

1	POWER QUERY	5
1.1	Import Individual Files	5
1.2	Import Files from a Folder	5
1.2.1	Import Data	5
1.2.2	Disable Automatic Data Type Detection	6
1.2.3	Combine Multiple Data Sources	6
1.2.4	Adjust the Date Type	7
1.3	Copy Applied Steps	7
1.4	Import Files from a SharePoint Folder	7
1.5	Merge Data Sources (Replacement for VLOOKUP)	8
1.5.1	Split Column Contents	8
1.5.2	Merge Data Tables	8
1.5.3	Create a PivotTable	10
1.6	Append Different Tables	10
1.7	Column Operations	11
1.7.1	Remove Duplicates	11
1.7.2	Duplicate Date Columns and Perform Date Calculations	11
1.7.3	Fill Missing Values	11
1.7.4	Column From Examples	12
1.7.5	Conditional Column	12
1.8	Pivot Transformations	12
1.8.1	Unpivot Data	12
1.8.2	Pivot Data	13
1.9	Display Data Quality	14
1.9.1	Clean Data	14
1.10	Data Imports	15
1.10.1	Import Multiple Workbooks and Worksheets	15
1.10.2	Importing and Parsing XML and JSON Data	16
1.11	SQL Queries	17
1.12	Web Queries	18
1.13	Join Types	18
1.13.1	Join Examples	19
1.13.2	Inner Join	21
1.13.3	Left Anti Join	21
1.14	Grouping	21
1.14.1	Group by Multiple Criteria	21
1.14.2	Grouping with Nested Tables	22
1.15	Expression Examples	23
2	POWER QUERY ADVANCED	24

2.1	Analyzing complex Excel files	24
2.1.1	Simple average with List()	24
2.1.2	Add file names and sheet names to a list	24
2.1.3	Delete older duplicates.....	25
2.1.4	Create a date list from From..To	25
2.1.5	Determine previous periods using a self-join	25
2.1.6	Web query for stock data: an introduction to M.....	26
2.2	M-Code	26
2.2.1	M code structure	26
2.2.2	Help	27
2.2.3	Data types.....	28
2.2.4	Types of values.....	28
2.2.5	Create a table (table constructor).....	29
2.2.6	References.....	29
2.2.7	each and _	30
2.3	Import fragmented data (workbooks/tables).....	30
2.3.1	One workbook, all sheets/columns identical, pivoted: solution using helper columns	30
2.3.2	Multiple workbooks from a folder, all sheets/columns identical, pivoted: solution using helper columns 31	
2.3.3	Multiple workbooks from a folder, all with different columns, pivoted: solution using M and a function 31	
2.4	Useful M functions	32
2.5	Practical imports.....	34
2.5.1	One workbook, all sheets/columns identical, pivoted: solution using helper columns	34
2.5.2	Multiple workbooks from a folder, all sheets/columns identical, pivoted: solution using helper columns 34	
2.6	Functions and parameters	35
2.6.1	Define functions.....	35
2.6.2	Define parameter queries (transform many files at once)	35
2.6.3	Pass data from Excel to a query	36
2.6.4	If a Formula Firewall error occurs	37
2.7	Techniques	37
2.7.1	Parameter.....	37
2.8	Miscellaneous.....	39
2.8.1	Insert rows	39
2.8.2	Calculate the previous day.....	39
2.8.3	Datediff.....	39
2.8.4	List data from...to in rows.....	39
3	POWER BI DESKTOP.....	41
3.1	Data Imports	41
3.1.1	Multi-File CSV Import.....	41
3.1.2	Interim Report	42
3.1.3	Publish Reports	43
3.1.4	Publish Reports to SharePoint.....	43
3.1.5	Excel Import with Relationships.....	43
3.1.6	Import SharePoint Data	45
3.2	Create a Simple Report.....	46

3.2.2	Create Measures	47
3.2.3	Insert Visuals	48
3.2.4	Filter Visuals	49
3.3	Create Tables Using a Function	49
3.3.1	Insert New Columns	49
3.4	Types of Table Relationships	50
3.5	Master Date Table	50
3.6	Interim Report with Treemap and Slicer	51
3.6.1	Filter for All Pages	52
3.7	Create DAX Measures (Aggregates)	52
3.8	Create Dashboards	53
3.8.1	Matrix (PivotTable)	53
3.8.2	Column Chart with Drill-Down	54
3.8.3	Line Chart	55
3.8.4	Card and Multi-Row Card	55
3.8.5	Slicer	55
3.8.6	Import New Visuals	56
3.9	KPI and Conditional Formatting	56
3.10	Publish Reports	57
3.11	Compare Prior-Year Periods	57
3.11.1	Calculate	57
4	POWER BI DESKTOP ADVANCED	59
4.1	Filter Concepts and Filter Propagation	59
4.2	DAX	60
4.2.1	Summarize Discount Percentage Bands	60
4.2.2	Calculate	60
4.3	Inactive Relationships (USERELATIONSHIP)	61
4.4	Filter Context	61
4.4.1	Context Transition	62
4.5	Parameter	62
4.5.1	Dynamic Dimensions (Field Parameters)	62
4.5.2	Numeric Parameters (Numeric Range)	63
4.6	Count/Calculate Orphaned Records	63
4.7	AI Analyses	63
4.7.1	Changes in the Chart	63
4.8	Advanced DAX	64
4.8.1	Previous Periods	64
4.8.2	YTD (Year to Date)	65
4.8.3	IF()	66

4.8.4	Switch	66
4.8.5	Group Measures in a Folder	67
4.8.6	Calculate Totals for Percentage Shares	67
4.8.7	Variables in DAX	68
4.8.8	OR/AND/IN	69
4.8.9	Combine AND/IN with FILTER().....	69
4.8.10	Time-Based Analysis.....	69
4.8.11	Calculate Capacities	72
4.8.12	Row Context	72
4.8.13	KEEPFILTERS().....	73
4.8.14	DATATABLE()	73
4.8.15	Access Previous Rows in DAX (Beta)	74
4.8.16	Lookupvalue	74
4.8.17	VALUES()	74
4.8.18	SELECTEDVALUE()	74
4.8.19	LOOKUPVALUE()	74
4.8.20	ALL() ALLSELECTED() ALLEXCEPT().....	75
4.8.21	CONVERT()	75
4.8.22	SUMMARIZE()	76
4.9	Visuals.....	76
4.9.1	Drillthrough.....	76
4.9.2	Bookmarks and Interactions	77
4.9.3	Tooltip	77
4.9.4	Small Multiples	78
4.9.5	Map Visuals	78
4.9.6	Map	79
4.9.7	Filled Map	79
4.9.8	Sparkline	79
4.9.9	Ribbon Chart.....	80
4.9.10	Decomposition Tree	80
4.9.11	Scatter Chart.....	81
4.9.12	Add visuals (example: Infographic Designer)	81
4.10	Power BI Service.....	82
4.10.1	Automatically refresh online data (datasets)	82
4.10.2	Set Up the Power BI Data Gateway	83
4.10.3	Import Data from Teams	84
4.10.4	Create a Dashboard	84
4.10.5	Analyze Power BI (Portal) Data in Excel	85
4.11	Miscellaneous.....	86
4.11.1	Display a List of Image URLs in a Table.....	86

1 Power Query

1.1 Import Individual Files

Folder: First Import

Import the *.txt file:

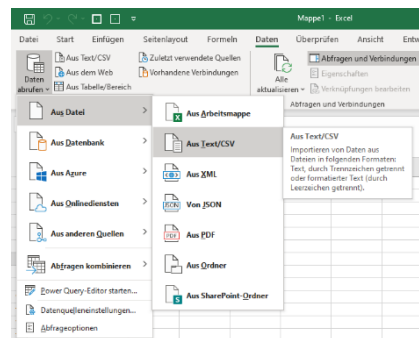


Figure 124

Changing the data source in Power Query:

- Home > Data Source Settings

1.2 Import Files from a Folder

Folder: Multi CSV

1.2.1 Import Data

You can import external data from sources such as:

- From Database (e.g. SQL, Access)
- From File (or From Folder!)
- JSON/XML

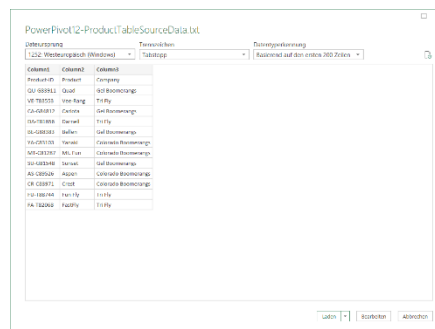


Figure 125

Load

- Create a connection only
- Directly into Excel

Note: The *Add this data to the Data Model* checkbox makes the data source available in Power Pivot.

TIP: For text files, use the command at the top left: Use First Row as Headers.

1.2.2 Disable Automatic Data Type Detection

Because Power Query's automatic data type detection often identifies data incorrectly, it can be useful to disable it via *File > Options and settings > Query Options*. Under *Current Workbook > Data Load*, select *Never detect column types and headers for unstructured sources*.

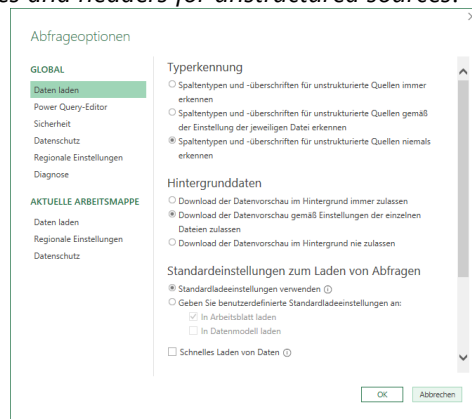


Figure 126

1.2.3 Combine Multiple Data Sources

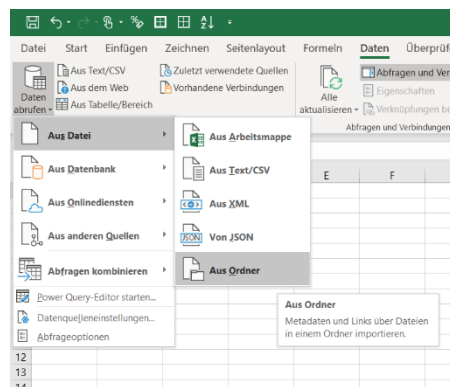


Figure 127

You can now use the *Combine Data* button to combine all data sources:

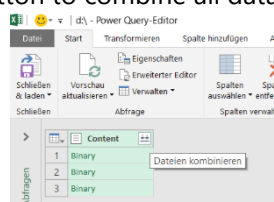


Figure 128

1.2.4 Adjust the Date Type

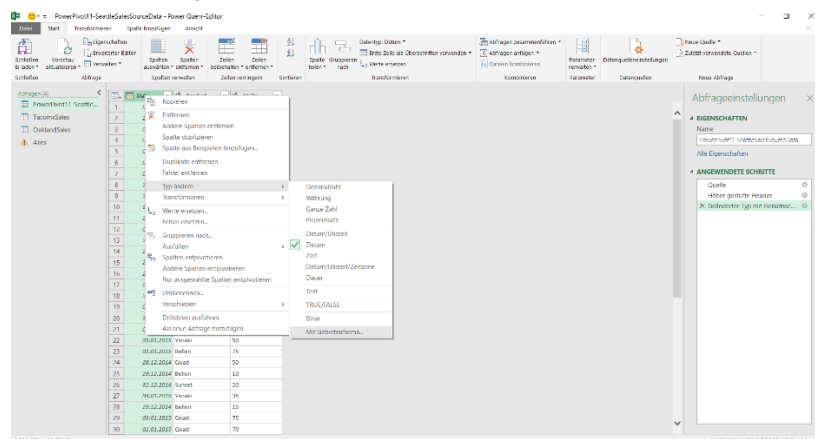


Figure 129

This action is recorded as an applied transformation step and is automatically reapplied whenever the data source is refreshed.

1.3 Copy Applied Steps

- 1) Right-click the source query and select *Advanced Editor*.
- 2) Copy the script and click Cancel
- 3) Right-click the target query and select *Advanced Editor*.
- 4) Paste the script and adjust the file path if necessary

1.4 Import Files from a SharePoint Folder

Folder: <https://edvadsde.sharepoint.com/sites/Teamsite> ([Teamsite – Home \(sharepoint.com\)](#))

First, copy the URL of the required SharePoint team site:

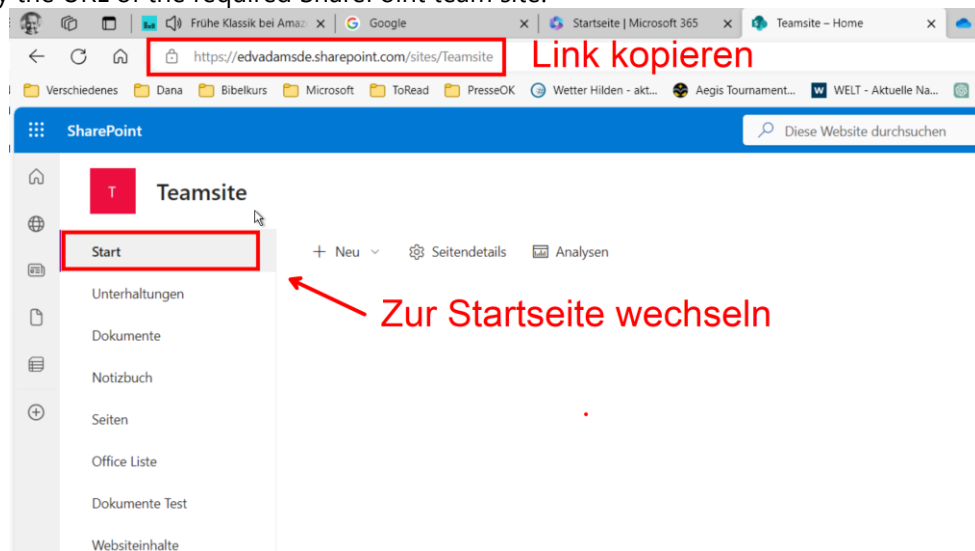


Figure 130

Then start the import:

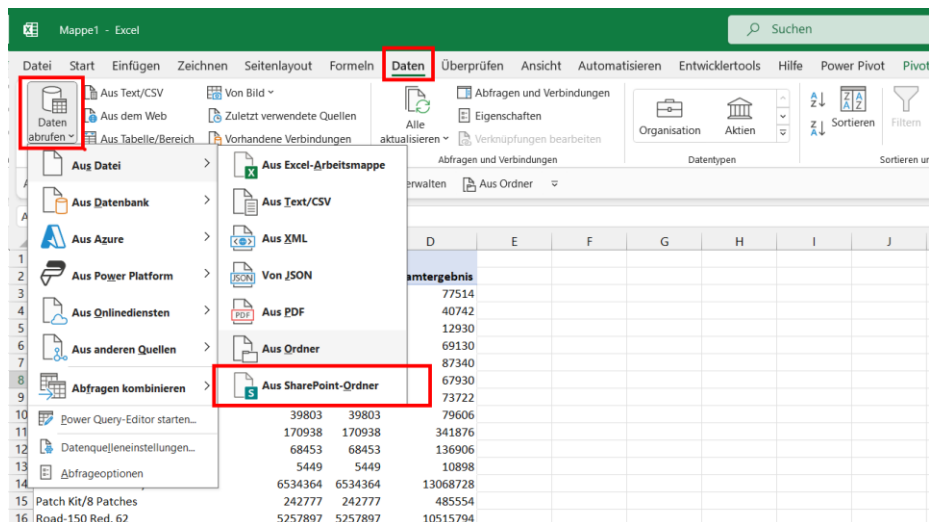


Figure 131

Then paste the link into the following dialog box.

1.5 Merge Data Sources (Replacement for VLOOKUP)

Folder: Link Data Sources

- Import Excel table
- Import text file
- Standardize the common key column (ID) (see next section: 6.5.1)

TIP:

For text files, you can easily promote the first row to column headers:

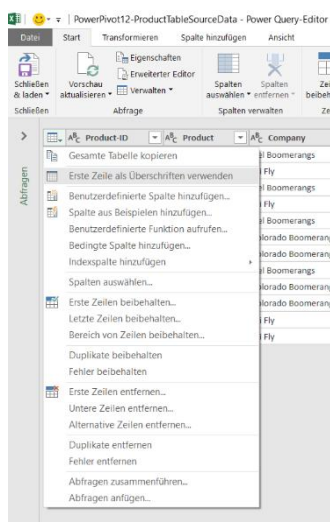


Figure 132

1.5.1 Split Column Contents

- *Data > Queries & Connections > Queries > Edit*
- *Transform > Split Column*

1.5.2 Merge Data Tables

In Excel: Select the relevant queries and choose *Merge*.

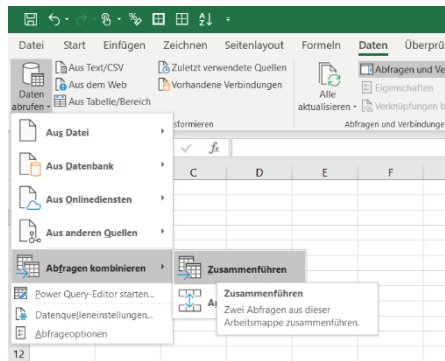


Figure 133

In Power Query: Select *Home > Merge Queries* (or *Home > Merge Queries > Merge Queries as New*).

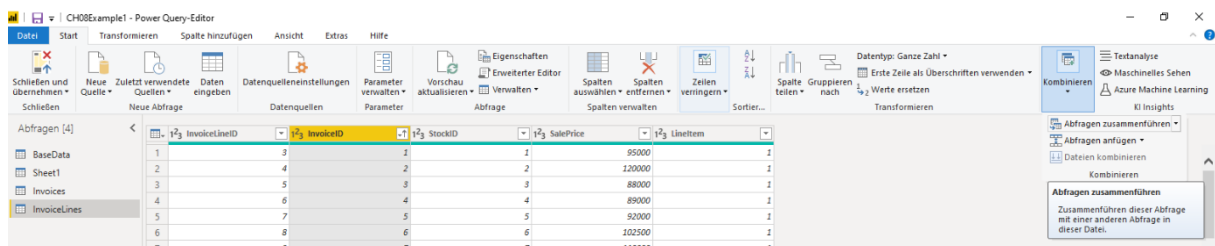


Figure 134

Then select the columns to be linked:

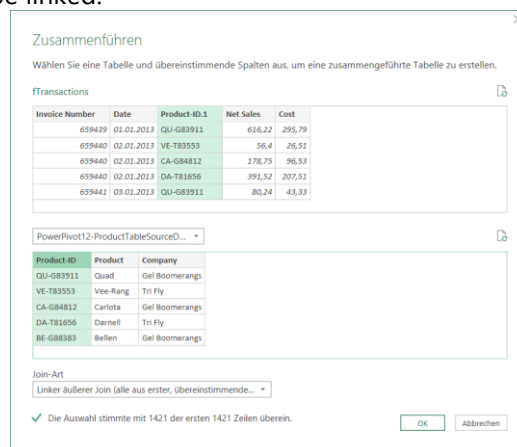


Figure 135

Then select the columns to include in the merged query:

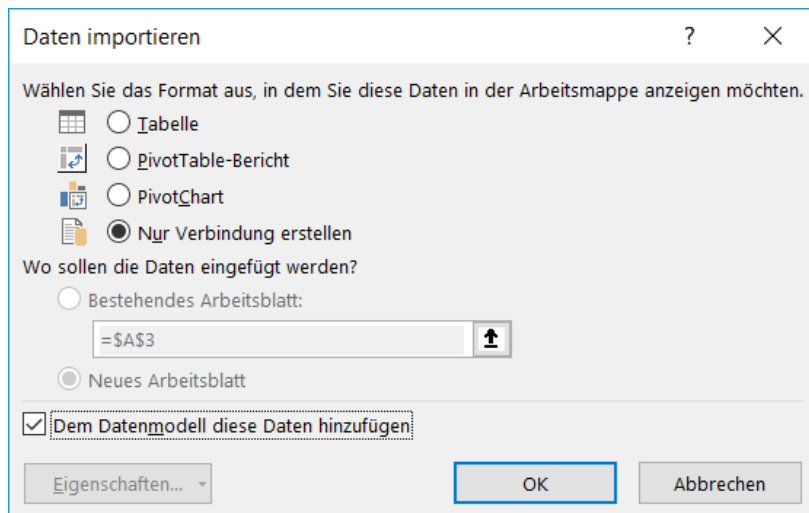


Figure 136

Note: If the data is added to the Data Model via the checkbox, this data source is available for further processing in Power Pivot.

In the expanded column header, select the columns to include in the query:

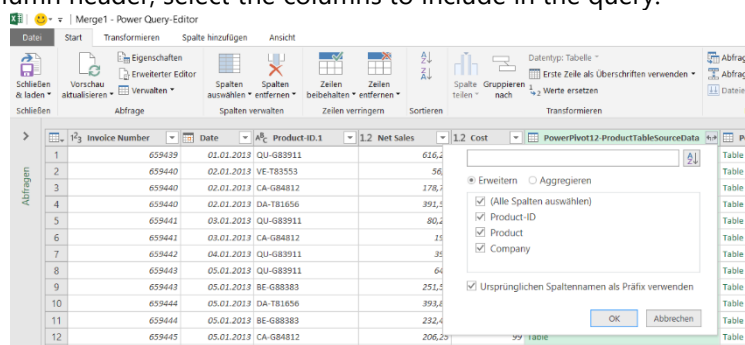


Figure 137

1.5.3 Create a PivotTable

Right-click the query in Excel's *Queries & Connections* pane and select *Load To...* to create, for example, a new PivotTable from the data.

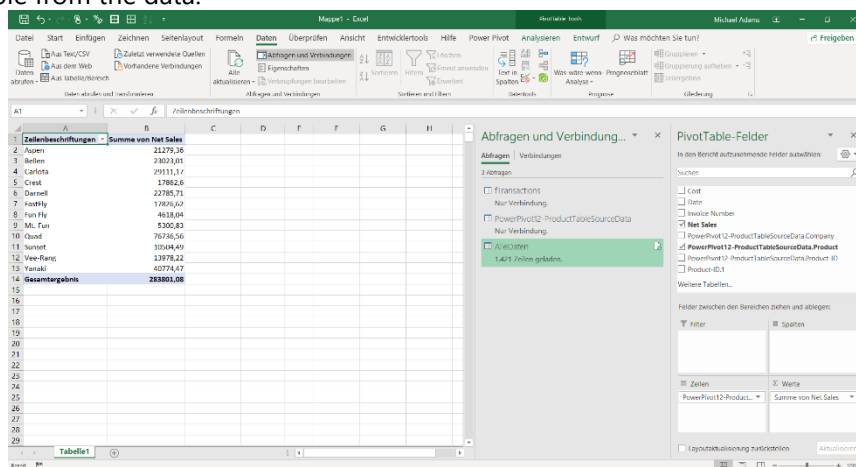


Figure 138

1.6 Append Different Tables

Folder: Append

- Create a new workbook

- Load all three tables and use *Load To* to create a connection only

Because the source queries have different columns, they must be standardized:

- In the *Sales2017* table, *Units*, *NetCostEquivalent*, and *Price* must be multiplied. To do this, select the columns to be multiplied and choose *Add Column > Standard > Multiply*:

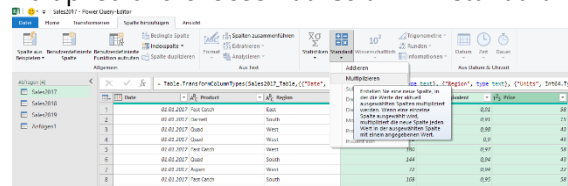


Figure 139

- Rename the result column to *Revenue* so that the column names are identical across all tables.
- You can now remove *Units*, *NetCostEquivalent*, and *Price*.
- In the *Sales2019* table, remove the redundant *Color* column.
- Select *Home > Append Queries > Append Queries as New*, choose *Three or more tables*, and select all three queries:

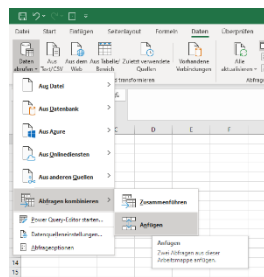


Figure 140

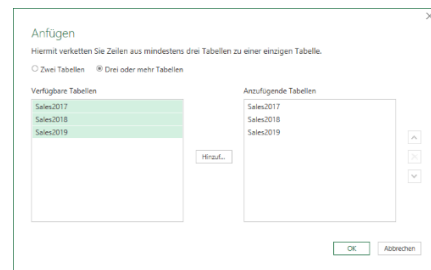


Figure 141

Tip: Columns are appended based on their **column names**, not their contents. To combine differently named columns into a single column, rename them so that they have the same column name.

1.7 Column Operations

1.7.1 Remove Duplicates

Folder: Duplicates

First, import the entire folder and combine it into a list.

Then select the columns whose combined values identify duplicate rows and choose *Home > Remove Rows > Remove Duplicates*.

1.7.2 Duplicate Date Columns and Perform Date Calculations

Folder: CSV Multi-Import

To derive information from a date, first duplicate the date column and then apply the required date transformation:

- *Add Column > Duplicate Column*
- In the copy: *Add Column > Date*, e.g.:
 - *Add Column > Date > Age*
 - Then: *Transform > Duration > Total Years*

1.7.2.1 Calculate the Date Difference

`Duration.Days([End]-[Start])`

1.7.3 Fill Missing Values

File: Misc/CarMakeAndModelMatrix.xlsx and HierarchieSAP.txt

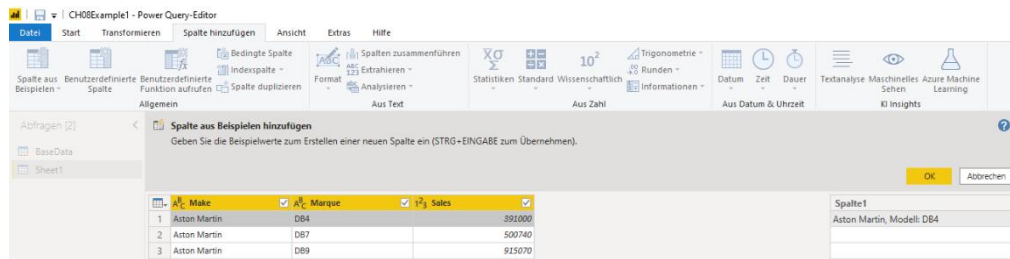
To assign each row to the appropriate value, select the relevant column and choose *Transform > Fill > Down*, or *Transform > Fill > Up* as required.

1.7.4 Column From Examples

File: Misc/CarMakeAndModelMatrix.xlsx

This feature lets you create a new column from examples without writing formulas or commands:

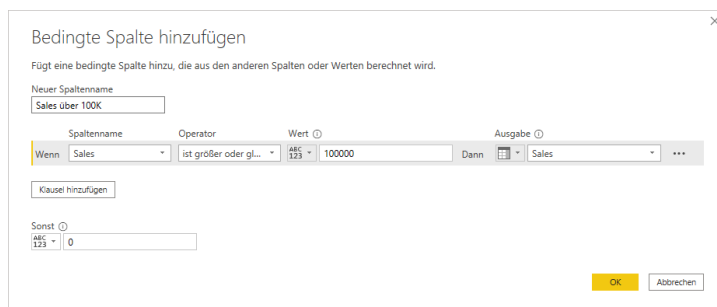
- *Add Column > Column From Examples > From All Columns*
- Enter the required example data here:



1.7.5 Conditional Column

File: Misc/CarMakeAndModelMatrix.xlsx

To create a new column using if-then logic, select *Add Column > Conditional Column* and then define the required condition(s):



1.8 Pivot Transformations

1.8.1 Unpivot Data

Folder: *Unpivot*

- *Home > Remove Rows > Remove Top Rows (2)*
- *Home > Use First Row as Headers*
- Filter out blank rows (*null*)
- Delete the total column
- Select the month columns and choose *Transform > Unpivot Columns*:

Category	Subcategory	Productname	Producttype	Produkt	Umsatz
Accessories	Bike Racks	Hitch Rack - 4-Bike		487 Jan, 2016	2222
Accessories	Bike Racks	Hitch Rack - 4-Bike		487 Feb, 2016	2033
Accessories	Bike Racks	Hitch Rack - 4-Bike		487 Mar, 2016	4234
Accessories	Bike Racks	Hitch Rack - 4-Bike		487 Apr, 2016	2179
Accessories	Bike Racks	Hitch Rack - 4-Bike		487 May, 2016	2637
Accessories	Bike Racks	Hitch Rack - 4-Bike		487 Jun, 2016	3279
Accessories	Bike Racks	Hitch Rack - 4-Bike		487 Jul, 2016	2238
Accessories	Bike Racks	Hitch Rack - 4-Bike		487 Aug, 2016	5287
Accessories	Bike Racks	Hitch Rack - 4-Bike		487 Sep, 2016	3883
Accessories	Bike Racks	Hitch Rack - 4-Bike		487 Oct, 2016	2484
Accessories	Bike Racks	Hitch Rack - 4-Bike		487 Nov, 2016	3442
Accessories	Bike Racks	Hitch Rack - 4-Bike		487 Dec, 2016	3552
Accessories	Bike Stands	Alt Purpose Bike Stand		488 Jan, 2016	686
Accessories	Bike Stands	Alt Purpose Bike Stand		488 Feb, 2016	3693
Accessories	Bike Stands	Alt Purpose Bike Stand		488 Mar, 2016	2868
Accessories	Bike Stands	Alt Purpose Bike Stand		488 Apr, 2016	4882
Accessories	Bike Stands	Alt Purpose Bike Stand		488 May, 2016	3439
Accessories	Bike Stands	Alt Purpose Bike Stand		488 Jun, 2016	4812
Accessories	Bike Stands	Alt Purpose Bike Stand		488 Jul, 2016	2774
Accessories	Bike Stands	Alt Purpose Bike Stand		488 Aug, 2016	3003
Accessories	Bike Stands	Alt Purpose Bike Stand		488 Sep, 2016	2402
Accessories	Bike Stands	Alt Purpose Bike Stand		488 Oct, 2016	4413
Accessories	Bike Stands	Alt Purpose Bike Stand		488 Nov, 2016	3882
Accessories	Bike Stands	Alt Purpose Bike Stand		488 Dec, 2016	2243
Accessories	Bottles and Cages	Water Bottle - 30 oz.		477 Jan, 2016	3892
Accessories	Bottles and Cages	Water Bottle - 30 oz.		477 Feb, 2016	4727
Accessories	Bottles and Cages	Water Bottle - 30 oz.		477 Mar, 2016	3658
Accessories	Bottles and Cages	Water Bottle - 30 oz.		477 Apr, 2016	4449
Accessories	Bottles and Cages	Water Bottle - 30 oz.		477 May, 2016	4051

Figure 142

1.8.2 Pivot Data

Folder: *Pivot*

First, select the column to be pivoted—in our example, *Regions*—and choose *Transform > Pivot Column*. Then select the values column to be aggregated, in this case *Revenue*:

Spalte pivotieren

Verwenden Sie die Namen in Spalte "Produkt" zum Erstellen neuer Spalten.

Wertespalte

Aggregatwertfunktion

Weitere Informationen über "Spalte pivotieren"

OK Abbrechen

Figure 143

Result:

Produkt	West	Süd	Nord	Ost	
Außenvelozahn	7767	13723,2	2118,03	null	
Inbus	22648,12	1434,65	5201,83	null	
Innenzschkant	12042,56	19386,8	16295,65	7264,24	
Innenzschkant-TR	5910,87	6410,88	801,53	1108,46	
Innenvelozahn	13810,58	2459,5	3847,07	null	
Phillips-Kreuzschlitz	785,52	null	13382,19	10012,69	
Positiv-Kreuzschlitz	14272,17	2655,1	8082,3	5410,36	
Rändelkopf	9630,17	9007,17	null	3503,89	
Schlitz	8210,62	10921,05	15889,13	null	
Sechskant	6314,17	10881,86	24676,85	11374,19	
TorxSet	4504,58	4341,72	12333,07	9393,69	
Torx	5028,85	11654,29	null	9004	
Torx-TR	12844,36	null	20626,51	9069,81	
Tri-Wing	12928,52	1786,76	3963,07	6500,32	
Vierkant	10012,31	6116,29	null	null	

Figure 144

Tip: *Transform > Transpose* switches rows and columns.

1.9 Display Data Quality

Folder: Pivot raw data

The View tab provides several options for displaying and assessing data quality:

- Formula bar for editing expressions
- Column quality shows errors or empty data
- Column distribution shows the data distribution
- Column profile shows details about the selected column at the bottom of the screen

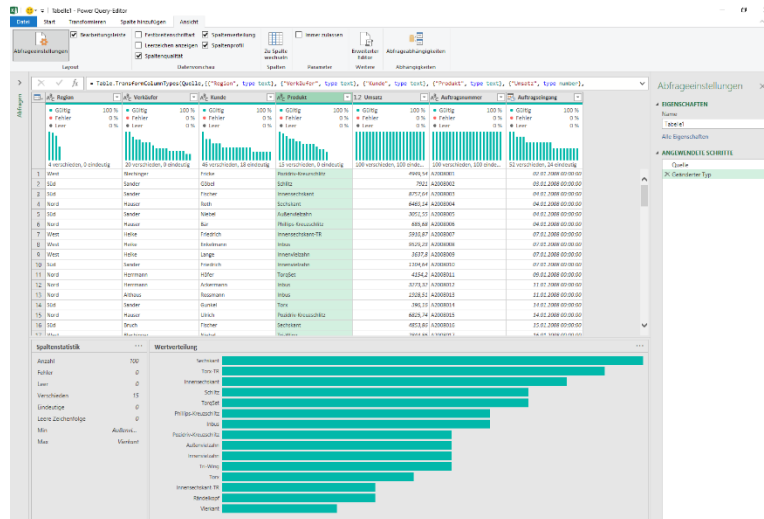


Figure 145

1.9.1 Clean Data

You can perform basic data-cleaning operations directly from the column profile:

- Remove duplicates
- Remove empty
- Remove errors

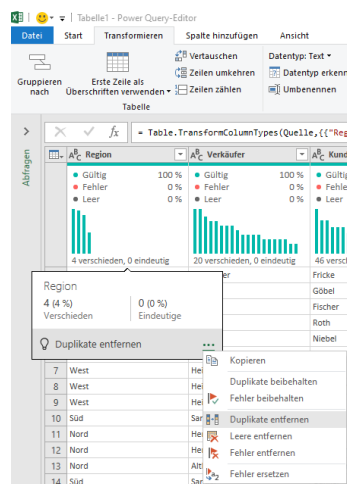


Figure 146

1.10 Data Imports

1.10.1 Import Multiple Workbooks and Worksheets

Folder: Mappen und Tabellen Multiimport\Data

This process consolidates multiple workbooks and worksheets into a single Excel table.

First, select the folder that contains the files to be imported and load it into Power Query as a connection. See also Section 6.10.

Then remove all columns except the first:

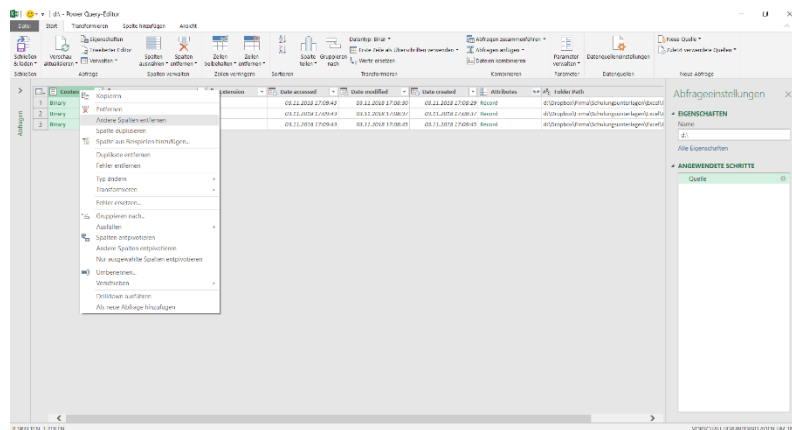


Figure 147

Add a new ExcelData column:

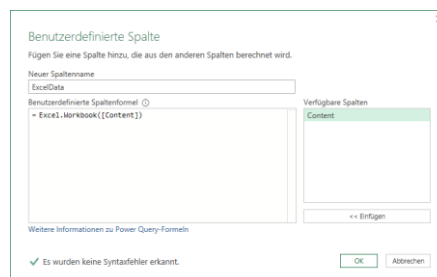


Figure 148

Remove the column containing the Binary values, and then expand the contents to obtain the following result:

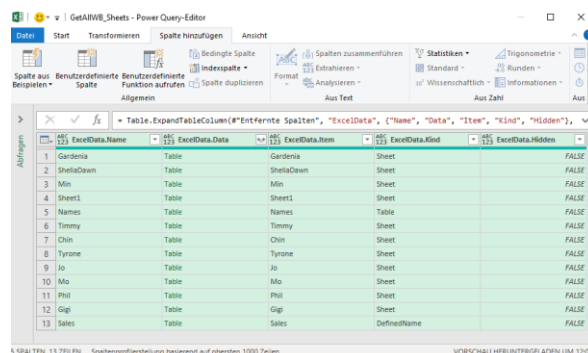


Figure 149

Then:

- Kind: filter by Sheet
- Item: <>Sheet
- Remove Kind, Name, and Hidden

Clicking a Table entry reveals that the column headers are incorrect. Correct them as follows:

- Add a Custom Column named PromoteHeaders: Table.PromoteHeaders([ExcelData.Data])
- *Expand PromoteHeaders*

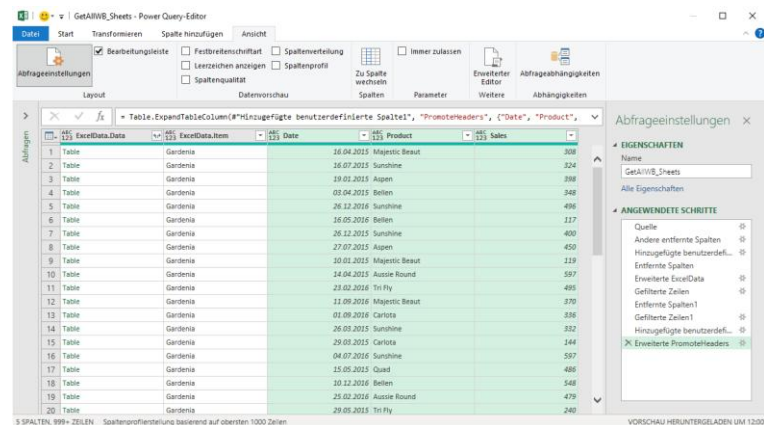


Figure 150

Rename ExcelData.Item, for example, to SalesRep.

Note: If the date is to be grouped in a PivotTable, its data type must be set to Date in Power Query.

1.10.2 Importing and Parsing XML and JSON Data

JSON: JavaScript Object Notation

XML: Extensible Markup Language

1.10.2.1 Extract Data from a Column

Folder: XML_JSON

First, connect to XMLInColumn.xlsx. Then select the column containing the XML data (AvailableColors). Next, select Add Column > Parse > XML.

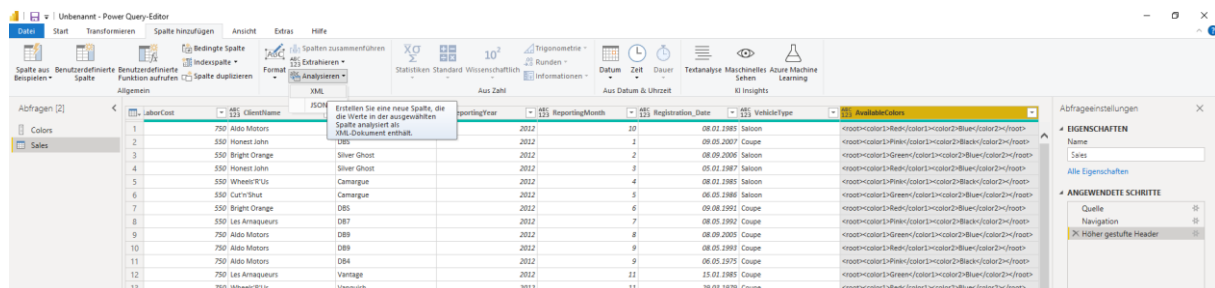


Figure 151

1.10.2.2 Load and Parse Complex JSON Files

Folder: XML_JSON

- First, connect to CarSalesJSON_Complex.Json.
- Then convert the list to a table
- Expand the list using the Expand to New Rows command (see Figure 152).
- Finally, expand each required Record individually.

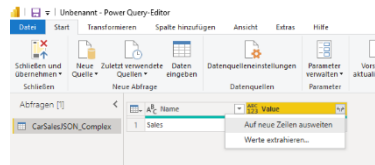


Figure 152

Difference between a list and a table: A list contains row-by-row data that cannot be analyzed at column level. A table contains columns, so it often makes sense to convert a list to a table.

1.11 SQL Queries

A connection to SQL Server can be initiated via *Data > Get Data > From Database > From SQL Server Database*:

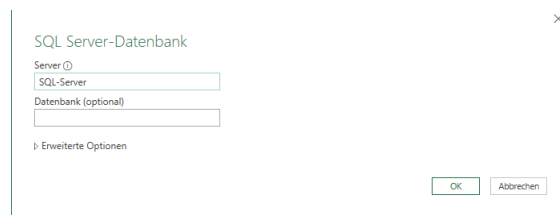


Figure 153

The server can be hosted locally or accessed over the web.

If the server is accessible, you will be prompted for your sign-in credentials...

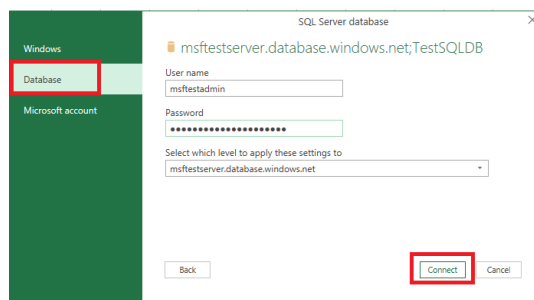


Figure 154

The connection to the databases available to you is then established:

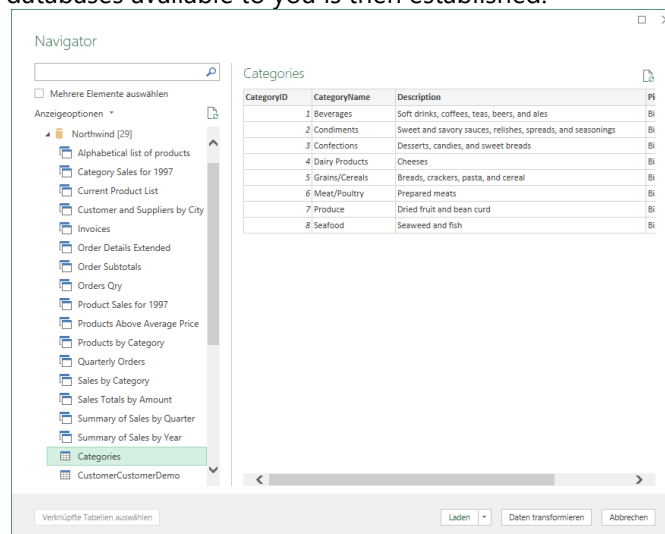


Figure 155

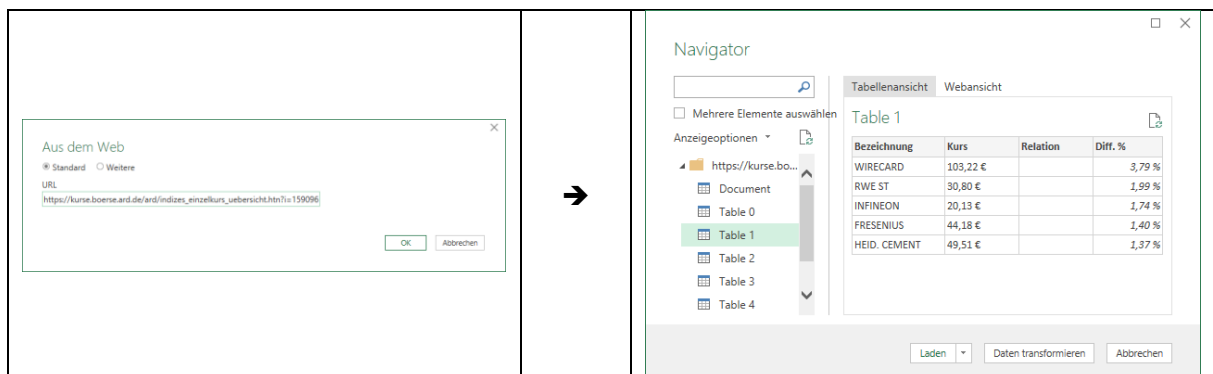
1.12 Web Queries

For example:

<https://www.tagesschau.de/wirtschaft/boersenkurse/>

Data can be retrieved from websites by selecting *Data > Get Data > From Other Sources > From Web*.

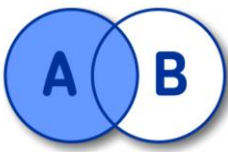
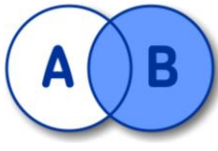
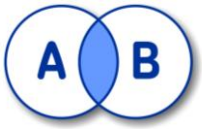
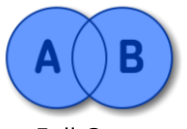
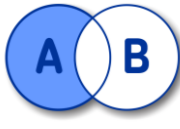
Simply enter the web address, and Power Query displays all sources found at that address:



1.13 Join Types

Folder: Join types

Different join types can be used when merging tables:

Type	Effect
 Left Outer	Returns all records from the left table and matching records from the right table. This corresponds to VLOOKUP.
 Right Outer	Returns all records from the right table and matching records from the left table.
 Inner Join	Shows only records that occur in both tables (AND)
 Full Outer	Shows all records, whether or not there is a match in the other table
 Left Anti	Returns records from the left table that have no matching record in the right table.
 Right Anti	Returns records from the right table that have no matching record in the left table.

Note: The Fuzzy matching option lets you specify how closely the fields used for merging must match.

1.13.1 Join Examples

1.13.1.1 Simple Left Outer

For each sale in the left table, retrieve the product price from the right table. This corresponds to the VLOOKUP() or XLOOKUP() function.

Product	Units	Product	Price
Quad	48	Carlota	\$26
Kangaroo	168	Quad	\$43
Carlota	132	Sunshine	\$19
Carlota	72	Majestic Beaut	\$27
Sunshine	108		
Quad	156		
Carlota	96		
Sunshine	60		
Sunshine	24		
Carlota	120		
Quad	24		

Figure 156

1.13.1.2 Multi-Column Left Outer Join

To use multiple key criteria simultaneously, such as product and color, hold down the Ctrl key and select the corresponding columns in each table when creating the merge in Power Query, as shown in Figure 158.

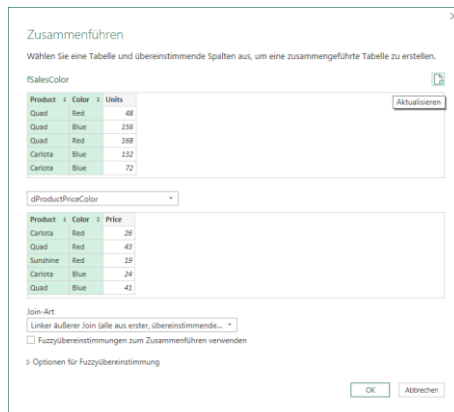


Figure 157

Product	Color	Units	Product	Color	Price
Quad	Red	48	Carlota	Red	\$26,00
Quad	Blue	156	Quad	Red	\$43,00
Quad	Red	168	Sunshine	Red	\$19,00
Carlota	Blue	132	Carlota	Blue	\$24,00
Carlota	Blue	72	Quad	Blue	\$41,00
Sunshine	Red	108	Sunshine	Blue	\$18,00
Quad	Blue	156			
Carlota	Red	96			
Sunshine	Red	60			
Sunshine	Blue	24			
Carlota	Blue	120			
Quad	Blue	24			

Figure 158

1.13.1.3 Left Outer with Multiple Matches in the Target Table

Multiple matches cannot be looked up in a target table using VLOOKUP:

Invoice No.	Discount%	Invoice No.	Product	Sales
4588	0,065	4588	Carlota	130
4589	0,0375	4588	Quad	559
4590	0,12	4588	Sunshine	114
		4589	Quad	559
		4589	Sunshine	209
		4590	Carlota	2869

Figure 159

Unlike VLOOKUP, a Power Query join can return and evaluate multiple matches from the target table.

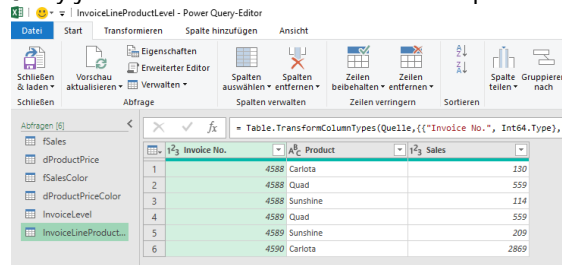


Figure 160

1.13.2 Inner Join

Which participants were in both cities?

Berlin		Frankfurt
Sol Marroquin		Raven Beatty
Kiera Mcfall		Roxanna Mercier
Raven Beatty		Fanny Denning
Elinore Dees		Lesha Nobles
Wei Lockwood		Wei Lockwood
Donald Eldridge		Gertrudis Fitzpatrick
Claudio Beam		Angelita Packer
Angelita Packer		Beaulah Wenger
Reyna Luke		Malvina Hamer
Beaulah Wenger		Bernita Crutcher
Malvina Hamer		Shiela Anaya
Vivan Keeney		Yolonda Armstead

Figure 161

1.13.3 Left Anti Join

Which participants were only in Berlin?

Berlin		Frankfurt
Sol Marroquin		Raven Beatty
Kiera Mcfall		Roxanna Mercier
Raven Beatty		Fanny Denning
Elinore Dees		Lesha Nobles
Wei Lockwood		Wei Lockwood
Donald Eldridge		Gertrudis Fitzpatrick
Claudio Beam		Angelita Packer
Angelita Packer		Beaulah Wenger
Reyna Luke		Malvina Hamer
Beaulah Wenger		Bernita Crutcher
Malvina Hamer		Shiela Anaya
Vivan Keeney		Yolonda Armstead

Figure 162

1.14 Grouping

Folder: Gruppieren, file: PQ -GroupBy:

- Load the table into Power Query via *Data > From Table/Range*.
- In Power Query, select *Transform > Group By*.

Gruppieren nach

Geben Sie die gewünschte Ausgabe und die Spalte an, nach der gruppiert werden soll.

☒ Standard ☐ Weitere

Product

Neuer Spaltenname: Gesamt

Vorgang: Summe

Spalte: Sales

OK Abbrechen

Figure 163

Notes:

- You can also group by multiple columns. First select the columns to group by, and then apply the grouping.
- Advanced lets you group by additional criteria and aggregations.

1.14.1 Group by Multiple Criteria

Folder: Gruppieren, file: PQ_GroupBy_MoreCriteria.xlsx

The following multi-column grouping shows whether a SalesRep sold a product more than once and the total sales amount:

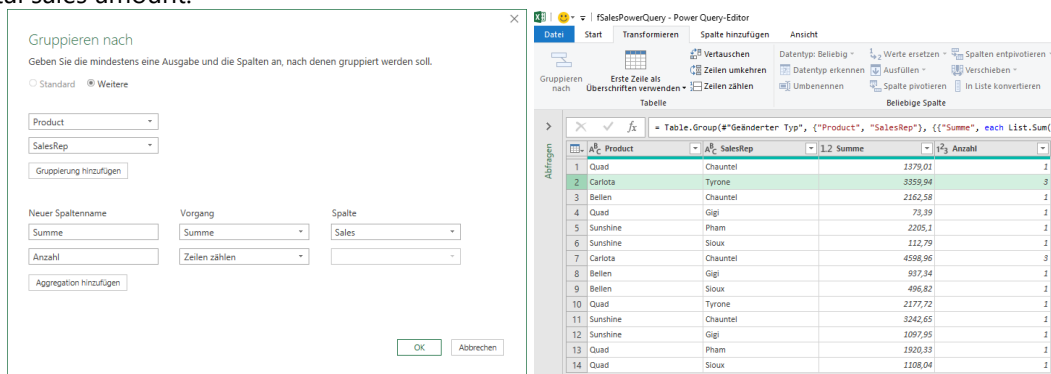


Figure 164

1.14.2 Grouping with Nested Tables

Note: Additional topic outside the scope of the seminar

To make a group's detail data (nested table) available, use Advanced to add an All Rows aggregation:

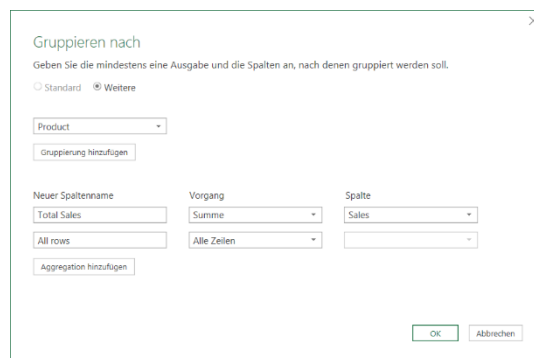


Figure 165

The nested table can now be viewed and used for further analysis:

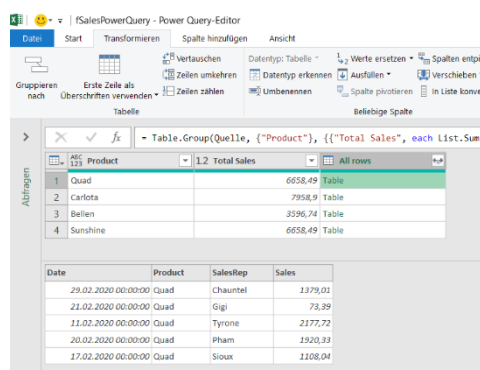


Figure 166

You can now obtain a list of the required values from this nested table. To do so, create a new Custom Column and enter the following formula:

```
=List.Distinct([All rows][SalesRep])
```

This list can then be converted to comma-separated values:

```
=Text.Combine(List.Distinct([All rows][SalesRep]), ", ")
```

1.15 Expression Examples

- `if [Datum]=2 then "a" else "b"`
- `"Jahr:"&Number.ToText(Date.Year([Datum]))`
- `Date.WeekOfYear([Datum],Day.Monday)`

2 Power Query advanced

2.1 Analyzing complex Excel files

2.1.1 Simple average with List()

Folder: Average of a column

Add Column > Custom Column, then:

```
= List.Average(#"Previous Step"[ColumnName])
```

e.g.:

```
let
    Source = Excel.CurrentWorkbook(){[Name="Table1"]}[Content],
    #"Changed Type" = Table.TransformColumnTypes(Source,{{"UMSATZ", Int64.Type}}),
    #"Added Custom Column" = Table.AddColumn(#"Changed Type", "Custom", each
List.Average(#"Changed Type"["UMSATZ"]))
in
    #"Added Custom Column"
```

2.1.2 Add file names and sheet names to a list

Folder: MultipleExcelImport-FileNames

- Import everything from the folder
- *Right-click File Name > Transform > lowercase*
- Filter by .xls
- File Name: *right-click > Split Column, or Transform > Split Column*
 - Tip: *File > Options and settings > Query Options > Data Load > Type Detection*: disable automatic type detection
- Keep only the Content and City columns
- Add Column > Custom Column
 - =Excel.Workbook([Content],true) returns the table names
 - true: first row of each table = header
- Expand the table containing all items
- Filter
 - SalesRep.Name filter by does not begin with: Sheet (case-sensitive!)
 - SalesRep.Kind = Sheet
- Remove unnecessary columns; the following remains:

Name	Tabellenblatt.Name	Tabellenblatt.Data
1 oakland	Fran	Table
2 oakland	Galb	Table
3 oakland	Popli	Table
4 portland	Sioux	Table
5 portland	Chin	Table
6 portland	Tyrone	Table
7 sanfrancisco	Miki	Table
8 sanfrancisco	Han	Table
9 sanfrancisco	Alden	Table
10 seattle	Sindy	Table
11 seattle	Mo	Table
12 seattle	Gigi	Table
13 tacoma	Sue	Table
14 tacoma	Pham	Table
15 tacoma	Shelia	Table

Figure 1

Finally, expand the table column (Data):

	Name	Date	Salesrep	Product	Sales
1	Oakland	08.11.2017	Fran	Aussie Round	220
2	Oakland	24.11.2017	Fran	Tri Fly	84
3	Oakland	18.12.2017	Fran	Tri Fly	210
4	Oakland	18.11.2017	Fran	Bellen	48
5	Oakland	28.12.2016	Fran	Bellen	72
6	Oakland	11.11.2017	Fran	Sunshine	96
7	Oakland	12.09.2017	Fran	Aussie Round	80
8	Oakland	19.11.2016	Fran	Appen	144
9	Oakland	28.11.2017	Fran	Aussie Round	204
10	Oakland	18.12.2016	Fran	Tri Fly	72
11	Oakland	07.11.2017	Fran	Tri Fly	48
12	Oakland	04.11.2017	Fran	Appen	96
13	Oakland	24.10.2017	Fran	Carlota	72
14	Oakland	12.02.2016	Fran	Appen	96
15	Oakland	07.12.2017	Fran	Tri Fly	264
16	Oakland	27.11.2017	Fran	Quid	72
17	Oakland	20.11.2017	Fran	Carlota	12
18	Oakland	02.12.2017	Fran	Appen	72
19	Oakland	01.11.2016	Fran	Quid	72
20	Oakland	18.10.2017	Fran	Appen	132
21	Oakland	18.12.2016	Fran	Quid	208
22	Oakland	19.12.2017	Fran	Aussie Round	72
23	Oakland	07.12.2016	Fran	Sunshine	240
24	Oakland	15.12.2017	Fran	Aussie Round	84
25	Oakland	25.12.2016	Fran	Majestic Beaut	240
26	Oakland	19.12.2016	Fran	Carlota	84
27	Oakland	31.06.2017	Fran	Tri Fly	24
28	Oakland	28.11.2017	Fran	Carlota	108
29	Oakland	12.12.2016	Fran	Sunshine	120
30	Oakland	24.11.2016	Fran	Bellen	24
31	Oakland	24.11.2016	Fran	Quid	204
32	Oakland	18.12.2016	Fran	Tri Fly	72
33	Oakland	18.12.2017	Fran	Bellen	72
34	Oakland	16.04.2016	Fran	Sunshine	96
35	Oakland	07.09.2016	Fran	Sunshine	84
36	Oakland	30.11.2017	Fran	Bellen	72
37	Oakland	04.11.2017	Fran	Majestic Beaut	80

• Figure 2

2.1.3 Delete older duplicates

Folder: Duplicates/DeleteOlderDuplicates

- Load both tables and append them in a new query.
- Sort Date ascending, Brand ascending, and Impressions descending.
- Select the columns from Date through Campaign and remove duplicates

Note: The second duplicate row is always removed, meaning the one with the lower number of impressions.

2.1.4 Create a date list from From..To

Here, individual rows are to be generated from From...To intervals:

	ProductFamli	From	To	ID	ProductFamli	Date	ID
1	Fiction	01.11.2021	04.11.2021	A	Fiction	01.11.2021	A
2	Sport	03.11.2021	08.11.2021	B	Fiction	02.11.2021	A
3	News	02.11.2021	06.11.2021	C	Fiction	03.11.2021	A
4					Fiction	04.11.2021	A
5					Sport	03.11.2021	B
6					Sport	04.11.2021	B
7					Sport	05.11.2021	B
8					Sport	06.11.2021	B
9					Sport	07.11.2021	B
10					Sport	08.11.2021	B
11					News	02.11.2021	C
12					News	03.11.2021	C
13					News	04.11.2021	C
14					News	05.11.2021	C
15					News	06.11.2021	C
16							
17							

Figure 3

This can be achieved using the List function:

`=List.Dates([From],Duration.Days([To]-[From])+1,#duration(1,0,0,0))`

2.1.5 Determine previous periods using a self-join

To calculate with values from previous rows, create two index columns offset by 1 and join them using a Left Outer self-join:

	ABC 123	Year	ABC 123	Quarter	1.2	Quantity	1.3	Index	1.3	Index.1	1.2	Sales.Quantity
1		2014		1		719		0		1		null
2		2014		2		888		1		2		719
3		2014		3		881		2		3		888
4		2014		4		1004		3		4		881
5		2015		1		1150		4		5		1004
6		2015		2		1172		5		6		1150
7		2015		3		1178		6		7		1172
8		2015		4		1445		7		8		1178
9		2016		1		14095		8		9		1445
10		2016		2		21533		9		10		14095
11		2016		3		21495		10		11		21533
12		2016		4		25761		11		12		21495

Figure 4

Zusammenführen

Wählen Sie Tabellen und übereinstimmende Spalten aus, um eine zusammengeführte Tabelle zu erstellen.

Sales

Year	Quarter	Quantity	Index	Index.1
2014	1	719	0	1
2014	2	888	1	2
2014	3	881	2	3
2014	4	1004	3	4
2015	1	1150	4	5

Sales

Year	Quarter	Quantity	Index	Index.1
2014	1	719	0	1
2014	2	888	1	2
2014	3	881	2	3
2014	4	1004	3	4
2015	1	1150	4	5

Join-Art

Linker äußerer Join (alle aus erster, übereinstimmende...)

☒ Fuzzyübereinstimmungen zum Zusammenführen verwenden

> Optionen für Fuzzyübereinstimmung

OK Abbrechen

Figure 5

This means that no match is found in the "joined table" for index 0 (null), while index 1 returns the first value from the "joined table," and so on.

Differences and percentage changes can then be calculated:

Differences:

[Quantity Current] - [Quantity Prev]

Percentage changes:

Number.Round((([Quantity Current]-[Quantity Prev])/[Quantity Prev],3)

2.1.6 Web query for stock data: an introduction to M

ExamplefileLoop_Work

https://edvadamsde-my.sharepoint.com/:x/g/personal/adams_edv-adams_de/EQVNBdHokB1NhtUj_7h9A8kB5OPlDIYmSWCOTJO-sh0kAA?e=IGB2f3

2.2 M-Code

2.2.1 M code structure

Editing:

- Applied Steps
- Formula Bar
- Advanced Editor

TIP: Zoom in the editor: Ctrl + Shift + +/-

Structure:

Let

#"Identifier" = Expression

In

Result



Figure 6

StepName1 is also called an identifier, variable, or transformation step.

Additional steps can be added using `fx`.

Example:

```
let
    Units = 2,
    Turnover = Units*10,
    #"Final result" = Text.Combine({"Turnover total:", Number.ToText(Turnover)})
in
    #"Final result"
```

Comments are written as in DAX: `//` or `/*...*/`

TIP: `Ctrl+K, Ctrl+C` | `Ctrl+K, Ctrl+U` // comment/uncomment a block

2.2.1.1 Identifier notation

Without spaces: `RemovedColumns = ...`

With spaces: `#"Changed Type" = ...`

Figure 7

2.2.2 Help

`=FunctionName`

displays help for a function.

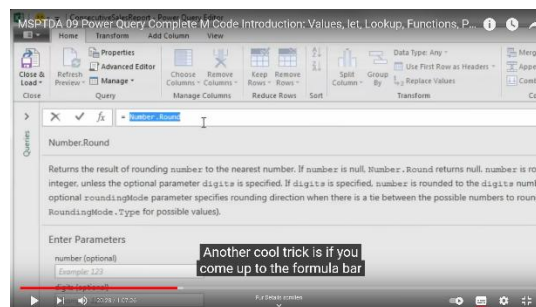


Figure 8

`= #shared`

displays all functions:

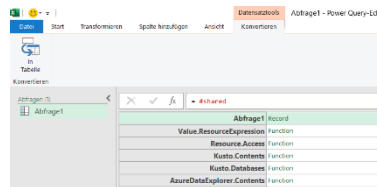


Figure 9

Then simply convert to a table and filter for the desired function.

2.2.3 Data types

Kind	Literal
Null	null
Logical	true false
Number	0 1 -1 1.5 2.3e-5
Time	#time(09,15,00)
Date	#date(2013,02,26)
DateTime	#datetime(2013,02,26, 09,15,00)
DateTimeZone	#datetimezone(2013,02,26, 09,15,00, 09,00)
Duration	#duration(0,1,30,0)
Text	"hello"
Binary	#binary("AQID")
List	{1, 2, 3}
Record	[A = 1, B = 2]
Table	#table({"X", "Y"}, {{0,1},{1,0}})
Function	(x) => x + 1
Type	type { number } type table [A = any, B = text]

Figure 10

2.2.4 Types of values

Values =

Primitive

List

Record

Table

Function

15956.8

Workday

null

true

8/6/2018

	List
1	15956.8
2	4669.8
3	1919.97
4	6311.83

TypeDay	Weekend
TotalSales	4669.8
NumberSales	4

A ¹	TypeDay	1.2 TotalSales	1.2 NumberSales
1	Workday	15956.8	10
2	Weekend	4669.8	4
3	Holiday	1919.97	5
4	Donation	6311.83	4

Queries [3]

f_x EffectiveAnnualRate

f_x TransformFairTableFunction

f_x AssumedAge

Figure 11

Add primitive values:

Click fx and enter a value (text, number, etc.).

Lists

Lists can be nested

and some functions expect nested lists as arguments.

Note: Product Name is a generalized identifier, so quotation marks are not required.

See also: [#table - PowerQuery M | Microsoft Learn](#)

Reference a value (drill down): = Source{0}[Product] => Source/Row/Column



This reference depends on the sort order

This reference is independent of the sort order

Returns: record

Returns: list

Returns: table

2.2.7 each and _

In a row context, the underscore represents a record.

In this example, the Products column is returned from each record:

```
= Table.AddColumn(ToList,"New", each _[Products])
```

The _ is context-dependent; in the following example, it represents the current list. The element at index 0 (Approbation/A) is always returned from the Quali field of that list:

ABC 123	Quali	ABC 123	new
0	List		Approbation
0	List		Approbation
0	List		Approbation
0	List		Approbation
0	List		Approbation
0	List		Approbation
0	List		Approbation
0	List		Approbation
0	List		A

Figure 12

```
= Table.AddColumn("#Grouped Rows","new",each _[Quali]{0})
```

Note: _ is the shorthand notation for a function/parameter passing:

```
= Table.AddColumn("#Grouped Rows","new", (LE)=>LE[Quali]{0})
```

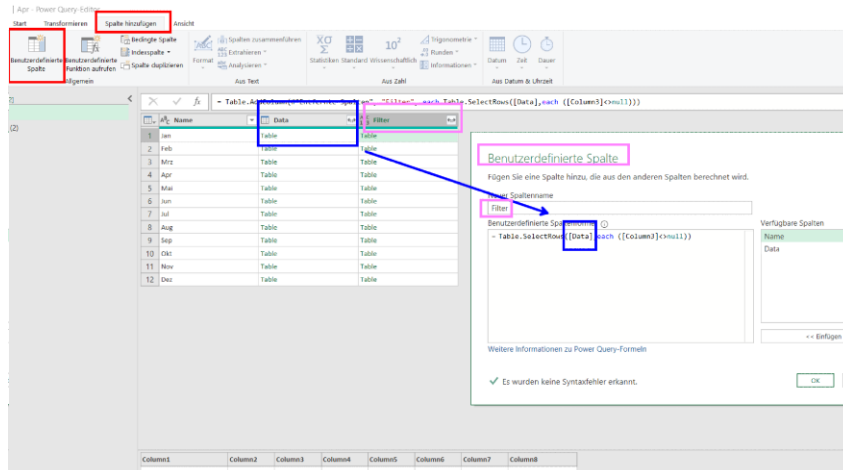
Here, the variable LE represents the current row record. LE is passed to the function, and LE[Quali]{0} retrieves the first item from the Quali list.

2.3 Import fragmented data (workbooks/tables)

2.3.1 One workbook, all sheets/columns identical, pivoted: solution using helper columns

Folder: Workfiles-Advanced\MultiTransform-Unpivot\Data\Reporting2020_SameColumns.xlsx"

- Import a single Excel file
- Keep one table (query) and duplicate it
- Apply transformations in the duplicate and copy the M functions
- Then insert them in a custom column and use the table name from the preceding column



- Finally, delete the helper columns and expand the last column

2.3.2 Multiple workbooks from a folder, all sheets/columns identical, pivoted: solution using helper columns

Folder: Workfiles-Advanced\MultiTransform-Unpivot\Data\Reporting2020_SameColumns.xlsx"

- Import Excel files from a folder; keep only tables with SameColumns
- Keep only the Content column
- Custom column: Excel.Workbook([Content])
- Expand the column
- Keep only the Data column
- The helper-column solution from 7.3.1 can now be used

2.3.3 Multiple workbooks from a folder, all with different columns, pivoted: solution using M and a function

- Import Excel files from a folder; keep only tables with DiffColumns
- Keep only the Content column
- Custom column: Excel.Workbook([Content])
- Expand the column
- Then keep only file names, sheet names, and table data:

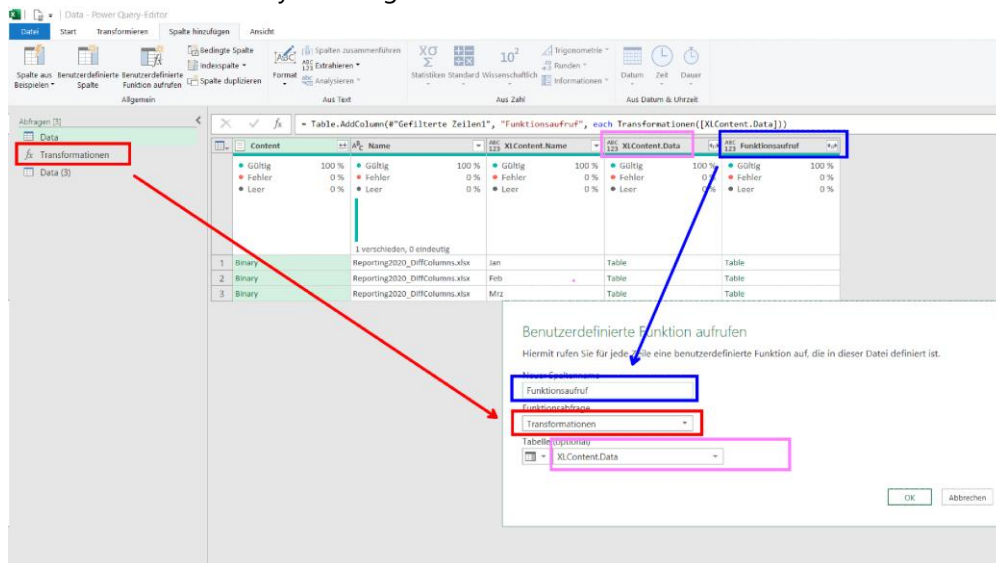
	Name	ABC 123	Name.1	ABC 123	Data
1	Reporting2019_DiffColumns.xlsx		Jan		Table
2	Reporting2019_DiffColumns.xlsx		Feb		Table
3	Reporting2019_DiffColumns.xlsx		Mrz		Table
4	Reporting2019_DiffColumns.xlsx		Apr		Table
5	Reporting2019_DiffColumns.xlsx		Mai		Table
6	Reporting2019_DiffColumns.xlsx		Jun		Table
7	Reporting2019_DiffColumns.xlsx		Jul		Table
8	Reporting2019_DiffColumns.xlsx		Aug		Table
9	Reporting2019_DiffColumns.xlsx		Sep		Table
10	Reporting2019_DiffColumns.xlsx		Okt		Table
11	Reporting2019_DiffColumns.xlsx		Nov		Table
12	Reporting2019_DiffColumns.xlsx		Dez		Table
13	Reporting2020_DiffColumns.xlsx		Jan		Table
14	Reporting2020_DiffColumns.xlsx		Feb		Table
15	Reporting2020_DiffColumns.xlsx		Mrz		Table
16	Reporting2020_DiffColumns.xlsx		Apr		Table
17	Reporting2020_DiffColumns.xlsx		Mai		Table
18	Reporting2020_DiffColumns.xlsx		Jun		Table
19	Reporting2020_DiffColumns.xlsx		Jul		Table
20	Reporting2020_DiffColumns.xlsx		Aug		Table
21	Reporting2020_DiffColumns.xlsx		Sep		Table
22	Reporting2020_DiffColumns.xlsx		Okt		Table
23	Reporting2020_DiffColumns.xlsx		Nov		Table
24	Reporting2020_DiffColumns.xlsx		Dez		Table

- Duplicate the query and display one table from the Data column
- Apply the desired transformations here and copy them in the Advanced Editor
- Create a blank query, turn it into a function, and insert the transformations copied in the previous step between let and in. Then adjust the source names:

```
(Table) =>
let
    Source = Table,

    #"Filtered Rows1" = Table.SelectRows(Source, each ([Column3] <> null) and ([Column1] <> "Grand Total")),
    #"Removed Columns" = Table.RemoveColumns(#"Filtered Rows1", {"Column8"}),
    #"Promoted Headers" = Table.PromoteHeaders(#"Removed Columns", [PromoteAllScalars=true]),
    #"Unpivoted Columns" = Table.UnpivotOtherColumns(#"Promoted Headers", {"Name", "XLContent.Name", "Data (2).Row Labels"}, "Attribute", "Value")
in
    #"Unpivoted Columns"
```

- Create a new column by invoking a custom function:



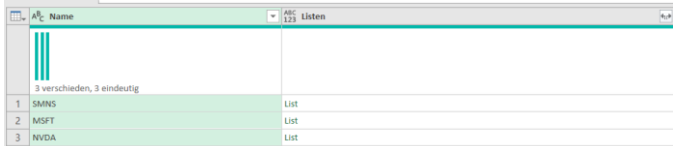
- Remove unnecessary columns

ABC	Name	ABC	XLContent.Name	ABC	Funktionsaufruf
	- Gültig 100 % - Fehler 0 % - Leer 0 %		- Gültig 100 % - Fehler 0 % - Leer 0 %		- Gültig 100 % - Fehler 0 % - Leer 0 %
	1 verschieden, 0 eindeutig				
1	Reporting2020_DiffColumns.xlsx	Jan		Table	
2	Reporting2020_DiffColumns.xlsx	Feb		Table	
3	Reporting2020_DiffColumns.xlsx	Mrz		Table	

- Expand the data in the function-call column

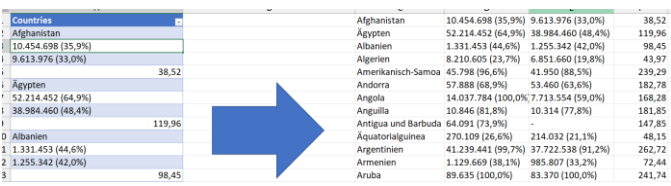
2.4 Useful M functions

Excel.CurrentWorkbook()	Returns Excel objects except worksheets
Folder.Files(FilePath)	Imports all files from a folder
Table.ColumnNames(table)	Returns all column names of a table as a list
Table.ToList(Table)	Creates a list from a table
Table.FromList(List)	Creates a table from a list
Table.ColumnNames	Returns column names
Table.SelectColumns	Selects columns
Table.SelectRows	Filters
Table.RenameColumns	Table.RenameColumns(Table, { {"CustomerNum", "CustomerID"}, {"PhoneNum", "Phone"} })
List.Zip	List 1: {1, 2, 3} List 2: {'a', 'b', 'c'} Result: {1, 'a'}, {2, 'b'}, {3, 'c'}
Table.RenameColumns AND List.Zip	Table.RenameColumns(Table, List.Zip({CurrentColumnNames, NewColumnNames}))

<code>Table.Group(table as table, key as any, aggregatedColumns as list, optional groupKind as nullable number, optional comparer as nullable function) as table</code>	Groups tables key: By which columns? aggregatedColumns: Calculated columns groupKind: with GroupKind.Local, only consecutive rows are treated as a group rather than the entire table.												
<code>Table.ExpandList</code>	Expands all lists in a list column <code>Table.ExpandListColumn (Table,ColumnName)</code>  <table><tr><th></th><th>Name</th><th>Listen</th></tr><tr><td>1</td><td>SMNS</td><td>List</td></tr><tr><td>2</td><td>MSFT</td><td>List</td></tr><tr><td>3</td><td>NVDA</td><td>List</td></tr></table>		Name	Listen	1	SMNS	List	2	MSFT	List	3	NVDA	List
	Name	Listen											
1	SMNS	List											
2	MSFT	List											
3	NVDA	List											

List.split

Transforms a number of rows into columns



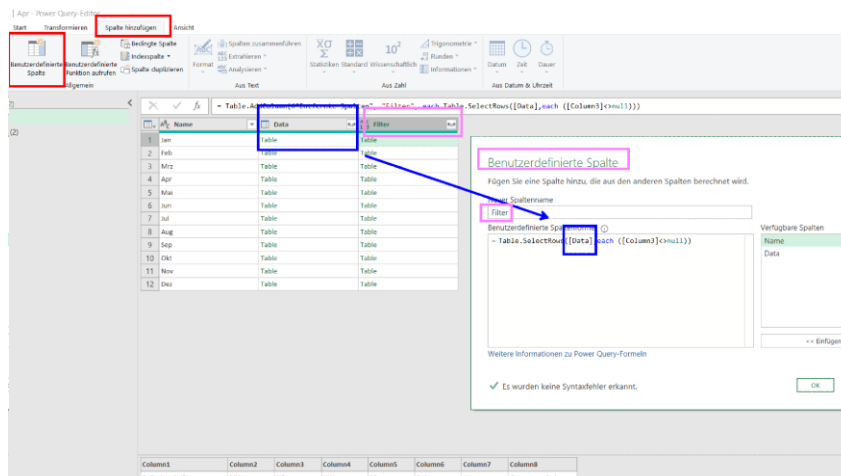
Example:
`result=#table(4,List.Split(Source[Countries],4))`
`result=#table(4,... => creates 4 columns`

2.5 Practical imports

2.5.1 One workbook, all sheets/columns identical, pivoted: solution using helper columns

Folder: Workfiles-Advanced\MultiTransform-Unpivot\Data\Reporting2020_SameColumns.xlsx"

- Import a single Excel file
- Keep one table (query) and duplicate it
- Apply transformations in the duplicate and copy the M functions
- Then insert them in a custom column and use the table name from the preceding column



- Finally, delete the helper columns and expand the last column

2.5.2 Multiple workbooks from a folder, all sheets/columns identical, pivoted: solution using helper columns

Folder: Workfiles-Advanced\MultiTransform-Unpivot\Data\Reporting2020_SameColumns.xlsx"

- Import Excel files from a folder; keep only tables with SameColumns
- Keep only the Content column
- Custom column: Excel.Workbook([Content])
- Expand the column
- Keep only the Data column

2.6 Functions and parameters

2.6.1 Define functions

Folder: Functions

Home > New Source > Other Sources > Blank Query, then Home > Advanced Editor:

```
let
    CalculateTurnover=(Units, Price)=>Number.Round(Units*Price, 2)
in
```

Use a function:

Add Column > Invoke Custom Function:



Figure 13

2.6.2 Define parameter queries (transform many files at once)

Folder: Parameter Queries

To transform and combine multiple files at once, first create a dedicated parameter query. It receives each table as input and applies the same transformations to all tables.

Create a parameter query:

- Open any one of the files to be transformed and transform it as needed, e.g. (it is best to remove the automatic Changed Type step):
 - Transform > Transpose
 - Home > Use First Row as Headers
 - Transform > Unpivot Columns
- In Advanced Editor, remove the first Source=... step
- Create a new parameter, e.g. TableName, and use it in place of the table name:

Erweiterter Editor

Bonny

```
(TableName)=>
let
    parameter = TableName
    #"Transponierte Tabelle" = Table.Transpose(TableName),
    #"Höher gestufte Header" = Table.PromoteHeaders(#"Transponierte Tabelle", [PromoteAllScalars=true]),
    #"Entpivotierte Spalten" = Table.UnpivotOtherColumns(#"Höher gestufte Header", {"Date", "Product/Fair"}, "Attribut", "Wert")
in
    #"Entpivotierte Spalten"
```

Figure 14

- Save the query, e.g. as TransformTable

Use the parameter query

- Import the folder and transform the data
- Keep only Content
- Create a custom column, e.g. GetWorkbookContent, and retrieve the workbook contents:

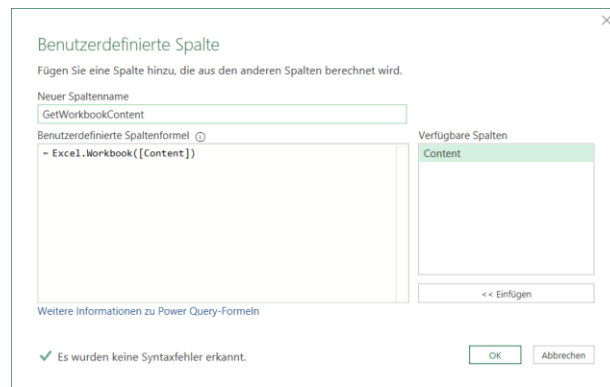


Figure 15

- Remove the Content (Binary) column
- Expand the table and keep only the Data column, because it contains the worksheet contents
- *Add Column > Invoke Custom Function:*

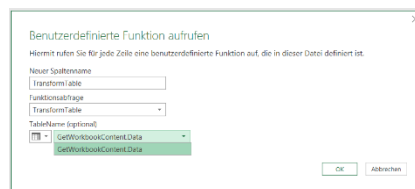


Figure 16

- Now pass GetWorkbookContent.Data to our new TransformTable function.
- Finally, remove the first column and expand the data

2.6.3 Pass data from Excel to a query

Folder: Parameter Queries

In this example, the folder import path is to be passed from Excel to the parameter query from section 7.6.2:

- Excel: create a table and enter the path there
- Name the table, e.g. tPath
- Create a new query from this table
- Drill down into the first cell:

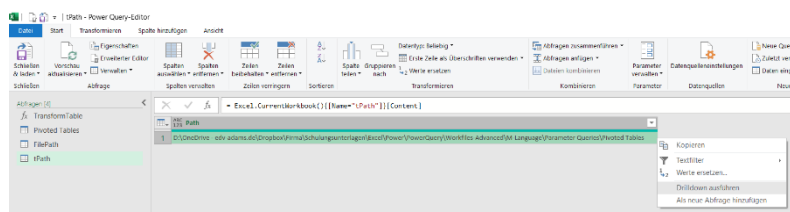


Figure 17

- In Advanced Editor, insert a new step that passes the previous step to Folder.Files:

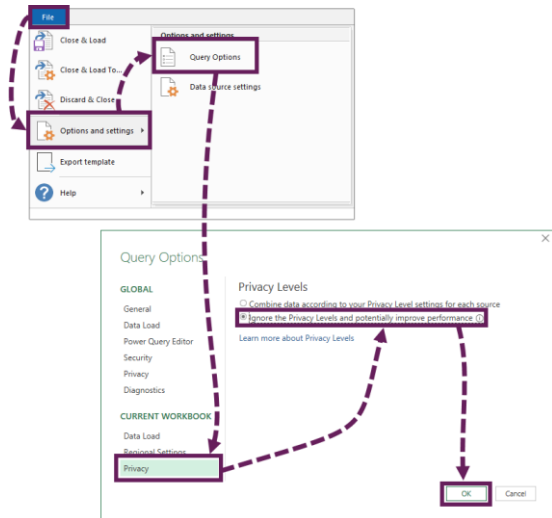
```
let
    Source = Excel.CurrentWorkbook(){[Name="tPath"]}[Content],
    IPPath = Source{0}[Path], //Get row 0, column Path from table tPath
    FilePath=Folder.Files(IPPath)
in
    FilePath
```

Finally, pass the result of this query to the desired query as the "file path":

```
let
    Source = FilePath,
    ...
    ...
```

Note: retrieve Excel workbook objects: Excel.Workbook(), Excel.CurrentWorkbook()

2.6.4 If a Formula Firewall error occurs



<https://exceloffthegrid.com/formula-firewall-in-power-query/>

2.7 Techniques

2.7.1 Parameter

DB: AdventureWorks_Database.xlsx

Create a drop-down list to serve as the filter criterion for a query (Territories):

- Import the Territories table
- **TIP:** Right-click a column > Add as New Query creates a list from that column. Apply this to the Country column here.
- Remove duplicates from this list
- Home > Manage Parameters > New Parameter:

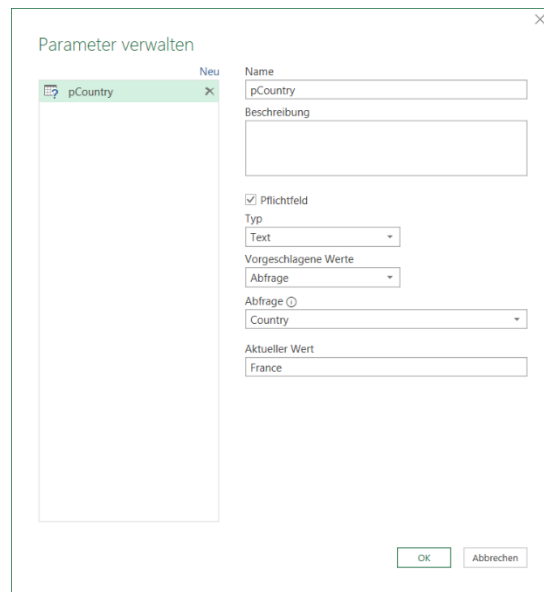


Figure 18

- Open the target query in which the parameter will be used as a filter criterion and apply any filter to Country
- Now change the M code from
`= Table.SelectRows(#"Changed Type", each ([Country] = "Australia"))`
to:
`= Table.SelectRows(#"Changed Type", each ([Country] = pCountry))`
(pCountry is the name of the parameter)

Use parameters in Power Query:

Home > Manage Parameters > Manage Parameters:

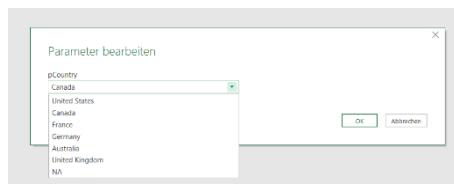


Figure 19

Finally, refresh the filtered query.

Use parameters in Power BI Desktop:

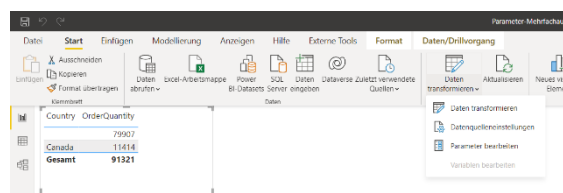


Figure 20

2.8 Miscellaneous

2.8.1 Insert rows

#"Insert Row" = Table.InsertRows("#Added Custom Column", 0, {[From = 1, To = "data2", Currency="AA", Custom="dd"]})

2.8.2 Calculate the previous day

Date.AddDays([Order Intake],-1) (Only for continuous data with no gaps!)

2.8.3 Datediff

Duration.Days(Duration.From([To]-[From]))

2.8.4 List data from...to in rows

Folder: Time interval FROM TO

- Load the table into a query
- Remove the time from the date
- Insert a new column: =List.Dates([From],Duration.Days([To]-[From])+1,#duration(1,0,0,0))

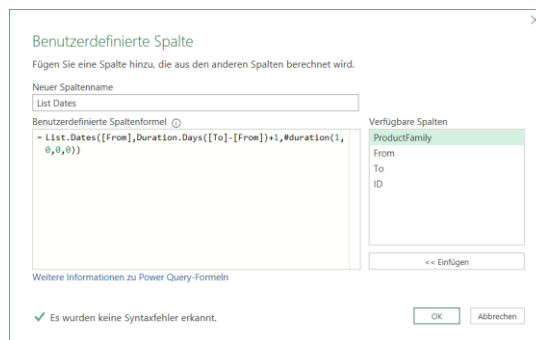


Figure 21

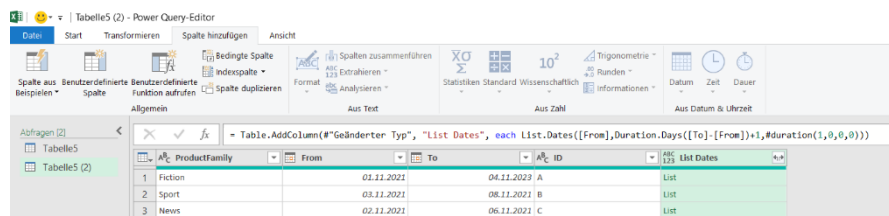


Figure 22

- Expand the List Dates column into rows using the double-arrow icon
- Remove the From/To columns:

source
ttings
Sources

Advanced Editor

testFunction1

```
(totalLoops as number, Loop as number, Value as number) =>  
let  
  CurrentLoop = Loop + 1,  
  CurrentValue = Value * 2,  
  output =  
    if CurrentLoop >= totalLoops  
    then CurrentValue  
    else @testFunction1(totalLoops, CurrentLoop, CurrentValue)  
in  
  output
```

	ProductFamily	ID	Date
1	Fiction	A	01.11.2021
2	Fiction	A	02.11.2021
3	Fiction	A	03.11.2021
4	Fiction	A	04.11.2021
5	Sport	B	03.11.2021
6	Sport	B	04.11.2021
7	Sport	B	05.11.2021
8	Sport	B	06.11.2021
9	Sport	B	07.11.2021
10	Sport	B	08.11.2021
11	News	C	02.11.2021
12	News	C	03.11.2021
13	News	C	04.11.2021
14	News	C	05.11.2021
15	News	C	06.11.2021

Figure 23

3 Power BI Desktop

3.1 Data Imports

Folder: MultiCSV-to-Dashboard/Start

This chapter begins by importing and modeling all required data.

3.1.1 Multi-File CSV Import

To import multiple files from a folder, select *Home* > *Get data* > *More*, and then choose Folder.

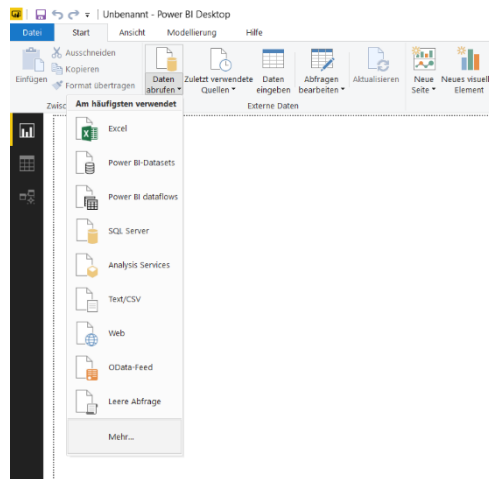


Figure 36

Select the folder that contains the files to import, and then select Transform Data.

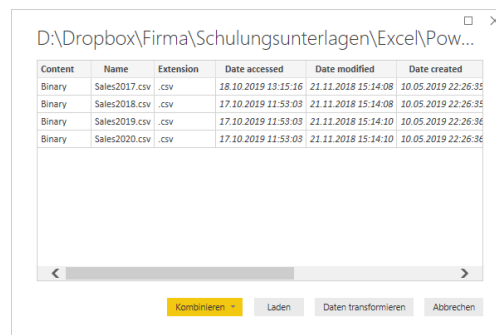


Figure 37

This opens the integrated Power Query Editor. You can also open it later by selecting *Home* > *Transform data*.



Now use the context menu to remove all columns except the first. Then select Combine > Combine Files to combine all source files into a single table.

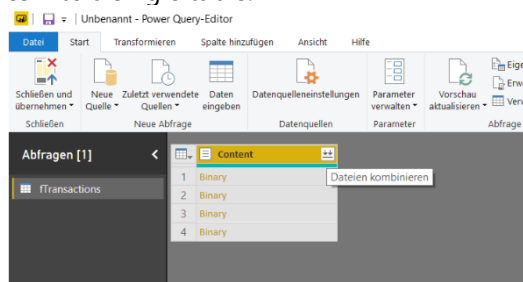


Figure 38

3.1.1.1 Transform Values

It is now advisable to convert the date column to a standard date format. For the currency column, right-click the column header and select Change Type > Using Locale. Choose the appropriate locale, and then round the values to two decimal places.

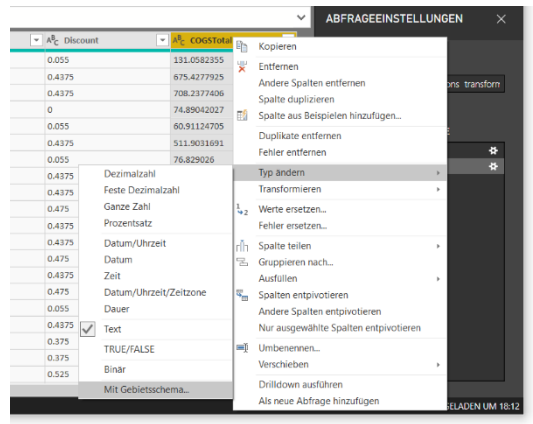


Figure 39

Select *Home > Close & Apply* to apply all changes in Power BI.

Note: If the revenue figures were converted incorrectly (Figure 40), delete the Changed Type step. Then right-click the column header and select Change Type > Using Locale to convert the values with the correct locale.

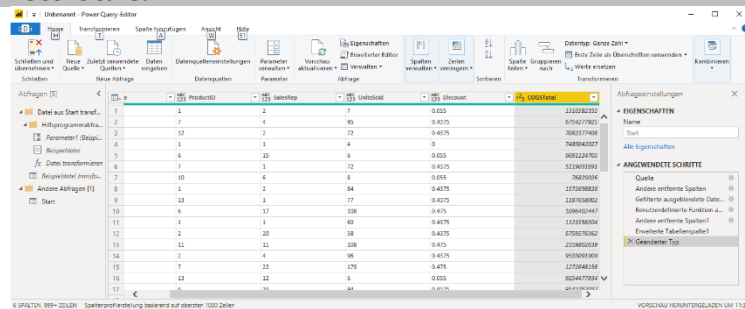


Figure 40

3.1.2 Interim Report

For example, the data prepared so far can be used to create a report like this:

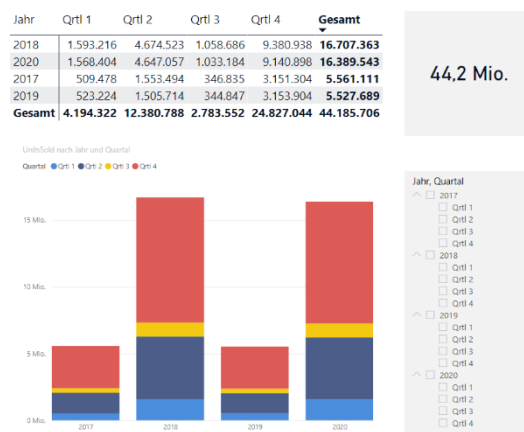


Figure 41

3.1.3 Publish Reports

First, create a free Microsoft account if necessary. When you start Power BI, sign in with your account. You can then select *Home* > *Publish* to upload your project to powerbi.com.

Then sign in to powerbi.com and view your project online. Interactions such as filtering and drill-down are also available within the project.

Using Share, you can share a report with other Power BI Pro users. They can then view the report either online or in the Power BI app on a mobile device.

Note: To optimize the report for mobile devices, select *View* > *Mobile layout*.

3.1.4 Publish Reports to SharePoint

To do this, sign in to the Power BI service and select Share:

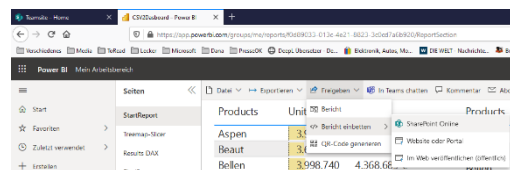


Figure 42

In the Power BI service, open the report, and then select *File* > *Embed report* > *SharePoint Online*. Copy the report URL provided for SharePoint Online.

In SharePoint:

- Enable page editing
- Insert a new page section
- Add the Power BI web part to the SharePoint page.
- Select Add report, and then paste the Power BI report URL into the Power BI report link box.

3.1.5 Excel Import with Relationships

Folder: Excel-CSV-Relation /016-MSPTDA-Excel.xlsx and fSales-2017.csv

First, import all four Excel tables and the CSV file by selecting *Home* > *Get data*. In the Navigator window, select Transform Data to open the data in Power Query Editor.

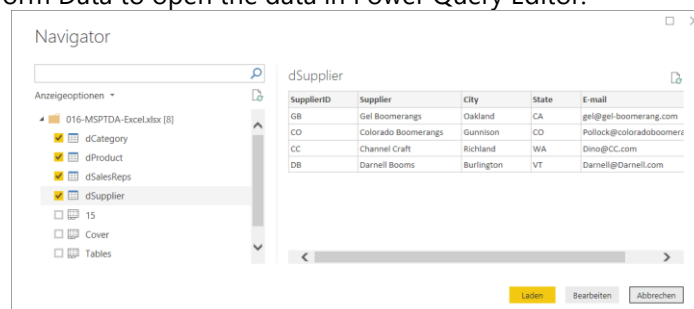


Figure 43

In Power Query Editor, select *Home* > *Merge Queries*, and then choose the tables to link: dProduct and dCategory. Select the matching CategoryID column in each table.

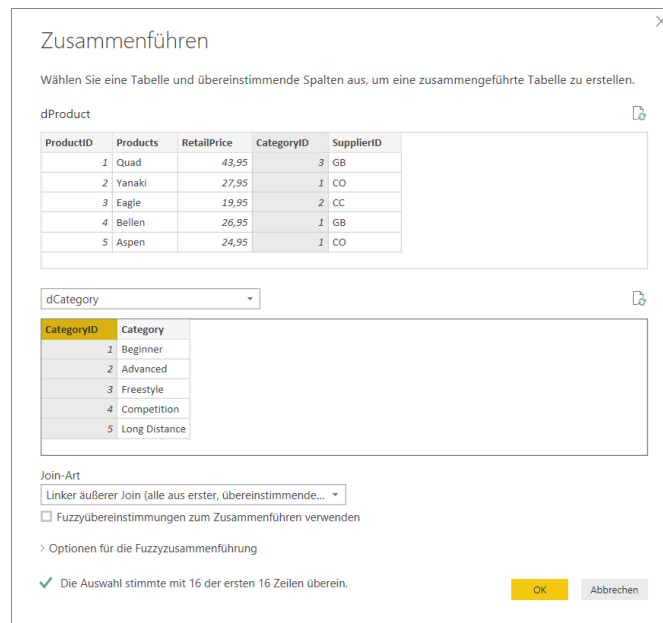


Figure 44

The merged result can now be expanded using this icon. Retain only the columns you need, in this case Category:

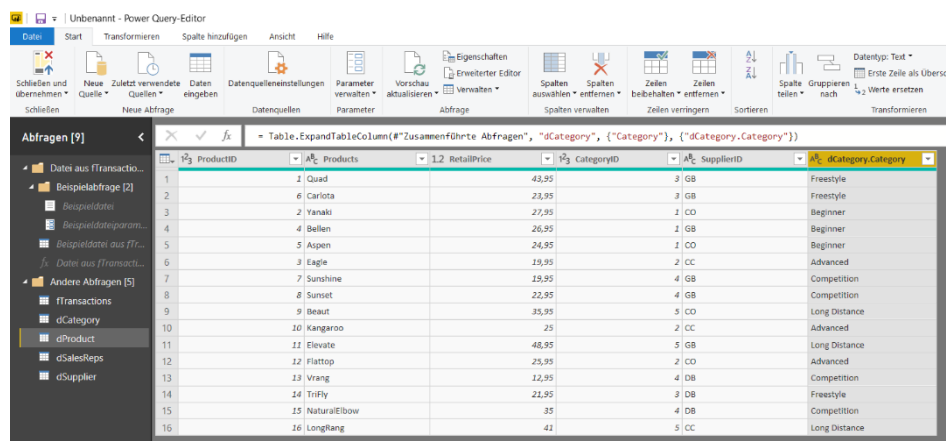


Figure 45

Select *Home > Close & Apply* to apply all changes in Power BI. Use the Model button on the left to review and edit table relationships. Power BI has created the relationships automatically:

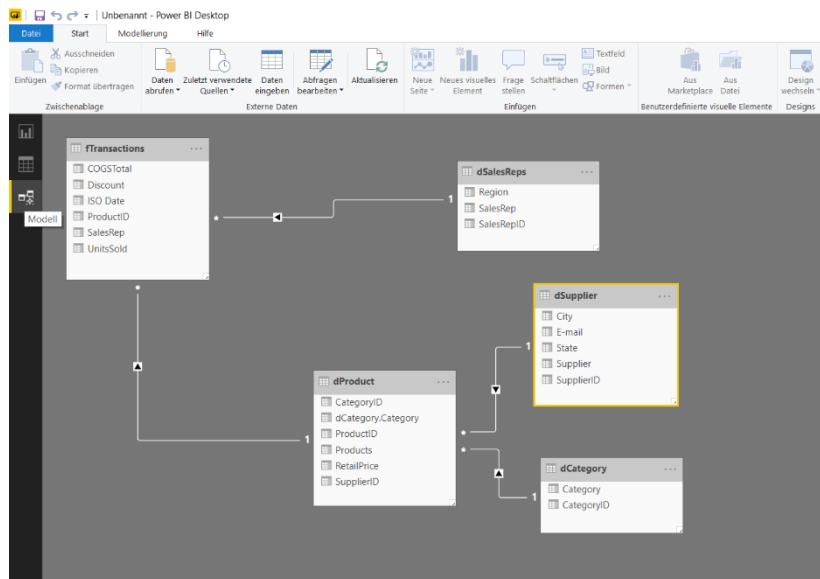


Figure 46

Note: If the relationships were not created automatically, create them manually in Model view by dragging a key field from one table onto the matching key field in another table.

3.1.6 Import SharePoint Data

In Power BI Desktop, select *Home > Get data > More*, search for SharePoint folder, select the connector, and then select Connect.

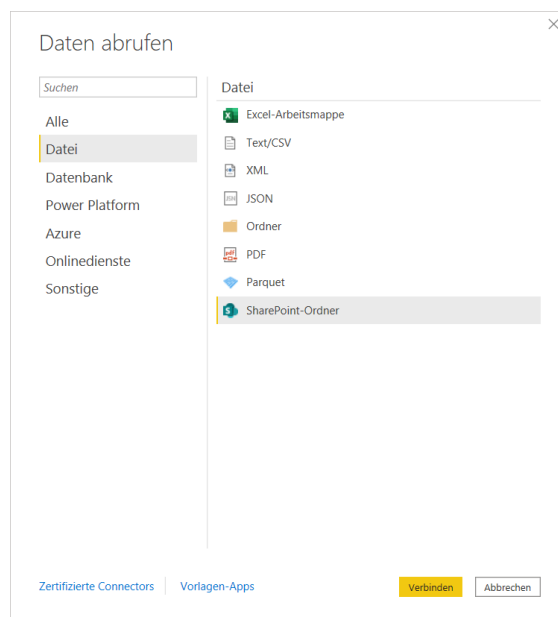


Figure 47

Copy the SharePoint site address from the browser and paste it, for example:

<https://edvadamsde.sharepoint.com/sites/Seminare>

Then transform the data, filter for the required file(s) in Power Query, and create a report with visuals based on those file(s).

https://edvadsde.sharepoint.com/sites/Seminare

Content	Name	Extension	Date accessed	Date modified	Date created	Attributes
Binary	Schulungsinhalte_Office_GrpfM_MCom_2018.docx	.docx	null	09.05.2021 21:31:41	09.05.2021 21:31:41	Record https://edvadsde.sharepoint.com/sites/Seminare/_layouts/15/onedoc.aspx?d=09052021213141&e=.docx
Binary	VBAlinhalt-Aufbau-Ausführlich.docx	.docx	null	09.05.2021 21:31:41	09.05.2021 21:31:41	Record https://edvadsde.sharepoint.com/sites/Seminare/_layouts/15/onedoc.aspx?d=09052021213141&e=.docx
Binary	Schulungsinhalte_Office_Word_Excel_PPT.docx	.docx	null	09.05.2021 21:31:42	09.05.2021 21:31:42	Record https://edvadsde.sharepoint.com/sites/Seminare/_layouts/15/onedoc.aspx?d=09052021213142&e=.docx
Binary	VBAlinhalt-Grundlagen-Ausführlich.docx	.docx	null	09.05.2021 21:31:43	09.05.2021 21:31:43	Record https://edvadsde.sharepoint.com/sites/Seminare/_layouts/15/onedoc.aspx?d=09052021213143&e=.docx
Binary	Was schließt du.xlsx	.xlsx	null	11.06.2021 12:26:26	09.05.2021 22:03:14	Record https://edvadsde.sharepoint.com/sites/Seminare/_layouts/15/onedoc.aspx?d=09052021220314&e=.xlsx
Binary	Sales-2017.csv	.csv	null	26.05.2021 18:22:59	26.05.2021 18:14:41	Record https://edvadsde.sharepoint.com/sites/Seminare/_layouts/15/onedoc.aspx?d=09052021181441&e=.csv
Binary	Excel-DimensionenData.xlsx	.xlsx	null	26.05.2021 18:24:17	26.05.2021 18:14:17	Record https://edvadsde.sharepoint.com/sites/Seminare/_layouts/15/onedoc.aspx?d=09052021181417&e=.xlsx
Binary	dProduct.xlsx	.xlsx	null	05.10.2021 09:21:14	05.10.2021 09:21:14	Record https://edvadsde.sharepoint.com/sites/Seminare/_layouts/15/onedoc.aspx?d=05102021092114&e=.xlsx
Binary	Angebot-Semi - 2.mht	.mht	null	09.05.2021 21:30:55	09.05.2021 21:27:46	Record https://edvadsde.sharepoint.com/sites/Seminare/_layouts/15/onedoc.aspx?d=09052021212746&e=.mht
Binary	Seite ohne Titel - 2.mht	.mht	null	09.05.2021 21:56:34	09.05.2021 21:56:08	Record https://edvadsde.sharepoint.com/sites/Seminare/_layouts/15/onedoc.aspx?d=09052021215608&e=.mht

Figure 48

Once the required report has been created, upload it to the Power BI service as usual via *Home/Publish*.

3.2 Create a Simple Report

Next, create the following report:

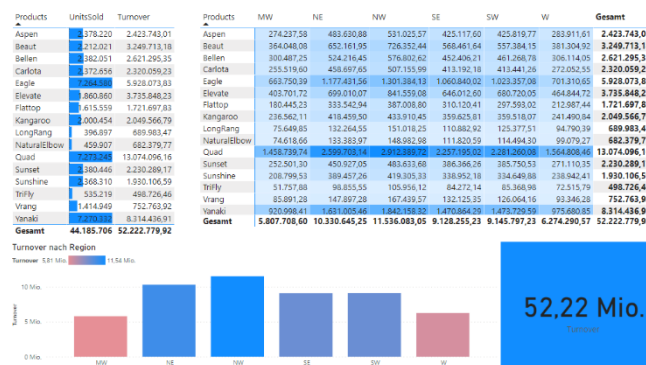


Figure 49

Key features of the report:

- Drill-down in the chart
- Conditional formatting in all elements (including the card showing 52 million)
- Configurable interactions with other visuals

3.2.1.1 Drill-Down in the Chart

For example, to avoid displaying colored segments in a stacked chart and instead enable drill-down to more detail by clicking, drag the detailed information to the Axis field:

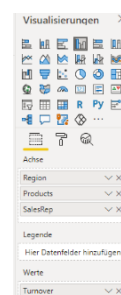


Figure 50

3.2.2 Create Measures

To calculate revenue in the table on the left side of the report, we need a measure that performs this calculation:

```
Turnover = SUMX(fTrans;fTrans[UnitsSold]*(1-fTrans[Discount])*RELATED(dProduct[RetailPrice]))
```

The expression within SUMX() calculates each individual revenue amount:

```
fTrans[UnitsSold]*(1-fTrans[Discount])*RELATED(dProduct[RetailPrice])
```

Because the product price is not stored in the fTrans table, RELATED() must retrieve it from the dProduct table.

3.2.3 Insert Visuals

Table

First, insert the Table visual:

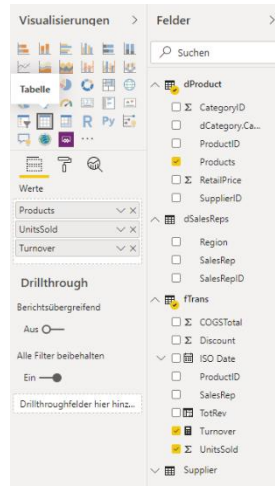


Figure 51

Then add the required fields to the visual: Products, UnitsSold, and Turnover (Measure). Use the chevron next to UnitsSold to add data bars as conditional formatting.

Matrix

This is equivalent to a PivotTable for displaying multidimensional data. Insert the visual and drag the required fields (Product, Region, Turnover) to the appropriate areas:

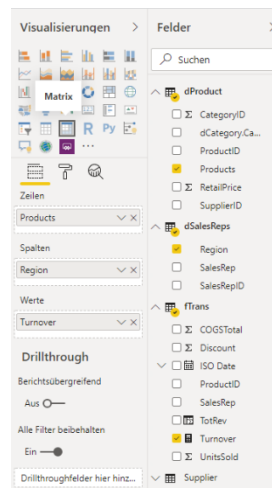


Figure 52

Using the Conditional Formatting chevron, enable a background color for revenue (Turnover) to add a heat map to the table.

Chart

Finally, insert a stacked column chart and drag the required fields to their respective areas:

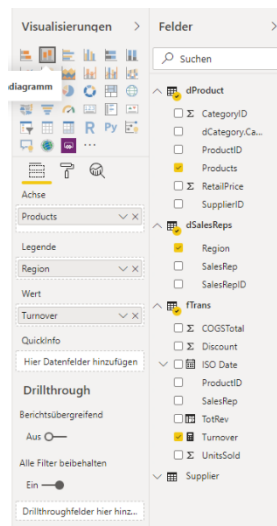


Figure 53

TIP: To adjust the sort order on the x-axis, select the three-dot menu in the upper-right corner of the chart, then select Sort by.

3.2.4 Filter Visuals

To filter within a visual, first select it, then choose the appropriate filter on the right side of the screen:

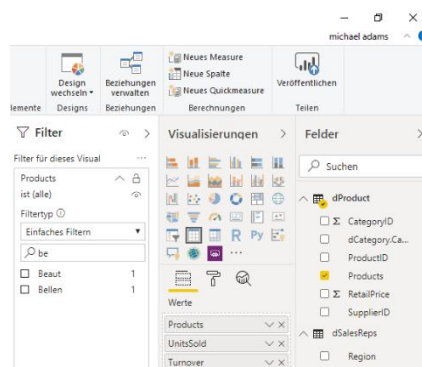


Figure 54

3.3 Create Tables Using a Function

Folder/file: Result from Chapter 4.1

Create a calculated table by selecting *Modeling* > *New table*, for example:

DateTable = CALENDAR("1.1.18";"31.12.18") – Creates a table with a start and end date

Tip: The start and end dates can also be derived from an existing date column:

Date = CALENDAR(MIN(fSales[ISO Date]);MAX(fSales[ISO Date]))

3.3.1 Insert New Columns

Add calculated columns by selecting *Modeling* > *New column*, for example:

- MonthNumber = MONTH(DateTable[Date])
- MonthText = FORMAT('dDate'[Date];"MMM")
- Year = YEAR('dDate'[Date])
- Quarter = ROUNDUP('dDate'[MonthNumber]/3;0)

3.3.1.1 Region as a Grouping Column

Regions should be reduced to N/W/S/E:

```
Region Simple = LEFT(dSalesReps[Region];1)
```

3.3.1.2 Account for the Fiscal Year

Assume your company's fiscal year starts on April 1. The quarter numbers must therefore be shifted. In this example, January through March would be the fourth quarter, and the first quarter would begin in April:

```
FiscalQuarter = IF(dDate[Quarter]=1;4;dDate[Quarter]-1)
```

The fiscal year is calculated in a similar way:

```
FiscalYear = IF(dDate[Quarter]=1;dDate[Year]-1;dDate[Year])
```

Fiscal period (Q1-17, Q1-18, etc.)

- Create: FiscalPeriod = "Q"&dDate[FiscalQuarter]&"-"&RIGHT(dDate[FiscalYear];2)
- Helper column for sorting: FiscalPeriodSort = dDate[FiscalYear]*10+dDate[FiscalQuarter]

The data is sorted by quarter here

To sort FiscalPeriod by the helper column, select the FiscalPeriod column in Data view. Then select *Column tools > Sort by column* and choose the helper sort column.

3.4 Types of Table Relationships

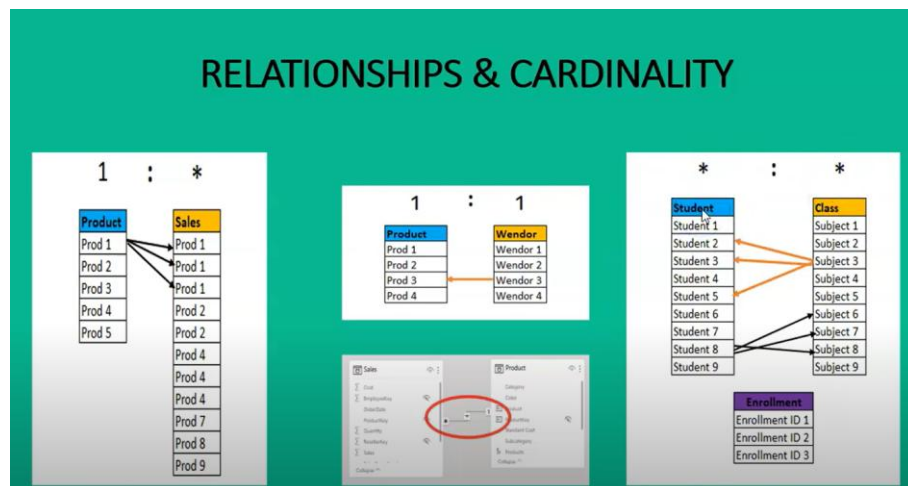


Figure 55

3.5 Master Date Table

To ensure that time-intelligence functions work correctly, select the date table in the Fields pane. Then select *Table tools > Mark as date table > Mark as date table*.

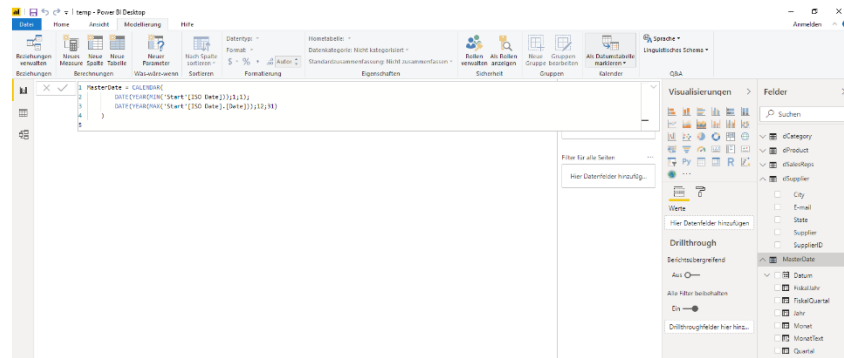


Figure 56

Next, link the date table to the fact table (fTransactions) in the relationships area:

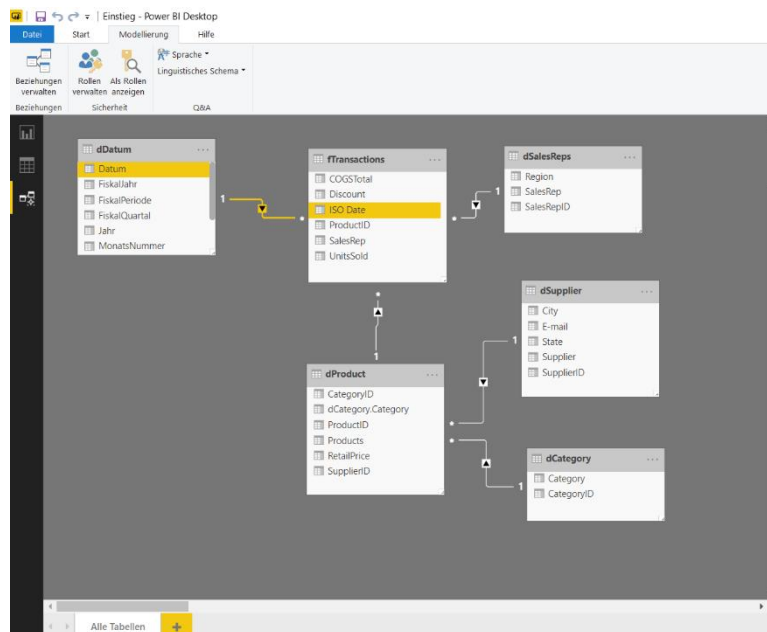


Figure 57

Finally, it is advisable to right-click and remove any unnecessary columns from the report view:

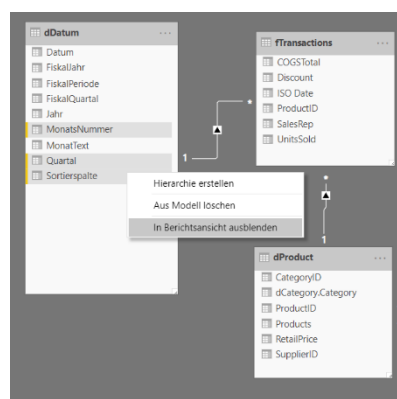


Figure 58

3.6 Interim Report with Treemap and Slicer

In this report, the treemap shows the distribution of revenue by region and salesperson. The slicer containing the fiscal year filters all visuals in the report.

We also need the following measures:

- $\text{PrevPeriodTurnover} = \text{CALCULATE}(\text{Turnover}; \text{SAMEPERIODLASTYEAR}(\text{dDate}[\text{Datum}]))$
- $\text{Change\%} = \text{DIVIDE}([\text{Turnover}] - [\text{PrevPeriodTurnover}]; [\text{PrevPeriodTurnover}]; 0)$

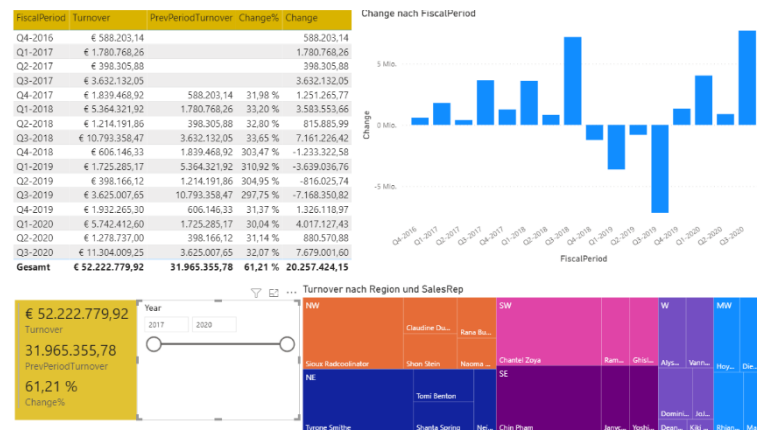


Figure 59

3.6.1 Filter for All Pages

If you drag fields into this filter area, the filter applies to every page in your file:

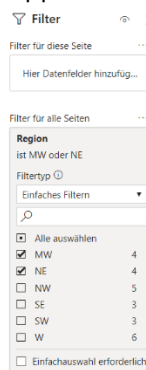


Figure 60

3.7 Create DAX Measures (Aggregates)

Create explicit measures by selecting *Modeling > New measure*, for example:

- Total Units:
 $\text{TotalUnits} = \text{SUM}(\text{fTrans}[\text{UnitsSold}])$
- Total COGS:
 $\text{TotalCOGS} = \text{SUM}(\text{fTrans}[\text{COGSTotal}])$
- Total Revenue:
 $\text{TotalRevenue} = \text{SUMX}(\text{fTrans}; \text{RELATED}(\text{dProduct}[\text{RetailPrice}]) * \text{fTrans}[\text{UnitsSold}] * (1 - \text{fTrans}[\text{Discount}]))$
- Total Gross Profit
 $\text{TotalGrossProfit} = [\text{TotalRevenue}] - [\text{TotalCOGS}]$
- % Gross Profit
 $\text{\%GrossProfit} = \text{DIVIDE}([\text{TotalGrossProfit}]; [\text{TotalRevenue}])$

- Average daily revenue:
 $\text{DailyAvgRevenue} = \text{AVERAGEX}(\text{fTrans}; \text{fTrans}[\text{UnitsSold}] * (1 - \text{fTrans}[\text{Discount}] * \text{RELATED}(\text{dProduct}[\text{RetailPrice}]))$

Select these measures using their checkboxes to obtain:

Products	TotalUnits	TotalCOGS	TotalRevenue	TotalGrossProfit	%GrossProfit
Aspen	5396613	€ 49.634.131,61	73.178.453,48	23.544.321,87	32,17 %
Beaut	4958560	€ 66.465.864,10	96.900.202,32	30.434.338,22	31,41 %
Belien	5418324	€ 56.598.810,92	79.313.910,84	22.715.099,92	28,64 %
Carlota	5382974	€ 54.647.276,53	70.051.940,28	15.404.663,75	21,99 %
Eagle	16442391	€ 147.080.721,22	178.567.634,07	31.486.912,85	17,63 %
Elevate	4162114	€ 90.942.292,59	111.229.212,15	20.286.919,56	18,24 %
FlatTop	3647668	€ 35.880.779,76	51.713.952,84	15.833.173,08	30,62 %
Kangaroo	4521036	€ 43.418.349,05	61.609.867,56	18.191.518,51	29,53 %
LongRang	798488	€ 12.315.982,63	18.538.270,33	6.222.287,71	33,56 %
NaturalElbow	922782	€ 13.726.003,59	18.279.237,64	4.553.234,05	24,91 %
Quad	16434763	€ 307.701.577,04	393.195.511,48	85.493.934,44	21,74 %
Sunset	5397626	€ 48.293.067,15	67.314.410,39	19.021.343,24	28,26 %
Sunshine	5399398	€ 38.388.457,60	58.551.836,99	20.163.379,39	34,44 %
TriFly	1063876	€ 8.465.064,29	13.209.596,32	4.744.532,03	35,92 %
Virang	3187668	€ 36.728.011,49	22.557.157,12	-14.170.854,37	-62,82 %
Vanaki	16433913	€ 163.193.753,18	250.117.297,39	86.923.544,21	34,75 %
Gesamt	99568214	€ 1.173.480.142,75	1.564.328.491,21	390.848.348,47	24,99 %

Figure 61

Note on filter context: The results are calculated for each item; that is, the relevant value is calculated for Aspen, Beaut, and so on. The row-level filter context is therefore taken from the Products column.

3.8 Create Dashboards

As a general rule, create reports in four steps:

- Select a visualization
- Select measures (see Chapter 4.7) using their checkboxes
- Select fields (table columns)
- Configure formatting

3.8.1 Matrix (PivotTable)

Use the Matrix visual to insert a PivotTable into the report. Then drag the required fields to the Rows/Columns/Values containers.

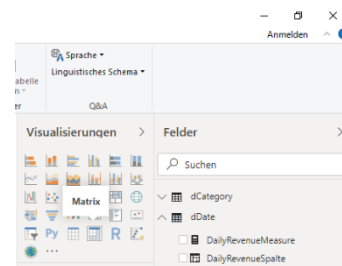


Figure 62

The drop-down arrow for the value field can also be used to display percentage shares.

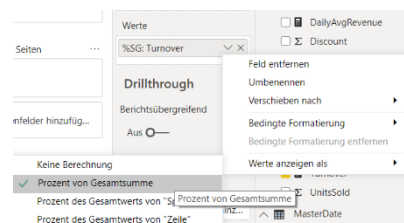


Figure 63

You can then format the visual. Open the Format visual pane by selecting the paintbrush icon, expand the Grid section, and adjust the text size.

To add data bars, open the drop-down menu for the relevant field in the Values well, select Conditional formatting, and then select Data bars.

Result:

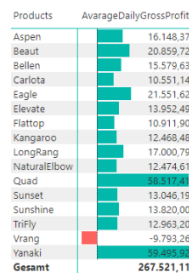


Figure 64

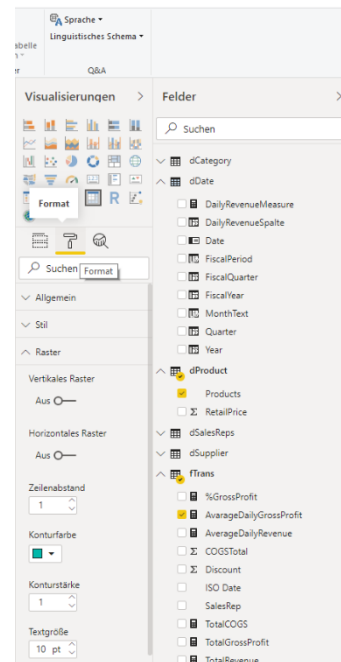


Figure 65

3.8.2 Column Chart with Drill-Down

Create a hierarchy

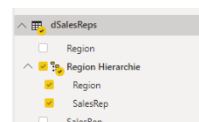
To support drill-down, create a hierarchy from Region and SalesRep. In the Data pane, drag SalesRep onto Region to create the hierarchy.

Note: If the months (text) are sorted incorrectly, sort the month indirectly by month number.

Drill-down

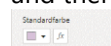
Create a column chart, and then add the hierarchy to the x-axis field well. To view detailed data, turn on drill-down mode and select a column.

The double down arrow displays all data at the next level.

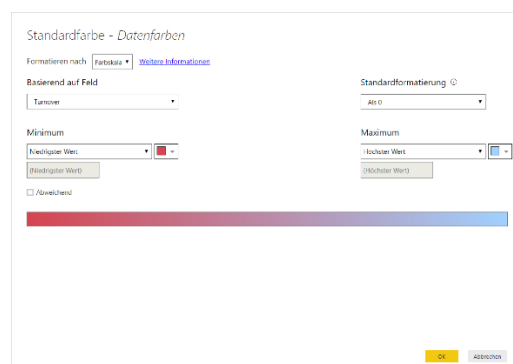


Conditional formatting

Open Format visual > Columns > Colors, and then select the fx button.



- Based on field: Turnover
- Select colors for minimum/maximum values



Note: Conditional formatting can also be applied to tables in the same way.

3.8.3 Line Chart

Insert a line chart using the Line Chart visual. Display the fiscal quarters on the x-axis (field: FiscalPeriod) and place AverageDailyGrossProfit on the y-axis.

Adjust x-axis sorting

In the chart's three-dot menu, first select the sorting field and sort order:

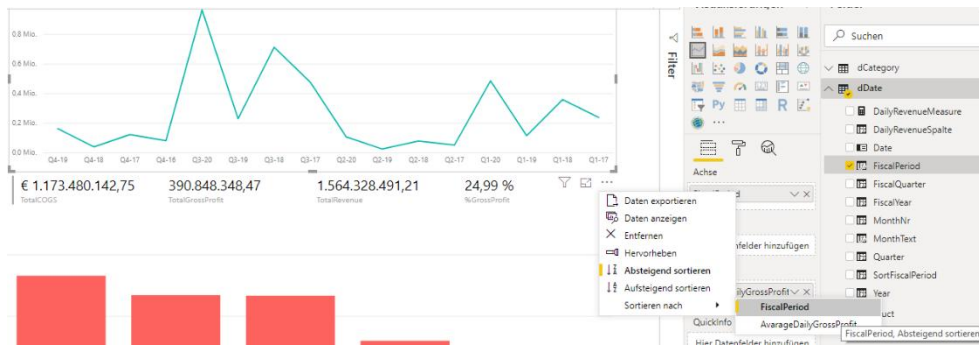


Figure 66

Sorting issue:

Because the current order is alphabetical, use the helper sort column created in Chapter 4.3.1.2. In Data view, select the FiscalPeriod column, and then select *Column tools* > *Sort by column* > *SortFiscalPeriod*.

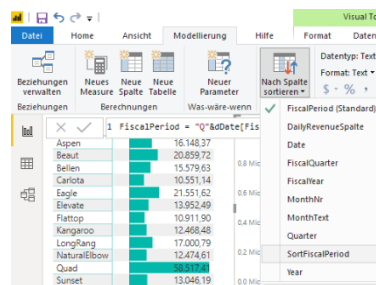


Figure 67

3.8.4 Card and Multi-Row Card

This element displays measures: Card displays one value, while Multi-row card displays multiple values:



Figure 68

TIP: To insert a text box, select *Insert* > *Text box*.

3.8.5 Slicer

You can add several slicers from the visual selection to make filtering data easier. Insert a slicer and then select the required field. Repeat this process for all additional slicers.

3.8.5.1 Bookmarks

The current filter settings can be saved as a bookmark.

To do so, select *View* > *Bookmarks* to open the Bookmarks pane. Apply the required filters, and then select *Add* to create a bookmark.

3.8.6 Import New Visuals

To import a custom visual such as Chiclet Slicer, select the ellipsis in the Visualizations pane, and then choose Get more visuals.



Figure 69

Sign in and select, for example, Chiclet Slicer. After installation, the new element is available in the current file.

Tip: If you need the visual permanently, start with a blank file, import the visual or visuals, and save the file as a template (*.pbit). The next time you use Power BI, start it by double-clicking the template.

3.9 KPI and Conditional Formatting

In a table or matrix visual, open the drop-down menu for the required field in the Values well, and then select *Conditional formatting* > *Icons*.

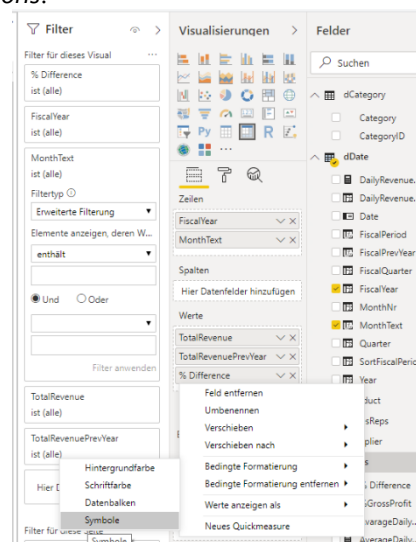


Figure 70

In the following dialog box, select the required thresholds and icons:

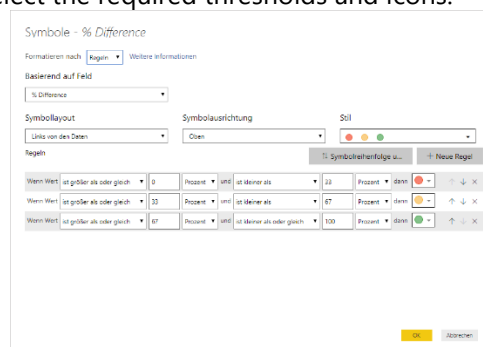


Figure 71

Finished report:

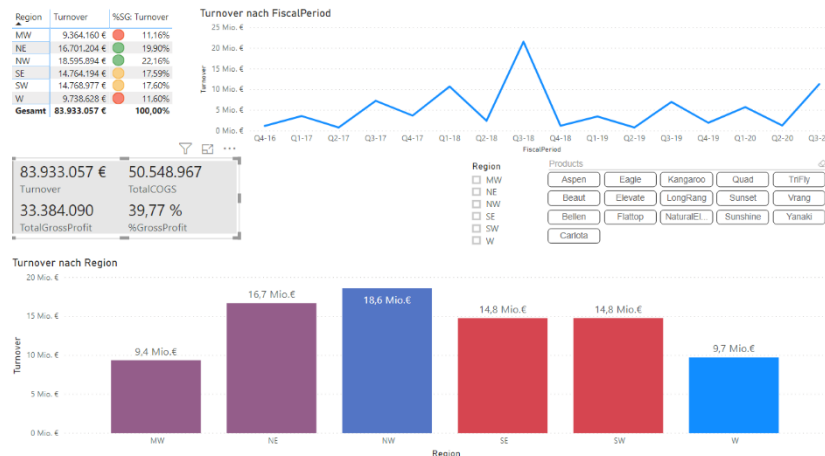


Figure 72

3.10 Publish Reports

First, create a free Microsoft account if necessary. When you start Power BI, sign in with your account. You can then use *Home/Publish to upload your project to powerbi.com*. Then sign in to powerbi.com and view your project online. Additional editing is also possible within the project, allowing each user to create individual visualizations based on the prepared data.

3.11 Compare Prior-Year Periods

In the master date table, the prior-year date can be calculated as follows:
`LastYear = SAMEPERIODLASTYEAR(MasterDate[Date])`

3.11.1 Calculate

The `CALCULATE()` function evaluates an expression in a specific filter context:
`=CALCULATE(Expression;Filter1;Filter2;...)`

Example 1

`UnitsBellen = CALCULATE([TotalUnits],FILTER(dProduct,dProduct[Products]="Bellen"))`

Example 2

```
TurnoverLastYear =
    CALCULATE(
        SUMX(
            fTrans;fTrans[UnitsSold]*(1-fTrans[Discount])*RELATED(dProduct[RetailPrice]));
        SAMEPERIODLASTYEAR(MasterDate[Date])
    )
```

Here, `SUMX()` defines the calculation expression
`SAMEPERIODLASTYEAR()` defines the filter context
 Next, calculate the difference between revenue and prior-year revenue:
`PrevYearDiff = [Turnover]-[TurnoverLastYear]`

Then analyze it by month in a PivotTable (matrix):

MonthText	2017	2018	2019	2020	Gesamt
Jan	223.821,94 €	617.297,73 €	-562.348,03 €	150.624,61 €	429.396,25 €
Feb	246.311,96 €	519.030,74 €	-530.534,32 €	159.647,44 €	394.455,81 €
Mrz	706.272,38 €	1.366.203,08 €	-1.373.762,81 €	409.700,59 €	1.108.413,24 €
Apr	1.430.940,39 €	2.914.612,06 €	-2.965.246,26 €	956.720,20 €	2.337.026,39 €
Mai	1.472.126,39 €	2.954.848,12 €	-3.008.895,76 €	928.804,72 €	2.346.883,46 €
Jun	658.469,74 €	1.297.647,15 €	-1.303.931,49 €	406.317,34 €	1.058.502,74 €
Jul	273.994,82 €	549.277,48 €	-541.254,95 €	151.656,95 €	433.674,30 €
Aug	260.511,42 €	549.633,29 €	-556.211,40 €	174.160,95 €	428.094,26 €
Sep	262.105,51 €	532.861,20 €	-534.585,14 €	156.586,86 €	416.968,44 €
Okt	1.587.221,84 €	3.147.761,88 €	-3.224.407,00 €	1.017.182,25 €	2.527.758,96 €
Nov	2.883.828,11 €	5.735.012,08 €	-5.806.140,84 €	1.800.883,14 €	4.613.582,49 €
Dez	2.793.214,15 €	5.439.678,88 €	-5.561.232,63 €	1.491.007,40 €	4.162.667,80 €
Gesamt	12.798.818,66 €	25.623.863,68 €	-25.968.550,64 €	7.803.292,44 €	20.257.424,15 €

Figure 73

To enable drill-down in the chart, create a hierarchy from Year and Month:

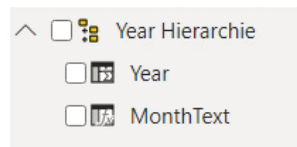


Figure 74

Then create a column chart:

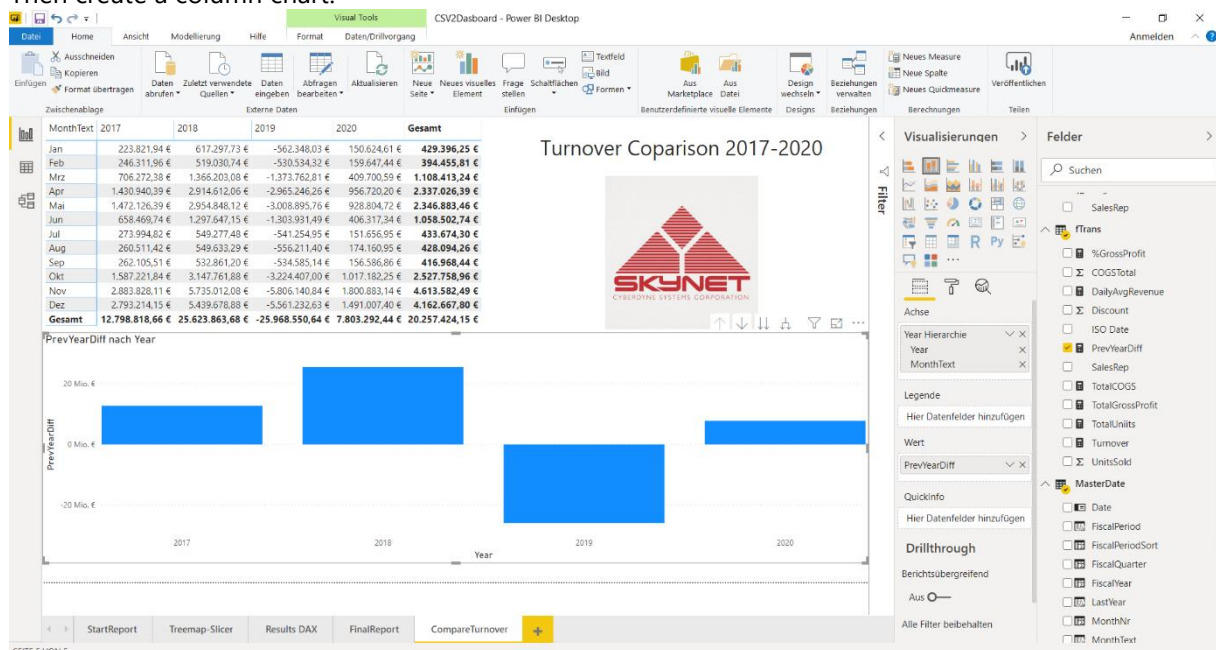


Figure 75

In the finished report, add labels by selecting *Insert > Text box* and add graphics by selecting *Insert > Image*.

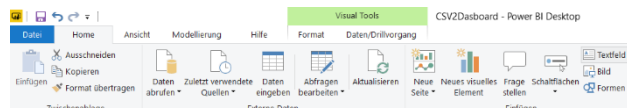


Figure 76

4 Power BI Desktop Advanced

4.1 Filter Concepts and Filter Propagation

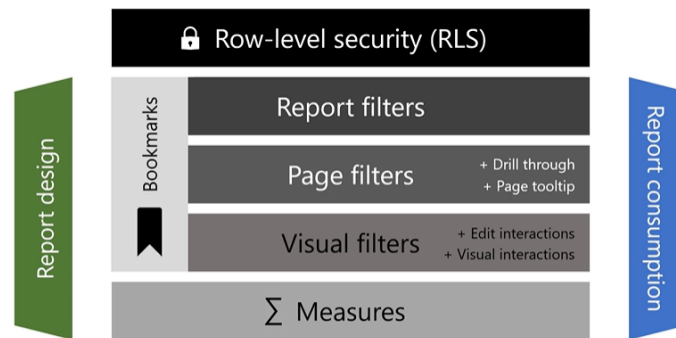


Figure 24

Details: [Introduction to designing reports for filtering - Training | Microsoft Learn](#)

TIP: Disable the Filters pane by selecting the eye icon:

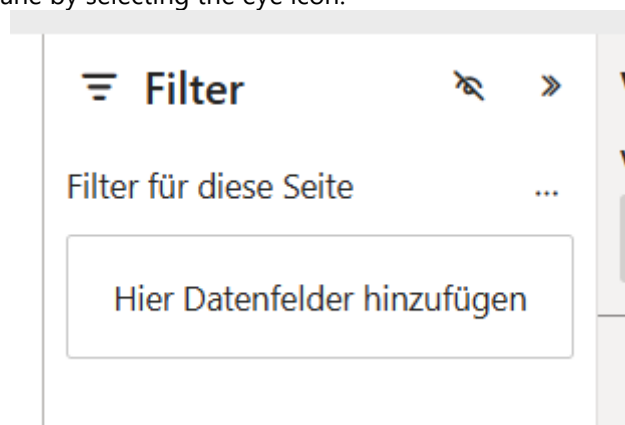


Figure 25

In addition, the visual header icons...

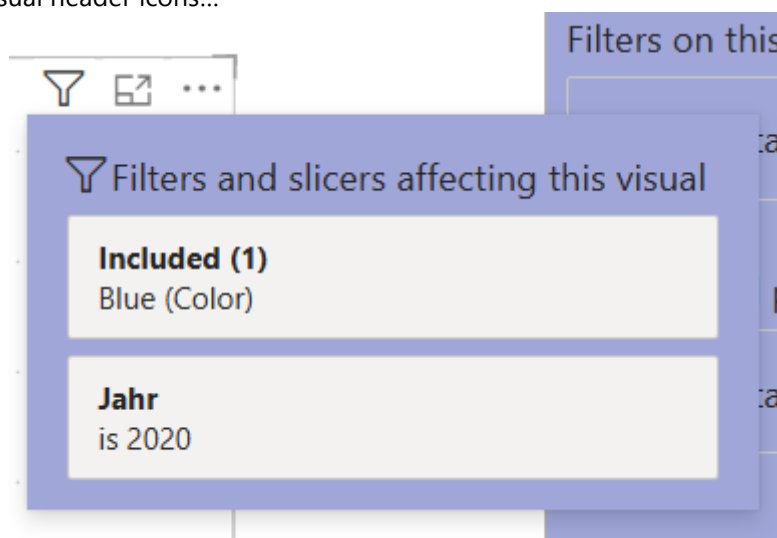


Figure 26

...can be disabled individually in the visual settings or globally in the report options.

4.2 DAX

4.2.1 Summarize Discount Percentage Bands

Folder: MultiCSVtoDashboard

Create a data table manually:

```
Table = DATATABLE("number"; INTEGER;  
    {  
        {1};{2};{3};{4};{5};{6} etc etc  
    }  
)
```

A better approach is to create the data table dynamically using GENERATESERIES():

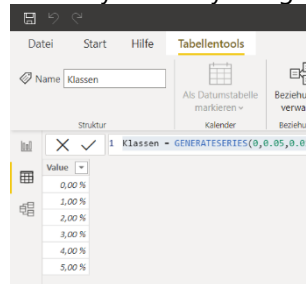
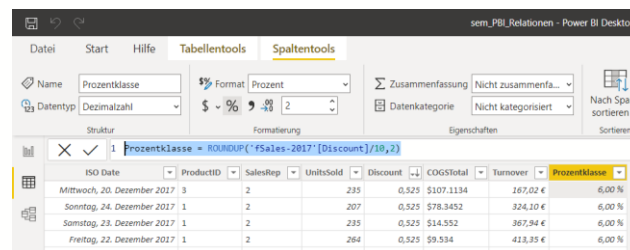


Figure 27

Calculate the percentage class in fSales:

Prozentklasse = `ROUNDUP('fSales-2017'[Discount]/10,2)`



ISO Date	ProductID	SalesRep	UnitsSold	Discount	COGSTotal	Turnover	Prozentklasse
Mittwoch, 20. Dezember 2017	3	2	235	0,525	\$107.1134	167,02 €	6,00 %
Sonntag, 24. Dezember 2017	1	2	207	0,525	\$78.3452	324,10 €	6,00 %
Sonntag, 23. Dezember 2017	1	2	235	0,525	\$14.552	367,94 €	6,00 %
Freitag, 22. Dezember 2017	1	2	264	0,525	\$9.534	413,35 €	6,00 %

Figure 28

Then link fSales/Prozentklassen to the data table referenced above.

4.2.1.1 Display Blank Values as Zero in Visuals

First, enable the display of blank cells:

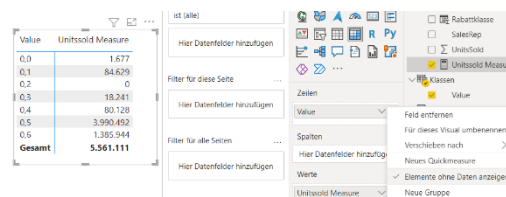


Figure 29

To display zeros as well, create a new measure and add zero at the end:

Units Total = `SUM('fSales-2017'[UnitsSold])+0`

4.2.2 Calculate

Folder: Story/2_FiscalPeriods

CALCULATE changes a visual's filter context and applies its own filter for a calculation, for example:

- Bikes = `CALCULATE([Turnover],dProduct[Category]="Bikes")`
- Mountain Bikes = `CALCULATE([Turnover],dProduct[Subcategory]="Mountain Bikes")`
- Turnover 2020 = `CALCULATE([Turnover],YEAR(fSales[OrderDateKey])=2020)`

4.3 Inactive Relationships (USERELATIONSHIP)

Folder: Orders and Payments

Turnover is to be analyzed by payment date:

Jahr	TurnoverByOrder	TurnoverByPayment
2022	77.437,00	72.335,93
2023	102.623,40	110.235,53
2024	4.048.280,61	2.047.124,05
Gesamt	4.228.341,01	2.229.695,51

Figure 30

To do this, create an additional relationship from the MasterDate table to the payment date:

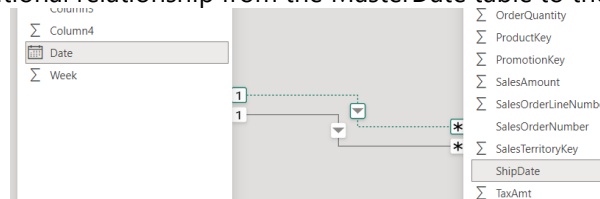


Figure 31

USERELATIONSHIP() can now be used as a filter criterion for CALCULATE() to return turnover by payment date:

TurnoverByPayment =

```
CALCULATE(
    SUMX(Sales, Sales[UnitPrice]*Sales[OrderQuantity]),
    USERELATIONSHIP('Calendar'[Date], Sales[ShipDate])
)
```

4.4 Filter Context

Folder: ContextTransition/FilterContext.pbix (DB: Product_Turnover_ContextTransition)

If a calculated column always shows the same result, this indicates a missing filter context. If the latest order date is to be determined in a calculated column in the Customer table, this does not work because the filter context is missing:

MaxDateWrong = MAXX(Sales, Sales[OrderDate])

To determine the latest order date for each customer, RELATEDTABLE() must therefore be used to establish the relationship between the customer and the sales table.

New column in the Customers table:

MaxDateOK = MAXX(RELATEDTABLE(Sales), Sales[OrderDate])

Determine the customer's number of sales:

SalesCount = COUNTROWS(RELATEDTABLE(Sales))

Recommended video on this topic: [\(77\) Understanding Row Context in DAX using Power BI - YouTube](#)

4.4.1 Context Transition

Folder: Folder: ContextTransition/FilterContext.pbix

Use CALCULATE in a column to establish context—for example, to identify which data belongs to each customer:

MaxDateCALCULATE = CALCULATE(MAXX(Sales,Sales[OrderDate]))

For related tables, a visual automatically establishes filter context. Select the aggregation type (Min, Max, and so on) in the field settings:

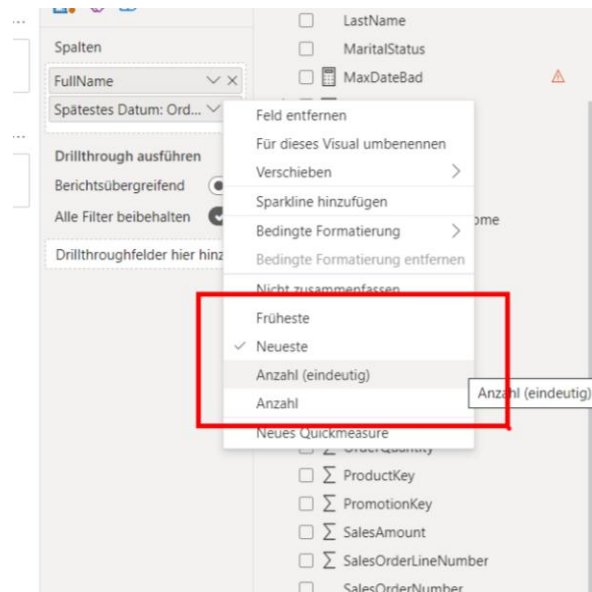


Figure 32

If a total is created as an additional column in the detail table, the overall turnover is calculated and repeated, for example:

Umsatzspalte Ohne Kontext (falsch) =

SUMX(fTransactions,fTransactions[Units]*RELATED(dProducts[Price]))

A measure does not require CALCULATE() because every measure is implicitly wrapped in CALCULATE(). [MaxDateMEASURE = MAXX(Sales,Sales[OrderDate])]

HöchstesDatum mit Kontext (richtig)=

[MaxDateMEASURE]

4.5 Parameter

4.5.1 Dynamic Dimensions (Field Parameters)

Switchable dimensions can be used in visuals, for example to switch a chart axis between month, product, or region. This can be done easily using field parameters:

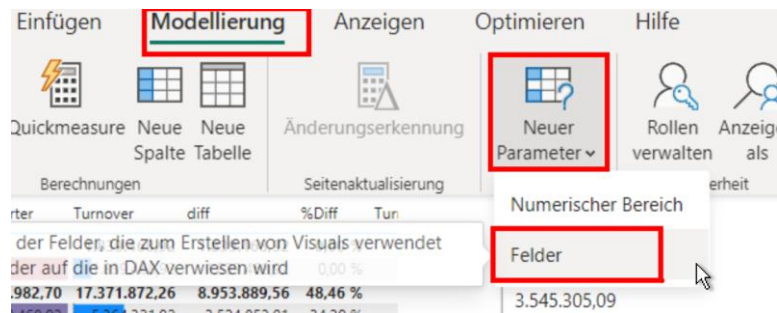


Figure 33

4.5.2 Numeric Parameters (Numeric Range)

Enables users to select a value or factor in a visual. This value can then be used for any calculations in measures.

Note: The `SELECTEDVALUE()` function returns the currently selected value.

4.6 Count/Calculate Orphaned Records

Calculate values for orphaned records in a detail table:

Measure that determines the number of orphaned records in the sales list:

```
Count Rows Non Exist =
//Counts nonexistent (orphaned) items/data rows in fSales
CALCULATE (
    COUNTROWS(fSales),
    ISBLANK(dProduct[ProductID])
)
```

Measure that determines the total units for orphaned records in the sales list:

```
Summe Units Sold Non Exist =
//Sums UnitsSold for nonexistent (orphaned) items in fSales
CALCULATE (
    SUMX(fSales,fSales[UnitsSold]),
    ISBLANK(dProduct[ProductID])
)
```

4.7 AI Analyses

4.7.1 Changes in the Chart

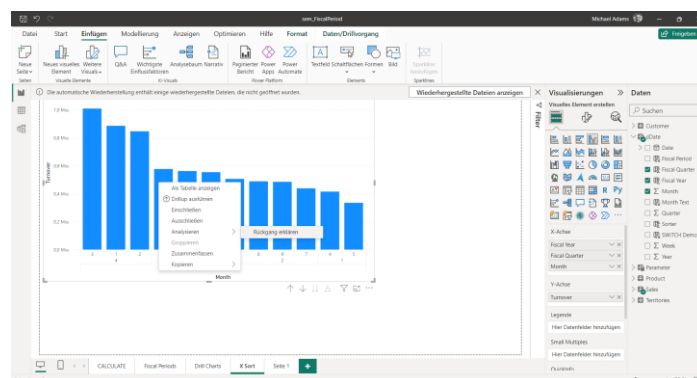


Figure 34

4.8 Advanced DAX

4.8.1 Previous Periods

4.8.1.1 Compare Fiscal Periods

Folder: Story/2_Fiscal Periods

This requires a MasterDate table, which you can create quickly using DAX:

```
dDate = ADDCOLUMNS(CALENDAR("2022-1-1", "2022-6-30"), "Year", YEAR([Date]), "Quarter", QUARTER([Date]), "Month", MONTH([Date]))
```

Or dynamically using the MIN-MAX date:

```
dDate = ADDCOLUMNS(
    CALENDAR(MIN(fSales[OrderDateKey]), MAX(fSales[OrderDateKey])),
    "Year", YEAR([Date]), "Quarter", QUARTER([Date]), "Month", MONTH([Date])
)
```

Additional columns can be added manually:

The screenshot shows the Power BI Desktop interface with the 'Spaltentools' (Column Tools) ribbon active. The formula bar contains the DAX formula for 'dDate'. Below the formula bar, a table is displayed with columns for Date, Year, Quarter, Month, Fiscal Quarter, Fiscal Year, Fiscal Period, Sorter, Last Year, Fiscal Month, Fiscal Date, and Fiscal Month Text. The table contains data for the first six months of 2018.

Date	Year	Quarter	Month	Fiscal Quarter	Fiscal Year	Fiscal Period	Sorter	Last Year	Fiscal Month	Fiscal Date	Fiscal Month Text
01.07.18	2018	3	7	2	2018	Q2-2018	20182	01.07.17	4	01.04.18	Apr
02.07.18	2018	3	7	2	2018	Q2-2018	20182	02.07.17	4	02.04.18	Apr
03.07.18	2018	3	7	2	2018	Q2-2018	20182	03.07.17	4	03.04.18	Apr
04.07.18	2018	3	7	2	2018	Q2-2018	20182	04.07.17	4	04.04.18	Apr
05.07.18	2018	3	7	2	2018	Q2-2018	20182	05.07.17	4	05.04.18	Apr
06.07.18	2018	3	7	2	2018	Q2-2018	20182	06.07.17	4	06.04.18	Apr

Figure 35

Calculate turnover for the previous period:

Turnover Last Period = `CALCULATE([Turnover], DATEADD(dDate[Date], -1, QUARTER))` + 0

Tun

Absolute change:

Turnover diff =

```
IF(
    ISBLANK([Turnover last year]),
    "N/A",
    [Turnover] - [Turnover last year]
)
```

Calculate percentage change:

Change % = Turnover % = `DIVIDE([Turnover] - [Turnover last year], [Turnover last year], "--")`

4.8.1.2 Calculate Time Intervals Relative to the Current Date Using DATESBETWEEN()

_Last 3 Months prev Year =

```
VAR LastMonth=
    CALCULATE(
        MONTH(MAX('Calendar'[Date])),
        ALL()
    )
VAR LastYear=
```

```

    CALCULATE(
        YEAR(MAX('Calendar'[Date])),
        ALL()
    )
VAR Spend= //This solution uses DATESBETWEEN
    CALCULATE(
        SUM(advertising[Spend]),
        KEEPFILTERS(
            DATESBETWEEN(
                'Calendar'[Date],
                DATE(LastYear-1,LastMonth-3,1), //LastMonth-3: Go back 3 months
                DATE(LastYear-1,LastMonth,1)
            )
        )
    )
RETURN Spend

```

4.8.1.3 Display Custom Time Intervals Using FILTER()

_Last X Months FILTER =

```

VAR LastMonth=
    CALCULATE(
        MONTH(MAX('Calendar'[Date])),
        ALL()
    )
VAR LastYear=
    CALCULATE(
        YEAR(MAX('Calendar'[Date])),
        ALL()
    )
VAR Spend= //This solution uses FILTER()
    CALCULATE(
        SUM(advertising[Spend]),
        KEEPFILTERS( //Retains the visual's filter and adds the following filter
            FILTER(
                'Calendar', //LastMonth-3 goes back 3 months
                'Calendar'[Date]>=DATE(LastYear,LastMonth-3,1) &&
'Calendar'[Date]<=DATE(LastYear,LastMonth,1) ||
                'Calendar'[Date]>=DATE(LastYear-1,LastMonth-3,1) &&
'Calendar'[Date]<=DATE(LastYear-1,LastMonth,1)
            )
        )
    )
RETURN Spend

```

Diff: Current-Last

%Diff: Current/Last-1

4.8.2 YTD (Year to Date)

Detailed example in DAX_Functions_YTD.pbix

4.8.2.1 YTD Jan-Jul

Determine spending from Jan–Jul, with July used here

Spend YTD Jan-Jul =

```
TOTALYTD(SUM(Spend[mGrossMediaSpend]),dDate[Date],MONTH(dDate[Date])<=7)
```

4.8.2.2 YTD Through the Latest (Current) Month

Spend latest Month =

```
VAR _LastMonth = CALCULATE (
    MONTH ( MAX ( dDate[Date] ) ),
    ALL ( dDate ) //Here, ALL() completely ignores the filter context
)
VAR LM = CALCULATE (
    SUM ( Spend[mGrossMediaSpend] ),
    KEEPFILTERS ( MONTH ( dDate[Date] ) <= _LastMonth ) //The filter context is retained and
                                                    //an additional filter is added
)
RETURN LM
```

4.8.3 IF()

Folder: Dax_Functions/Dax Functions

Note: All row calculations run in row context, meaning that a calculation is performed in every individual row of the data table.

Calculate age:

```
Age = DATEDIFF(Customer[BirthDate],TODAY(),YEAR)
```

Calculate age class:

Age Class =

```
//Determines four age groups
if(Customer[Age]>50, "50+",
if(Customer[Age]>40, "40-50",
if(Customer[Age]>30, "30-40",
"u30"))))
```

4.8.4 Switch

Calculate the number of sales:

```
Nr of Sales = calculate(COUNTROWS(Sales))+0
```

Classify regions by turnover:

Volume =

```
SWITCH(TRUE(),
    Territories[Nr of Sales]>=10000,"HIGH",
    Territories[Nr of Sales]>=5000,"MED",
    Territories[Nr of Sales]>=1,"LOW",
    "NA"
)
```

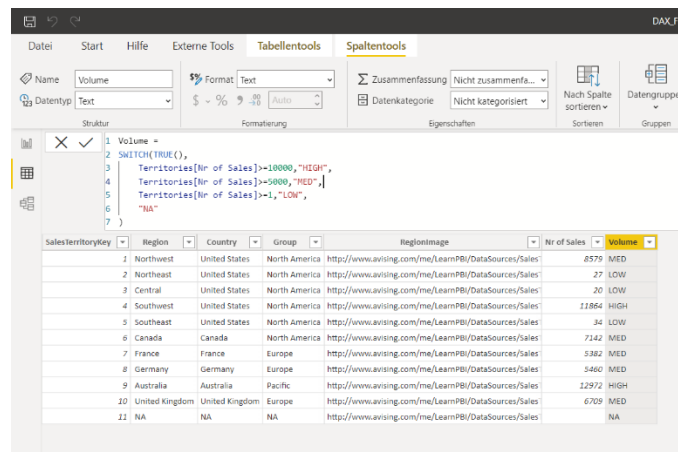


Figure 36

4.8.5 Group Measures in a Folder

Folder: DAX_Funtions/DAX_Functions_Measures.pbix

Tip: Organize columns and measures into folders in the data model. Select the required objects, and then group them under a folder name:

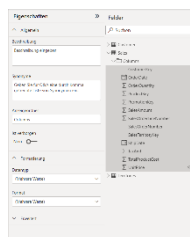


Figure 37

4.8.6 Calculate Totals for Percentage Shares

File: DAX_Functions_Measures.pbix/Calculate If (Database: AdventureWorks_Database.xlsx by AviS)

To calculate percentage shares, totals must be determined for each row. CALCULATE() and ALL() can first be used to override the visual's filter context:

Total Sales (All Countries) =

//Total for all regions excluding blank records

```

if(
    ISBLANK([Turnover]),
    BLANK(),
    CALCULATE(
        [Turnover],
        all(Territories[Country]) //REMOVEFILTERS() can be used instead of ALL()
    )
)

```

Country	Turnover	Total Sales (All Countries)	Country %
Australia	€ 9.051.766	29.307.836,53	30,89 %
Canada	€ 1.966.991	29.307.836,53	6,71 %
France	€ 2.640.526	29.307.836,53	9,01 %
Germany	€ 2.890.708	29.307.836,53	9,86 %
United Kingdom	€ 3.387.491	29.307.836,53	11,56 %
United States	€ 9.370.355	29.307.836,53	31,97 %
Gesamt	€ 29.307.837	29.307.836,53	100,00 %

Figure 38

Country %: Turnover can then be divided by Total Sales.

4.8.7 Variables in DAX

Use variables to improve readability and performance.

General example:

Revenue YoY % =

```
VAR RevenuePriorYear =
    CALCULATE(
        [Revenue],
        SAMEPERIODLASTYEAR('Date'[Date])
    )
RETURN
    DIVIDE(
        [Revenue] - RevenuePriorYear,
        RevenuePriorYear
    )
```

Example of a percentage comparison:

Total Sales (All Countries) =

```
//Total for all regions excluding blank records
var SalesAllCountries=
if(
    ISBLANK([Turnover]),
    BLANK(),
    CALCULATE(
        [Turnover],
        all(Territories[Country]) //REMOVEFILTERS() can be used instead of ALL()
    )
)
var PercentCountries=
    [Turnover]/SalesAllCountries
return PercentCountries
```

Note: Why not use Show values as... %?

When a slicer is used, percentage shares refer to the items selected in the slicer. However, a measure can be permanently fixed to the overall total (all countries):

Country	Turnover	Total Sales (All Countries)	Country %	%SG: Turnover
Australia	€ 9.051.766	29.307.836,53	30,89 %	82,15%
Canada	€ 1.966.991	29.307.836,53	6,71 %	17,85%
Gesamt	€ 11.018.757	29.307.836,53	37,60 %	100,00%

Figure 39

4.8.8 OR/AND/IN

Use OR in CALCULATE to combine France and Germany and compare them with another country, for example:

France and Germany =

```
CALCULATE(
    [Turnover],
    Territories[Country]="germany" || Territories[Country]="France"
)
```

Alternatively:

France and Germany =

```
CALCULATE(
    [Turnover],
    or(Territories[Country]="germany", Territories[Country]="France")
)
```

Or using IN:

France and Germany =

```
CALCULATE(
    [Turnover],
    Territories[Country] IN {"germany", "France"}
)
```

4.8.9 Combine AND/IN with FILTER()

Large Sales =

```
FILTER(
    Sales,
    Sales[SalesAmount]>2000 && RELATED('Calendar'[WeekdayNum]) IN {1,2,3,4,5}
)
```

4.8.10 Time-Based Analysis

4.8.10.1 Total turnover with discount in Q1

Turnover Q1 Rabatt mit USt =

VAR Turnover =

```
SUMX(
    fSales,
    IF(
        MONTH(fSales[ISO Date])>=1 && MONTH(fSales[ISO Date])<=3,
        [Turnover] * .7,
    )
)
```

```

    )
    )
VAR Steuer = 0.19
VAR Result = Turnover*Steuer
RETURN Result

```

4.8.10.2 Running Total

File: DAX_Functions_YTD.pbix

TOTALYTD function: YearToDay = `TOTALYTD([Turnover], 'Calendar'[Date])`

TOTALYTD creates a running total by summing through the end of the respective period (month, quarter, etc.).

Important: A correct, related date table must exist.

Note: After drilling down in the table, months may need to be sorted indirectly using a helper column.

The individual years can be compared in the chart for the Jan–Jul period. (The Year field must be placed in the Legend bucket.)

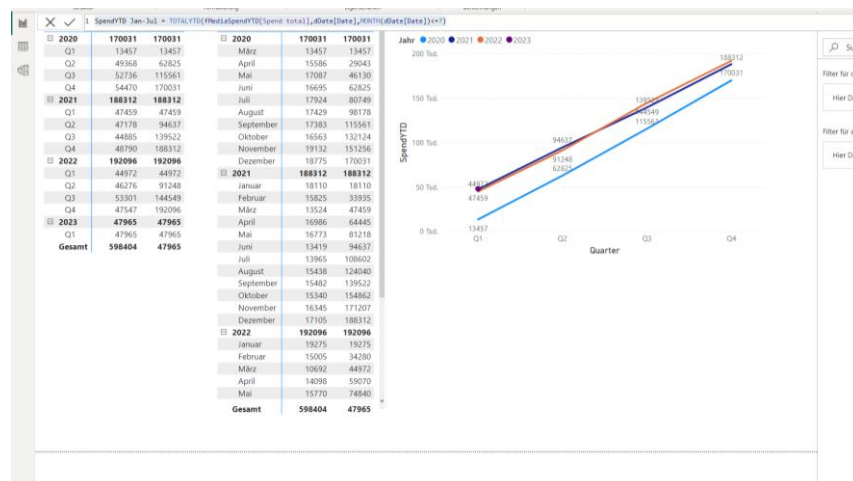


Figure 40

Determine spending from Jan–Jul:

`SpendYTD Jan-Jul = TOTALYTD(fMediaSpendYTD[Spend total], dDate[Date], MONTH(dDate[Date]) <= 7)`

Additional topics: Spend YTD Jan–Jul vs last year, % diff vs last year in the DAX_Functions_YTD sample file

4.8.10.3 Running total for the fiscal year

If the (fiscal) year-end is defined in TOTALYTD, the running total restarts on that date:

`Fiscal YTD = TOTALYTD([Turnover], 'Calendar'[Date], "30.6")`

Year	Turnover	YearToDay	Fiscal YTD
2014	7.072.084,23	7.072.084,23	3.805.710,58
Jan	473.388,16	473.388,16	473.388,16
Feb	506.191,69	979.579,85	979.579,85
Mar	473.943,03	1.453.522,88	1.453.522,88
Apr	513.329,47	1.966.852,36	1.966.852,36
May	543.993,40	2.510.845,76	2.510.845,76
Jun	755.527,89	3.266.373,65	3.266.373,65
Jul	596.746,56	3.863.120,21	596.746,56
Aug	350.816,89	4.213.936,90	1.147.583,25
Sep	644.135,20	5.058.072,10	1.791.698,45
Oct	663.692,29	5.721.764,39	2.455.390,74
Nov	673.556,20	6.395.320,59	3.128.946,93
Dec	676.763,65	7.072.084,23	3.805.710,58
2015	5.762.134,24	5.762.134,24	3.037.501,33
Jan	500.365,15	500.365,15	4.306.075,73
Feb	546.091,47	1.046.366,62	4.852.077,20
Mar	350.466,99	1.396.833,60	5.202.544,18
Apr	415.390,23	1.812.223,83	5.617.934,41
May	335.095,09	2.147.318,92	5.953.029,50
Jun	577.313,99	2.724.632,91	6.530.343,50
Jul	438.865,17	3.163.498,08	438.865,17
Aug	489.090,33	3.652.588,41	927.955,50
Sep	485.574,79	4.138.163,20	1.413.530,29
Oct	506.399,26	4.644.562,46	1.919.929,55
Nov	562.772,56	5.207.335,02	2.482.702,11
Dec	554.799,22	5.762.134,24	3.037.501,33
2016	16.473.618,05	16.473.618,05	9.720.059,11
Jan	886.668,84	886.668,84	3.924.170,17
Feb	847.413,51	1.734.082,35	4.771.583,68
Mar	1.010.258,13	2.744.340,48	5.781.841,81
Apr	1.080.449,58	3.824.790,06	6.862.291,39
May	1.196.981,11	5.021.771,17	8.059.272,50
Jun	1.731.787,77	6.753.558,94	9.791.060,27
Jul	1.340.244,95	8.093.803,89	1.340.244,95
Aug	1.462.479,83	9.556.283,72	2.802.724,78
Sep	1.480.905,18	11.037.188,90	4.283.629,96
Oct	1.608.750,53	12.645.939,43	5.892.380,49
Nov	1.878.317,51	14.524.256,94	7.770.698,00
Dec	1.949.361,11	16.473.618,05	9.720.059,11
2017			
Jan			9.720.059,11
Feb			9.720.059,11
Mar			9.720.059,11
Apr			9.720.059,11
May			9.720.059,11
Jun			9.720.059,11
Gesamt	29.307.836,53		

Figure 41

Note: Extra rows can be filtered out through the TOTALYTD filter argument using the FILTER() function:

Fiscal YTD =

TOTALYTD([Turnover], 'Calendar'[Date], FILTER('Calendar', 'Calendar'[Year]<2017), "30.6")

2016	16.473.618,05	16.473.618,05	9.720.059,11
Jan	886.668,84	886.668,84	3.924.170,17
Feb	847.413,51	1.734.082,35	4.771.583,68
Mar	1.010.258,13	2.744.340,48	5.781.841,81
Apr	1.080.449,58	3.824.790,06	6.862.291,39
May	1.196.981,11	5.021.771,17	8.059.272,50
Jun	1.731.787,77	6.753.558,94	9.791.060,27
Jul	1.340.244,95	8.093.803,89	1.340.244,95
Aug	1.462.479,83	9.556.283,72	2.802.724,78
Sep	1.480.905,18	11.037.188,90	4.283.629,96
Oct	1.608.750,53	12.645.939,43	5.892.380,49
Nov	1.878.317,51	14.524.256,94	7.770.698,00
Dec	1.949.361,11	16.473.618,05	9.720.059,11
2017			
Jan			9.720.059,11
Feb			9.720.059,11
Mar			9.720.059,11
Apr			9.720.059,11
May			9.720.059,11
Jun			9.720.059,11
Gesamt	29.307.836,53		

Figure 42

Determine turnover for weekends only:

Turnover Weekend =

```
TOTALYTD(
    [Turnover],
    'Calendar'[Date],
    FILTER(
        'Calendar',
        WEEKDAY('Calendar'[Date],2) IN {6,7}
    )
)
```

4.8.10.4 Balance at the End of a Period

LASTDATE() is suitable for determining the balance at the end of a period, such as month-end:

Last Balance Date =

```
CALCULATE(
    Sales[Sum Units Balance],
    LASTDATE('Calendar'[Date])
)
```

A problem can occur when no date exists at month-end—for example, when month-end falls on a weekend. This can be resolved using LASTNONBLANK():

Last Nonblank Balance =

```
CALCULATE(
    Sales[Sum Units Balance],
    LASTNONBLANK(
        'Calendar'[Date],
        Sales[Sum Units Balance])
)
```

This returns the last nonblank date at the end of a period, such as the end of a month.

Conversely, FIRSTNONBLANK() and FIRSTDATE() can also be used.

More information: Mitchel Pearson, Pragmatic Works: [\(26\) !\[\]\(b623c03fa487d1698dace6203c519726_img.jpg\) Beginning Power BI DAX Functions Tutorial \[Full Course\] - YouTube](#)

4.8.11 Calculate Capacities

Folder: Advanced/Auslastung Kapazitäten

Problem: Different calculations must be performed depending on the drill level in a chart.

ISINSCOPE() determines the hierarchy level to which a chart is currently drilled. Combined with SWITCH(), the corresponding action can then be derived in a measure:

Verfügbar = SWITCH(

```
    TRUE();
    ISINSCOPE(dDate[Month]);SUM(dRessourcen[Capacity Hours per Year])/12;
    ISINSCOPE(dDate[Quarter]);SUM(dRessourcen[Capacity Hours per Year])/4;
    ISINSCOPE(dDate[Year]);SUM(dRessourcen[Capacity Hours per Year]
```

))

Note: For ISINSCOPE() to work as in the example above, a MasterDate table must exist in the data model.

4.8.12 Row Context

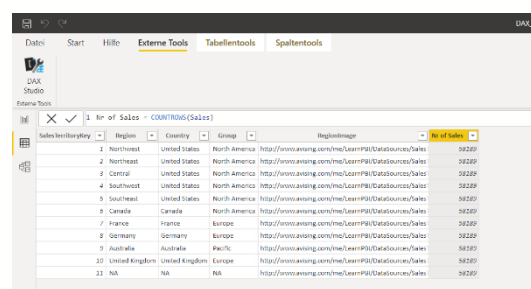
Folder: Dax_Functions/Dax Functions

This expression returns only the total number of rows in a related table because the filter context is missing:

Nr of Sales = COUNTROWS(Sales)

This also returns only the overall result or total:

Nr of Sales = SUMX(Sales,Sales[SalesAmount])



Sales hierarchy	Region	Country	Group	RegionImage	Nr of Sales
1 Northwest	United States	North America	http://www.avising.com/mc/learn/PB/DataSource/Sales	36289	
2 Northeast	United States	North America	http://www.avising.com/mc/learn/PB/DataSource/Sales	36289	
3 Central	United States	North America	http://www.avising.com/mc/learn/PB/DataSource/Sales	36289	
4 Southwest	United States	North America	http://www.avising.com/mc/learn/PB/DataSource/Sales	36289	
5 Southeast	United States	North America	http://www.avising.com/mc/learn/PB/DataSource/Sales	36289	
6 Canada	Canada	North America	http://www.avising.com/mc/learn/PB/DataSource/Sales	36289	
7 France	France	Europe	http://www.avising.com/mc/learn/PB/DataSource/Sales	36289	
8 Germany	Germany	Europe	http://www.avising.com/mc/learn/PB/DataSource/Sales	36289	
9 Australia	Australia	Pacific	http://www.avising.com/mc/learn/PB/DataSource/Sales	36289	
10 United Kingdom	United Kingdom	Europe	http://www.avising.com/mc/learn/PB/DataSource/Sales	36289	
11 N/A	N/A	N/A	http://www.avising.com/mc/learn/PB/DataSource/Sales	36289	

Figure 43

Solutions:

Nr of Sales = `COUNTROWS(RELATEDTABLE(Sales))`

Nr of Sales = `SUMx(RELATEDTABLE(Sales),Sales[SalesAmount])`

`RELATEDTABLE()` RETURNS ALL DETAIL-TABLE ROWS RELATED TO THE CORRESPONDING ROW IN THE MASTER TABLE.

or:

Nr of Sales = `calculate(COUNTROWS(Sales))`

Nr of Sales = `CALCULATE(SUMx(Sales,Sales[SalesAmount]))`

`CALCULATE()` RETRIEVES THE FILTER CONTEXT FROM THE MASTER TABLE.

4.8.13 KEEPFILTERS()

Use this function with `CALCULATE()`. Use `KEEPFILTERS()` when `CALCULATE` should add specific filters without completely overriding the existing filter context.

Example:

Spend latest Month =

```
VAR _LastMonth = CALCULATE (
    MONTH ( MAX ( dDate[Date] ) ),
    ALL ( dDate ) //Here, ALL() completely ignores the filter context
)
VAR LM = CALCULATE (
    SUM ( Spend[mGrossMediaSpend] ),
    KEEPFILTERS ( MONTH ( dDate[Date] ) <= _LastMonth ) //The filter context is retained and
                                                         //zusätzlicher Filter wird hinzugefügt
)
RETURN LM
```

4.8.14 DATATABLE()

Creates a table but does not allow scalar expressions (calculated values).

```
dUserRegions= DataTable("Name", STRING,
    "Region", STRING
    ,{
        {" User1", "East"},
        {" User2", "East"},
        {" User3", "West"},
        {" User4", "West"},
        {" User4", "East"}
    }
)
```

Alternative: DAX Table Constructor

```
Table_2 = { (1,"Mark", 112), (3,"John", 121), (3,"Chan", "Hello") }
```

You cannot specify column names here, but scalar values are supported.

4.8.15 Access Previous Rows in DAX (Beta)

First, an index column must be created:

IndexDAX = RANKX('fSales-2020','fSales-2020'[ID],,ASC)

Note: RANKX() assigns consecutive numbers only to unique values.

4.8.16 Lookupvalue

```
LOOKUPVALUE(  
    <result_columnName>,  
    <search_columnName>,  
    <search_value>  
    [, <search2_columnName>, <search2_value>]...  
    [, <alternateResult>]  
)
```

Example: Display turnover (total) for the year 2017 AND calendar week 2:

```
This Turnover = LOOKUPVALUE(  
    LastWeek[total],LastWeek[Week],2,LastWeek[Year],2017  
)
```

Year	Week	total	Index	Index.1	csw.total	Diff
2017	2	9564	0	1		9564
2017	3	25103	1	2	9564	15539
2017	4	26371	2	3	25103	1268

Figure 44

Alternatively:

```
This Turnover2 = SUMX(  
    FILTER(LastWeek, LastWeek[Week]=2 && LastWeek[Year]=2017),  
    LastWeek[total]  
)
```

4.8.17 VALUES()

Returns a column as a table and removes duplicates.

Example: SUMX(VALUES(Product[Color]),[Sales Amount])

(Calculates turnover by color)

4.8.18 SELECTEDVALUE()

If a column has been filtered down to a single value, the function returns that value. Otherwise, the alternate result is returned.

Example: Select Period = SELECTEDVALUE('Select Period'[Value],"Alternate result")

4.8.19 LOOKUPVALUE()

Returns the information associated with the search criterion, similar to XLOOKUP.

4.8.20 ALL() ALLSELECTED() ALLEXCEPT()

- ALL() returns all records and overrides all other filters.
- ALLSELECTED() returns all records selected in the slicer.
- ALLEXCEPT() returns all records except those excluded by the specified filter. In the example below, except Country; the visual's row context—which filters row by row by Country—is taken into account

TIP: ALLEXCEPT() can be used for row-context-based grouping, for example
 CALCULATE(Berechnung,ALLEXCEPT(Tabelle, Gruppenspalte))

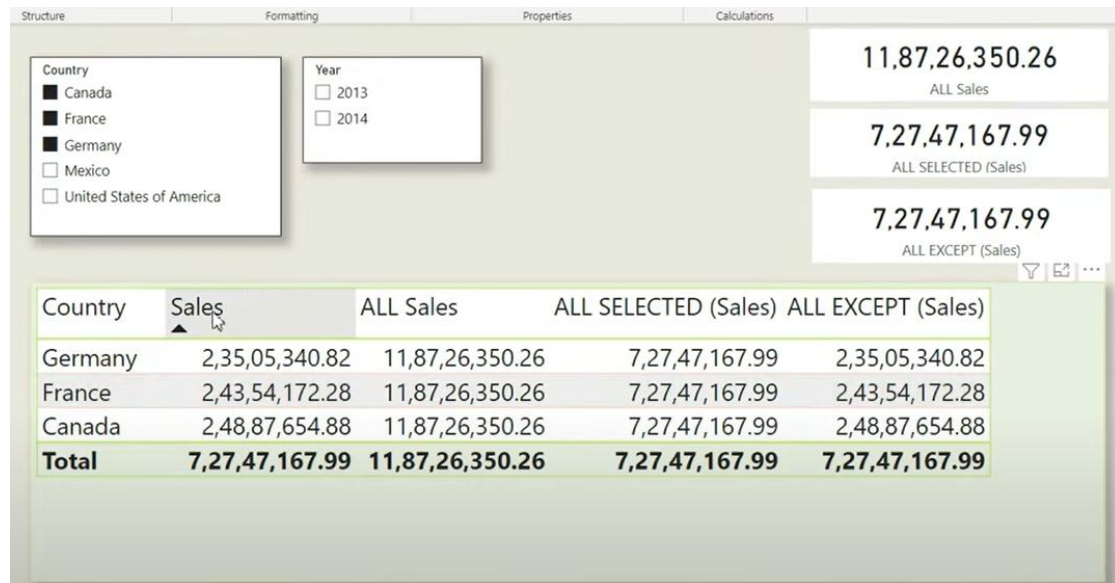


Figure 45

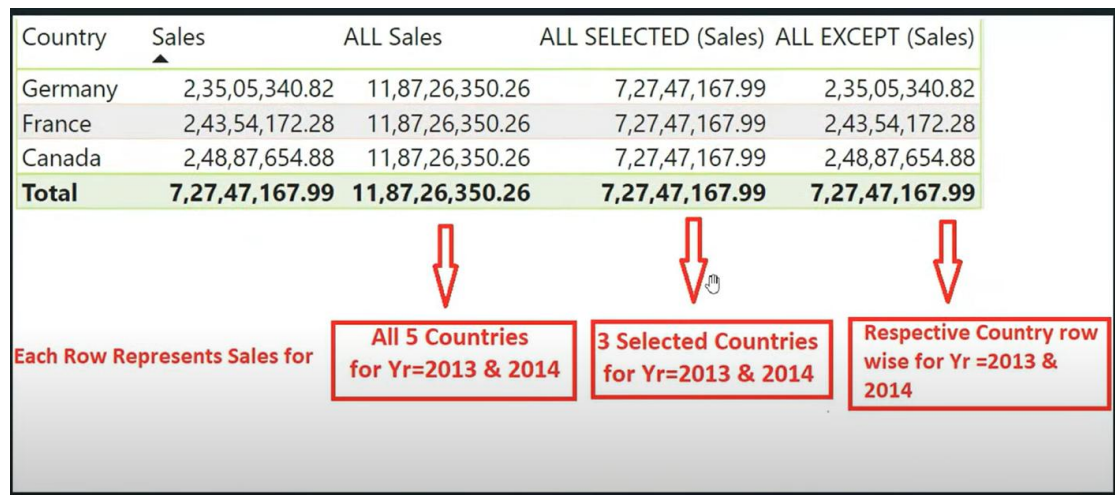


Figure 46

See also: [All Vs AllSelected Vs AllExcept | Power BI Dax Tutorial \(youtube.com\)](https://www.youtube.com/watch?v=...)

4.8.21 CONVERT()

Converts the data type of a value

MSRRegion =

IF(

```
CALCULATE(SUMX(Sales,Sales[Umsatz]),Sales[Region]="Nord")=BLANK(),
```

```
"-", //TEXT data type
```

```
CONVERT(CALCULATE(SUMX(Sales,Sales[Umsatz])),STRING))
```

In the ELSE branch of IF, the result of CALCULATE() must be converted to text to make the data type consistent across the entire column

4.8.22 SUMMARIZE()

The result is a table

Groups by columns and creates new calculated columns with aggregated results.

```
SUMMARIZE(
    fSales, //Group from this table
    fSales[SalesTerritoryKey],dProducts[Color], //Group these columns
    "Turnover",
    SUMX(fSales,fSales[OrderQuantity]*RELATED(dProducts[ListPrice])) //New column
    with aggregation
)
```

4.9 Visuals

4.9.1 Drillthrough

Story folder: Fiscal Periods/Sales Overview and Sales Details

Requires a dedicated detail page that can be drilled through to by right-clicking.

For example, if a specific year is selected on page one, the Year field must be placed in the Drillthrough bucket on the detail-page visual. The visual can then show the months of the year selected on page one.

Example:

- Main page shows countries
- Drillthrough page shows city details, such as turnover by city:

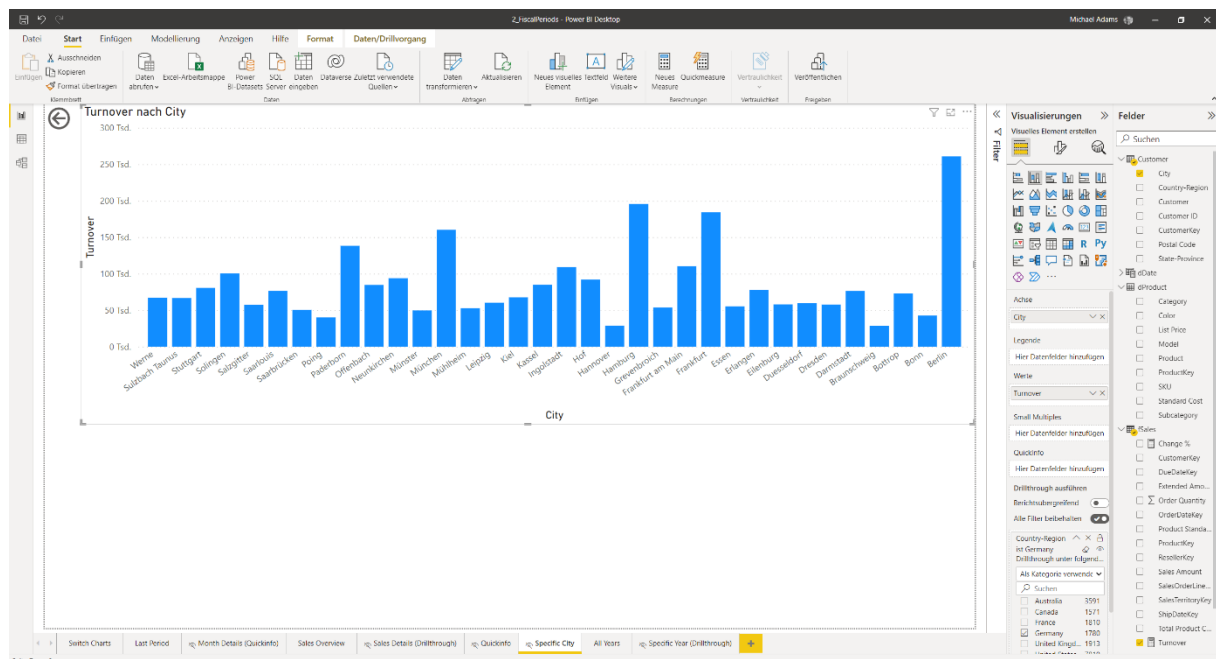


Figure 47

Tip: Drillthrough can also be presented using a button:

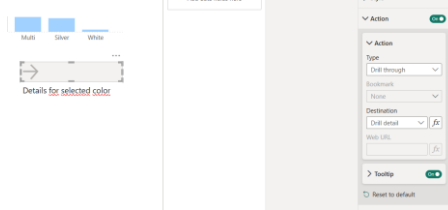


Figure 48

Notes

- Toggle visual: [Swap Power BI Visuals to add FLEXIBILITY in your reports - YouTube](#)

4.9.2 Bookmarks and Interactions

Story folder: 1_SwitchVisuals

Set up a bookmark:

- Enable the Bookmarks pane
- *Bookmark/Add*
- Data means that filter settings are also saved

Set up an image action

- Insert image
- Format pane: Action, Type: Bookmark

4.9.2.1 Switch between years and quarters in the visual

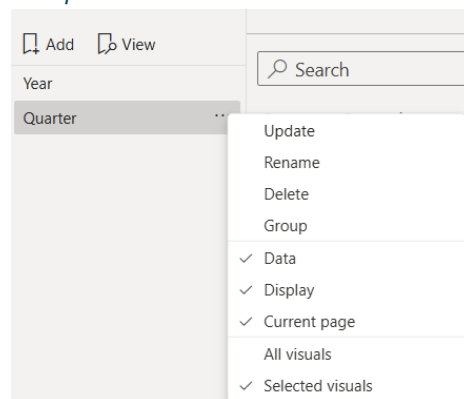


Figure 49

First drill to years or quarters, then select Selected visuals in the new bookmark

4.9.3 Tooltip

Folder: Advanced/2_FiscalPeriods.pbix Blatt Last Period und Sales Details

First, a tooltip page must be set up to display the desired details:

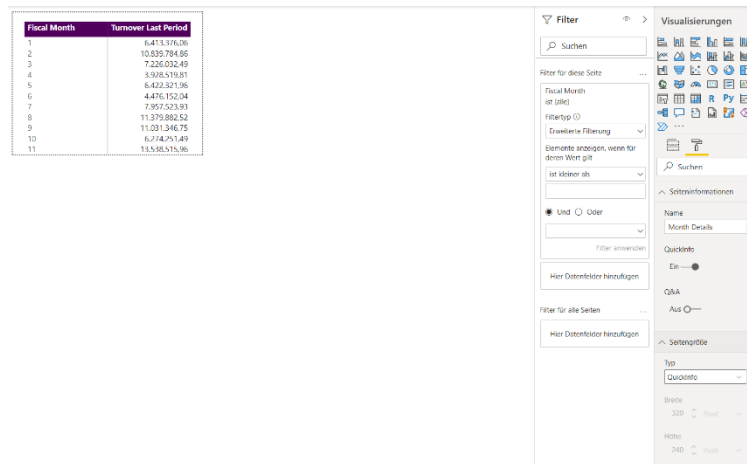


Figure 50

TIP: It is advisable to hide the tooltip page so that it is not displayed in the report.

Then enable the tooltip on the desired visual and select the detail page:

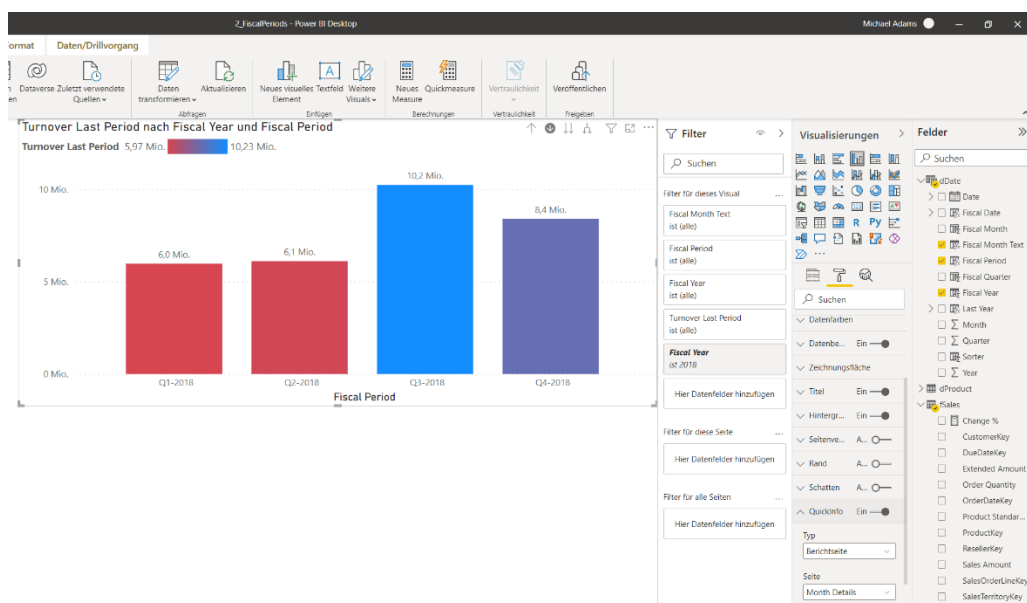


Figure 51

4.9.4 Small Multiples

Folder: Small_Multiples

First, import the data and create the data model.

Sales table: Quarters can be broken down by individual years

Alternatively, break down gender (M/F) by year.

4.9.5 Map Visuals

Folder: Advanced/2_FiscalPeriods.pbix, sheet Sales Overview

Note: If map visuals are not enabled, they must be enabled in the Power BI admin portal:

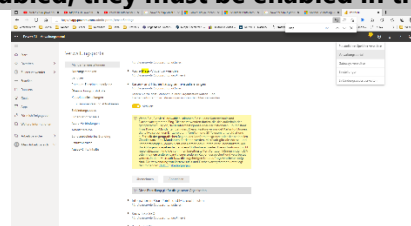


Figure 52

4.9.6 Map

Displays values as bubbles on the map.
Formatting: Bubbles/size and individual colors

4.9.7 Filled Map

Displays regions as filled areas. Use a color scale for the fill colors.

4.9.8 Sparkline

Folder: Advanced/2_FiscalPeriods.pbix/Sparklines

After total turnover has been determined in a table, for example, sparklines can be created in an additional column: Insert/Insert sparkline

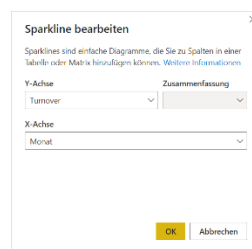


Figure 53

Here, for example, months and their turnover can be selected to visualize the turnover trend:

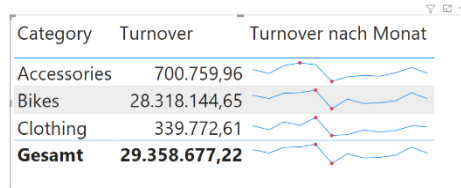
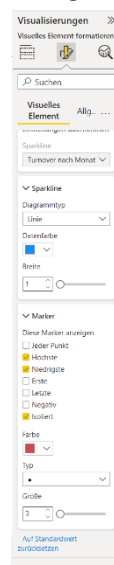


Figure 54

Formatting settings for sparklines:



Data settings for sparklines:

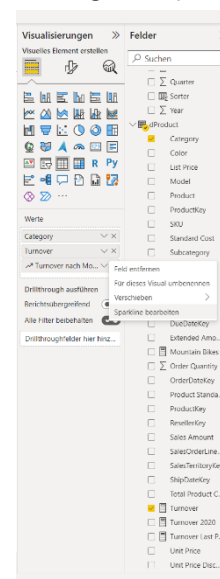


Figure 55

4.9.9 Ribbon Chart

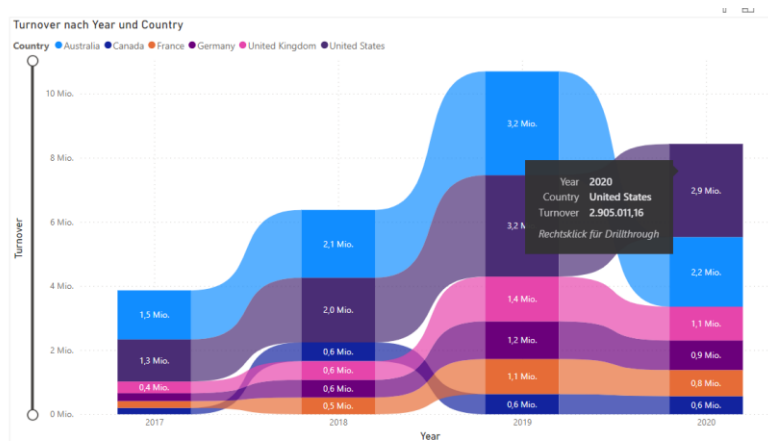


Figure 56

Recommended settings:

- Zoom slider
- Data labels
- If necessary, use a filter to reduce the chart to the key categories

4.9.10 Decomposition Tree

Enables detailed analysis of figures or data, for example by:

- Country
- Product color
- Gender

When the highest or lowest value is selected, the AI function automatically chooses the next categories:

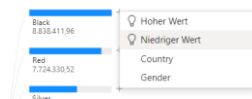


Figure 57

Alternatively, the category can be selected manually.

The following example uses the lowest-value category to identify which color is purchased least, by country and gender:

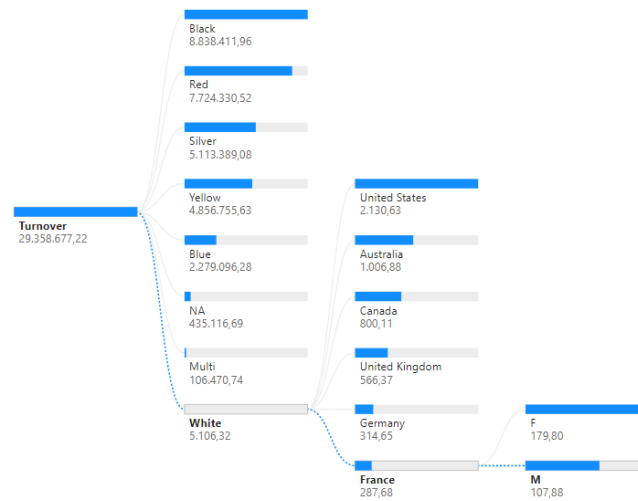


Figure 58

4.9.11 Scatter Chart

Enables analysis of multiple dimensions, for example

- Turnover, profit, units, region, and time

The timeline can even be animated using the Play button.

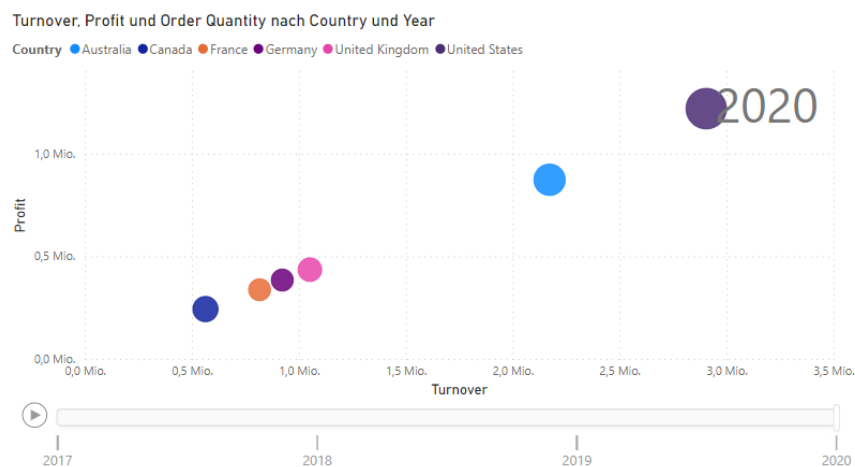


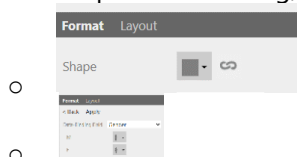
Figure 59

4.9.12 Add visuals (example: Infographic Designer)

Preparation: Filter to France and Color=White; use Gender and Turnover as the data

Compare order turnover by gender:

- Format/Shape: Data Binding, to assign shapes to the data points



- Multiple Units: ON
- Value Color: ON -> color setting
- Power BI formatting settings: Chart Type: Bar

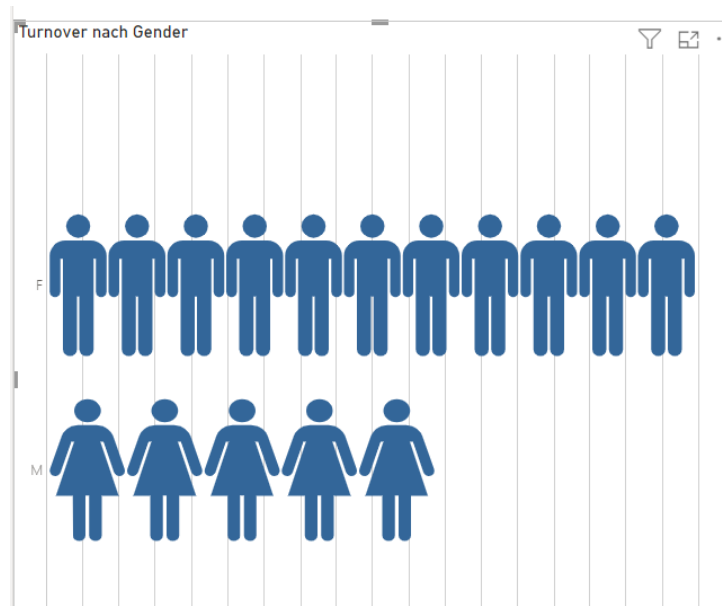


Figure 60

Tip: Further breakdown, for example by Legend or Row By Country

4.10 Power BI Service

4.10.1 Automatically refresh online data (datasets)

Prerequisite: Store the data in SharePoint.

- Sign in to the Power BI portal (for example, <https://app.powerbi.com>).
- Select the desired workspace
- For the desired dataset (orange icon), select the ellipsis (three-dot) menu and then Settings:

Name	Type	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Include in app
Bericht aus Sharepoint	Report	Seminare	10/5/21, 9:41:38 AM	—	—	—	Yes
Bericht aus Sharepoint	Dataset	Seminare	10/5/21, 9:41:38 AM	N/A	—	—	Yes
Sales_Units	Report	Seminare	10/2/21, 10:58:25 AM	—	—	—	Yes
Sales_Units	Dataset	Seminare	10/2/21, 10:58:25 AM	N/A	—	—	Yes
Sem_PBI_Einstieg	Report	Seminare	10/5/21, 8:32:33 AM	—	—	—	Yes
Sem_PBI_Einstieg	Dataset	Seminare	10/5/21, 8:32:33 AM	N/A	—	—	Yes

Figure 61

- Under Data Source Credentials, enter the SharePoint credentials:

Figure 62

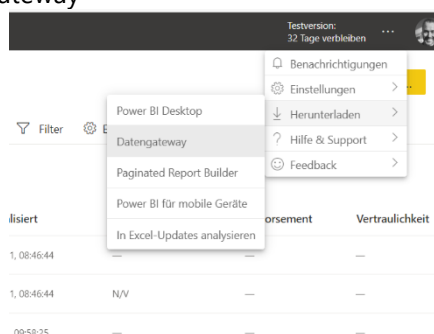
- Under Scheduled Refresh, enable automatic refresh:

Figure 63

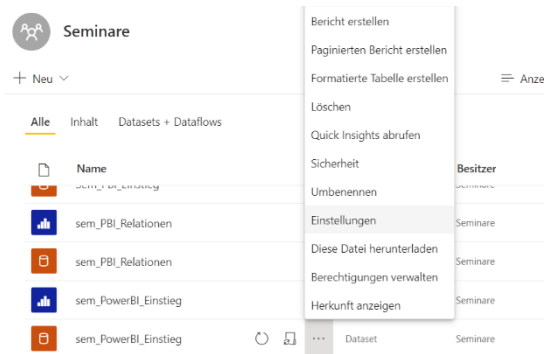
4.10.2 Set Up the Power BI Data Gateway

Used to refresh data stored on-premises.

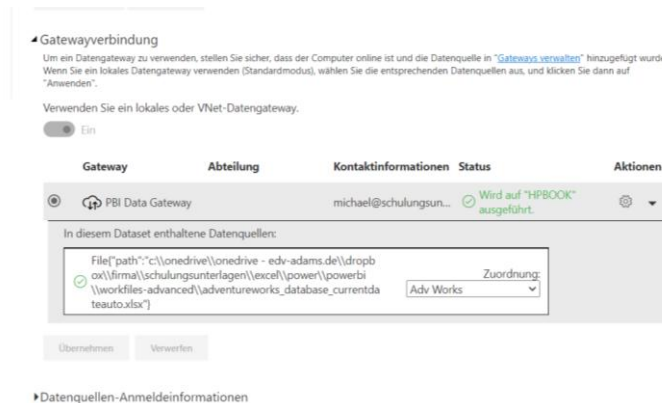
Download and install the data gateway



Select the settings for the desired dataset:



Add or manage a gateway connection:



4.10.3 Import Data from Teams

- In Teams, go to Files.
- Select Open in SharePoint.
- In SharePoint, open the Excel file in the Excel app.
- In the Excel file, select Info and copy the path.
- In Power BI, select Get Data > From Web > Basic.
- Paste the link from the Excel file.
- Remove "?web=1" from the link.
- Select OK.

4.10.4 Create a Dashboard

A dashboard is a collection of visuals from other reports. Create a new one:

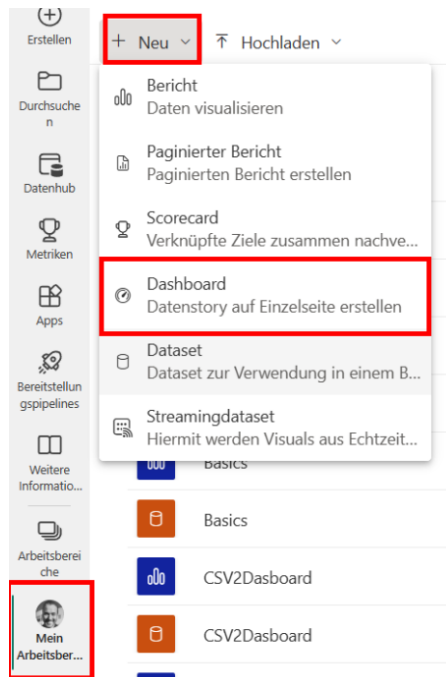


Figure 64

Report pages can then be added to a dashboard:

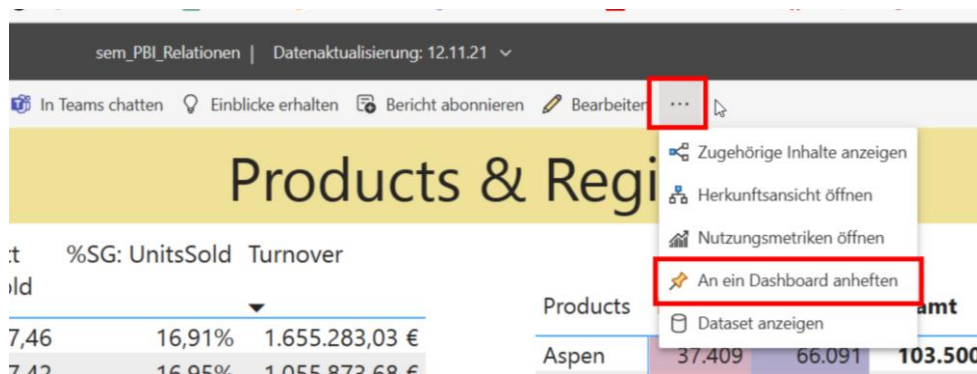


Figure 65

4.10.5 Analyze Power BI (Portal) Data in Excel

In the Power BI portal, select the desired project and open the three-dot menu:

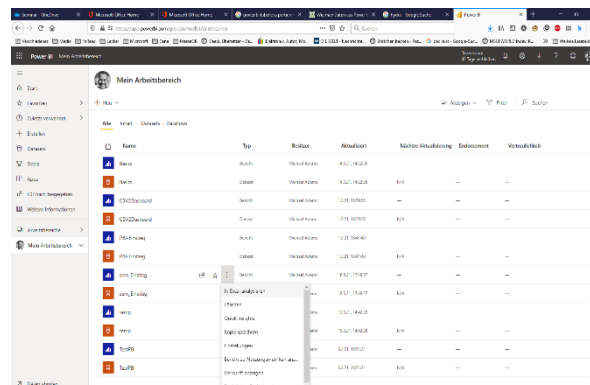


Figure 66

This establishes an MDX connection to Excel:

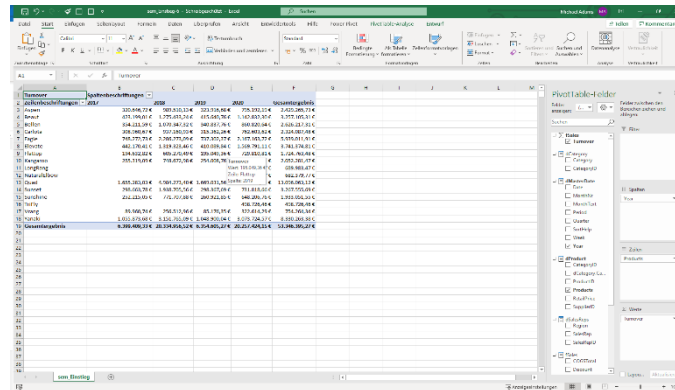


Figure 67

All dashboard fields can now be used; individual measures can be retrieved using CUBEVALUE(), and an inserted slicer can also be taken into account:

=CUBEWERT("pbiazure://api.powerbi.com a782bb76-26ca-4baf-9500-e41af5eb9cc1 Model";"[Measures].[Turnover]";Datenschnitt_dCategory.Category)

4.11 Miscellaneous

4.11.1 Display a List of Image URLs in a Table

First import the table containing the list of image links and change the data type in Power BI Desktop to Image URL:

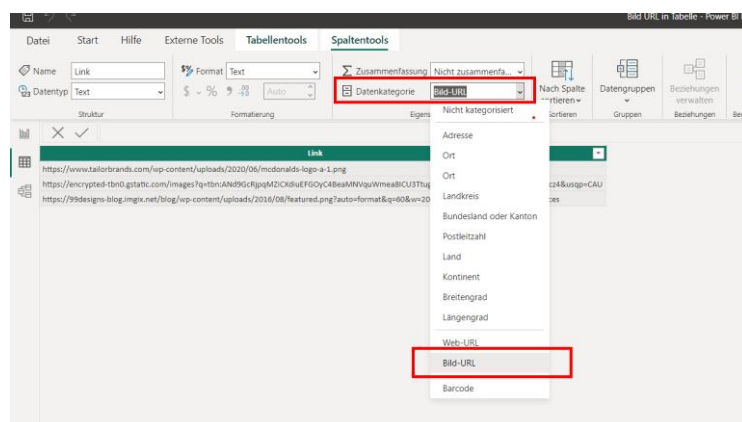


Figure 68

Then insert a table visual and use the table containing the image URLs as its data source:

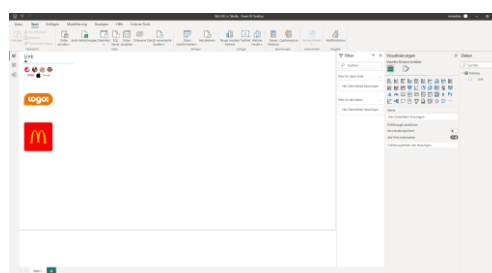


Figure 69

Image scaling can be configured in the visual's formatting properties:

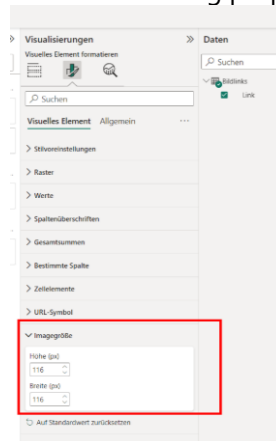


Figure 70

