du kommer videre på din "Power"-rejse, er det sandsynligt, at du bruger data i Power Pivot, hvilket kræver indlæsning i datamodellen.

Hvis du vil ændre standardindstillingerne for Luk og indlæs, skal du åbne Power Query editoren og derefter klikke på **Fil** → **Indstillinger** → **Indstillinger for forespørgsel**.

I dialogboksen Indstillinger skal du vælge indstillingen **Global** → **Dataindlæsning**.



I denne dialogboks finder vi indstillingen **Angiv brugerdefinerede indstillinger for standardindlæsning**:

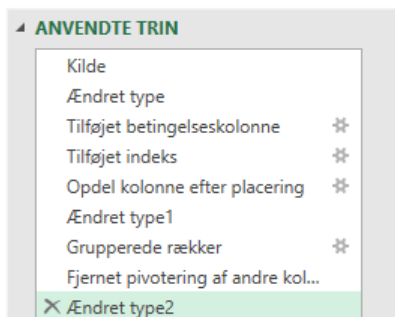
- **Indlæs i regneark** - hvis det er afkrydset, indlæses dataene til regnearket, hvis det ikke er afkrydset, indlæses det kun som en forbindelse.
- **Indlæs til datamodel** - hvis det er afkrydset, indlæses data i datamodellen, hvis det ikke er afkrydset, indlæses intet i datamodellen. For at bruge dataene i Power Pivot, skal denne indstilling være afkrydset.

Valg af **Brug af standard for indlæsning** svarer til at afkrydse **indlæs regneark** og fjerne afkrydsning af **indlæsning til Datamodel**.



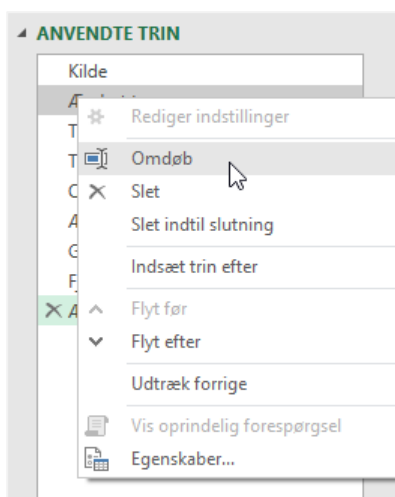
## Giv trinene et meningsfuldt navn

Jeg må indrømme at jeg ikke har fulgt mine egne råd i denne bog. Det at omdøbe trinene med et meningsfuldt navn kan virkelig spare dig tid. Det virker måske ikke sådan på nuværende tidspunkt, men på længere sigt vil det være. Det er så let at have forespørgsler, hvor de anvendte trin ser sådan ud:



Hvad foretog de forskellige trin sig? Der er alt for mange at huske på, så vi bliver nødt til at fortsætte med at vende tilbage til forhåndsvisningspanelet og formellinjen i et forsøg på at finde ud af, hvad vi foretog os. Hvis vi bare havde omdøbt trinnene, da vi oprettede forespørgslen.

Det er nemt at omdøbe trinnene – **Højre klik** på trinnet og vælg **Omdøb** i menuen.



Dette ændrer også navnet på trinnet i den avancerede editor.

## Sørg altid for at formellinjen er synlig

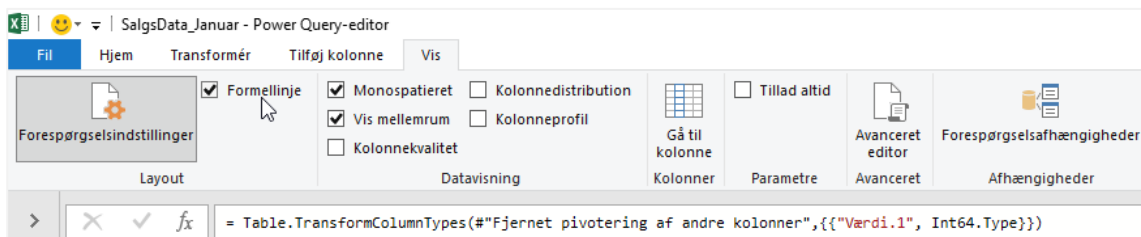
Formellinjen er den perfekte balance mellem kompleksiteten i vinduet Avanceret editor og enkelheden i hovedbrugergrænsefladen.

M koden hjælper os med at:

- Lære og værdsæt, hvad Power Query laver i baggrunden.
- Lave hurtige små kodeændringer, hvor det er nødvendigt.

Hvis du kommer fra en Excel verden, er formellinjen et velkendt element i vinduet. Vi forstår muligvis ikke alt heri, men det betyder ikke, at den ikke er nyttigt.

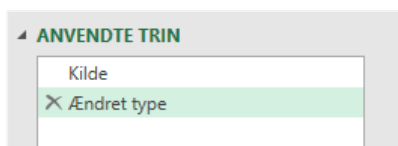
For at aktivere formellinjen skal du klikke på **Vis** → **Formellinjen** i Power Query editoren.



## Forhindre automatisk registrering af datatype

Hver gang vi importerer data, gætter Power Query på, hvilken datatype hver kolonne indeholder. Baseret på dette gæt tilføjes et nyt trin til listen: Anvendte trin.

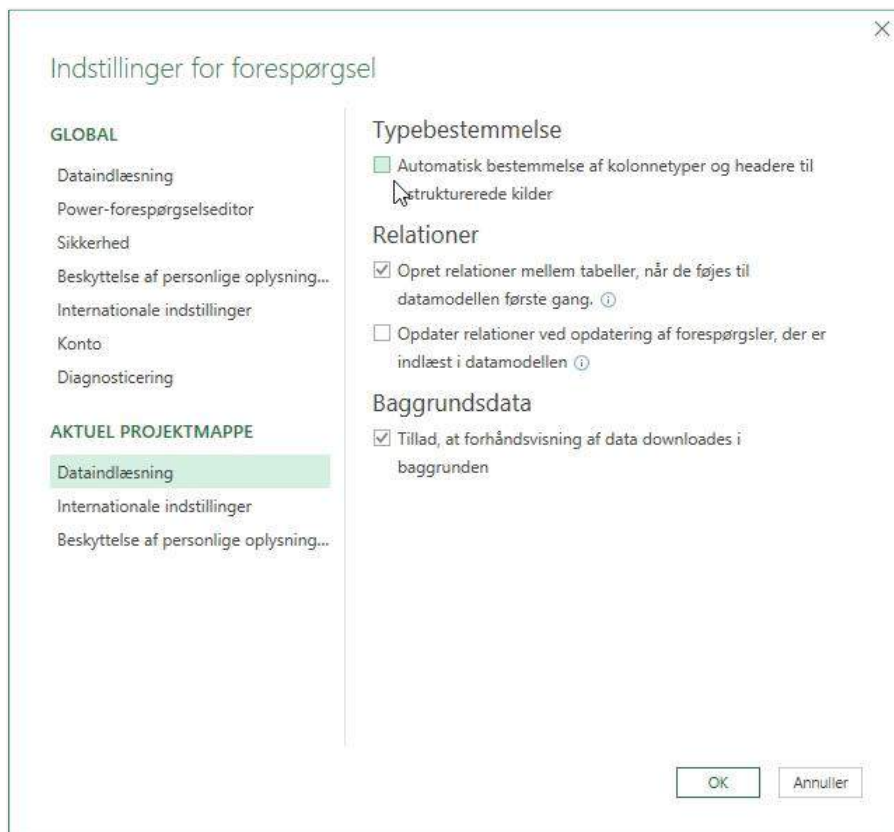
Hvis dataene ikke er strukturerede, f.eks. En CSV-fil eller et navngivet område, fremhæves den første række automatisk til overskriftsrække.



Vi vil indimellem opdage, at vi regelmæssigt sletter disse trin. Derfor kan det være hurtigere og lettere at deaktivere registrering af datatypetypen. Klik på **Fil** → **Indstillinger** → **Indstillinger for forespørgsel** i Power Query editoren.

I dialogboksen Indstillinger skal du vælge gruppen **Aktuel projektmappe** → **Dataindlæsning**.

Fjern afkrydsningen fra **Automatisk bestemmelse af kolonnetyper og headere til ustrukturerede kilder**, og klik derefter på **OK**.

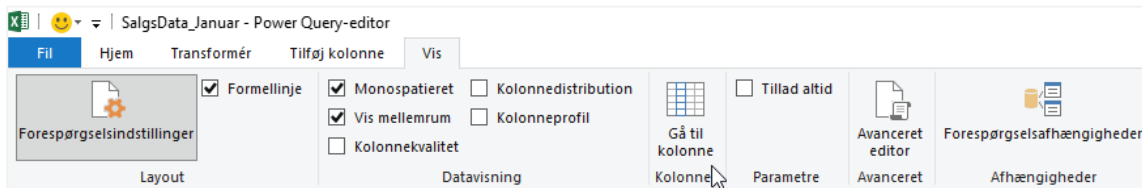


Eventuelle forespørgsler, der oprettes, vil nu kun indeholde kildetrinnet. Det er op til os at beslutte, om og hvornår vi skal fremhæve overskrifter og hvilke datatyper, der skal bruges.

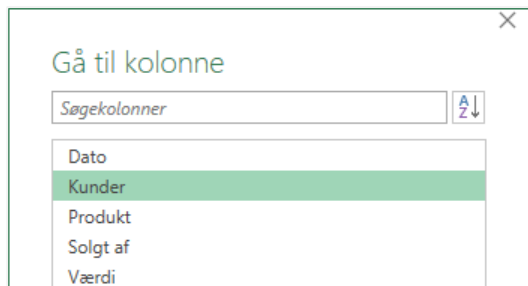
## Brug Gå til kolonne

Mens du arbejder med en tabel, der indeholder masser af kolonner, kan det være tidskrævende at finde den ønskede kolonne; bevægelse frem og tilbage med den vandrette rullebjælke er ikke hurtig. Brug af venstre og højre markørtaster kan være en nyttig, men der er en bedre måde, kaldet **Gå til kolonne**.

Gå til kolonne-ikonet skjuler sig på fanen Vis (som vi sjældent bruger). Klik på **Vis** → **Gå til kolonne**.



Dialogboksen Gå til kolonne er enkel og let at bruge.



Der er ingen flervalgsmuligheder eller komplekse markeringer, du skal du blot vælge kolonnenavnet og derefter klikke på OK (eller dobbeltklikke på kolonnenavnet for at undgå at klikke på OK).

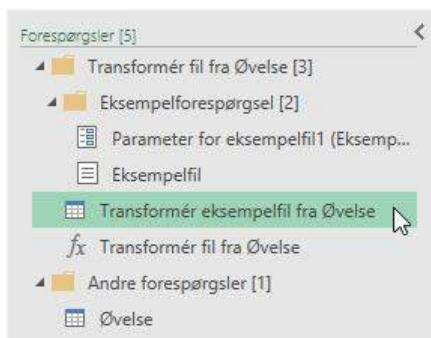
Feltet øverst giver os mulighed for at søge efter enhver tekststreng. Dernæst er der ikonet for at sortere i alfabetisk eller naturlig rækkefølge. Søgning har ingen indflydelse på kildedataene; det er udelukkende til at finde en kolonne.

## Opret mapper til gruppering af forespørgsler

Jo flere forespørgsler vi har, des mere rodet bliver panelet Projektmappe-forespørgsler og forespørgselslisten.

Eksemplerne, vi har brugt i denne bog, har været relativt ligetil med nogle få forespørgsler. Men i den virkelige verden er der sandsynligvis meget mere.

Det er en fremragende idé at organisere forespørgsler ved at placere dem i grupper. Det gør Power Query, når der kombineres flere filer.



Hvordan du beslutter at organisere grupperne er op til dig. Men parametre og brugerdefinerede funktioner er ideelle kandidater til at lægge i deres egen mappe. Når du arbejder videre med Power Query, kan du muligvis begynde at oprette separate forespørgsler til udpakning, transformering og indlæsning af data; disse er også nyttige at organisere i grupper.

Mapper kan oprettes i panelet Projektmappe-forespørgsler eller i listen over forespørgsler.

### Forespørgselsliste (i Power Query editoren)

Hvis du bruger listen over forespørgsler i Power Query editoren, skal du **højre klikke** på listen og vælge **Ny gruppe**. Giv gruppen et navn, og klik derefter på **OK**. En ny mappe oprettes.



Træk forespørgsler ind i mapperne efter behov.

### Panelet Projektmappe-forespørgsler (i Excel)

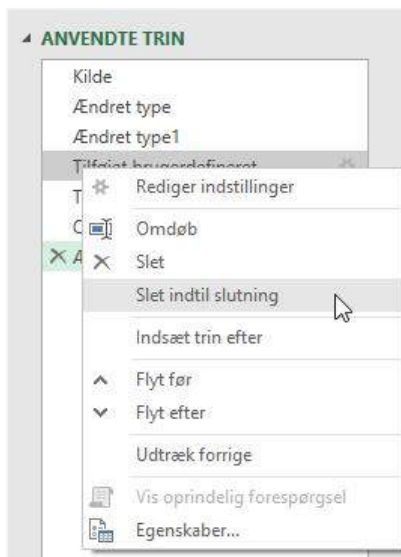
Processen til oprettelse af mapper i panelet Projektmappe-forespørgsler er lidt anderledes. **Højre klik** på listen og vælg **Flyt til gruppe** for at se mulighederne for at oprette mapper eller flytte forespørgsler.

## Slet trin indtil slutningen

At opbygge effektive forespørgsler kræver øvelse. Selv da kan vi når vi følger en bestemt vej, opdage, at det valg, vi tog for 15 trin siden, ikke var den bedste beslutning. Derfor er alle de efterfølgende trin bygget på et mindre optimalt valg.

Vi kunne slette hvert trin én for én for at komme tilbage til punktet før beslutningen; dette vil kræve en masse klik. I stedet kan vi opnå det med bare to klik!

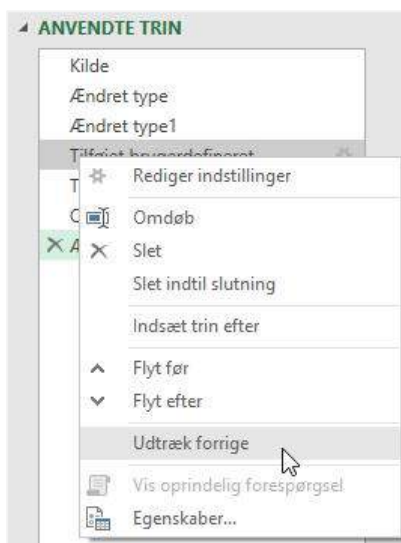
I panelet Anvendte trin højre klikkes på det første trin, vi ønsker at slette, og vælg **Slet indtil slutning**. Alle de efterfølgende trin fjernes.



## Opdel en forespørgsel i to

Undertiden er den første del af en forespørgsel ideel som de første trin i en anden forespørgsel. Du kan være fristet til at opbygge en anden forespørgsel, der indeholder de samme trin ... men det er en dårlig idé. I stedet er det bedre at opdele forespørgslen i to dele og derefter bruge den første del som kilde til begge dele.

**Højre klik** på trinnet, hvor du vil opdele forespørgslen, og klik derefter på **Udtræk forrige** (et lidt mærkeligt navn efter min mening).



Giv forespørgslen et nyt navn, og klik på **OK**. Du har nu to forespørgsler:

1. Den første del af den indledende forespørgsel.
2. Den anden del af den indledende forespørgsel, der bruger (1) som kilde.

Den første del af den indledende forespørgsel kan nu bruges som kilde til andre.

## Kopier og indsæt forespørgsler i en ny projektmappe

Hvis du har oprettet en smart forespørgsel, hvorfor kun bruge den en gang? Det kan let kopiere og indsætte mellem projektmapper.

I Excels panel **Projektmappe-forespørgsler** skal du **højre klikke** på forespørgslen og vælge **Kopier**. I målprojektmappen skal du **højre klikke** i panelet **Projektmappe-forespørgsler** og vælge **Indsæt**. Det er let.

Dette indsætter selve forespørgslen sammen med andre objekter, der henvises til. Naturligvis, hvis der er datakilder, der bruges inden for kildeprojektmappen, bliver disse nødt til at blive udnævnt til data i målprojektmappen.

## Brug af kommentarer

På samme måde som at omdøbe trin, er kommentarer en af de ting, som vi tror, vi vil komme til på et senere tidspunkt ... men det bliver aldrig til noget. I løbet af få dage har vi glemt, hvorfor vi har taget visse beslutninger, eller hvilke specifikke kodelinjer der er beregnet til at gøre hvad. Når dette sker, spilder vi en masse tid på at prøve at forstå vores tidligere beslutningsproces. Dette er grunden til, at kommentarer er så vigtige.

### Indsættelse af kommentarer i den Avancerede editor / Formellinjen

En enkeltlinjekommentar i den avancerede editor er betegnet med to skråstreg (//)

```
// Dette er en enkelt linje kommentar
```

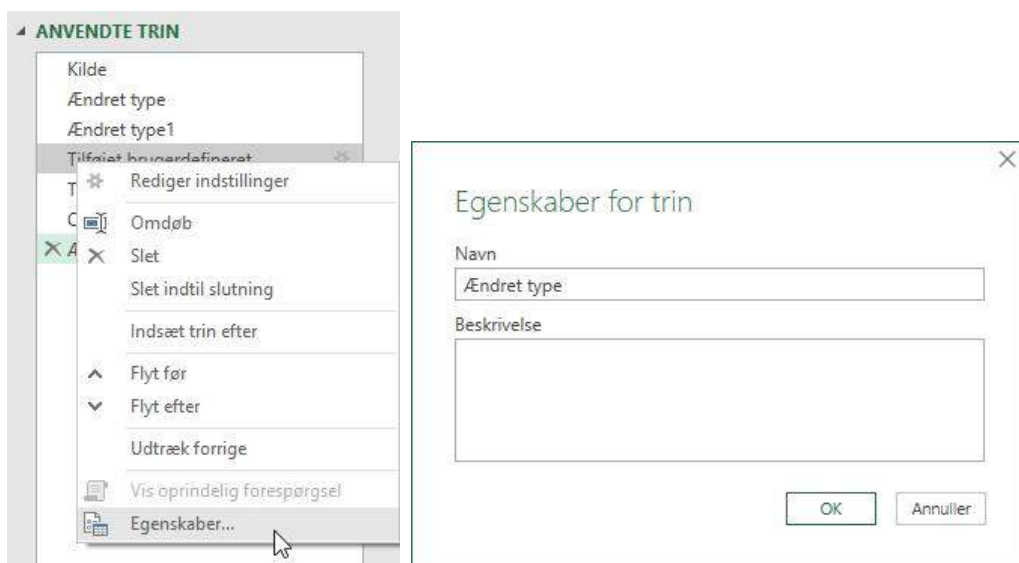
Kommentarer på flere linjer involverer en blanding af skråstreger og stjerner som vist nedenfor:

```
/ * Dette er en kommentar på flere linjer: linje 1
linje 2
linje 3 * /
```

Kommentarer indtastet ved hjælp af disse metoder er kun synlige i den avanceret editor; de vises ikke i formellinjen. Kommentarer kan tilføjes i formellinjen, men når et andet trin vælges, er disse kommentarer ikke længere synlige.

### Indsættelse af kommentarer i trinegenskaberne

Hvis vi **højreklikker** på et hvilket som helst trin, er der muligheden **Egenskaber ...** i menuen. Klik på det for at åbne dialogboksen Egenskaber for trin, og skriv derefter kommentaren i beskrivelsesfeltet. Klik til sidst på **OK**.



Enhver tekst, der indtastes i boksen Beskrivelse, er inkluderet i den Avanceret editor som en enkelt linjekommentar.

## **Tak for denne gang**

Det var hvad et kunne blive til i denne Power Query-bog. Hvis du har fulgt med, håber jeg, at du nu føler dig i stand til at lege med Power Query. Når du har brugt Power Query i lidt tid, anbefaler jeg, at du vender tilbage og arbejder dig igennem bogen igen. Det er utroligt så mange ting, der giver mere mening anden gang.

Tak for at du læste og fulgte med.

