

Advanced Data Analytics



Power BI

Let me introduce...

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

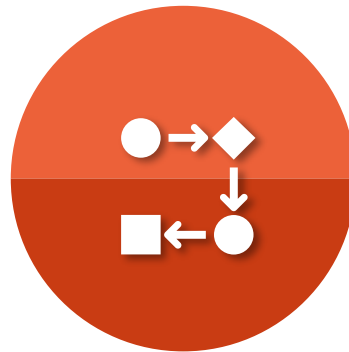


Our Journey in 3 sessions



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

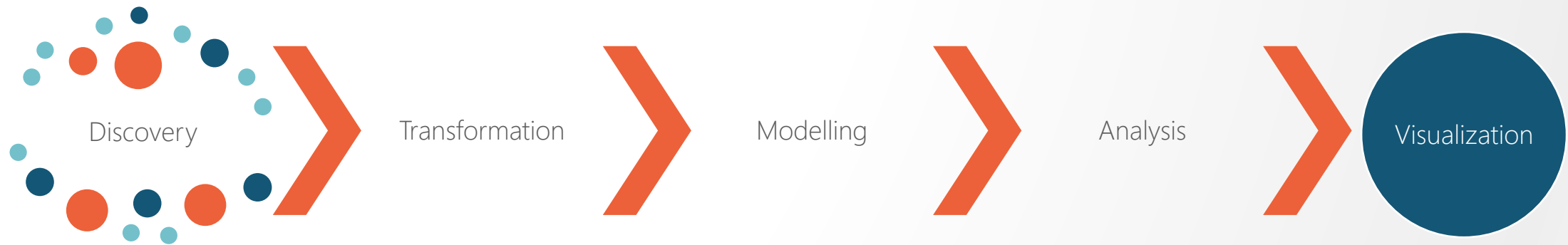
Definition

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

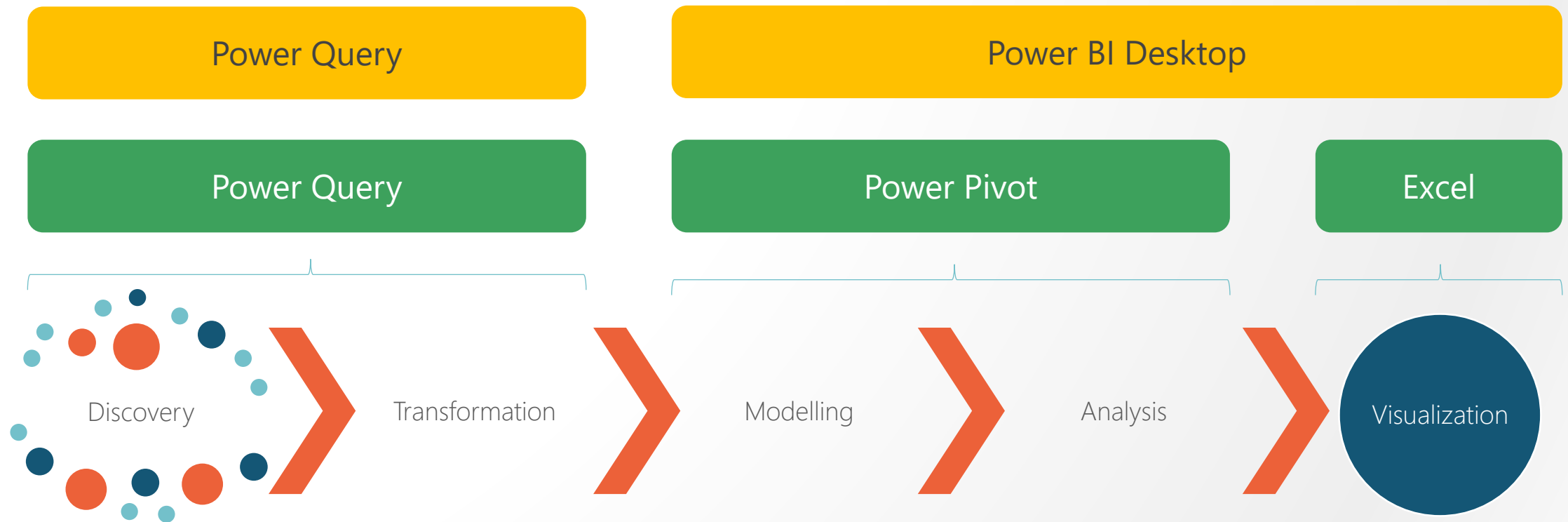
Microsoft self service BI

- 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

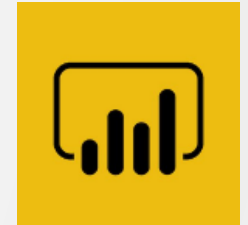
Self-service-BI



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



Getting started with Power BI

Power BI Desktop vs Online

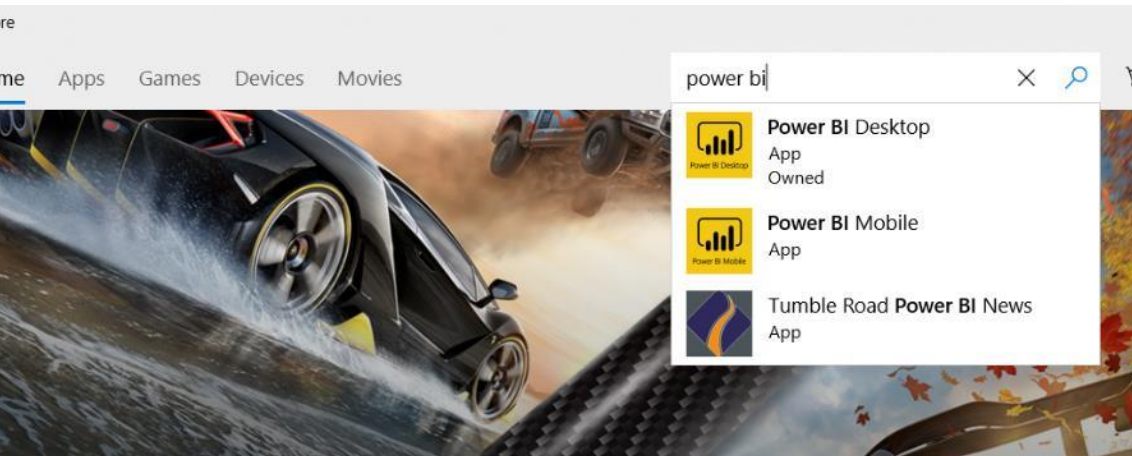
Power BI Desktop is a **staging tool**, designed to create reports that need to be published on PowerBI.com

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



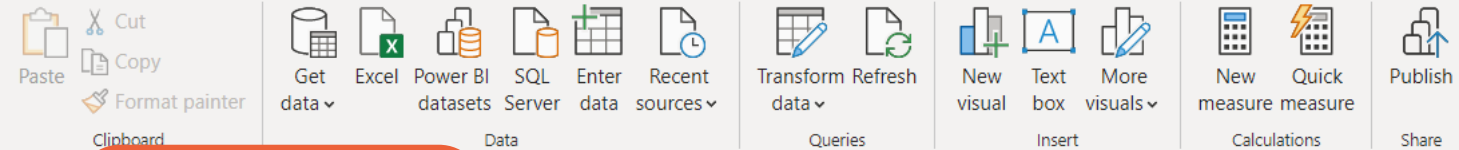
This block contains two overlapping screenshots. The background is a screenshot of the Power BI Desktop application window. The title bar says 'Power BI Desktop'. The main area is divided into sections: 'Get data' with a 'Recent sources' list containing 'rob_collie.pbix' and 'Geert.pbix', and 'Open other reports'. The foreground is a white sign-up form titled 'Welcome to Power BI Desktop'. It asks 'Where can we send you the latest tips and tricks for Power BI?' and includes input fields for 'First Name *', 'Last Name *', 'Email Address *', 'Enter your phone number *', 'Country/region *' (a dropdown), 'Company name *', and 'Job Role*' (a dropdown). Below these fields is a paragraph about Microsoft's use of contact information and a 'Done' button. At the bottom of the form is a link: 'Already have a Power BI account? Sign in'. Below the form, there is a yellow banner with 'TUTORIALS' and a list of links: 'Get started with Power BI Desktop', 'Download a sample', 'Watch our training videos', 'See what others have built', and 'All guided learning'. There is also a 'VIEW ALL VIDEOS' link and a checkbox for 'Show this screen on startup'.

Help and resources

More resources at the end of the slideshow

The screenshot displays the Microsoft Power BI Desktop application window. The title bar at the top reads "Untitled - Power BI Desktop" and includes the user name "Geert de Kooter" along with standard window controls. The ribbon menu at the top contains tabs for File, Home, Insert, Modeling, View, and Help. The "Help" tab is currently selected and highlighted with an orange callout bubble containing the word "Help". Below the ribbon, the main workspace area shows a prompt to "Add data to your report" with the instruction "Once loaded, your data will appear in the Fields pane." Below this prompt are four buttons: "Import data from Excel", "Import data from SQL Server", "Paste data into a blank table", and "Try a sample dataset". On the right side of the interface, the "Visualizations" pane is visible, showing various chart and table icons. The "Fields" pane is partially visible on the far right.

File Home Insert Modeling View Help External Tools



Report

Data

Model

Filter

Visuals

Fields

Search

- Analysis
- Countries
- Customers
- Dates
- Employees
- Orders
- Products

Values

Add data fields here

Drill through

Cross-refer

Off

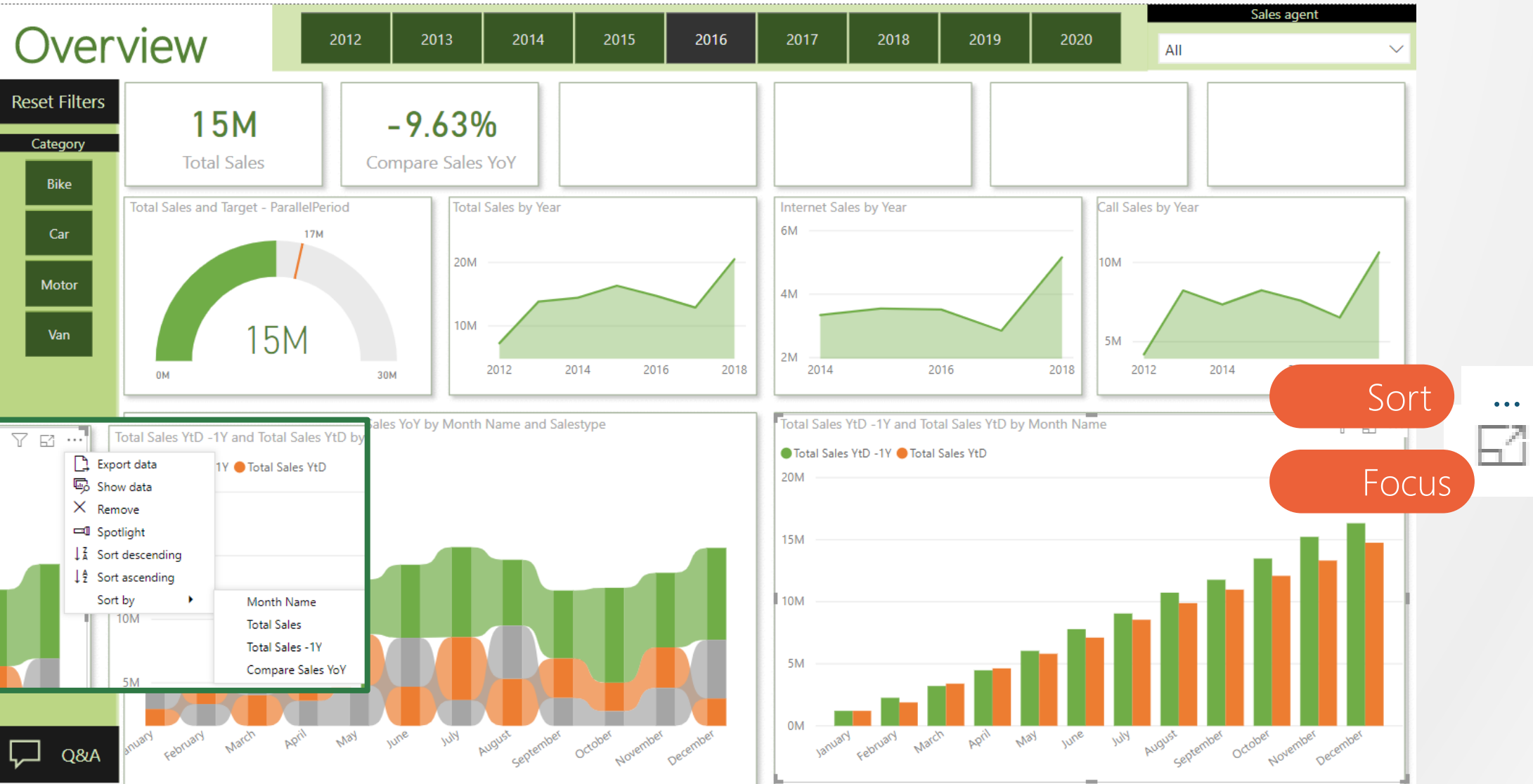
Keep all filters

On

Add drill-through fields here

Tables

Use Reports



Online

The screenshot shows the Microsoft Power BI online interface. At the top, there's a dark blue header with the 'LEARNIA' logo. Below it, a browser window displays the URL 'https://app.powerbi.com/home?noSignUpCheck=1'. The Power BI interface has a dark sidebar on the left with a menu: Home, Favorites, Recent, Create, Datasets, Apps, Shared with me, Learn, Workspaces, and My workspace. The main area is white and features a greeting 'Good afternoon, Geert' with a subtitle 'Find and share actionable insights to make data-driven decisions'. A yellow '+ New Report' button is in the top right. Below the greeting is a section 'Favorites + frequents' containing a grid of items: 'Training' (App), 'My workspace' (Workspace), 'Finance' (Workspace), 'Finance' (App), 'Example_2019' (Report), '5_B_03_Visuals_import' (Report), 'oktober 2020' (Dashboard), '5_E_finacial_Flow' (Report), and 'Main_exercises' (Report). Two orange callout boxes are present: one labeled 'Apps' pointing to the 'Apps' menu item, and another labeled 'Workspaces' pointing to the 'Workspaces' menu item.

Power BI Home

Search

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_finacial_Flow
Report

Main_exercises
Report

Apps

Workspaces

Online Workspace

Power BI My workspace

Search

My workspace

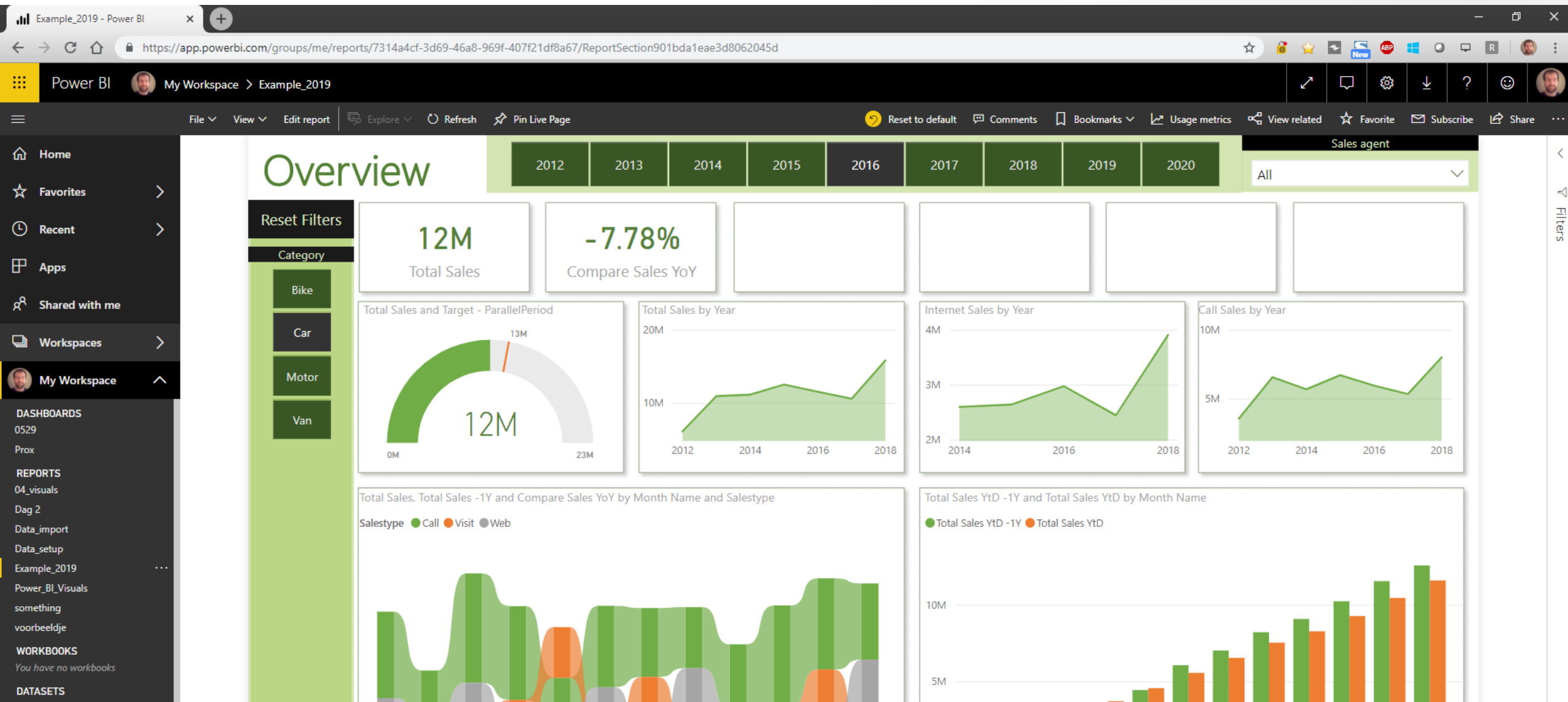
+ New View Filters Search

Name	Type	Owner	Refreshed	Next refresh	Endorsement
5_B_03_Visuals_import	Report	Geert de Kooter	26-9-20 09:42:52	—	—
5_B_03_Visuals_import	Dataset	Geert de Kooter	26-9-20 09:42:52 ⚠	N/A	—
5_E_finalcial_Flow	Report	Geert de Kooter	11-3-20 15:39:26	—	—
5_E_finalcial_Flow	Dataset	Geert de Kooter	11-3-20 15:39:26 ⚠	N/A	—
Example 2019	Report	Geert de Kooter	23-10-20 13:19:38	—	—

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

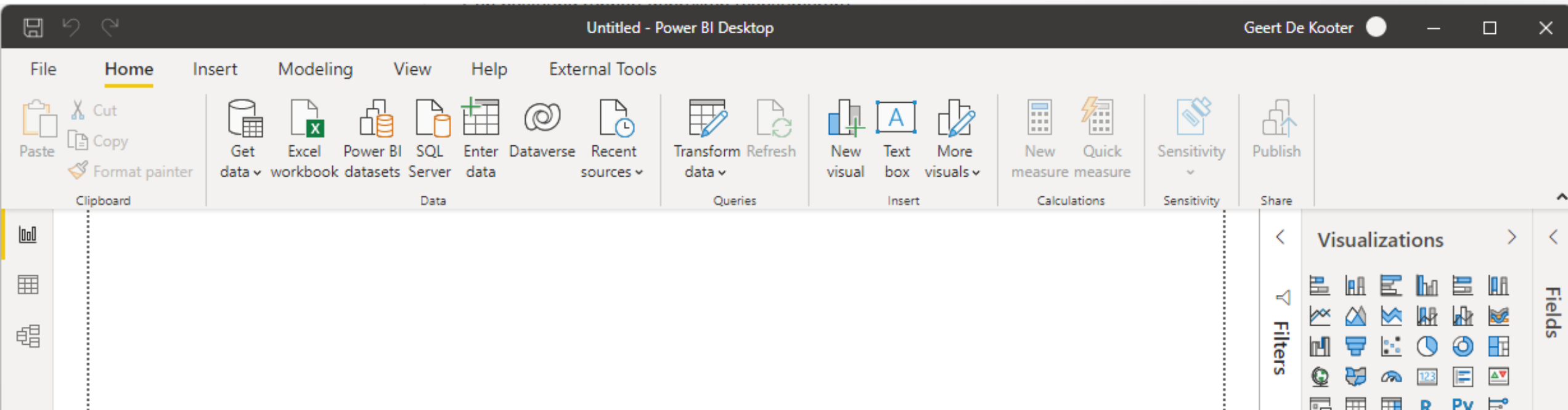




Create a report

Start a new Power BI report

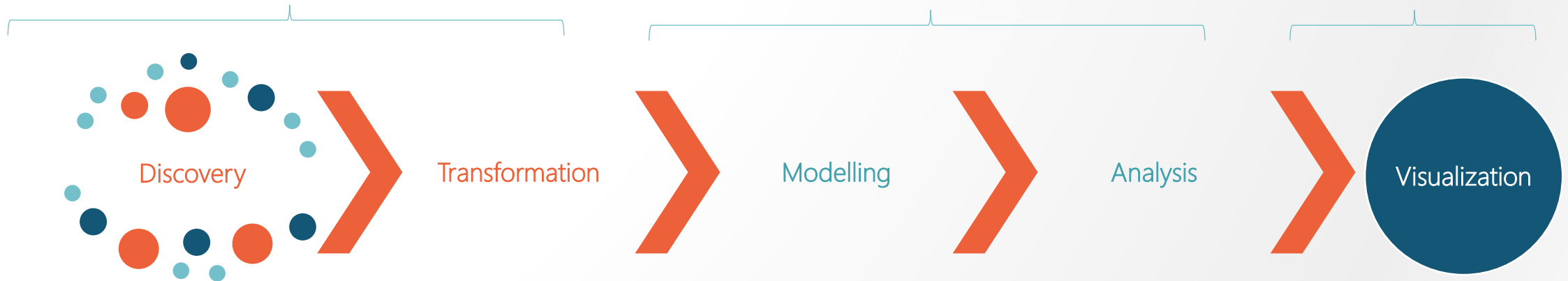
- Start Power BI Desktop
 - Per session there is one PBIX-file
 - Power BI Desktop can run multiple sessions simultaneously
 - Never without PBIX
- Save the file as **QuickRun.pbix**



Self-service-BI

Power Query

Power BI Desktop





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

The screenshot shows the Microsoft Power BI Desktop interface. The title bar indicates 'Untitled - Power BI Desktop' and the user 'Geert De Kooter'. The 'Table tools' ribbon is active, showing options for 'Name', 'Format', 'Data type', 'Data category', 'Sort by column', 'Data groups', 'Manage relationships', and 'New column'. The 'Data type' dropdown is set to 'Decimal number'. The 'Data category' dropdown is set to 'Uncategorized'. The 'Format' dropdown is set to 'General'. The 'Data type' dropdown is highlighted with an orange callout box labeled 'Data type'. The 'Format' dropdown is highlighted with an orange callout box labeled 'Format'. The 'Data' view is selected, showing a table with columns: Quantity, PricePerUnit, CurrencyCode, BtwTvaVat, Orderdate, Customerkey, Salestype, Status, SalesRep, and SendDate. The 'Data' view is highlighted with a teal callout box labeled 'Data'. The 'Fields' pane on the right shows the 'Orders' table and the 'BtwTvaVat' column.

Quantity	PricePerUnit	CurrencyCode	BtwTvaVat	Orderdate	Customerkey	Salestype	Status	SalesRep	SendDate
1	13680,56	EUR	0,21	Wednesday, 22 October 2014	C00091	Web	Cancelled	E0004	
1	10557,93	EUR	0,21	Friday, 2 January 2015	C00186	Web	Cancelled	E0004	
1	4103,31	EUR	0,21	Wednesday, 11 March 2015	C00186	Visit	Cancelled	E0008	
1	16971,67	EUR	0,21	Friday, 13 March 2015	C00062	Call	Cancelled	E0028	
1	19139,29	EUR	0,21	Friday, 20 March 2015	C00025	Web	Cancelled	E0028	

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

The screenshot shows the Power BI Desktop interface with the Power Query Editor open. The main window is titled "Untitled - Power BI Desktop" and the user is "Geert De Kooter". The "Table tools" ribbon is active, showing options like "Transform data", "New table", "Manage roles", "View as", "Sensitivity", and "Publish". A blue callout bubble labeled "Transform data" points to the "Transform data" button.

The Power Query Editor window is titled "Untitled - Power Query Editor" and has a ribbon with "File", "Home", "Transform", "Add Column", "View", "Tools", and "Help". The "Transform" ribbon is active, showing options like "Close & Apply", "Close", "Refresh", "Preview", "Query", "Manage Columns", "Reduce Rows", "Sort", "Transform", "Data Type: Text", "Use First Row as Headers", and "Replace Values". A red callout bubble labeled "Use First row as Headers" points to the "Use First Row as Headers" button.

The "Queries" pane on the left shows a list of queries: "ProductKey", "Orders", "Category", "Products", and "SubCategory". A blue callout bubble labeled "Queries" points to this pane.

The "Close & Apply" button is highlighted with a red callout bubble labeled "Close & Apply".

The formula bar shows the query formula: `= Table.TransformColumnTypes(Category_Sheet,{{"Column1", type text}, {"Column2", type text}})`.

The data preview shows a table with two columns: "Column1" and "Column2". The data is as follows:

Column1	Column2
Category	Category
Bike	Bike
Car	Car
Motor	Motor
Van	Van

The "Query Settings" pane on the right shows the "PROPERTIES" section with the "Name" set to "Category". The "APPLIED STEPS" section shows the steps: "Source", "Navigation", and "Changed Type".

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 Datasource Recent sources

Clipboard Data

Transform data Refresh data

Queries

Manage relationships

Relationships

Manage roles View as

Security

Q&A setup Language Linguistic schema

Q&A

Sensitivity

Sensitivity

Publish

Share

Model

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

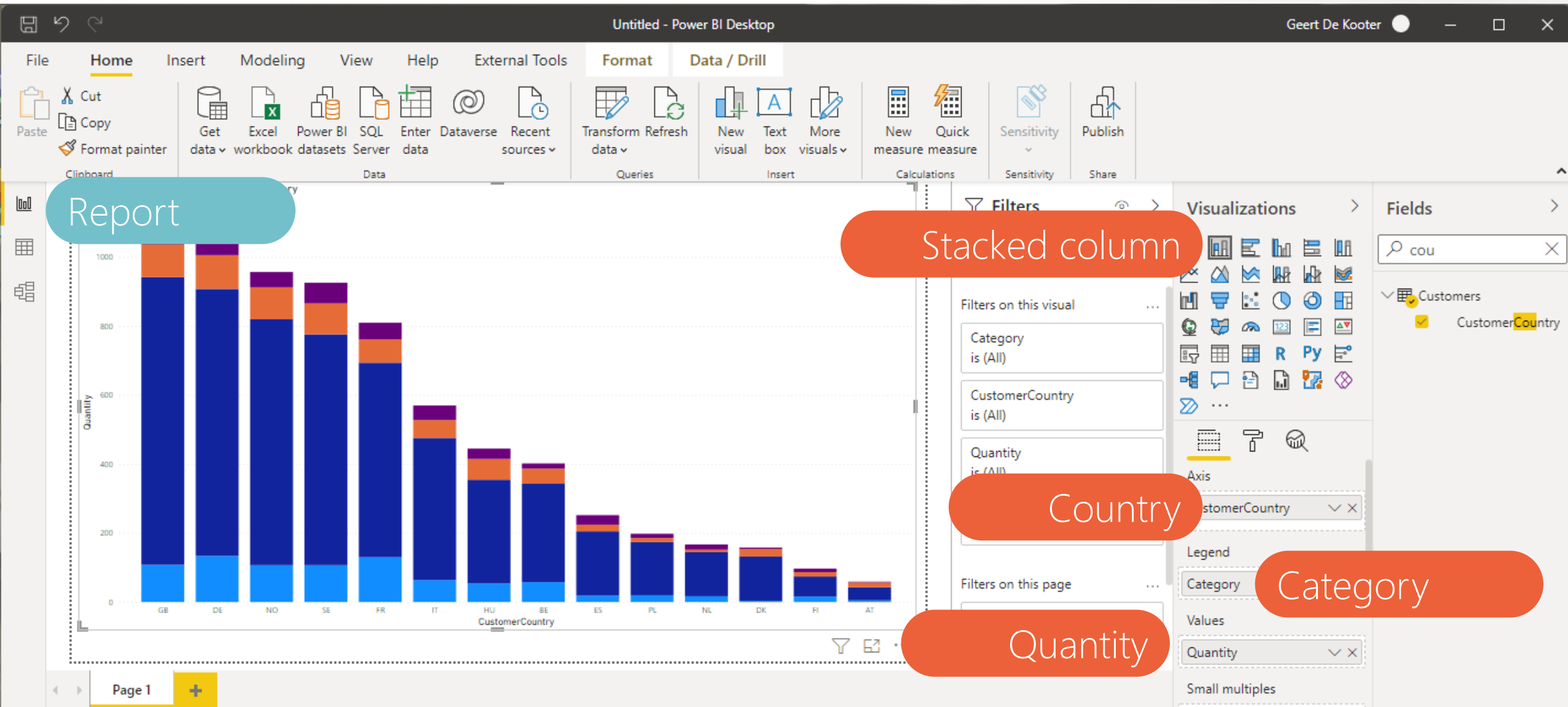
Fields

Search

- Category
- Orders
- Products
- SubCategory

Power BI will try to create the relations

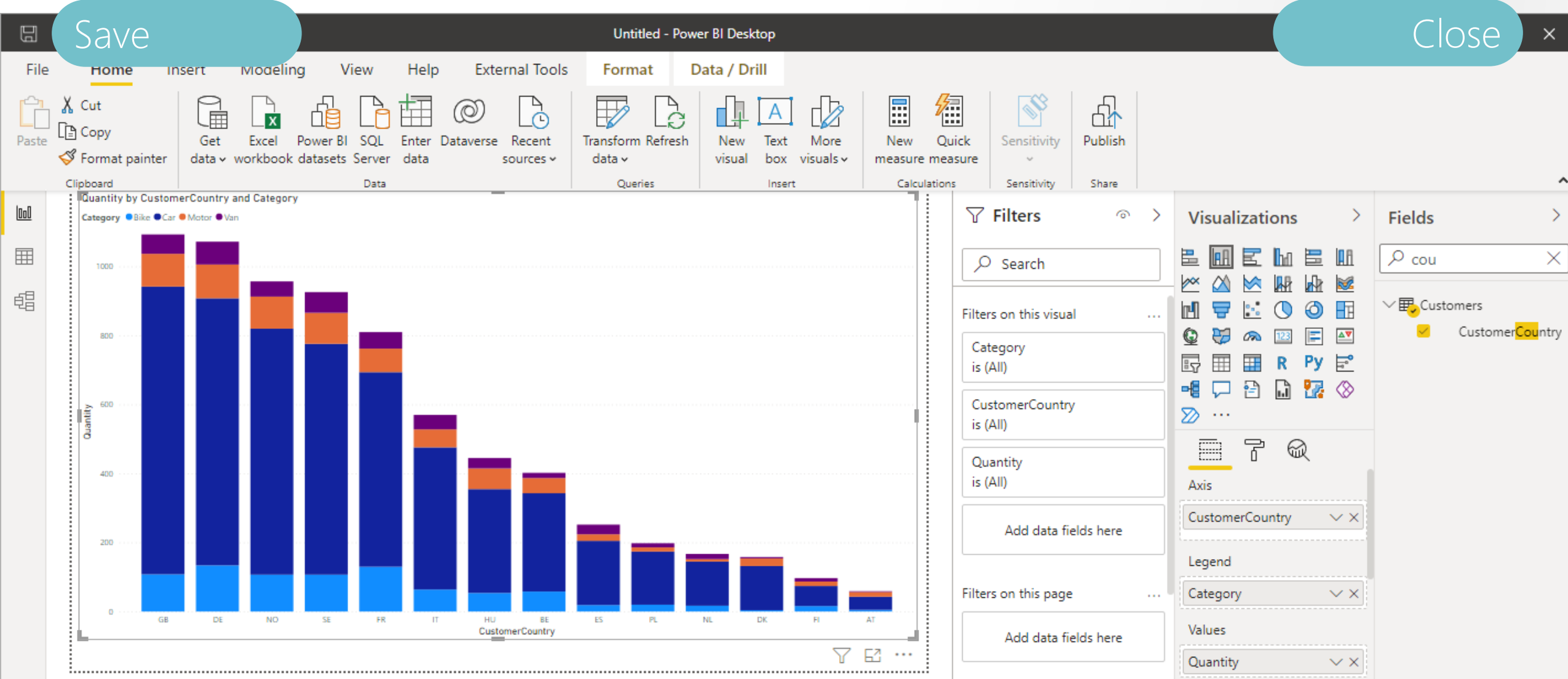
Create a chart



Save and close the file

Save

Close





Power Query - Intro

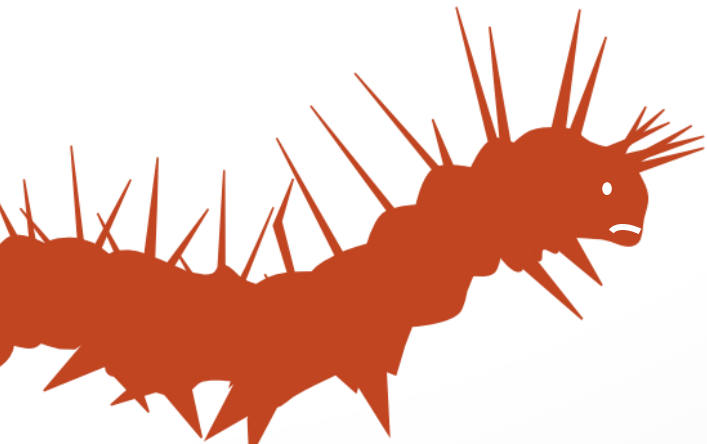
Power Query Data sources

- 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

The screenshot shows the Power BI Desktop interface. The title bar indicates the file is 'Example_2019 - Power BI Desktop'. The ribbon is set to the 'Home' tab, with the 'View' and 'Modeling' sub-tabs visible. The 'External data' group on the ribbon contains five buttons: 'Get Data', 'Recent Sources', 'Enter Data', 'Edit Queries', and 'Refresh'. These buttons are highlighted with orange callouts: 'Get Data' is in a rounded rectangle below the button, 'Edit Queries' is in a rounded rectangle below the button, and 'Refresh' is in a rounded rectangle above the button. The 'Visualizations' pane on the right shows various chart types and a search bar. The 'Fields' pane on the right shows a list of data fields under the 'Analysis' category, including 'Actual Custo...', 'Average Days...', 'Call Sales', 'Cancelled Deals', 'Compare Sale...', 'Grand total o...', 'Internet Sales', 'Sales Expected', 'Sales late Deli...', 'Share Sales o...', 'Target - Parall...', 'Total Invoiced', and 'Total Sales'.

Example_2019 - Power BI Desktop

File Home View Modeling Help

Clipboard External data Insert Custom visuals Themes Relationships Calculations Share

Get Data Recent Sources Enter Data Edit Queries Refresh

External data

Refresh

Get Data Edit Queries

Visualizations Fields

Search

Analysis

- Actual Custo...
- Actual Custo...
- Average Days ...
- Call Sales
- Call Sales Share
- Cancelled Deals
- Compare Sale...
- Grand total o...
- Grand total o...
- Internet Sales
- Sales Expected
- Sales late Deli...
- Share Sales o...
- Target - Parall...
- Total Invoiced
- Total Sales

Get Data

The screenshot shows the Microsoft Power BI Desktop interface. The ribbon at the top includes tabs for 'Bestand', 'Home', and 'View'. The 'Get Data' tab is selected, and a 'Get Data' dialog box is open. The dialog box has a search bar and a list of data sources. The 'Database' category is selected, showing a list of databases including 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, and Impala. The 'Visualizations' pane on the right shows various chart types, and the 'Fields' pane shows a search bar and a list of fields.

Get Data

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

Visualizations

Fields

Search

Values

Drag data fields here

FILTERS

Page level filters

Drag data fields here

Report level filters

Drag data fields here

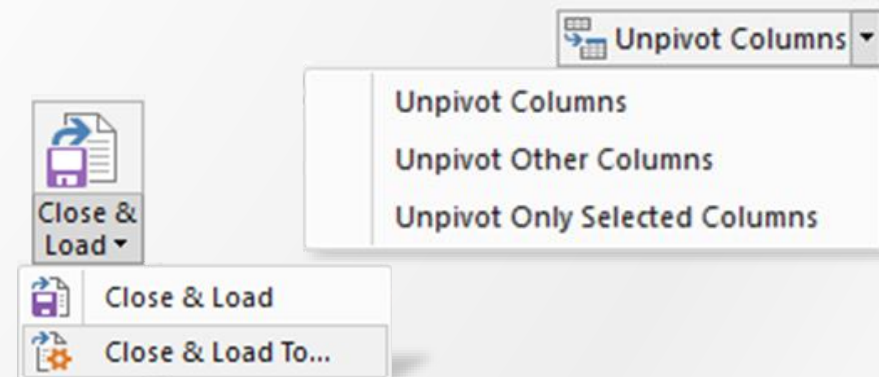
DRILLTHROUGH

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

DaysOff - Power Query Editor

File Home Transform Add Column View

Group By Use First Row as Headers Count Rows

Transpose Reverse Rows

Data Type: Text Detect Data Type Rename

Replace Values Fill Pivot Column

Unpivot Columns Move Convert to List

Unpivot Columns / Unpivot other columns

Queries

fx = Table.RemoveColumns(#"Changed Type",{"Department", "Name"})

	EmployeeID	31/01/2017	28/02/2017	31/03/2017	30/04/2017
1	E0001	3	null	null	
2	E0002	3	2	2	
3	E0003	3	null	null	
4	E0004	4	null	1	
5	E0005	3	null	null	
6	E0006	2	null	null	
7	E0007	4	null	3	
8	E0008	3	2	1	
9	E0009	3	null	2	
10	E0010	3	null	null	
11	E0011	4	2	null	
12	E0012	3	3	3	
13	E0013	3	3	null	
14	E0014	4	null	2	
15	E0015	3	null	2	
16	E0016	2	2	2	
17	E0017	1	1	3	
18	E0018	3	3	null	

DaysOff - Power Query Editor

File Home Transform Add Column View

Group By Use First Row as Headers Count Rows

Transpose Reverse Rows

Data Type: Date Detect Data Type Rename

Replace Values Unpivot Columns

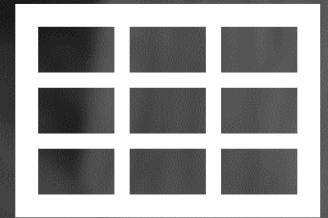
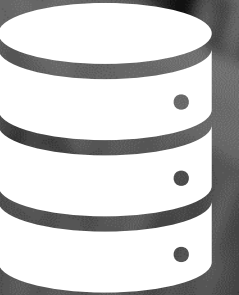
Unpivot Columns
Unpivot Other Columns
Unpivot Only Selected Columns

Split Column Format Text Column

Queries

fx = Table.TransformColumnTypes(#"Unpivoted Other Columns",{"Attribute", type date})

	EmployeeID	Attribute	Value
1	E0001	31/01/2017	3
2	E0001	30/04/2017	2
3	E0001	31/07/2017	2
4	E0001	31/08/2017	12
5	E0001	30/11/2017	6
6	E0001	31/12/2017	1
7	E0002	31/01/2017	3
8	E0002	28/02/2017	2



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

om SQL Server Database

om Microsoft Access Database

om Analysis Services

om SQL Server Analysis Services Database

om Oracle Database

om IBM Db2 Database

om MySQL Database

om PostgreSQL Database

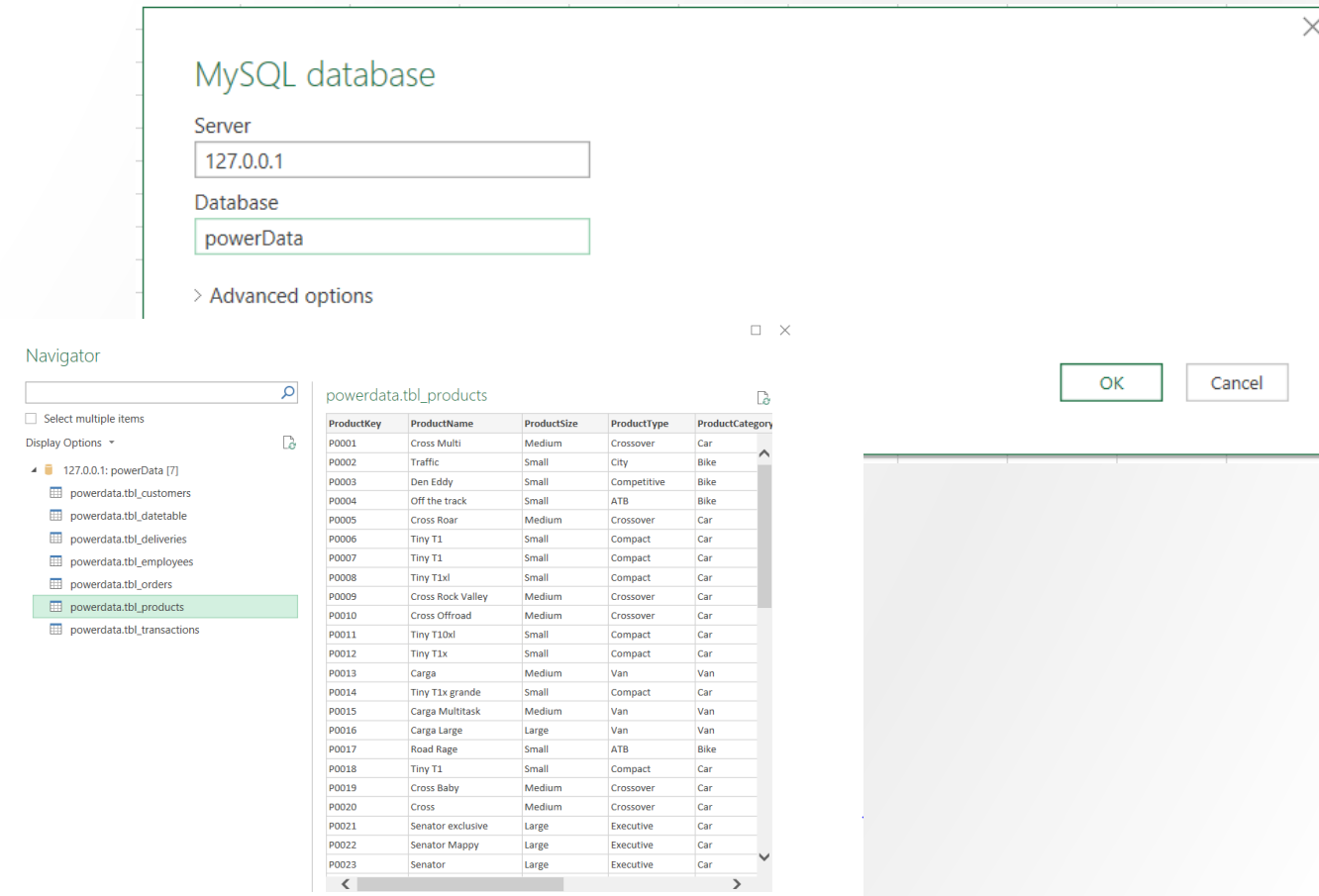
om Sybase Database

om Teradata Database

om SAP HANA Database

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. 2^e word
 2. 2^e 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. Criteria: **High >15000,**
Normal >8000 else **Low**
2. Close the query and save the file as **2_C_01.xlsx**

Query Settings

Properties

Name: Products

All Properties

Applied Steps

Source

Navigation

Changed Type

Added Custom Column

Added Conditional Column

Add rule

Add Conditional Column

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

New column name: PriceRange

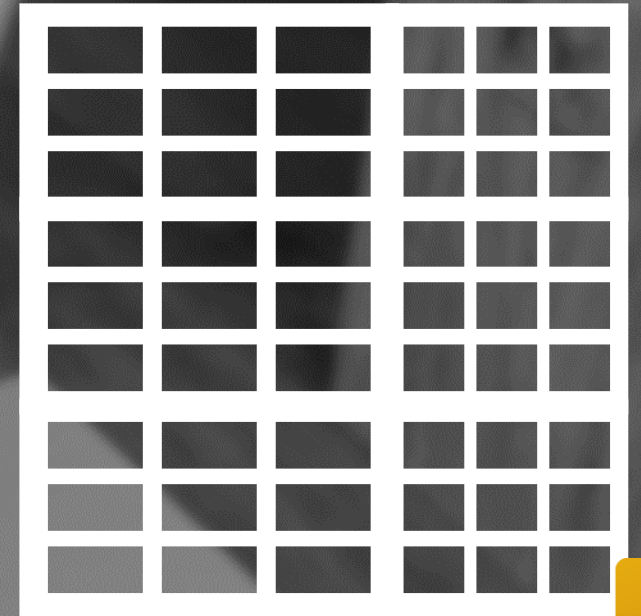
	Column Name	Operator	Value	Then	Output
If	CatalogPrice	is greater than or...	15000	Then	High
Else If	CatalogPrice	is greater than or...	8000	Then	Normal

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**

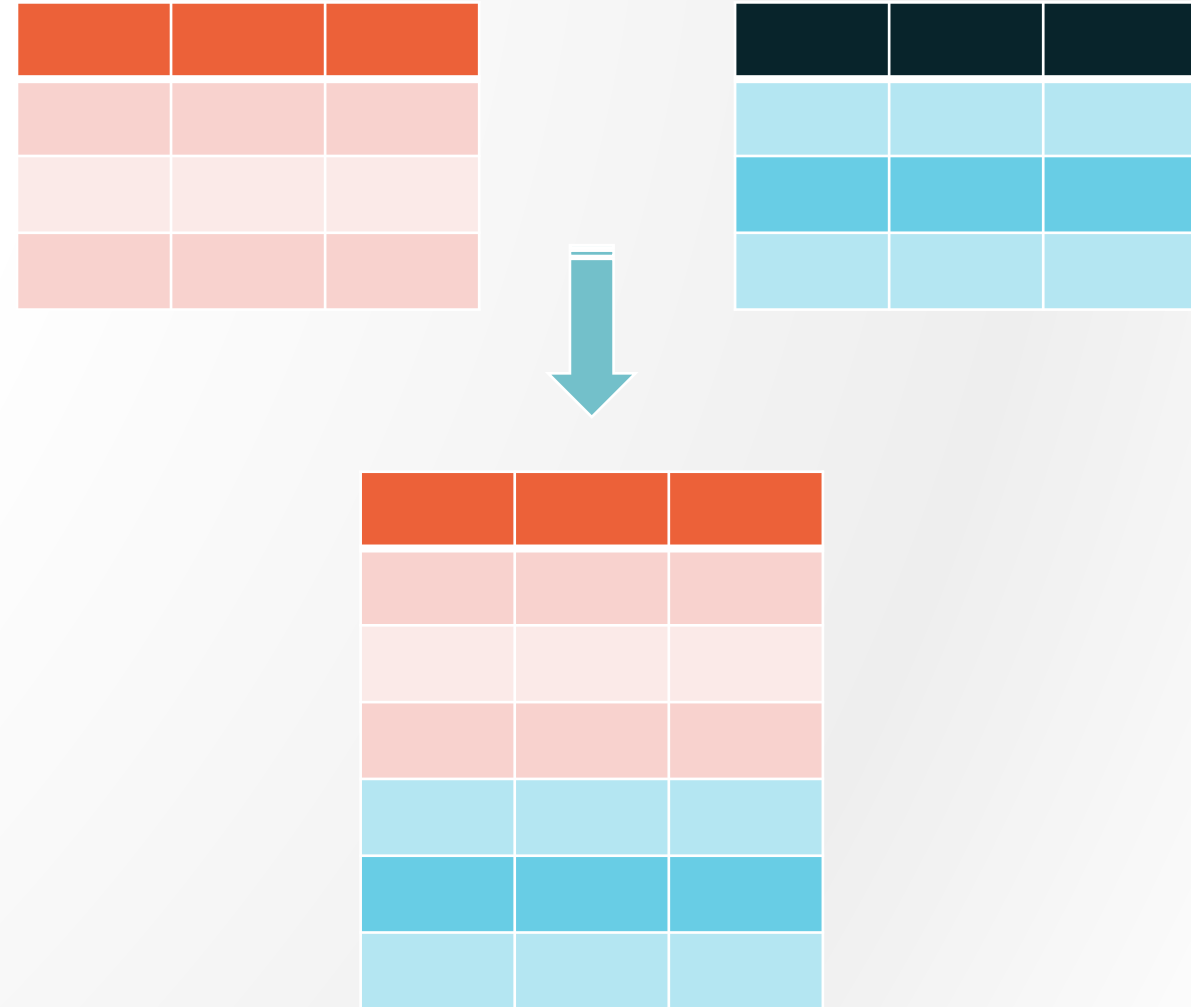


Append Tables

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**

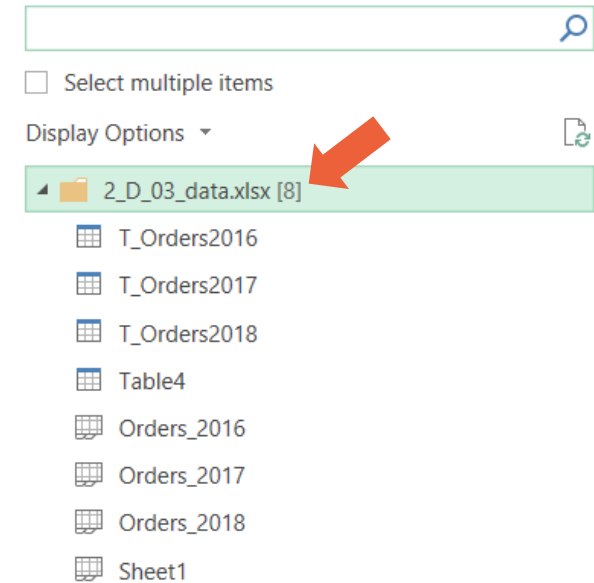
All files from a folder

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**

Navigator





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

Power BI - Power Query Tips

Turning Metadata into a column

Sometimes exports contain metadata in the top-rows. For multiple reasons we want this data to appear as a column, repeating the same values multiple times.

Before

ReportDate	22/03/2019				
Source	DB-ext-Sales				
ID	Customer	Status	Duration		
2019003	Goldendox	Closed	5		
2019004	Goldendox	Closed	4		
2019005	Indonlam	Closed	4		
2019006	An-fix	Closed	2		

After

ID	Customer	Status	Duration	ReportDate
2019003	Goldendox	Closed	5	Friday 22 March 2019
2019004	Goldendox	Closed	4	Friday 22 March 2019
2019005	Indonlam	Closed	4	Friday 22 March 2019
2019006	An-fix	Closed	2	Friday 22 March 2019

Import the Source file

1. Home > External data > Get Data > Excel
2. Select File and click **Open**
3. Select the sheet and click on **Edit / Transform data**
4. Change the name of the query, press **<Enter>** to confirm

Activate the formula bar

If the formula bar is not active, activate it

1. View > Layout > ☒ Formula Bar

Capture the metadata from the first line

1. Go to the last query-step in the Query Settings-pane, most likely "Changed Type"
2. Right-click on the value to capture and choose **Drill Down**
3. Right-click on the step Column1 and change the name to "Capture reportDate"

Restore the table

1. Click on **fx** next to your formula bar
2. Change the = #"Capture ReportDate" in to = #"Changed Type" to refer to a previous state of the query

Tip: Power Query is Case-sensitive

3. Right-click on the step Custom1 and change the name to "Restore table"

Clean up

1. Use Home > Reduce Rows > Remove Rows > Remove top rows to clean the unnecessary rows
2. Use Home > Transform > Use First Row as headers

Insert the captured value

1. Click on Add Column > Custom Column
2. Insert a name for the custom column, e.g. "ReportDate"
3. Insert the formula = #"Capture ReportDate"

You refer to the name given to step were the metadata was captured.

Apply the correct data type

Click on **ReportDate** and select the correct data type

Finish the query

Home > Close & Apply the query

© LEARNIA

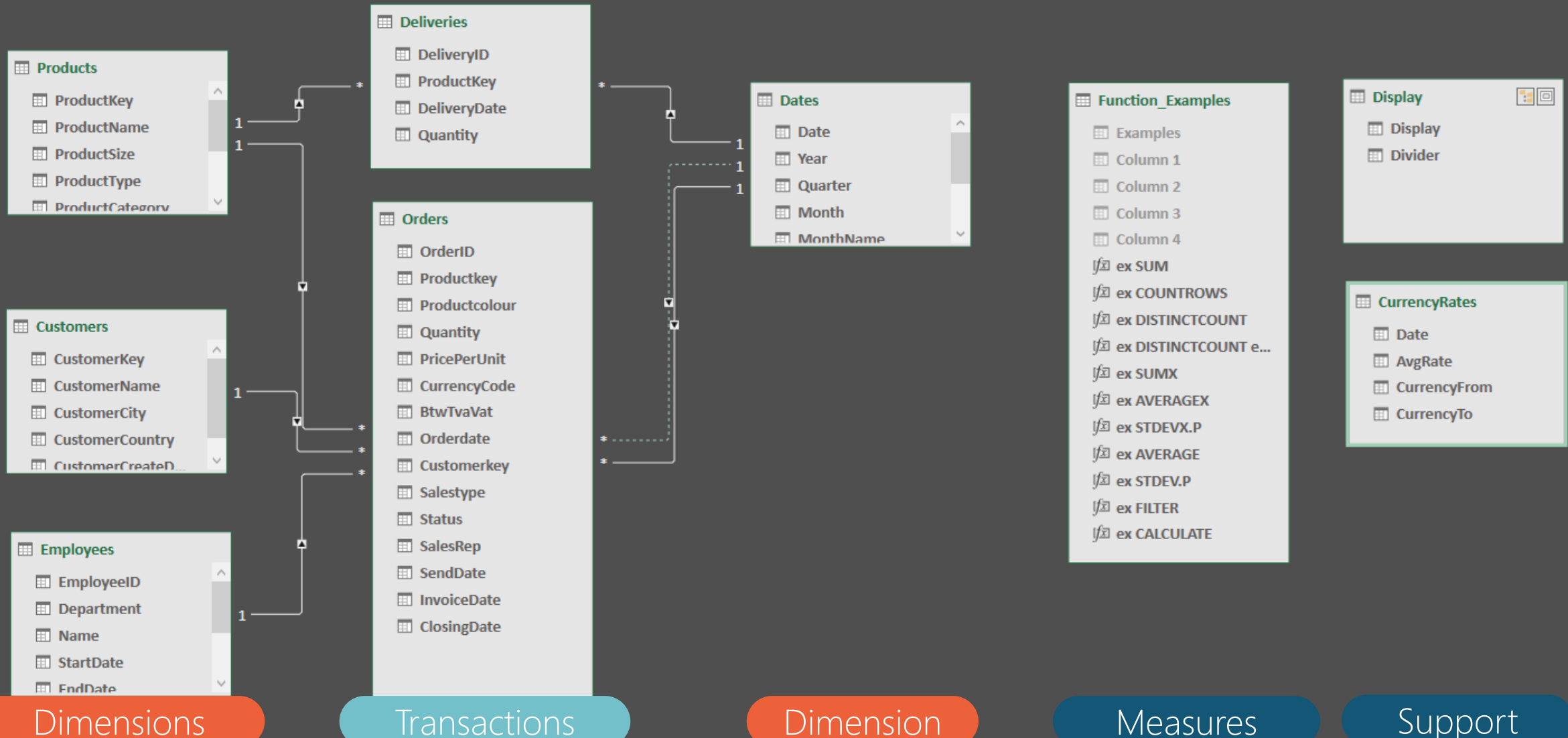
5

www.learnia.be



Part 2 : Modelling

Data Model



Create relations - Demo

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

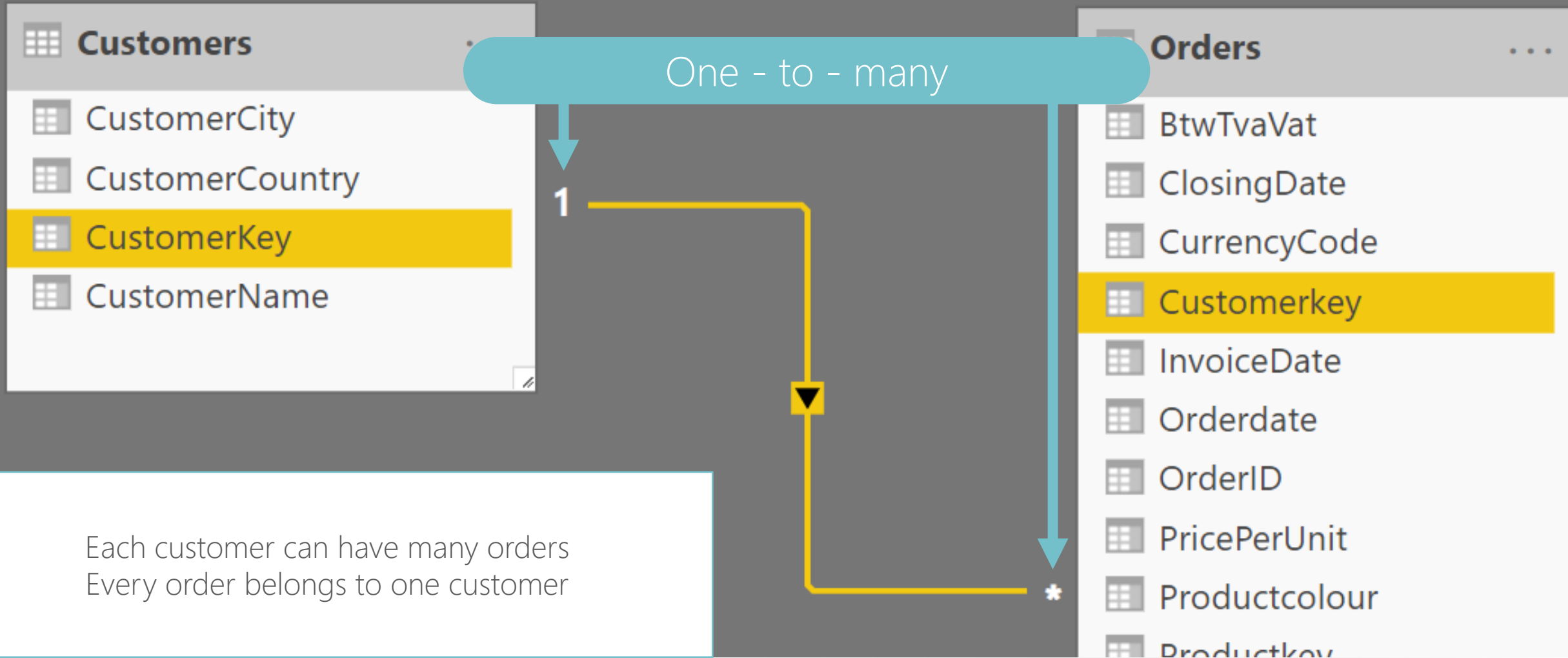
Dimension

- Customerkey
- ProductKey
- EmployeeID
- Date
- Date

Orders

CustomerKey
ProductKey
SalesRep
OrderDate
InvoiceDate

One-to-many



Multiple (active) relations

Being active is normally controlled via the function: USERRELATIONSHIP

Orders
BtwTvaVat
ClosingDate
CurrencyCode
Customerkey
InvoiceDate
Orderdate
OrderID
PricePerUnit
Productcolour

Dbl-click

Edit relationship ×

Select tables and columns that are related.

Orders

Productkey	Productcolour	Quantity	PricePerUnit	CurrencyCode	BtwTvaVat	Orderdate	CurrencyCode
6	Blue	1	13680,56	EUR	0,21	woensdag 22 oktober 2014	CO
1	Blue	1	10557,93	EUR	0,21	vrijdag 2 januari 2015	CO
3	Blue	1	4103,31	EUR	0,21	woensdag 11 maart 2015	CO

Dates

Date	Year	Quarter	Month	Month Name	mmmYYYY	mm YYYY	MonthKey
dinsdag 1 mei 2012	2012	2	5	May	MAY 2012	52012	201205
woensdag 2 mei 2012	2012	2	5	May	MAY 2012	52012	201205
donderdag 3 mei 2012	2012	2	5	May	MAY 2012	52012	201205

Cardinality

Many to one (*:1)

Cross filter direction

Single

☒ Make this relationship active

☐ Apply security filter in both directions

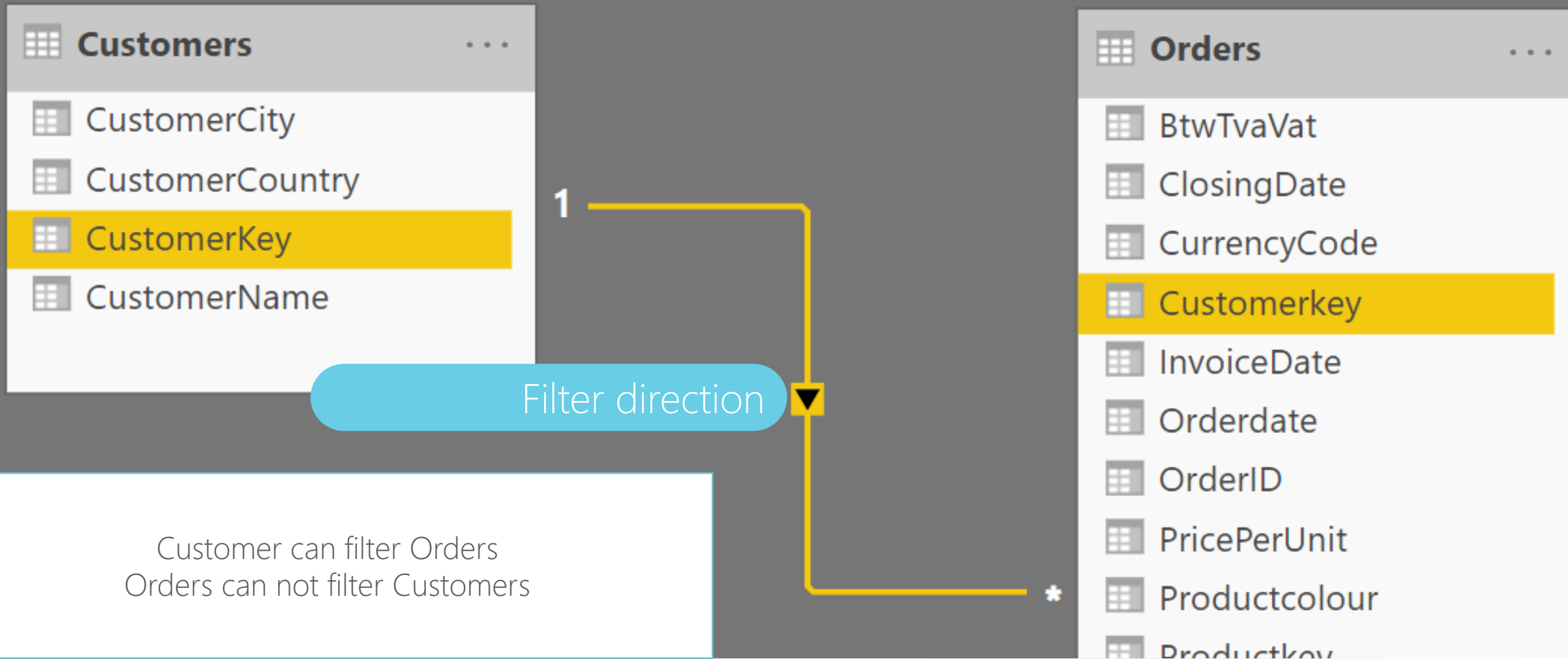
☐ Assume referential integrity

Month Name
MonthKey
Quarter
Year

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



Data
Analyses
eXpression
Language



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

3_B_01_CalculatedColumns - Power BI Desktop

File Home **Modeling** Help

Manage Relationships Relationships New Measure Calculations New Column New Table What If Sort by Column Sort Data type: Decimal Number Format: € Euro (123 €) \$ % , .00 2 Home Table: Data Category: Uncategorized Default Summarization: Sum Manage Roles Security View as Roles Groups New Group Edit Groups Mark as Date Table Calendars Language Linguistic Schema Q&A

1 Amont + Tax = Orders[Quantity] * Orders[PricePerUnit] * 1,21

OrderID	Productkey	Productcolour	Quantity	PricePerUnit	Orderdate	Customerkey	Status	Amont + Tax	Some & None	% Discount
O0000061	P0007	Violet	1	20860,19	maandag 26 maart 2012	C00007	Closed	25.240,83 €	None	7,25%
O0000098	P0018	Violet	1	9960,2	donderdag 24 mei 2012	C00006	Closed	12.051,84 €	Some	9,25%
O0000112	P0034	Violet	1	14390,85	maandag 11 juni 2012	C00072	Closed	17.412,93 €	None	11,50%

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
 $=\text{ROUNDUP}(\text{Orders}[\text{OrderMonth}] / 3 ; 0)$
- Show the line total :
Line Total
 $= \text{Orders}[\text{Quantity}] * \text{Orders}[\text{PricePerUnit}]$



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
- MMaxDiscount
=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"
)

Create 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

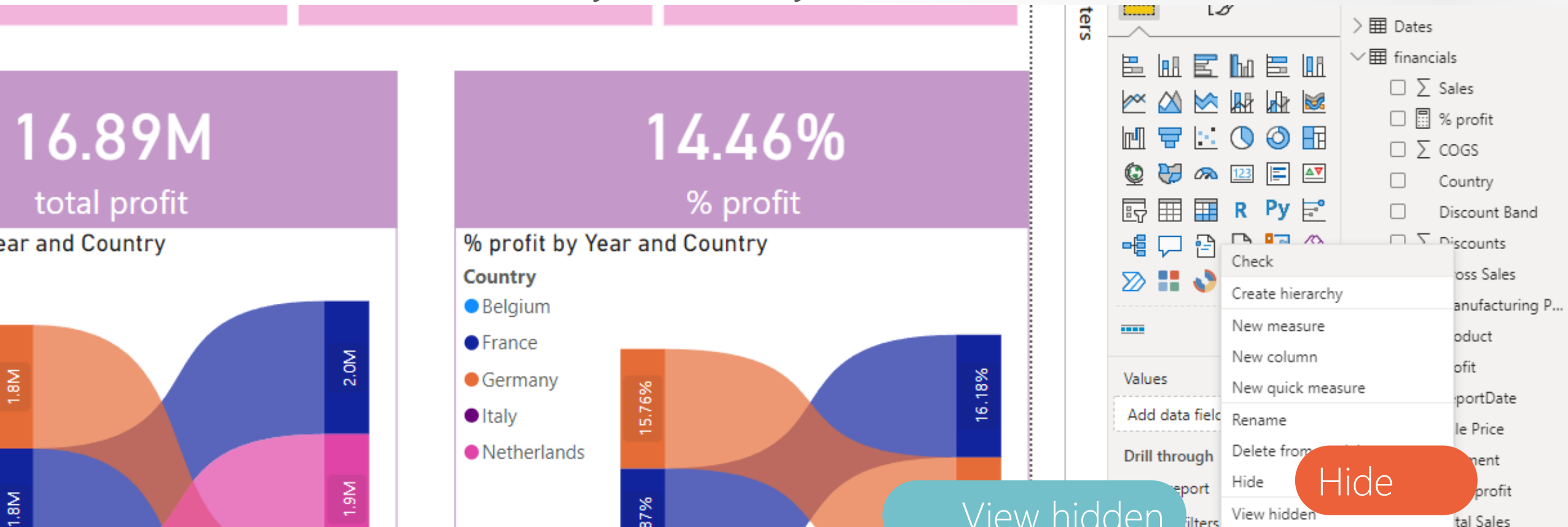




Organize your model

Organize your Model

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



Sort columns

- Column tools > Sort by column

The screenshot shows the Power BI Desktop interface with the 'Table tools' ribbon selected and the 'Column tools' tab active. The 'Month Name' column is selected in the data table. The 'Sort by column' dropdown menu is open, showing options: Date, mm YYYY, mmm, mmmYYYY, Month (highlighted), MonthKey, Quarter, and Year. An orange callout bubble points to the 'Sort by column' icon in the ribbon. Another orange callout bubble points to the 'Data' icon in the left-hand pane.

5_E_financial_Flow_results - Power BI Desktop

File Home Help External Tools Table tools Column tools

Name: Month Name Format: Text Summarization: Don't summarize Data category: Uncategorized

Structure Formatting Properties

Sort by column

Sort by column

Month Name

Date

mm YYYY

mmm

mmmYYYY

Month

MonthKey

Quarter

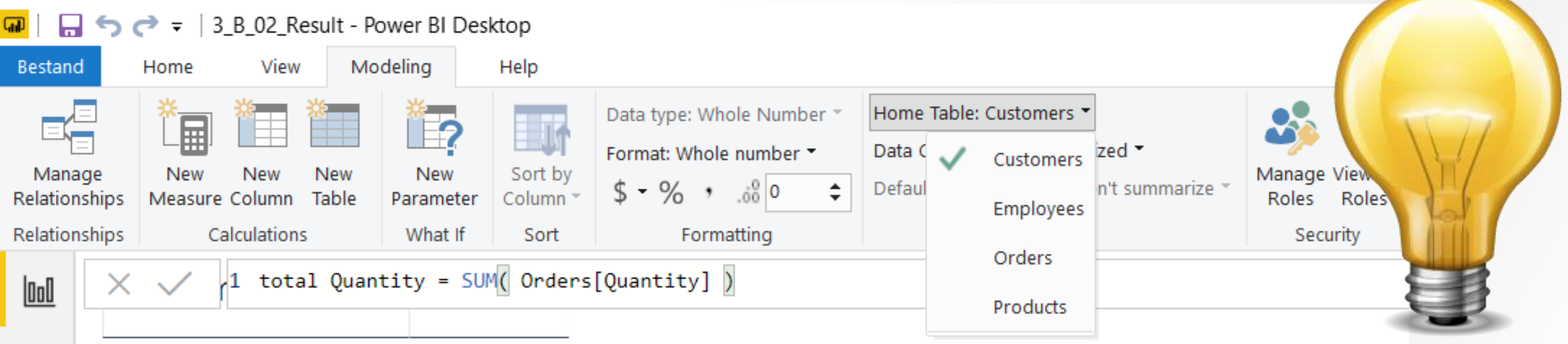
Year

Data

	Year	Quarter	Month	Month Name	mmmYYYY	mm YYYY	mmm	
	2018	2018	1	1	January	JAN 2018	01 2018	JAN
Tuesday, 2 January 2018	2018	1	1	1	January	JAN 2018	01 2018	JAN
Wednesday, 3 January 2018	2018	1	1	1	January	JAN 2018	01 2018	JAN
Thursday, 4 January 2018	2018	1	1	1	January	JAN 2018	01 2018	JAN
Friday, 5 January 2018	2018	1	1	1	January	JAN 2018	01 2018	JAN
Saturday, 6 January 2018	2018	1	1	1	January	JAN 2018	01 2018	JAN
Sunday, 7 January 2018	2018	1	1	1	January	JAN 2018	01 2018	JAN
Monday, 8 January 2018	2018	1	1	1	January	JAN 2018	01 2018	JAN
Tuesday, 9 January 2018	2018	1	1	1	January	JAN 2018	01 2018	JAN
Wednesday, 10 January 2018	2018	1	1	1	January	JAN 2018	01 2018	JAN
Thursday, 11 January 2018	2018	1	1	1	January	JAN 2018	01 2018	JAN
Friday, 12 January 2018	2018	1	1	1	January	JAN 2018	01 2018	JAN
Saturday, 13 January 2018	2018	1	1	1	January	JAN 2018	01 2018	JAN

Where to create a measure

- Measures can be stored in any table of your data model
- In measures can easily be moved from one 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 lose 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

5_E_financial_Flow_results - Power BI Desktop

Search

File Home Help External Tools Table tools Column tools

Clipboard: Paste, Cut, Copy

Data: Get data, Excel workbook, Data hub, SQL Server, Enter data, Dataverse, Recent sources

Queries: Transform data, Refresh data

Relationships: Manage relationships

Calculations: New measure, Quick measure, New column, New table

Security: Manage roles, View as

Sensitivity: Sensitivity

Share: Publish

Date	Year	Quarter	Month	Month Name	mmmY
Monday, 1 January 2018	2018	1	1	January	JAN 20
Tuesday, 2 January 2018	2018	1	1	January	JAN 20
Wednesday, 3 January 2018	2018	1	1	January	JAN 20
Thursday, 4 January 2018	2018	1	1	January	JAN 20
Friday, 5 January 2018	2018	1	1	January	JAN 20
Saturday, 6 January 2018	2018	1	1	January	JAN 20
Sunday, 7 January 2018	2018	1	1	January	JAN 20
Monday, 8 January 2018	2018	1	1	January	JAN 20
Tuesday, 9 January 2018	2018	1	1	January	JAN 20
Wednesday, 10 January 2018	2018	1	1	January	JAN 20
Thursday, 11 January 2018	2018	1	1	January	JAN 20
Friday, 12 January 2018	2018	1	1	January	JAN 20
Saturday, 13 January 2018	2018	1	1	January	JAN 20
Sunday, 14 January 2018	2018	1	1	January	JAN 20

Create Table

	Column1	+
1		
+		

Name:

Load Edit Cancel



Part 3 : Analysis



Basic aggregations

Create a measure

Power BI Desktop interface showing the process of creating a new measure.

Top Screenshot: The 'Table tools' ribbon is active. The 'Name' field is set to 'Orders'. The 'New measure measure column table' button is highlighted with an orange callout that says 'New measure'.

Bottom Screenshot: The 'Measure tools' ribbon is active. The 'Name' field is set to 'Measure'. The 'Format' dropdown is set to '\$%'. The 'Data category' dropdown is set to 'Uncategorized'. The 'New measure measure' button is highlighted.

Formula Bar: The formula entered is: `1 Total Quantity = SUM(Orders[Quantity])`

Data Table:

OrderID	Productkey	Productcolour	Quantity	PricePerUnit	BtwTvaVat	Customerkey	Salestype	Status	SalesRep	ClosingDate	Customer
00000025	P0012	Green	1	10840,41	0,21	C00008	Call	Closed	E0004	Monday, 30 January 2012	Zumr
00000031	P0003	Green	1	4249,13	0,21	C00049	Call	Closed	E0004	Wednesday, 22 February 2012	Lamfr
00000035	P0001	Green	1	18550,12	0,21	C00035	Call	Closed	E0004	Wednesday, 22 February 2012	Sun-a
00000045	P0042	Green	1	15460,74	0,21	C00049	Phone	Closed	E0004	Saturday, 3 March 2012	Lamfr
00000057	P0048	Green	1	4280,62	0,21	C00036	Visit	Closed	E0004	Monday, 26 March 2012	Strin

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 to calculate **Percentages**

Arguments

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

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
	1,98	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 creates a table with unique rows

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 % :=
$$\text{Total Deals Net} / \text{Total All Deals Net}$$
- Nothing to do ...
 - Rewrite **Deals Net %** without using the **Total All Deals Net**
Tip: use variables



ALLSELECTED

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

2	Row Labels	Total Deals Net	Total All Deals Net	Total ALLSELECTED Deals Net
3	AT	71.422	41.096.949	4.767.847
4	BE	367.110	41.096.949	4.767.847
5	DE	553.136	41.096.949	4.767.847
6	DK	237.944	41.096.949	4.767.847
7	ES	114.073	41.096.949	4.767.847
8	FI	129.975	41.096.949	4.767.847
9	FR	238.993	41.096.949	4.767.847
10	GB	877.558	41.096.949	4.767.847
11	HU	360.177	41.096.949	4.767.847
12	IT	540.587	41.096.949	4.767.847
13	NL	37.340	41.096.949	4.767.847
14	NO	571.904	41.096.949	4.767.847
15	PL	83.853	41.096.949	4.767.847

Row Labels	Total Deals Net
Bike	1.305.134
Car	31.341.483
Motor	4.767.847
Grand Total	41.096.949

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**

$$= \text{DIVIDE}(\text{[Total Sales Walk-in]}; \text{[Total Sales]})$$

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]) ;

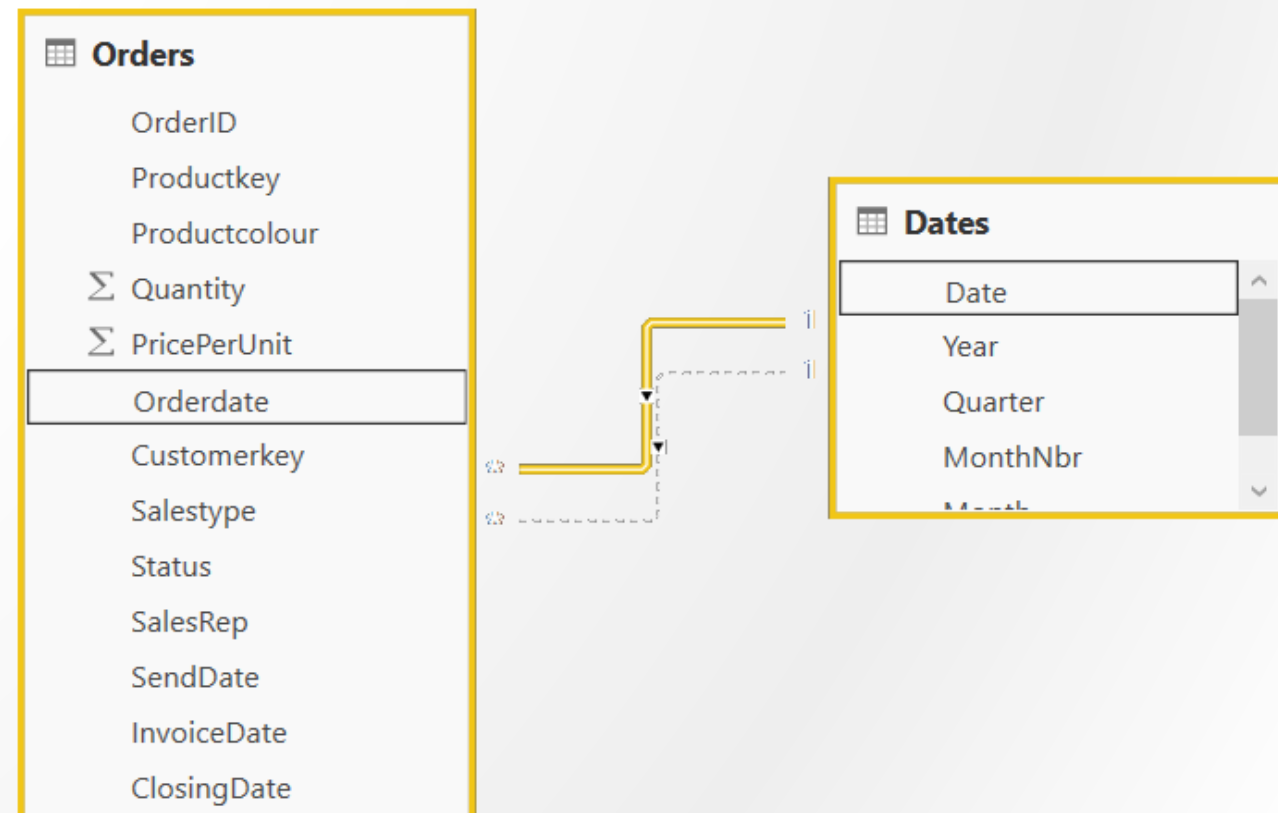
BLANK()

)

	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 activates 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



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])

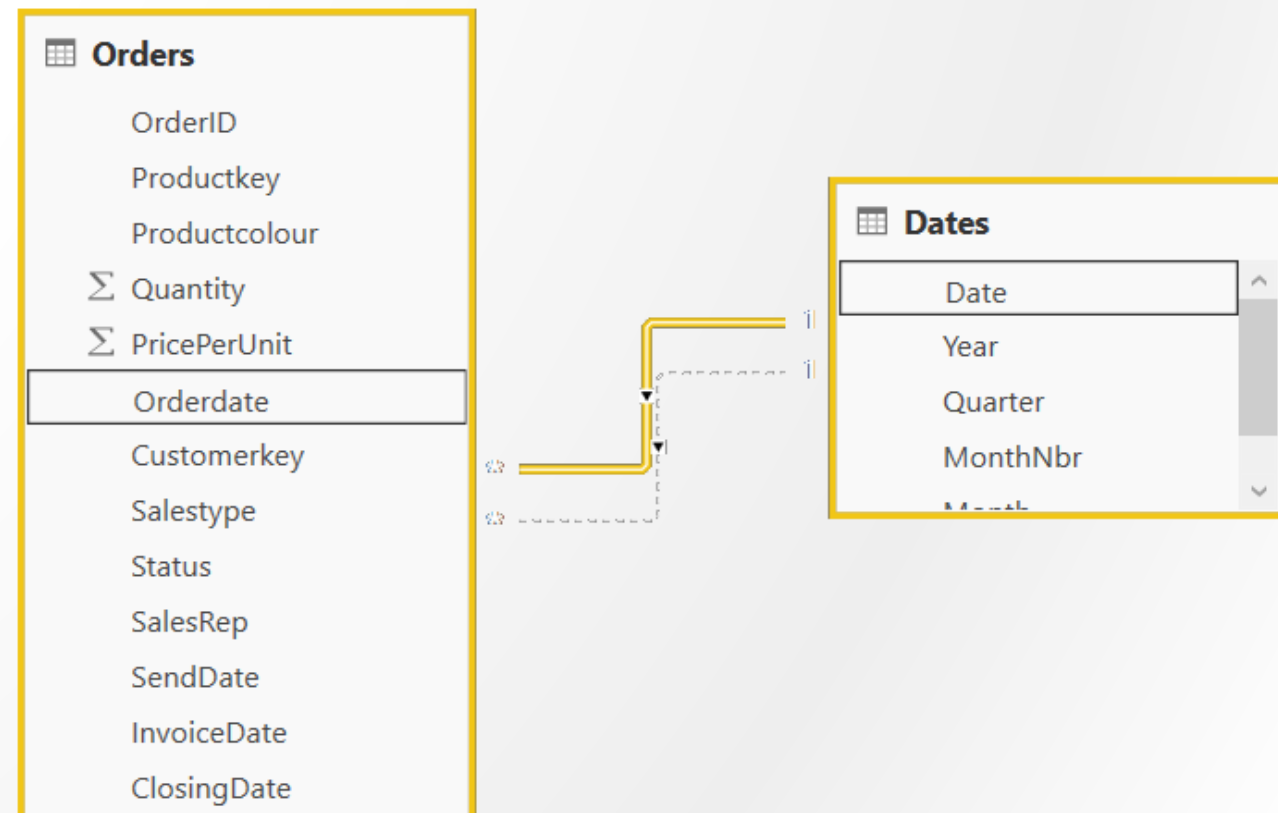


ALLEXCEPT()

- 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 activates 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...

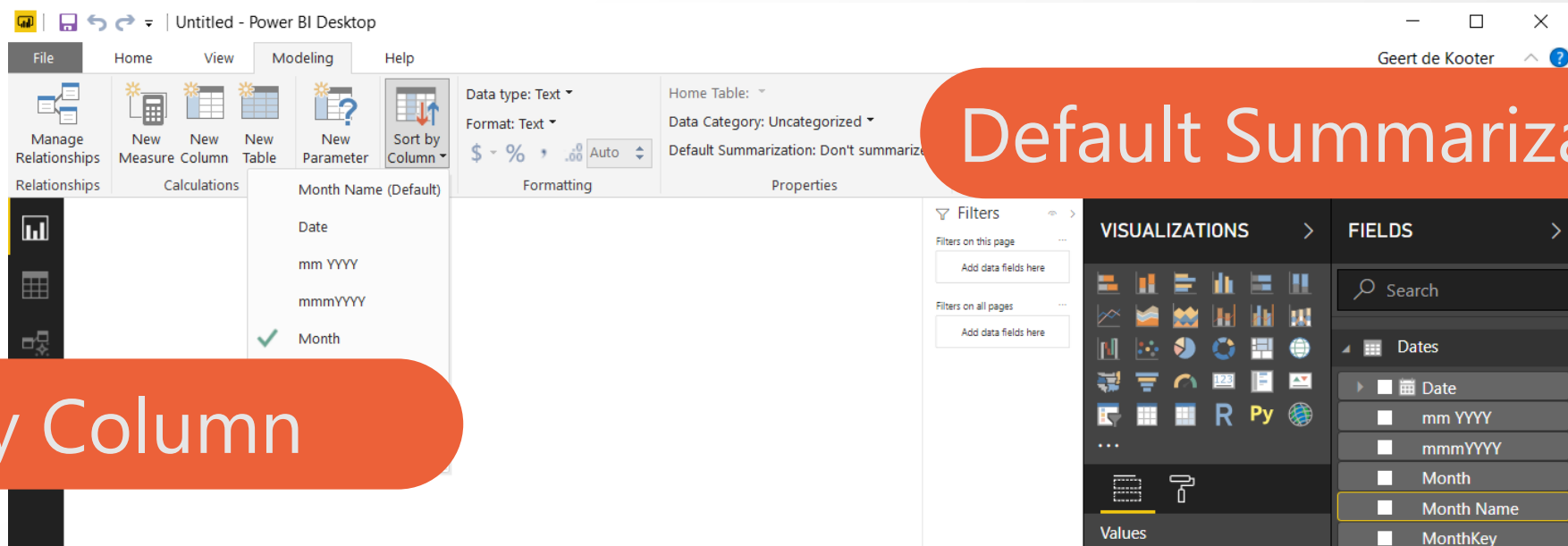
	DateActual	DateKey	YearKey	HalfYearKey	QuarterKey	MonthKey	MonthName	MaandName	MoisName	DayOfWeek	DayName
7	6/01/2010	20100106	2010	201001	201001	201001	January	Januari	janvier	3	Wednesday
8	7/01/2010	20100107	2010	201001	201001	201001	January	Januari	janvier	4	Thursday
9	8/01/2010	20100108	2010	201001	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	ianvier	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

Year	Month Name	Total Sales	Total Sales YtD
2012	January	1,115,147.60	1,115,147.60
2013	February	1,316,994.12	2,432,141.72
2014	March	1,127,663.42	3,559,805.14
2015	April	1,145,678.66	4,705,483.80
2016	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()

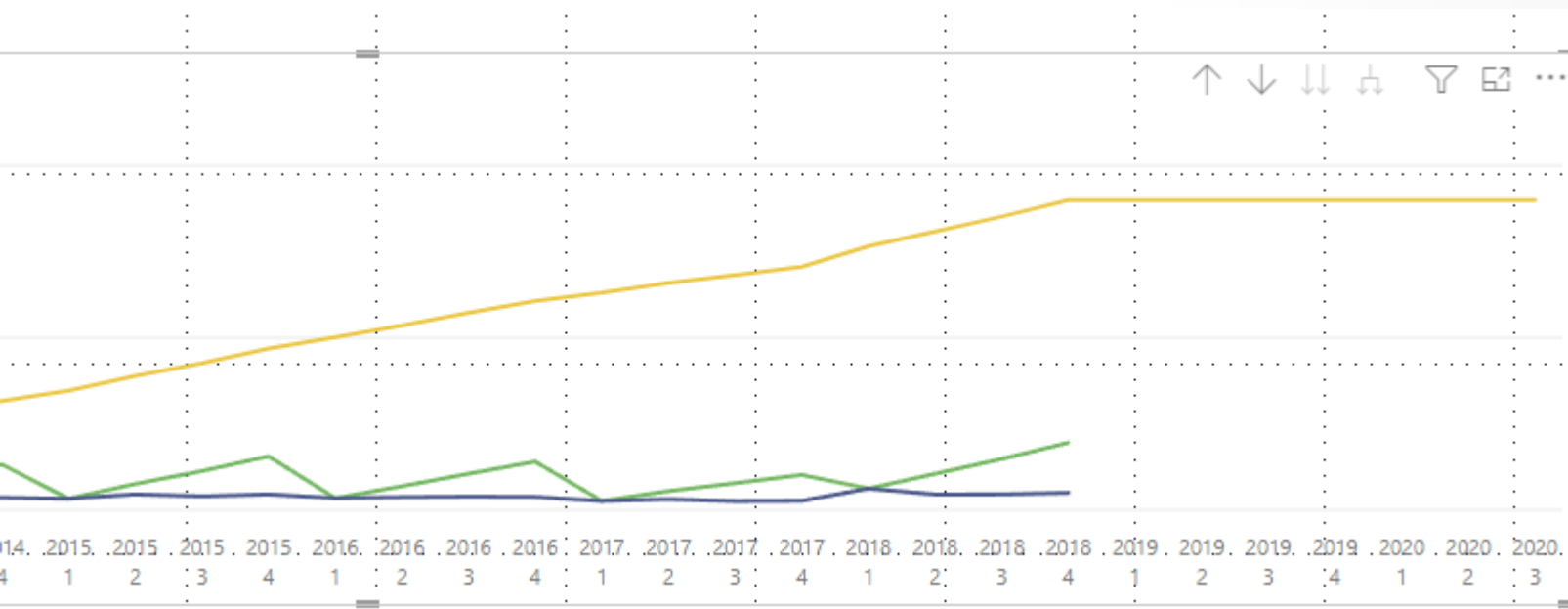
- Creates a table containing all dates between the two given dates
- Arguments
 - DATESBETWEEN(Dates[Date] ; Startdate ; Enddate)
- So Total Sales YTD could therefore be written as:
= CALCULATE (
 [Total Sales] ;
 DATESBETWEEN(Dates[Date];
 DATE (YEAR (MAX (Dates[Date])) ; 1; 1) ;
 MAX(Dates[Date])
)
)

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:
 $\text{DateWithSales} = \text{Dates}[\text{Date}] \leq \text{MAX} (\text{Orders}[\text{Orderdate}])$
- 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 = $\text{SUM}(\text{Products}[\text{ProductStockStart}])$
- Create a measure to calculate the current level of the stock
 - Total Stock = $[\text{Qty Stock Start}] + [\text{Total Delivered To Date}] - [\text{Total Quantity To Date}]$
- 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
- ☒ 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

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

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

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

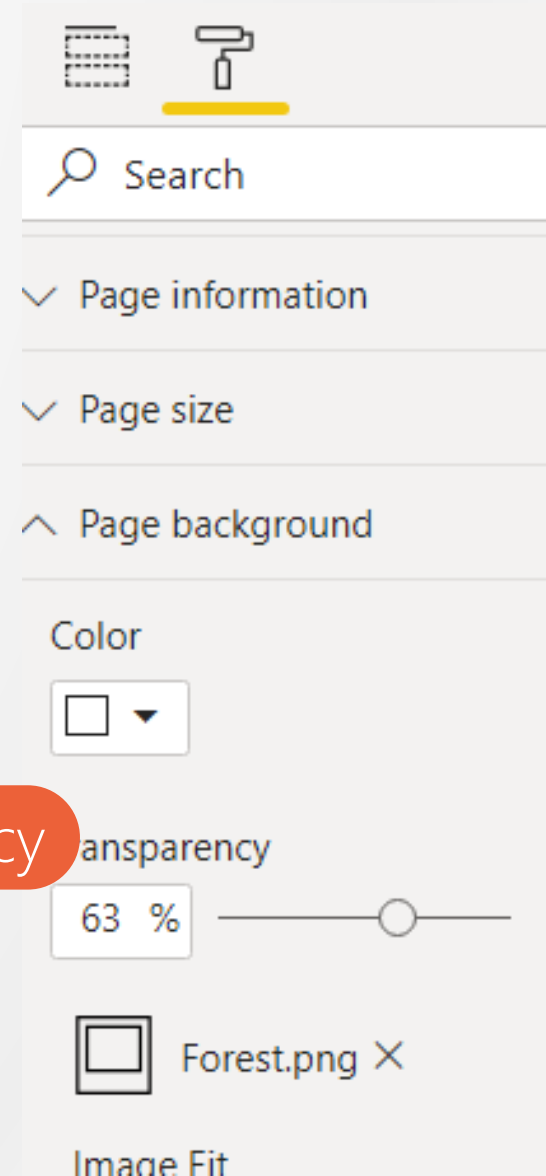
- Apply a background image or color
- Click on the page and choose 

Alignment of visuals

- Use gridlines
 - **View / Show Gridlines**

Or

- Apply a background image with the necessary canvas



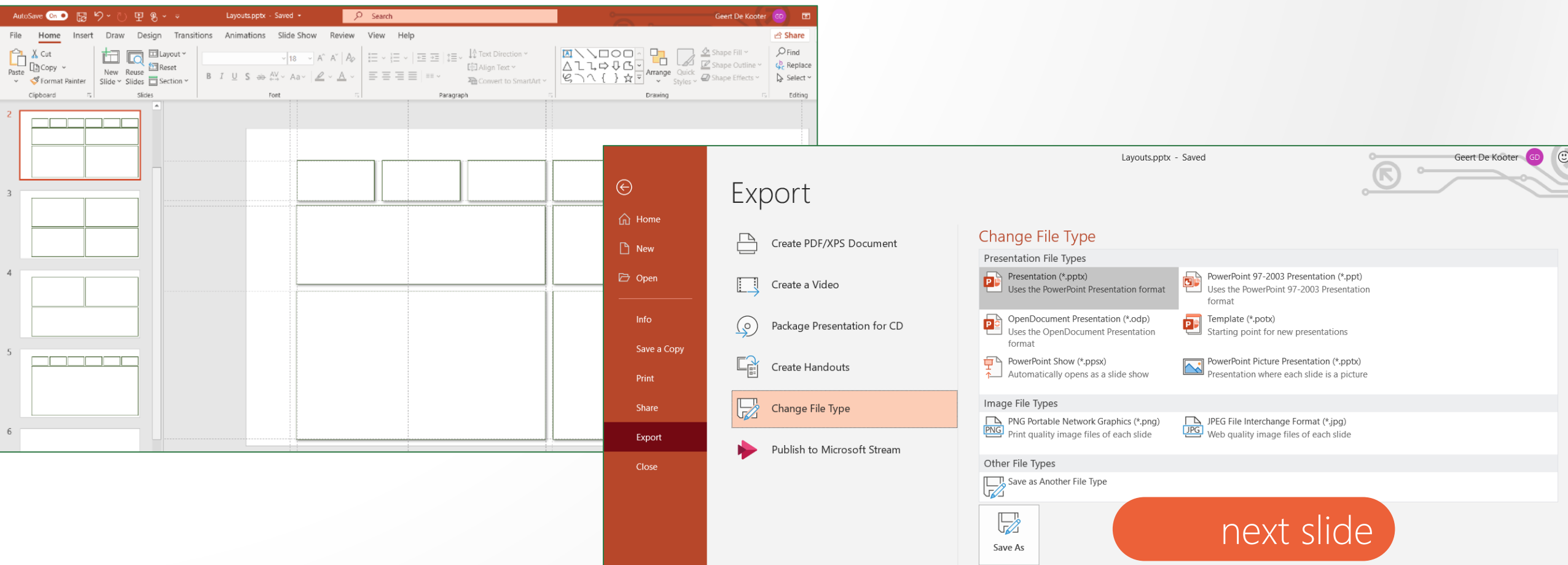
Transparency

63 %

Next slide

Create a Background canvas

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



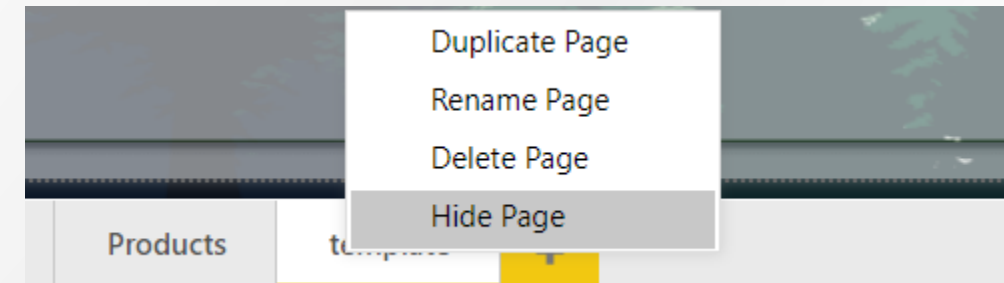
Apply a Background canvas

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

Tip

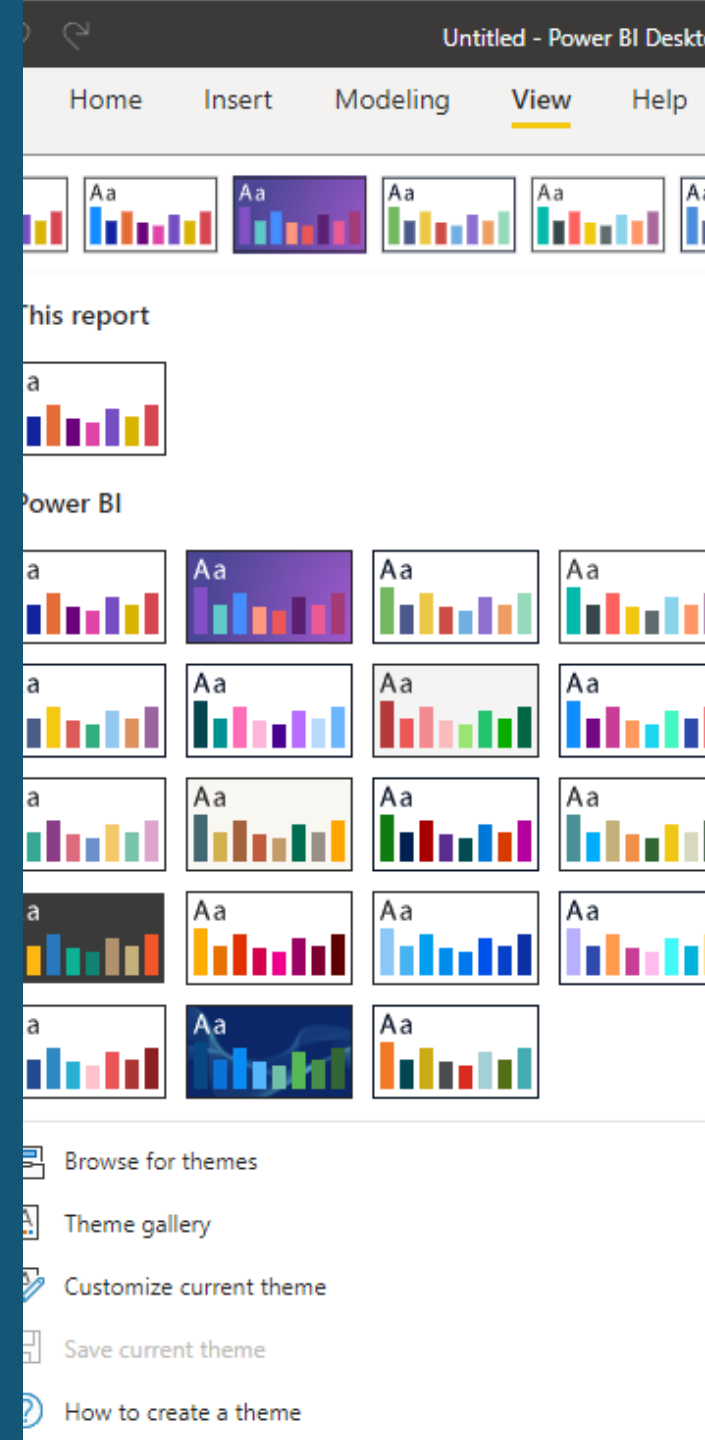
Canvas and background are applied to the current page

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



Themes

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



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



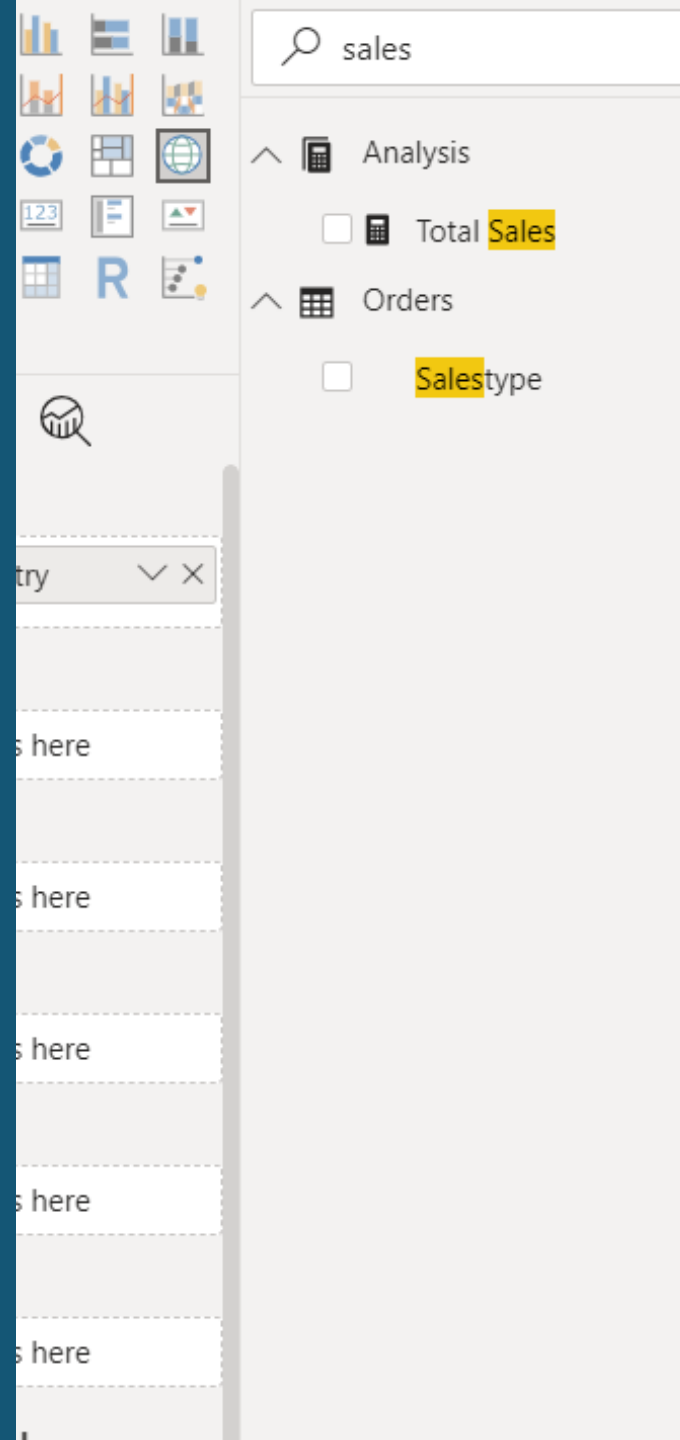


Add visuals

Adding a visual

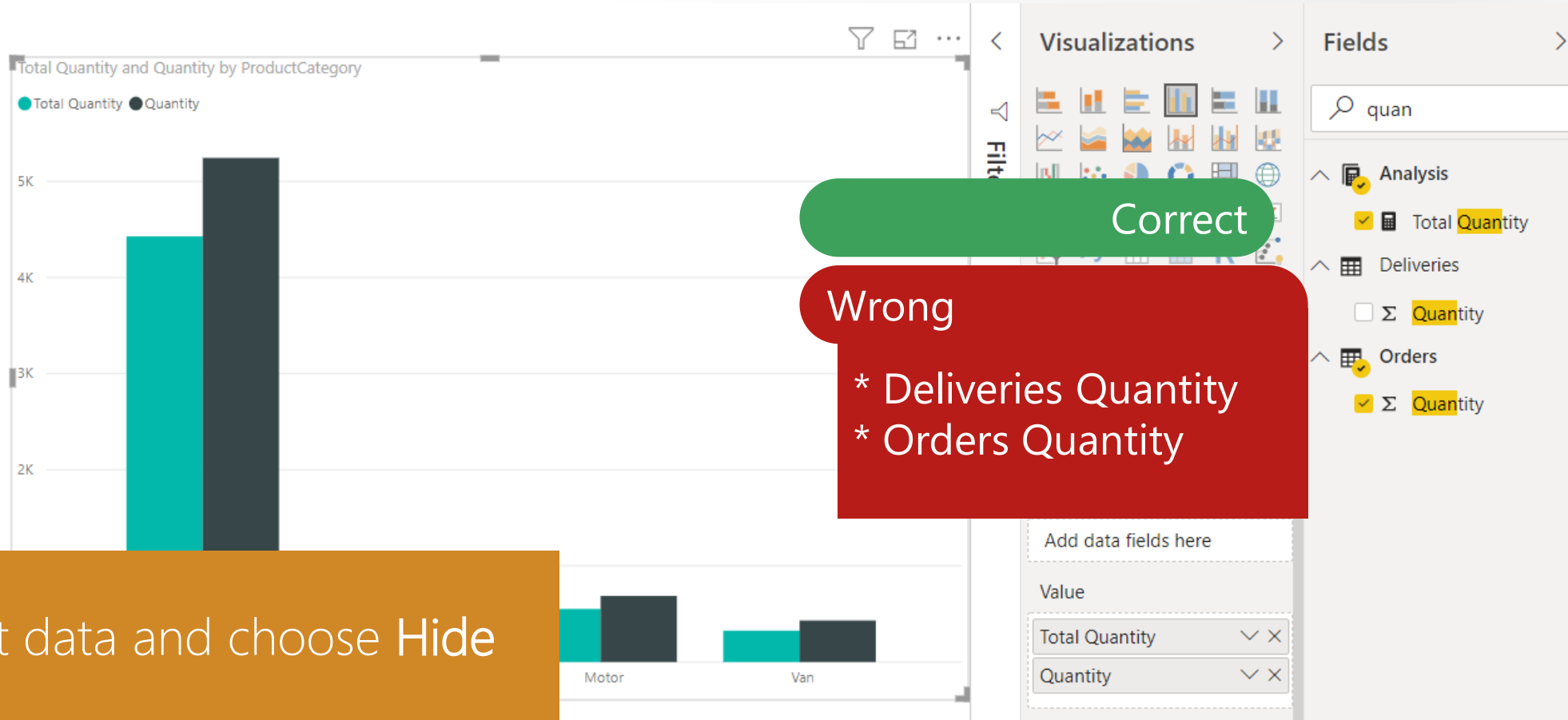
Visual – Data – Format – Filter

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



Tip

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



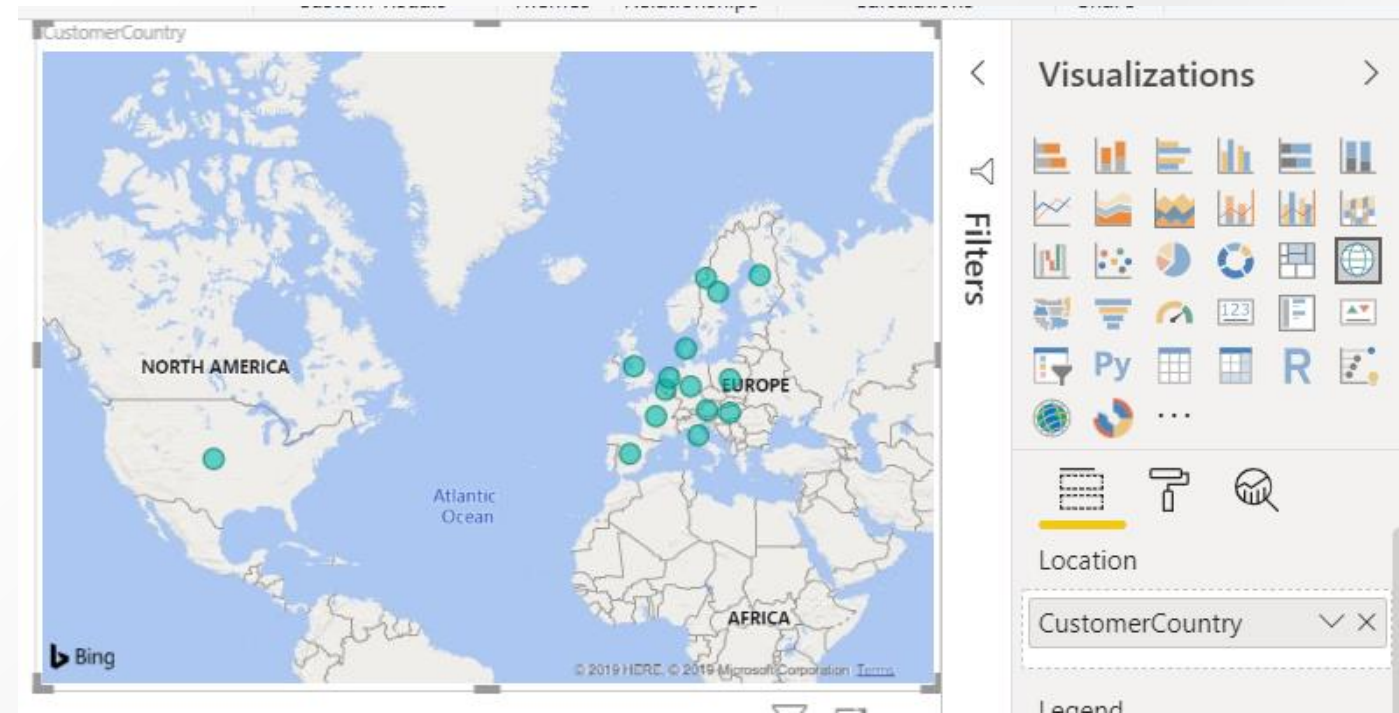
Tip:
Right click on the fact data and choose Hide

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

Visual

Search

General

Visualizations

Format visual



Search

Visual

General

✓ X-axis

On

✓ Values

Font

Segoe UI

9

B *I* U

Color

  *fx*

Max area height

25 %

Concatenate labels

On

> Title

On

Reset to default

> Y-axis

On

> Legend

> Small multiple title

> Small multiple grid



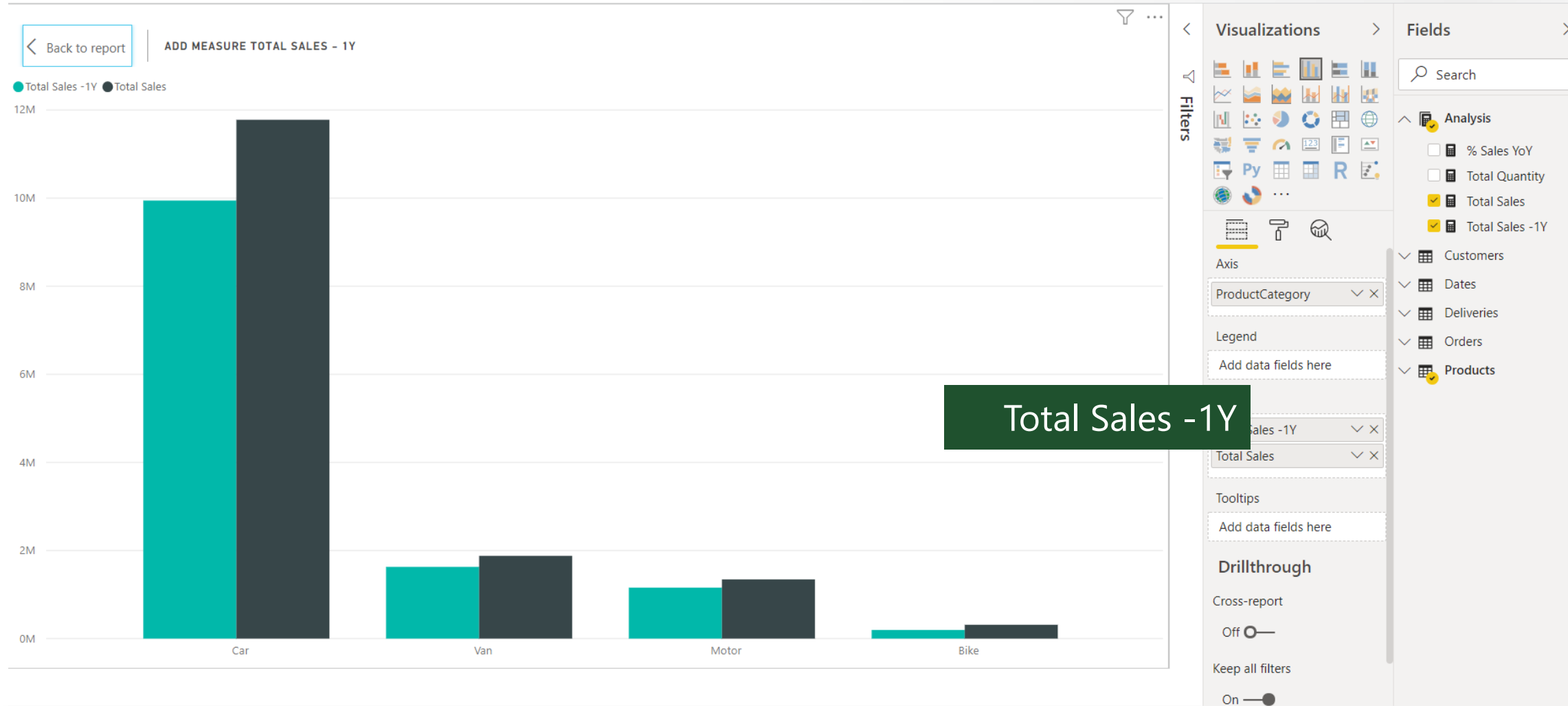
Context

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

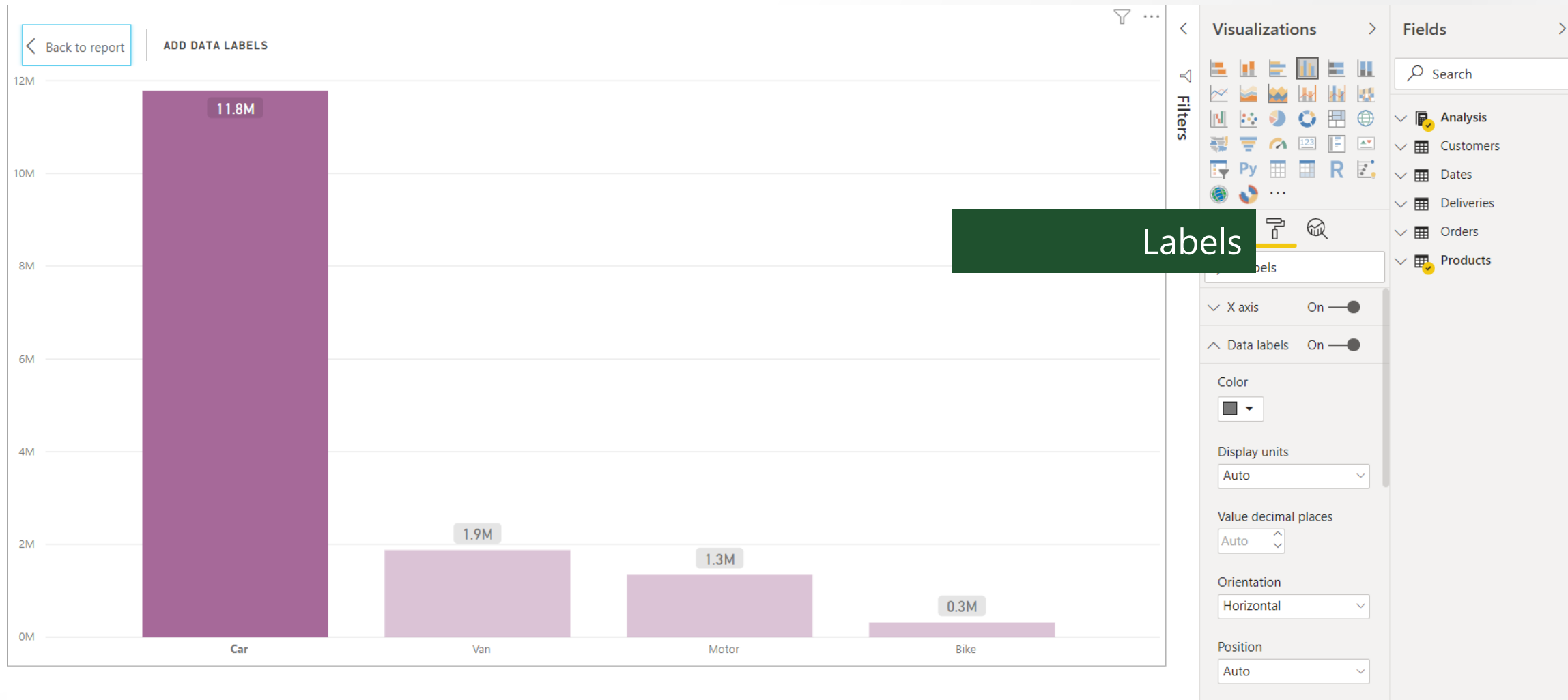
Another measure

Comparing with last year's values using a measure



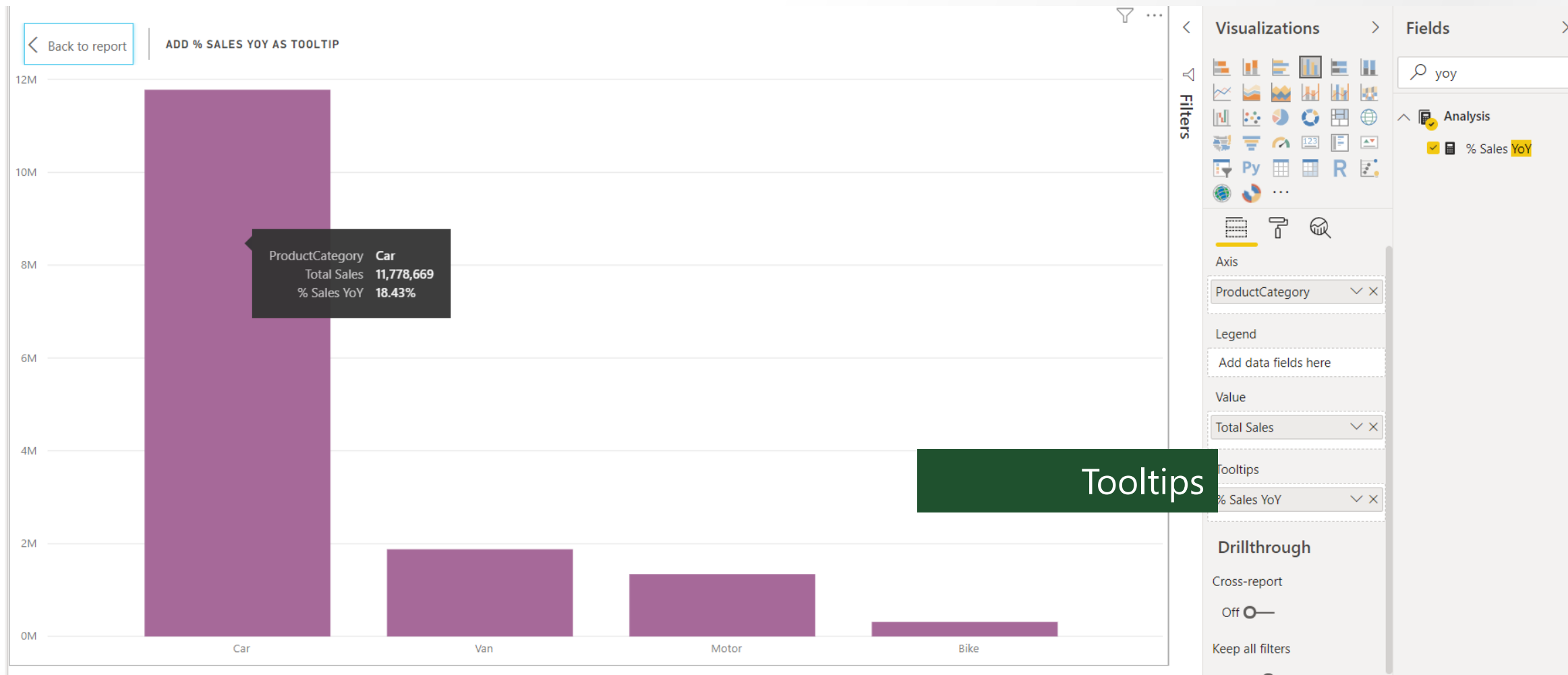
Data labels

Apply data labels for easy interpretation



Normal tooltips

Add additional information in the tooltip



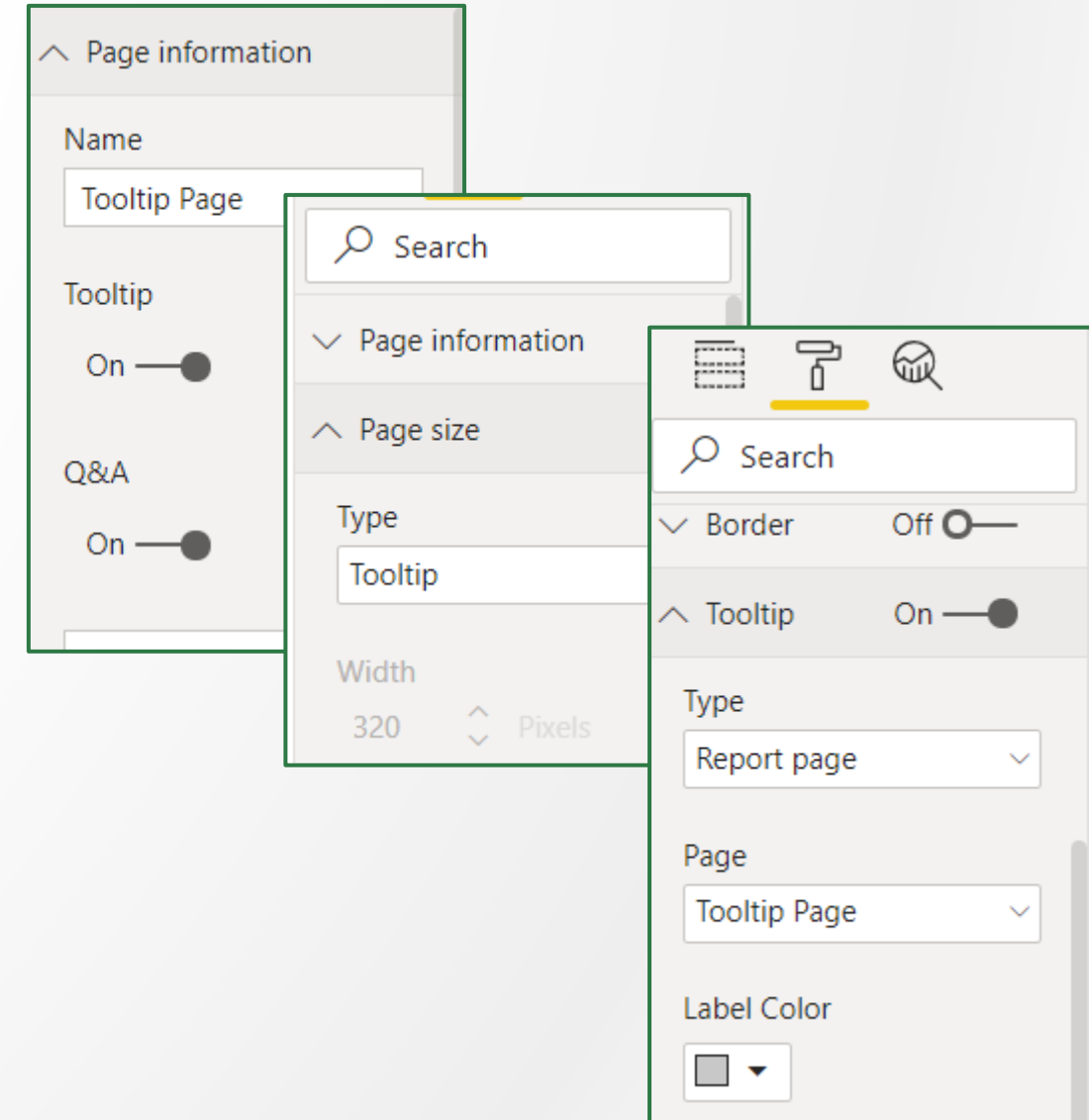
Tooltip Page

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

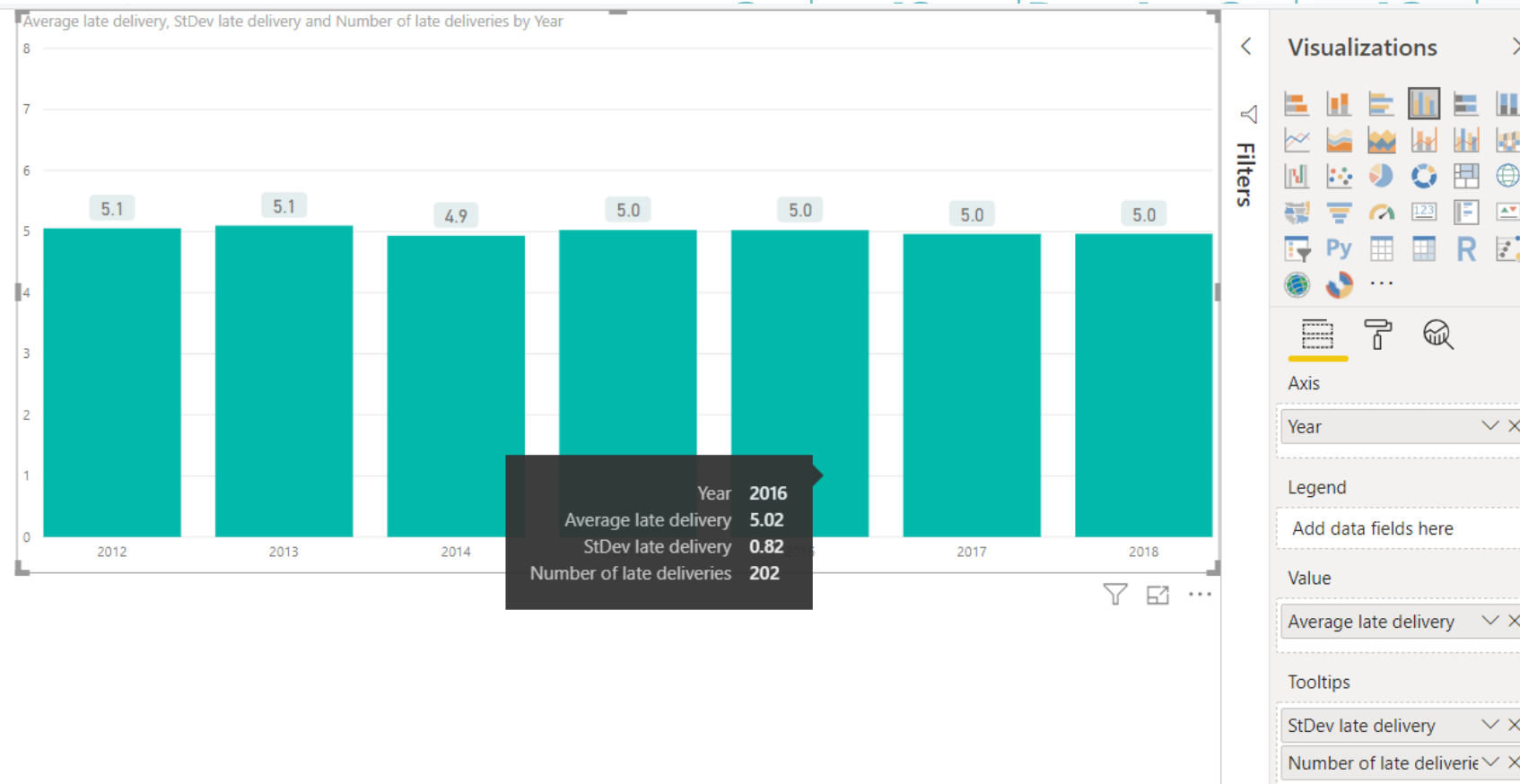
```
= STDEVX.P( FILTER( Orders ;  
    Orders[SendDate] - Orders[Orderdate] > 3 ) ;  
    Orders[SendDate] - Orders[Orderdate]  
)
```

Tooltips exercise

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

= COUNTROWS(FILTER(Orders ;

[LateDeliveryDate] > 3)

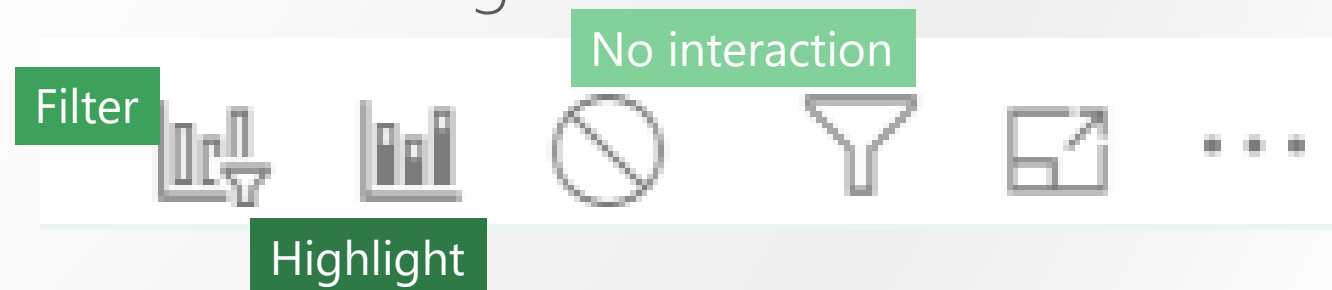


Slicers

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, multiple modes

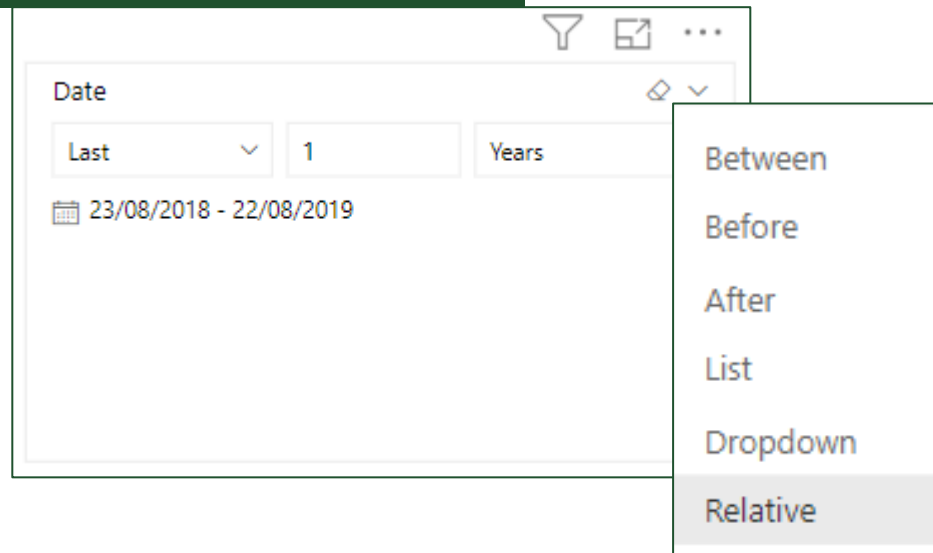
The diagram illustrates four different slicer modes in a data visualization tool:

- Dropdown:** A slicer with a single dropdown menu showing 'All'.
- List:** A slicer with a list of categories: Bike, Car, Motor, and Van. A green box labeled 'List' is placed over it.
- List - vertical:** A slicer with a vertical list of categories: Bike, Car, Motor, and Van. A green box labeled 'List - vertical' is placed over it.
- List - horizontal:** A slicer with a horizontal list of categories: Bike, Motor, Car, and Van. A green box labeled 'List - horizontal' is placed over it.

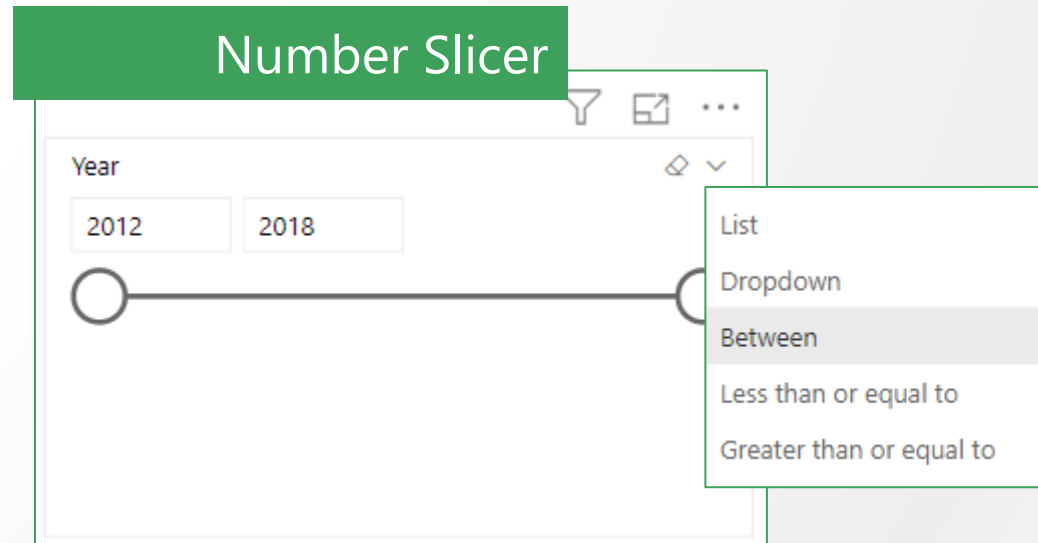
Arrows point from the 'List', 'List - vertical', and 'List - horizontal' modes to a central 'List' mode. A green box labeled 'General/ Orientation/ Horizontal' is placed over the 'List' mode, with an arrow pointing to the 'General' settings panel on the right. The 'General' panel shows the 'Orientation' dropdown set to 'Horizontal'.

Date and number slicers

Date Slicer



Number Slicer



Getting coffee

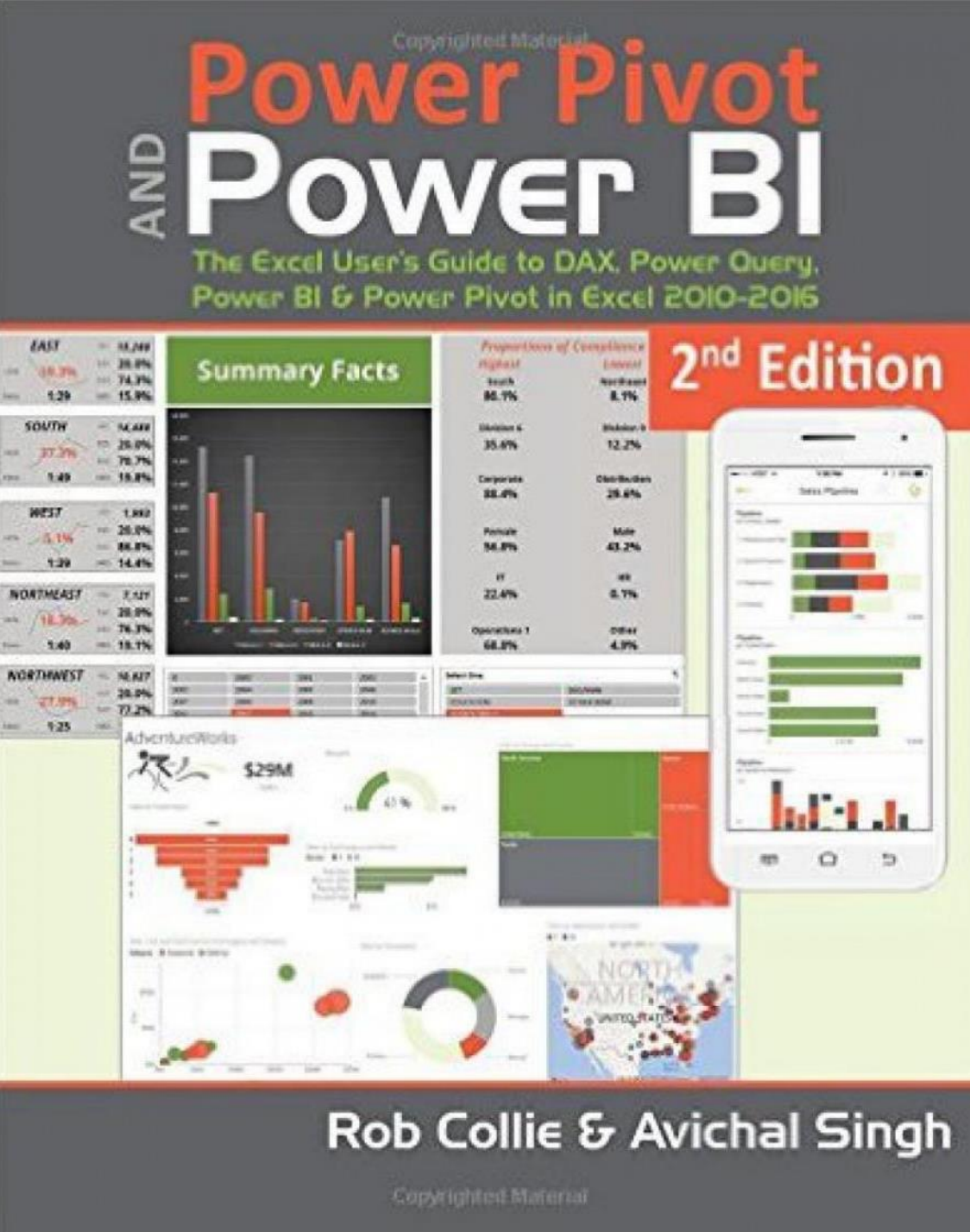




References & Guides

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 Excel and Power BI

Collect, Combine and Transform Data using Power Query in ...

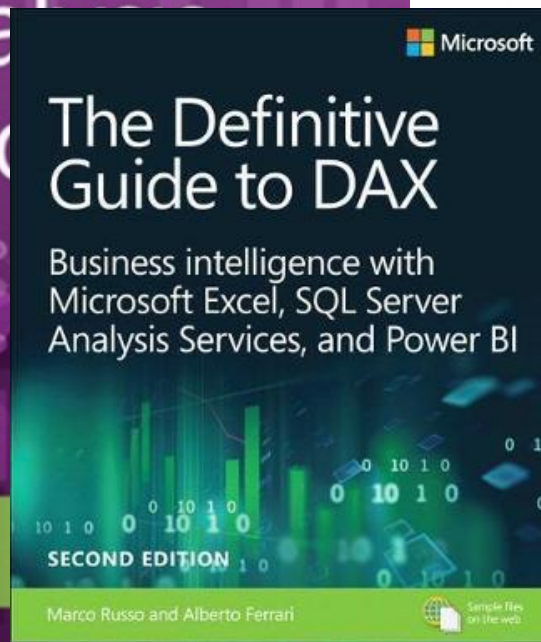
- Gil Raviv
- English
- Focus on
 - Power Query
- Missing



The Definitive Guide to DAX

Business intelligence with Microsoft Excel, SQL Server Analysis Services, and Power BI

Marco Russo and Alberto Ferrari



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