LEARNIA
Partners in efficient learning

Advanced Data Analytics

Power BI

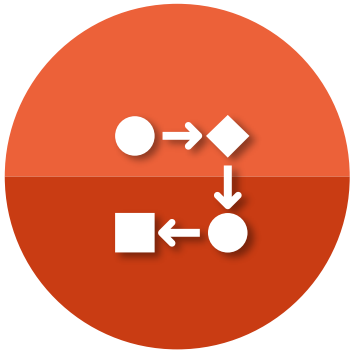
Let me introduce…

- Peter De Laet
  - Trainer – Training manager
  - LEARNIA
  - peter.delaet@learnia.be

Our Journey in 3 sessions

INTRO

What is BI
Power Query
Power Pivot
Power BI



POWER QUERY

Extract
Transform
Load

MODELLING

Create Relations
Calculated Columns
Show/Hide/Sort

ANALYSIS

Create Measures
Aggregating expressions
Add Filters
CALCULATE

What is Business Intelligence

Business Intelligence

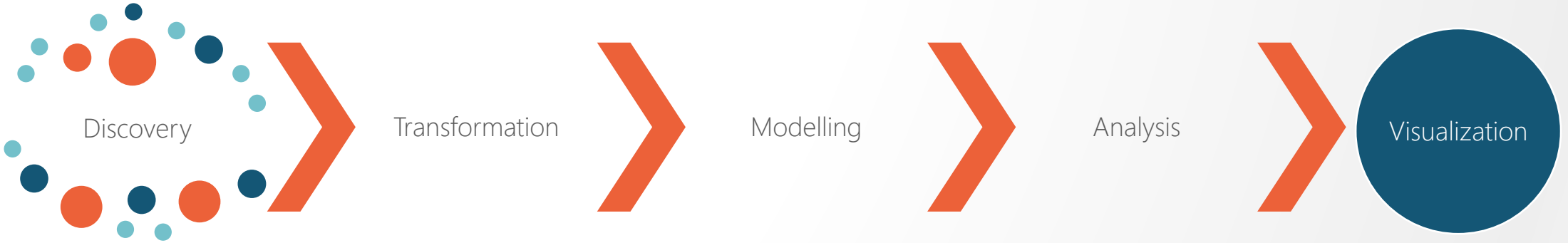
Definition

The term Business Intelligence (BI) refers to technologies, applications and practices for the collection, integration, analysis, and presentation of business information.

The purpose of Business Intelligence is to support better business decision making.

https://en.wikipedia.org/wiki/Business_intelligence

Process



1. Collect/Discover data from various systems
2. Transform and uniform the data
3. Model your data by creating relations between various tables
4. Analyse the data
5. Visualize the data

What is Power BI

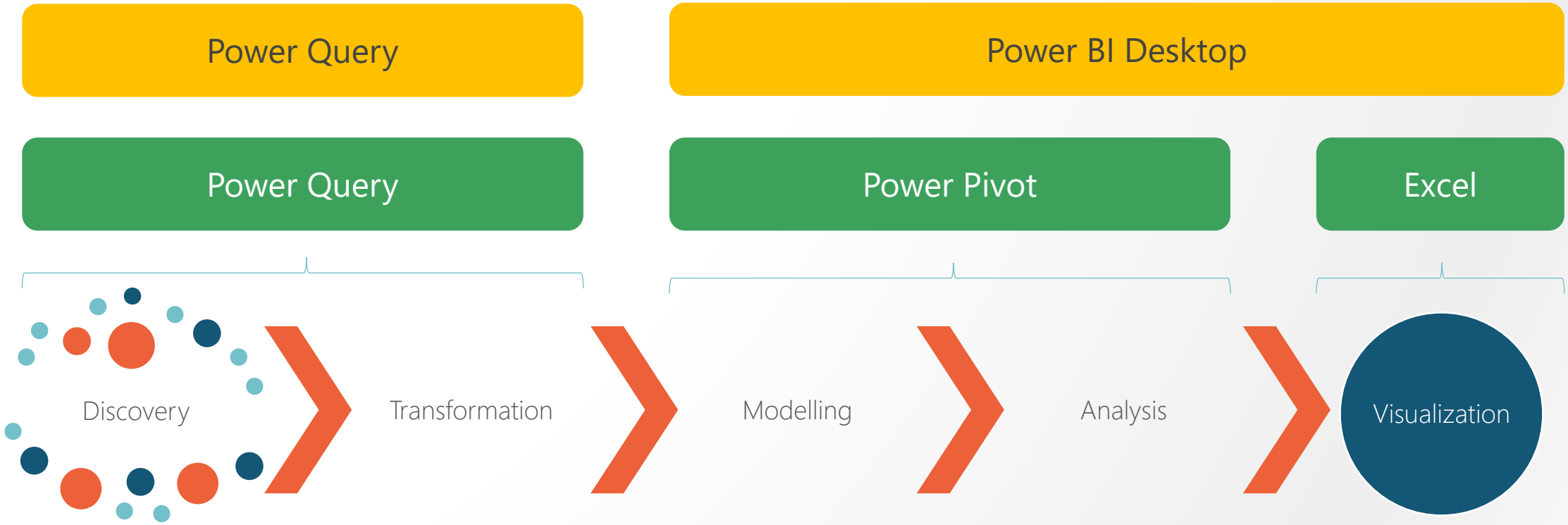
Microsoft self service BI

- Power BI Desktop & PowerBI.com

But also…

- A set of Add-ins for Excel
  - Power Pivot
  - Power Query
  - 3D Maps (aka Power Map)
  - Power View

Self-service-BI



Microsoft Self-service-BI

Power BI =

1. An online platform
   with a supporting desktop application

2. Set of add-ins for Excel
   (Power Pivot, Power Query …)

Both are independent but are using similar techniques

LEARNIA

Getting started with Power BI

Power BI Desktop vs Online

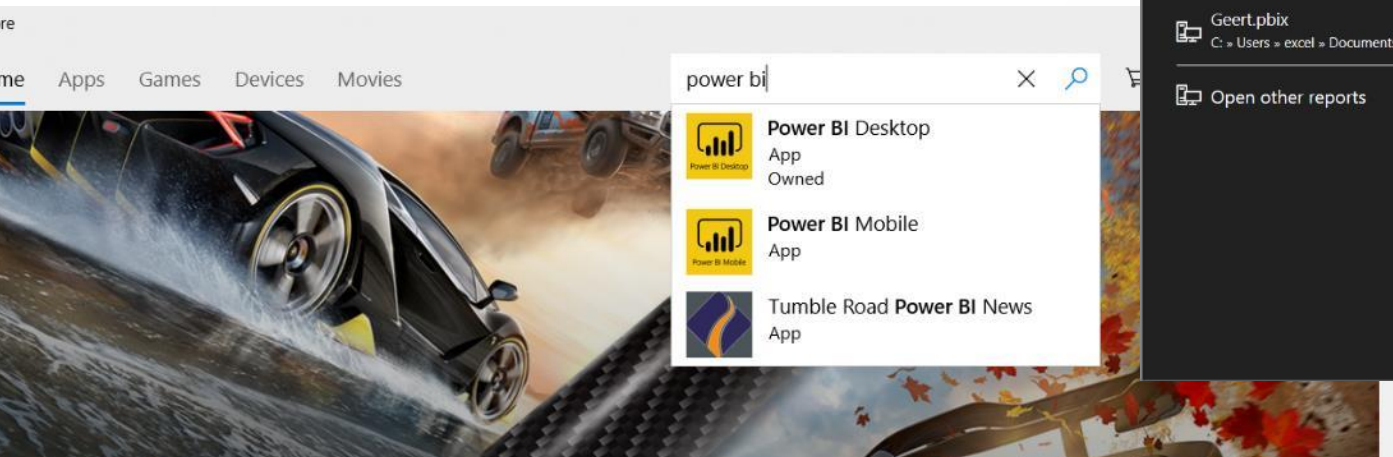
Power BI Desktop is a staging tool, designed to create reports that need to be published on PowerBI.com

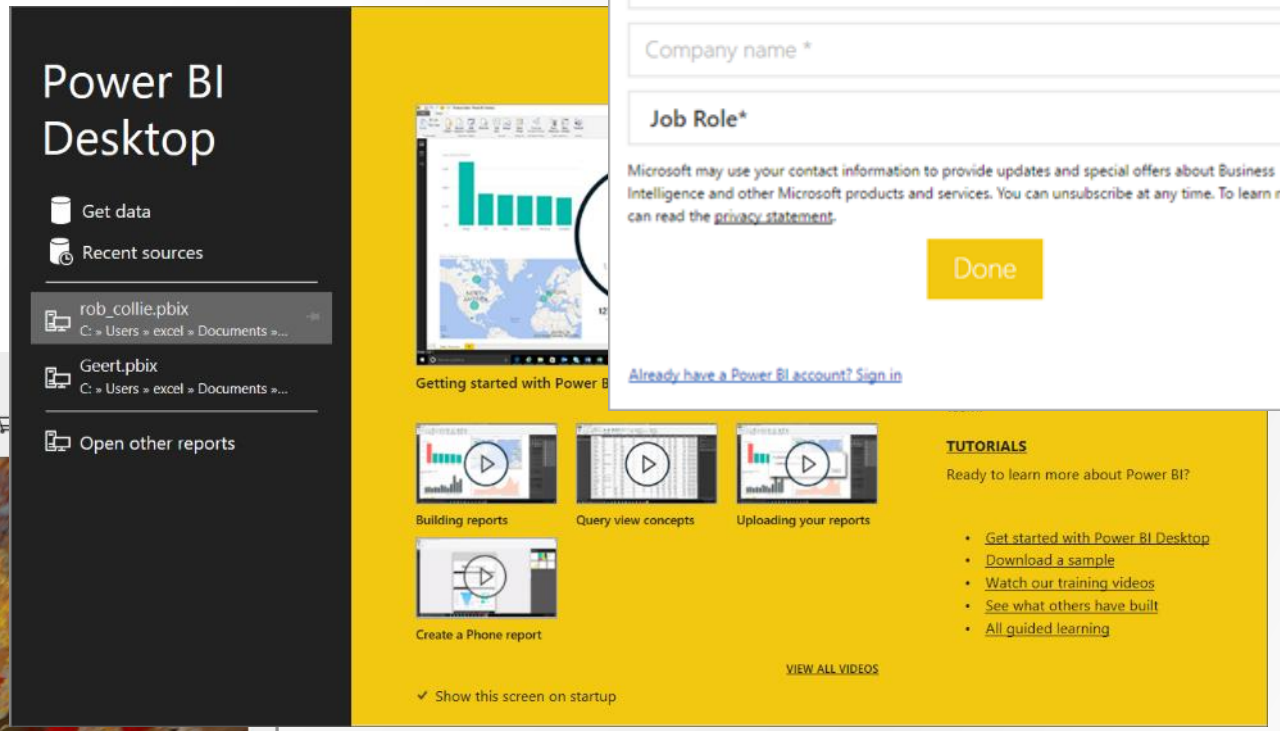
PowerBI.com reports

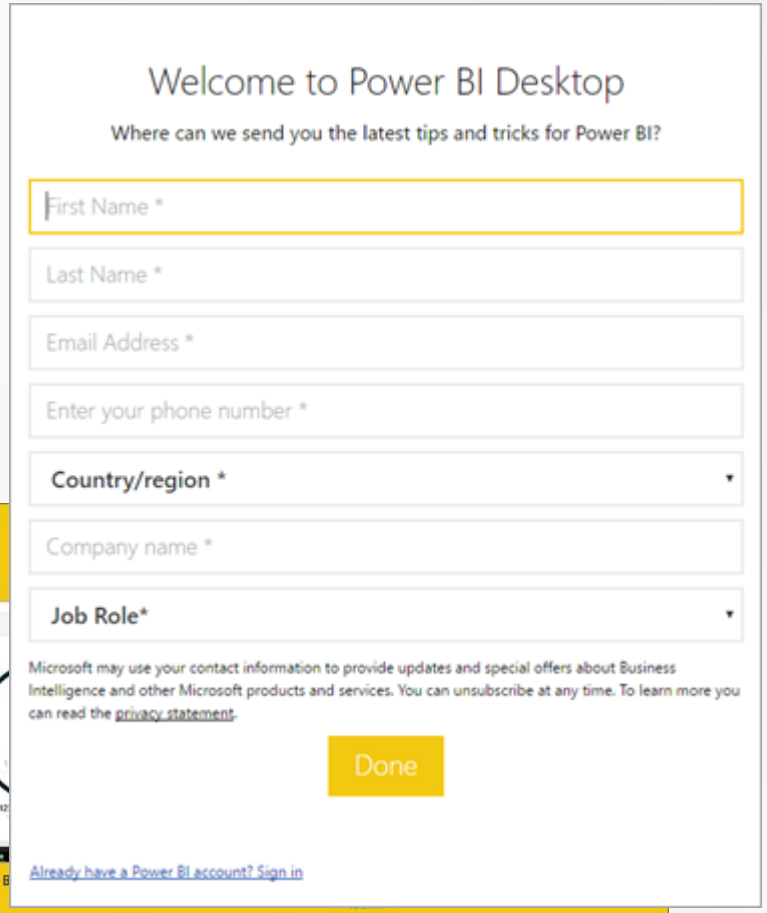
- Are read-only for consumers
- Are editable for you (although you better not)
- Can automatically be refreshed

Download

- From Microsoft Store / Software Centre
- Search for Power BI
- Sign in/up after installation

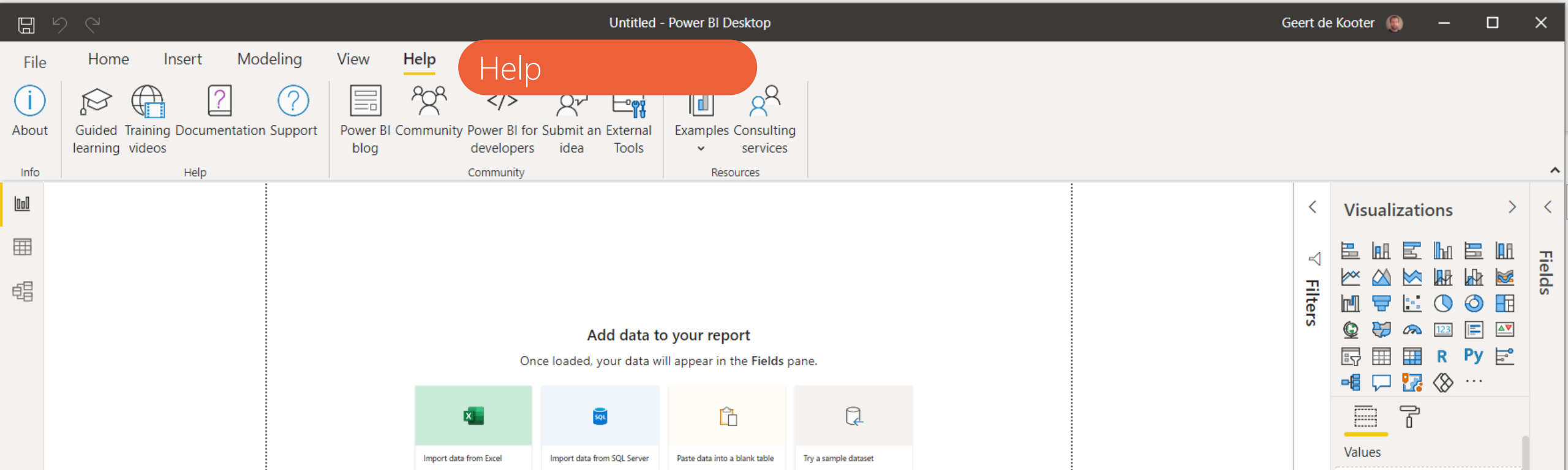


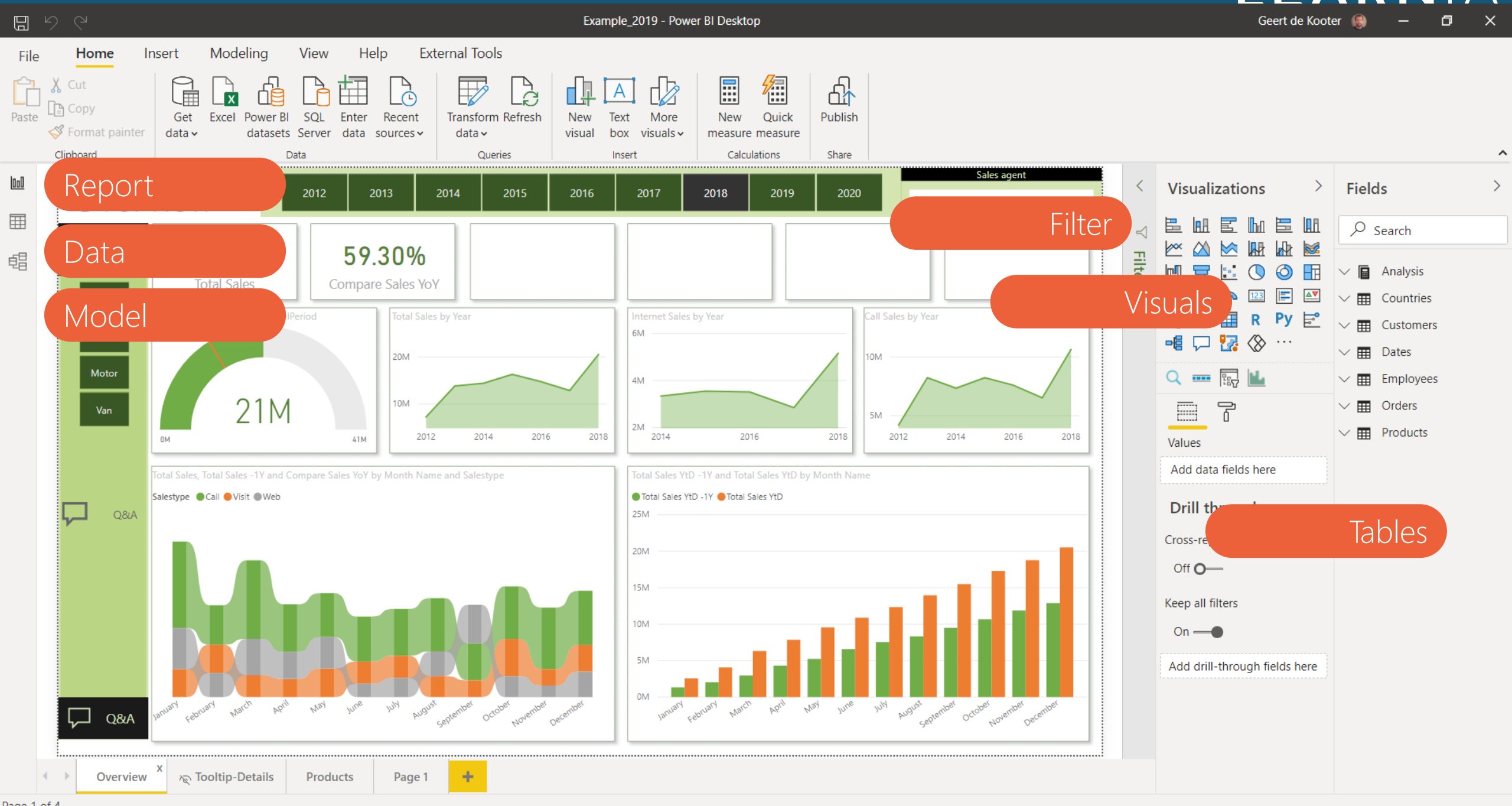




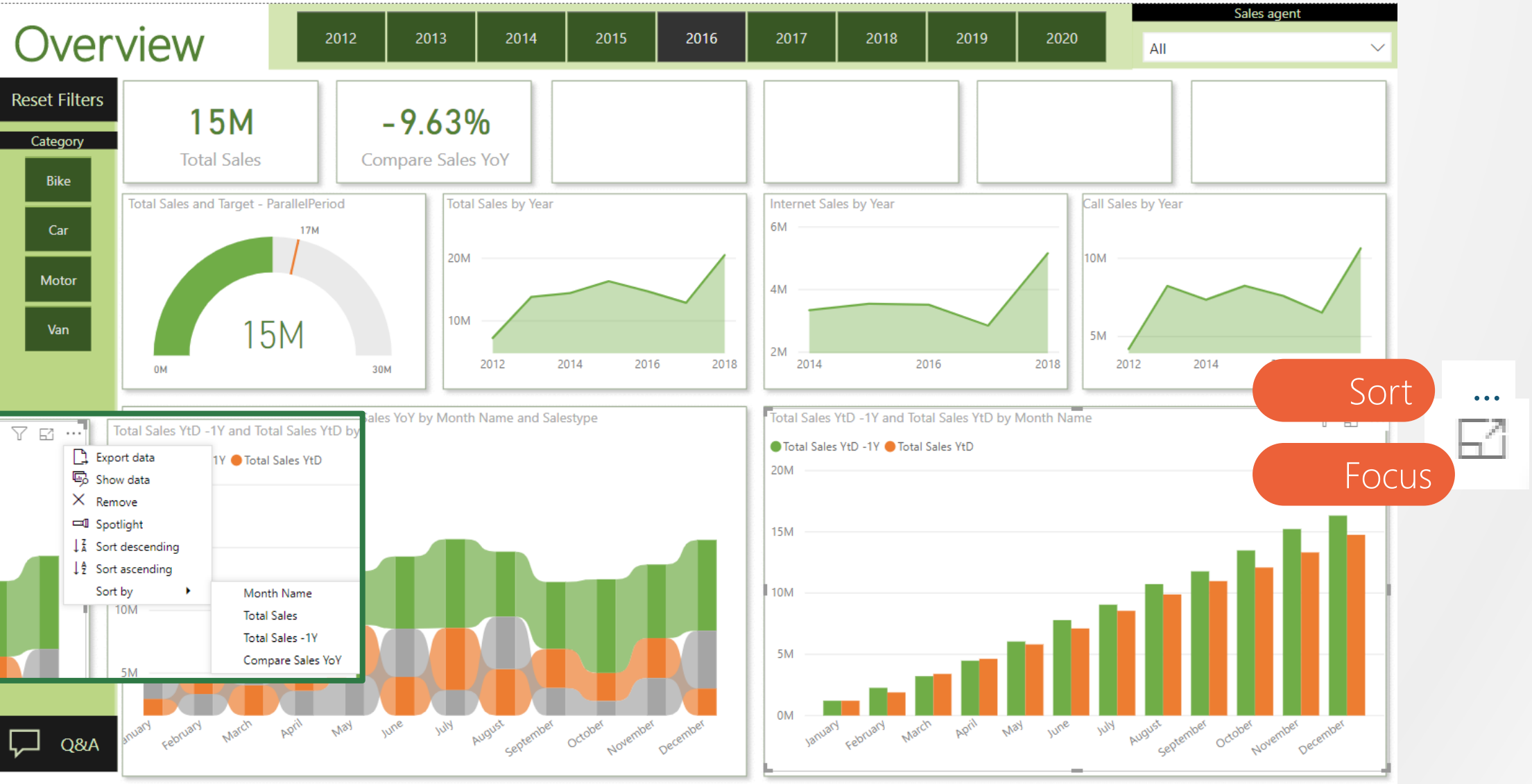
Help and resources

More resources at the end of the slideshow

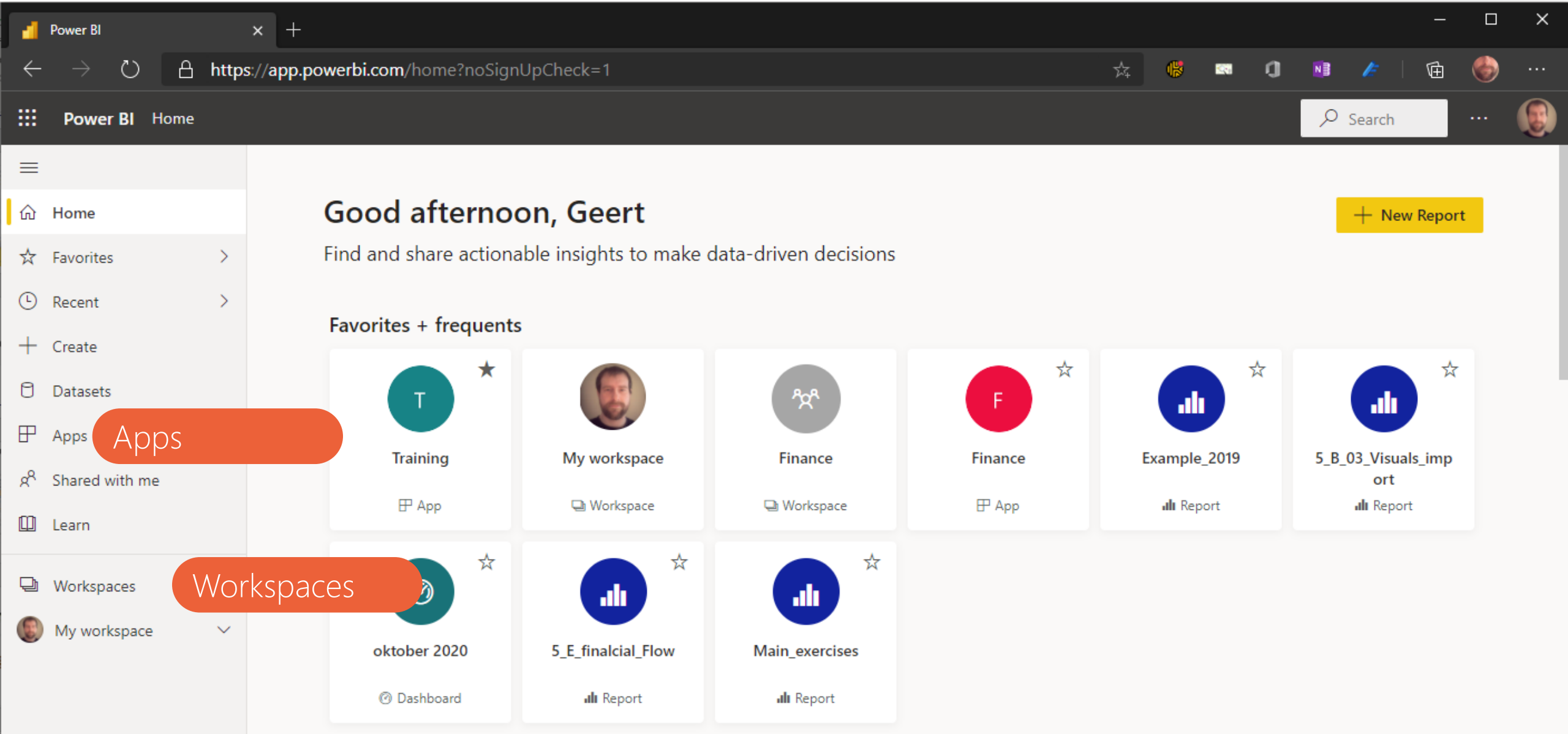



Example_2019 - Power BI Desktop
Geert de Kooter
File
Home
Insert
Modeling
View
Help
External Tools
Paste
Cut
Copy
Format painter
Clipboard
Get data
Excel
Power BI datasets
SQL Server
Enter data
Recent sources
Data
Transform data
Refresh
Queries
New visual
Text box
More visuals
Insert
New measure
Quick measure
Calculations
Publish
Share
Report
Data
Model
Filter
Visuals
Tables
2012
2013
2014
2015
2016
2017
2018
2019
2020
Sales agent
59.30%
Compare Sales YoY
Total Sales
21M
Total Sales by Year
Internet Sales by Year
Call Sales by Year
Motor
Van
Q&A
Total Sales, Total Sales -1Y and Compare Sales YoY by Month Name and Salestype
Salestype Call Visit Web
Total Sales YtD -1Y and Total Sales YtD by Month Name
Total Sales YtD -1Y
Total Sales YtD
Visualizations
Values
Add data fields here
Drill through
Cross-report
Off
Keep all filters
On
Add drill-through fields here
Fields
Search
Analysis
Countries
Customers
Dates
Employees
Orders
Products
Overview
Tooltip-Details
Products
Page 1

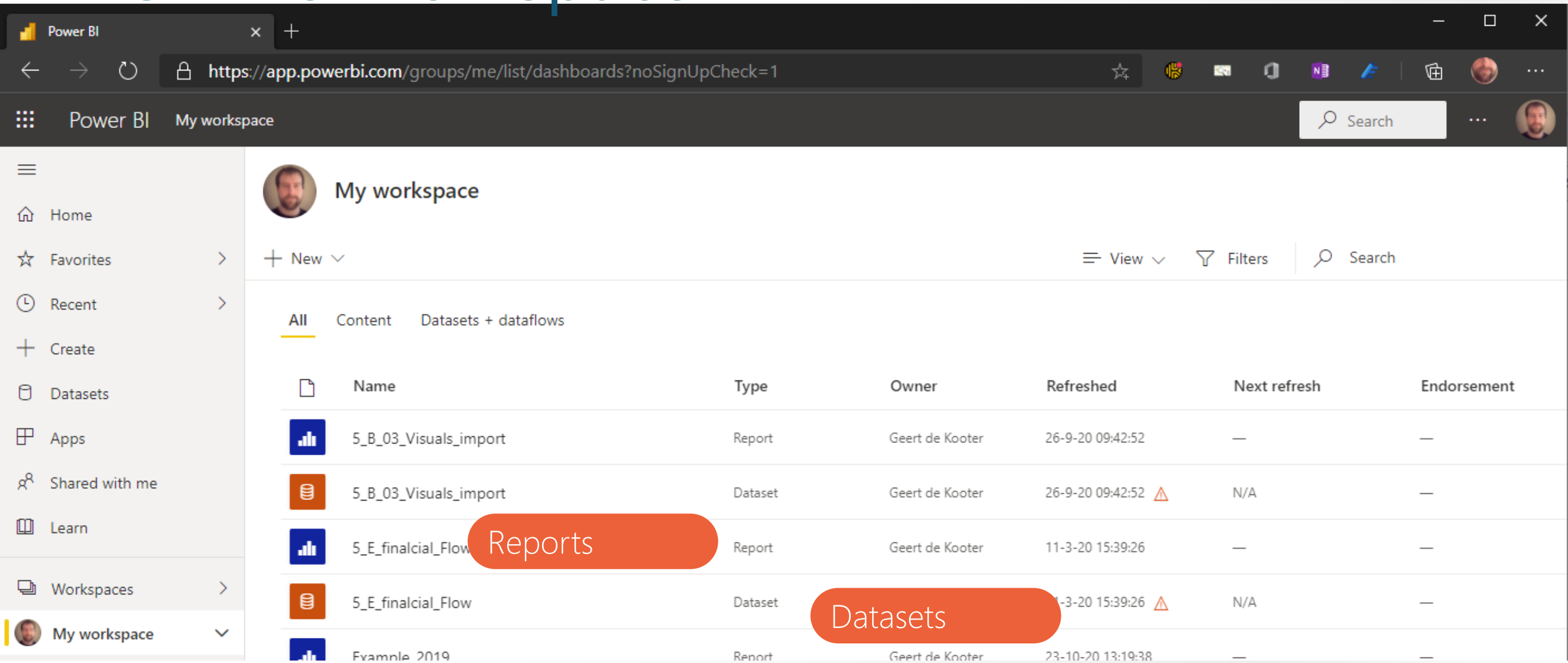
Use Reports


Overview
2012
2013
2014
2015
2016
2017
2018
2019
2020
Sales agent
All
Reset Filters
Category
Bike
Car
Motor
Van
15M
Total Sales
-9.63%
Compare Sales YoY
Total Sales and Target - ParallelPeriod
17M
15M
0M
30M
Total Sales by Year
20M
10M
2012
2014
2016
2018
Internet Sales by Year
6M
4M
2M
2014
2016
2018
Call Sales by Year
10M
5M
2012
2014
Total Sales YtD -1Y and Total Sales YtD by
Export data
Show data
Remove
Spotlight
Sort descending
Sort ascending
Sort by
Month Name
Total Sales
Total Sales -1Y
Compare Sales YoY
Total Sales YtD -1Y and Total Sales YtD by Month Name
Total Sales YtD -1Y
Total Sales YtD
20M
15M
10M
5M
0M
January
February
March
April
May
June
July
August
September
October
November
December
Sort
Focus
Q&A

Online


Power BI
https://app.powerbi.com/home?noSignUpCheck=1
Power BI Home
Search
Home
Favorites
Recent
Create
Datasets
Apps
Shared with me
Learn
Workspaces
My workspace
Apps
Workspaces
Good afternoon, Geert
Find and share actionable insights to make data-driven decisions
New Report
Favorites + frequents
Training
App
My workspace
Workspace
Finance
Workspace
Finance
App
Example_2019
Report
5_B_03_Visuals_import
Report
oktober 2020
Dashboard
5_E_finalcial_Flow
Report
Main_exercises
Report

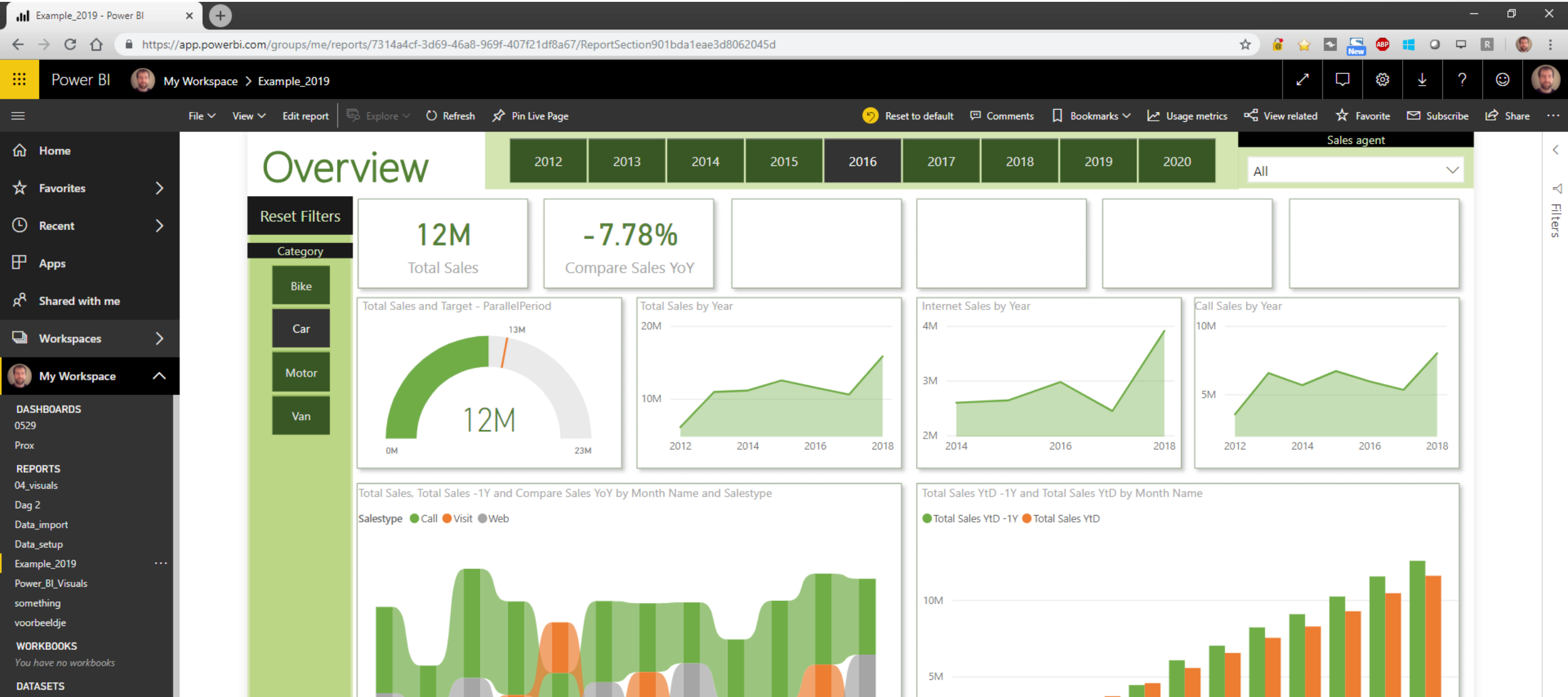
Online Workspace



Elements

- Dashboards
  - Overview of all your important visuals
- Reports
  - Your reports
- Workbooks
  - Shared Excel Files
- Data sets
  - The data models used to create reports

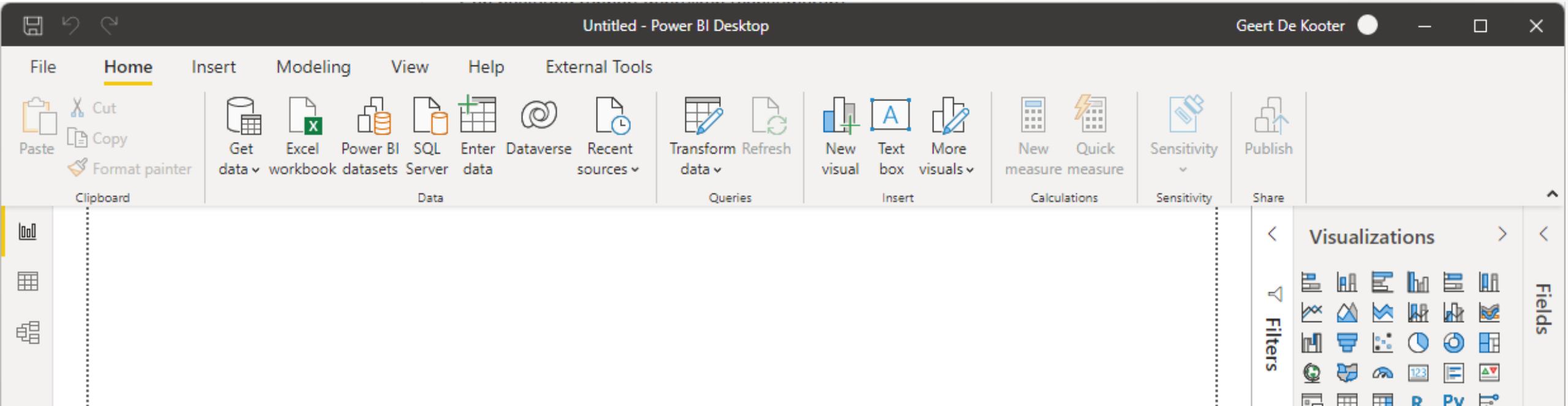
Consuming Online Reports



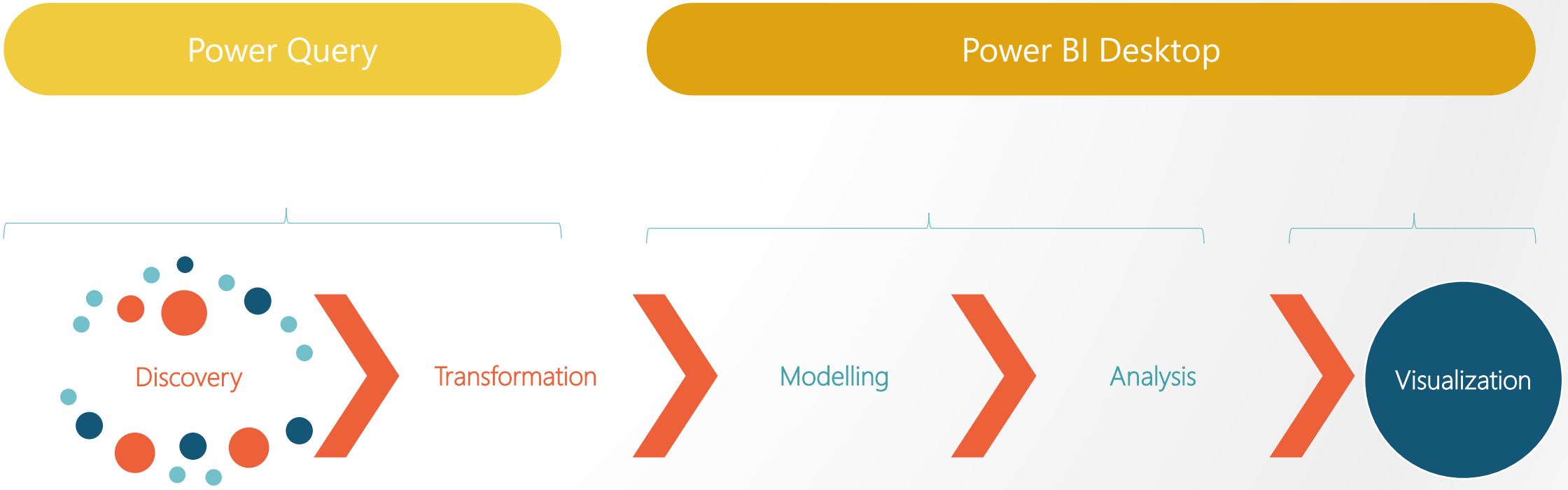
Create a report

Start a new Power BI report

- Start Power BI Desktop
  - Per session there is one PBIX-file
  - Power BI Desktop can run multiple sessions simultaneously
  - Never without PBIX
- Save the file as QuickRun.pbix



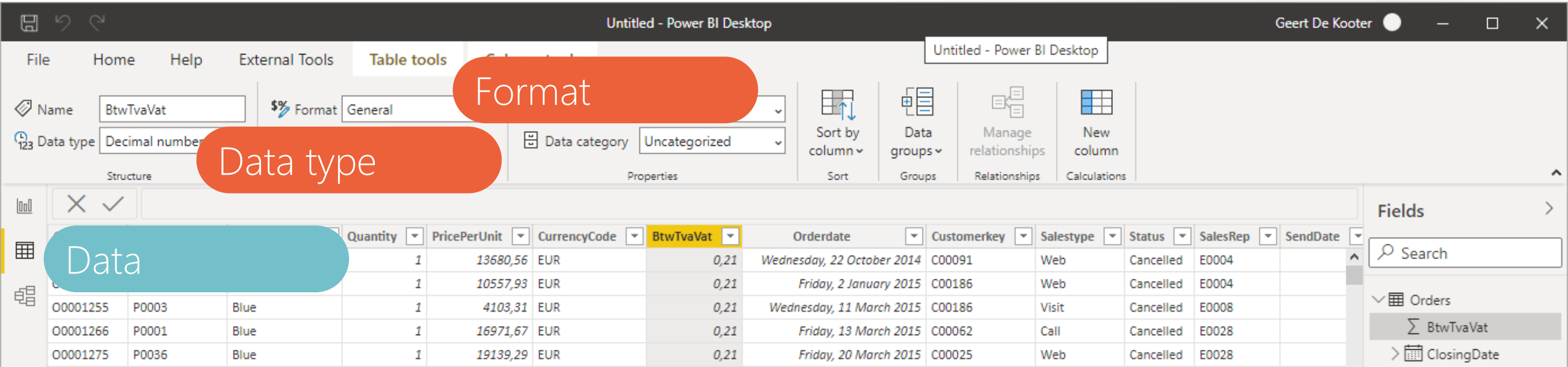
Self-service-BI



Power BI - Demo

DEMO

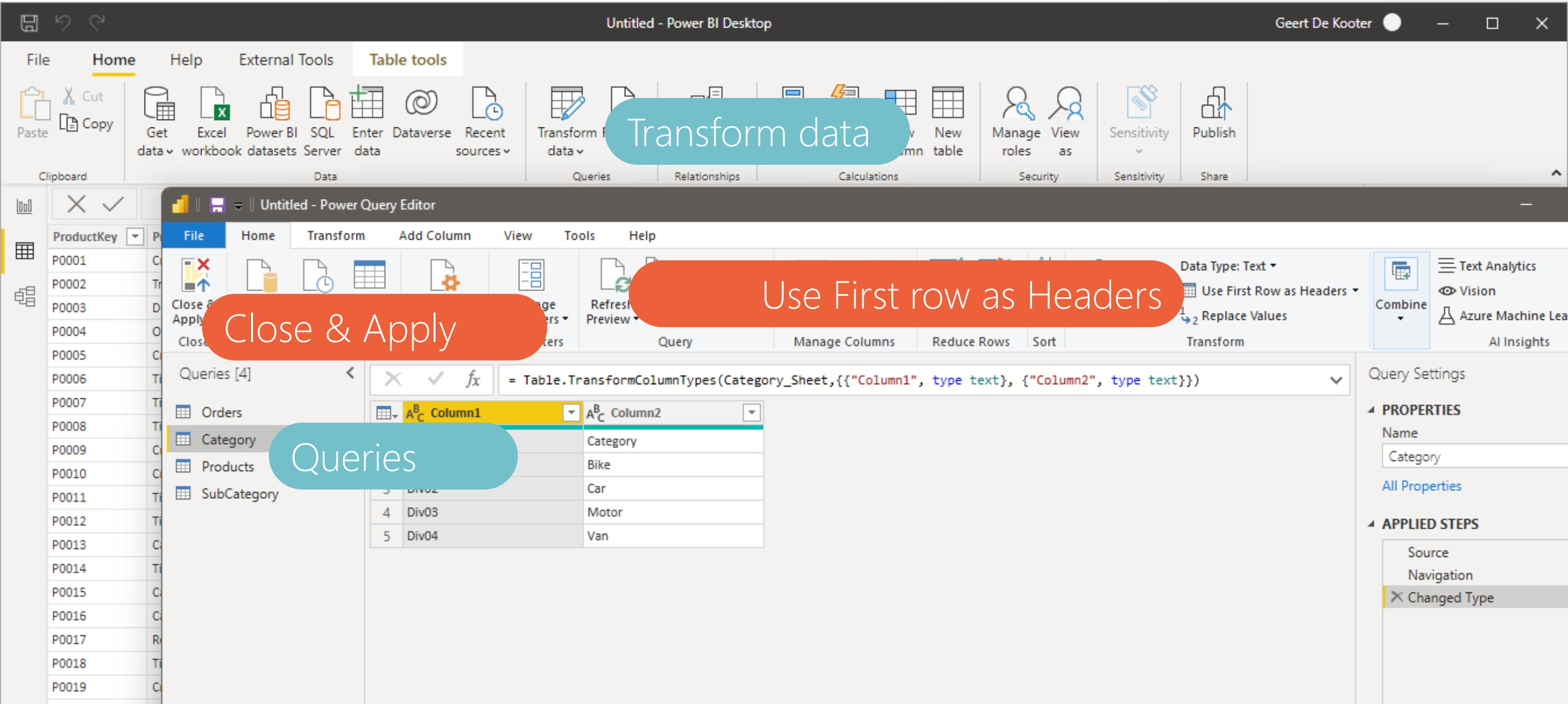
1. Load the data from Source Data / Orders.xlsx
2. Select the sheet “Orders”
3. Click on Load
4. Switch to the Data view
5. Check the datatypes



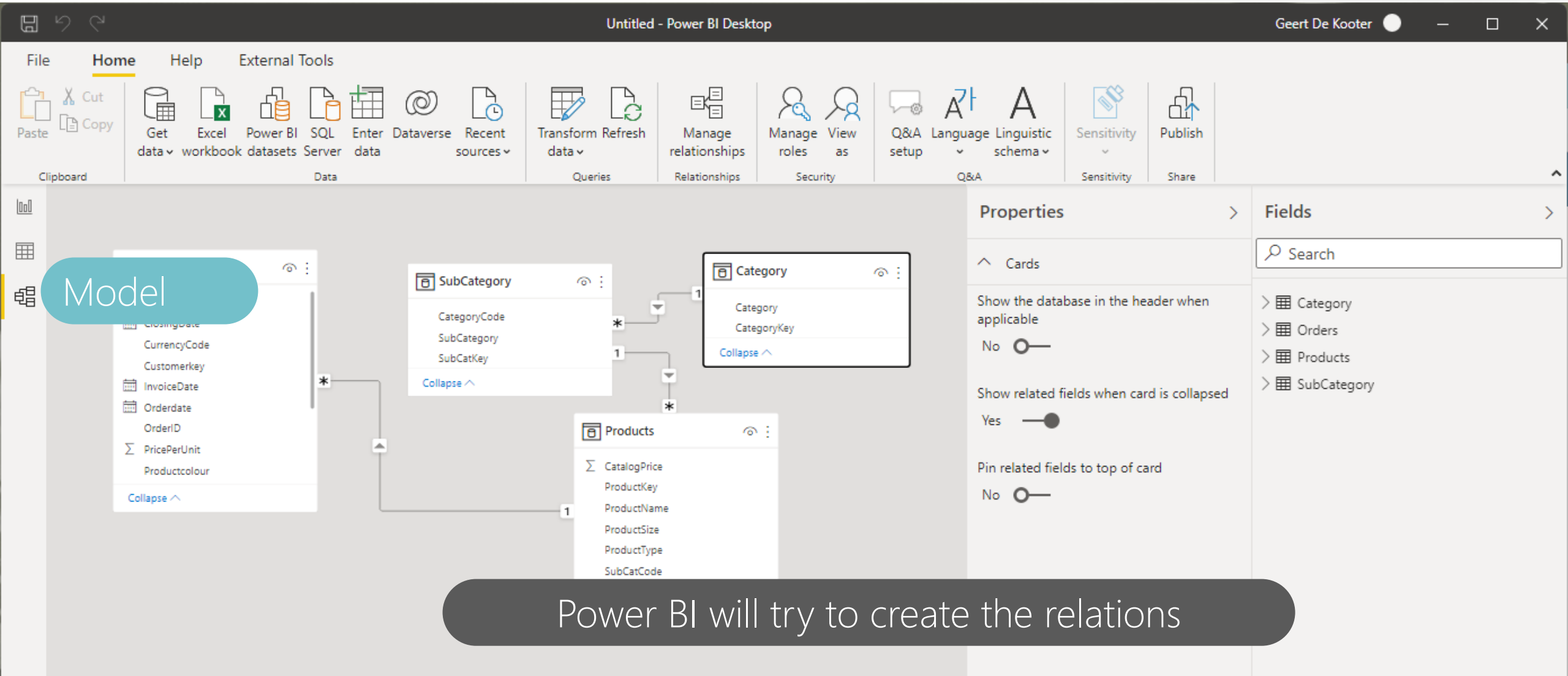
Exercise

- Load customers from Source Data / DimCustomers.xlsx
  - 3 times the same data
  - Select the Excel Table
- Change the name to "Customers"
- Load products from the file Source Data / Products.xlsx
  - 3 different sheets
  - Select all sheets
- Check and fix the queries
  - Category, the first row is not recognized
  - Subcategory, the first row is not recognized
  - Home/ Transform data

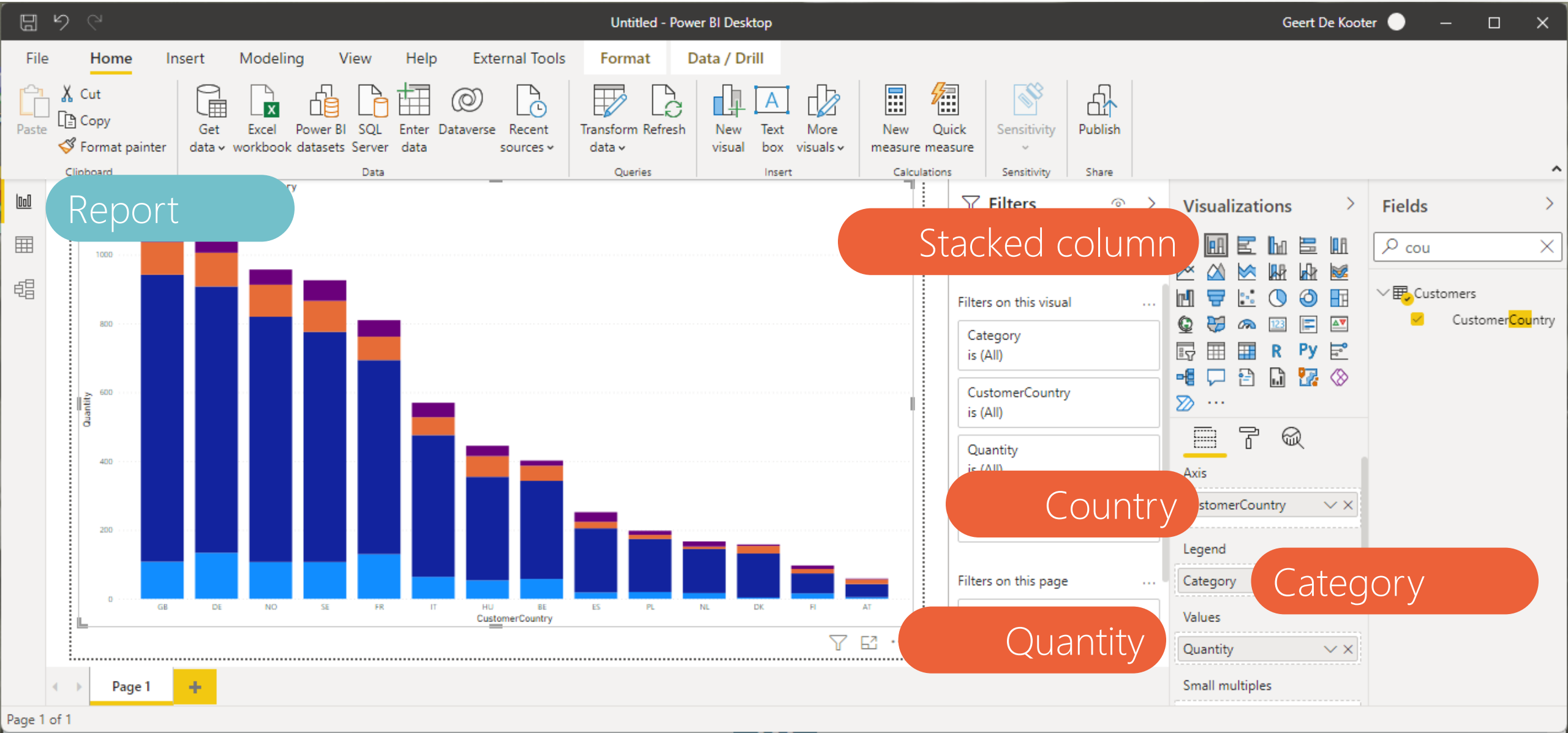
Power Query – fix de headers



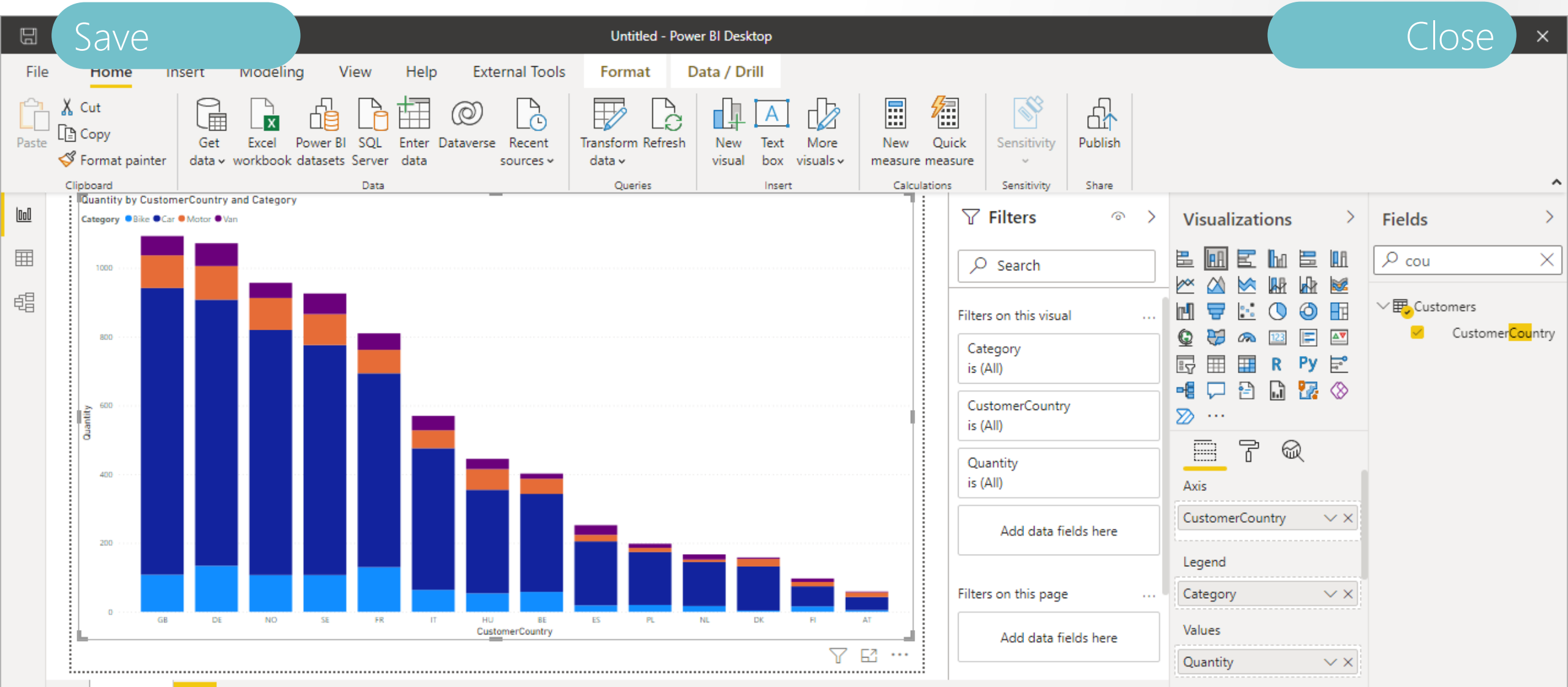
Modelling


Untitled - Power BI Desktop
Geert De Kooter
File
Home
Help
External Tools
Paste
Cut
Copy
Get data
Excel workbook
Power BI datasets
SQL Server
Enter data
Dataverse
Recent sources
Transform data
Refresh
Manage relationships
Manage roles
View as
Q&A setup
Language
Linguistic schema
Sensitivity
Publish
Clipboard
Data
Queries
Relationships
Security
Q&A
Sensitivity
Share
Model
CurrencyCode
Customerkey
InvoiceDate
Orderdate
OrderID
PricePerUnit
Productcolour
Collapse
SubCategory
CategoryCode
SubCategory
SubCatKey
Collapse
Category
Category
CategoryKey
Collapse
Products
CatalogPrice
ProductKey
ProductName
ProductSize
ProductType
SubCatCode
Properties
Cards
Show the database in the header when applicable
No
Show related fields when card is collapsed
Yes
Pin related fields to top of card
No
Fields
Search
Category
Orders
Products
SubCategory
Power BI will try to create the relations

Create a chart



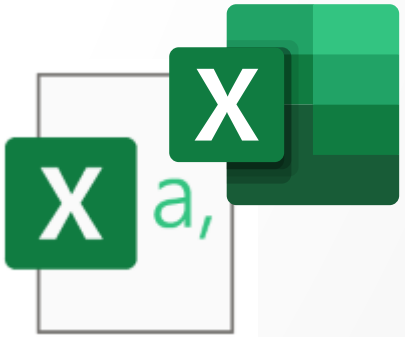
Save and close the file



Power Query - Intro

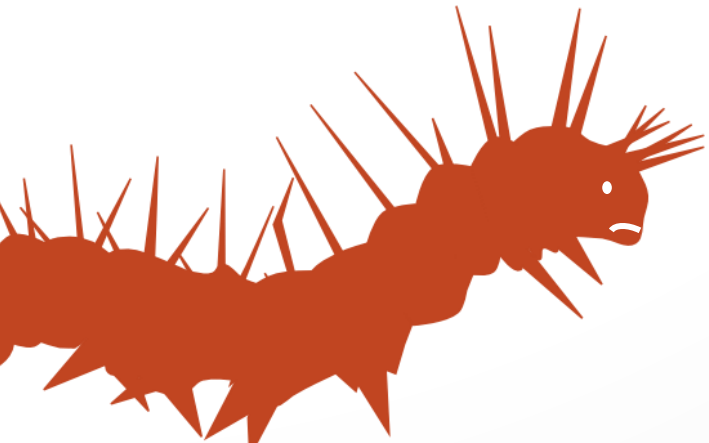
Power Query Data sources

- Power Query can handle 45+ data sources in Power BI
- Power Query can handle 25+ data sources in Excel



Power Query

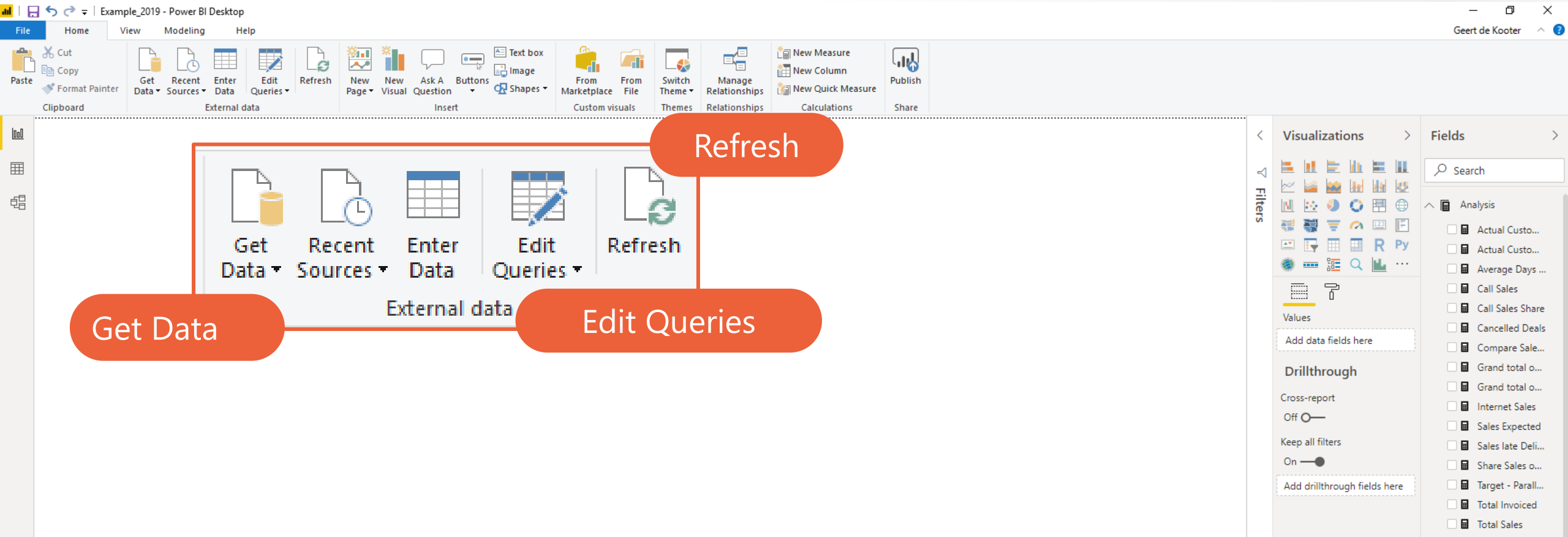
- Extract the data
- Transforms the data
  - Format, Remove rows, UnPivot, Date2Month …
- Merges data
  - Import from folder, replace …
- Loads the data into your model



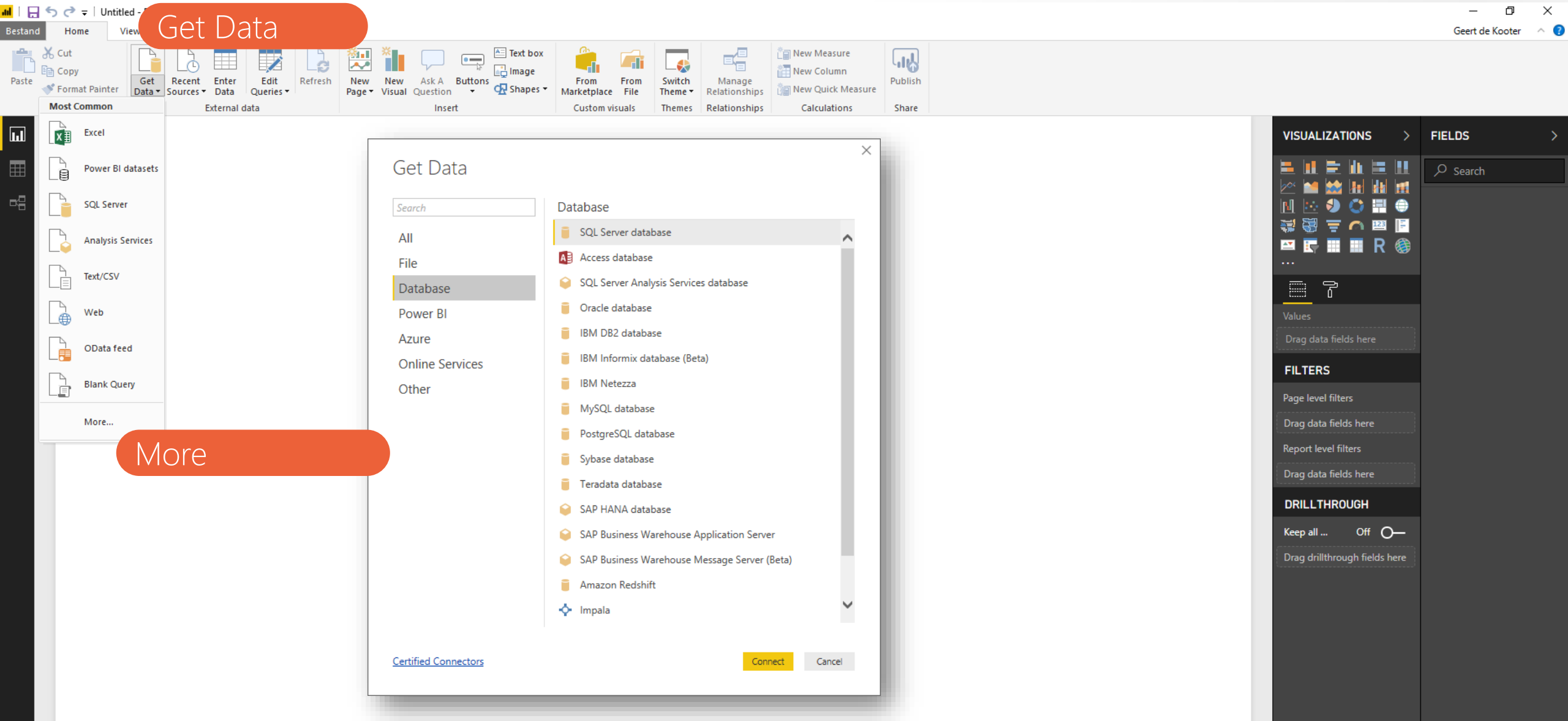
Example

1. Create a new Excel File
2. Use Data/ Get Data / From File 2_A_01.xlsx
3. Select column 1-3
4. Transform/ Unpivot other columns
5. Rename the columns “Attribute” and “Value”
6. Change the datatype
7. …
8. Home > Close and Load

Power BI – Query interface



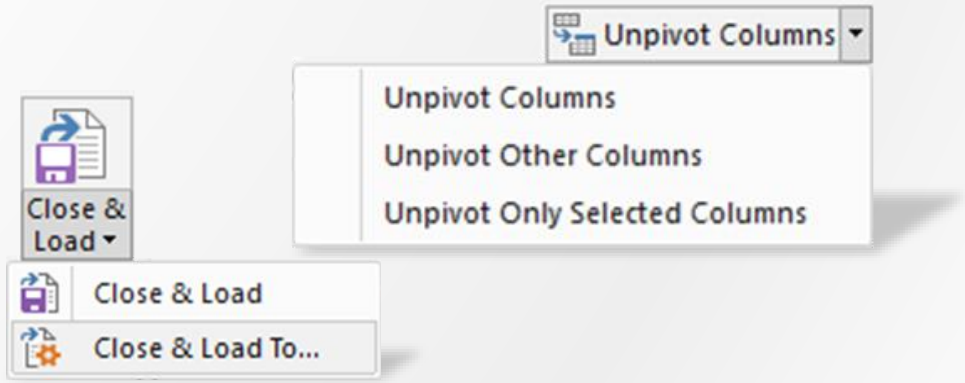
Get Data


Get Data
Bestand
Home
View
Geert de Kooter
Paste
Cut
Copy
Format Painter
Get Data
Recent Sources
Enter Data
Edit Queries
Refresh
New Page
New Visual
Ask A Question
Buttons
Text box
Image
Shapes
From Marketplace
From File
Switch Theme
Manage Relationships
New Measure
New Column
New Quick Measure
Publish
External data
Insert
Custom visuals
Themes
Relationships
Calculations
Share
Most Common
Excel
Power BI datasets
SQL Server
Analysis Services
Text/CSV
Web
OData feed
Blank Query
More...
More
Get Data
Search
All
File
Database
Power BI
Azure
Online Services
Other
Database
SQL Server database
Access database
SQL Server Analysis Services database
Oracle database
IBM DB2 database
IBM Informix database (Beta)
IBM Netezza
MySQL database
PostgreSQL database
Sybase database
Teradata database
SAP HANA database
SAP Business Warehouse Application Server
SAP Business Warehouse Message Server (Beta)
Amazon Redshift
Impala
Certified Connectors
Connect
Cancel
VISUALIZATIONS
FIELDS
Search
Values
Drag data fields here
FILTERS
Page level filters
Drag data fields here
Report level filters
Drag data fields here
DRILLTHROUGH
Keep all ...
Off
Drag drillthrough fields here

Interface

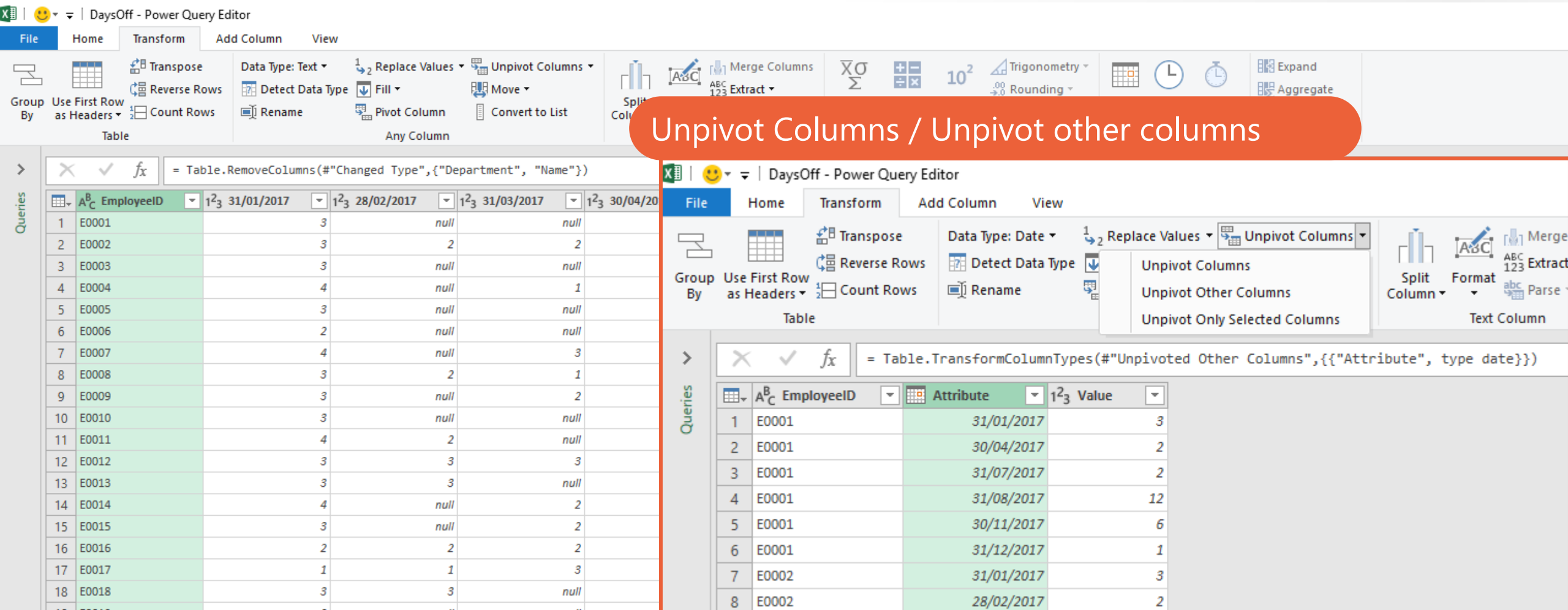
- Formula Bar
  - View/ Layout/ ☑ Formula Bar
- Transform-tab
  - Change the selected column
- Add column-tab
  - Create a new column by transforming the selected column

- Buttons have “secret” options



Unpivot Columns

- Transform/ Unpivot Columns



Data connections

External Excel files into Excel

1. Create a new file
2. Get Data > From File > From Workbook
3. Select 2_B_01_data.xlsx
4. Select
   - The Table tbl_Products
   - The Sheet Production
5. Tbl_products: Split the column Description
6. Production: Fill the missing years
7. Home > Close & Load

Text/CSV-Files

1. Create a new Excel file
2. Get Data > From File > From Text/CSV
3. Select B_02_data.csv
4. Close the query

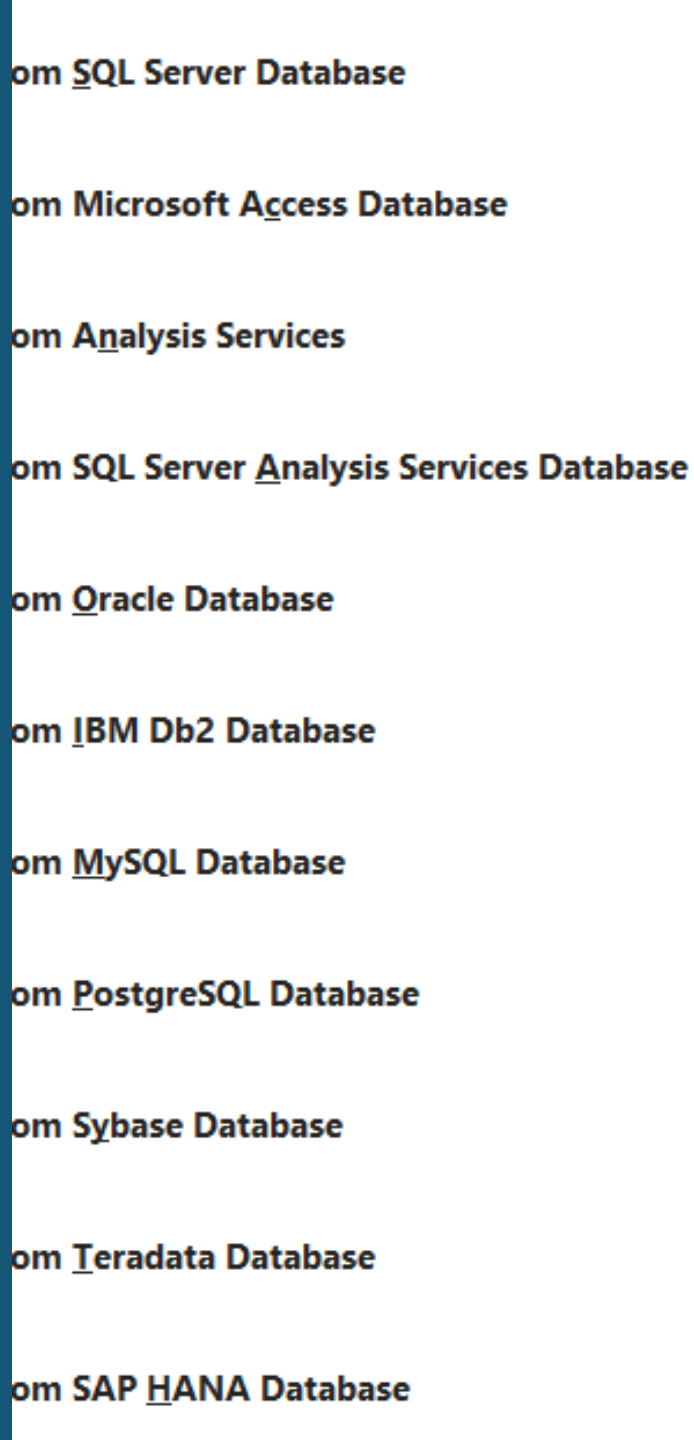
Text/CSV-Files

1. Create a new file
2. Get Data > From File > From Text/CSV
3. Import 2_B_03_data.txt
4. Promote the first row
   The Startdate is not correctly recognized
   - File > Options & Settings > Query options > Regional settings
     English: United States
   - Or use Add Column / Column from Example
5. Refresh Preview
6. Change the datatype of Startdate
7. Close the query

Databases

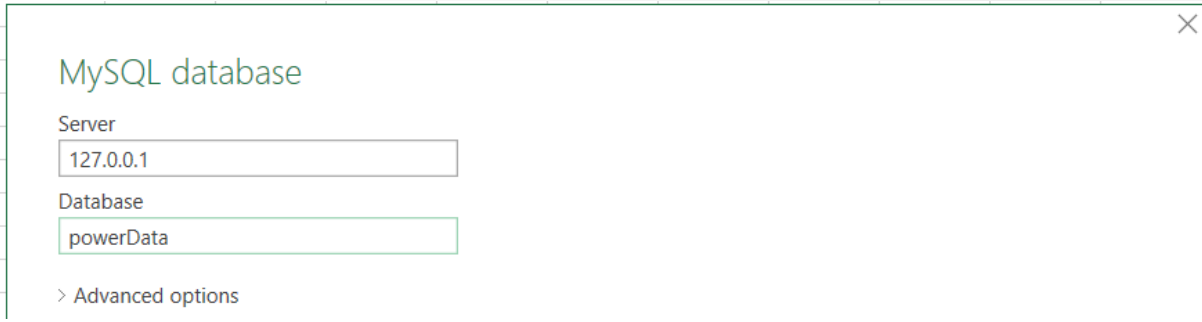
- Multiple databases are supported
  - Get Data > From Database

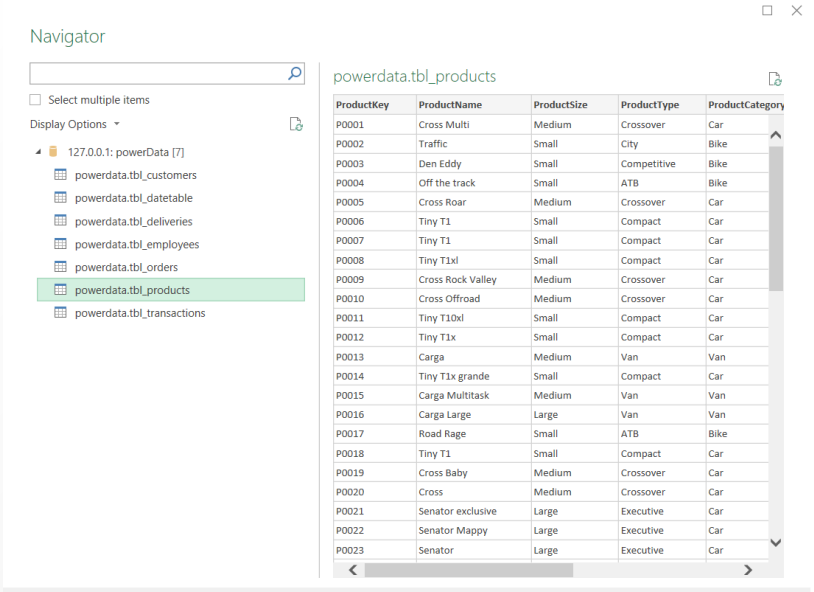
- Via ODBC connections the support can be extended
  - Get Data > From Other Sources



Demo MYSQL/SQL

1. Get Data > From Database > From MySQL Database
2. Address and Database name
3. Username and password
4. Select Tables
5. …





Access

1. Create a new File
2. Get Data > From Database > From Microsoft Access …
3. Select 2_B_04_data.accdb or 2_B_04_data_alternative.accdb
4. Search and import Purchase orders or Orders
5. …
6. Close the query

Because of a version mismatch between the Access file and your version of Excel/Power BI, the import could fail.
https://docs.microsoft.com/en-us/power-bi/desktop-access-database-errors

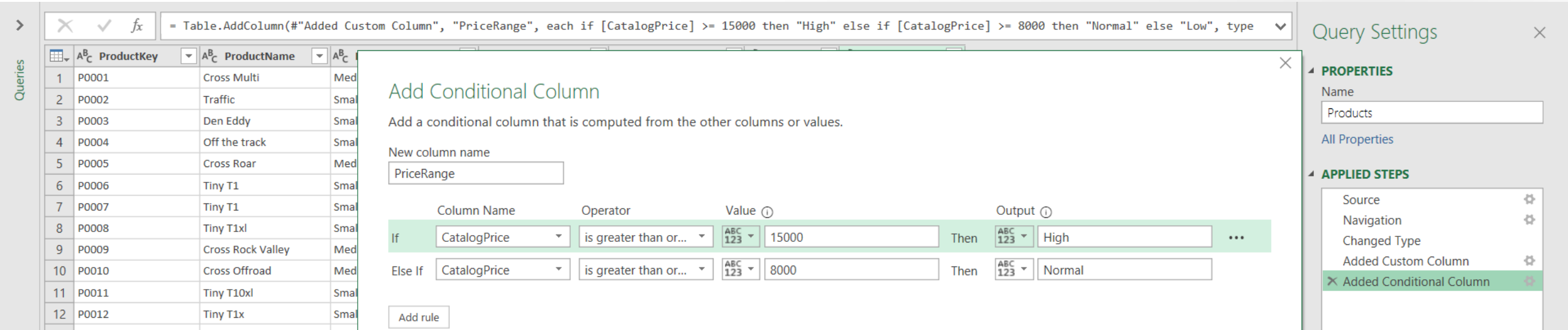
Transformations

Column from Examples

1. Create a new file
2. Get Data > From File > From Workbook
3. Select 2_C_01_data.xlsx
4. Select the table Products
5. Select the column Description
6. Click Add Column > Column from Examples
7. Create the next 3 examples
    1. 2e word
    2. 2e word in CAPITALS
    3. “Initials” based on the first character of every word
8. Click OK to confirm

Conditional column

1. Click Add Column > Conditional column
   1. Name PriceRange
   2. Critaria: High >15000,
      Normal>8000 else Low
2. Close the query and save the file as 2_C_01.xlsx



Date transformations

1. Create a new file
2. Get Data > From File > From Workbook
3. Select 2_C_02_data.xlsx
4. Add a column containing only the Starttime
5. Create multiple columns based on the column Start
    1. Year, Quarter, Month
    2. Monthname (Language!)
6. Create a Custom Column Duration
    1. =[End]-[Start]
    2. Transform Total Minutes & Roundup

From Text to column

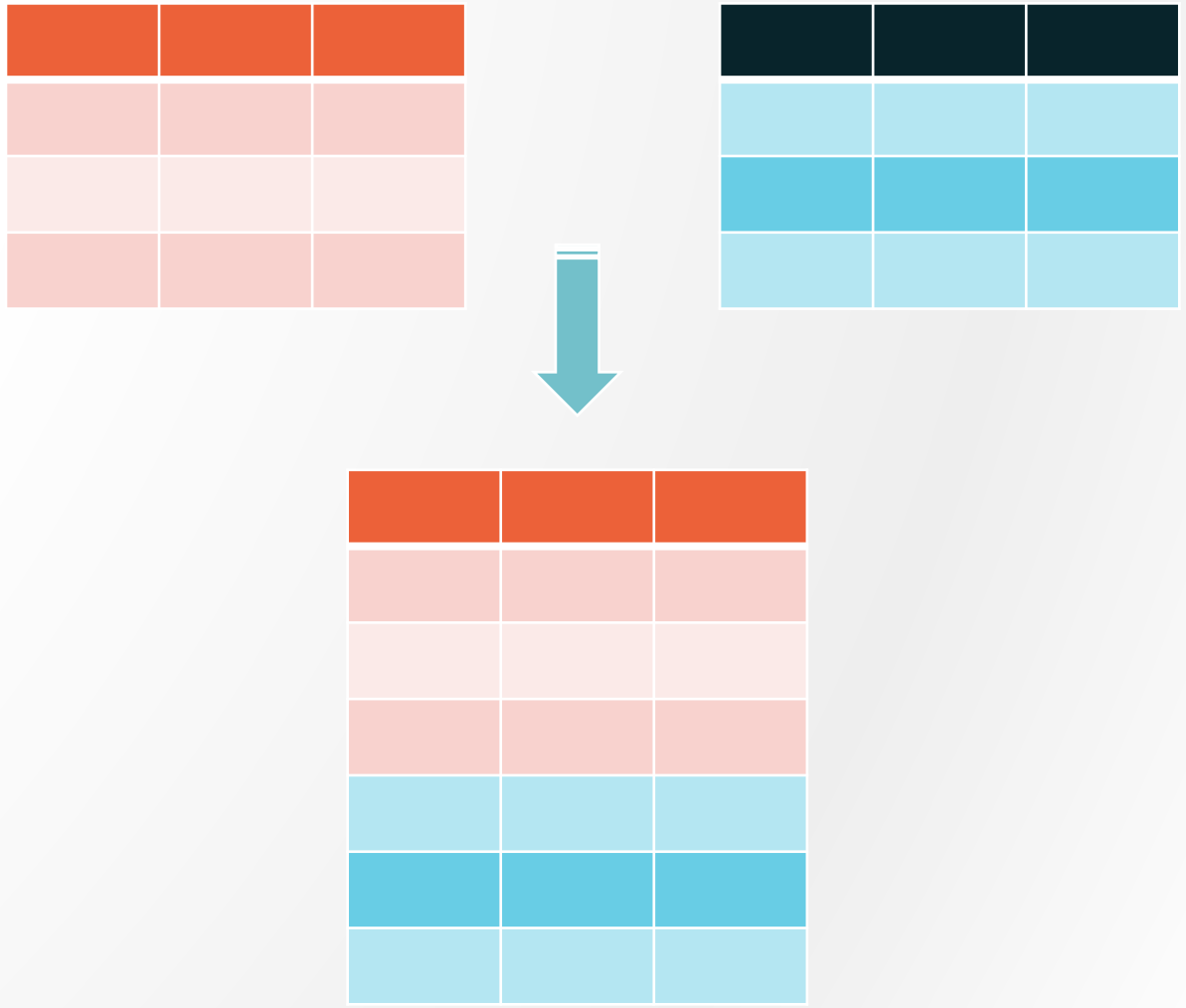
1. Duplicate the previous query
   - the data is still coming from 2_C_02_data.xlsx
2. Delete all columns except ID and Categories
3. Choose Split Column
4. Choose Split into Rows (adv options)
5. Remove redundant spaces using Trim
   - Transform > Format > Trim
6. Close the queries and save the file as 2_C_02.xlsx

Append Tables

Appending Tables

Situations

- 2 Separate lists
- All files from a folder
- Multiple sheets from 1 workbook



2 Separate lists

1. Create a new file
2. Import 2_D_01_data1.xlsx and 2_D_01_data2.csv
3. Append these 2 queries using
   Append Queries as new
4. Call the query Contacts
5. Solve the issues, by renaming the columns

6. Close the queries, export only the combined list
7. Save the file as 2_D_01

All files from a folder

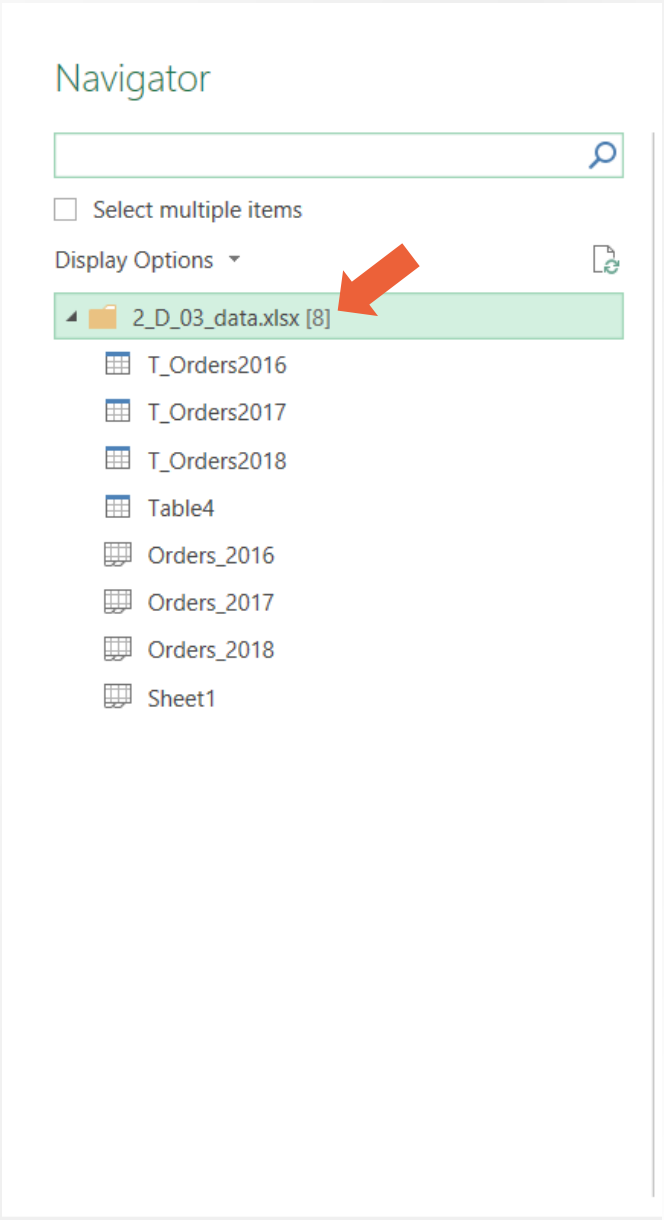
1. Create a new file
2. Get Data > From File > From Folder
3. Select the folder 2_D_02_data
4. Use Combine to combine the data by example

5. Save the file as 2_D_0201

Sheets from 1 file

1. Create a new file
2. Get data > from file > from excel
3. Select 2_D_03_data.xlsx
4. Select the folder icon and edit the query
5. Combine the T_Orders...-Tables

6. Save the file as 2_D_03



LEARNIA

Power Query - Exercises

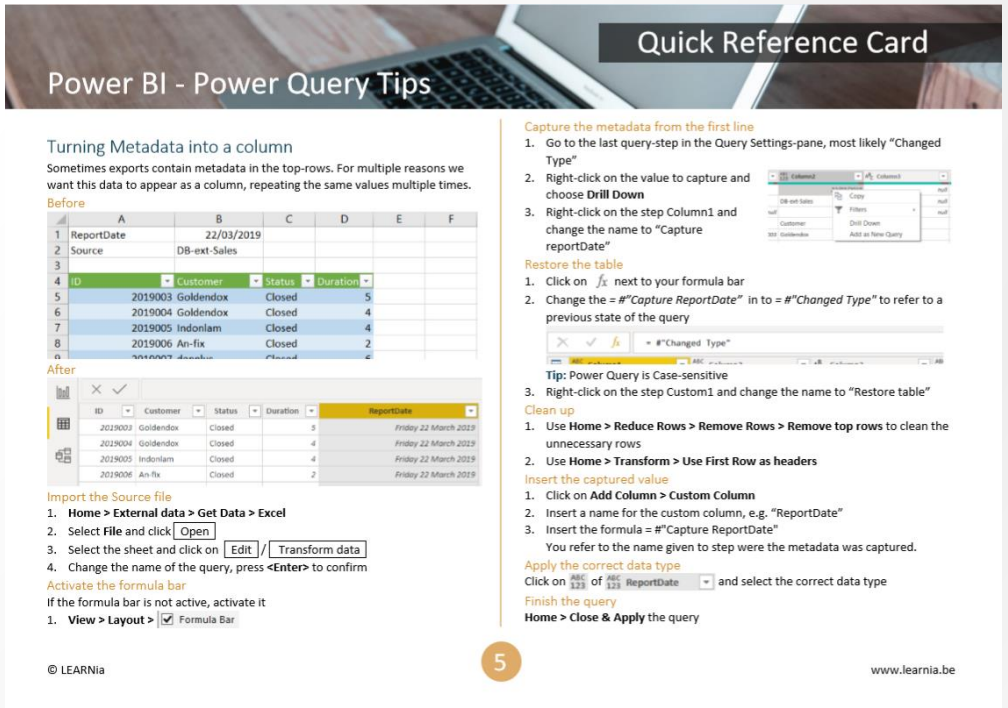
Exercise Orders

1. Create a new file and import 2_E_01_Orders.xlsx
2. Save the file as 2_E_01
3. Remove the first rows with meta info
4. Turn the correct row into the header row
5. Remove "None" in sendDate and InvoiceDate
6. Remove the summary "Totals“ rows at the bottom
7. Close the query

Reuse meta info as a column

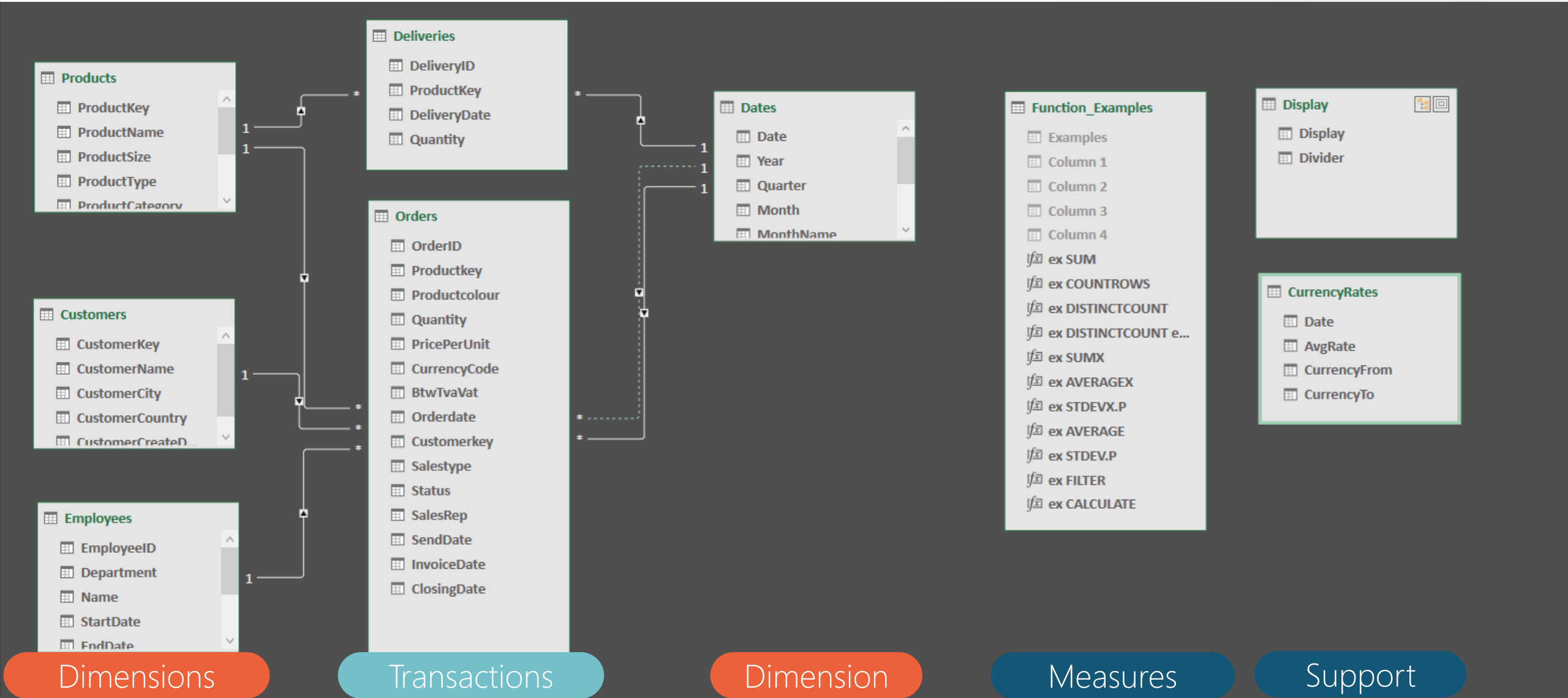
1. Create a new file and import 2_E_03_metainfo.xlsx
2. Save the file as 2_E_03
3. Reuse the meta info using the method from the handout

4. Close the query



Part 2 : Modelling

Data Model



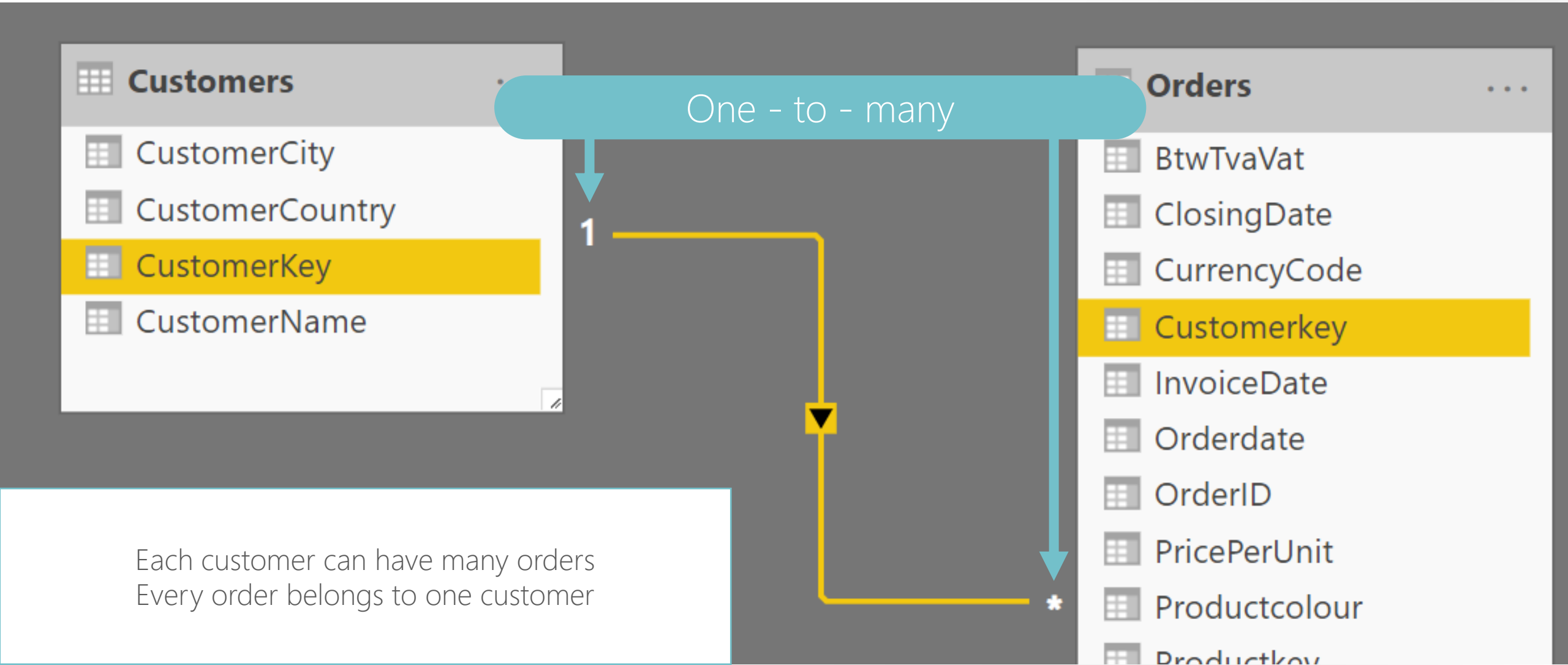
Dimensions | Transactions | Dimension | Measures | Support

Create relations - Demo

1. Open 3_A_01_Demo
2. Go to the diagram view
3. Drag & Drop corresponding fields

Dimension	Orders
• Customerkey	CustomerKey
• ProductKey	ProductKey
• EmployeeID	SalesRep
• Date	OrderDate
• Date	InvoiceDate

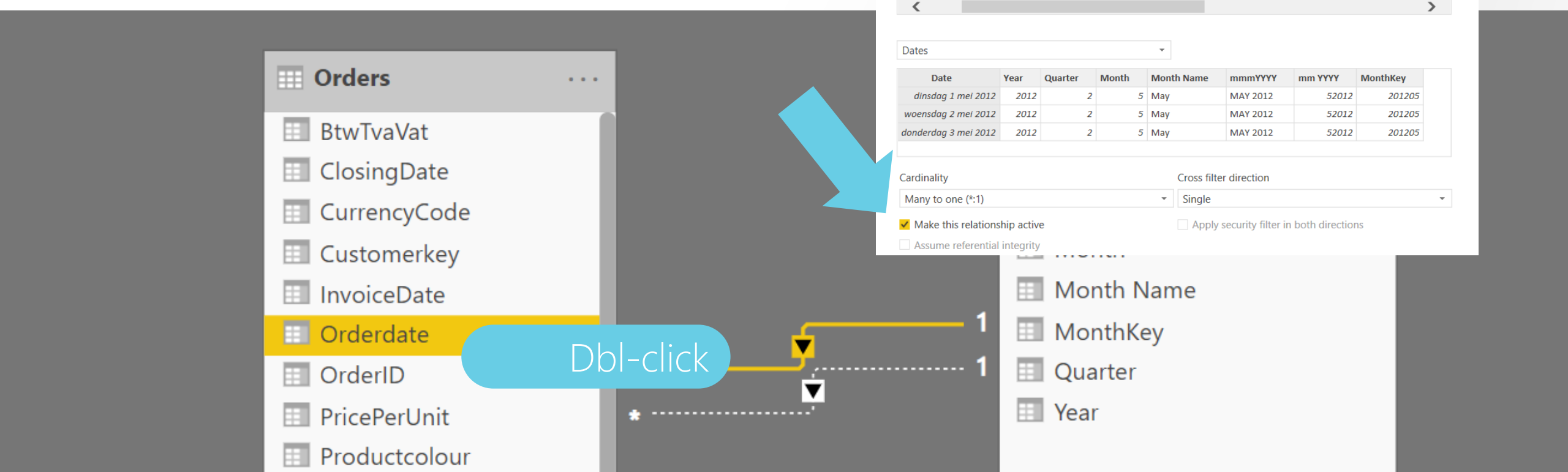
One-to-many



Each customer can have many orders
Every order belongs to one customer

Multiple (active) relations

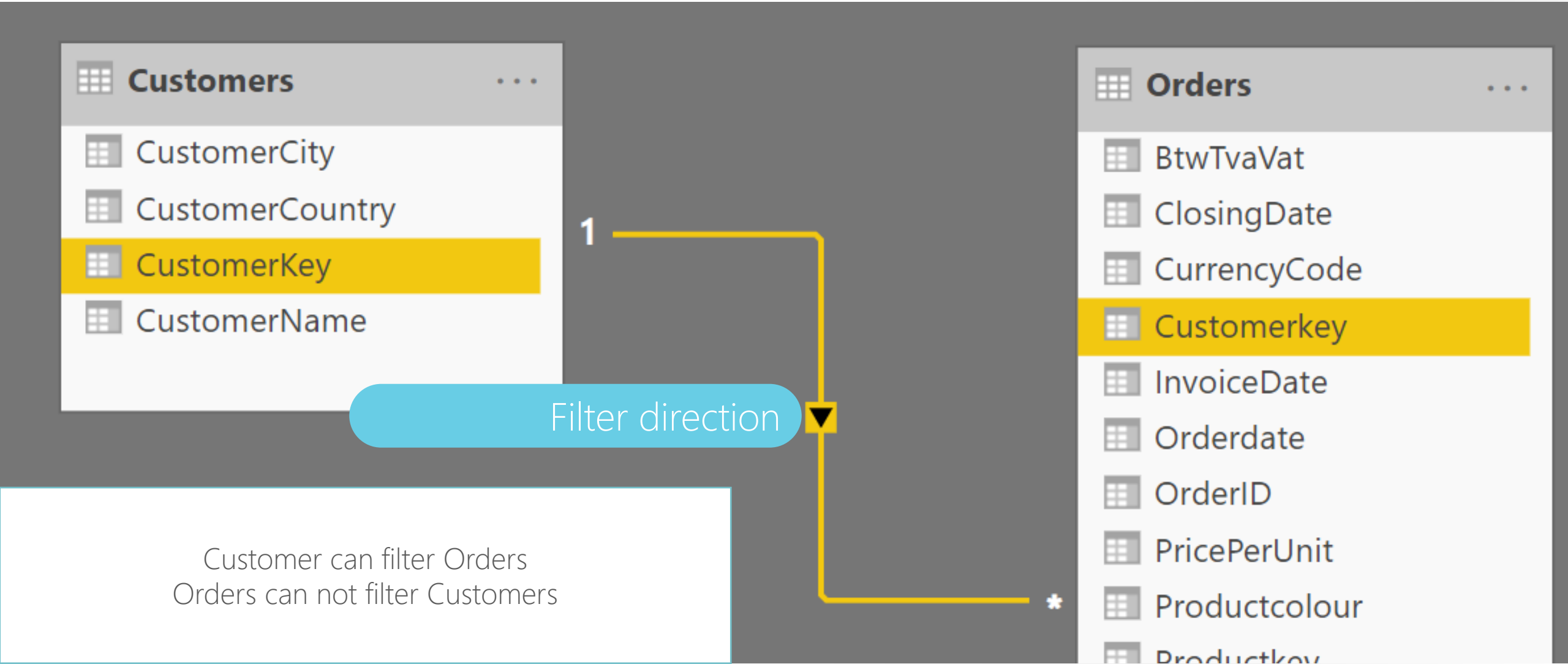
Being active is
normally controlled via
the function: USERELATIONSHIP



Relations

- Relationships have the right direction by default
  - one-many (Dimension – Facts)
  - Based on unique values in columns
- At least one of the columns should contain unique values
- Multiple relations between 2 tables if needed
  - But only one can be active
- Double relations do not exist
  - Create an additional combined key-column

Filter direction



Customer can filter Orders
Orders can not filter Customers

LEARNIA

DAX – Calculated columns

Data Analyses eXpression language

- DAX allows you to analyse and to extend your imported data.

- Excel vs DAX functions
- Excel formulas refer to cells and ranges
- DAX formulas refer to columns or tables

Two types of calculations

- Calculated Columns
  - Calculations made in the data model table
- Measures (Calculated Field)
  - Summaries(aggregations) added to the PivotTable

Calculated Columns

- Calculations made in the data model table
  - Are mainly used as a filter
  - Use space; results are saved in the data model
  - Recalculated on Refresh

- Example

OrderDate	OrderYear	Function OrderYear
23/03/2016	2016	=YEAR(Orders[OrderDate])
15/04/2017	2017	=YEAR(Orders[OrderDate])

Referencing conventions

Referring to a column

- Always include the table name =YEAR(Orders[OrderDate])

OrderDate	OrderYear	Function OrderYear
23/03/2016	2016	=YEAR([OrderDate]) ✗
28/02/2018	2018	=YEAR(Orders[OrderDate]) ✓

Writing tips

- Type your functions, tables, columns and measures
- Use the <↑> <↓> to select
- Use the <TAB> to confirm your selection
- Use <ENTER> to confirm your calculation

- Additional space after a FUNCTIONNAME(
  additional space before)
- Add line breaks using <SHIFT>+<ENTER>
- Use <TAB> to add whitespace
- Type // or -- to add comments in your formula

DAX Code

```
=SUMX(FILTER(Orders; Orders[SalesType] IN {“Phone”,”Call”}) ;
Orders[UnitPrice]*Orders[Quantity])
```

✗

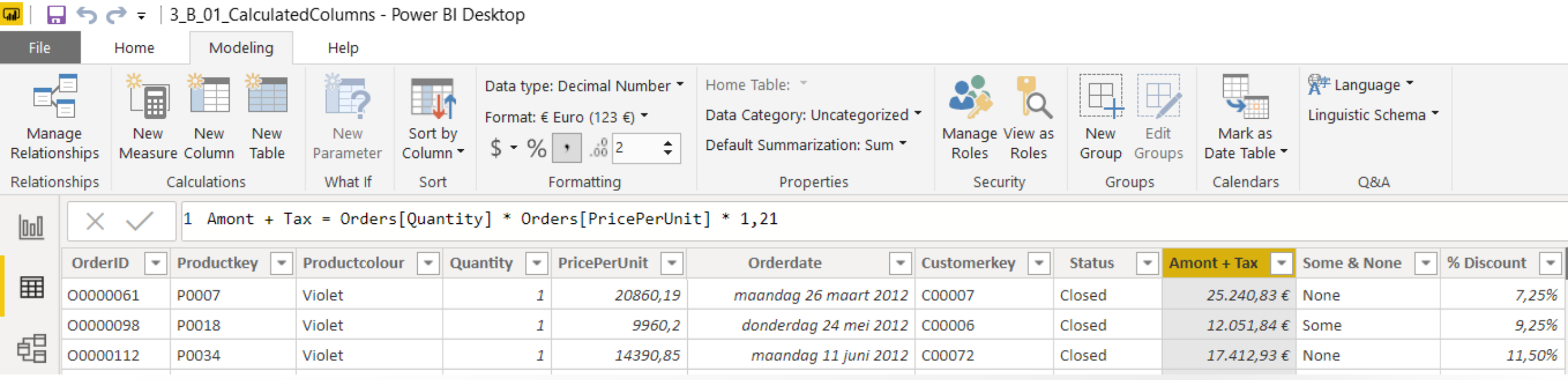
```
=SUMX(
        FILTER(
            Orders;
            Orders[SalesType] IN {“Phone”,”Call”}
            ) ;
        Orders[UnitPrice]*Orders[Quantity]
    )
```

✓

Calculated columns

Calculated Columns in Power BI

- Right-click on a column and select New Column
- In the formula bar type the name in front of the =
  - Spaces and some special characters are allowed
- Enter the formula in the formula bar
- Click ✔ or press <ENTER> to confirm



OrderDate Transformations

Open 3_B_01_CalculatedColumns

- Show the year of the OrderDate:
  OrderYear
  =YEAR(Orders[OrderDate])
- Show the month of the OrderDate:
  OrderMonth
  =MONTH(Orders[OrderDate])

Calculations

In 3_B_01_CalculatedColumns

- Show the quarter of the OrderDate:
  OrderQuarter
  =ROUNDUP(Orders[OrderMonth] / 3 ; 0)

- Show the line total :
  Line Total
  = Orders[Quantity]* Orders[PricePerUnit]

IF … Then … else

1. Create a **CostRange** column
2. Click Add Column > Conditional column
    1. Name CostRange
    2. Requirements : High >15000,
       Normal>8000
       else Low

- =IF (Products[ProductCost]> 15000 ;
  "High" ;
  IF(Products[ProductCost] > 8000 ;
  "Normal" ;
  "Low")
)

SWITCH

- Nested-IFs are hard to control
  Therefore use SWITCH

- MAxDiscount
  =SWITCH (Order[SalesType] ; *// the test*
  "Web" ; 15% ; *//Current value; answer*
  "Walk-in" ; 15% ;
  "Call" ; 10% ;
  6% *//all other situations*
)

The current value should be a single value, "Web", 15 … True

⚠ so not >5200

SWITCH TRUE

- If you need ranges: SWITCH-TRUE

- CostRangeSwitch
  = SWITCH (TRUE() ;
  Products[ProductCost]> 15000 ; "High" ;
  Products[ProductCost]> 8000 ; "Normal" ;
  "Low"
)

Crearte a HighEnd column

- “Expensive cars” : Category =“CAR” && CatalogPrice > 20000
- “Expensive bikes” : Category =“BIKE” && CatalogPrice > 5000
- “Expensive Vans” : Category =“VAN” && CatalogPrice > 25000

```
=SWITCH( TRUE() ;
    Products[ProductCategory]="CAR" && Products[CatalogPrice]>20000 ; TRUE() ;
    Products[ProductCategory]="BIKE" && Products[CatalogPrice]>5000 ; TRUE() ;
    Products[ProductCategory]="VAN" && Products[CatalogPrice]>25000 ; TRUE() ;
    FALSE()
    )
```

Use your relations

- Two tables are related via the one-many relation

  RELATED-function
  - A FACT table retrieving information from the DIM
  - One value is returned

  LOOKUPVALUE-function
  - A relation does not need to exist
  - One value is returned

  RELATEDTABLE-function
  - A DIM table retrieving information from the FACT
  - Multiple lines are returned

RELATED

- In 3_B_01_CalculatedColumns
- Add the ProductName to the orders table:
  ProductName
  =RELATED(Product[ProductName])

- Calculate the margin per line: Line Margin
- Quantity * (product.cost- price)

LOOKUPVALUE : Big chief

- Create a column that will display the name of the CEO: Big Chief
- BIG chief

```
= LOOKUPVALUE( Employees[Name] ,
               Employees[Position] , "CEO" )
```

LOOKUPVALUE : Chief

- Create a column that will display the name of the "reportsto" : Chief
-
```
Chief = LOOKUPVALUE(
        Employees[Name] ,
        Employees[EmployeeID] ,
        Employees[ReportsTo] )
```

LOOKUPVALUE : Department Chief

- Create a column that will display the name of the "manager" : of the corresponding department
-
```
Manager = LOOKUPVALUE(
        Employees[Name],
        Employees[Position] ,  "manager" ,
        Employees[Department] , Employees[Department],
        Employees[EndDate] , BLANK()
 )
```

RELATEDTABLE

In 3_B_01_CalculatedColumns

- In the table Customers is a calculated column
  Orders
  = COUNTROWS(Orders)
  The results is not correct. Why?

- Create a new column that fixes the problem: Orders Rows

```
= COUNTROWS (
                RELATEDTABLE( Orders )
                )
```

Exercise

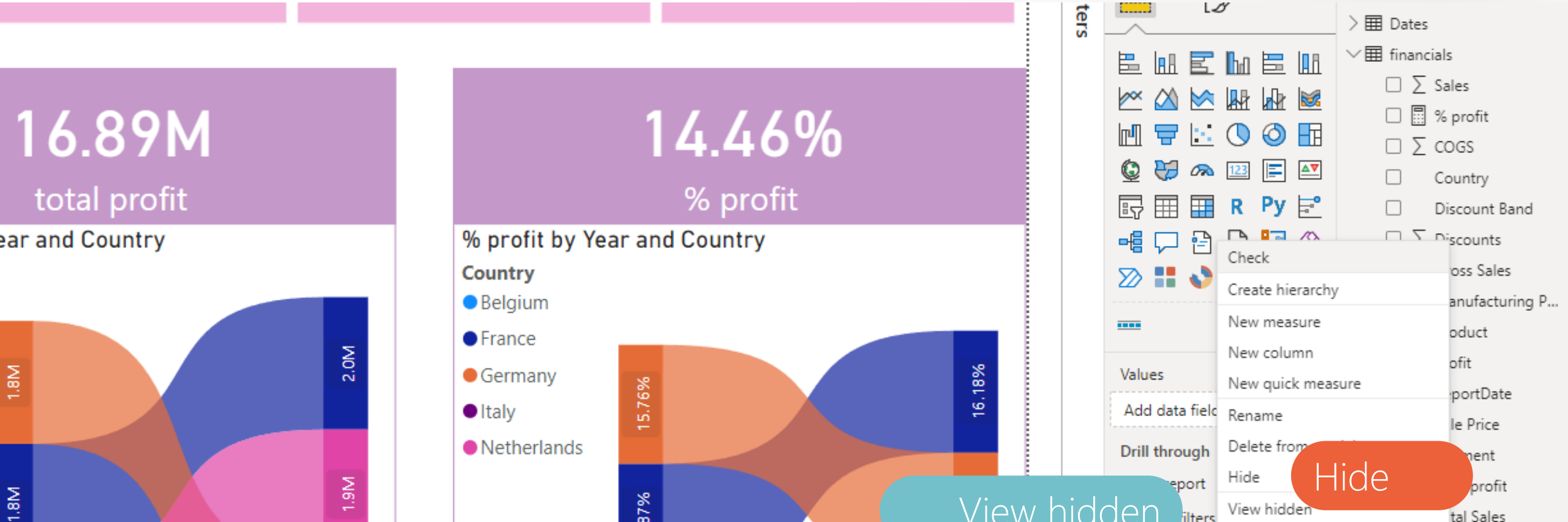
1. Open 3_B_02_Columns
2. For every order calculate the absolute discount comparing the Catalogprice (in Products) and the PricePerUnit (in Orders)
   - Column name: Discount abs
   - Set the Format as: Decimal Number (2 decimals)

3. For every product, count how many order lines exist
   - Column name: Order Lines

Organize your model

Organize your Model

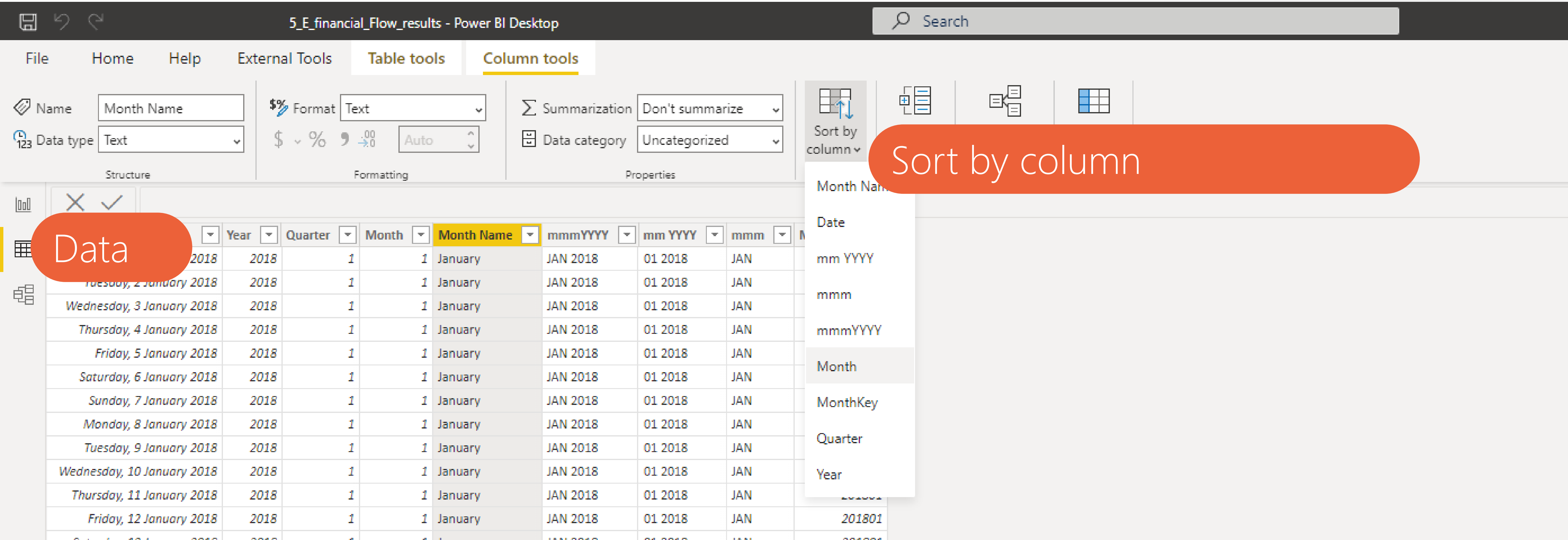
- Two related tables often share a column
  - Customers.CustomerID vs Orders.CustomerID
- These columns are (only)necessary for the relation



LEARNIA

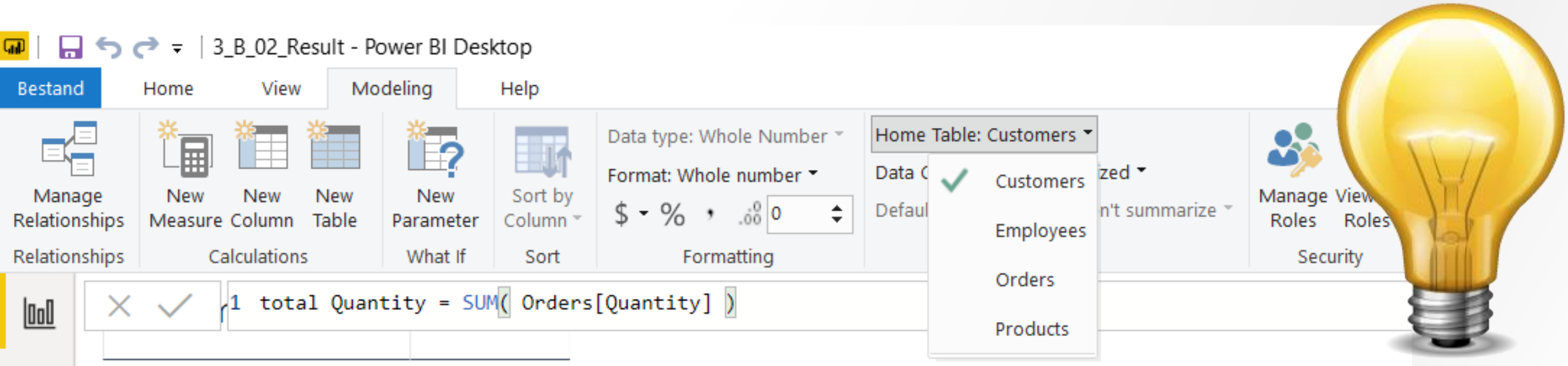
Sort columns

- Column tools > Sort by column



Where to create a measure

- Measures can be stored in any table of your data model
- In measures can easily be moved from on table to another
- Therefore, never include the table name (location) when referring to a measure



Measures table

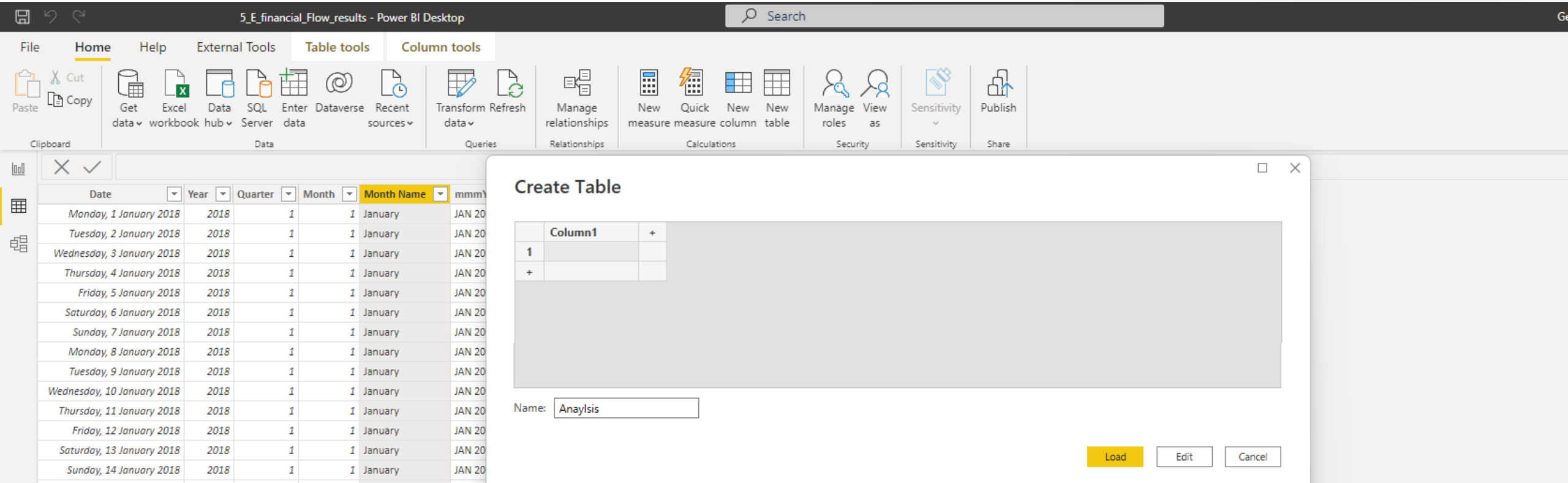
Save all your measures in a separate table

Advantages

- All measures are in a single location
- When tables need to be reloaded (not refreshed) you don’t loose your measures

Create a measure "Table"

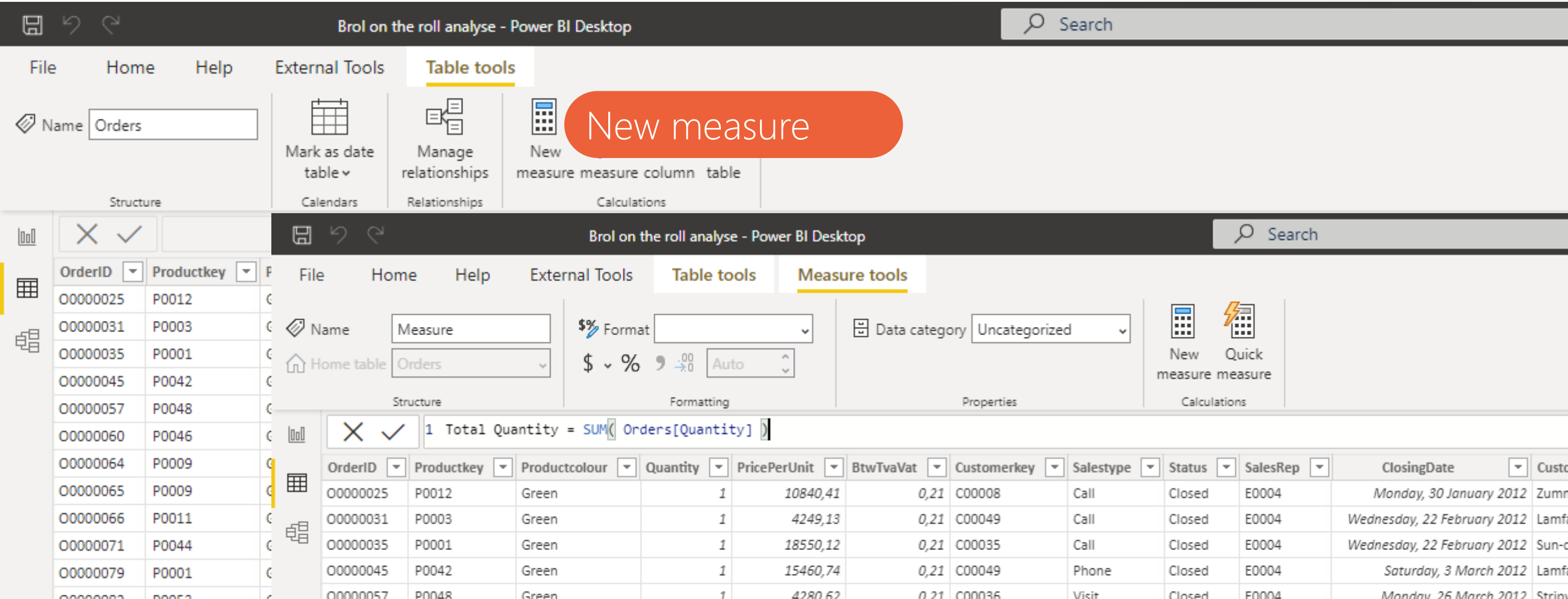
1. Home > Data > Enter data
2. Type a name for the measure table: "Analysis"
3. After the first measure is created, remove the empty Column1



Part 3 : Analysis

Basic aggregations

Create a measure



Basic aggregations

- Functions
  - SUM, MAX, MIN …
  - COUNT, COUNTA, DISTINCTCOUNT …
  - COUNTROWS
  - AVERAGE, STDEV.P, STDEV.S …

- These functions are similar to the functions in Excel
- Only one argument: TableName or ColumnName
  - COUNTROWS(TableName)
  - SUM(TableName[ColumnName])

COUNT/SUM functions

- Open 4_A_01_Demo
- The number of Orders: Order Lines
  =COUNTROWS(Orders)

- The number of Customers that Ordered: Actual Customers
  =DISTINCTCOUNT(Orders[Customerkey])

- Sum of Quantity ordered: Quantity Total
  =SUM(Orders[Quantity])

Create Visual

- Create a simple visual showing these measures
- Show the values vertically
- Break the values per Customercountry
- Insert a slicer for the ProductCategory

AVERAGE and STDEV

- In 4_A_01_Demo
- Calculate the average Quantity ordered: Quantity Avg
  =AVERAGE(Orders[Quantity])

- Calculate the Standard Deviation of the ordered Quantity: Quantity StDev
  =STDEV.P(Orders[Quantity])

- Use the Measures in a visual together with CustomerCountry

Getting started with variables

VAR-RETURN

Use variables to simplify complex measures

```
=IF( ( [COST] - [PRICE] )> 2400 ;
        ( [COST] - [PRICE] ) * 0,06 ;
        ( [COST] - [PRICE] ) * 0,078
        )
```

```
=VAR Delta = ([COST] - [PRICE])

RETURN
IF(Delta >2400 ;
        Delta * 0,06 ;
        Delta * 0,078
      )
```

- The VARiable is calculated only once
- The VARiable only exists in this measure

Aggregating Expressions

Aggregating Expressions

- Functions
  - SUMX, MAXX, MINX, AVERAGEX, STDEVX.P, STDEVX.S …

Arguments

- Expressions inside the function
  - SUMX(Orders;
    [Price]*[Quantity])

- Temporary Tables
  - AVERAGEX(
    TOPN(10;Customers;[revenue]);
    [revenue])

SUMX

- Open 4_B_01_Demo
- SUM of Expected Orders: Total Orders

```
=SUMX( Orders;
                    Orders[Quantity] * Orders[PricePerUnit] )
```

- Sales with Tax: Total TVA

```
=SUMX( Orders;
                    Orders[Quantity] *
                    Orders[PricePerUnit]*Orders[BtwTvaVat] )
```

- Total Orders incl TVA

```
= [Total Orders] + [Total orders TVA]
```

SUMX - Challenge

- Calculate the total margin: Total Margin
  Tip: SUM(Quantity*(Price-Cost))

```
=SUMX( Orders;
        [Quantity]
        *   (
            [PricePerUnit] - RELATED( Products[ProductCost] )
            )
        )
```

- Create a visual showing these values per country

Filters

FILTER()

- FILTER() is a table function; it returns a filtered table
- FILTER(Table ; Filter)

FILTER(Orders; Orders[Status]="Cancelled")

Generates a table containing all Cancelled Orders

⚠ So this can never be an answer of a measure

Cancelled Deals

- Sum of the Cancelled Deals : Total Cancelled Deals

```
=SUMX(
        FILTER(  Orders ;
                  Orders[Status]="Cancelled"
                ) ;
        Orders[Quantity] * Orders[PricePerUnit]
        )
```

Closed Deals

- Sum of the Closed Orders: Total Sales

```
=   SUMX(
            FILTER(  Orders ;
                        Orders[Status]="closed"
                    ) ;
            Orders[Quantity] * Orders[PricePerUnit]
            )
```

- Extra: Create a measure calculating the Success Rate

  = [Total Sales] / [Total Orders]

Average to delivery

- Average Days to delivery =

```
AVERAGEX( Orders;
          Orders[Senddate]-Orders[Orderdate] )
```

Strange Results

- Average Days to delivery =

```
AVERAGEX(
          FILTER(  Orders ;
                   Orders[Senddate]<> BLANK()
                   ) ;
          Orders[Senddate]-Orders[Orderdate] )
```

Filter Functions: ALL

- ALL cancels all filters in measure
- This allows us to create GrandTotals and those are important tot calculate Percentages

Arguments

- ALL(TableName)
- ALL(TableName[ColumnName])

⚠ ALL creates a table with unique rows

Row Labels	Total Deals Net	Total All Deals Net
AT	355.041,36	41.096.949,03
BE	2.572.384,63	41.096.949,03
DE	6.047.208,19	41.096.949,03
DK	1.011.113,56	41.096.949,03
ES	1.612.944,76	41.096.949,03
FI	577.618,21	41.096.949,03
FR	4.819.322,63	41.096.949,03
GB	5.249.925,92	41.096.949,03
HU	2.090.269,29	41.096.949,03
IT	3.963.779,76	41.096.949,03
NL	833.227,03	41.096.949,03
	,98	41.096.949,03
	,96	41.096.949,03
SE	5.091.360,15	41.096.949,03
US	806.901,60	41.096.949,03
Grand Total	**41.096.949,03**	**41.096.949,03**

All Orders

- Open 4_B_02_Demo
- Total Deals Net:=
  SUMX(Orders ;
  Orders[Quantity] * Orders[PricePerUnit])

- Total All Deals Net:=
  SUMX(ALL(Orders) ;
  Orders[Quantity] * Orders[PricePerUnit])

- Total All Deals Net NOT:=
  SUMX(ALL(Orders[PricePerUnit]; Orders[Quantity]) ;
  Orders[Quantity] * Orders[PricePerUnit])

Bereken het percentage tov Grand Total

- the Total All Deals Net-measure is not always very relevant as such
- Divide Total Deals Net by Total All Deals Net
- Deals Net % :=

  Total Deals Net / Total All Deals Net

- Nothing to do …
  - Rewrite Deals Net % withour using the Total All Deals Net
    Tip: use variables

ALLSELECTED

- ALLSELECTED Works similar to ALL
- But keeps the filters from outside the visual

Row Labels	Total Deals Net	Total All Deals Net	Total AllSELECTED Deals Net
AT	71.422	41.096.949	4.767.847
BE	367.110	41.096.949	4.767.847
DE	553.136	41.096.949	4.767.847
DK	237.944	41.096.949	4.767.847
ES	114.073	41.096.949	4.767.847
FI	129.975	41.096.949	4.767.847
FR	238.993	41.096.949	4.767.847
GB	877.558	41.096.949	4.767.847
HU	360.177	41.096.949	4.767.847
IT	540.587	41.096.949	4.767.847
NL	37.340	41.096.949	4.767.847
NO	571.904	41.096.949	4.767.847
PL	83.853	41.096.949	4.767.847

Row Lab	Total Deals Net
Bike	1.305.134
Car	31.341.483
Motor	4.767.847

Motor 4.767.847

ProductCategory

- Bike
- Car
- Motor

CALCULATE

CALCULATE

- CALCULATE alters the applied filters.

- Structure: CALCULATE(expression ; Filter1 ; Filter2 …)
  - Filter: column = static value
    - =, <, >, <=, >= and <>
  - Filter function: ALL, ALLEXCEPT, SAMEPERIODLASTYEAR…

Example

- Count Orders=
  CALCULATE(COUNTROWS(orders) ;
  Orders[Status]="Closed")

Sales by Type

- Total Sales via Internet: Total Sales Internet

```
= CALCULATE( [Total Sales]
                ; Orders[SalesType] = "web" )
```

- Total Sales via Internet: Total Sales Walk-in

```
= CALCULATE( [Total Sales]
                ; Orders[SalesType] = "Visit" )
```

Arguments

- = Equals
- && logical AND
- || logical OR

- Total Sales BeNeLux=

```
CALCULATE( [Total Sales];
                Customer[Country]="BE"    ||
                Customer[Country]="LUX"   ||
                Customer[Country]="NL" )
```

Sales by Type

- Total Sales via Internet: Total Sales Call

```
Total Sales Calls=
                CALCULATE(   [Total Sales]
                                        ; Orders[SalesType] = "Call"
                                        || Orders[SalesType] = "Phone")
```

Results per SalesType

- Share Walk-in to Total Sales: Share Walk-in

```
=   DIVIDE(     [Total Sales Walk-in ] ;
                [Total Sales]
            )
```

Productcolour	Internet Sales	Walk-in Sales	Total Sales	Share Walk-in
Violet	€ 5,881,606	€ 8,201,548	€ 27,893,907	29.4%
Blue	€ 4,917,333	€ 6,298,664	€ 23,532,352	26.8%
Green	€ 4,096,590	€ 10,058,501	€ 32,307,829	31.1%
Indigo	€ 2,515,962	€ 4,113,186	€ 13,329,882	30.9%
Orange	€ 85,080	€ 609,970	€ 1,780,880	34.3%
Total	**€ 17,496,571**	**€ 29,281,869**	**€ 98,844,850**	**29.6%**

Filter Functions: ALL

- ALL cancels all filters applied to the measure
- This allows the creation of GrandTotals and these are important to calculate Shares

Arguments

- ALL(TableName)
- ALL(TableName[ColumnName])

Shares per ProductType/Category

- Calculate the Total Sales of all Products

Total Sales All Products

```
=   CALCULATE( [Total Sales];
                    ALL( Products )
                    )
```

- Calculate the Share of every product

Share Per Product

```
=DIVIDE( [Total Sales];
                [Total Sales All Products];
                BLANK()
                )
```

ProductCategory	Total Sales	Total Sales All Products	Share Per Product
Bike	**2,535,584.09**	**89,642,052.04**	**2.83%**
ATB	621,257.37	89,642,052.04	0.69%
City	1,489,745.98	89,642,052.04	1.66%
Competitive	424,580.74	89,642,052.04	0.47%
Car	**70,309,475.19**	**89,642,052.04**	**78.43%**
Compact	28,392,077.77	89,642,052.04	31.67%
Crossover	9,898,478.30	89,642,052.04	11.04%
Executive	11,908,095.20	89,642,052.04	13.28%
Family	19,349,342.76	89,642,052.04	21.59%
Sports	761,481.16	89,642,052.04	0.85%
Motor	**8,771,069.96**	**89,642,052.04**	**9.78%**
Executive	1,665,405.98	89,642,052.04	1.86%
Sports	7,105,663.98	89,642,052.04	7.93%
Van	**8,025,922.80**	**89,642,052.04**	**8.95%**
Van	8,025,922.80	89,642,052.04	8.95%
Total	**89,642,052.04**	**89,642,052.04**	**100.00%**

Country Comparison

- Create Total Sales all Countries

```
= CALCULATE(
    [Total Sales] ;
    All( Customers[CustomerCountry] )
    )
```

CustomerCountry	Total Sales	Total Sales all Countries	Share Per Country
AT	622,643.19	89,642,052.04	0.69%
BE	4,898,037.62	89,642,052.04	5.46%
DE	12,851,616.56	89,642,052.04	14.34%
DK	1,974,720.17	89,642,052.04	2.20%
ES	3,231,665.46	89,642,052.04	3.61%
FI	1,290,745.27	89,642,052.04	1.44%
FR	10,081,442.19	89,642,052.04	11.25%
GB	14,228,365.56	89,642,052.04	15.87%
HU	5,032,954.25	89,642,052.04	5.61%
IT	7,818,521.82	89,642,052.04	8.72%
NL	2,059,563.67	89,642,052.04	2.30%
NO	11,694,999.84	89,642,052.04	13.05%
PL	2,299,844.57	89,642,052.04	2.57%
SE	9,805,560.95	89,642,052.04	10.94%
US	1,751,370.92	89,642,052.04	1.95%
Total	**89,642,052.04**	**89,642,052.04**	**100.00%**

- Create Share Per Country

```
= DIVIDE( [Total Sales] ; [Total Sales all Countries] )
```

ALLEXCEPT()

- ALLEXCEPT will cancel all filters except the one specified
- Create Total Sales per Year
  - = CALCULATE(
    [Total Sales] ;
    ALLEXCEPT(Dates; Dates[Year])
)
- Create Share per Year
  - = DIVIDE ([Total Sales] ;
    [Total Sales per Year]
)

Extra : ISFILTERED

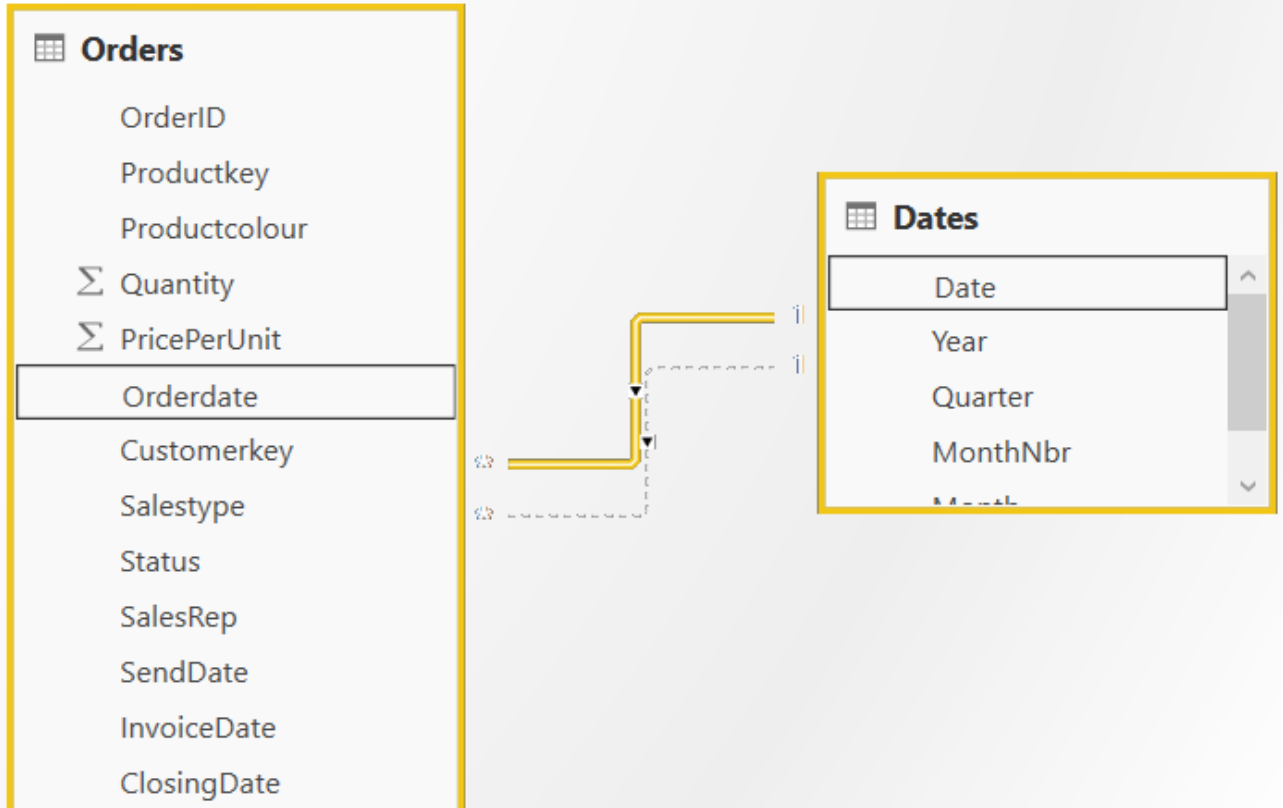
- Use ISFILTERED to test if a filter is being used.
- You can remove Share over Categories if there is no filter on either ProductType or ProductSize …

```
Share over Categories 2:=
IF(
        ISFILTERED( Products[ProductSize] ) ||
            ISFILTERED( Products[ProductType] ) ;
    DIVIDE (  [Total Sales] ; [Total Sales per Category]  ) ;
    BLANK()
    )
```

	A	B	C
1	Row Labels	Total Sales	Share over Categories 2
2	**Bike**	**2.535.584,09**	
3	ATB	621.257,37	24,5%
4	City	1.489.745,98	58,8%
5	Competitive	424.580,74	16,7%
6	**Car**	**70.309.475,19**	
7	Compact	28.392.077,77	40,4%
8	Crossover	9.898.478,30	14,1%
9	Executive	11.908.095,20	16,9%
10	Family	19.349.342,76	27,5%
11	Sports	761.481,16	1,1%
12	**Motor**	**8.771.069,96**	
13	Executive	1.665.405,98	19,0%
14	Sports	7.105.663,98	81,0%
15	**Van**	**8.025.922,80**	
16	Van	8.025.922,80	100,0%
17	**Grand Total**	**89.642.052,04**	

USERELATIONSHIP()

- When multiple relations exist between 2 tables, only one is active.
- USERELATIONSHIP actives an inactive existing relationship



Total Invoiced

- Create a measure that calculates how much was invoiced in a given period. Total Invoiced

- Total Invoiced

```
= CALCULATE( [Total Sales];
             USERELATIONSHIP(
             Dates[date] ; Orders[InvoiceDate]  )
             )
```

- Compare the results in a table
  Total Sales and Total Invoiced
  grouped by years and months

Limits of CALCULATE

- None, except…
  - There are safeguards to ensure high performance
  - Filters are not allowed to be too complex, like
    - Column = Measure
    - Column = Formula
    - Column = Column
    - Measure = Measure
      - =, <, >, <=, >= and <>

- So this doesn't work
  - ```
    Sales late delivered =
      CALCULATE( [Total Sales]
                        ; Orders[SendDate]-Orders[Orderdate] > 5
                        )
    ```

For more complex situations use FILTER()
    ```

FILTER()

- FILTER(Table ; Filter)

- Sales late delivered :=

```
CALCULATE( [Total Sales]
                ; FILTER( Orders
                        ; Orders[SendDate]-Orders[Orderdate] > 5 )
                )
```

Summarize

Summarize

- Via Summarize you can compile tables that can then be analyzed.
- Especially for calculating averages over certain periods
-
- For example, calculate a Monthly Average Sales

```
Monthly Average Sales:=
        AVERAGEX(
        SUMMARIZE( Orders ; Dates[MonthKey] ; "Sales" ; [Total Sales] );
        [Sales] )
```

Row Labels	Total Sales	Monthly Average Sales
2012	**6.301.455,50**	**525.121,29**
January	653.073,82	653.073,82
February	645.474,11	645.474,11
March	407.455,66	407.455,66
April	616.329,98	616.329,98
May	509.035,93	509.035,93
June	575.806,35	575.806,35
July	497.095,45	497.095,45
August	551.021,63	551.021,63
September	421.585,60	421.585,60
October	422.104,58	422.104,58
November	625.384,00	625.384,00
December	377.088,39	377.088,39
2013	**12.123.624,53**	**1.010.302,04**
January	1.027.436,00	1.027.436,00
February	758.060,24	758.060,24
March	1.027.184,71	1.027.184,71
April	923.258,86	923.258,86
May	986.308,31	986.308,31
June	1.171.801,30	1.171.801,30
July	1.137.907,97	1.137.907,97
August	809.371,00	809.371,00
September	900.568,20	900.568,20
October	1.268.086,90	1.268.086,90
November	1.088.328,17	1.088.328,17
December	1.025.312,87	1.025.312,87
2014	**12.928.843,76**	**1.077.403,65**
January	1.115.147,60	1.115.147,60
February	1.316.994,12	1.316.994,12
March	1.127.663,42	1.127.663,42
April	1.145.678,66	1.145.678,66
May	973.388,73	973.388,73

Monthly Average Sales

- The average is the total of the 12 months divided by 12

1.010.302,04

12.123.624,53 / 12

LEARNIA

CALCULATE

CALCULATE

- CALCULATE alters the applied filters.

- Structure: CALCULATE(expression ; Filter1 ; Filter2 …)
  - Filter: column = **static value**
    - =, <, >, <=, >= and <>
  - Filter function: ALL, ALLEXCEPT, SAMEPERIODLASTYEAR…

Example

- Count Orders=
  CALCULATE(COUNTROWS(orders) ;
  Orders[Status]="Closed")

Sales by Type

- Total Sales via Internet: Total Sales Internet

```
= CALCULATE( [Total Sales]
                ; Orders[SalesType] = "web" )
```

- Total Sales via Internet: Total Sales Walk-in

```
= CALCULATE( [Total Sales]
                ; Orders[SalesType] = "Visit" )
```

Filter Functions: ALL

- ALL cancels all filters applied to the measure
- This allows the creation of GrandTotals and these are important to calculate Shares

Arguments

- ALL(TableName)
- ALL(TableName[ColumnName])

Shares per ProductType/Category

- Calculate the Total Sales of all Products

Total Sales All Products

```
=   CALCULATE( [Total Sales];
                    ALL( Products )
                    )
```

- Calculate the Share of every product

Share Per Product

```
=DIVIDE( [Total Sales];
                [Total Sales All Products];
                BLANK()
                )
```

ProductCategory	Total Sales	Total Sales All Products	Share Per Product
Bike	**2,535,584.09**	**89,642,052.04**	**2.83%**
ATB	621,257.37	89,642,052.04	0.69%
City	1,489,745.98	89,642,052.04	1.66%
Competitive	424,580.74	89,642,052.04	0.47%
Car	**70,309,475.19**	**89,642,052.04**	**78.43%**
Compact	28,392,077.77	89,642,052.04	31.67%
Crossover	9,898,478.30	89,642,052.04	11.04%
Executive	11,908,095.20	89,642,052.04	13.28%
Family	19,349,342.76	89,642,052.04	21.59%
Sports	761,481.16	89,642,052.04	0.85%
Motor	**8,771,069.96**	**89,642,052.04**	**9.78%**
Executive	1,665,405.98	89,642,052.04	1.86%
Sports	7,105,663.98	89,642,052.04	7.93%
Van	**8,025,922.80**	**89,642,052.04**	**8.95%**
Van	8,025,922.80	89,642,052.04	8.95%
Total	**89,642,052.04**	**89,642,052.04**	**100.00%**

Country Comparison

- Create Total Sales all Countries

```
= CALCULATE(
    [Total Sales] ;
    All( Customers[CustomerCountry] )
    )
```

CustomerCountry	Total Sales	Total Sales all Countries	Share Per Country
AT	622,643.19	89,642,052.04	0.69%
BE	4,898,037.62	89,642,052.04	5.46%
DE	12,851,616.56	89,642,052.04	14.34%
DK	1,974,720.17	89,642,052.04	2.20%
ES	3,231,665.46	89,642,052.04	3.61%
FI	1,290,745.27	89,642,052.04	1.44%
FR	10,081,442.19	89,642,052.04	11.25%
GB	14,228,365.56	89,642,052.04	15.87%
HU	5,032,954.25	89,642,052.04	5.61%
IT	7,818,521.82	89,642,052.04	8.72%
NL	2,059,563.67	89,642,052.04	2.30%
NO	11,694,999.84	89,642,052.04	13.05%
PL	2,299,844.57	89,642,052.04	2.57%
SE	9,805,560.95	89,642,052.04	10.94%
US	1,751,370.92	89,642,052.04	1.95%
Total	**89,642,052.04**	**89,642,052.04**	**100.00%**

- Create Share Per Country

```
= DIVIDE( [Total Sales] ; [Total Sales all Countries] )
```

ALLEXCEPT()

- ALLEXCEPT will cancel all filters except the one specified
- Create Total Sales per Year

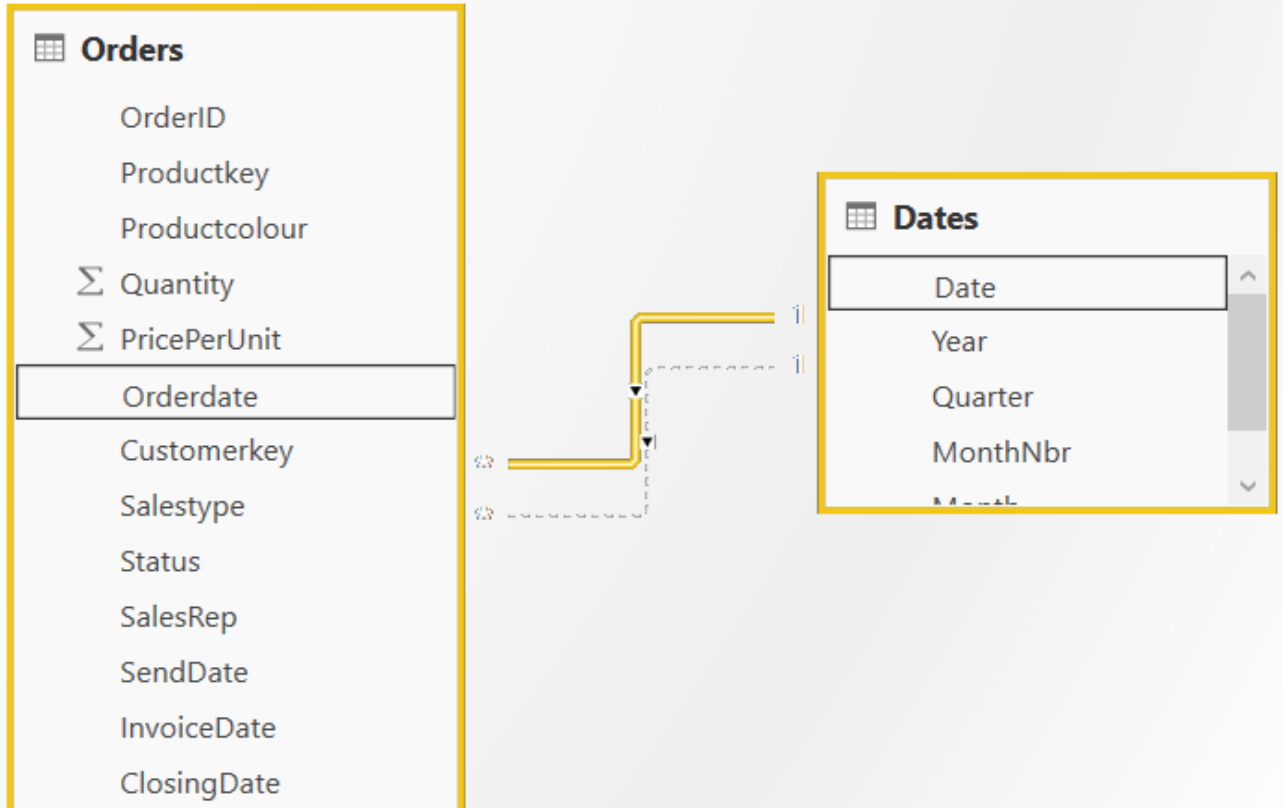
```
= CALCULATE(
        [Total Sales] ;
        ALLEXCEPT( Dates; Dates[Year] )
        )
```

- Create Share per Year

```
= DIVIDE (  [Total Sales] ;
          [Total Sales per Year]
        )
```

USERELATIONSHIP()

- When multiple relations exist between 2 tables, only one is active.
- USERELATIONSHIP actives an inactive existing relationship



Total Invoiced

- Create a measure that calculates how much was invoiced in a given period. Total Invoiced

- Total Invoiced

```
=
    CALCULATE( [Total Sales];
                USERELATIONSHIP(
                Dates[date] ; Orders[InvoiceDate]  )
                )
```

- Compare the results in a table Total Sales and Total Invoiced grouped by years and months

Limits of CALCULATE

- None, except…
  - There are safeguards to ensure high performance
  - Filters are not allowed to be too complex, like
    - Column = Measure
    - Column = Formula
    - Column = Column
    - Measure = Measure
      - =, <, >, <=, >= and <>

- This does NOT work
  - ```
    Sales late delivered =
      CALCULATE( [Total Sales]
                 ; Orders[SendDate]-Orders[Orderdate] > 5
                 )
    ```

For more complex situations use FILTER()
    ```

FILTER()

- FILTER() nulls the build-in safeguards
- FILTER() is a table function, it returns a filtered table
- FILTER(Table ; Filter)

- Sales late delivered

```
= CALCULATE( [Total Sales]
                    ; FILTER( Orders
                            ; Orders[SendDate]-Orders[Orderdate] > 5 )
                    )
```

CALCULATE with/without FILTER

- Actual Orders =
  CALCULATE([Deals Expected]
  ; Orders[Status] = "Closed")

- Actual Filtered Orders =
  CALCULATE(
  [Deals Expected]
  ;FILTER (Orders
  ; Orders[Status] = "Closed")
)

Adds/replaces a filter

Filters the table

Status	Deals Expected	Actual Orders	Actual Filtered Orders
Cancelled	357	2810	
Closed	2810	2810	2810
Total	**3167**	**2810**	**2810**

Time intelligence

Date Intelligence

- Date intelligence function
  - TOTALYTD, TOTALQTD, TOTALMTD (running totals)
  - SAMEPERIODLASTYEAR
  - DATEADD
  - …
- These functions need a Dimension to work; a Date Table
- The date table should be linked to your dates in the facts table

The Date Table

Date Table

- One keycolumn containing dates
- All dates in the reference period should exist
  - e?.g. all days between 1/1/2010 and 31/12/2020
  - Power Pivot: Design/ Mark as Date table
  - Power BI: Modeling / Mark as Date table
- Additional information should be added like
  - Year, weeknumber, IsWeekday, MonthName…

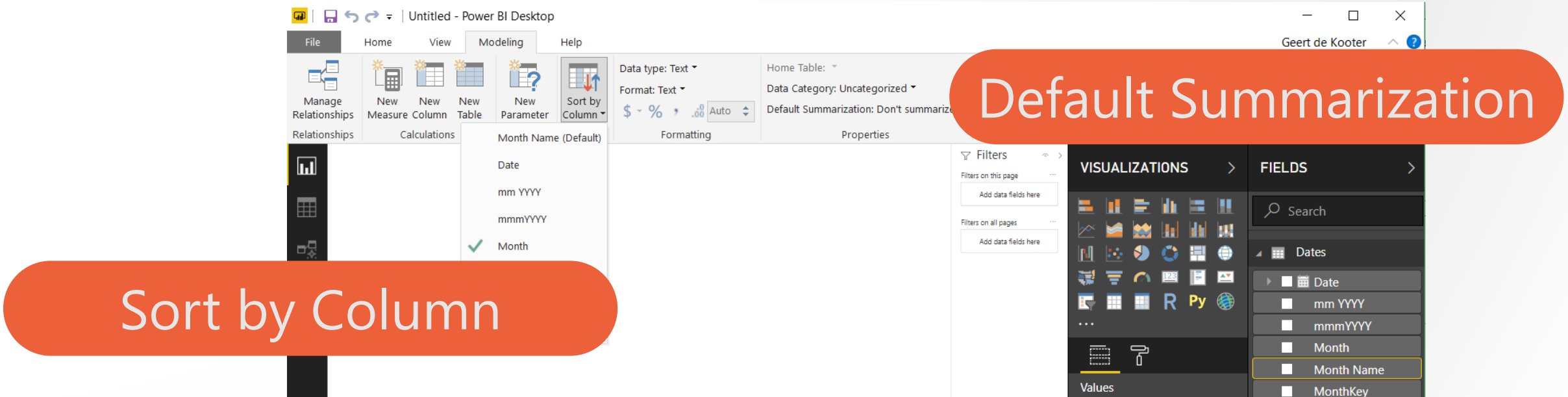
	DateActual	DateKey	YearKey	HalfYearKey	QuarterKey	MonthKey	MonthName	MaandName	MoisName	DayOfWeek	DayName
7	6/01/2010	20100106	2010	201001	201001	201001	January	Januari	janvier	3	Wednesday
8	7/01/2010	20100107	2010	201001	201001	201001	January	Januari	janvier	4	Thursday
9	8/01/2010	20100108	2010	201001	201001	201001	January	Januari	janvier	5	Friday
10	9/01/2010	20100109	2010	201001	201001	201001	January	Januari	janvier	6	Saturday
11	10/01/2010	20100110	2010	201001	201001	201001	January	Januari	janvier	7	Sunday
12	11/01/2010	20100111	2010	201001	201001	201001	January	Januari	janvier	1	Monday

Where to get it from

- A date table can be
  - Imported like any other source
  - Calculated in Power Query via M
    - See hand-out
  - Calculated in DAX (only Power BI)

Steps

1. Import the Date-table
2. Modeling / Mark as Date Table
    1. Select the key column
3. Avoid summarization on columns
4. Sort the Month Name on using the Month (number)



Import a DateTable

- Open 4_D_01_DateIntel
- Create a Date table using the M code in 4_D_01_M_DataTable.txt
- Change the name of the table to: Dates
- Mark the table as Date Table
- After loading the data
  - Change the sort order of the columns
    - Month, Month Year and Month (Abbr)
  - Switch off the default summarization for:
    - Year and Month
  - Hide the Monthkey column
- Create relations
- Use Year, Quarter and Month Name in the visual

ISO Weeknumbers

- Copy the contents from M_function_ISOWeeknum.txt
- Create a blank query to create a custom function
  - Name: fx_ISOweeknumber
- Paste the contents in the advanced editor

- Go to the Dates-query
- Add Column / Invoke custom function
  - Name: Weeknumber
  - Select the Dates[Date] column

Year To Date

Naming conventions

Acronym	Description
YTD	Year-to-date
QTD	Quarter-to-date
MTD	Month-to-date
WTD	Week-to-date
R12M	Rolling 12 Months
PY	Previous year
PQ	Previous quarter
PM	Previous month
YOY	Year-over-year
PYC	Previous year complete
PQC	Previous quarter complete
PMC	Previous month complete

TOTALYTD()

- Creates a running total (per calendar year)
- Arguments
  - TOTALYTD(Expression ; Dates[Date])

Dates[Date] = keycolumn

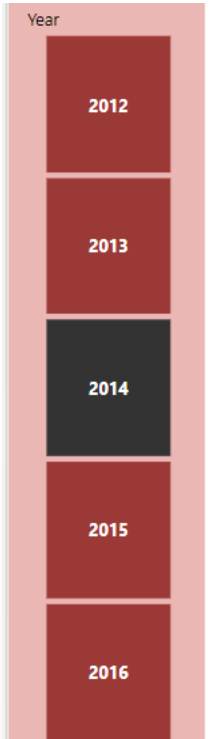
- Example
- OrderCountYTD=
  TOTALYTD(COUNTROWS(Orders);
  Dates[Date])

Total Sales YTD

- Open 4_D_02_DateCalculations
- Create a running total on the existing measure Total Sales : Total Sales YTD

```
=TOTALYTD( [Total Sales] ; Dates[Date] )
```

- Add Total Sales YTD to the visual/PivotTable



Month Name	Total Sales	Total Sales YtD
January	1,115,147.60	1,115,147.60
February	1,316,994.12	2,432,141.72
March	1,127,663.42	3,559,805.14
April	1,145,678.66	4,705,483.80
May	973,388.73	5,678,872.53
June	1,106,549.08	6,785,421.61
July	1,066,347.12	7,851,768.73
August	874,250.72	8,726,019.45
September	865,974.64	9,591,994.09
October	1,164,448.13	10,756,442.22
November	1,041,629.02	11,798,071.24
December	1,130,772.52	12,928,843.76
Total	**12,928,843.76**	**12,928,843.76**

DATESYTD()

- Creates a table containing all dates up to the last date in the context of the selected year
- Arguments
  - DATESYTD(Dates[Date])

Dates[Date] = keycolumn

- So Total Sales YTD could also be written as:

```
= CALCULATE (
                [Total Sales] ;
                DATESYTD( Dates[Date] )
                )
```

DATESBETWEEN()

- Creates a table containing all dates between the two given dates
- Arguments
  - DATESBETWEEN(Dates[Date] ; Startdate ; Enddate)

- So Total Sales YTD could therefore be written as:

```
= CALCULATE (
            [Total Sales] ;
            DATESBETWEEN( Dates[Date];
                        DATE ( YEAR ( MAX ( Dates[Date] ) ) ; 1; 1 ) ;
                        MAX( Dates[Date] )
                        )
                  )
```

DATESBETWEEN() - Total To Date

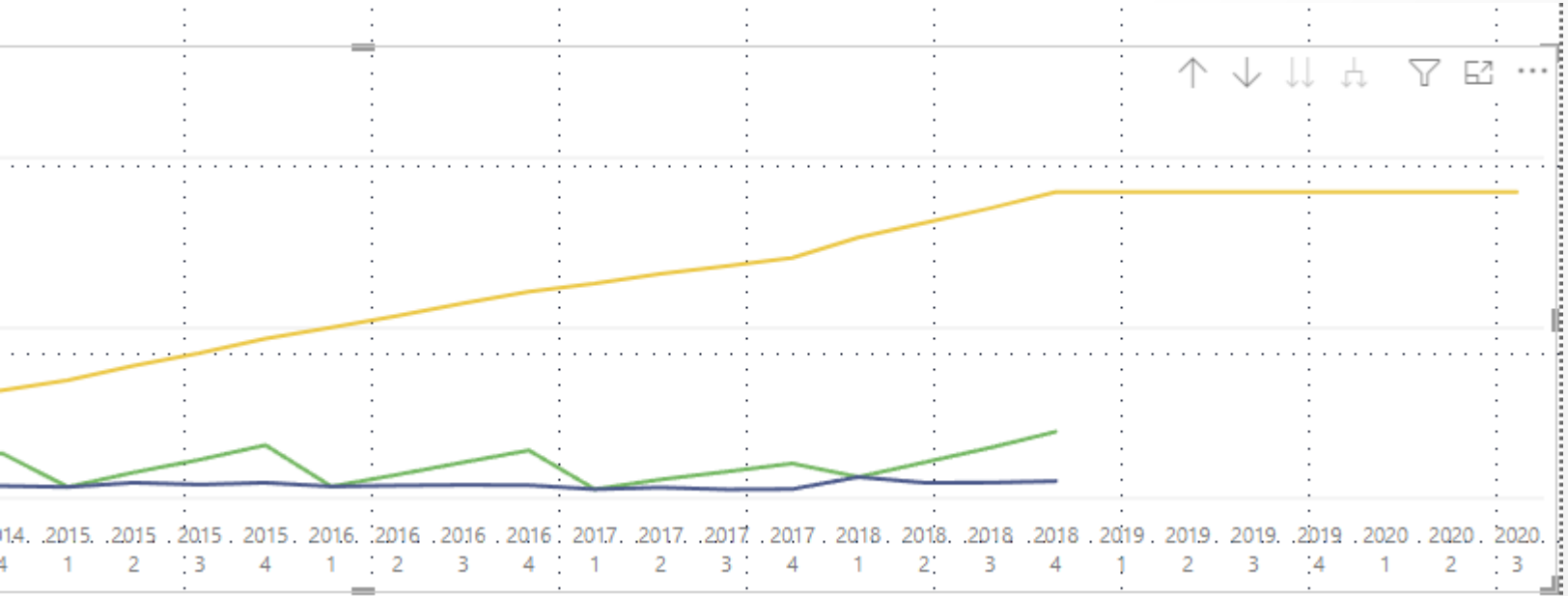
- DATESBETWEEN doesn't seem to be very useful …
- But what if you don't want to restart at the beginning of the year

- Total Sales to Date could therefore be written as:

```
= CALCULATE (
                    [Total Sales] ;
                    DATESBETWEEN( Dates[Date];
                                        DATE( 1900 ; 1; 1) ;
                                        MAX( Dates[Date]      )
                                        )
                    )
```

Total to Date - Presentation problem

- When using a TD there are answers even if there are no new sales
- Check for every date if it is before the last "orderdate"
- Calculated column:
  DateWithSales = Dates[Date] <= MAX (Orders[Orderdate])
- Use the value to filter the visual



DatesInPeriod - Rolling 12 Months

- DATESINPERIOD returns all dates in a given period

- Total Sales R12M :=
  CALCULATE(
  [Total Sales] ;
  DATESINPERIOD(
  Dates[Date];
  MAX(Dates[Date]);
  -12;
  MONTH)
)

Total Quantity to Date

Total Quantity

```
= SUMX(FILTER( Orders; Orders[Status]= "closed"); Orders[Quantity] )
```

Total Quantity To Date

```
= CALCULATE(
    [Total Quantity ];
      DATESBETWEEN( Dates[Date];
                    DATE( 1900 ; 1; 1) ;
                    MAX( Dates[Date]      )
                    )
    )
```

Total Delivered to Date

Quantity Delivered

```
= SUM( Deliveries[Quantity] )
```

Total Delivered To Date

```
= CALCULATE(
    [Quantity Delivered ];
      DATESBETWEEN( Dates[Date];
                    DATE( 1900 ; 1; 1) ;
                    MAX( Dates[Date]      )
                    )
    )
```

Total Stock

- Create a measure to get the total start value for the stock
  - Qty Stock Start = SUM(Products[ProductStockStart])

- Create a measure to calculate the current level of the stock
  - Total Stock = [Qty Stock Start] + [Total Delivered To Date] - [Total Quantity To Date]

- Create a table using displaying these values per month
- Add a slicer for the year

Total Stock

MonthOfYear	Qty Stock Start	Qty Delivered	Qty Delivered TD	Qty Ordered	Qty Ordered TD	Qty In Stock
1	199	66	2141	58	2119	221
2	199	59	2200	67	2186	213
3	199	69	2269	61	2247	221
4	199	81	2350	79	2326	223
5	199	91	2441	90	2416	224
6	199	111	2552	114	2530	221
7	199	87	2639	84	2614	224
8	199	102	2741	108	2722	218
9	199	72	2813	68	2790	222
10	199	92	2905	95	2885	219
11	199	106	3011	104	2989	221
12	199	70	3081	67	3056	224
Total	**199**	**1006**	**3081**	**995**	**3056**	**224**

Year

- [ ] Y 2010
- [ ] Y 2011
- [ ] Y 2012
- [ ] Y 2013
- [ ] Y 2014
- [x] Y 2015
- [ ] Y 2016

Date comparisons

SAMEPERIODLASTYEAR

- Calculate the measure over the same period of the previous year
- Requested period
  - Total Sales = CALCULATE([Deals Expected] ; Orders[Status] = "Closed")

- Period previous year: Actual Orders PY
  - Total Sales PY=

```
CALCULATE(
    [Total Sales] ;
    SAMEPERIODLASTYEAR( Dates[date] )
    )
```

Turnover Previous Year

- Calculate the Total Sales of the previous year:
  Total Sales PY

```
=CALCULATE( [Total Sales];
                    SAMEPERIODLASTYEAR( Dates[Dates] ) )
```

- Create a Table/Matrix visual
- Calculate the growth Year over Year : Total Sales YoY

```
=DIVIDE(   [Total Sales] - [Total Sales PY] ;
                    [Total Sales PY] ;
                    BLANK()
                    )
```

DATEADD()

- DATEADD(Dates; nbr Intervals; Interval)
- Calculate the Total Sales of 2 years ago using DATEADD:
  Total Sales -2Y

```
=CALCULATE( [Total Sales]
                ; DATEADD( Dates[Date]
                                ; -2
                                ; Year )
                )
```

Total Sales Previous Quarter

- Calculate the Total Sales of the selected period 1 quarter ago :

Total Sales PQ

```
=CALCULATE( [Total Sales]
                ; DATEADD( Dates[Date] )
                        ; -1
                        ; Quarter
                )
```

Total Sales QoQ

```
=DIVIDE(    [Total Sales] - [Total Sales PQ] ;
                [Total Sales PQ] ;
                BLANK()
                )
```

PARALLELPERIOD()

- PARALLELPERIOD(Dates; nbr Intervals; interval)
  - Returns the total of complete given interval

Year	Total Sales	Total Sales-1Year	Total Sales -1Y parallel
2016	**€ 15,837,994**	**€ 17,196,288**	**€ 17,196,288**
Q1	€ 3,752,443	€ 3,371,715	€ 17,196,288
Q2	€ 3,977,716	€ 4,791,135	€ 17,196,288
Q3	€ 4,198,259	€ 4,204,881	€ 17,196,288
Q4	€ 3,909,576	€ 4,828,557	€ 17,196,288
2017	**€ 17,729,447**	**€ 15,837,994**	**€ 15,837,994**
Q1	€ 3,874,512	€ 3,752,443	€ 15,837,994
Q2	€ 5,006,865	€ 3,977,716	€ 15,837,994
Q3	€ 4,141,500	€ 4,198,259	€ 15,837,994
Q4	€ 4,706,570	€ 3,909,576	€ 15,837,994
Total	**€ 33,567,441**	**€ 33,034,282**	**€ 33,034,282**

```
TotalSales-1Y parallel =
CALCULATE( [Total Sales];
        PARALLELPERIOD( Dates[Date]
        ;-1; YEAR )
        )
```

Compare to previous month

- Imagine a situation where you want to compare the selected month (month to date) with the total of last month

Day	Total Sales	perc last month
12		40.54%
13		40.54%
14		40.54%
15	71,375.45	45.56%
16	223,958.43	61.30%
17	94,685.76	67.96%
18	5,407.46	68.34%
19		68.34%
20		68.34%
21		68.34%
22	9,036.47	68.97%
23	62,036.99	73.33%
24	47,050.01	76.64%
25	324,529.70	99.45%
26	52,140.08	103.12%
27		103.12%
28		103.12%
29	27,971.48	105.09%
30	87,270.51	111.22%
31	117,345.00	119.47%
Total	**1,699,479.20**	**119.47%**

Compare to previous month

- Total Sales PMC

```
= CALCULATE( [Total Sales] ;
                PARALLELPERIOD( Dates[Date]; -1 ; MONTH )
                )
```

- Perc Last Month

```
= DIVIDE(
    TOTALMTD([Total Sales]; Dates[Date]) ;
    [Total Sales PMC] )
```

Visualisations in Power BI

Visualisaties in Power BI

- Report design and Page-settings
- Add, Edit and Delete visuals
- Format visuals
- Slicers, Filters and interactions
- Tooltips
- Buttons and bookmarks

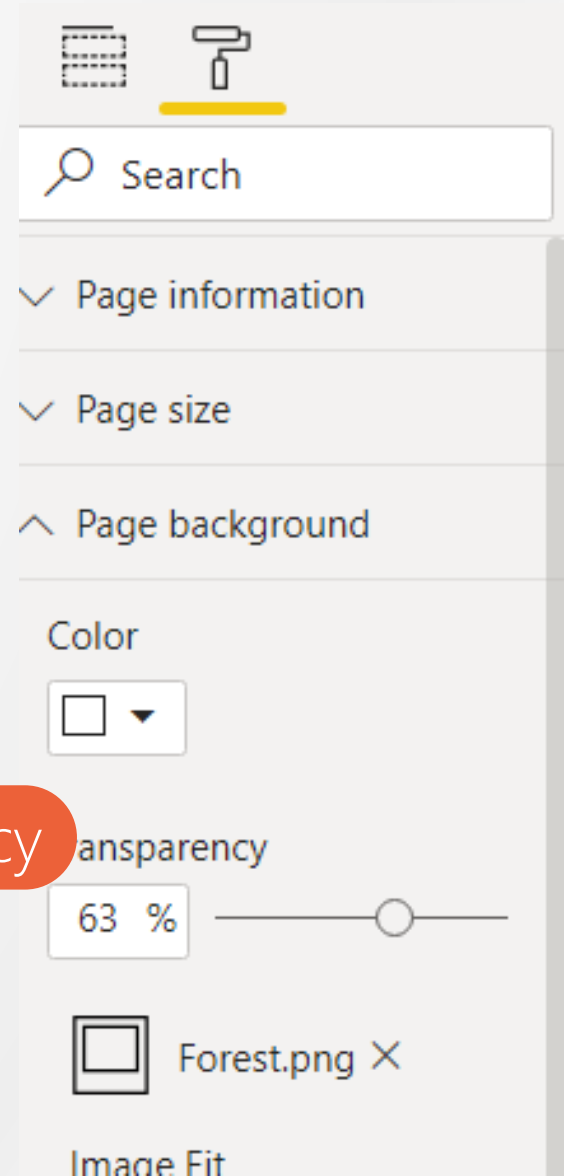
Report design and Page-settings

- Apply a background image or color
- Click on the page and choose 

Alignment of visuals

- Use gridlines
  - View / Show Gridlines

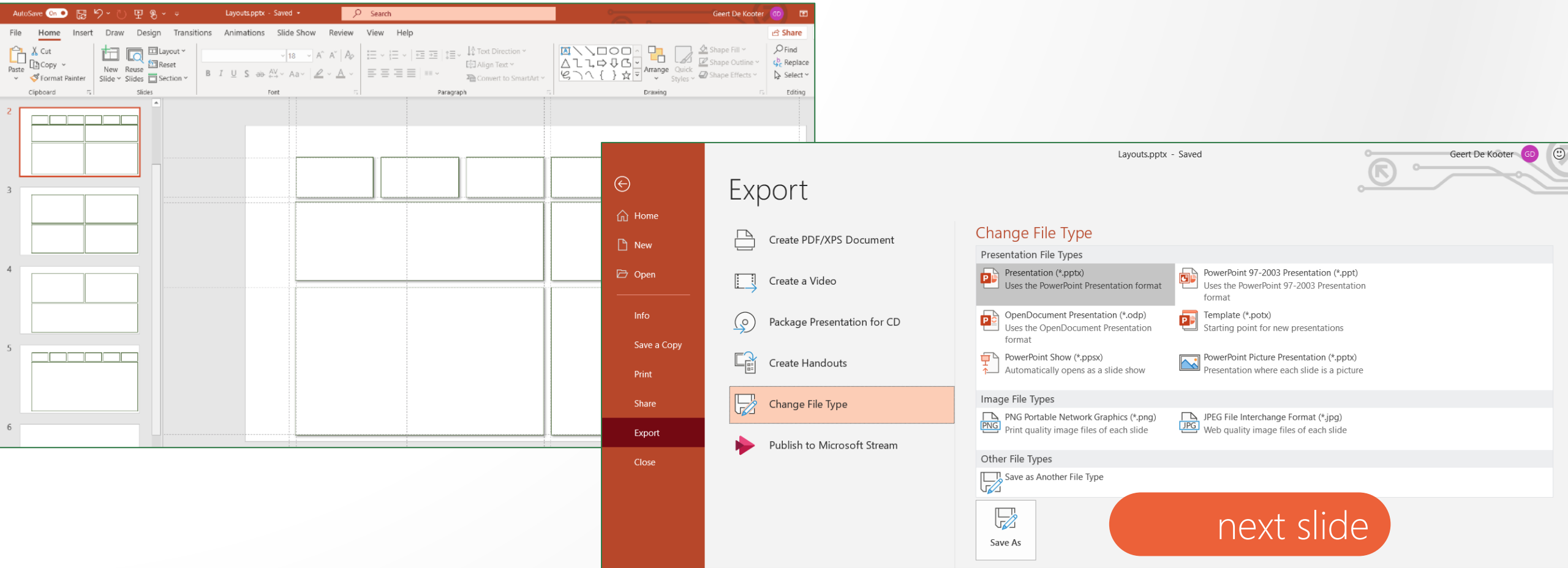
  Or
- Apply a background image with the necessary canvas



Next slide

Create a Background canvas

- Open PowerPoint and draw a canvas
- Choose File / Export and save as PNG



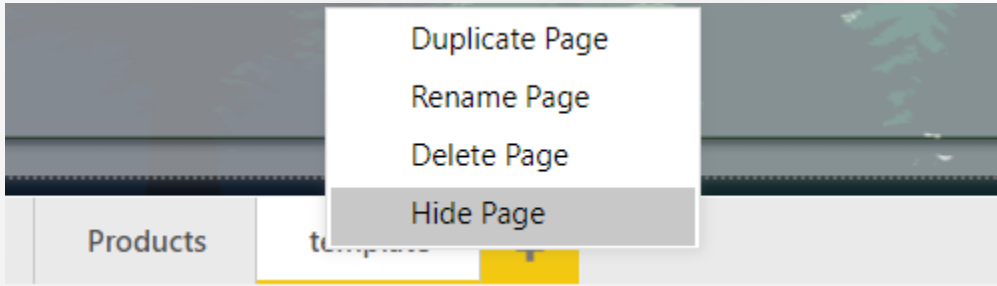
Apply a Background canvas

- Click on the page and choose
- In the section Page background, click Add image
- Change the Transparency

Tip

Canvas and background are applied to the current page

- Right click to create a duplicate
- Create a "hidden template"



Themes

- Standard themes
- Browse for themes voor Company branded themes
- Customize & save current Theme

LEARNIA

Apply a background and theme

- Open 5_A_01_Page_Report.pbix
- Set Forest.png as wallpaper
- Apply the Training.json theme
- Reorganise the visuals

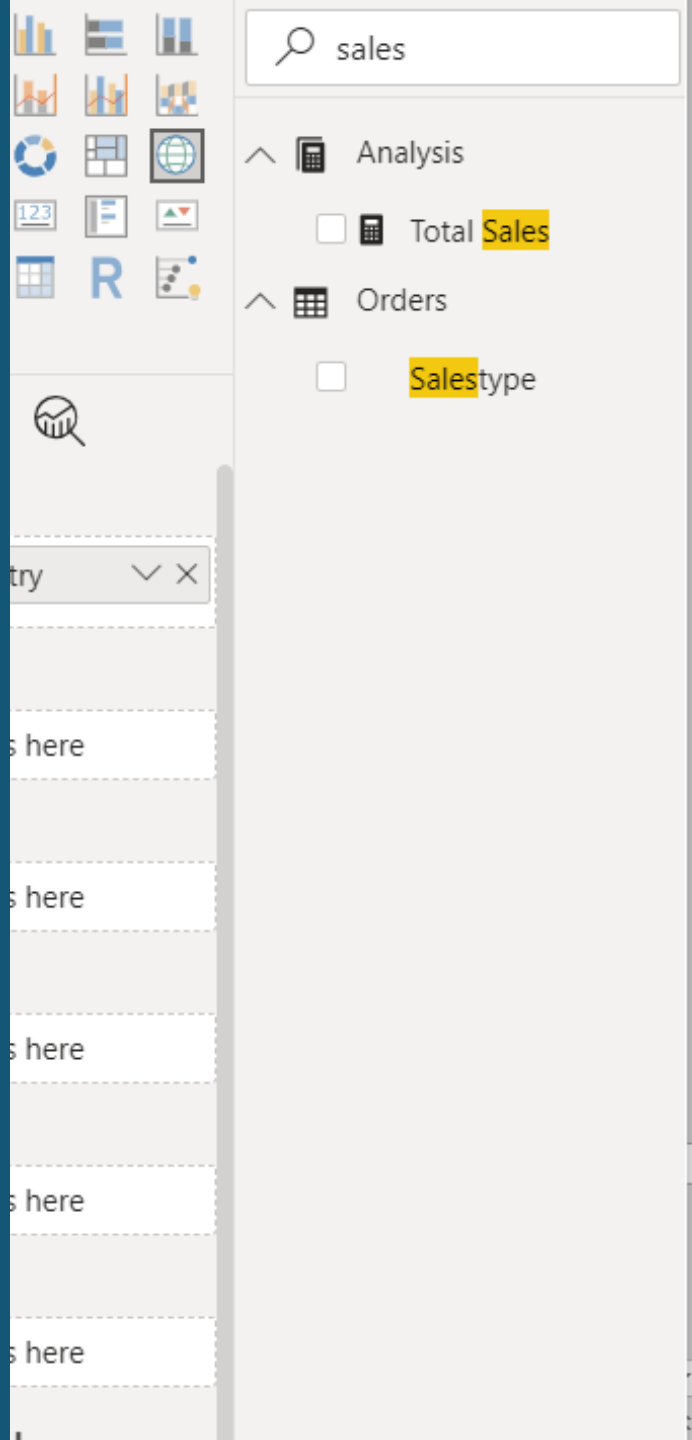
- Change the transparency of the
  - Wallpaper
  - Visuals

Add visuals

Adding a visual

Visual – Data – Format – Filter

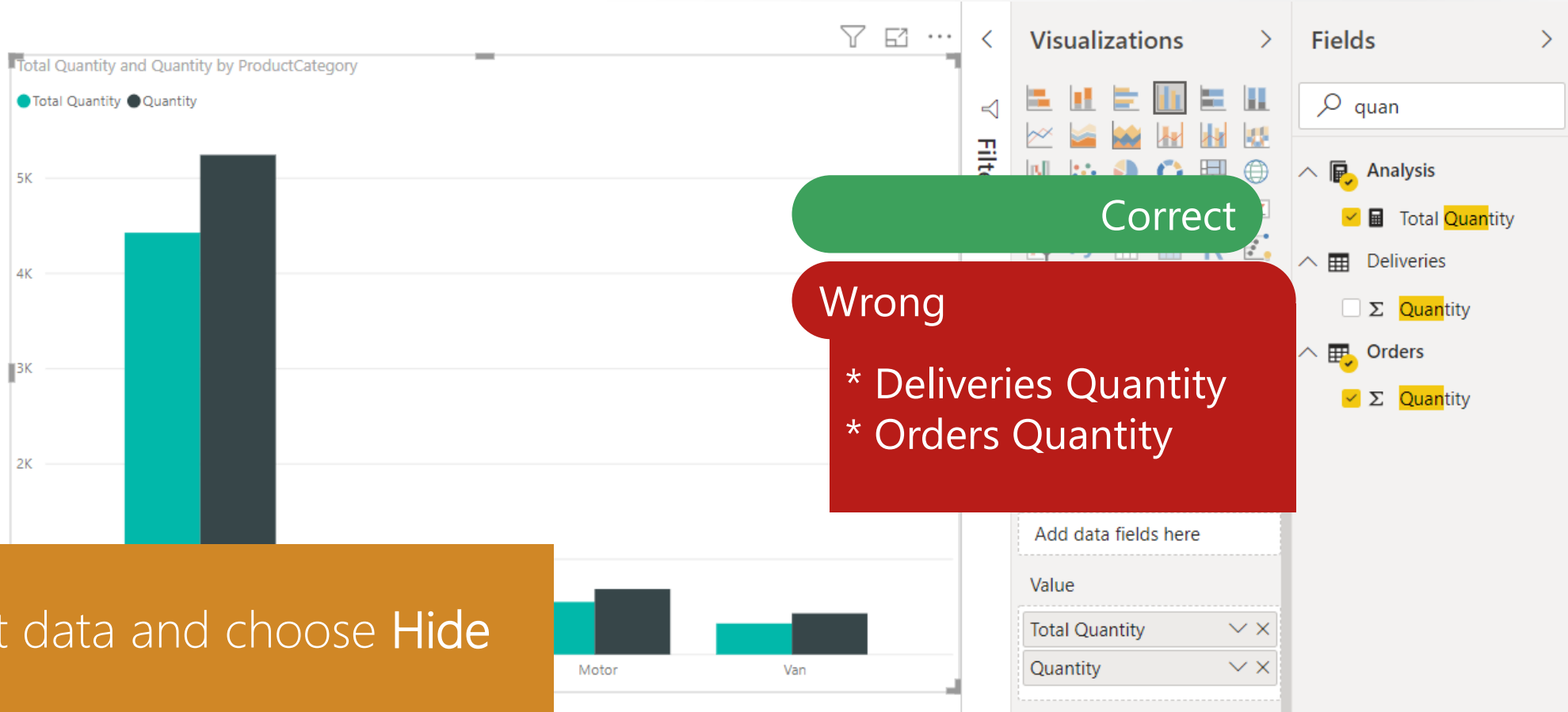
1. Click anywhere on your Page
2. Select the desired visual
3. Find and add the Fields and Measures
4. Change the formatting
5. Adjust filters toe



LEARNIA

Tip

- Don't add summaries of fact data
- Create and use measures !!!



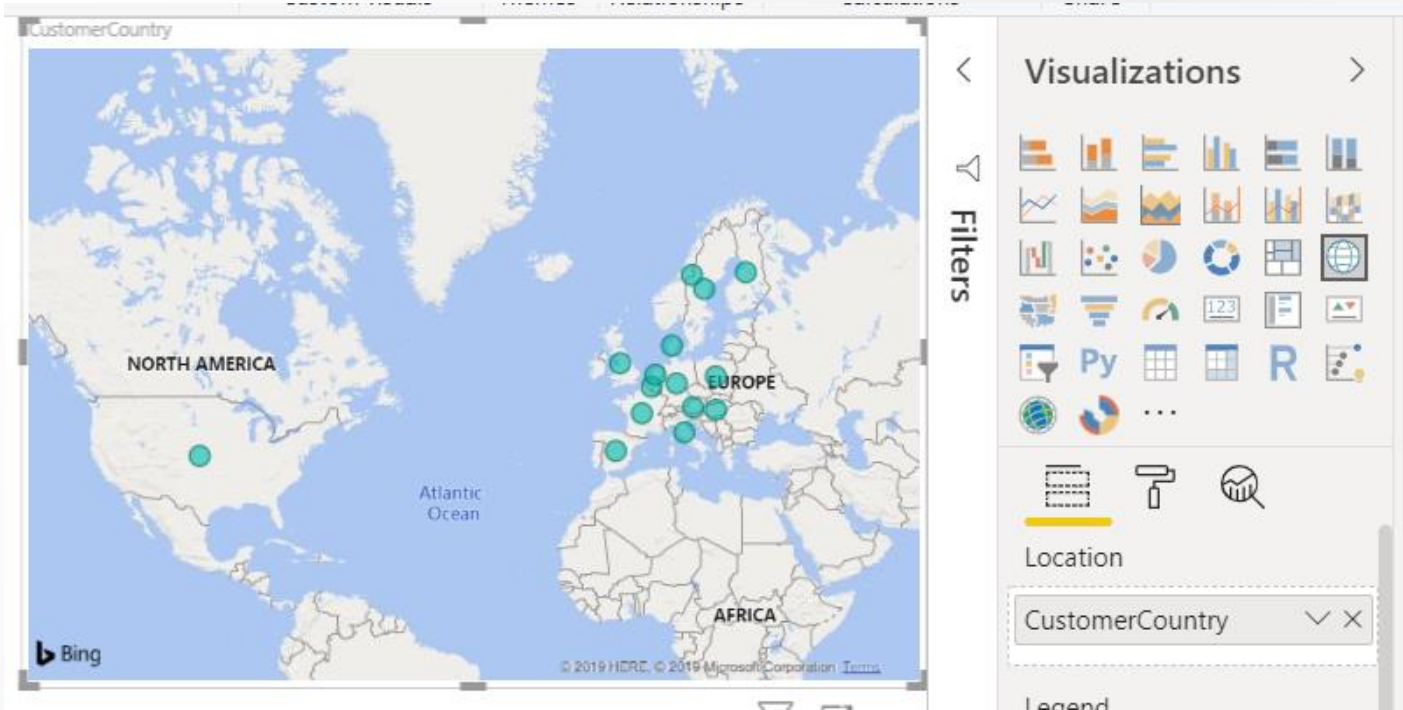
Tip:
Right click on the fact data and choose **Hide**

Default behaviour: Data - Visual

- Measure → Columnchart
- Tekst → Table
- Geographical data → Map

Change the visualtype

1. Select the visual
2. Click on a different type



Formatting of the Visuals

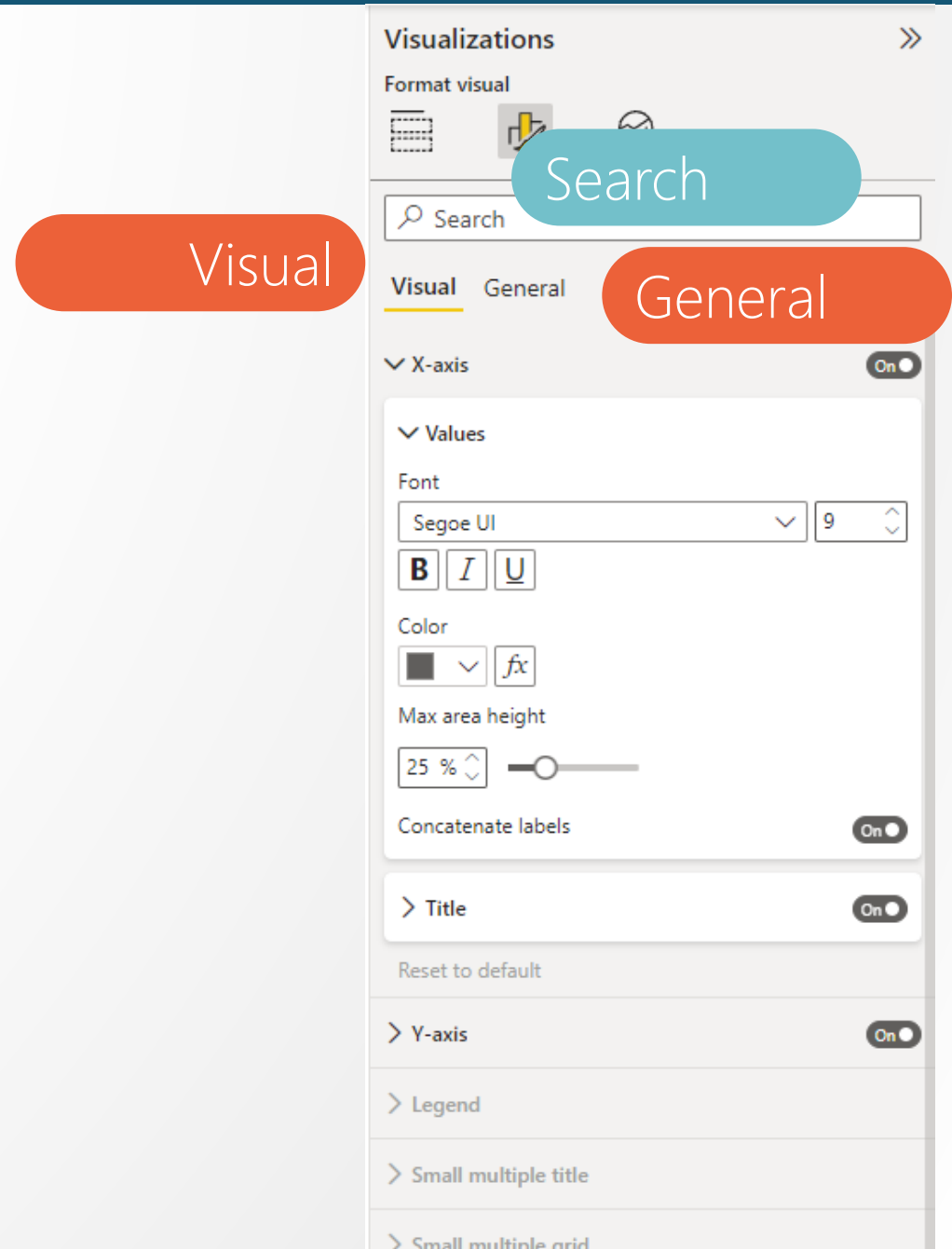
- Use a theme for the default settings
- Click 

Charts

- Each element can be given a different color

Tables

- Default formatting in the Style section
- Striped rows and columns
- Conditional Format



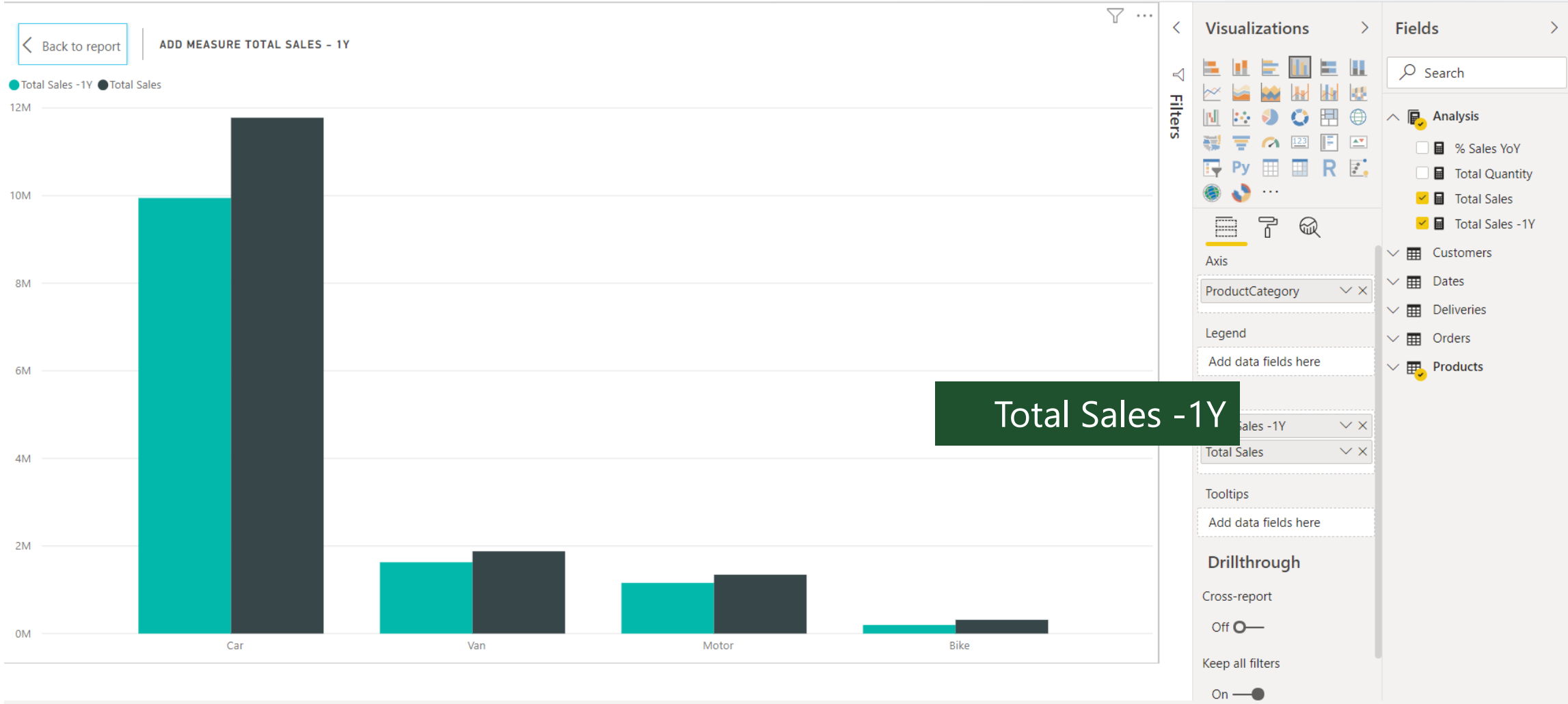
Context

Context

- Applying context to values gives a deeper meaning to the number you are looking at
  - This year's value compared to last year's
  - This category compared to all categories
  - An Average with the Standard Deviation and the size of the population

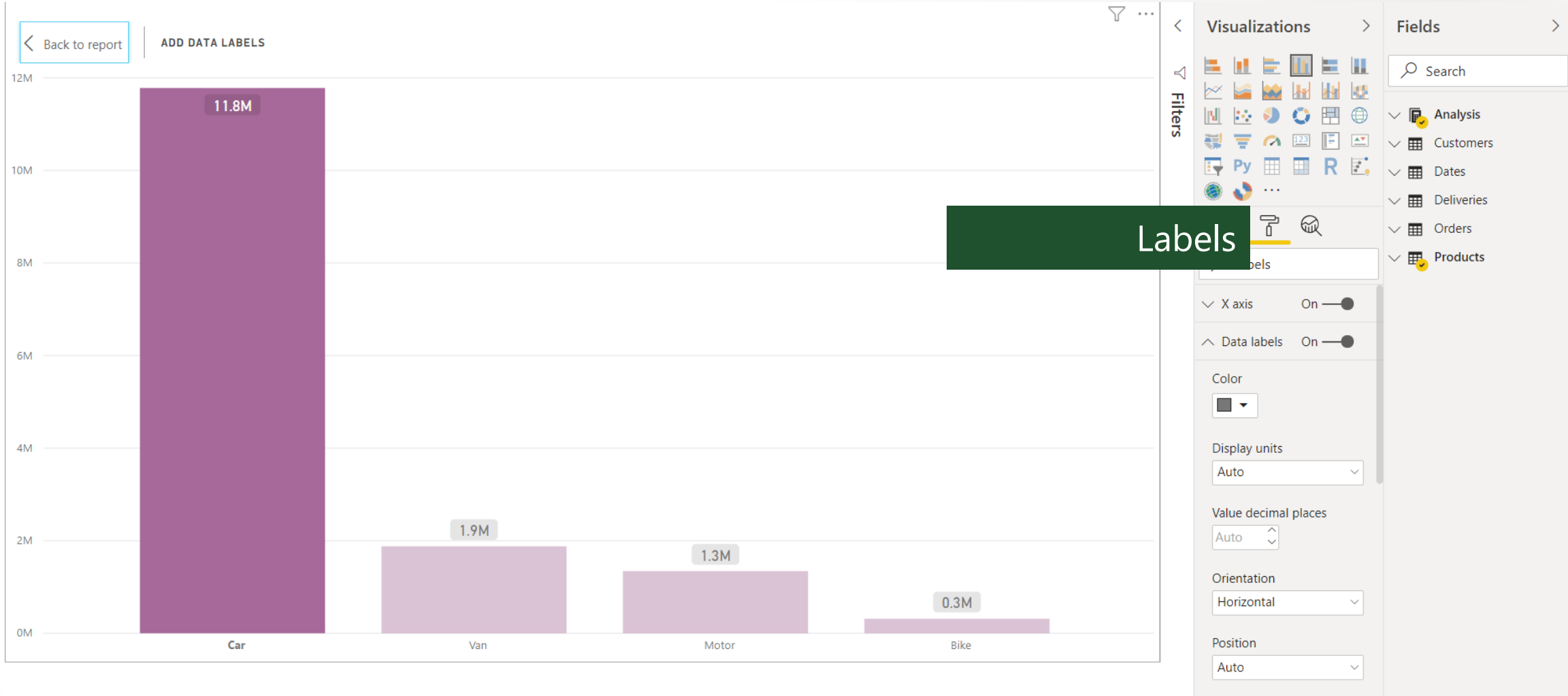
Another measure

Comparing with last year's values using a measure



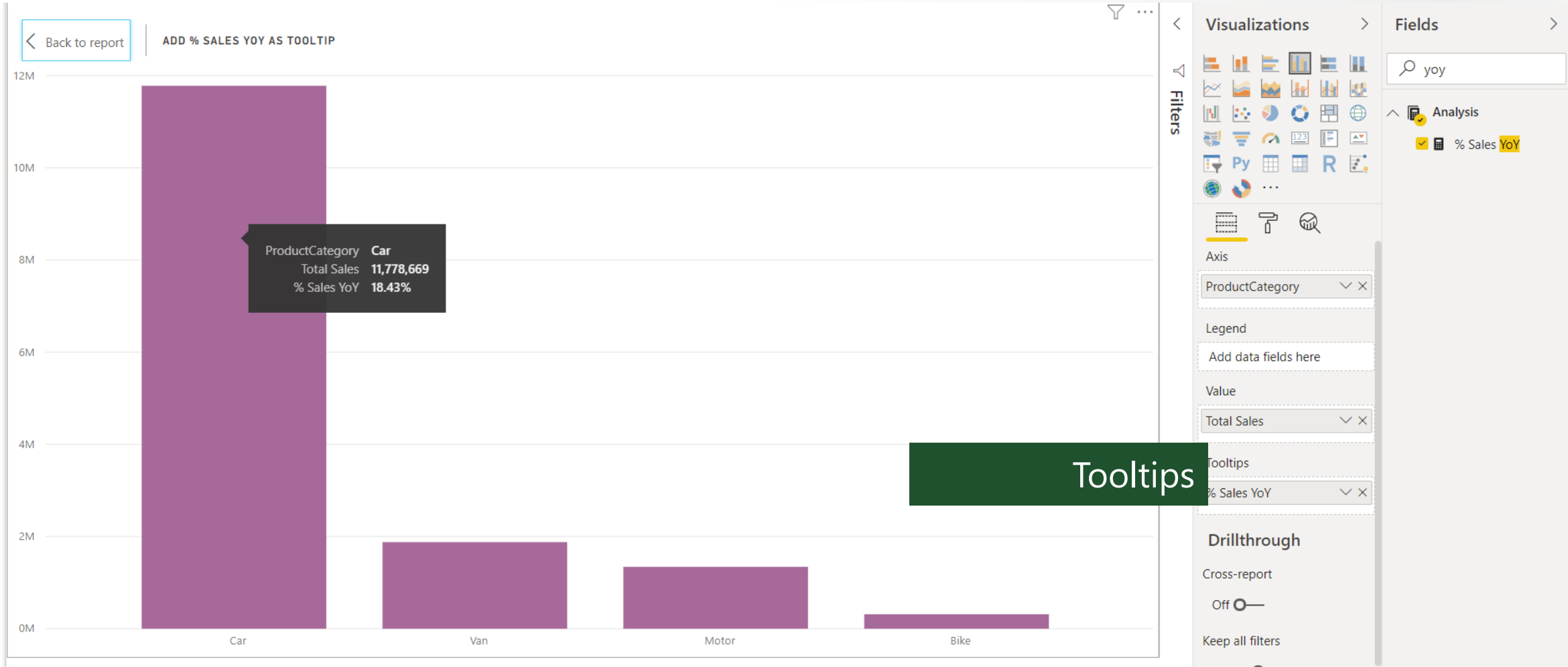
Data labels

Apply data labels for easy interpretation



Normal tooltips

Add additional information in the tooltip



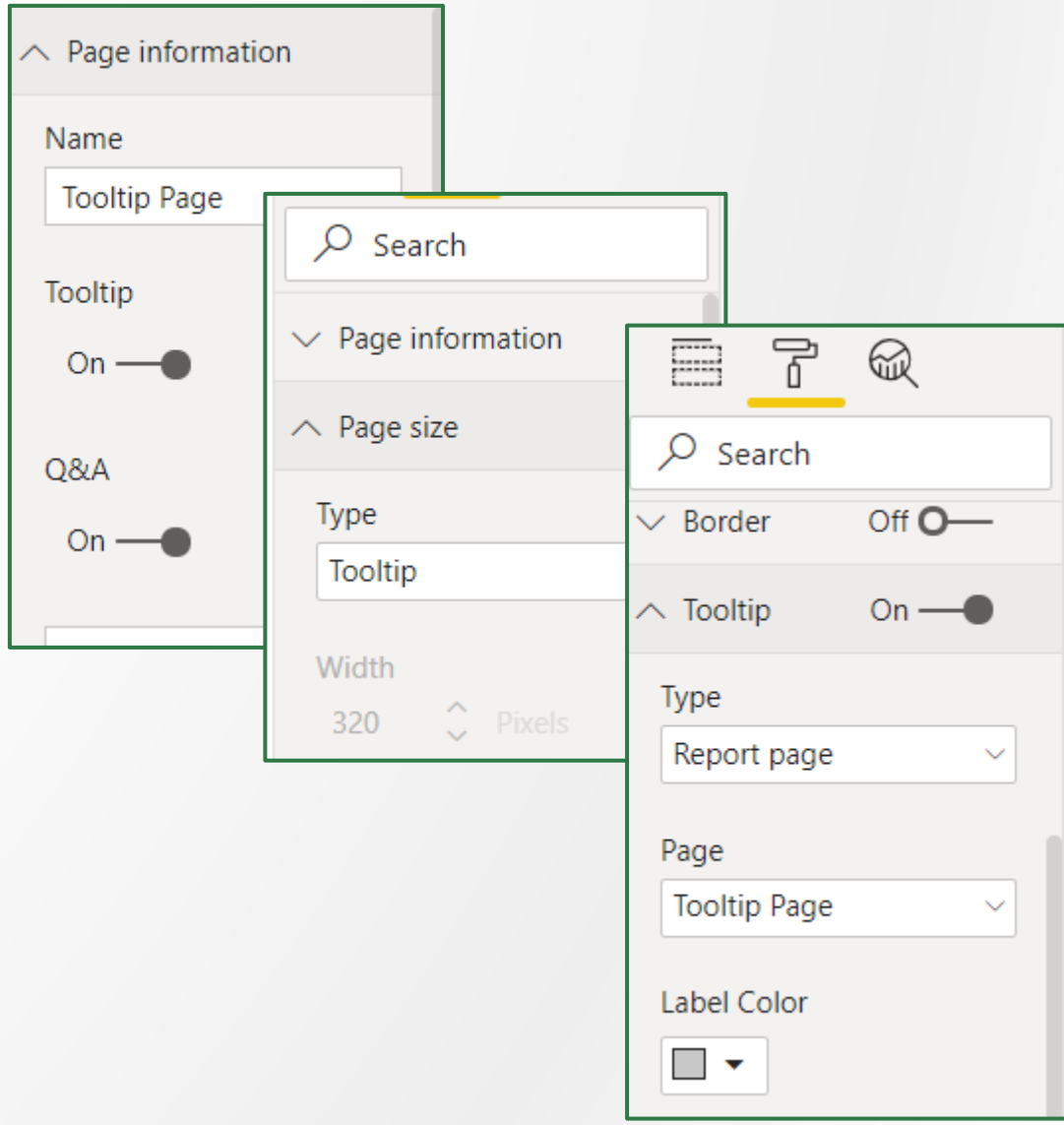
Tooltip Page

Create a tooltip page

1. Create a separate page
2. Change the page settings
    1. Page Information: Tooltip – On
    2. Page size: Tooltip
3. Add visuals

Use a tooltip page

1. Select a visual
2. Settings Tooltip, select the page



Tooltip Exercise

- In the file 5_B_01_Visuals.pbix page Exercise

  Average delivery in days for late deliveries (Delivery > 3 days)

1. Add labels to the visual

2. Calculate the StDev for late deliveries and add this to the tooltip

   StDev late delivery

```
= STDEVX.P(  FILTER( Orders ;
            Orders[SendDate] - Orders[Orderdate] > 3 ) ;
            Orders[SendDate] - Orders[Orderdate]
)
```

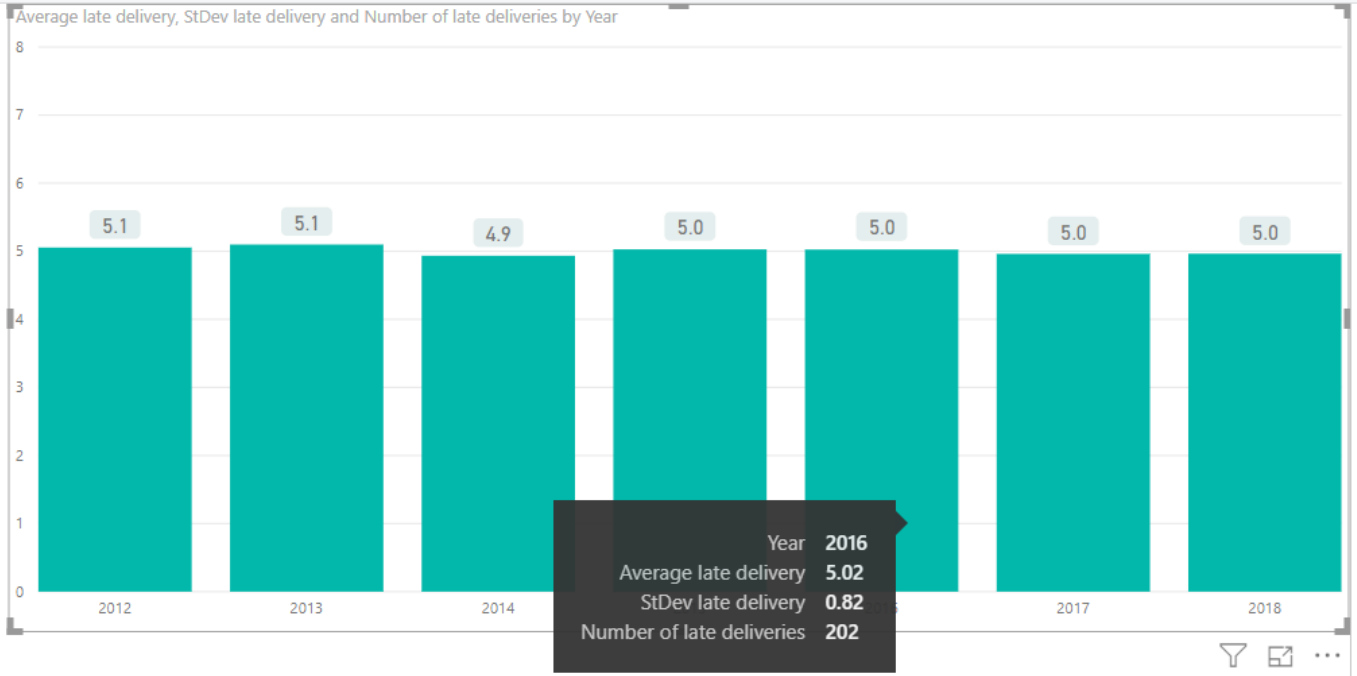
Tooltips exercise

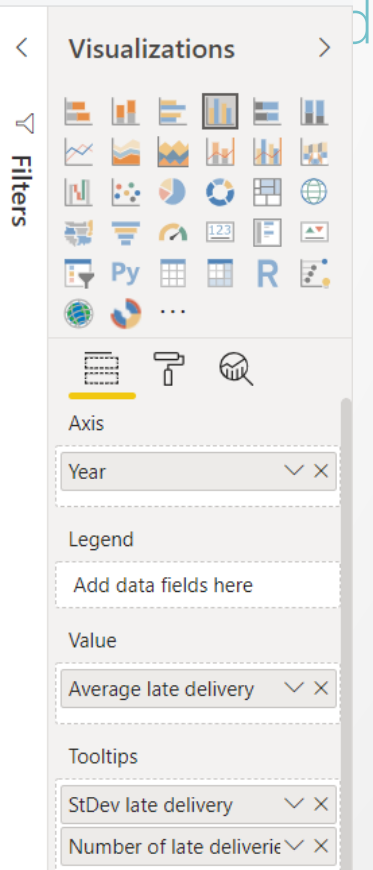
Count the number of late deliveries and add this to the tooltip

Number of late deliveries

= COUNTROWS(FILTER(Orders ;

date] > 3)





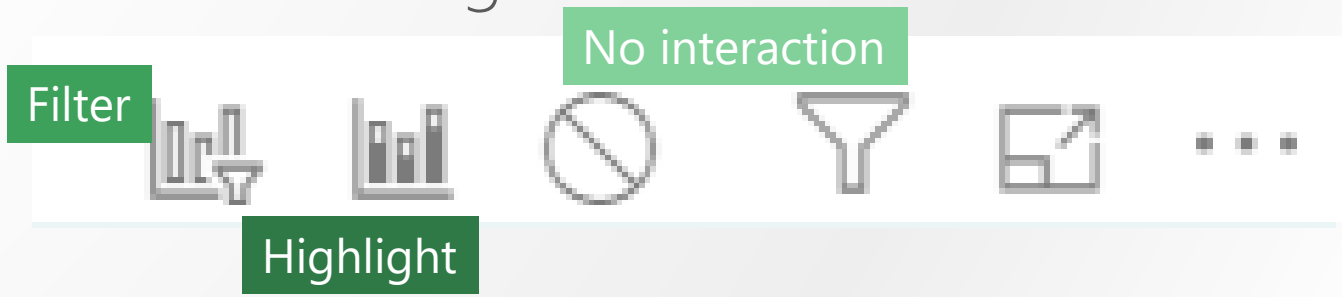
Slicers

1. Go to 5_B_02_Interactions.pbix page Slicers exercise
2. Add a SalesType slicer
    1. Allow multiple selections via CTRL
    2. Change the list to buttons
    3. Change the colour of the buttons
3. Add a slicer on the year
    1. Change the list to buttons
    2. Make it only interact with the PIE chart

Interaction

- Visuals interact with each other
- The interaction can be stopped/changed

1. Select the source visual
   - The visual that should/shouldn't control other visuals
2. Click **Visual Tools – Format** and click **Edit interactions**
3. Use the buttons on the other visuals to change the interaction



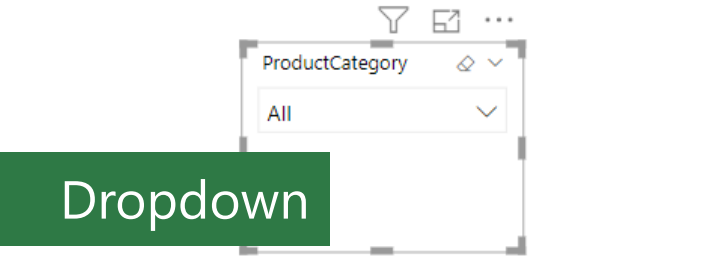
Change the interactions

Open the file 5_B_02_Interactions.pbix

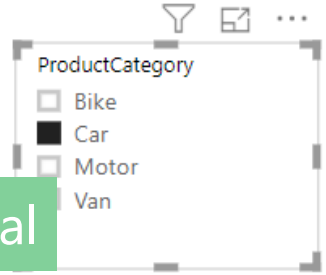
1. Change the interactions between the Year slicer and the charts
2. Remove the interaction between the different Total … by Year charts

Slicers

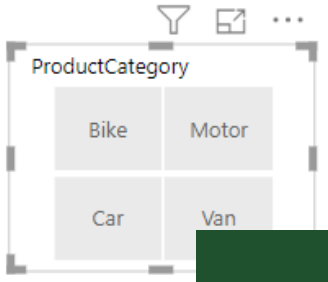
- Slicers are a very visual way to filter the data
- One slicer, mutliple modes



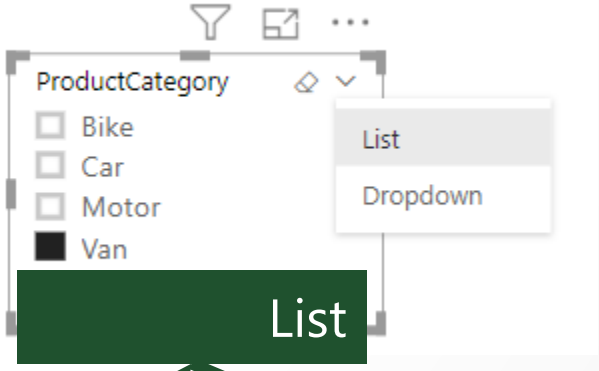
Dropdown



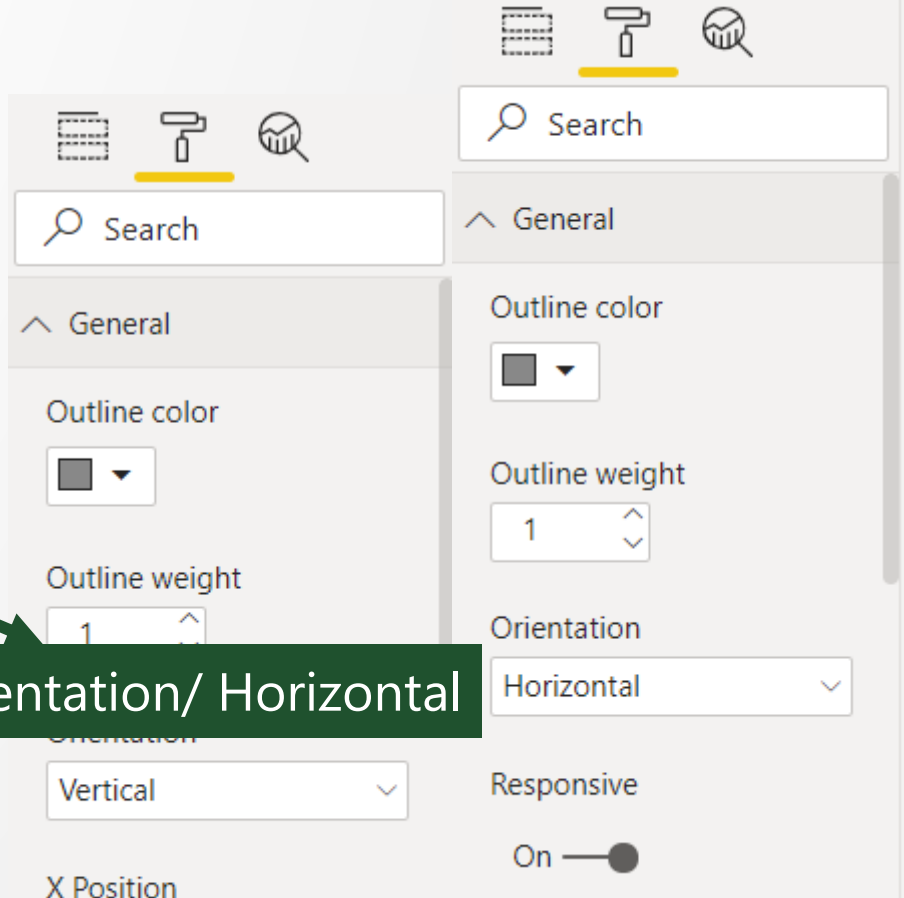
List - vertical



List - horizontal

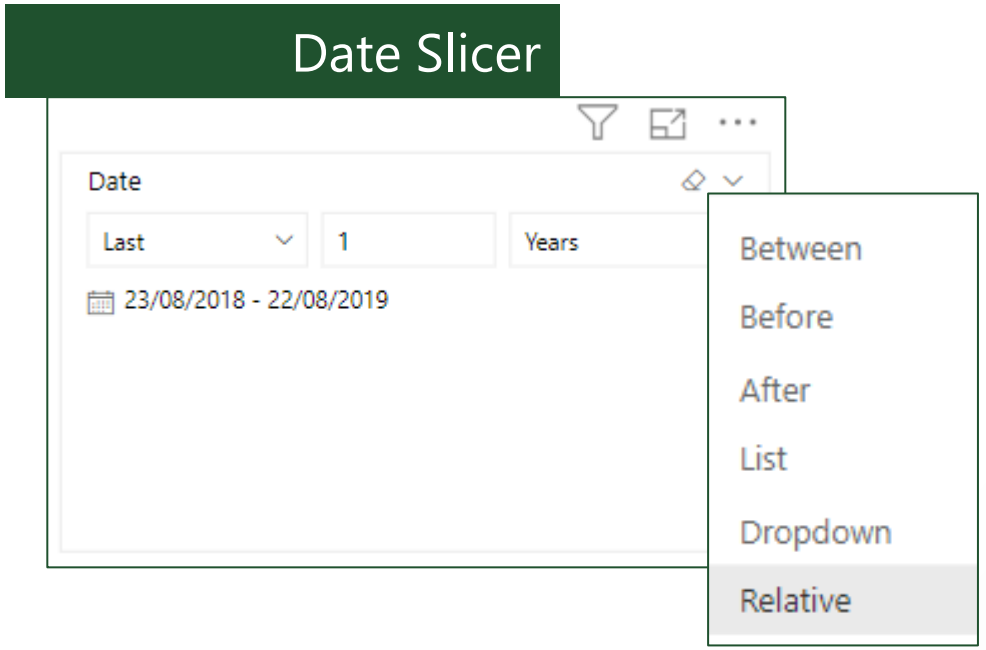


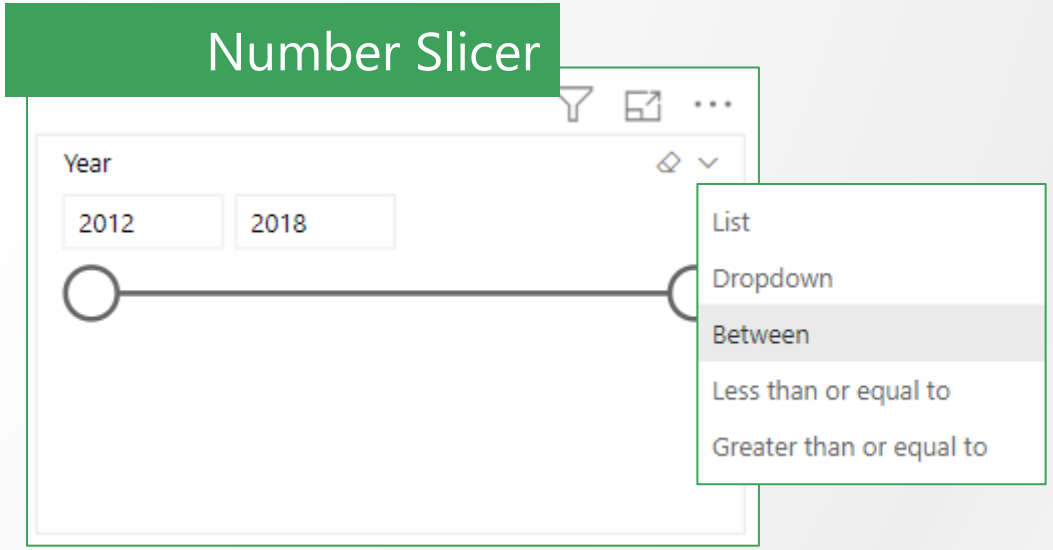
List



General/ Orientation/ Horizontal

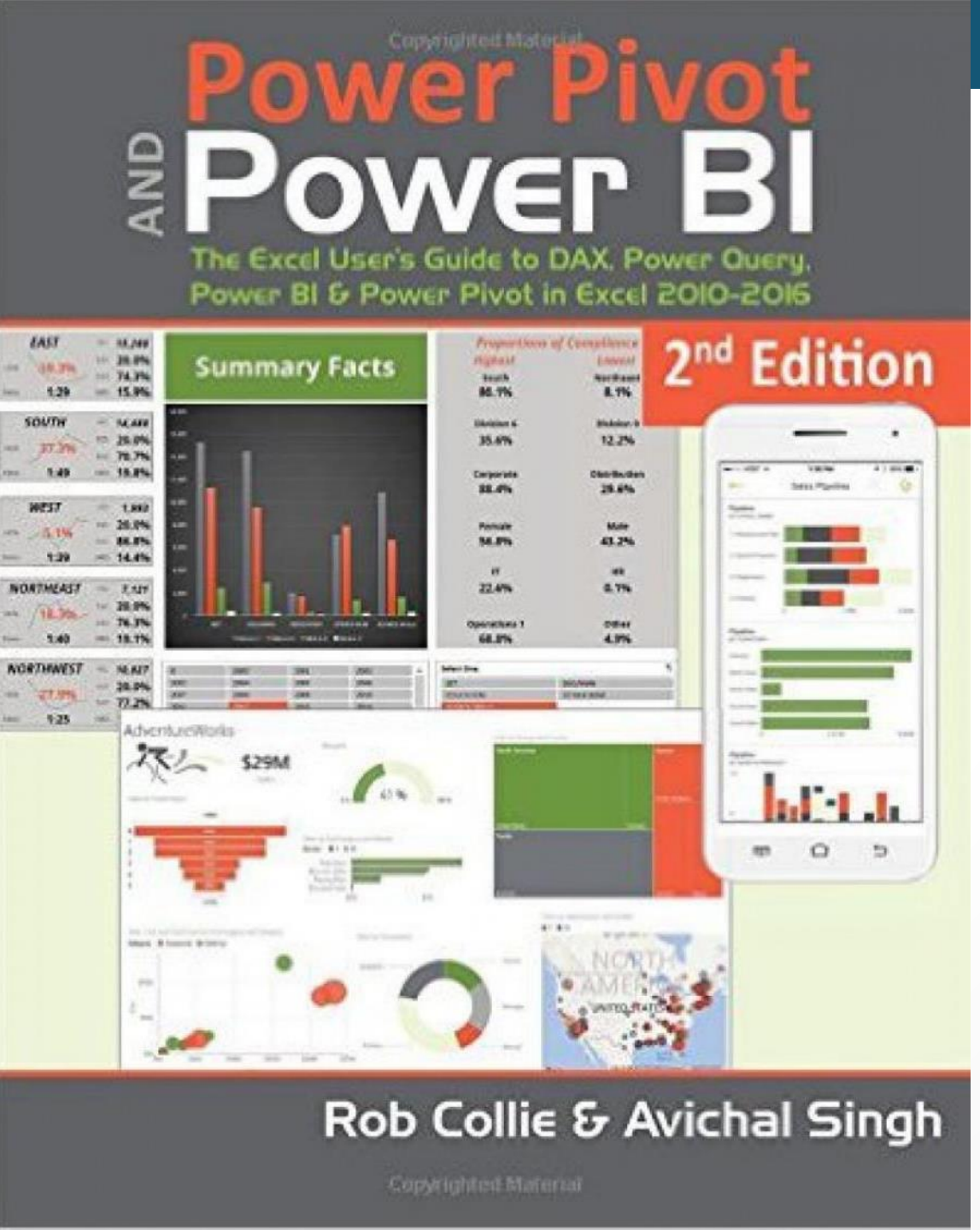
Date and number slicers





Getting **coffee**

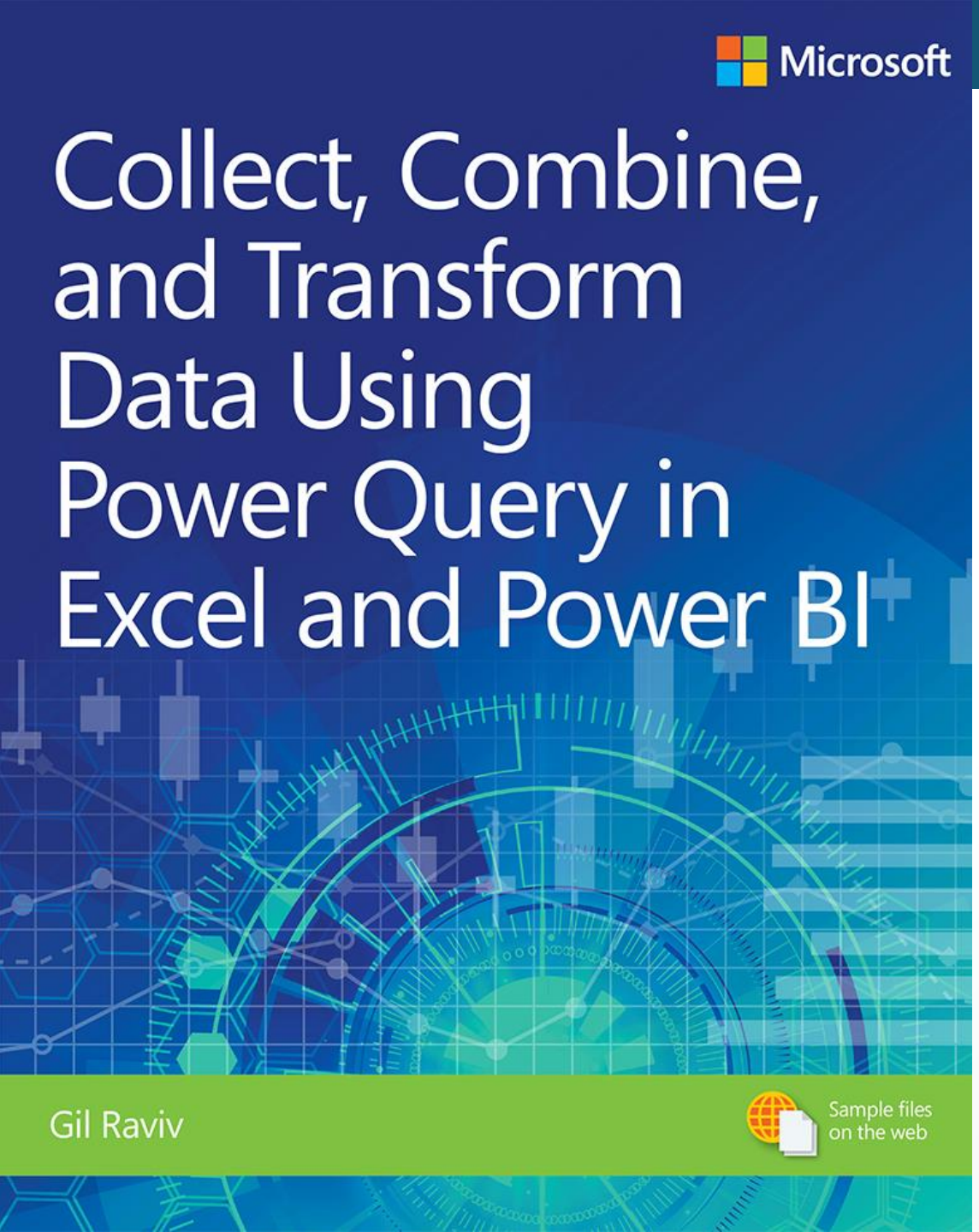
References & Guides



LEARNIA

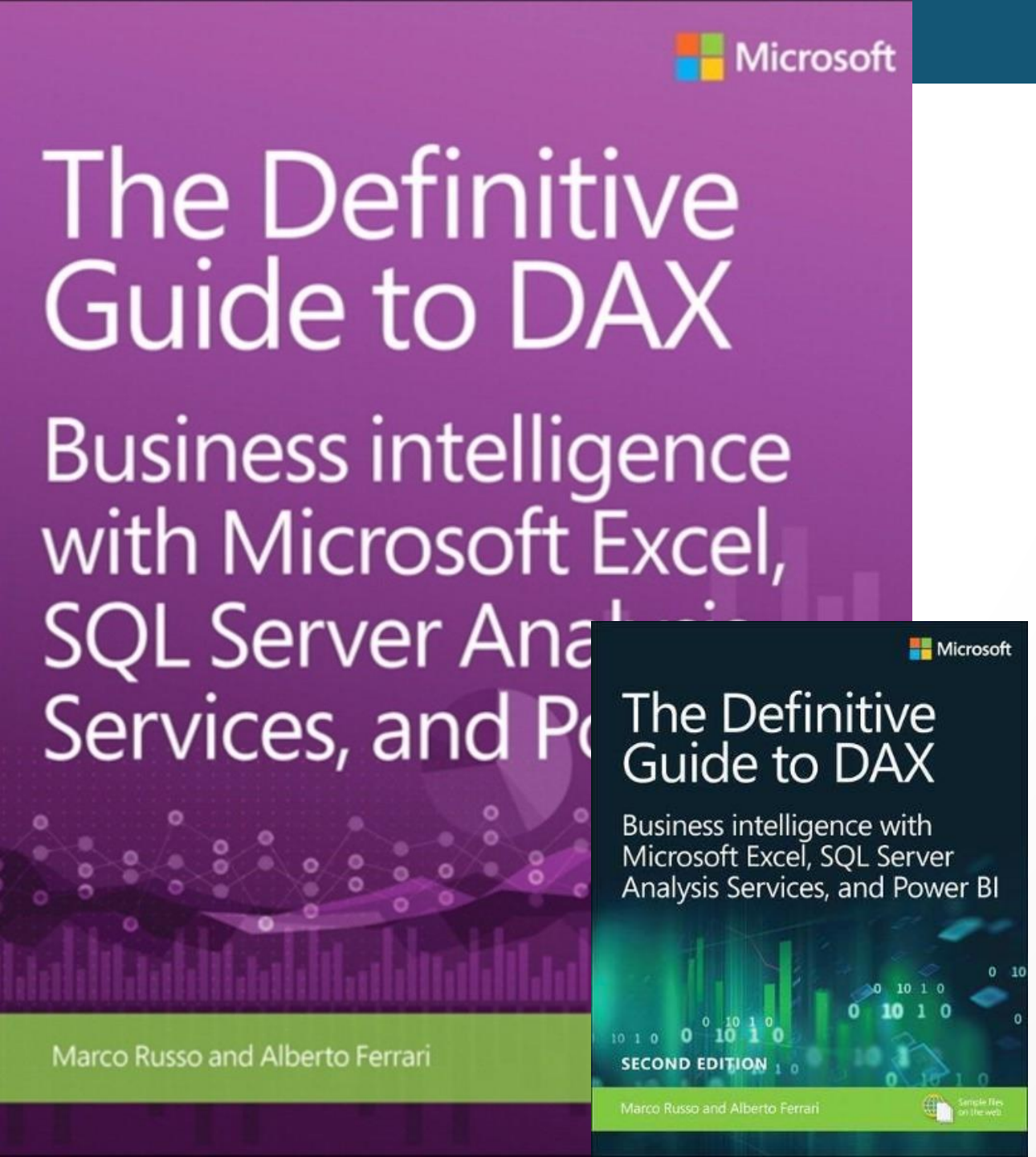
Power Pivot and Power BI

- Rob Collie & Avichal Singh
- English
- Focus
  - From basics to advanced
  - All Power Pivot versions supported



Collect, Combine and Transform Data using Power Query in …

- Gil Raviv
- English
- Focus on
  - Power Query
- Missing



The Definitive Guide to DAX

- Alberto Ferrari & Marco Russo
- English
- Focus on
  - PowerPivot DAX
- Missing
  - Power BI
  - Power Query

Websites

- https://docs.microsoft.com/en-us/power-bi/
  - Official Documentation
- https://powerbi.microsoft.com/en-us/blog/
  - Official Blog (What's new)
- http://www.powerpivotpro.com/
  - Rob Collie
- https://www.sqlbi.com/
  - Marco Russo and Alberto Ferrari
- https://www.kasperonbi.com/
  - Kasper de Jonge

Peter De Laet

Trainer &
Training Manager
Peter.Delaet@learnia.be

LEARNIA
Partners in efficient learning

DANK U

MERCI GRACIAS

DANKE THANK YOU