

Microsoft-PowerBI-kurser-1

Indhold

Microsoft-PBI-kurser-1.....	1
Oversigt over dataanalyser	6
Beskrivende analyser	7
Diagnosticerende analyser	7
Forudsigende analyser.....	7
Præskriptive analyser	8
Kognitive analyser.....	8
Eksempel.....	8
Roles	8
Align flere ens visuals lodret og vandret:	12
Install new visual	13
**** Kom i gang med at bygge vha. Power BI	13
Brug Power BI	13
Komponenterne i Power BI.....	15
Opret et dashboard: kortfattet udtræk fra én/flere rapporter	16
Præsentation og brug af Power BI-tjenesten	17
Distribuer indhold vha en App.....	19
Apps er den ideelle delingsløsning i enhver organisation.	21
Videnkontrol	23
**** Føj målinger(mesaures) til Power BI Desktop-modeller	26
enkelt-stående, som udregner aggregeret værdi, som sum(feltværdi), ses som lommeregner	26
kolonne-dannende : ses med lille tabel med sigma i ikonen.....	26
tabel-dannende, icon viser tabel med lommeregner	26
Implicitte målinger er automatiske funktionsmåder, der understøttes af visualiseringer, fx aggregeringer.....	26
eksplicitte målinger bruger DAX-formler, der opsummerer modeldata.	26
Numeriske kolonner understøtter den største række sammenlægnings, aggregerings-funktioner:	32
Opsummer ikke-numeriske kolonner	32
Opret simple målinger(measures).....	33
Sammenlign beregnede kolonner med målinger	37
De områder, hvor beregnede kolonner og målinger er forskellige, omfatter:.....	37
*** Beskriv Power BI Desktop-modeller	39

Faktatabeller	40
Dimensionstabeller	41
Analyseforespørgsler	43
Konfigurer rapportvisuals	44
*** Choose a Power BI model framework	48
Introduction	49
Describe Power BI model fundamentals	50
Determine when to develop an import model, alternativt Direct Query	54
Import model benefits	54
Import model limitations: max 10GB, opdateres få gange dagligt	55
Data refresh: Imported data must be periodically refreshed, kan kun ske få gange dagligt	56
Determine when to develop a DirectQuery model	56
DirectQuery model benefits	57
Model large or fast-changing data sources(data er altid opdateret)	57
DirectQuery model limitations	58
Boost DirectQuery model performance	58
DirectQuery user-defined aggregation tables	59
Determine when to develop a composite model	59
Composite model benefits	59
Composite model limitations	60
Choose a model framework	62
Choose the composite model framework to:	62
Knowledge check	63
**** Forbered, importer data til analyse med Power BI	64
Hent data fra filer	67
Hent data fra relationsdatakilder,SQL-DB	71
Opret dynamiske rapporter med parametre	79
Hent data fra en NoSQL-database	90
Importér en JSON-fil	91
Hent data fra onlinetjenester	94
Vælg en lagertilstand	95
Hent data fra Azure Analysis Services	98
Løs problemer med ydeevnen	99
Næste undermodul: Løs fejl under import af data	103

Øvelse – Forbered data i Power BI Desktop	107
Column Quality, lav check ved dataindlæsning	108
Get data from a CSV file.....	113
videnstest	115
**** Design en semantisk model i Power BI	117
Introduktion.....	119
Arbejd med tabeller.....	123
Konfigurer egenskaber for tabeller og kolonner.....	125
Oprette, generer en fælles dato-tabel.....	131
M: først lave liste med datoer, dernæst konerter listen til tabel	134
Markeres som den officielle datotabel.....	139
Create relationship med nye date-tabel	141
Arbejd med dimensios-tabeller, valg af hierarkier	142
Overordnet-underordnet-hierarki, lave rolle-hierarki fra Employee tabel	146
Dimensioner med forskellige roller	148
Definer datagranularitet	149
Arbejde med relationer og alitet	156
Mange til en- eller (*:1) en til mange-relation (1: *) er det almindelige, sunde	156
Filtrering følger pileretninger	157
En til en-relation (1:1), anbefales ikke: saml data i én tabel.....	157
Mange til mange-relation (.): anbefales ikke, kan nemt give inkonsistente data	157
Løs modelleringsudfordringer	160
Lab-øvelser, Setup local lab environment.....	161
Matrix-visual kan fint vies værdier efter hierarkier, hurtigste måde	173
Test.....	189
**** Optimer en model i forhold til ydeevne i Power BI	191
Introduktion til optimering af ydeevnen	192
Mindske modelstørrelsen og andre modelforbedringer:	193
Gennemse performance for målinger, relationer og grafik,	195
Identificere flaskehalse for rapportydeevne.....	195
Performance-analyse-værktøj	196
Løse problemer og optimere performance	198
Visuals.....	199
Performanceanalyse	199

DAX-forespørgsel	199
Semantisk model	200
Metadata	201
Automatisk dato/klokkeslætsfunktion	203
Brug variabler til at forbedre ydeevnen og fejlfinding.....	204
Brug variabler til at foretage fejlfinding af flere trin.....	206
Reducer kardinalitet	207
design med en blandet model:.....	209
Optimere DirectQuery-modeller med lagring på tabelniveau.....	210
Opret og administrer data-sammenlægninger, til grovere granularitet	215
Du kan oprette sammenlægninger på forskellige måder, og hver metode giver dig de samme resultater, f.eks.:.....	216
Test din viden.....	219
Resumé	220
**** Skriv DAX-formler til Power BI Desktop-modeller	222
Introduktion.....	223
What If-analyse.....	226
Skriv DAX-formler	228
A single value for column 'WoW' in table 'Table Current' cannot be determined.	228
Tryk på Skift+Enter for at angive et return på formellinjen	231
Hvis du trykker på Enter alene, bekræftes formelen.....	231
DAX-datatyper	233
Arbejd med DAX-funktioner	234
Aritmetiske operatorer	236
Brug DAX-variabler.....	240
Sammenknyt to tabeller i DAX-kode, med Related()	241
Tjek din viden.....	242
**** Føj beregnede tabeller og kolonner til Power BI Desktop-modeller	244
Opret kopi af Date, til ny Ship-Date., så der kan filtreres efter ship-dates i Sales med relation	247
Opret en datotabel	249
Merge to tabeller med Union().....	250
Opret beregnede kolonner	252
Detaljerede Dato-kolonner oprettes.....	253
Få mere at vide om rækkekontekst	255
Vælg en teknik til tilføjelse af en kolonne.....	256

Tjek din viden.....	257
**** Brug DAX-time intelligence-funktioner i Power BI Desktop-modeller	259
Brug DAX-time intelligence-funktioner	262
Opsummeringer over tid-funktioner, fx Year to Date, YTD	262
Sammenligninger over tidsrum, fx Revenu i måned med samme måned året før.....	264
Yderligere time intelligence-beregninger, antal forekomster i tidsrum.....	266
Snapshotberegninger	269
Øvelse – Create avancerede DAX-beregninger i Power BI Desktop.....	273
In this task, you'll create a matrix visual to support testing your new measures.	274
The CALCULATE() function	276
Create a YTD measure	281
Videns-check.....	284
**** Enforce security in Power BI models	286
Restrict access to Power BI model data.....	288
RLS=Row-Level Security: controlling access to data by row	288
Static rules	290
If there are many rules to set up and you anticipate adding new rules in the future, consider creating dynamic rules instead.....	291
Dynamic rules	291
Restrict access to Power BI model objects	293
Object-level security (OLS) can restrict access to specific tables and columns, and metadata.....	294
A perspective defines viewable subsets of model objects som views in DB	294
OLS is a feature inherited from Azure Analysis Services (AAS) and SQL Server Analysis Services (SSAS) .	294
Apply good modeling practices	296
Exercise: Enforce model security	296
-Create static roles.....	301
User ser kun de aktivt tilvalgte filter-værdier	303
Validate the static roles	303
In the View as roles window, select the Australia role.	304
Publish the reports til WebService-delen	304
Configure row-level security (<i>optional</i>) -for assets in Web-service delen.....	305
In a real-world solution, each role should be mapped to a security group.....	307
In this task, you will clean up the solution by removing the dataset and the model roles.	307
-Create a dynamic role.....	309
Add the Salesperson table	309

Configure the relationship for nye SalesPerson table	313
The purpose of the Salesperson table is to enforce data permissions. When hidden, report authors and the Q&A experience won't see the table or its fields.....	314
Create a dynamic role.....	314
Check your knowledge	319
Summary.....	320

<https://learn.microsoft.com/da-dk/training/browse/?products=power-bi%2Cpower-platform&subjects=data-management&levels=intermediate&roles=data-analyst>

Oversigt over dataanalyser

<https://learn.microsoft.com/da-dk/training/modules/data-analytics-microsoft/3-roles>

Fuldført 100 XP

- 6 minutter

Før data kan bruges til at fortælle en historie, skal de igennem en proces, der gør det muligt at bruge dem i fortællingen. Dataanalyseprocessen består i at identificere, rense, transformere og modellere data, så du kan finde meningsfulde og nyttige oplysninger. Dataene bruges derefter til at skabe en fortælling via rapporter til de analyser, der ligger til grund for vigtige beslutningsprocesser.



I takt med at verden bliver mere datadreven, bliver historiefortælling gennem dataanalyser en vigtig komponent for både store og små virksomheder. Det er årsagen til, at organisationer fortsat ansætter dataanalytikere.

Datadrevne virksomheder træffer beslutninger, der er baseret på den historie deres data fortæller, og i dagens datadrevne verden bliver data ikke udnyttet fuldt ud, hvilket er en udfordring for de fleste virksomheder. Dataanalyser er og bør være en vigtig del for alle organisationer, så de kan fastsætte, hvad

der har en indvirkning på deres virksomhed, og blandt andet evaluere kundesynspunkter, udføre markeds- og produktundersøgelser og identificere tendenser eller anden dataindsigt.

Mens dataanalyseprocessen fokuserer på at rydde op i, udforme og visualisere data, bør begrebet dataanalyse, og hvor vigtigt det er for virksomheden, ikke undervurderes. Kernekomponenterne i analysen er opdelt i følgende kategorier, så du kan analysere data:

- Beskrivende
- Diagnosticerende
- Forudsigende
- Præskriptive
- Kognitive

Beskrivende analyser

Beskrivende analyser hjælper med at besvare spørgsmål om, hvad der er sket, ud fra historiske data. Beskrivende analyseteknikker opsummerer store semantiske modeller for at beskrive resultater for interessenter.

Gennem udviklingen af KPI'er (nøgletalsindikatorer) kan disse strategier hjælpe med at spore, om vigtige målsætninger opfyldes eller ej. Målepunkter, f.eks. afkast (ROI), bruges i mange brancher, og særlige målepunkter er udviklet til at spore ydeevnen i bestemte brancher.

Et eksempel på beskrivende analyser omfatter oprettelsen af rapporter for at danne sig et overblik over en organisations salgs- og økonomidata.

Diagnosticerende analyser

Diagnosticerende analyser hjælper med at besvare spørgsmål om, **hvorfor bestemte hændelser fandt sted**. Teknikker til diagnosticerende analyser supplerer de grundlæggende beskrivende analyser, og de bruger resultaterne fra beskrivende analyser til at finde frem til årsagerne til disse hændelser. Herefter undersøges disse nøgletal yderligere for at finde ud af, hvorfor de er blevet bedre eller værre. Dette sker normalt i tre trin:

1. Identificer uregelmæssigheder i dataene. Disse uregelmæssigheder kan være uventede ændringer i et målepunkt eller på et bestemt marked.
2. Indsaml data, der er relateret til disse uregelmæssigheder.
3. Brug statistiske teknikker til at finde relationer og tendenser, der forklarer disse afvigelser.

Forudsigende analyser

Forudsigende analyser hjælper med at besvare spørgsmål om, hvad der vil ske i fremtiden. Forudsigende analyseteknikker bruger historiske data til at identificere tendenser og finde ud af, om de måske vil gentage sig. Forudsigende analyseværktøjer giver værdifuld indsigt i, hvad der kan ske fremover. Teknikkerne omfatter en række statistiske teknikker og teknikker til maskinel indlæring, f.eks. neurale netværk, beslutningstræer og regression.

Præskriptive analyser

Præskriptive analyser hjælper med at besvare spørgsmål om, **hvilke tiltag der er nødvendige for at nå et mål**. Ved hjælp af indsigt fra præskriptive analyser kan organisationer træffe datadrevne beslutninger. Denne teknik giver virksomheder mulighed for at træffe velfunderede beslutninger i usikre tider. Præskriptive analyseteknikker er afhængige af maskinel indlæring som en af strategierne til at finde mønstre i store semantiske modeller. Organisationer kan ved at analysere tidligere beslutninger og hændelser estimere sandsynligheden for forskellige resultater.

Kognitive analyser

Kognitive analyser **forsøger at udtrække fornuftslutninger fra eksisterende data og mønstre**, drage konklusioner på baggrund af eksisterende vidensbaser og derefter føje disse resultater til vidensbasen til fremtidige fornuftslutninger – en form for et selvlærende feedbackloop. Kognitive analyser hjælper dig med at få mere at vide om, hvad der kan ske, hvis omstændigheder ændrer sig, og bestemme, hvordan du kan håndtere disse situationer.

Udledninger er ikke strukturerede forespørgsler, der er baseret på en regeldatabase. De er snarere ustrukturerede hypoteser, der indsamles fra flere kilder og med en varierende grad af pålidelighed. Effektive kognitive analyser afhænger af algoritmerne til maskinel indlæring og gør brug af flere koncepter til behandling af naturligt sprog for at gøre hidtil uudnyttede datakilder forståelige, f.eks. logge fra call center-samtaler og produktanmeldelser.

Eksempel

En detailforretning bruger beskrivende analyser til at aktivere rapportering og datavisualiseringer og identificere mønstre fra de foregående år for at afgøre, hvilke produkter der muligvis bliver populære næste år. Virksomheden kan også se på understøttende data for at forstå, hvorfor et bestemt produkt var populært, og om den pågældende tendens fortsætter. Det kan hjælpe dem med at afgøre, om de fortsat skal lagere det pågældende produkt.

En virksomhed kan fastslå, om et bestemt produkt var populært i en bestemt **Forretningsanalytiker**

Selvom der er visse lighedspunkter mellem en dataanalytiker og en forretningsanalytiker, er den største forskel på disse to roller det, de foretager sig med dataene. En forretningsanalytiker er tættere på selve virksomheden og er specialist i at fortolke de data, der kommer fra visualiseringen. Rollen som dataanalytiker og forretningsanalytiker kan ofte varetages af en og samme person.

Roles

Dataanalytiker

En dataanalytiker gør det muligt for virksomheder at maksimere værdien af deres dataaktiver via visualiserings- og rapporteringsværktøjer som f.eks. Microsoft Power BI. Dataanalytikere er **ansvarlige for profilering, rensning og transformering af data. Deres ansvar omfatter også design og opbygning af skalerbare og effektive semantiske modeller og aktivering og implementering af avancerede analysefunktioner i rapporter til analyse. En dataanalytiker samarbejder med de pågældende interessenter om at identificere relevante og nødvendige data- og rapporteringskrav og får derefter til opgave at omdanne rådata til relevant og meningsfuld indsigt.**

En dataanalytiker er også ansvarlig for administrationen af Power BI-aktiver, herunder rapporter, dashboards, arbejdsområder og de underliggende semantiske modeller, der bruges i rapporterne. De varetager implementeringen og konfigurationen af de korrekte sikkerhedsprocedurer sammen med kravene fra interessenterne for at sikre, at alle Power BI-aktiver og de tilhørende data beskyttes.

Dataanalytikere arbejder sammen med datateknikere om at fastslå og finde relevante datakilder, der opfylder kravene fra interessenterne. Derudover samarbejder dataanalytikere med datateknikeren og databaseadministratoren om at sikre, at analytikeren har korrekt adgang til de nødvendige datakilder. Dataanalytikeren arbejder også sammen med datateknikeren for at identificere nye processer eller forbedre eksisterende processer til indsamling af data til analyse.

Datatekniker

Datateknikere klargør og konfigurerer dataplatformsteknologier lokalt og i cloudmiljøet. De administrerer og sikrer flowet af strukturerede og ustrukturerede data fra flere kilder. De dataplatforme, de bruger, kan omfatte relationsdatabaser, ikke-relationelle databaser, datastreams og fillagre. Datateknikere sikrer også, at datatjenester integreres sikkert og problemfrit på tværs af dataplatforme.

En datateknikers primære ansvar omfatter brug af datatjenester lokalt og i skyen samt brug af værktøjer til at overføre, udsende og transformere data fra flere kilder. Datateknikere samarbejder med forretningsinteressenter om at identificere og overholde datakrav. De designer og implementerer løsninger.

Selvom der kan være sammenfald mellem opgaverne og ansvarsområderne for en datatekniker og en databaseadministrator, beskæftiger en datatekniker sig med andet end blot at overvåge en database og den server, hvor databasen hostes, men varetager sandsynligvis ikke den overordnede administration af driftsdata.

En datatekniker tilfører en masse værdi til både business intelligence-projekter og datavidenskabelige projekter. Når datateknikeren indsamler data, hvilket også betegnes som dataændring, kører projekterne hurtigere, fordi dataspecialister kan fokusere på deres egne arbejdsområder.

Som dataanalytiker skal du arbejde tæt sammen med en datatekniker for at sikre, at du kan få adgang til de mange forskellige strukturerede og ustrukturerede datakilder, fordi de understøtter dig i at optimere semantiske modeller, som typisk betjenes fra et moderne data warehouse eller data lake.

Både databaseadministratorer og business intelligence-medarbejdere kan fungere som datatekniker. De skal blot uddannes i de værktøjer og den teknologi, der bruges til at behandle store mængder data.

Dataspecialist

Dataspecialister udfører avancerede analyser for at trække værdier ud af data. Deres arbejde kan variere fra beskrivende til præsriptive analyser. En beskrivende analyse evaluerer data gennem en proces, der kaldes EDA (Exploratory Data Analysis (undersøgende dataanalyse)). Præsriptive analyser bruges i maskinel indlæring til at anvende modelleringsteknikker, der kan registrere afvigelser eller mønstre. Disse analyser er en vigtig del af budgetmodeller.

Beskrivende og præsriptive analyser er blot en del af dataspecialisternes arbejde. Nogle dataspecialister arbejder også med dyb læring, hvor de udfører iterative eksperimenter for at løse et komplekst dataproblem ved hjælp af tilpassede algoritmer.

I anekdotisk dokumentation fremgår det, at det meste af arbejdet i et datavidenskabeligt projekt bruges på dataændring og engineering af funktioner. Dataspecialister kan gøre eksperimentprocessen hurtigere, når datateknikerne udnytter deres færdigheder til at ændre data.

På overfladen kan det se ud som om, at en dataspecialists og en dataanalytikers arbejdsopgaver er meget forskellige, men det er ikke tilfældet. Et dataspecialist bruger dataene til at få svar på spørgsmål og udarbejder ofte en hypotese eller et eksperiment og henvender sig til dataanalytikeren for at hjælp med datavisualiseringen og -rapporteringen.

Databaseadministrator

En databaseadministrator implementerer og administrerer de driftsmæssige aspekter ved cloudbaserede og hybride dataplatformsløsninger, som bygger på Microsoft Azure-datatjenester og Microsoft SQL Server. En databaseadministrator er ansvarlig for den overordnede tilgængelighed og ensartede ydeevne og optimeringer af databaseløsningerne. De arbejder sammen med interessenter for at identificere og implementere politikker, værktøjer og processer til sikkerhedskopiering af data og planer for gendannelse af data.

En databaseadministrators rolle er forskellig fra en datateknikers rolle. En databaseadministrator overvåger og administrerer den overordnede tilstand for en database og den tilhørende hardware, mens en datatekniker er involveret i processen med at ændre dataene, dvs. indtagelse, transformation, validering og rensning af data for at imødekomme forretningsmæssige behov og krav.

Databaseadministratoren er også ansvarlig for at administrere dataenes samlede sikkerhed, tildele og begrænse brugeradgang og -rettigheder til dataene i henhold til forretningsmæssige behov og krav.

Næste undermodul: En dataanalytikers opgaver

[Tidligere](#)

Næste undermodul: Roller i data

[Tidligere](#)

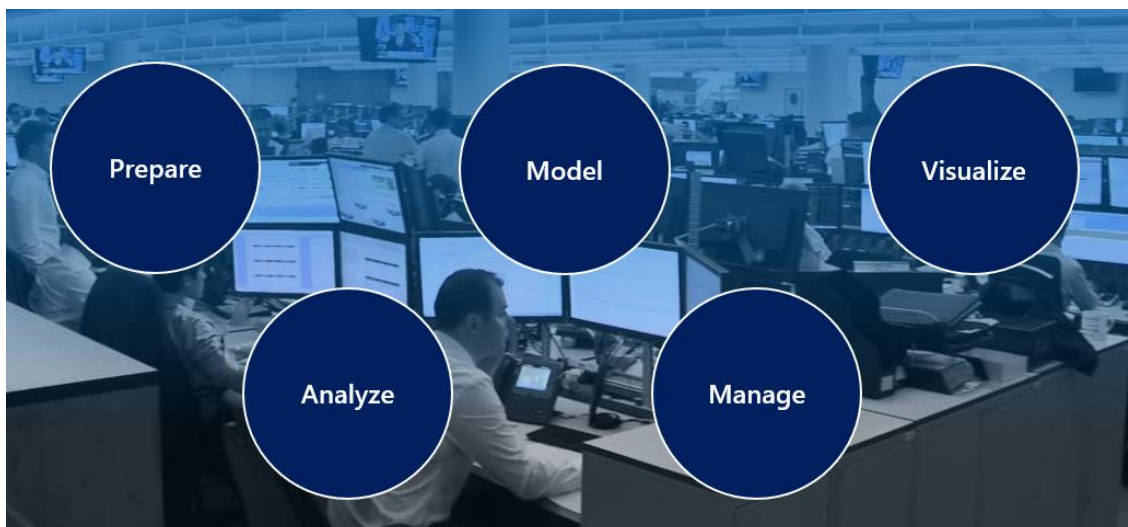
En dataanalytikers opgaver

Fuldført 100 XP

- 10 minutter

En dataanalytiker er en af flere vigtige roller i en organisation, der hjælper med at afdække og gøre oplysninger forståelige for at holde virksomheden afbalanceret og driftseffektiv. Det er derfor af altafgørende betydning, at du som dataanalytiker forstår dit ansvar og de opgaver, der udføres næsten dagligt. Dataanalytikere er afgørende for at hjælpe organisationer med at få værdifuld indsigt i de enorme mængder data, som de har, og de arbejder tæt sammen med andre i organisationen for at hjælpe med at afdække værdifulde oplysninger.

Følgende figur viser de fem nøgleområder, som du beskæftiger dig med under dataanalyseprocessen.



1.

Hvilken datarolle muliggør avancerede analysefunktioner specifikt via rapporter og visualiseringer?

Dataspecialist

En dataspecialist klarlægger de spørgsmål, der har brug for svar, og henvender sig til dataanalytikeren for at få hjælp med datavisualiseringen og -rapporteringen.

Datatekniker

Dataanalytiker

En dataanalytiker bruger passende visualiseringer til at hjælpe beslutningstagerne med at få omfattende og meningsfuld indsigt fra data.

2.

Hvilken dataanalytiker-opgave har en kritisk effektivitets påvirkning på rapportering og dataanalyse?

Modellering

En optimeret og justeret semantisk model fungerer bedre og giver en bedre dataanalyseoplevelse.

Analyse

Visualisere

3.

Hvilken af følgende muligheder er den vigtigste vigtige fordel ved dataanalyse?

Afgørende analyser

Velfunderede forretningsbeslutninger

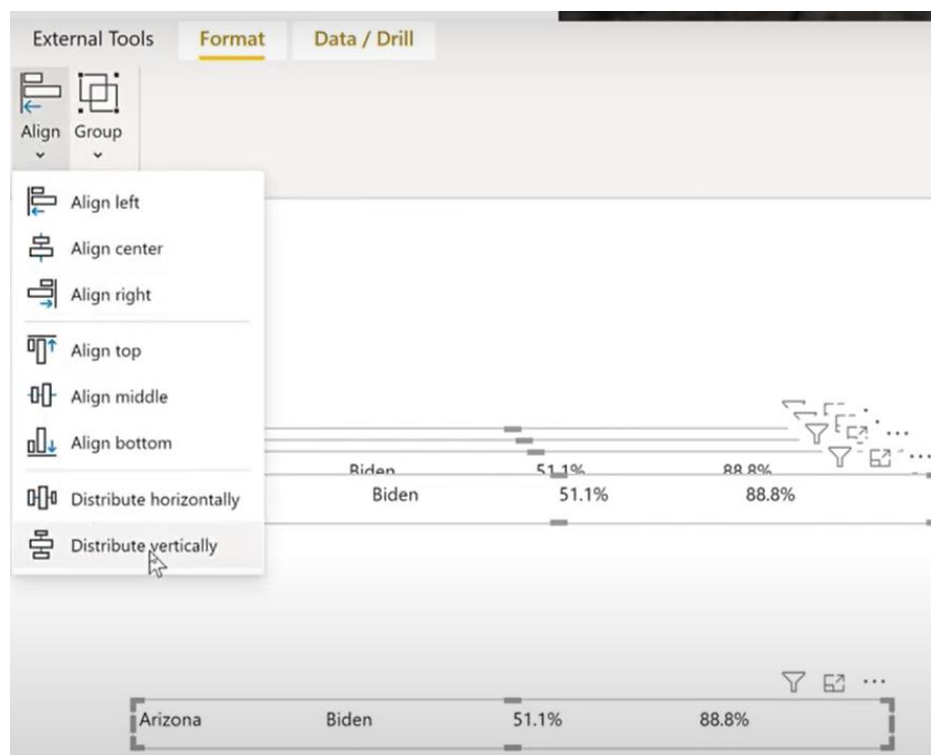
En af de vigtigste fordele ved dataanalyse er muligheden for at få værdifuld viden fra en virksomheds dataaktiver, så du kan foretage de rette og optimale forretningsbeslutninger.

Komplekse rapporter

1. [Learn](#)
2. [Træning](#)
3. [Gennemse](#)

Align flere ens visuals lodret og vandret:

Marker flere med <ctrl> klik,, så format



Install new visual

Download som .pbviz -fil, så --- import denne fil

**** [Kom i gang med at bygge vha. Power BI](#)

Introduktion

<https://learn.microsoft.com/da-dk/training/modules/get-started-with-power-bi/1-introduction>

Fuldført 100 XP

- 1 minut

Microsoft Power BI er en komplet rapporteringsløsning, der tilbyder dataforberedelse, datavisualisering, distribution og administration via udviklingsværktøjer og en onlineplatform.

Power BI kan skalere fra simple rapporter ved hjælp af en enkelt datakilde til rapporter, der kræver kompleks datamodellering og ensartede temaer. Brug Power BI til at oprette visuelt flotte, interaktive rapporter, der fungerer som analyse- og beslutningsprogrammet bag gruppeprojekter, divisioner eller hele organisationer.

Power BI er et vigtigt værktøj for dataanalytikere og deres organisation. Men alle datamedarbejdere drager fordel af at forstå, hvordan Power BI arbejder med at udforske og præsentere dataindsigt i organisationer.

Næste undermodul: Brug Power BI

[Næste](#)

Brug Power BI

Fuldført 100 XP

- 3 minutter

Hvis du vil oprette rapporter med Power BI, skal du først forstå de nødvendige værktøjer. Power BI indeholder tre primære komponenter:

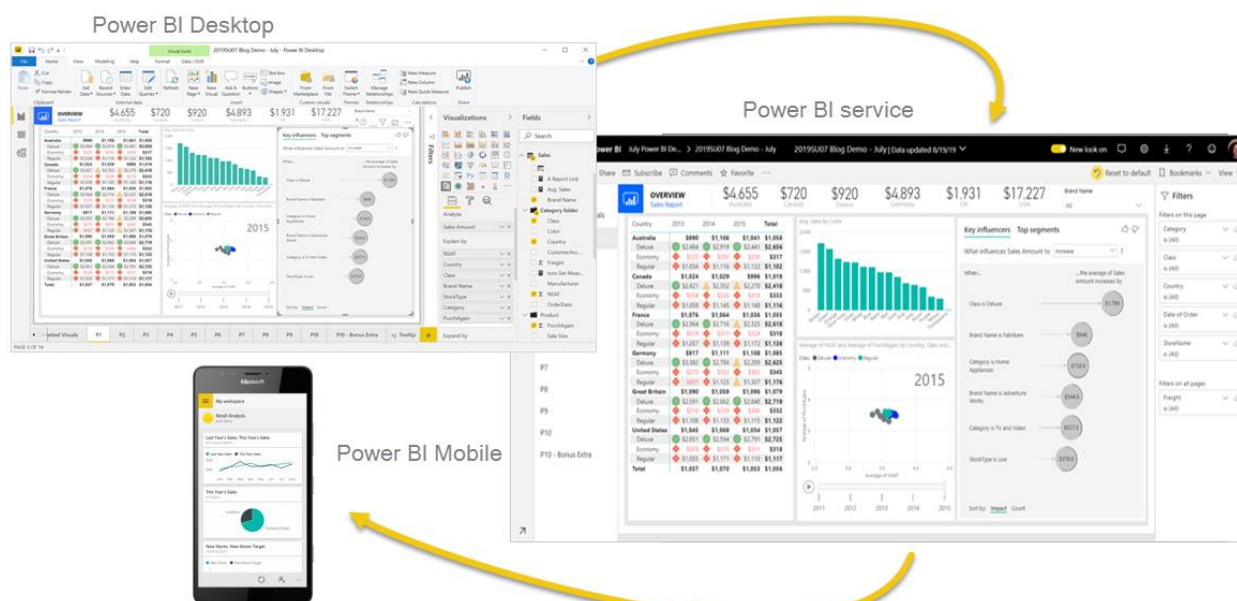
- Power BI Desktop (skrivebordsprogram)
- Power BI-tjeneste (onlineplatform)
- Power BI – Mobil (mobilapp på tværs af platforme)

Power BI Desktop er det udviklingsværktøj, der er tilgængeligt for dataanalytikere og andre rapportoprettere. Mens Power BI-tjeneste giver dig mulighed for at organisere, administrere og distribuere

dine rapporter og andre Power BI-elementer. Power BI Desktop kan downloades gratis enten via Windows Store eller direkte online.

Du kan få adgang til Power BI-tjeneste på app.powerbi.com med en skole- eller arbejdskonto. Hvis din organisation ikke allerede bruger Power BI, kan du stadig udforske tjenesten ved at få en gratis prøveversion eller tilmelde dig en gratis [Microsoft 365 Developer-konto](#).

Power BI – Mobil giver forbrugerne mulighed for at få vist rapporter i et mobiloptimeret format. Du kan oprette disse optimerede rapportvisninger i Power BI Desktop.



Udforsk flowet i Power BI

Der er et almindeligt flow, når du opretter rapporter med Power BI. Først skal du starte med Power BI Desktop for at oprette forbindelse til data og oprette rapporten. Derefter publicerer du rapporten til Power BI-tjeneste og distribuerer den til forbrugere.

Flowet for Power BI er:

1. Opret forbindelse til data med Power BI Desktop.
2. Transformér og modelér data med Power BI Desktop.
3. Create visualiseringer og rapporter med Power BI Desktop.
4. Publicer rapport til Power BI-tjeneste.
5. Distribuer og administrer rapporter i Power BI-tjeneste.

Power BI-tjeneste giver dig også mulighed for at oprette dashboards på højt niveau, der foretager detailudledning til rapporter, og apps, så du nemt kan gruppere relaterede rapporter til brugere i et enkelt format.

Tip

Se flere oplysninger om Power BI i [referencedokumentationen](#).

Næste undermodul:

Komponenterne i Power BI

Komponenterne i Power BI

Fuldført 100 XP

- 8 minutter

Komponenterne i Power BI er **semantiske modeller** og **visualiseringer**. Create en semantisk model og derefter bruge visualiseringer til at oprette en rapport. Lad os udforske disse elementer mere detaljeret, og hvordan de relaterer til flowet i Power BI.

Create en semantisk model

En **semantisk model** består af alle forbundne data, transformationer, relationer og beregninger. Hvis du vil følge flowet i Power BI, skal du først oprette forbindelse til data, transformere data og oprette relationer og beregninger for at oprette en semantisk model.

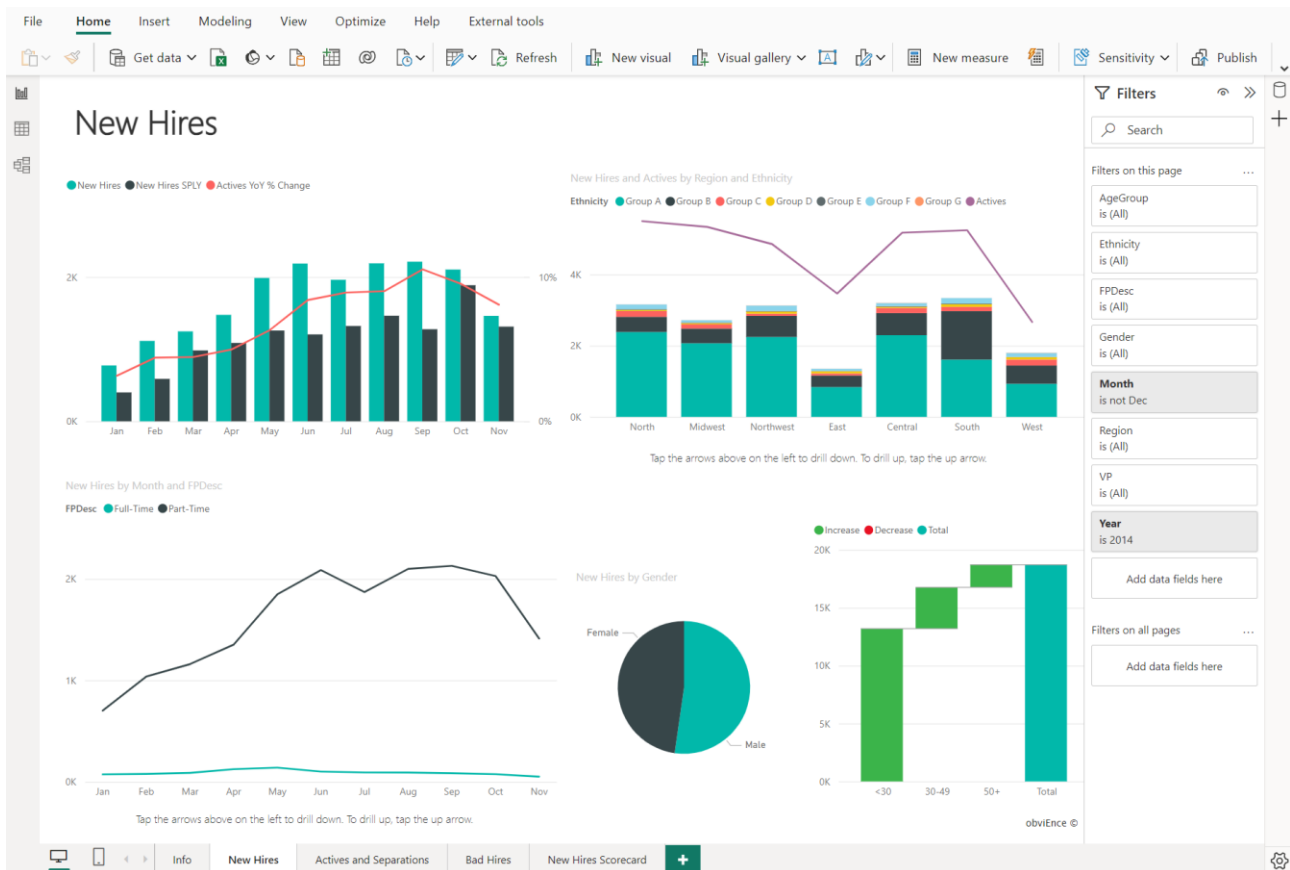
Opret først forbindelse til lige så mange datakilder, du har brug for. Derefter skal du rense og transformere dataene til dine behov. Tilføj relationer mellem tabeller og beregninger for at udvide den semantiske model. Efter alt dette kan du nu oprette en rapport.

Create visualiseringer i en rapport

Når du opretter en **visualisering** (også kaldet *visualisering*) i Power BI Desktop, føjer du den til **lærredet** for en rapportside. Vælg dine visualiseringer for at oprette sider i din rapport. Den er ideel til at holde hver side enkel med relaterede data, så forbrugerne nemt kan se indsigten.

Power BI er en løsning med lav kode, hvilket betyder, at du kan "*trække og slippe*" datafeltet direkte på lærredet. Power BI vælger en visualisering til dit datafelt. Du kan nemt ændre mellem visualiseringer for de samme felter og føje eller fjerne datafelter til visualiseringen.

En af de mest værdifulde funktioner i Power BI-rapporter er interaktiviteten mellem visualiseringer. Forbrugere kan vælge forskellige datapunkter i visualiseringen og se, hvordan det påvirker de andre visualiseringer. Afhængigt af dit design kan de også foretage detaljeudgang fra én visualisering til flere detaljer eller filtrere baseret på forskellige felter i rapporten.

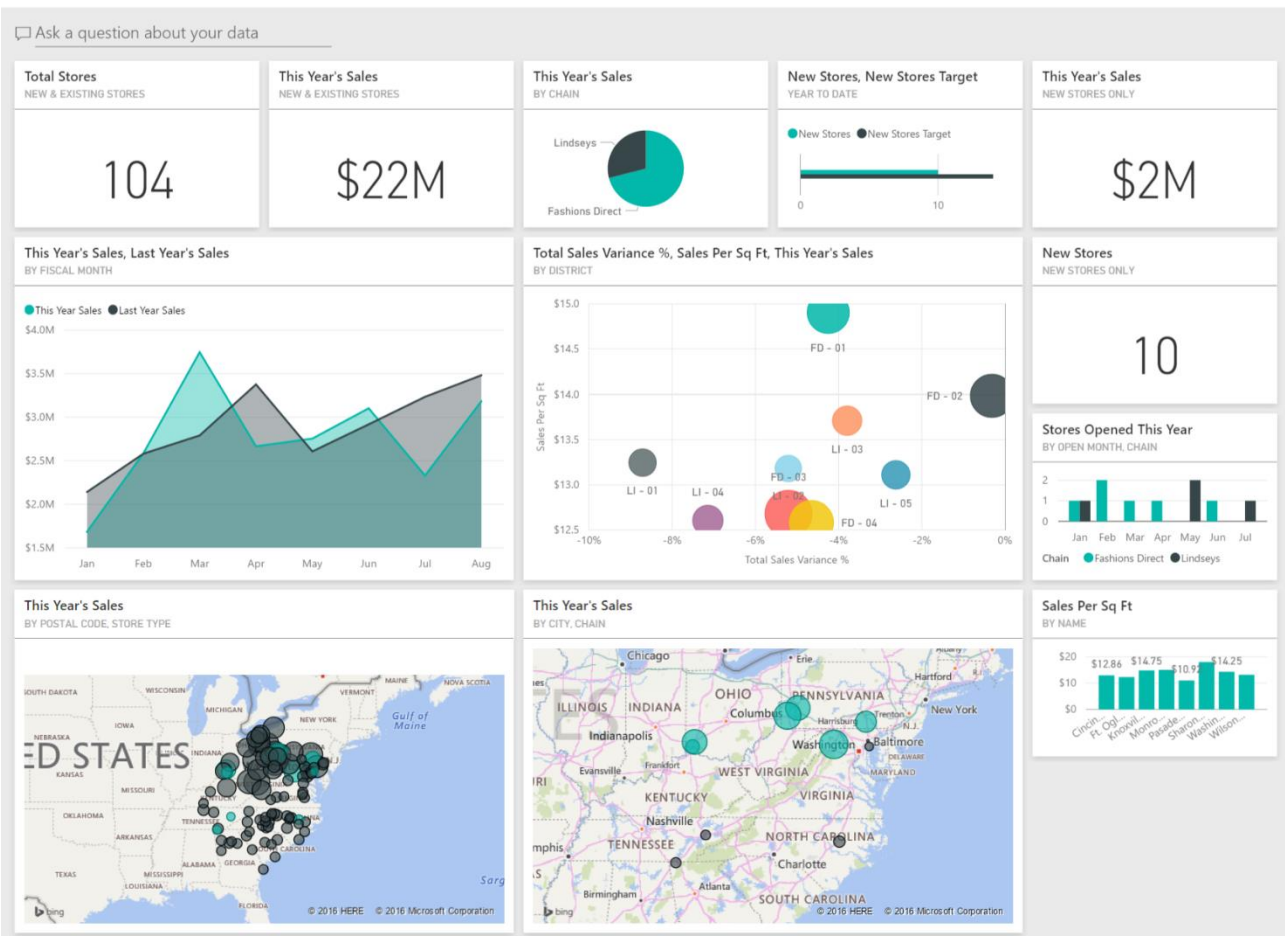


Når du er tilfreds med din rapport, publicerer du den på Power BI-tjeneste.

Opret et dashboard: kortfattet udtræk fra én/flere rapporter

I Power BI-tjeneste kan du også oprette **dashboards**, når du har publiceret en rapport. Dashboards består af en enkelt side, der består af **felter**. Føj felter til et dashboard ved at fastgøre en visualisering i en rapport til dashboardet. Felter er ikke interaktive som visualiseringer, så når en bruger interagerer med feltet, går de til den underliggende rapport for at få flere oplysninger -Så der er vel interaktion alligevel?

Dashboards er en glimrende måde at levere oplysninger på højt niveau til forbrugere. På samme måde som et dashboard i et køretøj skal du inkludere de vigtigste oplysninger i et dashboard. Derefter kan forbrugerne gå til rapporten for at få flere oplysninger.



For at opsummere er byggestenene i Power BI **semantiske modeller** og **visualiseringer**. Ved hjælp af Power BI Desktop kan du oprette den semantiske model og bruge visualiseringer til at oprette rapporter.

I Power BI-tjeneste kan du distribuere indhold til dine forbrugere og bruge rapporter til at oprette dashboards.

Næste undermodul:

Præsentation og brug af Power BI-tjenesten

Tidligere

Præsentation og brug af Power BI-tjenesten

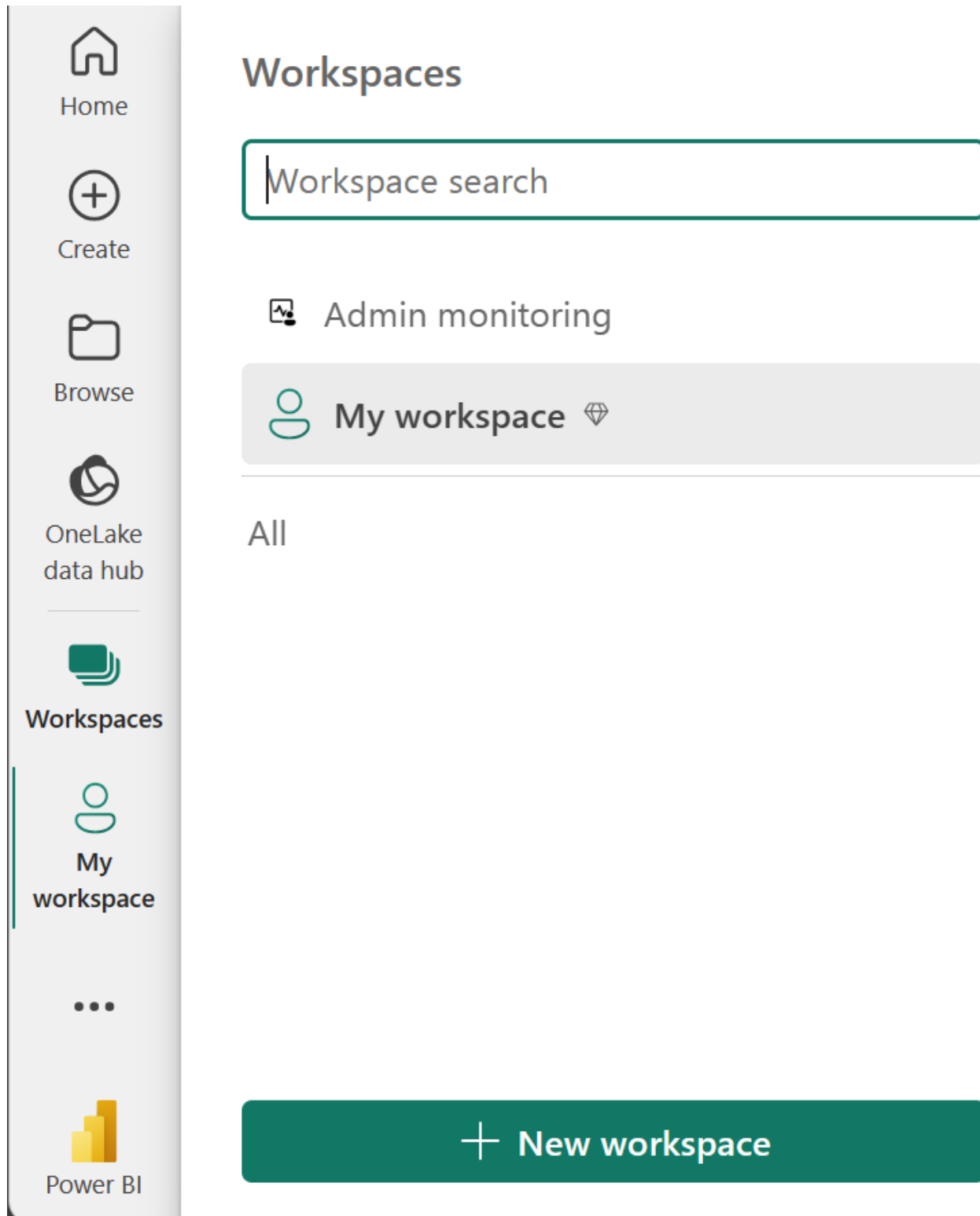
Fuldført 100 XP

- 5 minutter

Nu, hvor du forstår, hvordan du opretter en rapport, kan vi udforske Power BI-tjeneste. Den Power BI-tjeneste giver en enkel og interaktiv brugeroplevelse, hvor du kan tage dine dataanalyser til det næste niveau.

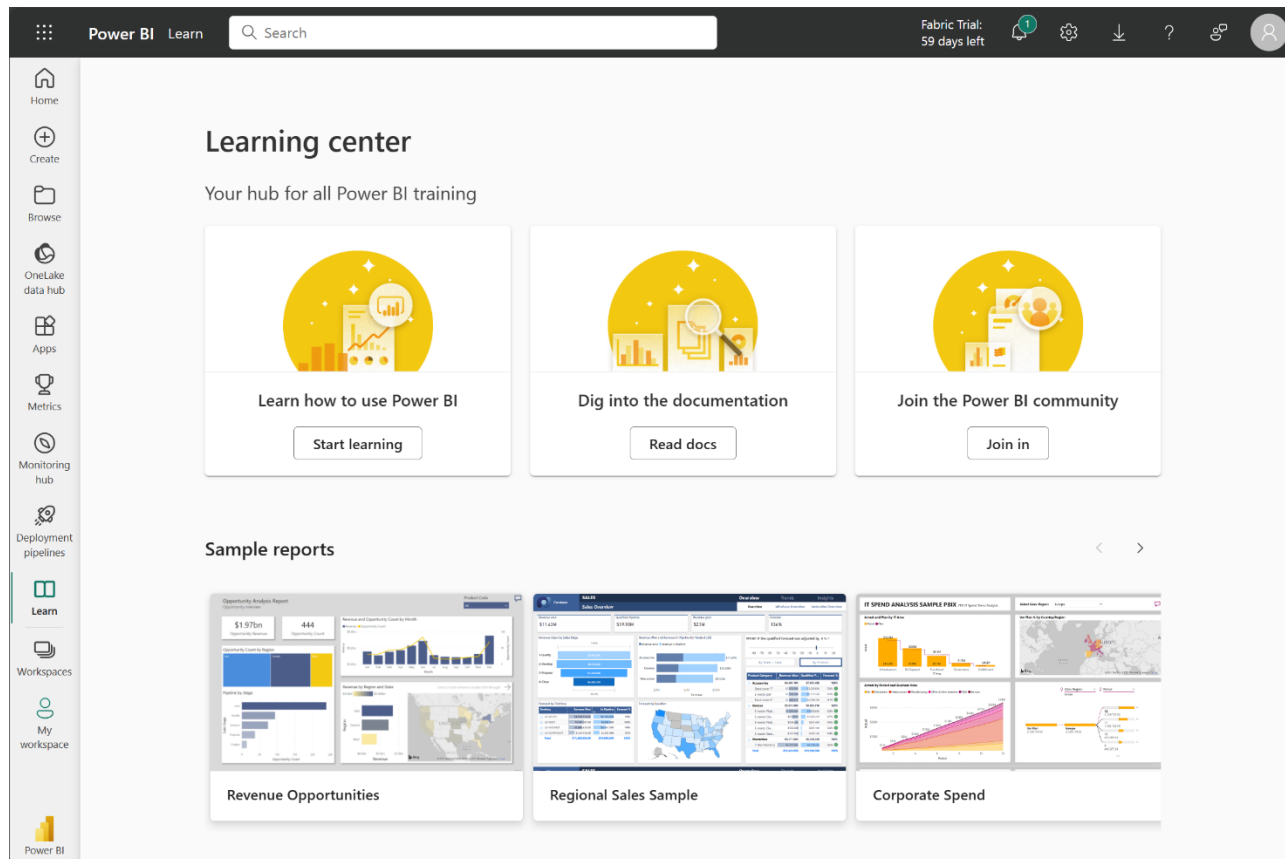
Organiser elementer med arbejdsområder

Arbejdsområder er grundlaget for Power BI-tjeneste. Når du publicerer en rapport, skal du vælge et arbejdsområde. Som standard har alle brugere adgang til brugerens eget *arbejdsområde*, hvilket kun er ideelt til test. Når du vil dele indhold med andre, **skal du altid** oprette og bruge et delt arbejdsområde.



Udforsk eksempel-rapporter

Hvis du endnu ikke har oprettet en rapport, tilbyder Power BI flere eksempel-rapporter, som du kan udforske. Disse rapporter indlæses i Mit arbejdsområde, så du kan udforske dem privat. Du kan få adgang til eksempel-rapporter i afsnittet **Learn** i navigationsruden.



Distribuer indhold vha en App

I et arbejdsområde kan du oprette en **app**, som **giver forbrugerne en forenklet brugergrænseflade til at få adgang til rapporter og dashboards**. I appkonfigurationen skal du konfigurere appen, vælge det indhold, der skal inkluderes (begrænset til det aktuelle arbejdsområde), og vælge din målgruppe.

Når du har oprettet en app, skal du opdatere appen efter hver ændring af elementer i arbejdsområdet. Kravet om at opdatere appen giver dig mulighed for at styre, hvilken version af indholdet der er synlig for din målgruppe.

Power BI Apps

Search

Home

Create

Browse

OneLake data hub

Apps

Metrics

Monitoring hub

Deployment pipelines

Learn

Workspaces

PL-300

1 Setup*

2 Content

3 Audience

Build your app

App name *

Demo App

Description *

Enter a summary

Describe your app.

App logo

Upload

Delete

App theme color

Contact information

☒ Show app publisher

☐ Show items contacts from the workspace

☐ Show specific individuals or groups

Enter a name or email address

Copy link

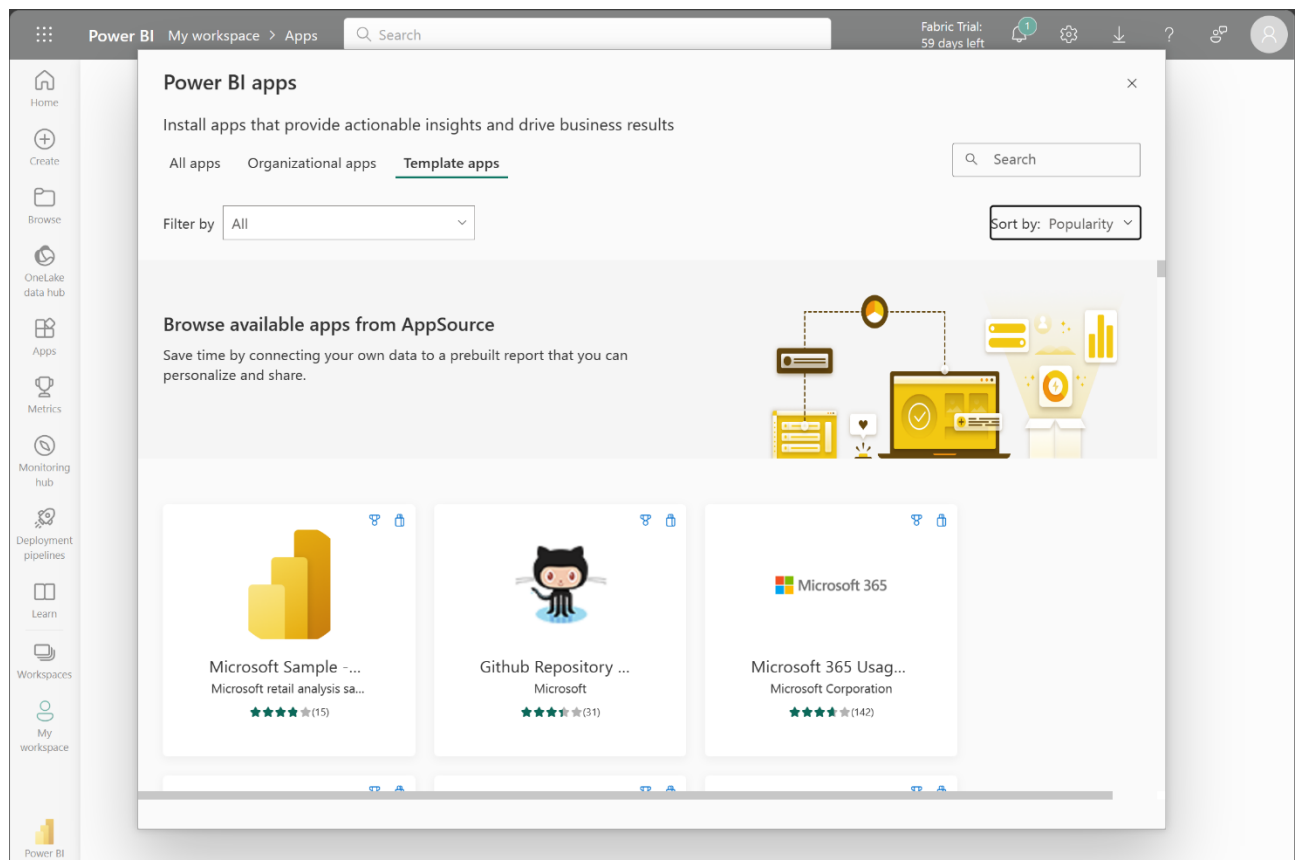
Advanced settings

Apps er den ideelle delingsløsning i enhver organisation.

Selvom du kan give adgang til arbejdsområdet, kan tilladelser til arbejdsområdet give brugerne adgang til mere indhold end ønsket. Deling af individuelle elementer er også et problem, hvis du foretager ændringer, som forbrugerne ikke skal se endnu.

Udforsk skabelonapps

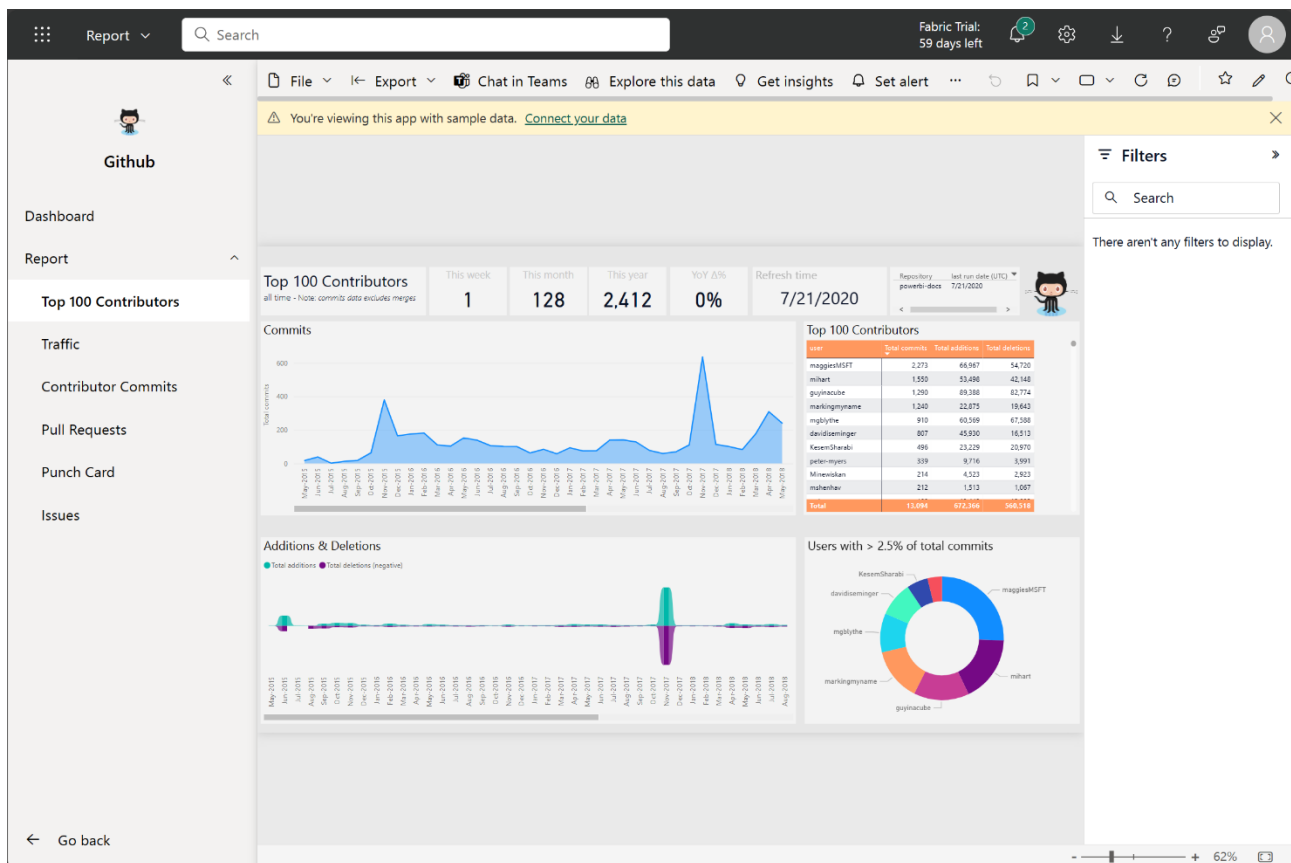
Nu, hvor du forstår, hvad en app er, kan vi se på **skabelonapps**. Med skabelonapps kan du finde en eksisterende app, der passer til dine behov, og derefter kan du oprette forbindelse til dine data. Disse apps kan være en god måde hurtigt at dele indsigt på med en minimal indsats.



Tip

Hvis du vil have adgang til skabelonapps, skal du vælge ikonet Apps i venstre navigationsrude > Hent apps > Skabelonapps.

På følgende skærbillede har vi installeret GitHub-skabelonappen og udvidet rapporten. Vi kan se forskellige rapportsider, herunder *De 100 største bidragydere* og *Pull-anmodninger*. Hvis din organisation bruger GitHub, kan brug af denne skabelonapp nemt understøtte dine behov uden at starte fra starten.



Opdater en semantisk model

For at understøtte dine data, der ændres hele tiden, kan du konfigurere planlagte opdateringer af dine semantiske modeller i Power BI-tjeneste. Opdateringer efter behov er også tilgængelige.

Home Create Browse OneLake data hub Apps Metrics Monitoring hub Deployment pipelines Learn Workspaces PL-300 Power BI

General Dashboards **Semantic models** Workbooks Dataflows Datamarts (Preview) App

Demo

Sales Analysis

Settings for Sales Analysis

[View semantic model](#)

This semantic model has been configured by angie@beepboopbop.onmicrosoft.com.

[Refresh history](#)

Semantic model description

Gateway and cloud connections

Data source credentials

Parameters

Query Caching

Refresh

Configure a refresh schedule

Define a data refresh schedule to import data from the data source into the semantic model. [Learn more](#)

☐ Off

Refresh frequency

Daily

Time zone

(UTC-05:00) Eastern Time (US and Ca

Time

[Add another time](#)

Send refresh failure notifications to

☒ Semantic model owner

☐ These contacts:

Enter email addresses

Apply Discard

Tip

Du kan finde flere oplysninger om alle tidsplaner for opdateringer i [dokumentationen Til opdatering af data](#).

Næste undermodul:

Videnkontrol

Fuldført200 XP

- 3 minutter

1.

Hvad er den almindelige aktivitetsflow i Power BI?

Opret en rapport i tjenesten Power BI , dele den til mobil Power BI og interagere med den Power BI Desktop.

Opret en rapport i Power BI Desktop, del den med tjenesten Power BI , og brug rapporter i tjenesten og på Power BI mobil.

Rigtig. Tjenesten Power BI giver dig mulighed for at få vist og bruge rapporter og dashboards, men du kan ikke bruge figurdata.

Opret en rapport på mobil Power BI , del den med Power BI Desktop, og brug rapporter i Power BI tjenesten.

2.

Hvilken af følgende er byggesten til Power BI?

Felter og visualiseringer.

Semantiske modeller og visualiseringer.

Rigtig. Uden en semantisk model kan du ikke oprette visualiseringer, og rapporter består af visualiseringer.

Semantiske modeller og rapporter.

3.

Hvad er en samling rapporter og dashboards kaldt ind Power BI?

Lærredet.

Planlagt opdatering

En app.

Rigtig. En app er en samling færdige visualiseringer, der er arrangeret i dashboards og rapporter på forhånd. Du kan hente apps, der opretter forbindelse til mange onlinetjenester, fra denne AppSource.

Næste undermodul:

Resumé

Fuldført100 XP

- 1 minut

Microsoft Power BI tilbyder en komplet dataanalyteløsning, der omfatter dataforberedelse, visualisering og distribution. **Semantiske modeller** og **visualiseringer** er komponenterne i Power BI.

Flowet og komponenterne i Power BI omfatter:

- **Power BI Desktop** til oprettelse af semantiske modeller og rapporter med visualiseringer.
- **Power BI-tjeneste** til oprettelse af dashboards fra publicerede rapporter og distribution af indhold med apps.
- **Power BI – Mobil** til adgang på farten til det Power BI-tjeneste indhold, der er udviklet til mobilenheder.

Ved hjælp af Power BI kan du træffe dataorienterede beslutninger på tværs af din organisation.

Alle enheder er fuldført:



<https://learn.microsoft.com/api/achievements/share/da-dk/bjarnehansen-7412/VD8RHAFM?sharingId=39714CB1FE13C67B>

<https://learn.microsoft.com/api/achievements/share/da-dk/bjarnehansen-7412/KGRMYQ2B?sharingId=39714CB1FE13C67B>

**** Føj målinger(mesaures) til Power BI Desktop-modeller

3 typer measure:

enkelt-stående, som udregner aggregeret værdi, som sum(feltværdi), ses som lommeregner
kolonne-dannende : ses med lille tabel med sigma i ikonen

tabel-dannende, icon viser tabel med lommeregner

Resumé

Fuldført 100 XP

- 1 minut

I dette modul har du lært, at Power BI-målinger er enten implicitte eller eksplicitte.

Implicitte målinger er automatiske funktionsmåder, der understøttes af visualiseringer, fx aggregeringer

eksplicitte målinger bruger DAX-formler, der opsummerer modeldata.

Eksplicitte målinger er vigtige, fordi de giver dig mulighed for at oprette komplekse DAX-formler for at opnå de nøjagtige beregninger, som dine rapportvisualiseringer har brug for. Selvom du har lært at oprette enkle og sammensatte målinger i dette modul, lærer du i senere moduler at oprette mere effektive målinger ved hjælp af filterændringsfunktioner og iteratorfunktioner.

Introduktion

Fuldført 100 XP

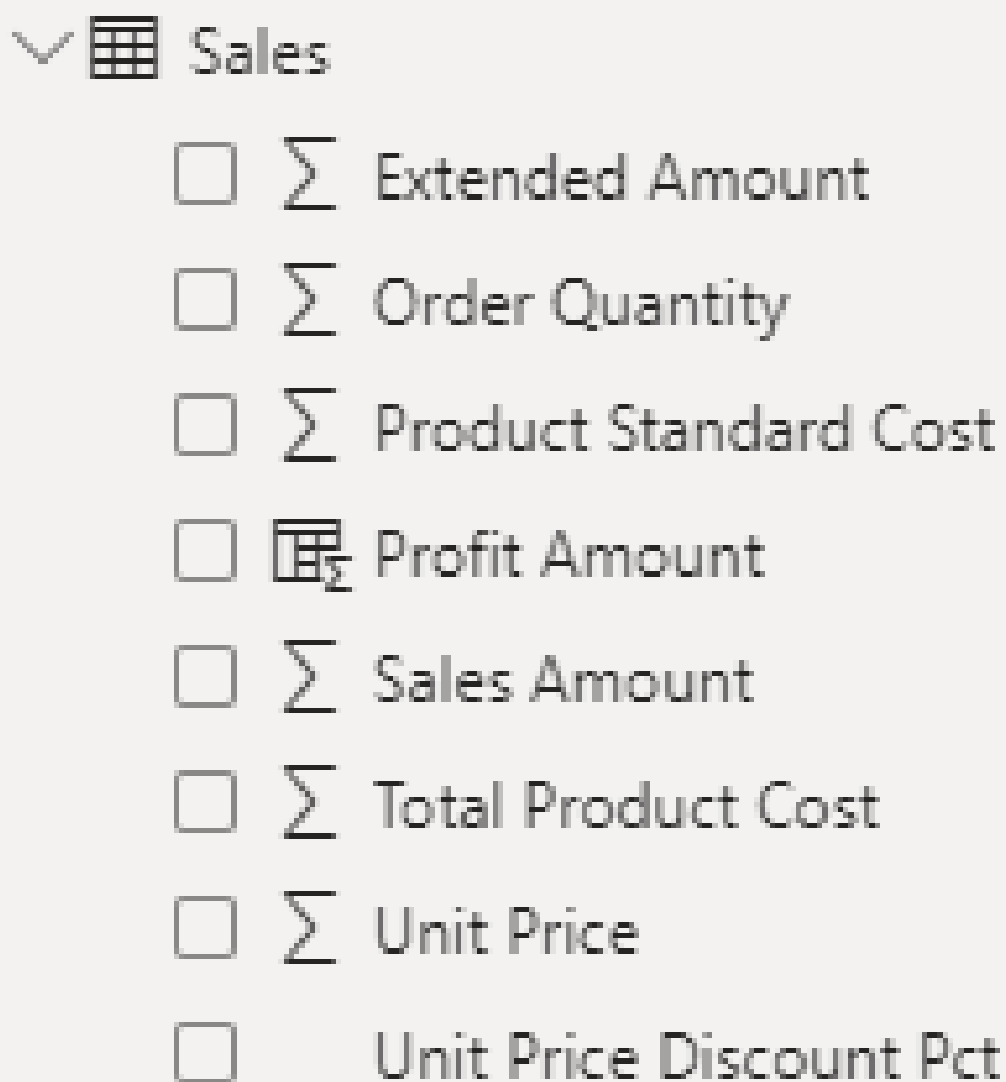
- 3 minutter

Målinger i Microsoft Power BI-modeller er enten *implicitte* eller *eksplicitte*. Implicitte målinger er automatiske funktionsmåder, der gør det muligt for visuals at opsummere modelkolonnedata. Eksplicitte målinger, også blot kaldet *målinger*, er beregninger, du kan føje til din model. I dette modul fokuseres der på, hvordan du kan bruge implicitte målinger.

I ruden **Felter** angiver en kolonne, der vises med sigmasymbolet (Σ), to fakta:

- Det er en numerisk kolonne.
- Den opsummerer kolonneværdier, når de bruges i et visual (når de føjes til en feltbrønd, der understøtter opsummering).

På følgende billede kan du se, at tabellen **Sales** kun indeholder felter, der kan opsummeres, herunder den beregnede kolonne **Profit Amount**.



Som datakonstruktør kan du styre, om og hvordan kolonnen opsummerer ved at angive egenskaben **Opsummering** til **Opsummer ikke** eller til en bestemt sammenlægningsfunktion. Når du angiver egenskaben **Opsummering** til **Opsummer ikke**, vises sigma-symbolet ikke længere ud for kolonnen i ruden **Felter**.

Hvis du vil se, hvordan forfattere af rapporter kan bruge implicitte målinger, kan du først downloade og åbne filen [Adventure Works DW 2020 M04. pbix](#).

I rapporten skal du fra tabellen **Sales** føje feltet **Sales Amount** til det matrixvisual, der grupperer regnskabsår og måned i rækkerne.

Fiscal Year	Sales Amount
 FY2018	\$23,860,891.17
2017 Jul	\$1,423,357.32
2017 Aug	\$2,057,902.45
2017 Sep	\$2,523,947.55
2017 Oct	\$561,681.48
2017 Nov	\$4,764,920.16
2017 Dec	\$596,746.56

Hvis du vil bestemme, hvordan kolonnen opsummeres i ruden med visualefter, skal du vælge pilen for feltet **Sales Amount** og derefter gennemse indstillingerne for genvejsmenuen.

Remove field

Rename for this visual

Move to



Add a sparkline

Conditional formatting



Remove conditional formatting



Sum

Average

Minimum

Maximum

Count (Distinct)

Count

Standard deviation

Variance

Median

Bemærk, at der er en markering ud for sammenlægningsfunktionen **Sum**. Denne markering angiver, at kolonnen opsummeres, ved at kolonneværdier *lægges sammen*. Det er også muligt at ændre sammenlægningsfunktionen ved at vælge en af de andre indstillinger som Gennemsnit, Minimum osv.

Føj derefter feltet **Unit Price** til matrixvisual'et.

Fiscal Year	Sales Amount	Unit Price
 FY2018	\$23,860,891.17	\$1,273.63
2017 Jul	\$1,423,357.32	\$1,817.15
2017 Aug	\$2,057,902.45	\$1,181.42
2017 Sep	\$2,523,947.55	\$1,030.56
2017 Oct	\$561,681.48	\$3,228.05
2017 Nov	\$4,764,920.16	\$1,009.71
2017 Dec	\$596,746.56	\$3,174.18

Standardopsummeringen er nu angivet til **Gennemsnits unitpris**, sket automatisk(modelkonstruktøren ved, at det ikke er hensigtsmæssigt at lægge enhedsprisværdier sammen, fordi det er prissatser, som ikke kan lægges sammen).

Remove field

Rename for this visual

Move



Move to



Add a sparkline

Conditional formatting



Remove conditional formatting

Sum

✓ Average

Minimum

Maximum

Count (Distinct)

Count

Standard deviation

Variance

Det er en **Implicitte måling/beregning**(sket automatisk uden eksplicit 'besked'), det giver mulighed for, at rapportforfatterne starter med en standardopsummeringsteknik, og giver dem mulighed for at ændre den, så den passer til deres visuelle krav.

Numeriske kolonner understøtter den største række sammenlægnings, aggregeringsfunktioner:

- Sum
- Gennemsnit
- Minimum
- Maximum
- Antal (adskilt)
- Antal
- Standardafvigelse
- Varians
- Median

Opsummer ikke-numeriske kolonner

Kolonner, der ikke er numeriske, kan opsummeres. Sigma-symbolet vises dog ikke ud for ikke-numeriske kolonner i ruden **Felter**, fordi de ikke opsummeres som standard.

Tekstkolonner giver mulighed for følgende sammenlægninger:

- Første (alfabetisk)
- Sidste (alfabetisk)
- Antal (adskilt)
- Antal

Datokolonner giver mulighed for følgende sammenlægninger:

- Tidligste
- Seneste
- Antal (adskilt)
- Antal

Booleske kolonner giver mulighed for følgende sammenlægninger:

- Antal (adskilt)
- Antal

Fordele ved implicitte målinger

Der er flere fordele forbundet med implicitte målinger. Implicitte målinger er enkle begreber at lære og bruge, og de giver fleksibilitet i den måde, rapportforfattere visualiserer modeldata på. **Derudover betyder de mindre arbejde for dig som datakonstruktør, da du ikke behøver at oprette eksplicite beregninger.**

Begrænsninger af implicitte målinger

Implicitte målinger har begrænsninger. Selvom det er muligt at angive en hensigtsmæssig opsummeringsmetode, kan rapportforfattere vælge at lægge en kolonne sammen på ikke-hensigtsmæssige måder. I matrixvisual'et kan du f. eks. ændre sammenlægningsfunktionen for **Unit Prices** til **Sum**.

Fiscal Year	Sales Amount	Sum of Unit Price
☒ FY2018	\$23,860,891.17	\$13,905,519.77
2017 Jul	\$1,423,357.32	\$1,164,795.52
2017 Aug	\$2,057,902.45	\$1,115,258.56
2017 Sep	\$2,523,947.55	\$1,291,296.17
2017 Oct	\$561,681.48	\$561,681.48
2017 Nov	\$4,764,920.16	\$2,159,760.38
2017 Dec	\$596,746.56	\$596,746.56

Rapportvisualiseringen overholder din konfiguration, men den har nu produceret kolonnen **Sum af enhedspris**, som præsenterer vildledende data.

Den største begrænsning for implicitte målinger er, at de kun fungerer for enkle scenarier, hvilket betyder, at de kun kan opsummere kolonneværdier, der bruger en bestemt sammenlægningsfunktion. I situationer, hvor du har brug for at beregne forholdet mellem de enkelte måneders salgsbeløb i forhold til det årlige salgsbeløb, skal du derfor oprette en eksplicit måling ved at skrive en DAX-formel (Data Analysis Expressions) for at opfylde dette mere avancerede krav.

Implicitte målinger fungerer ikke, når der sendes en forespørgsel til modellen ved hjælp af MDX (flerdimensionelle udtryk). Dette sprog forventer eksplicite målinger og kan ikke opsummere kolonnedata. Den bruges, når der forespørges om en semantisk Power BI-model ved hjælp af [Analysér i Excel](#), eller når en [sideinddelt Power BI-rapport](#) bruger en forespørgsel, der genereres af den grafiske MDX-forespørgselsdesigner.

Opret simple målinger(measures)

Opret simple målinger

Fuldført100 XP

- 3 minutter

Du kan skrive en DAX-formel for at føje en måling til en vilkårlig tabel i din model. En formel for en måling skal returnere en skalarværdi eller en enkelt værdi.

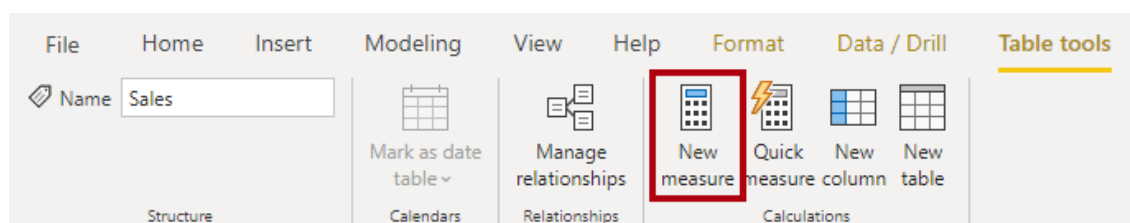
Bemærk

I tabelmodellering findes begrebet beregnet måling ikke. Ordet *beregnet* bruges til at beskrive beregnede tabeller og beregnede kolonner. Det adskiller dem fra tabeller og kolonner, der stammer fra Power Query, hvor begrebet eksPLICIT måling ikke findes.

Målinger lagrer ikke værdier i modellen. De bruges i stedet på forespørgselstidspunktet til at returnere opsummeringer af modeldata. Desuden kan målingerne ikke referere direkte til en tabel eller en kolonne. De skal sende tabellen eller kolonnen til en funktion for at oprette en opsummering.

En *simpel* måling er en måling, som lægger værdierne i en enkelt kolonne sammen. Den gør, hvad implicitte målinger automatisk gør.

I det næste eksempel skal du føje en måling til tabellen **Sales**. Vælg tabellen **Sales** i ruden **Felter**. Vælg **Ny måling** fra gruppen **Beregninger** på det kontekstafhængige bånd **Tabelværktøjer** for at oprette en måling.



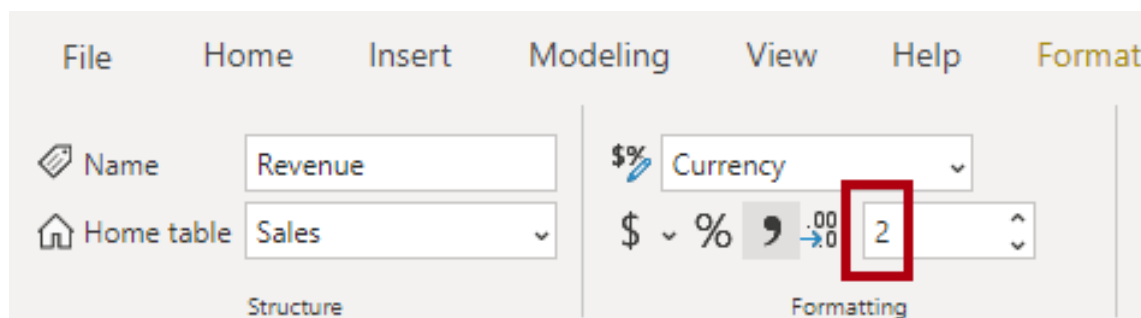
Angiv følgende målingsdefinition på formellinjen, og tryk derefter på **Enter**.

DAXKopier

Revenue =SUM(Sales[Sales Amount])

Målingsdefinitionen føjer målingen **Revenue** til tabellen **Sales**. Den bruger [SUM](#) DAX-funktionen til at opsummere værdierne i kolonnen **Sales Amount**.

Angiv decimalerne til **2** på det kontekstafhængige bånd **Målingsværktøjer** i gruppen **Formatering**.



Tip

Angiv formateringsindstillingerne, umiddelbart efter at du har oprettet en måling, for at sikre, at værdierne i alle rapportvisuals er vises pænt og konsistent.

Føj derefter målingen **Revenue** til matrixvisual'et. Bemærk, at den giver samme resultat som den implicitte måling **Sales Amount**.

Fjern **Sales Amount** og **Sum of Unit Price** fra matrixvisual'et.

Derefter skal du oprette flere målinger. Opret målingen **Cost** ved hjælp af følgende målingsdefinition, og angiv derefter formatet med to decimaler.

is equivalent to `PRODUCTX( Table, Table[Column] )`.

DAXKopiér

Cost =`SUM(Sales[Total Product Cost])`

Opret målingen **Profit**, og angiv derefter formatet med to decimaler.

DAXKopiér

Profit =`SUM(Sales[Profit Amount])`

Bemærk, at kolonnen **Profit Amount** er en beregnet kolonne. Dette emne gennemgås senere i dette modul.

Opret derefter målingen **Quantity**, og formatér den som et heltal med tusindtalsseparator.

DAXKopiér

Quantity =

`SUM(Sales[Order Quantity])`

Opret tre enhedsprismålinger, og angiv derefter formatet for hvert af dem med to decimalpladser. Bemærk de forskellige DAX-sammenlægningsfunktioner, der bruges: [MIN](#), [MAX](#) og [AVERAGE](#).

DAXKopiér

Minimum Price =

`MIN(Sales[Unit Price])`

DAXKopiér

Maximum Price =

`MAX(Sales[Unit Price])`

DAXKopiér

Average Price =

`AVERAGE(Sales[Unit Price])`

Nu kan du skjule kolonnen **Unit Price**, hvilket medfører, at rapportforfatterne mister muligheden for at opsummere kolonnen undtagen ved hjælp af dine målinger.

Tip

Tilføjelse af målinger og skjulning af kolonner er den måde du som datakonstruktør kan begrænse opsummeringsmuligheder på.

Opret derefter følgende to målinger, der tæller antallet af ordrer og ordrelinjer. Formatér begge målinger med nul decimalpladser.

DAXKopier

Order Line Count = COUNT(Sales[SalesOrderLineKey]) //121K

DAXKopier

Order Count =DISTINCTCOUNT('Sales Order'[Sales Order]) //31K

Hver række må have unik værdi af customerKey, productKey

Der kan nemt være flere (customerKey, productKey) for en order

DAX-funktionen [COUNT](#) tæller antallet af værdier, der ikke er TOMME, i en kolonne, mens [DISTINCTCOUNT](#) DAX-funktionen tæller antallet af entydige værdier i en kolonne. Da en ordre kan have en eller flere ordrelinjer, har kolonnen **Sales Order** dublerede værdier. En adskilt optælling af værdier i denne kolonne tæller antallet af ordrer korrekt.

Du kan også vælge den bedste måde at skrive målingen **Ordre Line Count** på. I stedet for at tælle værdier i en kolonne er det semantisk tydeligere at bruge [COUNTROWS](#) DAX-funktionen. I modsætning til de tidligere introducerede sammenlægningsfunktioner, som aggregerer kolonneværdier, COUNTROWS tæller funktionen antallet af rækker *for en tabel*.

Rediger formelen for målingen **Ordrelinjeantal** , som du oprettede ovenfor, til følgende parametre:

DAXKopier

Order Line Count =COUNTROWS(Sales)

Føj hver enkelt måling til matrixvisual'et.

Alle de målinger, du har oprettet, anses for at være simple målinger, fordi de lægger en enkelt kolonne eller en enkelt tabel sammen.

Opret sammensatte målinger, består af flere measures

Fuldført100 XP

- 1 minut

Når en måling refererer til en eller flere målinger, kaldes det en *sammensat måling*.

I dette eksempel skal du ændre målingen **Profit** ved hjælp af følgende målingsdefinition. Formatér målingen med to decimalpladser.

DAXKopier

Profit =[Revenue] - [Cost]

Føj derefter målingen **Profit** til matrixvisual'et.

Nu, hvor din model gør det muligt at opsummere avancen, kan du slette den beregnede kolonne **Profit** beregnet kolonne.

Ved at fjerne denne beregnede kolonne har du optimeret den semantiske model. Hvis du fjerner disse kolonner, resulterer det i en reduceret semantisk modelstørrelse og kortere dataopdateringstider. Den beregnede kolonnen **Profit Amount** var ikke påkrævet, fordi målingen **Profit** kan producere det påkrævede resultat direkte.

Profit divided by Revenue = DIVIDE([Profit], [Revenue])*100 //profit %

Sammenlign beregnede kolonner med målinger

Fuldført 100 XP

- 1 minut

For DAX-begyndere kan beregnede kolonner og målinger virke forvirrende. I det følgende afsnit gennemgås lighederne og forskellene mellem disse.

Der er følgende ligheder mellem beregnede kolonner og målinger:

- Beregninger, som du kan føje til din semantiske model.
- De er begge defineret ved hjælp af en DAX-formel.
- Du refererer til dem i DAX-formler ved at sætte deres navne i kantede parenteser.

De områder, hvor beregnede kolonner og målinger er forskellige, omfatter:

- **Formål** – Beregnede kolonner udvider en tabel med en ny kolonne, mens målinger definerer, hvordan modeldata skal opsummeres.
- **Evaluering** – Beregnede kolonner evalueres ved hjælp af *rækkekontekst* på dataopdateringstidspunktet, mens målinger evalueres ved hjælp af *filterkontekst* på forespørgselstidspunktet. Filterkontekst introduceres i et senere modul. Det er et vigtigt emne at forstå og mestre, så du kan opnå komplekse opsummeringer.
- **Lager** – Beregnede kolonner (i tabeller i importlagringstilstand) lagrer en værdi for hver række i tabellen, men en måling lagrer aldrig værdier i modellen.
- **Visuel brug** – Beregnede kolonner kan (som alle kolonner) bruges til at filtrere, gruppere eller opsummere (som en implicit måling), mens målinger er designet til at opsummere.

Næste undermodul: Tjek din viden

1.

Hvilket udsagn om målinger er korrekt?

Målinger gemmer værdier i den semantiske model.

Målinger skal føjes til den semantiske model for at opnå opsummering.

Målinger kan referere direkte til kolonner.

Målinger kan referere direkte til andre målinger.

Målinger kan referere til andre målinger. Det kaldes en sammensat måling.

2.

Hvilken DAX-funktion kan opsummere en tabel?

SUM

Funktionen SUM kan kun opsummere en kolonne.

AVERAGE

COUNTROWS

Funktionen COUNTROWS opsummerer en tabel ved at returnere antallet af rækker.

DISTINCTCOUNT

3.

Hvilke af følgende udsagn, der beskriver ligheden mellem målinger og beregnede kolonner i en importmodel, er sandt?

De kan opnå opsummering af modelldata.

En beregnet kolonne kan opsummeres (implicit måling), og en måling opnår altid opsummering.

De gemmer værdier i den semantiske model.

De evalueres under dataopdatering.

Mens beregnede kolonner evalueres på dataopdateringstidspunktet, evalueres målinger kun på forespørgselstidspunktet.

De kan oprettes ved hjælp af hurtigberegninger.

Fortsæt det fremragende arbejde



Føj målinger til Power BI
Desktop-modeller

Du har opnået et resultat!

Tillykke, men hvad skal du gøre nu?

Lad os først dele dit resultat

Du afsætter tid til at lære noget nyt, så lad dit netværk tage del i din sejr!



Mist ikke dit momentum, bliv ved med at lære

Nedenfor finder du anbefalet indhold, der kan hjælpe dig på vej!

Næste modul i dette læringsforløb

<https://learn.microsoft.com/api/achievements/share/da-dk/bjarnehansen-7412/J9ZNWZMT?sharingId=39714CB1FE13C67B>

1. [Learn](#)
2. [Træning](#)
3. [Gennemse](#)

*** [Beskriv Power BI Desktop-modeller](#)

Introduktion

Fuldført 100 XP

- 10 minutter

I Microsoft Power BI understøtter den semantiske model udvikling af rapporter og dashboards. Hvis du vil oprette de rigtige rapporter og dashboards, skal du have en klar forståelse af de spørgsmål, dine rapport- og dashboardbrugere vil stille. Dit næste fokus bør være på det semantiske modeldesign, der bedst understøtter visualiseringerne i dine rapporter og dashboards. Når du har fundet frem til et modeldesign, er du klar til at bruge Power BI Desktop til at udvikle modellen.

Når du udvikler modellen, skal du udføre følgende opgaver:

- Oprette forbindelse til data
- Omdanne og forbered data
- Definere forretningslogik ved at tilføje DAX-beregninger (Data Analysis Expressions)
- Publicere modellen til Power BI

Hvis du forstår strukturen af semantiske modeller, kan det hjælpe dig med at designe den rigtige model til at understøtte dine rapporter og dashboards. En semantisk model kan udvikles på mange måder, men en eller flere af disse måder er mere optimal. Optimale modeller er vigtige for at levere en god ydeevne for forespørgsler og for at minimere opdateringstider for data og brugen af tjenesteressourcer, herunder hukommelse og CPU. Jo færre ressourcer, der bruges, jo flere modeller kan der hostes, og jo mere kan omkostningerne reduceres.

Næste undermodul: Design af stjerneschemaer

[Næste](#)

Design af stjerneschemaer

Fuldført 100 XP

- 5 minutter

Det er usædvanligt, at en semantisk Power BI-model består af en enkelt tabel. En model med en enkelt tabel kan være et enkelt design, måske ét, der egner sig til en dataudforskningsopgave eller blåstempling, men er ikke et optimalt modeldesign. En optimal model overholder designprincipperne for *stjerneschemaer*. Stjerneschema refererer til en designtilgang, der ofte bruges af designere af relationsdata warehouse, fordi det præsenterer en brugervenlig struktur, og det understøtter analyseforespørgsler med høj ydeevne.

Dette designprincip kaldes et stjerneschema, da det klassificerer modeltabeller som enten *faktum* eller *dimension*. I et diagram danner en faktatabel midten af en stjerne, mens dimensionstabeller, der placeres rundt om faktatabellen, repræsenterer stjernens spidser.

Faktatabeller

En faktatabels rolle er at lagre en akkumulering af rækker, der repræsenterer observationer eller hændelser, som registrerer en bestemt forretningsaktivitet. Hændelser, der lagres i en salgsfaktatabel, kan f. eks. være

salgsordrer og ordrelinjerne. Du kan også bruge en faktatabel til at registrere lagerbevægelser, lagerbeholdninger eller daglige valutakurser. Faktatabeller indeholder generelt mange rækker. Efterhånden akkumuleres rækkerne i faktatabellen. I analyseforespørgsler (som defineres senere i dette modul) opsummeres faktatabeldata for at oprette værdier som f.eks. salg og mængde.

Dimensionstabeller

Dimensionstabeller beskriver dine forretningsenheder, der typisk repræsenterer personer, steder, produkter eller begreber. En tabel med datodimensioner, der indeholder én række for hver dato, er et almindeligt eksempel på en tabel med konceptdimensioner. Kolonnerne i dimensionstabeller tillader filtrering og gruppering af faktatabeldata.

Hver dimensionstabel skal have en entydig kolonne, der kaldes nøglekolonnen. En entydig kolonne indeholder ikke dublerede værdier, og der må aldrig være manglende værdier i den. I en produktdimensionstabel kan kolonnen have navnet **ProductKey** eller **ProductID**. De øvrige kolonner indeholder sandsynligvis beskrivende værdier, f. eks. produktnavn, underkategori, kategori, farve osv. I analyseforespørgsler bruges disse kolonner til at filtrere og gruppere data.

Sammenlign efter fakta-og dimensionstabeller

I følgende figur sammenlignes karakteristika for fakta- og dimensionstabeller.

Udvid tabel

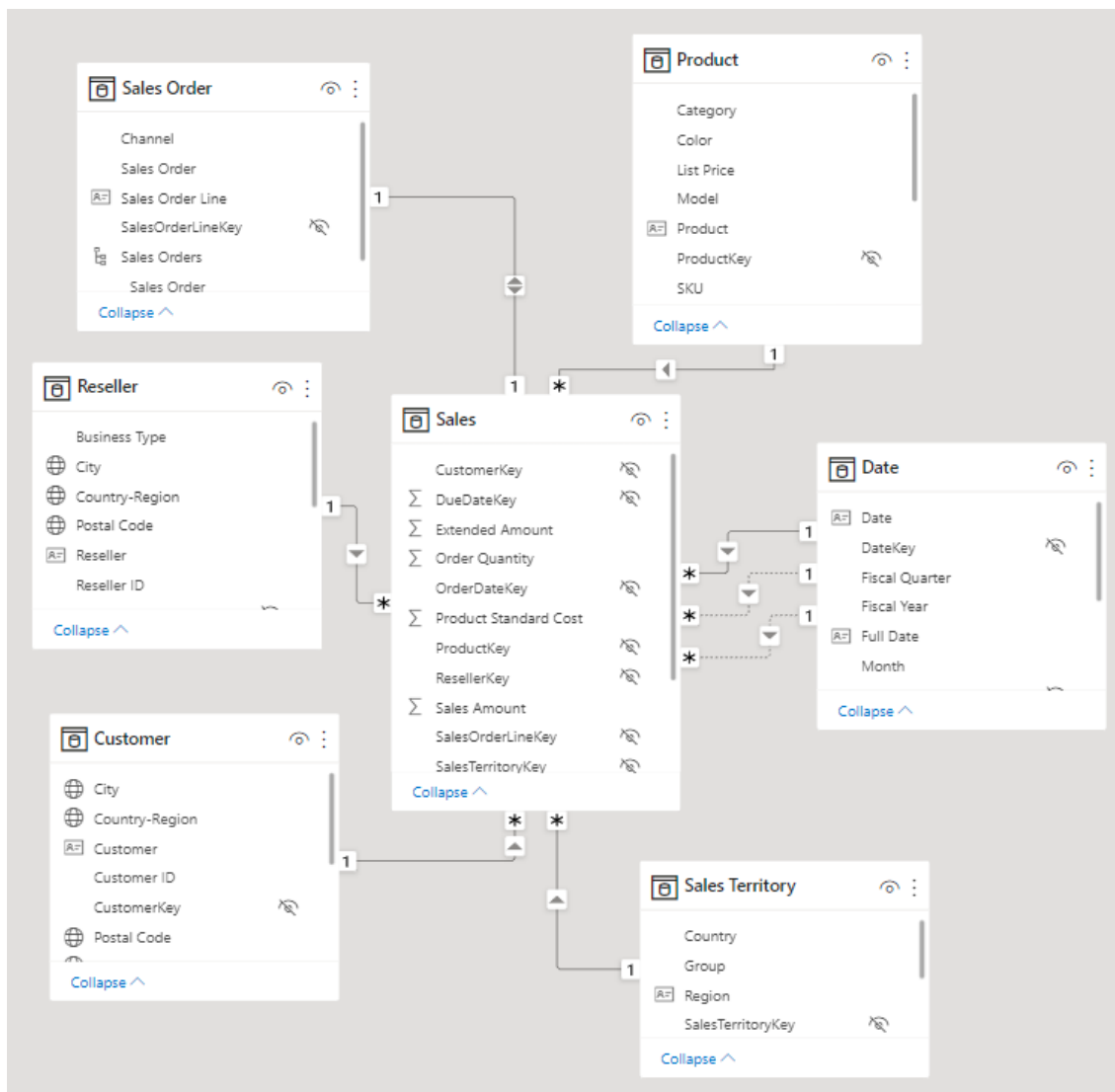
Egenskab	Dimensionstabel	Faktatabel
Modelformål	Lagrer forretningsenheder	Lagrer begivenheder eller observationer
Tabelstruktur	Omfatter en nøglekolonne og beskrivende kolonner til filtrering og gruppering	Omfatter dimensionsnøglekolonner og målingskolonner, der kan opsummeres
Datamængde	Indeholder typisk færre rækker (i forhold til faktatabeller)	Kan indeholde mange rækker
Formål med forespørgsel	At filtrere og gruppere	At opsummere

Relater stjerneskematabeller

Dimensionstabeller relateret til faktatabeller i modellen ved hjælp af en-til-mange-relationer. Relationerne gør det muligt for filtre og grupper, der anvendes på kolonnerne i dimensionstabellen, at overføre til faktatabellen. Dette designmønster er almindeligt.

Dimensionstabeller kan bruges til at filtrere flere faktatabeller, og faktatabeller kan filtreres efter flere dimensionstabeller. Det er dog ikke god praksis at knytte en faktatabel direkte til en anden faktatabel.

Hvis du vil øve dette koncept, kan du downloade filen [Adventure Works DW 2020 M01. pbix](#), åbne filen og derefter skifte til modeldiagrammet.

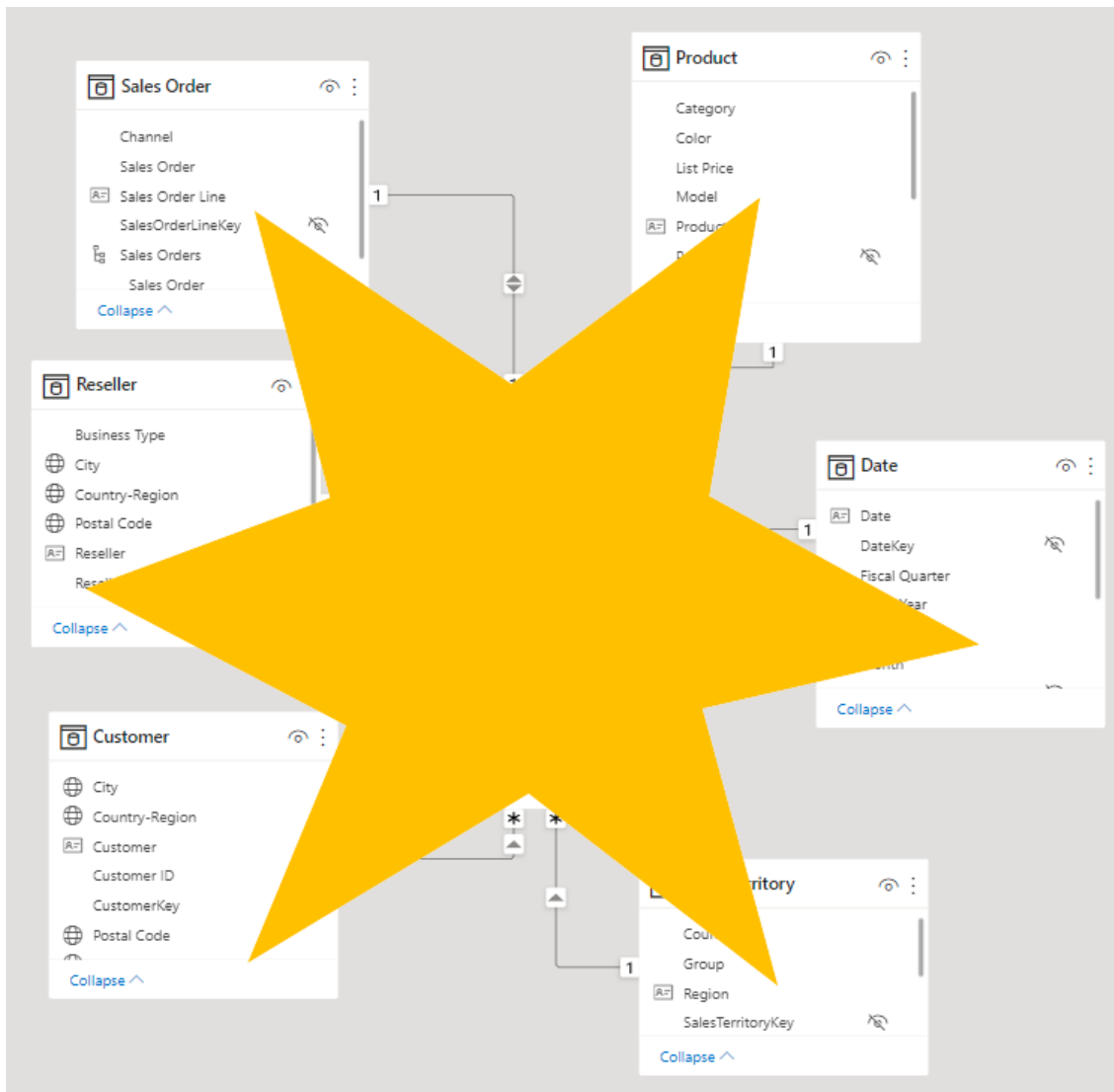


Bemærk, at modellen består af syv tabeller, hvoraf den ene hedder **Sales** og er faktatabellen. De resterende tabeller er dimensionstabeller og har følgende navne:

- Customer
- Dato
- Product
- Reseller
- Sales Order
- Sales Territory

Læg mærke til relationerne mellem dimensions- og faktatabellerne, og at hvert relationsfilter peger på faktatabellen. Det betyder, at når der anvendes filtre på kolonnerne med dimensionstabeller (til at filtrere eller gruppere efter kolonneværdier), filtreres de relaterede fakta og opsummeres.

Hvis du kigger nærmere på mønsteret, kan du muligvis se en stjerneform.



Du kan få flere oplysninger om design af stjerneschemaer under [Forstå stjerneschema og vigtigheden for Power BI](#).

Næste undermodul: Analyseforespørgsler

[Tidligere](#)

Analyseforespørgsler

Fuldført 100 XP

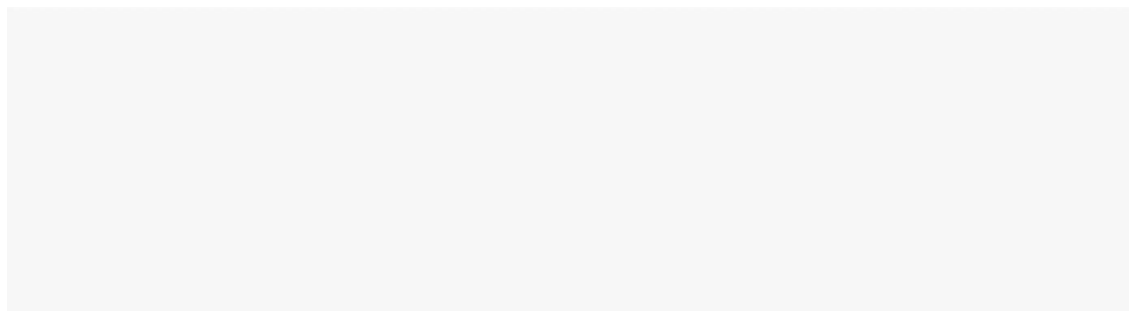
- 2 minutter

En *analyseforespørgsel* er en forespørgsel, der giver et resultat fra en semantisk model. Ethvert Power BI-visual sender en analytisk forespørgsel til Power BI i baggrunden for at forespørge modellen. Analyseforespørgslen skrives som en DAX-forespørgselssætning (Data Analysis Expressions). Du behøver

dog ikke at skrive en oprindelig DAX-sætning. Du skal kun konfigurere rapportvisualiseringer ved at tilknytte semantiske modelfelter.

En analyseforespørgsel har tre faser, der implementeres i følgende rækkefølge:

1. Filtrer
2. Gruppér
3. Opsummer



Filtrering eller anvendelse af udsnit, er målrettet de relevante data. I Power BI-rapporter kan filtre anvendes på tre forskellige områder: hele rapporten, en bestemt side eller en bestemt visualisering. Filtrering anvendes også i baggrunden, når sikkerhed på rækkeniveau gennemtvinges. Hvert rapportvisual kan arve filtre, eller der kan anvendes filtre direkte på det.

Gruppering eller opdeling, inddeler forespørgselsresultater i grupper.

Opsummering opretter et resultat med en enkelt værdi. Numeriske kolonner opsummeres normalt ved hjælp af opsummeringsmetoder (sum, count og mange andre). Disse metoder er simple opsummeringer. Du kan opnå mere komplekse opsummeringer, f. eks. en procentdel af hovedtotalen, ved at definere målinger, der er skrevet i DAX.

Ikke alle analytiske forespørgsler behøver at filtrere, gruppere og opsummere:

- Rapportvisuals filtreres normalt, måske efter en tidsperiode eller en geografisk placering.
- Gruppering er valgfri. F.eks. bruges et kortvisual, der bruges til at vise en enkelt værdi, ikke til gruppering.
- Rapportvisuals opsummerer normalt. En bemærkelsesværdig undtagelse er dog udsnitsvisual'et, som ikke bruges til opsummering.

Konfigurer rapportvisuals

Fuldført100 XP

- 5 minutter

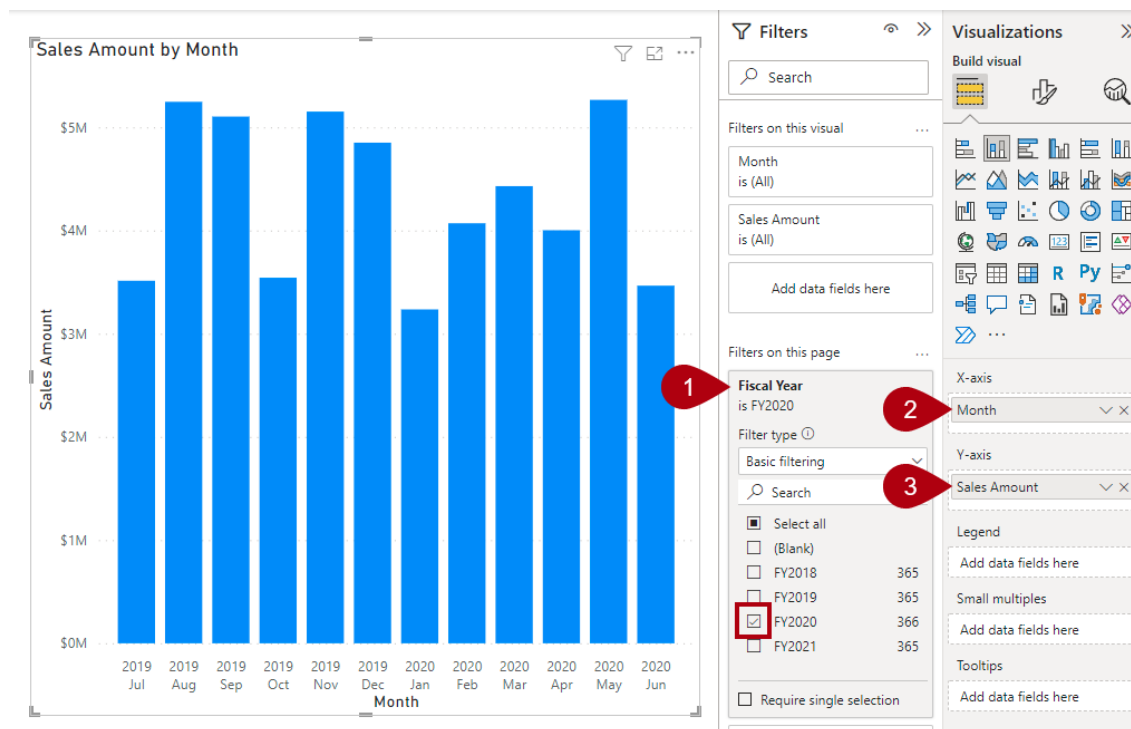
Rapportforfattere opretter rapportdesign ved at føje rapportvisuals og andre elementer til sider. Andre elementer omfatter tekstfelter, knapper, former og billeder. Hvert af disse elementer er konfigureret uafhængigt af semantiske modelfelter.

På designtidspunktet skal du benytte følgende fremgangsmåde for at tilføje og konfigurere et rapportvisual:

1. Vælg en visualtype, f. eks. et liggende søjlediagram.
2. Knyt semantiske modelfelter, der vises i ruden **Felter**, til visualiseringsfeltbrønderne. For et liggende søjlediagram er brønderne **Y-akse**, **X-akse**, **Forklaring**, **Små multipla** og **Værktøjstip**.
3. Konfigurer tilknyttede felter. Det er muligt at omdøbe tilknyttede felter eller til at ændre feltet, så det er opsummerer eller ej. Hvis feltet opsummerer, kan du vælge opsummeringsmetoden.
4. Anvend formateringsindstillinger, f. eks. akseegenskaber, datanavne og mange andre.

I følgende eksempel kan du se, hvordan du konfigurerer analyseforespørgslen for et rapportvisual. Start med at åbne filen [Adventure Works DW 2020 M01.pbix](#) Power BI Desktop og derefter følge disse trin:

1. Føj et stablet søjlediagram til rapportsidens.
2. **Filtrer** siden ved hjælp af **Regnskabsår** i tabellen **Dato**, og vælg **FY2020**.
3. **Gruppér** visualiseringen ved at føje **Month** fra tabellen **Date** til **brønden med X-aksen**.
4. **Opsummer** visualiseringen ved at føje **Sales Amount** fra tabellen **Sales** til **brønden på Y-aksen**.



Felter er en fælles betegnelse, der bruges til at beskrive en modelressource, der kan bruges til at konfigurere et visual. De tre forskellige modelressourcer, der er felter, omfatter:

- Kolonner

- Hierarkiniveauer
- Målinger

Hver af disse ressourcetyper kan bruges til at konfigurere et visual, som konfigurerer en analyseforespørgsel i baggrunden. I følgende tabel illustreres det, hvordan du bruger de enkelte modelressourcer.

Udvid tabel

Modelressource	Filtrer	Gruppér	Opsummer
Kolonne	X	X	X
Hierarkiniveau	X	X	
Måling	X		X

Kolonner

Brug kolonner til at filtrere, gruppere og opsummere kolonneværdier. Opsummering af numeriske kolonner er almindelig, og det kan gøres ved hjælp af sum, antal, distinkt antal, minimum, maksimum, gennemsnit, median, standardafvigelse eller varians. Du kan også opsummere tekstkolonner ved hjælp af først (alfabetisk rækkefølge), sidste, antal eller distinkt antal. Derudover kan du opsummere datokolonner ved hjælp af tidligste, seneste, antal eller distinkt antal.

Datakonstruktøren kan angive standardegenskaben for opsummering på designtidspunktet. Denne egenskab kan angives til en af de understøttede opsummeringstyper eller **Opsummer ikke**. Denne indstilling betyder, at kolonnen som standard kun skal bruges til at gruppere. Hvis din semantiske model har en numerisk kolonne, der gemmer årsværdierne, er det hensigtsmæssigt at angive standardopsummering til **Opsummer ikke**, fordi kolonnen sandsynligvis kun bruges til gruppering eller filtrering, og denne numeriske opsummering af år, f.eks. et gennemsnit, ikke giver et meningsfuldt resultat.

Hierarkiniveauer

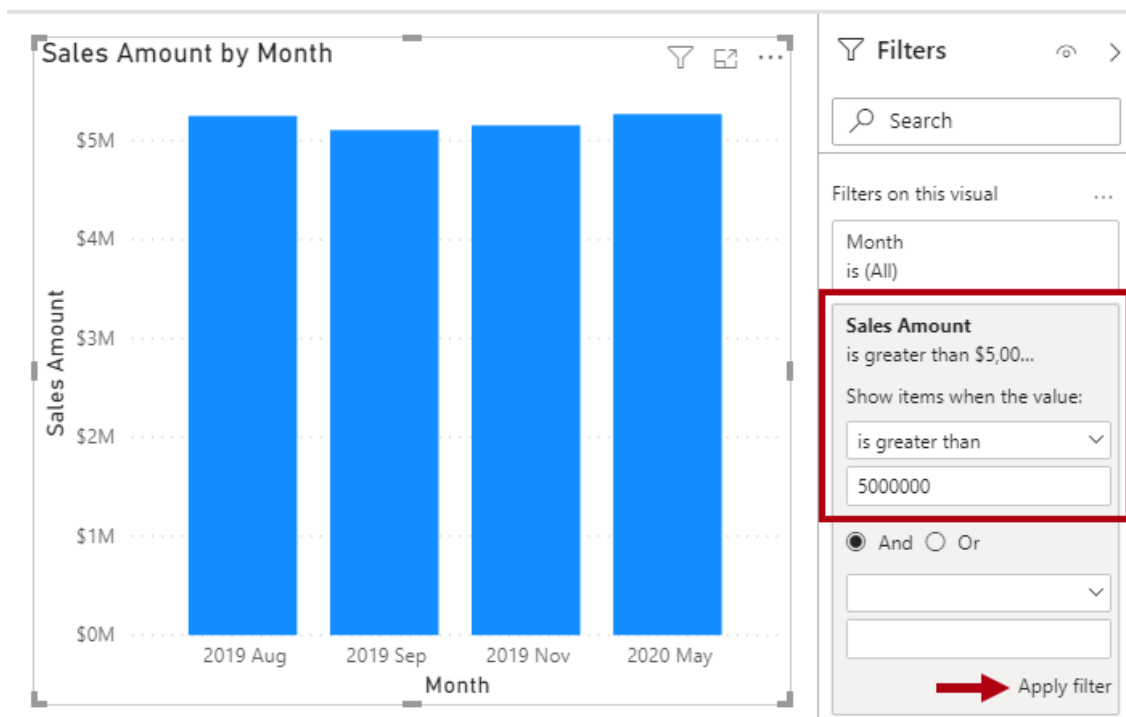
Hierarkiniveauer er baseret på kolonner, og de kan bruges til at filtrere og gruppere, men ikke til at opsummere. Rapportforfattere kan opsummere den kolonne, hierarkiniveauet er baseret på, hvis den er synlig i ruden **Felter**.

Målinger

Målinger er designet til at opsummere modeldata. De kan ikke bruges til at gruppere data. Målinger kan dog bruges til at filtrere data i ét særligt tilfælde: Hvis du vil bruge en måling til at filtrere et visual, når målingen vises i visual'et, og filteret er et filter på visualniveau (dvs. ikke en rapport eller et filter på sideniveau). På denne måde anvendes et målingsfilter, *efter* at analyseforespørgslen har opsummeret data. Denne proces gøres for at fjerne grupper, hvor målingsfilterbetingelsen ikke er sand. (For dem, der kender SQL-syntaksen, er en måling, der bruges til at filtrere en visualisering, som delsætningen HAVING i en SELECT sætning).

Følgende figur viser visual'et i form af det stablede søjlediagram, der er justeret til at vise grupper (måneder), når salgsbeløbene overstiger 5 millioner USD. Denne justering udføres i ruden **Filtre** ved at anvende et filter i feltet **Salgsbeløb**: Konfigurer filteret til at vise elementer, når værdien *er større end* 5.000.000. Husk at vælge **Anvend filter**, der er placeret i nederste højre hjørne af kortet.

Bemærk, at kun fire grupper (måneder) har salgsbeløb, der overstiger 5 millioner USD.



Næste undermodul: Tjek din viden

[Tidligere Næste](#)

Har du brug for hjælp? Se i vores [fejlfindingsvejledning](#), eller angiv specifik feedback ved at [rapportere et problem](#).

1.

Hvilken type objekt opretter du i et Power BI Desktop modeldesign for at oprette forbindelse til flere tabeller?

Relation

Korrekt. En relation forbinder to tabeller for at tillade filtrering og yderligere analyse.

Kolonne

Måling

2.

Hvilket af følgende udsagn om design af et stjerneschema er korrekt?

Faktatabeller lagrer akkumuleringer af forretningshændelser.

Korrekt. Faktatabeller gemmer akkumuleringer af forretningshændelser, f. eks. salgsordrer eller valutakurser.

Faktatabeller lagrer akkumuleringer af forretningsenheder.

Faktatabeller skal have en entydig kolonne.

3.

I hvilken rækkefølge implementerer en analyseforespørgsel sine faser?

Filtrer, Gruppér og Opsummer

Korrekt. Filtrer bruges først for at begrænse de data, der forespørges. Gruppér opdeler derefter forespørgselsresultatet i grupper. Opsummer opretter derefter sammenlægninger af enkeltværdier for hver gruppe.

Gruppér, Filtrer, Opsummer

Opsummer, Filtrer, Gruppér

*** Choose a Power BI model framework

- 46 min.
- Module
- 8 Units

Feedback

Intermediate

Data Analyst

Power BI

Describe model frameworks, their benefits and limitations, and features to help optimize your Power BI data models.

Learning objectives

By the end of this module, you'll be able to:

- Describe Power BI model fundamentals.
- Determine when to develop an import model.
- Determine when to develop a DirectQuery model.

- Determine when to develop a composite model.
- Choose an appropriate Power BI model framework.

[Start](#)Tilføj

-
-
-

Prerequisites

- Experience developing Power BI data models, reports, and dashboards.

Dette modul er en del af disse læringsforløb

- [Modeldata med Power BI](#)
- [Prepare data for tabular models in Power BI](#)

Modulvurdering

Besvar {percent}% af spørgsmålene korrekt for at kunne bestå.

[Tag modulvurderingen](#)

- [Introduction](#)3 min.
- [Describe Power BI model fundamentals](#)9 min.
- [Determine when to develop an import model](#)7 min.
- [Determine when to develop a DirectQuery model](#)9 min.
- [Determine when to develop a composite model](#)7 min.
- [Choose a model framework](#)3 min.
- [Knowledge check](#)5 min.
- [Summary](#)3 min.

Introduction

Completed100 XP

- 3 minutes

For over two decades, Microsoft continues to make deep investments in enterprise business intelligence (BI). Azure Analysis Services (AAS) and SQL Server Analysis Services (SSAS) are based on mature BI data modeling technology used by countless enterprises. The same technology is also at the heart of Power BI data models.

Power BI offers you a choice when designing your model. **You can use Power BI Desktop to develop your model, and you can develop it by using different frameworks.** These frameworks help to deliver fast performance, near real-time results, or both.

This module introduces the frameworks, their benefits and limitations, and features to help optimize your models. Lastly, it provides you with guidance to help you choose the right framework and features for your project.

Learning objectives

By the end of this module, you'll be able to:

- Describe Power BI model fundamentals.
- Determine when to develop an import model.
- Determine when to develop a DirectQuery model.
- Determine when to develop a composite model.
- Choose an appropriate Power BI model framework.

Næste undermodul: Describe Power BI model fundamentals

[Næste](#)

Describe Power BI model fundamentals

Completed100 XP

- 9 minutes

This unit introduces Power BI model terms. It's important that you understand these terms in order to choose the appropriate model framework for your project. This unit describes the following terms:

- Data model
- Power BI dataset
- Analytic query
- Tabular model
- Star schema design
- Table storage mode
- Model framework

Data model

A Power BI data model is a query-able data resource that's optimized for analytics. Reports can query data models by using one of two analytic languages: Data Analysis Expressions (DAX) or Multidimensional

Expressions (MDX). Power BI uses DAX, while paginated reports can use either DAX or MDX. The Analyze in Excel features uses MDX.

Tip

A data model is also described as semantic model, especially in enterprise scenarios. Commonly, in the context of a data discussion and in this module, a data model is simply referred to as a model.

Power BI dataset

You develop a Power BI model in Power BI Desktop, and once published to a workspace in the Power BI service, it's then known as a dataset. A dataset is a Power BI artifact that's a source of data for visualizations in Power BI reports and dashboards.

Bemærk

Not all datasets originate from models developed in Power BI Desktop. Some datasets represent connections to external-hosted models in AAS or SSAS. Others can represent real-time data structures, including push datasets, streaming datasets, or hybrid datasets. This module is concerned only with models developed in Power BI Desktop.

Analytic query

Power BI reports and dashboards must query a dataset. When Power BI visualizes dataset data, it prepares and sends an analytic query. An analytic query produces a query result from a model that's easy for a person to understand, especially when visualized.

An analytic query has three phases that are executed in this order:

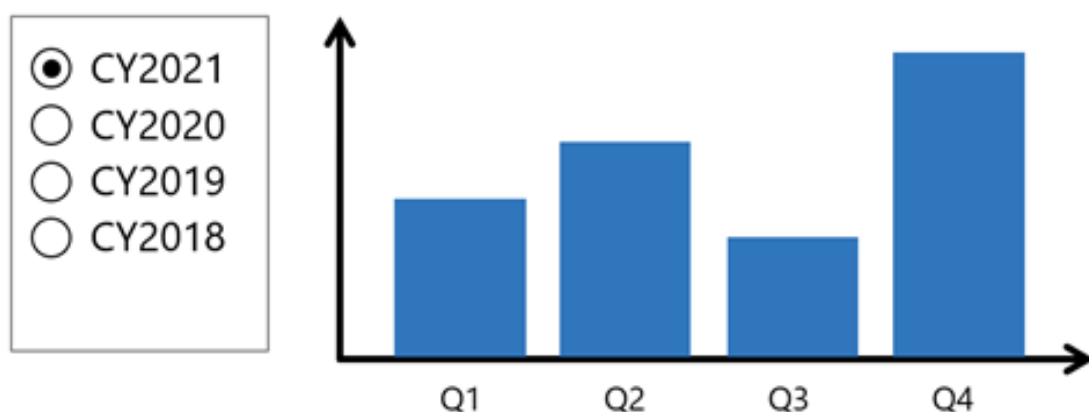
1. Filter
2. Group
3. Summarize

Filtering (sometimes known as slicing) narrows down on a subset of the model data. Filter values aren't visible in the query result. Most analytic queries apply filters because it's common to filter by a time period, and usually other attributes. Filtering happens in different ways. In a Power BI report, you can set filters at report, page, or visual level. Report layouts often include slicer visuals to filter visuals on the report page. When the model enforces row-level security (RLS), it applies filters to model tables to restrict access to specific data. Measures, which summarize model data, can also apply filters.

Grouping (sometimes known as dicing) divides query result into groups. Each group is also a filter, but unlike the filtering phase, filter values are visible in the query result. For example, grouping by customer filters each group by customer.

Summarization produces a single value result. Typically, a report visual summarizes a numeric field by using an aggregate function. Aggregate functions include sum, count, minimum, maximum, and others. You can achieve simple summarization by aggregating a column, or you can achieve complex summarization by creating a measure using a DAX formula.

Consider an example: A Power BI report page includes a slicer to filter by a single year. There's also a column chart visual that shows quarterly sales for the filtered year.



In this example, the slicer **filters** the visual by calendar year 2021. The column chart **groups** by quarters (of the filtered year). Each column is a group that represents a visible filter. The column heights represent the **summarized** sales values for each quarter of the filtered year.

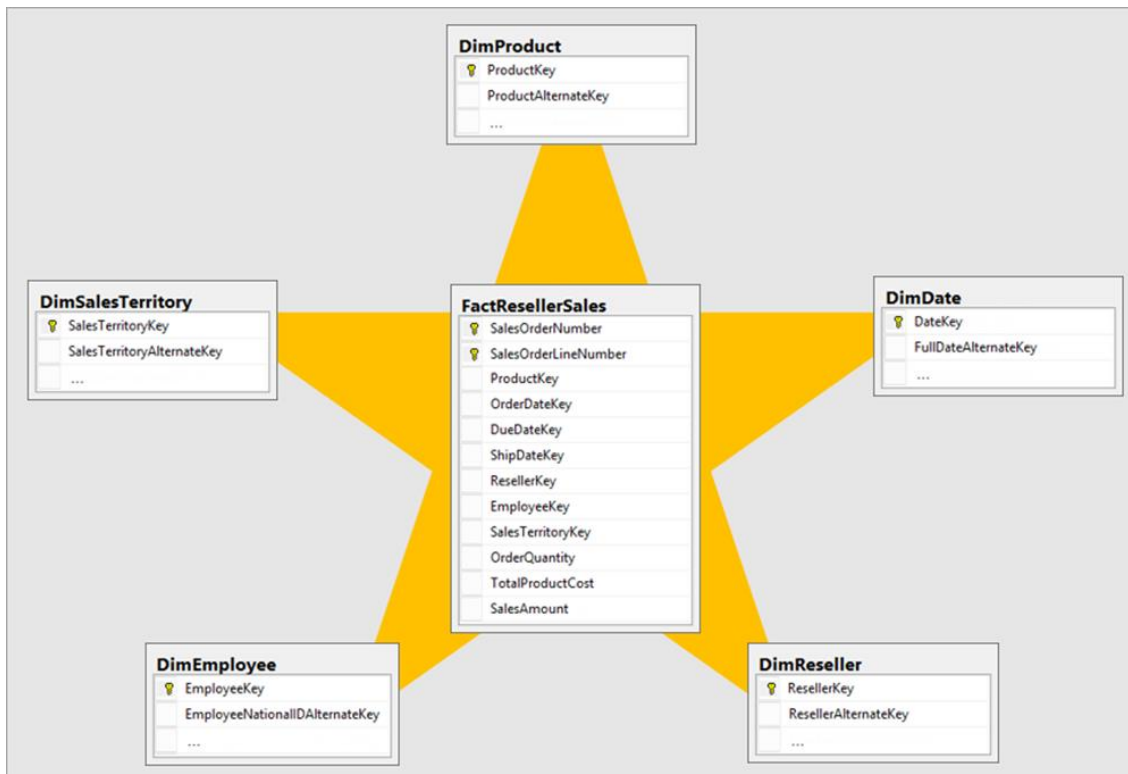
Tabular model

A Power BI model is a tabular model. A tabular model comprises one or more tables of columns. It can also include relationships, hierarchies, and calculations.

Star schema design

To produce an optimized and easy-to-use tabular model, we recommend you produce a star schema design. Star schema design is a mature modeling approach widely adopted by relational data warehouses. It requires you to classify model tables as either dimension or fact.

Dimension tables describe business entities; the things you model. Entities can include products, people, places, and concepts including time itself. Fact tables store observations or events, and can be, for example, sales orders, stock balances, exchange rates, or temperature readings. A fact table contains dimension key columns that relate to dimension tables, and numeric measure columns. A fact table forms the center of a star, and the related dimension tables form the points of the star.



In an analytic query, dimensions table columns filter or group. Fact table columns are summarized.

For more information, see [Understand star schema and the importance for Power BI](#).

Table storage mode

Each Power BI model table (except calculated tables) has a storage mode property. The storage mode property can be either **Import**, **DirectQuery**, or **Dual**, and it determines whether table data is stored in the model.

- **Import** – Queries retrieve data that's stored, or cached, in the model.
- **DirectQuery** – Queries pass through to the data source.
- **Dual** – Queries retrieve stored data or pass through to the data source. Power BI determines the most efficient plan, striving to use cached data whenever possible.

Model framework

Table storage mode settings determine the model framework, which can be either import, DirectQuery, or composite. The following units in this module describe each of these frameworks and provides guidance on their use.

- An import model comprises tables that have their storage mode property set to **Import**.
- A DirectQuery model comprises tables that have their storage mode property set to **DirectQuery**, and they belong to the same source group. Source group is described later in this module.
- A composite model comprises more than one source group.

Næste undermodul: Determine when to develop an import model

[Tidligere](#)

Determine when to develop an import model, alternativt Direct Query

100 XP

- 7 minutes

An import model comprises tables that have their storage mode property set to **Import**. It includes calculated tables, which you define with a DAX formula, too.

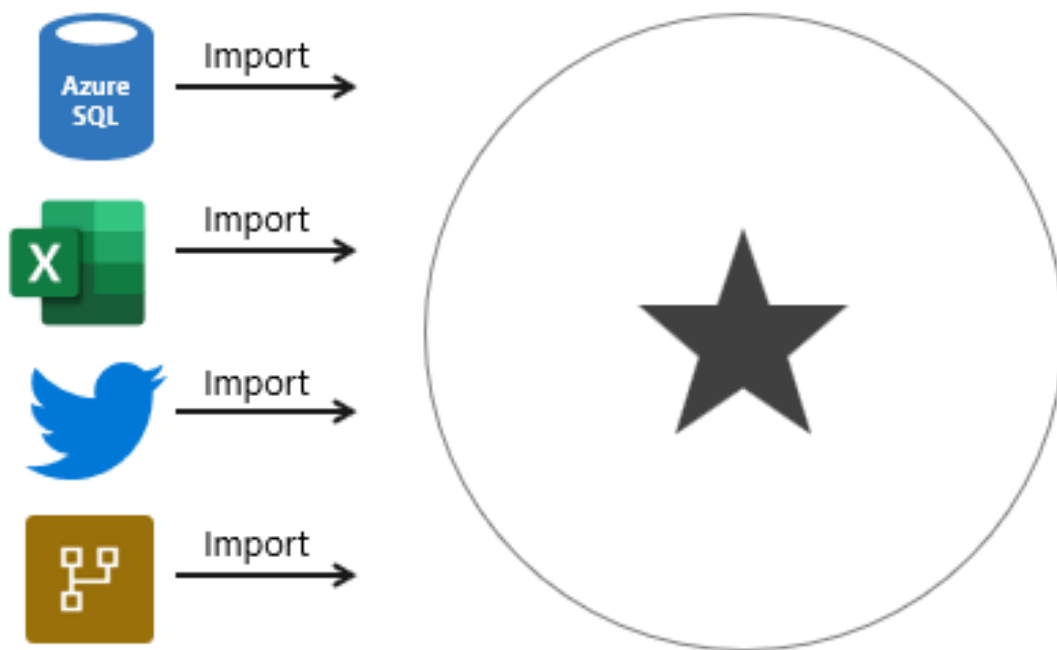
Import model benefits

Import models are the most frequently developed model framework because there are many benefits.

Import models:

- Support all Power BI data source types, including databases, files, feeds, web pages, dataflows, and more.
- Can integrate source data. For example, one table sources its data from a relational database while a related table sources its data from a web page.
- Support all DAX and Power Query (M) functionality.
- Support calculated tables.
- **Deliver the best query performance. That's because the data cached in the model is optimized for analytic queries** (filter, group, and summarize) and the model is stored entirely in memory.

In short, import models offer you the most options and design flexibility, and they deliver fast performance. For this reason, Power BI Desktop defaults to use import storage mode when you "Get data."



Import model limitations: max 10GB, opdateres få gange dagligt

Despite the many compelling benefits, there are limitations of import models that you must bear in mind. Limitations are related to **model size** and **data refresh**.

Model size

Power BI imposes dataset size restrictions, which limit the size of a model. **When you publish the model to a shared capacity, there's a 1-GB limit per dataset.** When this size limit is exceeded, the dataset will fail to refresh. When you publish the model to a dedicated capacity (also known as Premium capacities), it can grow beyond 10-GB, providing you enable the [Large dataset storage format setting](#) for the capacity.

You should always strive to reduce the amount of data stored in tables. This strategy helps to reduce the duration of model refreshes and speed up model queries. There are numerous data reduction techniques that you can apply, including:

- Remove unnecessary columns
- Remove unnecessary rows
- Group by and summarize to raise the grain, mere grovkornet opløsning, of fact tables
- Optimize column data types with a preference for numeric data
- Preference for custom columns in Power Query instead of calculated columns in the model
- Disable Power Query query load
- Disable auto date/time
- Use DirectQuery table storage, as described in later units of this module.

For more information, see [Data reduction techniques for Import modeling](#).

Bemærk

The 1-GB per dataset limit refers to the compressed size of the Power BI model, not the volume of data being collected from the source system.

Data refresh: Imported data must be periodically refreshed, kan kun ske få gange dagligt
Imported data must be periodically refreshed. Dataset data is only as current as the last successful data refresh. To keep data current, you set up scheduled data refresh, or report consumers can perform an on-demand refresh.

Power BI imposes limits on how often scheduled refresh operations can occur. **It's up to eight times per day in a shared capacity, and up to 48 times per day in a dedicated capacity.**

You should determine whether this degree of latency is tolerable. It often depends on the velocity (or volatility) of the data, and the urgency to keep users informed about the current state of data. **When scheduled refresh limits aren't acceptable, consider using DirectQuery storage tables, or creating a hybrid table. Or take a different approach, and create a real-time dataset instead.**

Tip

Hybrid tables are described in [unit 4](#). For information about real-time datasets, work through the [Monitor data in real-time with Power BI](#) module.

You must also consider refresh workload and duration. **By default, to refresh a table, Power BI removes all data and reloads it again. These operations can place an unacceptable burden on source systems, especially for large fact tables.** To reduce this burden, you can set up the incremental refresh feature. Incremental refresh automates the creation and management of time-period partitions, and intelligently update only those partitions that require refresh.

When your data source supports incremental refresh, it can result in faster and more reliable refreshes, and reduced resource consumption of Power BI and source systems.

Advanced data modelers can customize their own partitioning strategy. Automation scripts can create, manage, and refresh table partitions. For more information, see [Power BI usage scenarios: Advanced data model management](#). This usage scenario describes using the XMLA endpoint available with Power BI Premium.

Determine when to develop a DirectQuery model

Completed100 XP

- 9 minutes

A DirectQuery model comprises tables that have their storage mode property set to **DirectQuery**, and they belong to the same source group.

A source group is a set of model tables that relate to a data source. There are two types:

- **Import** – Represents all import storage mode tables including calculated tables. There can only be one import source group in a model.
- **DirectQuery** – Represents all DirectQuery storage mode tables that relate to a specific data source.

Bemærk

An import model and a DirectQuery model only comprise a single source group. **When there's more than one source group, the model framework is known as a composite model.** Composite models are described in [Unit 5](#).

DirectQuery model benefits

There are several benefits to developing a DirectQuery model.

Model large or fast-changing data sources(data er altid opdateret)

A DirectQuery model is a great framework choice when your source data exhibits volume and/or velocity characteristics. **Because DirectQuery tables don't require refresh, they're well suited to large data stores,** like a data warehouse. It's impractical and inefficient, if not impossible, to import an entire data warehouse into a model. When the source data changes rapidly and users need to see current data, a DirectQuery model can deliver near real-time query results.

When a report queries a DirectQuery model, Power BI passes those queries to the underlying data source.



Enforce source RLS

DirectQuery is also useful when the source database enforces row-level security (RLS). Instead of replicating RLS rules in your Power BI model, the source data base can enforce its rules. This approach works only for some relational databases, and it involves setting up single sign-on for the dataset data source. For more information, see [Azure SQL Database with DirectQuery](#).

Data sovereignty restrictions

If your organization has security policies that restrict data leaving their premises, then it isn't possible to import data. A DirectQuery model that connects to an on-premises data source may be appropriate. (You can also consider installing [Power BI Report Server](#) for on-premises reporting.)

Create specialized datasets

Typically, DirectQuery mode supports relational database sources. That's because Power BI must translate analytic queries to native queries understood by the data source.

However, there's one powerful exception. You can connect to a Power BI dataset (or Azure Analysis Services model) and convert it to a DirectQuery local model. A local model is a relative term that describes a model's relationship to another model. In this case, the original dataset is a remote model, and the new dataset is the local model. These models are chained, which is a term used to describe related models. You can chain up to three models in this way.

This capability to chain models supports the potential to personalize and/or extend a remote model. The simplest thing you can do is rename objects, like tables or columns, or add measures to the local model. You can also extend the model with calculated columns or calculated tables, or add new import or DirectQuery tables. However, these extensions result in the creation of new source groups, which means the model becomes a composite model. That scenario is described in [Unit 5](#).

For more information, see [Using DirectQuery for Power BI datasets and Azure Analysis Services](#).

DirectQuery model limitations

There are many limitations related to DirectQuery models that you must bear in mind. Here are the main limitations:

- Not all data sources are supported. Typically, only major relational database systems are supported. Power BI datasets and Azure Analysis Services models are supported too.
- All Power Query (M) transformations are not possible, because these queries must translate to native queries that are understood by source systems. So, for example, it's not possible to use pivot or unpivot transformations.
- Analytic query performance can be slow, especially if source systems aren't optimized (with indexes or materialized views), or there are insufficient resources for the analytic workload.
- Analytic queries can impact on source system performance. It could result in a slower experience for all workloads, including OLTP operations.

Boost DirectQuery model performance

When there's a justification to develop a DirectQuery model, you can mitigate some limitations in two ways.

Data source optimizations

You can optimize the source database to ensure the expected analytic query workload performs well. Specifically, you can create indexes and materialized views, and ensure the database has sufficient resources for all workloads.

Tip

We recommend that you always work in collaboration with the database owner. It's important that they understand the additional workload a DirectQuery model can place on their database.

DirectQuery user-defined aggregation tables

You can add user-defined aggregation tables to a DirectQuery model. User-defined aggregation tables are special model tables that are hidden (from users, calculations, and RLS). They work best when they satisfy higher-grain, en grovkornet opløsning, analytic queries over large fact tables. When you set the aggregation table to use DirectQuery storage mode, it can query a materialized view in the data source. **You can also set an aggregation table to use import storage mode or enable automatic aggregations**, and these options are described in Unit 4.

For more information, see [DirectQuery model guidance in Power BI Desktop](#).

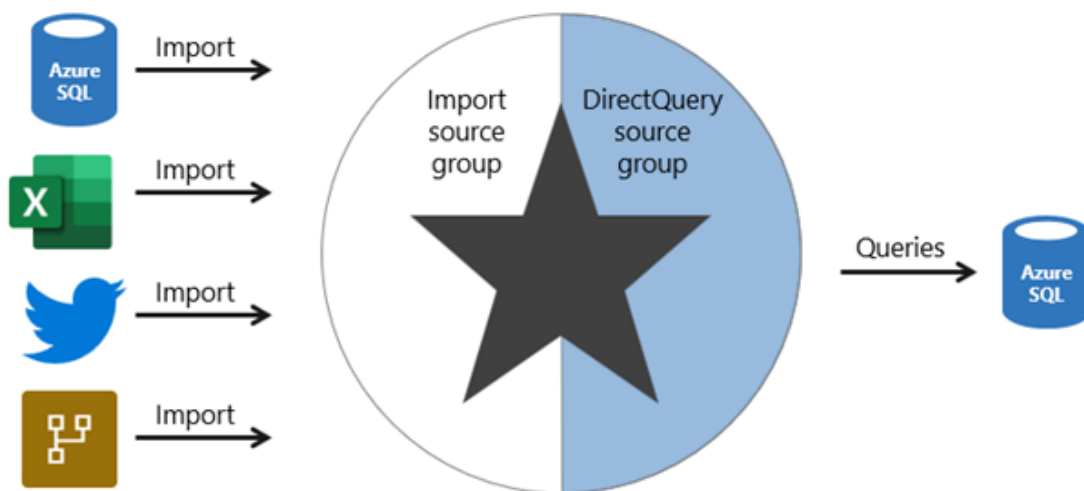
Næste undermodul: Determine when to develop a composite model

Determine when to develop a composite model

Completed 100 XP

- 7 minutes

A composite model comprises more than one source group. Typically, there's always the import source group and a DirectQuery source group.



Bemærk

Generally, the benefits and limitations associated with import and DirectQuery storage modes apply to composite models.

Composite model benefits

There are several benefits to developing a composite model.

Composite models provide you with design flexibility. You can choose to integrate data using different storage modes, striking the right balance between imported data and pass-through data. Commonly, enterprise models benefit from using DirectQuery tables on large data sources and by boosting query performance with imported tables. Power BI features that support this scenario are described later in this unit.

Composite models can also boost the performance of a DirectQuery model by providing Power BI with opportunity to satisfy some analytic queries from imported data. Querying cached data almost always performs better than pass-through queries.

Lastly, when your model includes DirectQuery tables to a remote model, like a Power BI dataset, you can extend your model with new calculated columns and tables. It results in a specialized model based on a core model. For more information, see [Power BI usage scenarios: Customizable managed self-service BI](#).

Composite model limitations

There are several limitations related to composite models.

- Import (or dual, as described later) storage mode tables still require periodic refresh. Imported data can become out of sync with DirectQuery sourced data, so it's important to refresh it periodically.
- When an analytic query must combine imported and DirectQuery data, Power BI must consolidate source group query results, which can impact performance. To help avoid this situation for higher-grain queries, you can add import aggregation tables to your model (or enable automatic aggregations) and set related dimension tables to use dual storage mode. This scenario is described later in this unit.
- When chaining models (DirectQuery to Power BI datasets), modifications made to upstream models can break downstream models. Be sure to assess the impact of modifications by performing [dataset impact analysis](#) first.
- Relationships between tables from different source groups are known as limited relationships. A model relationship is limited when the Power BI can't determine a "one" side of a relationship. Limited relationships may result in different evaluations of model queries and calculations. For more information, see [Relationship evaluation](#).

Boost DirectQuery model performance with import data

When there's a justification to develop a DirectQuery model, you can mitigate some limitations by using specific Power BI features that involve import tables.

Import aggregation tables

You can add import storage mode user-defined aggregation tables or enable automatic aggregations. This way, Power BI directs higher-grain, mere grovkornet, fact queries to a cached aggregation. To boost query performance further, ensure that related dimension tables are set to use dual storage mode.

Automatic aggregations are a Premium feature. For more information, see [Automatic aggregations](#).

Dual storage mode

A dual storage mode table is set to use both import and DirectQuery storage modes. At query time, Power BI determines the most efficient mode to use. Whenever possible, Power BI attempts to satisfy analytic queries by using cached data.

Dual storage mode tables work well with import aggregation tables. They allow Power BI to satisfy higher-grain queries entirely from cached data.

Slicer visuals and filter card lists, which are often based on dimension table columns, render more quickly because they're queried from cached data.

Deliver real-time data from an import model

When you set up an import table with incremental refresh, you can enable the **Get the latest data in real-time with DirectQuery** option.

Incremental refresh and real-time data

Refresh large tables faster with incremental refresh. Plus, get the latest data in real time with DirectQuery (Premium only). [Learn more](#)

ⓘ These settings will apply when you publish the dataset to the Power BI service. Once you do that, you won't be able to download it back to Power BI Desktop. [Learn more](#)

1. Select table

FactInternetSales

2. Set import and refresh ranges

☒

 Incrementally refresh this table

Archive data starting

5

Years

 before refresh date

Data imported from 12/21/2016 to 12/18/2021.

Incrementally refresh data starting

3

Days

 before refresh date

Data will be incrementally refreshed from 12/18/2021 to 12/21/2021.

3. Choose optional settings

☒

 Get the latest data in real time with DirectQuery (Premium only) [Learn more](#)

By enabling this option, Power BI automatically creates a table partition that uses DirectQuery storage mode. In this case, the table becomes a hybrid table, meaning it has import partitions to store older data, and a single DirectQuery partition for current data.

When Power BI queries a hybrid table, the query uses the cache for older data, and passes through to the data source to retrieve current data.

This option is only available with a Premium license.

For more information, see [Configure incremental refresh and real-time data](#).

Næste undermodul: Choose a model framework

[Tidligere](#)

Choose a model framework

Completed 100 XP

- 3 minutes

Here's some general guidance about choosing the appropriate model framework for your project. **It especially applies to enterprise solutions, where data volumes are large, query throughput is high, and fast responsiveness is important.**

Most importantly, choose the import model framework whenever possible. This framework offers you the most options, design flexibility, and delivers fast performance. Be sure to apply data reduction techniques to ensure that Power BI loads the least amount of data possible.

Choose the DirectQuery model framework when your data source stores large volumes of data and/or your report needs to deliver near real-time data.

Choose the composite model framework to:

- Boost the query performance of a DirectQuery model.
- Deliver near real-time query results from an import model.
- Extend a Power BI dataset (or AAS model) with additional data.

You can boost the query performance of a DirectQuery model by using aggregation tables, which can use import or DirectQuery storage mode. When using import aggregation tables, be sure to set related dimension tables to use dual storage mode. That way, Power BI can satisfy higher-grain queries entirely from cache.

You can deliver near real-time query results from in import model by creating a hybrid table. In this case, Power BI adds a DirectQuery partition for the current period.

Lastly, you can create specialized models by chaining to a core model by using DirectQuery. This type of development is typically done by a business analyst who extends core models, which IT delivers and supports.

Vigtigt

Plan carefully. In Power BI Desktop, it's always possible to convert a DirectQuery table to an import table. But it's not possible to convert an import table to a DirectQuery table.

Næste undermodul:

Knowledge check

1.

Geoffrey is a data modeler at Adventure Works who **developed a DirectQuery model** that connects to the data warehouse. To improve the query performance of higher-grain sales queries, Geoffrey added an import aggregation table. What else should Geoffrey do to improve query performance of the higher-grain queries?

Set related dimension tables as aggregation tables.

Set related dimension tables to dual storage mode.

Correct. Setting the related dimension tables to dual storage mode will allow Power BI to satisfy higher-grain queries entirely from cache.

Set related dimension tables to import storage mode.

Incorrect. Setting the related dimension tables to import storage mode will allow Power BI to satisfy higher-grain queries entirely from cache. However, lower-grain queries will take longer to process because they can't be satisfied entirely from the data source.

2.

Breana is a data modeler at Adventure Works who developed a manufacturing model, which **is an import model**. Breana needs to ensure that manufacturing reports **deliver real-time results**. Which type of table should Breana create?

Aggregation table.

Incorrect. An aggregation table supports fast higher-grain queries.

Hybrid table.

Correct. A hybrid table includes a DirectQuery partition for the current period to deliver near-real time results.

Partitioned table.

3.

Mousef is a business analyst at Adventure Works who wants to create a new model by extending the sales dataset, which is delivered by IT. Mousef wants to add a new table of census population data sourced from a web page. Which model framework should Mousef use?

Composite.

Correct. A composite model would comprise a DirectQuery source group containing the sales dataset tables, and an import source group containing the imported web page data.

DirectQuery.

Live connection.

<https://learn.microsoft.com/api/achievements/share/da-dk/bjarnehansen-7412/9XK2KV7U?sharingId=39714CB1FE13C67B>

Fortsæt det fremragende arbejde



Choose a Power BI model framework

Du har opnået et resultat!

Tillykke, men hvad skal du gøre nu?

Lad os først dele dit resultat

Du afsætter tid til at lære noget nyt, så lad dit netværk tage del i din sejr!



Mist ikke dit momentum, bliv ved med at lære

Nedenfor finder du anbefalet indhold, der kan hjælpe dig på vej!

Næste modul i dette læringsforløb

**** Forbered, importer data til analyse med Power BI

- 3 t. 47 min.
- Læringsforløb
- 0 af 2 moduler er fuldført

Øvet

Dataanalytiker

Power BI

Du får mere at vide om, hvordan du bruger Power Query til at udtrække data fra forskellige datakilder, vælge en lagringstilstand og forbindelsestype. Du lærer også at profilere, rense og indlæse data i Power BI, før du modellerer dine data.

I dette læringsforløb får du hjælp til at forberede dig til [certificeringen Microsoft Certified: Data Analyst Associate](#).

Forudsætninger

Ingen

[Start](#)Tilføj

-
-

Indlæs din kode

Introduktion

Fuldført100 XP

- 2 minutter

Som de fleste af os arbejder du for en virksomhed, hvor du er forpligtet til at oprette Microsoft Power BI-rapporter. Dataene findes i flere forskellige databaser og filer. Disse datalagre er forskellige fra hinanden, nogle findes i Microsoft SQL Server, nogle findes i Microsoft Excel, men alle dataene er relateret til hinanden.

Bemærk

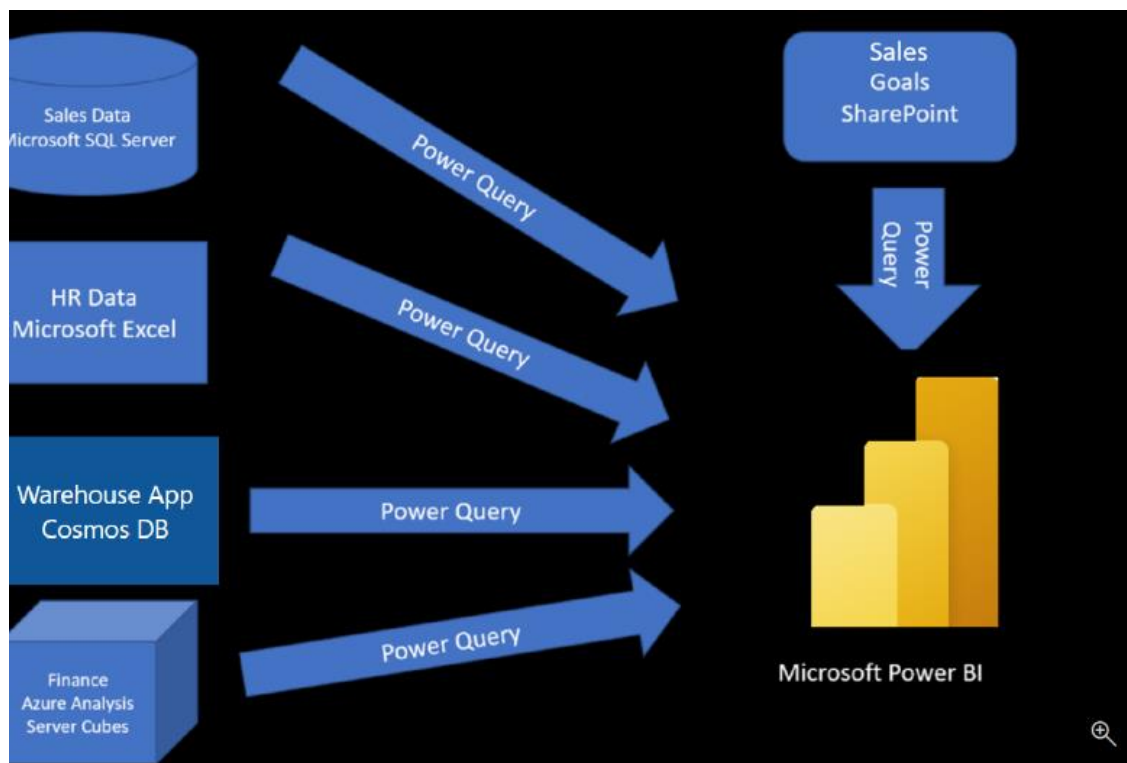
Modulsektionerne før laboratoriet er udelukkende oplysende. Du får mulighed for at arbejde med reelle data i laboratoriet.

I dette moduls scenarie arbejder du for Tailwind Traders. Du har fået til opgave af direktionen at udarbejde en række rapporter, der er afhængige af data på flere forskellige placeringer. Den database, der sporer salgstransaktioner, er i SQL Server, en relationsdatabase, der indeholder, hvilke varer hver kunde har købt, og hvornår. Den sporer også, hvilke medarbejdere der udførte salget sammen med navnet på medarbejderen og medarbejder-id'et. Denne database indeholder dog ikke medarbejderens ansættelsesdato, vedkommendes titel, eller hvem vedkommendes leder er. For at få disse oplysninger skal du have adgang til filer, som HR opbevarer i Excel. Du har gentagne gange bedt dem om at bruge en SQL-database, men de har endnu ikke haft mulighed for at implementere det.

Når en vare sendes, registreres forsendelsen i lagerprogrammet, som er nyt i virksomheden. Udviklerne valgte at gemme data i Cosmos DB som et sæt JSON-dokumenter.

Tailwind Traders har et program, der hjælper med økonomiske projekteringer, så de kan forudsige, hvad deres salg vil være i de kommende måneder og år, baseret på tidligere tendenser. Disse projekteringer

gemmes i Microsoft Azure Analysis Services. Her er en visning af de mange datakilder, du bliver bedt om at kombinere data fra.



Du skal først udtrække data fra forskellige datakilder, før du kan udarbejde rapporter. Interaktion med SQL Server er forskellig fra Excel, så du bør lære nuancerne i begge systemer. Når du har fået indsigt i systemerne, kan du bruge Power Query til at hjælpe dig med at rense dataene, f.eks. omdøbning af kolonner, erstatning af værdier, fjernelse af fejl og kombination af forespørgselsresultater. Power Query er også tilgængelig i Excel. Når dataene er blevet rensat og organiseret, er du klar til at oprette rapporter i Power BI. Til sidst skal du publicere din kombinerede semantiske model og rapporter til Power BI-tjeneste. Herfra kan andre bruge din semantiske model og oprette deres egne rapporter, eller de kan bruge de rapporter, du allerede har bygget. Hvis en anden har bygget en semantisk model, du vil bruge, kan du også oprette rapporter ud fra det!

Dette modul fokuserer på det første trin til at hente dataene fra de forskellige datakilder og importere dem i Power BI ved hjælp af Power Query.

Når du har gennemført dette modul, vil du være i stand til følgende:

- Identificer og opret forbindelse til en datakilde
- Hent data fra en relationsdatabase, såsom Microsoft SQL Server
- Hent data fra en fil, såsom Microsoft Excel
- Hent data fra programmer
- Hent data fra Azure Analysis Services
- Vælg en lagertilstand

- Løs problemer med ydeevnen
- Løs fejl under import af data

Næste undermodul: Hent data fra filer

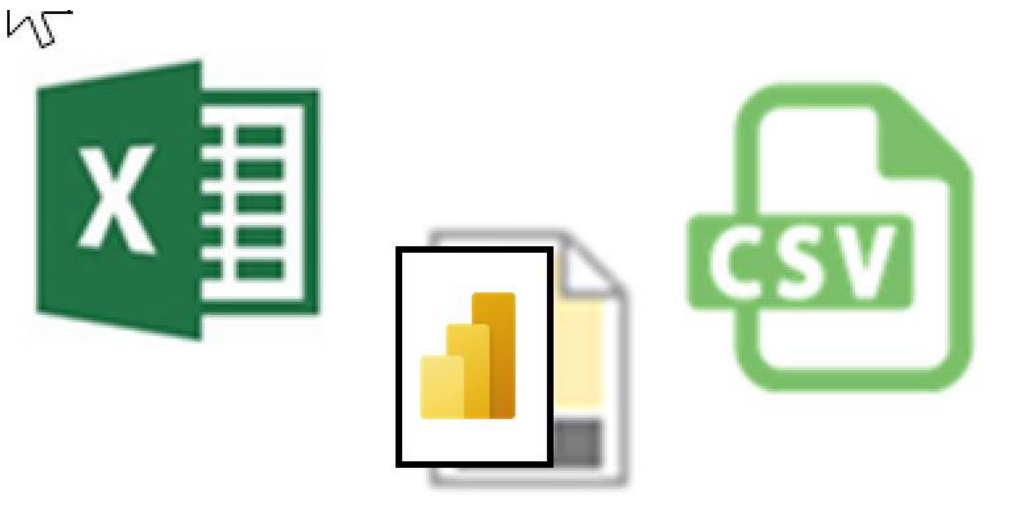
[Næste](#)

Hent data fra filer

Fuldført 100 XP

- 10 minutter

Organisationer eksporterer og gemmer ofte data i filer. Et muligt filformat er en flad fil. En flad fil er en filtype, der kun indeholder én datatabel, og hver række med data har samme struktur. Filen indeholder ikke hierarkier. Du kender sandsynligvis de mest almindelige typer af flade filer, som er filer med kommaseparerede værdier (.csv), afgrænsede tekstfiler (.txt) og filer med fast bredde. En anden filtype er outputfiler fra forskellige programmer, f.eks. Microsoft Excel-projektmapper (.xlsx).



Med Power BI Desktop kan du hente data fra mange filtyper. Du kan finde en liste over de tilgængelige indstillinger, når du bruger funktionen **Hent data** i Power BI Desktop. De efterfølgende afsnit indeholder en beskrivelse af, hvordan du importerer data fra en Excel-fil, der er gemt på en lokal computer.

Scenarie

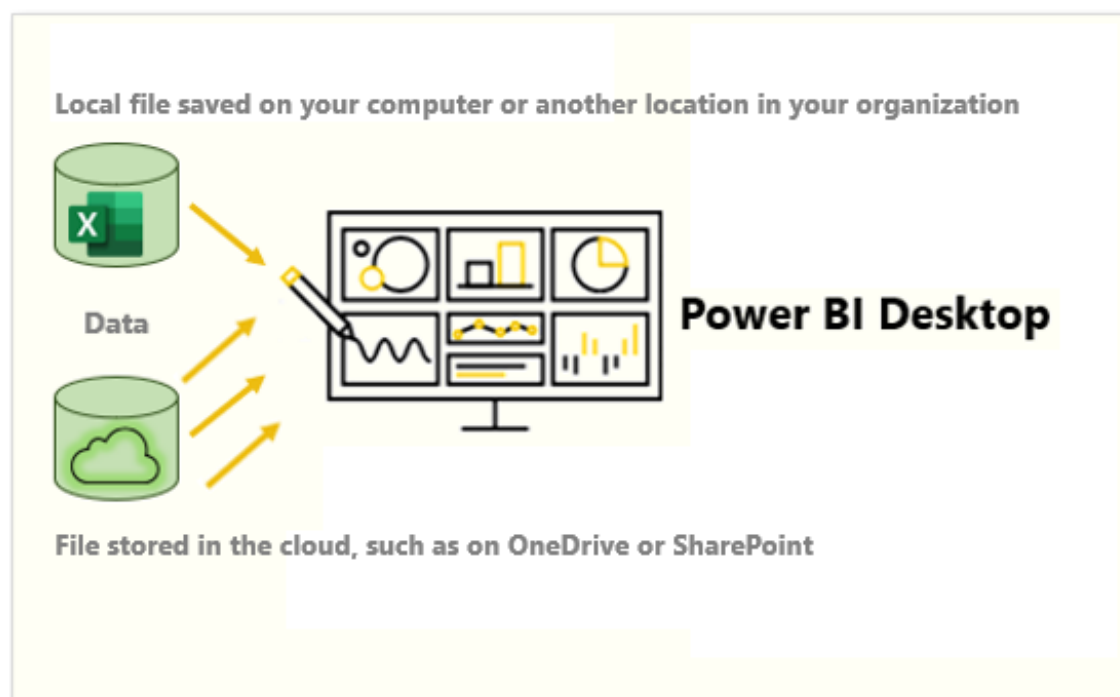
HR-teamet hos Tailwind Traders har forberedt en flad fil, der indeholder nogle af organisationens medarbejderdata, f.eks. navn på medarbejder, ansættelsesdato, stilling og leder. De har bedt dig om at udarbejde Power BI-rapporter ved hjælp af disse data og data, der findes i flere andre datakilder.

Placering af flad fil

Det første trin er at bestemme, hvilken filplacering du vil bruge til at eksportere og gemme dine data.

Dine Excel-filer kan være placeret på en af følgende placeringer:

- **Lokal** – Du kan importere data fra en lokal fil til Power BI. Filen flyttes ikke til Power BI, og der forbliver ikke et link til den. I stedet oprettes der en ny semantisk model i Power BI, og data fra Excel-filen indlæses i den. **Ændringer af den oprindelige Excel-fil afspejles derfor ikke i din semantiske Power BI-model. Du kan bruge import af lokale data til data, der ikke ændres.**
- **OneDrive for Business** – Du kan hente data fra OneDrive for Business til Power BI. Denne metode er effektiv til at holde en Excel-fil og din semantiske model, rapporter og dashboards i Power BI synkroniseret. Power BI opretter regelmæssigt forbindelse til din fil på OneDrive. **Hvis der findes ændringer, opdateres din semantiske model, dine rapporter og dashboards automatisk i Power BI.**
- **OneDrive – Personlig** – Du kan bruge data fra filer på en personlig OneDrive-konto og få mange af de samme fordele som med OneDrive for Business. Du skal dog logge på med din personlige OneDrive-konto og vælge indstillingen **Log mig ikke af**. Kontakt systemadministratoren for at finde ud af, om denne type forbindelse er tilladt i din organisation.
- **SharePoint – Teamwebsteder** – Lagring af dine Power BI Desktop-filer på SharePoint-teamwebsteder svarer til at gemme på OneDrive for Business. Den primære forskel er den måde, som du opretter forbindelse til filen på fra Power BI. Du kan angive en URL-adresse eller oprette forbindelse til rodmappen.



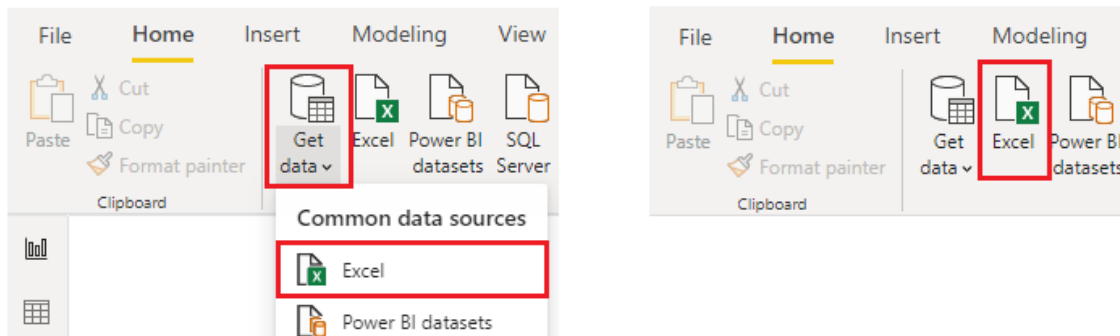
Brug af en cloudmulighed, f.eks. OneDrive eller SharePoint-teamwebsteder, er den mest effektive måde at holde din fil og din semantiske model, rapporter og dashboards synkroniseret på i Power BI. Men hvis dine data ikke ændres regelmæssigt, er det en passende mulighed at gemme filer på en lokal computer.

Opret forbindelse til data i en fil

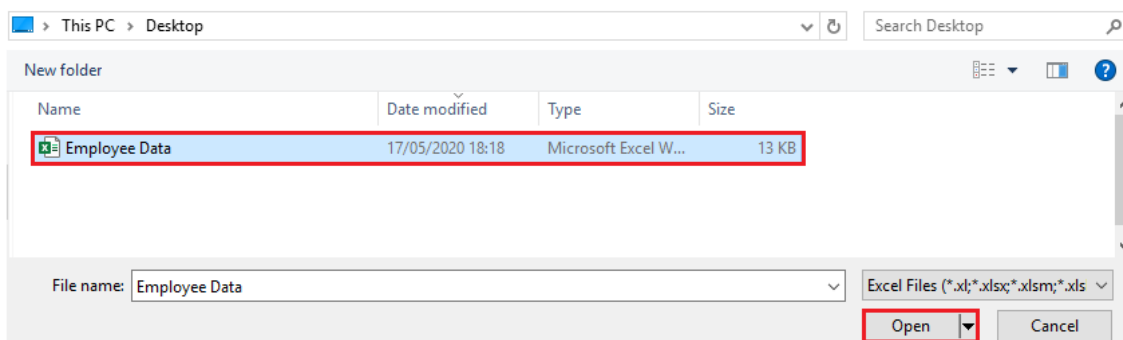
Vælg **Hent data** under fanen **Hjem** i Power BI. På den viste liste skal du vælge den indstilling, du har brug for, f.eks. **Tekst/CSV** eller **XML**. I dette eksempel skal du vælge **Excel**.

Tip

Fanen **Hjem** indeholder indstillinger for hurtig adgang til datakilder, f.eks. **Excel**, ud for knappen **Hent data**.



Afhængigt af dit valg skal du finde og åbne din datakilde. Du bliver muligvis bedt om at logge på en tjeneste, f.eks. OneDrive, for at godkende din anmodning. I dette eksempel skal du åbne **Excel-projektmappen Medarbejderdata**, der er gemt på skrivebordet (Husk, at der ikke er nogen filer til praksis, det er hypotetiske trin).



Vælg de fildata, der skal importeres

Når filen har oprettet forbindelse til Power BI Desktop, åbnes vinduet **Navigator**. I dette vindue kan du se de data, der er tilgængelige i din datakilde (Excel-filen i dette eksempel). Du kan vælge en tabel eller enhed for at få forhåndsvist indholdet og sikre, at de korrekte data indlæses i Power BI-modellen.

Markér afkrydsningsfeltet eller afkrydsningsfelterne for den eller de tabeller, som du vil overføre til Power BI. Dette valg aktiverer knapperne **Indlæs** og **Transformér data** som vist på følgende billede.

Navigator

The screenshot shows the Power BI Navigator interface. On the left, under 'Display Options', the file 'Employee Data.xlsx [1]' is listed, and the 'Employee Data' table is selected and highlighted with a red box. On the right, the 'Employee Data' table is displayed with the following data:

Department	Extension	Position Title	Join Date	Experience
MARKETING	425	Marketing Advisor	01/01/2019	2 years
MARKETING	206	Marketing Advisor	01/03/2018	3 years
MARKETING	207	Brand Manager	01/01/2015	6 years
MARKETING	349	Senior Brand Manager	04/10/2010	10 years
MARKETING	425	Marketing - Coordinator	05/02/2019	2 years
MARKETING	210	Marketing - Coordinator	06/02/2019	2 years
MARKETING	208	Marketing Consulatant	07/05/2015	6 years
MARKETING	249	Marketing Consulatant	08/02/2015	6 years
OPERATIONS	425	Supervisor	09/01/2010	11 years
OPERATIONS	216	Administrator	10/06/2018	2 years
OPERATIONS	215	Operations Manager	11/04/2015	6 years
OPERATIONS	350	Senior Finance Manager	12/05/2010	11 years

At the bottom right of the window, there are three buttons: 'Load' (highlighted in yellow), 'Transform Data', and 'Cancel'.

Nu kan du vælge knappen **Indlæs** for automatisk at indlæse dine data i Power BI-modellen eller vælge knappen **Transformér data** for at starte Power Query-editor, hvor du kan gennemse og rense dine data, før du indlæser dem i Power BI-modellen.

Vi anbefaler ofte, at du transformerer data, men denne proces bliver drøftet senere i dette modul. I dette eksempel kan du vælge **Indlæs**.

Skift kildefil

Du skal måske ændre placeringen af en kildefil for en datakilde under udviklingen, eller hvis placeringen af et fillager ændres. For at holde dine rapporter opdateret skal du opdatere filstierne til forbindelse til Power BI.

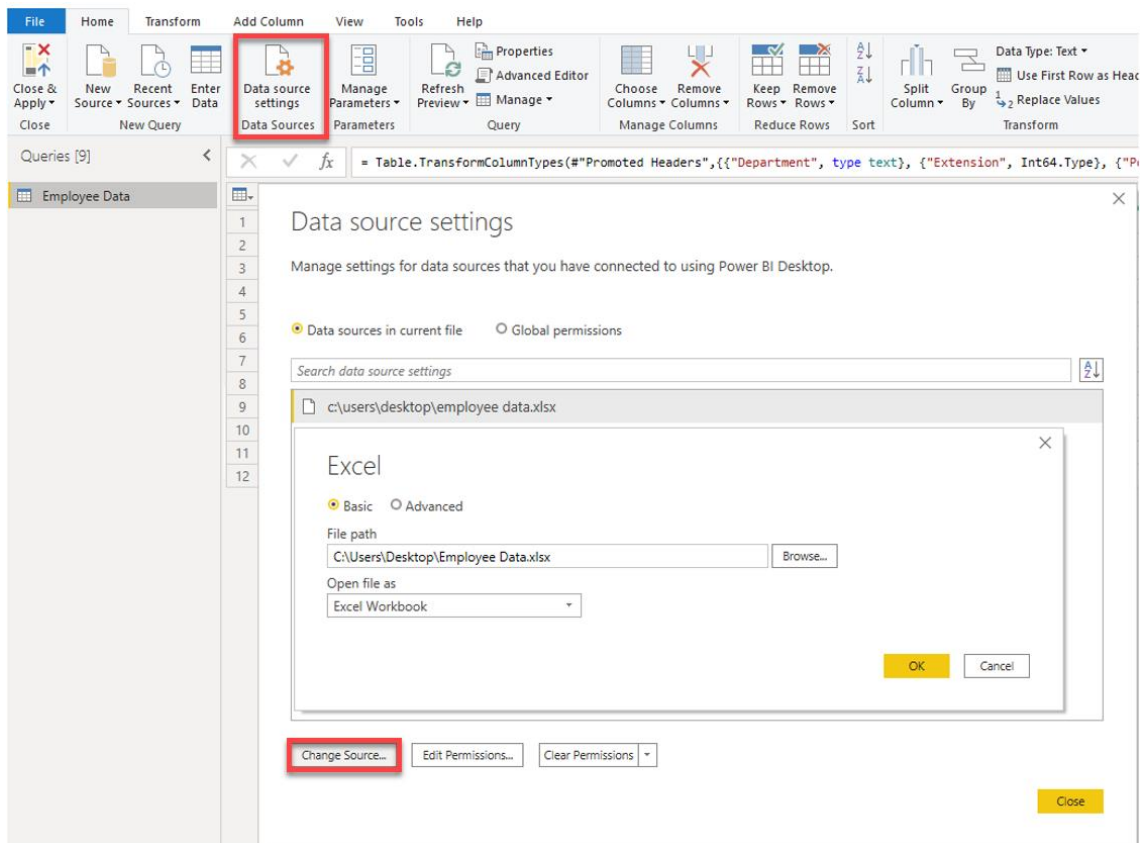
Power Query giver dig mange måder at udføre denne opgave på, så du kan foretage denne type ændring, når det er nødvendigt.

1. Indstillinger for datakilde
2. Forespørgselsindstillinger
3. Avanceret editor

Advarsel!

Hvis du ændrer en filsti, skal du sørge for, at du opretter forbindelse til den samme fil med den samme filstruktur igen. Alle strukturelle ændringer af en fil, f.eks. sletning eller omdøbning af kolonner i kildefilen, afbryder rapporteringsmodellen.

Prøv f.eks. at ændre filstien til datakilden under indstillingerne for datakilden. Vælg **Indstillinger for datakilde** i Power Query. I vinduet **Indstillinger for datakilde** skal du vælge din fil og derefter vælge **Skift kilde**. Opdater **filstien**, eller brug indstillingen **Gennemse** til at finde den fil, vælg **OK**, og vælg derefter **Luk**.



Næste undermodul: Hent data fra relationsdatakilder

[Tidligere](#)

Hent data fra relationsdatakilder, SQL-DB

Fuldført 100 XP

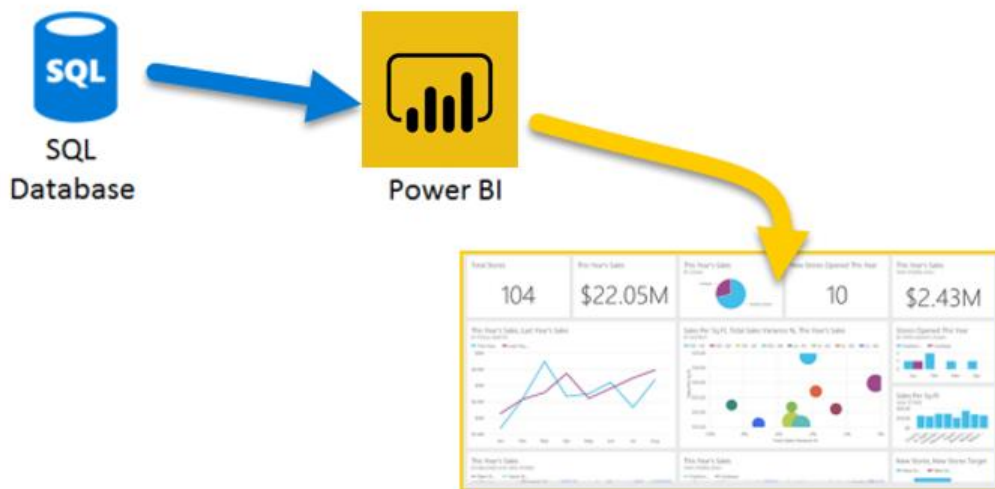
- 14 minutter

Hvis din organisation bruger en relationsdatabase til salg, kan du bruge Power BI Desktop til at oprette direkte forbindelse til databasen i stedet for at bruge eksporterede flade filer.

Hvis du opretter forbindelse mellem Power BI og din database, hjælper det dig med at overvåge din virksomheds status og identificere tendenser, så du kan forudsige salgstal, planlægge budgetter og angive indikatorer og mål for præstation. Power BI Desktop kan oprette forbindelse til mange relationsdatabaser, der enten befinder sig i cloudmiljøet eller i det lokale miljø.

Scenarie

Salgsteamet hos Tailwind Traders har anmodet om, at du opretter forbindelse til organisationens lokale SQL Server-database og henter salgsdataene til Power BI Desktop, så du kan oprette salgsrapporter.

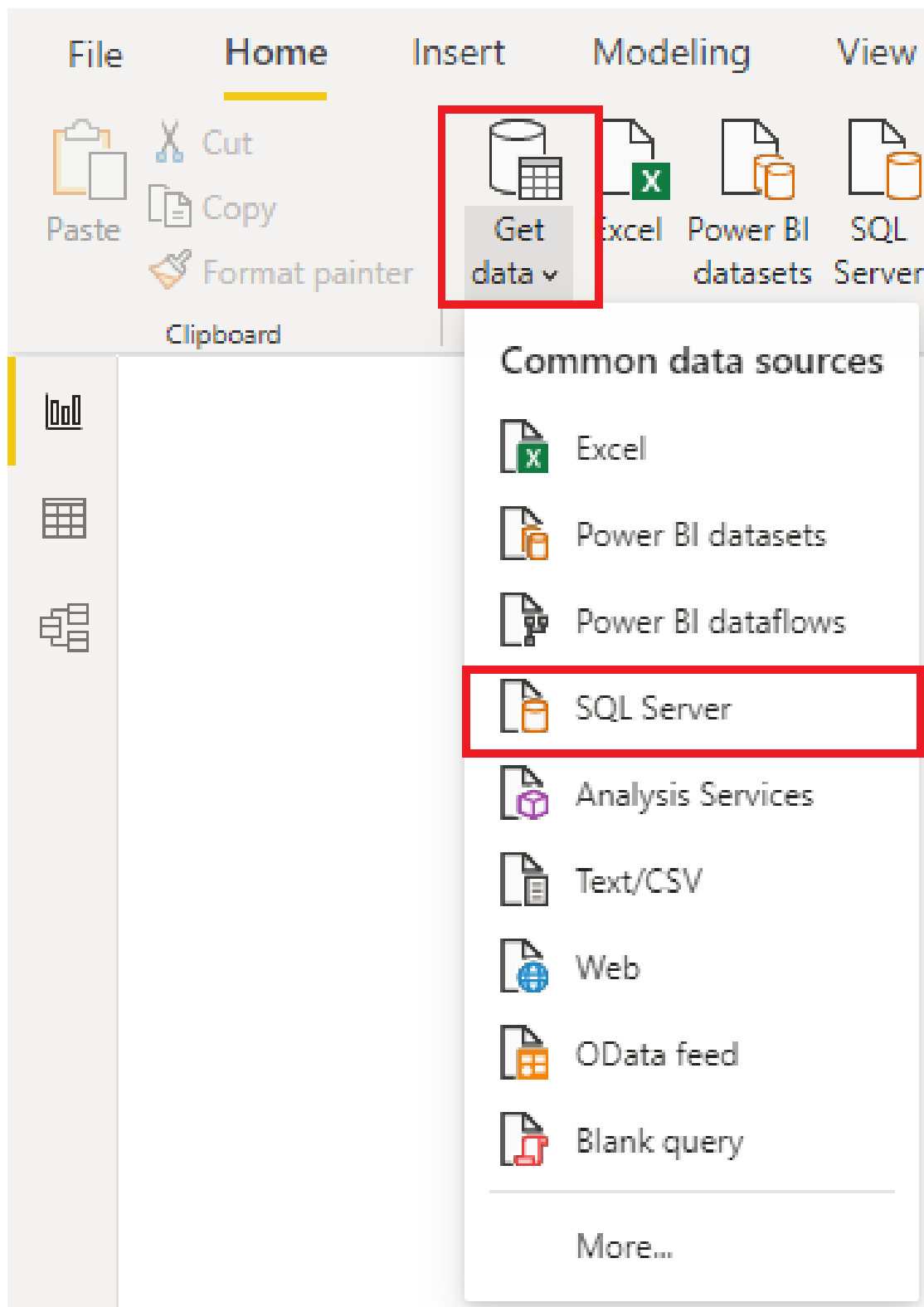


Opret forbindelse til data i en relationsdatabase

Du kan bruge funktionen **Hent data** i Power BI Desktop og vælge den relevante indstilling for din relationsdatabase. I dette eksempel skal du vælge indstillingen **SQL Server** som vist på følgende skærm billede.

Tip

Ud for knappen **Hent data** findes der indstillinger for hurtig adgang til datakilden, f.eks. **SQL Server**.



Dit næste trin er at angive navnet på din databaseserver og et navn på databasen i vinduet **SQL Server-database**. De to indstillinger i dataforbindelsestilstanden er: **Importér** (valgt som standard, anbefales) og **DirectQuery**. Du vælger for det meste **Importér**. Andre avancerede indstillinger er også tilgængelige i **SQL Server databasevinduet**, men du kan ignorere dem i øjeblikket.

SQL Server database

Server ①

Database (optional)

Data Connectivity mode ①

☒ Import

☐ DirectQuery

▶ Advanced options

OK

Cancel

Når du har tilføjet server- og databasenavne, bliver du bedt om at logge på med et brugernavn og en adgangskode. Du har tre logonindstillinger:

- **Windows** – Brug din Windows-konto (Legitimationsoplysninger til Azure Active Directory).
- **Database** – Brug dine legitimationsoplysninger til databasen. SQL Server har f.eks. sit eget system til logon og godkendelse, der nogle gange bruges. Hvis databaseadministratoren gav dig et unikt logon til databasen, skal du måske angive disse legitimationsoplysninger under fanen **Database**.
- **Microsoft-konto** – Brug legitimationsoplysningerne til din Microsoft-konto. Denne mulighed bruges ofte til Azure-tjenester.

Vælg en logonindstilling, angiv dit brugernavn og din adgangskode, og vælg derefter **Opret forbindelse**.

SQL Server database

Windows

Database

Microsoft account

Use your Windows credentials to access this database.

☒ Use my current credentials

☐ Use alternate credentials

User name

Password

Select which level to apply these settings to

localhost

Back

Connect

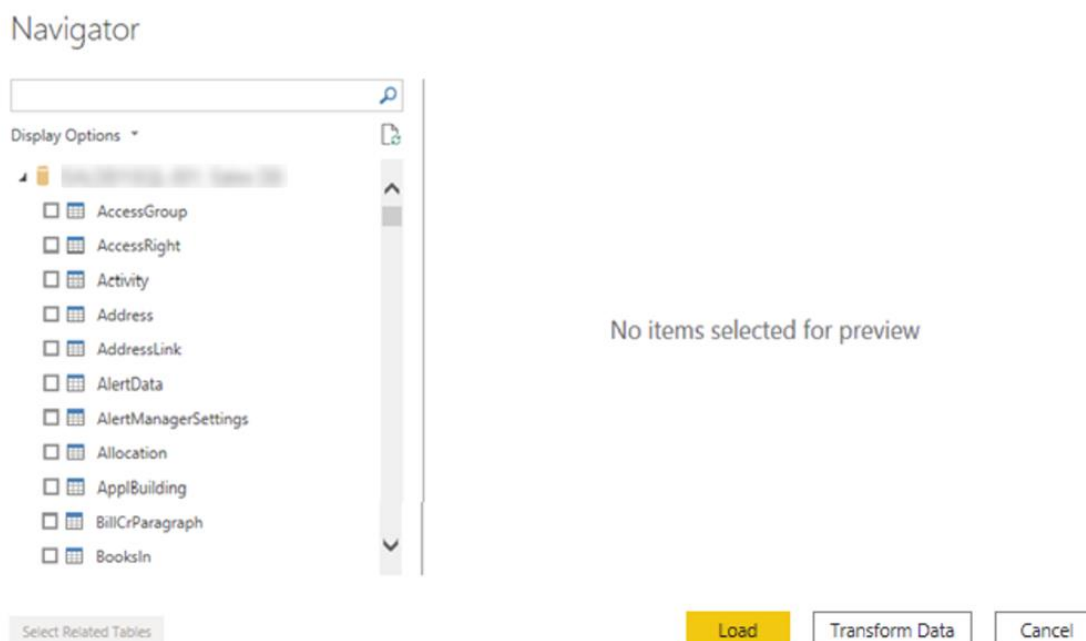
Cancel

Vælg data til import

Når databasen er oprettet forbindelse til Power BI Desktop, viser vinduet **Navigator** de data, der er tilgængelige i datakilden (SQL-databasen i dette eksempel). Du kan vælge en tabel eller enhed for at få forhåndsvist indholdet og sikre, at de korrekte data indlæses i Power BI-modellen.

Markér afkrydsningsfelterne i den eller de tabeller, du vil hente til Power BI Desktop, og vælg derefter enten indstillingen **Indlæs** eller **Transformér data**.

- **Indlæs** – Indlæs automatisk dine data i en Power BI-model i den aktuelle tilstand.
- **Transformér data** – Åbn dine data i Microsoft Power Query, hvor du kan udføre handlinger, f.eks. sletning af unødvendige rækker eller kolonner, gruppering af dine data, fjernelse af fejl og mange andre opgaver i datakvalitet.



Importér data ved at skrive en SQL-forespørgsel

En anden måde, du kan importere data på, er ved at skrive en SQL-forespørgsel for kun at angive de tabeller og kolonner, du har brug for.

Hvis du vil skrive SQL-forespørgslen, skal du angive server- og databasenavnene i **vinduet SQL Server database** og derefter vælge pilen ud for **Avancerede indstillinger** for at udvide denne sektion og få vist dine indstillinger. Skriv din forespørgselssætning i feltet **SQL-sætning**, og vælg derefter **OK**. I dette eksempel skal du bruge **Select** SQL-sætningen til at indlæse kolonnerne ID, NAME og SALESAMOUNT fra tabellen SALES.

SQL Server database

Server ①

Database (optional)

Advanced options

Command timeout in minutes (optional)

SQL statement (optional, requires database)

```
SELECT
  ID
, NAME
, SALESAMOUNT
FROM SALES
```

☒ Include relationship columns

☐ Navigate using full hierarchy

☐ Enable SQL Server Failover support

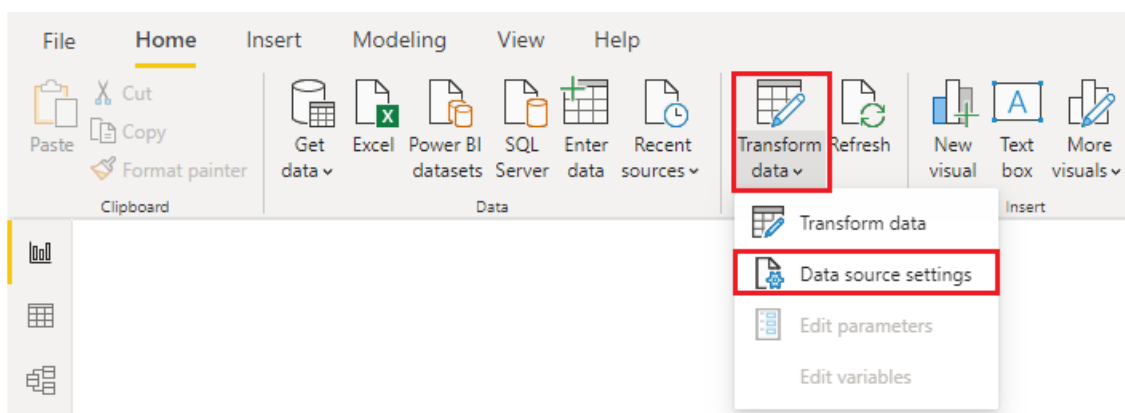
OK

Cancel

Skift indstillinger for datakilde

Efter du har oprettet en forbindelse til datakilden og indlæst data i Power BI Desktop, kan du når som helst vende tilbage og ændre dine forbindelsesindstillinger. Denne handling kræves ofte pga. en sikkerhedspolitik i organisationen, f.eks. hvis adgangskoden skal ændres hver 90. dag. Du kan ændre datakilden og redigere eller rydde tilladelser.

Vælg **Transformér data** under fanen **Hjem**, og vælg derefter indstillingen **Indstillinger for datakilde**.

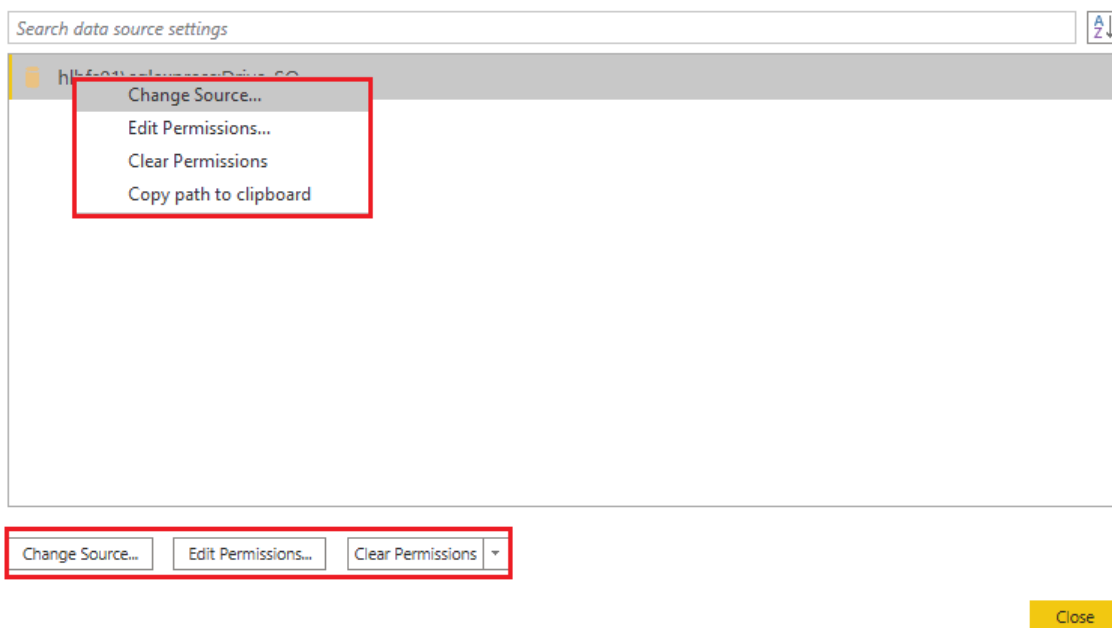


På den viste liste over datakilder skal du vælge den datakilde, du vil opdatere. Derefter kan du højreklikke på denne datakilde for at få vist de tilgængelige opdateringsmuligheder, eller du kan bruge knapperne til opdateringsmuligheder nederst til venstre i vinduet. Vælg den opdateringsmulighed, du har brug for, ret indstillingerne efter behov, og anvend derefter ændringerne.

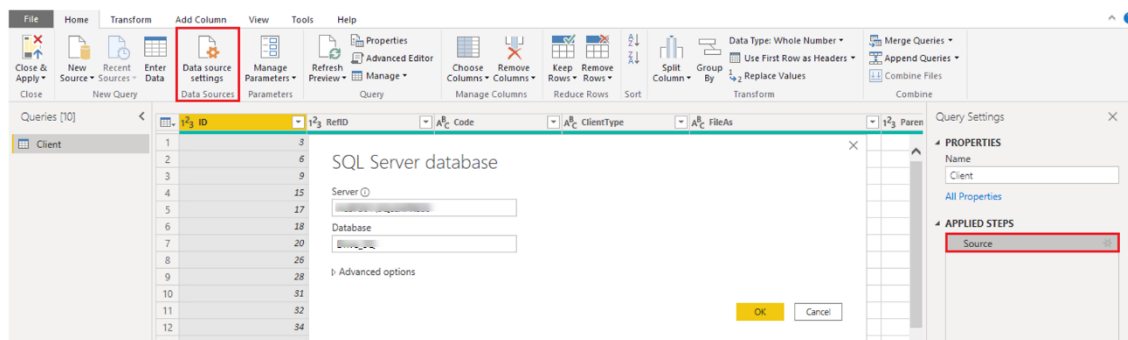
Data source settings

Manage settings for data sources that you have connected to using Power BI Desktop.

☒ Data sources in current file ☐ Global permissions



Du kan også ændre indstillingerne for din datakilde inde fra Power Query. Vælg tabellen, og vælg derefter indstillingen **Indstillinger for datakilde** på båndet **Hjem**. Du kan også gå til panelet **Forespørgselsindstillinger** i højre side af skærmen og vælge ikonet Indstillinger ud for Kilde (eller dobbelt Vælg kilde). I det vindue, der vises, skal du opdatere oplysningerne om serveren og databasen og derefter vælge **OK**.



Når du har foretaget ændringerne, skal du vælge **Luk og Anvend** for at anvende disse ændringer på indstillingerne for datakilden.

Skriv en SQL-sætning

Som tidligere nævnt kan du importere data i din Power BI-model ved hjælp af en SQL-forespørgsel. SQL står for Structured Query Language og er et standardiseret programmeringssprog, der bruges til at administrere relationsdatabaser og udføre forskellige handlinger til administration af data.

Tænk på det scenarie, hvor din database indeholder en stor tabel, som består af salgsdata for flere år. Salgsdata fra 2009 er ikke relevante for den rapport, du opretter. Denne situation er, hvor SQL er nyttig, fordi det kun giver dig mulighed for at indlæse det påkrævede datasæt ved at angive nøjagtige kolonner og rækker i DIN SQL-sætning og derefter importere dem til din semantiske model. Du kan også joinforbinde forskellige tabeller, køre specifikke beregninger, oprette logiske sætninger og filtrere data i SQL-forespørgslen.

Følgende eksempel er en simpel forespørgsel, hvor ID, NAVN og SALGSBELØB er valgt i tabellen SALG.

SQL-forespørgslen starter med en **Select-sætning**, som giver dig mulighed for at vælge de specifikke felter, du vil hente fra databasen. I dette eksempel vil du gerne indlæse kolonnerne ID, NAVN og SALGSBELØB.

SQLKopiér

```
SELECT  
  
ID  
  
, NAME  
  
, SALESAMOUNT  
  
FROM
```

FRA angiver navnet på den tabel, du vil hente data fra. I dette tilfælde er det tabellen SALG. Følgende eksempel er den fulde SQL-forespørgsel:

SQLKopiér

```
SELECT  
  
ID  
  
, NAME  
  
, SALESAMOUNT  
  
FROM  
  
SALES
```

Når du bruger en SQL-forespørgsel til at importere data, skal du prøve at undgå at bruge jokertegnet (*) i forespørgslen. Hvis du bruger jokertegnet (*) i SELECT-sætningen, importerer du alle de kolonner, du ikke har brug for, fra den angivne tabel.

Følgende eksempel viser forespørgslen med jokertegnet.

SQLKopiér

```
SELECT *  
  
FROM
```

SALES

Jokertegnet (*) importerer alle kolonner i tabellen **Sales** . Denne metode anbefales ikke, fordi den medfører redundante data i din semantiske model, hvilket medfører problemer med ydeevnen og kræver ekstra trin for at normalisere dine data til rapportering.

Alle forespørgsler skal også have en **WHERE**-sætning. Med denne sætning filtreres rækkerne, så der kun vælges filtrerede poster, som du vil have. Hvis du i dette eksempel vil hente de seneste salgsdata efter den 1. januar 2020, skal du tilføje en **WHERE-delsætning** . Den udvidede forespørgsel vil ligne følgende eksempel.

SQLKopiér

SELECT

ID

, NAME

, SALESAMOUNT

FROM

SALES

WHERE

OrderDate >= '1/1/2020'

Det er en bedste praksis at undgå at gøre dette direkte i Power BI. Overvej i stedet at skrive en forespørgsel som denne i en visning. En visning er et objekt i en relationsdatabase, der ligner en tabel. Visninger har rækker og kolonner og kan indeholde næsten alle former for operatorer i SQL-sproget. Hvis Power BI bruger en visning, når der hentes data, gøres der brug af forespørgselsdelegering, som er en funktion i Power Query. Forespørgselsdelegering forklares senere, men kort sagt så betyder det, at Power Query optimerer datahentning i henhold til, hvordan dataene skal bruges senere.

Opret dynamiske rapporter med parametre

Fuldført100 XP

- 6 minutter

Dynamiske rapporter er rapporter, hvor dataene kan ændres af en udvikler i henhold til brugerspecifikationer. Dynamiske rapporter er værdifulde, fordi en enkelt rapport kan bruges til flere formål. Hvis du bruger dynamiske rapporter, har du færre individuelle rapporter, der skal oprettes, hvilket vil spare tid og ressourcer i organisationen.

Du kan bruge parametre ved at bestemme de værdier, du vil have vist data for i rapporten, og rapporten opdateres tilsvarende ved at filtrere datafor dig.

Når du opretter dynamiske rapporter, kan du give brugerne af dine rapporter mulighed for selv at ændre datakilden og filtrere de data, der vises i rapporterne.

I følgende eksempel har du oprettet en rapport for salgsteamet hos Tailwind Traders, der viser salgsdataene i SQL Server-databasen. Rapporten giver en holistisk visning af, hvordan salgsteamet præsterer. Selvom rapporten er meget nyttig, vil medlemmer af salgsteamet gerne kunne filtrere rapporten, så de kun får vist deres egne data og nemmere spore deres præstation i forhold til deres salgsmål.

Opret dynamiske rapporter til enkelte værdier

Hvis du vil oprette en dynamisk rapport, skal du først skrive din SQL-forespørgsel. Brug derefter funktionen **Hent data** i Power BI Desktop til at oprette forbindelse til databasen.

I dette eksempel opretter du forbindelse til databasen på SQL Server ved at følge disse trin:

1. Når du har angivet dine serveroplysninger, skal du i vinduet **SQL Server-database** vælge **Avancerede indstillinger**.
2. Indsæt SQL-forespørgslen i feltet **SQL-sætning**, og vælg derefter **OK**.

```
SELECT [SalesOrderID]
      ,[OrderDate]
      ,[OnlineOrderFlag]
      ,[SalesOrderNumber]
      ,[CustomerID]
      ,[SalesPersonID]
      ,[TerritoryID]
      ,[SubTotal]
      ,[TaxAmt]
      ,[Freight]
      ,[TotalDue]
FROM [TailwindTraders].[Sales].[SalesOrderHeader] s
WHERE s.[SalesPersonID] = 279
```

SQL Server database

Server ⓘ

ABC

localhost\SQLEXPRESS

Database (optional)

ABC

TailwindTraders

Data Connectivity mode ⓘ

☒ Import
 ☐ DirectQuery

⌄ Advanced options

Command timeout in minutes (optional)

SQL statement (optional, requires database)

```

SELECT
    ,[TaxAmt]
    ,[Freight]
    ,[TotalDue]
FROM [TailwindTraders].[Sales].[SalesOrderHeader] s
WHERE s.[SalesPersonID] = 279

```

☒ Include relationship columns
 ☐ Navigate using full hierarchy
 ☐ Enable SQL Server Failover support

OK

Cancel

Når forbindelsen oprettes, vises dataene i eksempelvinduet.

3. Vælg **Rediger** for at åbne data i Power Query-editor.

Derefter skal du oprette parameteren ved at følge disse trin:

1. Vælg **Administrer parametre > Ny parameter** under fanen **Hjem**.
2. Rediger parameterens standardnavn til noget mere beskrivende i vinduet **Parametre**, så dens formål fremgår klart. I dette tilfælde skal du ændre navnet til **SalesPerson**.
3. Vælg **Tekst** på listen **Type**, og vælg derefter **Vilkårlig værdi** på listen **Foreslået værdi**.
4. Vælg **OK**.

Manage Parameters

New

ABC SalesPerson

Name
SalesPerson

Description

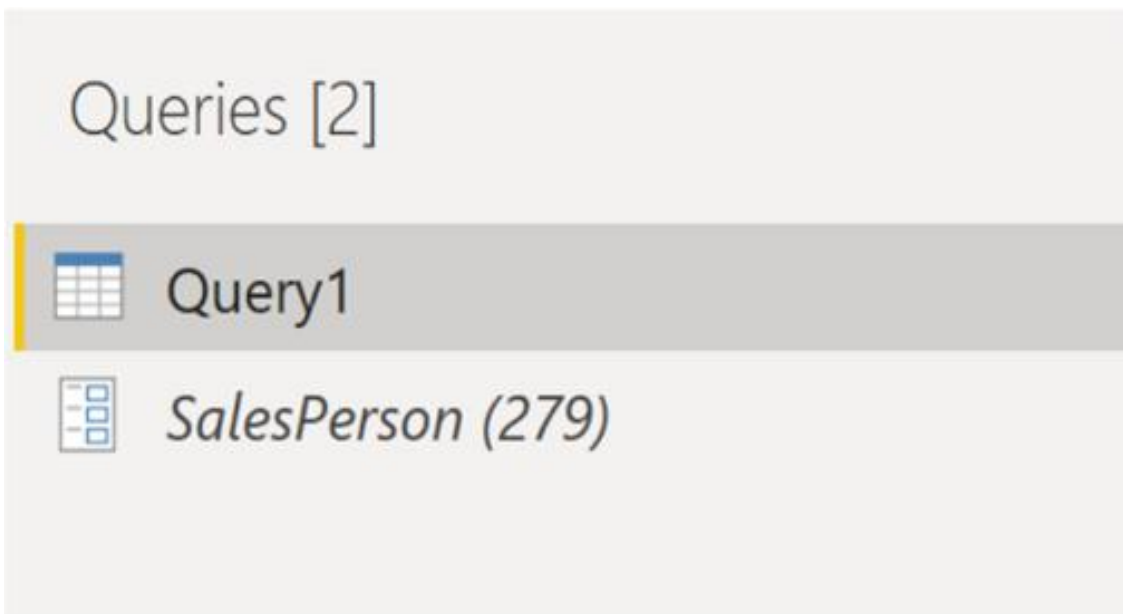
☒ Required

Type
Text

Suggested Values
Any value

Current Value
279

Der vises en ny forespørgsel for den parameter, du har oprettet.



Nu skal du justere koden i SQL-forespørgslen, så du kan vurdere den nye parameter:

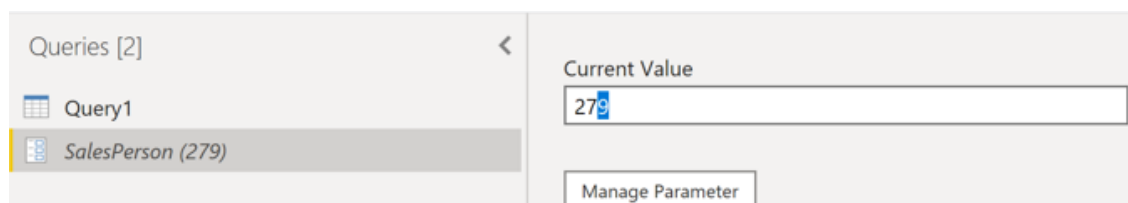
1. Højreklik på **Query1**, og vælg derefter **Avanceret editor**.
2. Erstat den eksisterende værdi i execute-sætningen med et ampersand (&) efterfulgt af dit parameternavn (**SalesPerson**), som vist på følgende billede.



3. Kontrollér, at der ikke vises nogen fejl nederst i vinduet, og vælg derefter **Udført**.

Selvom du ikke kan se en forskel på skærmen, kørte Power BI forespørgslen.

4. Hvis du vil have bekræftet, at forespørgslen blev kørt, kan du køre en test ved at vælge parameterforespørgslen og angive en ny værdi i feltet **Aktuel værdi**.



5. Der vises muligvis et advarselsikon ud for forespørgslen. Hvis det er tilfældet, skal du vælge den pågældende forespørgsel for at få vist advarslen, hvor det er angivet, at der kræves tilladelse for at køre den oprindelige databaseforespørgsel. Vælg **Rediger tilladelse**, og vælg derefter **Kør**.

Når forespørgslen kører korrekt, vises den nye værdi i parameteren.

	A ^B SalesOrderNumber	1 ² SalesOrderID	1 ² SalesPersonID	OrderDate
1	SO43659	43659	279	31/05/2011 00:00:00
2	SO43660	43660	279	31/05/2011 00:00:00
3	SO43681	43681	279	31/05/2011 00:00:00
4	SO43684	43684	279	31/05/2011 00:00:00
5	SO43685	43685	279	31/05/2011 00:00:00
6	SO43694	43694	279	31/05/2011 00:00:00
7	SO43695	43695	279	31/05/2011 00:00:00
8	SO43696	43696	279	31/05/2011 00:00:00

6. Vælg **Luk og anvend** for at vende tilbage til rapporteditoren.

Nu kan du anvende parameteren på rapporten:

1. Vælg **Rediger forespørgsler > Rediger parametre**.
2. Angiv en ny værdi i vinduet **Rediger parametre**, og vælg derefter **OK**.
3. Vælg **Anvend ændringer**, og kør derefter den oprindelige forespørgsel igen.

Når du nu får vist dataene, kan du se dataene for den nye værdi, der blev sendt gennem parameteren.

SalesOrderNumber	SalesOrderID	SalesPersonID	OrderDate	CustomerID	TerritoryID	SubTotal	TaxAmt	Freight	TotalDue
SO43659	43659	279	31/05/2011 00:00:00	29825	5	20565.6206	1971.5149	616.0984	23153.2339
SO43660	43660	279	31/05/2011 00:00:00	29672	5	1294.2529	124.2483	38.8276	1457.3288
SO43681	43681	279	31/05/2011 00:00:00	29661	5	13787.5434	1323.0668	413.4584	15524.0686
SO43684	43684	279	31/05/2011 00:00:00	29912	5	5596.4705	537.2612	167.8941	6301.6258
SO43685	43685	279	31/05/2011 00:00:00	30084	5	2736.5678	263.201	82.2503	3082.0191
SO43694	43694	279	31/05/2011 00:00:00	29549	5	20645.634	1978.3257	618.2268	23242.1865
SO43695	43695	279	31/05/2011 00:00:00	29958	5	39373.781	3787.4632	1183.5823	44344.8265
SO43696	43696	279	31/05/2011 00:00:00	29849	5	419.4589	40.2681	12.5838	472.3108
SO43845	43845	279	01/07/2011 00:00:00	29888	5	8580.0739	823.6669	257.3959	9661.1367
SO43861	43861	279	01/07/2011 00:00:00	29749	5	23401.1062	2244.4088	701.3777	26346.8927
SO43862	43862	279	01/07/2011 00:00:00	29945	5	31000.7804	2987.8703	933.7095	34922.3602

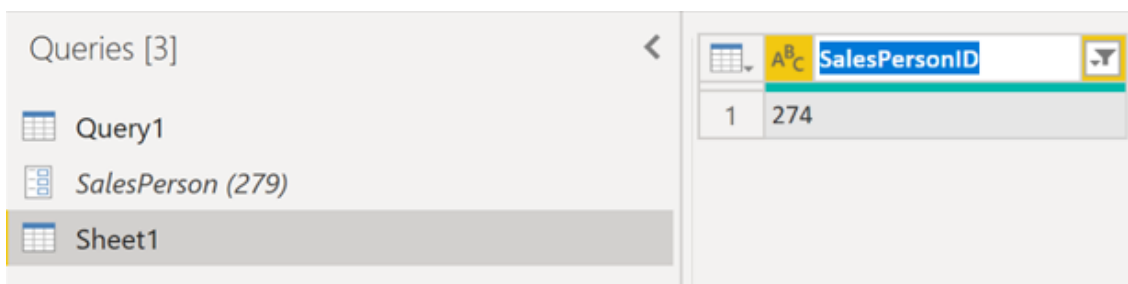
Du kan nu oprette en rapport, der viser data for en bestemt værdi ad gangen. Der kræves flere trin for at få vist data for flere værdier på samme tid.

Opret dynamiske rapporter for flere værdier

Hvis du vil have flere værdier ad gangen, skal du først oprette et Microsoft Excel-regneark, der indeholder en tabel, som består af én kolonne, der indeholder listen over værdier.

Derefter skal du bruge funktionen **Hent data** i Power BI Desktop for at oprette forbindelse til dataene i Excel-regnearket, og derefter følge disse trin:

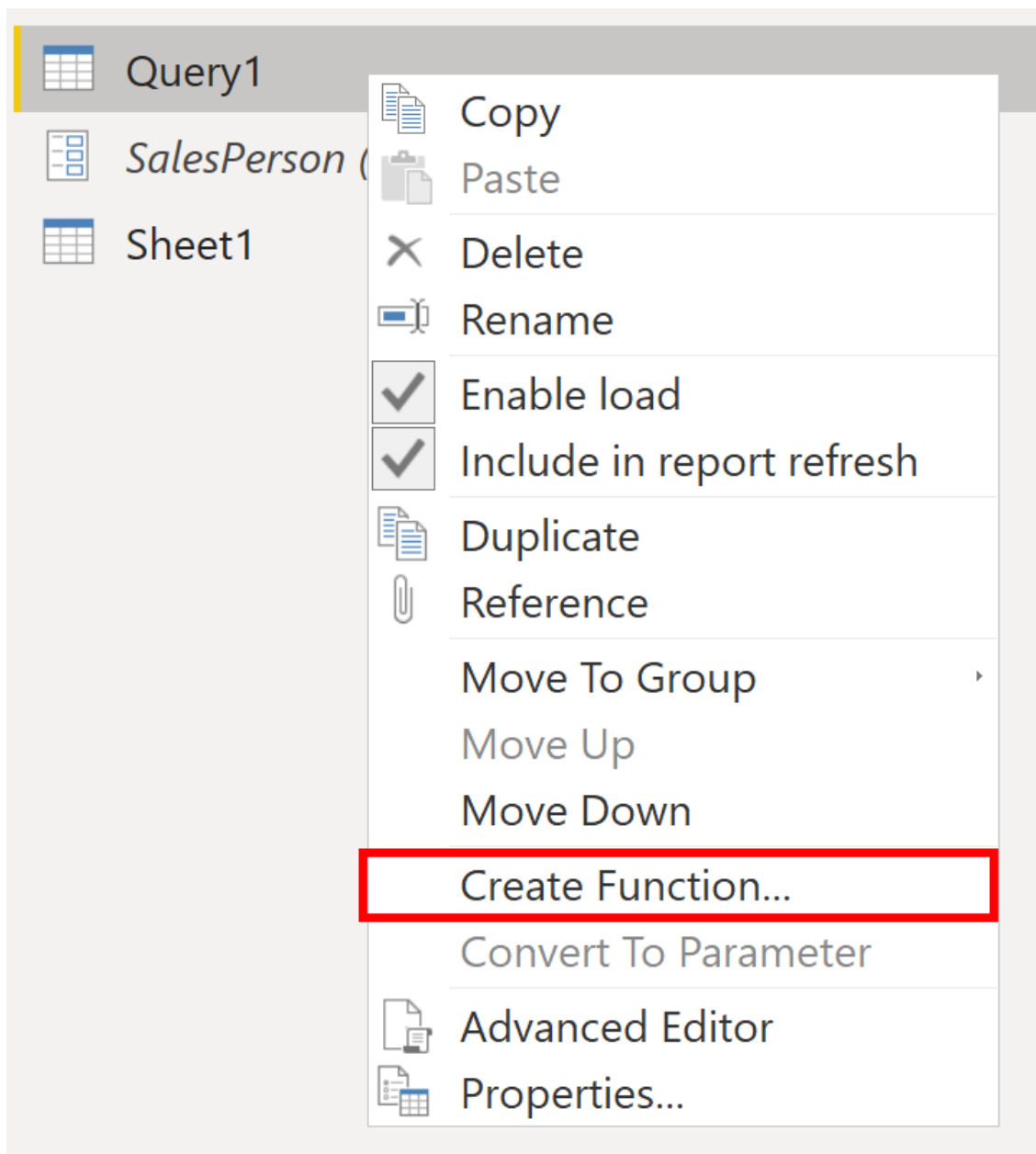
1. I vinduet **Navigatør** skal du vælge **Rediger** for at åbne dataene i Power Query-editor, hvor du kan se en ny forespørgsel til datatabellen.



2. Omdøb kolonnen i tabellen til noget mere beskrivende.
3. Skift kolonnedatatypen til **Tekst**, så den stemmer overens med parametertypen, og du undgår problemer med datakonvertering.
4. I afsnittet **Egenskaber** i forespørgslen kan du også ændre navnet på datakilden til noget mere beskrivende. I dette eksempel skal du angive **SalesPersonID**.

Derefter skal du oprette en funktion, der overfører den nye **SalesPersonID-forespørgsel** til **Forespørgsel1**:

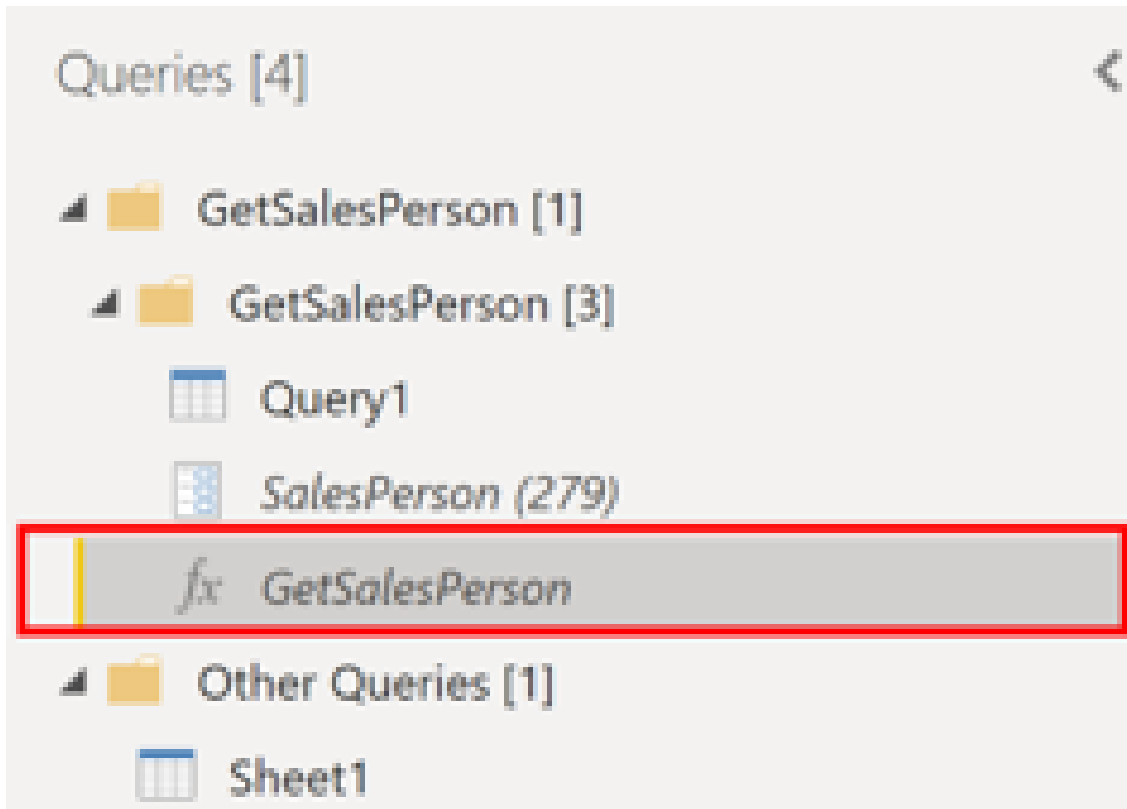
1. Højreklik på **Query1**, og vælg **Opret funktion**.



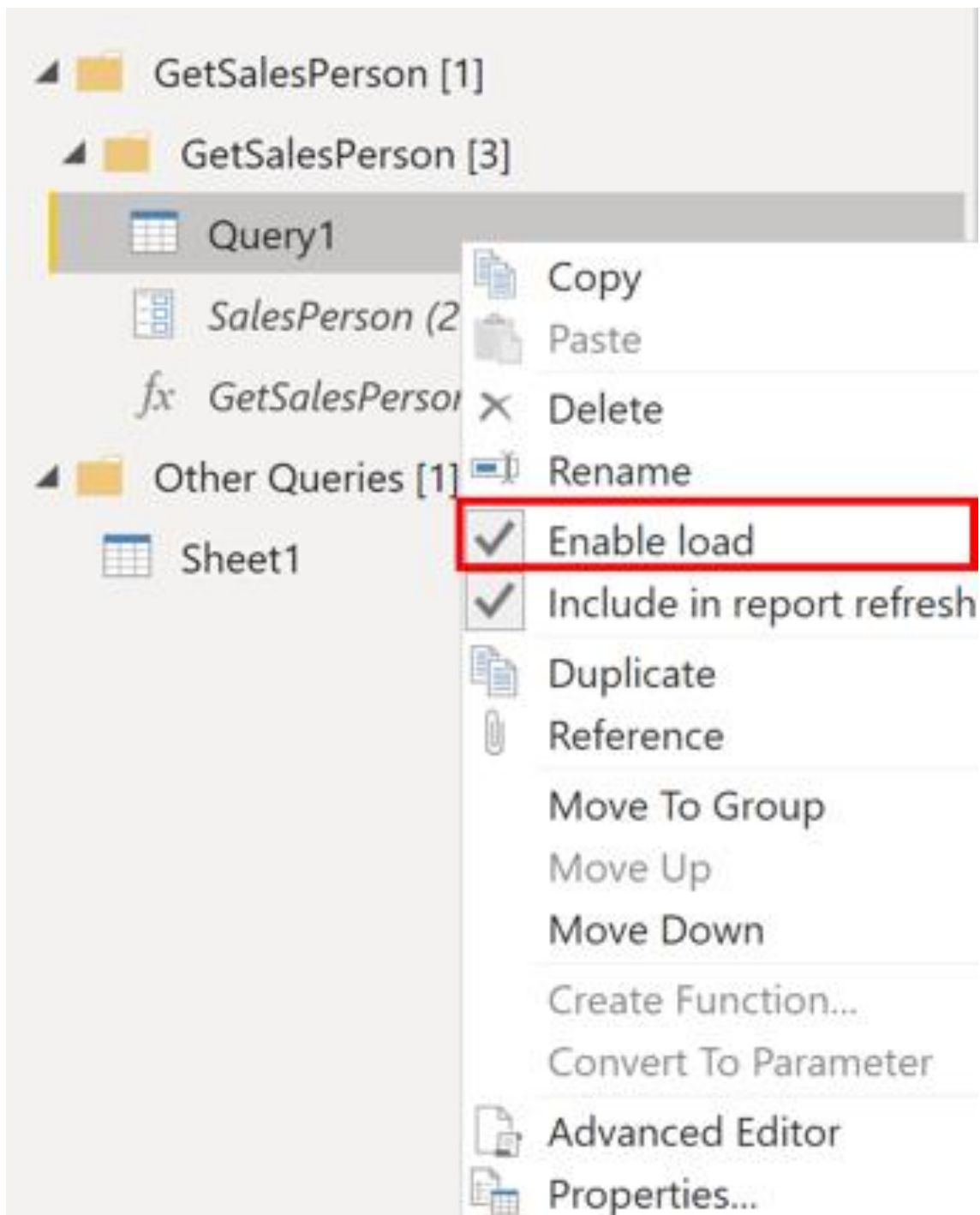
2. Angiv et navn til funktionen, og vælg derefter **OK**.

A screenshot of the 'Create Function' dialog box. The title bar says 'Create Function'. Below the title, it says 'Enter a name for the new function.' There is a text input field labeled 'Function name' containing the text 'GetSalesPerson'. Below the input field, it says 'Parameters: SalesPerson'. At the bottom right, there are two buttons: 'OK' (yellow) and 'Cancel' (grey).

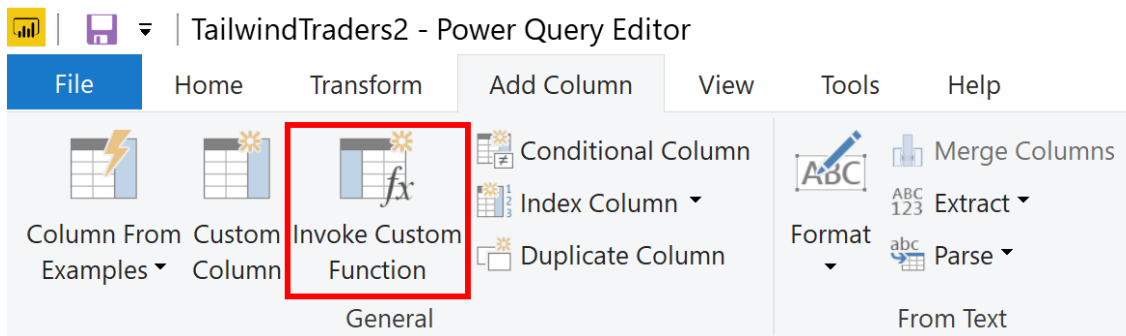
Den nye funktion vises i ruden **Forespørgsler** .



3. Hvis du vil sikre dig, at **Query1** ikke vises på feltlisten for rapporten, hvilket kunne forvirre brugerne, kan du deaktivere den, så den ikke indlæses i rapporten, ved at højreklikke på **Query1** igen og derefter vælge **Aktiver indlæsning** (valgt som standard) for at deaktivere funktionen.



4. Vælg den **SalesPersonID-forespørgsel**, du indlæste fra Excel-regnearket, og vælg derefter **Aktivér brugerdefineret funktion** under fanen **Tilføj kolonne** for at køre den brugerdefinerede funktion, du har oprettet.



5. Vælg funktionen på listen **Funktionsforespørgsel** i vinduet **Aktivér brugerdefineret funktion**.

Navnet **på den nye kolonne** opdateres automatisk, og den tabel, der indeholder de værdier, du vil overføre gennem parameteren, vælges som standard.

6. Vælg **OK**, og kørs om nødvendigt den oprindelige forespørgsel.

Der vises en ny kolonne til funktionen **GetSalesFromSalesPerson** ud for kolonnen **SalesPersonID**.

	ABC SalesPersonID	ABC 123 GetSalesFromSalesPerson
1	274	Table

7. Vælg ikonet med to pile i den nye kolonneoverskrift, og markér derefter afkrydsningsfelterne for de kolonner, du vil indlæse. I dette afsnit kan du se, hvilke oplysninger der er tilgængelige i rapporten for hver værdi (salgsperson-id).
8. Ryd afkrydsningsfeltet **Anvend det oprindelige kolonnenavn som præfiks**, fordi du ikke behøver at få vist et præfiks med kolonnenavnene i rapporten.
9. Vælg **OK**.

ABC SalesPersonID ABC 123 GetSalesFromSalesPerson

1

Expand Aggregate

- ☒ (Select All Columns)
- ☒ SalesOrderNumber
- ☒ SalesOrderID
- ☒ SalesPersonID
- ☒ OrderDate
- ☒ CustomerID
- ☒ TerritoryID
- ☒ SubTotal
- ☒ TaxAmt
- ☒ Freight
- ☒ TotalDue
- ☒ OnlineOrderFlag

☐ Use original column name as prefix

OK Cancel

Nu burde du kunne se data for de kolonner, du har markeret, for hver værdi (sælger-id).

Queries [4]

- GetSalesPerson [1]
- GetSalesPerson [3]
 - Query1
 - SalesPerson (279)
 - fx GetSalesPerson
- Other Queries [1]
 - fSales

fx = Table.Distinct("#Expanded GetSalesFromSalesPerson ", {"Sa

	ABC SalesPersonID	ABC SalesOrderNumber	ABC SalesOrderID
1	274	SO43849	43849
2	275	SO43670	43670
3	276	SO43663	43663
4	277	SO43667	43667
5	278	SO43677	43677
6	279	SO43659	43659
7	280	SO43664	43664

Du kan om nødvendigt føje flere værdier (sælger-id'er) til kolonnen **SalesPersonID** i Excel-regnearket, eller du kan ændre de eksisterende værdier.

10. Gem ændringerne, og gå derefter tilbage til Power Query-editor.
11. Vælg **Opdater eksempel** under fanen **Hjem**, og kød derefter den oprindelige forespørgsel igen (om nødvendigt). Du kan se salget for de nye sælger-id'er, du har føjet til regnearket.
12. Vælg **Luk og Anvend** for at vende tilbage til rapporteditoren, hvor du kan se de nye kolonnenavne i ruden Felter.

Nu kan du begynde at oprette din rapport.

Næste undermodul: Hent data fra en NoSQL-database

[Tidligere Næste](#)

Har du brug for hjælp? Se i vores [fejlfindingsvejledning](#), eller angiv spec

Hent data fra en NoSQL-database

Fuldført100 XP

- 5 minutter

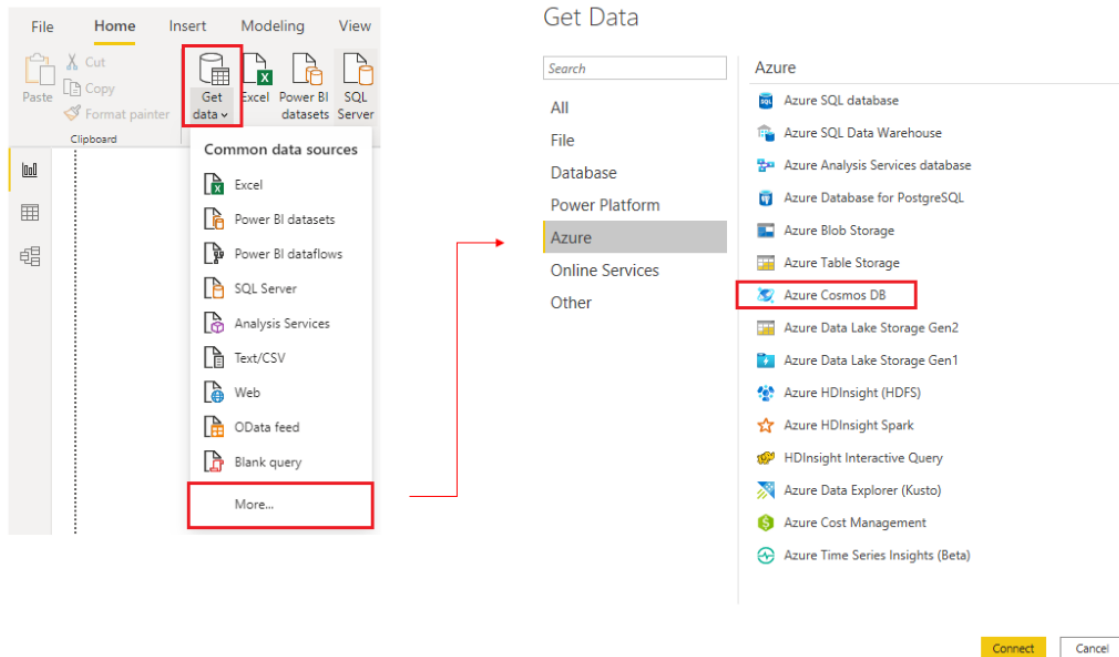
Nogle organisationer bruger ikke en relationsdatabase, men bruger i stedet en *NoSQL-database*. En NoSQL-database (også kaldet ikke-SQL, ikke kun SQL eller *ikke-relationel*) er en fleksibel type database, der ikke bruger tabeller til at gemme data.

Scenarie

Softwareudviklere hos Tailwind Traders oprettede et program til administration af forsendelse og sporing af produkter fra deres lagre. Programmet bruger Cosmos DB, en NoSQL-database, som datalager. Data gemmes som JSON-dokumenter, som er åbne standardfilformater, der primært bruges til at overføre data mellem en server og et webprogram. Du skal importere disse data til en semantisk Power BI-model til rapportering.

Opret forbindelse til en NoSQL-database (Azure Cosmos DB)

I dette scenarie skal du bruge funktionen **Hent data** i Power BI Desktop. Denne gang skal du dog vælge indstillingen **Mere...** for at finde og oprette forbindelse til den type database, du bruger. I dette eksempel skal du vælge **Azure-kategorien**, vælge **Azure Cosmos DB** og derefter vælge **Opret forbindelse**.



I vinduet **Eksempelforbindelse** skal du vælge **Fortsæt** og derefter angive dine databaselegitimationsoplysninger. I dette eksempel kan du angive databaseoplysningerne i vinduet **Azure Cosmos DB**. Du kan angive url-adressen til Azure Cosmos DB-kontoens slutpunkt, som du vil hente dataene fra (du kan hente URL-adressen fra bladet **Nøgler** i Azure Portal). Alternativt kan du angive navnet på databasen eller samlingen eller bruge navigatoren til at vælge databasen og samlingen til at identificere datakilden.

Hvis du opretter forbindelse til et slutpunkt for første gang, som du er i dette eksempel, skal du sørge for at angive din kontonøgle. Du kan finde denne nøgle i feltet **Primær nøgle** i bladet **Skrivebeskyttede nøgler** i Azure Portal.

Importér en JSON-fil

Hvis du arbejder med data, der er gemt i JSON-format, er det ofte nødvendigt at udtrække og normalisere dataene først. Dette skyldes, at JSON-data ofte gemmes i et indlejret eller ustruktureret format, hvilket gør det svært at analysere eller rapportere direkte.

I dette eksempel skal dataene udtrækkes og normaliseres, før du kan rapportere om dem, så du skal transformere dataene, før du indlæser dem i Power BI Desktop.

Når du har oprettet forbindelse til databasekontoen, åbnes vinduet **Navigatør**, hvor der vises en liste over databaser under den pågældende konto. Vælg den tabel, du vil importere. I dette eksempel skal du vælge tabellen Produkt. På fanen til forhåndsvisning vises kun **Postelementer**, da alle poster i dokumentet repræsenteres som typen Post i Power BI.

Navigator

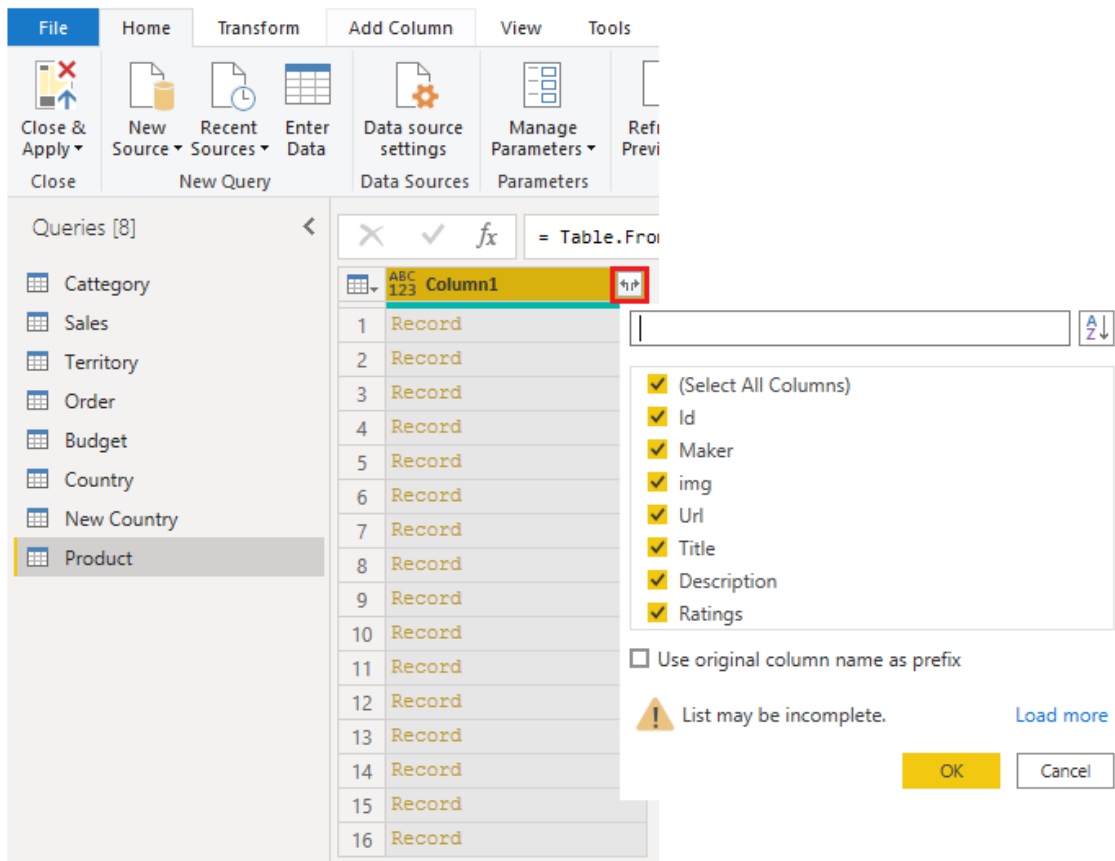
The screenshot shows the Power BI Navigator interface. On the left, there is a search bar and a 'Display Options' dropdown. Below this, a tree view shows the following structure:

- https://analytics.documents.azure.com [5]
 - warehouse
 - ☒ Product

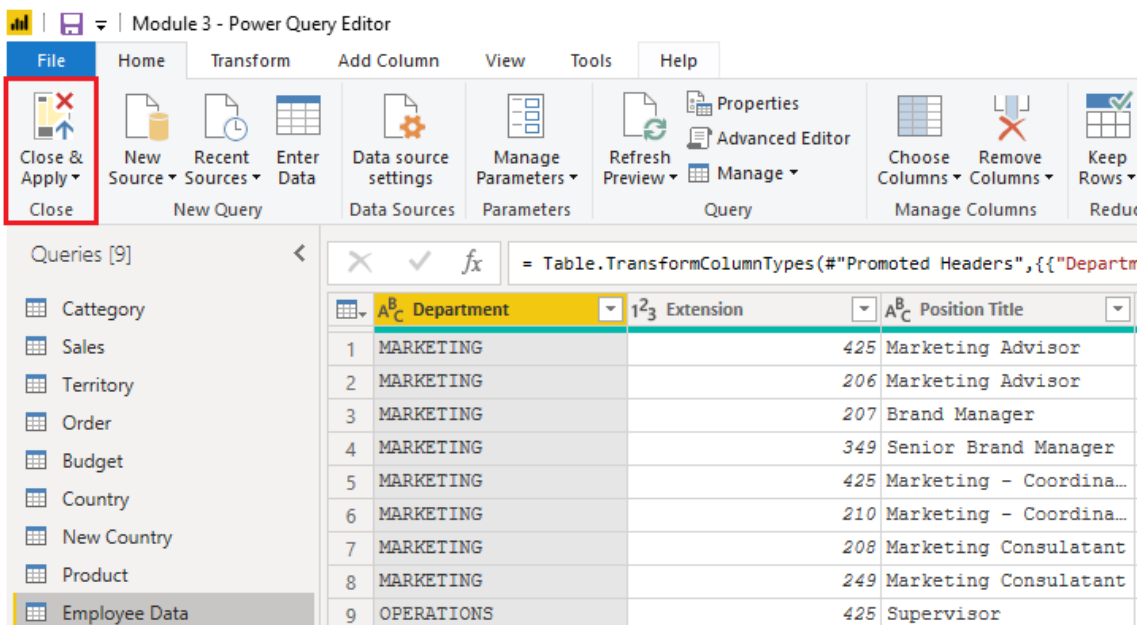
On the right, the 'Product' table is displayed with a header 'Document' and 15 rows of 'Record'. A vertical scrollbar is on the right side of the table. At the bottom, there are three buttons: 'Load' (yellow), 'Edit' (grey), and 'Cancel' (grey).

Vælg knappen **Rediger** for at åbne posterne i Power Query.

I Power Query skal du vælge knappen **Udvid** til højre for **kolonne1-overskriften**, som viser genvejsmenuen med en liste over felter. Markér de felter, du vil indlæse i Power BI Desktop, fjern markeringen i afkrydsningsfeltet **Brug oprindeligt kolonnenavn som præfiks**, og vælg derefter **OK**.



Gennemse de valgte data for at sikre, at du er tilfredse med dem, og vælg derefter **Luk & Anvend** for at indlæse dataene i Power BI Desktop.



Dataene ligner nu en tabel med rækker og kolonner. Data fra Cosmos DB kan nu relateres til data fra andre datakilder og kan evt. bruges i en Power BI-rapport.

Næste undermodul: Hent data fra on

Hent data fra onlinetjenester

Fuldført 100 XP

- 5 minutter

Organisationer bruger ofte en række softwareprogrammer til at understøtte deres daglige handlinger, f.eks. SharePoint, OneDrive, Dynamics 365, Google Analytics osv. Disse programmer producerer deres egne data. Power BI kan kombinere data fra flere programmer for at skabe mere meningsfuld indsigt og mere meningsfulde rapporter.

Scenarie

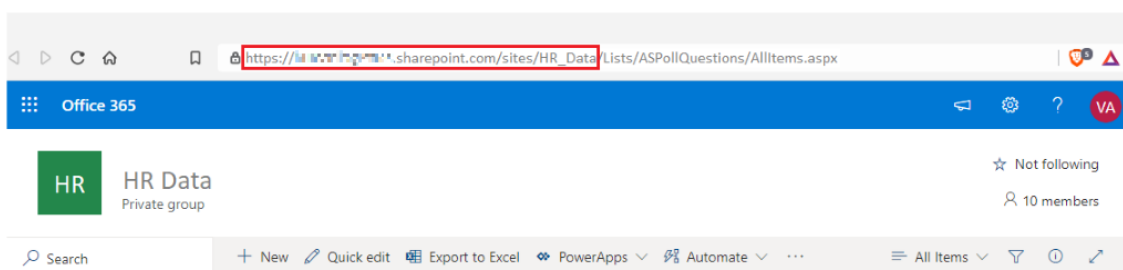
Tailwind Traders bruger SharePoint til samarbejde og til at gemme salgsdata. Det er starten på det nye regnskabsår, og salgscheferne vil gerne angive nye mål for salgsteamet. Den formular, som ledelsen bruger, findes i SharePoint. Du skal oprette forbindelse til disse data i Power BI Desktop, så salgsmålene kan bruges sammen med andre salgsdata for at bestemme tilstanden af salgspipeline.

I følgende afsnit undersøges det, hvordan du bruger funktionen Power BI Desktop **Hent data** til at oprette forbindelse til datakilder, der er produceret af eksterne programmer. For at illustrere denne proces har vi angivet et eksempel, der viser, hvordan du opretter forbindelse til et SharePoint-websted og importerer data fra en onlineliste.

Opret forbindelse til data i et program

Når du opretter forbindelse til data i et program, begynder du på samme måde, som når du opretter forbindelse til de andre datakilder: ved at vælge funktionen **Hent data** i Power BI Desktop. Vælg derefter den indstilling, du har brug for, i kategorien **Onlinetjenester**. I dette eksempel skal du vælge **SharePoint-onlineliste**.

Når du har valgt **Opret forbindelse**, bliver du bedt om at angive din SharePoint-URL-adresse. Denne URL-adresse er den, du bruger til at logge på dit SharePoint-websted via en webbrowser. Du kan kopiere URL-adressen fra dit SharePoint-websted og sætte den ind i forbindelsesvinduet i Power BI. Du behøver ikke at angive din fulde URL-filsti. Du behøver kun at indlæse webstedets URL-adresse, fordi du kan vælge den specifikke liste, du vil indlæse, når du har forbindelse. Afhængigt af den URL-adresse, du har kopieret, skal du muligvis slette den sidste del af URL-adressen, som vist på følgende billede.



Når du har angivet din URL-adresse, skal du vælge **OK**. Power BI skal godkende forbindelsen til SharePoint, så log på med din Microsoft-konto, og vælg derefter **Opret forbindelse**.

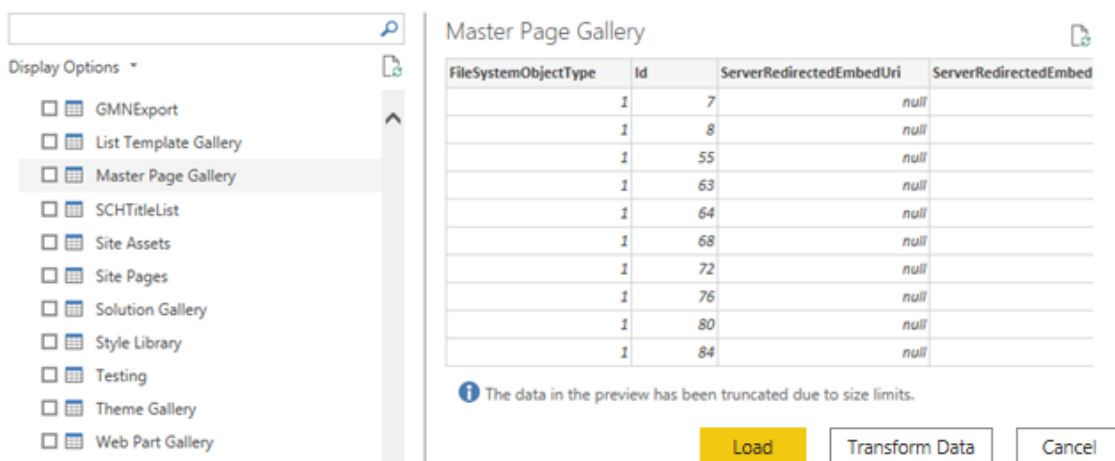


Vælg de programdata, der skal importeres

Når Power BI har oprettet forbindelse til SharePoint, vises vinduet **Navigatør**, ligesom når du opretter forbindelse til andre datakilder. I vinduet vises tabellerne og enhederne på dit SharePoint-websted. Vælg den liste, du vil indlæse i Power BI Desktop. På samme måde som når du importerer fra andre datakilder, har du mulighed for automatisk at indlæse dine data i en Power BI-model eller starte Power Query-editor for at transformere dine data, før du indlæser dem.

I dette eksempel skal du vælge indstillingen **Indlæs**.

Navigatør



Næste undermodul: Vælg en lagertilstand

Vælg en lagertilstand

Fuldført100 XP

- 6 minutter

Den mest populære måde at bruge data i Power BI på er ved at importere dem til en semantisk Power BI-model. Import af dataene betyder, at dataene gemmes i Power BI-filen og bliver publiceret sammen med

Power BI-rapporterne. Denne proces hjælper dig med at gøre det nemmere at interagere direkte med dine data. Denne tilgang vil dog måske ikke fungere for alle organisationer.

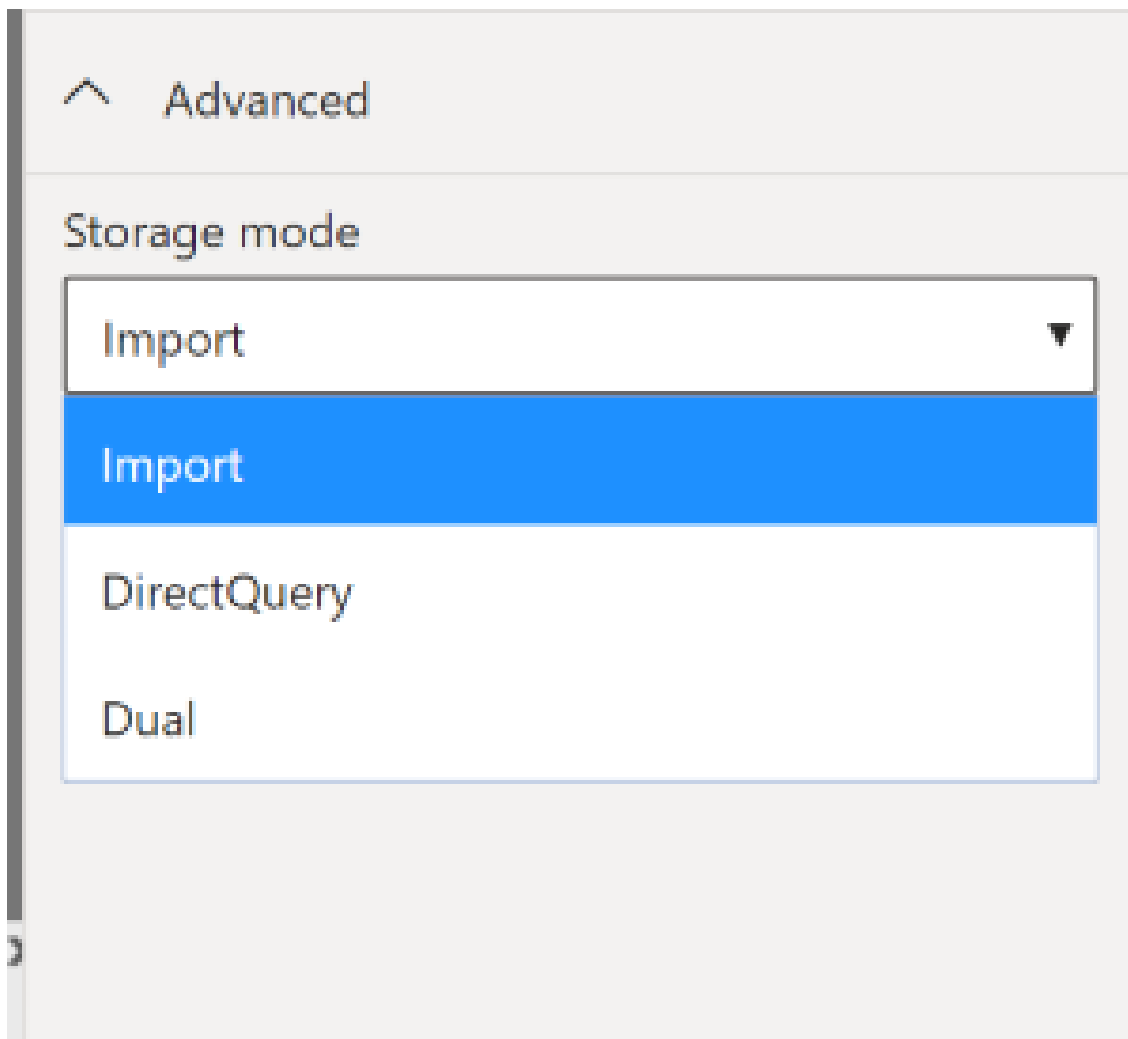
Hvis du vil fortsætte med scenariet, skal du oprette Power BI-rapporter til salgsafdelingen hos Tailwind Traders, hvor import af dataene ikke er en ideel metode. Den første opgave, du skal udføre, er at oprette dine semantiske modeller i Power BI, så du kan oprette visualiseringer og andre rapportelementer. Salgsafdelingen har mange forskellige semantiske modeller af forskellig størrelse. Af sikkerhedsmæssige årsager har du ikke tilladelse til at importere lokale kopier af dataene i dine rapporter, så direkte import af data er ikke længere en mulighed. Du har derfor brug for at oprette en direkte forbindelse til salgsafdelingens datakilde. I følgende afsnit beskrives det, hvordan du kan sikre, at disse forretningskrav er opfyldt, når du importerer data i Power BI.

Nogle gange kan der være sikkerhedskrav i forbindelse med dine data, der gør det umuligt at importere en kopi direkte. Eller dine semantiske modeller kan simpelthen være for store og vil tage for lang tid at indlæse i Power BI, og du vil undgå at skabe en flaskehals i ydeevnen. **Power BI løser disse problemer ved hjælp af DirectQuery-lagertilstand, som giver dig mulighed for at forespørge dataene direkte i datakilden og ikke importere en kopi i Power BI. DirectQuery er nyttig, da det sikrer, at du altid får vist den nyeste version af dataene.**

De tre forskellige typer lagertilstande, du kan vælge imellem er:

- Importér
- DirectQuery
- Dual (sammensat objekt)

Du kan få adgang til lagertilstande ved at skifte til visningen **Model**, vælge en datatabel og i den resulterende rude Egenskaber vælge, hvilken tilstand du vil bruge på rullelisten **Lagertilstand**, som vist i følgende visualisering.



Lad os se nærmere på de forskellige typer lagertilstande.

Importtilstand

Med importtilstanden kan du oprette en lokal Power BI-kopi af dine semantiske modeller fra din datakilde. Du kan bruge alle funktionerne i Power BI-tjenesten med denne lagertilstand, herunder Spørgsmål og svar samt Hurtig indsigt. Dataopdateringer kan planlægges eller efter behov. Importtilstand er standarden for udarbejdelse af nye Power BI-rapporter.

DirectQuery-tilstand

Indstillingen **DirectQuery** er nyttig, når du ikke vil gemme lokale kopier af dine data, fordi dine data ikke **cachelagres**. Du kan i stedet forespørge om de specifikke tabeller, du skal bruge, ved hjælp af oprindelige Power BI-forespørgsler, hvorefter de påkrævede data hentes fra den underliggende datakilde. I bund og grund opretter du en direkte forbindelse til datakilden. **Brug af denne model sikrer, at du altid får vist de nyeste data, og at alle sikkerhedskrav er opfyldt.** Denne tilstand er desuden velegnet til, når du har store semantiske modeller at hente data fra. I stedet for at ydeevnen bremses, fordi der skal indlæses store mængder data i Power BI, kan du bruge DirectQuery til at oprette en forbindelse til kilden, hvilket også løser problemer med ventetid.

Dual (sammensat tilstand)

I Dual-tilstand kan du identificere nogle data, der skal importeres direkte og andre data, der skal forespørges. Alle tabeller, der overføres til din rapport, er et produkt af både Import- og DirectQuery-tilstand. Hvis du bruger Dual-tilstand, kan Power BI vælge den mest effektive form for datahentning.

Du kan finde flere oplysninger om lagringstilstande i [Lagertilstande](#).

Næste undermodul: Hent data fra Azure Analysis Services

[Tidligere](#)

Hent data fra Azure Analysis Services

Fuldført 100 XP

- 5 minutter

Azure Analysis Services er en fuldt administreret platform som en tjeneste (PaaS), der leverer semantiske modeller i virksomhedsklassen i cloudmiljøet. Du kan bruge avancerede miks- og modelleringsfunktioner til at kombinere data fra flere datakilder, definere målepunkter og sikre dine data i en enkelt semantisk tabelmodel, der er tillid til. Den semantiske model gør det nemmere og hurtigere for brugerne at udføre ad hoc-dataanalyser ved hjælp af værktøjer som Power BI.

For at fortsætte scenariet så bruger Tailwind Traders Azure Analysis Services til at gemme økonomiske projekteringsdata. Du er blevet bedt om at sammenligne disse data med de faktiske salgsdata i en anden database. Hentning af data fra Azure Analysis Services server svarer til at hente data fra SQL Server, da du kan:

- Godkend til serveren.
- Vælg den model, du vil bruge.
- Vælg, hvilke tabeller du har brug for.

De bemærkelsesværdige forskelle mellem Azure Analysis Services og SQL Server er:

- Analysis Services-modeller har allerede oprettet beregninger.
- Hvis du ikke har brug for en hel tabel, kan du forespørge dataene direkte. I stedet for at bruge Transact-SQL (T-SQL) til at forespørge dataene, ligesom du ville gøre i SQL Server, kan du bruge et flerdimensionelt udtryk (MDX – Multi-Dimensional Expressions) eller DAX (Data Analysis Expressions).

Opret forbindelse til data i Azure Analysis Services

Som tidligere nævnt bruger du funktionen **Hent data** i Power BI Desktop. Når du vælger **Analysis Services**, bliver du bedt om at angive serveradressen og databasenavnet med to indstillinger: **Importér** og **Opret direkte forbindelse**.

SQL Server Analysis Services database

Server ⓘ

asazure://westus.azure.windows.net/azureanalysiservicesiketest

Database (optional)

adventureworks

☐ Import

☒ Connect live

▸ MDX or DAX query (optional)

OK

Cancel

Opret forbindelse live er en mulighed for Azure Analysis Services. Azure Analysis Services bruger tabelmodellen og DAX til at udføre beregninger, der svarer til Power BI. Disse modeller er kompatible med hinanden. Når du bruger indstillingen Opret liveforbindelse, kan du bevare data- og DAX-beregninger på deres oprindelige placering uden at skulle importere dem alle sammen i Power BI. Azure Analysis Services kan have en tidsplan for hurtig opdatering, hvilket betyder, at når data opdateres i tjenesten, opdateres Power BI-rapporter straks, uden at det er nødvendigt at starte en tidsplan for opdatering i Power BI. Denne proces kan forbedre rettidigheden af dataene i din rapport.

Ligesom med en relationsdatabase kan du vælge, hvilke tabeller du vil bruge. Hvis du vil forespørge Azure Analysis Services-modellen direkte, kan du bruge DAX eller MDX.

Du importerer sandsynligvis dataene direkte til Power BI. Et acceptabelt alternativ er, at du importerer alle andre data, som du gerne vil have (fra Excel, SQL Server osv.), i Azure Analysis Services-modellen og derefter bruger en liveforbindelse. Denne fremgangsmåde forenkler din løsning ved at holde de semantiske modellerings- og DAX-målinger på ét sted.

Du kan få flere oplysninger om, hvordan du opretter forbindelse mellem Power BI og Azure Analysis Services, i [dokumentationen Opret forbindelse til Power BI](#).

Næste undermodul: Løs problemer med ydeevnen

Løs problemer med ydeevnen

Fuldført 100 XP

- 10 minutter

Af og til har organisationer brug for at håndtere problemer med ydeevnen, når de kører rapporter. Power BI indeholder værktøjet Effektivitetsanalyse, som hjælper dig med at løse problemer og strømline processen.

Forestil dig et scenarie, hvor du opretter rapporter til salgsteamet i din organisation. Du har importeret data, som findes i flere tabeller i salgsteamets SQL-database, ved at oprette en dataforbindelse til databasen via DirectQuery. Når du opretter foreløbige visualiseringer og filtre, vil du bemærke, at der sendes

forespørgsler til nogle tabeller hurtigere end til andre, og nogle filtre tager længere tid at behandle sammenlignet med andre.

Optimer ydeevnen i Power Query

Ydeevnen i Power Query afhænger af ydeevnen på datakildens niveau. Der er mange forskellige datakilder, som Power Query tilbyder, og teknikkerne til justering af ydeevnen for hver kilde er lige brede. Hvis du f.eks. udtrækker data fra en Microsoft SQL Server, skal du følge vejledningen til justering af ydeevnen for dette produkt. Gode SQL Server teknikker til justering af ydeevnen omfatter oprettelse af indeks, hardwareopgraderinger, justering af udførelsesplan og datakomprimering. Disse emner ligger uden for denne artikels område, og de dækkes kun som et eksempel for at opbygge kendskab til din datakilde og hæste fordelene ved at bruge Power BI og Power Query.

Power Query drager fordel af god ydeevne ved datakilden via en teknik kaldet Forespørgselsdelegering.

Forespørgselsdelegering

Forespørgselsdelegeringen i Power Query-editor hjælper dig med at øge ydeevnen af dine Power BI-rapporter. *Forespørgselsdelegering* er den proces, hvor de transformationer og redigeringer, du foretager i Power Query-editor spores samtidig som oprindelige forespørgsler eller simple **Select** SQL-sætninger, mens du aktivt foretager transformationer. Årsagen til at implementere denne proces er at sikre, at disse transformationer kan finde sted på den oprindelige datakildeserver og ikke overvælde Power BI-beregningsressourcer.

Du kan bruge Power Query til at indlæse data i Power BI. Brug derefter Power Query-editor til at transformere dine data, f.eks. omdøbning eller sletning af kolonner, tilføjelse, fortolkning, filtrering eller gruppering af dine data.

Tænk på et scenarie, hvor du har omdøbt nogle få kolonner i salgsdataene og flettet en kolonne med by og stat i formatet "by stat". I mellemtiden sporer funktionen til forespørgselsdelegering disse ændringer i de oprindelige forespørgsler. Når du derefter indlæser dine data, sker transformationen uafhængigt i den oprindelige kilde, hvilket sikrer, at ydeevnen er optimeret i Power BI.

Fordelene ved forespørgselsdelegering omfatter:

- **Større effektivitet i opdatering af data og trinvis opdateringer.** Når du importerer datatabeller ved hjælp af forespørgselsdelegering, er Power BI bedre i stand til at tildele ressourcer og opdatere dataene hurtigere, fordi Power BI ikke behøver at gennemgå hver transformation lokalt.
- **Automatisk kompatibilitet med DirectQuery- og Dual-lagertilstande.** Alle datakilder i DirectQuery- og Dual-lagertilstande skal have egenskaber til behandling af back end-serveren for at oprette en direkte forbindelse, hvilket betyder, at forespørgselsdelegering er en automatisk egenskab, du kan bruge. Hvis alle transformationer kan reduceres til en enkelt **Select**-sætning, kan forespørgselsdelegering forekomme.

I følgende scenarie vises forespørgselsdelegering i handlingen. I dette scenarie kan du anvende et sæt forespørgsler på flere tabeller. Når du har tilføjet en ny datakilde ved hjælp af Power Query, og du bliver dirigeret til Power Query-editor, skal du gå til ruden **Forespørgselsindstillinger** og højreklikke på det senest anvendte trin, som vist i følgende figur.

▲ APPLIED STEPS

Source	⚙️
Navigation	⚙️
Promoted Headers	⚙️
✕ Changed Type	

⚙️

Edit Settings

🖨️

Rename

✕

Delete

Delete Until End

Insert Step After

⬆️

Move Up

⬇️

Move Down

Extract Previous

📄

View Native Query

📊

Diagnose

📄

Properties...

Hvis indstillingen **Vis oprindelig forespørgsel** ikke er tilgængelig (vises ikke med fed), er forespørgselsdelegering ikke mulig for dette trin, og du skal arbejde bagud i området **Anvendte trin**, indtil du når det trin, hvor **Vis oprindelig forespørgsel** er tilgængelig (vises med fed skrift). Denne proces viser den oprindelige forespørgsel, der bruges til at transformere den semantiske model.

Oprindelige forespørgsler er ikke mulige for følgende transformationer:

- Tilføjelse af en indeksskolonne

- Fletning og tilføjelse af kolonner i forskellige tabeller med to forskellige kilder
- Ændring af en kolonnes datatype

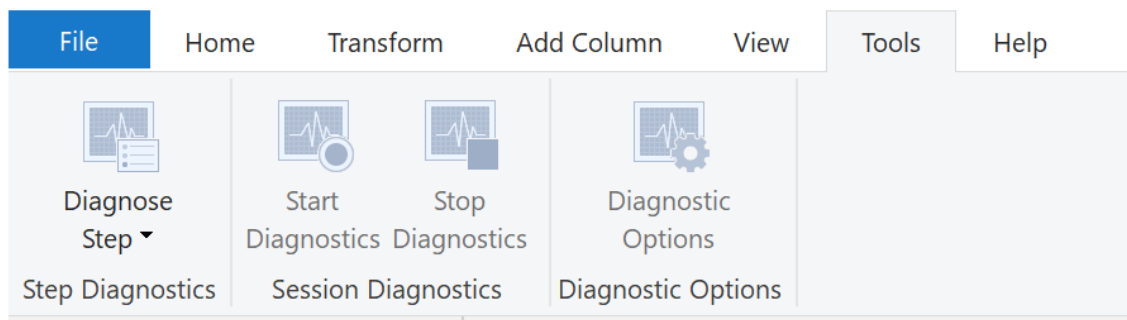
En god tommelfingerregel er, **at hvis du kan overføre en transformation til en SQL-sætning af typen *Select*, som indeholder operatoren og parametre, f.eks. GRUPPER EFTER, SORTER EFTER, HVOR, FOREN ALLE og JOINFORBIND, så kan du bruge forespørgselsdelegering.**

Forespørgselsdelegering er én mulighed for at optimere ydeevnen, når du henter, importerer og forbereder data, men en anden mulighed er forespørgselsdiagnosticering.

Forespørgselsdiagnosticering

Et andet værktøj, som du kan bruge til at undersøge ydeevnen af forespørgslen, er *forespørgselsdiagnosticering*. Du kan afgøre, hvilke flaskehalse der kan være, når du indlæser og transformerer dine data, opdaterer dine data i Power Query, kører SQL-sætninger i Power Query-editor osv.

Du får adgang til forespørgselsdiagnosticering i Power Query-editor ved at gå til **Værktøjer** på båndet Hjem. Når du er klar til at begynde at transformere dine data eller foretage andre ændringer i Power Query-editor, skal du vælge **Start diagnosticering** i afsnittet **Sessionsdiagnosticering**. Når du er færdig, skal du sørge for at vælge **Stop diagnosticering**.



Hvis du vælger **Diagnosticer trin**, kan du se, hvor lang tid det tager at køre det pågældende trin, som vist på følgende billede. Dette valg kan fortælle dig, om et trin tager længere tid at fuldføre end andre, hvilket derefter fungerer som udgangspunkt for yderligere undersøgelse.

Queries [17]							
		Id	A⁰C Query	A⁰C Step	Exclusive Duration	A⁰C Category	A⁰C Data Source Kind
Diagnostics [2]		1	1.1	Product	Changed Type	0.00:00:00.0457545	Evaluator
	Diagnostics Detailed...	2	1.2	Product	Source	0.00:00:01.9567741	Evaluator
	Diagnostics_2020-0...						

Dette værktøj er nyttigt, når du vil analysere ydeevnen på Power Query side for opgaver, f.eks. indlæsning af semantiske modeller, kørsel af dataopdateringer eller kørsel af andre transformerende opgaver.

Andre teknikker til optimering af ydeevnen

Andre måder at optimere ydeevnen af forespørgslen på i Power BI omfatter:

- **Behandl så mange data som muligt i den oprindelige datakilde.** Med Power Query og Power Query-editor kan du behandle dataene, men den behandlingskraft, der skal bruges til at fuldføre denne opgave, kan reducere ydeevnen i andre områder af dine rapporter. **Generelt er det en god idé at behandle så meget som muligt i den oprindelige datakilde.**

- **Brug oprindelige SQL-forespørgsler.** Når du bruger DirectQuery til SQL-databaser, f.eks. i vores scenarie, skal du sørge for, at du ikke henter data fra lagrede procedurer eller almindelige tabeludtryk (CTE'er).
- **Adskil dato og klokkeslæt, hvis de er forbundet.** Hvis nogle af dine tabeller indeholder kolonner, der kombinerer dato og klokkeslæt, skal du sørge for, at du adskiller dem i entydige kolonner, før du importerer dem i Power BI. Denne tilgang vil øge komprimeringsegenskaberne.

Du kan finde flere oplysninger i [Vejledning til forespørgselsdelegering](#) og [Forespørgselsdelegering](#).

Næste undermodul: Løs fejl under import af data

[Tidligere Næste](#)

Har du brug for hjælp? Se i vores [fejlfindingsvejledning](#), eller angiv specifik feedback ved at [rapportere et problem](#).

Løs fejl under import af data

Fuldført 100 XP

- 7 minutter

Under import af data til Power BI kan der opstå fejl, der skyldes faktorer såsom:

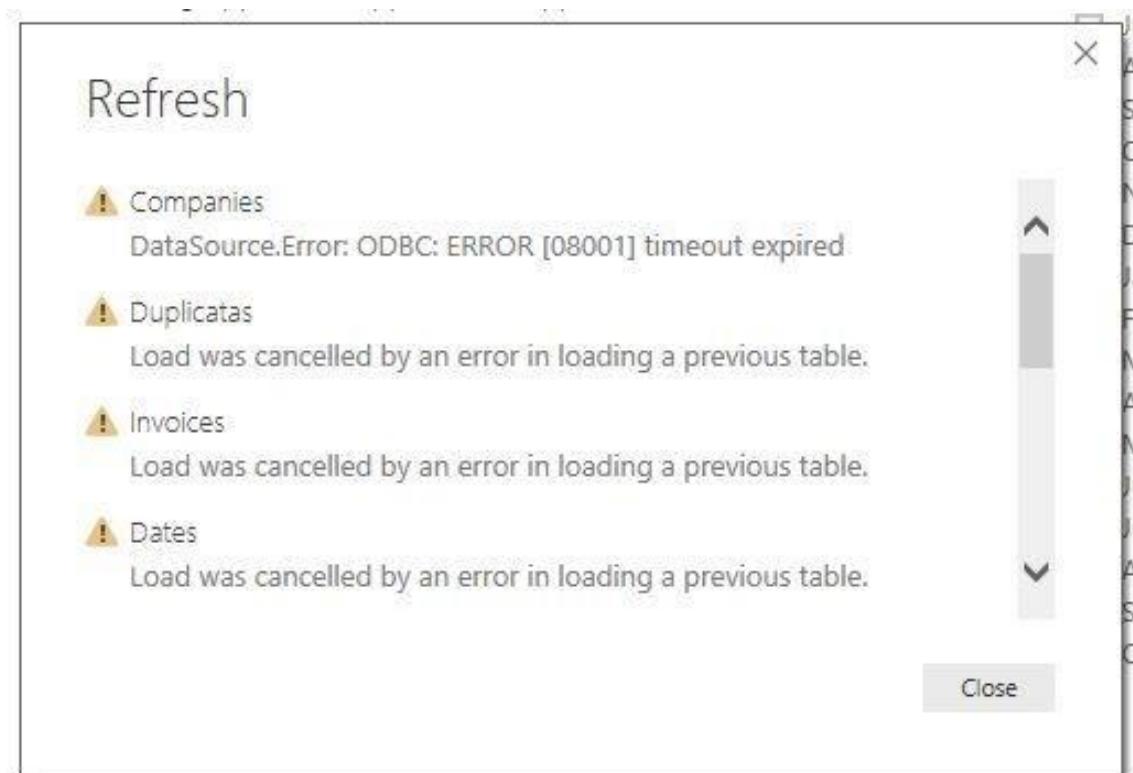
- Power BI importerer fra mange datakilder.
- Hver datakilde kan have adskillige fejlmeddelelser (og nogle gange hundredvis).
- Andre komponenter kan forårsage fejl, f.eks. harddiske, netværk, softwaretjenester og operativsystemer.
- Data kan ofte ikke overholde et bestemt skema.

I følgende afsnit beskrives nogle af de mere almindelige fejlmeddelelser, som du måske får vist i Power BI.

Timeout for forespørgsel er udløbet

I forbindelse med relationskildesystemer er der ofte mange mennesker, som bruger de samme data i den samme database samtidigt. Nogle relationssystemer og deres administratorer forsøger at begrænse en bruger i at lægge beslag på alle hardwareressourcer ved at angive timeout for en forespørgsel. Disse timeouts kan konfigureres til et hvilket som helst tidsinterval fra så lidt som 5 sekunder og helt op til 30 minutter eller mere.

Hvis du f.eks. henter data fra din organisations SQL Server, får du måske vist den fejl, der vises på følgende figur.



Fejl i forespørgsel i Power BI: Timeout er udløbet

Denne fejl angiver, at du har hentet for mange data i henhold til din organisations politikker. Administratorer inkluderer denne politik for at undgå, at et andet program eller en pakke af programmer, som også bruger den pågældende database, bliver langsomt.

Du kan løse denne fejl ved at hente færre kolonner eller rækker fra en enkelt tabel. Selvom du skriver SQL-sætninger, kan det være en almindelig praksis at inkludere grupperinger og sammenlægninger. Du kan også joinforbinde flere tabeller i en enkelt SQL-sætning. Derudover kan du udføre komplicerede underforespørgsler og indlejrede forespørgsler i en enkelt sætning. Disse kompleksiteter udbygger kravene til processen for forespørgsler i relationssystemet og kan i høj grad forlænge implementeringstiden.

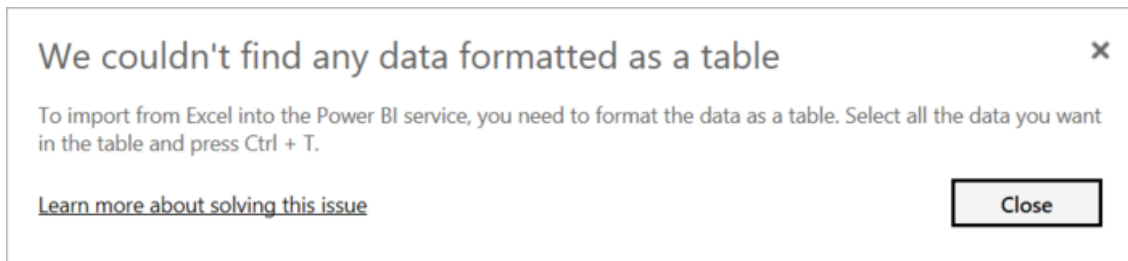
Hvis du har brug for rækker, kolonner og kompleksitet, bør du overveje at tage små datadele og derefter samle dem igen ved hjælp af Power Query. Du kan f.eks. kombinere halvdelen af kolonnerne i én forespørgsel og den anden halvdel i en anden forespørgsel. Power Query kan flette disse to forespørgsler sammen igen, når du er færdig.

Vi kunne ikke finde data, der er formateret som en tabel

Indimellem får du vist fejlen "Vi kunne ikke finde data, der er formateret som en tabel" under import af data fra Microsoft Excel. Denne fejl forklarer heldigvis sig selv. Power BI forventer at finde data, der er formateret som en tabel fra Excel. Fejlen fortæller dig endda løsningen. Udfør følgende trin for at løse problemet:

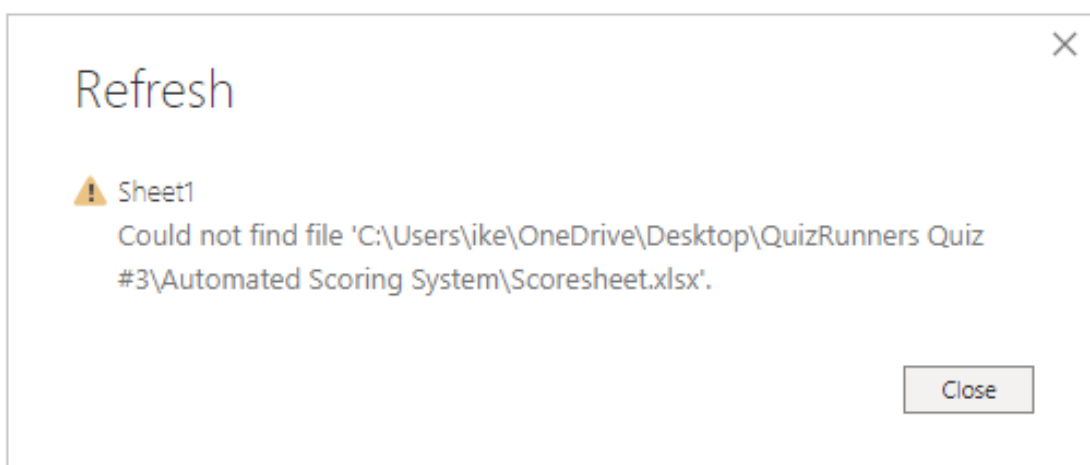
1. Åbn Excel-projektmappen, og fremhæv de data, du vil importere.
2. Tryk på tastaturgenvejen **Ctrl-T**. Den første række vil sandsynligvis være dine kolonneoverskrifter.

3. Bekræft, at kolonneoverskrifterne afspejler den måde, du vil navngive dine kolonner på. Prøv derefter at importere dataene fra Excel igen. Denne gang bør det virke.



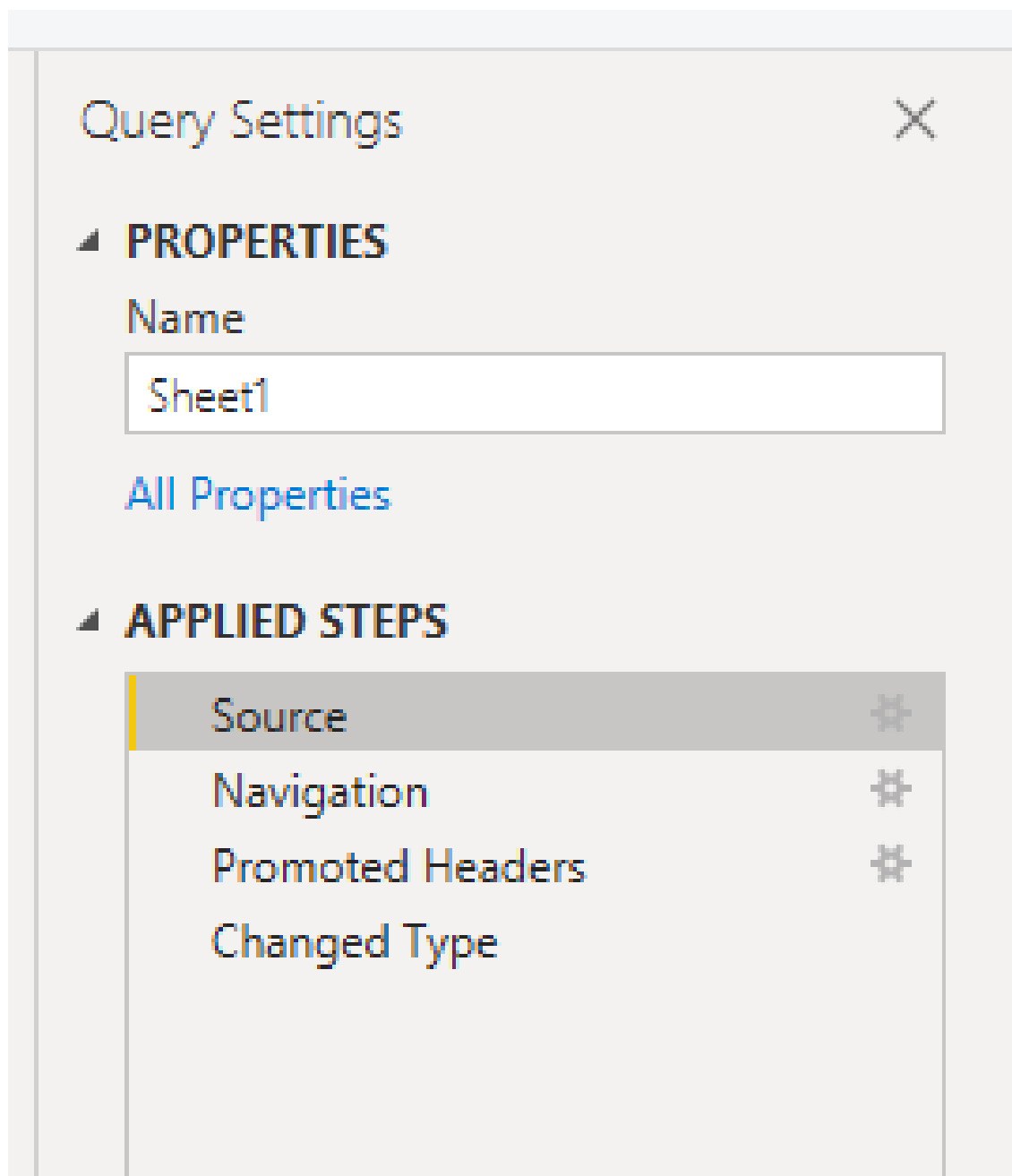
Filen blev ikke fundet

Når du importerer data fra en fil, får du muligvis vist fejlen "Filen blev ikke fundet".

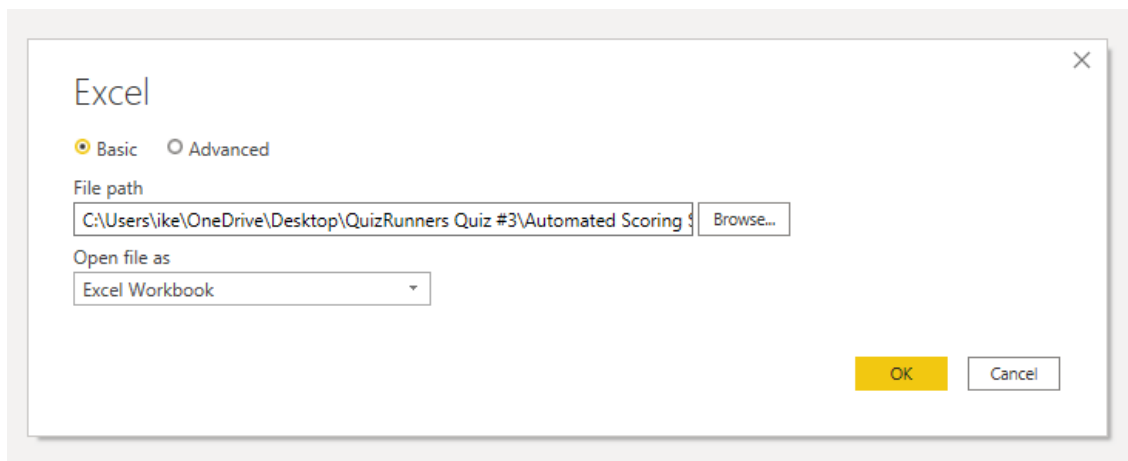


Denne fejl skyldes normalt, at filen er blevet flyttet fra én placering til en anden, eller at tilladelserne til filen er blevet ændret. Hvis årsagen er den førstnævnte, skal du finde filen og ændre kildeindstillingerne.

1. Åbn Power Query ved at vælge knappen **Transformér data** i Power BI.
2. Fremhæv den forespørgsel, der forårsager fejlen.
3. Under **Forespørgselsindstillinger** til venstre skal du vælge tandhjulsikonet ud for **Kilde**.



4. Ret filplaceringen til den nye placering.



Fejl i datatype

Når du importerer data i Power BI, ser kolonnerne nogle gange tomme ud. Denne situation opstår pga. en fejl i fortolkningen af datatypen i Power BI. Løsningen til denne fejl er unik for datakilden. Hvis du f.eks. importerer data fra SQL Server og ser tomme kolonner, kan du prøve at konvertere til den korrekte datatype i forespørgslen.

I stedet for at bruge denne forespørgsel:

```
SELECT CustomerPostalCode FROM Sales.Customers
```

Skal du bruge denne forespørgsel:

```
SELECT CAST(CustomerPostalCode as varchar(10)) FROM Sales.Customers
```

Ved at angive den korrekte type ved datakilden forhindrer du mange af disse almindelige fejl i datakilden.

Du får måske vist forskellige typer fejl i Power BI, der skyldes de alsidige datakildesystemer, hvor dine data opbevares.

Hvis du oplever en fejl, der ikke er dækket, kan du søge i [Microsoft-dokumentationen](#) efter fejlmeddelelsen og den løsning, du har brug for.

Næste undermodul: Øvelse – Forbered data i Power BI Desktop

Øvelse – Forbered data i Power BI Desktop

Fuldført 100 XP

- 30 minutter

Dette undermodul indeholder et laboratorium, der skal gennemføres.

Brug de gratis ressourcer i laboratoriet til at gennemføre øvelserne i dette undermodul. Du bliver ikke faktureret **for laboratoriemiljøet**. Du skal dog muligvis medbringe dit eget abonnement afhængigt af laboratoriet.

Microsoft leverer denne laboratorieøvelse og relateret indhold til uddannelsesmæssige formål. Alle de viste oplysninger ejes af Microsoft og er udelukkende beregnet til at lære om de produkter og tjenester, der er omfattet af dette Microsoft Learn-modul.

Start laboratorie

Den anslåede tid til at fuldføre øvelsen er 30 minutter.

Bemærk

Der leveres en virtuel maskine, der indeholder de klientværktøjer, du har brug for, sammen med øvelsesvejledningen. Brug knappen ovenfor til at starte den virtuelle maskine.

Der er et begrænset antal samtidige sessioner tilgængelige – hvis det hostede miljø ikke er tilgængeligt, kan du prøve igen senere.

Du kan også bruge disse [konfigurationsvejledninger](#) til at oprette dit eget laboratoriemiljø og derefter følge disse [øvelsesinstruktioner](#).

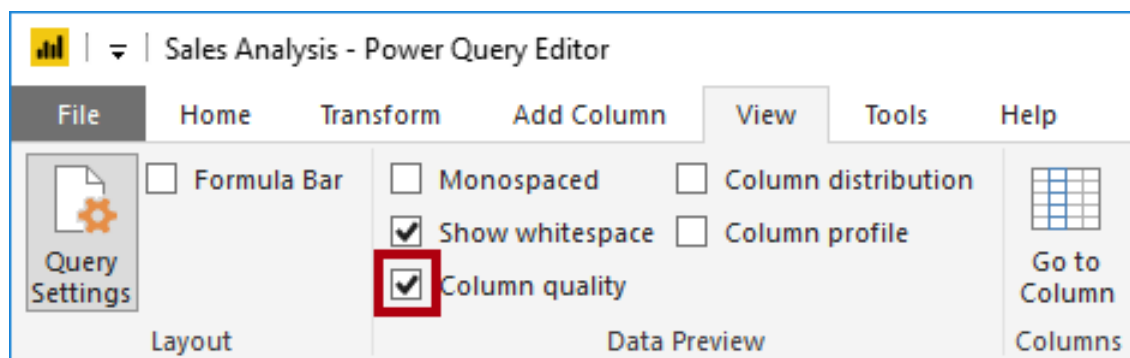
Næste undermodul: Tjek din viden

[Tidligere](#)

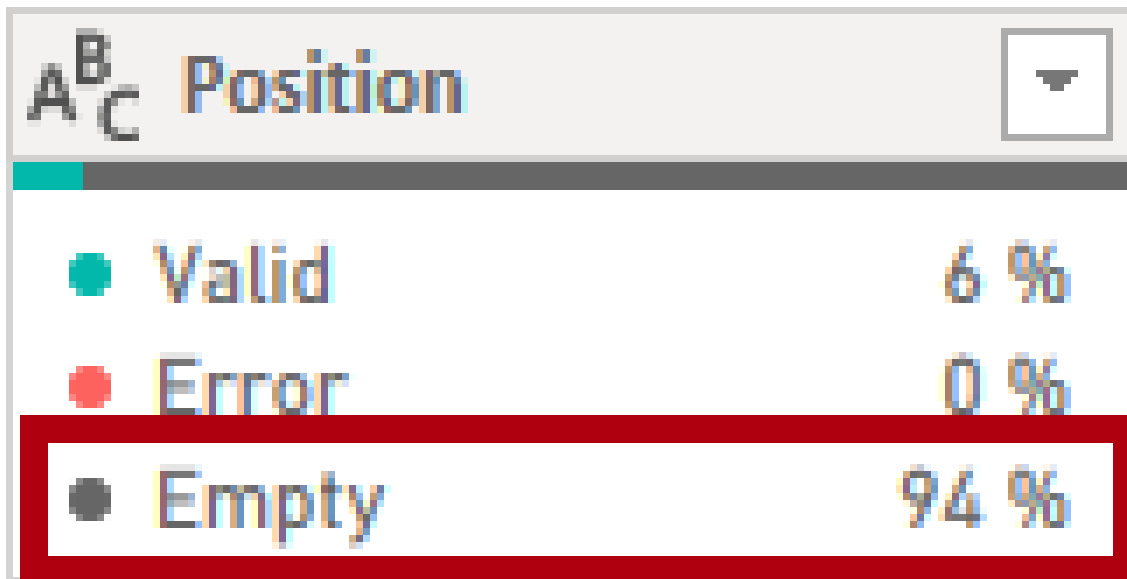
Column Quality, lav check ved dataindlæsning

MS-CorrelationId: 00000000-5b86-46cf-bb56-b45ef2fcef69

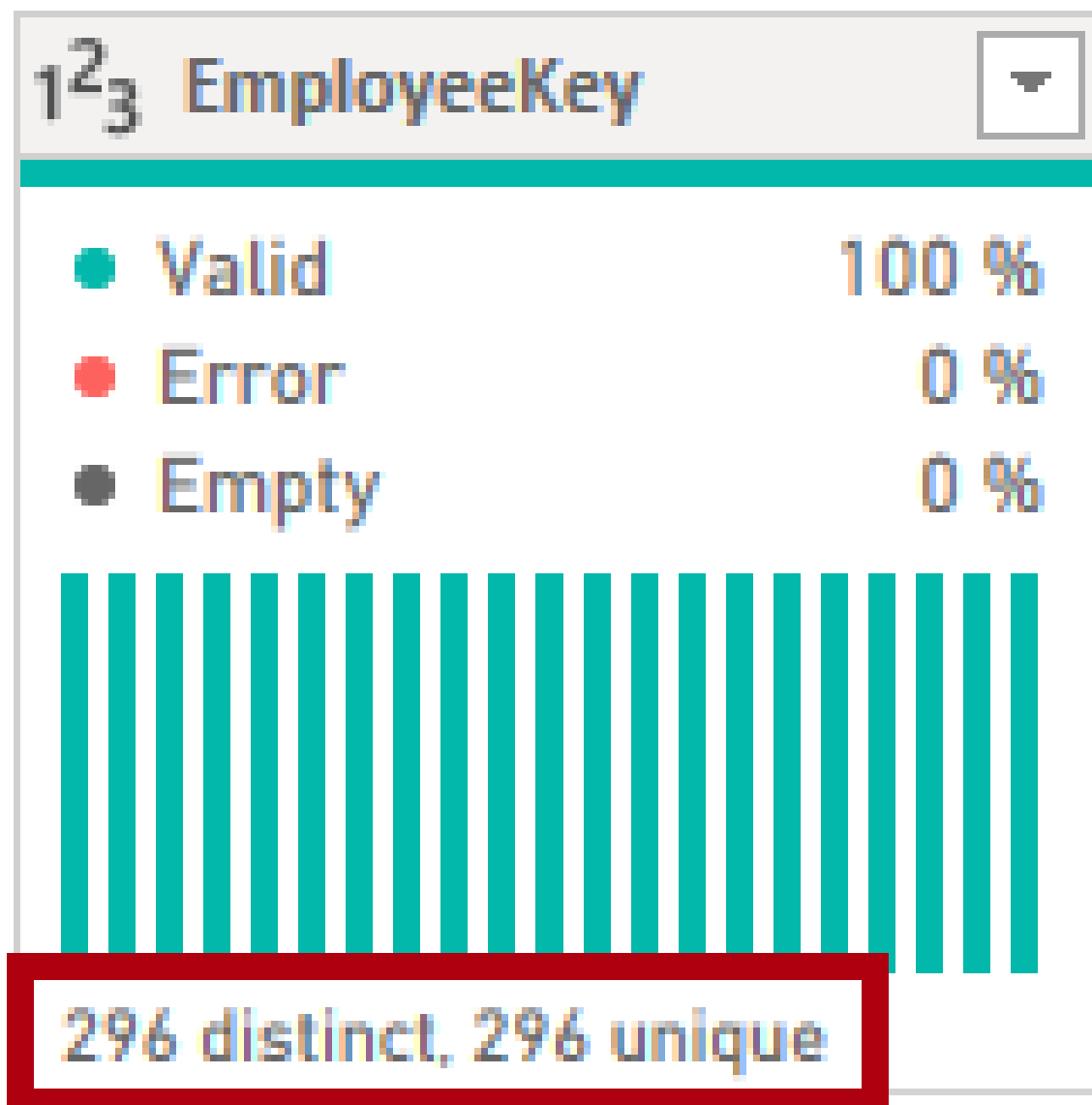
1. To assess column quality, on the **View** ribbon tab, from inside the **Data Preview** group, check **Column Quality**. The column quality feature allows you to easily determine the percentage of valid, error, or empty values found in columns.



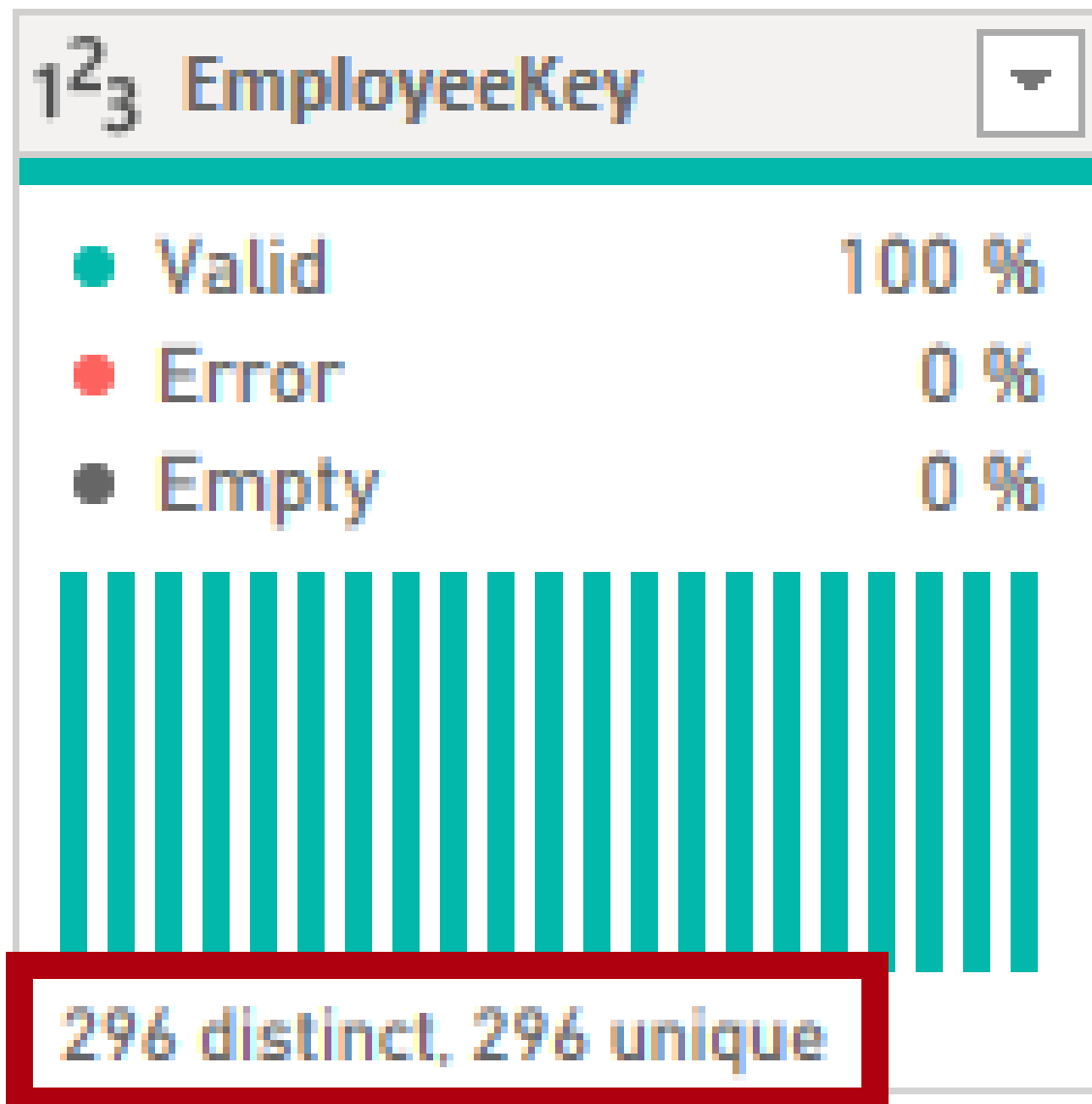
2. Notice that the **Position** column has 94% empty (null) rows.



3. To assess column distribution, on the **View** ribbon tab, from inside the **Data Preview** group, check **Column Distribution**.
4. Review the **Position** column again, and notice that there are four distinct values, and one unique value.
5. Review the column distribution for the **EmployeeKey** column—there are 296 distinct values, and 296 unique values.



1. Review the **Position** column again, and notice that there are four distinct values, and one unique value.
2. Review the column distribution for the **EmployeeKey** column—there are 296 distinct values, and 296 unique values.



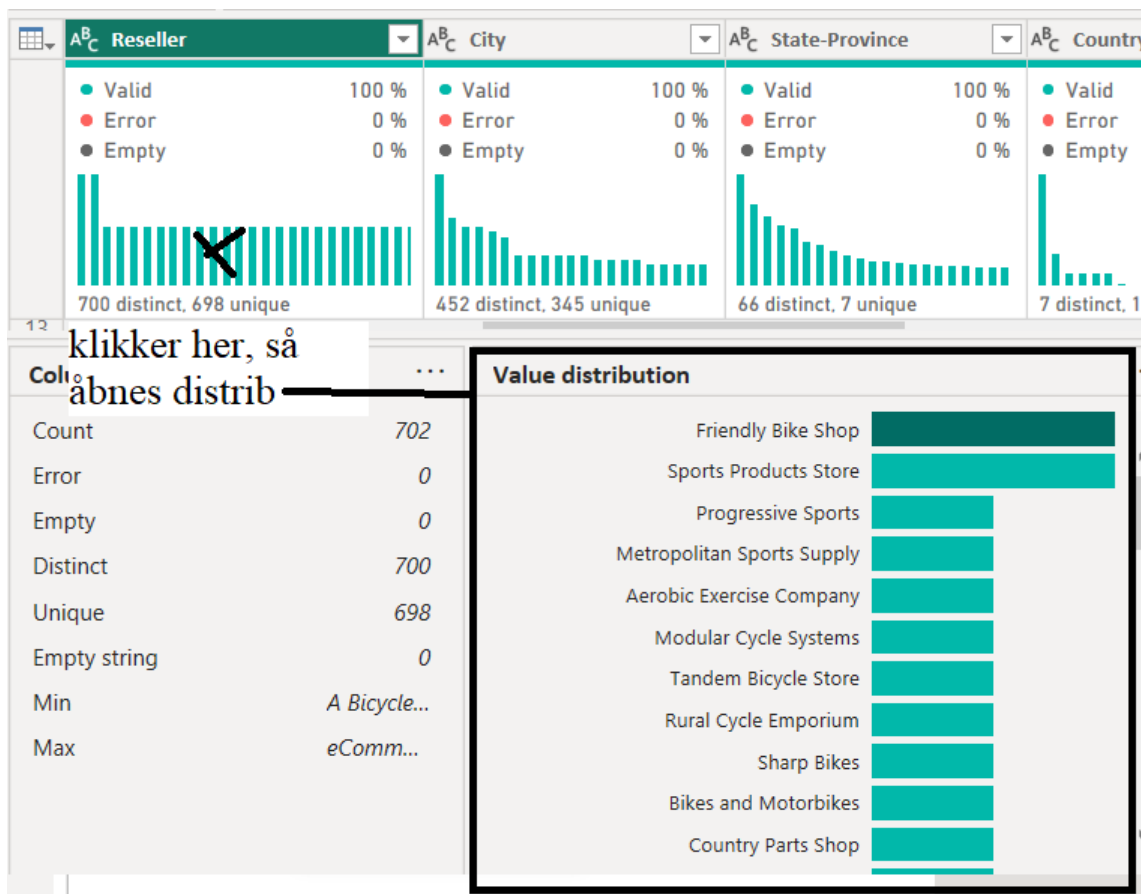
Note: When the distinct and unique counts are the same, it means the column contains unique values. When modeling, it's important that some model tables have unique columns. These unique columns can be used to create one-to-many relationships, which you'll do in the **Model Data in Power BI Desktop** lab.

3. In the **Queries** pane, select the **DimProduct** query.

The **DimProduct** table contains one row per product sold by the company.

4. In the **Queries** pane, select the **DimReseller** query.

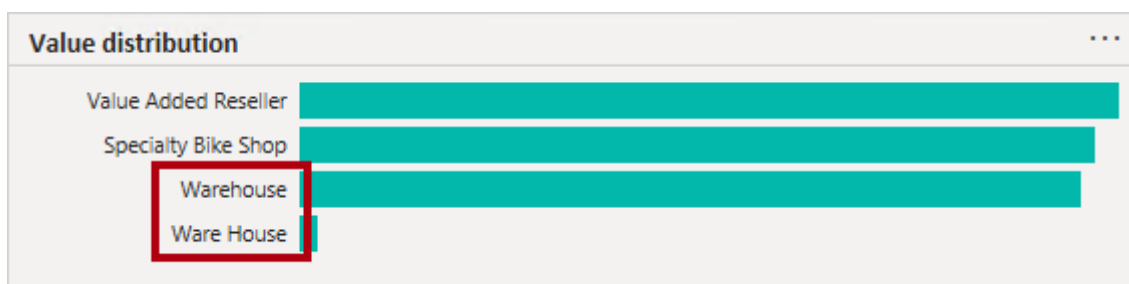
The **DimReseller** table contains one row per reseller. Resellers sell, distribute, or value add to the Adventure Works products.



To view column values, on the **View** ribbon tab, from inside the **Data Preview** group, check **Column Profile**.

5. Select the **BusinessType** column header, and notice the new pane beneath the data preview pane. Review the column statistics and value distribution in the data preview pane.

*Notice the data quality issue: there are two labels for warehouse (**Warehouse**, and the misspelled **Ware House**).*



6. Hover the cursor over the **Ware House** bar, and notice that there are five rows with this value.
7. In the **Queries** pane, select the **DimSalesTerritory** query.

*The **DimSalesTerritory** table contains one row per sales region, including **Corporate HQ** (headquarters). Regions are assigned to a country, and countries are assigned to groups. In the **Model Data in Power BI Desktop** lab, you'll create a hierarchy to support analysis at region, country, or group level.*

8. In the **Queries** pane, select the **FactResellerSales** query.

The **FactResellerSales** table contains one row per sales order line—a sales order contains one or more line items.

9. Review the column quality for the **TotalProductCost** column, and notice that 8% of the rows are empty.

Missing **TotalProductCost** column values is a data quality issue.

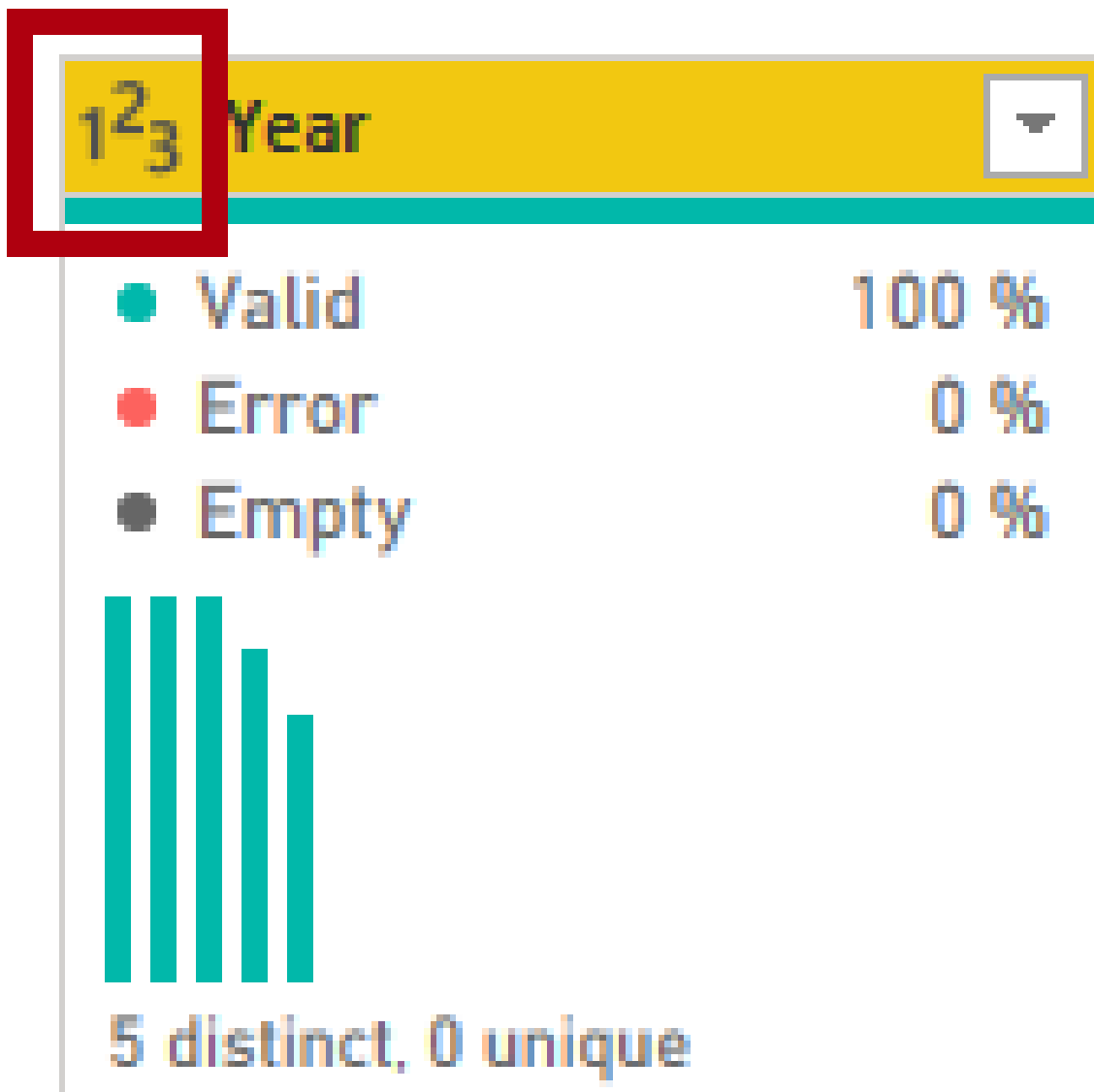
Get data from a CSV file

In this task, you'll create a new query based on CSV files.

1. To add a new query, in the **Power Query Editor** window, on the **Home** ribbon tab, from inside the **New Query** group, select the **New Source** down-arrow, and then select **Text/CSV**.
2. Navigate to the **01-prepare-data > ResellerSalesTargets.csv** file. Select **Open**.
3. In the **ResellerSalesTargets.csv** window, review the preview data. Select **OK**.
4. In the **Queries** pane, notice the addition of the **ResellerSalesTargets** query.

The **ResellerSalesTargets** CSV file contains one row per salesperson, per year. Each row records 12 monthly sales targets (expressed in thousands). The business year for the Adventure Works company commences on July 1.

5. Notice that no column contains empty values. When there isn't a monthly sales target, a hyphen character is stored instead.
6. Review the icons in each column header, to the left of the column name. The icons represent the column data type. **123** is whole number, and **ABC** is text.



7. Repeat the steps to create a query based on the **ColorFormats.csv** file.

The **ColorFormats** CSV file contains one row per product color. Each row records the HEX codes to format background and font colors.

You should now have two new queries, **ResellerSalesTargets** and **ColorFormats**.

Queries [8]



DimEmployee



DimEmployeeSalesTerritory



DimProduct



DimReseller



DimSalesTerritory



FactResellerSales



ResellerSalesTargets



ColorFormats

videnstest

1.

Når du opretter forbindelse til en SQL Server database for at hente data, hvilket sprog skal du bruge til at udtrække data?

DAX

T-SQL

Korrekt. T-SQL er det forespørgselssprog, du skal bruge til SQL Server.

MDX

2.

Du opretter en Power BI-rapport med data fra en Azure Analysis Services MDX-kube. Når dataene opdateres i kuben, vil du gerne se dem øjeblikkeligt i Power BI-rapporten. Hvordan skal du oprette forbindelse?

Importér

T-SQL

Direkte forbindelse

Korrekt. Dette afspejler ændringer af kubens øjeblikkeligt.

3.

Hvad kan du gøre for at forbedre ydeevnen, når du henter data i Power BI?

Fjern unødvendige kolonner og rækker

Korrekt. Brug altid den mindste mængde data, der kræves til dit projekt.

Eksportér databasefiler til CSV for at indlæse

Kombiner kolonner med dato og klokkeslæt i en enkelt kolonne

Resultater

Fortsæt det fremragende arbejde



Få data i Power BI

Du har opnået et resultat!

Tillykke, men hvad skal du gøre nu?

Lad os først dele dit resultat

Du afsætter tid til at lære noget nyt, så lad dit netværk tage del i din succes.



<https://learn.microsoft.com/api/achievements/share/da-dk/bjarnehansen-7412/YQ6JPJDR?sharingId=39714CB1FE13C67B>

**** Design en semantisk model i Power BI

- 1 t. 23 min.
- Modul
- 10 Enheder

Feedback

Øvet

Dataanalytiker

Power BI

Microsoft Power Platform

Processen med at oprette en kompliceret semantisk model i Power BI er ligetil. Hvis dine data kommer fra mere end ét transaktionssystem, kan du, endelig, arbejde med en masse tabeller, før du ved af det. Oprettelse af en stor semantisk model handler om at forenkle uorden. Et stjerneschema er én måde at forenkle en semantisk model på, og du får mere at vide om terminologien og implementeringen af den i dette modul. Du får også mere at vide om, hvorfor det er vigtigt at vælge den korrekte datagranularitet, hvad angår Power BI-rapporters effektivitet og brugbarhed. Endelig får du mere at vide om, hvordan du forbedrer ydeevnen med dine semantiske Power BI-modeller.

Læringsmål

I dette modul kommer du til at:

- Oprette fælles datotabeller
- Konfigurere mange til mange-relationer
- Løse problemer med cirkulære relationer
- Designe stjerneschemaer

[Start](#)Tilføj

-
-
-

Forudsætninger

Ingen

Dette modul er en del af disse læringsforløb

- [Develop dynamic reports with Microsoft Power BI](#)
- [Modeldata med Power BI](#)
- [Validate your Microsoft Power Platform Solution Architect skills](#)
- [Introduktion](#)5 min.
- [Arbejd med tabeller](#)3 min.
- [Opret en datotabel](#)9 min.
- [Arbejd med dimensioner](#)6 min.
- [Definer datagranularitet](#)4 min.
- [Arbejde med relationer og kardinalitet](#)5 min.

- [Løs modelleringsudfordringer](#) 2 min.
- [Øvelse – Modeldata i Power BI Desktop](#) 45 min.
- [Tjek din viden](#) 3 min.
- [Resumé](#) 1 min.

Introduktion

Fuldført 100 XP

- 5 minutter

Oprettelse af en fantastisk semantisk model er en af de vigtigste opgaver, som en dataanalytiker kan udføre i Microsoft Power BI. Når du er god til det, kan du gøre det nemmere for andre at forstå dine data, hvilket vil gøre det nemmere for både brugerne og dig selv at bygge værdifulde Power BI-rapporter.

Siderne i dette modul er kun vejledende. Der er ingen datafiler. Du har en chance for at arbejde med rigtige data i laboratorierne.

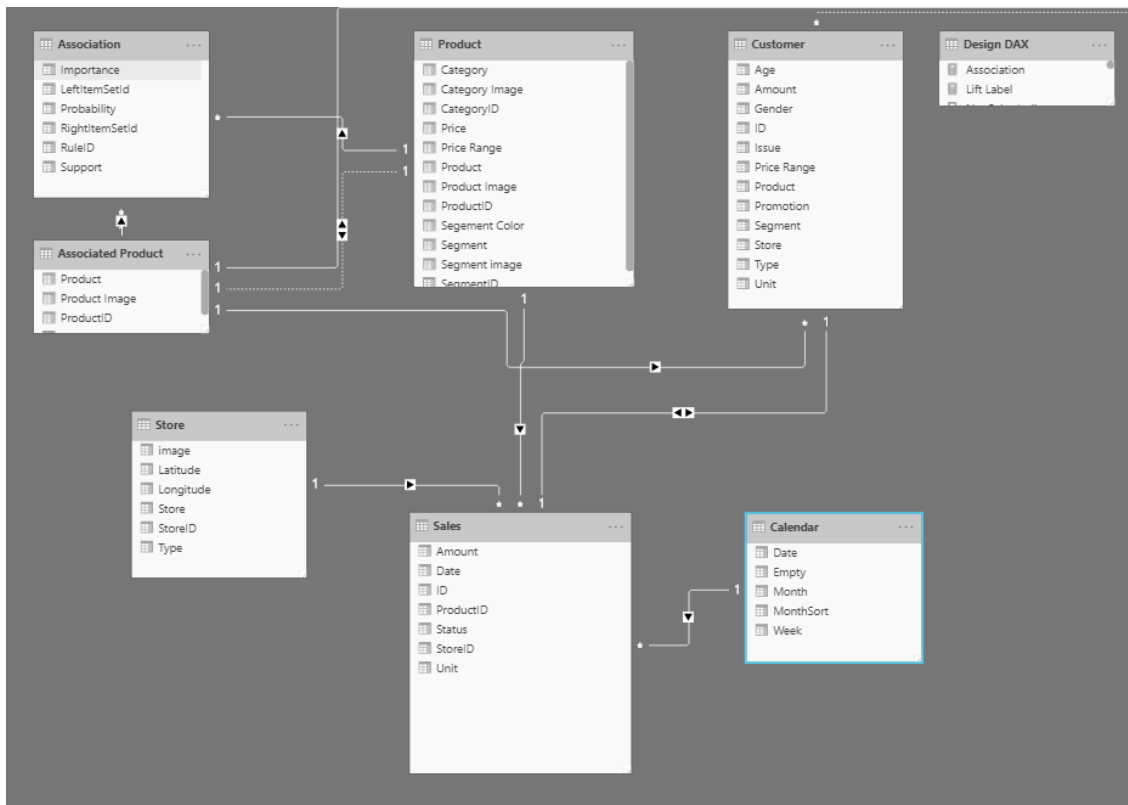
En god semantisk model giver følgende fordele:

- Udforskning af data er hurtigere.
- Det er nemmere at oprette sammenlægninger.
- Rapporter er mere nøjagtige.
- Det er hurtigere at skrive rapporter.
- Det er nemmere at vedligeholde rapporter fremover.

Det er svært at angive regler for, hvad der gør en god semantisk model, fordi alle data er forskellige, og brugen af disse data varierer. **Generelt er en mindre semantisk model bedre, fordi den fungerer hurtigere og er nemmere at bruge.** Det er dog lige så problematisk at definere, hvad en mindre semantisk model indebærer, fordi det er et heuristisk og subjektivt koncept.

En mindre semantisk model består typisk af færre tabeller og færre kolonner i hver tabel, som brugeren kan se. Hvis du importerer alle de nødvendige tabeller fra en salgsdatabase, men det samlede antal tabeller er 30 tabeller, vil brugeren ikke finde den intuitiv. Hvis du skjuler disse tabeller i fem tabeller, bliver den semantiske model mere intuitiv for brugeren, hvorimod den kan være overvældende, hvis brugeren åbner en tabel og finder 100 kolonner. Hvis du fjerner unødvendige kolonner for at give et mere håndterbart antal, øges sandsynligheden for, at brugeren læser alle kolonnenavne. For at opsummere skal du stræbe efter enkelhed, når du designer dine semantiske modeller.

Følgende billede er et eksempel på en semantisk model. Felterne indeholder tabeller med data, hvor hvert linjeelement i feltet er en kolonne. De linjer, der forbinder felterne, repræsenterer relationer mellem tabellerne. Disse relationer kan være komplekse, selv i en sådan forenklet model. Den semantiske model kan nemt blive uorganiseret, **og det samlede antal tabeller i modellen kan gradvist stige. Det kræver en konstant indsats at holde din semantiske model enkel, omfattende og præcis.**



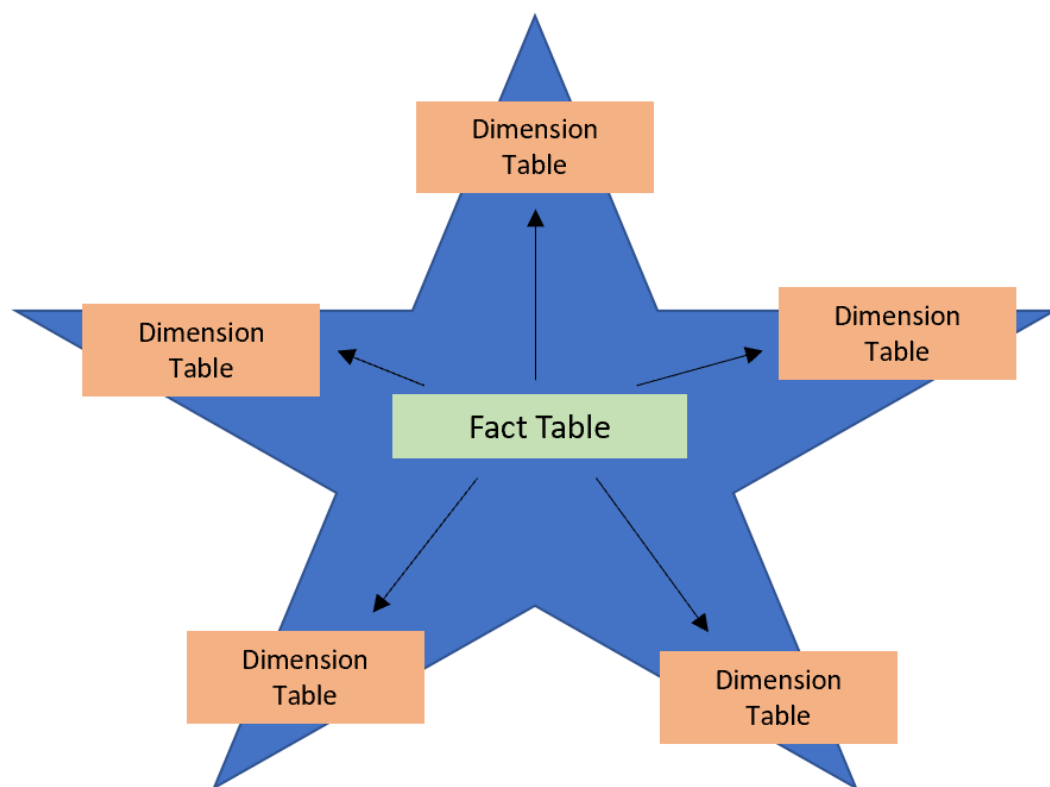
Relationer defineres mellem tabeller via primære og fremmede nøgler. Primære nøgler er kolonner, der identificerer hver entydig datarække, der ikke er null. Hvis du f. eks. har tabellen Customers, kan du have et indeks, der identificerer hver enkelt kunde. Den første række har id'et 1, den anden række et id på 2 osv. Hver række tildeles en entydig værdi, som denne enkle værdi kan referere til: den primære nøgle. Denne proces bliver vigtig, når du refererer til rækker i en anden tabel, hvilket er det, fremmede nøgler gør. Der dannes relationer mellem tabeller, når forskellige tabeller har primære og fremmede nøgler tilfælles.

Power BI gør det muligt at bygge relationer på basis af tabeller med forskellige datakilder, en effektiv funktion, der gør det muligt for dig at hente én tabel fra Microsoft Excel og en anden fra en relationsdatabase. Du skal derefter oprette relationen mellem disse to tabeller og behandle dem som en samlet semantisk model.

Nu, hvor du har lært om de relationer, der udgør dataskemaet, kan du udforske en bestemt type skemadesign, stjerneschemaet, som er optimeret til høj ydeevne og anvendelighed.

Stjerneschemaer

Du kan designe et stjerneschema for at forenkle dine data. Det er ikke den eneste måde at forenkle dine data på, men det er en populær metode. Derfor skal alle Power BI-dataanalytiker kende den. I et stjerneschema er hver tabel i din semantiske model defineret som en dimension eller en faktatabel, som vist i følgende visualisering.



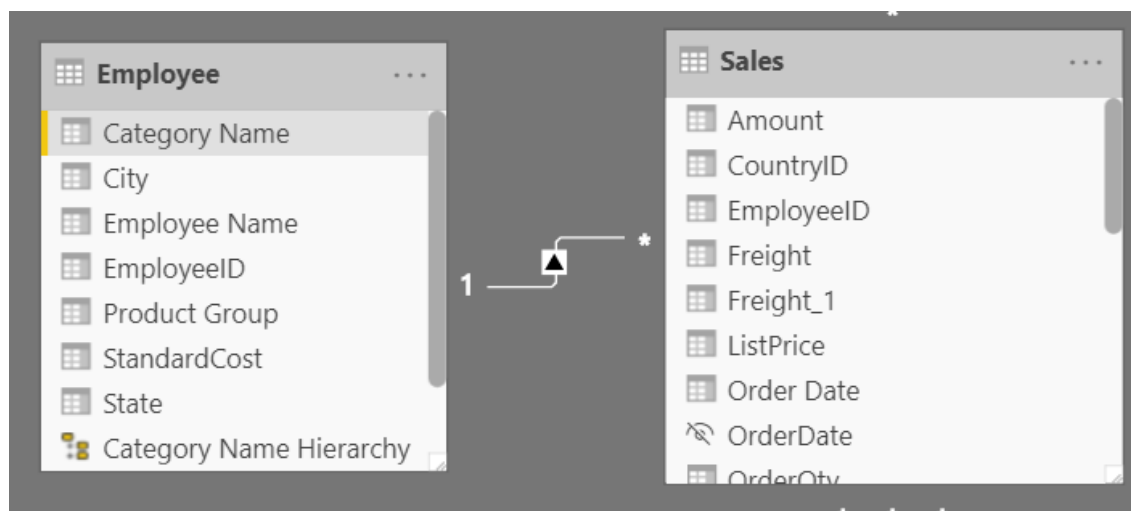
Faktatabeller indeholder observatoriske dataværdier eller hændelsesdataværdier: salgsordrer, produktoptællinger, priser, transaktionsdatoer og -klokkeslæt samt antal. Faktatabeller kan indeholde flere gentagne værdier. Et produkt kan f. eks. optræde flere gange i flere rækker for forskellige kunder på forskellige datoer. Disse værdier kan samles i visuals. Et visual for de samlede salgsordrer er f. eks. en sammenlægning af alle salgsordrer i faktatabellen. I faktatabeller er det almindeligt at se kolonner, der er udfyldt med tal og datoer. Tallene kan være måleenheder, f. eks. salgsbeløb, eller de kan være nøgler, f. eks. et kunde-id. Datoerne repræsenterer den tid, der registreres, f. eks. ordredato eller afsendelsesdato.

Dimensionstabeller indeholder oplysninger om dataene i faktatabeller: produkter, placeringer, medarbejdere og ordretyper. Disse tabeller har forbindelse til faktatabellen vha. nøglekolonner. Dimensionstabeller bruges til at filtrere og gruppere dataene i faktatabeller. Faktatabellerne indeholder derimod de målbare data, f.eks. salg og indtægt, og hver række repræsenterer en entydig kombination af værdier fra dimensionstabellerne. Du kan gruppere dataene i visual'et for de samlede salgsordrer, så du kan se de samlede salgsordrer efter produkt, hvor produktet er data i dimensionstabellen.

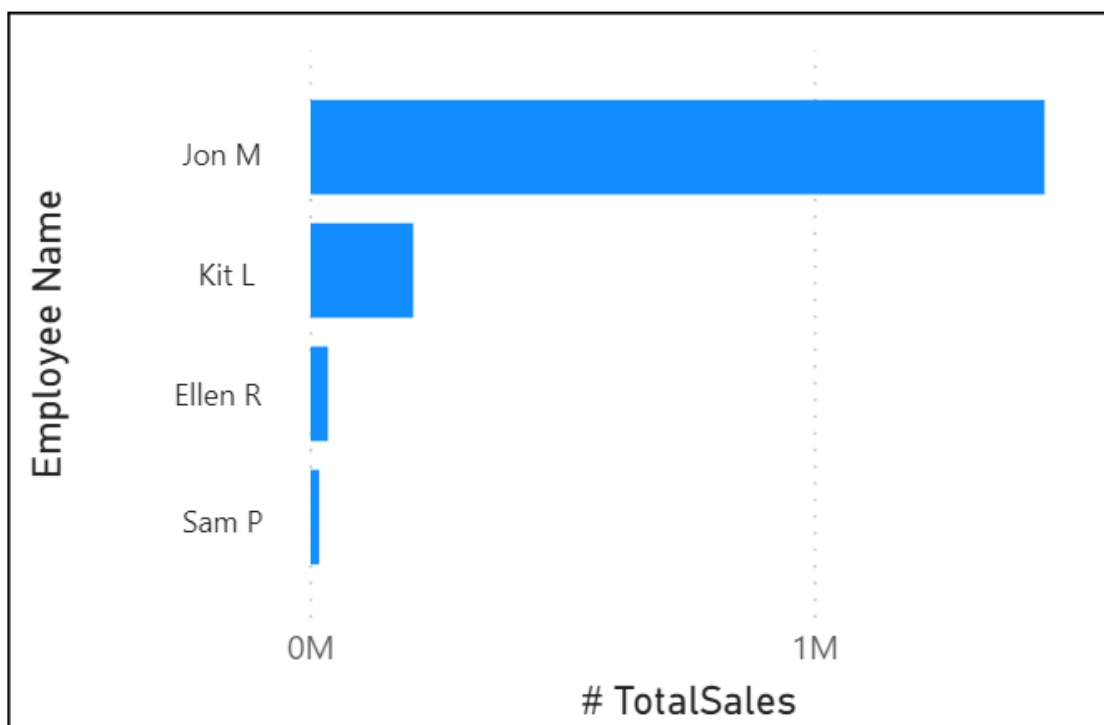
Faktatabeller er meget større end dimensionstabeller, fordi der forekommer mange hændelser i faktatabeller, f.eks. individuelt salg. Dimensionstabeller er typisk mindre, da du er begrænset til det antal elementer, du kan filtrere og gruppere på. Et år indeholder f.eks. kun så mange måneder, og USA består kun af et bestemt antal stater.

Nu hvor du har fået disse oplysninger om faktatabeller og dimensionstabeller, tænker du måske på, hvordan du kan bygge dette visual i Power BI.

De relevante data findes i to tabeller, Employee og Sales, som vist i følgende semantiske model. Da tabellen Sales indeholder salgsordreværdierne, der kan samles, anses den for at være en faktatabel. Tabellen Employees indeholder det specifikke medarbejdernavn, som filtrerer salgsordrerne, og den er derfor en dimensionstabel. Den kolonne, de to tabeller har tilfælles, hvilket er den primære nøgle i tabellen Employee, er **EmployeeID**, og du kan oprette en relation mellem de to tabeller på basis af denne kolonne.



Når du opretter denne relation, kan du bygge visual'et i henhold til kravene, som vist i følgende figur. Hvis du ikke oprettede denne relation, samtidig med at du er opmærksom på den kolonne, de to tabeller har tilfælles, ville det være sværere at oprette visual'et.



Stjerneschemaer og den underliggende semantiske model er grundlaget for organiserede rapporter. Jo mere tid du bruger på at oprette disse forbindelser og designe, jo nemmere er det at oprette og vedligeholde rapporter.

Næste undermodul: Arbejd med tabeller

[Næste](#)

Arbejd med tabeller

Fuldført 100 XP

- 3 minutter

Når brugerne ser færre tabeller, vil de få glæde af at bruge din semantiske model betydeligt mere. Antag f. eks., at du har importeret mange tabeller fra mange datakilder, og at visual'et nu ser rodet ud. I dette tilfælde skal du sikre, at din semantiske model og tabelstruktur forenkles, før du begynder at arbejde med at oprette rapporter.

En enkel tabelstruktur har følgende fordele:

- Den er nem at navigere i på grund af de kolonne- og tabelegenskaberne, der er specifikke og brugervenlige.
- Den har flettede eller tilføjede tabeller for at forenkle tabellerne i datastrukturen.
- Den har gode meningsfyldte relationer mellem tabellerne.

I følgende afsnit forklares det yderligere, hvordan du kan arbejde med tabellerne for at sikre en enkel og læsbar tabelstruktur.

Konfigurer semantisk model, og opret relationer mellem tabeller

Hvis du allerede har hentet dine data og rensset dem i Power Query, kan du gå til fanen **Model**, hvor den semantiske model er placeret. På følgende billede kan du se, hvordan relationen mellem tabellerne **Order** og **Sales** kan ses via kolonnen **OrderDate**.

med at oprette relationer mellem kolonner, der er navngivet på samme måde. Relationer kan være inaktive eller aktive. Der kan kun være én aktiv relation mellem tabeller, hvilket er beskrevet i et fremtidigt modul.

Selvom funktionen **Administrer relationer** gør det muligt for dig at konfigurere relationer mellem tabeller, kan du også konfigurere egenskaber for tabeller og kolonner for at sikre organiseringen af din tabelstruktur.

[Konfigurer egenskaber for tabeller og kolonner](#)

Visningen **Model** i Power BI Desktop omfatter mange indstillinger for kolonneegenskaber, du kan få vist eller opdatere. En enkel metode til at åbne denne menu for at opdatere tabellerne og felterne er ved at Ctrl+klikke eller Skift+klikke på elementer på denne side.

Properties



^ General

Name

Date

Description

Enter a description

Synonyms

date

Display folder

Enter the display folder

Is hidden

No ☐

[Skriv her]

Under fanen **generelt** kan du:

- Redigere navnet på og beskrivelsen af denne kolonne.
- Tilføje synonymer, der kan bruges til at identificere kolonnen, når du bruger funktionen Spørgsmål og svar.
- Tilføje en kolonne i en mappe for at organisere tabelstrukturen yderligere.
- Skjule eller vise kolonnen.

Under fanen **Formatering** kan du:

- Ændre datatype.
- Formatere datoen.

Lad os f. eks. antage, at datoerne i din kolonne er formateret som vist på det forrige skærmbillede i formatet af "Wednesday, March 14, 2001". Hvis du vil ændre formatet, så datoen er i formatet "dd/mm/åååå", skal du vælge rullemenuen under **Alle dato-klokkeslætformater** og derefter vælge det relevante datoformat som vist i følgende figur.

Custom format

Custom

All date formats

*3/14/2001 (m/d/yyyy)

Wednesday, March 14, 2001 (dddd, mmmm d, yyyy)

March 14, 2001 (mmmm d, yyyy)

Wednesday, 14 March, 2001 (dddd, d mmmm, yyyy)

14 March, 2001 (d mmmm, yyyy)

3/14/2001 (m/d/yyyy)

3/14/01 (m/d/yy)

03/14/01 (mm/dd/yy)

03/14/2001 (mm/dd/yyyy)

01/03/14 (yy/mm/dd)

2001-03-14 (yyyy-mm-dd)

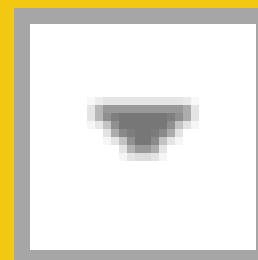
14-Mar-01 (dd-mmm-yy)

March 2001 (mmmm yyyy)

March 14 (mmmm d)

Når du har valgt det rette datoformat, skal du vende tilbage til kolonnen **Date**, hvor du kan se, at formatet er ændret, som vist i følgende figur.

Date



01/01/2019

01/06/2019

01/13/2019

01/20/2019

01/27/2019

[Skriv her]

Under fanen **Avanceret** kan du:

- Sortere efter en bestemt kolonne.
- Tildele en bestemt kategori til dataene.
- Opsummere dataene.
- Finde ud af, om kolonnen eller tabellen indeholder null-værdier.

Desuden har Power BI en ny funktionalitet, som du kan bruge til opdatering af disse egenskaber i mange tabeller og felter ved at Ctrl+klikke eller Skift+klikke på elementer.

Disse eksempler er kun nogle af de mange typer transformationer, du kan foretage for at forenkle tabelstrukturen. Dette trin er vigtigt at udføre, før du begynder at oprette visuals, så du ikke behøver at flytte frem og tilbage, når du foretager formateringsændringer. Denne proces til formatering og konfiguration af tabeller kan også udføres i Power Query.

Næste undermodul: Oprette en datotabel

Oprette, generer en fælles dato-tabel

Du kan opbygge en fælles datotabel på følgende måder:

- Kildedata
- DAX
- Power Query

Kildedata

I enkelte tilfælde har kildedatabaser og datalagersteder allerede deres egne datotabeller. Hvis den administrator, der har designet databasen, har gennemført et grundigt job, kan disse tabeller bruges til at udføre følgende opgaver:

- Identifikation af firmaferier
- Separat kalender og regnskabsår
- Identificer weekender vs. ugedage

Kildedatatabeler er modne og klar til omgående brug. Hvis du har en tabel som sådan, skal du tage den med i din semantiske model og ikke bruge andre metoder, der beskrives i dette afsnit. Det anbefales, at du bruger en kildedatotabel, da den sandsynligvis deles med andre værktøjer, som du bruger sammen med Power BI.

Hvis du ikke har en kildedatatabel, kan du bygge en fælles datotabel på andre måder.

DAX

Du kan bruge DAX-funktionerne `KALENDERAUTO()` eller `KALENDER()` til at opbygge din fælles datotabel. Funktionen `KALENDER()` returnerer et sammenhængende datointerval baseret på en start- og slutdato, der er angivet som argumenter i funktionen. Alternativt returnerer funktionen `CALENDARAUTO()` et

sammenhængende, komplet datointerval, der automatisk bestemmes ud fra din semantiske model. Startdatoen vælges som den tidligste dato, der findes i din semantiske model, og slutdatoen er den seneste dato, der findes i din semantiske model plus data, der er udfyldt i den regnskabsmåned, du kan vælge at medtage som et argumenter i funktionen KALENDERAUTO(). I dette eksempel bruges funktionen KALENDER(), fordi du kun vil have vist data fra 31. maj 2011 (den første dag, hvor Salg begyndte at spore disse data) og fremad for de næste 10 år.

I Power BI Desktop skal du **vælge Ny** tabel og derefter angive den i følgende DAX-formel:

DAX Kopiér

Dates = CALENDAR(DATE(2011, 5, 31), DATE(2022, 12, 31))

1 Dates = CALENDAR(DATE(2011, 5, 31), DATE(2022, 12, 31))

Du har nu en kolonne med datoer, du kan bruge. Men denne kolonne er dog lidt anderledes. Du vil også have vist kolonnerne for kun året, månedsnummeret, ugen i året og ugedagen. Du kan udføre denne opgave ved **at vælge Ny** kolonne på båndet og indtaste følgende DAX-ligning, der henter året fra datotabellen.

DAX Kopiér

Year = YEAR(Dates[Date])

✕ ✓ 1 Year = YEAR(Dates[Date])	
Date	Year
Tuesday, May 31, 2011	2011
Wednesday, June 1, 2011	2011
Thursday, June 2, 2011	2011
Friday, June 3, 2011	2011
Saturday, June 4, 2011	2011

Du kan udføre samme proces for at hente månedsnummer, ugenummer og dag i ugen:

DAXKopiér

MonthNum = MONTH(Dates[Date])

DAXKopiér

WeekNum = WEEKNUM(Dates[Date])

DAXKopier

DayoftheWeek = FORMAT(Dates[Date], "DDDD")

Når du er færdig, indeholder tabellen de kolonner, der er vist i følgende figur.

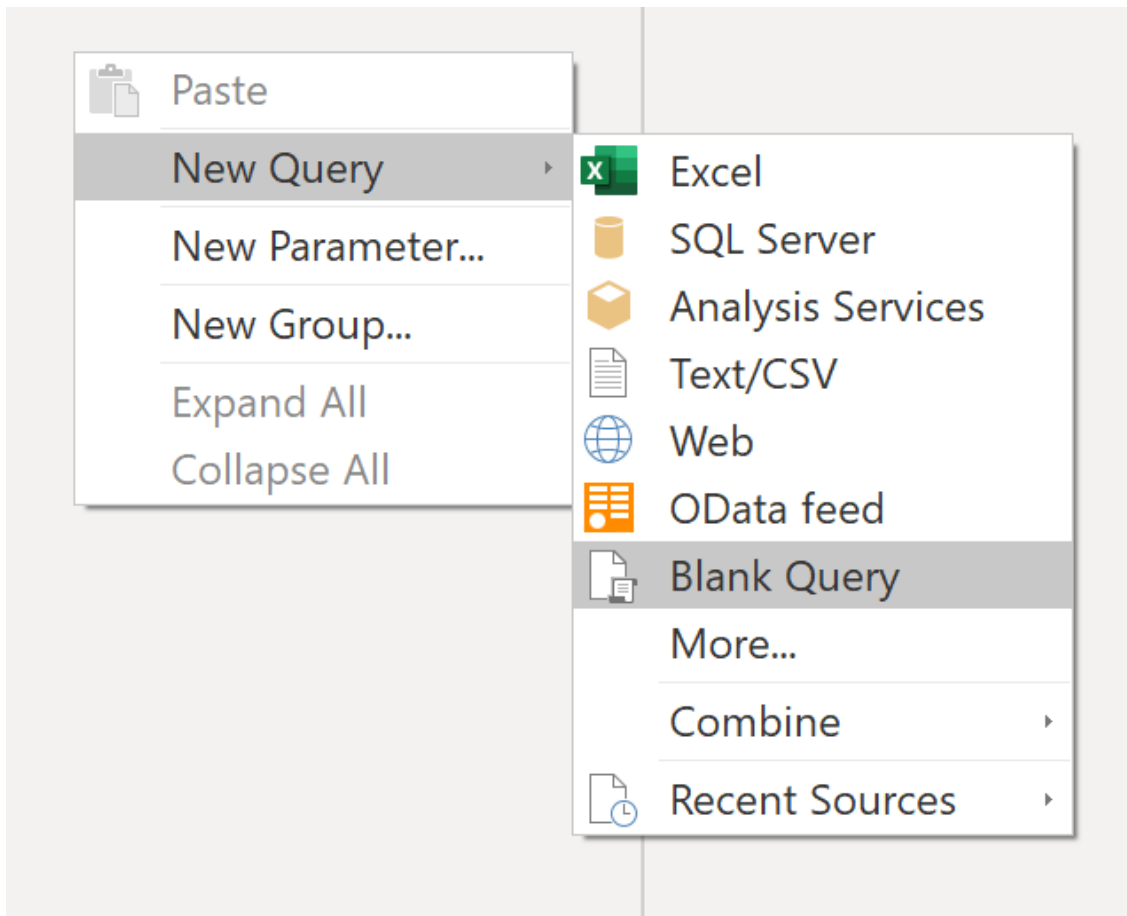
Date	Year	MonthNum	WeekNum	DayoftheWeek
Tuesday, May 31, 2011	2011	5	23	Tuesday
Wednesday, June 1, 2011	2011	6	23	Sunday
Thursday, June 2, 2011	2011	6	23	Monday
Friday, June 3, 2011	2011	6	23	Tuesday

Du har nu oprettet en fælles datotabel ved hjælp af DAX. I denne proces føjes den nye tabel kun til den semantiske model. du vil stadig have brug for at etablere relationer mellem datotabellen og tabellerne Salg og Ordre, og derefter markere tabellen som den officielle datotabel i din semantiske model. Før du udfører disse opgaver, skal du dog overveje en anden måde at opbygge en fælles datotabel på: ved hjælp af denne metode Power Query.

Power Query

Du kan bruge M-sprog, som er det udviklingssprog, der bruges til at opstille forespørgsler i Power Query, til at definere en fælles datotabel.

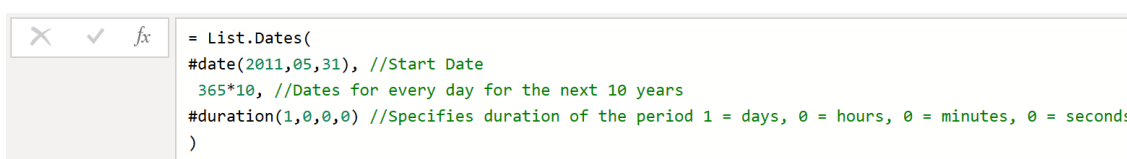
Vælg **Transformer data** i Power BI Desktop, som du bliver direkte til Power Query. Højreklik på det tomme område i ruden Forespørgsler i venstre rude for at åbne følgende rullemenu, **hvor du skal vælge Ny > tom forespørgsel**.



I den nye forespørgselsvisning, der fås som resultat, skal du angive følgende M-formel for at oprette en kalendertabel:

M: først lave liste med datoer, dernæst konverter listen til tabel

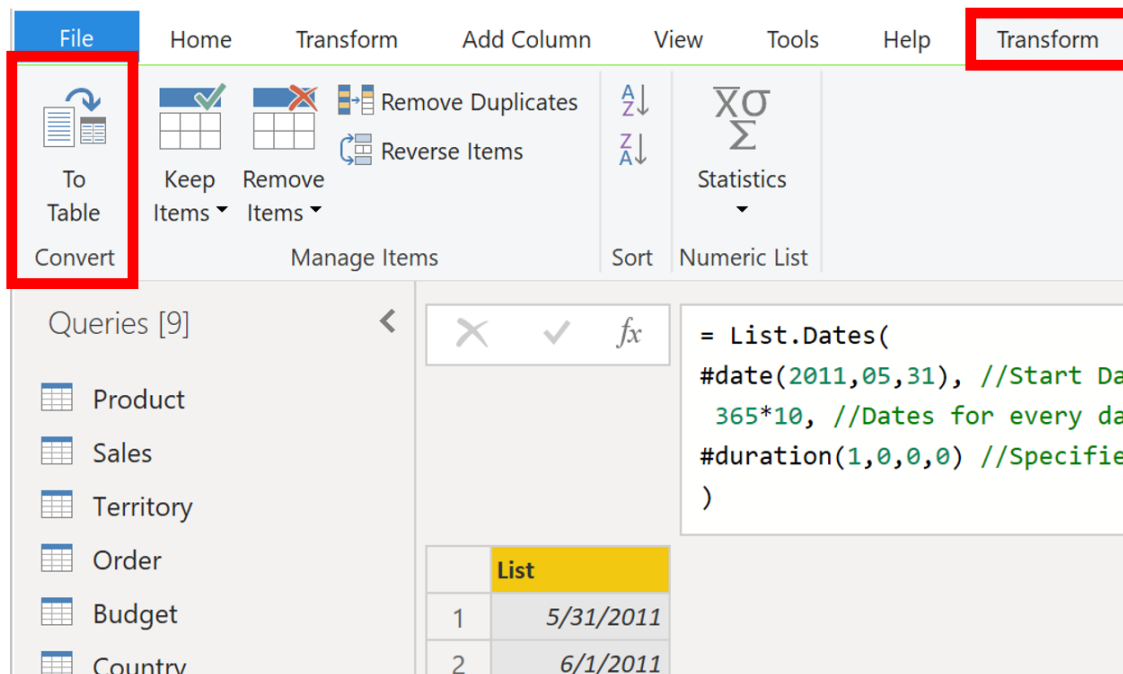
= List.Dates(#date(2011,05,31), 365*10, #duration(1,0,0,0))



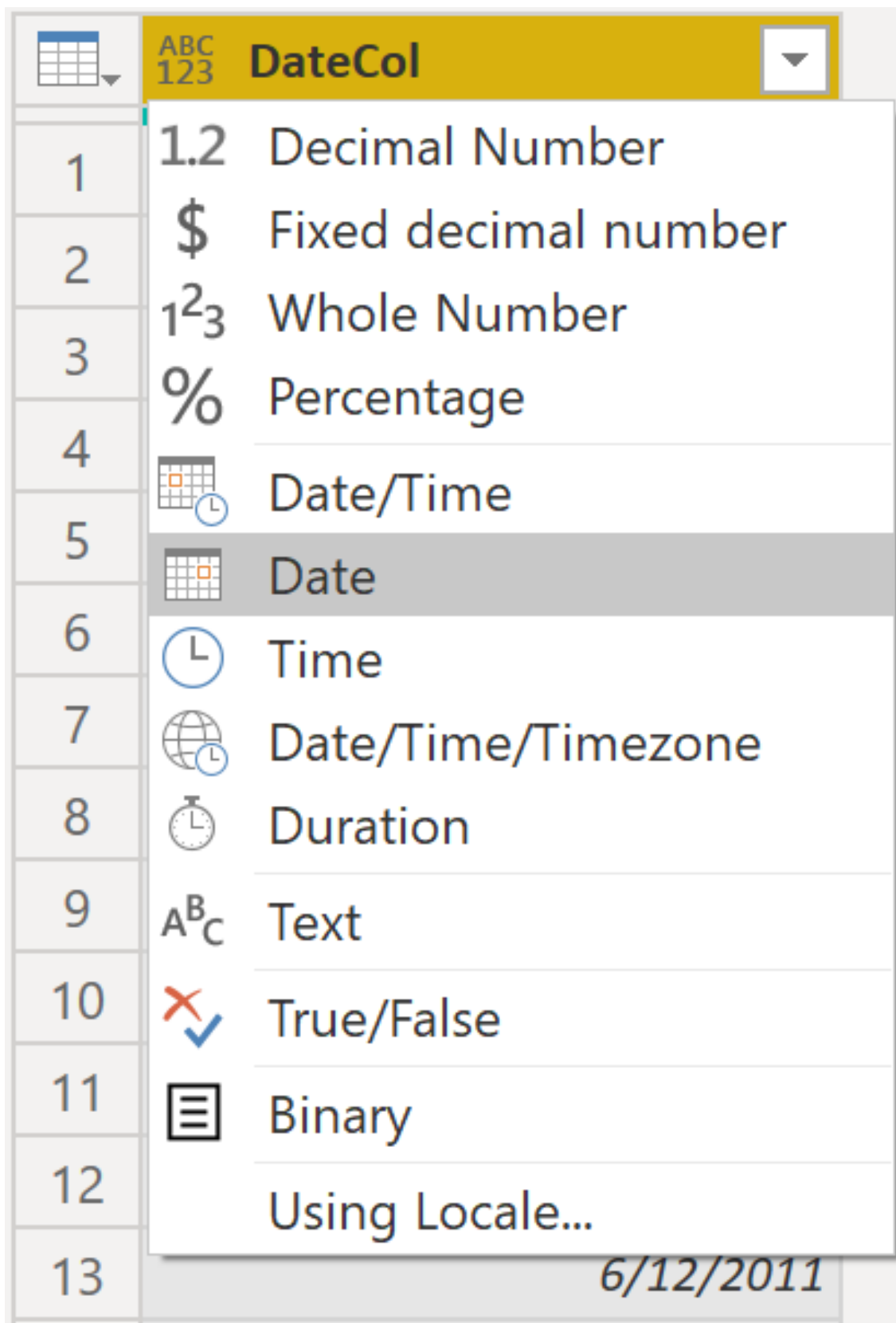
Når det kommer til salgsdataene, skal startdatoen afspejle den tidligste dato, du har i dataene: 31. maj 2011. Derudover ønsker du at få vist datoer for de næste 10 år, herunder datoer i fremtiden. Denne indfaldsvinkel sikrer, at du – efterhånden som nye salgsdata flyder ind – ikke behøver oprette denne tabel igen. Du kan også ændre varigheden. I dette tilfælde skal du bruge et datapunkt for hver dag, men du kan også øges med timer, minutter og sekunder. I følgende figur vises resultatet.

	List
1	5/31/2011
2	6/1/2011
3	6/2/2011
4	6/3/2011
5	6/4/2011
6	6/5/2011
7	6/6/2011
8	6/7/2011
9	6/8/2011

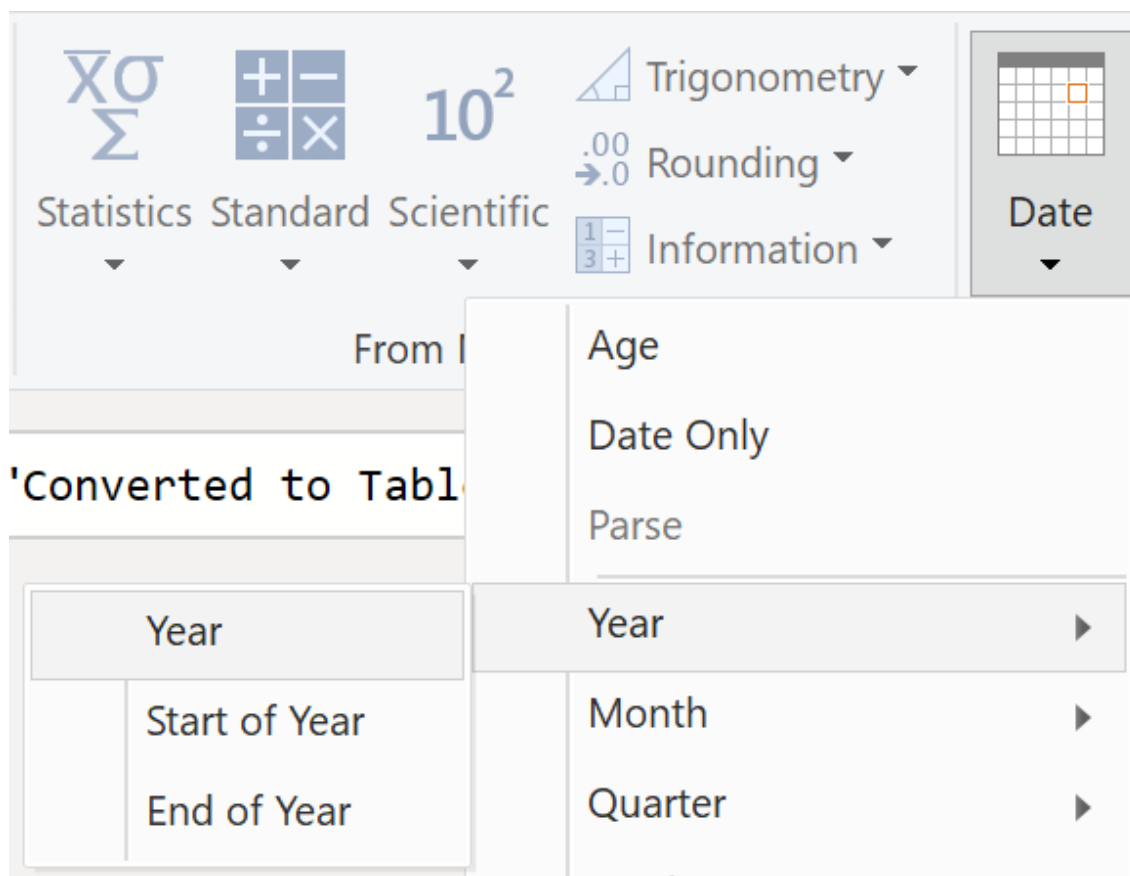
Når du har realiseret, at processen er lykkedes, bemærker du, at du har en liste over datoer i stedet for en tabel over datoer. Du kan rette denne fejl ved at gå til fanen **Transformering på båndet og vælge Konverter til tabel >** . Som navnet foreslår, konverterer denne funktion listen til en tabel. Du kan også omdøbe kolonnen til **DateCol**.



Derefter skal du føje kolonner til den nye tabel for at få vist datoer i form af år, måned, uge og dag, så du kan opbygge et hierarki i det visuelle udtryk. Din første opgave består i at ændre kolonnetype **ved at vælge ikonet ud for navnet på kolonnen og vælge datotypen i den resulterende rullemenu**.



Når du er færdig med at vælge datatype , kan du tilføje kolonner for år, måneder, uger og dage. Gå til **Tilføj** kolonne, vælg rullemenuen under **Dato**, og vælg derefter **År** som vist i følgende figur.



Bemærk, Power BI at der er tilføjet en kolonne over alle år, som er trukket fra **DateCol**.

DateCol	Year
5/31/2011	2011
6/1/2011	2011
6/2/2011	2011

Udfør den samme proces for måneder, uger og dage. Når du er færdig med denne proces, indeholder tabellen de kolonner, der er vist i følgende figur.

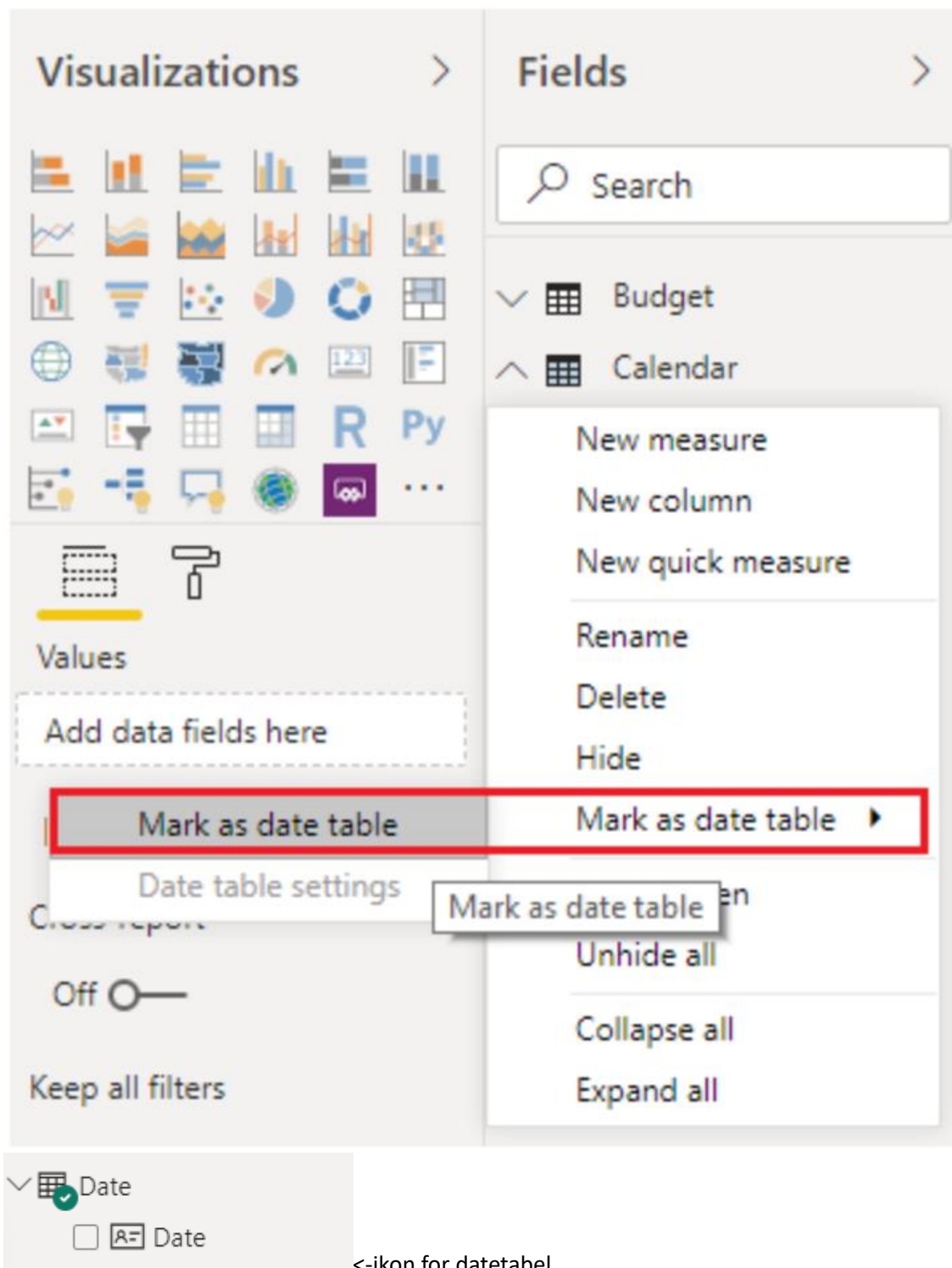
	DateCol	Year	Month	Week of Year	Day Name
1	5/31/2011	2011	5	23	Tuesday
2	6/1/2011	2011	6	23	Wednesday
3	6/2/2011	2011	6	23	Thursday

Nu har du brugt til Power Query at opbygge en fælles datotabel.

De foregående trin viser, hvordan tabellen kommer ind i den semantiske model. Nu skal du markere Power BI tabellen som den officielle datotabel, så du kan genkende den for alle fremtidige værdier og sikre, at formateringen er korrekt.

Markeres som den officielle datotabel

Den første opgave ved afmærkning af tabellen som den officielle datotabel er at finde den nye tabel i **uden** Felter. Højreklik på navnet på tabellen, og vælg derefter Markér **som datotabel** som vist i følgende figur.



Når du markerer tabellen som en datotabel, Power BI udføres valideringer for at sikre, at data indeholder null-værdier på nul, er entydige og indeholder fortløbende datoværdier for en periode. Du kan også vælge

bestemte kolonner i tabellen, der skal markeres som datoen, hvilket kan være nyttigt, når du har mange kolonner i tabellen. Højreklik på tabellen, vælg Marker **som datotabel**, og vælg derefter indstillinger for **datotabel**. Følgende vindue vises, hvor du kan vælge, hvilken kolonne der skal markeres som **Dato**.

Mark as date table

Select a column to be used for the date. The column must be of the data type 'date' and must contain only unique values. [Learn more](#)

Date column

 Validated successfully

 When you mark this as a date table, the built-in date tables that were associated with this table are removed. Visuals or DAX expressions referring to them may break.
[Learn how to fix visuals and DAX expressions](#)

OK

Cancel

Hvis du **vælger Markér som datotabel**, fjernes automatisk genererede hierarkier fra feltet Dato i den tabel, du har markeret som en datotabel. I forbindelse med andre datofelter findes det automatiske hierarki stadig, indtil du får vist en relation mellem dette felt og datotabellen, **eller indtil du deaktiverer funktionen Auto-dato/- klokkeslæt**. Du **kan manuelt føje et hierarki til den fælles datotabel ved at højreklikke på kolonnerne for år, måned, uge eller dag i ruden Felter og derefter vælge Nyt hierarki**. Denne proces beskrives yderligere senere i dette modul.

Opbyg din grafik

Hvis du vil opbygge grafik mellem tabellerne Salg og Ordre, skal du oprette en relation mellem denne nye fælles datotabel og tabellerne Salgsordrer og Ordre. Som følge heraf kan du opbygge visuelle elementer ved hjælp af den nye datotabel. Du kan udføre denne opgave > ved at gå til fanen Administrer relationer for model, **hvor du kan oprette relationer mellem den fælles datotabel og tabellerne Ordre og Salg ved hjælp af kolonnen OrderDate**. Følgende skærmbillede viser et eksempel på en sådan relation.

Create relationship med nye date-tabel

Create relationship

Select tables and columns that are related.

Sales

SalesOrderID	OrderDate	Sort_of_Sales	Freight	Freight_1	ProductID	OrderQty	SpecialOffe
52242	Sunday, July 7, 2013	Internet	0.1248	0.1248	870	1	
52592	Sunday, July 14, 2013	Internet	0.1248	0.1248	870	1	
52694	Tuesday, July 16, 2013	Internet	0.1248	0.1248	870	1	

DatesTable

DateCol	Year	Month	Week of Year	Day Name
Tuesday, May 31, 2011	2011	5	23	Tuesday
Tuesday, June 7, 2011	2011	6	24	Tuesday
Tuesday, June 14, 2011	2011	6	25	Tuesday

Cardinality

Many to one (*:1)

Cross filter direction

Both

☒ Make this relationship active

☐ Assume referential integrity

☐ Apply security filter in both directions

OK

Cancel

Når du har opbygget relationerne, kan du visualisere det samlede salg og ordreantal med den almindelige datotabel, du har udviklet vha. DAX eller metoden Power Query .

Når du skal fastlægge det samlede salg, skal du tilføje alle salg, fordi kolonnen Beløb i salgstabellen kun kigger på omsætningen for hvert salg og ikke på den samlede salgsindtomsætning. Du kan fuldføre denne opgave ved at bruge følgende beregning af målepunkt, som forklares i senere diskussioner. Den beregning, du skal bruge, når du bygger dette målepunkt, er følgende:

DAX -Kopier

#Total Sales = SUM(Sales[Amount'])

Når du er færdig, kan du oprette en tabel ved at vende tilbage til fanen **Visualiseringer** og vælge tabel **visualisering** . Du vil have vist totalordrer og salg efter år og måned, så du vil kun medtage kolonnerne År og Måned fra datotabellen, **kolonnen OrderQty** og **#TotalSales** måling. Når du får mere at vide om hierarkier, kan du også opbygge et hierarki, der giver Detailudledning fra år til måneder. I dette eksempel kan du få dem vist side om side. Du har nu oprettet en visual med en fælles datotabel.

Year	Month	OrderQty	# TotalSales	^
2011	5	825	853,422	
2011	6	141	460,085	
2011	7	2209	3,130,880	
2011	8	2904	3,917,345	
2011	9	157	503,668	
2011	10	5382	7,426,033	
2011	11	230	740,105	
2011	12	1040	1,815,966	
2012	1	3967	6,319,337	
2012	2	1442	2,106,429	
2012	3	3184	4,575,015	✓
Total		274914	170,964,700	

Næste undermodul: Arbejd med dimensioner

[Tidligere Næste](#)

Har du brug for hjælp? Se i vores [fejlfindingsvejledning](#), eller angiv specifik feedback ved at [rapportere et problem](#).

Arbejd med dimensios-tabeller, valg af hierarkier

Fuldført100 XP

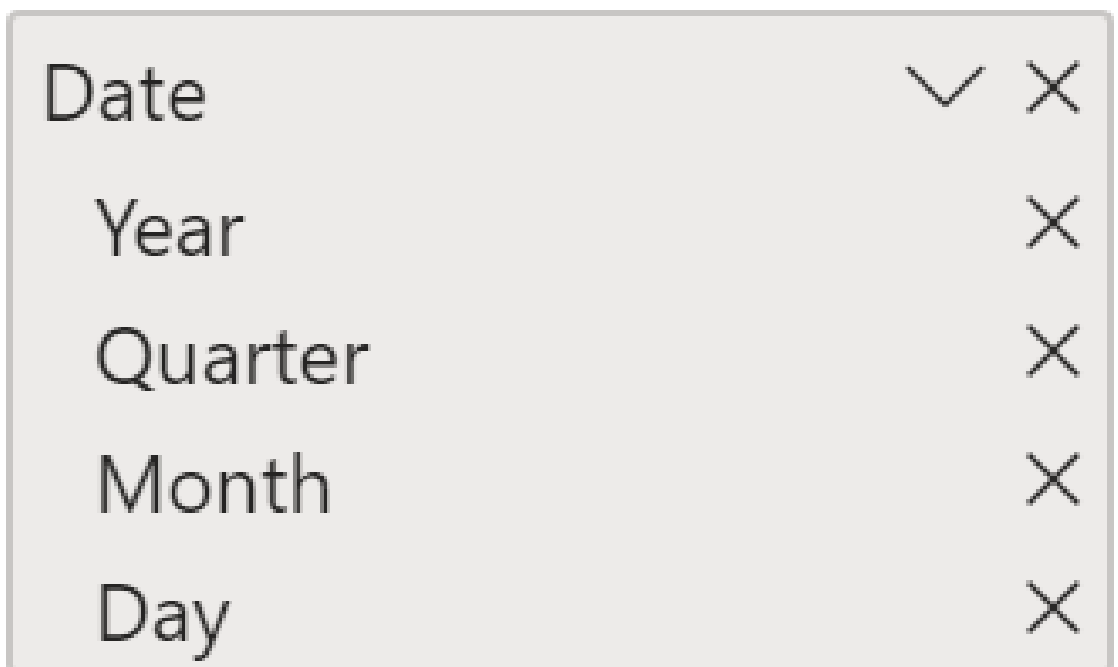
- 6 minutter

Når du opretter et stjerneschema, har du dimensions- og faktatabeller. Faktatabeller indeholder oplysninger om begivenheder som salgsordrer, forsendelsesdatoer, forhandlere og leverandører. Dimensionstabeller indeholder detaljer om forretningsenheder, f. eks. produkter og tidspunkter, og de er forbundet til faktatabeller via en relation.

Du kan bruge hierarkier som én kilde til at hjælpe dig med at finde detaljer i dimensionstabeller. Disse hierarkier udgør naturlige segmenter i dine data. Du kan f. eks. have et hierarki med datoer, hvor dine datoer kan segmenteres i år, måneder, uger og dage. Hierarkier er nyttige, fordi de giver dig mulighed for at foretage detailudledning i de specifikke data i stedet for kun at få vist dataene på et højt niveau.

Hierarkier

Når du opretter visuals, angiver Power BI automatisk værdier for datotypen som et hierarki (hvis tabellen ikke er markeret som en datotabel).

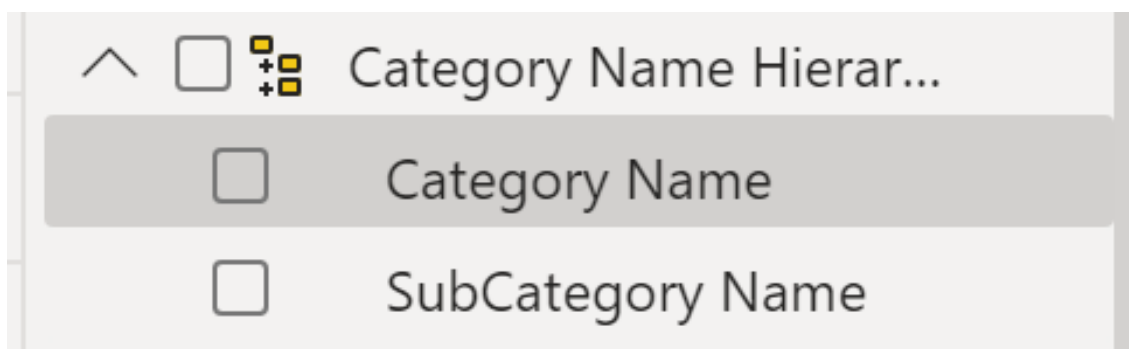


I den foregående **Date**-kolonne vises datoen mere og mere detaljeret som år, kvartaler, måneder og dage. Du kan også oprette hierarkier manuelt.

Du kan f. eks. forestille dig en situation, hvor du vil oprette et stablet liggende søjlediagram med **det samlede salg efter kategori og underkategori**. Du kan udføre denne opgave ved at oprette et hierarki i tabellen **Product** for kategorier og underkategorier. Hvis du vil oprette et hierarki, skal du gå til ruden **Felter** i Power BI og derefter højreklikke på den kolonne, du vil oprette hierarkiet for. Vælg **Nyt hierarki** som vist i følgende figur.



Træk derefter kolonnen Subcategory til det nye hierarki, som du har oprettet, og slip den. Denne kolonne vil blive tilføjet som et underordnet niveau i hierarkiet.



Nu kan du oprette visual'et ved at vælge et stablet liggende søjlediagram i ruden **Visualiseringer**. Tilføj dit **Hierarki for kategorinavn** i feltet **Akse** og **Salg i alt** i feltet **Værdier**.

Axis

Category Name Hierarcl ✓ ✕

Category Name ✕

SubCategory Name ✕

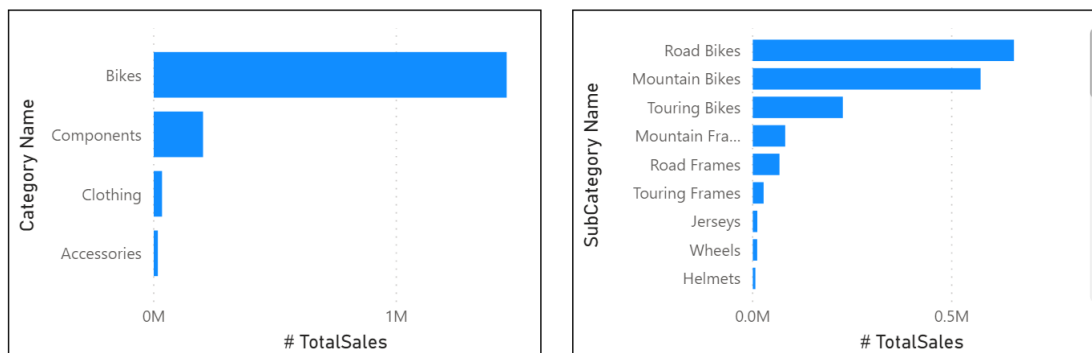
Legend

Add data fields here

Values

TotalSales ✓ ✕

Du kan foretage detailudledning i visual'et for at få vist både **Category** og **Subcategory**, afhængigt af hvad du vil se. Hierarkier giver dig mulighed for at få vist stigende dataniveauer i en enkelt visning.



Nu, hvor du har fået større indblik i hierarkier, kan du tage et skridt yderligere og undersøge overordnet-underordnet-hierarkier og deres rolle i flere relationer mellem faktatabeller og dimensionstabeller.

Overordnet-underordnet-hierarki, lave rolle-hierarki fra Employee tabel

I det følgende eksempel viser tabellen Employee i databasen vigtige oplysninger om medarbejderne, deres ledere og deres id'er. Når du kigger på denne tabel, kan du se, at **Roy F** vises flere gange i kolonnen **Manager**. Som du kan se på billedet, kan flere medarbejdere have samme leder, hvilket angiver et hierarki mellem ledere og medarbejdere.

	1 ² ₃ Employee ID	A ^B _C Employee	1 ² ₃ Manager ID	A ^B _C Manager
1	1010	Roy F	null	
2	1011	Pam H	1010	Roy F
3	1012	Guy L	1010	Roy F
4	1013	Roger M	1011	Pam H
5	1014	Kaylie S	1011	Pam H
6	1015	Mike O	1012	Guy L
7	1016	Rudy Q	1012	Guy L

Kolonnen **Manager** bestemmer hierarkiet og er derfor det overordnede niveau, mens de underordnede niveauer er medarbejderne. I dette eksempel vil du have vist alle niveauer i dette hierarki. I Power BI vises alle niveauer i hierarkiet ikke som standard, så det er dit ansvar at sikre, at du kan se alle niveauer i dette hierarki eller "fladgøre", så du kan se mere detaljerede data.

Fladgør overordnet-underordnet-hierarki

Processen med at få vist flere underordnede niveauer, der er baseret på en overordnet på øverste niveau, kaldes at *fladgøre*, *flatten*, *hierarkiet*. I denne proces opretter du flere kolonner i en tabel for at få vist den hierarkiske sti fra det overordnede til det underordnede niveau i samme post. Du skal bruge PATH(), som er en simpel DAX-funktion, der returnerer en tekstversion af ledelsesstien for hver medarbejder, og funktionen PATHITEM() til at adskille denne sti i de enkelte niveauer i ledelseshierarkiet.

Vigtigt

DAX er endnu ikke beskrevet, men beskrives i et andet modul. Denne funktion er medtaget i dette afsnit, fordi den giver en beskrivelse af hierarkier. Hvis brugen af DAX i denne sammenhæng er forvirrende, kan du se DAX-modulet og derefter vende tilbage til dette afsnit.

Gå til fanen **Modellering**, mens du får vist tabellen, og vælg **Ny kolonne**. Angiv følgende funktion i den formellinje, der vises, for at oprette tekststien mellem medarbejderen og lederen. Denne handling opretter en beregnet kolonne i DAX.

DAX Kopiér

Path = PATH(Employee[Employee ID], Employee[Manager ID])

1

Path = PATH(Employee[Employee ID], Employee[Manager ID])

▼

Manager ID

10

PATH(ID_ColumnName, Parent_ColumnName)

Returns a string which contains a delimited list of IDs, starting with the top/root of a hierarchy and ending with the specified ID.

Den fuldførte sti mellem medarbejderen og lederen vises i den nye kolonne, som det ses på følgende skærmbillede.

Employee ID	Manager ID	Employee	Manager	Path
1010		Roy F		1010
1011	1010	Pam H	Roy F	1010 1011
1012	1010	Guy L	Roy F	1010 1012
1013	1011	Roger M	Pam H	1010 1011 1013
1014	1011	Kaylie S	Pam H	1010 1011 1014
1015	1012	Mike O	Guy L	1010 1012 1015
1016	1012	Rudy Q	Guy L	1010 1012 1016

Hvis du kigger på Roger M, er stien til id'er **1010 | 1011 | 1013**, hvilket betyder, at ét niveau over Roger M (ID 1013) findes hans chef Pam H (ID 1011), og ét niveau over Pam H findes hendes leder Roy F (ID 1010). I denne række er Roger M nederst i hierarkiet på det underordnede niveau, og Roy F er øverst i hierarkiet på det overordnede niveau. Denne sti oprettes for hver medarbejder. Hvis du vil fladgøre hierarkiet, kan du adskille de enkelte niveauer ved hjælp af funktionen PATHITEM.

Hvis du vil have vist alle tre niveauer i hierarkiet separat, kan du oprette fire kolonner på samme måde, som du tidligere gjorde det, ved at angive følgende ligninger. Du skal bruge funktionen PATHITEM til at hente den værdi, der er placeret på det tilsvarende niveau i hierarkiet.

- Niveau 1 = PATHITEM(Employee[Path],1)
- Niveau 2 = PATHITEM(Employee[Path],2)
- Niveau 3 = PATHITEM(Employee[Path],3)

1

Level 1 = PATHITEM(Employee[Path], 1)

▼

Manager ID

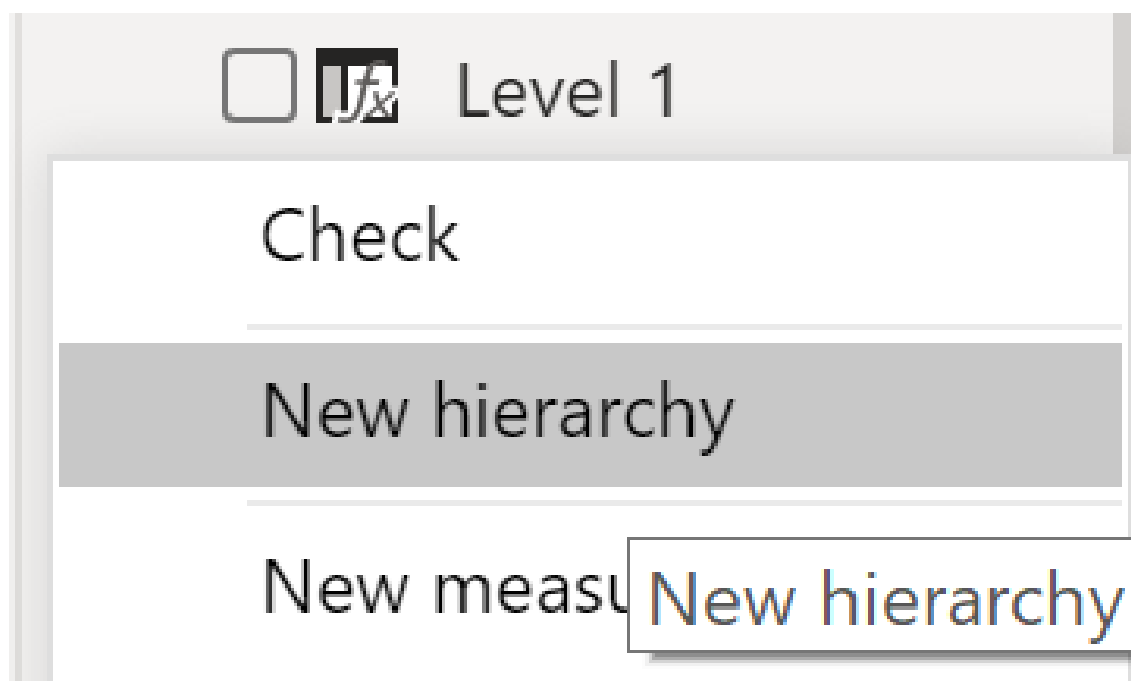
PATHITEM(Path, Position, [Type])

Returns the nth item in the delimited list produced by the Path function.

Når du er færdig, kan du se, at du nu har alle hierarkiets niveauer i din tabel. Roy F, det er vel Quincy Howard!?, er øverst i hierarkiet, og når du gennemgår **niveau 2-3**, kan du se, at lederne og medarbejderne er tilknyttet hinanden.

Employee ID	Name	Manager	Manager ID	Path	Level 1	Level 2	Level 3	Level 4
1000	Quincy Howard			1000	1000			
1001	Mallory Yang	Quincy Howard	1000	1000 1001	1000	1001		
1002	Donovan Maynard	Quincy Howard	1000	1000 1002	1000	1002		
1003	Giselle McClain	Mallory Yang	1001	1000 1001 1003	1000	1001	1003	
1004	Melvin Marsh	Mallory Yang	1001	1000 1001 1004	1000	1001	1004	
1005	Ria Snow	Giselle McClain	1003	1000 1001 1003 1005	1000	1001	1003	1005
1006	Callie Savage	Giselle McClain	1003	1000 1001 1003 1006	1000	1001	1003	1006

Nu kan du oprette et hierarki i ruden **Felter**, som du tidligere har gjort det. Højreklik på **Niveau 1**, da dette er det første hierarkiniveau, og vælg derefter **Nyt hierarki**. Træk og slip derefter **niveau 2** og **niveau 3** til dette hierarki.

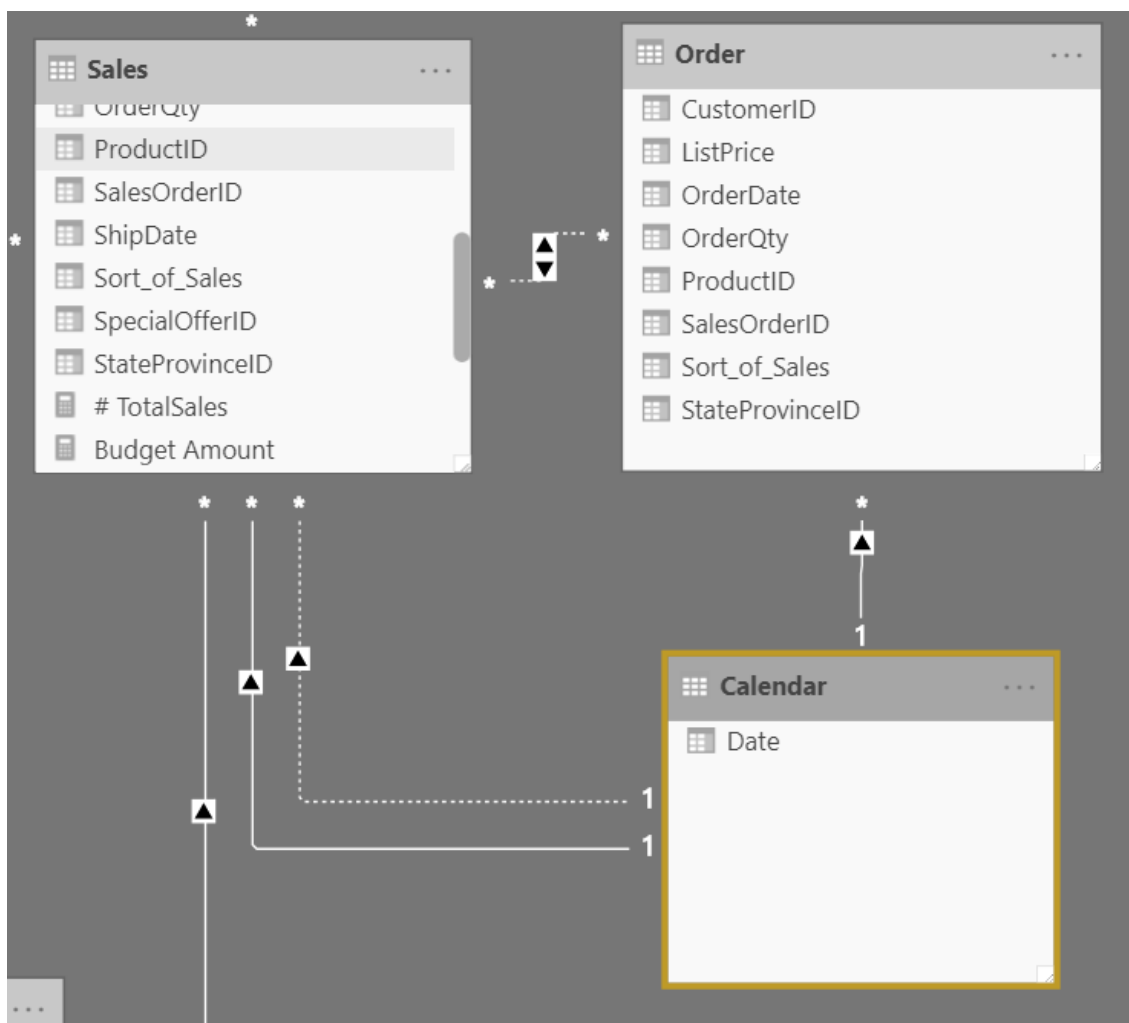


Du har nu fladgjort et hierarki, så du kan få vist individuelle niveauer.

Tidligere har du set dimensioner, der kun har én relation med en faktatabel. Der opstår dog situationer, hvor dimensionstabellen har flere relationer til en faktatabel.

Dimensioner med forskellige roller

Dimensioner med forskellige roller har flere gyldige relationer med faktatabeller, hvilket betyder, at den samme dimension kan bruges til at filtrere flere kolonner eller tabeller med data. Derfor kan du filtrere data forskelligt, afhængigt af hvilke oplysninger du har brug for at hente. Dette emne er kompliceret, så dette afsnit omfatter kun en introduktion. Når du skal arbejde dimensioner med forskellige roller, kræver det komplekse DAX-funktioner, der beskrives i senere afsnit.



I det foregående visual vises tabellerne Calendar, Sales og Orders. Calendar er dimensionstabellen, mens Sales og Orders er faktatabeller. Dimensionstabellen har to relationer: én med Sales og én med Orders. Dette eksempel viser en dimension med forskellige roller, fordi tabellen Calendar kan bruges til at gruppere data i både Sales og Orders. Hvis du vil oprette en visualisering, hvor tabellen Calendar refererer til tabellerne Orders og Sales, fungerer tabellen Kalender som en dimension med forskellige roller.

Definer datagranularitet

Fuldført 100 XP

- 4 minutter

Datagranularitet er det detaljeniveau, der er repræsenteret i dine data, hvilket betyder, at jo større granularitet, jo mere detaljerede er dine data.

Datagranularitet er et vigtigt emne for alle dataanalytikere, uanset hvilke Power BI-værktøjer du bruger. Definition af den korrekte datagranularitet kan have en stor indvirkning på Power BI-rapporters og -visuals' effektivitet og brugbarhed.

Definition af datagranularitet

Forestil dig et scenarie, hvor din virksomhed administrerer 1.000 køletrailere. Med få minutters mellemrum bruger trailerne et Microsoft Azure IoT-program til at registrere deres aktuelle temperatur. Denne temperatur er vigtig for din organisation, fordi hele lasten ville blive ødelagt, hvis kølefunktionen ikke fungerede korrekt, og koste virksomheden tusindvis af dollars. Med så mange biler og sensorer genereres der omfattende data hver dag. Dine rapportbrugere ønsker ikke at gennemse et utal af poster for at finde dem, de er særligt interesseret i.

Hvordan kan du ændre granulariteten af dataene for at gøre den semantiske model mere anvendelig?

I dette scenarie vil du muligvis importere dataene ved at bruge et dagligt gennemsnit for hver truck. Denne fremgangsmåde reducerer posterne i databasen til én post for hver trailer for hver dag. Hvis du beslutter, at fremgangsmåden er acceptabel til sporing af omkostninger og fejl, kan du bruge den pågældende datagranularitet. Du kan også vælge den senest registrerede temperatur, eller du kan nøjes med at importere poster, der ligger over eller under et normalt temperaturinterval. Enhver af disse metoder reducerer de samlede poster, du importerer, samtidig med, at du stadig får vist data, der er omfattende og værdifulde.

Du kan beslutte at bruge datagranularitet, der er defineret ugentligt, månedligt eller kvartalsvist i forskellige scenarier. Det gælder generelt, at jo færre poster, du arbejder med, jo hurtigere fungerer dine rapporter og visuals. Denne fremgangsmåde kan oversættes til en hurtigere opdateringshastighed for hele den semantiske model, hvilket kan betyde, at du kan opdatere oftere.

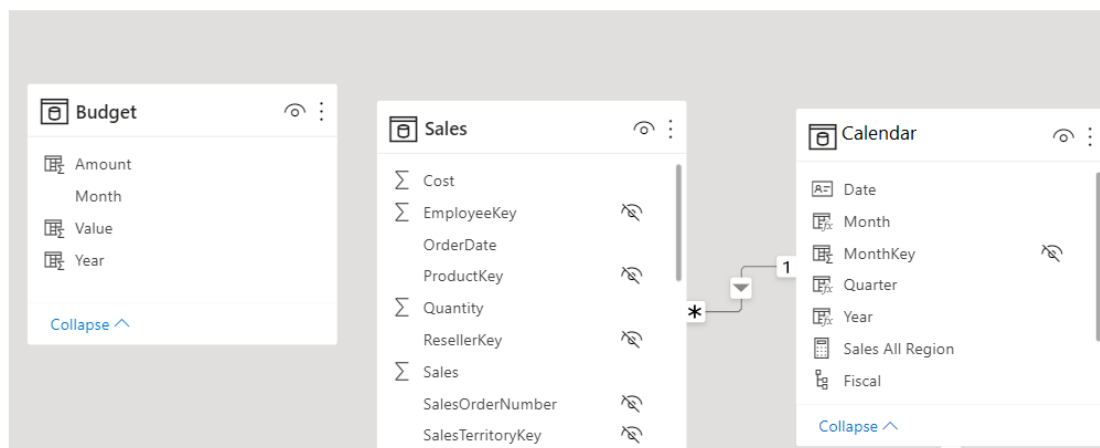
Fremgangsmåden har dog en ulempe. Hvis brugerne vil foretage detailudledning i hver enkelt transaktion, kan en opsummering af granulariteten forhindre dem i at gøre det, hvilket kan have en negativ indvirkning på brugeroplevelsen. Det er vigtigt at forhandle niveauet af datagranularitet med rapportbrugere, så de forstår konsekvenserne af disse valg.

Skift datagranularitet for at oprette en relation mellem to tabeller

Datagranularitet kan også have indflydelse, når du opretter relationer mellem tabeller i Power BI.

Forestil dig f.eks., at du opretter rapporter til salgsteamet hos Tailwind Traders. Du er blevet bedt om at oprette en matrix med det samlede salg og budget over tid ved hjælp af tabellerne Calendar, Sales og Budget. Du kan se, at det laveste niveau af tidsbaserede detaljer, der vises i tabellen Sales, er dage, f.eks. 5/1/2020, 6/7/2020 og 6/18/2020. Budgettabellen går f.eks. kun til det månedlige niveau, f.eks. er budgetdataene 5/2020 og 6/2020. Disse tabeller har forskellige granulariteter, der skal afstemmes, før du kan oprette en relation mellem tabeller.

I følgende figur vises din aktuelle semantiske model.



Som vist i den foregående figur, mangler der en relation mellem Budget og Calendar. Derfor skal du oprette denne relation, før du kan oprette visual'et. Bemærk, at hvis du transformerer kolonnerne **Year** og **Month** i tabellen Calendar til en ny kolonne og udfører den samme transformation i tabellen Budget, kan du matche formatet for kolonnen **Date** i tabellen Calendar. Derefter kan du oprette en relation mellem de to kolonner. Hvis du vil fuldføre denne opgave, skal du sammenkæde kolonnerne **Year** og **Month** og derefter ændre formatet.

Year 	Month 
2013	1
2013	2
2013	3
2013	4
2013	5
2013	6
2013	7
2013	8

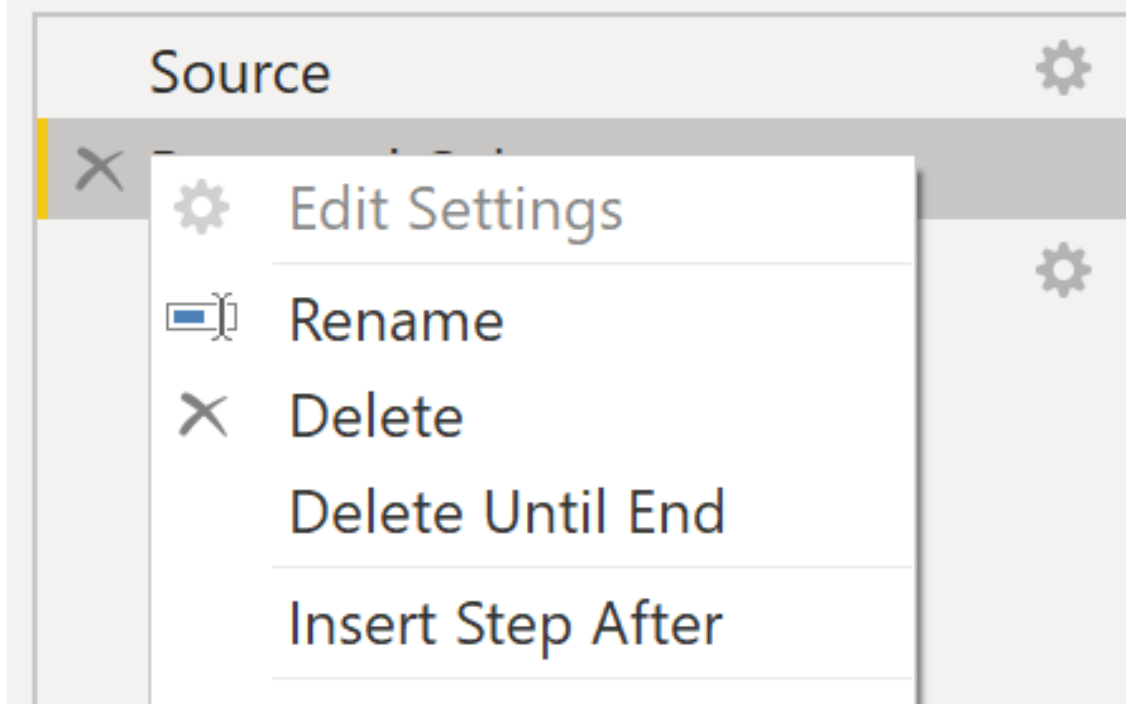
Budget

Date 
01/01/2011
01/02/2011
01/03/2011
01/04/2011
01/05/2011
01/06/2011
01/07/2011

Calendar

Vælg **Transformér data** på båndet. Højreklik på det sidste trin i **Anvendte trin** i ruden til højre, og vælg derefter **Indsæt trin efter**.

APPLIED STEPS



Vælg **Brugerdefineret kolonne** under **Tilføj kolonne** på båndet Hjem. Angiv følgende ligning, der sammenkæder kolonnerne **Year** og **Month**, og tilføj derefter en bindestreg mellem kolonnenavnene.

M Kopiér

```
Column = Table.AddColumn("#Renamed Columns", "Custom", each [Year] & "-" & [Month])
```

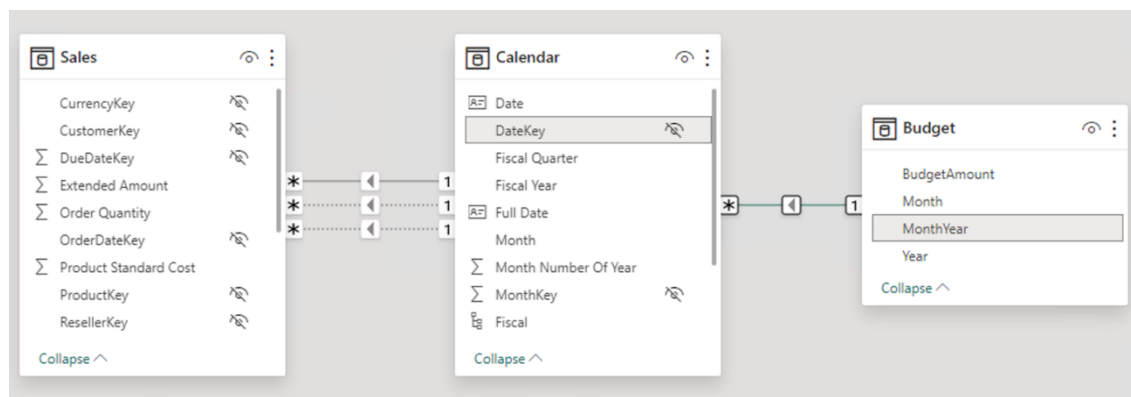
Skift datatypen til **Dato**, og omdøb derefter kolonnen. Tabellen Budget skal ligne følgende figur.

Year ▼	Month ▼	BudgetAmount ▼	MonthYear ▼
2011	1	9493	2011-1
2011	2	2490	2011-2
2011	3	16585	2011-3
2011	4	5575	2011-4
2011	5	11611	2011-5
2011	6	14017	2011-6
2011	7	11919	2011-7
2011	8	3832	2011-8
2011	9	7094	2011-9
2011	10	6216	2011-10
2011	11	7769	2011-11
2011	12	12552	2011-12
2012	1	12781	2012-1
2012	2	7889	2012-2
2012	3	10329	2012-3
2012	4	5350	2012-4
2012	5	19216	2012-5
2012	6	5849	2012-6
2012	7	19269	2012-7
2012	8	3080	2012-8
2012	9	2381	2012-9

Nu kan du oprette en relation mellem tabellerne Budget og Calendar.

Opret en relation mellem tabeller

Power BI registrerer automatisk relationer, men du kan også gå til **Administrer relationer > Ny** og oprette relationen i kolonnen **Dato**. Relationen skal ligne følgende figur.



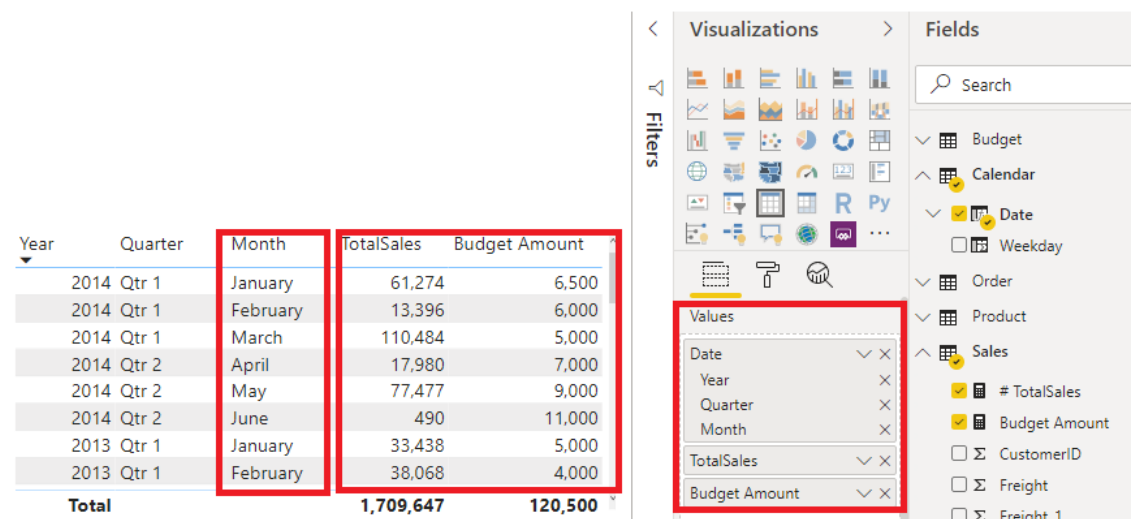
Ved at fuldføre denne opgave har du sikret, at granulariteten er den samme mellem dine forskellige tabeller. Nu skal du oprette DAX-målinger for at beregne **Total Sales** og **BudgetAmount**. Gå til ruden **Data** i Power BI Desktop, vælg **Ny måling**, og opret derefter to målinger med følgende ligninger:

DAX Kopiér

TotalSales = SUM(Sales[Total Sales])

BudgetAmount = SUM (Budget[BudgetAmount])

Vælg tabelvisualiseringen i ruden **Visualisering**, og angiv derefter disse målinger og **dato** i feltet **Værdier**. Du har nu nået målet om at oprette en matrix over det samlede salg og budgetter over tid.



Næste undermodul: Arbejde med relationer og alitet

[Tidligere Næste](#)

Har du brug for hjælp? Se i vores [fejlfindingsvejledning](#), eller angiv specifik feedback ved at [rapportere et problem](#).

Arbejde med relationer og alitet

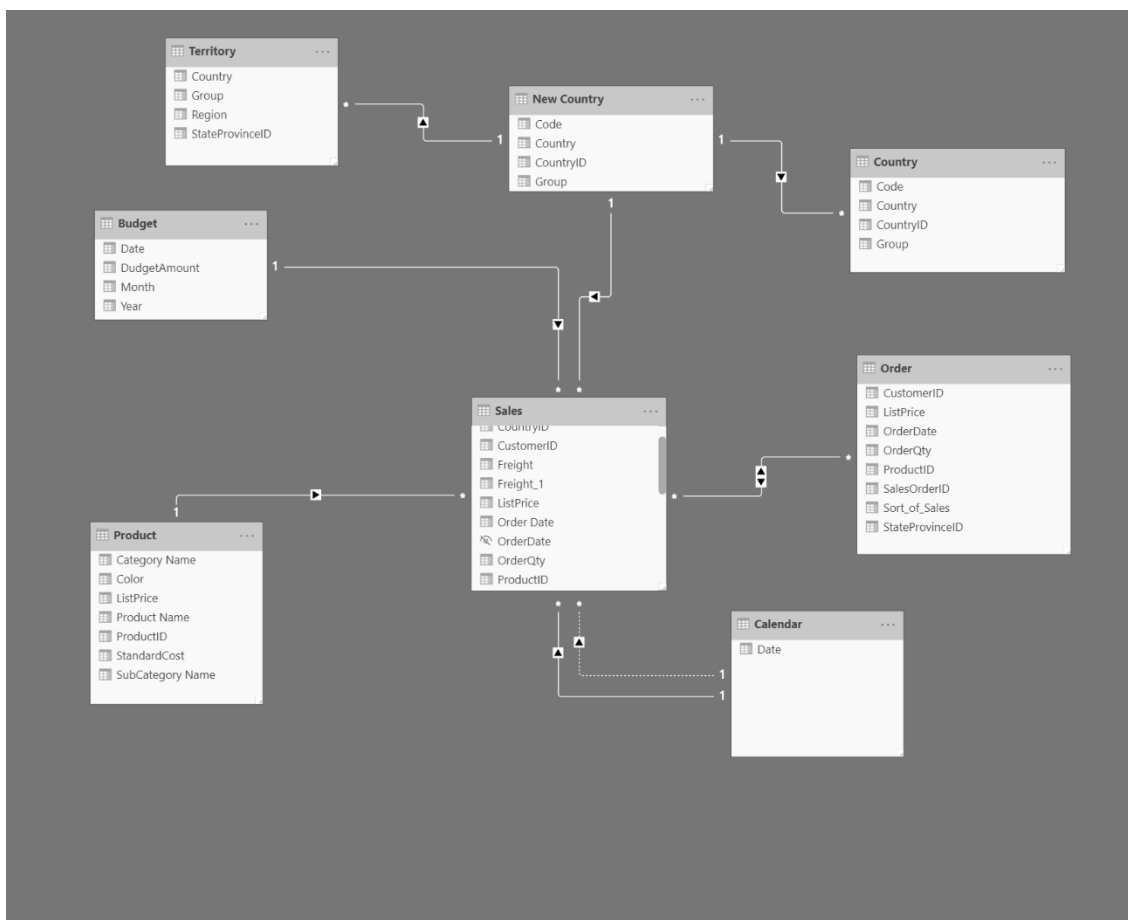
Fuldført 100 XP

- 5 minutter

I modsætning til andre databasestyringssystemer har Power BI begrebet *retningsforhold* til en relation.

Denne retningsbestemthed får en vigtig rolle ved filtrering af data mellem flere tabeller. Når du indlæser data, søger Power BI du automatisk efter relationer, der findes i dataene, ved at matche kolonnenavne. Du kan også bruge **administrer relationer til** at redigere disse indstillinger manuelt.

Du har f.eks. hentet mange tabeller fra salgsdatabasen, og følgende billede er et eksempel på din semantiske model. Power BI har automatisk registreret flere relationer, men du kan ikke finde ud af, hvad de betyder. Du vil sikre dig, at relationerne afspejler de relationer, der findes i dataene, nøjagtigt.



Relationer

De følgende er forskellige typer relationer, som du kan finde i Power BI.

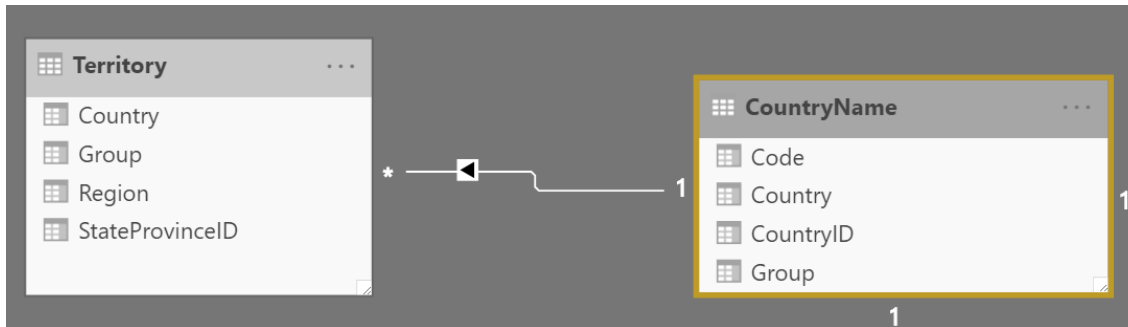
Mange til en- eller (*:1) en til mange-relation (1: *) er det almindelige, sunde

- Beskriver en relation, hvor du har mange forekomster af en værdi i en kolonne, der kun er relateret til en entydig tilsvarende forekomst i en anden kolonne.
- Beskriver retningsretningen mellem fakta- og dimensionstabeller.

- Er den mest almindelige type retningsbestemthed og er standard Power BI , når du automatisk opretter relationer.

Et eksempel på en én til mange-relation kan være mellem tabellerne Landenavn og Distrikt, hvor du kan have mange distrikter, der er tilknyttet ét entydigt land.

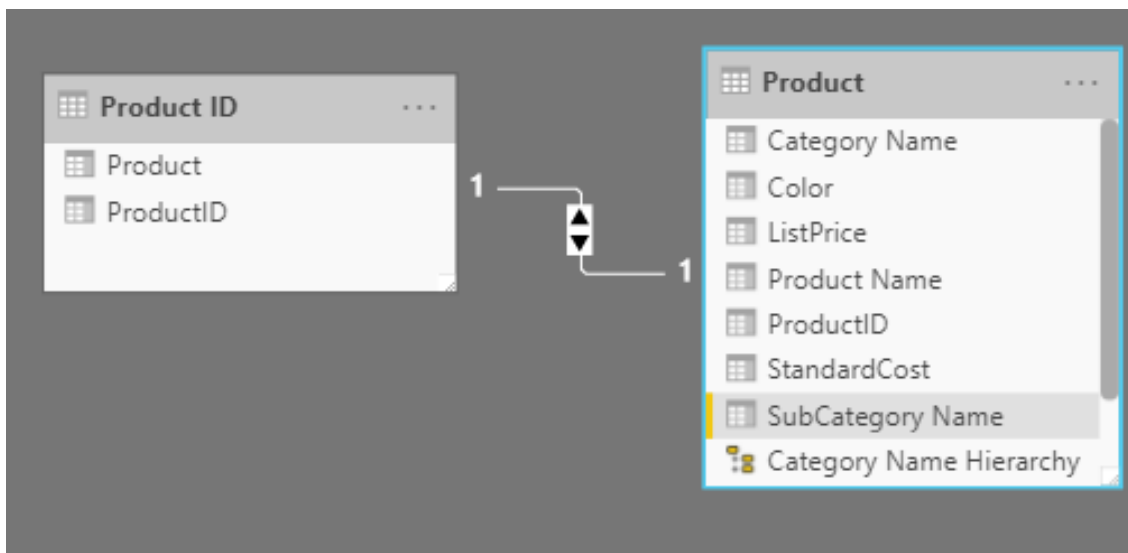
Filtrering følger pileretninger



En til en-relation (1:1), anbefales ikke: saml data i én tabel

- Beskriver en relation, hvor der kun er én forekomst af en værdi fælles mellem to tabeller.
- Kræver entydige værdier i begge tabeller.
- **Det anbefales ikke, at du gør dette**, da denne relation gemmer overflødige oplysninger og foreslår, at modellen ikke er designet korrekt. Det er bedre at kombinere tabellerne.

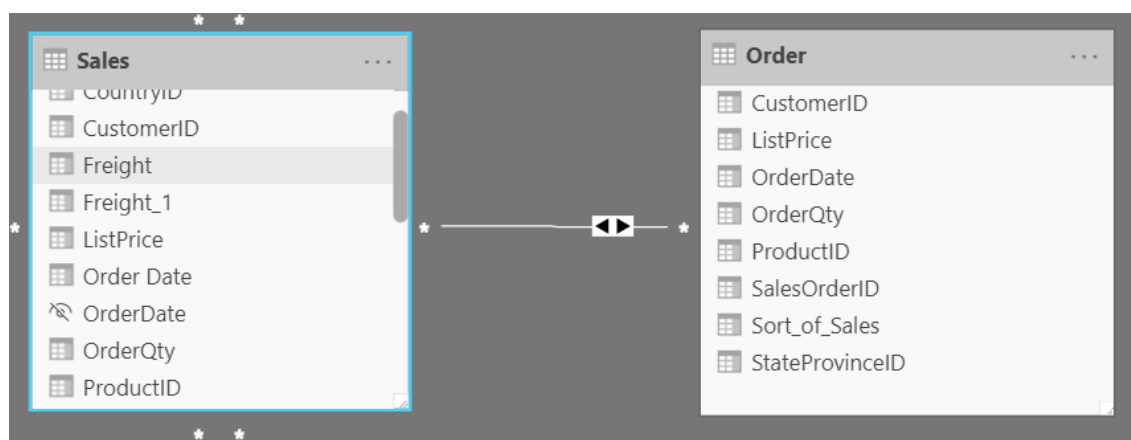
Et eksempel på en en til en-relation er, hvis du havde produkter og produkt-i to forskellige tabeller. **Det er overflødigt at oprette en en til en-relation, og disse to tabeller skal kombineres.**



Mange til mange-relation (.): anbefales ikke, kan nemt give inkonsistente data

- Beskriver en relation, hvor mange værdier er til fælles mellem to tabeller.
- Kræver ikke entydige værdier i nogen af tabellerne i en relation.
- **Anbefales ikke**; manglende entydige værdier introducerer guity og dine brugere ved muligvis ikke, hvilken kolonne med værdier der henviser til hvad.

I nedenstående figur vises f.eks. en mange til mange-relation mellem tabellerne **Salg** og **Ordre** i kolonnen **OrderDate** , da flere salg kan have tilknyttet flere ordrer. **Der introduceres Guity, fordi begge tabeller kan have samme ordredato.**



Krydsfilterretning

Data kan filtreres på én eller begge dele af en relation.

Med en **enkelt retning på tværs af filtre:**

- Der kan kun bruges én tabel i en relation til at filtrere dataene. Tabel 1 kan f.eks. filtreres efter tabel 2, men tabel 2 kan ikke filtreres efter tabel 1.

Tip

Følg pilens retning i forholdet mellem tabellerne for at vide, hvilken retning filteret vil flyde i. Du vil typisk have, at disse pile skal pege på faktatabellen.

- I forbindelse med en en til mange-relation eller mange til en-relation vil den krydsfilterretningen være fra den ene side, hvilket betyder, at filtrering sker i tabellen med mange værdier.

Med **både krydsfilterretninger** eller **tovejs-krydsfiltrering:**

- Den ene tabel i en relation kan bruges til at filtrere den anden. En dimensionstabel kan f.eks. filtreres gennem faktatabellen, og faktatabellerne kan filtreres gennem dimensionstabellen.
- **Du kan have en lavere ydeevne, når der bruges tovejs-på tværs af filtrering med mange til mange-relationer.**

En god ide at **være forsigtig med tovejs-krydsfiltrering:** Du skal ikke aktivere tovejs-krydsfiltreringsrelationer, medmindre du fuldt ud forstår de forskellige foranstaltninger, der gør det. Aktivering af den kan medføre uoverensmældt vareprøve, overprøve og uventede resultater og potentiel ydeevne.

Fortrolighed og krydsfilterretning

I forbindelse med 1 til 1-relationer er den eneste tilgængelige indstilling tovejs-krydsfiltrering. Data kan filtreres på hver side af dette forhold og resultere i én entydig værdi. Du kan f.eks. filtrere på ét produkt-id og returnere et enkelt produkt, og du kan filtrere på et produkt og returnere et enkelt produkt-id.

Ved mange til mange-relationer kan du vælge at filtrere i én retning eller i begge retninger ved at bruge **tovejs-krydsfiltrering**. Den kompleksitet, der er knyttet til tovejs-krydsfiltrering, overestimeres i en mange til mange-relation, fordi der vil være flere stier mellem forskellige tabeller. Hvis du opretter et målepunkt, en beregning eller et filter, kan der forekomme utilsigtede konsekvenser, når dataene filtreres, Power BI og afhængigt af hvilken relation programmet vælger, når filteret anvendes, kan det endelige resultat blive forskelligt. Denne situation gælder også for tovejs-relationer, og hvorfor du skal være forsigtig, når du bruger dem.

Derfor bliver mange til mange-relationer og/eller tovejs-relationer komplicerede. **Medmindre du er sikker på, hvordan dataene ser ud, når de aggregeres, kan disse typer åbne relationer med flere filtreringsretninger introducere flere stier gennem dataene.**

Oprette mange til mange-relationer

Overvej det scenarie, hvor du har til opgave at opbygge et billede/visual, der undersøger budgetter for kunder og konti. Du kan have flere debitorer på samme konto og flere konti med samme debitor, så du ved, at du skal oprette en mange til mange-relation.

CustID	CustName
1022	Roy M
1023	Bob K
1024	Ellen L
1025	Mitch W
1026	Regan Q
1027	Lulu S
1028	Aliya R

CustomerTable

CustomerID	AccountID	AccountName
1022	12	BHP
1023	12	BHP
1024	13	RogerInc
1024	14	MyShip
1026	15	Holdings Unl.
1025	16	Key Biz Insiders
1028	17	Ty Inc
1022	17	Ty Inc

AccountTable

Du kan oprette dette forhold ved at gå til **Administrer relationer>ny**. I det vindue, der bliver resultatet, skal du oprette en relation mellem **kolonnen Debitor-id** i CustomerTable og AccountTable. Relationen er angivet til mange til mange, og filtertypen er i begge retninger. Straks vil du blive advaret om, at du kun skal bruge denne type forhold, hvis det forventes, at ingen af kolonnerne vil have entydige værdier, fordi du kan få uventede værdier[nej, det må vel være ok at én kunde har én konto, men de fleste skal have flere koblinger. Da du vil filtrere i begge retninger, skal du vælge **tovejs-krydsfiltrering**.

Vælg **OK**. Du har nu oprettet en mange til mange-relation.

×

Create relationship

Select tables and columns that are related.

AccountTable

CustomerID	AccountID	AccountName
1022	12	BHP
1023	12	BHP
1024	13	RogerInc

CustomerTable

CustID	CustName
1022	Roy M
1023	Bob K
1024	Ellen L

Cardinality

Many to Many (*:*)

Cross filter direction

Both

☒ Make this relationship active

☐ Apply security filter in both directions

☐ Assume referential integrity

⚠

This relationship has cardinality Many-Many. This should only be used if it is expected that neither column (CustomerID and CustID) contains unique values, and that the significantly different behavior of Many-many relationships is understood. [Learn more](#)

OK

Cancel

Yderligere oplysninger finder du i [Mange til mange-relationer Power BI](#).

Næste undermodul: Løs modelleringsudfordringer

[Tidligere](#)

Løs modelleringsudfordringer

Fuldført100 XP

- 2 minutter

Modellering af data handler om at oprette og vedligeholde relationer, så du effektivt kan visualisere dataene i den formular, din virksomhed har brug for. Når du opretter disse relationer, er **en af de typiske faldgruber cirkulære relationer**.

Du udvikler f. eks. rapporter for salgsteamet og undersøger relationerne mellem tabeller. I en dårligt designet semantisk model har Tabel 1 en mange til en-relation med en kolonne i tabel 2, men Tabel 2 har en en til mange-relation med Tabel 3, der har sin egen relation til Tabel 1. Dette net af relationer er svært at administrere, og det bliver en uoverskuelig opgave at oprette visuals, da det ikke længere er tydeligt, hvilke relationer der findes. Derfor er det vigtigt, at du kan identificere cirkulære relationer, så dine data bliver brugbare.

Relationsafhængigheder

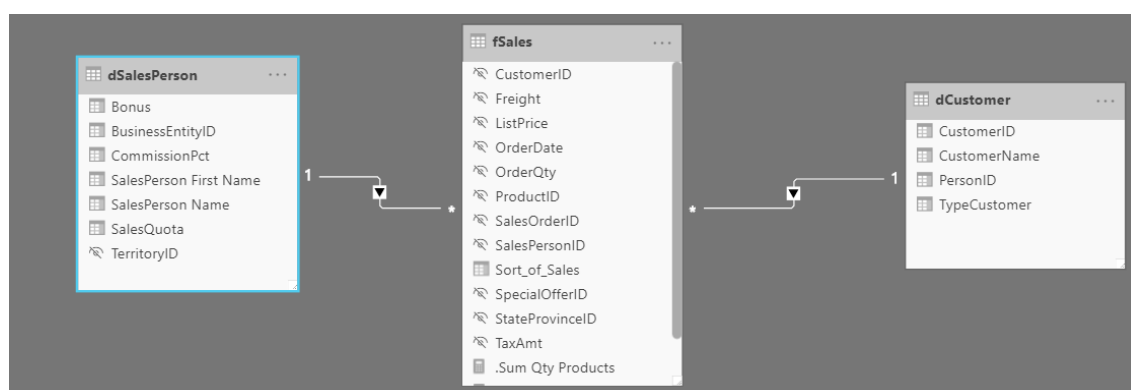
For at forstå cirkulære relationer skal du først forstå afhængigheder.

Du har f. eks. følgende beregnede **Total** for en kolonne i tabellen Sales.

`Sales['TotalCost'] = Sales['Quantity'] * Sales['Price']`

TotalCost afhænger af **Quantity** og **Price**, så hvis der opstår en ændring i Quantity eller Price, er der også en ændring i **TotalCost**. I dette eksempel beskrives der en kolonnes afhængighed af andre kolonner, men du kan også have afhængigheder mellem målinger, tabeller og relationer.

Der er følgende relationer mellem **dSalesPerson**, **fSales** og **dCustomer**. En ændring i **dCustomer** resulterer i en ændring i **fSales**, hvilket resulterer i ændringer i **dSalesPerson**. Disse typer afhængigheder kan findes i relationer.



Næste undermodul: Øvelse – Modeldata i Power BI Desktop

Tidligere Næste

Har du brug for hjælp? Se i vores [fejlfindingsvejledning](#), eller angiv specifik feedback ved at [rapportere et problem](#).

Express server: Server=localhost\SQLEXPRESS;Database=master;Trusted_Connection=True;

Install media : C:\SQL2022\Express_ENU

Resource folder: C:\Program Files\Microsoft SQL Server\160\SSE\Resources

Lab-øvelser, Setup local lab environment

Ideally, you should complete these labs in a hosted lab environment. If you want to complete them on your own computer, you can do so by installing the following software.

- All setup and resource files can be [downloaded from GitHub](#).
 - Extract the 'AllFiles' folder to D:/ and rename it to 'D:\Allfiles'.

You may experience unexpected dialogs and behavior when using your own environment. Due to the wide range of possible local configurations, the course team cannot support issues you may encounter in your own environment.

Instructions using Windows 11

 The instructions below are for a Windows 11 computer. Connecting from a different OS may not result in the same experience.

Power BI Desktop

1. Download and install from the Microsoft store. If you do not have access to the Microsoft store, download from the [web](#). Power BI Desktop is the primary application for these labs.
 - Use the default options in the installer.

M365 Developer Account

For some of the exercises, you will need to log into Power BI with an organizational account. You can use your own, but if you don't have access, you can create a free [M365 Developer account](#).

SQL Server Database Engine

1. The lab connects to a localhost SQL Server instance. The following instructions will help you install SQL Server and configure the default options. You only need to install the Database Engine feature.
 - Download the free [Developer copy of install media](#)
 - [Install SQL Server from the Installation Wizard \(Setup\)](#)

 You can use an existing SQL Server instance if you have access, instead of installing a local version. However, you'll need to modify the connection string from "localhost" to your instance name.

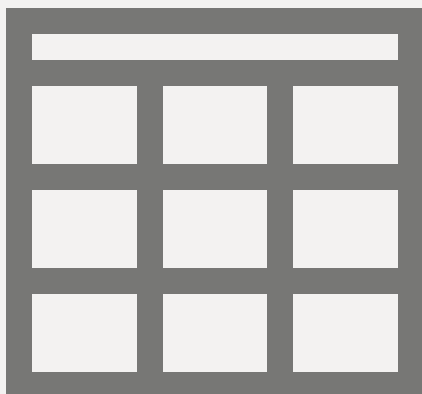
Microsoft Edge

1. Install the latest version of [Microsoft Edge](#) to access Power BI service online.
-
1. To view all table fields, in the **Data** pane, right-click an empty area, and then select **Expand All**.
 2. To create a table visual, in the **Data** pane, from inside the **Product** table, check the **Category** field.
 3. To add another column to the table, in the **Data** pane, check the **Sales | Sales** field.
 4. Notice that the table visual lists four product categories, and that the sales value is the same for each, and the same for the total.

Category	Sum of Sales
Accessories	77,548,570.20
Bikes	77,548,570.20
Clothing	77,548,570.20
Components	77,548,570.20
Total	77,548,570.20

*The issue is that the table is based on fields from different tables. The expectation is that each product category displays the sales for that category. However, because there isn't a model relationship between these tables, the **Sales** table isn't filtered. You'll now add a relationship to propagate filters between the tables.*

5. Select the **Model** view icon from the left navigation pane, and select **Manage Relationships**.



6. In the **Manage Relationships** window, notice that no relationships are yet defined. To create a relationship, select **New relationship**.
7. Configure the relationship from **Product** table to **Sales** table. Notice the following elements were automatically configured:
 - **ProductKey columns in each table are selected.** *The columns were selected because they share the same name and data type. You may need to find matching columns with different names in real data.*
 - **Cardinality type is One To Many (1:*)**. *The cardinality was automatically detected, because Power BI understands that the **ProductKey** column from the **Product** table contains unique values. One-to-many relationships are the most common cardinality, and all relationship you create in this lab will be this type.*
 - **Cross Filter Direction type is Single.** *Single filter direction means that filters propagate from the “one side” to the “many side”. In this case, it means filters applied to the **Product** table will propagate to the **Sales** table, but not in the opposite direction.*
 - **Make This Relationship Active is checked.** *Active relationships propagate filters. It’s possible to mark a relationship as inactive so filters don’t propagate. Inactive relationships can exist when there are multiple relationship paths between tables. In this case, model calculations can use special functions to activate them.*

← New relationship



Select tables and columns that are related.

From table

Product

Background C...	Category	Color	Font Color Fo...	Product	ProductKey	Subcatego
#000000	Components	Black	#FFFFFF	HL Road Fram...	210	Road Fr
#000000	Accessories	Black	#FFFFFF	Sport-100 Hel...	215	Helmets
#000000	Accessories	Black	#FFFFFF	Sport-100 Hel...	216	Helmets

To table

Sales

EmployeeKey	OrderDate	ProductKey	Quantity	ResellerKey	SalesOrderNu...	SalesTerrit
282	Friday, Augus...	235	2	312	SO43897	4
282	Friday, Augus...	351	2	312	SO43897	4
282	Friday, Augus...	348	2	312	SO43897	4

Cardinality

One to many (1:*)

Cross-filter direction

Single

☒ Make this relationship active

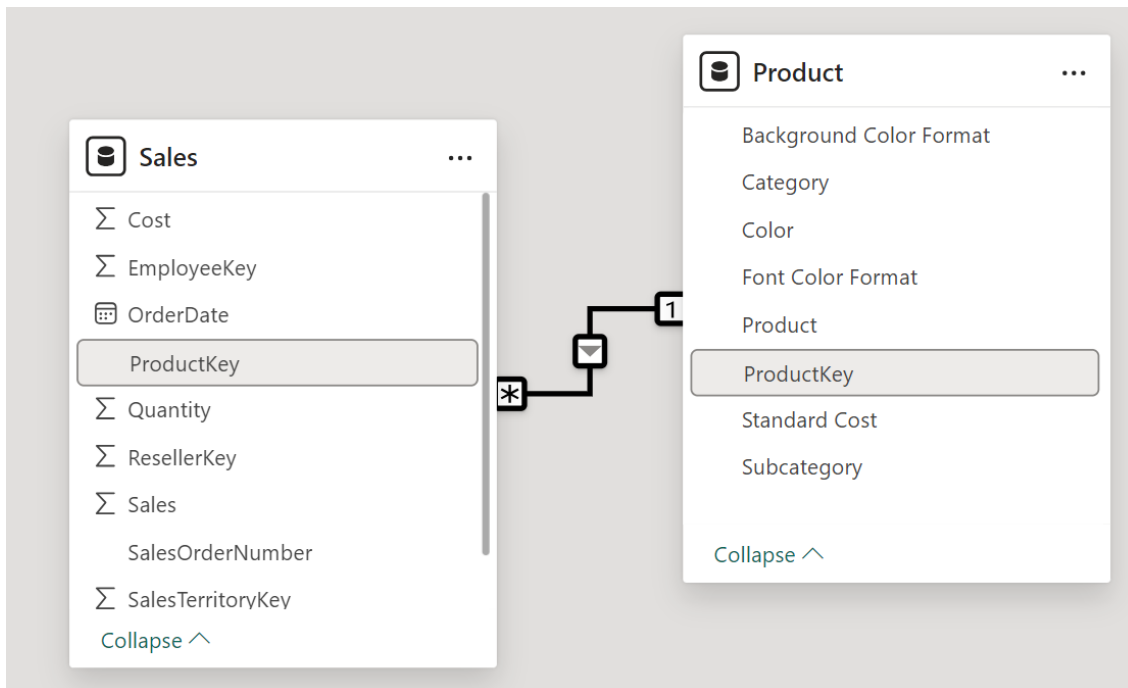
☐ Apply security filter in both directions

☐ Assume referential integrity

Save

Cancel

8. Select **OK**, notice in the **Manage Relationships** window that the new relationship is listed, and then select **Close**.



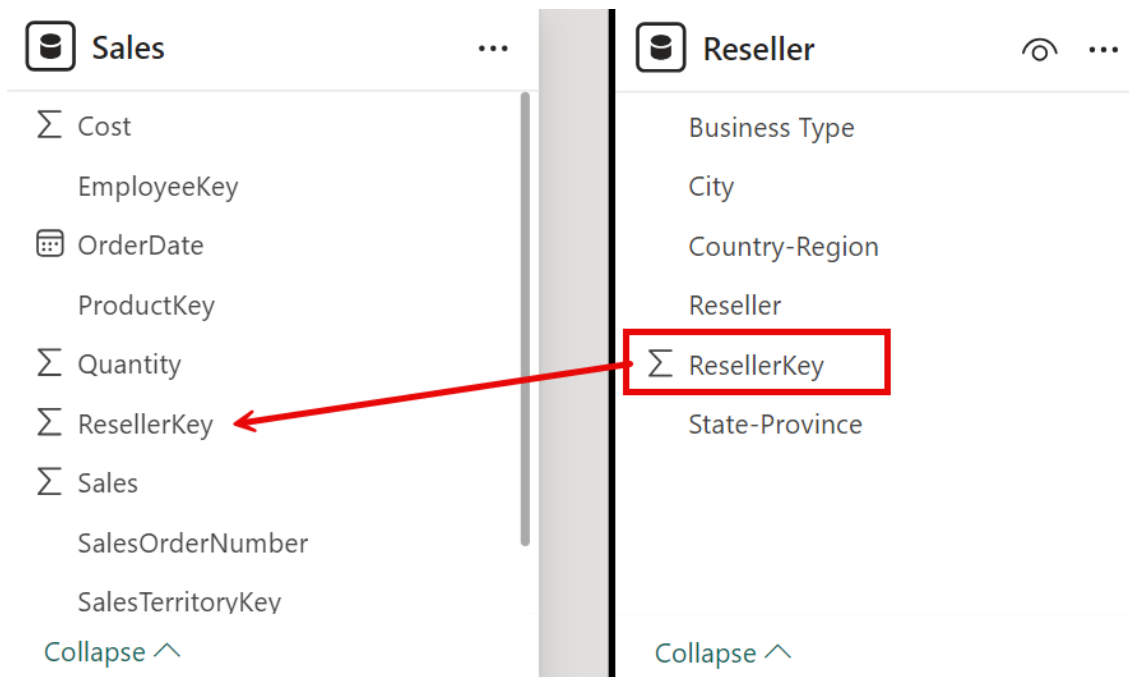
Notice there's now a connector between the two tables (*it doesn't matter if the tables are positioned next to each other*). - You can interpret the cardinality that is represented by the **1** and **(*)** indicators. - Filter direction is represented by the arrow head. - A solid line represents an active relationship; a dashed line represents an inactive relationship. - Hover the cursor over the relationship to highlight the related columns.

Create additional relationships

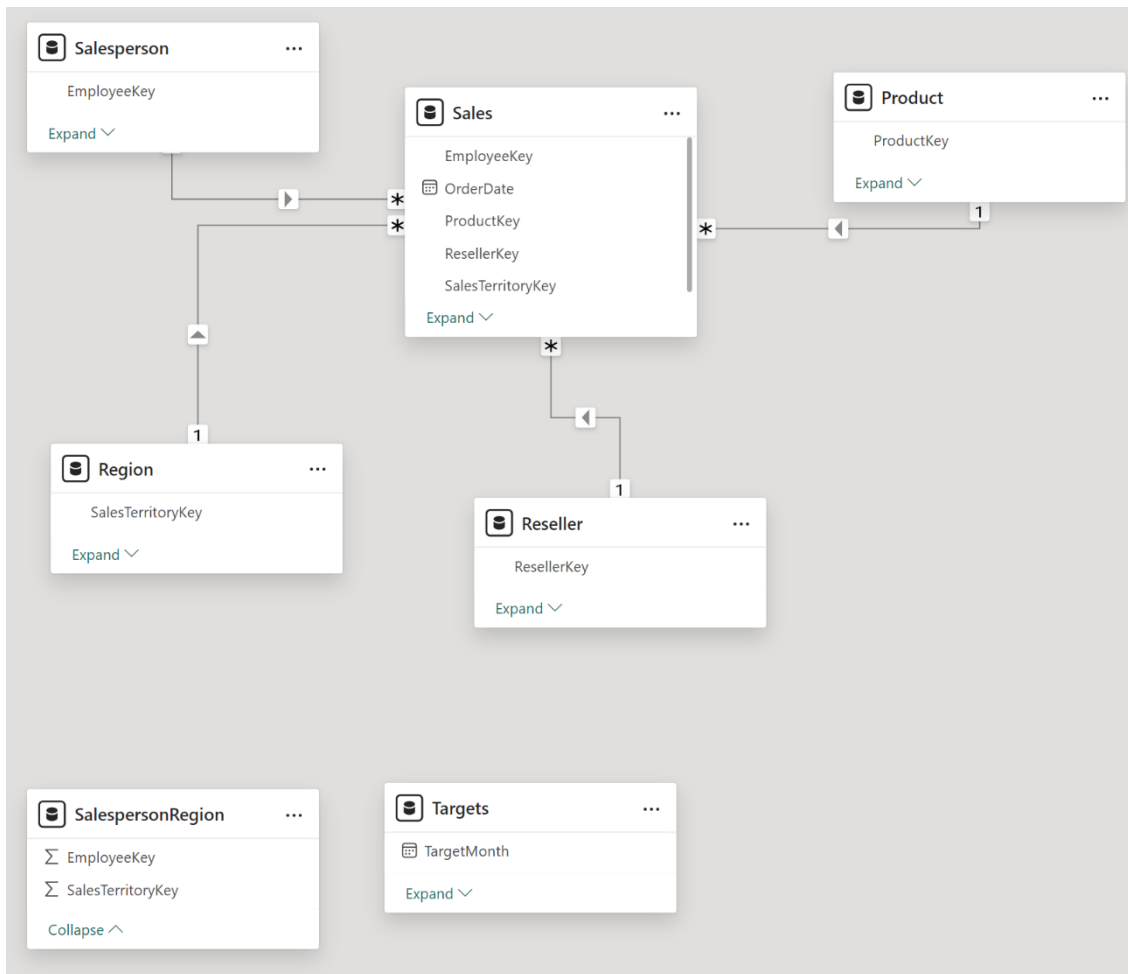
There's an easier way to create a relationship. In the model diagram, you can drag and drop columns to create a new relationship.

1. To create a new relationship using a different technique, from the **Reseller** table, drag the **ResellerKey** column onto the **ResellerKey** column of the **Sales** table.

Tip: *Sometime a column doesn't want to be dragged. If this situation arises, select a different column, and then select the column you intend to drag again, and then try again. Ensure that you see the new relationship added to the diagram.*



2. Use the new technique to create the following two model relationships:
 - **Region | SalesTerritoryKey to Sales | SalesTerritoryKey**
 - **Salesperson | EmployeeKey to Sales | EmployeeKey**
3. In the diagram, arrange the tables so that the **Sales** table is positioned in the center of the diagram, and the related tables are arranged about it. Position the disconnected tables to the side.



4. In the report view, notice that the table visual updated to display different values for each product category.

*Filters applied to the **Product** table now propagate to the **Sales** table.*

Category	Sum of Sales
Accessories	539,106.09
Bikes	64,069,033.16
Clothing	1,714,056.05
Components	11,226,374.90
Total	77,548,570.20

5. **Save** the Power BI Desktop file.

Configure the Product table

In this task, you'll configure the **Product** table with a hierarchy and display folder.

1. In **Model view > Data pane**, if necessary, expand the **Product** table to reveal all fields.
2. To create a hierarchy, in the Data pane, right-click the **Category** column, and then select **Create Hierarchy**.
3. Update the name to **Products** (right-click or double-click to rename).
4. To add the second level to the hierarchy, in the **Properties** pane, in the **Hierarchy** dropdown list, select **Subcategory** (you might need to scroll down inside the pane).
5. To add the third level to the hierarchy, in the **Hierarchy** dropdown list, select **Product**.
6. To complete the hierarchy design, select **Apply Level Changes**.

Hierarchy

Select a column to add level... ▾

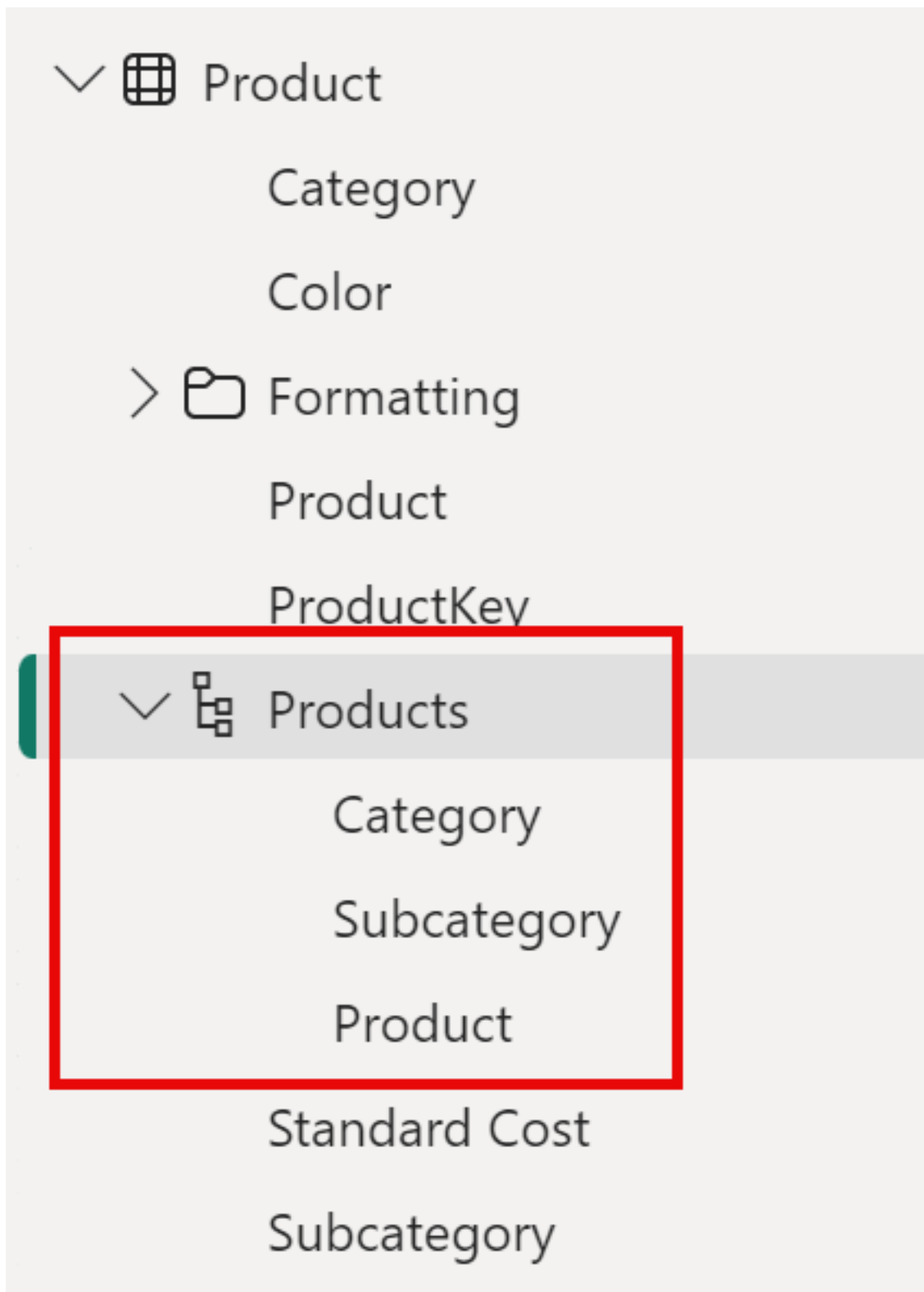
↕ Category (Category) ✕

↕ Subcategory (Subcategory) ✕

↕ Product (Product) ✕

Apply Level Changes

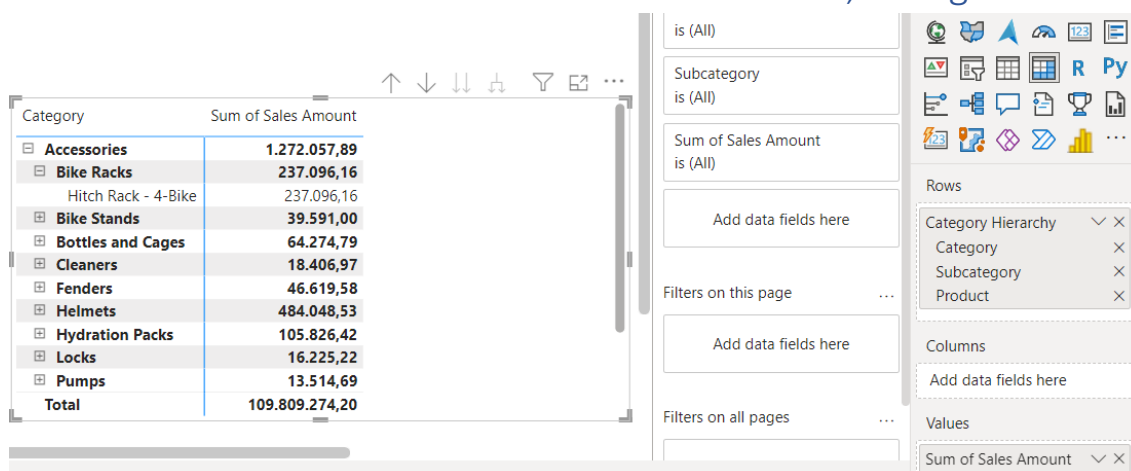
7. In the **Data** pane, notice the **Products** hierarchy. To reveal the hierarchy levels, expand the **Products** hierarchy.



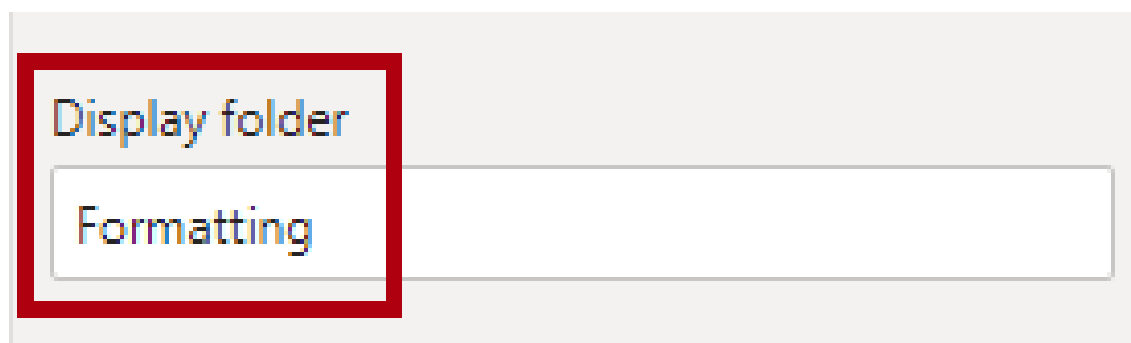
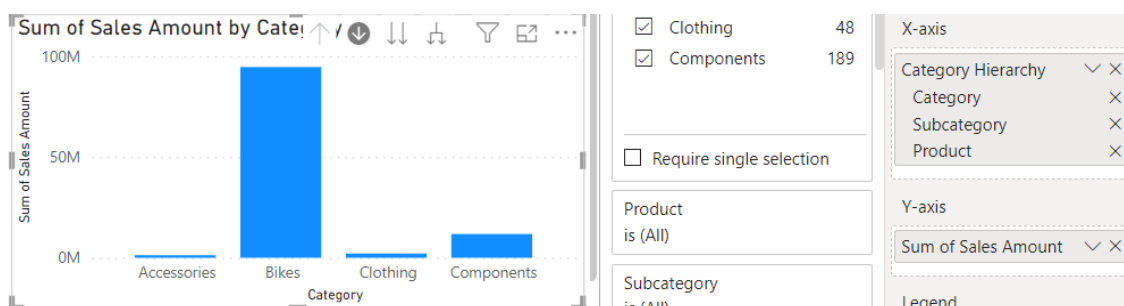
8. To organize columns into a display folder, in the **Data** pane, first select the **Background Color Format** column.
9. While pressing the **Ctrl** key, select the **Font Color Format** column.

10. In the **Properties** pane, in the **Display Folder** box, enter **Formatting**.

Matrix-visual kan fint vies værdier efter hierarkier, hurtigste måde



Ligesom clustered column chart kan bruge det



11. In the **Data** pane, notice that the two columns are now inside a folder.

Display folders are a great way to organize tables, especially for tables that comprise many fields. They're logical presentation only.

![[Picture 349]](Linked_image_Files/03-configure-data-model-in-power-bi-desktop_image30.png)

Configure the Region table

In this task, you'll configure the **Region** table with a hierarchy and updated categories.

1. In the **Region** table, create a hierarchy named **Regions**, with the following three levels:

- Group
 - Country
 - Region
2. Select the **Country** column (not the **Country** hierarchy level).
 3. In the **Properties** pane, expand the **Advanced** section (at the bottom of the pane), and then in the **Data Category** dropdown list, select **Country/Region**.

The screenshot shows the 'Advanced' section of the Properties pane in Power BI. The 'Advanced' section is expanded, indicated by an upward-pointing arrow icon. Below it, there are three settings:

- Sort by column:** A dropdown menu showing 'Country (Default)'.
- Data category:** A dropdown menu showing 'Country/Region'. This section is highlighted with a red box.
- Summarize by:** A dropdown menu showing 'None'.

Data categorization can provide hints to the report designer. In this case, categorizing the column as country or region provides more accurate information to Power BI when it renders a map visualization.

Configure the Reseller table

In this task, you'll configure the **Reseller** table to add a hierarchy and update data categories.

1. In the **Reseller** table, create a hierarchy named **Resellers**, with the following two levels:
 - Business Type
 - Reseller

2. Create a second hierarchy named **Geography**, with the following four levels:
 - Country-Region
 - State-Province
 - City
 - Reseller
3. Set the **Data Category** for the following columns (not within the hierarchy):
 - Country-Region to **Country/Region**
 - State-Province to **State or Province**
 - City to **City**

Configure the Sales table

In this task, you'll configure the **Sales** table with updated descriptions, formatting, and summarization.

1. In the **Sales** table, select the **Cost** column.
2. In the **Properties** pane, in the **Description** box, enter: *Based on standard cost.*

*Descriptions can be applied to tables, columns, hierarchies, or measures. In the **Data** pane, description text is revealed in a tooltip when a report author hovers their cursor over the field.*

1. Select the **Quantity** column.
2. In the **Properties** pane, from inside the **Formatting** section, slide the **Thousands Separator** property to **Yes**.
3. Select the **Unit Price** column.
4. In the **Properties** pane, from inside the **Formatting** section, set the **Decimal Places** property to **2**.
5. In the **Advanced** group (you may need to scroll down to locate it), in the **Summarize By** dropdown list, select **Average**.

*By default, numeric columns will summarize by summing values together. This default behavior isn't suitable for a column like **Unit Price**, which represents a rate. Setting the default summarization to average will produce a meaningful result.*

Bulk update properties

In this task, you'll update multiple columns using single bulk updates. You'll use this approach to hide columns, and format column values.

1. From the **Model view** > **Data** pane, select the **Product | ProductKey** column.
2. While pressing the **Ctrl** key, select the following 13 columns (spanning multiple tables):
 - Region | SalesTerritoryKey
 - Reseller | ResellerKey

- Sales | EmployeeKey
- Sales | ProductKey
- Sales | ResellerKey
- Sales | SalesOrderNumber
- Sales | SalesTerritoryKey
- Salesperson | EmployeeID
- Salesperson | EmployeeKey
- Salesperson | UPN
- SalespersonRegion | EmployeeKey
- SalespersonRegion | SalesTerritoryKey
- Targets | EmployeeID

3. In the **Properties** pane, slide the **Is Hidden** property to **Yes**.

The columns were hidden because they're either used by relationships or will be used in row-level security configuration or calculation logic.

*You'll use the **SalesOrderNumber** in a calculation in the **Create DAX Calculations in Power BI Desktop lab**.*

1. Multi-select the following three columns:

- Product | Standard Cost
- Sales | Cost
- Sales | Sales

2. In the **Properties** pane, from inside the **Formatting** section, set the **Decimal Places** property to **0** (zero).

Explore the model interface

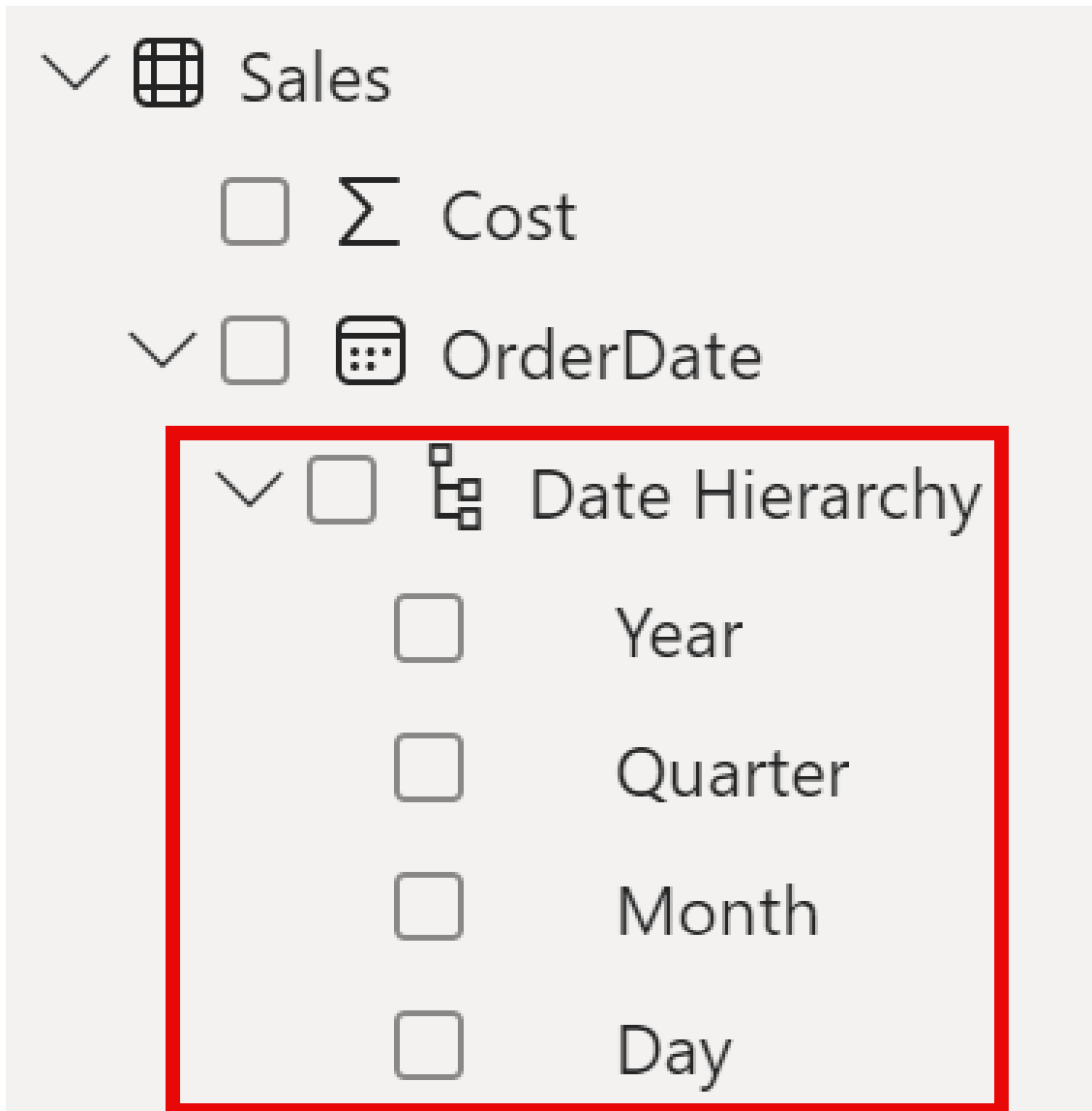
In this task you'll switch to Report view, review the data model interface, and configure the auto date/time setting.

1. Switch to **Report** view.

2. In the **Data** pane, notice the following:

- Columns, hierarchies and their levels are fields, which can be used to configure report visuals
- Only fields relevant to report authoring are visible
- The **SalespersonRegion** table isn't visible—because all of its fields are hidden
- Spatial fields in the **Region** and **Reseller** table are adorned with a spatial icon

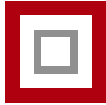
- Fields adorned with the sigma symbol (Σ) will summarize, by default
 - A tooltip appears when hovering the cursor over the **Sales | Cost** field
3. Expand the **Sales | OrderDate** field, and then notice that it reveals a **Date Hierarchy**. The **Targets | TargetMonth** field delivers a similar hierarchy.



Important: *These hierarchies weren't created by you. They were created automatically as a default setting. There's a problem, however. The Adventure Works financial year commences on July 1 of each year. But, in these automatically created date hierarchies, the date hierarchy year commences on January 1 of each year.*

1. To turn off the **Auto date/time** setting, Navigate to **File > Options and Settings > Options**.
2. Under the **Current File** section, navigate to **Data Load > Time Intelligence**, and uncheck **Auto Date/Time**.

Time intelligence



Auto date/time ⓘ [Learn more](#)

3. In the **Data** pane, notice that the date hierarchies are no longer available.

Create quick measures

In this task, you'll create two quick measures to calculate profit and profit margin. A quick measure creates the calculation formula for you. They're easy and fast to create for simple and common calculations.

1. In the **Data** pane, right-click the **Sales** table, and then select **New Quick Measure**.

Select

New measure

New column

New quick measure

2. In the **Quick Measures** window, in the **Calculation** dropdown list, from inside the **Mathematical Operations** group, select **Subtraction**.

Quick measure



Select a calculation to create a measure or describe the measure you need and we'll generate suggestions in DAX, which you can customize later.

Calculations



Suggestions with Copilot

Select a calculation



Quarter-to-date total

Month-to-date total

Year-over-year change

Quarter-over-quarter change

Month-over-month change

Rolling average

Totals

Running total

Total for category (filters applied)

Total for category (filters not applied)

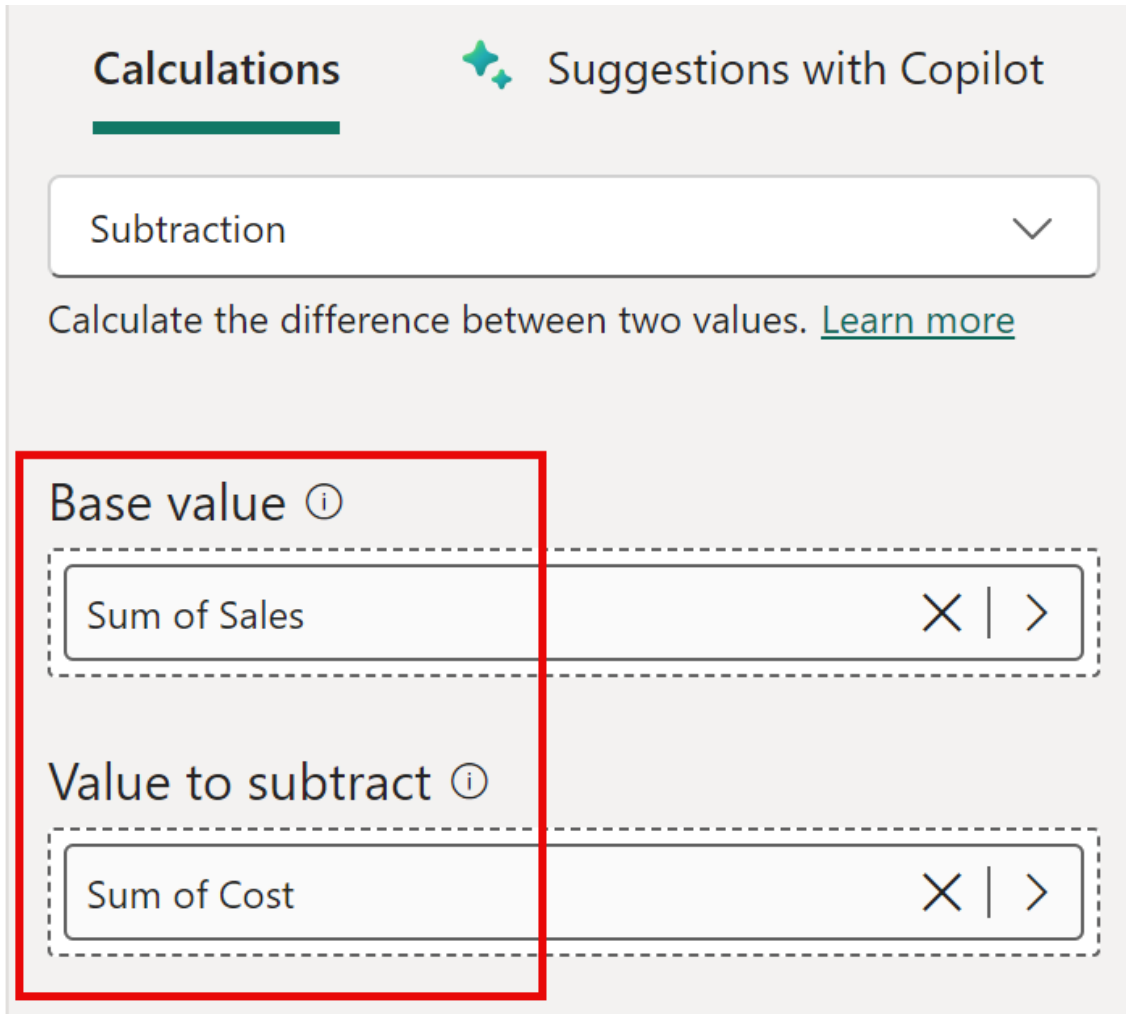
Mathematical operations

Addition

Subtraction

[Skriv her]

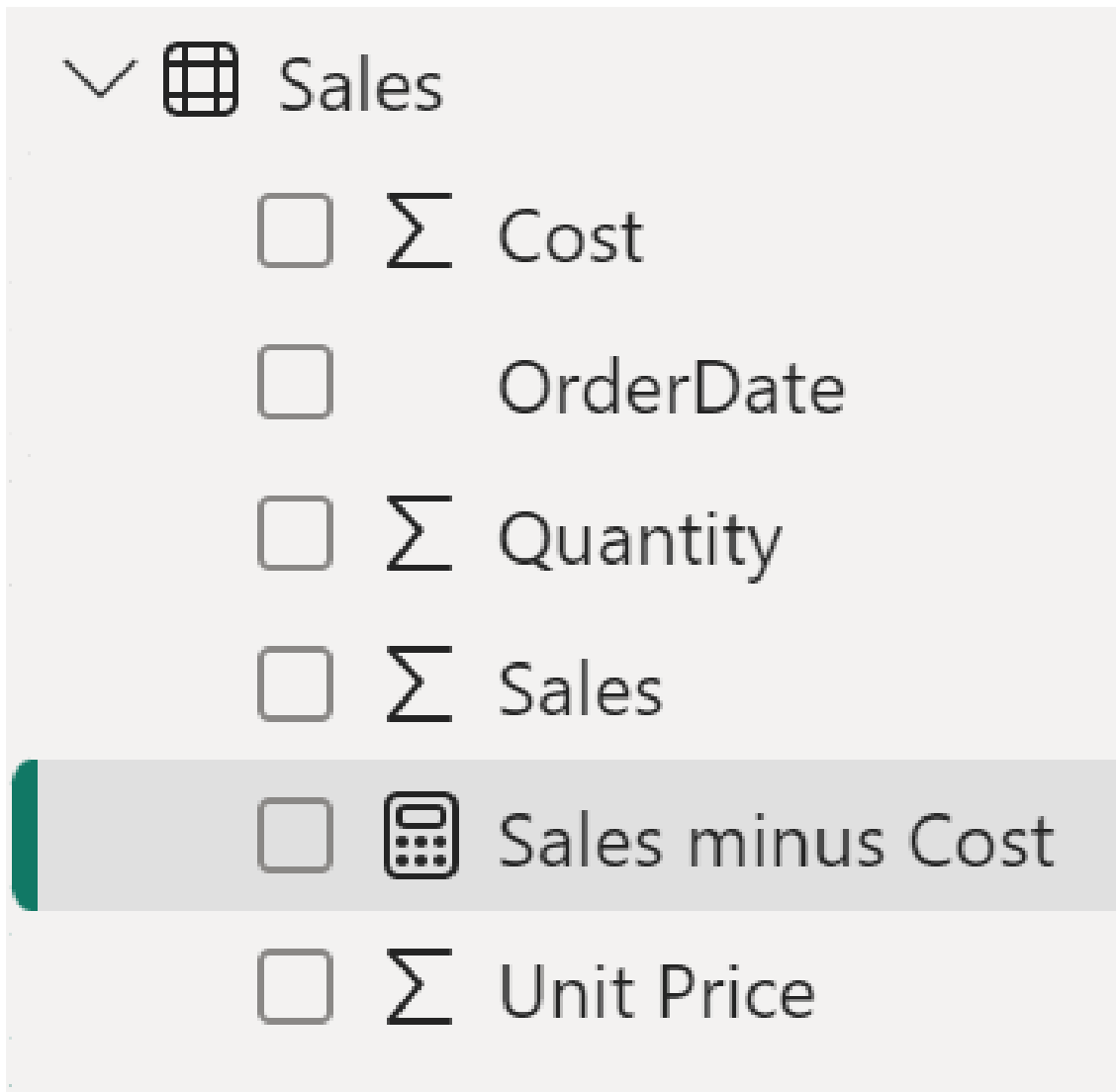
3. In the **Data** pane of the **Quick Measures** window, expand the **Sales** table.
4. Drag the **Sales** field into the **Base Value** box.
5. Drag the **Cost** field into the **Value to Subtract** box, then select **Add**.



The screenshot shows the 'Calculations' pane in Power BI. At the top, there's a 'Subtraction' dropdown menu with a downward arrow. Below it, a text prompt says 'Calculate the difference between two values. [Learn more](#)'. The main area contains two input fields. The first is labeled 'Base value ⓘ' and contains 'Sum of Sales'. The second is labeled 'Value to subtract ⓘ' and contains 'Sum of Cost'. Both input fields have a dashed border and a right-side control with 'X | >' icons. A red rectangular box highlights the 'Base value' and 'Value to subtract' sections.

6. In the **Data** pane, inside the **Sales** table, notice that new measure.

Measures are indicated by the calculator icon.



7. To rename the measure, right-click it, select **Rename**, then rename to **Profit**.

*Tip: To rename a field, you can also double-click it, or select it and press **F2**.*

8. In the **Sales** table, add a second quick measure, based on the following requirements:
 - Use the **Division** mathematical operation
 - Set the **Numerator** to the **Sales | Profit** field
 - Set the **Denominator** to **Sales | Sales** field
 - Rename the measure as **Profit Margin**
9. Ensure the **Profit Margin** measure is selected, and then on the **Measure Tools** contextual ribbon, set the format to **Percentage**, with two decimal places.

Name	Profit Margin
Home table	Sales

Structure

\$ %	Format	Percentage	▼
\$ ▼	%	,	. ⁰⁰ / ₀
			2 ▲▼

Formatting

10. To test the two measures, first select the **Table** visual.

✓  Sales

- ☐ Σ Cost
- ☐ OrderDate
- ☒  Profit
- ☒  Profit Margin
- ☐ Σ Quantity
- ☒ Σ Sales
- ☐ Σ Unit Price

12. Select and drag the right guide to widen the table visual.

Category	Sales	Profit	Pr
Accessories	\$539,106	\$188,081	
Bikes	\$64,069,033	(\$435,679)	
Clothing	\$1,714,056	\$245,857	
Components	\$11,226,375	\$1,001,235	
Total	\$77,548,570	\$999,495	



13. Verify that the measures produce reasonable results that are correctly formatted.

Category	Sales	Profit	Profit Margin
Accessories	\$539,106	\$188,081	34.89 %
Bikes	\$64,069,033	(\$435,679)	-0.68 %
Clothing	\$1,714,056	\$245,857	14.34 %
Components	\$11,226,375	\$1,001,235	8.92 %
Total	\$77,548,570	\$999,495	1.29 %

Create a many-to-many relationship

In this task, you'll create a many-to-many relationship between the **Salesperson** table and the **Sales** table.

1. In Power BI Desktop, in Report view, in the **Data** pane, check the follow two fields to create a new **table** visual.
 - Salesperson | Salesperson
 - Sales | Sales

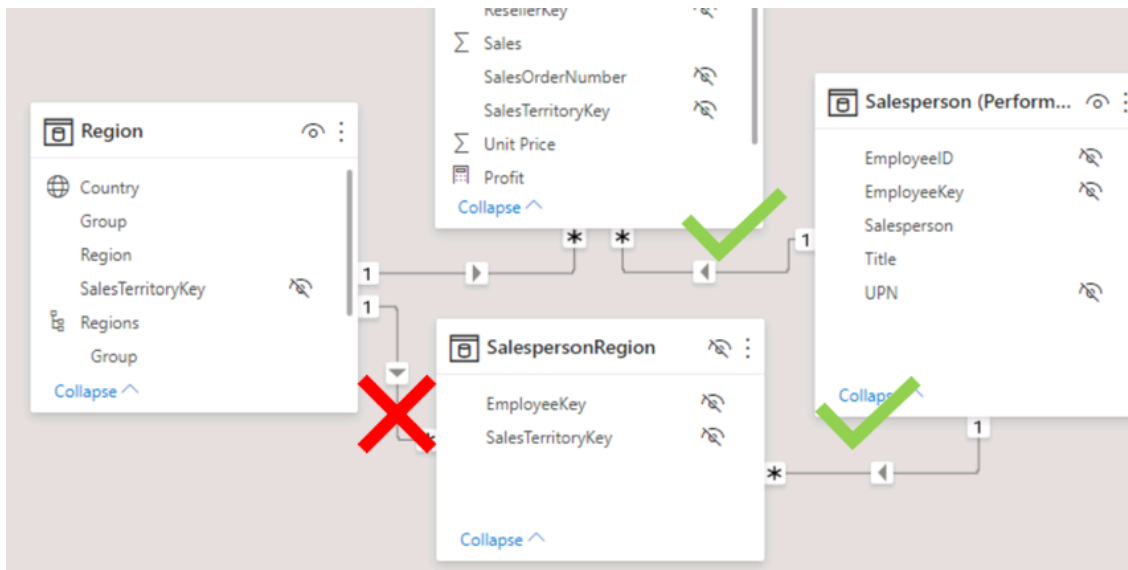
Salesperson	Sales
Amy Alberts	\$737,568
David Campbell	\$3,614,761
Garrett Vargas	\$3,486,102
Jae Pak	\$8,099,817
Jillian Carson	\$9,755,992
José Saraiva	\$5,536,439
Linda Mitchell	\$10,158,635
Lynn Tsoflias	\$1,216,917
Michael Blythe	\$8,952,751
Pamela Ansman-Wolfe	\$3,187,720
Rachel Valdez	\$1,681,208
Total	\$77,548,570

2. The table displays sales made by each salesperson. However, there's another relationship between salespeople and sales. Some salespeople belong to one, two, or possibly more sales regions. In addition, sales regions can have multiple salespeople assigned to them.
3. From a performance management perspective, a salesperson's sales (based on their assigned territories) need to be analyzed and compared with sales targets. You'll create relationships to support this analysis in the next exercise.
4. Notice that **Michael Blythe** has sold almost \$9 million.
5. Switch to Model view, then drag the **SalespersonRegion** table to position it between the **Region** and **Salesperson** tables.
6. Use the drag-and-drop technique to create the following two model relationships:
 - **Salesperson | EmployeeKey** to **SalespersonRegion | EmployeeKey**
 - **Region | SalesTerritoryKey** to **SalespersonRegion | SalesTerritoryKey**

The **SalespersonRegion** table can be considered to be a bridging table.

7. Switch to Report view, and then notice that the visual hasn't updated—the sales result for Michael Blythe hasn't changed.
8. Switch back to Model view, and then follow the relationship filter directions (arrowhead) from the **Salesperson** table.

Consider that the **Salesperson** table filters the **Sales** table. It also filters the **SalespersonRegion** table, but it doesn't continue by propagating filters to the **Region** table (the arrowhead is pointing the wrong direction).

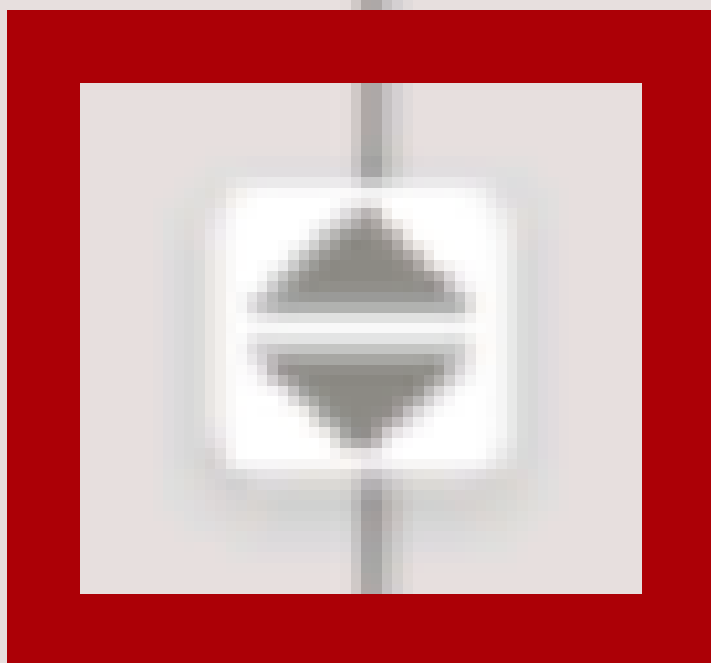


9. To edit the relationship between the **Region** and **SalespersonRegion** tables, double-click the relationship.
10. In the **Edit Relationship** window, in the **Cross Filter Direction** dropdown list, select **Both**.
11. Check the **Apply Security Filter in Both Directions** checkbox, then select OK.

The screenshot shows the **Edit Relationship** window for the relationship between **Region** and **SalespersonRegion**. The **Cardinality** is set to **Many to one (*:1)**. The **Cross filter direction** dropdown is set to **Both**. The **Make this relationship active** checkbox is checked. The **Assume referential integrity** checkbox is unchecked. The **Apply security filter in both directions** checkbox is checked. The **OK** and **Cancel** buttons are at the bottom right.

12. Notice that the relationship has a double arrowhead now.

1



[Skriv her]

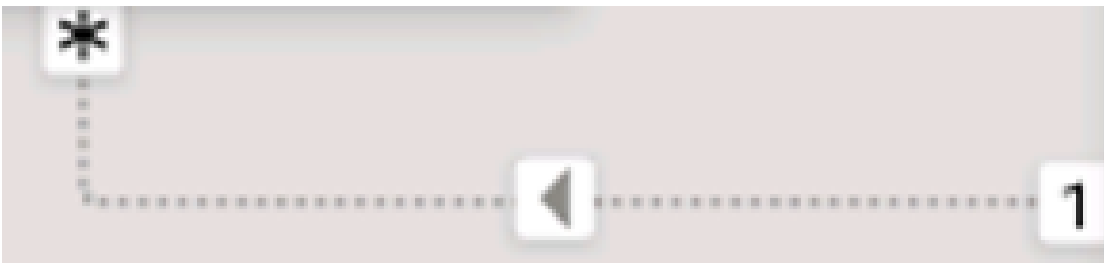
13. Switch to Report view, and then notice that the sales values have still not changed.

*The issue now relates to the fact that there are two possible filter propagation paths between the **Salesperson** and **Sales** tables. This ambiguity is internally resolved, based on a “least number of tables” assessment. To be clear, you shouldn’t design models with this type of ambiguity—the issue will be addressed in part later in this lab, and by the completion of the **Create DAX Calculations in Power BI Desktop** lab.*

14. Switch to Model view to force filter propagation via the bridging table. Edit (double-click) the relationship between the **Salesperson** and **Sales** tables.
15. In the **Edit Relationship** window, uncheck the **Make This Relationship Active** checkbox, and select **OK**.

The filter propagation will now follow the only active path.

16. In the diagram, notice that the inactive relationship is represented by a dashed line.



17. Switch to Report view, and then notice that the sales for Michael Blythe are now nearly \$22 million.

Salesperson	Sales
Amy Alberts	\$10,288,626
Brian Welcker	\$77,548,570
David Campbell	\$12,004,822
Garrett Vargas	\$13,875,633
Jae Pak	\$8,410,883
Jillian Carson	\$7,633,387
José Saraiva	\$13,875,633
Linda Mitchell	\$25,634,503
Lynn Tsoflias	\$1,391,025
Michael Blythe	\$21,987,348
Pamela Ansman-Wolfe	\$30,005,939
Total	\$77,548,570

18. Notice also, that the sales for each salesperson—if added—would exceed the table total.

It's a common observation of a many-to-many relationship due to the double, triple, etc. counting of regional sales results. Consider Brian Welcker, the second salesperson listed. His sales amount equals the total sales amount. It's the correct result due to the fact he's the Director of Sales; his sales are measured by the sales of all regions.

*While the many-to-many relationship is now working, it's now not possible to analyze sales made by a salesperson (because the relationship is inactive). You'll be able to reactivate the relationship when you introduce a calculated table that will allow analyzing sales made in the sales region(s) assigned to the salesperson (for performance analysis) in the **Create DAX Calculations in Power BI Desktop** lab.*

19. Switch to Model view, and then in the diagram, select the **Salesperson** table.

20. In the **Properties** pane, in the **Name** box, replace the text with **Salesperson (Performance)**.

The renamed table now reflects its purpose: it's used to report and analyze the performance of salespeople based on the sales of their assigned sales regions.

Relate the Targets table

In this task, you'll create a relationship to the **Targets** table.

1. Create a relationship from the **Salesperson (Performance) | EmployeeID** column and the **Targets | EmployeeID** column.
2. In Report view, add the **Targets | Target** field to the table visual.
3. Resize the table visual so all columns are visible.

Salesperson	Sales	Target
Amy Alberts	\$10,288,626	\$19,450,000
Brian Welcker	\$77,548,570	\$221,700,000
David Campbell	\$12,004,822	\$19,625,000
Garrett Vargas	\$13,875,633	\$23,675,000
Jae Pak	\$8,410,883	\$13,575,000
Jillian Carson	\$7,633,387	\$13,675,000
José Saraiva	\$13,875,633	\$18,875,000
Linda Mitchell	\$25,634,503	\$40,850,000
Lynn Tsoflias	\$1,391,025	\$3,210,000
Michael Blythe	\$21,987,348	\$31,150,000
Pamela Ansman-Wolfe	\$30,005,939	\$53,850,000
Total	\$77,548,570	\$676,210,000

It's now possible to visualize sales and targets—but take care for two reasons. First, there's no filter on a time period, and so targets also include future target amounts. Second, targets aren't additive, and so the total shouldn't be displayed. They can either be disabled by formatting the visual or removed by using calculation logic.

Lab complete

Congratulations!

You have successfully completed this Module, to mark the lab as complete click **End**.

Test

1.

Hvad betyder datagranularitet?

Den mængde data, du har, hvilket betyder, at højere granularitet betyder, at der er flere data at arbejde med.

Detaljeniveauet i dine data, hvilket betyder, at højere granularitet betyder mere detaljerede data.

Korrekt. Datagranularitet refererer til, hvor fint eller groft dine data opdeles eller aggregeres.

Datagranularitet er en type skemadesign.

2.

Hvad er forskellen mellem en faktatabel og en dimensionstabel?

Faktatabeller indeholder observatoriske data, mens dimensionstabeller indeholder oplysninger om bestemte enheder i dataene.

Korrekt. Faktatabeller indeholder observatoriske data som f. eks. salgsordrer, medarbejdere, forsendelsesdatoer osv., mens dimensionstabeller indeholder oplysninger om bestemte enheder, f. eks. produkt-id'er og datoer.

Faktatabeller indeholder oplysninger om bestemte enheder, mens dimensionstabeller indeholder oplysninger om observatoriske data.

Der er ingen forskel.

3.

Vælg det bedste svar til at forklare relationskardinalitet?

Kardinalitet er målingen af entydige værdier i en tabel.

Korrekt. Et eksempel på høj kardinalitet ville være en sales-tabel, da den har et højt antal entydige værdier.

Kardinalitet er dataenes granularitet.

Kardinalitet er den tid, det tager at indlæse dataene.

Fortsæt det fremragende arbejde



Design en semantisk model i
Power BI

Du har opnået et resultat!

Tillykke, men hvad skal du gøre nu?

Lad os først dele dit resultat

Du afsætter tid til at lære noget nyt, så lad dit netværk tage del i din sejr!

<https://learn.microsoft.com/api/achievements/share/da-dk/bjarnehansen-7412/AP96JGR7?sharingId=39714CB1FE13C67B>

**** Optimer en model i forhold til ydeevne i Power BI

<https://learn.microsoft.com/da-dk/training/modules/optimize-model-power-bi/1-introduction>

- 1 t. 9 min.
- Modul
- 8 Enheder

Feedback

Øvet

Dataanalytiker

Power BI

Microsoft Power Platform

Optimering af ydeevnen, også kendt som justering af ydeevnen, omfatter ændring af den semantiske models aktuelle tilstand, så den kører mere effektivt. **Når din semantiske model er optimeret, fungerer den bedre.**

Læringsmål

Når du har fuldført dette modul, vil du kunne udføre følgende:

- Gennemgå ydeevnen af målinger, relationer og visualiseringer.
- Brug variabler til at forbedre ydeevnen og fejlfinding.
- Gør ydeevnen bedre ved at reducere kardinalitetsniveauer.
- Optimer DirectQuery-modeller med lagring på tabelniveau.
- Opret og administrer sammenlægninger.

[Start](#)Gem

-
-
-

Forudsætninger

Ingen

Dette modul er en del af disse læringsforløb

- [Modeldata med Power BI](#)
- [Work with semantic models in Microsoft Fabric](#)
- [Introduktion til optimering af ydeevnen](#)6 min.
- [Gennemgå ydeevnen af målinger, relationer og visualiseringer](#)19 min.
- [Brug variabler til at forbedre ydeevnen og fejlfinding](#)6 min.
- [Reducer kardinalitet](#)7 min.
- [Optimer DirectQuery-modeller med lager på tabelniveau](#)18 min.
- [Opret og administrer sammenlægninger](#)7 min.
- [Tjek din viden](#)3 min.
- [Resumé](#)3 min.

Introduktion til optimering af ydeevnen

Fuldført100 XP

- 6 minutter

Optimering af ydeevnen, også kaldet justering af ydeevnen, omfatter ændring af den semantiske models aktuelle tilstand, så den kører mere effektivt. Når din semantiske model er optimeret, fungerer den i bund og grund bedre.

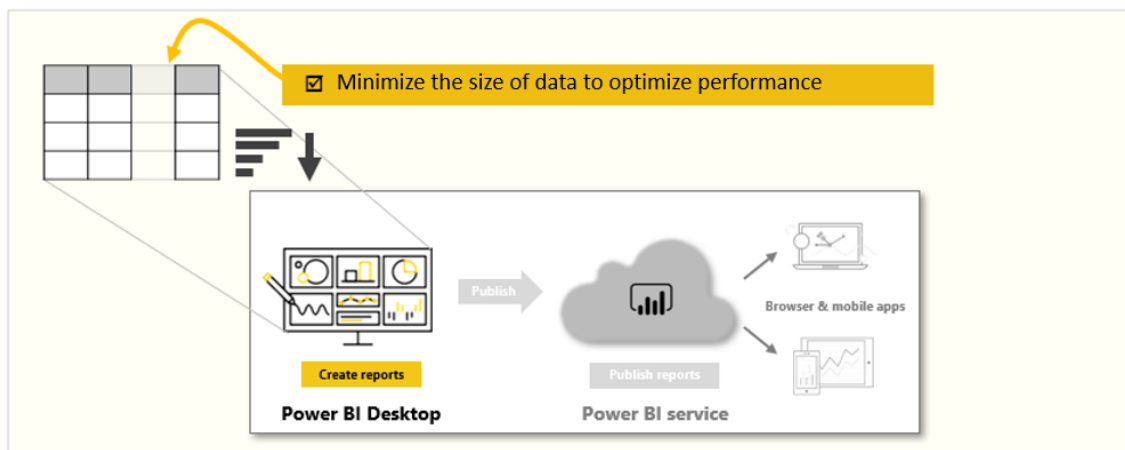
Du kan muligvis se, at din rapport kører godt i test- og udviklingsmiljøer, men når den udrulles til produktion til bredere forbrug, opstår der problemer med ydeevnen. Fra en rapportbrugers perspektiv er dårlig ydeevne karakteriseret ved, at det tager længere tid at indlæse rapportsider, og det tager længere tid at opdatere visualiseringer. Denne dårlige ydeevne giver en negativ brugeroplevelse.

Som dataanalytiker bruger du ca. 90 % af din tid på at arbejde med dine data, og ni gange ud af ti er dårlig ydeevne et direkte resultat af en dårlig semantisk model, forkerte DAX (Data Analysis Expressions) eller blandingen af de to. Processen med at designe en semantisk model til ydeevne kan være kedelig, og den undervurderes ofte. Men hvis du løser problemer med ydeevnen under udvikling, har du en robust Semantisk Power BI-model, der returnerer bedre rapporteringsydeevne og en mere positiv brugeroplevelse. I sidste ende vil du også kunne bevare den optimerede ydeevne. Efterhånden som din organisation vokser, vokser størrelsen af dens data, og dens semantiske model bliver mere kompleks. Ved at optimere din semantiske model tidligt kan du afhjælpe den negative indvirkning, som denne vækst kan have på ydeevnen af din semantiske model.

En semantisk model i mindre størrelse bruger færre ressourcer (hukommelse) og opnår hurtigere opdatering af data, beregninger og gengivelse af visualiseringer i rapporter. Processen til optimering af ydeevnen omfatter derfor at minimere størrelsen af den semantiske model og gøre den mest effektive brug af dataene i modellen, hvilket omfatter:

Mindske modelstørrelsen og andre modelforbedringer:

- Sikring af, at de korrekte datatyper bruges.
- Sletning af unødvendige kolonner og rækker.
- Undgåelse af gentagne værdier.
- Erstatning af numeriske kolonner med målinger.
- Reduktion af kardinaliteter.
- Analysering af modellens metadata.
- Opsummering af data, hvor det er muligt.



I dette modul introduceres du til de trin, processer og begreber, der er nødvendige for at optimere en semantisk model til ydeevne på virksomhedsniveau. Vær dog opmærksom på, at selvom den grundlæggende vejledning til ydeevne og bedste praksis i Power BI fører dig langt for at optimere en semantisk model til forespørgselydeevne, skal du sandsynligvis samarbejde med en datatekniker for at drive optimering af semantisk model i kildedatakilderne.

Forestil dig f.eks., at du **arbejder som Microsoft Power BI-udvikler for Tailwind Traders**. Du har fået en opgave til at gennemse en semantisk model, der blev bygget for et par år siden af en anden udvikler, en person, der siden har forladt organisationen.

Den semantiske model opretter en rapport, der har modtaget negativ feedback fra brugerne. Brugerne er glade for de resultater, de kan se i rapporten, men de er ikke tilfredse med rapportens ydeevne. Det tager for lang tid at indlæse siderne i rapporten, og tabellerne opdateres ikke hurtigt nok, når der foretages bestemte valg. Ud over denne feedback har it-teamet fremhævet, at filstørrelsen på denne bestemte semantiske model er for stor, og det belaster organisationens ressourcer.

Du skal gennemse den semantiske model for at identificere den grundlæggende årsag til problemer med ydeevnen og foretage ændringer for at optimere ydeevnen.

I slutningen af dette modul kan du:

- Gennemgå ydeevnen af målinger, relationer og visualiseringer.
- Brug variabler til at forbedre ydeevnen og fejlfinding.
- Gør ydeevnen bedre ved at reducere kardinalitetsniveauer.
- Optimer DirectQuery-modeller med lagring på tabelniveau.
- Opret og administrer sammenlægninger.

Gennemse performance for målinger, relationer og grafik

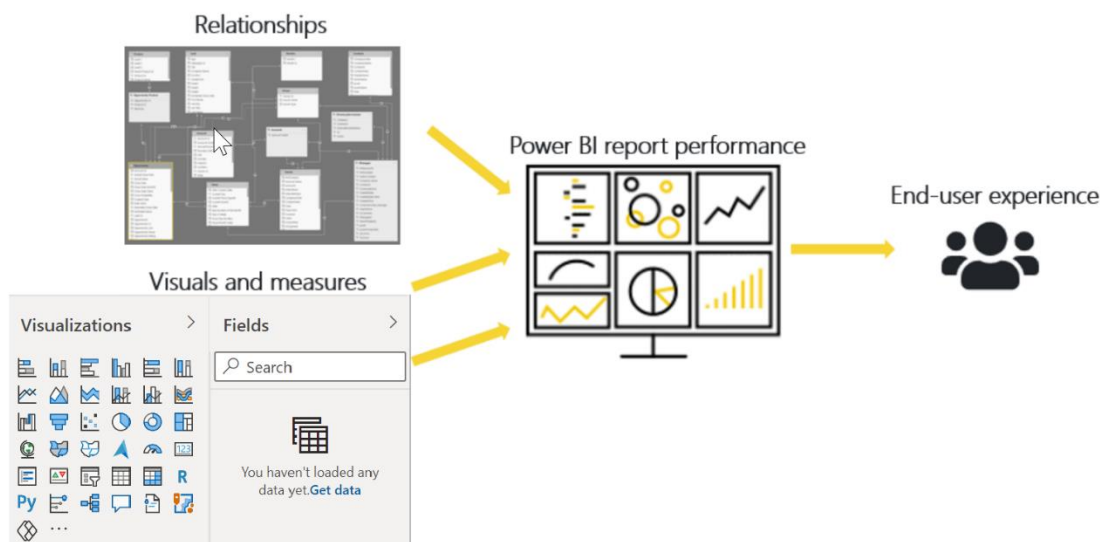
[Næste](#)

Gennemse performance for målinger, relationer og grafik,

Fuldført 100 XP

- 19 minutter

Hvis din semantiske model har flere tabeller, komplekse relationer, komplicerede beregninger, flere visuals eller overflødige data, findes der et potentiale for dårlig rapportydeevne. En rapportes dårlige ydeevne medfører en negativ brugeroplevelse.



For at optimere ydeevnen skal du først identificere, hvor problemet kommer fra. Du skal med andre ord finde ud af, hvilke elementer i rapporten og den semantiske model der forårsager problemer med ydeevnen. Derefter kan du løse disse problemer og dermed forbedre ydeevnen.

Identificere flaskehalse for rapportydeevne

For at opnå optimal performance i rapporterne skal du oprette en effektiv semantisk model, der kører forespørgsler og målinger hurtigt. Når du har et godt grundlag, kan du forbedre modellen yderligere ved at analysere forespørgselsplanerne og afhængighederne og derefter foretage ændringer for at optimere performance yderligere.

Du bør gennemse målingerne og forespørgslerne i din semantiske model for at sikre, at du bruger den mest effektive måde til at få de ønskede resultater på. Udgangspunktet bør være at identificere flaskehalse, der findes i koden. Når du identificerer den langsommere forespørgsel i den semantiske model, **kan du først fokusere på den største flaskehals og oprette en prioritetsliste, der kan bruges til at gennemgå de andre problemer.**

Analysér ydeevne

Du kan bruge performanceanalyser **som** en hjælp Power BI Desktop til at finde ud af, hvordan hvert af rapportelementerne fungerer, når brugerne samarbejder med dem. Du kan f.eks. bestemme, hvor lang tid det tager for en bestemt visuel grafik at opdatere den, når den startes via en brugerind interaktion. **Performanceanalyse kan hjælpe** dig med at identificere de elementer, der bidrager til de problemer med performance, som kan være nyttige ved fejlfinding.

Før du kører **performanceanalyser**, skal du sikre dig, at du får de mest nøjagtige resultater i analysen (testen), så du starter med en klar visuel cache og en klar cache til dataprogrammet.

- **Visuel cache** – Når du indlæser en visuel cache, kan du ikke rydde denne visuelle cache uden at lukke Power BI Desktop og åbne den igen. For at undgå cachelagring i spil, skal du starte analysen med en ren visuel cache.

Hvis du vil sikre, at der er en klar visuel cache, Power BI Desktop skal du føje en tom side til din (.pbix)-fil og derefter gemme og lukke filen, når den pågældende side er valgt. Åbn den Power BI Desktop (.pbix)-fil igen, som du vil analysere. Den åbnes på den tomme side.

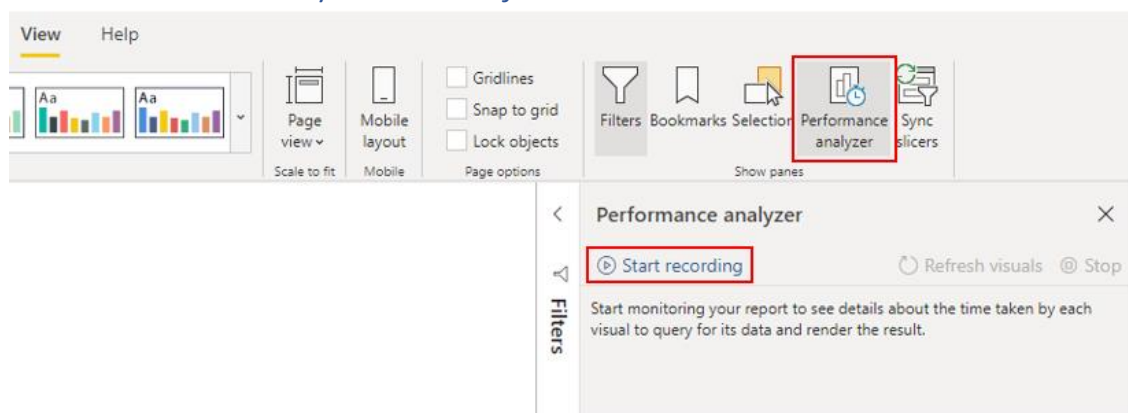
- **Cache i dataprogrammet** – Når en forespørgsel køres, cachelagres resultaterne, så analysens resultater vises på samme måde. Du skal rydde datacachen, før visualiseringen køres igen.

Hvis du vil rydde datacachen, kan du enten Power BI Desktop genstarte eller forbinde DAX Studio til den semantiske model og derefter kalde Clear Cache.

Når du har ryddet cachelagrene Power BI Desktop og åbnet filen på den tomme side, skal du gå til fanen Vis og vælge indstillingen **Performanceanalyse**.

Hvis du vil starte analyseprocessen, skal du vælge **Start** registrering, vælge siden i den rapport, du vil analysere, og arbejde sammen med elementerne i den rapport, du vil måle. Du kan se resultaterne af dine interaktioner i ruden **Performanceanalyse**, når du arbejder. Når du er færdig, skal du vælge **stopknappen**.

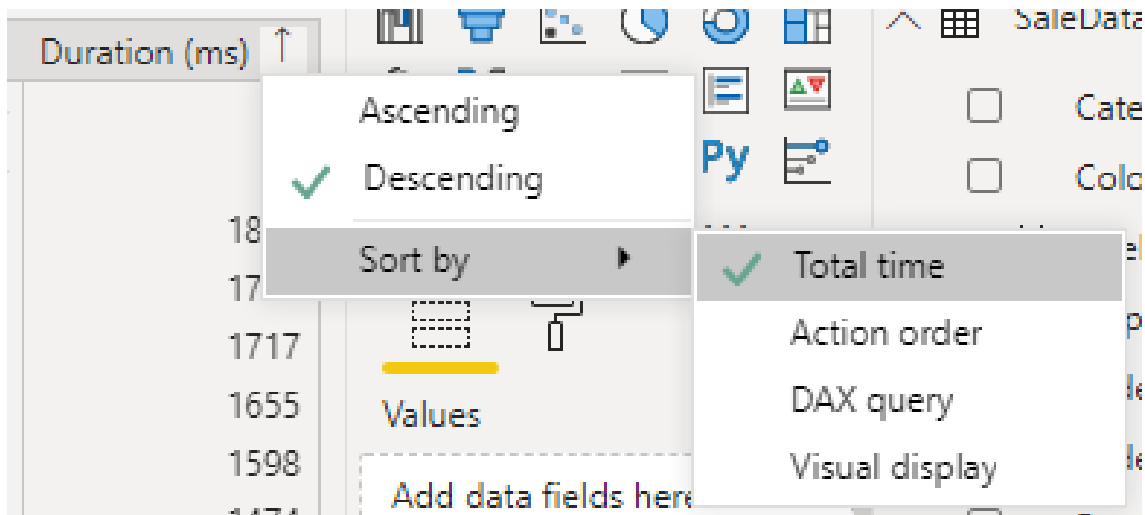
Performance-analyse-værktøj



Du kan finde mere detaljerede oplysninger i [Brug Performance Analyzer til at undersøge performance for rapportelementer](#).

Gennemse resultater

Du kan gennemse resultaterne af din performancetest i ruden **Performanceanalyse**. Hvis du vil gennemse opgaverne i rækkefølge efter varighed, skal du længste til korteste ved at højreklikke på ikonet **Sorter ud for kolonnehovedet varighed (ms)** og derefter vælge Total tid i faldende rækkefølge.



Logoplysningerne for de enkelte visuelle elementer viser, hvor lang tid det har taget (varighed) at udføre følgende opgavekategorier:

- **DAX-forespørgsel** – Den tid det har taget det for det visuelle element at sende forespørgslen sammen med den tid, det har taget Analysis Services at returnere resultaterne.
- **Visuel visning** – Den tid det har taget den visuelle visning at vise på skærmen, herunder den tid, det tager at hente webbilleder eller kodning.
- **Anden** – Den tid det har taget den visuelle fremstilling at forberede forespørgsler, vente på, at andre billeder er udført, eller udføre andre opgaver til behandling af baggrundsoplysninger. Hvis denne kategori viser en lang varighed, er den eneste reelle metode til at reducere denne varighed at optimere DAX-forespørgsler for andre visuals eller reducere antallet af visuals i rapporten.

Performance analyzer ×	
Start recording	Refresh visuals Stop
Clear Export	
Name	Duration (ms)
Recording started (8/06/2020 10:59:48 a.m.)	-
Changed page	-
Card	1913
DAX query	94
Visual display	21
Other	1797
Copy query	
Slicer	1855
Sales by Quarter and Region	1770
Sales by Region and Color	1669
Card	1499
Card	1431
Slicer	1356
Sales by Region	1329
Sales by Quarter	1270

Resultaterne af analysetesten hjælper dig med at forstå din semantiske model og identificere de elementer, du skal optimere. Du kan sammenligne varigheden af de enkelte elementer i rapporten og identificere de elementer, der har en lang varighed. Du skal fokusere på disse elementer og undersøge, hvorfor det tager dem så lang tid at blive indlæst på rapportens side.

Hvis du vil analysere dine forespørgsler mere detaljeret, kan du bruge DAX Studio, som er et gratis, åbent kildeværktøj, der leveres af en anden tjeneste.

Løse problemer og optimere performance

Resultaterne af din analyse identificerer områder for forbedring og muligheder for performanceoptimering. Du vil muligvis finde ud af, at du skal foretage forbedringer af visualiseringer, DAX-forespørgslen eller andre elementer i din semantiske model. Følgende oplysninger indeholder vejledning i, hvad du skal søge efter, og hvilke ændringer du kan foretage.

Visuals

Hvis du identificerer grafik som flaskehals, der medfører dårlig performance, skal du finde en måde at forbedre ydeevnen på med minimal effekt på brugeroplevelsen.

Overvej antallet af visuelle elementer på rapport siden. Hvis der ikke er så mange visuelle elementer, giver det bedre ydeevne. Spørg dig selv, om en visuel elementer er rigtig nødvendigt, og om det giver slutbrugereren værdi. Hvis svaret er nej, skal du fjerne det visuelle element. I stedet for at bruge flere visualiseringer på siden kan du overveje andre måder at angive flere oplysninger på, f.eks. værktøjstip til gennemrulling af sider og rapport sider.

Kontroller antallet af felter i de enkelte visuelle elementer. Jo mere visuel grafik du har i rapporten, jo større er risikoen for performanceproblemer. Jo flere billeder, der er, jo flere kan rapporten desuden vises, hvad der ellers kan være, hvis rapporten mister identiteten. Den øvre grænse for visuelle elementer er 100 felter (målinger eller kolonner), så det bliver langsomt at indlæse en grafik med mere end 100 felter. Spørg dig selv, om du har brug for alle disse data visuelt. Du kan måske finde ud af, at du kan reducere antallet af felter, du bruger i øjeblikket.

Performanceanalyse

DAX-forespørgsel

Når du undersøger resultaterne i **ruden Performanceanalyse**, kan du se, Power BI Desktop hvor lang tid det har taget programmet at evaluere hver forespørgsel (i millisekunder). **Et godt udgangspunkt er en DAX-forespørgsel, der tager længere tid end 120 millisekunder.** I dette eksempel identificerer du en bestemt forespørgsel, der har en stor varighed.

[-] Sales by Year	270
DAX query	2754
Visual display	57
Other	160
Copy query	

Performanceanalyser fokuserer på potentielle problemer, men ved ikke, hvad der skal gøres for at forbedre dem. Det kan være en god ide at foretage en yderligere undersøgelse af, hvorfor det tager så lang tid at behandle denne måling. Du kan bruge DAX Studio til at undersøge dine forespørgsler mere detaljeret.

Vælg f.eks. **Kopier forespørgsel** for at kopiere beregningsformlen til Udklipsholder, og indsæt den i Dax Studio. Derefter kan du gennemse trin detaljer. I dette eksempel forsøger du at tælle det samlede antal produkter med ordreantal, der er større end eller lig med fem.

Kopier

Count Customers =

```
CALCULATE (
    DISTINCTCOUNT ( Order[ProductID] ),
    FILTER ( Order, Order[OrderQty] >= 5 )
)
```

Når du har analyseret forespørgslen, kan du bruge din egen viden og erfaring til at identificere, hvor performance-problemer ligger. Du kan også prøve at bruge forskellige DAX-funktioner for at se, om de forbedrer ydeevnen. I følgende eksempel er FILTER-funktionen udskiftet med funktionen KEEPFILTER. Da testen blev kørt igen i Performanceanalyse, blev varigheden kortere som et resultat af funktionen KEEPFILTER.

Kopier

Count Customers =

```
CALCULATE (
    DISTINCTCOUNT ( Order[ProductID] ),
    KEEPFILTERS (Order[OrderQty] >= 5 )
)
```

I dette tilfælde kan du erstatte filterfunktionen med FUNKTIONEN KEEPFILTER for at reducere varigheden af evaluering for denne forespørgsel væsentligt. Når du foretager denne ændring for at kontrollere, om varigheden er blevet forbedret eller ej, skal du rydde datacachen **og derefter køre performanceanalyseprocessen** igen.

Sales by Year	270
DAX query	54
Visual display	57
Other	160
Copy query	

Semantisk model

Hvis varigheden af målinger og visuelle elementer viser lave værdier (med andre ord hvis de har en kort varighed), er de ikke årsagen til problemer med ydeevnen. Hvis DAX-forespørgslen i stedet viser en høj varighedsværdi, er det i stedet sandsynligt, at der skrives et målepunkt forkert, eller at der er opstået et problem med den semantiske model. Problemet kan skyldes relationerne, kolonnerne eller metadataene i modellen, eller det kan være status **for indstillingen Automatisk dato/ klokkeslæt**, som forklaret i følgende afsnit.

Relationer

Du skal gennemse relationerne mellem tabellerne for at sikre, at du har oprettet de korrekte relationer. Kontroller, at egenskaberne for relationer er konfigureret korrekt. En kolonne på én side, der indeholder entydige værdier, kan f.eks. være konfigureret forkert som en kolonne på mange sider. Du får mere at vide om, hvordan alitet påvirker ydeevnen senere i dette modul.

Kolonner

Det er en **god ide ikke at importere datakolonner, som du ikke har brug for**. Hvis du vil undgå at slette kolonner Power Query i Editor, skal du prøve at tage højde for dem ved kilden, når du indlæser data i Power BI Desktop. Men hvis du ikke kan fjerne overflødige kolonner fra kildeforespørgslen, eller dataene allerede er

importeret i den rå tilstand, Power Query kan du altid bruge Editor til at undersøge hver kolonne. Spørg dig selv, om du rigtig har brug for hver kolonne, og prøv at identificere de fordele, som hver enkelt føjer til din semantiske model. **Hvis du opdager, at en kolonne ikke tilføjer nogen værdi, skal du fjerne den fra den semantiske model.** Antag f.eks., at du har en id-kolonne med tusindvis af entydige rækker. Du ved, at du ikke vil bruge denne bestemte kolonne i en relation, så den bruges ikke i en rapport. Derfor bør du overveje denne kolonne som unødvendig og overflødig, når den spilder plads i den semantiske model.

Når du fjerner en unødvendig kolonne, reducerer du størrelsen på den semantiske model, hvilket gengæld resulterer i en mindre filstørrelse og hurtigere opdateringstid. Da den semantiske model kun indeholder relevante data, vil den overordnede rapportydeevne blive forbedret.

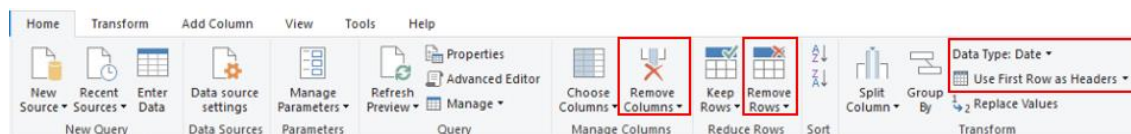
Du kan finde flere oplysninger i [teknikkerne til datareduktion i Import af modellering](#).

Metadata

Metadata er oplysninger om andre data. Power BI metadata indeholder oplysninger om din semantiske model, f.eks. navn, datatype og format for hver af kolonnerne, databasens skema, rapportdesign, sidste gang filen blev ændret, dataopdateringshastigheden og meget mere.

Når du indlæser data Power BI Desktop i, er det en god ide at analysere de tilsvarende metadata, så du kan identificere eventuelle uoverensstemmelser med din semantiske model og normalisere dataene, før du begynder at oprette rapporter. Hvis du kører analyser af metadataene, forbedres den semantiske modelydeevne, for når du analyserer metadataene, **kan du identificere unødvendige kolonner, fejl i dataene, forkerte datatyper, mængden af data**, der indlæses (store semantiske modeller, herunder transaktions- eller historiske data, vil tage længere tid at blive indlæst), og meget mere.

Du kan bruge Power Query Editor til Power BI Desktop at undersøge kolonnerne, rækkerne og værdierne for de rådata. Du kan derefter bruge de tilgængelige værktøjer, f.eks. dem, der er fremhævet i følgende skærmbillede, til at foretage de nødvendige ændringer.

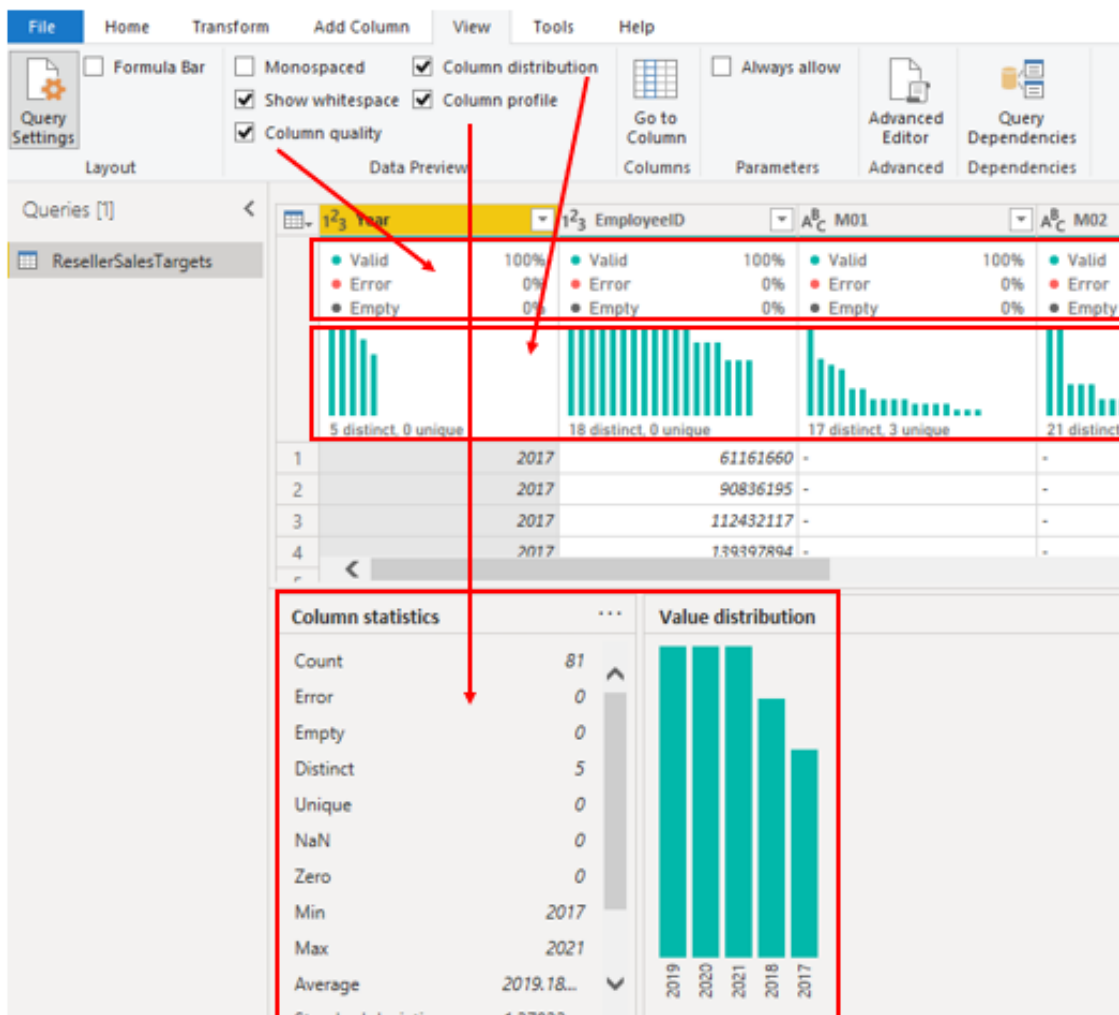


Valgmulighederne Power Query omfatter:

- **Unødvendige kolonner** – Evaluerer behovet for hver kolonne. Hvis en eller flere kolonner ikke bruges i rapporten, og derfor ikke er nødvendige, skal du fjerne dem ved at bruge indstillingen Fjern kolonner under fanen Startside.
- **Unødvendige rækker** – Kontrollerer de første få rækker i den semantiske model for at se, om de er tomme, eller om de indeholder data, du ikke har brug for i rapporterne. Hvis det er nødvendigt, fjernes disse rækker ved hjælp af indstillingen Fjern rækker under fanen Startside.
- **Datatype** - Evaluerer kolonnedatatyperne for at sikre, at de er korrekte. Hvis du identificerer en datatype, der er forkert, kan du ændre den ved at markere kolonnen, vælge Datatype **under fanen Transformer** og derefter vælge den korrekte datatype på listen.
- **Forespørgselsnavne** – Undersøger navnene på forespørgslen (tabellen) i **ruden** Forespørgsler. På samme måde som du brugte navne på kolonneoverskrifter, bør du ændre navnene på forespørgselsnavne, der er negative, til navne, der er mere genkendelige eller mere genkendelige,

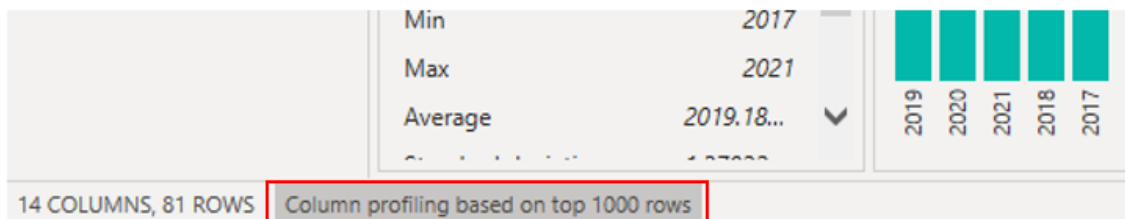
og som brugeren er mere fortrolig med. Du kan omdøbe en forespørgsel ved at højreklikke på den pågældende forespørgsel, vælge Omdøb , redigere navnet efter behov og derefter trykke på **Enter**.

- **Kolonnededetaljer-Editor** Power Query har følgende tre dataindstillinger forhåndsversion du kan bruge til at analysere de metadata, der er knyttet til kolonnerne. Du kan finde disse indstillinger under fanen Vis , som det er illustreret i følgende skærbillede.
 - **Kolonnekvalitet** – Bestemmer, hvor stor en procentdel af varerne i kolonnen der er gyldige, har fejl eller er tomme. Hvis den gyldige procent ikke er 100, skal du undersøge årsagen, rette fejlene og udfylde tomme værdier.
 - **Kolonnefordeling** – Viser hyppigheden og fordelingen af værdierne i hver af kolonnerne. Du vil undersøge dette yderligere senere i dette modul.
 - **Kolonneprofil** – Viser søjlestatistikdiagram og et kolonnefordelingsdiagram.



Bemærk

Hvis du gennemgår en stor semantisk model med mere end 1.000 rækker, og du vil analysere hele den semantiske model, skal du ændre standardindstillingen nederst i vinduet. Vælg **kolonneprofilering baseret på øverste 1000 rækker kolonneprofilering>baseret på hele datasættet**.



Andre metadata, du skal overveje, er oplysningerne om den semantiske model som helhed, f.eks. opdateringshastigheden for filstørrelse og data. Du kan finde disse metadata i den tilknyttede Power BI Desktop (.pbix)-fil. De data, du indlæser Power BI Desktop i, komprimeres og gemmes på disken af vertiPac-lagringsprogrammet. Størrelsen på den semantiske model har direkte indflydelse på dens ydeevne. en semantisk model i mindre størrelse bruger mindre ressourcer (hukommelse) og giver hurtigere dataopdatering, beregninger og visning af visuelle elementer i rapporter.

Automatisk dato/klokkeslætsfunktion

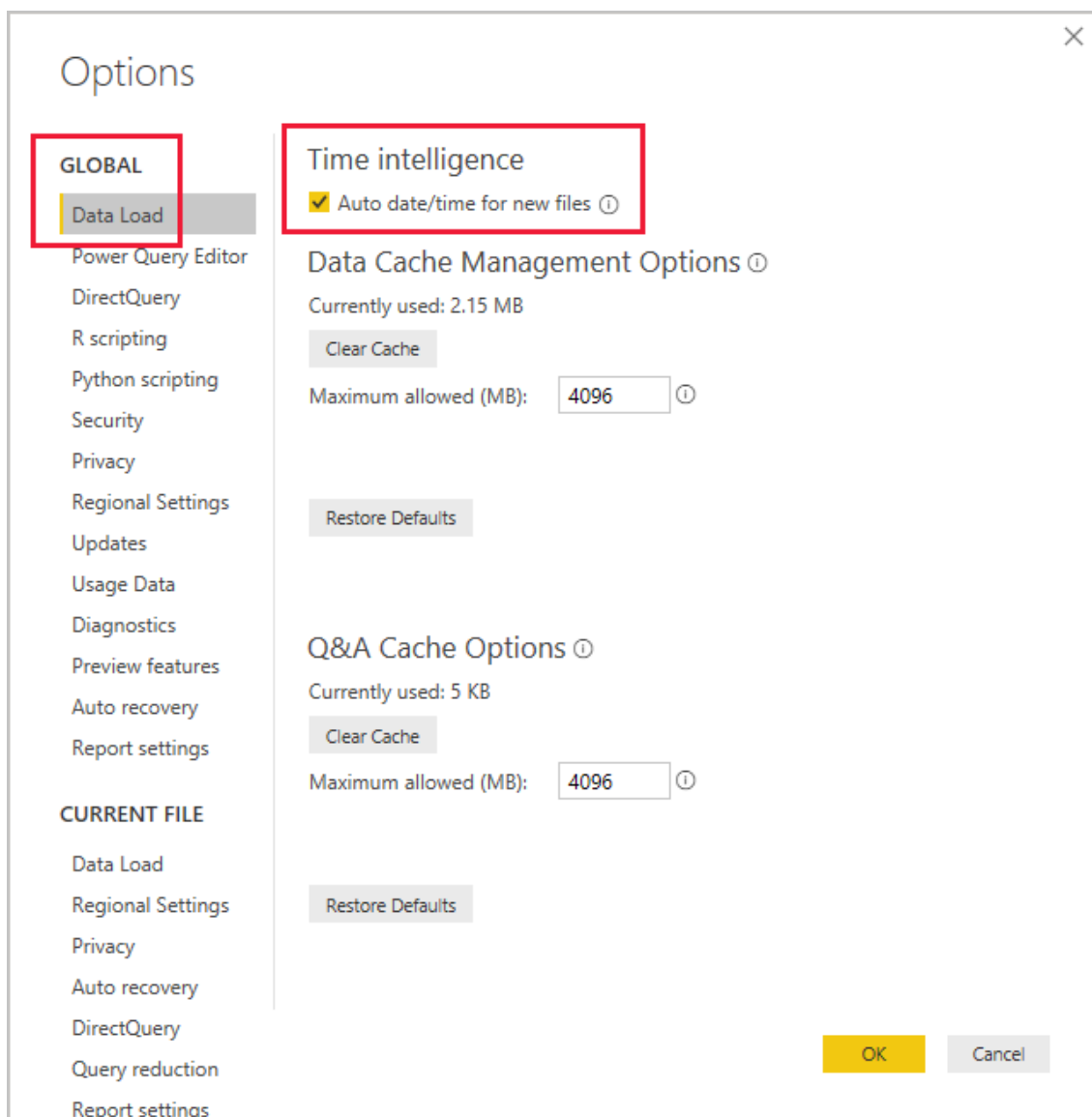
En anden emne, der skal overvejes, når performance optimeres , er **indstillingen Automatisk dato/ klokkeslæt** i Power BI Desktop. Som standard er denne funktion aktiveret globalt, Power BI Desktop hvilket betyder, **at der automatisk oprettes en skjult beregnet tabel for hver datokolonne, forudsat at visse betingelser er opfyldt**. De nye skjulte tabeller går ud over de tabeller, du allerede har i den semantiske model.

Indstillingen **Automatisk dato/klokkeslæt** giver dig mulighed for at arbejde med timeoplysninger, når du filtrerer, grupperer og ruller ned gennem kalendertidsperioder. Det anbefales, at du **kun holder indstillingen Automatisk dato/ klokkeslæt** aktiveret, når du arbejder med kalendertidsperioder, og når du har forenklede modelkrav i forhold til tid.

Hvis datakilden allerede definerer en **datodimensionstabel**, skal den pågældende tabel bruges til ensartet definition af tid i organisationen, **og du skal deaktivere den globale indstilling for automatisk dato/ klokkeslæt**. Deaktivering af denne indstilling kan reducere størrelsen på den semantiske model og reducere opdateringstiden.

Du kan aktivere/deaktivere **denne indstilling for Automatisk dato/** Power BI Desktop klokkeslæt globalt, så den gælder for alle dine filer, eller du kan aktivere/deaktivere indstillingen for den aktuelle fil, så den kun gælder for den enkelte fil.

Hvis du vil aktivere/deaktivere **denne indstilling for Automatisk dato/ klokkeslæt**, >> skal du gå til **Indstillinger for filindstillinger og derefter vælge enten siden Global eller Aktuel** fil. Vælg Dataload på begge sider , og marker **eller fjern derefter markeringen i afkrydsningsfeltet i sektionen Time Intelligence** efter behov.



Du kan finde en oversigt over og en generel introduktion **til funktionen Automatisk dato/ klokkeslæt** under [Anvend automatisk dato/klokkeslæt i Power BI Desktop](#).

Næste undermodul: Brug variabler til at forbedre ydeevnen og fejlfinding

[Tidligere](#)

Brug variabler til at forbedre ydeevnen og fejlfinding

Fuldført100 XP

- 6 minutter

Du kan bruge variabler i dine DAX-formler som en hjælp til at skrive mindre komplekse og mere effektive beregninger. Variabler bruges ikke nok af udviklere, der begynder at bruge Power BI Desktop, men de er effektive, og du bør bruge dem som standard, når du opretter målinger.

Nogle udtryk omfatter brugen af mange indlejrede funktioner og genbrug af logik for udtryk. Disse udtryk tager længere tid at behandle og er svære at læse og derfor at foretage fejlfinding af. Hvis du bruger variabler, kan du spare tid på behandling af forespørgsler. Denne ændring er et skridt i den rigtige retning mod optimering af ydeevnen for en semantisk model.

Brugen af variabler i din semantiske model giver følgende fordele:

- **Forbedret ydeevne** – Variabler kan udføre målinger på en mere effektiv måde, fordi de fjerner behovet for, at Power BI evaluerer det samme udtryk flere gange. Du kan opnå de samme resultater i en forespørgsel på halvdelen af den oprindelige behandlingstid.
- **Forbedret læsbarhed** – Variabler har korte navne, der beskriver sig selv, og bruges i stedet for et tvetydigt udtryk med flere ord. Det kan være, at du synes, at det er nemmere at læse og forstå formler, når der bruges variabler.
- **Forenklet fejlfinding** – Du kan bruge variabler til at foretage fejlfinding af en formel og teste udtryk, hvilket kan være nyttigt i forbindelse med fejlfinding.
- **Reduceret kompleksitet** – Variabler kræver ikke brug af DAX-funktionerne EARLIER eller EARLIEST, som er svære at forstå. Disse funktioner var påkrævet, før variablerne blev introduceret, og var skrevet i komplekse udtryk, der introducerede nye filterkontekster. Nu, hvor du kan bruge variabler i stedet for disse funktioner, kan du skrive færre komplekse formler.

Brug variabler til at forbedre ydeevnen

I følgende tabel vises en definition af en måling på to forskellige måder for at illustrere, hvordan du kan bruge en variabel til at foretage en måling på en mere effektiv måde. Bemærk, at formlen gentager udtrykket, der beregner "samme periode sidste år", men på to forskellige måder: I den første forekomst bruges den normale DAX-beregningsmetode, og i den anden bruges variabler i beregningen.

I den anden række i tabellen kan du se den forbedrede definition af målingen. I denne definition bruges VAR-nøgleordet til at introducere en variabel med navnet **SalesPriorYear**, og der bruges et udtryk til at tildele resultatet "samme periode sidste år" til denne nye variabel. Den bruger derefter variabelen to gange i DIVIDE-udtrykket.

Uden variabel

DAXKopier

Sales YoY Growth =

DIVIDE (

([Sales] - CALCULATE ([Sales], PARALLELPERIOD ('Date'[Date], -12, MONTH))),

CALCULATE ([Sales], PARALLELPERIOD ('Date'[Date], -12, MONTH))

)

Med variabel

DAXKopier

Sales YoY Growth =

VAR SalesPriorYear =

CALCULATE ([Sales], PARALLELPERIOD ('Date'[Date], -12, MONTH))

VAR SalesVariance =

DIVIDE (([Sales] - SalesPriorYear), SalesPriorYear)

RETURN

SalesVariance

I den første definition af målingen i tabellen er formelen ineffektiv, fordi den kræver, at Power BI evaluerer det samme udtryk to gange. Den anden definition er mere effektiv, fordi Power BI på grund af variablen kun skal evaluere PARALLELPERIOD-udtrykket én gang.

Hvis din semantiske model har flere forespørgsler med flere målinger, kan brugen af variabler reducere den samlede behandlingstid for forespørgsler til halvdelen og forbedre den semantiske models overordnede ydeevne. Denne løsning er desuden enkel. Forestil dig besparelserne, i takt med at formlerne bliver mere komplicerede, f.eks. når du arbejder med procentdele og kører totaler.

Brug variabler til at forbedre læsbarheden

Udover at forbedre ydeevnen vil du måske bemærke, hvordan brugen af variabler gør det nemmere at læse koden.

Når du bruger variabler, er det bedste praksis at bruge beskrivende navne til variablerne. I forrige eksempel kaldes variablen **SalesPriorYear**, hvilket tydeligt angiver, hvad variablen beregner. Forestil dig resultatet af at bruge en variabel med navnet **X**, **temp** eller **variable1**. Formålet med denne variabel ville ikke være tydelig for alle.

Brug af tydelige, kortfattede, betydningsfulde navne vil hjælpe med at gøre det nemmere at forstå, hvad du forsøger at beregne, og det vil være meget nemmere for andre udviklere at vedligeholde rapporten i fremtiden.

Brug variabler til at foretage fejlfinding af flere trin

Du kan bruge variabler som en hjælp til at foretage fejlfinding af en formel og identificere, hvad problemet er. Variabler hjælper med at forenkle fejlfinding af din DAX-beregning ved at evaluere hver variabel separat og kalde dem igen efter udtrykket RETURN.

I følgende eksempel tester du et udtryk, der er tildelt en variabel. Hvis du vil foretage fejlfinding, skal du midlertidigt omskrive returneringsudtrykket for at skrive til variablen. Målingsdefinitionen returnerer kun variablen **SalesPriorYear**, fordi det er det, der kommer efter returneringsudtrykket.

DAXKopier

Sales YoY Growth % =

VAR SalesPriorYear = CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))

```
VAR SalesPriorYear% = DIVIDE((Sales - SalesPriorYear), SalesPriorYear)
```

```
RETURN SalesPriorYear% Sales YoY Growth %
```

Sales YoY Growth % =

```
VAR SalesPriorYear = CALCULATE(Sales[SalesAmount]; PARALLELPERIOD('Date'[Date]; -12;MONTH))
```

```
VAR SalesPriorYearPct = DIVIDE((Sales[SalesAmount] - SalesPriorYear); SalesPriorYear)
```

```
RETURN SalesPriorYearPct :giver fejl med Sales[SalesAmount] ?!
```

Return-udtrykket viser kun værdien **SalesPriorYear%**. Denne teknik giver dig mulighed for at genindlæse udtrykket, når du har fuldført fejlfindingen. Den gør det også nemmere at forstå beregninger pga. reduceret kompleksitet af DAX-koden.

Næste undermodul: Reducer kardinalitet

[Tidligere](#)

Reducer kardinalitet

Fuldført 100 XP

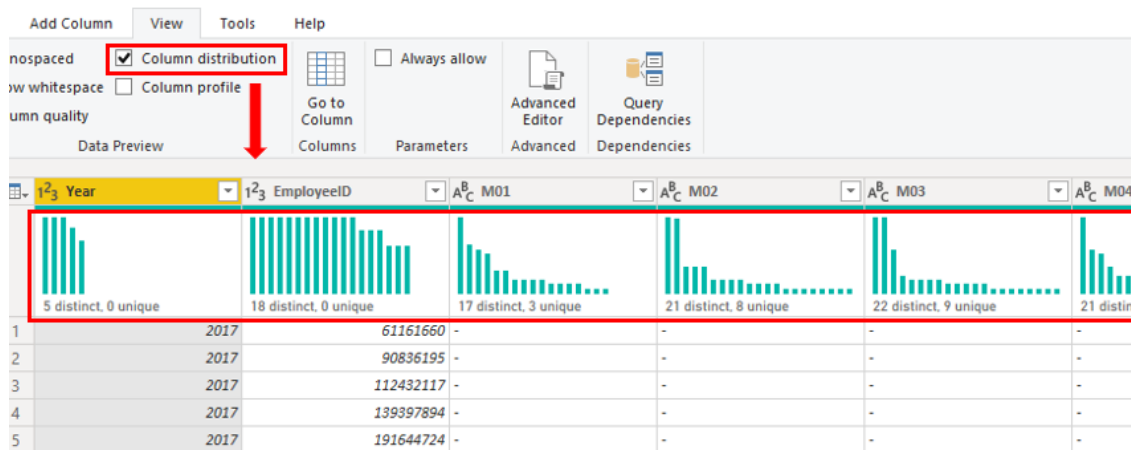
- 7 minutter

Kardinalitet er et begreb, der bruges til at beskrive, hvor unikke værdierne i en kolonne er. Kardinalitet bruges også i forbindelse med relationer mellem to tabeller, hvor det beskriver retningen af relationen.

Identificer kardinalitetsniveauer i kolonner

Tidligere når du brugte Power Query-editor til at analysere metadataene, blev der ved hjælp af indstillingen **Kolonnedistribution** under fanen **Vis** vist statistik om, hvor mange særskilte og unikke elementer der var i hver kolonne i dataene.

- **Antal særskilte værdier** – Det samlede antal forskellige værdier, der blev fundet i en given kolonne.
- **Antal unikke værdier** – Det samlede antal værdier, der kun forekommer én gang i en given kolonne.



En kolonne, der har mange gentagne værdier i sit område (entydigt antal er lavt), vil have et lavt kardinalitetsniveau. **Omvendt vil en kolonne, der indeholder mange unikke værdier (antallet af unikke værdier er højt), vil have et højt kardinalitetsniveau.**

Lavere kardinalitet fører til mere optimeret ydeevne, så du skal muligvis reducere antallet af kolonner med høj kardinalitet i din semantiske model.

Reducer relationens kardinalitet

Når du importerer flere tabeller, skal du måske udføre en analyse ved hjælp af dataene fra alle disse tabeller. Det er nødvendigt at have relationer mellem disse tabeller for at kunne beregne resultaterne nøjagtigt og vise de korrekte oplysninger i dine rapporter. Power BI Desktop hjælper med at gøre det nemmere at oprette disse relationer. Faktisk skal du i de fleste tilfælde ikke gøre noget; funktionen til automatisk registrering gør det for dig. Men du har måske indimellem brug for selv at oprette relationer eller foretage ændringer af en relation. Uanset hvad er det vigtigt at forstå relationer i Power BI Desktop, og hvordan du opretter og redigerer dem.

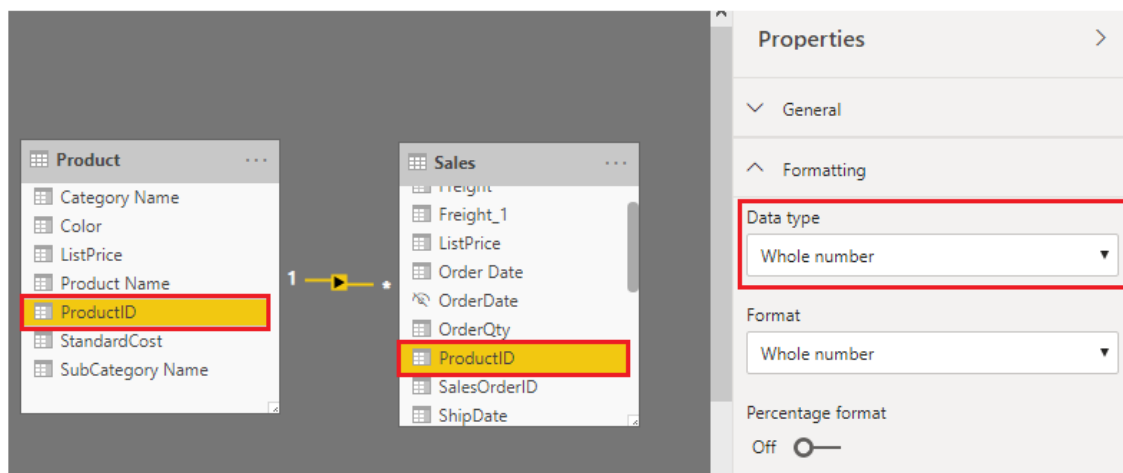
Når du opretter eller redigerer en relation, kan du konfigurere yderligere indstillinger. Power BI Desktop konfigurerer som standard yderligere indstillinger automatisk baseret på et bedste bud, hvilket kan være forskelligt for hver relation, afhængigt af dataene i kolonnerne.

Relationerne kan have forskellig kardinalitet. Kardinalitet er retningen af relation, og hver modelrelation skal have en kardinalitetstype defineret. Indstillingerne for kardinalitet i Power BI er:

- **Mange til en (*:1)** – Denne relation er den mest almindelige standardtype. Det betyder, at kolonnen i én tabel kan have mere end én forekomst af en værdi, og den anden relaterede tabel, som ofte er opslagstabellen, har kun én forekomst af en værdi.
- **En til en (1:1)** – I denne relationstype har kolonnen i én tabel kun én forekomst af en bestemt værdi, og den anden relaterede tabel har kun én forekomst af en bestemt værdi.
- **En til mange (1:*)** – I denne relationstype har kolonnen i én tabel kun én forekomst af en bestemt værdi, og den anden relaterede tabel kan have mere end én forekomst af en værdi.
- **Mange til mange (:)** – Med sammensatte modeller kan du etablere en mange til mange-relation mellem tabeller, hvilket fjerner kravene om unikke værdier i tabeller. Det fjerner også tidligere midlertidige løsninger, såsom introduktion af nye tabeller kun for at etablere relationer.

Under udvikling opretter og redigerer du relationer i din model, så når du opretter nye relationer i din model, uanset hvilken kardinalitet du har valgt, skal du altid sikre, at begge de kolonner, du bruger til at deltage i en relation, deler den samme datatype. Din model vil aldrig fungere, hvis du forsøger at oprette en relation mellem to kolonner, hvor den ene kolonne har datatypen tekst, og den anden har datatypen heltal.

I følgende eksempel har feltet **ProductID** datatypen **Heltal** i tabellerne Produkt og Salg. Kolonner med datatypen **Heltal** fungerer bedre end kolonner med datatypen **Tekst**.



Gør ydeevnen bedre ved at reducere kardinalitetsniveauer

Power BI Desktop tilbyder forskellige teknikker, som du kan bruge til at reducere de data, der indlæses i semantiske modeller, f.eks. opsummering. Reduktion af dataene, der indlæses i din model, forbedrer relationernes kardinalitet i rapporten. Derfor er det vigtigt, at du bestræber dig på at minimere de data, der skal indlæses i dine modeller. Dette gælder især for store modeller, eller modeller, som du forventer, vil vokse og blive store med tiden.

Den mest effektive teknik til at reducere størrelsen af en model er nok at bruge en oversigtstabel fra datakilden. En detaljeret tabel kan indeholde alle transaktioner, hvorimod en oversigtstabel kan indeholde én post pr. dag pr. uge eller pr. måned. Det kan f.eks. være gennemsnittet af alle transaktioner pr. dag.

I en kildetabel med salgsfakta kunne der f.eks. gemmes én række for hver ordrelinje. Du kunne reducere dataene markant ved at opsummere alle salgsmetrikker, hvis du grupperer efter dato, kunde og produkt, og hvis der ikke er behov for individuelle transaktionsdetaljer.

Overvej derefter, at der kan opnås en endnu større reduktion af data ved at gruppere efter dato på månedsniveau. Derved kunne modellens størrelsen blive reduceret med op til 99 %, men det er ikke længere muligt at rapportere på dagsniveau eller niveauet for individuelle ordrer. Det er altid nødvendigt med en afvejning i forhold til detaljeniveauet af dataene, når man skal beslutte, om man vil opsummere data af faktatypen. En ulempe er, at du kan miste muligheden for at analysere data, fordi detaljerne ikke længere findes. Denne afvejning kan afhjælpes ved hjælp af et design med en blandet model.

design med en blandet model:

I Power BI Desktop resulterer et design af typen Blandet tilstand i en sammensat model. Det giver dig i bund og grund mulighed for at fastlægge en lagertilstand for hver tabel. Hver tabel kan derfor have egenskaben **Lagertilstand** angivet til **Import** eller **DirectQuery**.

En effektiv teknik til at reducere modellens størrelse er at angive egenskaben **Lagertilstand** for større tabeller af faktatypen til **DirectQuery**. Denne designtilgang fungerer godt sammen med de teknikker, der bruges til at opsummere dataene. De opsummerede salgsdata kunne f.eks. bruges til at opnå rapportering af "oversigter" med høj ydeevne. **Der kan oprettes en side med detaljeadgang for at vise detaljeret salg for en specifik (og smal) filterkontekst, hvor alle salgsordrer vises i kontekst.** Siden med detaljeadgang ville indeholde visualiseringer baseret på en DirectQuery-tabel for at hente dataene for salgsordrerne (detaljer om salgsordre).

Du kan finde flere oplysninger i [Teknikker til reduktion af data til importudformning](#).

Næste undermodul: Optimere DirectQuery-modeller med lagring på tabelniveau

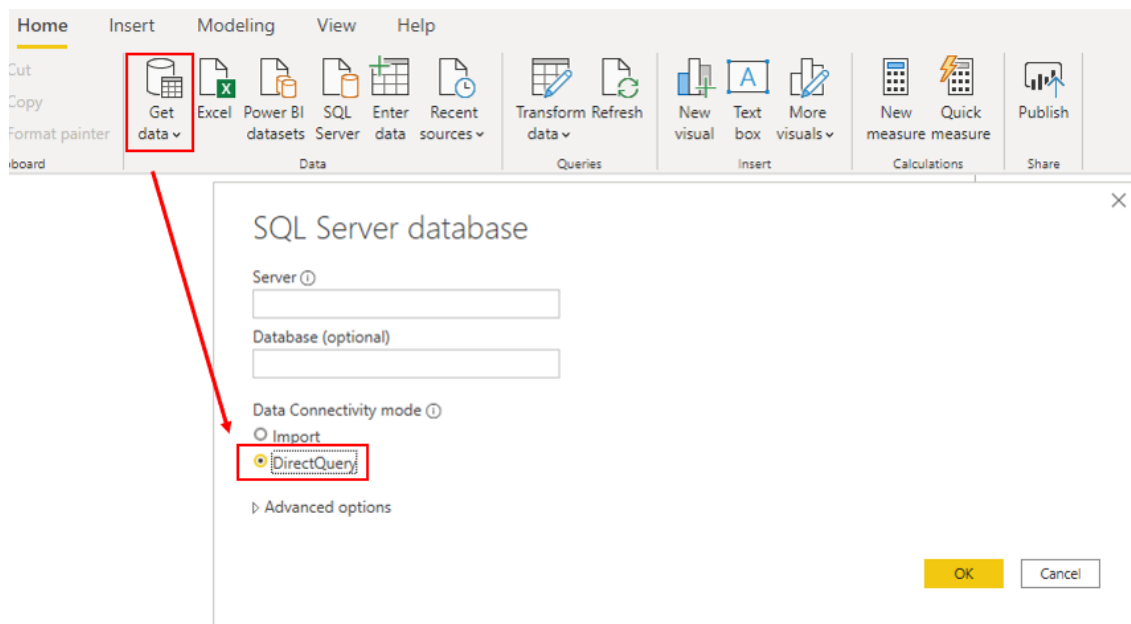
[Tidligere](#)

Optimere DirectQuery-modeller med lagring på tabelniveau

Fuldført 100 XP

- 18 minutter

DirectQuery er en måde at hente data på Power BI Desktop. DirectQuery-metoden involverer direkte forbindelse til data i dets kildelager inde fra Power BI Desktop. Den er et alternativ til at importere data til Power BI Desktop.



Når du bruger DirectQuery-metoden, afhænger den overordnede brugeroplevelse af den underliggende datakildes ydeevne meget. **Hvis forespørgslerne svar langsommere, vil det medføre en negativ brugeroplevelse, og forespørgsler kan give time out i scenarierne for antallet af forespørgsler.** Desuden vil antallet af brugere, der åbner rapporter på én gang, have indflydelse på den belastning, der er placeret på datakilden. Hvis rapporten f.eks. indeholder 20 visualiseringer, og 10 personer bruger rapporten, findes der 200 forespørgsler eller flere i datakilden, fordi hver visuel grafik vil udstede en eller flere forespørgsler.

Din models ydeevne Power BI vil muligvis ikke kun påvirke den underliggende datakildes performance, men også af andre ukontrollerbare faktorer, som f.eks.:

- Netværks latenstid. hurtigere netværk returnerer data hurtigere.
- Datakildeens serverydeevne, og hvor mange andre arbejdsbyrder der er på den pågældende server. Overvej f.eks., at en serveropdatering finder sted, mens flere hundrede personer bruger samme server af forskellige årsager.

Derfor udgør brug af DirectQuery en risiko for kvaliteten af modellens ydeevne. Hvis du vil optimere ydeevnen i denne situation, skal du have kontrol over eller have adgang til kildedatabasen.

Yderligere oplysninger finder du i [DirectQuery-modelvejledningen Power BI Desktop](#).

Konsekvenser af at bruge DirectQuery

Det er en god ide at importere data Power BI Desktop til, men organisationen skal muligvis bruge tilstanden DirectQuery-dataforbindelser af en af følgende årsager (fordele ved DirectQuery):

- Det er passende, hvis data ændres ofte og tæt på realtidsrapportering.
- Den kan håndtere store data uden at skulle aggregere på forhånd.
- Den anvender begrænsninger for adgang til data for at overholde de lovgivningsmæssige krav.
- Den kan bruges med en flerdimensionel datakilde, der indeholder målinger, som f.eks. SAP Business Warehouse . BW.

Hvis din organisation har brug for at bruge DirectQuery, skal du klart forstå, hvordan det er, inden for Power BI Desktop og være opmærksom på begrænsningerne for organisationen. Derefter er du i stand til at gøre noget for at optimere DirectQuery-modellen så meget som muligt.

DirectQuery-forbindelsers funktionalitet

Når du bruger DirectQuery til at oprette forbindelse til data i Power BI Desktop, fungerer denne forbindelse på følgende måde:

- Når du først bruger **funktionen Hent** data Power BI Desktop, skal du vælge kilden. Hvis du opretter forbindelse til en relationskilde, kan du vælge et sæt tabeller, og hver af dem definerer en forespørgsel, der logisk returnerer et sæt data. Hvis du vælger en multidimensionel kilde, f.eks. GIFT, kan du kun vælge kilden.
- Når du indlæser dataene, importeres der ingen data til Power BI Desktop det, men kun skemaet indlæses. Når du bygger grafik inden for Power BI Desktop, sendes forespørgsler til den underliggende kilde for at hente de nødvendige data. Den tid, det tager at opdatere det visuelle element afhænger af den underliggende datakildes ydeevne.
- Hvis der foretages ændringer i de underliggende data, afspejles de ikke omgående i de eksisterende visualiseringer Power BI pga. cachelagring. Du skal udføre en opdatering for at se disse ændringer. De nødvendige forespørgsler er til stede for de enkelte visuelle elementer, og visuals opdateres tilsvarende.
- Når du udgiver rapporten til tjenesten Power BI , vil den resultere i en semantisk model i Power BI brug på samme måde som til import. Der er dog ikke inkluderet data med den semantiske model.

- Når du åbner en eksisterende rapport i brug Power BI eller bygger en ny, forespørges den underliggende kilde igen for at hente de nødvendige data. Afhængigt af placeringen af den oprindelige kilde kan du være nødt til at konfigurere en data gateway for det lokale miljø.
- Du kan pinde visualiseringer eller hele rapportsider som dashboard-felter. Felterne opdateres automatisk i en plan, f.eks. hver time. Du kan styre frekvensen af denne opdatering, så den opfylder dine behov. Når du åbner et dashboard, afspejler felterne dataene på tidspunktet for den seneste opdatering og omfatter muligvis ikke de seneste ændringer, der er foretaget i den underliggende datakilde. Du kan altid opdatere et åbent dashboard for at sikre, at det er opdateret.

Begrænsninger for DirectQuery-forbindelser

Brugen af DirectQuery kan have negative konsekvenser. Begrænsningerne varierer, afhængigt af den bestemte datakilde, der bruges. Der skal tages højde for følgende punkter:

- **Performance** – Som det tidligere er beskrevet, afhænger den overordnede brugeroplevelse af den underliggende datakildes ydeevne meget.
- **Sikkerhed** – Hvis du bruger flere datakilder i en DirectQuery-model, er det vigtigt at forstå, hvordan data bevæger sig mellem de underliggende datakilder og de tilknyttede sikkerhedsmæssige konsekvenser. Du skal også identificere, om der gælder sikkerhedsregler for dataene i den underliggende kilde, da Power BI alle brugere kan se disse data.
- **Data transformation** – Sammenlignet med importerede data har data, der stammer fra DirectQuery, begrænsninger med hensyn til anvendelse af teknikker af data transformation i Power Query Editor. Hvis du f.eks. opretter forbindelse til en OLAP kilde, f.eks. OPLYSNINGER BW, kan du ikke foretage nogen transformationer overhovedet. Hele den eksterne model tages fra datakilden. Hvis du vil foretage transformationer til dataene, skal du gøre det i den underliggende datakilde.
- **Modellering** – Nogle af de modelleringsfunktioner, du har med importerede data, er ikke tilgængelige, eller de er begrænsede, når du bruger DirectQuery.
- **Rapportering** – Næsten alle de rapporteringsfunktioner, du har med importerede data, understøttes også for DirectQuery-modeller, forudsat at den underliggende kilde har et passende ydeevneniveau. Når rapporten publiceres i tjenesten Power BI, vil funktionerne Quick Insights og Q&A dog ikke blive understøttet. Brugen af funktionen Gå på opdagelse i Excel vil sandsynligvis resultere i en dårligere ydeevne.

Du kan finde mere detaljerede oplysninger om begrænsningerne ved brug af DirectQuery i [Implications of using DirectQuery](#).

Nu, hvor du har en kort forståelse af, hvordan DirectQuery fungerer, og de begrænsninger, den medfører, kan du gøre noget for at forbedre ydeevnen.

Optimere ydeevne

Vi fortsætter med scenariet Debitwind Traders under gennemgangen af den semantiske model, og du opdager, at forespørgslen brugte DirectQuery Power BI Desktop til at oprette forbindelse til kildedataene. Denne brug af DirectQuery er årsagen til, at brugerne arbejder med dårlig rapportydeevne. Det tager for lang tid at indlæse siderne i rapporten, og tabellerne opdaterer ikke hurtigt nok, når der foretages bestemte valg. Du skal gøre en handling for at optimere DirectQuery-modellens ydeevne.

Du kan undersøge de forespørgsler, der sendes til den underliggende kilde, og prøve at identificere årsagen til den dårlige forespørgselsydeevne. Du kan derefter foretage ændringer i og Power BI Desktop den underliggende datakilde for at optimere den overordnede performance.

Optimer data i Power BI Desktop

Når du har optimeret datakilden så meget som muligt, Power BI Desktop **kan du gøre yderligere ved at bruge performanceanalyser**, hvor du kan anvende forespørgselsforespørgsler til at validere forespørgselsplaner.

Du kan analysere varigheden af forespørgsler, der sendes til den underliggende kilde, for at identificere de forespørgsler, der det tager lang tid at indlæse. Du kan med andre ord identificere, hvor flaskehalse findes.

Du behøver ikke at bruge en særlig indfaldsvinkel, når du optimerer en DirectQuery-model. du kan anvende samme optimeringsteknik, som du brugte på de importerede data til at tilpasse dataene fra DirectQuery-kilden. Du kan f.eks. reducere antallet af visuelle elementer på rapport siden eller reducere antallet af felter, der bruges visuelt. Du kan også fjerne unødvendige kolonner og rækker.

Du kan finde en mere detaljeret vejledning til, hvordan du optimerer en DirectQuery-forespørgsel, i: [Vejledning i DirectQuery-modellen Power BI Desktop i og vejledning til brug af DirectQuery](#).

Optimer den underliggende datakilde (tilknyttet database)

Dit første stop er datakilden. Du skal tilpasse kildedatabasen så meget som muligt, da alt, hvad du gør for at forbedre ydeevnen for kildedatabasen, vil til gengæld forbedre Power BI DirectQuery. De handlinger, du kan udføre i databasen, gør de bedste.

Overvej, hvordan følgende standarddatabasemetoder anvendes i de fleste situationer:

- Undgå at bruge komplekse beregnede kolonner, da beregningsudtrykket integreres i kildeforespørgsler. Det er mere effektivt at skubbe udtrykket tilbage til kilden, da det undgår et push-down. Du kan også overveje at føje kolonnekolonner til surrogatnøgler til tabeller af typen Dimension.
- Gennemse indekserne, og kontroller, at det aktuelle indeks er korrekt. Hvis du har brug for at oprette nye indekser, skal du sikre dig, at de er korrekte.

Se vejledningsdokumenterne i datakilden, og implementer deres performanceanbefalinger.

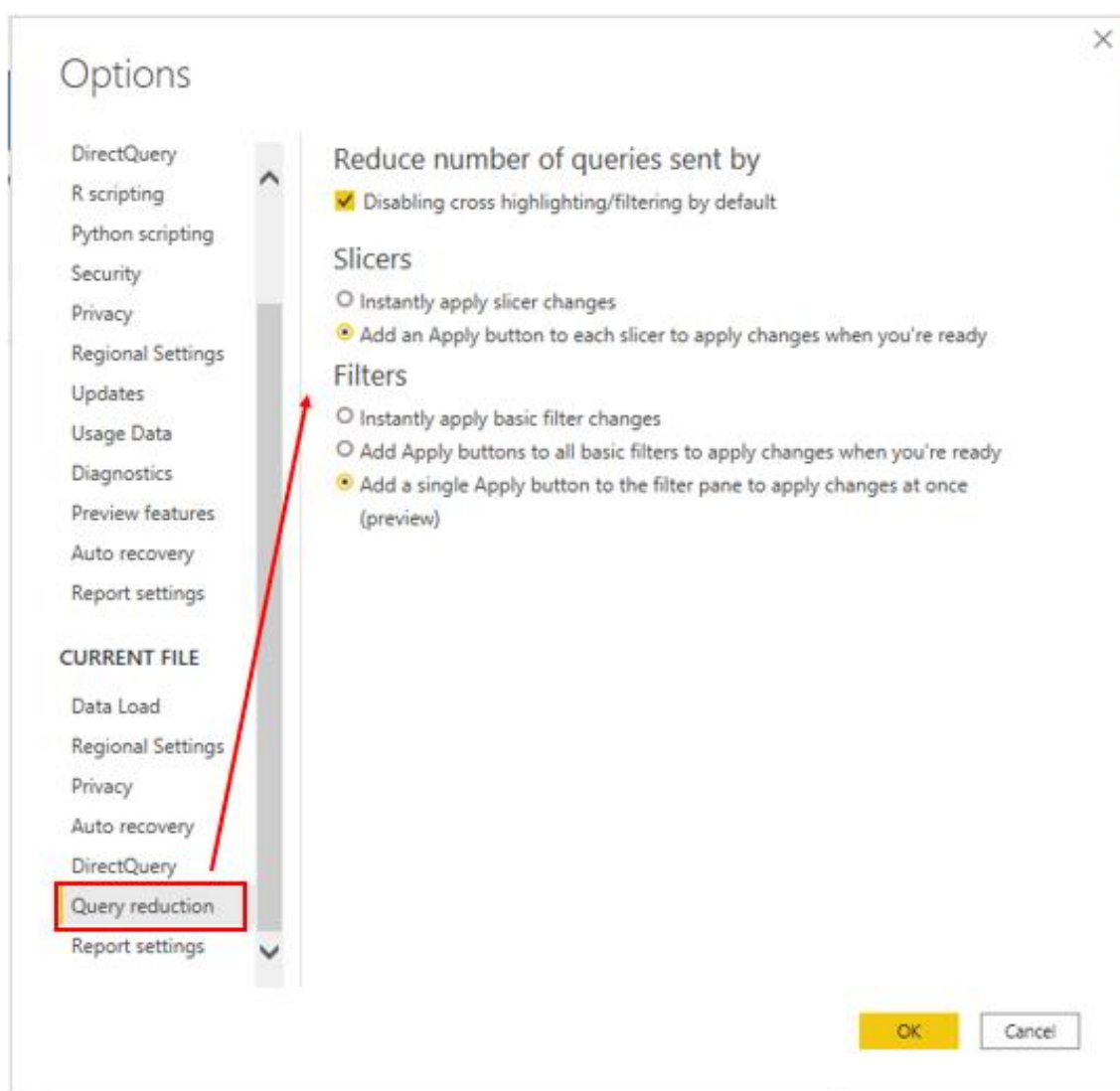
Tilpasse indstillingerne for forespørgselsreduktion

Power BI Desktop giver dig mulighed for at sende færre forespørgsler og deaktivere visse interaktioner, der vil resultere i en dårlig erfaring, hvis det tager lang tid at køre de forespørgsler, der er resultatet. Hvis du anvender disse indstillinger, forhindres det, at forespørgsler fortløbende søger i datakilden, hvilket bør forbedre ydeevnen.

I dette eksempel redigerer du standardindstillingerne for at anvende de tilgængelige indstillinger for datareduktion på modellen. Du kan få adgang til indstillingerne ved at vælge **>Indstillinger for>** filindstillinger, rulle ned på siden og derefter vælge indstillingen **For forespørgselsreduktion**.

Der er følgende indstillinger for reduktion i forespørgsler:

- **Reducer antallet af forespørgsler, der sendes af** – Som standard påvirker alle visuelle elementer hinanden. Hvis dette afkrydsningsfelt markeres, deaktiveres standardinteraktionen. Du kan vælge, hvilke visuelle elementer der skal bruges sammen, ved at bruge funktionen **Rediger interaktioner**.
- **Udstedere** – Som standard er indstillingen **Anvend ændringer afvaluta** valgt. Hvis du vil tvinge rapportbrugerne til manuelt at anvende ændringer i personlæseren, skal du vælge knappen Tilføj en anvend-knap på hver enkelt person, **så ændringerne kan anvendes, når du er klar til det**.
- **Filtre** – Som standard er indstillingen **Anvend grundlæggende filterændringer** valgt. Hvis du vil tvinge rapportbrugerne til at anvende filterændringer manuelt, skal du vælge en af de alternative indstillinger:
 - **Føj en anvend knap til alle grundlæggende filtre for at anvende ændringer, når du er klar**
 - **Føj en enkelt knap til filtrerruden for at anvende ændringer på én gang (forhåndsversion)**



Næste undermodul:

Opret og administrer data-sammenlægninger, til grovere granularitet

Tidligere

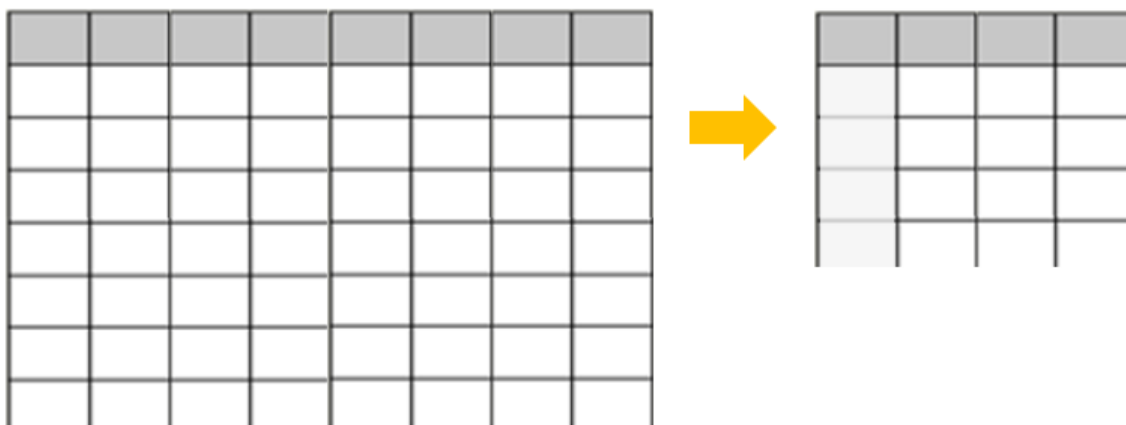
Opret og administrer data-sammenlægninger

Fuldført 100 XP

- 7 minutter

Når du samler data, opsummerer du disse data og viser dem med en lavere detaljeringsgrad (på et højere niveau). Du kan f.eks. opsummere alle salgsdata og gruppere dem efter dato, kunde, produkt osv.

Sammenlægningsprocessen reducerer tabelstørrelserne i den semantiske model, så du kan fokusere på vigtige data og hjælpe med at forbedre forespørgselydeevnen.



Din organisation kan beslutte at bruge sammenlægninger i deres semantiske modeller af følgende årsager:

- Hvis du arbejder med en stor mængde data (big data), giver sammenlægningerne en bedre ydeevne af forespørgsler og hjælper dig med at analysere og vise indsigt i disse mange data. Samlede data cachelagres og bruger derfor en brøkdel af de ressourcer, der kræves til detaljerede data.
- Hvis du oplever en langsom opdatering, hjælper sammenlægninger dig med at sætte fart på opdateringsprocessen. Den mindre cachestørrelse reducerer opdateringstiden, så dataene kommer hurtigere ud til brugerne. I stedet for at opdatere det, der kan være millioner af rækker, opdaterer du i stedet en mindre mængde data.
- Hvis du har en stor semantisk model, kan sammenlægninger hjælpe dig med at reducere og bevare størrelsen på din model.
- Hvis du forventer, at din semantiske model bliver større i fremtiden, kan du bruge sammenlægninger som et proaktivt skridt i retning af fremtidig korrektur af din semantiske model ved at mindske risikoen for problemer med ydeevne og opdatering og overordnede forespørgselsproblemer.

Hvis du fortsætter med Tailwind Traders-scenariet, har du taget flere trin for at optimere ydeevnen af den semantiske model, men it-teamet har meddelt dig, at filstørrelsen stadig er for stor. Filstørrelsen er i øjeblikket 1 gigabyte (GB), så du skal reducere den til omkring 50 megabytes (MB). Under gennemgangen af ydeevnen identificerede du, at den forrige udvikler ikke brugte sammenlægninger i den semantiske model,

så du nu vil oprette nogle sammenlægninger til salgsdataene for at reducere filstørrelsen og optimere ydeevnen yderligere.

Opret sammenlægninger

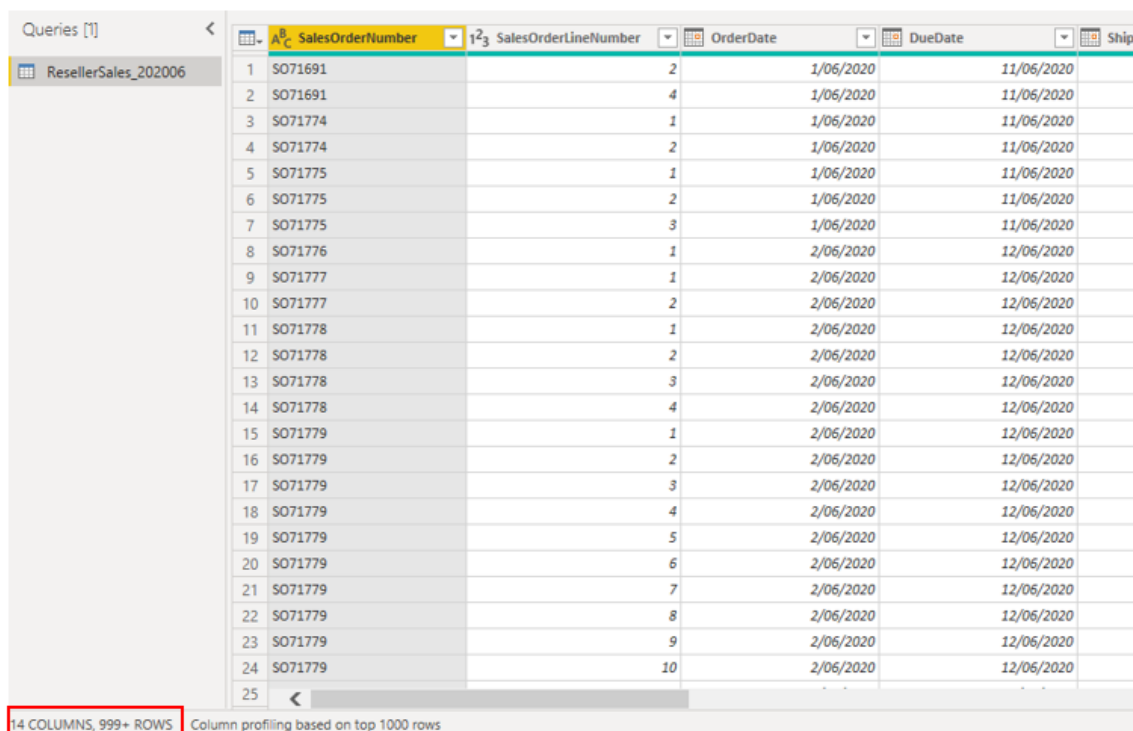
Før du begynder at oprette sammenlægninger, skal du beslutte dig for, med hvilken detaljeringsgrad (niveau) du vil oprette dem. **I dette eksempel vil du samle salgsdataene på dagsniveau.**

Når du har besluttet dig for detaljeringsgraden, er det næste trin at beslutte, hvordan du vil oprette sammenlægningerne.

Du kan oprette sammenlægninger på forskellige måder, og hver metode giver dig de samme resultater, f.eks.:

- Hvis du har adgang til databasen, kan du oprette en tabel med sammenlægningen og derefter importere denne tabel i Power BI Desktop.
- Hvis du har adgang til databasen, kan du oprette en visning af sammenlægningen og derefter importere denne visning i Power BI Desktop.
- I Power BI Desktop kan du bruge Power Query-editor til at oprette sammenlægningerne trinvis.

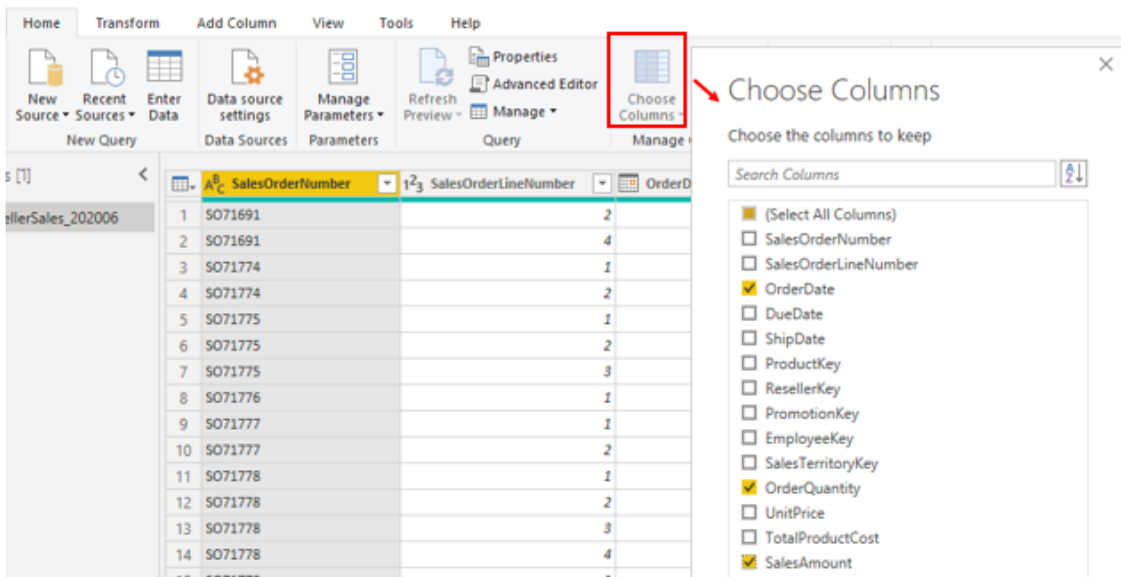
I dette eksempel åbner du en forespørgsel i Power Query-editor og bemærker, at dataene ikke er blevet samlet. Der er over 999 rækker, som vist på følgende skærmbillede.



	1 SalesOrderNumber	2 SalesOrderLineNumber	3 OrderDate	4 DueDate	5 ShipDate
1	SO71691	2	1/06/2020	11/06/2020	
2	SO71691	4	1/06/2020	11/06/2020	
3	SO71774	1	1/06/2020	11/06/2020	
4	SO71774	2	1/06/2020	11/06/2020	
5	SO71775	1	1/06/2020	11/06/2020	
6	SO71775	2	1/06/2020	11/06/2020	
7	SO71775	3	1/06/2020	11/06/2020	
8	SO71776	1	2/06/2020	12/06/2020	
9	SO71777	1	2/06/2020	12/06/2020	
10	SO71777	2	2/06/2020	12/06/2020	
11	SO71778	1	2/06/2020	12/06/2020	
12	SO71778	2	2/06/2020	12/06/2020	
13	SO71778	3	2/06/2020	12/06/2020	
14	SO71778	4	2/06/2020	12/06/2020	
15	SO71779	1	2/06/2020	12/06/2020	
16	SO71779	2	2/06/2020	12/06/2020	
17	SO71779	3	2/06/2020	12/06/2020	
18	SO71779	4	2/06/2020	12/06/2020	
19	SO71779	5	2/06/2020	12/06/2020	
20	SO71779	6	2/06/2020	12/06/2020	
21	SO71779	7	2/06/2020	12/06/2020	
22	SO71779	8	2/06/2020	12/06/2020	
23	SO71779	9	2/06/2020	12/06/2020	
24	SO71779	10	2/06/2020	12/06/2020	
25					

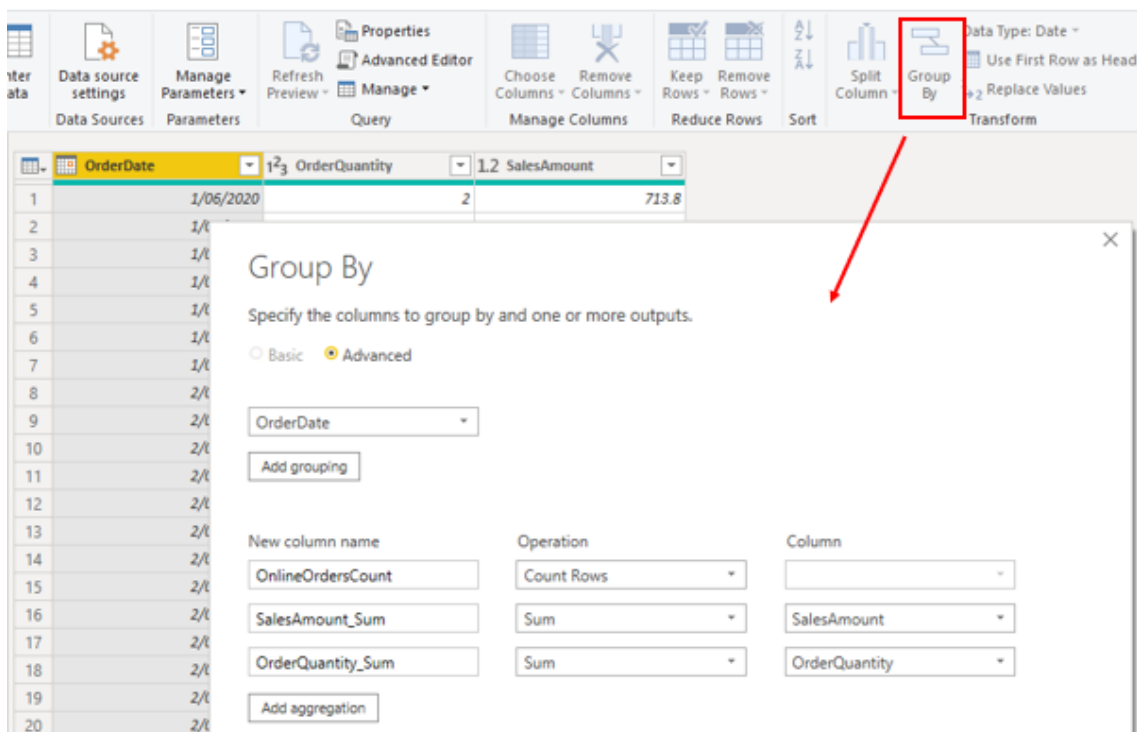
14 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows

Du vil samle dataene efter kolonnen **OrderDate** og have vist kolonnerne **OrderQuantity** og **SalesAmount**. Start med at vælge **Vælg kolonner** under fanen **Hjem**. I det viste vindue skal du vælge de kolonner, du vil have i sammenlægningen, og derefter vælge **OK**.



Når de valgte kolonner vises på siden, skal du vælge indstillingen **Gruppér efter** under fanen **Hjem**. I det viste vindue skal du vælge den kolonne, du vil gruppere efter (**OrderDate**), og angive navnet på den nye kolonne (**OnlineOrdersCount**).

Vælg indstillingen **Avanceret**, og vælg derefter knappen **Tilføj sammenlægning** for at få vist en anden kolonnerække. Angiv et navn på sammenlægningskolonnen, vælg kolonnens funktion, og vælg derefter den kolonne, som du vil linke sammenlægningen til. Gentag disse trin, indtil du har tilføjet alle sammenlægningerne, og vælg derefter **OK**.



Det kan tage et par minutter, før sammenlægningen vises, men når det sker, kan du se, hvordan dataene er blevet transformeret. Dataene samles i hver dato, og du kan se værdierne for antallet af ordrer og den respektive sum af salgsbeløbet og ordreantallet.

	OrderDate	1.2 OnlineOrdersCount	1.2 SalesAmount_Sum	1.2 OrderQuantity_Sum
1	1/06/2020	7	3221.28	14
2	2/06/2020	78	68495.62	225
3	3/06/2020	151	213860.64	738
4	4/06/2020	89	87484.01	218
5	5/06/2020	224	292106.92	978
6	6/06/2020	163	145808.57	572
7	7/06/2020	125	110115.85	557
8	8/06/2020	156	177334.37	441

Vælg knappen **Luk og Anvend** for at lukke Power Query-editor og anvende ændringerne på din semantiske model. Vend tilbage til siden **Power BI Desktop**, og vælg derefter knappen **Opdater** for at se resultaterne. Se skærmen, fordi der vises en kort meddelelse om antallet af rækker, som din semantiske model nu har. Dette antal rækker bør være markant mindre end det antal, du startede med. Du kan også se dette antal, når du åbner Power Query-editor igen, som vist på følgende skærbillede. I dette eksempel blev antallet af rækker reduceret til 30.

22	22/06/2020	55	73935.41	129
23	23/06/2020	116	191212.91	371
24	24/06/2020	20	11193.33	26
25	25/06/2020	62	65857.75	183

4 COLUMNS, 30 ROWS Column profiling based on top 1000 rows

Husk, at du startede med over 999 rækker. Brug af sammenlægning har reduceret antallet af rækker i din semantiske model markant, hvilket betyder, at Power BI har færre data at opdatere, og at din model skal fungere bedre.

Administrer sammenlægninger

Når du har oprettet sammenlægninger, kan du administrere dem i Power BI Desktop og foretage ændringer af deres funktionsmåde, hvis det er nødvendigt.

Du kan åbne vinduet **Administrer sammenlægninger** i en hvilken som helst visning i Power BI Desktop. Højreklik på tabellen i ruden **Felter**, og vælg derefter **Administrer sammenlægninger**.

For hver sammenlægningskolonne kan du vælge en indstilling på rullelisten **Opsummering** og foretage ændringer af den valgte detaljerede tabel og kolonne. Når du er færdig med at administrere sammenlægningerne, skal du vælge **Anvend alle**.

Du kan finde flere detaljerede oplysninger om, hvordan du opretter og administrerer sammenlægninger i [Brug sammenlægninger i Power BI Desktop](#).

Ind i model, husk manage key, virker:

The screenshot shows the Power BI Desktop interface. On the left, a table is displayed with columns 'Year' and 'Sum of SalesAmount_sum'. The table data is as follows:

Year	Sum of SalesAmount_sum
2017	11.928.555,52
Qtr 3	6.005.207,32
July	1.423.357,32
August	2.057.902,45
September	2.523.947,55
Qtr 4	5.923.348,19
2018	30.516.891,80
Qtr 1	5.965.011,12
Qtr 2	5.967.324,53
Total	109.809.274,20

On the right, the Fields pane is visible. It shows a list of fields on the left and a list of fields on the right. The fields on the left are 'Date - Year is (All)' and 'Sum of SalesAmount_... is (All)'. The fields on the right are 'Date', 'Year', 'Quarter', 'Month', and 'Day'. The 'Date' field is selected in the 'Rows' section. The 'Sum of SalesAmount_...' field is selected in the 'Values' section.

Næste undermodul: Test din viden

[Tidligere](#)

Test din viden

Fuldført200 XP

- 3 minutter

Besvar følgende spørgsmål for at se, hvad du har lært.

1.

Hvilke fordele får du af at analysere metadataene?

Fordelen ved at analysere metadataene er, at du tydeligt kan identificere data, der er inkompatible med din semantiske model.

Fordelen ved at analysere metadataene er, at du tydeligt kan identificere data, der er inkompatible med din semantiske model.

Fordelen ved at analysere metadataene er at sætte dig ind i dataene <forkert>

Fordelen ved at analysere metadataene er, at du tydeligt kan identificere data, der er inkompatible med din semantiske model.

Fordelen ved at analysere metadataene er at kende antallet af rækker, kolonner og tabeller, der indlæses i modellen.

2.

Hvad du kan opnå ved at fjerne unødvendige rækker og kolonner?

Det er ikke nødvendigt at slette unødvendige rækker og kolonner, og det er en god ide at bevare alle metadata.

Hvis du sletter unødvendige rækker og kolonner, reduceres den semantiske modelstørrelse, og det er en god ide kun at indlæse de nødvendige data i den semantiske model.

Hvis du sletter unødvendige rækker og kolonner, reduceres den semantiske modelstørrelse, og det er en god ide kun at indlæse de nødvendige data i den semantiske model.

Hvis du sletter unødvendige rækker og kolonner, kan det beskadige den semantiske models struktur.

3.

Hvilken af følgende sætninger om relationer er Power BI Desktop sand?

Der kan kun oprettes relationer mellem kolonner, der indeholder den samme datatype.

Ukorrekt. Dette gælder ikke, da der kan oprettes relationer mellem kolonner, der indeholder forskellige datatyper i Power BI Desktop.

Der kan kun oprettes relationer mellem tabeller, der indeholder det samme antal rækker.

Der kan oprettes relationer mellem tabeller, der indeholder forskellige typer data.

Rigtig. Der kan oprettes et arbejdsforhold, så længe der er mindst én fælles kolonne mellem dem.

Næste undermodul:

[Resumé](#)

[Tidligere](#)

Resumé

Fuldført100 XP

- 3 minutter

I dette moduls scenarie var en af organisationens Power BI Desktop semantiske modeller ineffektiv og forårsagede problemer. Brugere var utilfredse med rapportens ydeevne, og størrelsen af modellen var for stor, så det belastede organisationens ressourcer.

Du blev bedt om at gennemse den semantiske model for at identificere årsagen til problemer med ydeevnen og foretage ændringer for at optimere ydeevnen og reducere modellens størrelse.

Power BI Desktop indeholder en række værktøjer og funktioner, som du kan bruge til at analysere og optimere ydeevnen af dens semantiske modeller. Du begyndte optimeringsprocessen ved hjælp af **Effektivitetsanalyse** og andre værktøjer for at gennemgå ydeevnen af målinger, relationer og visualiseringer

og foretog derefter ændringer baseret på analyseresultaterne. Derefter anvendte du variabler for at skrive mindre komplekse og mere effektive beregninger. Du kiggede derefter nærmere på kolonnedistributionen og reducerede kardinaliteten af dine relationer. På dette tidspunkt var den semantiske model mere optimeret. Du overvejede, hvordan situationen ville være anderledes, hvis din organisation brugte en DirectQuery-model, og derefter identificerede du, hvordan du skulle optimere ydeevnen fra Power BI Desktop og kildedatabasen. Til sidst brugte du sammenlægninger til at reducere størrelsen på den semantiske model markant.

Hvis Power BI Desktop ikke giver dig mulighed for at optimere ineffektive semantiske modeller, skal du bruge meget tid på dine flere datakilder for at forbedre dataene der. Uden Effektivitetsanalyse ville du især ikke have identificeret årsagerne til problemerne med ydeevnen i dine rapporter og flaskehalsene i forespørgslerne, der skulle ryddes. Derfor ville brugerne blive frustrerede og umotiverede og måske undlade at bruge rapporterne.

Nu, hvor du har optimeret rapporten, kan brugerne tilgå de data, de har brug for hurtigere, så de er mere produktive og har større jobtilfredshed. Den reduktion, du foretog af modellens størrelse, vil lette belastningen på ressourcerne, hvilket giver organisationen en række fordele. Du har fuldført den opgave, du blev stillet.

[Brug Effektivitetsanalyse til at undersøge ydeevnen af rapportelementer](#)

[Anvend automatisk dato/klokkeslæt i Power BI Desktop](#)

[Teknikker til reduktion af data til importudformning](#)

[Hjælp til DirectQuery-model i Power BI Desktop](#)

[Brug sammenlægninger i Power BI Desktop](#)

Alle enheder er fuldført:

[Tidligere](#)

Fortsæt det fremragende arbejde



Optimer en model i forhold
til ydeevne i Power BI

Du har opnået et resultat!

Tillykke, men hvad skal du gøre nu?

Lad os først dele dit resultat

Du afsætter tid til at lære noget nyt, så lad dit netværk tage del i din sejr!



Mist ikke dit momentum, bliv ved med at lære

Nedenfor finder du anbefalet indhold, der kan hjælpe dig på vej!

<https://learn.microsoft.com/api/achievements/share/da-dk/bjarnehansen-7412/UYBKND3?sharingId=39714CB1FE13C67B>

**** Skriv DAX-formler til Power BI Desktop-modeller

- 27 min.
- Modul
- 8 Enheder

Feedback

Øvet

Appbygger

Dataanalytiker

Power BI

Microsoft Power Platform

I dette modul får du oplysninger om, hvordan du skriver DAX-formler for at oprette beregnede tabeller, beregnede kolonner og målinger, der er forskellige typer af modelberegninger. Derudover lærer du at skrive

og formatere DAX-formler, der består af udtryk, som bruger funktioner, operatorer, referencer til modelobjekter, konstanter og variabler.

Læringsmål

Når du har fuldført dette modul, vil du være i stand til følgende:

- Beskrive de forskellige DAX-beregningstyper.
- Skrive DAX-formler.
- Beskrive DAX-datatyper.
- Arbejde med DAX-funktioner.
- Bruge DAX-operatorer.
- Bruge DAX-variabler.

[Start](#)Tilføj

-
-

Forudsætninger

Erfaring med oprettelse af Microsoft Power BI Desktop-modeller og design af Power BI-rapportlayouts.

Dette modul er en del af disse læringsforløb

- [Brug DAX i Power BI Desktop](#)
- [Modeldata med Power BI](#)
- [Introduktion](#)5 min.
- [Skriv DAX-formler](#)6 min.
- [DAX-datatyper](#)2 min.
- [Arbejd med DAX-funktioner](#)3 min.
- [Brug DAX-operatorer](#)5 min.
- [Brug DAX-variabler](#)1 min.
- [Tjek din viden](#)4 min.
- [Resumé](#)1 min.

Introduktion

Fuldført100 XP

- 5 minutter

Ved hjælp af DAX (Data Analysis Expressions) kan du føje tre typer beregninger til din semantiske model:

- Beregnede tabeller
- Beregnede kolonner
- Målinger

Bemærk

DAX kan også bruges til at definere regler for sikkerhed på rækkeniveau (RLS), som er udtryk, der gennemtvinger filtre i modeltabeller. Regler anses dog ikke for at være modelberegninger, så de er ikke omfattet af dette modul. Du kan finde flere oplysninger under [Sikkerhed på rækkeniveau \(RLS\) med Power BI](#).

Beregnede tabeller

Du kan skrive en DAX-formel for at føje en beregnet tabel til din model. Formlen kan duplikere eller transformere *eksisterende modeldata* eller oprette en dataserie for at oprette en ny tabel. Beregnede tabeldata importeres altid til din model, så det øger størrelsen på modellageret og kan forlænge dataopdateringstiden.

Bemærk

En beregnet tabel kan ikke oprette forbindelse til eksterne data. Du skal bruge Power Query for at fuldføre opgaven.

Beregnede tabeller kan være nyttige i forskellige scenarier:

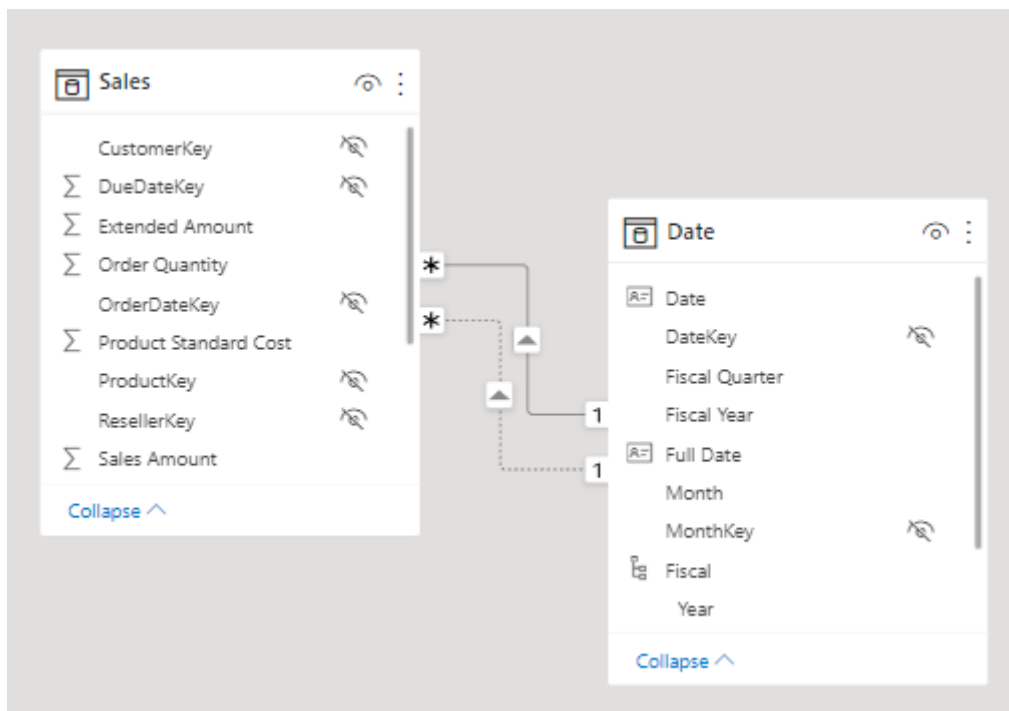
- Datotabeller
- Dimensioner med forskellige roller
- What If-analyse

Datotabeller

Datotabeller skal anvende særlige tidsfiltre, der kaldes *time intelligence*. DAX-time intelligence-funktioner fungerer kun korrekt, når der er konfigureret en datotabel. Når dine kildedata ikke indeholder en datotabel, kan du oprette en som beregnede tabeller ved hjælp [CALENDAR](#) af DAX-funktionerne eller [CALENDARAUTO](#).

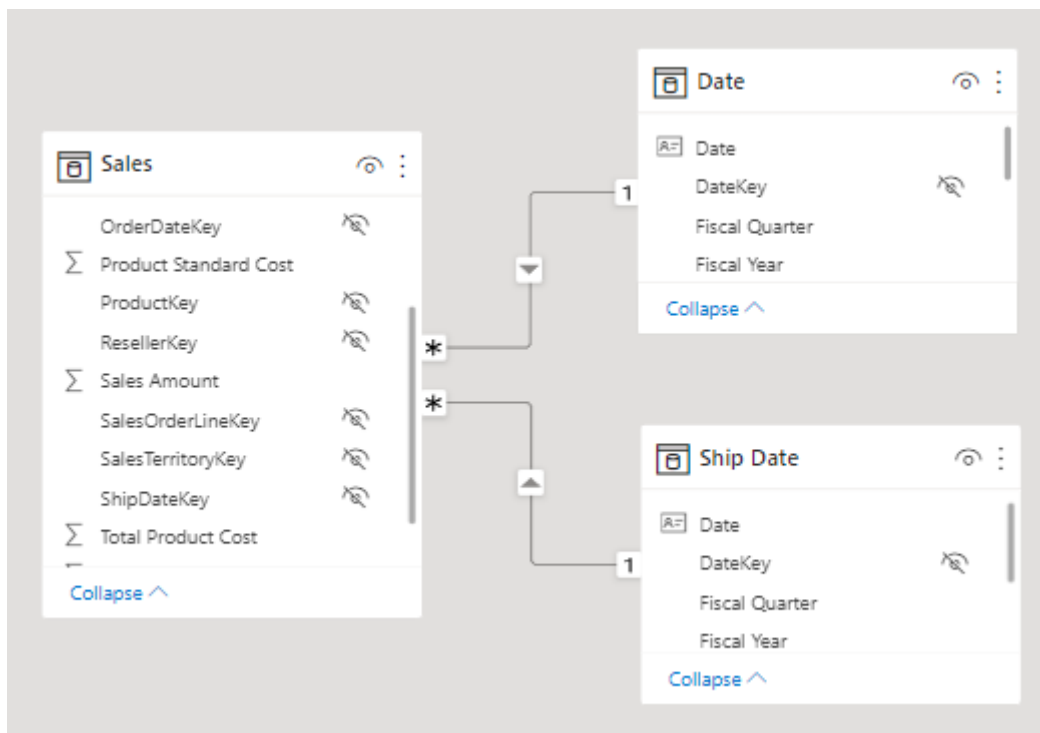
Dimensioner med forskellige roller

Når to modeltabeller har flere relationer, kan det skyldes, at din model har en dimension med forskellige roller. Hvis du f.eks. har en tabel med navnet **Sales**, der indeholder to datokolonner, **OrderDateKey** og **ShipDateKey**, er begge kolonner relateret til kolonnen **Date** i tabellen **Date**. I dette tilfælde beskrives tabellen **Date** som en dimension med forskellige roller, da den kan spille rollen *ordredato* eller *afsendelsesdato*.



Microsoft Power BI-modeller tillader kun én aktiv relation mellem tabeller, som er angivet som en ubrudt linje i modeldiagrammet. Den aktive relation bruges som standard til at overføre filtre, hvilket i dette tilfælde ville være fra tabellen **Date** til kolonnen **OrderDateKey** i tabellen **Sales** . Alle resterende relationer mellem de to tabeller er inaktive. I et modeldiagram er relationerne repræsenteret som stiplede linjer. Inaktive relationer bruges kun, når der udtrykkeligt anmodes om dem i en beregnet formel ved hjælp [USERELATIONSHIP](#) af DAX-funktionen.

Måske kan et bedre modeldesign have to datotabeller, hver med en aktiv relation til tabellen **Sales** . Rapportbrugere kan derfor filtrere efter ordredato eller afsendelsesdato eller begge dele på samme tid. En beregnet tabel kan duplikere dataene i tabellen **Date** for at oprette tabellen **Ship Date** .



What If-analyse

Power BI Desktop understøtter en funktion, der kaldes [What If-parametre](#). Når du opretter en What If-parameter, føjes der automatisk en beregnet tabel til din model.

What If-parametre gør det muligt for brugere af rapporten at vælge eller filtrere efter værdier, der er gemt i den beregnede tabel. Formler for målinger kan bruge den eller de valgte værdier på en meningsfuld måde.

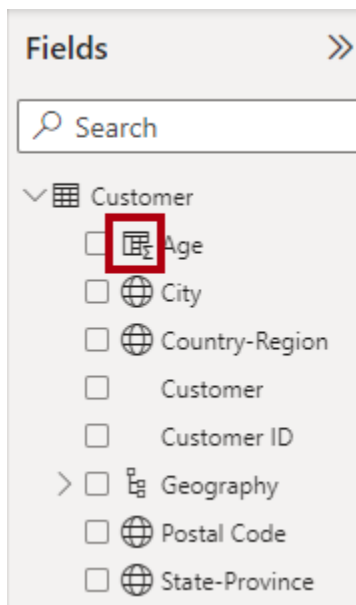
En **What If-** parameter kan f. eks. tillade, at rapportbrugeren vælger en hypotetisk valutakurs, og en måling kan dividere indtægtsværdier (i en lokal valuta) med den valgte sats.

Beregne What If-tabeller ikke er relateret til andre modeltabeller, fordi de ikke bruges til at overføre filtre. Derfor kaldes de nogle gange for *frakoblede tabeller*.

Beregne kolonner

Du kan skrive en DAX-formel for at føje en beregnet kolonne til en vilkårlig tabel i din model. Formlen evalueres for hver tabelrække, og den returnerer en enkelt værdi. Når formelen føjes til en tabel med lagringstilstanden Import, evalueres den, når den semantiske model opdateres, og den øger din models lagerstørrelse. Når formlen føjes til en tabel med lagringstilstanden DirectQuery, evalueres formelen af den underliggende kildedatabase, når der sendes en forespørgsel til tabellen.

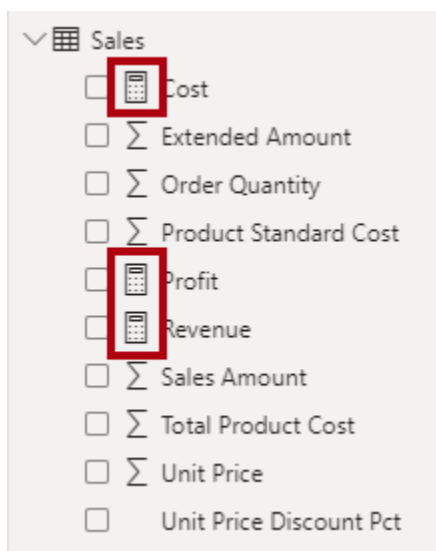
Beregne kolonner udvides med et særligt ikon i ruden **Felter**. I følgende eksempel vises en enkelt beregnet kolonne i tabellen **Kunde** med navnet **Alder**.



Målinger

Du kan skrive en DAX-formel for at føje en måling til en vilkårlig tabel i din model. Formlen bruges til opsummering af modeldata. Formlen skal returnere en enkelt værdi på samme måde som en beregnet kolonne. I modsætning til beregnede kolonner, der evalueres på dataopdateringstidspunktet, evalueres målinger på forespørgselstidspunktet. Deres resultater gemmes aldrig i modellen.

Målinger vises med lommeregnerikonet i ruden **Felter**. I følgende eksempel vises tre målinger i tabellen **Sales : Cost, Profit og Revenue**.



Sommetider kan målinger beskrives som *eksplicitte målinger*. Eksplicitte målinger er nærmere betegnet modelberegninger, der er skrevet i DAX, og kaldes normalt blot *målinger*. Ja, begrebet *implicitte målinger* findes også. Implicitte målinger er kolonner, der kan opsummeres i visuals på forenklede måder, f. eks. count, sum, minimum, maximum osv. Du kan identificere implicitte målinger i ruden **Felter**, fordi de vises med sigmasymbolet (Σ).

Bemærk

Enhver kolonne kan opsummeres, når den føjes til et visual. Derfor kan de konfigureres som implicitte målinger, uanset om de vises med sigma-symbolet eller ej, når de føjes til en visualisering.

Begrebet *beregnet måling* findes ikke i tabelmodellering. Ordet *beregnet* bruges til at beskrive beregnede tabeller og beregnede kolonner, hvilket adskiller dem fra tabeller og kolonner, der stammer fra Power Query. Begrebet eksplicit måling findes ikke i Power Query.

Næste undermodul: Skriv DAX-formler

Skriv DAX-formler

A single value for column 'WoW' in table 'Table Current' cannot be determined.

This can happen when a measure formula refers to a column that contains many values without specifying an aggregation such as min, max, count, or sum to get a single result.

Error This can happen when a measure formula refers to a column that contains many values without.

Looks like you're trying to create a measure instead of a calculated column. If you use the same exact code, but as a calculated column, you shouldn't have an issue.

The problem is that measures are calculated over the entire data set unless otherwise given a filter context. So the error message in that case is accurate.

Hope this helps

David

[View solution in original post](#)

Fuldført100 XP

- 6 minutter

Hver modelberegningstype, beregnet tabel, beregnet kolonne eller måling defineres ud fra sit navn efterfulgt af lighedstegnet (=), som igen efterfølges af en DAX-formel. Brug følgende skabelon til at oprette en modelberegning:

DAXKopier

<Calculation name> = <DAX formula>

Definitionen af den beregnede tabel **Ship Date** , der duplikerer dataene i tabellen **Date** , er f.eks.:

DAXKopier

Ship Date = 'Date'

En DAX-formel består af udtryk, der returnerer et resultat. Resultatet er enten et tabelobjekt eller en skalarværdi. Formler for beregnede tabeller skal returnere et tabelobjekt. formuler for beregnede kolonner og målinger skal returnere en skalarværdi (enkelt værdi).

Formler samles ved hjælp af:

- DAX-funktioner
- DAX-operatorer
- Referencer til modelobjekter
- Konstantværdier, f.eks. tallet 24 eller konstantteksten "FY" (forkortelse for regnskabsår)
- DAX-variabler
- Mellemrum

Tip

Når du indtaster DAX-formler i Power BI Desktop, har du fordel af *IntelliSense*. IntelliSense er en hjælp til kodefuldførelse, der viser funktioner og modelressourcer. Når du vælger en DAX-funktion, får du også en definition og en beskrivelse. Vi anbefaler, at du bruger IntelliSense, så du hurtigt kan oprette nøjagtige formuler.

DAX-funktioner

På samme måde som Microsoft Excel er DAX et funktionelt sprog, hvilket betyder, at formuler er afhængige af funktioner for at nå bestemte mål. DAX-funktioner har typisk argumenter, der tillader overførsel af variabler. Formuler kan bruge mange funktionskald og indlejrer ofte funktioner i andre funktioner.

I en formel skal funktionsnavne efterfølges af parenteser. Variablerne angives i parentes.

Bemærk

Nogle funktioner kræver ikke argumenter, eller argumenter kan være valgfri.

Arbejde med DAX-funktioner beskrives senere i dette modul.

DAX-operatorer

Formuler er også afhængige af operatorer, som kan udføre aritmetiske beregninger, sammenligne værdier, arbejde med strenge eller teste betingelser.

DAX-operatorer beskrives mere detaljeret senere i dette modul.

Referencer til modelobjekter

Formuler kan kun referere til tre typer modelobjekter: tabeller, kolonner og målinger. En formel kan ikke referere til et hierarki eller et hierarkiniveau. Husk, at et hierarkiniveau er baseret på en kolonne, så formelen kan referere til et hierarkiniveaus kolonne.

Tabelreferencer

Når du refererer til en tabel i en formel officielt, er tabelnavnet omsluttet af enkelte anførselstegn. I følgende definition af en beregnet tabel kan du se, at tabellen **Date** er omsluttet af enkelte anførselstegn.

DAXKopiér, fx med New table tast -> åbnes editor

Ship Date = 'Date'

Enkelte anførselstegn kan dog udelades, når følgende betingelser er opfyldt:

1. Tabelnavnet indeholder ikke integrerede mellemrum.
2. Tabelnavnet er ikke et reserveret ord, der bruges af DAX. Alle DAX-funktionsnavne og -operatorer er reserverede ord. *Date er et DAX-funktionsnavn, hvilket forklarer, hvorfor du skal omslutte den i enkelte anførselstegn, når du refererer til en tabel med navnet Date.*

I følgende definition af en beregnet tabel er det muligt at udelade de enkelte anførselstegn, når der refereres til tabellen Airport:

DAXKopiér

Arrival Airport = Airport

Kolonnereferencer

Når du refererer til en kolonne i en formel, skal kolonnenavnet være omsluttet af kantede parenteser. Alternativt kan dens tabelnavn være foranstillet. Følgende målingsdefinition refererer f. eks. til kolonnen **Sales Amount**.

DAXKopiér

Revenue = SUM([Sales Amount])

Da kolonnenavne er entydige i en tabel, men ikke nødvendigvis entydige i modellen, kan du gøre kolonnereferencen utvetydig ved at foranstille dens tabelnavn. Denne utvetydige kolonne kaldes en *fuldt kvalificeret kolonne*. Nogle DAX-funktioner kræver angivelse af fuldt kvalificerede kolonner.

Tip

Hvis du vil forbedre læsbarheden af dine *formler anbefaler vi, at du altid foranstiller en kolonnereferences tabelnavn.*

Det tidligere eksempel på en målingsdefinition kan omskrives som:

DAXKopiér

Revenue = SUM(Sales[Sales Amount])

Målingsreferencer

Når du refererer til en måling i en formel, f.eks. referencer til kolonnenavne, skal målingsnavnet være omsluttet af kantede parenteser. Følgende målingsdefinition refererer f. eks. til målingerne **Revenue** og **Cost**.

DAXKopiér

Profit = [Revenue] - [Cost]

Hvis du er DAX-begynder, kan det faktisk, at kolonne- og målingsreferencer altid er omsluttet af kantede parenteser, medføre forvirring, når du forsøger at læse en formel. Men når du bliver dygtige til grundlæggende DAX-elementer, kan du afgøre, hvilken type objekt det er, fordi kolonner og målinger bruges på forskellige måder i DAX-formler.

Tip

Det er muligt at foranstille en målingsreferences tabelnavn. **Målingerne er imidlertid et objekt på modelniveau. Selvom de er knyttet til en hjemmetabel, er det kun en kosmetisk relation for at organisere målinger logisk i ruden Felter.**

Derfor anbefaler vi, at du altid foranstiller en kolonnereferences tabelnavn, mens det modsatte gælder for målinger: **Vi anbefaler, at du aldrig foranstiller en målingsreferences tabelnavn.**

Du kan finde flere oplysninger under [Kolonne- og målingsreferencer](#).

DAX-variabler

Formler kan deklarere DAX-variabler for at lagre resultater.

Hvordan og hvornår du skal bruge DAX-variabler, beskrives senere i dette modul.

Mellemrum

Mellemrum refererer til tegn, som du kan bruge til at formatere formler på en måde, der er hurtig og nem at forstå. Blanktegn omfatter:

- Mellemrum
- Tabulatorer
- return

Blanktegn er valgfri, og det ændrer ikke din formellogik eller påvirker ydeevnen negativt. Det anbefales på det kraftigste, at du vælger en formattypografi og anvender den konsekvent og overvejer følgende anbefalinger:

- Brug mellemrum mellem operatorer.
- Brug faner til at indrykke indlejrede funktionskald.
- Brug return til at adskille funktionsargumenter, især når de er for lange til at kunne være på en enkelt linje. Formatering på denne måde gør det lettere at foretage fejlfinding, især hvis formelen mangler en parentes.
- Hellere for mange mellemrum end for få.

Tip

Tryk på **Skift+Enter** for at angive et return på formellinjen

Hvis du trykker på **Enter** alene, bekræftes formelen.

Overvej følgende målingsdefinition, der skrives på en enkelt linje, og som omfatter fem DAX-funktionskald:

DAXKopier

```
Revenue YoY % = DIVIDE([Revenue] - CALCULATE([Revenue], SAMEPERIODLASTYEAR('Date'[Date])),  
CALCULATE([Revenue], SAMEPERIODLASTYEAR('Date'[Date])))
```

Følgende eksempel er den samme målingsdefinition, men nu formateret, hvilket gør den lettere at læse og forstå:

DAXKopier

Revenue YoY % =

```
DIVIDE(  
    [Revenue]  
    - CALCULATE(  
        [Revenue],  
        SAMEPERIODLASTYEAR('Date'[Date])  
    ),  
    CALCULATE(  
        [Revenue],  
        SAMEPERIODLASTYEAR('Date'[Date])  
    )  
)
```

) virker ikke

Duer ikke

virker ikke

Sales YoY Growth % =

```
VAR SalesPriorYear = CALCULATE(Sales[SalesAmount]; PARALLELPERIOD('Date'[Date]; -12;MONTH))
```

```
VAR SalesPriorYearPct = DIVIDE((Sales[SalesAmount] - SalesPriorYear); SalesPriorYear)
```

```
RETURN SalesPriorYearPct
```

pgm forstår ikke Sales[SalesAmount]: skal være summations-var, altså enkelt-measure, her er det en kolonne værdi, [Revenue]=SUM(Sales[SalesAmount])

dette kan kompilere, og virker nu

Revenue YoY% =

```
var Revenue=SUM(Sales[SalesAmount])
```

```
VAR SalesPriorYearPct=DIVIDE(
```

```
    Revenue- CALCULATE(Revenue;SAMEPERIODLASTYEAR('Date'[Date]));
```

```
    CALCULATE(Revenue;SAMEPERIODLASTYEAR('Date'[Date]))
```

```
)
```

```
return SalesPriorYearPct
```

kunne længe ikke få det til fungere, skyldtes at Data tabellen ikke var markeret som officiel Data tabel

Revenue YoY% =

```
var Revenue=SUM(Sales[SalesAmount])
```

```
var RevenueLastYear=CALCULATE(Revenue;SAMEPERIODLASTYEAR('Date'[Date]))
```

```
VAR SalesPriorYearPct=DIVIDE(
```

```
    (Revenue- RevenueLastYear);RevenueLastYear))
```

```
return SalesPriorYearPct
```

Fiscal Year	Sum of SalesAmount_sum	Revenue	Sum of SalesAmount	Revenue YoY %	Revenue YoYPct
FY2018	23.860.891,17	23.348.493,12	23.348.493,12		
FY2018 Q1	6.005.207,32	5.187.522,85	5.187.522,85		
FY2018 Q2	5.923.348,19	6.559.394,34	6.559.394,34		
FY2018 Q3	5.965.011,12	5.887.191,56	5.887.191,56		
FY2018 Q4	5.967.324,53	5.714.384,37	5.714.384,37		
FY2019	34.070.108,50	33.636.168,52	33.636.168,52	44,06	44,06
FY2019 Q1	10.192.098,13	9.648.655,43	9.648.655,43	86,00	86,00
FY2019 Q2	8.392.458,01	8.642.446,40	8.642.446,40	31,76	31,76
FY2019 Q3	6.769.487,90	7.097.914,70	7.097.914,70	20,57	20,57
FY2019 Q4	8.716.064,46	8.837.154,00	8.837.154,00	11,22	11,22
Total	109.809.274,20	109.809.274,20	109.809.274,20	0,00	0,00

RevenueLastYear = CALCULATE([Revenue];PARALLELPERIOD('Date'[Date]; -12;MONTH)) intet tal ud
 Prøv selv at formatere målingen. Åbn filen [Adventure Works DW 2020 M02.pbix](#) Power BI Desktop, udvid derefter tabellen **Sales** i ruden **Felter**, og vælg derefter målingen **Revenue YoY %**. På formellinjen skal du bruge tabulator- og returtegn for at få det samme resultat som i det forrige eksempel. Når du tilføjer en retur, skal du huske at trykke på **Skift+Enter**.

Denne målingsdefinition kan forbedres yderligere for læsbarhed og ydeevne, hvilket forklares senere i dette modul.

Tip

Et fremragende formateringsværktøj fra en anden kilde, der kan hjælpe dig med at formatere dine beregninger, er [DAX-formateringsfunktionen](#). Med dette værktøj kan du indsætte din beregning og formatere den. Du kan derefter kopiere den formaterede beregning til Udklipsholder og indsætte den i Power BI Desktop igen.

Næste undermodul: DAX-datatyper

Tidligere

DAX-datatyper

Fuldført 100 XP

- 2 minutter

Semantiske modelkolonner har en angivet datatype, som sikrer, at alle kolonneværdier er i overensstemmelse med denne datatype. Kolonnedatatyper er defineret i Power Query eller i tilfælde af beregnede kolonner afledes de af formlen. Målingsdatatyper, der svarer til beregnede kolonnedatatyper, afledes af formlen.

Modeldatatyper er ikke det samme som DAX-datatyper, selvom der findes en direkte relation mellem dem. I følgende tabel kan du se modeldatatyperne og DAX-datatyperne. Bemærk det understøttede værdiområde for hver datatype.

Udvid tabel

Modeldatatype	DAX-datatype	Beskrivelse
Helt tal	64-bit heltal	-2^{63} til og med $2^{63}-1$
Decimaltal	64-bit ægte	Negativ: $-1,79 \times 10^{308}$ til og med $-2,23 \times 10^{-308}$ - nul (0) - positiv: $2,23 \times 10^{-308}$ til og med $1,79 \times 10^{308}$ 17 decimaltal
Boolesk	Boolesk	TRUE eller FALSE
Tekst	Streng	Unicode-tegnstreng
Dato	Dato/klokkeslæt	Gyldige datoer er alle datoer efter den 1. januar 1900
Valuta	Valuta	$-9,22 \times 10^{14}$ til og med $9,22 \times 10^{14}$ – begrænset til fire decimaltal med fast præcision
Ikke relevant	BLANK	I nogle tilfælde svarer det til en database (SQL) NULL

Datatypen BLANK

Datatype BLANK fortjener særlig omtale. DAX bruger BLANK til både database-NUL og tomme celler i Excel. BLANK betyder ikke nul. Måske kan det være nemmere at tænke på det som *fraværet af en værdi*.

To DAX-funktioner er relateret til datatypen BLANK: [BLANK](#) DAX-funktionen returnerer BLANK, mens [ISBLANK](#) DAX-funktionen tester, om et udtryk evalueres til BLANK.

Næste undermodul: Arbejd med DAX-funktioner

Arbejd med DAX-funktioner

Fuldført 100 XP

- 3 minutter

Biblioteket med DAX-funktioner består af hundredvis af funktioner, der hver især er designet til at opfylde et bestemt formål.

Da DAX stammer fra Power Pivot-tilføjelsesprogrammet til Microsoft Excel 2010, er der flere end 80 funktioner, som også findes i Excel. Det var en bevidst design strategi fra Microsofts side, der sikrer, at Excel-brugere hurtigt kan blive produktive med DAX.

Der findes dog mange funktioner, som du ikke kan finde i Excel, da de er specifikke for datamodellering:

- Funktioner til relationsnavigation
- Funktioner til ændring af filterkontekst
- Iteratorfunktioner
- Time intelligence-funktioner
- Stifunktioner

Tip

Hvis du vil søge efter dokumentation, der er relateret til en DAX-funktion, skal du i en websøgning angive nøgleordet *DAX*- efterfulgt af funktionens navn.

Du kan finde flere oplysninger i [Reference til DAX-funktion](#).

Funktioner, der stammer fra Excel

I følgende afsnit beskrives flere nyttige funktioner, som du måske allerede kender til, da de findes i Excel.

IF DAX-funktionen tester, om en betingelse, der er angivet som det første argument, er opfyldt. Den returnerer en værdi, hvis betingelsen er TRUE, og returnerer den anden værdi, hvis betingelsen er FALSE. Funktionens syntaks er:

DAXKopier

```
IF(<logical_test>, <value_if_true>[, <value_if_false>])
```

Tip

Et funktionsargument er valgfrit, når det vises omsluttet af kantede parenteser i dokumentationen.

Hvis **logical_test** evalueres til FALSE, og **value_if_false** ikke angives, returnerer funktionen BLANK.

Der er mange funktioner til opsummering i Excel, herunder SUM, COUNT, AVERAGE, MIN, MAX og mange andre. Den eneste forskel er, at du i DAX overfører en kolonnereference, mens du i Excel overfører et celleområde.

Mange matematiske og logiske funktioner samt tekst-, dato- og klokkeslæt- og informationsfunktioner fra Excel er også tilgængelige. Et lille eksempel på Excel-funktioner, der er tilgængelige i DAX, omfatter ABSf.eks. , ROUND, , SQRT, LEFTLEN, RIGHT, NOWYEARDATEMONTHUPPER, ISNUMBERTRUEFALSE, AND, OR, NOT, og .IFERROR

Funktioner, der ikke stammer fra Excel

To nyttige DAX-funktioner, der ikke er specifikke for modellering, og som ikke stammer fra Excel, er DISTINCTCOUNT og DIVIDE.

Funktionen DISTINCTCOUNT

Du kan bruge **DISTINCTCOUNT** DAX-funktionen til at tælle antallet af entydige værdier i en kolonne. Denne funktion er især effektiv i en analyseløsning. Tænk på, at antallet af kunder adskiller sig fra antallet af *forskellige* kunder. Sidstnævnte tæller ikke kunder, så forskellen er "hvor mange kunder" sammenlignet med "hvor mange *forskellige* kunder".

Funktionen DIVIDE

Du kan bruge DAX-funktionen **DIVIDE** til at opnå division. Du skal angive udtryk for tæller og nævner. Du kan også angive en værdi, der repræsenterer et *alternativt resultat*. Funktionens DIVIDE syntaks er:

DAXKopier

```
DIVIDE(<numerator>, <denominator>[, <alternate_result>]) skal bruge ; ikke ,
```

Funktionen DIVIDE håndterer automatisk division med nul sager. Hvis der ikke angives et andet resultat, og nævneren er nul eller BLANK, returnerer funktionen BLANK. Hvis der angives et alternativt resultat, returneres det i stedet for BLANK.

Denne funktion er praktisk, fordi udtrykket ikke først skal teste nævneværdien. Funktionen er også bedre optimeret til test af nævneværdien end funktionen IF . Ydeevnefordelen er vigtig, da det er dyrt at kontrollere, om der forekommer division med nul. Derudover resulterer brugen af DIVIDE funktionen i et mere præcist og elegant udtryk.

Tip

Vi anbefaler, at du bruger DIVIDE funktionen , når nævneren er et udtryk, der kan returnere nul eller BLANK. Hvis nævneren er en konstantværdi, anbefaler vi, at du bruger divisionsoperatoren (/), som introduceres senere i dette modul. I dette tilfælde garanteres det, at divisionen lykkes, og dit udtryk fungerer bedre, da unødvendige test undgås.

Næste undermodul: Brug DAX-operatorer

[Tidligere](#)

Brug DAX-operatorer

Fuldført100 XP

- 5 minutter

Dine DAX-formler kan bruge operatorer til at oprette udtryk, der udfører aritmetiske beregninger, sammenligner værdier, arbejder med strenge eller testbetingelser.

Tip

Mange DAX-operatorer og -rækkefølger er de samme som dem, der findes i Excel.

Aritmetiske operatorer

I følgende tabel vises de aritmetiske operatorer.

Udvid tabel

Operator	Beskrivelse
+	Tilføjelse
-	Subtraktion
*	Multiplikation
/	Division

Operator	Beskrivelse
^	Eksponentiering

Husk, at når du opdeler to udtryk, og når nævneren kan returnere nul eller BLANK, er det mere effektivt og sikrere at bruge [DIVIDE](#) DAX-funktionen.

Sammenligningsoperatorer

I følgende tabel vises de sammenligningsoperatorer, der bruges til at sammenligne to værdier. Resultatet er enten TRUE eller FALSE.

Udvid tabel

Operator	Beskrivelse
=	Lig med
==	Fuldstændig lig med
>	Større end
<	Mindre end
>=	Større end eller lig med.
<=	Mindre end eller lig med
<>	Ikke lig med

Alle sammenligningsoperatorer, undtagen **Fuldstændig lig med (==)**, behandler BLANK som lig med tallet nul, en tom streng (""), datoen den 30. december, 1899 eller FALSE. Det betyder, at udtrykket [Revenue] = 0 er TRUE, når værdien af [Revenue] enten er nul eller BLANK. Er derimod kun TRUE, [Revenue] == 0 når værdien af [Revenue] er nul.

Tekstsammenkædningsoperator

Brug og-tegnet (&) til at oprette forbindelse mellem eller sammenkæde to tekstværdier for at oprette én fortløbende tekstværdi. Se f. eks. følgende definition af en beregnet kolonne:

DAXKopier

```
Model Color = 'Product'[Model] & "-" & 'Product'[Color]
```

Logiske operatorer

Brug logiske operatorer til at kombinere udtryk, der giver et enkelt resultat. I følgende tabel vises alle logiske operatorer.

Udvid tabel

Operator	Beskrivelse
&&	Opretter en AND-betingelse mellem to udtryk, hvor hvert enkelt udtryk har et boolesk resultat. Hvis begge TRUE, returnerer kombinationen af udtrykkene også TRUE. Ellers returnerer kombinationen FALSE.
 (dobbeltrør)	Opretter en OR-betingelse mellem to logiske udtryk. Hvis et af følgende udtryk returnerer Sand, er resultatet Sand. Hvis begge udtryk er Falske, er resultatet Falsk.
IN	Opretter en logisk OR-betingelse mellem de rækker, der sammenlignes med en tabel. Bemærk! Der bruges tabelkonstruktørens syntaks.
NOT	Inverterer tilstanden for et boolesk udtryk (FALSE til TRUE og omvendt).

Et eksempel, der bruger den IN logiske operator, er målingsdefinitionen **ANZ Revenue**, som bruger [CALCULATE](#) DAX-funktionen til at gennemtvinge et bestemt filter for to lande: Australien og New Zealand.

Bemærk

Du får en introduktion til den effektive CALCULATE funktion, når du lærer, hvordan du ændrer filterkonteksten.

DAXKopier

ANZ Revenue =

```
CALCULATE(
    [Revenue],
    Customer[Country-Region] IN {
        "Australia",
        "New Zealand"
    }
)
```

Operatorrækkefølge

Når din DAX-formel indeholder flere operatorer, bruger DAX regler til at fastsætte evalueringsrækkefølgen, hvilket også kaldes *operatorrækkefølge*. Handlinger sorteres i henhold til følgende tabel.

Udvid tabel

Operator	Beskrivelse
^	Ekspontiering
-	Tegn (som i -1)
* og /	Multiplikation og division

Operator	Beskrivelse
NOT	NOT
+ og -	Addition og subtraktion
&	Sammenkædning af to tekststreng
=, <=, >, < >, <= >, < >	Sammenligning

Når operatorerne har samme prioritetsværdi, sorteres de fra venstre mod højre.

Normalt er operatorrækkefølgen den samme som den, der findes i Excel. Hvis du har brug for at tilsidesætte evalueringsrækkefølgen, skal du gruppere handlinger i parenteser.

Se f. eks. følgende definition af en beregnet kolonne:

DAXKopier

Extended Amount = Sales[Order Quantity] * Sales[Unit Price] * 1 - [Unit Price Discount Pct]

I dette eksempel på en definition af en beregnet kolonne, opnås der et forkert resultat, fordi multiplikation finder sted før subtraktion. I den følgende korrekte definition af en beregnet kolonne bruges der parenteser til at sikre, at subtraktionerne finder sted før multiplikationerne.

DAXKopier

Extended Amount = Sales[Order Quantity] * Sales[Unit Price] * (1 - [Unit Price Discount Pct])

Tip

Det kan være en udfordring at huske operatorrækkefølgen, især for DAX-begyndere. Derfor anbefaler vi, at du tester dine formler grundigt. Når formlerne ikke giver det korrekte resultat på grund af en forkert evalueringsrækkefølge, kan du eksperimentere ved at tilføje parenteser for at justere evalueringsrækkefølgen. Du kan også tilføje parenteser for at forbedre dine formlers læsbarhed.

Du kan finde flere oplysninger om DAX-operatorer og rækkefølgen i [DAX-operatorer](#).

Implicit konvertering

Når du skriver en DAX-formel, der bruger operatorer til at kombinere forskellige datatyper, behøver du ikke at konvertere typer eksplicit. DAX identificerer normalt datatyperne for modelobjekter, der refereres til, og udfører implicitte konverteringer, hvor det er nødvendigt, for at fuldføre den angivne handling.

Der kan dog være nogle begrænsninger for de værdier, der kan konverteres korrekt. Hvis en værdi eller en kolonne har en datatype, der ikke er kompatibel med den aktuelle handling, returnerer DAX en fejl. Et forsøg på at multiplicere en datoværdi resulterer f. eks. i en fejl, fordi den ikke er logisk.

BLANK håndteres forskelligt, afhængigt af hvilken operator der bruges. Den håndteres på samme måde, som Excel behandler BLANK på, men anderledes end den måde databaser (SQL) behandler NULL på. BLANK behandles som nul, når der anvendes aritmetiske operatorer på den, og som en tom streng, når den sammenkædes med en streng.

Tip

Det kan være en udfordring at huske, hvordan BLANK håndteres, især for DAX-begyndere. Derfor anbefaler vi, at du tester dine formler grundigt. Når TOMME værdier giver uventede resultater, kan du overveje at bruge [IF](#) funktionerne og [ISBLANK](#) DAX til at teste for BLANK og derefter reagere på en passende måde.

Næste undermodul: Brug DAX-variabler

[Tidligere](#) [Næste](#)

Har du brug for hjælp? Se i vores [fejlfindingsvejledning](#), eller angiv specifik feedback ved at [rapportere et problem](#).

Feedback

Brug DAX-variabler

Fuldført 100 XP

- 1 minut

Du kan deklarere DAX-variabler i dine formeludtryk. Når du deklarerer mindst én variabel, bruges en RETURN delsætning til at definere udtrykket, som derefter refererer til variablerne.

Vi anbefaler, at du bruger variabler, da de giver flere fordele:

- Forbedring af læsbarheden og vedligeholdelsen af dine formler.
- Ydeevnen forbedres, fordi variabler evalueres én gang, og kun når eller hvis der er brug for dem.
- Giver mulighed for nem test af en kompleks formel (på designtidspunktet) ved at returnere den relevant variabel.

I følgende eksempel vises der en formel, der deklarerer en variabel. Målingsdefinitionen **Revenue YoY %** omskrives for at deklarere en variabel, der har fået tildelt værdien for det foregående års indtægt.

DAXKopier

Revenue YoY % =

VAR RevenuePriorYear =

```
CALCULATE(  
    [Revenue],  
    SAMEPERIODLASTYEAR('Date'[Date])  
)
```

RETURN

```

DIVIDE(
    [Revenue] - RevenuePriorYear,
    RevenuePriorYear
)

```

[Revenue] er aggregeret værdi, Revenue = SUM(Sales[Sales Amount

```

Virker i Adventure Works DW2020 M04
Revenue YoY % =
VAR RevenuePriorYear =
    CALCULATE(
        [Revenue];
        SAMEPERIODLASTYEAR('Date'[Date])
    )
RETURN
    DIVIDE(
        [Revenue] - RevenuePriorYear;
        RevenuePriorYear
    ) * 100

```

Bemærk, at delsætningen RETURN refererer til variabelen to gange. Denne forbedrede målingsdefinitionsformel kører mindst halvdelen af tiden, da det ikke er nødvendigt at evaluere det foregående års indtægt to gange.

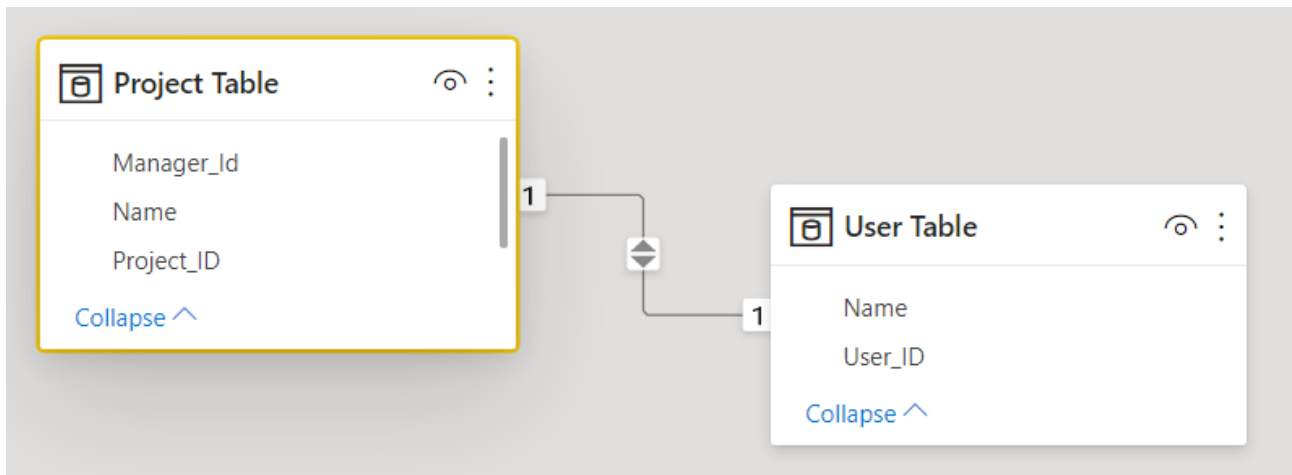
Omstrukturer målingen **Revenue YoY %** i Power BI Desktop-filen [Adventure Works DW 2020 M02. pbix](#) for at opnå det samme resultat som i det forrige eksempel.

Du finder flere oplysninger om brug af DAX-variabler, kan du se [Brug variabler til at forbedre dine formler](#).

Sammenknyt to tabeller i DAX-kode, med Related()

I don't use the direct query in Power BI, but if you have two separate tables and you can define the relationship between them, then it will work with your data.

1. Create a relationship between *Project Table* and *User Table* by dragging the column *Manager_Id* to *User_ID* in **Model View** of Power BI



2. Create a new calculated column in Project with following DAX:

3. `Project_Manager = RELATED('User Table'[Name])` //henter Name-feltværdien fra User

This will give you a new column in the Project table with the manager name corresponding to the Manager_Id

Project_ID	Name	Manager_Id	Project_Manager
2	Pointy	2	Baby
1	Aladeen	10	Musk

[S](#)

Tjek din viden

Fuldført 200 XP

- 4 minutter

Besvar følgende spørgsmål for at se, hvad du har lært.

1.

Du bruger Power BI Desktop til at udvikle en model. Den har en tabel med navnet Sales, der indeholder en kolonne med navnet CustomerKey. I rapporter skal du bruge en beregning for at vise det antal forskellige kunder, der har afgivet ordrer. Hvilken type DAX-beregning føjer du til modellen?

Beregnet tabel

Beregnet kolonne

Måling

Korrekt. Du kan føje en måling til tabellen sales ved hjælp af DAX-funktionen DISTINCTCOUNT.

2.

Du bruger Power BI Desktop til at udvikle en model. Den har en tabel med navnet Customer, der indeholder en kolonne med navnet DateOfBirth. Du skal gruppere kunder efter den aktuelle alder i rapporter. Hvilken type DAX-beregning vil du føje til tabellen Customer?

Beregnet tabel

Beregnet kolonne

Korrekt. En beregnet kolonne kan lagre den aktuelle alder for hver kunde. Denne kolonne kan derefter bruges i rapporter til at gruppere kunder efter alder.

Måling

Forkert. En måling kan ikke bruges til at gruppere kunder efter alder.

3.

Du bruger Power BI Desktop til at udvikle en model. Den har en tabel med navnet Geography, som har to relationer til tabellen Sales. Ét relationsfilter efter kundeområde og de andre filtre efter salgsområde. Du skal oprette en dimension med forskellige roller, så begge filtre er mulige. Hvilken type DAX-beregning føjer du til modellen?

Beregnet tabel

Korrekt. Tilføj en beregnet tabel, der duplikerer dataene i tabellen Geography og føjer en aktiv relation til tabellen Sales. Begge geografitabeller har nu aktive relationer, der gør det muligt for rapportbrugere at filtrere efter kundeområde eller salgsområde.

Beregnet kolonne

Måling

4.

Du skriver en DAX-formel, der føjer BLANK til tallet 20. Hvad er resultatet?

Resultatet er nul (0).

Resultatet er 20.

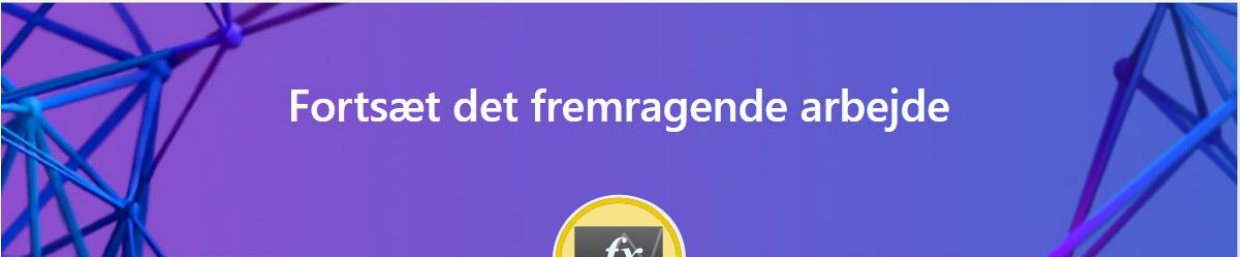
Korrekt. BLANK konverteres til nul og føjes derefter til tallet 20.

Resultatet er BLANK.

Forkert. Når du føjer en numerisk værdi til BLANK, repræsenteres BLANK som nul og føjes derefter til værdien.

Næste undermodul: Resumé

Resultater



Fortsæt det fremragende arbejde



Skriv DAX-formler til Power BI
Desktop-modeller

Du har opnået et resultat!

Tillykke, men hvad skal du gøre nu?

Lad os først dele dit resultat

Du afsætter tid til at lære noget nyt, så lad dit netværk tage del i din sejr!



<https://learn.microsoft.com/api/achievements/share/da-dk/bjarnehansen-7412/HAJ5NR28?sharingId=39714CB1FE13C67B>

**** Føj beregnede tabeller og kolonner til Power BI Desktop-modeller

- 19 min.
- Modul
- 6 Enheder

Feedback

Øvet

Appbygger

Dataanalytiker

Power BI

Microsoft Power Platform

Når du har afsluttet dette modul, kan du føje beregnede tabeller og beregnede kolonner til din semantiske model. Du vil også kunne beskrive rækkekontekst, der bruges til at evaluere formler for beregnede kolonner. Da det er muligt at føje kolonner til en tabel ved hjælp af Power Query, lærer du også, hvornår det er bedst at oprette beregnede kolonner i stedet for Power Query brugerdefinerede kolonner.

Læringsmål

Når du har fuldført dette modul, vil du være i stand til følgende:

- Oprette beregnede tabeller.
- Oprette beregnede kolonner.
- Identificere rækkekontekst.
- Bestemme, hvornår der skal bruges en beregnet kolonne i stedet for en brugerdefineret Power Query-kolonne.
- Føje en datotabel til din model ved hjælp af DAX-beregninger.

[Start](#)Tilføj

-
-

Forudsætninger

Få indblik i oprettelse af Microsoft Power BI Desktop-modeller og design af Power BI-rapportlayouts, og få en forståelse af, hvordan du skriver DAX-formler (Data Analysis Expressions)

Dette modul er en del af disse læringsforløb

- [Brug DAX i Power BI Desktop](#)
- [Develop dynamic reports with Microsoft Power BI](#)
- [Modeldata med Power BI](#)
- [Introduktion](#) 7 min.
- [Opret beregnede kolonner](#) 4 min.
- [Få mere at vide om rækkekontekst](#) 2 min.
- [Vælg en teknik til tilføjelse af en kolonne](#) 2 min.
- [Tjek din viden](#) 3 min.
- [Resumé](#) 1 min.

Introduktion

Fuldført 100 XP

- 7 minutter

calculated tables don't store data physically but instead create a virtual table on the fly, evaluating the formula each time it is queried. Calculated tables are not stored in the data model; they are dynamically generated. Calculated tables are recalculated if any of the tables they pull data from are refreshed or updated

Du kan skrive en DAX-formel (Data Analysis Expressions) for at føje en *beregnet tabel* til din model. Formlen kan duplikere eller transformere eksisterende modeldata for at oprette en ny tabel.

Bemærk

En beregnet tabel kan ikke oprette forbindelse til eksterne data. Du skal bruge Power Query for at fuldføre opgaven.

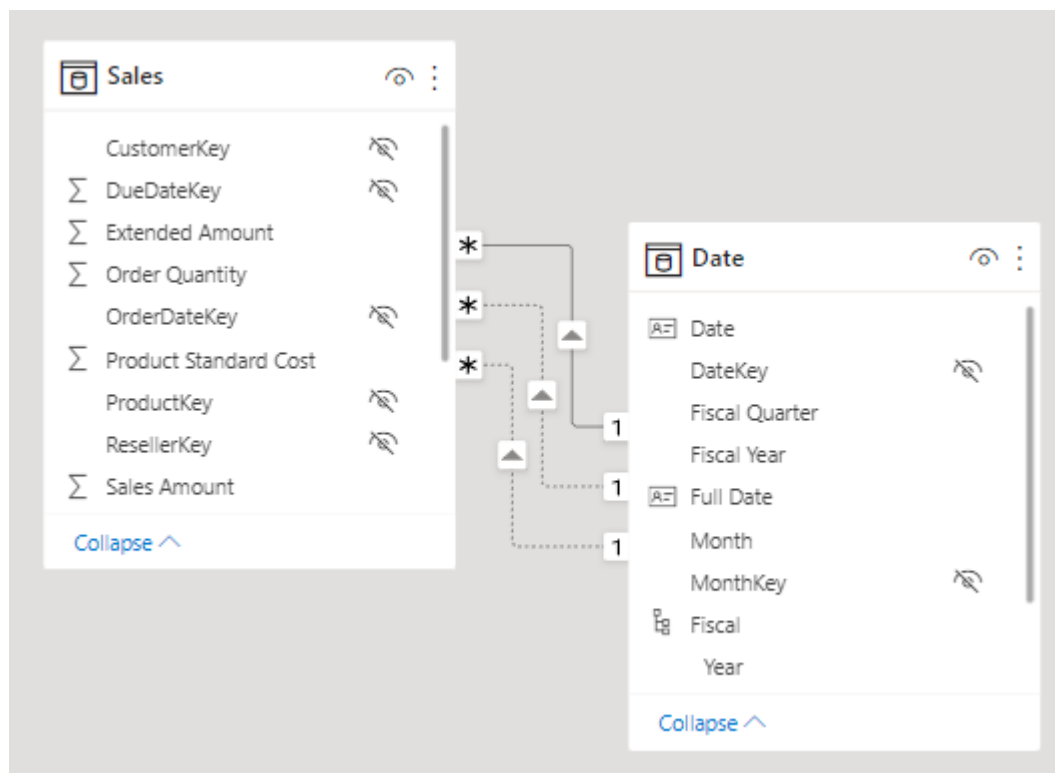
En formel for en beregnet tabel skal returnere et tabelobjekt. Den mest enkle formel kan duplikere en eksisterende modeltabel.

Beregnete tabeller har en ulempe: De øger størrelsen på modellageret, og de kan forlænge dataopdateringstiden. Årsagen er, at beregnede tabeller genberegnes, når de har formelafhængigheder i forhold til opdaterede tabeller.

Dupliker en tabel

I følgende afsnit beskrives den almindelige design udfordring, der kan løses ved at oprette en beregnet tabel. Først skal du hente og åbne filen [Adventure Works DW 2020 M03. pbix](#), og derefter skal du skifte til modeldiagrammet.

Bemærk, at tabellen **Sales** i modeldiagrammet har tre relationer til tabellen **Date**.



I modeldiagrammet vises der tre relationer, da der lagres salgsdata efter ordredato, afsendelsesdato og forfaldsdato i tabellen Sales. Hvis du kigger nærmere på kolonnerne **OrderDateKey**, **ShipDateKey** og **DueDateKey**, kan du se, at én relation er repræsenteret af en ubrudt linje, hvilket er den *aktive relation*. De øvrige relationer, der er repræsenteret af stiplede linjer, er *inaktive relationer*.

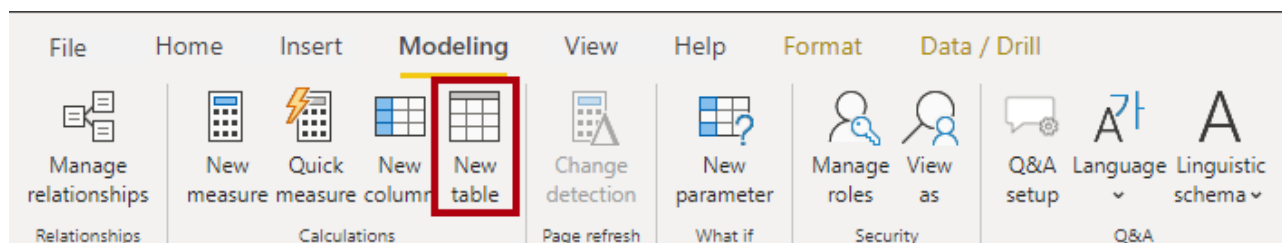
Bemærk

Der kan kun være én aktiv relation mellem to modeltabeller.

Hold markøren over den aktive relation i diagrammet for at fremhæve de relaterede kolonner, hvilket er den måde du interagerer med modeldiagrammet på for at få mere at vide om relaterede kolonner. **I dette tilfælde filtrerer den aktive relation kolonnen OrderDateKey i tabellen Sales**. Filtre, der anvendes på tabellen **Date**, overføres derfor til tabellen **Sales** for at filtrere efter ordredato. de filtrerer aldrig efter afsendelsesdato eller forfaldsdato.

Det næste trin er at slette de to inaktive relationer mellem tabellen **Date** og tabellen **Sales**. Hvis du vil slette en relation, skal du højreklikke på den og derefter vælge **Slet** i genvejsmenuen. Sørg for at slette begge inaktive relationer.

Tilføj derefter en ny tabel for at tillade, at rapportbrugere filtrerer salg efter afsendelsesdato. Skift til rapportvisning, og vælg derefter **Ny tabel** inde fra gruppen **Beregninger** på båndfanen **Udformning**.



Angiv følgende definition af en beregnet tabel på formellinjen (placeret under båndet), og tryk derefter på **Enter**.

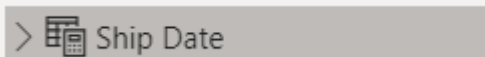
Opret kopi af Date, til ny Ship-Date., så der kan filtreres efter ship-dates i Sales med relation

DAXKopier

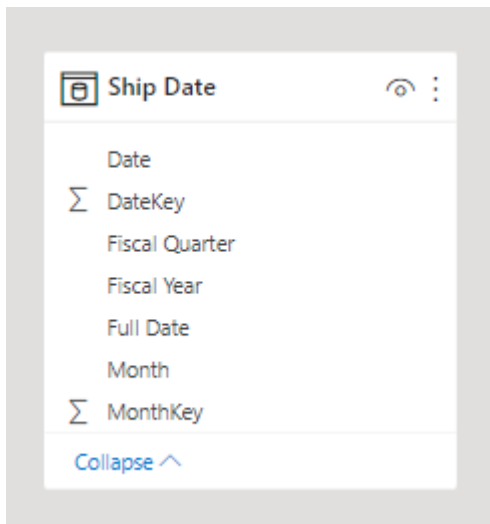
Ship Date = 'Date'

Den beregnede tabeldefinition duplikerer dataene i tabellen **Date** for at oprette en ny tabel med navnet **Ship Date**. Tabellen **Afsendelsesdato** indeholder præcis de samme kolonner og rækker som tabellen **Date**. Når dataene i tabellen **Date** opdateres, genberegnes tabellen **Ship Date**, så de altid er synkroniseret.

Kommer ind, nederst i data pane, uafhængigt af Sale:



Skift til modeldiagrammet, og bemærk derefter tilføjelsen af tabellen **Ship Date**.



Opret derefter en relation mellem kolonnen **DateKey** i tabellen **Ship Date** og kolonnen **ShipDateKey** i tabellen **Sales** . Du kan oprette relationen ved at trække kolonnen **DateKey** i tabellen **Ship Date** til kolonnen **ShipDateKey** i tabellen **Sales** .

En beregnet tabel duplikerer kun data. Den duplikerer ikke nogen modelgenskaber eller -objekter, f.eks. kolonnesynlighed eller hierarkier. Du skal konfigurere dem til den nye tabel, hvis det er nødvendigt.

Tip

Det er muligt at omdøbe kolonner i en beregnet tabel. I dette eksempel er det en god idé at omdøbe kolonner, så de beskriver deres formål bedre. Kolonnen **Fiscal Year** i tabellen **Ship Date** kan f.eks. omdøbes til **Ship Fiscal Year**. Når felter fra tabellen **Afsendelsesdato** bruges i visualiseringer, medtages deres navne derfor automatisk i billedtekster som f.eks. titlen på visualiseringen eller aksnavnene.

Hvis du vil fuldføre designet af tabellen **Afsendelsesdato** , kan du:

- Omdøb følgende kolonner:
 - **Date** til **Ship Date**
 - **Fiscal Year** til **Ship Fiscal Year**
 - **Fiscal Quarter** til **Ship Fiscal Quarter**
 - **Month** til **Ship Month**
 - **Full Date** til **Ship Full Date**
- Sortér kolonnen **Ship Full Date** efter kolonnen **Ship Date**.
- Sortér kolonnen **Ship Month** efter kolonnen **MonthKey**.
- Skjul kolonnen **MonthKey**.
- Opret et hierarki med navnet **Fiscal** med følgende niveauer:
 - Ship Fiscal Year
 - Ship Fiscal Quarter

- Ship Month
- Ship Full Date
- Markér tabellen Ship Date som en datotabel ved hjælp af kolonnen **Ship Date**.

Beregnete tabeller egner sig godt til arbejde i scenarier, når der findes flere relationer mellem to tabeller, som beskrevet tidligere. De kan også bruges til at føje en datotabel til din model. Datotabeller skal anvende særlige tidsfiltre, der kaldes *time intelligence*.

Opret en datotabel

I det næste eksempel oprettes der en anden beregnet tabel, denne gang ved hjælp [CALENDARAUTO](#) af DAX-funktionen.

Create den beregnede tabel **Due Date** ved hjælp af følgende definition.

DAXKopier

Due Date = CALENDARAUTO(6)

CALENDARAUTO DAX-funktionen tager et enkelt valgfrit argument, som er det sidste månedsnummer i året, og returnerer en tabel med en enkelt kolonne. Hvis du ikke angiver et månedsnummer, antages det, at det er 12 (for december). I Adventure Works slutter deres regnskabsår f. eks. den 30. juni hvert år, så værdien 6 (for juni) angives.

Funktionen scanner alle dato- og dato/klokkeslæt-kolonner i din model for at bestemme de tidligste og seneste værdier for gemte datoer. Derefter producerer den et komplet sæt datoer, der spænder over alle datoer i din model, så det sikres, at hele årets datoer er indlæst. Hvis den tidligste dato, der er gemt i din model, f.eks. er den 15. oktober 2021, er den første dato, der returneres af CALENDARAUTO funktionen, den 1. juli 2021. Hvis den seneste dato, der er gemt i modellen, er den 15. juni 2022, er den sidste dato, der returneres af funktionen CALENDARAUTO , den 30. juni 2022.

Funktionen garanterer, CALENDARAUTO at følgende krav til *at markere en datotabel* er opfyldt:

- Tabellen skal indeholde en kolonne med datatypen Dato.
- Kolonnen skal indeholde hele år.
- Der må ikke mangle nogen datoer i kolonnen.

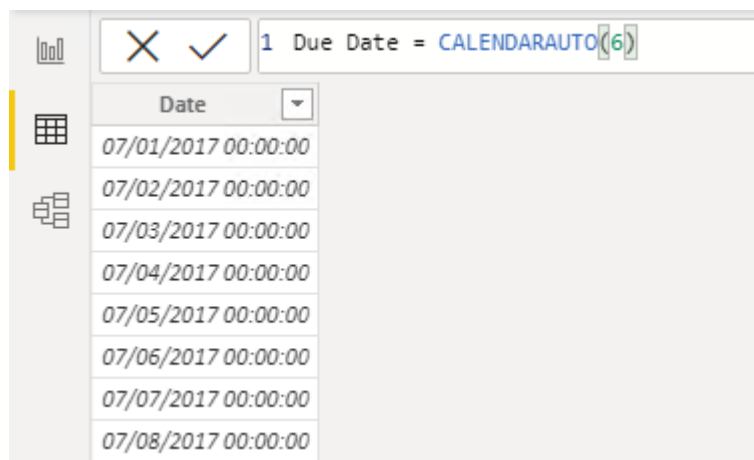
Tip

Du kan også oprette en datotabel ved hjælp [CALENDAR](#) af DAX-funktionen og angive to datoværdier, som repræsenterer datointervallet. Funktionen genererer én række for hver dato i intervallet. Du kan angive statiske datoværdier eller angive udtryk, der henter de tidligste/seneste datoer fra bestemte kolonner i din model.

Derefter skal du skifte til datavisning og derefter vælge tabellen **Due Date** i ruden **Felter**. Gennemgå nu kolonnen med datoer. Det kan være en god idé at sortere dem, så du få vist den tidligste dato i den første række, ved at vælge pilen i kolonneoverskriften **Date** og derefter sortere dem i stigende rækkefølge.

Bemærk

Sortering eller filtrering af kolonner ændrer ikke, den måde værdierne gemmes på. Disse funktioner hjælper dig med at udforske og forstå dataene.



Nu, hvor kolonnen **Date** er markeret, skal du gennemse meddelelsen på statuslinjen (placeret i nederste venstre hjørne). Den beskriver, hvor mange rækker tabellen lagrer, og hvor mange forskellige værdier der findes i den valgte kolonne.

Table: Due Date (1,461 rows) Column: Date (1,461 distinct values)

Når tabelrækkerne og de forskellige værdier er ens, betyder det, at kolonnen indeholder entydige værdier. Denne faktor er vigtig af to årsager: Den opfylder kravene til markering af en datotabel, og den gør det muligt at bruge denne kolonne i en modelrelation som en-siden.

Den beregnede tabel **Due Date** genberegnes, hver gang en tabel, der indeholder en datokolonne, opdateres. Det vil sige, at når en række indlæses i tabellen **Sales** med ordredatoen 1. juli 2022, udvides tabellen **Due Date** automatisk til at omfatte datoer til slutningen af det næste år: 30. juni 2023.

Tabellen **Due Date** kræver yderligere kolonner for at understøtte de kendte krav til filtrering og gruppering, især efter år, kvartal og måned.



Markér herefter Due Date om omdanner til Date-tabel

Merge to tabeller med Union()

For example, imagine you're a personnel manager who has a table of **Northwest Employees** and another table of **Southwest Employees**. You want to combine the two tables into a single table called **Western Region Employees**.

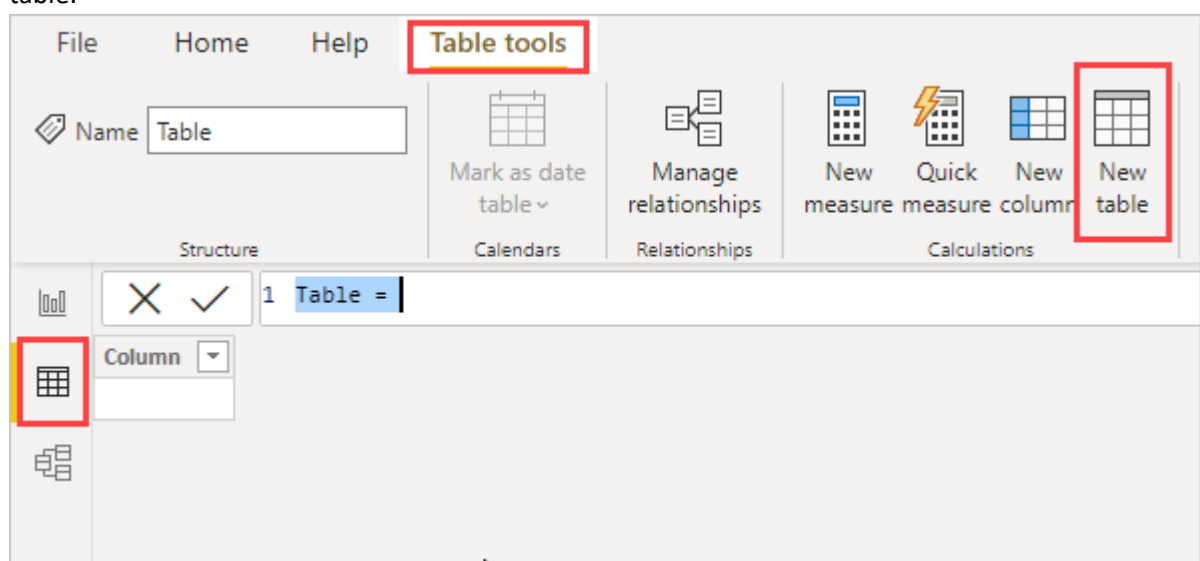
Northwest Employees

Employee	City	State	Tenure
Allen, Kerry	Eugene	OR	1
Baker, Cameron	Portland	OR	15
Morin, Max	Redmond	WA	10
Ramirez, Riley	Portland	OR	3
Rocha, Kim	Redmond	WA	15
Smith, Avery	Redmond	WA	15

Southwest Employees

Employee	City	State	Tenure
Connors, Morgan	San Diego	CA	10
Irwin, Jesse	Phoenix	AZ	3
Nguyen, Rory	Los Angeles	CA	3
Torres, Devon	Los Angeles	CA	2

1. In Report View, Table View, or Model View of Power BI Desktop, in the **Calculations** group select **New table**. It's a bit easier to do in **Table tools** in the Table View, because then you can immediately see your new calculated table.



2. Enter the following formula in the formula bar:

DAXCopy

Western Region Employees = UNION('Northwest Employees', 'Southwest Employees')

A new table named **Western Region Employees** is created, and appears just like any other table in the **Fields** pane. You can create relationships to other tables, add measures and calculated columns, and add the fields to reports just like with any other table.

<

Næste undermodul:

Opret beregnede kolonner

Næste

Opret beregnede kolonner

Fuldført100 XP

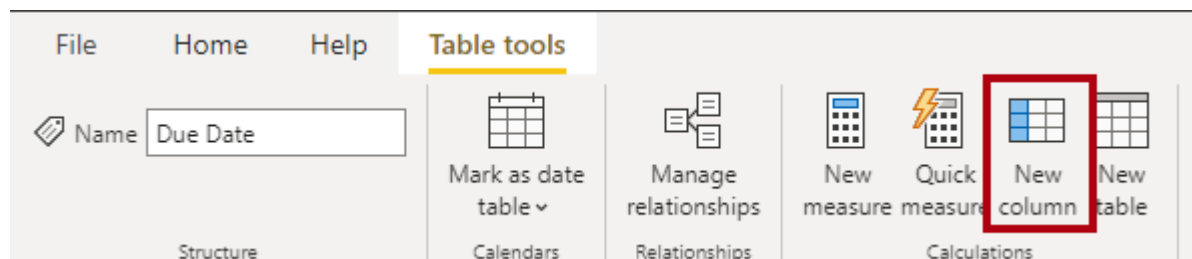
- 4 minutter

Du kan skrive en DAX-formel for at føje en *beregnet kolonne* til en vilkårlig tabel i din model. En formel for en beregnet kolonne skal returnere en skalarværdi eller en enkelt værdi, nemlig en værdi til hver række.

Beregnete kolonner i importmodeller har en ulempe: De øger størrelsen på modellageret, og de kan forlænge dataopdateringstiden. Årsagen er, at beregnede tabeller beregnes igen, når de har formelafhængigheder i forhold til opdaterede tabeller.

Kontrollér, at tabellen **Due Date** er valgt i ruden **Felter** i datavisning. Før du opretter en beregnet kolonne, skal du først omdøbe **Date** til **Due Date**.

Nu kan du føje en beregnet kolonne til tabellen **Due Date**. Vælg **Ny kolonne** fra gruppen **Beregninger** på det kontekstafhængige bånd **Tabelværktøjer** for at oprette en beregnet kolonne.



Angiv følgende definition af en beregnet kolonne på formellinjen, og tryk derefter på **Enter**.

Detaljerede Dato-kolonner oprettes

DAXKopier

Due Fiscal Year =

"FY"

```
& YEAR('Due Date'[Due Date])  
+ IF(  
    MONTH('Due Date'[Due Date]) > 6,  
    1  
)
```

Definitionen af den beregnede kolonne følger kolonnen **Due Fiscal Year** til tabellen **Due Date** . I nedenstående fremgangsmåde beskrives det, hvordan Microsoft Power BI evaluerer formelen for den beregnede kolonne:

1. Plustegnet (+) evalueres før tekst sammenkædningsoperatoren (&).
2. DAX-funktionen [YEAR](#) returnerer hele talværdien for forfaldsdatoåret.
3. [IF](#) DAX-funktionen returnerer værdien, når månedsnummeret for forfaldsdatoen er 7-12 (juli til december). Ellers returnerer den BLANK. Da Adventure Works-regnskabsåret f. eks. er juli-juni, **bruger de seneste seks måneder i kalenderåret det næste kalenderår som regnskabsåret.**
4. Årsværdien føjes til den værdi, der returneres af funktionen IF , som er værdien en eller BLANK. Hvis værdien er BLANK, konverteres den implicit til nul (0) for at tillade additionen at producere værdien for regnskabsåret.
5. Konstanttekstværdien "FY" sammenkædes med værdien for regnskabsåret, som implicit konverteres til tekst.

Tilføj en anden beregnet kolonne ved hjælp af følgende definition:

DAXKopier

Due Fiscal Quarter =

'Due Date'[Due Fiscal Year] & " Q"

```
& IF(  
    MONTH('Due Date'[Due Date]) <= 3,  
    3, //else  
    IF(  
        MONTH('Due Date'[Due Date]) <= 6,  
        4,  
        IF(  

```

```

        MONTH('Due Date'[Due Date]) <= 9,
        1,
        2
    )
)
)

```

Definitionen af den beregnede kolonne følger kolonnen **Due Fiscal Quarter** til tabellen **Due Date** . **Funktionen IF returnerer kvartalsnummeret (kvartal 1 er juli-september) i skatteåret**, og resultatet sammenkædes med kolonneværdien **Due Fiscal Year** og konstantteksten Q.

Tilføj en tredje beregnet kolonne ved hjælp af følgende definition:

DAXKopier

Due Month = FORMAT('Due Date'[Due Date], "yyyy mmm") //2019 Jul

Definitionen af den beregnede kolonne følger kolonnen **Due Month** til tabellen **Due Date** . [FORMAT](#) DAX-funktionen konverterer kolonneværdien **Due Date** til tekst ved hjælp af en formatstreng. I dette tilfælde opretter formatstrengen en mærkat, der beskriver året og det forkortede månedsnavn.

Bemærk

Der findes mange brugerdefinerede formater for dato/klokkeslæt. Du kan finde flere oplysninger under [Brugerdefinerede formater for dato og klokkeslæt til funktionen FORMAT](#).

Tilføj en fjerde beregnet kolonne ved hjælp af følgende definition:

DAXKopier

Due Full Date = FORMAT('Due Date'[Due Date], "yyyy mmm, dd") //2019 Jul, 01

Tilføj en femte beregnet kolonne ved hjælp af følgende definition:

DAXKopier

MonthKey =

(YEAR('Due Date'[Due Date]) * 100) + MONTH('Due Date'[Due Date])

Får rykket årstal 2 pladser frem, fx 201700 +7 ->201707

Den beregnede kolonne **MonthKey** multiplicerer forfaldsdatoåret med værdien 100 og tilføjer derefter månedsnummeret for forfaldsdatoen. Den opretter en numerisk værdi, der kan bruges til at sortere tekstværdierne for **Due Month** i kronologisk rækkefølge.

Kontrollér, at tabellen **Due Date** indeholder seks kolonner. Den første kolonne blev tilføjet, da den beregnede tabel blev oprettet, og de øvrige fem kolonner blev tilføjet som beregnede kolonner.

✕

✓

1 MonthKey =

2 (YEAR('Due Date'[Due Date]) * 100) + MONTH('Due Date'[Due Date])

Due Date	Due Fiscal Year	Due Fiscal Quarter	Due Month	Due Full Date	MonthKey
07/01/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	2017 Jul, 01	201707
07/02/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	2017 Jul, 02	201707
07/03/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	2017 Jul, 03	201707
07/04/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	2017 Jul, 04	201707
07/05/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	2017 Jul, 05	201707
07/06/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	2017 Jul, 06	201707
07/07/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	2017 Jul, 07	201707

Hvis du vil fuldføre designet af tabellen **Due Date** , kan du:

- Sortér kolonnen **Due Full Date** efter kolonnen **Due Date**.
- Sortér kolonnen **Due month** efter kolonnen **MonthKey**.
- Skjul kolonnen **MonthKey**.
- Opret et hierarki med navnet **Fiscal** med følgende niveauer:
 - Due Fiscal Year
 - Due Fiscal Quarter
 - Due Month
 - Due Full Date
- Markér tabellen **Due Date** som en datotabel ved hjælp af kolonnen **Due Date** .

Næste undermodul:

[Få mere at vide om rækkekontekst](#)

[Tidligere](#)

Få mere at vide om rækkekontekst

Fuldført100 XP

- 2 minutter

Nu, hvor du har oprettet beregnede kolonner, kan du få mere at vide om, hvordan deres formler evalueres.

Formlen for en beregnet kolonne evalueres for hver tabelrække. Desuden evalueres den inden for rækkekonteksten, hvilket betyder *den aktuelle række*. Se nærmere på definitionen af den beregnede kolonne **Due Fiscal Year**:

DAXKopier

Due Fiscal Year =

"FY"

```
& YEAR('Due Date'[Due Date])  
  
+ IF(  
    MONTH('Due Date'[Due Date]) <= 6,  
    1  
)
```

Når formelen evalueres for hver række, returnerer kolonnereferencen 'Due Date'[Due Date] kolonneværdien for *den pågældende række*. Du kan se, at Microsoft Excel har det samme koncept for arbejde med formler i [Excel-tabeller](#).

Rækkekonteksten udvides dog ikke ud over tabellen. Hvis formelen skal referere til kolonner i andre tabeller, har du to muligheder:

- Hvis tabellerne er relateret direkte eller indirekte, kan du bruge [RELATED](#) funktionen eller [RELATEDTABLE](#) DAX. Funktionen RELATED henter værdien på en-siden af relationen, mens RELATEDTABLE henter værdier på mange-siden. Funktionen RELATEDTABLE returnerer et tabelobjekt.
- Når tabellerne ikke er relateret, kan du bruge [LOOKUPVALUE](#) DAX-funktionen.

Prøv generelt at bruge funktionen, når det RELATED er muligt. Den fungerer normalt bedre end funktionen LOOKUPVALUE på grund af de måder, relationen og kolonnedataene gemmes og indekseres på.

Føj nu følgende definition af en beregnet kolonne til tabellen **Sales** :

DAXKopier

Discount Amount = **Sales[Order Quantity] * RELATED('Product'[List Price]) - Sales[SalesAmount]**

Officielle pris, herfra fratrækkes den reele pris, forskellen må være rabat

Definitionen af den beregnede kolonne fjører kolonnen **Discount Amount** til tabellen **Sales** . Power BI evaluerer formelen for den beregnede kolonne for hver række i tabellen **Sales** . Værdierne af kolonnerne **Order Quantity** og **Sales Amount** hentes i rækkekonteksten. **Da kolonnen Listepriis tilhører tabellen Product , RELATED er funktionen dog påkrævet for at hente listepriisværdien for salgsproduktet.**

Rækkekontekst bruges, når formler for beregnede kolonner evalueres. Den bruges også, når funktionsklassen *iteratorfunktioner* bruges. Iteratorfunktioner giver dig fleksibilitet til at oprette avancerede opsummeringer. Iteratorfunktioner beskrives i et senere modul.

Næste undermodul:

Vælg en teknik til tilføjelse af en kolonne

[Tidligere](#)

Vælg en teknik til tilføjelse af en kolonne

100 XP

- 2 minutter

Der er tre teknikker, du kan bruge til at føje kolonner til en modeltabel:

- Føj kolonner til en visning eller tabel (som en permanent kolonne), og hent dem derefter i Power Query. Denne mulighed giver kun mening, når datakilden er en relationsdatabase, og hvis du har færdigheder og tilladelser til at gøre det. Det er dog en god mulighed, da den understøtter nem vedligeholdelse og tillader genbrug af kolonnelogikken i andre modeller eller rapporter.
- Føj brugerdefinerede kolonner (ved hjælp af M) til Power Query forespørgsler.
- Føj beregnede kolonner (ved hjælp af DAX) til modeltabeller.

Uanset hvilken teknik du bruger, er resultatet det samme. **Rapportbrugere kan ikke bestemme en kolonnes oprindelse.** De tænker normalt ikke på, hvordan kolonnen blev oprettet, men på, om den leverer de rigtige data.

Når der er flere tilgængelige måder at tilføje en kolonne på, kan du overveje at bruge den fremgangsmåde, der passer bedst til dine færdigheder, og som understøttes af sproget (M eller DAX). **Præferencen er dog at tilføje brugerdefinerede kolonner i Power Query, når det er muligt, fordi de indlæses i modellen på en mere kompakt og optimal måde.**

Når du har brug for at føje en kolonne til en beregnet tabel, skal du sørge for at oprette en beregnet kolonne. Ellers anbefaler vi, at du kun bruger en beregnet kolonne, når den beregnede kolonneformel:

- Afhænger af opsummerede modeldata.
- **Skal bruge specialiserede modelleringsfunktioner, der kun er tilgængelige i DAX, f.eks. RELATED . funktionerne og RELATEDTABLE .** Specialiserede funktioner kan også omfatte [overordnet/underordnet-hierarkier i DAX](#), som er designet til at naturalisere en rekursiv relation til kolonner, f. eks. i en medarbejdertabel, hvor hver række lagrer en reference til chefrækken (der også findes i medarbejdertabellen).

Tjek din viden

1.Hvilket udsagn om beregnede tabeller er sandt?

Beregnede tabeller øger størrelsen på den semantiske model.

Beregnede tabeller lagrer data i modellen, og tilføjelse af dem resulterer i en større modelstørrelse.

Beregnede tabeller evalueres ved hjælp af rækkekontekst.

Beregnede tabeller oprettes i Power Query.

Beregnede tabeller kan ikke indeholde beregnede kolonner.

2.

Hvilket udsagn om beregnede kolonner er sandt?

Beregnede kolonner oprettes i Power Query-editor-vinduet.

Formler for beregnede kolonner evalueres ved hjælp af rækkekontekst.

Formler for beregnede kolonner evalueres ved hjælp af rækkekontekst.

Formler for beregnede kolonner kan kun referere til kolonner fra deres tabel.

Beregnede kolonner kan ikke være relateret til ikke-beregnede kolonner.

3.

Du er ved at udvikle en Power BI Desktop-model, der henter data fra en Excel-projektmappe. Projektmappen har en medarbejdertabel, der gemmer én række for hver medarbejder. Hver række har en reference til medarbejderens chef, som også er en række i medarbejdertabellen. Du skal føje flere kolonner til tabellen Employee i din model for at analysere løndata i organisationshierarkiet (f.eks. ledelsesniveau, lederniveau osv.). Hvilken teknik vil du bruge til at tilføje kolonnerne?

Føje permanente kolonner til en tabel ved hjælp af T-SQL.

Tilføje kolonneudtryk i en visning ved hjælp af T-SQL.

Tilføje beregnede kolonner ved hjælp af M.

Tilføje beregnede kolonner ved hjælp af DAX.

Du kan bruge DAX-overordnet/underordnet-funktioner til at naturalisere den rekursive (medarbejder chef) relation i kolonner.

Næste undermodul: Resumé

[Tidligere](#)

Resultater

Fortsæt det fremragende arbejde



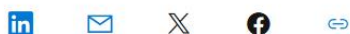
Føj beregnede tabeller og
kolonner til Power BI Desktop-
modeller

Du har opnået et resultat!

Tillykke, men hvad skal du gøre nu?

Lad os først dele dit resultat

Du afsætter tid til at lære noget nyt, så lad dit netværk tage del i din sejr!



<https://learn.microsoft.com/api/achievements/share/da-dk/bjarnehansen-7412/UYBTMSX3?sharingId=39714CB1FE13C67B>

**** Brug DAX-time intelligence-funktioner i Power BI Desktop-modeller

<https://learn.microsoft.com/da-dk/training/modules/dax-power-bi-time-intelligence/1-introduction>

- 1 t. 4 min.
- Modul
- 6 Enheder

Feedback

Øvet

Dataanalytiker

Appbygger

Power BI

Microsoft Power Platform

I slutningen af dette modul lærer du betydningen af time intelligence, og hvordan du føjer DAX-beregninger til time intelligence til din model.

Læringsmål

Når du har fuldført dette modul, vil du være i stand til følgende:

- Definere time intelligence.
- Bruge almindelige DAX-time intelligence-funktioner.
- Oprette nyttige intelligence-beregninger.

[Start](#)Tilføj

-
-
-

Forudsætninger

Du skal helst have erfaring med oprettelse af Microsoft Power BI Desktop-modeller og design af Power BI-rapportlayouts. Du bør også forstå, hvordan du opretter DAX-målinger (Data Analysis Expressions), og hvordan du arbejder med iteratorfunktioner og filterkontekst.

Dette modul er en del af disse læringsforløb

- [Brug DAX i Power BI Desktop](#)
- [Design and build tabular models](#)
- [Modeldata med Power BI](#)
- [Introduktion](#)1 min.
- [Brug DAX-time intelligence-funktioner](#)5 min.
- [Yderligere time intelligence-beregninger](#)5 min.
- [Øvelse – Create avancerede DAX-beregninger i Power BI Desktop](#)45 min.
- [Tjek din viden](#)3 min.
- [Resumé](#)5 min.

Introduktion

Fuldført100 XP

- 1 minut

Time intelligence relaterer til beregninger over tid. Det relaterer især til beregninger for datoer, måneder, kvartaler eller år og muligvis klokkeslæt. Det er sjældent, at du har brug for at beregne tid i timer, minutter eller sekunder.

I DAX-beregninger (Data Analysis Expressions) betyder time intelligence *ændring af filterkonteksten for datofiltre*.

I virksomheden Adventure Works begynder regnskabsåret f. eks. den 1. juli og slutter den 30. juni det efterfølgende år. De opretter et tabelvisual, der viser en månedlig indtægt og år-til-dato-indtægten (YTD).

Year	Revenue	Revenue YTD
 FY2018	\$23,860,891.17	\$23,860,891.17
2017 Jul	\$1,423,357.32	\$1,423,357.32
2017 Aug	\$2,057,902.45	\$3,481,259.78
2017 Sep	\$2,523,947.55	\$6,005,207.32
2017 Oct	\$561,681.48	\$6,566,888.80
2017 Nov	\$4,764,920.16	\$11,331,808.96
2017 Dec	\$596,746.56	\$11,928,555.52
2018 Jan	\$1,327,674.63	\$13,256,230.15
2018 Feb	\$3,936,463.31	\$17,192,693.45
2018 Mar	\$700,873.18	\$17,893,566.64
2018 Apr	\$1,519,275.24	\$19,412,841.88
2018 May	\$2,960,378.09	\$22,373,219.97
2018 Jun	\$1,487,671.19	\$23,860,891.17
 FY2019	\$34,070,108.50	\$34,070,108.50
2018 Jul	\$2,939,691.00	\$2,939,691.00
2018 Aug	\$3,964,801.20	\$6,904,492.20
2018 Sep	\$3,287,605.93	\$10,192,098.13

Filterkonteksten for **august 2017** indeholder de 31 datoer i august, som gemmes i tabellen **Date**. Men den beregnede år-til-dato-indtægt for **august 2017** anvender en anden filterkontekst. Det er den første dato i året til og med den sidste dato i filterkonteksten. I dette eksempel er det den 1. juli 2017 til og med den 31. august 2017.

Time intelligence-beregninger ændrer datofilterkontekster. De kan hjælpe dig med at besvare disse tidsrelaterede spørgsmål:

- Hvad er akkumuleringen af indtægter for året, kvartalet eller måneden?
- Hvilken indtægt var der i samme periode sidste år?
- Hvilken vækst i indtægten var der i samme periode sidste år?
- Hvor mange nye kunder afgav deres første ordre i hver måned?
- Hvad er værdien af den disponible lagerbeholdning for virksomhedens produkter?

I dette modul beskriver, hvordan du opretter time intelligence-målinger for at besvare disse spørgsmål.

Næste undermodul:

Brug DAX-time intelligence-funktioner

Næste

Brug DAX-time intelligence-funktioner

Fuldført 100 XP

- 5 minutter

DAX indeholder flere time intelligence-funktioner, der gør det nemmere at ændre filterkonteksten for datoer. Du kan skrive mange af disse intelligensformler ved hjælp af en CALCULATE funktion, der ændrer datofiltre, men det ville skabe mere arbejde.

Bemærk

Mange DAX time intelligence-funktioner bruger standarddatoperioder, især år, kvartaler og måneder. Hvis du har uregelmæssige tidsperioder (f. eks. regnskabsmåneder, der starter midt i en måned), eller hvis du har brug for at arbejde med uger eller tidsperioder (timer, minutter osv.), kan DAX time intelligence-funktionerne ikke bruges. I stedet **skal du bruge funktionen CALCULATE og angive håndlavede filtre for dato eller klokkeslæt.**

Krav til datotabel

Hvis du vil arbejde med DAX-time intelligence-funktioner, skal du opfylde kravet om at have mindst én *datotabel* i din model. En datotabel er en tabel, der overfylder følgende krav:

- Den skal have en kolonne med datatypen Dato (eller Dato/klokkeslæt), som kaldes *datokolonnen*.
- Datokolonnen må kun indeholde entydige værdier.
- Datokolonnen må ikke indeholde TOMME værdier.
- Datokolonnen må ikke have nogen manglende datoer.
- Datokolonnen skal strække sig over hele år. Et år er ikke nødvendigvis et kalenderår (januar-december).
- Datotabellen skal angives/markeres som en datotabel.

Du kan finde flere oplysninger i [Opret datotabeller i Power BI Desktop](#).

Opsummeringer over tid-funktioner, fx Year to Date, YTD

Én gruppe af DAX-time intelligence-funktioner bruges til opsummeringer over tid:

- [DATESYTD](#) – Returnerer en tabel med en enkelt kolonne, der indeholder datoer for år til dato (YTD) i den aktuelle filterkontekst. Denne gruppe indeholder [DATESMTD](#) også funktionerne og [DATESQTD](#) DAX for MTD (month-to-date) og kvartal-til-dato (QTD). Du kan overføre disse funktioner som filtre til DAX-funktionen [CALCULATE](#) .
- [TOTALYTD](#) – Evaluerer et udtryk for YTD i den aktuelle filterkontekst. De tilsvarende DAX-funktioner [TOTALQTD](#) [TOTALMTD](#) QTD og MTD er også inkluderet.

- [DATESBETWEEN](#) – Returnerer en tabel, der indeholder en kolonne med datoer, der starter med en given startdato og fortsætter indtil en bestemt slutdato.
- [DATESINPERIOD](#) – Returnerer en tabel, der indeholder en kolonne med datoer, der starter med en given startdato og fortsætter for det angivne antal intervaller.

Bemærk

Selvom funktionen TOTALYTD er nem at bruge, er du begrænset til at overføre ét filterudtryk. Hvis du har brug for at anvende flere filterudtryk, skal du bruge funktionen CALCULATE og derefter sende funktionen DATESYTD ind som et af filterudtrykene.

I følgende eksempel skal du oprette din første time intelligence-beregning, der bruger funktionen TOTALYTD. Syntaksen er følgende:

DAXKopier

TOTALYTD(<expression>, <dates>, [, <filter>][, <year_end_date>])

Funktionen kræver et udtryk, og, som det gælder for alle time intelligence-funktioner, en reference til datokolonnen i en markeret datotabel. Det er også muligt at angive et enkelt filterudtryk eller årets slutdato (kræves kun, når året ikke slutter den 31. december).

Download og åbn filen [Adventure Works DW 2020 M07. pbix](#). Føj derefter følgende målingsdefinition til tabellen **Sales**, der beregner ÅTD-indtægt. Formatér målingen som valuta med to decimalpladser.

DAXKopier

Revenue YTD =TOTALYTD([Revenue], 'Date'[Date], "6-30")

Værdien for årets slutdato repræsenterer "6-30" 30. juni. Så Q1: jul-sep

På **side 1** i rapporten skal du føje målingen **Revenue YTD** til matrixvisual'et. Bemærk, at der oprettes en opsummering af indtægtsbeløbene fra starten af året til og med den filtrerede måned.

Year	Revenue	Revenue YTD
FY2018	\$23,860,891.17	\$23,860,891.17
2017 Jul	\$1,423,357.32	\$1,423,357.32
2017 Aug	\$2,057,902.45	\$3,481,259.78
2017 Sep	\$2,523,947.55	\$6,005,207.32
2017 Oct	\$561,681.48	\$6,566,888.80
2017 Nov	\$4,764,920.16	\$11,331,808.96
2017 Dec	\$596,746.56	\$11,928,555.52
2018 Jan	\$1,327,674.63	\$13,256,230.15
2018 Feb	\$3,936,463.31	\$17,192,693.45
2018 Mar	\$700,873.18	\$17,893,566.64
2018 Apr	\$1,519,275.24	\$19,412,841.88
2018 May	\$2,960,378.09	\$22,373,219.97
2018 Jun	\$1,487,671.19	\$23,860,891.17

Fiscal Year	Sum of SalesAmount_sum	Revenue	Sum of SalesAmount	Revenue YoYPct	Revenue YTD
FY2018	23.860.891,17	23.860.891,17	23.860.891,17		23.860.891,17
FY2018 Q1	6.005.207,32	6.005.207,32	6.005.207,32		6.005.207,32
1. juli 2017	1.423.357,32	1.423.357,32	1.423.357,32		1.423.357,32
1. august 2017	2.057.902,45	2.057.902,45	2.057.902,45		3.481.259,78
1. september 2017	2.523.947,55	2.523.947,55	2.523.947,55		6.005.207,32
FY2018 Q2	5.923.348,19	5.923.348,19	5.923.348,19		11.928.555,52
FY2018 Q3	5.965.011,12	5.965.011,12	5.965.011,12		17.893.566,64
FY2018 Q4	5.967.324,53	5.967.324,53	5.967.324,53		23.860.891,17
FY2019	34.070.108,50	34.070.108,50	34.070.108,50	42,79	34.070.108,50
Total	109.809.274,20	109.809.274,20	109.809.274,20	0,00	

Sammenligninger over tidsrum, fx Revenu i måned med samme måned året før

En anden gruppe af DAX-time intelligence-funktioner bruges til flytning af tidsperioder:

- [DATEADD](#) – Returnerer en tabel, der indeholder en kolonne med datoer, der er flyttet enten frem eller tilbage i tiden med det angivne antal intervaller fra datoerne i den aktuelle filterkontekst.
- [PARALLELPERIOD](#) – Returnerer en tabel, der indeholder en kolonne med datoer, der repræsenterer en periode, der er parallel med datoerne i den angivne datokolonne i den aktuelle filterkontekst, hvor datoerne er flyttet et antal intervaller enten fremad i tiden eller tilbage i tiden.
- [SAMEPERIODLASTYEAR](#) – Returnerer en tabel, der indeholder en kolonne med datoer, der flyttes et år tilbage i tiden fra datoerne i den angivne datokolonne i den aktuelle filterkontekst.
- Mange hjælpe-DAX-funktioner til bagud- eller fremadnavigation i bestemte tidsperioder, som alle returnerer en tabel med datoer. Disse hjælpefunktioner omfatter [NEXTDAY](#), [NEXTMONTH](#), [NEXTQUARTER](#), [NEXTYEAR](#) og [PREVIOUSDAY](#), [PREVIOUSMONTH](#), [PREVIOUSQUARTER](#) og [PREVIOUSYEAR](#).

Nu skal du føje en måling til tabellen **Sales**, der beregner indtægten for det foregående år ved hjælp af funktionen **SAMEPERIODLASTYEAR**. Formatér målingen som valuta med to decimalpladser.

DAXKopier

Revenue PY =

VAR RevenuePriorYear = CALCULATE([Revenue], SAMEPERIODLASTYEAR('Date'[Date]))

RETURN

RevenuePriorYear

Føj derefter målingen **Revenue PY** til matrixvisual'et. Bemærk, at den giver resultater, der ligner det forrige års indtægtsbeløb.

Year	Revenue	Revenue YTD	Revenue PY
FY2018	\$23,860,891.17	\$23,860,891.17	
2017 Jul	\$1,423,357.32	\$1,423,357.32	
2017 Aug	\$2,057,902.45	\$3,481,259.78	
2017 Sep	\$2,523,947.55	\$6,005,207.32	
2017 Oct	\$561,681.48	\$6,566,888.80	
2017 Nov	\$4,764,920.16	\$11,331,808.96	
2017 Dec	\$596,746.56	\$11,928,555.52	
2018 Jan	\$1,327,674.63	\$13,256,230.15	
2018 Feb	\$3,936,463.31	\$17,192,693.45	
2018 Mar	\$700,873.18	\$17,893,566.64	
2018 Apr	\$1,519,275.24	\$19,412,841.88	
2018 May	\$2,960,378.09	\$22,373,219.97	
2018 Jun	\$1,487,671.19	\$23,860,891.17	
FY2019	\$34,070,108.50	\$34,070,108.50	\$23,860,891.17
2018 Jul	\$2,939,691.00	\$2,939,691.00	\$1,423,357.32
2018 Aug	\$3,964,801.20	\$6,904,492.20	\$2,057,902.45
2018 Sep	\$3,287,605.93	\$10,192,098.13	\$2,523,947.55
2018 Oct	\$2,157,287.40	\$12,349,385.53	\$561,681.48
2018 Nov	\$3,611,092.23	\$15,960,477.76	\$4,764,920.16
2018 Dec	\$2,624,078.39	\$18,584,556.15	\$596,746.56
2019 Jan	\$1,847,691.91	\$20,432,248.06	\$1,327,674.63
2019 Feb	\$2,829,361.64	\$23,261,609.70	\$3,936,463.31
2019 Mar	\$2,092,434.35	\$25,354,044.05	\$700,873.18
2019 Apr	\$2,405,970.99	\$27,760,015.05	\$1,519,275.24
2019 May	\$3,459,444.04	\$31,219,459.08	\$2,960,378.09
2019 Jun	\$2,850,649.42	\$34,070,108.50	\$1,487,671.19

Derefter skal du ændre målingen ved at omdøbe den til **Revenue YoY %** og derefter opdatere delsatningen RETURN for at beregne ændringsforholdet. Sørg for at ændre formatet til en procentdel med to decimaler.

DAXKopiér

Revenue YoY % =

VAR RevenuePriorYear = CALCULATE([Revenue], SAMEPERIODLASTYEAR('Date'[Date]))

RETURN

DIVIDE(

[Revenue] - RevenuePriorYear,

RevenuePriorYear

)

Bemærk, at målingen **Revenue YoY %** viser et forholdstal for ændringsfaktoren for det forrige års månedlige indtægt. Juli 2018 repræsenterer f. eks. en 106,53 % *forøgelse* i løbet af det forrige års månedlige indtægt, og november 2018 repræsenterer en 24,22 % *reduktion* for af det forrige års månedlige indtægt.

Year	Revenue	Revenue YTD	Revenue YoY %
FY2018	\$23,860,891.17	\$23,860,891.17	
2017 Jul	\$1,423,357.32	\$1,423,357.32	
2017 Aug	\$2,057,902.45	\$3,481,259.78	
2017 Sep	\$2,523,947.55	\$6,005,207.32	
2017 Oct	\$561,681.48	\$6,566,888.80	
2017 Nov	\$4,764,920.16	\$11,331,808.96	
2017 Dec	\$596,746.56	\$11,928,555.52	
2018 Jan	\$1,327,674.63	\$13,256,230.15	
2018 Feb	\$3,936,463.31	\$17,192,693.45	
2018 Mar	\$700,873.18	\$17,893,566.64	
2018 Apr	\$1,519,275.24	\$19,412,841.88	
2018 May	\$2,960,378.09	\$22,373,219.97	
2018 Jun	\$1,487,671.19	\$23,860,891.17	
FY2019	\$34,070,108.50	\$34,070,108.50	42.79%
2018 Jul	\$2,939,691.00	\$2,939,691.00	106.53%
2018 Aug	\$3,964,801.20	\$6,904,492.20	92.66%
2018 Sep	\$3,287,605.93	\$10,192,098.13	30.26%
2018 Oct	\$2,157,287.40	\$12,349,385.53	284.08%
2018 Nov	\$3,611,092.23	\$15,960,477.76	-24.22%
2018 Dec	\$2,624,078.39	\$18,584,556.15	339.73%
2019 Jan	\$1,847,691.91	\$20,432,248.06	39.17%
2019 Feb	\$2,829,361.64	\$23,261,609.70	-28.12%
2019 Mar	\$2,092,434.35	\$25,354,044.05	198.55%
2019 Apr	\$2,405,970.99	\$27,760,015.05	58.36%
2019 May	\$3,459,444.04	\$31,219,459.08	16.86%
2019 Jun	\$2,850,649.42	\$34,070,108.50	91.62%

Bemærk

Målingen **Revenue YoY %** viser en god brug af DAX-variabler. Målingen forbedrer formlens læsbarhed og giver dig mulighed for at udføre en enhedstest af en del af målingslogikken (ved at returnere værdien af variabelen **RevenuePriorYear**). Målingen er desuden en optimal formel, da det ikke er nødvendigt at hente værdien af det forrige års indtægt to gange. Når delsætningen har gemt den én gang i en variabel, bruges den RETURN to gange til variabelværdien.

Næste undermodul:

Yderligere time intelligence-beregninger, antal forekomster i tidsrum

[Tidligere](#)

Yderligere time intelligence-beregninger

Fuldført 100 XP

- 5 minutter

Der findes andre DAX-time intelligence-funktioner, som bruges til returnering af en enkelt dato. Du får mere at vide om disse funktioner ved at anvende dem i to forskellige scenarier.

DAX-funktionerne [FIRSTDATE](#) og [LASTDATE](#) returnerer den første og sidste dato i den aktuelle filterkontekst for den angivne kolonne med datoer.

Beregn nye forekomster

En anden brug af time intelligence-funktioner er at tælle nye forekomster. I følgende eksempel kan du se, **hvordan du kan beregne antallet af nye kunder i en bestemt periode**. En kunde registreres som ny i den tidsperiode, hvor den pågældende foretog sit første køb.

Din første opgave er at føje følgende måling til tabellen **Sales**, der tæller antallet af særskilte kunders *liv-til-dato* (LTD). **Life-to-date betyder fra starttidspunktet til den sidste dato i filterkonteksten; er det aug2017, er det åbenbart helt fra start til aug2017**. Formatér målingen som et heltal ved hjælp af tusindtalsseparatoren.

DAXKopier

```
Customers LTD =
VAR CustomersLTD =
    CALCULATE(
        DISTINCTCOUNT(Sales[CustomerKey]),
        DATESBETWEEN(
            'Date'[Date],
            BLANK(),
            MAX('Date'[Date])
        ),
        'Sales Order'[Channel] = "Internet"
    )
RETURN
    CustomersLTD
```

'Date'[Date] er den parameter der indeholder dato,

Føj målingen **Customer LTD** til matrixvisual'et. Bemærk, at resultatet er de forskellige kunder life-to-date (LTD) indtil udgangen af hver måned.

Year	Revenue	Revenue YTD	Revenue YoY %	Customers LTD
FY2018	\$23,860,891.17	\$23,860,891.17		2,459
2017 Jul	\$1,423,357.32	\$1,423,357.32		289
2017 Aug	\$2,057,902.45	\$3,481,259.78		448
2017 Sep	\$2,523,947.55	\$6,005,207.32		609
2017 Oct	\$561,681.48	\$6,566,888.80		783
2017 Nov	\$4,764,920.16	\$11,331,808.96		1,013
2017 Dec	\$596,746.56	\$11,928,555.52		1,201
2018 Jan	\$1,327,674.63	\$13,256,230.15		1,394
2018 Feb	\$3,936,463.31	\$17,192,693.45		1,571
2018 Mar	\$700,873.18	\$17,893,566.64		1,790
2018 Apr	\$1,519,275.24	\$19,412,841.88		1,992
2018 May	\$2,960,378.09	\$22,373,219.97		2,214
2018 Jun	\$1,487,671.19	\$23,860,891.17		2,459

Funktionen DATESBETWEEN returnerer en tabel, der indeholder en kolonne med datoer, der starter med en given startdato og fortsætter indtil en bestemt slutdato. Når startdatoen er BLANK, som her, bruges den første dato i kolonnen Date, altså fra første dato i virksomheden ca okt-2017. Modsat bruges den sidste dato i kolonnen Date, når slutdatoen er BLANK. I dette tilfælde bestemmes slutdatoen af funktionen MAX, der returnerer den sidste dato i filterkonteksten. Hvis august måned 2017 er i filterkontekst, returnerer funktionen MAX derfor den 31. august 2017, og funktionen DATESBETWEEN returnerer alle datoer helt fra start til den 31. august 2017. talværdi akkumuleres altså op

Derefter skal du ændre målingen ved at omdøbe den til **New Customers** og ved at tilføje en variabel mere for at lagre antallet af forskellige kunder *før* tidsperioden i filterkonteksten. Delsætningen RETURN trækker nu denne værdi fra LTD-kunder for at producere et resultat, som er antallet af nye kunder i tidsperioden.

DAXKopier

New Customers =

VAR CustomersLTD =

```
CALCULATE(
    DISTINCTCOUNT(Sales[CustomerKey]),
    DATESBETWEEN(
        'Date'[Date],
        BLANK(),
        MAX('Date'[Date])
    ),
    'Sales Order'[Channel] = "Internet"
)
```

VAR CustomersPrior =

```
CALCULATE(
    DISTINCTCOUNT(Sales[CustomerKey]),
    DATESBETWEEN(
        'Date'[Date],
        BLANK(),
        MIN('Date'[Date]) - 1
    )
)
```



```

),
'Sales Order'[Channel] = "Internet"
)
RETURN
CustomersLTD - CustomersPrior

```

Year	Revenue	Revenue YTD	Revenue YoY %	New Customers
FY2018	\$23,860,891.17	\$23,860,891.17		2,459
2017 Jul	\$1,423,357.32	\$1,423,357.32		289
2017 Aug	\$2,057,902.45	\$3,481,259.78		159
2017 Sep	\$2,523,947.55	\$6,005,207.32		161
2017 Oct	\$561,681.48	\$6,566,888.80		174
2017 Nov	\$4,764,920.16	\$11,331,808.96		230
2017 Dec	\$596,746.56	\$11,928,555.52		188
2018 Jan	\$1,327,674.63	\$13,256,230.15		193
2018 Feb	\$3,936,463.31	\$17,192,693.45		177
2018 Mar	\$700,873.18	\$17,893,566.64		219
2018 Apr	\$1,519,275.24	\$19,412,841.88		202
2018 May	\$2,960,378.09	\$22,373,219.97		222
2018 Jun	\$1,487,671.19	\$23,860,891.17		245

For variablen **CustomersPrior** skal du bemærke, at funktionen DATESBETWEEN indeholder datoer indtil den første dato i filterkonteksten *minus* én. Da Microsoft Power BI internt gemmer datoer som tal, kan du addere eller subtrahere tal for at flytte en dato.

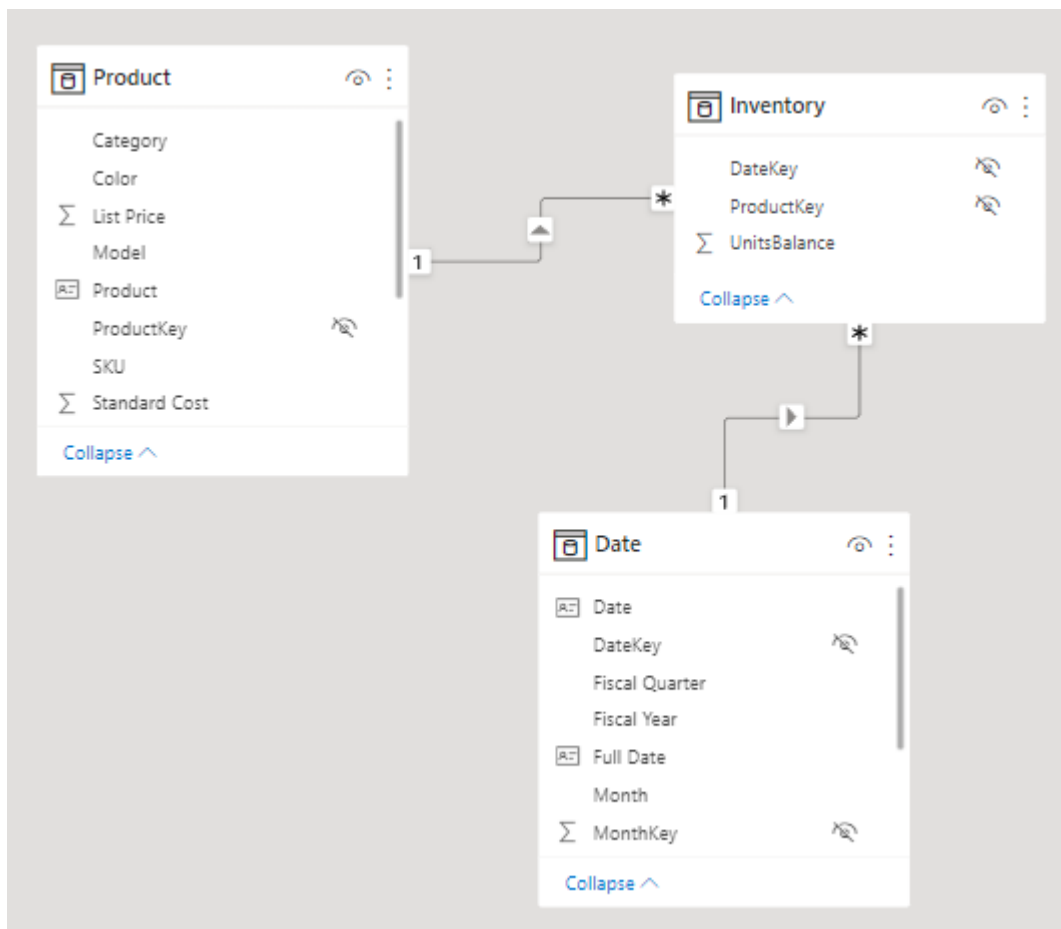
Snapshotberegninger

I nogle tilfælde gemmes faktadataene som snapshot i tid. Almindelige eksempler omfatter lagerbeholdningsniveauer eller kontosal-di. Et snapshot med værdier indlæses i tabellen med regelmæssige mellemrum.

Når du opsummerer snapshotværdier (f. eks. lagerbeholdningsniveauer), kan du opsummere værdier på tværs af alle dimensioner, undtagen dato. Hvis du tilføjer et lagerniveau på tværs af produktkategorier, får du en meningsfuld oversigt, men det gør du ikke ved at tilføje lagerniveau på tværs af datoer. Det er ikke hensigtsmæssigt at føje mere end ét lagerniveau til dagens lagerniveau, (medmindre du ønsker at udregne gennemsnittet af resultatet).

Når du opsummerer snapshot tabeller, kan formler for målinger være afhængige af DAX-time intelligence-funktioner for at gennemtvinge et enkelt datofilter.

I det følgende eksempel skal du udforske et scenarie for virksomheden Adventure Works. Skift til modelvisning, og vælg modeldiagrammet **Inventory**.



Bemærk, at diagrammet viser tre tabeller: **Produkt**, **Dato** og **Lager**. I tabellen **Inventory** gemmes snapshots af enhedssaldi for hver dato og hvert produkt. Det er vigtigt, at tabellen ikke har manglende datoer og ingen dublerede poster for et produkt på samme dato. Desuden gemmes den sidste snapshotpost for datoen den 15. juni 2020.

Skift nu til rapportvisning, og vælg **side 2** i rapporten. Føj kolonnen **UnitsBalance** i tabellen **Inventory** til matrixvisual'et. Standardopsummeringen er angivet til at opsummere værdier.

ProductKey	DateKey	UnitsBalance
210	20170701	438
210	20170702	438
210	20170703	438
210	20170704	438
210	20170705	438
210	20170706	438
210	20170707	438
210	20170708	438
210	20170709	438
210	20170710	438

Det er altså lagerbeholdningen af hvert produkt, der gemmes.

FY2020 Mountain-200 Bike Stock													
Product	2019 Jul	2019 Aug	2019 Sep	2019 Oct	2019 Nov	2019 Dec	2020 Jan	2020 Feb	2020 Mar	2020 Apr	2020 May	2020 Jun	Total
Mountain-200 Black, 38	4,884	4,649	4,784	4,873	4,605	4,943	5,208	4,872	5,208	5,040	5,208	2,520	56,794
Mountain-200 Black, 42	5,248	5,175	5,246	5,382	5,200	5,415	5,177	4,843	5,177	5,010	5,177	2,505	59,555
Mountain-200 Black, 46	5,523	5,617	5,287	5,419	5,273	5,663	5,890	5,510	5,890	5,700	5,890	2,850	64,512
Mountain-200 Silver, 38	5,412	5,190	5,114	5,245	5,220	5,406	5,890	5,510	5,890	5,700	5,890	2,850	63,317
Mountain-200 Silver, 42	5,252	5,262	5,043	5,266	5,072	5,208	5,022	4,698	5,022	4,860	5,022	2,430	58,157
Mountain-200 Silver, 46	5,004	5,019	4,960	5,012	4,847	5,231	5,053	4,727	5,053	4,890	5,053	2,445	57,294
Total	31,323	30,912	30,434	31,197	30,217	31,866	32,240	30,160	32,240	31,200	32,240	15,600	359,629

FY2020 Mountain-200 Bike Stock						
Month	2019 Jul		2019 Aug		2019 Sep	
Product	UnitsBalance	Stock on Hand	UnitsBalance	Stock on Hand	UnitsBalance	Stock on Hand
Mountain-200 Black, 38	4,884	151	4,649	171	4,784	99
Mountain-200 Black, 42	5,248	165	5,175	186	5,246	116
Mountain-200 Black, 46	5,523	182	5,617	184	5,287	131
Mountain-200 Silver, 38	5,412	171	5,190	173	5,114	129
Mountain-200 Silver, 42	5,252	177	5,262	169	5,043	109
Mountain-200 Silver, 46	5,004	181	5,019	158	4,960	126
Total	31,323	1027	30,912	1041	30,434	710

31.8 findes 171, Stock on Hold er beholdningen sidste dag i måneden, $4649/31=149$ er gennemsnits stok i aug. Der laves sum over hver dag i måneden med UnitBlanche

Denne visualkonfiguration er et eksempel på, hvordan en snapshotværdi ikke opsummeres. Det giver ikke meningsfuldt resultat at lægge de daglige snapshots sammen. Derfor skal du fjerne feltet **UnitsBalance** fra matrixvisual'et.

Nu skal du føje en måling til tabellen **Inventory**, der opsummerer værdien **UnitsBalance** for en enkelt dato. Datoen er den sidste dato i hver tidsperiode. Det opnås ved hjælp af funktionen LASTDATE. Formatér målingen som et heltal med tusindtalsseparatoren.

DAXKopier

Stock on Hand =

```
CALCULATE(  
    SUM(Inventory[UnitsBalance]),  
    LASTDATE('Date'[Date])  
)
```

Bemærk

Bemærk, at målingsformlen bruger funktionen SUM . Der skal bruges en aggregeringsfunktion (enkelt-målinger som her, tillader ikke direkte referencer til kolonner), men da der kun findes én række for hvert produkt for hver dato, fungerer funktionen SUM kun over en enkelt række, nemlig lastdate

Føj målingen **Stock on Hand** til matrixvisual'et. Værdien for hvert produkt er nu baseret på saldoen for den senest registrerede enhed for hver måned.

FY2020 Mountain-200 Bike Stock													
Product	2019 Jul	2019 Aug	2019 Sep	2019 Oct	2019 Nov	2019 Dec	2020 Jan	2020 Feb	2020 Mar	2020 Apr	2020 May	2020 Jun	Total
Mountain-200 Black, 38	151	171	99	172	30	168	168	168	168	168	168		
Mountain-200 Black, 42	165	186	116	176	76	167	167	167	167	167	167		
Mountain-200 Black, 46	182	184	131	172	111	190	190	190	190	190	190	190	
Mountain-200 Silver, 38	171	173	129	190	85	190	190	190	190	190	190		
Mountain-200 Silver, 42	177	169	109	170	120	162	162	162	162	162	162		
Mountain-200 Silver, 46	181	158	126	178	88	163	163	163	163	163	163		
Total	1,027	1,041	710	1,058	510	1,040	1,040	1,040	1,040	1,040	1,040		

Målingen returnerer BLANKs for juni 2020, da der ikke findes nogen poster for den sidste dato i juni. Ifølge dataene er det endnu ikke sket.

Filtrering efter den sidste dato i filter konteksten har indbyggede problemer: Der findes muligvis ikke en registreret dato, da den endnu ikke har fundet sted, eller muligvis fordi der ikke er registreret lagerbeholdninger i weekender.

Dit næste trin er at justere formelen for målingen for at bestemme den sidste dato, som har et resultat, der ikke er BLANK, og derefter filtrere efter den pågældende dato. Du kan opnå denne opgave ved hjælp af DAX-funktionen [LASTNONBLANK](#) .

Brug følgende målingsdefinition til at ændre målingen **Stock on Hand**.

DAXKopier

Stock on Hand =

```
CALCULATE(  
    SUM(Inventory[UnitsBalance]),  
    LASTNONBLANK(  
        'Date'[Date],  
        CALCULATE(SUM(Inventory[UnitsBalance]))  
    )  
)
```

Grønne er filtreringsmekanismen for **SUM(Inventory[UnitsBalance])**

I matrixvisual'et kan du se værdierne for juni 2020 og totalen (repræsenterer hele året).

2020 Apr	2020 May	2020 Jun	Total
168	168	168	168
167	167	167	167
190	190	190	190
190	190	190	190
162	162	162	162
163	163	163	163
1,040	1,040	1,040	1,040

Funktionen **LASTNONBLANK** er en iteratorfunktion. Den returnerer den sidste dato, der opretter et resultat, der ikke er BLANK. Den opnår dette resultat ved at gentage alle datoer i filterkonteksten i *faldende kronologisk rækkefølge*. (Omvendt gentages gentagelserne **FIRSTNONBLANK** i stigende kronologisk rækkefølge). For hver dato evalueres det overførte udtryk. Når der registreres et ikke-BLANK resultat, returnerer funktionen datoen. Denne dato bruges derefter til at filtrere funktionen **CALCULATE**.

Bemærk

Funktionen **LASTNONBLANK** evaluerer udtrykket i rækkekonteksten. Funktionen **CALCULATE** skal bruges til at overføre rækkekonteksten for at filtrere konteksten for at evaluere udtrykket korrekt.

Du bør nu skjule kolonnen **UnitsBalance** i tabellen **Inventory**. Det forhindrer rapportforfattere i at opsummere snapshots af enhedssaldi på en uhensigtsmæssig måde.

Næste undermodul:

Øvelse – Create avancerede DAX-beregninger i Power BI Desktop

[Tidligere](#)

Create Advanced DAX Calculations in Power BI Desktop

Lab story

In this lab, you'll create measures with DAX expressions involving filter context manipulation.

In this lab you learn how to:

- Use the **CALCULATE()** function to manipulate filter context
- Use Time Intelligence functions

This lab should take approximately 45 minutes.

Get started

To complete this exercise, first open a web browser and enter the following URL to download the zip folder:

<https://github.com/MicrosoftLearning/PL-300-Microsoft-Power-BI-Data-Analyst/raw/Main/Allfiles/Labs/05-create-dax-calculations-in-power-bi-desktop-advanced/05-advanced-dax.zip>

Extract the folder to the C:\Users\Student\Downloads\05-advanced-dax folder.

Open the **05-Starter-Sales Analysis.pbix** file.

Note: You can dismiss the sign-in by selecting **Cancel**. Close any other informational windows. Select **Apply Later**, if prompted to apply changes.

Create a matrix visual

In this task, you'll create a matrix visual to support testing your new measures.

1. In Power BI Desktop, **Report view**, create a new report page.
2. On **Page 3**, add a matrix visual.
3. Resize the matrix visual to fill the entire page.
4. To configure the matrix visual fields, from the **Data** pane, drag the **Region | Regions** hierarchy, and drop it inside the visual.

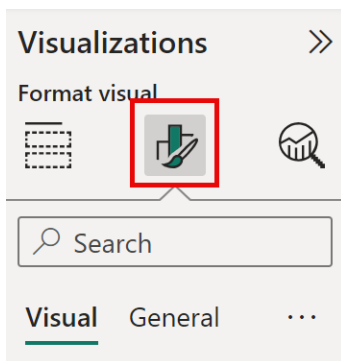
The labs use a shorthand notation to reference a field or hierarchy. It will look like this: **Region | Regions**. In this example, **Region** is the table name and **Regions** is the hierarchy name.

5. Add also the **Sales | Sales** field to the Values well.
6. To expand the entire hierarchy, at the top-right of the matrix visual, select the forked-double arrow icon twice.

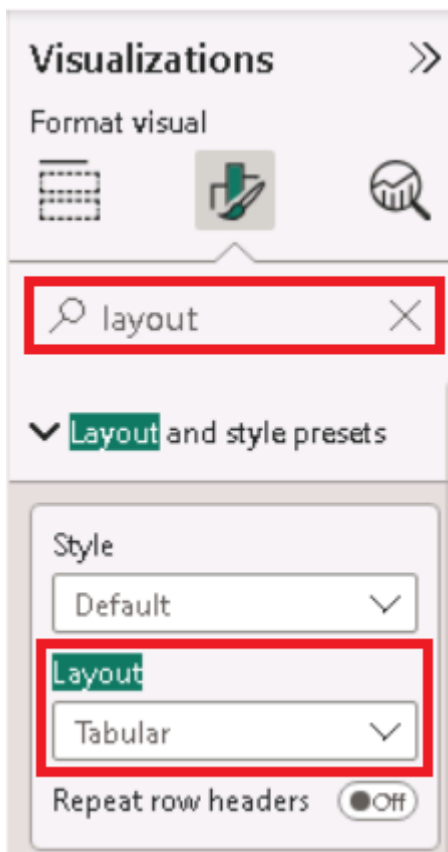
folder ud og viser overordnede niveauer

x2

7. To format the visual, in the **Visualizations** pane, select the **Format** pane. Ser ikke det



8. In the **Search** box, enter **Layout**.
9. Set the **Layout** property to **Tabular**.



10. Verify that the matrix visual now has 4 column headers.

Group	Country	Region	Sales
Europe	France	France	\$4,527,840
		Total	\$4,527,840
	Germany	Germany	\$1,877,743
		Total	\$1,877,743
	United Kingdom	United Kingdom	\$3,883,043
		Total	\$3,883,043
	Total		\$10,288,626
North America	Canada	Canada	\$13,875,633
		Total	\$13,875,633
	United States	Central	\$7,633,387
		Northeast	\$6,715,354
		Northwest	\$12,004,822
		Southeast	\$7,638,607
		Southwest	\$18,001,116
		Total	\$51,993,286
	Total		\$65,868,919
	Pacific	Australia	Australia
Total			\$1,391,025
Total		\$1,391,025	
Total			\$77,548,570

At Adventure Works, the sales regions are organized into groups, countries, and regions. All countries—except the United States—have just one region, which is named after the country. As the United States is such a large sales territory, it's divided into five sales regions.

You'll create several measures in this exercise, and then test them by adding them to the matrix visual.

Manipulate filter context

In this task, you'll create several measures with DAX expressions that use the CALCULATE() function to manipulate filter context.

The CALCULATE() function

is a powerful function used to manipulate the filter context. The first argument takes an expression or a measure (a measure is just a named expression). Subsequent arguments allow modifying the filter context.

1. Add a measure to the **Sales** table, based on the following expression:

Note: For your convenience, all DAX definitions in this lab can be copied from the **C:\Users\Student\Downloads\05-advanced-dax\Snippets.txt** file.

codeCopy

Sales All Region = CALCULATE(SUM(Sales[Sales]), REMOVEFILTERS(Region))

The REMOVEFILTERS() function removes active filters. It can take either no arguments, or a table, a column, or multiple columns as its argument.

In this formula, the measure evaluates the sum of the **Sales** column in a modified filter context, which removes any filters applied to the columns of the **Region** table.

2. Add the **Sales All Region** measure to the matrix visual.

Group	Country	Region	Sales	Sales All Region
Europe	France	France	\$4,527,840	\$77,548,570.2
		Total	\$4,527,840	\$77,548,570.2
	Germany	Germany	\$1,877,743	\$77,548,570.2
		Total	\$1,877,743	\$77,548,570.2
	United Kingdom	United Kingdom	\$3,883,043	\$77,548,570.2
		Total	\$3,883,043	\$77,548,570.2
	Total		\$10,288,626	\$77,548,570.2
North America	Canada	Canada	\$13,875,633	\$77,548,570.2
		Total	\$13,875,633	\$77,548,570.2
	United States	Central	\$7,633,387	\$77,548,570.2
		Northeast	\$6,715,354	\$77,548,570.2
		Northwest	\$12,004,822	\$77,548,570.2
		Southeast	\$7,638,607	\$77,548,570.2
		Southwest	\$18,001,116	\$77,548,570.2
		Total	\$51,993,286	\$77,548,570.2
	Total		\$65,868,919	\$77,548,570.2
	Pacific	Australia	Australia	\$1,391,025
Total			\$1,391,025	\$77,548,570.2
Total		\$1,391,025	\$77,548,570.2	
Total			\$77,548,570	\$77,548,570.2

3. Notice that the **Sales All Region** measure computes the total of all region sales for each region, country (subtotal) and group (subtotal).

The new measure is yet to deliver a useful result. When the sales for a group, country, or region is divided by this value it will produce a useful ratio known as “percent of grand total”.

4. In the **Data** pane, ensure that the **Sales All Region** measure is selected (when selected, it will have a dark gray background), and then in the formula bar, replace the measure name and formula with the following formula:

*Tip: To replace the existing formula, first copy the snippet. Then, select inside the formula bar and press **Ctrl+A** to select all text. Then, press **Ctrl+V** to paste the snippet to overwrite the selected text. Then press **Enter**.*

codeCopy

```
Sales % All Region =
DIVIDE(
SUM(Sales[Sales]),
CALCULATE(
SUM(Sales[Sales]),
```

```
REMOVEFILTERS(Region)
)
)
```

Bedre :

Sales % All Region =

DIVIDE(SUM(Sales[Sales]);[Sales All Region])*100

The measure has been renamed to accurately reflect the updated formula. The *DIVIDE()* function divides the **Sales** measure (not modified by filter context) by the **Sales** measure in a modified context, which removes any filters applied to the **Region** table.

5. In the matrix visual, notice that the measure has been renamed and that a different value now appears for each group, country, and region.
6. Format the **Sales % All Region** measure as a percentage with two decimal places.
7. In the matrix visual, review the **Sales % All Region** measure values.

Group	Country	Region	Sales	Sales % All Region
Europe	France	France	\$4,527,840	5.84 %
		Total	\$4,527,840	5.84 %
	Germany	Germany	\$1,877,743	2.42 %
		Total	\$1,877,743	2.42 %
	United Kingdom	United Kingdom	\$3,883,043	5.01 %
		Total	\$3,883,043	5.01 %
Total			\$10,288,626	13.27 %

8. Add another measure to the **Sales** table, based on the following expression, and format as a percentage:

codeCopy

Sales % Country =

```
DIVIDE(
SUM(Sales[Sales]),
CALCULATE(
SUM(Sales[Sales]),
REMOVEFILTERS(Region[Region])
)
)
```

9. Notice that the **Sales % Country** measure formula differs slightly from the **Sales % All Region** measure formula.

The difference is that the denominator modifies the filter context by removing filters on the **Region** column of the **Region** table, not all columns of the **Region** table. It means that any filters applied to the group or country columns are preserved. It will achieve a result that represents the sales as a percentage of country.

10. Add the **Sales % Country** measure to the matrix visual.
11. Notice that only the United States' regions produce a value that isn't 100%.

☐ North America	☐ Canada	Canada	\$13,875,633	17.89 %	100.00 %
		Total	\$13,875,633	17.89 %	100.00 %
	☐ United States	Central	\$7,633,387	9.84 %	14.68 %
		Northeast	\$6,715,354	8.66 %	12.92 %
		Northwest	\$12,004,822	15.48 %	23.09 %
		Southeast	\$7,638,607	9.85 %	14.69 %
		Southwest	\$18,001,116	23.21 %	34.62 %
		Total	\$51,993,286	67.05 %	100.00 %

You may recall that only the United States has multiple regions. All other countries comprise a single region, which explains why they're all 100%.

- To improve the readability of this measure in visual, overwrite the **Sales % Country** measure with this improved formula.

codeCopy

```

Sales % Country =
IF(
    ISINSCOPE(Region[Region]),
    DIVIDE(
        SUM(Sales[Sales]),
        CALCULATE(
            SUM(Sales[Sales]),
            REMOVEFILTERS(Region[Region])
        )
    )
)

```

The IF() function uses the ISINSCOPE() function to test whether the region column is the level in a hierarchy of levels. When true, the DIVIDE() function is evaluated. *When false, a blank value is returned because the region column isn't in scope.* Total eller tomt er ikke i scope, så giver tomt felt

SalesTerritoryKey	Region	Country	Group
1	Northwest	United States	North America
2	Northeast	United States	North America
3	Central	United States	North America
4	Southwest	United States	North America
5	Southeast	United States	North America
6	Canada	Canada	North America
7	France	France	Europe
8	Germany	Germany	Europe
9	Australia	Australia	Pacific
10	United Kingdom	United Kingdom	Europe

Regiontabel

- Notice that the **Sales % Country** measure now only returns a value when a region is in scope.

Group	Country	Region	Sales	Sales % All Region	Sales % Country
Europe	France	France	\$4,527,840	5.84 %	100.00 %
		Total	\$4,527,840	5.84 %	
	Germany	Germany	\$1,877,743	2.42 %	100.00 %
		Total	\$1,877,743	2.42 %	
	United Kingdom	United Kingdom	\$3,883,043	5.01 %	100.00 %
		Total	\$3,883,043	5.01 %	

14. Add another measure to the **Sales** table, based on the following expression, and format as a percentage:

codeCopy

```
Sales % Group =
DIVIDE(
SUM(Sales[Sales]),
CALCULATE(
SUM(Sales[Sales]),
REMOVEFILTERS(
Region[Region],
Region[Country]
)
)
)
```

To achieve sales as a percentage of group, two filters can be applied to effectively remove the filters on two columns.

15. Add the **Sales % Group** measure to the matrix visual.

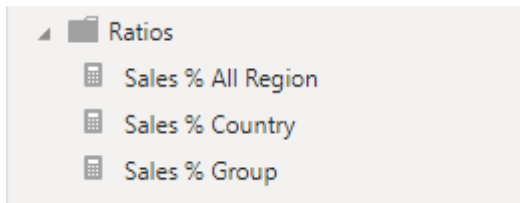
16. To improve the readability of this measure in visual, overwrite the **Sales % Group** measure with this improved formula.

codeCopy

```
Sales % Group =
IF(
ISINSCOPE(Region[Region])
|| ISINSCOPE(Region[Country]),
DIVIDE(
SUM(Sales[Sales]),
CALCULATE(
SUM(Sales[Sales]),
REMOVEFILTERS(
Region[Region],
Region[Country]
)
)
)
)
```

17. Notice that the **Sales % Group** measure now only returns a value when a region or country is in scope.

18. In Model view, place the three new measures into a display folder named **Ratios**.



19. Save the Power BI Desktop file.

*The measures added to the **Sales** table have modified filter context to achieve hierarchical navigation. Notice that the pattern to achieve the calculation of a subtotal requires removing some columns from the filter context, and to arrive at a grand total, all columns must be removed.*

Create a YTD measure

In this task, you'll create a sales year-to-date (YTD) measure by using time intelligence functions.

1. In Report view, on **Page 2**, notice the matrix visual that displays various measures with years and months grouped on the rows.
2. Add a measure to the **Sales** table, based on the following expression, and formatted to zero decimal places:

codeCopy

Sales YTD = TOTALYTD(SUM(Sales[Sales]), 'Date'[Date], "6-30")

*The TOTALYTD() function evaluates an expression—in this case the sum of the **Sales** column—over a given date column. The date column must belong to a date table marked as a date table.*

The function can also take a third optional argument representing the last date of a year. The absence of this date means that December 31 is the last date of the year. For Adventure Works, June is the last month of their year, and so "6-30" is used.

3. Add the **Sales** field and the **Sales YTD** measure to the matrix visual.
4. Notice the accumulation of sales values within the year.

Sales	Sales YTD
\$16,429,043	\$16,429,043
\$489,328	\$489,328
\$1,540,072	\$2,029,400
\$1,166,332	\$3,195,733
\$844,833	\$4,040,566
\$2,325,755	\$6,366,320
\$1,703,435	\$8,069,756
\$713,230	\$8,782,985
\$1,900,794	\$10,683,780
\$1,455,280	\$12,139,060
\$883,011	\$13,022,071
\$2,269,720	\$15,291,791
\$1,137,252	\$16,429,043
\$27,979,780	\$27,979,780
\$2,411,559	\$2,411,559
\$3,615,914	\$6,027,473

The **TOTALYTD()** function performs filter manipulation, specifically time filter manipulation. For example, to compute YTD sales for September 2017 (the third month of the fiscal year), all filters on the **Date** table are removed and replaced with a new filter of dates commencing at the beginning of the year (July 1, 2017) and extending through to the last date of the in-context date period (September 30, 2017).

Many Time Intelligence functions are available in DAX to support common time filter manipulations.

Create a YoY growth measure

In this task, you'll create a sales YoY growth measure by using a variable.

Variables help you simplify the formula and are more efficient if using the logic multiple times within a formula. Variables are declared with a unique name, and the measure expression must then be output after the **RETURN** keyword. Unlike some other coding language variables, DAX variables can only be used within the single formula.

1. Add another measure to the **Sales** table, based on the following expression:

```
SalesPriorYear =
CALCULATE(
SUM(Sales[Sales]),
PARALLELPERIOD(
'Date'[Date],
-12,
MONTH
)
)
```

The **SalesPriorYear** variable is assigned an expression that calculates the sum of the **Sales** column in a modified context that uses the **PARALLELPERIOD()** function to shift 12 months back from each date in filter context.

2. Add the **Sales YoY Growth** measure to the matrix visual.
3. Notice that the new measure returns BLANK for the first 12 months (because there were no sales recorded before fiscal year 2017).
4. Notice that the **Sales YoY Growth** measure value for **2018 Jul** is the **Sales** value for **2017 Jul**.

Sales	Sales YTD	Sales YoY Growth
\$16,429,043	\$16,429,043	
\$489,328	\$489,328	
\$1,540,072	\$2,029,400	
\$1,166,333	\$3,195,733	
\$844,833	\$4,040,566	
\$2,325,755	\$6,366,320	
\$1,703,435	\$8,069,756	
\$713,230	\$8,782,985	
\$1,900,794	\$10,683,780	
\$1,455,280	\$12,139,060	
\$883,011	\$13,022,071	
\$2,269,720	\$15,291,791	
\$1,137,252	\$16,429,043	
\$27,979,780	\$27,979,780	\$16,429,042.6
\$2,411,559	\$2,411,559	\$489,328.4
\$3,615,914	\$6,027,473	\$1,540,072.02

Now that the “difficult part” of the formula has been tested, you can overwrite the measure with the final formula that computes the growth result.

5. To complete the measure, overwrite the **Sales YoY Growth** measure with this formula, formatting it as a percentage with two decimal places:

```

Sales YoY Growth% =
VAR SalesPriorYear =
CALCULATE(
SUM(Sales[Sales]),
PARALLELPERIOD(
'Date'[Date],
-12,
MONTH
)
)
RETURN
DIVIDE(
(SUM(Sales[Sales]) - SalesPriorYear),
SalesPriorYear
)

```

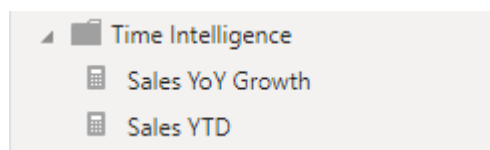
6. In the formula, in the **RETURN** clause, notice that the variable is referenced twice.

7. Verify that the YoY growth for **2018 Jul** is **392.83%**.

Year	Sales YTD2	SalesPriorYear	Sales YoY Growth2%
FY2018 Q1	3.195.732,87 kr.		
2017 Jul	489.328,4 kr.		
2017 Aug	2.029.400,42 kr.		
2017 Sep	3.195.732,87 kr.		
FY2018 Q2	8.069.755,58 kr.		
2017 Oct	4.040.565,58 kr.		
2017 Nov	6.366.320,49 kr.		
2017 Dec	8.069.755,58 kr.		
FY2018 Q3	12.139.059,85 kr.		
FY2018 Q4	16.429.042,6 kr.		
FY2019	27.979.779,53 kr.	16.429.042,6 kr.	70,31 %
FY2019 Q1	8.922.119,92 kr.	3.195.732,87 kr.	179,19 %
2018 Jul	2.411.558,85 kr.	489.328,4 kr.	392,83 %
Total		77.548.570,2 kr.	0,00 %

The YoY growth measure identifies almost 400% (or 4x) increase of sales during the same period of the previous year.

8. In Model view, place the two new measures into a display folder named **Time Intelligence**.



Lab complete

Videns-check

1.

Hvilken sætning beskriver bedst time intelligence i forbindelse med semantiske modelberegninger?

Rapportering af snapshotsaldi

Ændringer af filterkontekst, der inkluderer en datotabel

Time intelligence-beregninger ændrer datofilterkontekster.

Komplekse beregninger, der omfatter tid

Time intelligence-beregninger er ikke nødvendigvis komplekse.

Beregninger, der omfatter timer, minutter eller sekunder

2.

Du er ved at udvikle en semantisk model i Power BI Desktop. Du har lige tilføjet en datotabel ved hjælp af funktionen CALENDARAUTO. Du har udvidet den med beregnede kolonner, og du har relateret den til andre modeltabeller. Hvad skal du ellers gøre for at sikre, at DAX-time intelligence-beregninger fungerer korrekt?

Føj time intelligence-målinger til datotabellen.

Markér som en datotabel.

Du skal markere datotabellen, så Power BI kan filtrere de datoer, den indeholder, korrekt.

Tilføj et regnskabshierarki.

Tilføj en datokolonne.

3.

Du har en tabel, hvor der lagres snapshots af kontosalde for hver enkelt datoer, bortset fra weekender. Du skal sikre dig, at formlen for målingen kun filtrerer efter en enkelt dato. Hvis der ikke findes en post på den sidste dato i en tidsperiode, skal den bruge den seneste kontosaldo. Hvilken DAX-time intelligence-funktion skal du bruge?

FIRST

FIRSTNONBLANK

LAST

LASTNONBLANK

Funktionen LASTNONBLANK returnerer den sidste dato i den filterkontekst, hvor der findes en snapshotpost. Denne indstilling hjælper dig med at nå dit mål.

Resumé

Fuldført100 XP

- 5 minutter

I dette modul lærte du, at time intelligence-beregninger bruges til ændring af filterkonteksten for datofiltre. Du blev introduceret for mange DAX time intelligence-funktioner, der understøtter oprettelse af beregninger, f. eks. år-til-dato (YTD) eller år-for-år (YoY). Du har også lært, at life-to-date-beregninger (LTD) kan hjælpe dig med at tælle nye hændelser i faktadataene, og at snapshotdata kan filtreres på en måde, så de hjælper med at garantere, at der kun returneres en enkelt snapshotværdi.

Fortsæt det fremragende arbejde



Brug DAX-time intelligence-funktioner i Power BI Desktop-modeller

Du har opnået et resultat!

Tillykke, men hvad skal du gøre nu?

Lad os først dele dit resultat

Du afsætter tid til at lære noget nyt, så lad dit netværk tage del i din sejr!



<https://learn.microsoft.com/api/achievements/share/da-dk/bjarnehansen-7412/J9ZEGRUT?sharingId=39714CB1FE13C67B>

**** Enforce security in Power BI models

- 1 t. 21 min.
- Module
- 7 Units

Feedback

Intermediate

Data Analyst

Power BI

Enforce model security in Power BI using row-level security and object-level security.

Learning objectives

By the end of this module, you'll be able to:

- Restrict access to Power BI model data with RLS.
- Restrict access to Power BI model objects with OLS.
- Apply good development practices to enforce Power BI model security.

[Start](#)Tilføj

-
-
-

Prerequisites

- Experience developing Power BI data models by using Power BI Desktop.

Dette modul er en del af disse læringsforløb

- [Design and build tabular models](#)
- [Modeldata med Power BI](#)
- [Work with semantic models in Microsoft Fabric](#)

Modulvurdering

Besvar {percent}% af spørgsmålene korrekt for at kunne bestå.

[Tag modulvurderingen](#)

- [Introduction](#)3 min.
- [Restrict access to Power BI model data](#)14 min.
- [Restrict access to Power BI model objects](#)8 min.
- [Apply good modeling practices](#)3 min.
- [Exercise: Enforce model security](#)45 min.
- [Knowledge check](#)5 min.
- [Summary](#)3 min.

Introduction

Completed100 XP

- 3 minutes

You can enforce model security by restricting access to subset of data, and by restricting access to specific model tables and columns. You might restrict access because some report consumers aren't permitted to view specific data, like sales results of other sales regions. Achieving this requirement commonly involves

setting up row-level security (RLS), which involves defining roles and rules in that filter data model. You can also set up object-level security (OLS), to restrict access to entire tables or columns.

Learning objectives

By the end of this module, you'll be able to:

- Restrict access to Power BI model data with RLS.
- Restrict access to Power BI model objects with OLS.
- Apply good development practices to enforce Power BI model security.

Næste undermodul:

Restrict access to Power BI model data

[Næste](#)

Restrict access to Power BI model data

Completed100 XP

- 14 minutes

As a data modeler, you set up RLS by creating one or more roles. **A role has a unique name in the model, and it usually includes one or more rules. Rules enforce filters on model tables by using Data Analysis Expressions (DAX) filter expressions.**

RLS=Row-Level Security: controlling access to data by row

or RLS, refers to the practice of controlling access to data in a database by row, so that users are only able to access the data they are authorized for. This contrasts with database-level or table-level security which controls access to entire databases or tables, respectively

Bemærk

By default, a data model has no roles. A data model without roles means that users (who have permission to query the data model) have access to all model data.

Tip

It's possible to define a role that includes no rules. In this case, the role provides access to all rows of all model tables. This role set up would be suitable for an admin user who is allowed to view all data.

You can create, validate, and manage roles in Power BI Desktop. For Azure Analysis Services or SQL Server Analysis Services models, you can create, validate, and manage roles by using SQL Server Data Tools (SSDT).

You can also create and manage roles by using SQL Server Management Studio (SSMS), or by using a third-party tool, like [Tabular Editor](#).

To gain a better understanding of how RLS restricts access to data, watch the following animated image.

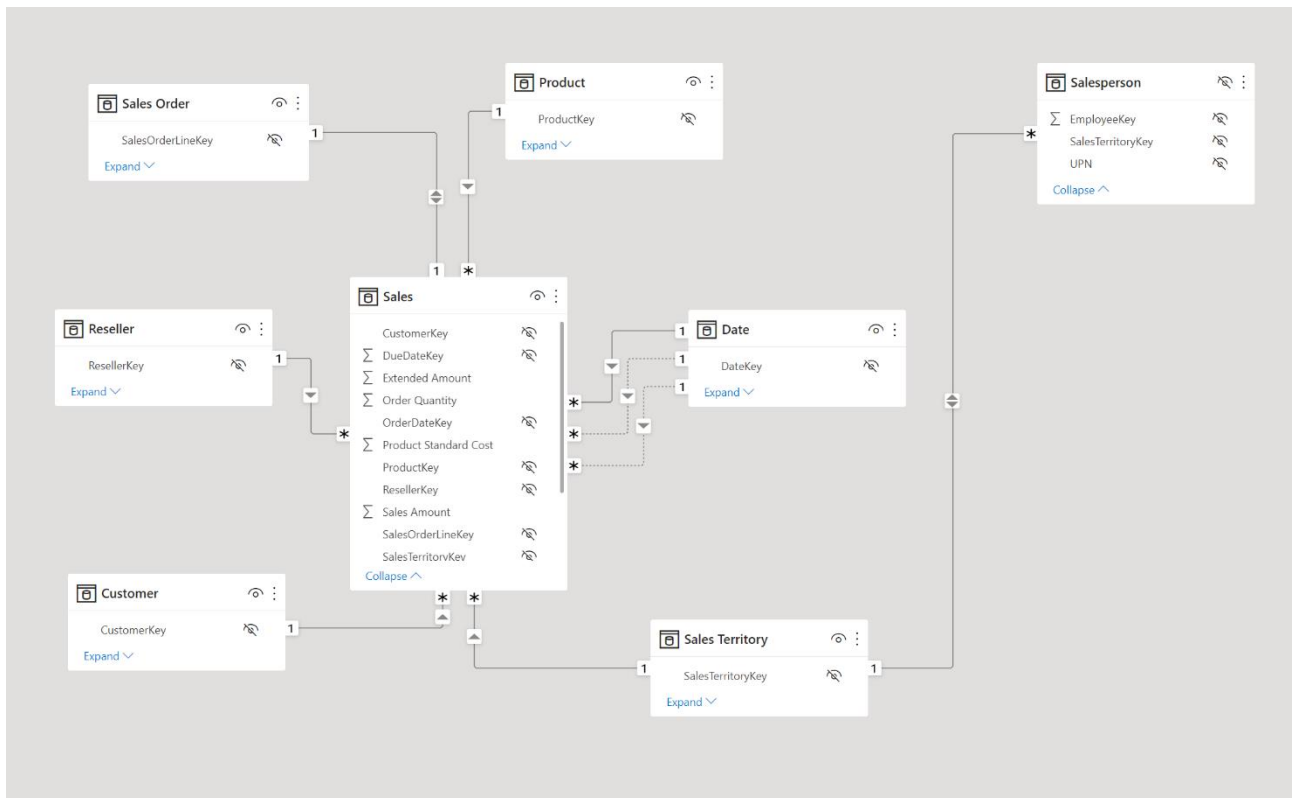


User kan kun se udvalgte rækker

Apply star schema design principles

We recommend you apply [star schema](#) design principles to produce a model comprising dimension and fact tables. **It's common to set up Power BI to enforce rules that filter dimension tables, allowing [model relationships](#) to efficiently propagate those filters to fact tables.**

The following image is the model diagram of the Adventure Works sales analysis data model. It shows a star schema design comprising a single fact table named **Sales**. The other tables are dimension tables that support the analysis of sales measures by date, sales territory, customer, reseller, order, product, and salesperson. Notice the model relationships connecting all tables. These relationships propagate filters (directly or indirectly) to the **Sales** table.



This model design supports examples presented in this unit.

Define rules

Rule expressions are evaluated within row context. **Row context means the expression is evaluated for each row using the column values of that row. When the expression returns TRUE, the user can “see” the row.**

Tip

To learn more about row context, work through the [Add calculated tables and columns to Power BI Desktop models](#) module. While this module describes adding model calculations, it includes a unit that introduces and describes row context.

You can define rules that are either static or dynamic.

Static rules

Static rules use DAX expressions that refer to constants.

Consider the following rule applied to the **Region** table that restricts data access to Midwest sales.

DAXKopier

```
'Region'[Region] = "Midwest"
```

The following steps explain how Power BI enforces the rule. It:

1. Filters the **Region** table, resulting in one visible row (for Midwest).
2. Uses the model relationship to propagate the **Region** table filter to the **State** table, resulting in 14 visible rows (the Midwest region comprise 14 states).
3. Uses the model relationship to propagate the **State** table filter to the **Sales** table, resulting in thousands of visible rows (the sales facts for the states that belong to the Midwest region).

The simplest static rule that you can create restricts access to all table rows. Consider the following rule applied to the **Sales Details** table (not depicted in the model diagram).

DAXKopier

FALSE()

This rule ensures users can't access any table data. **It could be useful when salespeople are allowed to see aggregated sales results (from the Sales table) but not order-level details.**

Defining static rules is simple and effective. Consider using them when you need to create only a few of them, as might be the case at Adventure Works where there are only six US regions. However, be aware of disadvantages: setting up static rules can involve significant effort to create and set up. It would also require you to update and republish the dataset when new regions are onboarded.

If there are many rules to set up and you anticipate adding new rules in the future, consider creating dynamic rules instead.

Dynamic rules

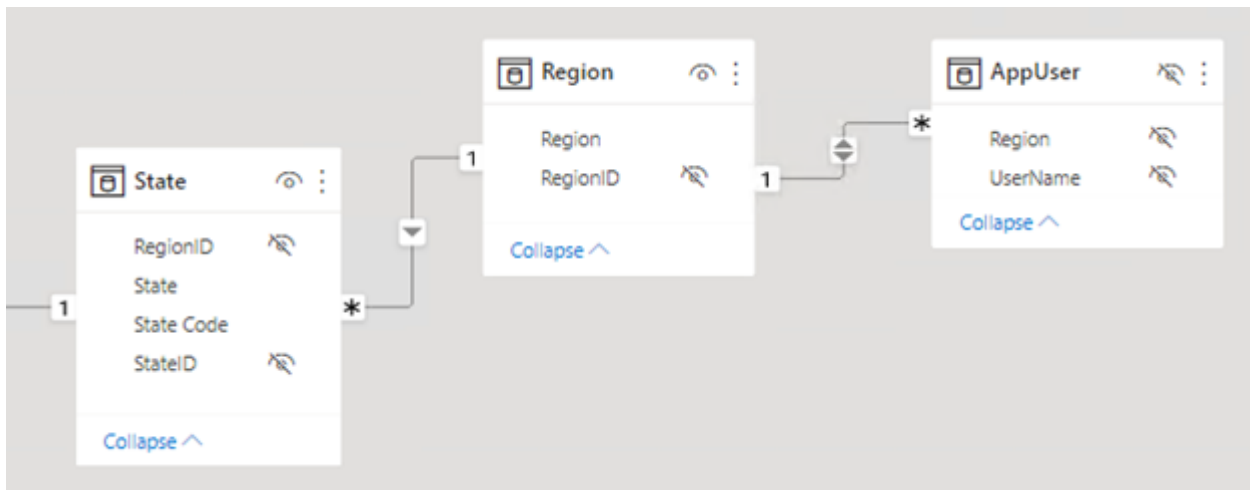
Dynamic rules use specific DAX functions that return environmental values (as opposed to constants). Environmental values are returned from three specific DAX functions:

- [USERNAME](#) or [USERPRINCIPALNAME](#) – Returns the Power BI authenticated user as a text value.
- [CUSTOMDATA](#) - Returns the **CustomData** property passed in the connection string. Non-Power BI reporting tools that connect to the dataset by using a connection string can set this property, like Microsoft Excel.

Bemærk

Be aware that the USERNAME function returns the user in the format of DOMAIN\username when used in Power BI Desktop. However, when used in the Power BI service, it returns the format of the user's User Principal Name (UPN), like username@adventureworks.com. Alternatively, you can use the USERPRINCIPALNAME function, which always returns the user in the user principal name format.

Consider a revised model design that now includes the (hidden) **AppUser** table. Each row of the **AppUser** table describes a username and region. A model relationship to the **Region** table propagates filters from the **AppUser** table.



The following rule applied to the **AppUser** table restricts data access to the region(s) of the authenticated user.

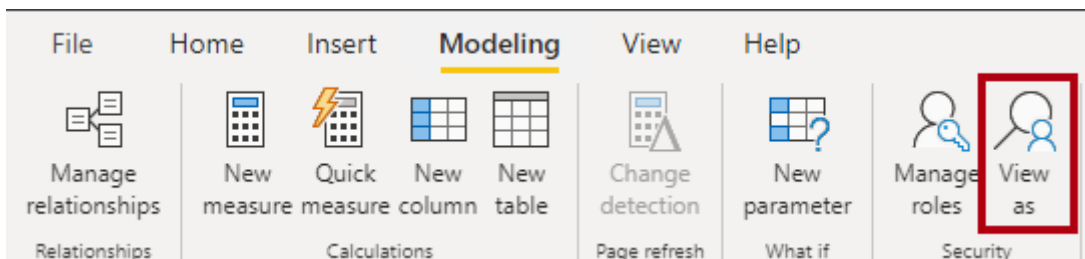
DAXKopier

```
'AppUser'[UserName] = USERPRINCIPALNAME()
```

Defining dynamic rules is simple and effective when a model table stores username values. They allow you to enforce a data-driven RLS design. For example, when salespeople are added to, or removed from, the **AppUser** table (or are assigned to different regions), this design approach just works.

Validate roles

When you create roles, it's important to test them to ensure they apply the correct filters. For data models created in Power BI Desktop, there's the **View as** function that allows you to see the report when different roles are enforced, and different username values are passed.



Set up role mappings

Role mappings must be set up in advance of users accessing Power BI content. Role mapping involves assigning Microsoft Entra security objects to roles. Security objects can be user accounts or security groups.

Tip

When possible, it's a **good practice to map roles to security groups**. That way, there will be fewer mappings, and you can delegate the group membership management to the network administrators.

For Power BI Desktop developed models, role mapping is typically done in the Power BI service. For Azure Analysis Services or SQL Server Analysis Services models, role mapping is typically done in SSMS.

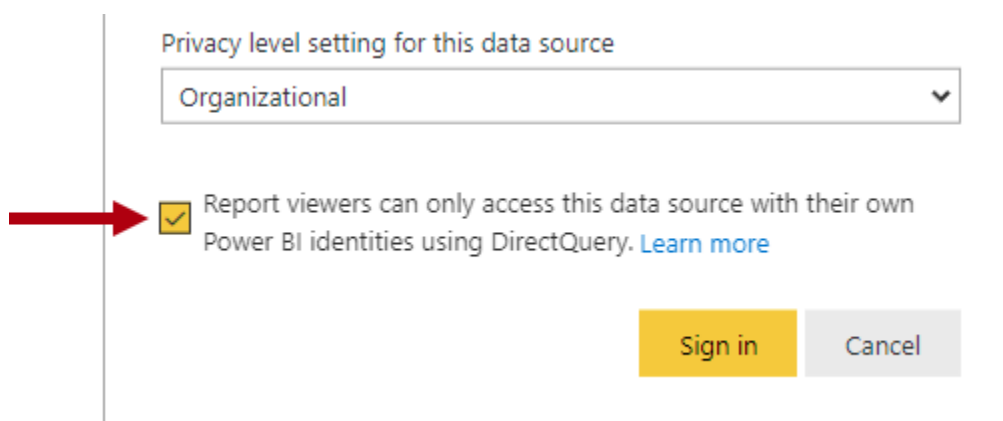
For more information about setting up RLS, see:

- [Row-level security \(RLS\) with Power BI](#)
- [Row-level security \(RLS\) guidance in Power BI Desktop](#)

Use single sign-on (SSO) for DirectQuery sources

When your data model has DirectQuery tables and their data source supports SSO, the data source can enforce data permissions. This way, the database enforces RLS, and Power BI datasets and reports honor the data source security.

Consider that Adventure Works has an Azure SQL Database for their sales operations that resides in the same tenant as Power BI. The database enforces RLS to control access to rows in various database tables. You can create a DirectQuery model that connects to this database without roles and publish it to the Power BI service. When you set the data source credentials in the Power BI service, you [enable SSO](#). When report consumers open Power BI reports, Power BI passes their identity to the data source. The data source then enforces RLS based on the identity of the report consumer.



For information about Azure SQL Database RLS, see [Row-level security](#).

Bemærk

Calculated tables and calculated columns that reference a DirectQuery table from a data source with SSO authentication aren't supported in the Power BI service.

For more information about data sources that support SSO, see [Single sign-on \(SSO\) for DirectQuery sources](#).

Næste undermodul:

Restrict access to Power BI model objects

[Tidligere](#)

Completed100 XP

- 8 minutes

As a data modeler, you can consider restricting user access to Power BI model objects.

Object-level security (OLS) can restrict access to specific tables and columns, and metadata.

Typically, you apply OLS to secure objects that store **sensitive data, like employee personal data**.

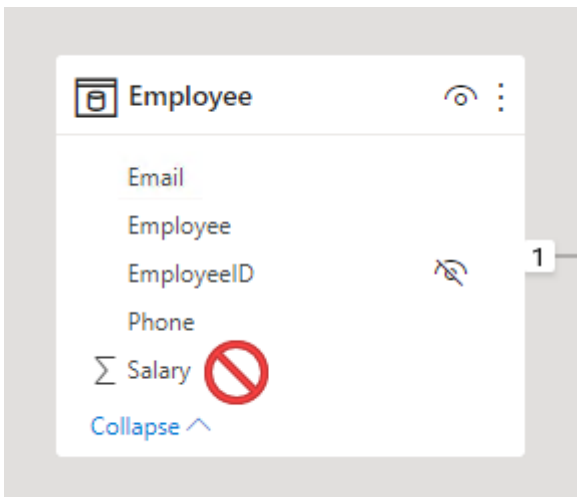
When Power BI enforces OLS, not only does it restrict access to tables and columns, but it can also secure metadata. When you secure metadata, it's not possible to retrieve information about secured tables and columns by using [Dynamic Management Views \(DMVs\)](#).

Vigtigt

Tabular models can hide tables and columns (and other objects) by using a [perspective](#).

A perspective defines viewable subsets of model objects som views in DB to help provide a specific focus for report authors. Perspectives are intended to reduce the complexity of a model, helping report authors find resources of interest. However, **perspectives aren't a security feature because they don't secure objects**. A user can still query a table or column even when it's not visible to them.

Consider an example at Adventure Works. This organization has a data warehouse dimension table named **DimEmployee**. The table includes columns that store employee name, phone, email address, and salary. While general report consumers can see employee name and contact details, they must not be able to see salary values. Only senior Human Resources staff are permitted to see salary values. So, the data modeler used OLS to grant access to the salary column only to specific Human Resources staff.



OLS is a feature inherited from Azure Analysis Services (AAS) and SQL Server Analysis Services (SSAS)

. The feature is available in Power BI Premium to provide backward compatibility for models migrated to Power BI. For this reason, it's not possible to completely set up OLS in Power BI Desktop.

Set up OLS

To set up OLS, you start by creating roles. You can create roles in Power BI Desktop in the same way you do when setting up RLS. Next, you need to add OLS rules to the roles. This capability isn't supported by Power BI Desktop, so you'll need to take a different approach.

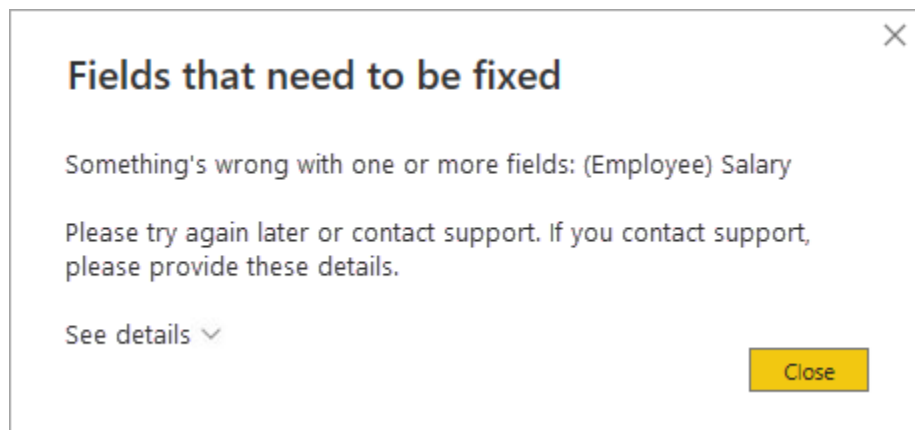
You add OLS rules to a Power BI Desktop model by using an [XML for Analysis \(XMLA\) endpoint](#). XMLA endpoints are available with Power BI Premium, and they provide access to the Analysis Services engine in the Power BI service. The read/write endpoint supports dataset management, application lifecycle management, advanced data modeling, and more. You can use XMLA endpoint-enabled APIs for scripting, such as [Tabular Model Scripting Language \(TMSL\)](#) or the [PowerShell SqlServer module](#). Or you can use a client tool, like SSMS. There are third-party tool options too, like [Tabular Editor](#), which is an open-source tool for creating, maintaining, and managing models.

By default, all model tables and columns aren't restricted. You can set them to **None** or **Read**. When set to **None**, users associated with the role can't access the object. When set to **Read**, users associated with the role can access the object. When you're restricting specific columns, ensure the table isn't set to **None**.

Once you've added the OLS rules, you can publish the model to the Power BI service. Use the same process for RLS to map accounts and security groups to the roles.

Considerations

In a Power BI report, when a user doesn't have permission to access a table or column, they'll receive an error message. The message will inform them that the object doesn't exist.



Consider carefully whether OLS is the right solution for your project. When a user opens a Power BI report that queries a restricted object (for them), the error message could be confusing and will result in a negative experience. To them, it looks like the report is broken. A better approach might be to create a separate set of models or reports for the different report consumer requirements.

Restrictions

There are restrictions to be aware of when implementing OLS.

You can't mix RLS and OLS in the same role. If you need to apply RLS and OLS in the same model, create separate roles dedicated to each type. Also, you can't set table-level security if it breaks a relationship chain. For example, if there are relationships between tables A and B, and B and C, you can't secure table B. If table B is secured, a query on table A can't transit the relationships between table A and B, and B and C. In this case, you could set up a separate relationship between tables A and C.



However, model relationships that reference a secured column will work, providing that the column's table isn't secured.

Lastly, while it isn't possible to secure measures, a measure that references secured objects is automatically restricted.

For more information, see [Object-level security](#).

Næste undermodul:

Apply good modeling practices

Tidligere

It's critical that your model enforces data permissions correctly and efficiently. The following list provides you with good development practices to apply.

- Strive to define fewer datasets (data models) with well-designed roles.
- Strive to create fewer roles by using dynamic rules. A data-driven solution is easier to maintain because you don't need to add new roles.
- When possible, create rules that filter dimension tables instead of fact tables. It will help to deliver faster query performance.
- Validate that the model design, including its relationships and relationship properties, are correctly set up.
- Use the USERPRINCIPALNAME function instead of USERNAME function. It provides consistency when validating the roles in Power BI Desktop and the Power BI service.
- Rigorously validate RLS and OLS by testing all roles.
- Ensure that the Power BI Desktop data source connection uses the same credentials that will be applied when set up in the Power BI service.

Exercise: Enforce model security

Completed 100 XP

- 45 minutes

Opretter delene i desktop version, kan ikke arbejde med Web-service delen

Denne enhed kræver en VM for at kunne fuldføres.

VM-tilstand giver dig et gratis webbaseret miljø via en virtuel maskine, så du kan fuldføre trinnene i dette undermodul.

Microsoft leverer denne laboratorieøvelse og relateret indhold til uddannelsesmæssige formål. Alle de viste oplysninger ejes af Microsoft og er udelukkende beregnet til at lære om de produkter og tjenester, der er omfattet af dette Microsoft Learn-modul.

Start VM-tilstand

Vigtigt

A virtual machine (VM) containing the client tools you need is provided, along with the exercise instructions. Use the button to open the VM. A limited number of concurrent sessions are available - if the hosted environment is unavailable, try again later.

In this exercise, you'll update a predeveloped data model to enforce security. Specifically, salespeople at the Adventure Works company should only be able to see sales data related to their assigned sales region. You learn how to:

- Create static roles
- Create dynamic roles
- Validate roles
- Map security principals to semantic model roles

When you finish the exercise, end the lab to close the VM. Don't forget to come back and complete the knowledge check to earn points for completing this module!

Bemærk

To complete this exercise on your own computer, you need a **Microsoft Fabric license**. See [Getting started with Fabric](#) for details of how to enable a free Fabric trial license.

You also need a Microsoft *school* or *work* account to do this. If you don't have one, you can [use the M365 Developer Program](#).

Launch the exercise and follow the instructions.

Sign in to the Power BI service

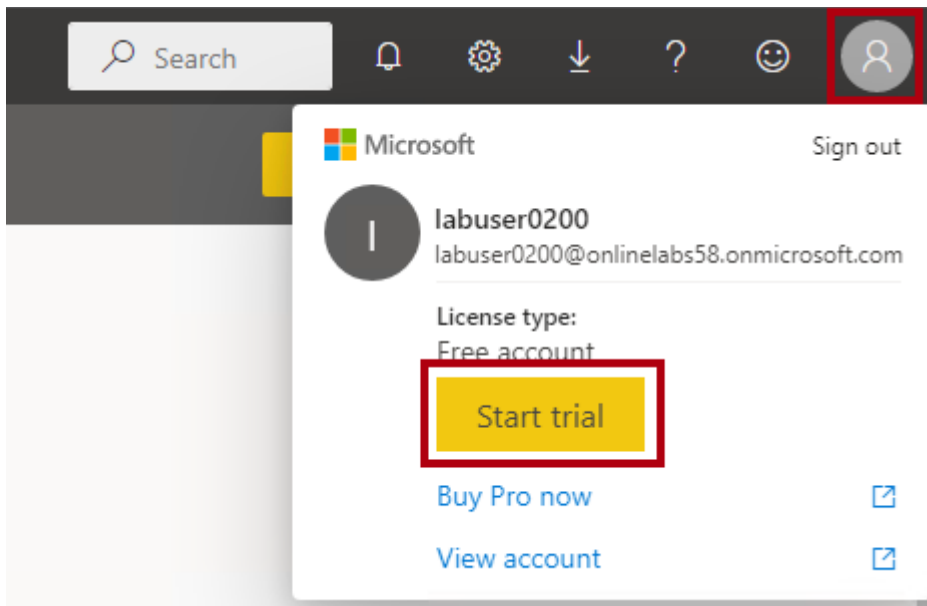
In this task, you will sign in to the Power BI service, start a trial license, and create a workspace.

Important: If you have already setup Power BI in your VM environment, continue to the next task.

1. In a web browser, go to <https://powerbi.com>.
2. Use the lab credentials to complete the sign in process.

Important: You must use the same credentials used to sign in from Power BI Desktop.

3. At the top-right, select the profile icon, and then select **Start trial**.



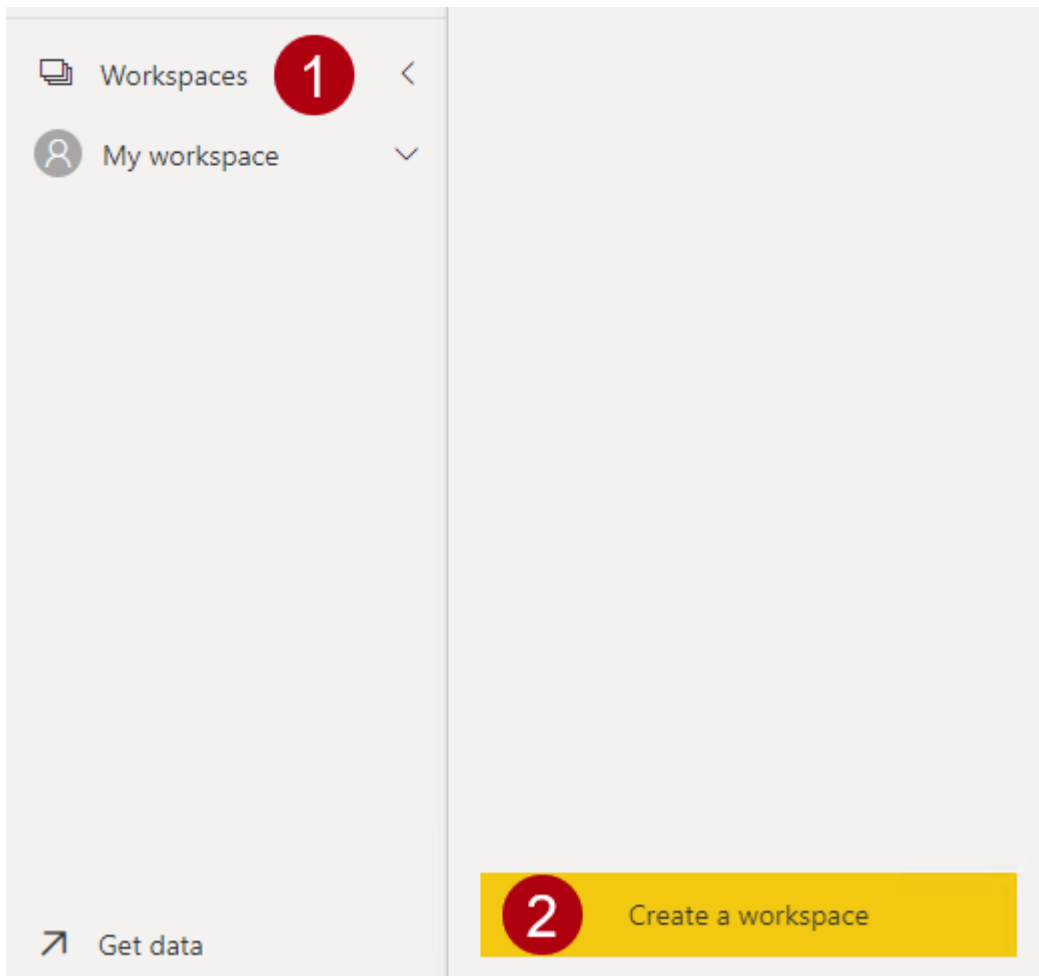
4. When prompted, select **Start trial**.
5. Do any remaining tasks to complete the trial setup.

*Tip: The Power BI web browser experience is known as the **Power BI service**.*

Create a workspace

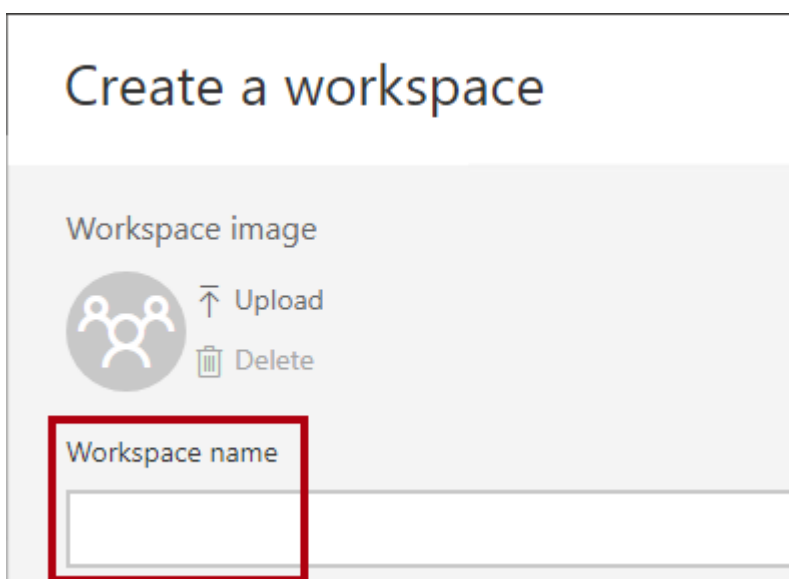
In this task, you will create a workspace.

1. In the Power BI service, to create a workspace, in the **Navigation** pane (located at the left), select **Workspaces**, and then select **Create workspace**.



2. In the **Create a workspace** pane (located at the right), in the **Workspace name** box, enter a name for the workspace.

The workspace name must be unique within the tenant.



3. Select **Save**.

Once created, the workspace is opened. In a later exercise, you will publish a dataset to this workspace.

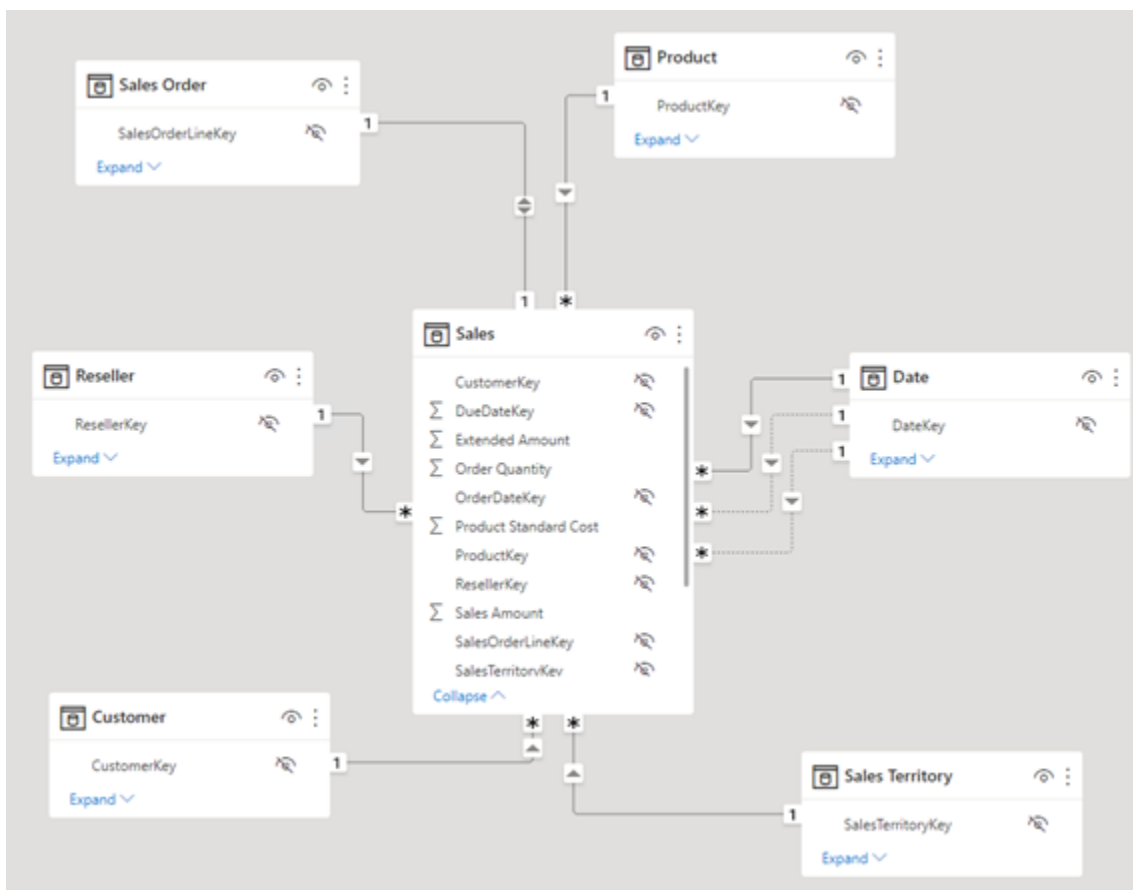
Review the data model

In this task, you will review the data model.

1. In Power BI Desktop, at the left, switch to **Model** view.

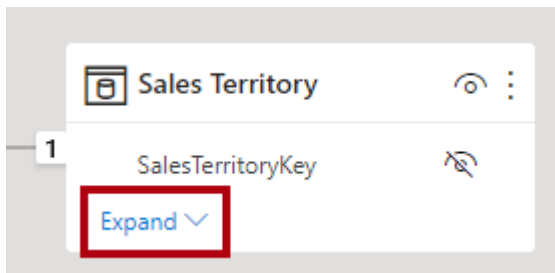


2. Use the model diagram to review the model design.



The model comprises six dimension tables and one fact table. The **Sales** fact table stores sales order details. It's a classic star schema design.

3. Expand open the **Sales Territory** table. **Region i 05-Solution-Sales-Analysis**



4. Notice that the table includes a **Region** column.

The **Region** column stores the Adventure Works sales regions. *At this organization, salespeople are only allowed to see data related to their assigned sales region. In this lab, you will implement two different row-level security techniques to enforce data permissions.*

-Create static roles

In this exercise, you will create and validate static roles, and then see how you would map security principals to the dataset roles.

Create static roles

In this task, you will create two static roles.

1. Switch to **Report** view.



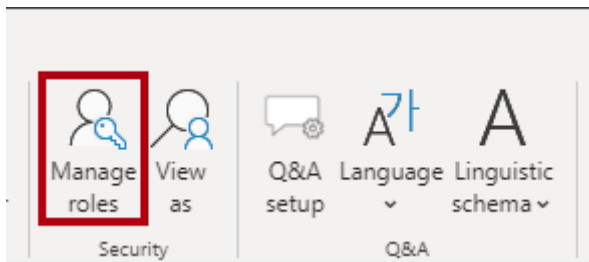
2. In the stacked column chart visual, in the legend, notice (for now) that it's possible to see many regions.

Sales Amount by Month and Region

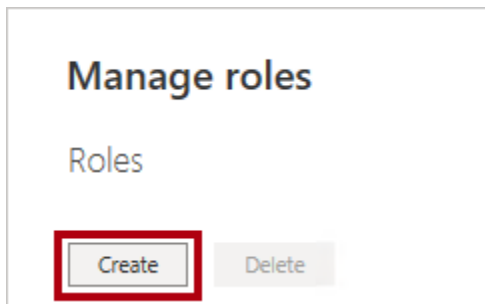
Region ● Australia ● Canada ● Central ● France ● Germany ● Northeast ● Northwest

For now, the chart looks overly busy. That's because all regions are visible. When the solution enforces row-level security, the report consumer will see only one region.

3. To add a security role, on the **Modeling** ribbon tab, from inside the **Security** group, select **Manage roles**.



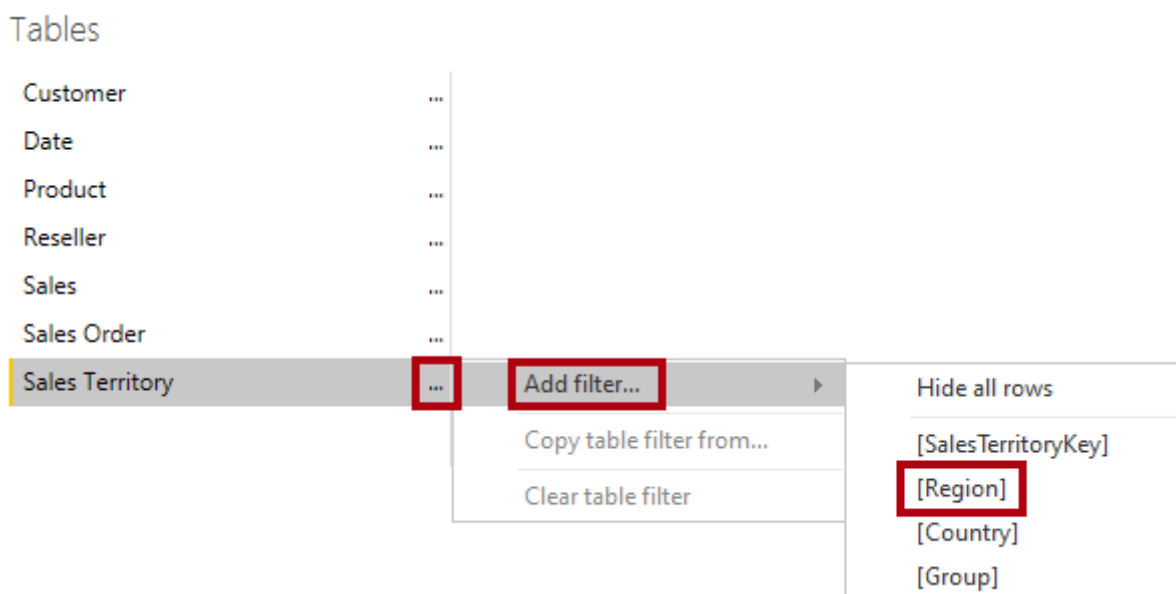
4. In the **Manage roles** window, select **Create**.



5. To name the role, replace the selected text with **Australia**, and then press **Enter**.



6. In the **Tables** list, for the **Sales Territory** table, select the ellipsis, and then select **Add filter > [Region]**.



7. In the **Table filter DAX expression** box, replace **Value** with **Australia**.

Table filter DAX expression

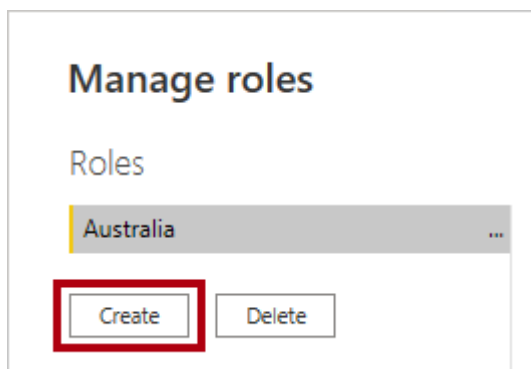
```
[Region] = "Australia"
```

User ser kun de aktivt tilvalgte filter-værdier

[Country]=United States : user ser kun poster opfyldende dette

*This expression filters the **Region** column by the value **Australia**.*

8. To create another role, press **Create**.



9. Repeat the steps in this task to create a role named **Canada** that filters the **Region** column by **Canada**.



In this lab, you'll create just the two roles. Consider, however, that in a real-world solution, a role must be created for each of the 11 Adventure Works regions.

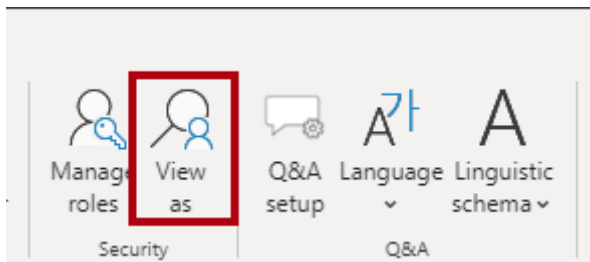
10. Select **Save**.



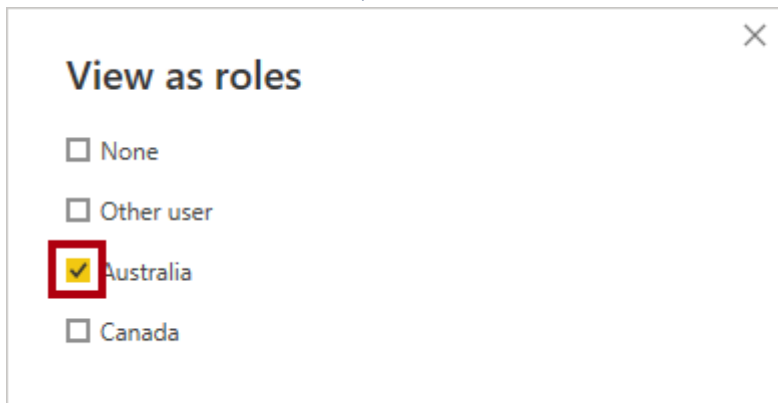
Validate the static roles

In this task, you will validate one of the static roles.

1. On the **Modeling** ribbon tab, from inside the **Security** group, select **View as**.



In the **View as roles** window, select the **Australia** role.



2. Select **OK**.

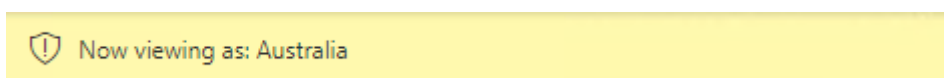


3. On the report page, notice that the stacked column chart visual shows only data for Australia.

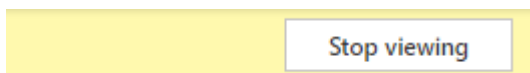
Sales Amount by Month and Region

Region ● Australia

4. Across the top of the report, notice the yellow banner that confirms the enforced role.



5. To stop viewing by using the role, at the right of the yellow banner, select **Stop viewing**.



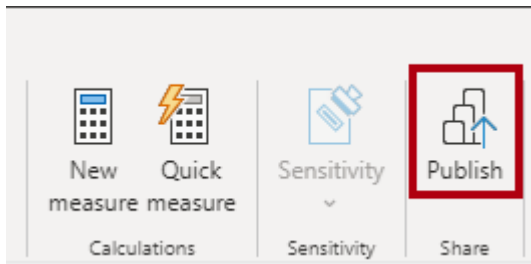
[Publish the reports til Webservice-delen](#)

In this task, you will publish the report.

1. Save the Power BI Desktop file.



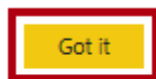
2. To publish the report, on the **Home** ribbon tab, select **Publish**.



3. In the **Publish to Power BI** window, select your workspace, and then select **Select**.



4. When the publishing succeeds, select **Got it**.



Configure row-level security (*optional*)-for assets in Web-service delen

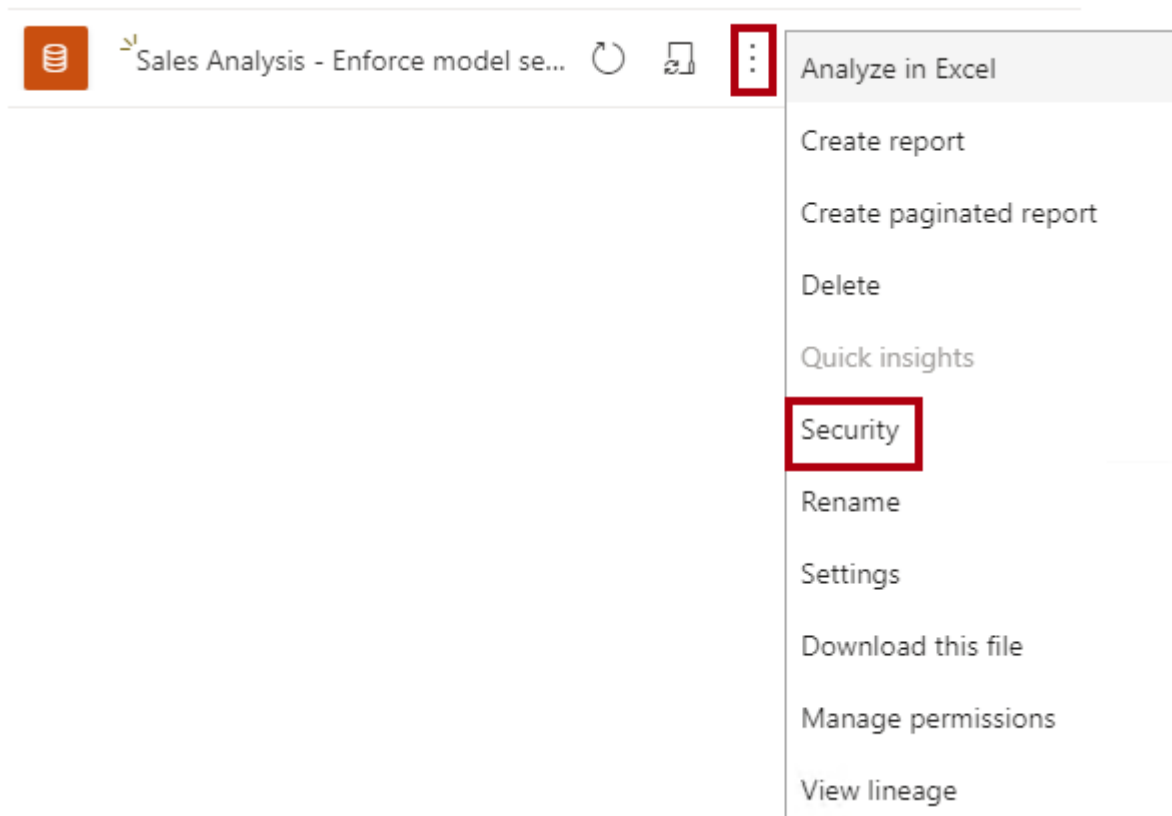
In this task, you will see how to configure row-level security in the Power BI service.

This task relies on the existence of a **Salespeople_Australia** security group in the tenant you are working in. This security group does NOT automatically exist in the tenant. If you have permissions on your tenant, you can follow the steps below. If you are using a tenant provided to you in training, you will not have the appropriate permissions to create security groups. Please read through the tasks, but note that you will not be able to complete them in the absence of the existence of the security group. **After reading through, proceed to the Clean Up task.**

1. Switch to the Power BI service (web browser).
2. In the workspace landing page, notice the **Sales Analysis - Enforce model security** dataset.

All Content Datasets + dataflows		
	Name	Type
	 Sales Analysis - Enforce model security	Report
	 Sales Analysis - Enforce model security	Dataset

3. Hover the cursor over the dataset, and when the ellipsis appears, select the ellipsis, and then select **Security**.



The **Security** option supports mapping Microsoft Azure Active Directory (Azure AD) security principals, which includes security groups and users.

4. At the left, notice the list of roles, and that **Australia** is selected.



5. In the **Members** box, commence entering **Salespeople_Australia**.

*Steps 5 through 8 are for demonstration purposes only, as they rely on the creation or existence of a **Salespeople_Australia** security group. If you have permissions and the knowledge to create security groups, please feel free to proceed. Otherwise, continue to the Clean Up task.*

Members (0)

People or groups who belong to this role



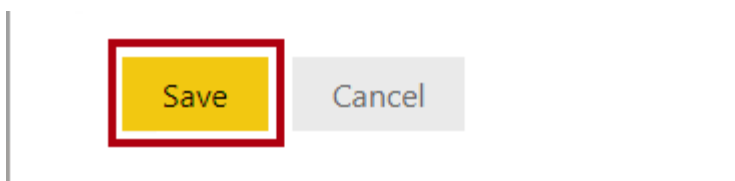
Salespeople_

- S Salespeople_Australia
- S Salespeople_Canada

6. Select **Add**.



7. To complete the role mapping, select **Save**.



*Now all members of the **Salespeople_Australia** security group are mapped to the **Australia** role, which restricts data access to view only Australian sales.*

In a real-world solution, each role should be mapped to a security group.

This design approach is simple and effective when security groups exist for each region. However, there are disadvantages: it requires more effort to create and set up. It also requires updating and republishing the dataset when new regions are onboarded.

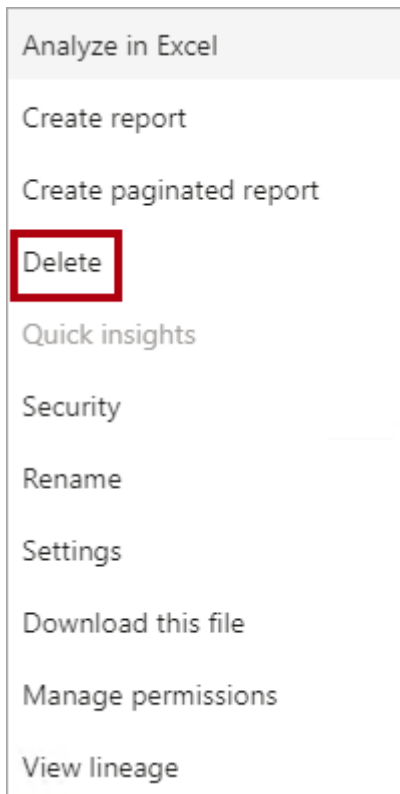
In the next exercise, you will create a dynamic role that is data-driven. This design approach can help address these disadvantages.

8. To return to the workspace landing page, in the **Navigation** pane, select the workspace.

Clean up the solution

In this task, you will clean up the solution by removing the dataset and the model roles.

1. To remove the dataset, hover the cursor over the dataset, and when the ellipsis appears, select the ellipsis, and then select **Delete**.

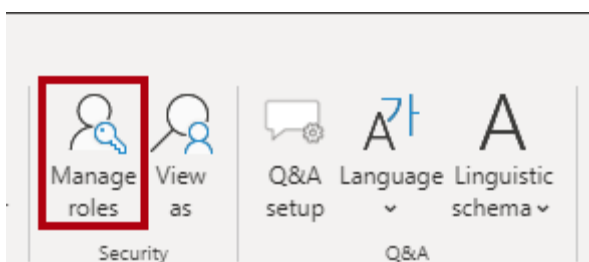


You will republish a revised dataset in the next exercise.

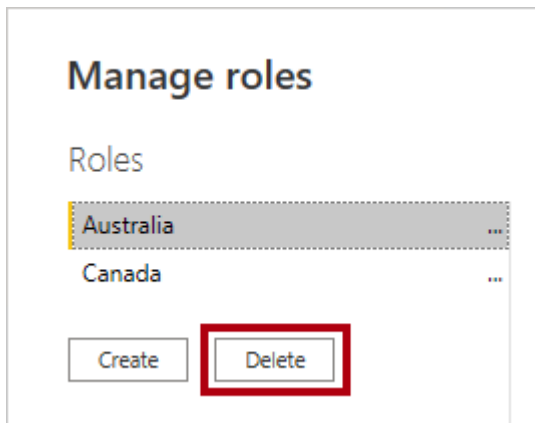
2. When prompted to confirm the deletion, select **Delete**.



3. Switch to Power BI Desktop.
4. To remove the security roles, on the **Modeling** ribbon tab, from inside the **Security** group, select **Manage roles**.



5. In the **Manage roles** window, to remove the first role, select **Delete**.



- When prompted to confirm the deletion, press **Yes, delete**.



- Also remove the second role.
- Select **Save**.



-Create a dynamic role

In this exercise, you will add a table to the model, create and validate a dynamic role, and then map a security principal to the dataset role.

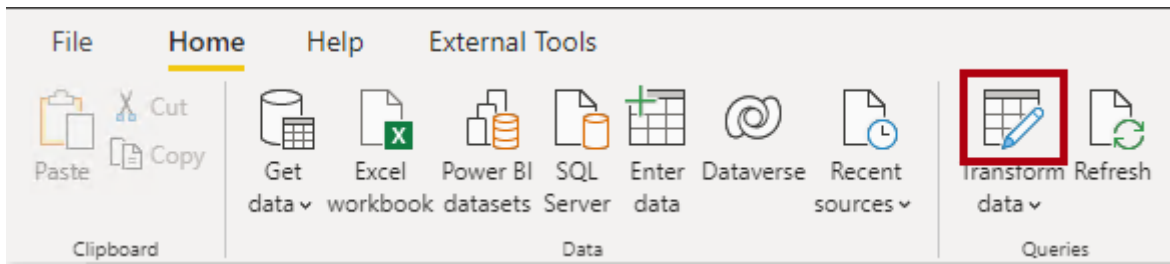
Add the Salesperson table

In this task, you will add the **Salesperson** table to the model.

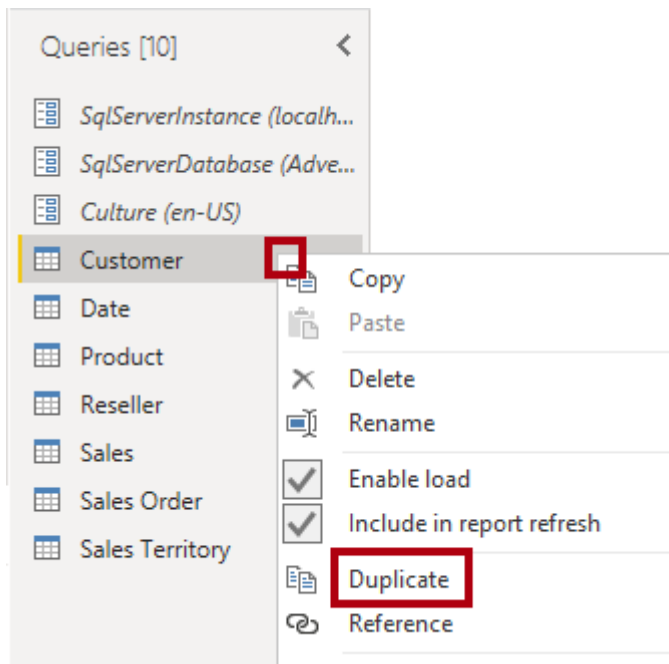
- Switch to **Model** view.



- On the **Home** ribbon tab, from inside the **Queries** group, select the **Transform data** icon.



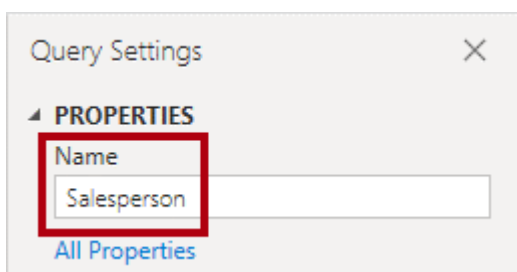
3. In the **Power Query Editor** window, in the **Queries** pane (located at the left), right-click the **Customer** query, and then select **Duplicate**.



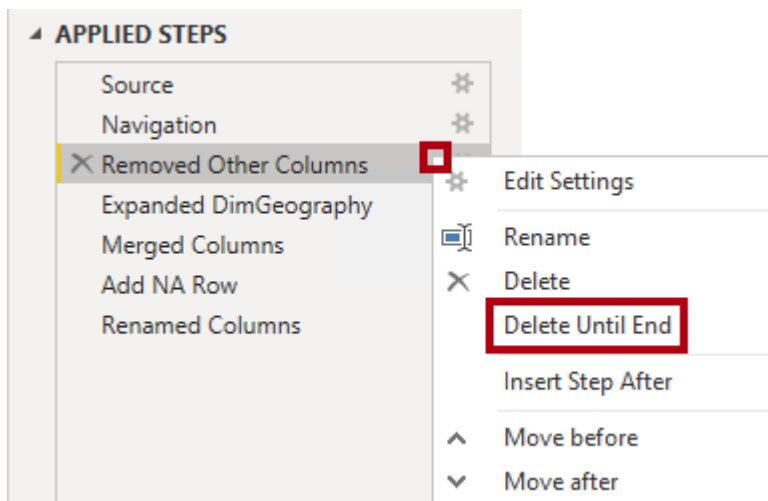
Laver det i **Product_sales.pbix** filen

Because the **Customer** query already includes steps to connect the data warehouse, duplicating it is an efficient way to commence the development of a new query.

4. In the **Query Settings** pane (located at the right), in the **Name** box, replace the text with **Salesperson**.



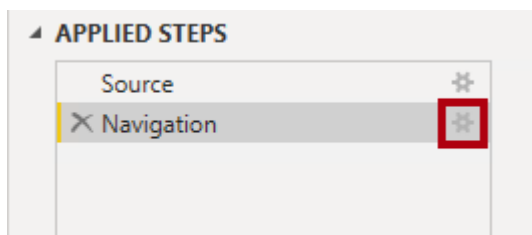
5. In the **Applied Steps** list, right-click the **Removed Other Columns** step (third step), and then select **Delete Until End**.



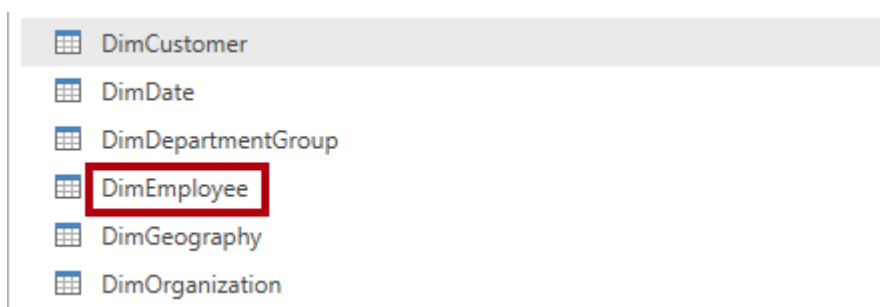
6. When prompted to confirm deletion of the step, select **Delete**.



7. To source data from a different data warehouse table, in the **Applied Steps** list, in the **Navigation** step (second step), select the gear icon (located at the right).



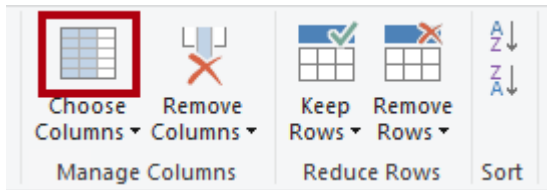
8. In the **Navigation** window, select the **DimEmployee** table.



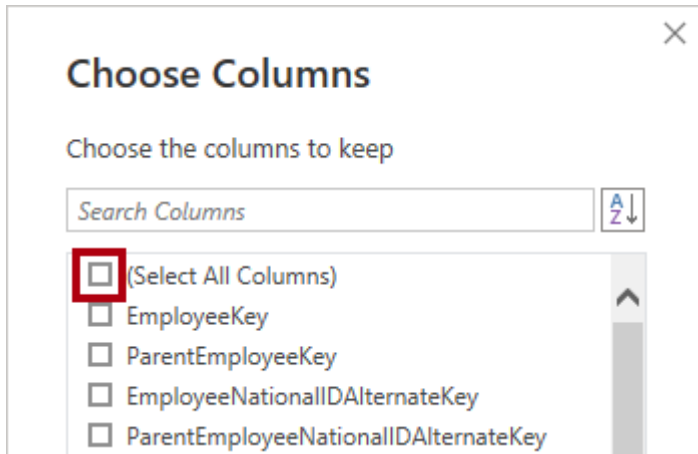
9. Select **OK**.



10. To remove unnecessary columns, on the **Home** ribbon tab, from inside the **Manage Columns** group, select the **Choose Columns** icon.



11. In the **Choose Columns** window, uncheck the **(Select All Columns)** item.



12. Check the following three columns:

- EmployeeKey
- SalesTerritoryKey
- EmailAddress

13. Select **OK**.



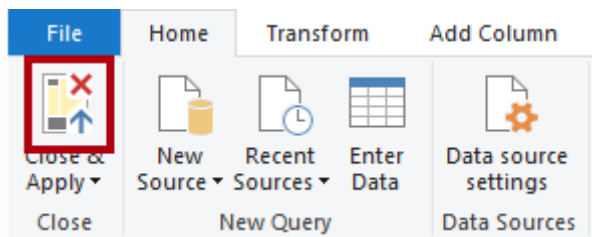
14. To rename the **EmailAddress** column, double-click the **EmailAddress** column header.

15. Replace the text with **UPN**, and then press **Enter**.

UPN is an acronym for User Principal Name. The values in this column match the Azure AD account names.

	123 EmployeeKey	123 SalesTerritoryKey	A ^B UPN
1	1	11	guy1@adventure-works.com
2	2	11	kevin0@adventure-works.com
3	3	11	roberto0@adventure-works.com

16. To load the table to the model, on the **Home** ribbon tab, select the **Close & Apply** icon.



17. When the table has added to the model, notice that a relationship to the **Sales Territory** table was automatically created.

Her er Salesperson table færdig, data er påfyldt, mangler kun relationship :

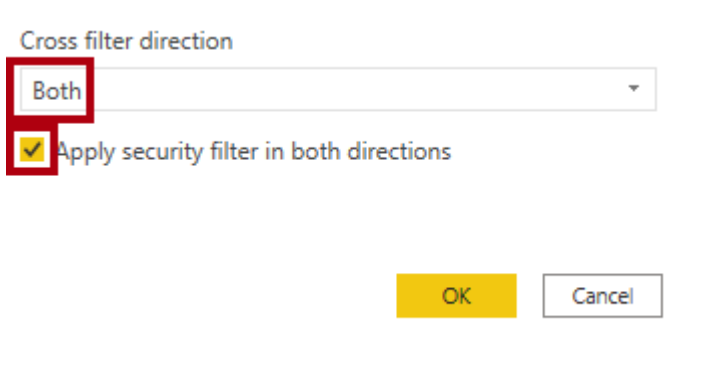
[Configure the relationship for nye SalesPerson table](#)

In this task, you will configure properties of the new relationship.

1. Right-click the relationship between the **Salesperson** and **Sales Territory** tables, and then select **Properties**.



2. In the **Edit relationship** window, in the **Cross filter direction** dropdown list, select **Both**.
3. Check the **Apply security filter in both directions** checkbox.

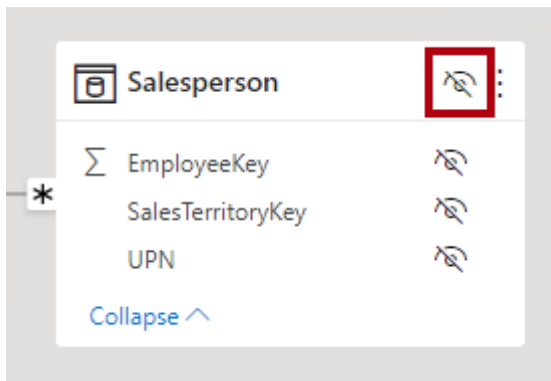


*Because there's a one-to-many relationship from the **Sales Territory** table to the **Salesperson** table, filters propagate only from the **Sales Territory** table to the **Salesperson** table. To force propagation in the other direction, the cross filter direction must be set to both.*

4. Select **OK**.



5. To hide the table, at the top-right of the **Salesperson** table, select the eye icon.

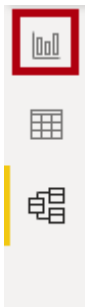


The purpose of the **Salesperson** table is to enforce data permissions. When hidden, report authors and the Q&A experience won't see the table or its fields.

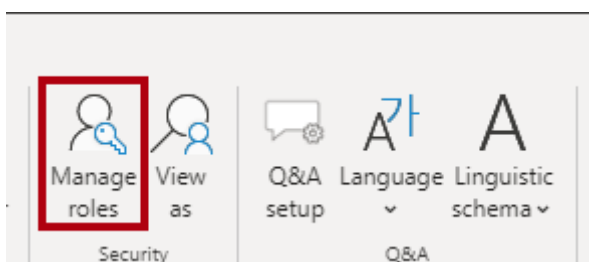
Create a dynamic role

In this task, you will create a dynamic role, which enforces permissions based on data in the model.

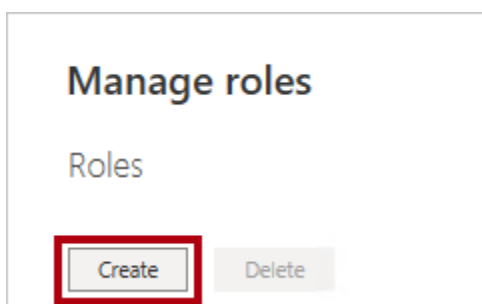
1. Switch to **Report** view.



2. To add a security role, on the **Modeling** ribbon tab, from inside the **Security** group, select **Manage roles**.



3. In the **Manage roles** window, select **Create**.

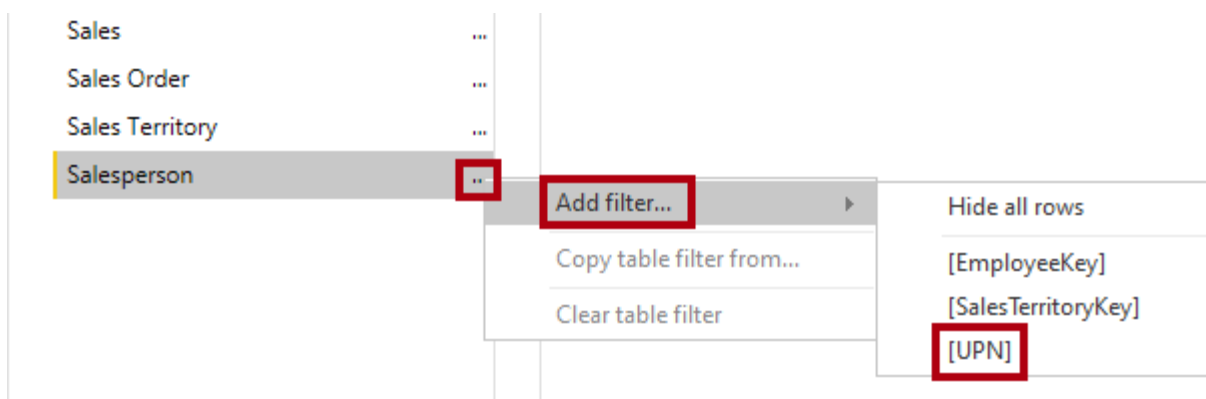


4. To name the role, replace the selected text with **Salespeople**.



This time, only one role needs to be created.

5. Add a filter to the **UPN** column of the **Salesperson** table.



6. In the **Table filter DAX expression** box, replace "Value" with **USERPRINCIPALNAME()**.

Table filter DAX expression



*This expression filters the **UPN** column by the **USERPRINCIPALNAME** function, which returns the user principal name (UPN) of the authenticated user.*

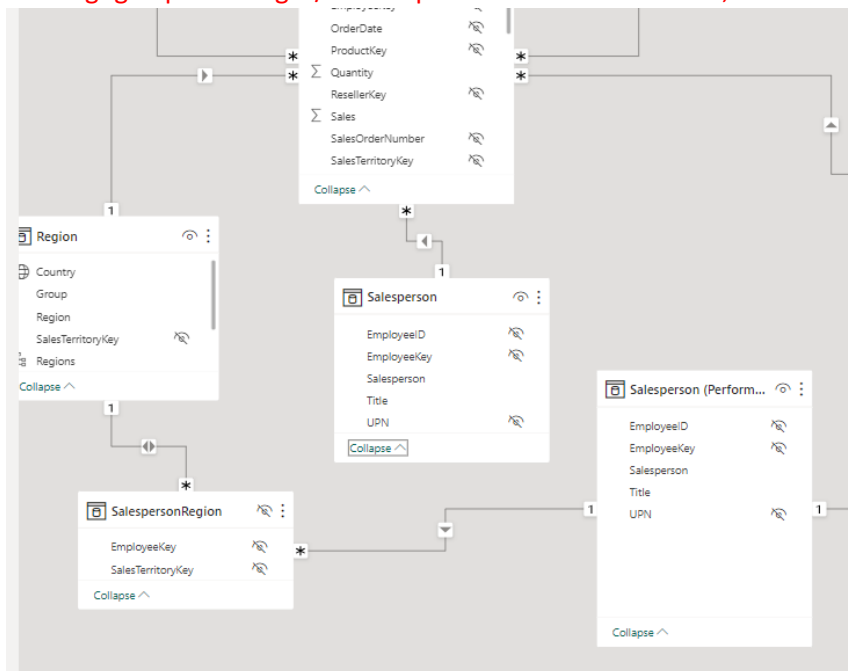
*When the **UPN** filters the **Salesperson** table, it filters the **Sales Territory** table, which in turn filters the **Sales** table. This way, the authenticated user will only see sales data for their assigned region.*

7. Select **Save**.



Hver Salesperson er tilknyttet Region via SalespersonRegion, som nu er dobbelt-rettet

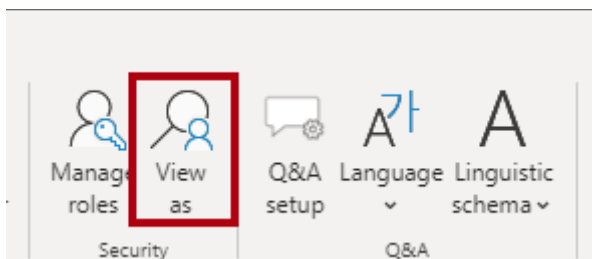
Med rigtige upn udvælges/filtres på én række i SalesPerson, som så udvælger bestemt Region



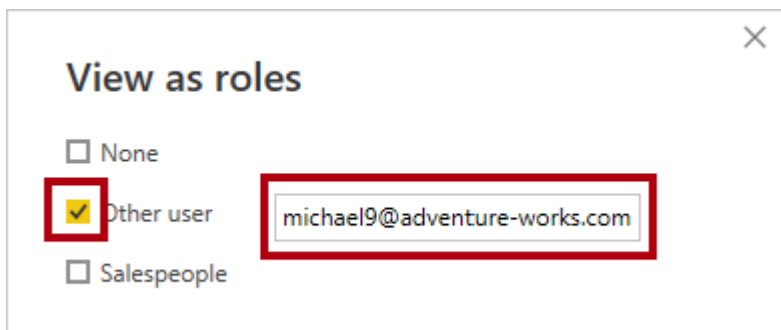
Validate the dynamic role

In this task, you will validate the dynamic role.

1. On the **Modeling** ribbon tab, from inside the **Security** group, select **View as**.

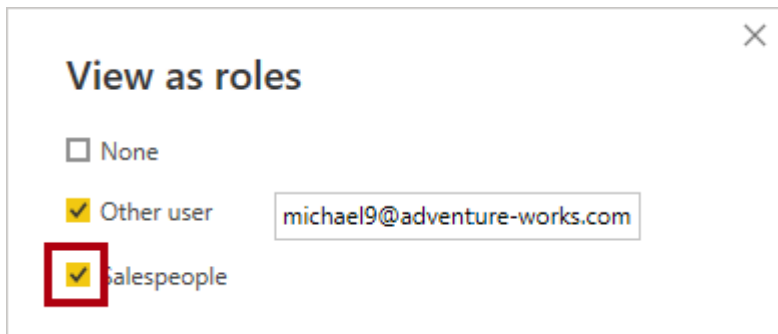


2. In the **View as roles** window, check **Other user**, and then in the corresponding box, enter: **michael9@adventure-works.com**



For testing purposes, **Other user** is the value that will be returned by the **USERPRINCIPALNAME** function. Note that this salesperson is assigned to the **Northeast** region.

3. Check the **Salespeople** role.



4. Select **OK**.

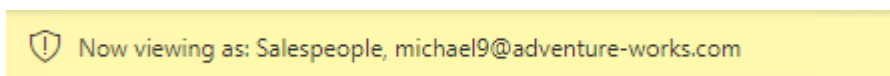


5. On the report page, notice that the stacked column chart visual shows only data for Northeast.

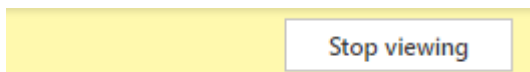
Sales Amount by Month and Region

Region ● Northeast

6. Across the top of the report, notice the yellow banner that confirms the enforced role.



7. To stop viewing by using the role, at the right of the yellow banner, select **Stop viewing**.



Finalize the design

In this task, you will finalize the design by publishing the report and mapping a security group to the role.

The steps in this task are deliberately brief. For full step details, refer to the task steps of the previous exercise.

1. Save the Power BI Desktop file.



2. Publish the report to the workspace you created at the beginning of the lab.
3. Close Power BI Desktop.
4. Switch to the Power BI service (web browser).
5. Go to the security settings for the **Sales Analysis - Enforce model security** dataset.
6. Map the **Salespeople** security group the **Salespeople** role.

Members (0)

People or groups who belong to this role

S

Salespeople

×

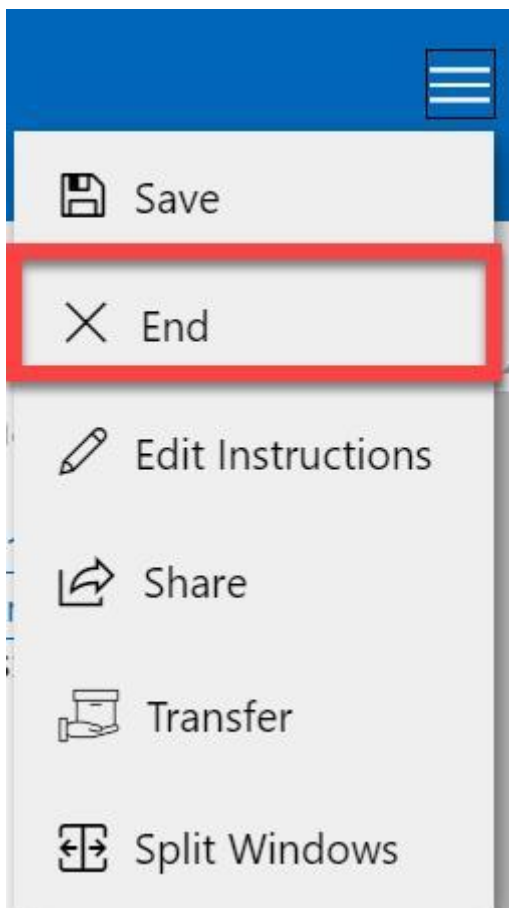
Enter email addresses

Now all members of the **Salespeople** security group are mapped to the **Salespeople** role. Providing the authenticated user is represented by a row in the **Salesperson** table, the assigned sales territory will be used to filter the sales table.

This design approach is simple and effective when the data model stores the user principal name values. When salespeople are added or removed, or are assigned to different sales territories, this design approach will simply work.

End the lab

Please be sure to end the lab(s) using the menu in the top right corner.



Check your knowledge

1.

Joshua is a data modeler at Adventure Works who is developing a model for a large data warehouse. The model must enforce RLS, and the Power BI reports that connect to the model should deliver the fastest possible performance. What should Joshua do?

Apply rules to dimension tables.

Correct. It's efficient to apply rules to dimension tables. Power BI filters the dimension tables (that usually have a small number of rows) and uses model relationships to propagate those filters to fact tables (that contain millions of rows).

Apply rules to hierarchies.

Apply rules to fact tables.

2.

Rupali is a data modeler at Adventure Works who is developing an import model to analyze employee timesheet data. The employee table stores the employee social security number (SSN) in a column. While the model will be available for all company managers, it will also be available to employees in the Payroll department. However, reports must only reveal employee SSNs to payroll employees. What feature should Rupali use to restrict access to the SSN column?

RLS.

SSO.

OLS.

Correct. You can use OLS (object-level security) to secure the SSN column in the employee table.

3.

Kasper is a data modeler at Adventure Works who is developing a model that must enforce RLS. It must restrict access to only the sales regions assigned to the report consumer. The source database includes a table that stores employee usernames and their assigned region(s). What should Kasper do?

Create an OLS role and use a dynamic rule.

Create an RLS role and use a dynamic rule.

Correct. An RLS role with a dynamic rule that uses the USERPRINCIPALNAME (or USERNAME) function can filter model data to the report consumer's assigned region(s).

Create an RLS role and use a static rule.

Summary

Completed 100 XP

- 3 minutes

When report consumers can see all model data there's no need to take special steps. However, when they should see subsets of model data or be restricted from certain tables or columns, it's critical that your model enforces appropriate security.

You typically restrict access to model data with RLS rules. However, when you develop a DirectQuery model for data sources that support SSO, Power BI can leverage the data source RLS. In this case, as a data modeler, you don't need to create model roles.

There are many good development practices you should apply to ensure data permissions are enforced accurately and efficiently.

Alle enheder er fuldført:

Resultater

Fortsæt det fremragende arbejde

Modeldata med Power BI

Enforce Power BI model security

Bestået modultilvurdering

Du har optjent 2-gennemførelser!

Tillykke, men hvad skal du gøre nu?

Lad os først dele dit resultat

Du afsætter tid til at lære noget nyt, så lad dit netværk tage del i din sejr!

in ✉ ✕ f ↗

<https://learn.microsoft.com/api/achievements/share/da-dk/bjarnehansen-7412/FVL3D9RX?sharingId=39714CB1FE13C67B>

<https://learn.microsoft.com/api/achievements/share/da-dk/bjarnehansen-7412/CF4SVJH9?sharingId=39714CB1FE13C67B>