

<b>1</b>	<b>POWER AUTOMATE</b> .....	<b>7</b>
<b>2</b>	<b>POWER AUTOMATE DESKTOP</b> .....	<b>12</b>
<b>3</b>	<b>POWERPIVOT</b> .....	<b>15</b>
<b>4</b>	<b>POWER BI DESKTOP</b> .....	<b>27</b>
<b>5</b>	<b>POWER BI DESKTOP ADVANCED</b> .....	<b>46</b>
<b>6</b>	<b>POWER BI – PL-300 ZERTIFIZIERUNGSTHEMEN</b> .....	<b>75</b>
<b>7</b>	<b>POWERQUERY</b> .....	<b>102</b>
<b>8</b>	<b>POWER QUERY ADVANCED</b> .....	<b>119</b>

<b>1</b>	<b>POWER AUTOMATE</b> .....	<b>7</b>
1.1	ALLGEMEINES.....	7
1.2	FLOW-VORLAGEN VERWENDEN .....	7
1.3	EINFACHE DATENEINGABE IN EXCEL .....	7
1.4	FLOW-BEISPIELE .....	7
1.4.1	<i>Mail wenn Neue Datei im OneDrive Ordner</i> .....	7
1.4.2	<i>Word zu PDF konvertieren und Nachricht an Teams</i> .....	7
1.4.3	<i>Record Working Hours</i> .....	7
1.4.4	<i>Mailanlagen in OneDrive Ordner kopieren</i> .....	8
1.5	OFFICE SKRIPT AUF ALLE DATEIEN EINES ORDNERS ANWENDEN .....	8
1.5.1	<i>Office Skript aufzeichnen</i> .....	8
1.5.2	<i>Skript mittels eines Flows auf Dateien im Ordner anwenden</i> .....	8
1.6	COMPOSE (VERFASSEN).....	9
1.7	CSV TO EXCEL .....	10
1.8	PERFORMANCE ANPASSEN .....	10
1.8.1	<i>Gleichzeitigkeit für Schleifen</i> .....	10
1.8.2	<i>Parallel Branching</i> .....	11
1.9	MEHRERE DATEIEN AUS ORDNER VERARBEITEN.....	11
<b>2</b>	<b>POWER AUTOMATE DESKTOP</b> .....	<b>12</b>
2.1	SETUP .....	13
2.2	ERSTE BEISPIELE.....	13
2.2.1	<i>Excel</i> .....	13
2.2.2	<i>Taschenrechner</i> .....	13
2.2.3	<i>Neue UI-Elemente zur Auswahl zufügen</i> .....	13
2.2.4	<i>Neues UI-Element aus vorhandener Auswahl hinzufügen:</i> .....	14
<b>3</b>	<b>POWERPIVOT</b> .....	<b>15</b>
3.1	INSTALLATION.....	15
3.2	DATENQUELLEN VERWALTEN .....	16
3.2.1	<i>Daten als Tabelle</i> .....	16
3.2.2	<i>Daten verknüpfen</i> .....	16
3.3	BERECHNUNGEN: DAX (DATA ANALYSIS EXPRESSIONS).....	16
3.3.1	<i>Horizontale Berechnungen</i> .....	16
3.3.2	<i>Gesamtergebnisse aus einer Tabelle berechnen</i> .....	17
3.3.3	<i>Gesamtergebnisse aus mehreren Tabellen berechnen (DAX-Funktionen)</i> .....	17

3.3.4	<i>Eigenen Filterkontext setzen</i>	17
3.3.5	<i>Berechnete Felder verwalten</i>	18
3.3.6	<i>Aufgabe</i>	18
3.4	MASTER-DATE-TABLE VERWENDEN	18
3.4.1	<i>Zielaufgabe</i>	19
3.5	DATENIMPORTE	20
3.5.1	<i>Datenmenge reduzieren</i>	20
3.5.2	<i>Import aus Sharepoint</i>	20
3.5.3	<i>Import aus verschiedenen Quellen</i>	20
3.5.4	<i>Import über Power Query</i>	21
3.6	VERKÄUFER: PERFORMANCE MESSEN	22
3.7	BEISPIEL: VERKNÜPFUNGEN MIT EINER MASTERDATE-TABLE	23
3.7.1	<i>Masterdatumstabelle erstellen</i>	23
3.7.2	<i>Datenmodell bearbeiten</i>	24
3.7.3	<i>Auswertung Jahr/Quartal/Monat/Woche per Slicer und Diagramm</i>	25
3.7.4	<i>Aufgabe</i>	25
3.8	KPI	26
<b>4</b>	<b>POWER BI DESKTOP</b>	<b>27</b>
4.1	DATENIMPORTE	27
4.1.1	<i>CSV-Multiimport</i>	27
4.1.2	<i>Zwischenbericht</i>	28
4.1.3	<i>Berichte veröffentlichen</i>	29
4.1.4	<i>Berichte auf SharePoint veröffentlichen</i>	29
4.1.5	<i>Excel-Import mit Relationen</i>	29
4.1.6	<i>Sharepoint Daten importieren</i>	31
4.2	EINFACHEN BERICHT ERSTELLEN	32
4.2.2	<i>Measures erstellen</i>	33
4.2.3	<i>Visuals einfügen</i>	34
4.2.4	<i>Visuals filtern</i>	35
4.3	TABELLEN PER FUNKTION ERSTELLEN	35
4.3.1	<i>Neue Spalten einfügen</i>	35
4.4	ARTEN VON TABELLENBEZIEHUNGEN	36
4.5	MASTER DATE-TABLE	36
4.6	ZWISCHENBERICHT MIT TREEMAP & SLICER	38
4.6.1	<i>Filter für alle Seiten</i>	38
4.7	DAX-MEASURES (AGGREGATE) ERSTELLEN	38
4.8	DASHBOARDS ERSTELLEN	39
4.8.1	<i>Matrix (Pivot Tabelle)</i>	39
4.8.2	<i>Säulendiagramm mit Drill-Down</i>	40
4.8.3	<i>Liniendiagramm</i>	41
4.8.4	<i>Karte/Mehrzeilige Zuordnung</i>	41
4.8.5	<i>Slicer (Datenschnitt)</i>	42
4.8.6	<i>Neue Visuals importieren</i>	42
4.9	KPI/BEDINGTE FORMATIERUNG	42
4.10	BERICHTE VERÖFFENTLICHEN	43
4.11	VORJAHRESPERIODEN VERGLEICHEN	43
4.11.1	<i>Calculate</i>	43
<b>5</b>	<b>POWER BI DESKTOP ADVANCED</b>	<b>46</b>
5.1	FILTERKONZEPTE UND FILTERPROPAGATION	46
5.2	DAX	47

---

5.2.1	Rabatt-Prozentklassen zusammenfassen.....	47
5.2.2	Calculate.....	47
5.3	INAKTIVE BEZIEHUNGEN (USERRELATIONSHIP).....	48
5.4	FILTERKONTEXT.....	48
5.4.1	Kontextübergang.....	49
5.5	PARAMETER.....	49
5.5.1	Dynamische Dimensionen (Feldparameter).....	49
5.5.2	Zahlenparameter (Numerischer Bereich).....	50
5.6	VERWAISTE DATENSÄTZE ZÄHLEN/BERECHNEN.....	50
5.7	KI ANALYSEN.....	50
5.7.1	Veränderungen im Diagramm.....	50
5.8	VERTIEFUNG DAX.....	51
5.8.1	Vorperioden.....	51
5.8.2	YTD (Year to date).....	52
5.8.3	IF().....	53
5.8.4	Switch.....	53
5.8.5	Measures im Ordner zusammenfassen.....	54
5.8.6	Gesamtwerte ermitteln um %-Anteile zu rechnen.....	54
5.8.7	Variablen in DAX.....	55
5.8.8	OR/AND/IN.....	56
5.8.9	AND/IN mit FILTER() kombinieren.....	56
5.8.10	Zeitauswertungen.....	56
5.8.11	Kapazitäten ermitteln.....	59
5.8.12	Zeilenkontext.....	59
5.8.13	KEEPFILTERS().....	60
5.8.14	DATATABLE().....	60
5.8.15	Zugriff auf vorherige Zeilen in DAX (Beta).....	61
5.8.16	Lookupvalue.....	61
5.8.17	VALUES().....	61
5.8.18	SELECTEDVALUE().....	61
5.8.19	LOOKUPVALUE().....	61
5.8.20	ALL() ALLSELECTED() ALLEXCEPT().....	61
5.8.21	CONVERT().....	62
5.8.22	SUMMARIZE().....	63
5.9	VISUALS.....	63
5.9.1	Drill through.....	63
5.9.2	Bookmarks und Interaktionen.....	64
5.9.3	Quickinfo.....	64
5.9.4	Small Mutiples.....	65
5.9.5	Map Visuals.....	65
5.9.6	Landkarte.....	66
5.9.7	Flächenkartogramm.....	66
5.9.8	Sparkline.....	66
5.9.9	Bänderdiagramm (Ribbon Chart).....	67
5.9.10	Analysebaum (Decomposition Tree).....	67
5.9.11	Punktogramm.....	68
5.9.12	Visuals Hinzufügen (Bsp.: Infographic Designer).....	68
5.10	POWER BI DIENST.....	69
5.10.1	Online gespeicherte Daten (Datasets) automatisch aktualisieren.....	69
5.10.2	Power BI Data Gateway einrichten.....	71
5.10.3	Daten aus Teams importieren.....	71
5.10.4	Dashboard anlegen.....	72

5.10.5	Daten aus Power BI (Portal) in Excel analysieren .....	72
5.11	VERSCHIEDENES .....	73
5.11.1	Liste von Bildern URLs in Tabelle anzeigen .....	73
<b>6</b>	<b>POWER BI – PL-300 ZERTIFIZIERUNGSTHEMEN .....</b>	<b>75</b>
6.1	POWER QUERY .....	75
6.1.1	Direct Query.....	75
6.1.2	SQL-Abfragen.....	75
6.1.3	Parameter für Datenquellen .....	76
6.1.4	Speichermodus.....	77
6.1.5	MasterDate in Power Query.....	77
6.2	DATA MODEL.....	78
6.2.1	Datumstabelle markieren .....	78
6.2.2	N:M.....	78
6.2.3	Row Level Security .....	79
6.2.4	Kreuzfilter .....	81
6.2.5	Datengranularität anpassen.....	83
6.2.6	Analyse in Excel.....	83
6.2.7	Hierarchie Pfade.....	83
6.2.8	Leistungsoptimierung.....	84
6.3	DAX.....	86
6.3.1	Filterkontext RELATED(), RELATEDTABLE() .....	86
6.3.2	Time Intel.....	87
6.3.3	CALCULATE(), CALCULATETABLE(), FILTER() .....	88
6.3.4	Filtermodifikatoren.....	89
6.3.5	VALUES, HASONEVALUE, SELECTEDVALUE, ISINSCOPE.....	89
6.3.6	Skalare Werte und EVALUATE.....	90
6.3.7	Visual Calculations.....	90
6.3.8	Quickmeasures.....	91
6.3.9	Semi-Additive Measures.....	91
6.3.10	Arten von DAX-Funktionen .....	92
6.4	DASHBOARDS.....	92
6.4.1	Visuals zum Dashboard zufügen.....	92
6.4.2	Bericht zum Dashboard hinzufügen .....	92
6.4.3	Seite entfernen .....	93
6.4.4	Design anpassen .....	93
6.4.5	Alert (Datenwarnung) hinzufügen.....	93
6.4.6	Berichtsarten.....	95
6.4.7	Paginated Reports.....	95
6.4.8	Icons mit Navigation und Mouseover.....	95
6.4.9	Datenanalysen.....	95
6.5	POWER BI SERVICE .....	97
6.5.1	On Prem Data Gateway .....	97
6.5.2	Rechte für Arbeitsbereiche.....	99
6.5.3	Quick Insights.....	101
<b>7</b>	<b>POWERQUERY .....</b>	<b>102</b>
7.1	EINZELNE DATEIEN IMPORTIEREN.....	102
7.2	DATEIEN AUS ORDNER IMPORTIEREN.....	102
7.2.1	Daten importieren .....	102
7.2.2	Automatische Datentyperkennung abschalten.....	102
7.2.3	Mehrere Datenquellen zusammenführen .....	103

7.2.4	<i>Datumstyp anpassen</i> .....	103
7.3	ANGEWENDETE SCHRITTE KOPIEREN.....	104
7.4	DATEIEN AUS EINEM SHAREPOINTORDNER IMPORTIEREN.....	104
7.5	DATENQUELLEN VERKNÜPFEN (ERSATZ FÜR SVERWEIS) .....	104
7.5.1	<i>Dateninhalte auftrennen</i> .....	105
7.5.2	<i>Datentabellen verknüpfen</i> .....	105
7.5.3	<i>Pivot-Tabelle anlegen</i> .....	106
7.6	UNTERSCHIEDLICHE TABELLEN ZUSAMMENFÜGEN.....	107
7.7	SPALTENOPERATIONEN.....	108
7.7.1	<i>Duplikate entfernen</i> .....	108
7.7.2	<i>Duplizieren und Rechnen mit Datum</i> .....	108
7.7.3	<i>Lücken auffüllen</i> .....	108
7.7.4	<i>Spalten aus Beispielen</i> .....	108
7.7.5	<i>Bedingte Spalte</i> .....	108
7.8	PIVOT TRANSFORMATIONEN .....	109
7.8.1	<i>Daten depivotieren</i> .....	109
7.8.2	<i>Daten pivotieren</i> .....	109
7.9	DATENQUALITÄT ANZEIGEN.....	110
7.9.1	<i>Daten bereinigen</i> .....	110
7.10	DATENIMPORTE.....	111
7.10.1	<i>Mappen und Tabellen Multiimport</i> .....	111
7.10.2	<i>Importieren und parsen von XML und JSON-Daten</i> .....	112
7.11	SQL-ABFRAGEN.....	113
7.12	WEB-ABFRAGEN .....	114
7.13	JOIN-ARTEN .....	114
7.13.1	<i>Beispiele für Joins</i> .....	115
7.13.2	<i>Inner Join</i> .....	117
7.13.3	<i>Left Anti Join</i> .....	117
7.14	GRUPPIERUNGEN.....	117
7.14.1	<i>Gruppieren nach mehreren Kriterien</i> .....	117
7.14.2	<i>Gruppierungen mit Untertabellen</i> .....	118
7.15	BEISPIELE FÜR AUSDRÜCKE .....	119
<b>8</b>	<b>POWER QUERY ADVANCED .....</b>	<b>119</b>
8.1	KOMPLEXE EXCEL-DATEIEN AUSWERTEN .....	119
8.1.1	<i>Einfacher Mittelwert mit List()</i> .....	119
8.1.2	<i>Dateinamen und Blattnamen in eine Liste übernehmen</i> .....	119
8.1.3	<i>Ältere Duplikate löschen</i> .....	120
8.1.4	<i>Datumsliste aus von..bis erzeugen</i> .....	120
8.1.5	<i>Vorperioden mittels SelfJoin ermitteln</i> .....	121
8.1.6	<i>Webabfrage Aktien, Einstieg M</i> .....	122
8.2	M-CODE.....	122
8.2.1	<i>Aufbau M-Code</i> .....	122
8.2.2	<i>Hilfen</i> .....	123
8.2.3	<i>Datentypen</i> .....	123
8.2.4	<i>Arten von Werten</i> .....	123
8.2.5	<i>Tabelle erstellen (Tabellenkonstruktor)</i> .....	124
8.2.6	<i>Referenzen</i> .....	124
8.2.7	<i>Each und _</i> .....	125
8.3	FRAGMENTIERTE DATEN (MAPPEN/TABELLEN) IMPORTIEREN.....	125
8.3.1	<i>EineMappe, alle Blätter/Spalten gleich, pivотиert, Lösung mit Hilfsspalten</i> .....	125

---

8.3.2	<i>Mehrere Mappen aus Ordner, alle Blätter/Spalten gleich, pivotiert, Lösung mit Hilfsspalten</i>	126
8.3.3	<i>Mehrere Mappen aus Ordner, alle unterschiedliche Spalten, pivotiert, Lösung mit M und Function</i>	126
8.4	NÜTZLICHE M-FUNKTIONEN.....	128
8.5	IMPORTE IN DER PRAXIS.....	129
8.5.1	<i>EineMappe, alle Blätter/Spalten gleich, pivotiert, Lösung mit Hilfsspalten.....</i>	129
8.5.2	<i>Mehrere Mappen aus Ordner, alle Blätter/Spalten gleich, pivotiert, Lösung mit Hilfsspalten</i>	129
8.6	FUNCTIONS UND PARAMETER .....	129
8.6.1	<i>Functions definieren.....</i>	129
8.6.2	<i>Parameterabfragen definieren (Viele Dateien auf einmal transformieren).....</i>	130
8.6.3	<i>Daten aus Excel an Abfrage übergeben.....</i>	131
8.6.4	<i>Bei Formula Firewall Fehlermeldung .....</i>	132
8.7	TECHNIKEN .....	132
8.7.1	<i>Parameter.....</i>	132
8.8	GEMISCHTES .....	134
8.8.1	<i>Zeilen einfügen .....</i>	134
8.8.2	<i>Vortag berechnen .....</i>	134
8.8.3	<i>Datediff.....</i>	134
8.8.4	<i>Daten von...bis in Zeilen auflisten.....</i>	134

# 1 Power Automate

## 1.1 Allgemeines

- RPA: Robotic Process Automation – ermöglicht die Einbindung auch alter (legacy) Applikationen

## 1.2 Flow-Vorlagen verwenden

Bei Mailingang Anlage im OneDrive Ordner ablegen

Vorlage: Office 365-E-Mail-Anlagen in OneDrive for Business speichern

[Office 365-E-Mail-Anlagen in angegebenem OneDrive for Business-Ordner speichern | Microsoft Power Automate](#)

## 1.3 Einfache Dateneingabe in Excel

Ordner: Eingaben in Excel

## 1.4 Flow-Beispiele

### 1.4.1 Mail wenn Neue Datei im OneDrive Ordner

- Neuer Automated Flow
- OneDrive: When a File is Created, zu überwachenden Ordner wählen
- Send Mail: Send an Email (V2)

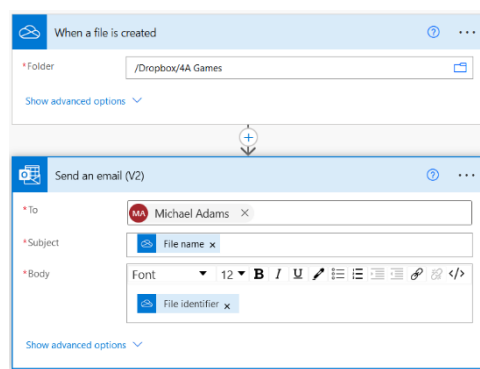


Abbildung 1

### 1.4.2 Word zu PDF konvertieren und Nachricht an Teams

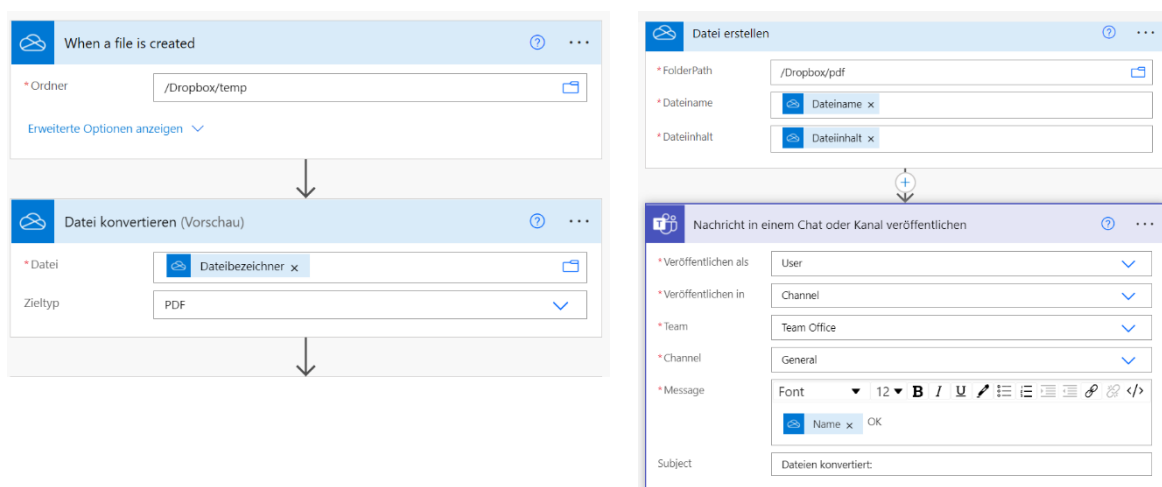


Abbildung 2

### 1.4.3 Record Working Hours

```
formatDateTime(triggerOutputs()['headers']['x-ms-user-timestamp'],'hh:mm tt')
if(triggerBody()['boolean'],'Start','End')
```

#### 1.4.4 Mailanlagen in OneDrive Ordner kopieren

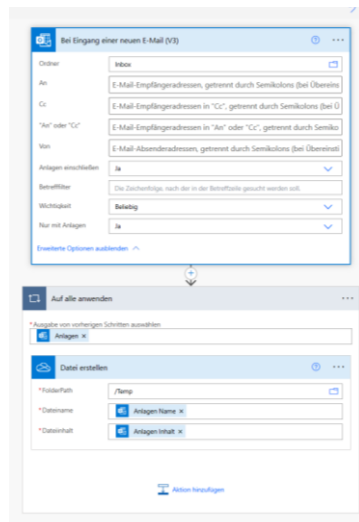


Abbildung 3

## 1.5 Office Skript auf alle Dateien eines Ordners anwenden

### 1.5.1 Office Skript aufzeichnen

Ein Office Skript ist mit einem Makro vergleichbar, mit dem Unterschied, dass diese Skripte später ganz einfach auf beliebige Dateien angewendet werden kann.

In Online Excel: *Automatisieren/Aufzeichnen*:

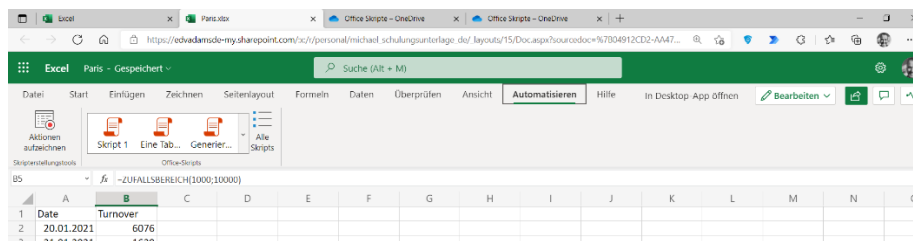


Abbildung 4

Danach die aufzuzeichnenden Aktionen durchführen. Nach der Aufzeichnung kann das Skript umbenannt werden.

Ab jetzt kann eine beliebige weitere Datei geöffnet werden und das Skript kann über *Ausführen* auf diese Datei angewendet werden.

### 1.5.2 Skript mittels eines Flows auf Dateien im Ordner anwenden

Steps:

- 1) Manuell getriggerten Flow anlegen
- 2) OneDrive: *Dateien im Ordner auflisten*
- 3) ExcelOnline: *Skript Ausführen*, für *Datei* wählen wir die ID auf der vorherigen Ordnerauflistung:

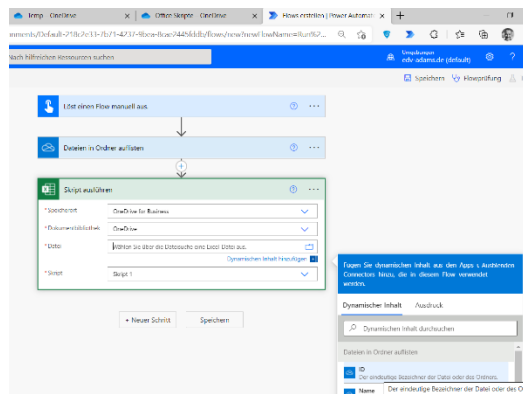


Abbildung 5

Power Automate erstellt automatisch ein Schleife, die alle Dateien durchgeht und das Skript auf diese anwendet:

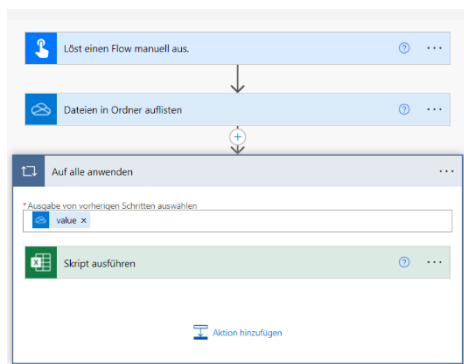


Abbildung 6

- 4) Innerhalb der Scheife:
  - a. Bedingung einfügen: *wenn endet mit .xlsx*
  - b. Skript in den *Dann-Zweig* ziehen

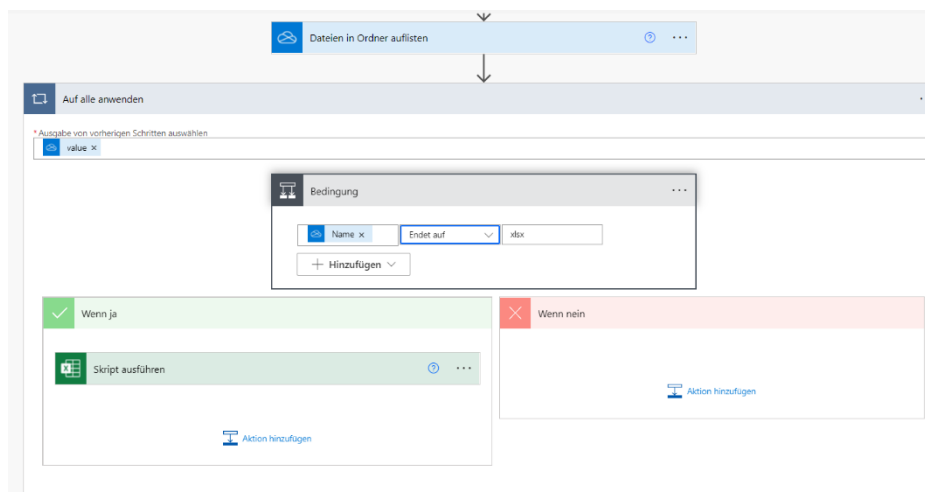


Abbildung 7

Damit wird das Skript innerhalb des Ordners nur auf Excel Dateien angewendet.

### 1.6 Compose (Verfassen)

Dient der Definition von Variablen.

Beispiel: Definition eines Arrays

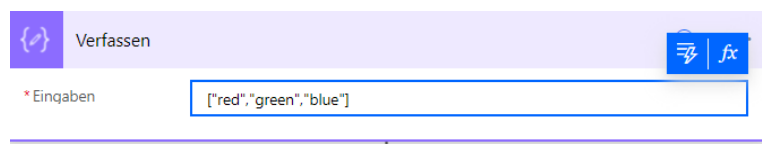


Abbildung 8

Element 2 ausgeben (gezählt wird ab Null): outputs('Verfassen')[1]

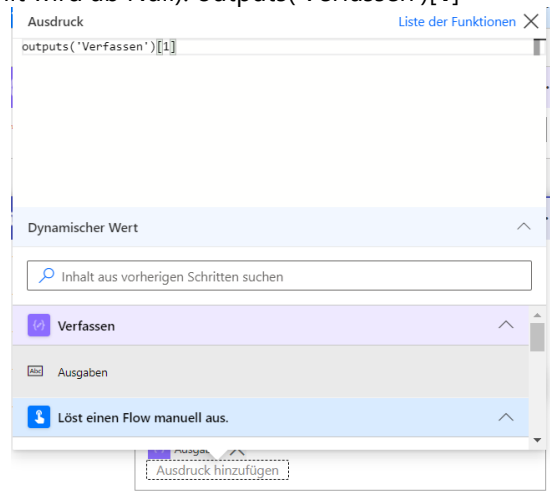


Abbildung 9

## 1.7 CSV to Excel

Hinweis: Das Modul CSV to Excel von Cloudmersive ist ein *Premiumprodukt* und kann bei Cloudmersive kostenlos aktiviert werden.

Ablauf:

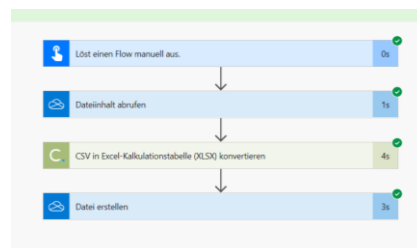


Abbildung 10

## 1.8 Performance anpassen

### 1.8.1 Gleichzeitigkeit für Schleifen

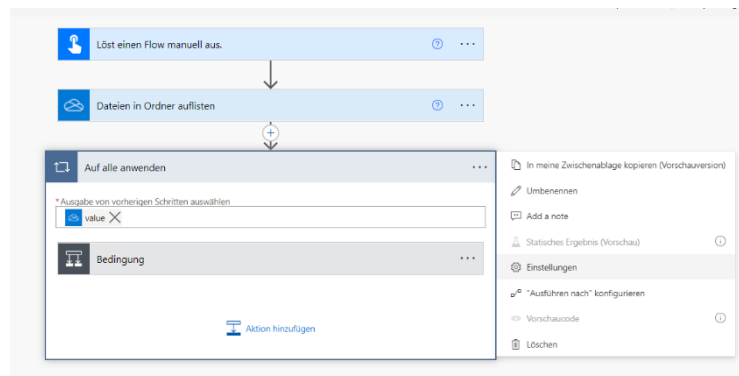


Abbildung 11

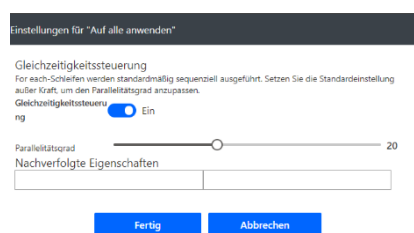


Abbildung 12

Schleifen können gleichzeitig ausgeführt werden und über *Parallelitätsgrad* kann die Verarbeitungsgeschwindigkeit erhöht werden:

**Achtung bei Variablen-Abhängigkeiten!** Wenn das Ergebnis einer Schleife von der anderen abhängt, sollte die Parallelverarbeitung nicht eingesetzt werden.

### 1.8.2 Parallel Branching

Prozesse können parallel ausgeführt werden:

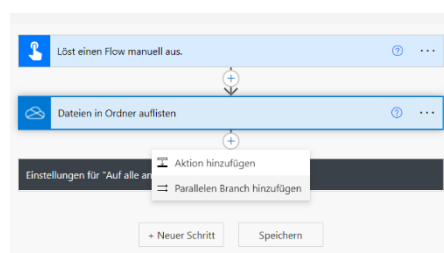


Abbildung 13

## 1.9 Mehrere Dateien aus Ordner verarbeiten

Ordner: MultiFiles

- Dateien im Ordner abrufen und Ordnerpfad eingeben
- Excel starten mit Datei „Order Tracker“
- For Each Schleife einfügen mit Iterationsvariable *Files*
- *Rename Files* in Schleife einfügen mit:
  - umzubenennende Datei: `%CurrentItem%`
  - neuem Dateinamen „Extract“
  - *If exists: Overwrite*
- Anwendung starten
  - Anwendungspfad von Word mit `Winword.exe` einfügen und ggf. alle Anführungszeichen entfernen

- Argument: /t mit "Dateipfad\Extract.docx" in ""
- Eine beliebige Datei aus Multifiles\data kopieren und in Extract.docx umbenennen und in Word öffnen
- Über die Schaltfläche *UI-Element hinzufügen* (bei *UI-Elemente*) gewünschte UI-Elemente aus der Tabelle im Worddokument zufügen
- Zugefügte Elemente sinnvoll umbenennen
- *Daten aus Fenster extrahieren* zufügen (Abbildung 14)
  - Bei *Fester* das erste Element (OrderNr) wählen
  - Daten Speichern in *Variable*
  - Erzeugt Variable sinnvoll benennen, z.B. %OrderNr%
  - Diesen Vorgang für die restlichen Elemente wiederholen

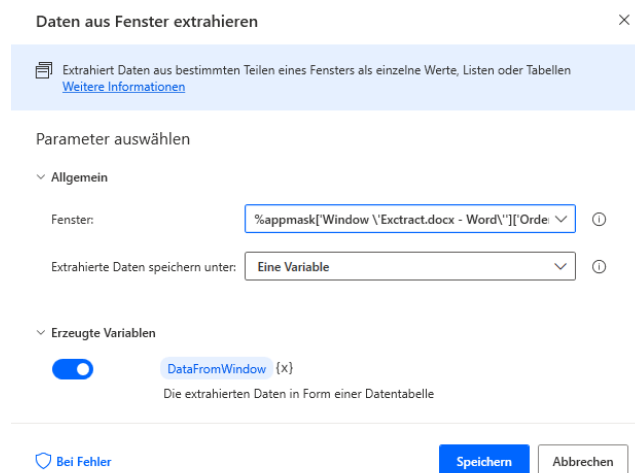


Abbildung 14

- *Erste freie Zeile aus Excel* zufügen
- *In Excel Arbeitsblatt schreiben* zufügen und folgendes einstellen:

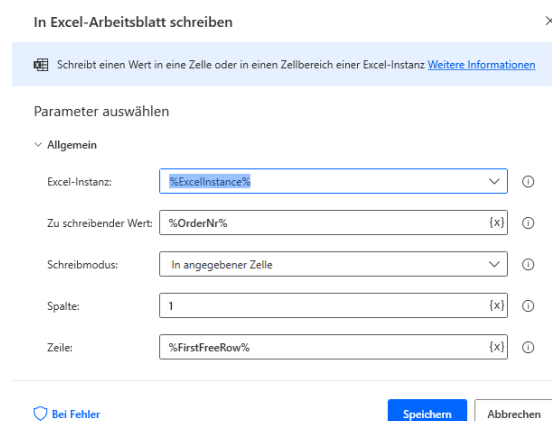


Abbildung 15

- Diesen Schritt für die restlichen Elemente wiederholen

## 2 Power Automate Desktop

## 2.1 Setup

- Login in Office.com
- Alle Apps/Power Automate
- Oben rechts: Install

## 2.2 Erste Beispiele

### 2.2.1 Excel

- Links aus Aktionen Excel/Starten
- Meldung Anzeigen
- Excel beenden

### 2.2.2 Taschenrechner

Aktion Anwendung starten (C:\Windows\System32\calc.exe):

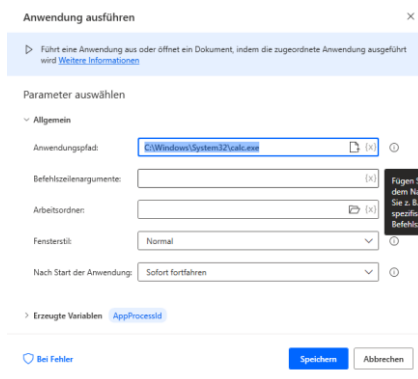


Abbildung 16

Desktop Recorder starten und die gewünschten Aktionen klicken:

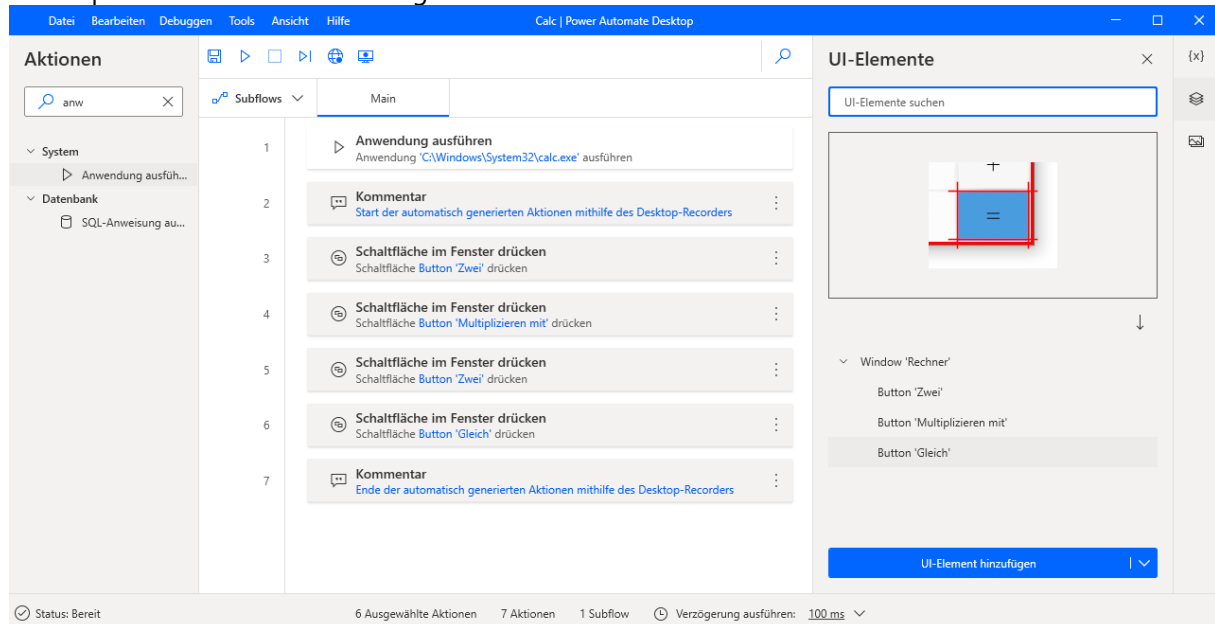


Abbildung 17

### 2.2.3 Neue UI-Elemente zur Auswahl zufügen

Über die drei Buttons auf der rechten Seite kann auf UI-Elemente umgeschaltet werden und über *UI Element hinzufügen* können weitere Bedienelemente des Taschenrechners bei gedr. STRG-Taste hinzugefügt werden.

Anschließend kann per Doppelklick auf einen vorhandenen Schritt ein anderes gespeichertes Element gewählt werden.

#### 2.2.4 Neues UI-Element aus vorhandener Auswahl hinzufügen:

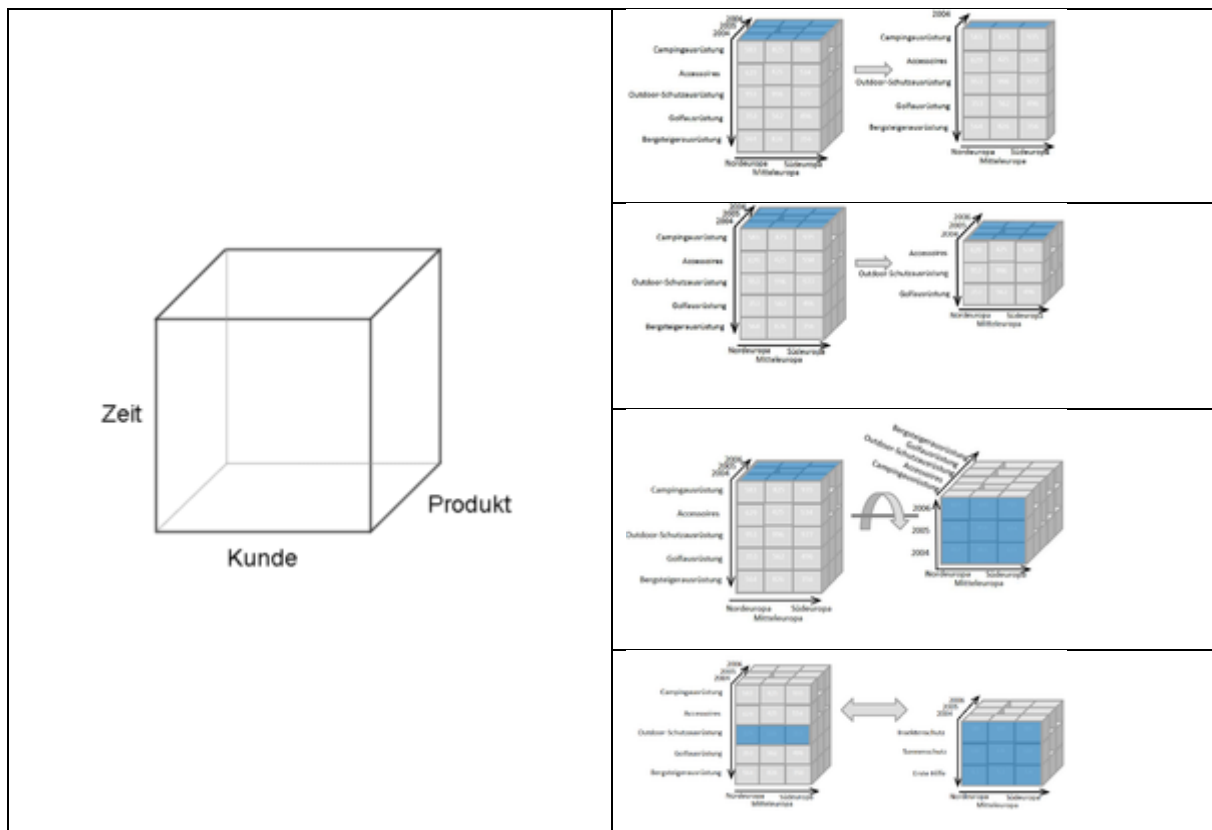
- Links in der Liste nach *Auf UI-Element in Fenster klicken* suchen
- Einzufügende Position in Liste wählen
- Doppelklick auf *Auf UI-Element in Fenster klicken*
- *UI-Element* Auswählen

**TIPP:** Der vorherige Schritt kann einfach kopiert und per Doppelklick angepasst werden

### 3 Powerpivot

Bei der Verarbeitung großer Datenmengen bietet Powerpivot folgende Vorteile:

- Trennung Reporting Datenhaltung
- Geschwindigkeit (Engine: xVelocity)
- Speicher sparen
- Daten verknüpfen (auch aus verschiedenen Quellen)
- Einsatz von OLAP (Online Analytical Processing) für Business Intelligence Anwendung (hier Powerpivot)
- Betrachtung eines OLAP-Cubes mittels:
  - Slicing
  - Dicing
  - Pivoting
  - Drill-Down



#### 3.1 Installation

Wählen Sie unter Optionen/Add-Ins/COM Add-Ins das Powerpivot Add-In.

### 3.2 Datenquellen verwalten

Es gibt zahlreiche Möglichkeiten Daten in Powerpivot einzufügen.

#### 3.2.1 Daten als Tabelle

Markieren Sie dazu die den gewünschten Datenbereich und wählen *Start/Als Tabelle formatieren*. Unter *Tabellentools* empfiehlt es sich anschließend einen Namen für die Tabelle zu definieren.

#### 3.2.2 Daten verknüpfen

Wechseln Sie in Powerpivot in die *Diagrammsicht* und verknüpfen korrespondierende Datenfelder per Drag&Drop:

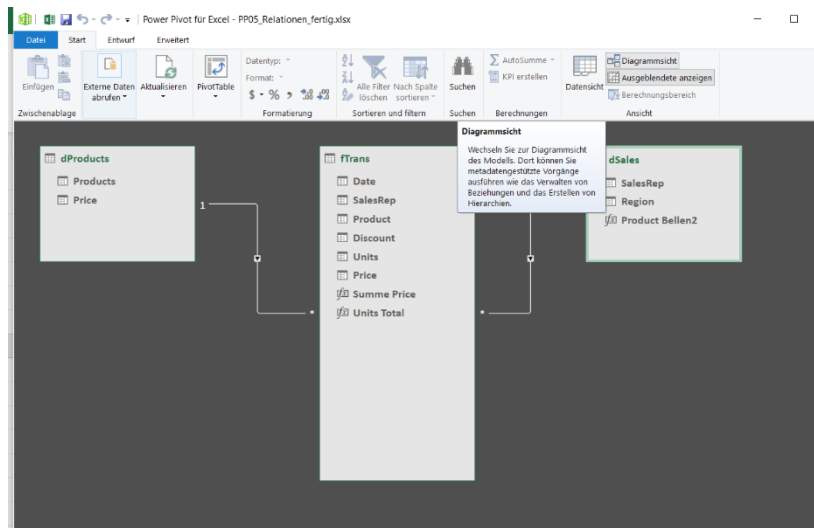


Abbildung 18

### 3.3 Berechnungen: DAX (Data Analysis Expressions)

#### 3.3.1 Horizontale Berechnungen

Horizontale Berechnungen erfolgen Zeile für Zeile. Wegen höherer Verarbeitungsgeschwindigkeit empfiehlt sich diese in Powerpivot durchzuführen. Hier spricht man von expliziten Berechnungen, im Gegensatz zu impliziten Berechnungen, die in der Pivot Tabelle über Berechnete Felder erfolgen.

##### 3.3.1.1 Berechnungen In einer Tabelle

Benötigte Daten: **PP05\_Relationen.xlsx**

Ziel:

Zeilenbeschriftungen	Summe von Brutto
Aspen	24,99 €
Bellen	27,31 €
Carlota	27,31 €
Crested Beaut	29,75 €
Doublers	95,14 €
FlatTop	32,73 €
Majestic Beaut	35,70 €
Quad	40,46 €
Sunbell	29,75 €
Sunset	27,97 €
Sunshine	23,74 €
V-Rang	22,61 €
<b>Gesamtergebnis</b>	<b>417,45 €</b>

**Daten referenzieren:** Tabellenname[Feldname]

**Brutto zeilenweise ausrechnen:** = fTrans[Price]\*1,19

### 3.3.1.2 Berechnungen in verknüpften Tabellen

Auf Felder aus anderen Tabellen kann über RELATED() zugegriffen werden. Beachten Sie, dass die anderen Tabellen verknüpft sein müssen.

**Beispiel:** =RELATED(tProduct[Price])\*fTrans[Units]\*(1-fTrans[Discount])

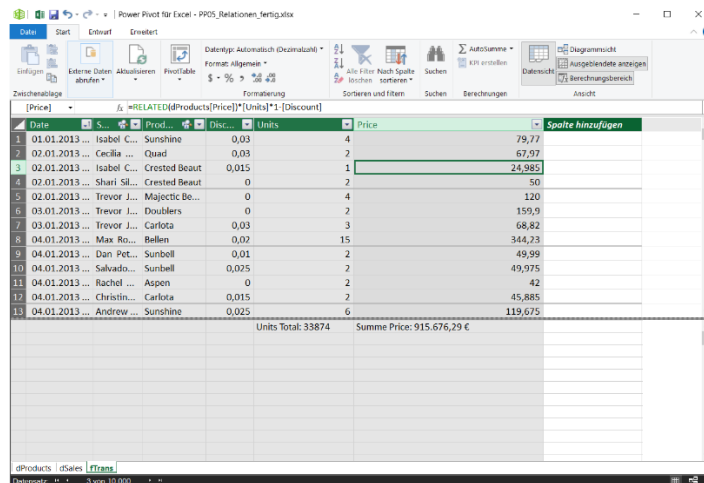


Abbildung 19

### 3.3.2 Gesamtergebnisse aus einer Tabelle berechnen

Benötigte Daten: **PP05\_Relationen.xlsx**

Ziel:

Zeilenbeschriftungen	Anzahl	Gesamtumsatz
East	5614	146141,5583
MidWest	12174	326109,4955
South	5908	158819,8718
West	10178	274631,2238
<b>Gesamtergebnis</b>	<b>33874</b>	<b>905702,1492</b>

Ein Berechnungsfeld wird wie folgt aufgebaut: Name:=Ausdruck

**Beispiel:** Anzahl Umsätze:=count([Price]\*1,19)

### 3.3.3 Gesamtergebnisse aus mehreren Tabellen berechnen (DAX-Funktionen)

Um Gesamtergebnisse aus Feldern anderer Tabellen zu bilden, benötigt man die DAX-Funktionen (Data Analysis Expressions)

**Aufbau:** Sumx(Domäne;Ausdruck)

**Beispiel:** Gesamtumsatz:=SUMX(fSales;fSales[Units]\*RELATED(dProducts[RetailPrice]))

Weitere Funktionen: AVERAGEX(), COUNTX(), MAXX(), MINX()

### 3.3.4 Eigenen Filterkontext setzen

Datei: PP05\_Relationen\_Tabellenbereiche.xlsx

Ziel:

Zeilenbeschriftungen	Umsatz	Umsatz Bellen
----------------------	--------	---------------

East	147.813,17 €	16.911,35 €
MidWest	329.653,54 €	32.905,85 €
South	160.532,91 €	17.255,77 €
West	277.676,68 €	28.223,96 €
<b>Gesamtergebnis</b>	<b>915.676,29 €</b>	<b>95.296,91 €</b>

- Standardmäßig filtert die Pivot-Tabelle
- Mit Filter() übergeht man den Filterkontext der Tabelle für eigene Auswertungen
- Filter() wird als Domäne in SUMX() & co. Eingesetzt
- Beispiel: Product Bellen  

$$\text{UmsatzBellen} := \text{SUMX}(\text{FILTER}(\text{trans}; [\text{Product}] = \text{"Bellen"}); \text{RELATED}(\text{prod}[\text{Price}]) * [\text{Units}] * (1 - [\text{Discount}]))$$

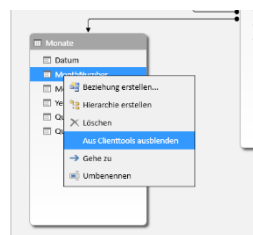
#### Hinweise:

- Formatierungen besser im Backend setzen, dadurch erfolgt die Formatübernahme automatisch in jede neue Pivot-Tabelle
- Explizite Berechnungen
  - können in allen Pivot-Tabellen verwendet werden
  - Verfügbar sind alle PPivot Funktionen
  - Berechnete Felder von Pivot-Tabellen funktionieren nicht im Kontext verknüpfter Tabellen
- Implizite Berechnungen
  - Direkt in Pivot-Tabelle

### 3.3.5 Berechnete Felder verwalten

Powerpivot/Berechnete Felder

Ausblenden aus Pivot-Feldern per Kontextmenü:



### 3.3.6 Aufgabe

Erstellen Sie eine Pivot-Tabelle mit den folgenden Spalten

- Gesamtanzahl von Umsätzen
- Summe Umsätze abzüglich Rabatt in EUR pro Region
- Summe Umsätze in US\$ umgerechnet (Kurs = 1,1)
- Summe Umsätze für Produkt Quad in EUR

## 3.4 Master-Date-Table verwenden

Sollen Informationen nach Datumsintervallen (Monat, Quartal usw.) ausgewertet werden, empfiehlt sich der Einsatz einer Master-Date-Table. In dieser Tabelle steht zunächst jedes Einzeldatum eines gewünschten Zeitraum. Feinere Unterteilungen (Monat, Quartal usw.) erfolgen über geeignete Datumsfunktionen.

Datum	MonthNumber	Month	Year	QuarterNum	Quarter
01.01.2013 00:00:00	1	Jan	2013	Q1	2013 - Q1
02.01.2013 00:00:00	1	Jan	2013	Q1	2013 - Q1
03.01.2013 00:00:00	1	Jan	2013	Q1	2013 - Q1
04.01.2013 00:00:00	1	Jan	2013	Q1	2013 - Q1
05.01.2013 00:00:00	1	Jan	2013	Q1	2013 - Q1
06.01.2013 00:00:00	1	Jan	2013	Q1	2013 - Q1
07.01.2013 00:00:00	1	Jan	2013	Q1	2013 - Q1
08.01.2013 00:00:00	1	Jan	2013	Q1	2013 - Q1
09.01.2013 00:00:00	1	Jan	2013	Q1	2013 - Q1
10.01.2013 00:00:00	1	Jan	2013	Q1	2013 - Q1
11.01.2013 00:00:00	1	Jan	2013	Q1	2013 - Q1
12.01.2013 00:00:00	1	Jan	2013	Q1	2013 - Q1

Abbildung 20 – Master Date Table

**Hinweis:** Sofern Powerpivot die Datengrundlage für eine Pivot-Tabelle bildet, können keine Gruppierungsfunktionen innerhalb der Pivot-Tabelle verwendet werden. Es muss eine Master-Date-Table verwendet werden, um Datumsgruppierungen erstellen zu können.

Anschließend wird die Datumstabelle (Im Bild, Tabelle *Monate*) mit dem Datumsfeld der Rohdaten verknüpft:

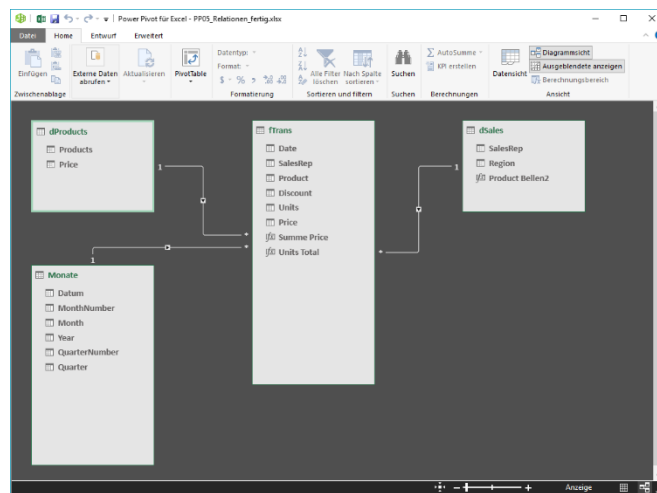


Abbildung 21

### 3.4.1 Zielaufgabe

Das Ziel ist eine Konsolidierung der Umsätze nach Jahr und Quartal. (Abbildung 22)

Datei: PP05\_Relationen.xlsx

Ziel:

Zeilenbeschriftungen	Units Total
2013 - Q1	1494
2013 - Q2	1539
2013 - Q3	1216
2013 - Q4	12381
2014 - Q1	1311
2014 - Q2	1280
2014 - Q3	1421
2014 - Q4	13232
<b>Gesamtergebnis</b>	<b>93874</b>

Abbildung 22

### 3.5 Datenimporte

#### 3.5.1 Datenmenge reduzieren

Benötigte Datei: PowerPivot02-01-LotsOfNonNormalizedData.xlsx

Nicht normalisierte Datenquellen in PowerPivot importieren:

Aus anderen Quellen/Excel Datei, hier ggf. auf *Tabelle enthält Überschriften* klicken

Vergleichen sie anschließend die Dateigröße.

#### 3.5.2 Import aus Sharepoint

Excel/Power BI: Daten abrufen/Aus Datei/Sharepoint Ordner

z.B.: <https://edvadsamsde.sharepoint.com/sites/Seminare/>

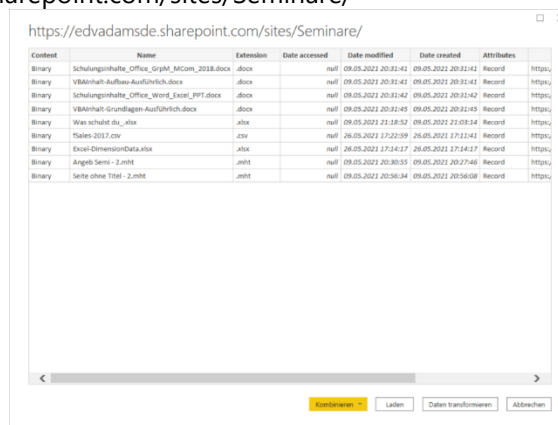


Abbildung 23

#### 3.5.3 Import aus verschiedenen Quellen

Benötigte Dateien:

PowerPivot06-SourceData2014-2015.accdb

PowerPivot06-SourceData.txt

Daten können aus verschiedenen Quellen importiert und dem Datenmodell hinzugefügt werden.

Wählen Sie dazu in Powerpivot *Start/Aus anderen Quellen*:

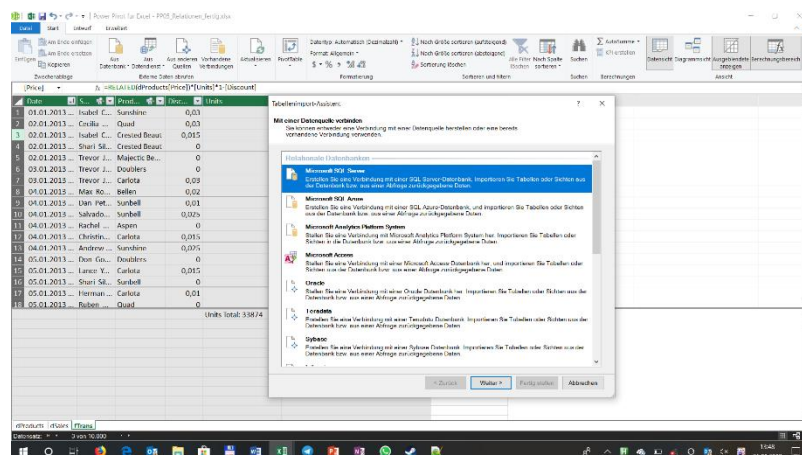


Abbildung 24

**Hinweis:** Wenn Sie externe Daten importieren, werden in Powerpivot alle Werte aktualisiert in dem Moment da sich einer der Verknüpften Datenquellen ändert. Dazu ist lediglich ein Klick auf *Aktualisieren* in Powerpivot erforderlich.

### 3.5.4 Import über Power Query

Benötigte Daten: Ordner PowerQuery:

- Excel-Dimensions.xlsx
- fSales-2019.csv

Das nächste Kapitel zeigt, wie Power Query aus Excel heraus aufgerufen wird.

#### 3.5.4.1 Datenabruf in Excel

- Daten/Daten abrufen/Aus Datei/Aus Arbeitsmappe
- Hier wählen Sie den Tabellenbereich *dProduct* und klicken auf *Daten/Transformieren* um die Daten in Power Query anzuzeigen
- Da allerdings keine weiteren Transformationen benötigt werden, wählen wir den Befehl *laden in...*

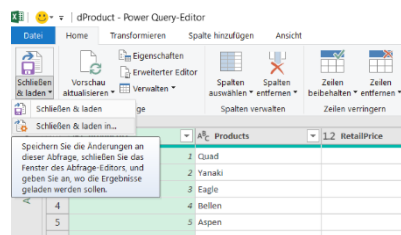


Abbildung 25

- Im nachfolgenden Dialog wählen wir nur Verbindung erstellen. Ferner aktivieren Sie die Checkbox *Dem Datenmodell hinzufügen*, um die Daten in Powerpivot verfügbar zu machen.

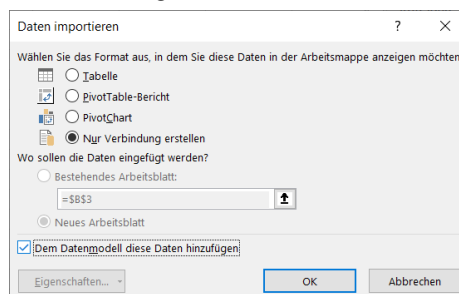


Abbildung 26

Als nächstes rufen wir die *fSales-2019.csv* auf die gleiche Weise ab und transformieren wie im Kap.

4.1.1.1 beschrieben die folgenden Spalten:

- ISO-Date
- Units Sold
- Discount

Im Ergebnis sehen unsere Daten wie folgt aus:



Zeilenbeschriftungen	KleinstesDatum	GrößtesDatum	AnzahlTage	Summe von Umsatz	UmsatzProTag	UmsatzProTag	Status
Althaus	11.01.2008	05.04.2008	85	25.170,76 €	296,13 €	🟡	
Aue	25.01.2008	10.03.2008	45	16.190,84 €	359,80 €	🟡	
Backhaus	07.03.2008	25.03.2008	18	4.945,22 €	274,73 €	🟡	
Blechinger	02.01.2008	01.04.2008	90	29.699,10 €	329,99 €	🟡	
Bruch	15.01.2008	01.04.2008	77	14.858,52 €	192,97 €	🔴	
Freytag	18.01.2008	05.03.2008	47	8.032,35 €	170,90 €	🔴	
Gasser	22.01.2008	26.03.2008	64	22.278,56 €	348,10 €	🟡	
Hauser	04.01.2008	25.03.2008	81	24.329,67 €	300,37 €	🟡	
Heike	07.01.2008	03.04.2008	87	36.320,63 €	417,48 €	🟡	
Herrmann	09.01.2008	03.03.2008	54	15.015,03 €	278,06 €	🟡	
Lemke	18.02.2008	01.04.2008	43	13.864,87 €	322,44 €	🟡	
Lüdemann	25.01.2008	28.03.2008	63	52.629,23 €	835,38 €	🟢	
Müller	05.03.2008	17.03.2008	12	10.072,54 €	839,38 €	🟢	
Nagel	01.02.2008	02.04.2008	61	38.042,49 €	623,65 €	🟢	
Rau	18.01.2008	05.04.2008	78	10.170,93 €	130,40 €	🔴	
Sander	03.01.2008	17.03.2008	74	43.787,08 €	591,72 €	🟡	
Uldall	10.03.2008	31.03.2008	21	18.518,36 €	881,83 €	🟢	
Wanka	15.02.2008	25.03.2008	39	23.812,38 €	610,57 €	🟢	
Weimar	05.02.2008	03.04.2008	58	7.132,74 €	122,98 €	🔴	
Wolf	21.01.2008	02.04.2008	72	32.477,25 €	451,07 €	🟡	
<b>Gesamtergebnis</b>	<b>02.01.2008</b>	<b>05.04.2008</b>	<b>94</b>	<b>447.348,55 €</b>	<b>4.759,03 €</b>	🟢	

### 3.7 Beispiel: Verknüpfungen mit einer MasterDateTable

Benötigte Dateien: PP05\_Relationen.xlsx

Ziel:

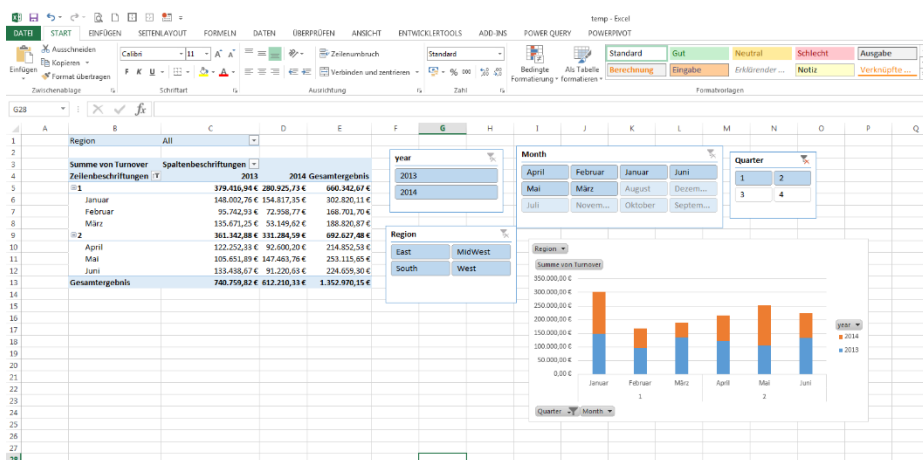


Abbildung 29

#### 3.7.1 Masterdatumstabelle erstellen

Monat holen: =Month()

Monatsnamen holen: =Format(Date;"MMMM")

Quartalsnummer holen: =Ceiling(Monatsnummer;3)/3 - teilt durch 3 und rundet auf 3 auf.

**Hinweis:** Formatierung als Text: Format(Wert;"00#")

[Quarter]	[Date]	[Year]	[QuarterNumber]	[Quarter]
1	01.01.2013 00:00:00	2013	1	2013 - Q1
1	02.01.2013 00:00:00	2013	1	2013 - Q1
1	03.01.2013 00:00:00	2013	1	2013 - Q1
1	04.01.2013 00:00:00	2013	1	2013 - Q1
1	05.01.2013 00:00:00	2013	1	2013 - Q1
1	06.01.2013 00:00:00	2013	1	2013 - Q1
1	07.01.2013 00:00:00	2013	1	2013 - Q1
1	08.01.2013 00:00:00	2013	1	2013 - Q1
1	09.01.2013 00:00:00	2013	1	2013 - Q1
1	10.01.2013 00:00:00	2013	1	2013 - Q1
1	11.01.2013 00:00:00	2013	1	2013 - Q1
1	12.01.2013 00:00:00	2013	1	2013 - Q1
1	13.01.2013 00:00:00	2013	1	2013 - Q1

Abbildung 30

### 3.7.2 Datenmodell bearbeiten

Jetzt werden alle Tabellen, inkl. der Datumstabelle verknüpft:

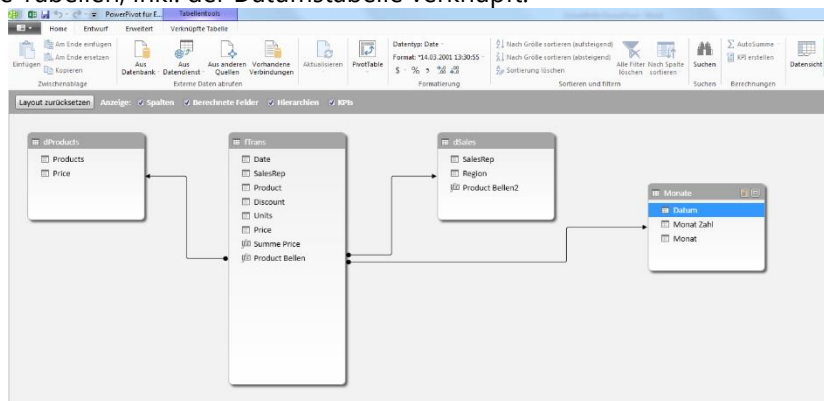


Abbildung 31

Ggf: Gewünschte Daten in die Verknüpfungstabelle holen oder berechnen, hier Price:

[Date]	[Sales]	[Product]	[Discount]	[Units]	[Price]
03.01.2013 00:00:00	Trevor Jones	Doublers	0	2	75,53
03.01.2013 00:00:00	Trevor Jones	Carlota	0,03	3	22,92
04.01.2013 00:00:00	Max Rodgers	Bellen	0,02	15	22,93
04.01.2013 00:00:00	Dan Peter...	Sunbell	0,01	2	24,99
04.01.2013 00:00:00	Salvador B...	Sunbell	0,025	2	24,98
04.01.2013 00:00:00	Rachel Go...	Aspen	0	2	21
04.01.2013 00:00:00	Christna F...	Carlota	0,015	2	22,94
04.01.2013 00:00:00	Andrew M...	Sunshine	0,025	6	19,99
05.01.2013 00:00:00	Don Go...	Doublers	0	4	75,53
05.01.2013 00:00:00	Lance Yates	Carlota	0,015	3	22,94
05.01.2013 00:00:00	Shari Silva	Sunbell	0	2	25
05.01.2013 00:00:00	Herrman W...	Carlota	0,01	1	22,94
05.01.2013 00:00:00	Ruben Nu...	Quad	0	19	34

Abbildung 32

Danach werden die Gesamtergebnisse gebildet:

[Date]	[Sales]	[Product]	[Discount]	[Units]	[Price]
01.01.2013 ...	Isabel C...	Sunshine	0,03	4	79,77
02.01.2013 ...	Cecilia ...	Quad	0,03	2	67,97
02.01.2013 ...	Isabel C...	Crested Beaut	0,015	1	24,99
02.01.2013 ...	Shari Sil...	Crested Beaut	0	2	50
02.01.2013 ...	Trevor J...	Majectic Be...	0	4	120
03.01.2013 ...	Trevor J...	Doublers	0	2	159,9
03.01.2013 ...	Trevor J...	Carlota	0,03	3	68,82
04.01.2013 ...	Max Ro...	Bellen	0,02	15	344,23
04.01.2013 ...	Dan Pet...	Sunbell	0,01	2	49,99
04.01.2013 ...	Salvado...	Sunbell	0,025	2	49,98
04.01.2013 ...	Rachel ...	Aspen	0	2	42
04.01.2013 ...	Christin...	Carlota	0,015	2	45,89
04.01.2013 ...	Andrew ...	Sunshine	0,025	6	119,68
05.01.2013 ...	Don Go...	Doublers	0	4	319,8
05.01.2013 ...	Lance Y...	Carlota	0,015	3	68,84
05.01.2013 ...	Shari Sil...	Sunbell	0	2	50
05.01.2013 ...	Herrman ...	Carlota	0,01	1	22,94
05.01.2013 ...	Ruben ...	Quad	0	19	646
05.01.2013 ...	Beth Tu...	Sunbell	0	3	75
05.01.2013 ...	Marian ...	Doublers	0,025	2	159,88
05.01.2013 ...	Carls M...	Quad	0,01	1	33,00
Units Total: 33874					Summe Price: 915.687,52 €

Abbildung 33

### 3.7.3 Auswertung Jahr/Quartal/Monat/Woche per Slicer und Diagramm

Jetzt kann in Excel über übliche Pivot-Funktionen die folgende Auswertung erstellt werden:

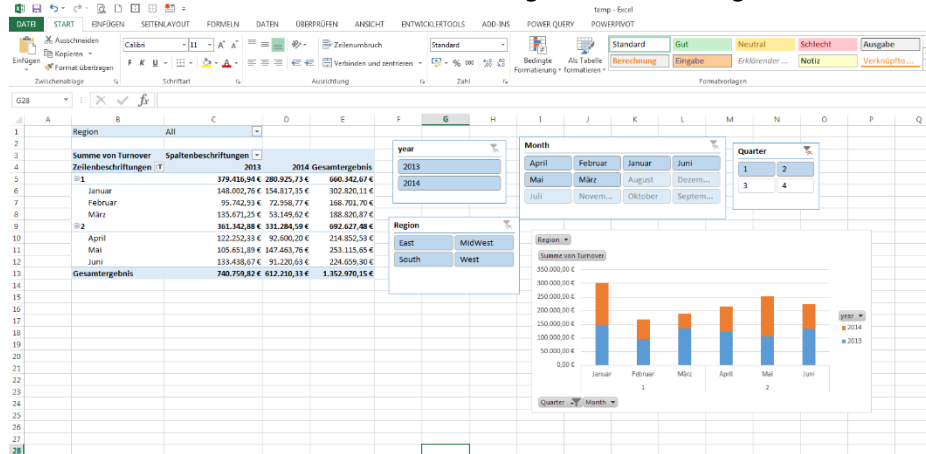


Abbildung 34

### 3.7.4 Aufgabe

- Fügen Sie alle drei Tabellen aus der Datei ins Datenmodell ein und verknüpfen sie entsprechend
- Erstellen Sie ein Masterdatumstabelle mit Monaten als Klartext, Quartals- und Wochennummern
- Verknüpfen Sie diese Tabelle in Ihrem Datenmodell
- Erstellen Sie einen Pivot-Table Report mit Pivot-Chart der es ermöglicht Jahre, Monate, Quartale und Regionen schnell und bequem über Datenschnitte zu selektieren

### 3.8 KPI

Datei: Artikel-Umsätze.xlsx

Ziel:

		Summe Umsatz	Summe Umsatz Ziel	Summe Umsatz Status
4	Althaus	25.170,76 €	22000	🟢
5	Aue	16.150,84 €	22000	🟡
6	Backhaus	4.945,22 €	22000	🔴
7	Blechinger	29.699,20 €	22000	🟢
8	Bruch	14.856,52 €	22000	🟢
9	Freitag	8.032,35 €	22000	🔴
10	Gaisler	22.278,56 €	22000	🟢
11	Häuser	24.329,07 €	22000	🟢
12	Heike	36.320,63 €	22000	🟢
13	Herrmann	15.035,03 €	22000	🟢
14	Lemke	13.864,87 €	22000	🟡
15	Ludemann	52.629,24 €	22000	🟢
16	Müller	10.072,54 €	22000	🟡
17	Nagel	38.042,49 €	22000	🟢
18	Rau	30.176,90 €	22000	🟢
19	Sander	42.787,68 €	22000	🟢
20	Uldall	18.518,36 €	22000	🟢
21	Wanka	23.812,38 €	22000	🟢
22	Weimar	7.332,24 €	22000	🔴
23	Wolf	32.477,25 €	22000	🟢
24	<b>Gesamtergebnis</b>	<b>447.348,55 €</b>	<b>22000</b>	🟢

KPI erstellen:

- Summe im Berechnungsbereich bilden
- Powerpivot/Home/KPI Erstellen
- Schwellwerte definieren

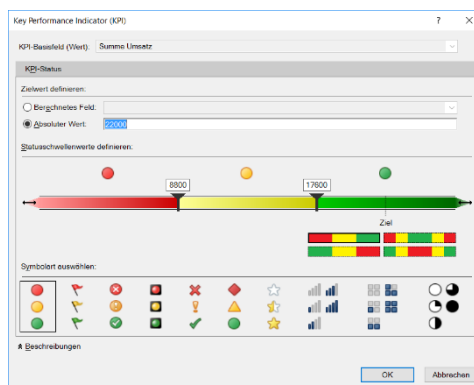


Abbildung 35

## 4 Power BI Desktop

### 4.1 Datenimporte

Ordner: MultiCSV-to-Dashboard/Start

In diesem Kapitel werden zunächst alle benötigten Daten importiert und modelliert.

#### 4.1.1 CSV-Multiimport

Um mehrere Dateien aus einem Ordner zu importieren, wählen Sie *Start/Daten abrufen/Mehr.../Ordner*

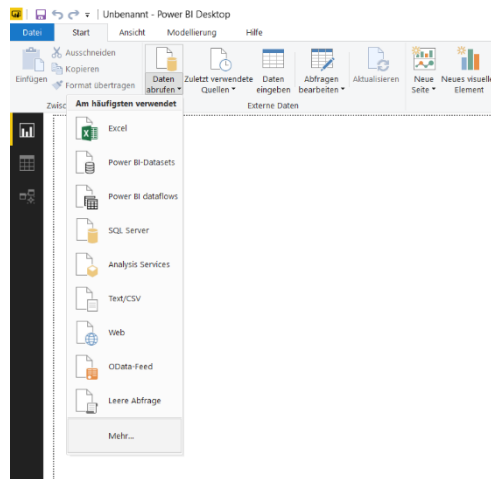


Abbildung 36

Wählen Sie den Ordner mit den zu importierenden Daten. Die Schaltfläche *Daten transformieren*

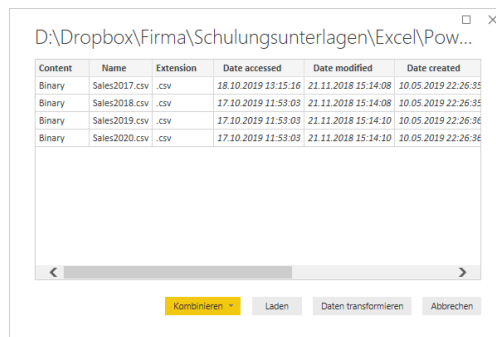


Abbildung 37

führt uns in den integrierten Powerquery-Editor. Nachträglich kann dies auch über *Abfragen bearbeiten* passieren:



Jetzt entfernen wir über das Kontextmenü alle anderen Spalten außer der ersten und fassen alle Datenquellen über die Schaltfläche *Daten kombinieren* zu einer Tabelle zusammen:

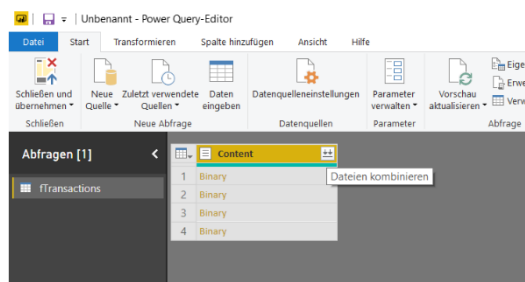


Abbildung 38

4.1.1.1 Werte transformieren

Jetzt empfiehlt es sich die Datumsspalte in reguläres Datum zu transformieren und die Währungszahlen via Rechtsklick *Typ ändern/mit Gebietschema* ins deutsche Zahlenformat zu konvertieren und auf zwei Stellen zu runden:

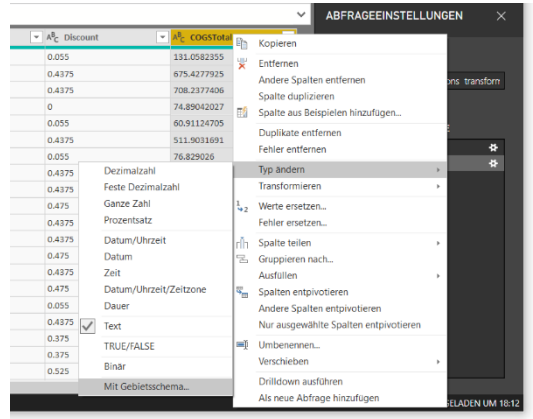


Abbildung 39

Über *Start/Schließen und übernehmen* werden alle Änderungen ins Power BI übernommen.

**Hinweis:** Falls die Umsatzzahlen falsch konvertiert wurden (Abbildung 40) ist der Schritt *Geänderter Typ* zu entfernen und anschließend das Feld über Rechtsklick *Typ ändern mit Gebietschema* in das korrekte Format zu überführen.

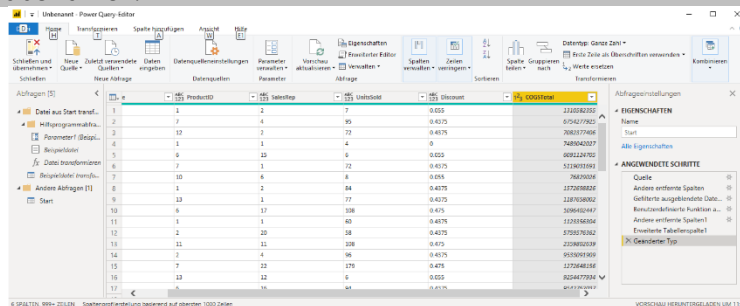


Abbildung 40

4.1.2 Zwischenbericht

Aus den bisherigen Daten kann z.B. ein solcher Bericht erstellt werden:

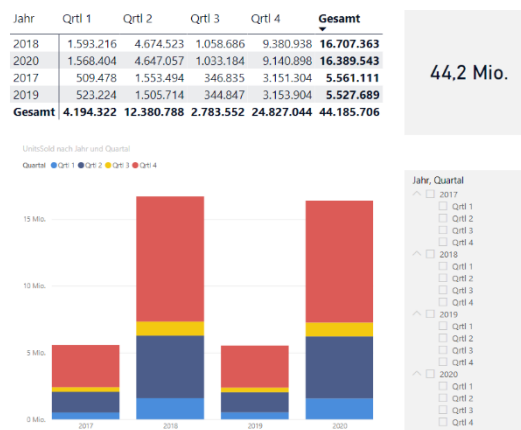


Abbildung 41

### 4.1.3 Berichte veröffentlichen

Legen Sie zunächst ggf. ein kostenloses Microsoft-Konto an. Wenn Sie Power BI starten, melden Sie sich mit Ihrem Konto an. Jetzt kann über Start/Veröffentlichen Ihr Projekt auf powerbi.com hochgeladen werden.

Loggen Sie sich anschließend auf powerbi.com ein und betrachten Ihr Projekt online. Darüber hinaus sind Interaktionen wie Filtern, Drilldown etc. innerhalb des Projektes möglich.

Über Freigeben kann ein Bericht an andere Power BI **Pro** Nutzer freigegeben werden. Diese können dann den Bericht entweder in Online oder in ihrer Power BI App auf einem Mobilgerät betrachten.

**Hinweis:** Über Anzeigen/Layout auf Mobilgeräten kann das Layout auf Mobilgeräte angepasst werden

### 4.1.4 Berichte auf SharePoint veröffentlichen

Dazu loggen Sie sich auf Ihrer Power BI Seite ein und klicken auf Freigeben:

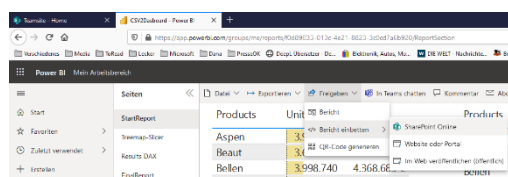


Abbildung 42

Über *Bericht einbetten/SharePoint Online* kopieren Sie den Einbettungslink in die Zwischenablage. In SharePoint:

- Seitenbearbeitung aktivieren
- Neuen Seitenbereich einfügen
- In Quicklinks nach Power BI suchen
- Über *Bericht hinzufügen* Link einfügen

### 4.1.5 Excel-Import mit Relationen

Ordner: Excel-CSV-Relation /016-MSPTDA-Excel.xlsx und fSales-2017.csv

Zunächst müssen alle vier Excel-Datentabellen und das CSV über *Start/Daten abrufen* importiert werden und können direkt über die Schaltfläche *Bearbeiten* in Powerquery bearbeitet werden.

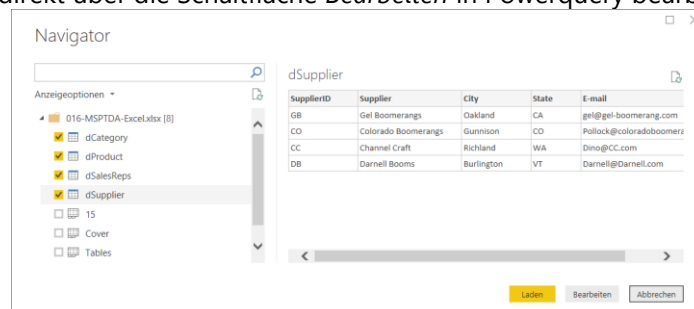


Abbildung 43

Jetzt können zusammenhängende Tabellen verknüpft werden. Dazu wählen Sie in Powerquery den Befehl *Start/Kombinieren/Abfragen zusammenführen* und selektieren die zu verknüpfenden Tabellen *dProduct* und *dCategory*. Anschließend werden die Verknüpfungsfelder – hier *CategoryID* – einfach angeklickt:

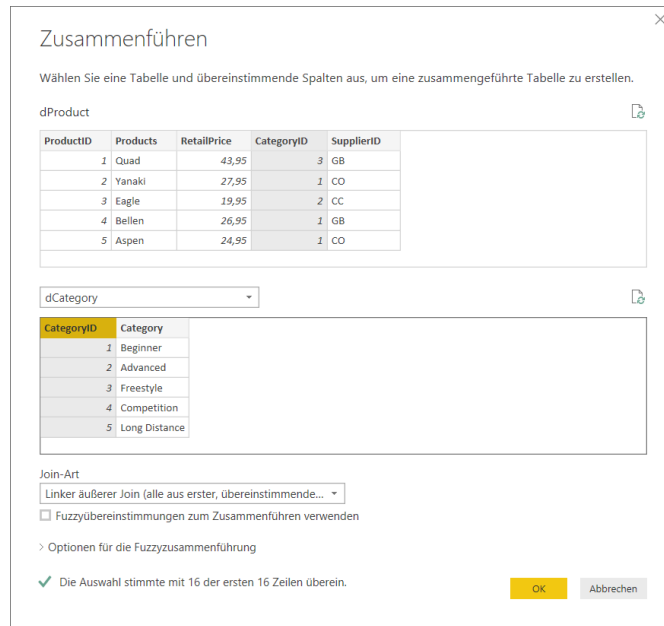



Abbildung 44

Das zusammengeführte Ergebnis kann jetzt über dieses Symbol  expandiert werden und wir behalten lediglich die Spalten, die wir benötigen, hier *Category*:

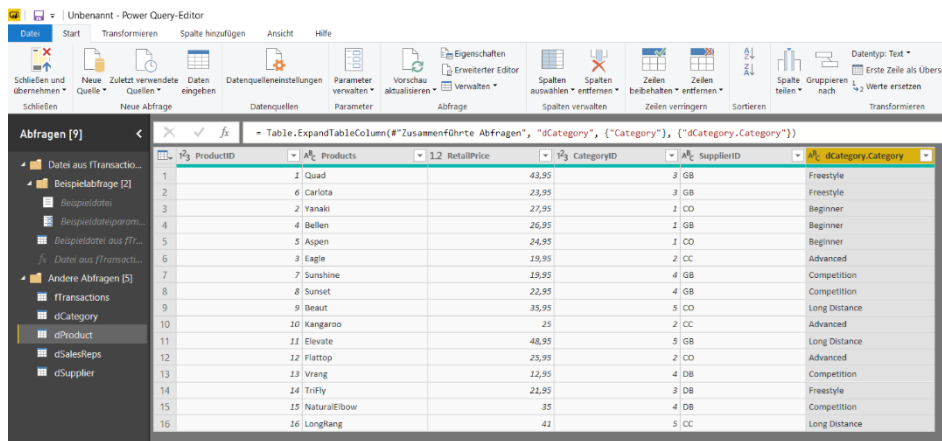


Abbildung 45

Über *Start/Schließen und übernehmen* werden alle Änderungen ins Power BI übernommen. Über die Schaltfläche *Modell* auf der linken Seite können Tabellenverknüpfungen geprüft und bearbeitet werden. Power BI hat für uns automatisch die Verknüpfungen angelegt:

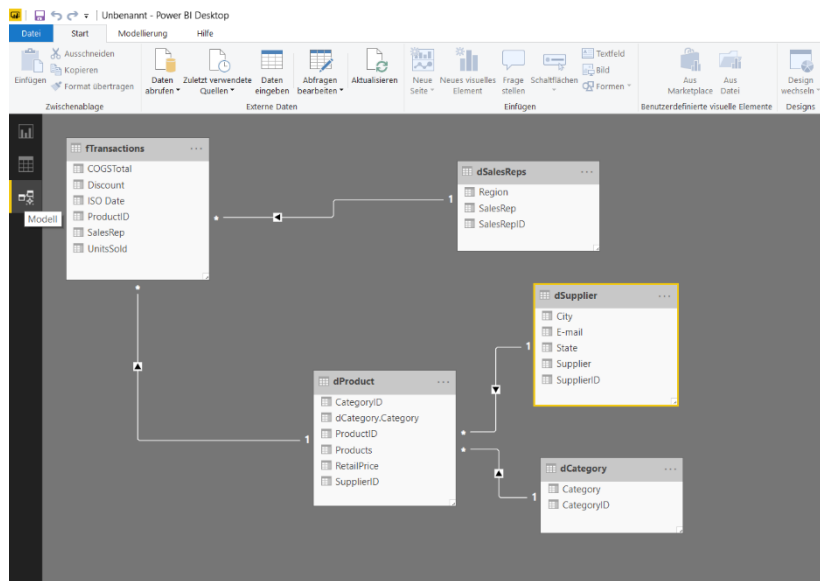


Abbildung 46

**Hinweis:** Falls die Verknüpfungen nicht automatisch angelegt wurden, muss dies manuell nachgeholt werden, indem man die Tabellen auf die Arbeitsfläche zieht und die Verbindungen per Drag&Drop anlegt.

#### 4.1.6 Sharepoint Daten importieren

Zunächst müssen die gewünschten Sharepoint Daten abgerufen werden:

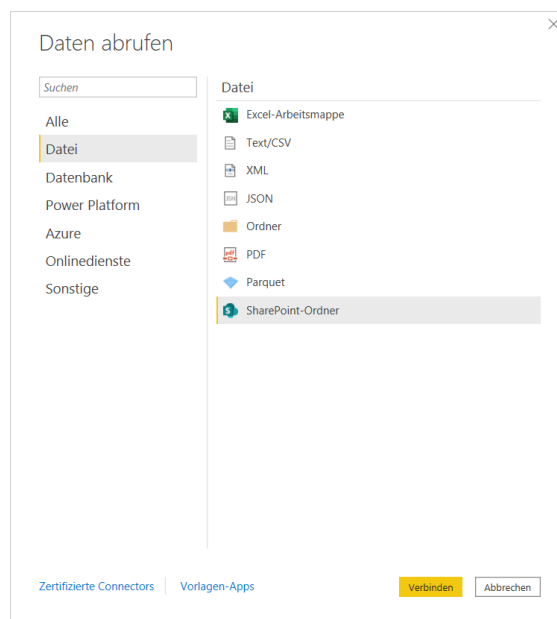


Abbildung 47

Sharepoint Webseitendresse aus dem Browser kopieren und einfügen, z.B.:

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

Danach *Daten transformieren* und in Power Query gewünschte Datei(en) ausfiltern und auf Basis dieser Datei(en) einen Report mit Visuals anlegen.

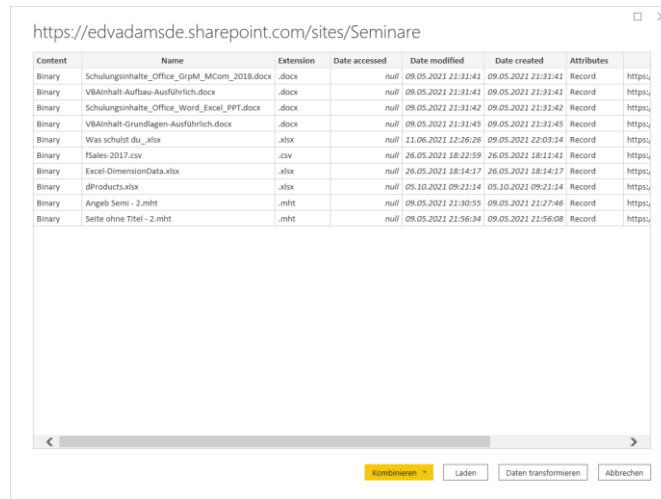


Abbildung 48

Nachdem der gewünschte Report erstellt wurde, kann dieser ganz normal über *Start/Veröffentlichen* auf den Power BI Dienst hochgeladen werden.

### 4.2 Einfachen Bericht erstellen

Als nächstes soll der nachfolgende Bericht erstellt werden:

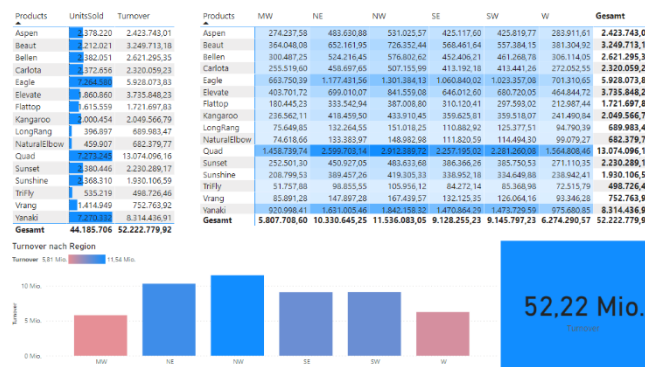


Abbildung 49

Die Besonderheiten im Bericht:

- Drilldown im Diagramm
- Bedingte Formatierungen in allen Elementen (Auch Zahlenkarte mit 52 Mio)
- Variierbare Interaktionen mit anderen Visuals

#### 4.2.1.1 Drilldown im Diagramm

Um z.B. im gestapelten Diagramm keine Farbsegmente anzuzeigen, sondern per Klick ein Drilldown zu weiteren Details zu ermöglichen müssen die Detailinformation auf *Achse* gezogen werden:

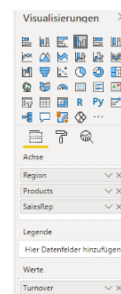


Abbildung 50

#### 4.2.2 Measures erstellen

Um den Umsatz in der linken Tabelle des Berichts erstellen zu können, benötigen wir ein Measure welches diesen berechnet:

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

Der Ausdruck innerhalb von SumX() berechnet die jeweiligen Einzelumsätze:

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

Da sich der Produktpreis nicht in der Tabelle *fTrans* befindet muss diese über RELATED() aus der Tabelle *dProduct* geholt werden.

### 4.2.3 Visuals einfügen

#### Table

Als erstes fügen wir das Visual *Table* ein:

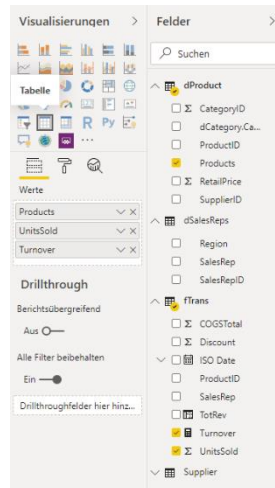


Abbildung 51

Danach können die gewünschten Felder in den Wertebereich gezogen werden. (*Products, UnitsSold, Turnover (Measure)*)

Über den Chevron bei *UnitsSold* können Datenbalken als bedingte Formatierung eingefügt werden.

#### Matrix

Dies entspricht einer *Pivot-Tabelle* zur Abbildung mehrdimensionaler Daten. Wir fügen das Visual ein und ziehen die gewünschten Felder (Produkt, Region, Umsatz) in die entsprechenden Bereiche:

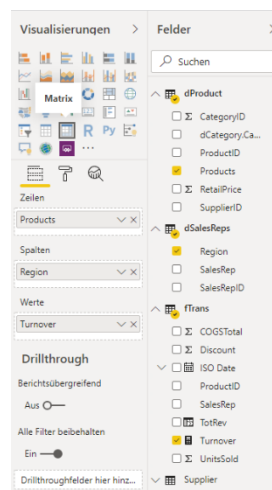


Abbildung 52

Über das Chevron *Bedingte Formatierung* können wir für den Umsatz (Turnover) die Hintergrundfarbe aktivieren um die Tabelle mit einer Heatmap zu versehen.

#### Diagramm

Zum Schluss fügen wir eine *gestapeltes Säulendiagramm* ein und ziehen die gewünschten Felder in ihre Bereiche:

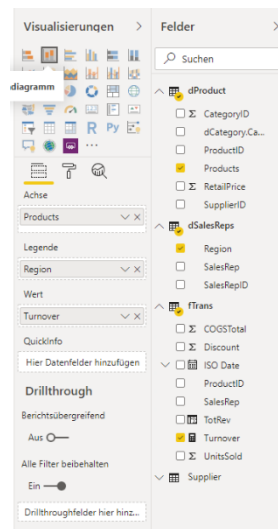


Abbildung 53

**TIPP:** Um die Reihenfolge der Sortierung auf der x-Achse anzupassen, wählen Sie das 3-Punkte Menü oben rechts im Diagramm und dann *sortieren nach*

#### 4.2.4 Visuals filtern

Um innerhalb des jeweiligen Visuals zu filtern, klicken Sie es zunächst an und wählen auf der rechten Bildschirmseite die entsprechende Filterung aus:

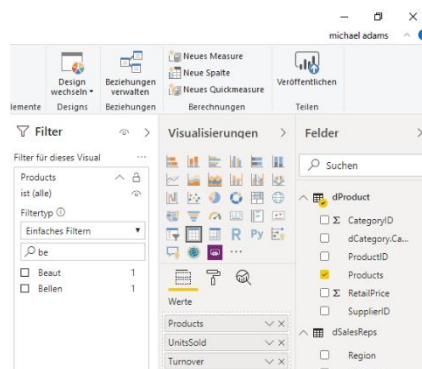


Abbildung 54

### 4.3 Tabellen per Funktion erstellen

**Ordner/Datei:** Ergebnis von Kap. 4.1

Neue Datentabellen können über *Modellierung/Neue Tabelle* per Funktion erstellt werden:

Datumstabelle = CALENDAR("1.1.18";"31.12.18") – Erstellt eine Tabelle mit Start- und Enddatum

**Tipp:** Das Start- und Enddatum können auch auf Basis einer existierenden Datumsspalte erstellt werden:

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

#### 4.3.1 Neue Spalten einfügen

Über *Modellierung/Neue Spalte* können weitere Berechnungsspalten angefügt werden, z.B.:

- MonatsNummer = MONTH(Datum[Datum])
- MonatText = FORMAT('dDatum'[Datum];"MMM")
- Jahr = YEAR('dDatum'[Datum])
- Quartal = ROUNDUP('dDatum'[MonatsNummer]/3;0)

4.3.1.1 Region als Gruppierungsspalte

Regionen sollen reduziert werden auf: N/W/S/E:  
 Region Simple = LEFT(dSalesReps[Region];1)

4.3.1.2 Fiskaljahr berücksichtigen

Angenommen das Fiskaljahr Ihres Unternehmens startet am 1.4. Damit müssen die Quartalsnummern verschoben werden. In diesem Beispiel wären die Monate Jan-Mrz das vierte Quartal womit im April das erste Quartal starten würde:

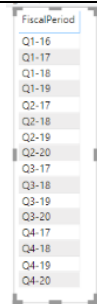
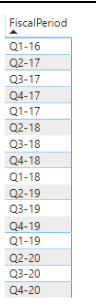
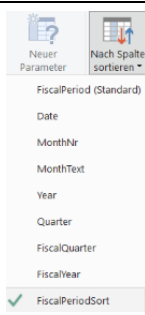
FiskalQuartal = IF(dDatum[Quartal]=1;4;dDatum[Quartal]-1)

Das Fiskaljahr ermitteln wir ähnlich:

FiskalJahr = IF(dDatum[Quartal]=1;dDatum[Jahr]-1;dDatum[Jahr])

Fiskalperiode (Q1-17, Q1-18 usw...)

- Erstellen: FiscalPeriod = "Q"&dDate[FiscalQuarter]&"-"&RIGHT(dDate[FiscalYear];2)
- Hilfspalte zum Sortieren: FiscalPeriodSort = dDate[FiscalYear]\*10+dDate[FiscalQuarter]

 <p>Hier erfolgt die Sortierung nach Quartalen</p>	 <p>Um die Sortierung für FiscalPeriod nach der Sortierhilfspalte zu aktivieren, klicken Sie in der Tabelle auf die Spalte FiscalPeriod und wählen <i>Nach Spalte Sortieren</i> und hier die Sortierhilfspalte (Indirektes Sortieren)</p> 
--	--

4.4 Arten von Tabellenbeziehungen

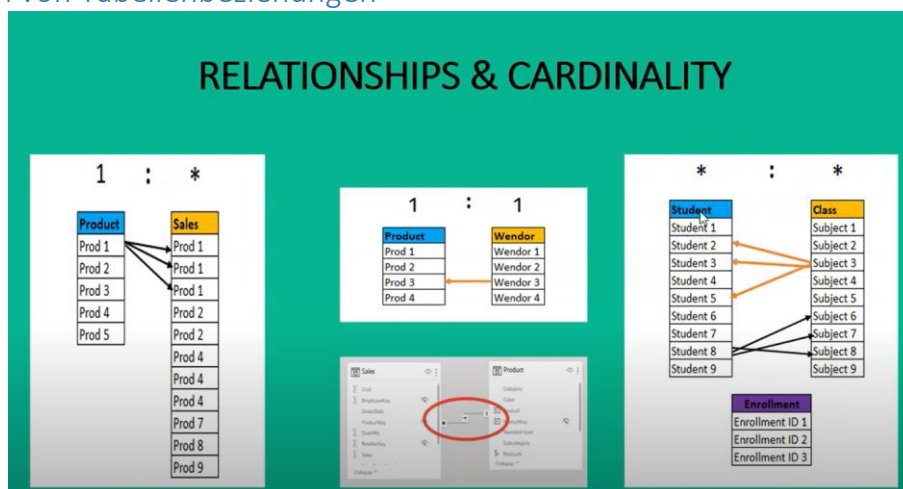


Abbildung 55

4.5 Master DateTable

Damit die Datumsanalysefunktionen korrekt arbeiten, sollte die Datumstabelle in Power BI über *Modellierung/Als Datumstabelle markieren* gekennzeichnet werden.

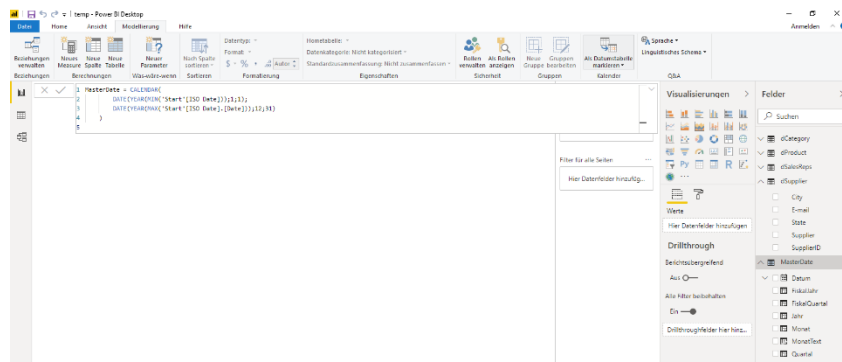


Abbildung 56

Anschließend sollte die Datumstabelle im Beziehungsbereich mit der Facts-Tabelle (fTransaktionen) verknüpft werden:

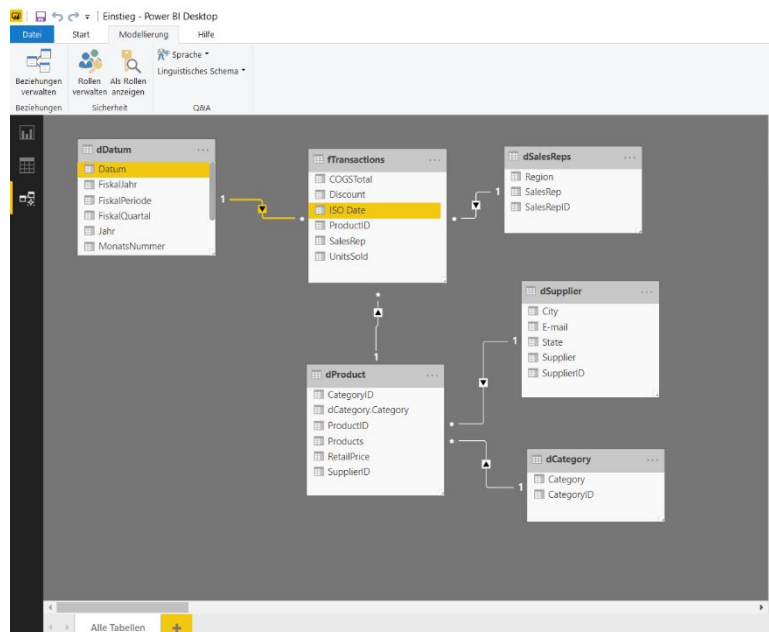


Abbildung 57

Zum Schluss empfiehlt es sich, nicht benötigte Spalten über Rechtsklick aus der Berichtsansicht zu löschen:

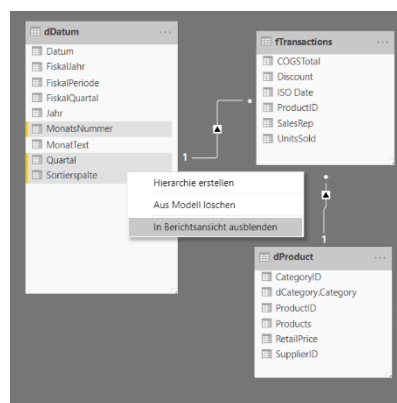


Abbildung 58

### 4.6 Zwischenbericht mit Treemap & Slicer

In diesem Bericht zeigt uns die Treemap die Verteilung der Umsätze nach Region und Verkäufer. Der Slicer (Datenschnitt) der das Fiskaljahr beinhaltet, filtert alle Visuals im Bericht.

Ferner benötigen wir folgende Measures:

- PrevPeriodTurnover = CALCULATE([Turnover];SAMEPERIODLASTYEAR(dDate[Datum]))
- Change% = DIVIDE([PrevPeriodTurnover];[Turnover];0)
- 

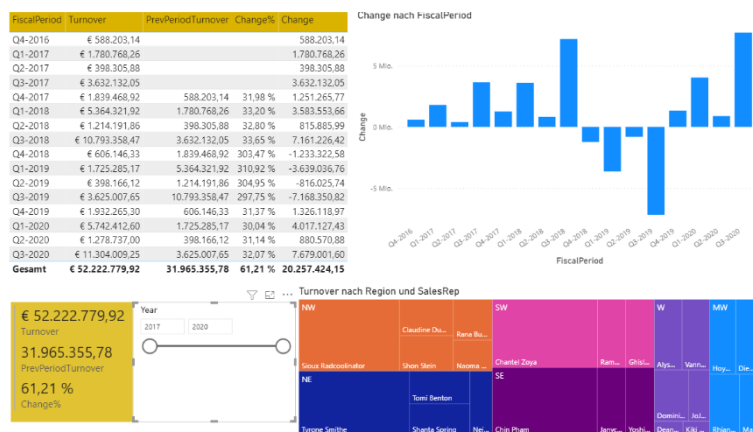


Abbildung 59

#### 4.6.1 Filter für alle Seiten

Wenn Sie in diesen Bereich Felder zum Filtern hineinziehen, wirkt dieser Filter auf alle Seiten Ihrer Datei:

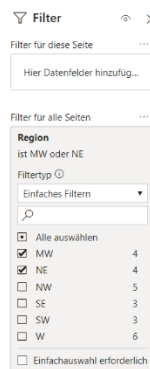


Abbildung 60

### 4.7 DAX-Measures (Aggregate) erstellen

Über Modellierung/Neues Measure können Explicite Measures erstellt werden, z.B.:

- Total Units:  
TotalUnits = SUM(fTrans[UnitsSold])
- Total COGS:  
TotalCOGS = SUM(fTrans[COGSTotal])

- Total Revenue:  
 $TotalRevenue = \text{SUMX}(fTrans; \text{RELATED}(dProduct[RetailPrice]) * fTrans[UnitsSold] * (1 - fTrans[Discount]))$
- Total Gross Profit  
 $TotalGrossProfit = [TotalRevenue] - [TotalCOGS]$
- % Gross Profit  
 $\%GrossProfit = \text{DIVIDE}([TotalGrossProfit]; [TotalRevenue])$
- Durchschnittlicher Tagesumsatz:  
 $DailyAvgRevenue = \text{AVERAGEX}(fTrans; fTrans[UnitsSold] * (1 - fTrans[Discount]) * \text{RELATED}(dProduct[RetailPrice]))$

Diese Measures können wir per Checkbox aktivieren und erhalten:

Products	TotalUnits	TotalCOGS	TotalRevenue	TotalGrossProfit	%GrossProfit
Aspen	5396613	€ 49.634.131,61	73.178.453,48	23.544.321,87	32,17 %
Beaut	4958560	€ 66.465.064,10	96.900.202,32	30.434.338,22	31,41 %
Bellen	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 %
Eievate	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 %
Vrang	3187688	€ 36.728.011,49	22.557.157,12	-14.170.854,37	-62,82 %
Vanaki	16433913	€ 163.193.733,18	250.117.297,39	86.923.544,21	34,75 %
<b>Gesamt</b>	<b>99568214</b>	<b>€ 1.173.480.142,75</b>	<b>1.564.328.491,21</b>	<b>390.848.348,47</b>	<b>24,99 %</b>

Abbildung 61

**Hinweis zum Filterkontext:** Hier werden die Ergebnisse **je Artikel** gebildet, d.h. es wird der jeweilige Wert gebildet für Aspen, Beaut usw... Der zeilenweise Filterkontext wird also der Products-Spalte entnommen.

## 4.8 Dashboards erstellen

Um Berichte zu erstellen, gehen Sie grundsätzlich in vier Schritten vor:

- Visualisierung auswählen
- Measures (siehe Kap. 4.7) per Checkbox aktivieren
- Felder (Tabellenspalten) auswählen
- Formatierung einstellen

### 4.8.1 Matrix (Pivot Tabelle)

Über das Visual Matrix kann auf dem Bericht eine Pivot-Tabelle eingefügt werden. Anschließend ziehen Sie die gewünschten Felder in die Zeile/Spalte/Werte Container.

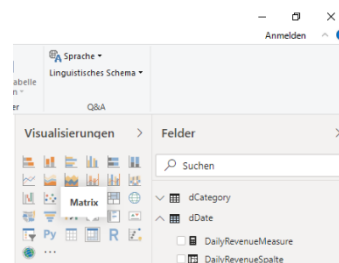


Abbildung 62

Über den Dropdown-Pfeil des Wertefeldes können auch die prozentualen Anteile angezeigt werden.

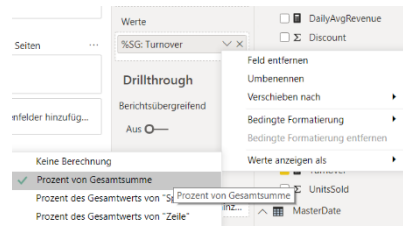


Abbildung 63

Anschließend kann die Visualisierung formatiert werden. Wechseln Sie dazu in den *Format* Bereich (Farbrolle), wechseln in die Gruppe *Raster* und ändern die Schriftgröße.

In der Gruppe *Bedingte Formatierung* können anschließend die *Datenbalken* aktiviert werden.

Ergebnis:

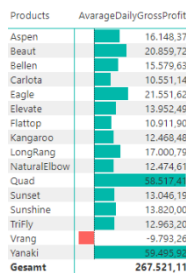


Abbildung 64

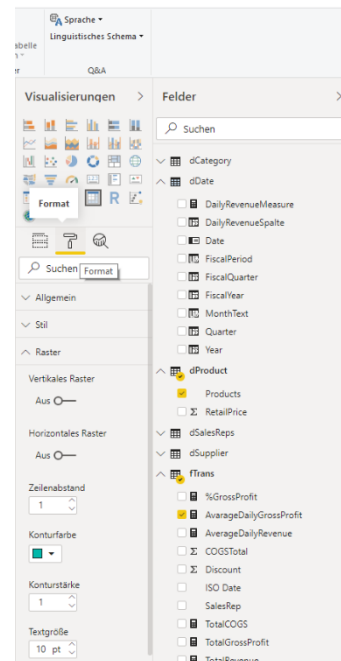
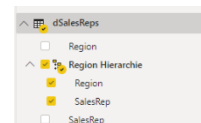


Abbildung 65

#### 4.8.2 Säulendiagramm mit Drill-Down

##### Hierarchie erstellen

Um später im Diagramm Detaildaten (Drill-Down) abrufen zu können erstellen Sie zunächst eine Hierarchie aus *Region* und *SalesRep*. Ziehen Sie dazu in der Feldliste die *SalesRep* auf die *Region* um die Hierarchie zu erstellen.



**Hinweis:** Falls die Monate (Text) falsch sortiert werden, muss der Monat indirekt nach Monatsnummer sortiert werden.

##### Drill-Down

Erstellen Sie jetzt ein Säulendiagramm und Ziehen die Hierarchie in den *Achse* Container. Jetzt können Sie über den Drill-Down Pfeil und Klick auf eine Säule Detaildaten abrufen. Der Doppelpfeil nach Unten zeigt alle Daten der nächsten Ebene.

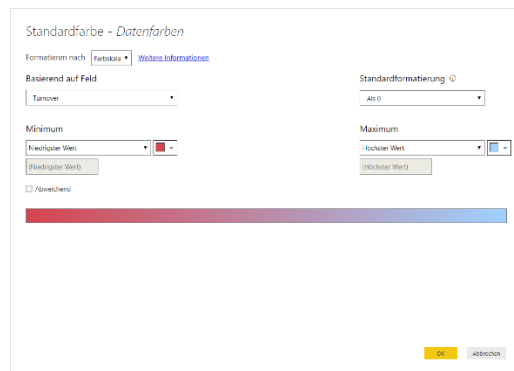


### Bedingte Formatierung

Format/Datenfarben/Button  $f_x$ :



- Basierend auf Feld: Turnover
- Farben für Min/Max Werte wählen



Hinweis: Die Bedingte Formatierung kann genauso auch auf Tabellen angewendet werden

### 4.8.3 Liniendiagramm

Über das Visual *Liniendiagramm* fügen Sie ein Liniendiagramm ein. Auf der x-Achse bilden wir die Fiskalquartale ab (Feld: *FiscalPeriod*) auf die y-Achse legen wir *AverageDailyGrossProfit*.

#### Sortierung der x-Achse anpassen

Wählen Sie im Dreipunkte Menü des Diagramms zunächst die Sortierung und Sortierreihenfolge nach dem gewünschten Feld:

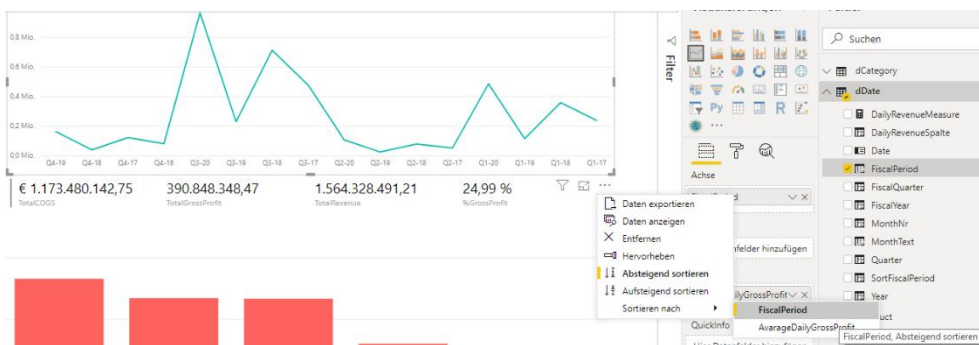


Abbildung 66

#### Sortierproblem:

Da die Sortierung jedoch alphabetisch erfolgt, benötigen wir jetzt die im Kap. 4.3.1.2 erstellte Sortier-Hilfsspalte in der die gewünschte Reihenfolge definiert ist. Um diese Sortierreihenfolge zu wählen markieren Sie rechts das Feld *FiscalPeriod* und wählen aus *Nach Spalte sortieren* die Sortier-Hilfsspalte (*SortFiscalPeriod*):

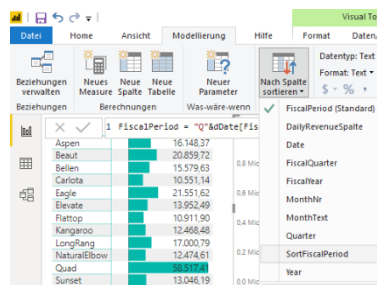


Abbildung 67

### 4.8.4 Karte/Mehrzeilige Zuordnung

Dieses Element dient der Anzeige von Measures, wobei *Karte* nur einen Wert anzeigt, *Mehrzeilige Zuordnung* mehrere Werte:



Abbildung 68

**TIPP:** Über *Home/Textfeld* kann ein Textfeld eingefügt werden.

#### 4.8.5 Slicer (Datenschnitt)

Aus der Auswahl der Visuals können mehrere Slicer zugefügt werden, um Daten anschließend komfortabel filtern zu können. Fügen Sie dazu einen Slicer ein und wählen Anschließend das gewünschte Feld. So verfahren Sie mit allen weiteren Slicern.

##### 4.8.5.1 Bookmarks

Die jeweilige Filtereinstellung kann als Lesezeichen gespeichert werden.

Aktivieren Sie dazu über *Ansicht* den *Lesezeichenbereich*. Filtern Sie wie gewünscht und fügen über *Hinzufügen* ein neues Lesezeichen ein.

#### 4.8.6 Neue Visuals importieren

Um z.B. einen mehrspaltigen Slicer (Chicklet-Slicer) zu importieren, wählen Sie in Visuals das Dreipunkte-Menü:



Abbildung 69

Loggen Sie sich ein und wählen z.B. den (Chicklet-Slicer). Nach der Installation steht Ihnen das neue Element innerhalb der aktuellen Datei zur Verfügung.

**Tipp:** Falls das Visual permanent benötigt wird, starten Sie mit einer leeren Datei, importieren das oder die Visuals und speichern die Datei als Vorlage (\*.pbit). Starten Sie Power BI beim nächsten mal durch Doppelklick auf die Vorlage.

### 4.9 KPI/Bedingte Formatierung

Rechtsklicken Sie dazu in der Feldauswahl auf das gewünschte Feld und wählen *Bedingte Formating/Symbole*:

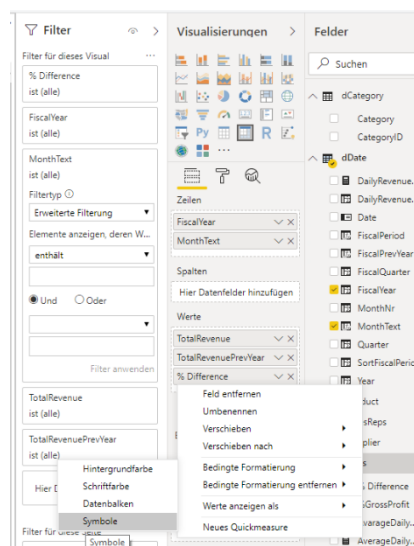


Abbildung 70

Wählen Sie im nachfolgenden Dialog die gewünschten Schwellwerte und Symbole:

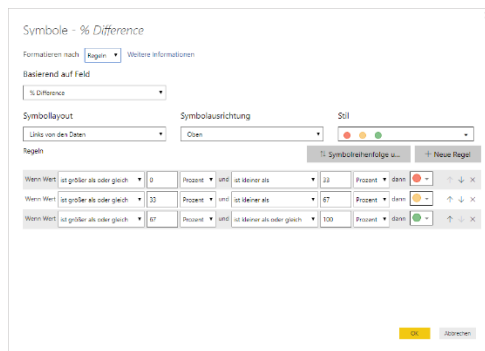


Abbildung 71

**Fertiger Bericht:**

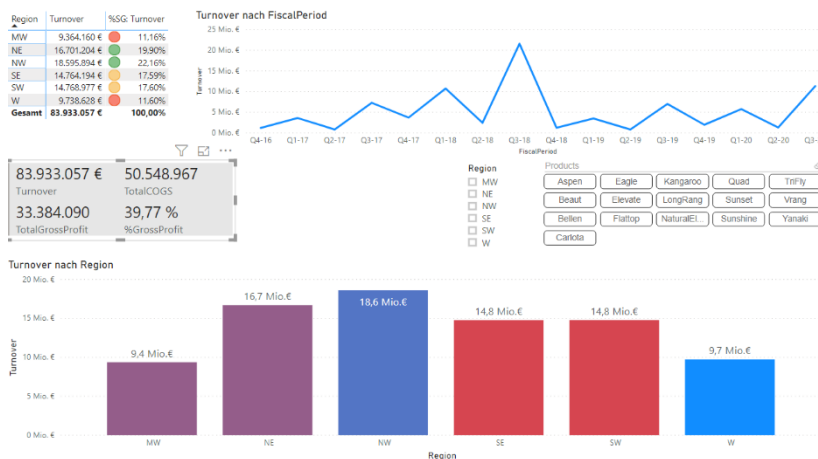


Abbildung 72

**4.10 Berichte veröffentlichen**

Legen Sie zunächst ggf. ein kostenloses Microsoft-Konto an. Wenn Sie Power BI starten, melden Sie sich mit Ihrem Konto an. Jetzt kann über Home/Veröffentlichen Ihr Projekt auf powerbi.com hochgeladen werden. Loggen Sie sich anschließend auf powerbi.com ein und betrachten Ihr Projekt online. Darüber hinaus sind weitere Bearbeitungen innerhalb des Projektes möglich. So kann jeder Benutzer seine individuellen Visualisierungen auf Basis der vorbereiteten Daten erstellen.

**4.11 Vorjahresperioden vergleichen**

In der Master-Datumstabelle kann das Datum des Vorjahres wie folgt ermittelt werden:  
 LastYear = SAMEPERIODLASTYEAR(MasterDate[Date])

**4.11.1 Calculate**

Die Funktion Calculate() ermittelt einen Berechnungsausdruck, im individuellen Filterkontext:  
 =CALCULATE(Ausdruck;Filter1;Filter2;...)

**Beispiel 1**

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

**Beispiel 2**

```

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

```

SUMX() bildet hier den Berechnungsausdruck

SAMEPERIODLASTYEAR() bildet den Filterkontext

Als nächstes bilden wir die Differenz zwischen Umsatz und Vorjahresumsatz:

PrevYearDiff = [Turnover]-[TurnoverLastYear]

Und werten diese, nach Monaten aufgeschlüsselt in einer Pivot-Tabelle (Matrix) aus:

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

Abbildung 73

Um Drilldown im Diagramm durchführen zu können bilden wir eine Hierarchie aus *Year* und *Month*:

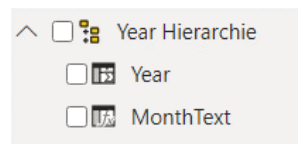


Abbildung 74

Anschließend erstellen wir ein Säulendiagramm:

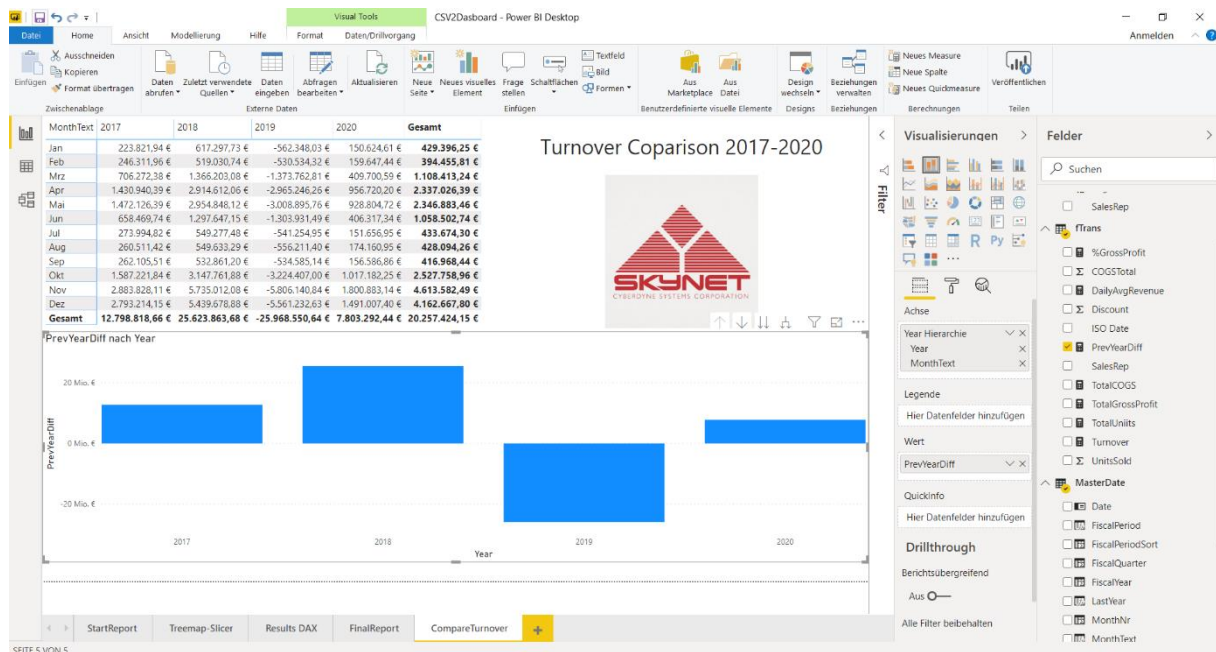


Abbildung 75

Im fertigen Report können über das Register *Textfeld* und *Bild* Beschriftungen und Grafiken eingefügt werden:

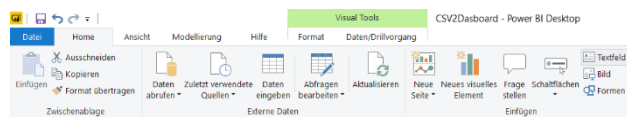


Abbildung 76

## 5 Power BI Desktop Advanced

### 5.1 Filterkonzepte und Filterpropagation

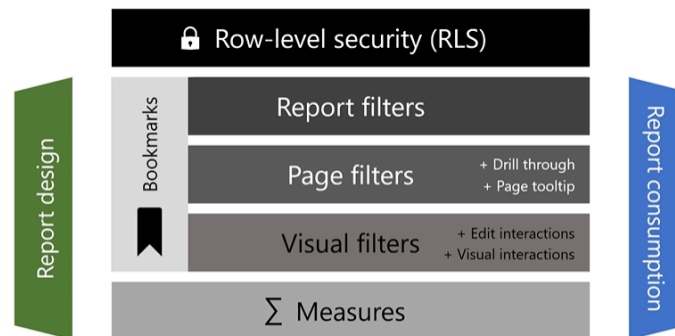


Abbildung 77

Details: [Einführung in das Entwerfen von Berichten zum Filtern - Training | Microsoft Learn](#)

TIPP: Filterbereich deaktivieren – Auge-Symbol:

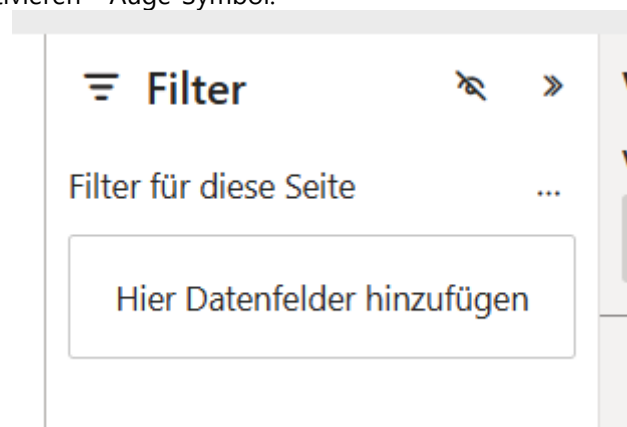


Abbildung 78

Ferner können die Visual-Icons (Header)...

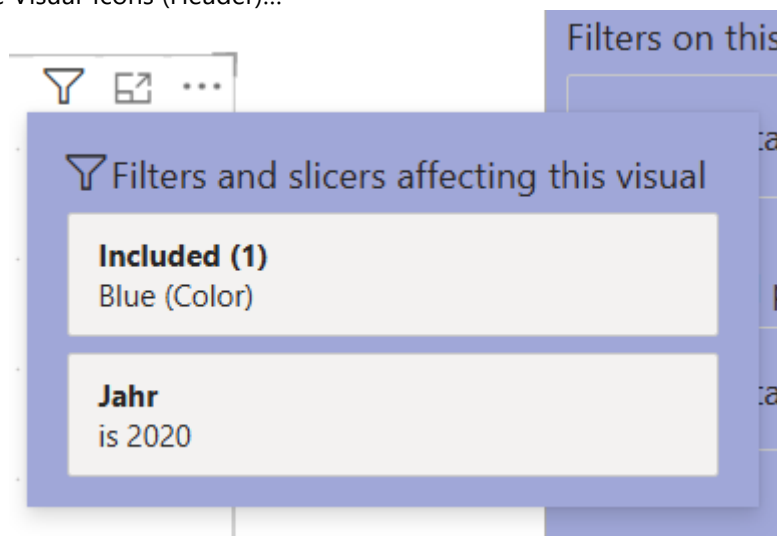


Abbildung 79

...entweder in den Visual-Einstellungen einzeln oder in den Optionen für den gesamten Bericht deaktiviert werden.

## 5.2 DAX

### 5.2.1 Rabatt-Prozentklassen zusammenfassen

**Ordner:** MultiCSVtoDashboard

**Datatable per Hand erzeugen:**

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

Besser: Datatable dynamisch erzeugen über GENERATESERIES():

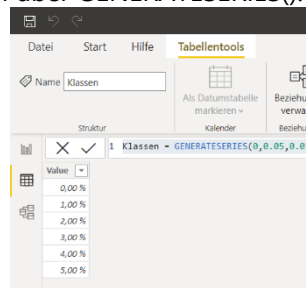


Abbildung 80

In fSales Prozentklasse berechnen:

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

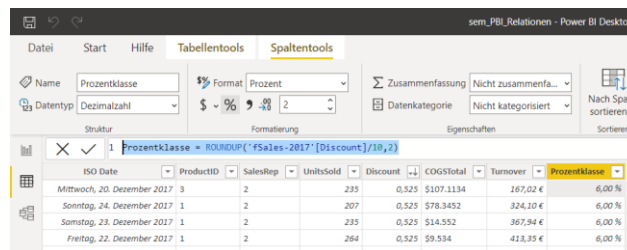


Abbildung 81

Dann fSales/Prozentklassen mit der Datentabelle aus **Fehler! Verweisquelle konnte nicht gefunden werden.** verknüpfen.

#### 5.2.1.1 Leere Werte in Visuals als Null anzeigen

Zunächst muss die Anzeige leerer Zellen aktiviert werden:

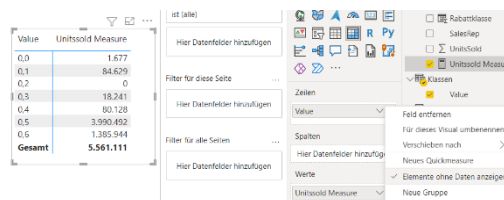


Abbildung 82

Um auch die *Null* anzuzeigen, legt man eine neue Measure an und addiert am Ende eine Null:

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

### 5.2.2 Calculate

Ordner: Story/2\_FiscalPeriods

Calculate ändert den Filterkontext eines Visuals und setzt für eine Kalkulation einen eigenen Filter ein, z.B.:

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

- Turnover 2020 = `CALCULATE([Turnover],YEAR(fSales[OrderDateKey])=2020)`

### 5.3 Inaktive Beziehungen (USERELATIONSHIP)

Ordner: Orders and Payments

Der Umsatz soll nach Bezahldatum ausgewertet werden:

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
<b>Gesamt</b>	<b>4.228.341,01</b>	<b>2.229.695,51</b>

Abbildung 83

Dazu wird eine zusätzliche Beziehung aus der MasterDate Tabelle zum Bezahldatum angelegt:

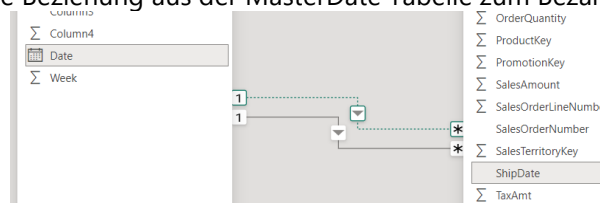


Abbildung 84

Jetzt kann USERELATIONSHIP() als Filterkriterium für CALCULATE() die Umsätze des Bezahldatums liefern:

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

### 5.4 Filterkontext

Ordner: ContextTransition/FilterContext.pbix (DB: Product\_Turnover\_ContextTransition)

Zeigt eine berechnete Spalte immer dasselbe Ergebnis, ist das ein Zeichen für fehlenden Filterkontext.

Soll in der Tabelle *Customer* in einer *berechneten Spalte* das letzte Bestelldatum ermittelt werden, funktioniert dies nicht, da der Filterkontext fehlt:

```
MaxDateWrong = MAXX(Sales, Sales[OrderDate])
```

Soll also für jeden Kunden das letzte Bestelldatum ermittelt werden so muss über RELATEDTABLE() die Beziehung von Kunde zur Umsatztable hergestellt werden.

Neue Spalte in Tabelle Kunden:

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

Anzahl Umsätze des Kunden ermitteln:

```
SalesCount = COUNTROWS(RELATEDTABLE(Sales))
```

Sehr gutes Video dazu: [\(77\) Understanding Row Context in DAX using Power BI - YouTube](#)

### 5.4.1 Kontextübergang

Ordner: Ordner: ContextTransition/FilterContext.pbix

Der Kontext, also z.B. welche Daten zu welchem Kunden gehören, kann in **einer Spalte** über CALCULATE hergestellt werden:

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

**Ein Visual** stellt bei verknüpften Tabellen, automatisch den Filterkontext her, wobei die Art der Aggregation (Min, Max, etc.) über in den Feldeinstellungen vorgenommen wird:

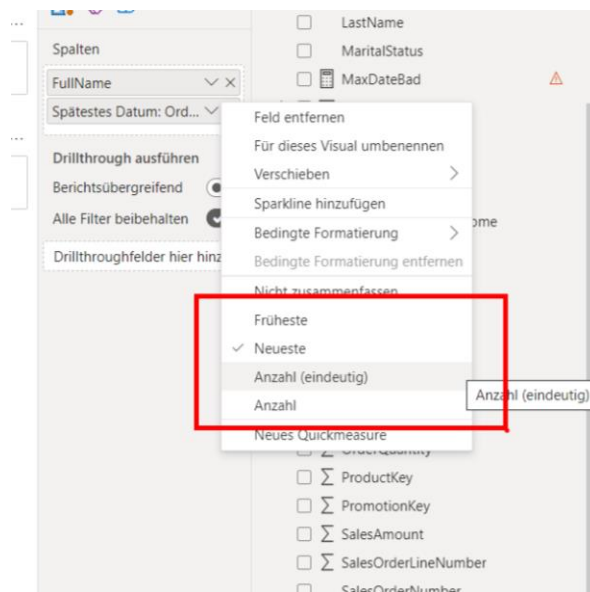


Abbildung 85

Wird in der Detailtabelle eine Summe als **zusätzliche Spalte** gebildet, so wird z.B. der Gesamtumsatz gebildet und wiederholt:

Umsatzspalte Ohne Kontext (falsch) =

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

Ein **Measure** benötigt kein Calculate(), da im Hintergrund jedes Measure in ein Calculate() verpackt wird. `[MaxDateMEASURE = MAXX(Sales,Sales[OrderDate])]`

HöchstesDatum mit Kontext (richtig)=

`[MaxDateMEASURE]`

## 5.5 Parameter

### 5.5.1 Dynamische Dimensionen (Feldparameter)

In Visuals können umschaltbare Dimensionen verwendet werden, um z.B. auf einer Diagrammachse zwischen Monat, Produkt oder Region umzuschalten. Dies geht ganz einfach über die Feldparameter:

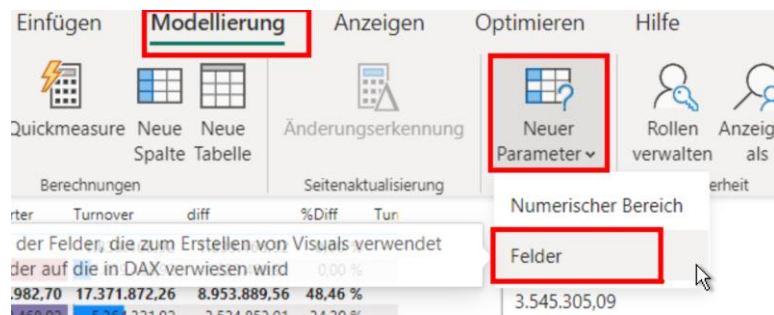


Abbildung 86

### 5.5.2 Zahlenparameter (Numerischer Bereich)

Ermöglicht die Auswahl eines Wertes/Faktors der in einem Visual einfach per Klick gewählt werden kann. Danach kann dieser Wert für beliebige Berechnungen in Measures verwendet werden.  
Hinweis: Die verwendete Funktion `SELECTEDVALUE()` liefert den aktuell geklickten Wert zurück.

## 5.6 Verwaiste Datensätze zählen/berechnen

Berechnungen für verwaiste Datensätze in einer Detailtabelle durchführen:

Measure, das die Anzahl der verwaisten Datensätze in der Umsatzliste ermittelt:

```
Count Rows Non Exist =
//Zählt nicht existente (verwaiste) Artikel/Datenzeilen in fSales
CALCULATE (
    COUNTROWS(fSales),
    ISBLANK(dProduct[ProductID])
)
```

Measure, das die Summe der Stückzahlen der verwaisten Datensätze in der Umsatzliste ermittelt:

```
Summe Units Sold Non Exist =
//Summiert UnitsSold für nicht existente (verwaiste) Artikel in fSales
CALCULATE (
    SUMX(fSales, fSales[UnitsSold]),
    ISBLANK(dProduct[ProductID])
)
```

## 5.7 KI Analysen

### 5.7.1 Veränderungen im Diagramm

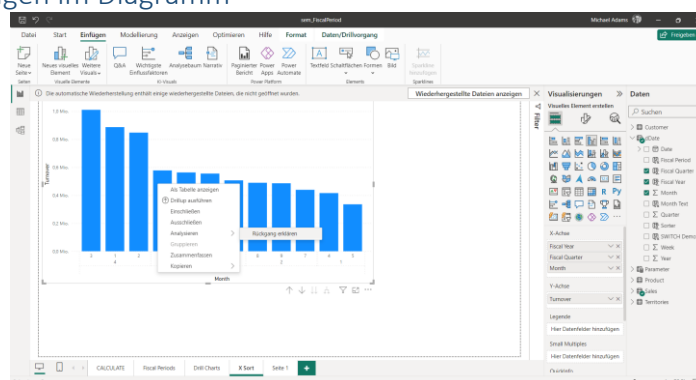


Abbildung 87

## 5.8 Vertiefung DAX

### 5.8.1 Vorperioden

#### 5.8.1.1 Fiskalperioden vergleichen

Ordner: Story/2\_Fiscal Periods

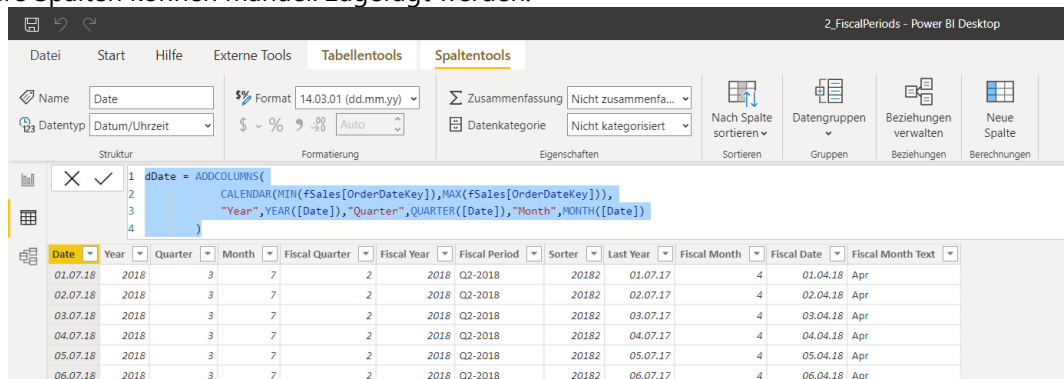
Dafür wird eine Masterdate-Table benötigt, die per DAX schnell angelegt werden kann:

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

Oder Dynamisch mit MIN-MAX Date:

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

Weitere Spalten können manuell zugefügt werden:



The screenshot shows the Power BI Desktop interface. The top ribbon includes 'Datei', 'Start', 'Hilfe', 'Externe Tools', 'Tabellentools', and 'Spaltentools'. The 'Spaltentools' ribbon is active, showing options for 'Name', 'Datentyp', 'Format', 'Zusammenfassung', 'Datenkategorie', 'Nach Spalte sortieren', 'Datengruppen', 'Beziehungen verwalten', and 'Neue Spalte'. The formula bar contains the following DAX code:

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

Below the formula bar, a table is displayed with the following columns: 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

Abbildung 88

Umsatz letzte Periode ermitteln:

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

Tun

Veränderung absolut:

Turnover diff =

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

Veränderung in % ermitteln:

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

#### 5.8.1.2 Zeitintervalle bezogen auf aktuelles Datum ermitteln mit DATESBETWEEN()

\_Last 3 Months prev Year =

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

```

    CALCULATE(
        YEAR(MAX('Calendar'[Date])),
        ALL()
    )
VAR Spend= //Diese Lösung verwendet DATESBETWEEN
    CALCULATE(
        SUM(advertising[Spend]),
        KEEPFILTERS(
            DATESBETWEEN(
                'Calendar'[Date],
                DATE(LastYear-1,LastMonth-3,1), //LastMonth-3: Gehe 3 Monate zurück
                DATE(LastYear-1,LastMonth,1)
            )
        )
    )
RETURN Spend

```

### 5.8.1.3 Beliebige Zeitintervalle darstellen über FILTER()

\_Last X Months FILTER =

```

VAR LastMonth=
    CALCULATE(
        MONTH(MAX('Calendar'[Date])),
        ALL()
    )
VAR LastYear=
    CALCULATE(
        YEAR(MAX('Calendar'[Date])),
        ALL()
    )
VAR Spend= //Diese Lösung läuft über FILTER()
    CALCULATE(
        SUM(advertising[Spend]),
        KEEPFILTERS( //Behält den Filter des Visuals und fügt den nachfolgenden Filter
hinzu
            FILTER(
                'Calendar', //LastMonth-3 geht um 3 Monate zurück
                '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

```

## 5.8.2 YTD (Year to date)

Ausführliches Beispiel in *DAX\_Functions\_YTD.pbix*

### 5.8.2.1 YTD Jan-Jul

Ausgaben ermitteln von Jan-Jul, wobei hier der Juli

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

#### 5.8.2.2 YTD letzter (Aktueller) Monat

```
Spend latest Month =
VAR _LastMonth = CALCULATE (
    MONTH ( MAX ( dDate[Date] ) ),
    ALL ( dDate ) //Hier wird der Filterkontext über ALL() komplett ignoriert
)
VAR LM = CALCULATE (
    SUM ( Spend[mGrossMediaSpend] ),
    KEEPFILTERS ( MONTH ( dDate[Date] ) <= _LastMonth ) //Filterkontext bleibt erhalten und
                                                    //ein zusätzlicher Filter wird hinzugefügt
)
RETURN LM
```

#### 5.8.3 IF()

Ordner: Dax\_Functions/Dax Functions

**Hinweis:** Alle Zeilenberechnungen laufen im Zeilenkontext, d.h. in jeder einzelnen Zeile der Datentabelle wird eine Berechnung durchgeführt.

##### Alter ermitteln:

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

##### Altersklasse ermitteln:

```
Age Class =
//Ermittelt vier Altersklassen
if(Customer[Age]>50, "50+",
if(Customer[Age]>40, "40-50",
if(Customer[Age]>30, "30-40",
"u30"))))
```

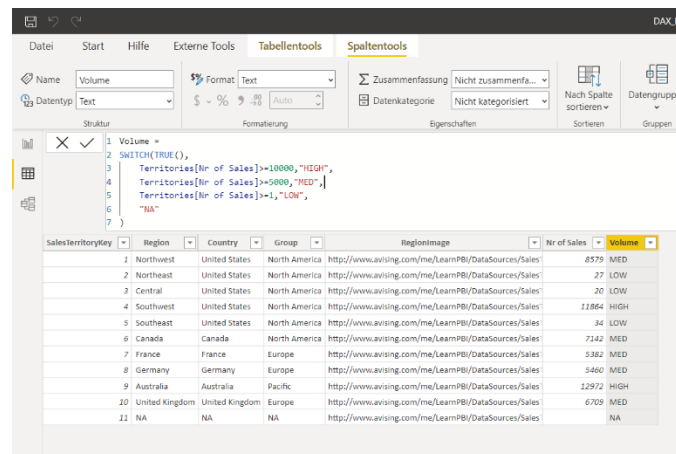
#### 5.8.4 Switch

Anzahl der Verkäufe ermitteln:

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

Regionen nach Umsatz klassifizieren:

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



DAX Formel:

```

1 Volume =
2 SWITCH(TRUE(),
3   Territories[Nr of Sales]>=10000,"HIGH",
4   Territories[Nr of Sales]>=5000,"MED",
5   Territories[Nr of Sales]>=1,"LOW",
6   "NA"
7 )

```

SalesTerritoryKey	Region	Country	Group	RegionImage	Nr of Sales	Volume
1	Northwest	United States	North America	http://www.avising.com/me/learnPB/DataSource/Sales	8579	MED
2	Northeast	United States	North America	http://www.avising.com/me/learnPB/DataSource/Sales	27	LOW
3	Central	United States	North America	http://www.avising.com/me/learnPB/DataSource/Sales	20	LOW
4	Southwest	United States	North America	http://www.avising.com/me/learnPB/DataSource/Sales	11864	HIGH
5	Southeast	United States	North America	http://www.avising.com/me/learnPB/DataSource/Sales	34	LOW
6	Canada	Canada	North America	http://www.avising.com/me/learnPB/DataSource/Sales	7142	MED
7	France	France	Europe	http://www.avising.com/me/learnPB/DataSource/Sales	5382	MED
8	Germany	Germany	Europe	http://www.avising.com/me/learnPB/DataSource/Sales	5460	MED
9	Australia	Australia	Pacific	http://www.avising.com/me/learnPB/DataSource/Sales	12972	HIGH
10	United Kingdom	United Kingdom	Europe	http://www.avising.com/me/learnPB/DataSource/Sales	6709	MED
11	NA	NA	NA	http://www.avising.com/me/learnPB/DataSource/Sales		NA

Abbildung 89

### 5.8.5 Measures im Ordner zusammenfassen

Ordner: DAX\_Funtions/DAX\_Functions\_Measures.pbix

Praktisch: Spalten und Measures können im Datenmodell nach Ordnern sortiert werden. Dazu müssen die gewünschten Objekte markiert werden und unter einem Ordernamen zusammengefasst werden:

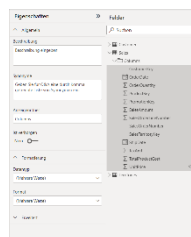


Abbildung 90

### 5.8.6 Gesamtwerte ermitteln um %-Anteile zu rechnen

File: DAX\_Functions\_Measures.pbix/Calculate If (Database: AdventureWorks\_Database.xlsx von AviS)

Um %-Anteile zu berechnen müssen je Zeile Gesamtwerte ermittelt werden. Dafür kann über CALCULATE() und ALL() zunächst der Filterkontext des Visuals übergangen werden:

Total Sales (All Countries) =

//Gesamt für alle Regionen ohne Leerdatensätze

```

if(
  ISBLANK([Turnover]),
  BLANK(),
  CALCULATE(
    [Turnover],
    all(Territories[Country]) //REMOVEFILTERS() möglich anstelle von 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 %
<b>Gesamt</b>	<b>€ 29.307.837</b>	<b>29.307.836,53</b>	<b>100,00 %</b>

Abbildung 91

*Country %*: Anschließend kann *Turnover* durch *Total Sales* dividiert werden.

### 5.8.7 Variablen in DAX

Für mehr Lesbarkeit und bessere Performace empfiehlt sich der Einsatz von Variablen.

#### Allgemeines Beispiel:

Revenue YoY % =

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

#### Beispiel für %-Vergleich:

Total Sales (All Countries) =

```
//Gesamt für alle Regionen ohne Leerdatensätze
var SalesAllCountries=
if(
    ISBLANK([Turnover]),
    BLANK(),
    CALCULATE(
        [Turnover],
        all(Territories[Country]) //REMOVEFILTERS() möglich anstelle von all()
    )
)
var PercentCountries=
    [Turnover]/SalesAllCountries
return PercentCountries
```

#### Hinweis: Warum nicht Werte anzeigen als...%?

Beim Einsatz eines Slicers beziehen sich %-Anteile auf die im Slicer selektierten Elemente.

Ein Measure kann aber auf die Gesamtheit (Alle Länder) permanent fixiert werden:

Country	Country	Turnover	Total Sales (All Countries)	Country %	%SSG: Turnover
<input type="checkbox"/> Australia	Australia	€ 9.051.786	29.307.836,53	20,89 %	82,15%
<input type="checkbox"/> Canada	Canada	€ 1.966.991	29.307.836,53	6,71 %	17,85%
<input type="checkbox"/> France	France				
<input type="checkbox"/> Germany	Germany				
<input type="checkbox"/> N/A					
<input type="checkbox"/> United Kingdom					
<input type="checkbox"/> United States					
	<b>Gesamt</b>	<b>€ 11.018.757</b>	<b>29.307.836,53</b>	<b>37,60 %</b>	<b>100,00%</b>

Abbildung 92

### 5.8.8 OR/AND/IN

Über ODER im Calculate können z.B. France ODER Germany zusammengefasst und mit einem anderen Land verglichen werden:

France and Germany =

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

Alternativ:

France and Germany =

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

**Oder mit IN:**

France and Germany =

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

### 5.8.9 AND/IN mit FILTER() kombinieren

Large Sales =

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

### 5.8.10 Zeitauswertungen

#### 5.8.10.1 Gesamtsumme Umsatz mit Rabatt in Q1

Turnover Q1 Rabatt mit USt =

VAR Turnover =

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

```

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

```

5.8.10.2 Laufende Summe

Datei: DAX\_Functions\_YTD.pbix

Funktion **TotalYTD**: YearToDay = TOTALYTD([Turnover], 'Calendar'[Date])

**TotalYTD** erzeugt laufende Summe, indem es bis zum Ende der jeweiligen Periode (Monat, Quartal etc. ) summiert

**Wichtig**: Es muss eine korrekte und verknüpfte Datumstabelle existieren.

**Hinweis**: Nach Aufdrillen der Tabelle muss beim Monaten ggf. nach Hilfsspalte indirekt sortiert werden

Im Diagramm können die einzelnen Jahre für den Bereich Jan-Jul verglichen werden. (Das Feld *Jahr* muss dabei in den Legende Bucket.)

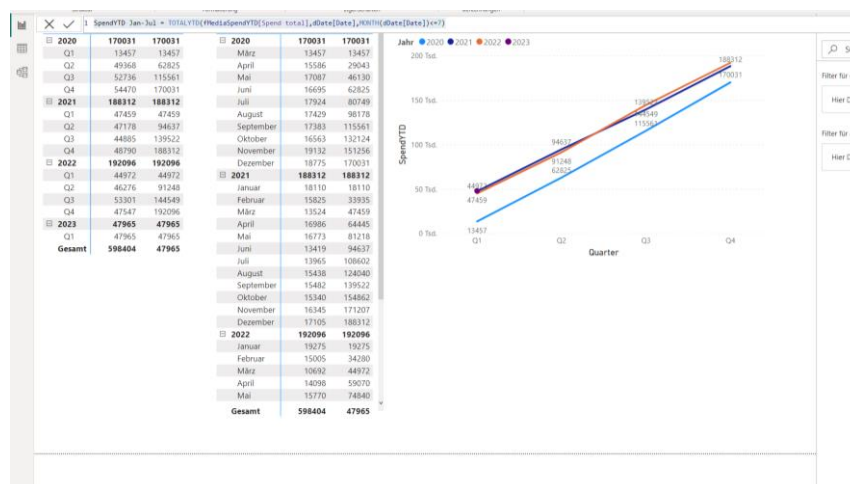


Abbildung 93

Ausgaben von Jan – Jul ermitteln:

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

**Weitere Themen**: Spend YTD Jan-Jul vs last Year, % Diff vs last Year in der Beispieldatei DAX\_Functions\_YTD

5.8.10.3 Laufende Summe für das Fiskaljahr

Wird das (Fiskal)Jahresende in TotalYTD definiert, Startet die Laufende Summe an diesem Datum neu:

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

Year	Turnover	YearToDay	Fiscal YTD
<b>2014</b>	<b>7.072.084,23</b>	<b>7.072.084,23</b>	<b>3.805.710,58</b>
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	598.746,56	3.865.120,21	3.865.120,21
Aug	550.216,59	4.415.336,80	4.415.336,80
Sep	644.135,20	5.059.472,00	5.059.472,00
Oct	663.692,29	5.723.164,29	5.723.164,29
Nov	673.556,20	6.396.720,49	6.396.720,49
Dec	676.763,65	7.072.084,23	7.072.084,23
<b>2015</b>	<b>5.762.134,24</b>	<b>5.762.134,24</b>	<b>3.037.501,33</b>
Jan	500.365,15	500.365,15	500.365,15
Feb	546.001,47	1.046.366,62	1.046.366,62
Mar	350.466,99	1.396.833,60	1.396.833,60
Apr	415.390,23	1.812.223,83	1.812.223,83
May	335.095,09	2.147.318,92	2.147.318,92
Jun	577.313,99	2.724.632,91	2.724.632,91
Jul	438.885,17	3.163.518,08	3.163.518,08
Aug	480.090,33	3.643.608,41	3.643.608,41
Sep	485.574,79	4.129.183,20	4.129.183,20
Oct	506.399,26	4.635.582,46	4.635.582,46
Nov	562.772,56	5.198.355,02	5.198.355,02
Dec	554.799,22	5.762.134,24	5.762.134,24
<b>2016</b>	<b>16.473.618,05</b>	<b>16.473.618,05</b>	<b>9.720.059,11</b>
Jan	886.668,84	886.668,84	886.668,84
Feb	847.413,51	1.734.082,35	1.734.082,35
Mar	1.010.258,13	2.744.340,48	2.744.340,48
Apr	1.080.449,58	3.824.790,06	3.824.790,06
May	1.196.981,11	5.021.771,17	5.021.771,17
Jun	1.731.787,77	6.753.558,94	6.753.558,94
Jul	1.340.244,95	8.093.803,89	8.093.803,89
Aug	1.462.479,83	9.556.283,72	9.556.283,72
Sep	1.480.905,18	11.037.188,90	11.037.188,90
Oct	1.608.750,53	12.645.939,43	12.645.939,43
Nov	1.878.317,51	14.524.256,94	14.524.256,94
Dec	1.949.361,11	16.473.618,05	16.473.618,05
<b>Gesamt</b>	<b>29.307.836,53</b>	<b>29.307.836,53</b>	<b>29.307.836,53</b>

Abbildung 94

**Hinweis:** Überschüssige Zeilen können über das Filterargument von TotalYTD mit Hilfe der FILTER() Funktion weggefiltert werden:

Fiscal YTD =

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

<b>2016</b>	<b>16.473.618,05</b>	<b>16.473.618,05</b>	<b>9.720.059,11</b>
Jan	886.668,84	886.668,84	886.668,84
Feb	847.413,51	1.734.082,35	1.734.082,35
Mar	1.010.258,13	2.744.340,48	2.744.340,48
Apr	1.080.449,58	3.824.790,06	3.824.790,06
May	1.196.981,11	5.021.771,17	5.021.771,17
Jun	1.731.787,77	6.753.558,94	6.753.558,94
Jul	1.340.244,95	8.093.803,89	8.093.803,89
Aug	1.462.479,83	9.556.283,72	9.556.283,72
Sep	1.480.905,18	11.037.188,90	11.037.188,90
Oct	1.608.750,53	12.645.939,43	12.645.939,43
Nov	1.878.317,51	14.524.256,94	14.524.256,94
Dec	1.949.361,11	16.473.618,05	16.473.618,05
<b>2017</b>			<b>9.720.059,11</b>
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
<b>Gesamt</b>	<b>29.307.836,53</b>	<b>29.307.836,53</b>	<b>29.307.836,53</b>

Abbildung 95

Umsätze nur für die die Wochenendtage ermitteln:

Turnover Weekend =

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

#### 5.8.10.4 Saldo am Periodenende

Um den Saldo am Ende einer Periode, z.B. am Monatsende, zu ermitteln bietet sich LASTDATE() an:

Last Balance Date =

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

Dabei kann das Problem auftreten, dass es zum Monatsende kein Datum gibt, da dies z.B. ein Wochenende war. Dies kann über LASTNONBLANK() gelöst werden:

Last Nonblank Balance =

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

Dies holt uns zum Ende einer Periode, z.B. zum Ende eines Monats, das letzte nichtleere Datum.

Umgekehrt kann auch mit FIRSTNONBLANK() und FIRSDATE() gearbeitet werden.

Mehr Info: Mitchel Pearson, Pragmatic Works: [\(26\) Beginning Power BI DAX Functions Tutorial \[Full Course\] - YouTube](#)

### 5.8.11 Kapazitäten ermitteln

**Ordner:** Advanced/Auslastung Kapazitäten

Problem: Je nach Drillstufe in einem Diagramm müssen unterschiedliche Berechnungen durchgeführt werden.

Auf welcher Hierarchiestufe ein Diagramm gerade gedrillt ist, ermittelt die Funktion ISINSCOPE() So kann in Kombination mit SWITCH() die jeweilige Aktion in einem Measure abgeleitet werden:

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]
```

))

Hinweis: Damit ISINSCOPE() wie im oberen Beispiel funktioniert muss eine Masterdate-Table im Datenmodell existieren.

### 5.8.12 Zeilenkontext

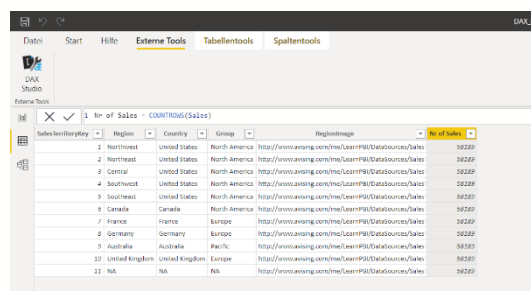
**Ordner:** Dax\_Functions/Dax Functions

Dieser Ausdruck liefert lediglich die Gesamtanzahl von Zeilen einer zugeordneten Tabelle, da hier der Filterkontext fehlt:

Nr of Sales = COUNTROWS(Sales)

Dies holt ebenfalls nur das Gesamtergebnis/Gesamtsumme

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



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

Abbildung 96

Lösungen:

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

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

RELATEDTABLE() holt die zur jeweiligen Zeile der Mastertabelle, alle zugehörigen Zeilen der Detailtabelle

oder:

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

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

CALCULATE() – holt den Filterkontext der Mastertabelle

### 5.8.13 KEEPFILTERS()

Wird vorzugsweise zusammen mit CALCULATE() eingesetzt. Soll CALCULATE nicht vollständig den Filterkontext übergehen, sondern lediglich einige eigene Filter zum Filterkontext **hinzufügen**, so ist KEEPFILTERS() zu verwenden.

#### Beispiel:

Spend latest Month =

```
VAR _LastMonth = CALCULATE (
    MONTH ( MAX ( dDate[Date] ) ),
    ALL ( dDate ) //Hier wird der Filterkontext über ALL() komplett ignoriert
)
VAR LM = CALCULATE (
    SUM ( Spend[mGrossMediaSpend] ),
    KEEPFILTERS ( MONTH ( dDate[Date] ) <= _LastMonth ) //Filterkontext bleibt erhalten und
                                                    //zusätzlicher Filter wird hinzugefügt
)
RETURN LM
```

### 5.8.14 DATATABLE()

Erstellt eine Tabelle, erlaubt aber keine skalaren Ausdrücke. (Berechnete Werte)

```
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") }
```

Hier besteht aber nicht die Möglichkeit, Spaltennamen mitzugeben, dafür sind skalare Werte möglich

### 5.8.15 Zugriff auf vorherige Zeilen in DAX (Beta)

Zunächst muss eine Indexspalte erzeugt werden:

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

**Hinweis:** Mittels RANKX() werden nur eindeutige Werte fortlaufend durchnummeriert.

### 5.8.16 Lookupvalue

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

**Beispiel:** Umsatz (Total) anzeigen für das Jahr 2017 UND Kalenderwoche 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

Abbildung 97

Alternativ:

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

### 5.8.17 VALUES()

Liefert eine Spalte als Tabelle zurück und entfernt Duplikate.

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

(Berechnet Umsätze je Farbe)

### 5.8.18 SELECTEDVALUE()

Sofern eine Spalte auf einen einzelnen Wert zusammengefiltert wurde, liefert die Funktion diesen zurück. Falls nicht, wird das Alternative Ergebnis zurückgegeben.

Beispiel: `Select Period = SELECTEDVALUE('Select Period'[Value], "Alternatives Ergebnis")`

### 5.8.19 LOOKUPVALUE()

Liefert wie X SVerweis die zum Suchkriterium zugehörige Information.

### 5.8.20 ALL() ALLSELECTED() ALLEXCEPT()

- ALL() zeigt alle Datensätze und übergeht alle anderen Filter
- ALLSELECTED() zeigt alle Datensätze die im Slicer selektiert wurden
- ALLEXCEPT() zeigt alle Datensätze außer.... Im Beispiel unten außer Country, hier wird der Zeilenkontext des Visuals - der zeilenweise nach Country filtert – berücksichtigt

TIP: ALLEXCEPT() kann zum Zeilenkontextbasiertem Gruppieren verwendet werden, z.B. CALCULATE(Berechnung,ALLEXCEPT(Tabelle, Gruppenspalte))

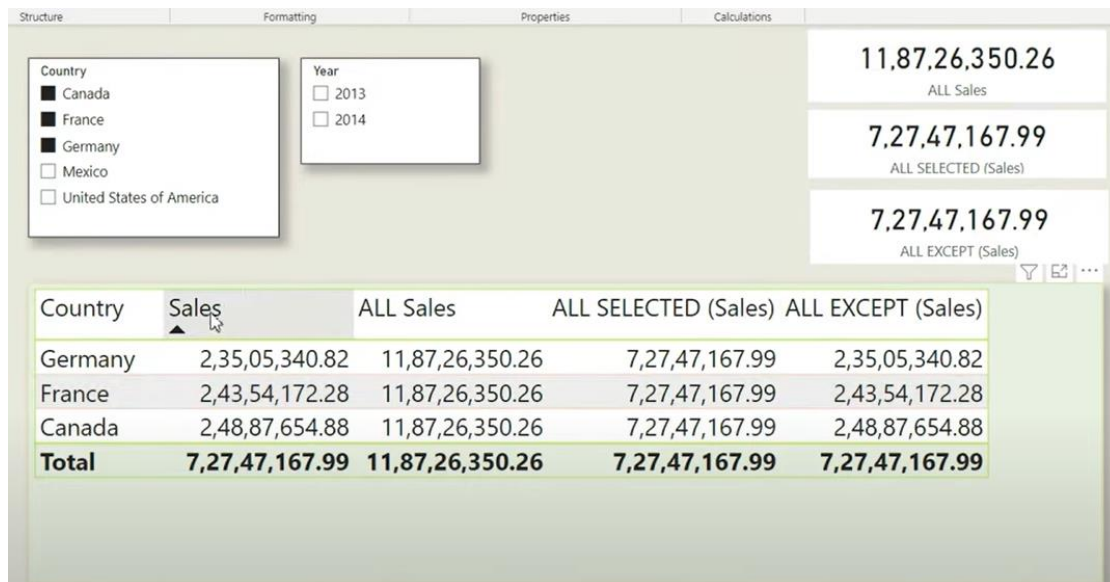


Abbildung 98

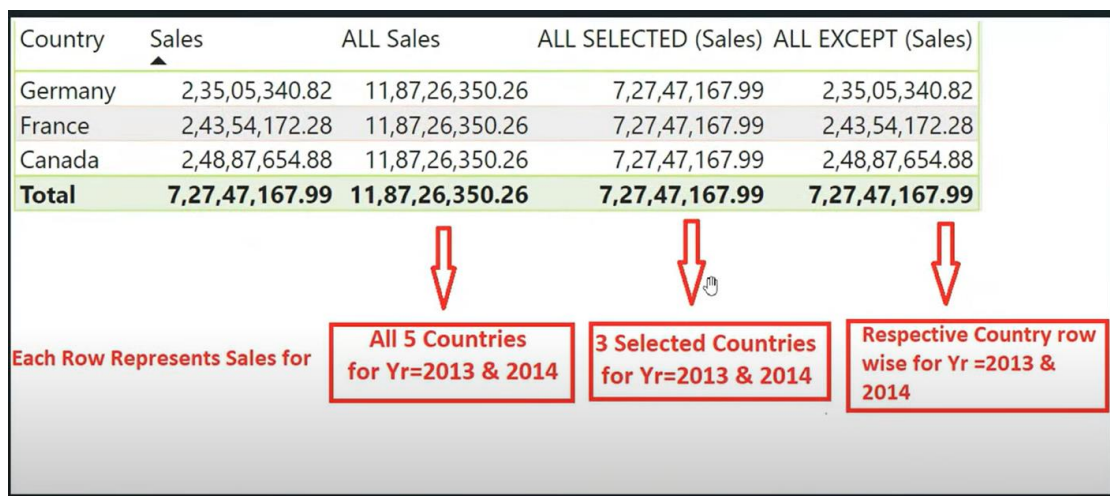


Abbildung 99

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

### 5.8.21 CONVERT()

Konvertiert den Datentyp eines Wertes

MSRRegion =

IF(

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

"-", //Datentyp TEXT

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

Hier muss im ELSE Zweig des IF das Ergebnis von CALCULATE() in einen Text konvertiert werden, um den Datentyp in der gesamten Spalte anzugleichen

### 5.8.22 SUMMARIZE()

Ergebnis ist eine Tabelle

Gruppiert nach Spalten, und erzeugt neue berechnete Spalten mit aggregierten Ergebnissen.

#### SUMMARIZE( fSales, //Aus dieser Tabelle gruppieren fSales[SalesTerritoryKey],dProducts[Color], //diese Spalten gruppieren "Turnover", SUMx(fSales,fSales[OrderQuantity]\*RELATED(dProducts[ListPrice])) //neue Spalte mit Aggregation ) )

## 5.9 Visuals

### 5.9.1 Drill through

Ordner Story: Fiscal Periods/Sales Overview und Sales Details

Benötigt eine eigene Detailseite zu der über die rechte Maustaste gedrillt werden kann.

Wird z.B. auf Seite eins ein bestimmtes Jahr angeklickt, muss auf dem Visual der Detailseite das Jahr in den Drillthrough Bucket gezogen werden. So können z.B. im Visual die Monate des auf Seite eins selektierten Jahres angezeigt werden.

Beispiel:

- Hauptseite zeigt Länder
- Drillthrough Seite zeigt Details zu Städten, z.B. Umsatz je Stadt:

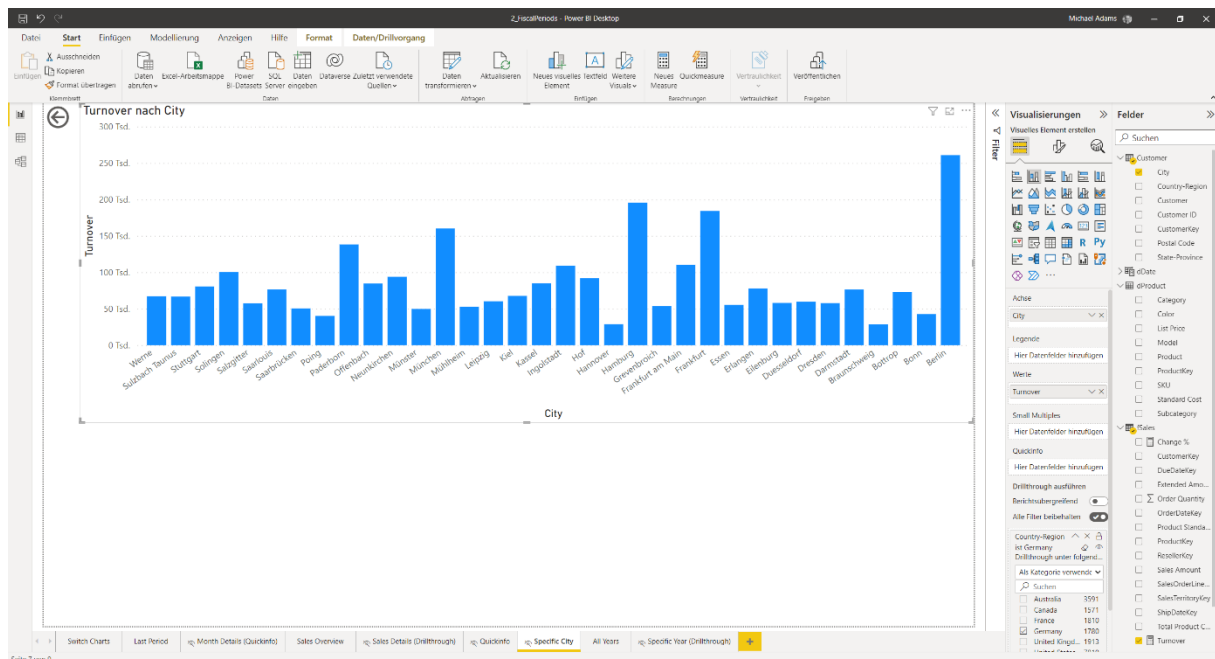


Abbildung 100

**Tipp:** Drillthrough kann auch mit einem Button veranschaulicht werden:



Abbildung 101

## Notizen

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

### 5.9.2 Bookmarks und Interaktionen

Ordner Story: 1\_SwitchVisuals

Bookmark einrichten:

- Ansicht Lesezeichen aktivieren
- *Bookmark/Hinzufügen*
- Mit *Daten* bedeutet, dass auch Filtereinstellungen gespeichert werden

Bildaktion einrichten

- Bild einfügen
- Farbröller: *Aktion*, Typ: *Lesezeichen*

#### 5.9.2.1 Umschalten zwischen Jahren und Quartalen im Visual

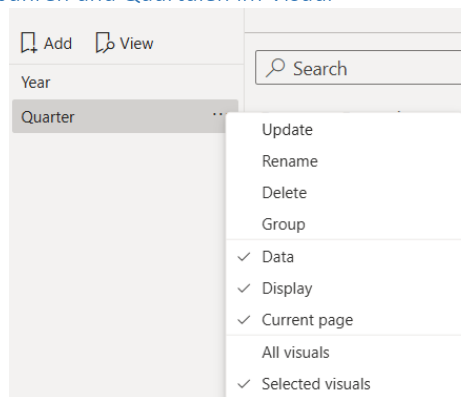


Abbildung 102

Zunächst auf Jahre oder Quartale drillen, dann im neuen Bookmark *Selected Visuals* wählen

### 5.9.3 Quickinfo

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

Zunächst muss eine Quickinfoseite eingerichtet werden auf der die gewünschten Details angezeigt werden:

Fiscal Month	Turnover Last Period
1	6,413,376,06
2	10,828,784,86
3	7,225,032,49
4	3,926,519,81
5	6,442,321,96
6	4,475,152,04
7	7,957,523,93
8	11,379,882,52
9	11,031,346,95
10	6,742,751,49
11	11,585,515,96

Abbildung 103

**TIPP:** Es empfiehlt sich die Quickinfoseite auszublenden, damit diese im Reporting nicht angezeigt wird.

Danach auf dem gewünschten Visual die Quickinfo aktivieren und die Detailseite auswählen:

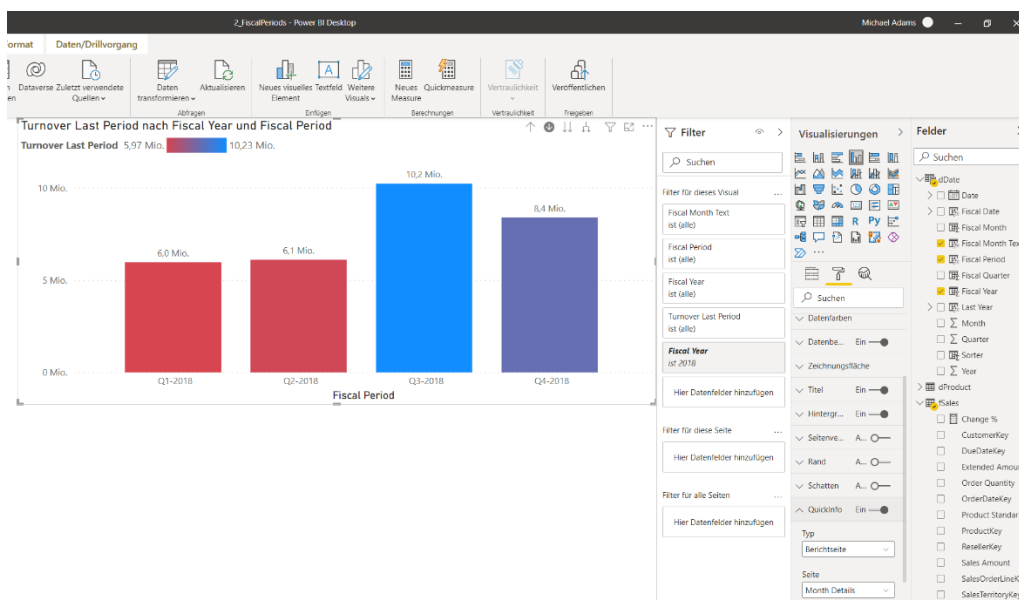


Abbildung 104

### 5.9.4 Small Multiples

Ordner: Small\_Multiples

Erst Daten importieren und Datenmodell erstellen

Tabelle Sales: Quartale können nach einzelnen Jahren aufgeschlüsselt werden

oder: Gender M/F nach Jahren aufschlüsseln.

### 5.9.5 Map Visuals

Ordner: Advanced/2\_FiscalPeriods.pbix, Blatt Sales Overview

**Hinweis:** Falls die Maps Visuals nicht aktiv sind, müssen diese im Power BI Verwaltungsportal aktiviert werden:

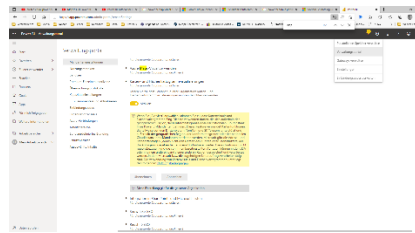


Abbildung 105

### 5.9.6 Landkarte

Zeigt Werte mit Blasen auf der Landkarte.

Formatierung: Blasen/Größe und individuelle Farben

### 5.9.7 Flächenkartogramm

Zeigt Regionen als Fläche. Hier sind die Ausfüllfarben als Farbskala nützlich

### 5.9.8 Sparkline

Ordner: Advanced/2\_FiscalPeriods.pbix/Sparklines

Nachdem z.B. ein Gesamtumsatz in einer Tabelle ermittelt wurde, können in einer zusätzlichen Spalte Sparklines erstellt werden: *Einfügen/Sparkline einfügen*

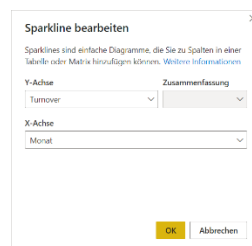


Abbildung 106

Hier können z.B. die Monate und deren Umsätze gewählt werden, um den Umsatzverlauf zu visualisieren:

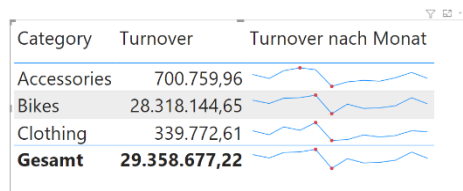


Abbildung 107

Formateinstellungen für Sparklines:

Dateneinstellungen für Sparklines:

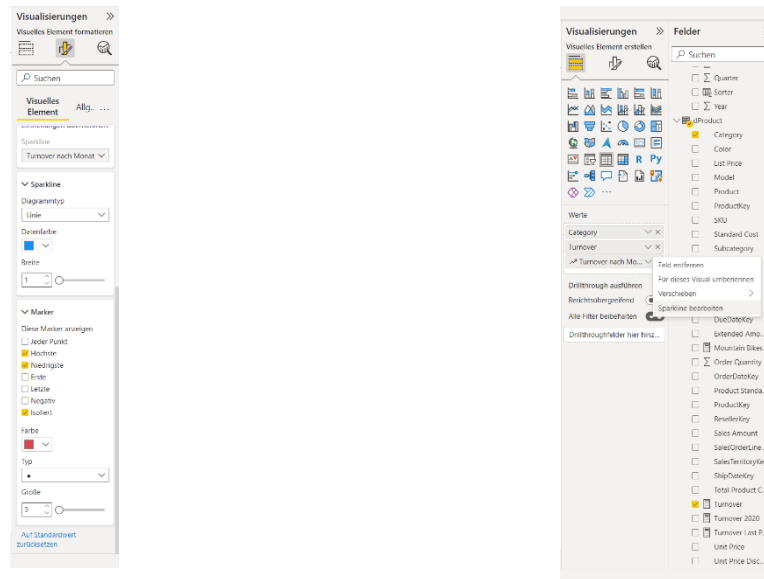


Abbildung 108

### 5.9.9 Bänderdiagramm (Ribbon Chart)

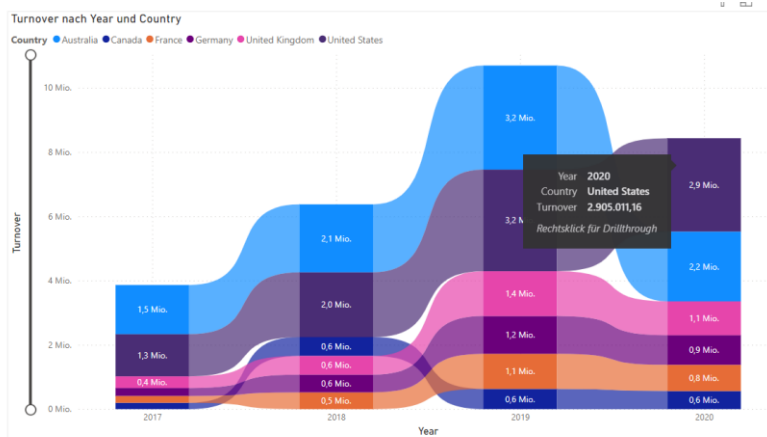


Abbildung 109

Empfehlenswerte Einstellungen:

- Zoomschieber
- Datenbeschriftungen
- Diagramm per Filter auf wesentliche Kategorien ggf. reduzieren

### 5.9.10 Analysebaum (Decomposition Tree)

Ermöglicht die Feinanalyse von Zahlen/Daten z.B. nach

- Land
- Produktfarbe
- Geschlecht

Über die KI Funktion werden bei Wahl des höchsten oder niedrigsten Wertes automatisch die nächsten Kategorien gewählt:

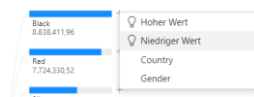


Abbildung 110

Alternativ kann die Kategorie selbst gewählt werden.  
 Im nachfolgenden Beispiel wird über die niedrigste Kategorie analysiert, welche Farbe, in welchem Land von welchem Geschlecht am wenigsten gekauft wird:

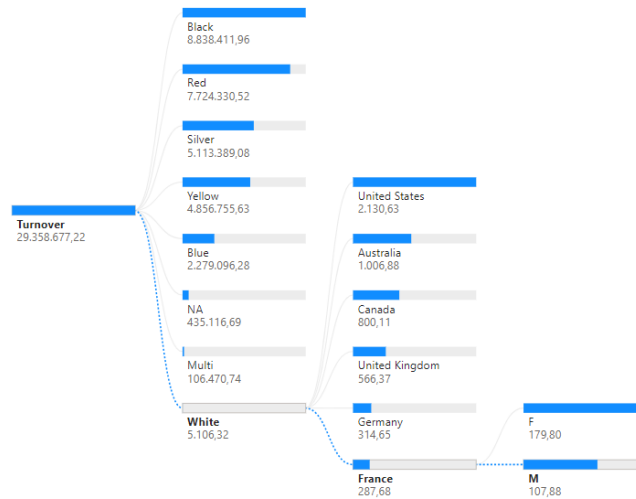


Abbildung 111

### 5.9.11 Punktdiagramm

Ermöglicht die Auswertung mehrerer Dimensionen z.B.

- Umsatz, Gewinn, Stückzahl, Region und Zeit

Über die Play-Taste kann sogar der Zeitliche Ablauf animiert werden.

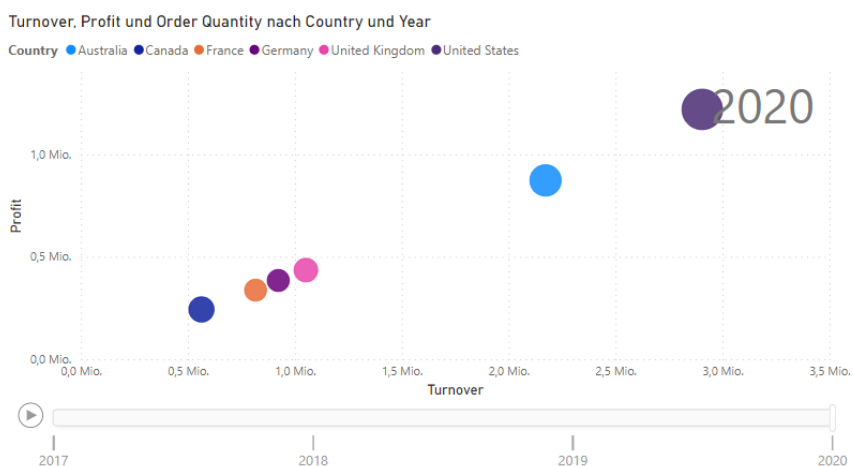


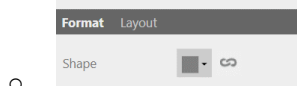
Abbildung 112

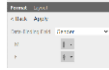
### 5.9.12 Visuals Hinzufügen (Bsp.: Infographic Designer)

Vorab: **Filter** auf France und Color=White, Daten auf Gender und Turnover

Vergleich nach Order Turnover und Geschlecht:

- Format/Shape: Data Binding, um den Datenpunkten die Shapes zuzuordnen





- Multiple Units: ON
- Value Color: ON -> Einstellung Farbe
- Formateinstellungen in PBI: Chart Type: Bar

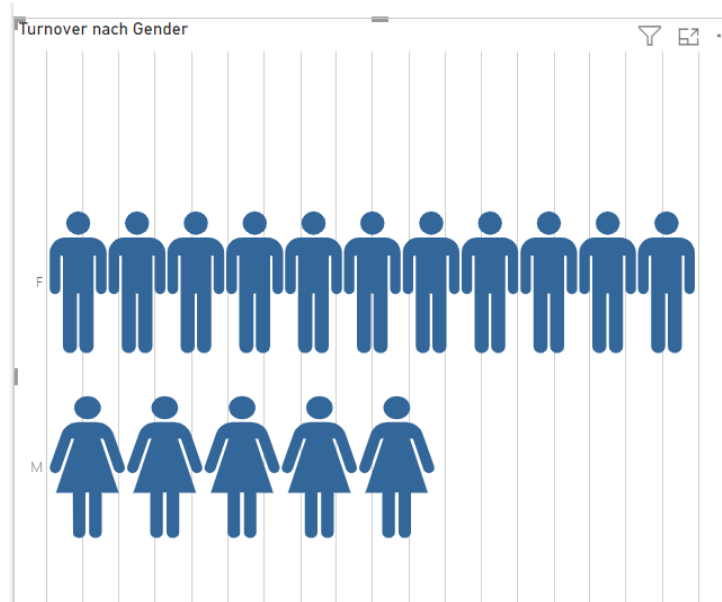


Abbildung 113

Tipp: Weitere Unterteilung z.B. nach *Legend* oder *Row By* nach Country

## 5.10 Power BI Dienst

### 5.10.1 Online gespeicherte Daten (Datasets) automatisch aktualisieren

Voraussetzung: Daten müssen auf Sharepoint liegen.

- Login auf dem Power BI Portal (z.B. <https://app.powerbi.com>)
- Gewünschten Arbeitsbereich wählen
- Für das gewünschte **Dataset** (Orangenes Icon) das Ellipsis (Drei Punkte) Menü wählen dann *Settings*:

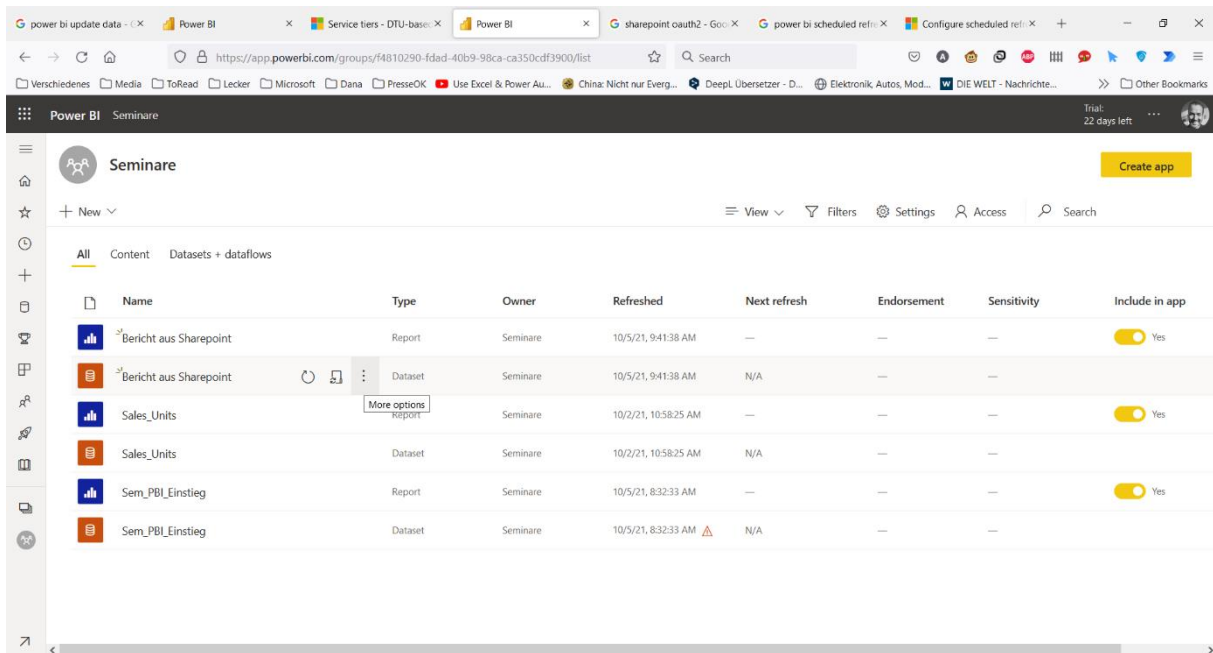


Abbildung 114

- Unter *Data Source Credentials* die Zugangsdaten zum Sharepoint hinterlegen:

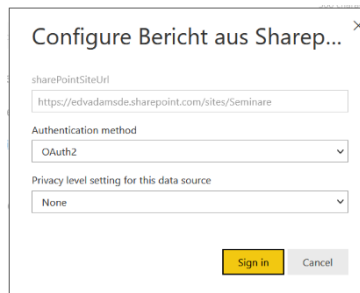


Abbildung 115

- Unter *Scheduled Refresh* die automatische Aktualisierung (Update) aktivieren:

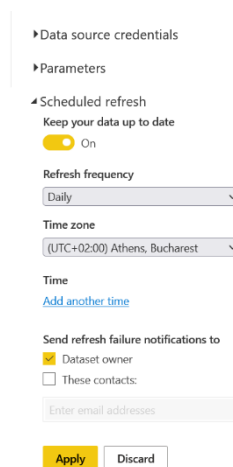
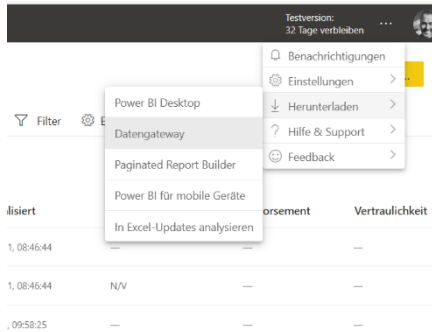


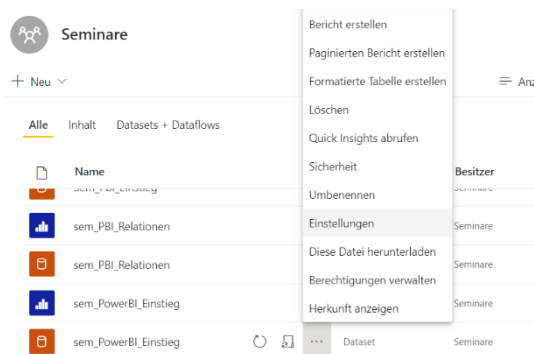
Abbildung 116

### 5.10.2 Power BI Data Gateway einrichten

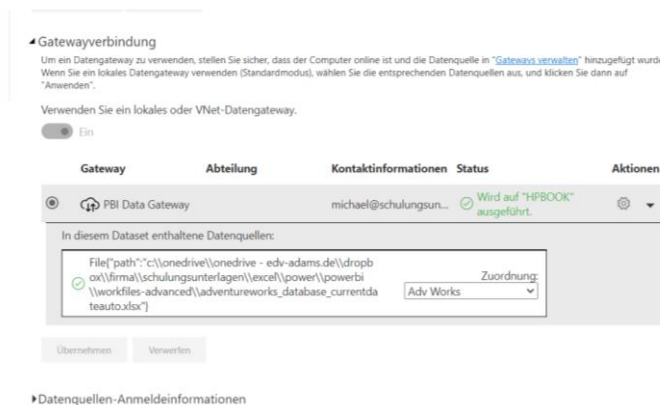
Dient der Aktualisierung bei On Premise abgelegten Daten.  
Data Gateway downloaden und installieren



Für das gewünschte Dataset die Einstellungen wählen:



Gatewayverbindung hinzufügen oder verwalten:



### 5.10.3 Daten aus Teams importieren

- Teams: Go to files.
- Click on "Open in SharePoint"
- Once in the SharePoint, open the Excel file on Excel app.
- On Excel file, click on "Info" & copy the path.
- Go to Power BI > Get Data > From Web > Basic.
- Paste the link from Excel file.
- Remove the "? web=1" from the link.
- Click on "OK"

### 5.10.4 Dashboard anlegen

Ein Dashboard ist eine Sammlung von Visuals aus anderen Berichten. Neues Anlegen:

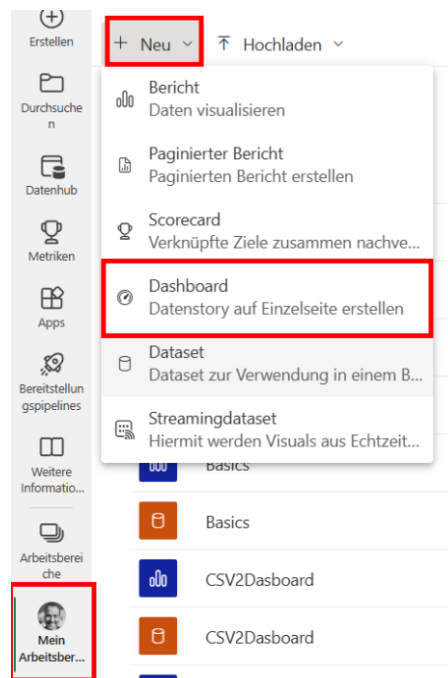


Abbildung 117

Anschließend können Berichtsseiten einem Dashboard hinzugefügt werden:

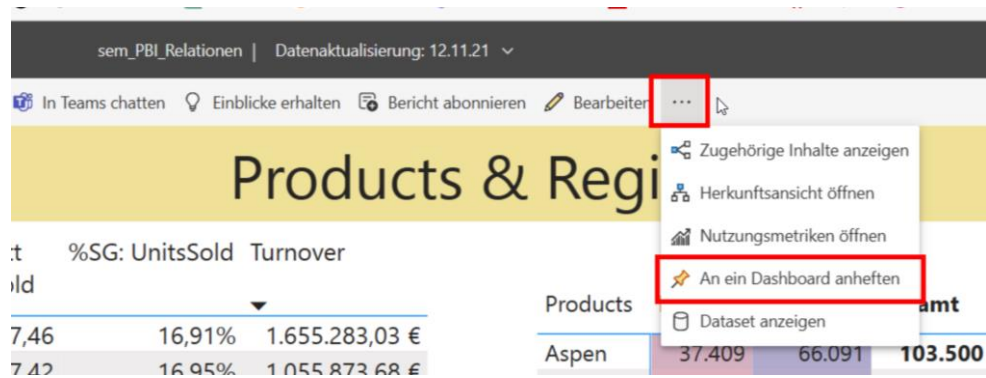


Abbildung 118

### 5.10.5 Daten aus Power BI (Portal) in Excel analysieren

Im Power BI Portal das gewünschte Projekt wählen und Dreipunkte Menü wählen:

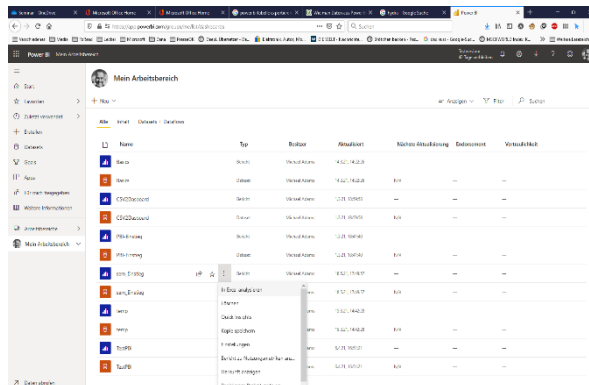
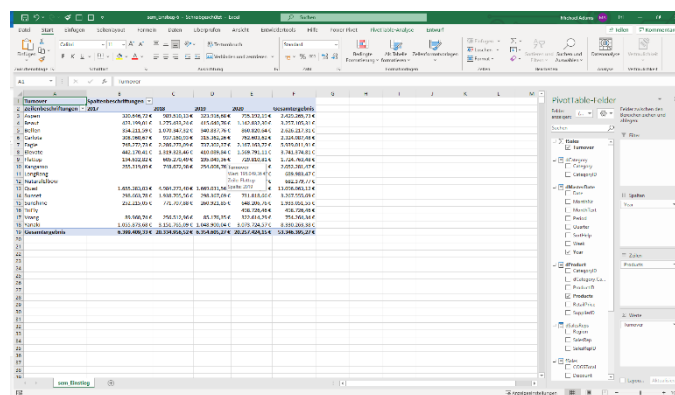


Abbildung 119

Danach erfolgt der Aufbau einer MDX Verbindung zu Excel:



Danach z.B. ein Tabellenvisual einfügen und die Tabelle mit Bild-URLs als Datenquelle verwenden:

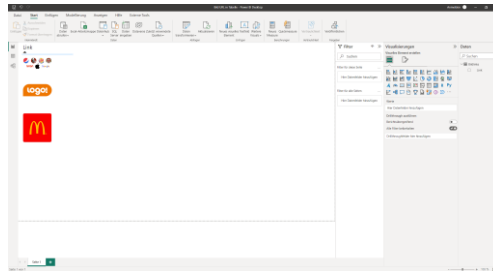


Abbildung 122

Die Skalierung der Bilder kann in den Formateigenschaften des Visuals eingestellt werden:

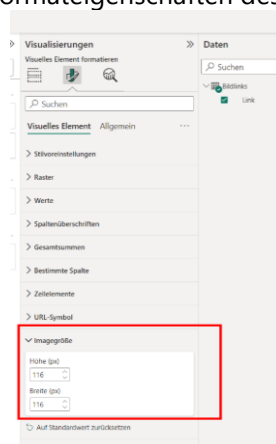


Abbildung 123

## 6 Power BI – PL-300 Zertifizierungsthemen

### 6.1 Power Query

#### 6.1.1 Direct Query

Ist langsamer als Import (Kopie)

Kann beschleunigt werden durch ein für Direct Query optimiertes Datenmodell

#### 6.1.2 SQL-Abfragen

Ordner: PL-300\Workfiles\Power Query Parameter

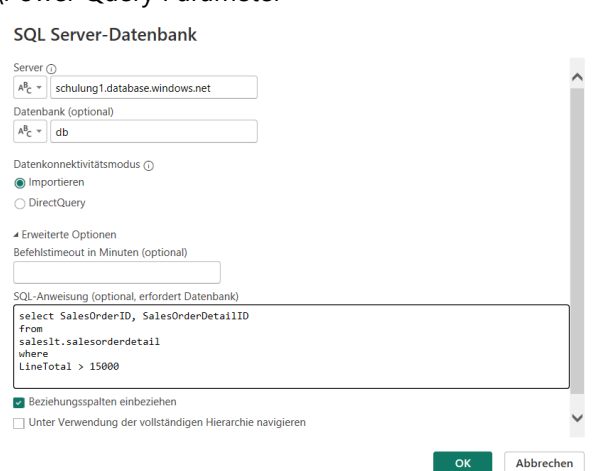


Abbildung 124

Diese Abfrage kann in Power Query im ersten Transformationsschritt über das Zahnrad bearbeitet werden.

#### Einzelwert als Parameter

In die SQL Abfrage können zuvor angelegte Parameter (hier Price) eingesetzt werden:  
 = Sql.Database("schulung1.database.windows.net", "db", [Query="select Name, Color, ListPrice#(lf)from#(lf)saleslt.Product#(lf)where #(lf)ListPrice > " & Price ])

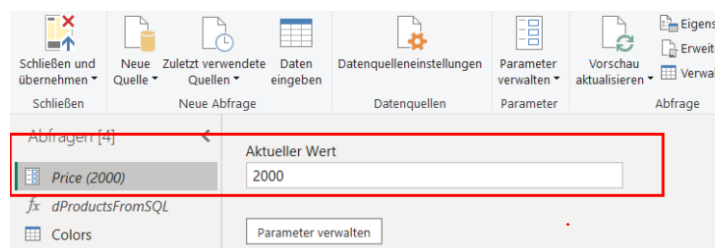


Abbildung 125

#### Function die Zeilenwerte als Parameter empfängt

(pColor as any)=>

let

```
Quelle = Sql.Database("schulung1.database.windows.net", "db",
[Query="select Name, Color,
ListPrice#(lf)from#(lf)saleslt.Product#(lf)where #(lf)Color = '" &
pColor & "'"])
```

```
//test = "select Name, Color,
ListPrice#(lf)from#(lf)saleslt.Product#(lf)where #(lf)Color = '" &
pColor
```

in

//test

Quelle

Start		Ergebnis		

### 6.1.3 Parameter für Datenquellen

Ordner: Parameters

Parameter ermöglichen das schnelle Ändern einer Datenbankquelle. Zunächst unter *Ansicht* Parameter zulassen:

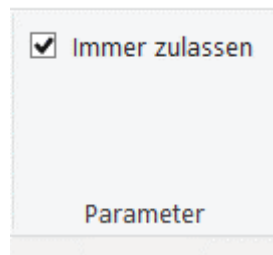


Abbildung 126

Dann in Power Query die Verbindung zu einer Datenquelle herstellen, wobei jetzt im Dropdown die Möglichkeit besteht ein Parameter zu wählen oder zu erstellen:

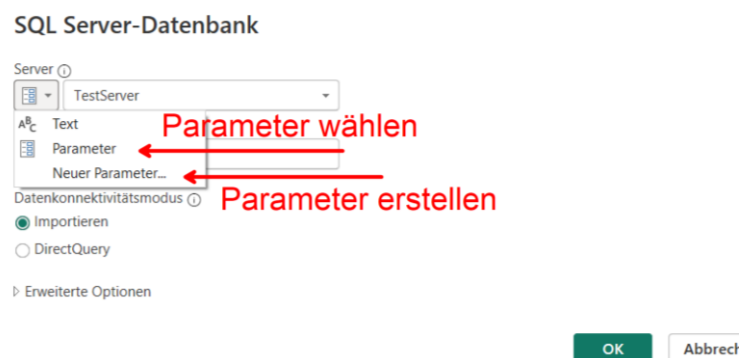


Abbildung 127

So können durch unterschiedliche Parameter schnell Verbindungen zu unterschiedlichen Datenbanken hergestellt werden:

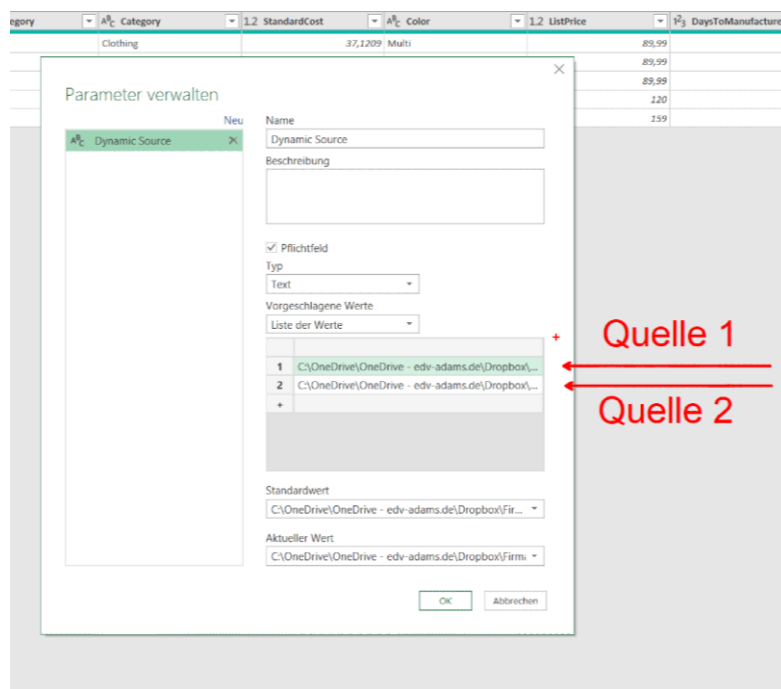


Abbildung 128

Anschließend kann die Datenquelle schnell und einfach über *Parameter bearbeiten* geändert werden:

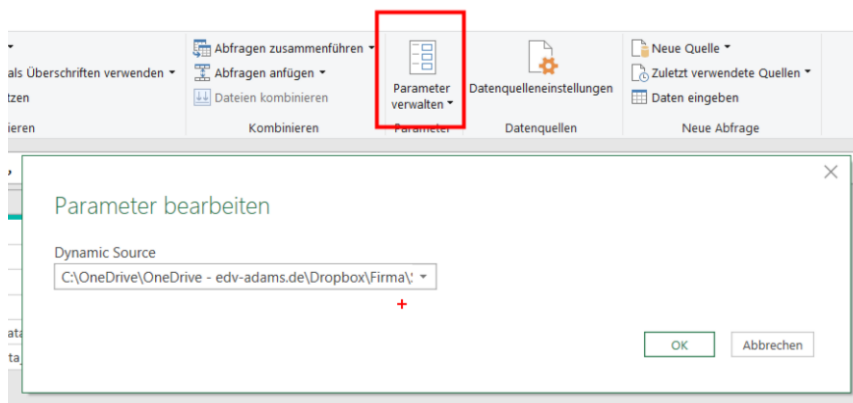


Abbildung 129

#### 6.1.4 Speichermodus

DirectQuery, wenn keine lokalen Kopien der Daten aus Sicherheitsgründen erlaubt sind. Danach ist z.B. GroupBy nicht möglich

#### 6.1.5 MasterDate in Power Query

MasterDate möglich via DAX oder direkt in Power Query:

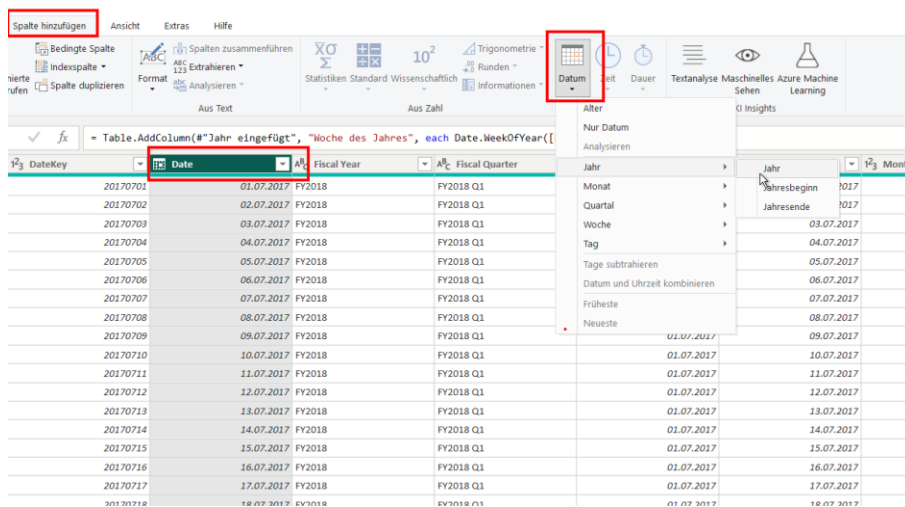


Abbildung 130

## 6.2 Data Model

### 6.2.1 Datumstabelle markieren

In Datensicht, Tabellentools/Als Dat-Tabelle markieren.

Hinweise:

- Das Markieren als Datumstabelle entfernt Hierarchien aus dem Datumsfeld
- CALENDAR(Start, End)
- CALENDERAUTO(fiscal\_year\_end\_Month) – Erstellt eine Liste für ein ganzes Jahr mit dem Endmonat *fiscal\_year\_end\_Month*.
- Beispiel für CALENDERAUTO(6): Die-Funktion durchläuft alle Datums- und Datums-/Uhrzeit-Spalten in Ihrem Modell, um die frühesten und spätesten gespeicherten Datumswerte zu ermitteln. Anschließend wird ein vollständiger Satz von Datumsangaben erzeugt, der alle Datumsangaben in Ihrem Modell umfasst, sodass sichergestellt wird, dass Datumsangaben für vollständige Jahre geladen werden. Wenn beispielsweise das früheste in Ihrem Modell gespeicherte Datum der 15. Oktober 2021 ist, lautet das erste von der CALENDERAUTO-Funktion zurückgegebene Datum 1. Juli 2021. Wenn das späteste in Ihrem Modell gespeicherte Datum der 15. Juni 2022 ist, lautet das letzte von der CALENDERAUTO-Funktion zurückgegebene Datum 30. Juni 2022.

### 6.2.2 N:M

Es ist zwar standardmäßig möglich eine N:M Beziehung im Datenmodell anzulegen, jedoch muss man sich darüber im Klaren sein, dass die Ergebnisse nicht unbedingt das sind, was man erwarten würde. Ferner funktioniert RELATED() nicht.

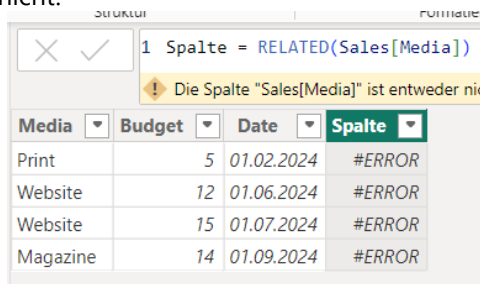


Abbildung 131

## Standard N:M

Hier fehlen Positionen (Website):

Media	Summe von Sales	Date
Magazine	11	17.04.24
Movie	13	15.08.24
Paper	9	25.09.24
Print	16	02.01.24
Video	5	18.02.24
Video	8	16.04.24
<b>Gesamt</b>	<b>62</b>	

Media	Summe von Budget	Date
Magazine	14	01.09.2024
Print	5	01.02.2024
Website	12	01.06.2024
Website	15	01.07.2024
<b>Gesamt</b>	<b>46</b>	

Media	Summe von Sales	Summe von Budget
Magazine	11	14
Movie	13	
Paper	9	
Print	16	5
Video	13	
<b>Gesamt</b>	<b>62</b>	<b>46</b>

Abbildung 132

### Lösung 1 – Denormalisierung in Power Query

Dabei werden die Tabellen aneinandergesetzt und ausgewertet:

Media	Summe von Sales	Summe von Budget
Magazine	11	14
Movie	13	
Paper	9	
Print	16	5
Video	13	
Website		27
<b>Gesamt</b>	<b>62</b>	<b>46</b>

Abbildung 133

### Lösung 2 – Relationstabelle

Hier wird die gemeinsame Kriteriumsspalte (Media) von Duplikaten befreit und mit jeweils 1:N an die beiden anderen Tabellen angebunden:

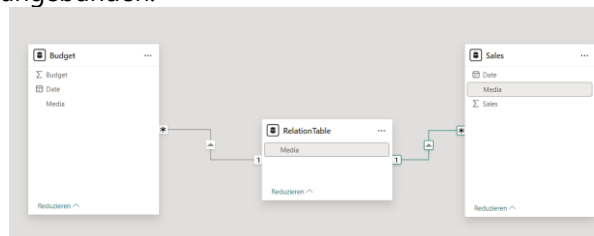


Abbildung 134

## 6.2.3 Row Level Security

### 6.2.3.1 Statische Rolle erstellen



Abbildung 135

Rolle erstellen:

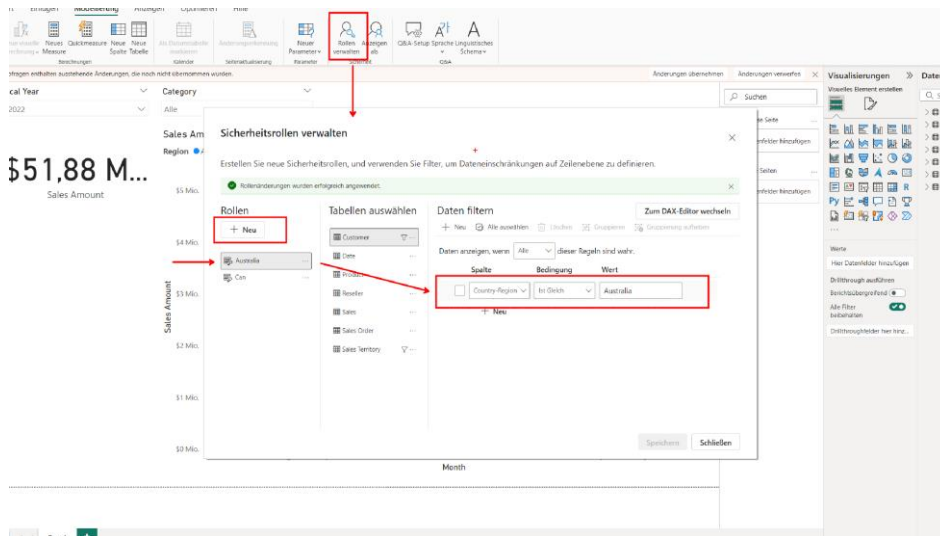


Abbildung 136

Rolle einem User oder Gruppe zuweisen:  
Für das gewünschte Dataset die Sicherheitseinstellungen wählen:

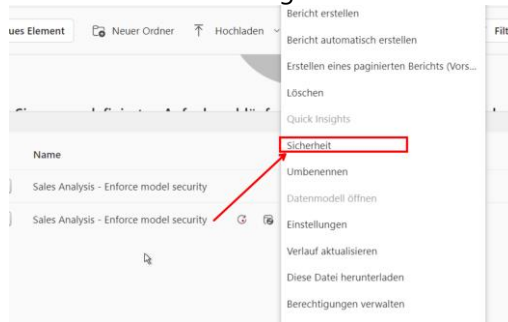


Abbildung 137

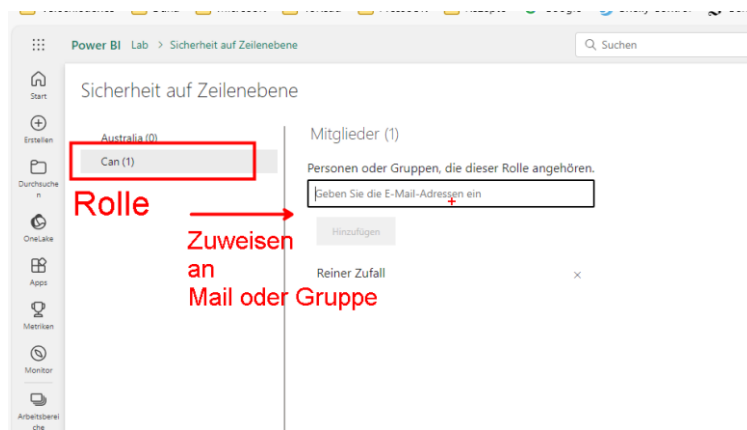


Abbildung 138

Ggf. in M365 vorher eine Gruppe anlegen.

6.2.3.2 Dynamische Rollen erstellen

Voraussetzung: Usernamen in der Tabelle über die der Zugriff geregelt werden soll:

UPN	SalesTerritoryKey	Region	Country	Group	Regi
adams@edv-adams.de	1	Northwest	United States	North America	http://www.avisings.com/me/LearnP
ivana@edv-adams.de	2	Northeast	United States	North America	http://www.avisings.com/me/LearnP
reiner@edv-adams.de	3	Central	United States	North America	http://www.avisings.com/me/LearnP
dana@edv-adams.de	4	Southwest	United States	North America	http://www.avisings.com/me/LearnP
ivana@edv-adams.de	5	Southeast	United States	North America	http://www.avisings.com/me/LearnP
reiner@edv-adams.de	6	Canada	Canada	North America	http://www.avisings.com/me/LearnP
dana@edv-adams.de	7	France	France	Europe	http://www.avisings.com/me/LearnP
adams@edv-adams.de	8	Germany	Germany	Europe	http://www.avisings.com/me/LearnP
ivana@edv-adams.de	9	Australia	Australia	Pacific	http://www.avisings.com/me/LearnP
reiner@edv-adams.de	10	United Kingdom	United Kingdom	Europe	http://www.avisings.com/me/LearnP
	11	NA	NA	NA	http://www.avisings.com/me/LearnP

Abbildung 139

Danach kann über *Rollen Verwalten* ein DAX-Ausdruck hinzugefügt werden, über den der Zugriff des jeweiligen Users auf die gewünschten Datensätze zugelassen ist:

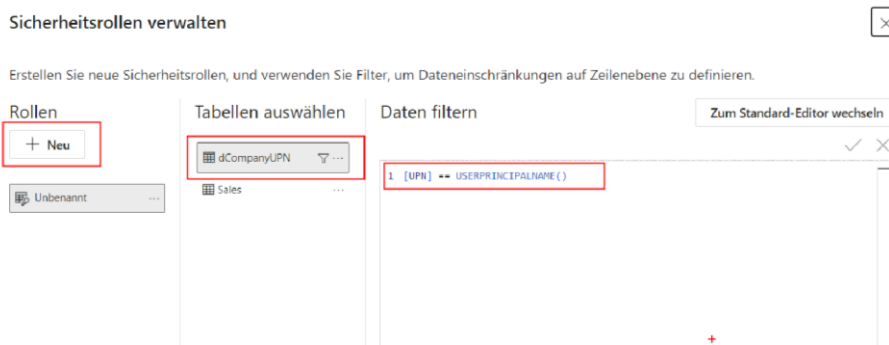


Abbildung 140

Über *Anzeigen als* kann die jeweilige Rolle getestet werden:

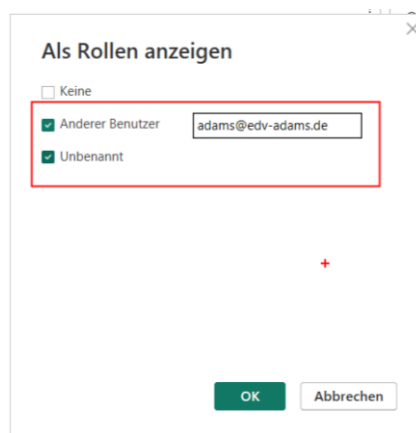


Abbildung 141

### 6.2.4 Kreuzfilter

Gegeben ist das folgende Datenmodell:

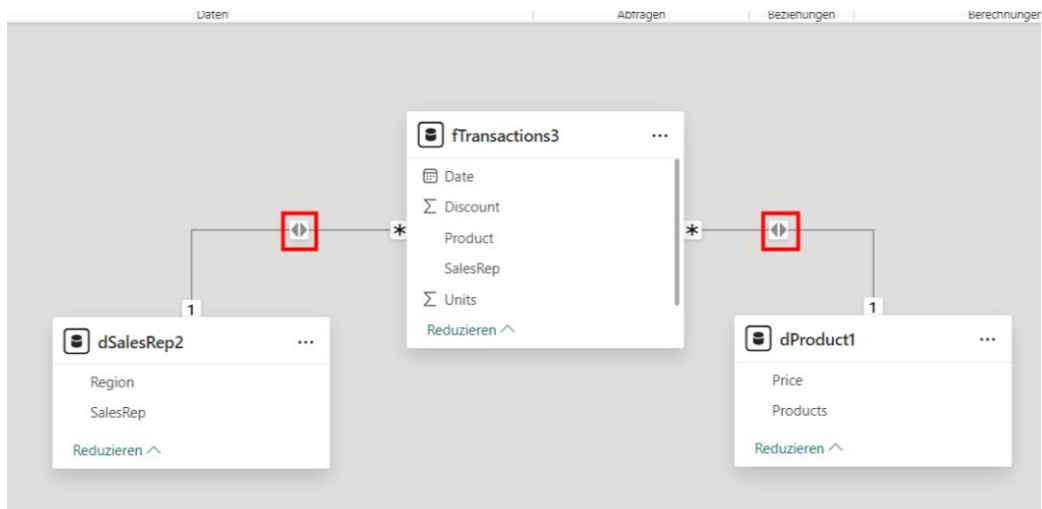


Abbildung 142

Um die zur Region zugehörigen Produkte anzeigen zu können (N:M) muss der Kreuzfilter in beide Richtungen aktiviert werden.

Ergebnis:

Product	Summe von Units
Carlota	4
Crested Beaut	2
Majectic Beaut	1
Sunbell	1
Sunshine	4
<b>Gesamt</b>	<b>12</b>

Abbildung 143

Da diese Technik sehr Ressourcenintensiv sein kann, empfiehlt sich eher das Zusammenfiltern der betroffenen Slicer über:

Cascading = INT(NOT ISEMPTY(fTransactions3))

Abbildung 144

### 6.2.5 Datengranularität anpassen

Gegeben ist die Tabelle *Budget*:

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

Abbildung 145

Die Granularität Year/Month passt nicht zur MasterDate Tabelle, daher die Anpassung:

- Year + Month verketteten
- In den Typ *Date* wandeln
- Danach kann die Beziehung hergestellt werden

### 6.2.6 Analyse in Excel

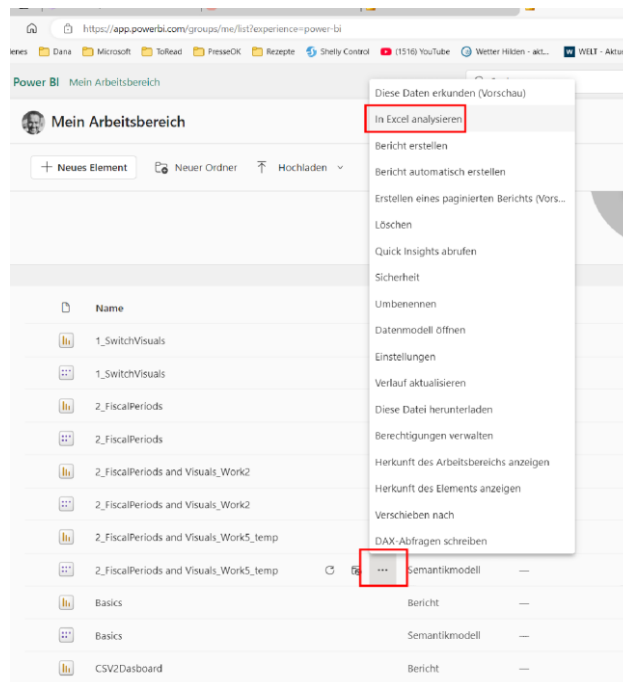


Abbildung 146

### 6.2.7 Hierarchie Pfade

Ordner: Path and Hierarchy

Über Funktionen wie PATH oder PATHITEM können Hierarchien in Dimensionstabellen zur gezielten Auswertung aufgeschlüsselt werden. Dies ist vorteilhaft, da so nicht nur eine Gesamtübersicht erstellt werden kann, sondern eine detaillierte Auswertung möglich ist.

Eine schnelle Visualisierung kann zunächst erstellt werden:

Visual Hinzufügen: hierarchy chart

Employee ID	Name	Position	Manager ID
1	James	General Manager	
3	Janette	EU Region Manager	1
7	Janine	Consultant	3
2	John	UK Region Manager	1
4	Kiera	Developer	2
5	Pam	Developer	3
6	Rosie	Consultant	2

Abbildung 147

### Path und Level ermitteln

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

L1 = `PATHITEM(Employees[Path],1)`

Pos In Hierarchy = `PATHLENGTH(Employees[Path])`

Employee ID	Name	Position	Manager ID	Path	L1	L2	L3	Pos In Hierarchy
1	James	General Manager		1	1			1
3	Janette	EU Region Manager	1	1 3	1	3		2
7	Janine	Consultant	3	1 3 7	1	3	7	3
2	John	UK Region Manager	1	1 2	1	2		2
4	Kiera	Developer	2	1 2 4	1	2	4	3
5	Pam	Developer	3	1 3 5	1	3	5	3
6	Rosie	Consultant	2	1 2 6	1	2	6	3

### Hierarchie erstellen

Über Rechtsklick auf das Feld der obersten Ebene kann eine neue Hierarchie erstellt werden. Weitere Stufen der Hierarchie können in der Modellansicht per Drag&Drop zugefügt werden:

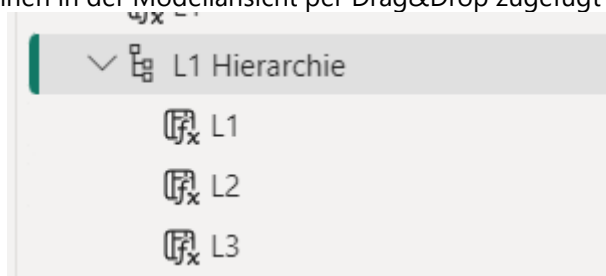
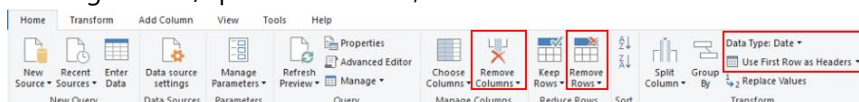


Abbildung 148

## 6.2.8 Leistungsoptimierung

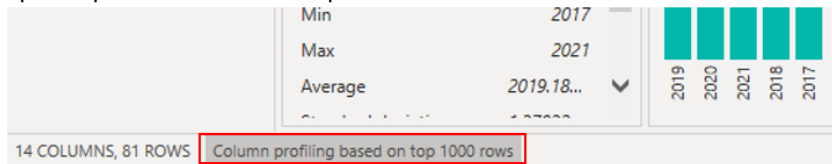
### 6.2.8.1 In Daten

- Unnötige Zeilen/Spalten entfernen, entweder:

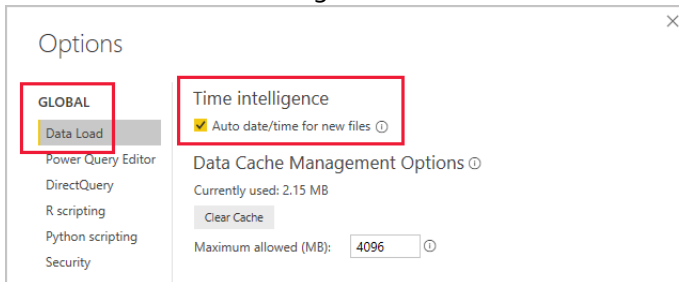


- Oder durch Gruppieren

- Spaltenprofil für alle Daten prüfen:



- Auto Time Intel aus und eigene MasterDateTable erstellen und als solches markieren



- Zusammenfassen (Gruppieren) von Daten um nicht benötigte Details zu entfernen (z.B. Tagesdatum und Produkt)
- Faktentabellen auf DirectQuery umstellen:
  - Es ist hilfreich, wenn häufige Datenänderungen anfallen und Berichte nahezu in Echtzeit erstellt werden müssen.
  - Es kann große Datenmengen verarbeiten, ohne dass diese vorab zusammengefasst werden müssen.
  - Es wendet Beschränkungen hinsichtlich der Datenhoheit an, um die gesetzlichen Vorgaben einzuhalten.
  - Es kann mit einer mehrdimensionalen Datenquelle verwendet werden, die Kennzahlen wie SAP Business Warehouse (BW) enthält.
    - Nachteile DirectQuery
      - Leistung – hängt von Quelle und Netz ab
      - Sicherheit – alle können die Daten sehen
      - Datentransformation – weniger Möglichkeiten
      - Berichterstellung – Q&A, Quick Insights nicht verfügbar
- Bei SQL-Abfragen nur relevante Daten importieren/Filtern

#### 6.2.8.2 Abfragenreduzierung

- Die Anzahl der Abfragen kann durch Deaktivierung aller standardmäßigen Interaktionen reduziert werden. Individuelle Interaktionen zwischen Visuals können weiterhin festgelegt werden.
- Bestätigungsschaltflächen für Slicer
- Bestätigungsschaltflächen für Standardfilter (Rechter Aufgabenbereich)

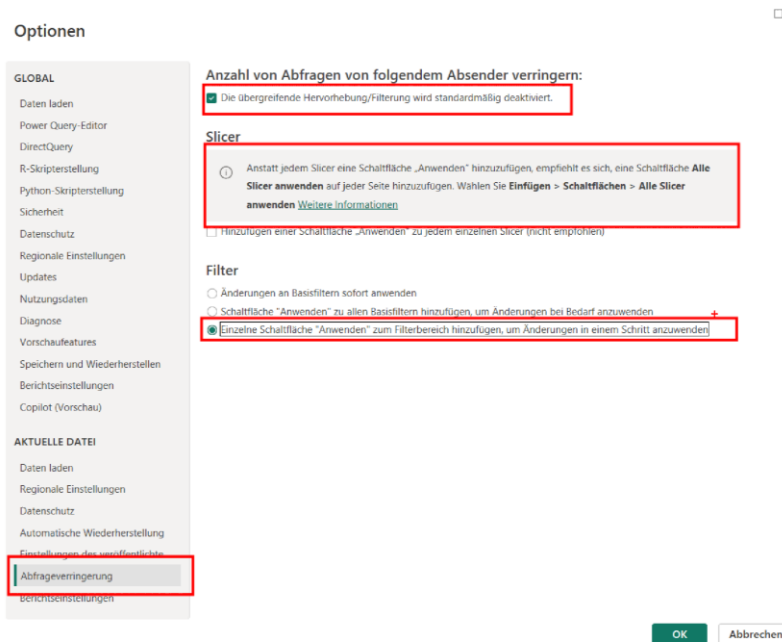


Abbildung 149

### 6.2.8.3 In DAX

- Verwendung von Variablen
  - Verbesserte Leistung
  - Verbesserte Lesbarkeit
  - Einfacheres Debuggen
  - Reduzierte Komplexität

#### Beispiel:

Revenue YoY % =

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

## 6.3 DAX

TIP: ctrl+k+c | ctrl+k+u //Block kommentieren/auskommentieren

### 6.3.1 Filterkontext RELATED(), RELATEDTABLE()

- RELATED() ruft den Wert auf der 1-Seite der Beziehung ab, Ergebnis: skalarer Wert
- RELATEDTABLE() ruft Werte auf der n-Seite der Beziehung ab, Ergebnis: Tabelle

- LOOKUPVALUE() entspricht XVerweis() und ruft einen eindeutigen Wert (1-Seite) bei nichtverknüpften Tabellen. Ergebnis: skalarer Wert (Gleiches Ergebnis wie RELATED(), wobei Related() performanter ist)

### 6.3.2 Time Intel

**Hinweis:** Zeitintel Funktionen ändern den Filterkontext!

**DatesYTD, DatesQTD, DatesMTD** erzeugt eine Datumsreihe (Tabelle) mit allen Daten seit Beginn der Periode bis zum aktuellen Datum im aktuellen Filterkontext.:

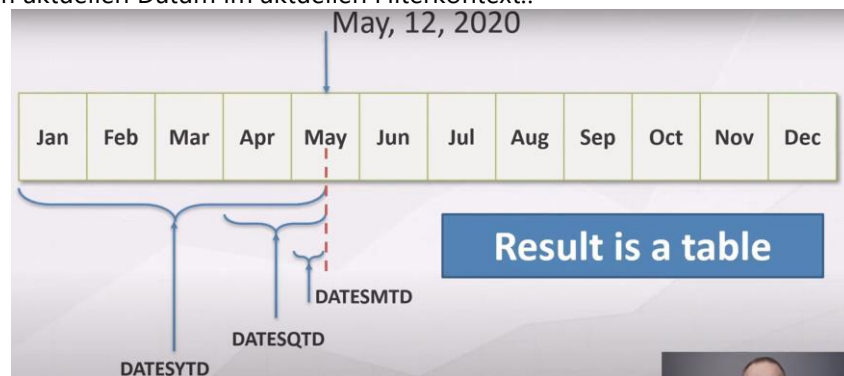


Abbildung 150

**DATESBETWEEN**(<Dates>, <StartDate>, <EndDate>)

Gibt eine Tabelle zurück, mit einer Datumsreihe von *Start* bis *Ende* aus einer beliebigen Spalte *Dates*.

**DATESINPERIOD**(<dates>, <start\_date>, <number\_of\_intervals>, <interval>)

Liefert eine Datumsreihe mit allen Einzeldaten einer bestimmten Anzahl von Intervallen.

⇒ Alle obigen Funktionen eignen sich als **Filter für CALCULATE**

### TOTALYTD vs CALCULATE

Units YTD CALCULATE = `CALCULATE(SUMX(fSales, fSales[Order Quantity]), DATESYTD(dDate[Date]), ["6-30"]) //6-30 => Start Fiskaljahr`

Units YTD TOTALYTD = `TOTALYTD(SUM(fSales[Order Quantity]), dDate[Date])`

Beide Funktionen liefern dasselbe Ergebnis, allerdings kann CALCULATE beliebig viele weitere Filter verwenden. TOTALYTD hingegen nur einen.

#### 6.3.2.1 Weitere

- [DATEADD](#)
- [PARALLELPERIOD](#)
- [SAMEPERIODLASTYEAR](#)
- [NEXTDAY](#), [NEXTMONTH](#), [NEXTQUARTER](#), [NEXTYEAR](#) sowie [PREVIOUSDAY](#), [PREVIOUSMONTH](#), [PREVIOUSQUARTER](#) und [PREVIOUSYEAR](#)
- [FIRSTDATE](#) und [LASTDATE](#)
- DATEVALUE - "1/8/2009", is converted to a datetime value equivalent to January 8th of 2009

#### 6.3.2.2 LTD und DATESBETWEEN

Wenn in DATESBETWEEN StartDate BLANK ist, dann ist StartDate das früheste Datum *in der Spalte dates* und **nicht** das früheste Datum im *Kontext des Visuals*

2017	Juli	290	01.07.2017 00:00:00
2017	August	449	01.07.2017 00:00:00
2017	September	610	01.07.2017 00:00:00
2017	Oktober	784	01.07.2017 00:00:00
2017	November	1014	01.07.2017 00:00:00
2017	Dezember	1202	01.07.2017 00:00:00
2018	Januar	1395	01.07.2017 00:00:00
2018	Februar	1573	01.07.2017 00:00:00

Abbildung 151

daher kann

```
DATESBETWEEN(dDate[Date],MINX(all(dDate[Date]),dDate[Date]),MAX(dDate[Date]))
```

durch

```
DATESBETWEEN(dDate[Date],BLANK(),MAX(dDate[Date]))
```

ersetzt werden.

Somit kann LTD (Life To Date) wie folgt berechnet werden:

```
LTD Measure = CALCULATE(
    DISTINCTCOUNT(fSales[CustomerKey]),
    //dDate[Date]=DATE(2017,7,4) || dDate[Date]=DATE(2017,7,5)
    DATESBETWEEN(dDate[Date],BLANK(),MAX(dDate[Date]))
)
```

### 6.3.3 CALCULATE(), CALCULATETABLE(), FILTER()

CALCULATETABLE und FILTER können beide als Filterargument für CALCULATE verwendet werden.

Der Unterschied: FILTER ist ein Iterator, geht also Zeile für Zeile vor. CALCULATETABLE arbeitet im Speichermodell und gibt als Ergebnis eine (Filter)Tabelle zurück. Da CALCULATETABLE somit performanter ist, ist es der FILTER-Funktion zu bevorzugen sofern möglich.

Ogleich dieser Ausdruck...

```
VAR France=
```

```
    CALCULATE(
        [Turnover],
        FILTER(
            dTerritories,
            dTerritories[Country]="France"
        )
    )
```

```
)
```

```
RETURN
```

```
    DIVIDE(France,[Turnover], "")
```

...das selbe Ergebnis liefert, ist dieser Ausdruck performanter und somit best practice:

Turnover France =

```
VAR France=
```

```
    CALCULATE(
        [Turnover],
        CALCULATETABLE(
            dTerritories,
            dTerritories[Country]="France"
        )
    )
```

```
)
)
RETURN
    DIVIDE(France,[Turnover], "")
```

Siehe dazu: [DAX Fridays Battle #188: FILTER vs CALCULATETABLE - Avoid using FILTER as a filter argument](#)

### 6.3.4 Filtermodifikatoren

Siehe: **PL-300/Filters**

REMOVEFILTERS()

Entfernt den Filter aus Region

Region	Turnover	Turnover RED	Percentage
Canada	3.134.375,92	3.134.376	6,82%
United States	14.680.390,91	14.680.391	31,93%
Central	3.277,55	14.680.391	0,01%
Northeast	8.563,92	14.680.391	0,02%
Northwest	5.806.971,92	14.680.391	12,63%
Southeast	19.442,30	14.680.391	0,04%
Southwest	8.842.135,22	14.680.391	19,23%
Pacific	14.180.824,84	14.180.825	30,84%
Australia	14.180.824,84	14.180.825	30,84%
<b>Total</b>	<b>45.978.015,30</b>	<b>45.978.015</b>	<b>100,00%</b>

Abbildung 152

KEEPFILTERS()

1 Turnover RED =  
2 CALCULATE(  
3 [Turnover],  
4 KEEPFILTERS(dProducts[Color]="red")  
5 )  
6 )

Color	Turnover	Turnover RED
Black	13.767.379,62	
Blue	3.670.684,49	
Multi	160.143,90	
NA	650.517,94	
Red	12.151.129,11	12.151.129,11
Silver	7.986.843,23	
White	7.695,44	
Yellow	7.583.621,58	
<b>Total</b>	<b>45.978.015,30</b>	<b>12.151.129,11</b>

Keepfilters - behält den aktuellen Filterkontext (des Visuals) und **fügt einen Filter hinzu**.  
Damit bleiben alle Zeilen außer RED leer, da z.B. in Zeile eins der Filter BLACK (Visualkontext) UND RED vom Measure aktiv sind.

Abbildung 153

### 6.3.5 VALUES, HASONEVALUE, SELECTEDVALUE, ISINSCOPE

VALUES liefert eine die Liste *eindeutiger* Elemente aus einer Spalte oder Tabelle – im jeweiligen Filterkontext!

Category	Count of CustomerKey	ValsCustomer
Accessories	34319	14564
Bikes	15205	9132
Clothing	8665	6567
<b>Total</b>	<b>58189</b>	<b>17918</b>

VALUES ermittelt hier die Anzahl *eindeutiger* Datensätze im Filterkontext des Visuals, also Anzahl der Datensätze zu jeder Category.  
Das Ergebnis von VALUES ist immer eine Tabelle.

Abbildung 154

SELECTEDVALUE liefert den aktuell selektierten (gefilterten/Filterkontext) Wert. Ergebnis ist BLANK wenn mehrere Werte selektiert sind.

HASONEVALUE prüft ob das Filterergebnis nur einen Wert beinhaltet

ISINSCOPE

Europe	False
France	False
France	True
Germany	True
United Kingdom	False
United Kingdom	True
NA	False
NA	False
NA	True
North America	False
Canada	False
Canada	True
United States	False
Central	True
Northeast	True
Northwest	True
Southeast	True
Southwest	True
Pacific	False
Australia	False
Australia	True
Total	False

ISINSCOPE gibt TRUE zurück, wenn sich die angegebene Spalte in der Hierarchie auf der gleichen Ebene befindet.

IsInScope =  
ISINSCOPE(dTerritories[Region])

TRUE=Alle Regionen der dritten Ebene, also der Region Ebene

Abbildung 155

### 6.3.6 Skalare Werte und EVALUATE

EVALUATE

ROW("Total Sales", SUM(Sales[Amount]))

### 6.3.7 Visual Calculations

Ordner: PL300/Visual\_Calculations

Visual Calculations beinhalten eigene DAX-Funktionen, z.B. RUNNINGSUM().

Besonderheiten:

- Gelten nur innerhalb des einen Visuals in dem sie platziert sind
- Es gilt nur die Filterung **des Visuals** und kein sonstiger Filterkontext
- Formatierung über FORMAT()

Month	Laufende Summe	/TD	Calculation
2018			
Gleitender Durchschnitt			
Prozent des übergeordneten Elements			
Prozent von Gesamtsumme			
Durchschnitt der untergeordneten Elemente			
Im Vergleich zu vorherigen			
Im Vergleich zum nächsten			
Im Vergleich zum ersten			
Im Vergleich zum letzten			
September	3.296.904,47	22.302.632,02	
Oktober	2.159.316,93	24.461.948,96	
November	3.612.282,67	28.074.231,63	
Dezember	2.627.079,29	30.701.310,92	

Abbildung 156

#### 6.3.7.1 Runningsum

RUNNINGSUM( <column>, [<axis>], [<blanks>], [<reset>] )

Column: Werte aus Spalte

Axis: Horizontal oder vertikal (ROWS,COLUMNS)

Reset: Laufende Summe zurücksetzen bei HIGHESTPARENT (z.B. Jahr), LOWESTPARENT (z.B. Tag) oder Ebene (1,2,3,..., z.B. Quartal)

6.3.7.2 Weitere Visual-Dax Funktionen

- MOVINGAVERAGE(Column, Window) Window=Anzahl benachbarter Werte
- COLLAPSE([Turnover],[Category]) //Benötigt KEIN ROWS – Liefert das Gesamtergebnis der übergeordneten Ebene
  - Beispiel: %-Anteil der über übergeordneten Ebene:  
= DIVIDE([Turnover], COLLAPSE([Turnover], rows))
- PREVIOUS(Field) – liefert den Wert der Vorperiode im Kontext des Visuals  
Beispiel für % der Vorperiode:  
= DIVIDE([Turnover] - PREVIOUS([Turnover]),[Turnover])

6.3.8 Quickmeasures

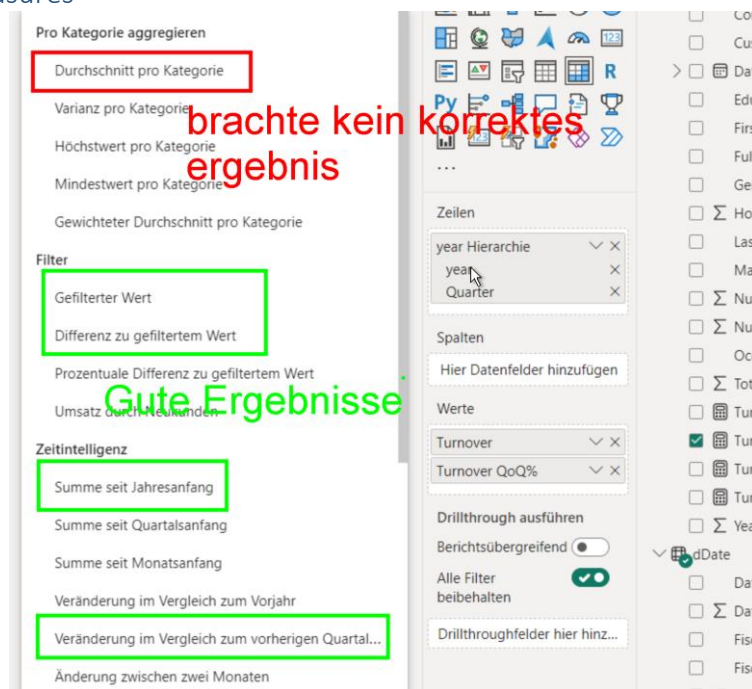


Abbildung 157

6.3.9 Semi-Additive Measures

Werden verwendet: um Salden am Ende einer Periode zu ermitteln und ggf. zu summieren. Dies summiert alle Einzelsalden, was nicht sinnvoll ist:

Stock (Wrong) =  
SUMX(MSFT\_Stock,MSFT\_Stock[Open])

Zeigt den Wert der letzten Periode, im aktuellen Kontext, was korrekt ist:

End of Period Balance Last Date =  
CALCULATE(  
MSFT\_Stock[Stock (Wrong)],  
LASTDATE(dDate[Date])  
) + 0

Da es sein kann, dass es sein kann passendes Datum in der fTable gibt, kann es vorteilhaft sein mit LASTNONBLANK() zu arbeiten, da dieses den letzten nicht-leeren Wert ermittelt:

```
End of Period Balance LastNonBlank =
    CALCULATE(
        MSFT_Stock[Stock (Wrong)],
        LASTNONBLANK(MSFT_Stock,MSFT_Stock[Stock (Wrong)])
    )
```

Notizen:

- Quelle = Table.FromRows(Json.Document(Binary.Decompress(Binary.FromText("i45WCkpNUYrViVZyyiIhTISk7OVYmMB", BinaryEncoding.Base64), Compression.Deflate)), let \_t = ((type nullable text) meta [Serialized.Text = true]) in type table [Color = \_t]),



### 6.3.10 Arten von DAX-Funktionen

Hinweis: Vergleichsoperator == (Genau gleich) wertet BLANK (nicht vorhandene) Werte nicht als Null (0) aus, somit ergibt BLANK==0 FALSE als Ergebnis.

## 6.4 Dashboards

### 6.4.1 Visuals zum Dashboard zufügen

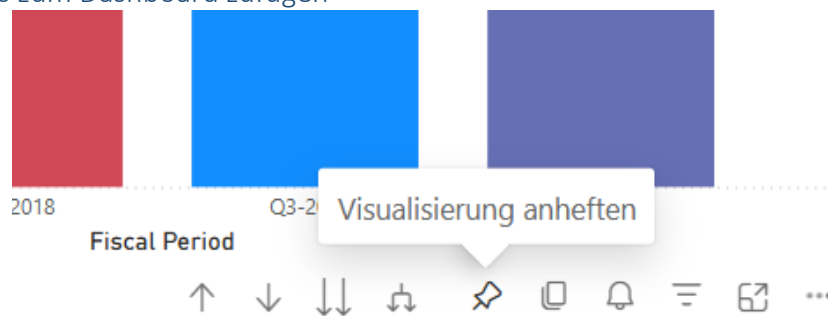


Abbildung 158

### 6.4.2 Bericht zum Dashboard hinzufügen

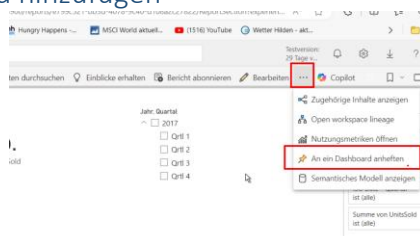


Abbildung 159

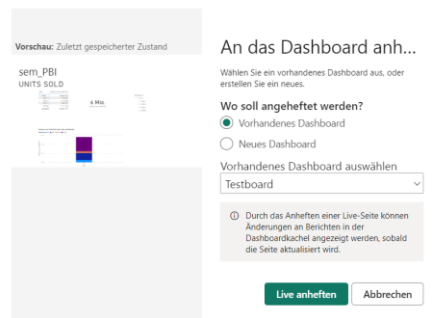


Abbildung 160

### 6.4.3 Seite entfernen

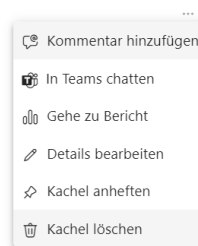


Abbildung 161

### 6.4.4 Design anpassen

Das Dashboarddesign kann unter folgenden Punkt an das CD angepasst, gespeichert und wieder abgerufen werden:

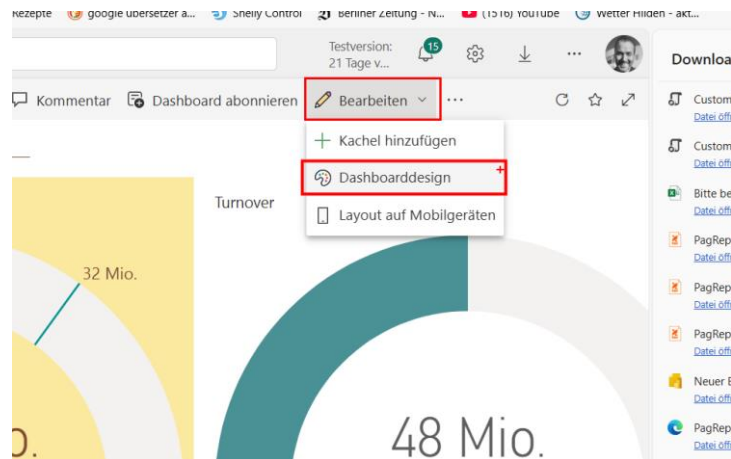


Abbildung 162

### 6.4.5 Alert (Datenwarnung) hinzufügen

Verfügbar für die Visuals: Karte, Tacho, KPI

Vorgehen:

- 1) Im Power BI Service den gewünschten Report öffnen und das zu überwachende Visual (Kachel) anpinnen:

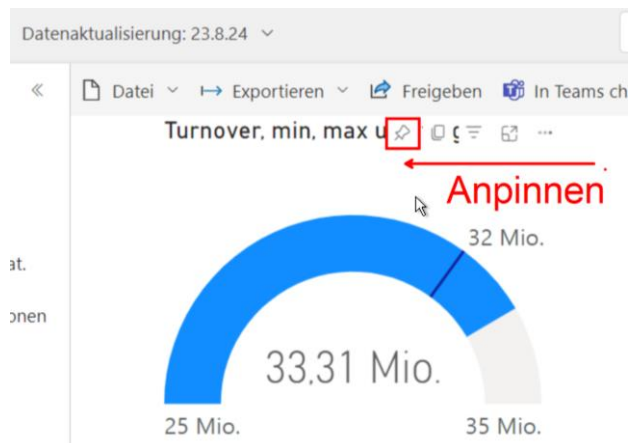


Abbildung 163

- zum korrespondierenden Dashboard wechseln und für die hinzugefügte Kachel die Warnungen aktivieren:

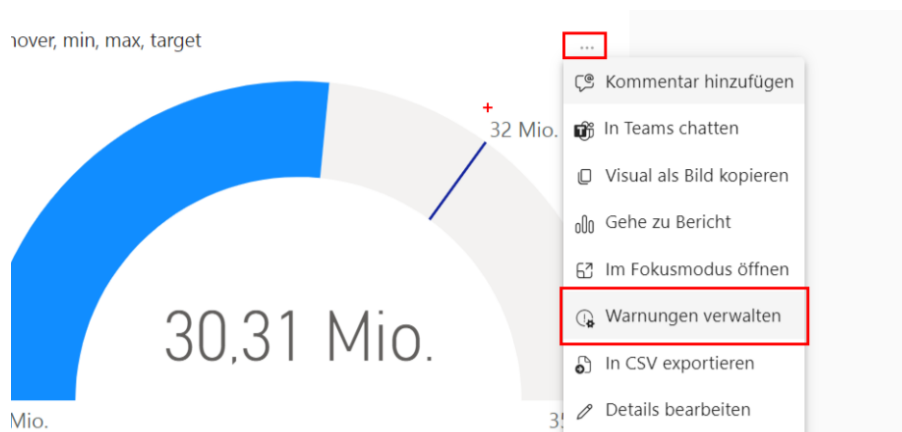


Abbildung 164

- Zum Schluss den Schwellwert und Art der Benachrichtigung einstellen

Datenwarnungen verwalten:

Warnungsname	Status	Quelle/Element
ne von OrderQuantity	Ein	Test
Turnover	Ein	SandboxDash
Turnover, min, max, target	Ein	SandboxDash
Turnover	Ein	Testboard

Abbildung 165

6.4.6 Berichtsarten

Siehe: [Einführung - Training | Microsoft Learn](#)

6.4.7 Paginated Reports

Paginierte Berichte dienen der druckoptimierten Ausgabe langer Listendaten  
 PBI Service:

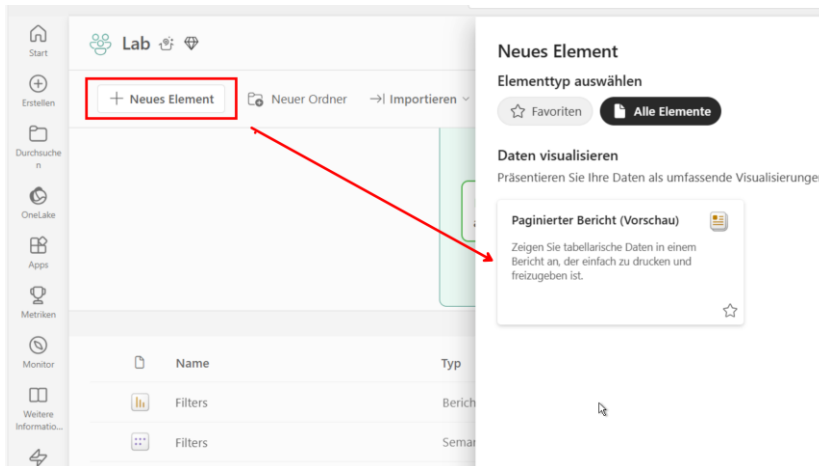


Abbildung 166

Dann das gewünschte Dataset auswählen und Bericht erstellen.

Gewünschte Datenfelder und Measures zufügen und in einem Arbeitsbereich speichern.

Optional: PowerBI Report Builder herunterladen, dort den Bericht öffnen und weiterführend bearbeiten und danach veröffentlichen.

6.4.8 Icons mit Navigation und Mouseover

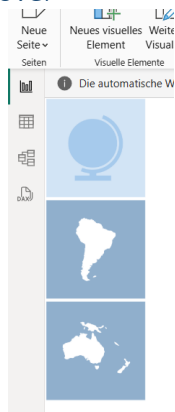


Abbildung 167

6.4.9 Datenanalysen

6.4.9.1 Histogramm (Verteilung) erstellen

In welcher Produktpreisgruppe, wurde welcher Umsatz gemacht?

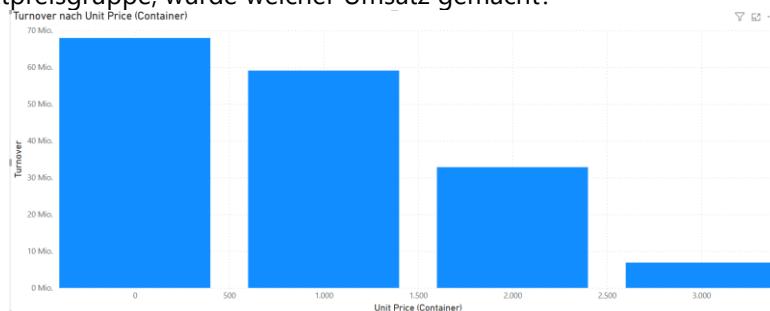


Abbildung 168

Zunächst für die X-Achse eine neue Gruppe erstellen:

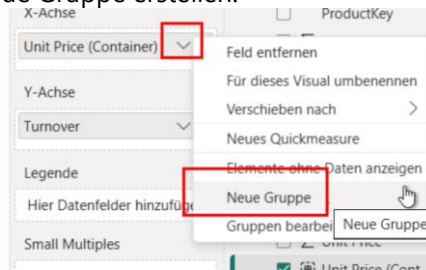


Abbildung 169

Dann Unterteilung festlegen:

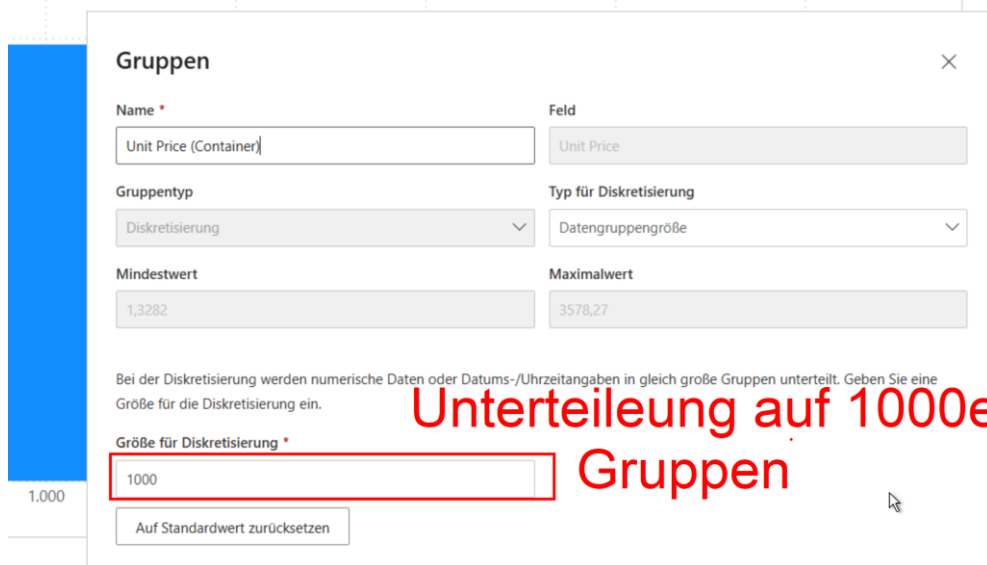


Abbildung 170

Hinweis: besteht die X-Achse aus Texten und nicht aus Zahlen, können individuelle Gruppen erstellt werden, z.B. nach bestimmten Ländern:

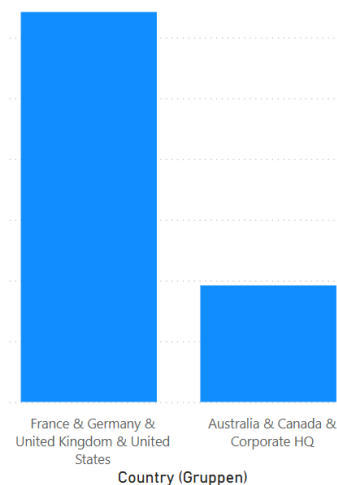


Abbildung 171

6.4.9.2 Q&A Visual

Dieses kann immer per Doppelklick eingefügt werden und erlaubt Fragen im Klartext (englisch) z.B.:

- top 3 product by turnover
- turnover by color

6.4.9.3 Einflussfaktoren für Anstieg und Rückgang



Abbildung 172

Über einen Rechtsklick auf die Daten kann eine Analyse der Einflussfaktoren für einen Anstieg oder Rückgang erfolgen.

Aus den angezeigten Vorschlägen können die jeweiligen Visuals dem Bericht zugefügt werden:

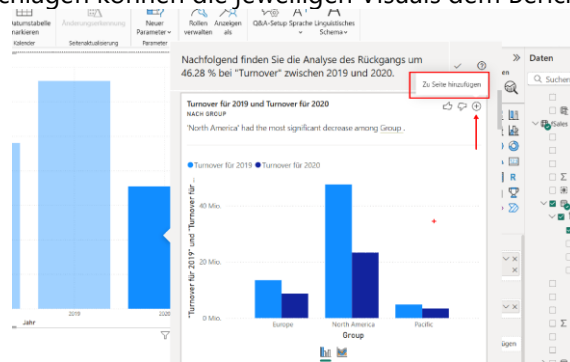


Abbildung 173

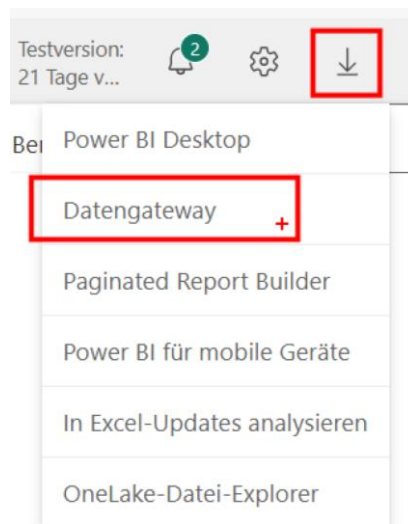
6.5 Power BI Service

6.5.1 On Prem Data Gateway

Download Data Gateway für **Standard-Mode**

**Personal Mode:** nur für eigene Ressourcen/Importierte Daten

**Standard Mode:** Weitere Quellen in der lokalen Infrastruktur (z.B. SQL-Server, PowerApps) und unterschiedliche User die das Gateway nutzen können



Get your data from on-premises data sources

Abbildung 174

Anmelden mit M365 Mail (Die gleiche wie den PBI-Login online)  
 GatewayNamen definieren. (Testgateway)  
 Alle weiteren Einstellungen erfolgen auf PowerBI.com, Gateway prüfen:

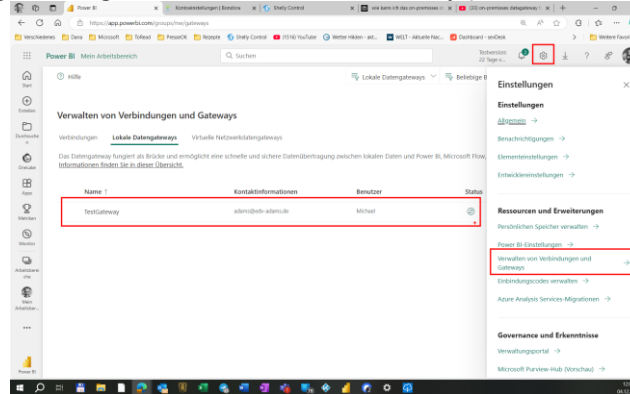


Abbildung 175

Einstellungen des Datasets:

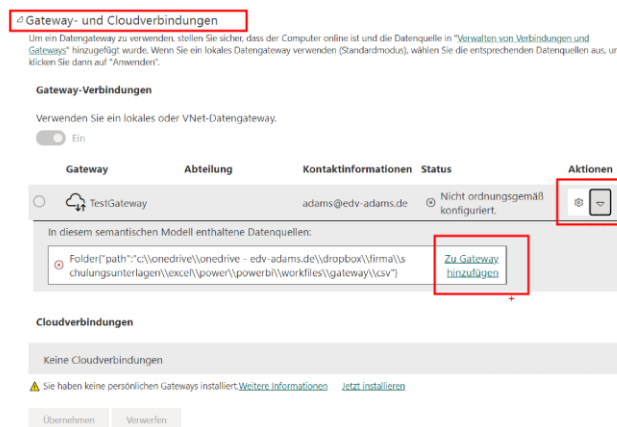


Abbildung 176

Verbindungseinstellungen definieren:

### Neue Verbindung ✕

**Gatewayclustername \***  
TestGateway ▼

**Verbindungsname \***  
TestVerbindung

**Verbindungstyp \***  
Ordner ▼

**Vollständiger Pfad \***  
C:\OneDrive\OneDrive - edv-adams.de\Dropbox\Firma\Sc...

**Authentifizierung** ⓘ

**Authentifizierungsmethode \***  
Windows ▼

**Windows-Benutzername \***  
hpbook\michael

**Windows-Kennwort \***  
.....

Testverbindung überspringen

**Allgemein**

**Datenschutzstufe \***  
Organisation ▼

Erstellen
Schließen

Abbildung 177

Verbindung dem Dataset zuordnen:

**Gateway-Verbindungen**

Verwenden Sie ein lokales oder VNet-Datengateway. ☑ Ein

Gateway	Abteilung	Kontaktinformationen	Status	Aktionen
TestGateway		adams@edv-adams.de	☑ Wird auf "HPBOOK" ausgeführt.	⚙️ ▼

In diesem semantischen Modell enthaltene Datenquellen:

Folder("path": "c:\onedrive\onedrive - edv-adams.de\dro

✔ pbox\firma\schulungsunterlagen\excel\power\powerb  
 \workfiles\gateway\csv")

Zuordnung:

TestVerbindung ▼

TestVerbindung

Zu Gateway hinzufügen

Abbildung 178

Nun können Datasets aktualisiert werden oder auf geplante Aktualisierungen eingestellt werden.

### 6.5.2 Rechte für Arbeitsbereiche

#### [Beschreibung der Rollen](#)

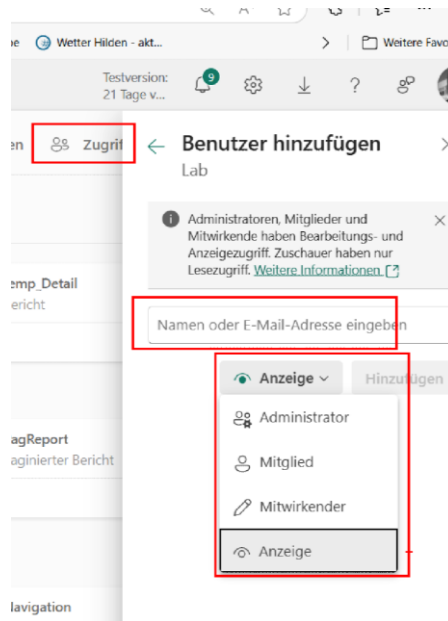


Abbildung 179

6.5.2.1 Apps erstellen

Eine App ist eine Zusammenstellung verschiedener Power BI Artefakte (z.B. Berichte) und dient der großflächigen Verteilung der Daten. Z.B. wird sie im Bereich APPS zum „Download“ angeboten – Download bedeutet Einbindung in die eigene App-Liste  
 Eine neue App wird in einem Arbeitsbereich erstellt:

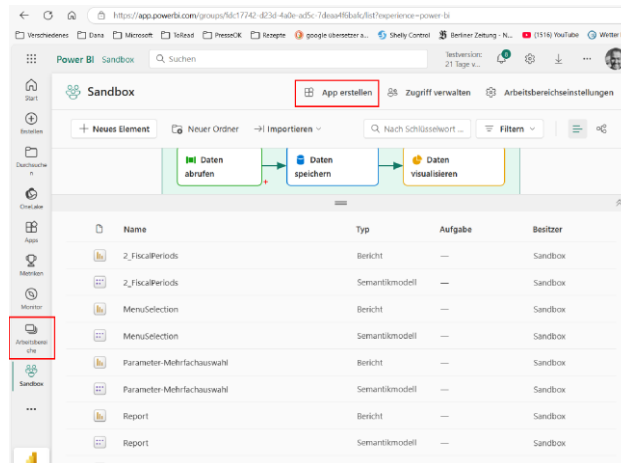


Abbildung 180

App empfehlen oder entfernen:

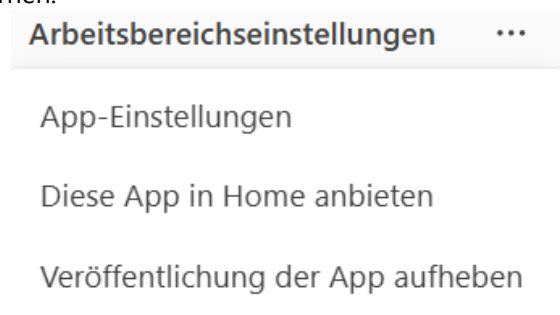


Abbildung 181

Eine App die jmd. Anders erstellt hat abrufen: APPS/Abrufen

Alternativ wird die App im Home-Bereich angeboten, sofern der Autor sie für den Home-Bereich angeboten hat, siehe Abbildung 181

### 6.5.3 Quick Insights

Automatische Erkenntnisse und Hinweise zum Semantikmodell:

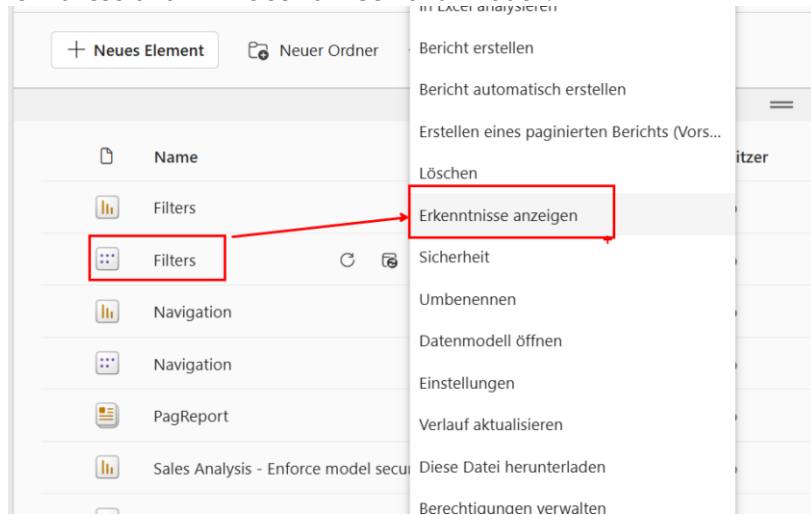


Abbildung 182

**Tipp:** die hier gewonnenen Erkenntnisse können in einem Dashboard zusammengefasst werden.

## 7 Powerquery

### 7.1 Einzelne Dateien importieren

Ordner: Erster Import

Die \*.txt Datei wird importiert:

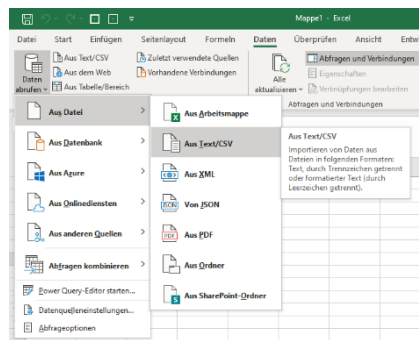


Abbildung 183

Änderung der Datenquelle in Powerquery:

- Start/Datenquelleneinstellungen

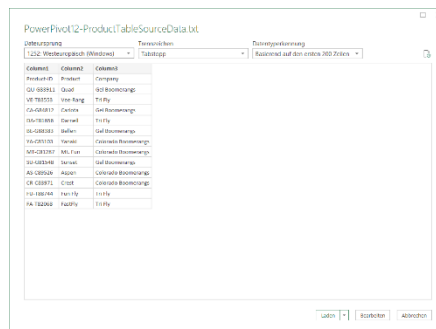
### 7.2 Dateien aus Ordner importieren

Ordner: Multi CSV

#### 7.2.1 Daten importieren

Externe Daten abrufen z.B.

- Aus Datenbank (z.B. SQL, Access)
- Aus Datei (Oder Aus Ordner!)
- JSON/XML



ProduktID	Produkt	Comping
001-00001	Blau	IG-Boomerang
001-00002	Blau	IG-Boomerang
001-00003	Blau	IG-Boomerang
001-00004	Blau	IG-Boomerang
001-00005	Blau	IG-Boomerang
001-00006	Blau	IG-Boomerang
001-00007	Blau	IG-Boomerang
001-00008	Blau	IG-Boomerang
001-00009	Blau	IG-Boomerang
001-00010	Blau	IG-Boomerang
001-00011	Blau	IG-Boomerang
001-00012	Blau	IG-Boomerang
001-00013	Blau	IG-Boomerang
001-00014	Blau	IG-Boomerang
001-00015	Blau	IG-Boomerang
001-00016	Blau	IG-Boomerang
001-00017	Blau	IG-Boomerang
001-00018	Blau	IG-Boomerang
001-00019	Blau	IG-Boomerang
001-00020	Blau	IG-Boomerang
001-00021	Blau	IG-Boomerang
001-00022	Blau	IG-Boomerang
001-00023	Blau	IG-Boomerang
001-00024	Blau	IG-Boomerang
001-00025	Blau	IG-Boomerang
001-00026	Blau	IG-Boomerang
001-00027	Blau	IG-Boomerang
001-00028	Blau	IG-Boomerang
001-00029	Blau	IG-Boomerang
001-00030	Blau	IG-Boomerang
001-00031	Blau	IG-Boomerang
001-00032	Blau	IG-Boomerang
001-00033	Blau	IG-Boomerang
001-00034	Blau	IG-Boomerang
001-00035	Blau	IG-Boomerang
001-00036	Blau	IG-Boomerang
001-00037	Blau	IG-Boomerang
001-00038	Blau	IG-Boomerang
001-00039	Blau	IG-Boomerang
001-00040	Blau	IG-Boomerang
001-00041	Blau	IG-Boomerang
001-00042	Blau	IG-Boomerang
001-00043	Blau	IG-Boomerang
001-00044	Blau	IG-Boomerang
001-00045	Blau	IG-Boomerang
001-00046	Blau	IG-Boomerang
001-00047	Blau	IG-Boomerang
001-00048	Blau	IG-Boomerang
001-00049	Blau	IG-Boomerang
001-00050	Blau	IG-Boomerang

Abbildung 184

Laden

- Als Verbindung
- Direkt in Excel

Hinweis: Die Checkbox *zum Datenmodell hinzufügen* macht die Datenquelle in PowerPivot verfügbar  
TIPP: Bei Textfiles, oben links: Erste Zeile als Überschriften verwenden.

#### 7.2.2 Automatische Datentyperkennung abschalten

Da die automatische Datentyperkennung von Powerquery häufig Daten falsch erkennt, kann es Sinn machen sie über *Datei/Abfrageoptionen* zu deaktivieren. Wählen Sie dazu die Option *Spaltentypen niemals erkennen*:



Abbildung 185

### 7.2.3 Mehrere Datenquellen zusammenführen

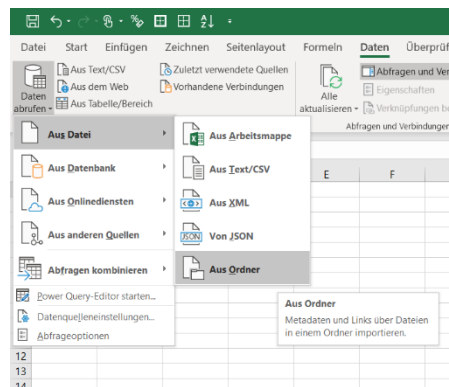


Abbildung 186

Über die Schaltfläche *Daten Kombinieren* können jetzt alle Datenquellen zusammengeführt werden:

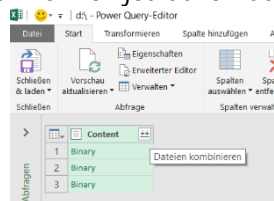


Abbildung 187

### 7.2.4 Datumstyp anpassen

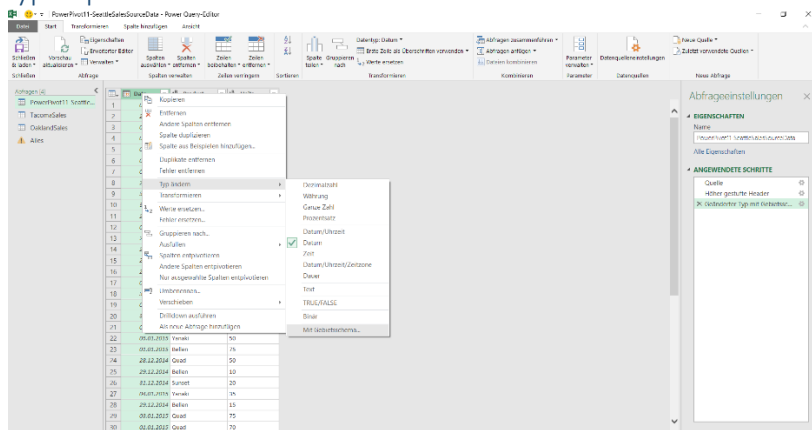


Abbildung 188

Diese Aktion wird als Konvertierungsschritt auf der rechten Seite gespeichert und stets auf die Datenquelle angewendet.

### 7.3 Angewendete Schritte kopieren

- 1) Rechtsklick auf Quellabfrage/Erweiterter Editor
- 2) Skript kopieren und Abbrechen
- 3) Rechtsklick auf Zielabfrage/Erweiterter Editor
- 4) Skript einfügen und ggf. Dateipfad anpassen

### 7.4 Dateien aus einem Sharepointordner importieren

Ordner: <https://edvadamsde.sharepoint.com/sites/Teamsite> ([Teamsite – Home \(sharepoint.com\)](https://edvadamsde.sharepoint.com/sites/Teamsite))  
 Zunächst den Link zur gewünschten Teamsite (Quelle) kopieren:

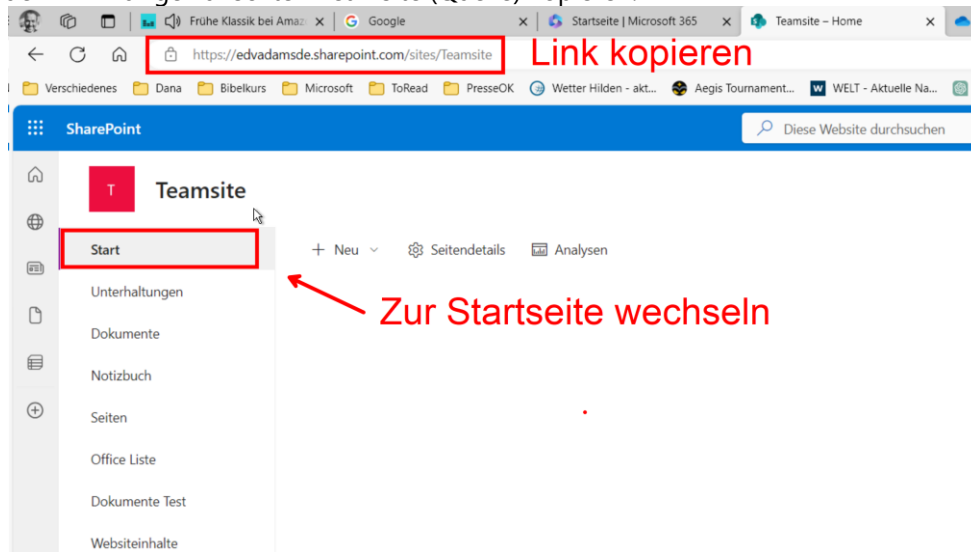


Abbildung 189

Danach Import starten:

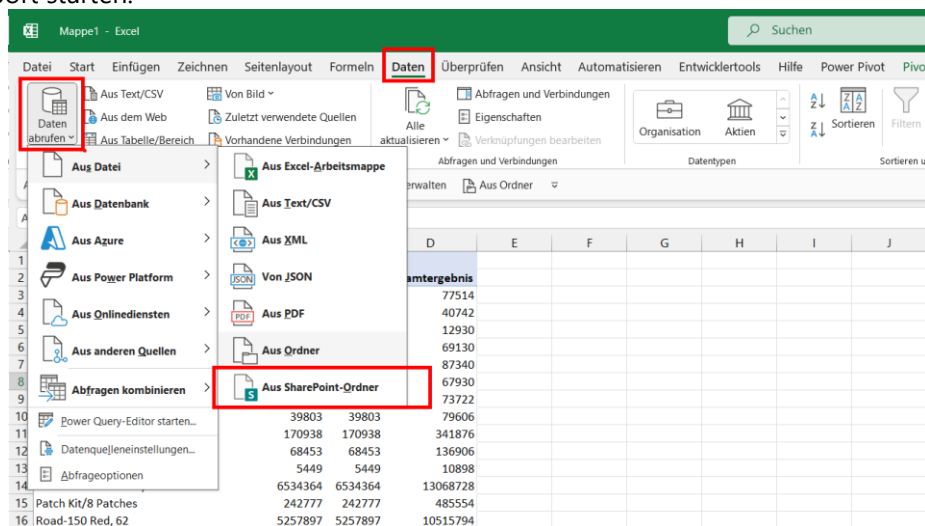


Abbildung 190

Und im nachfolgenden Dialog den Link einfügen.

### 7.5 Datenquellen verknüpfen (Ersatz für SVerweis)

Ordner: Datenquellen verknüpfen (Link Data Sources)

- Excel Tabelle importieren

- Textfile importieren
- Gemeinsame Kriterien-ID angleichen (Siehe nächster Abschnitt: 7.5.1)

TIPP:

Für Textfiles kann die erste Zeile sehr einfach als Überschrift übernommen werden:

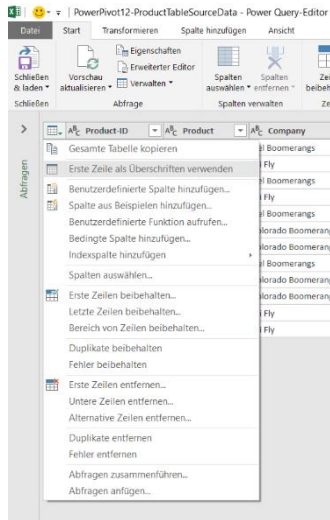


Abbildung 191

### 7.5.1 Dateninhalte auftrennen

- Daten Bearbeiten (PowerQuery)
- Transformieren/Spalte Teilen

### 7.5.2 Datentabellen verknüpfen

**In Excel:** Über den Befehl *Zusammenführen* werde die entsprechenden Abfragen ausgewählt.

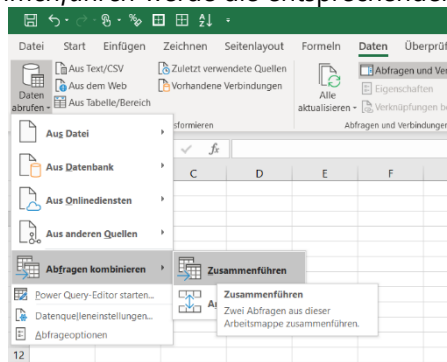


Abbildung 192

**In Powerquery:** Hier verwenden Sie *Start/Kombinieren/Abfragen zusammenführen*

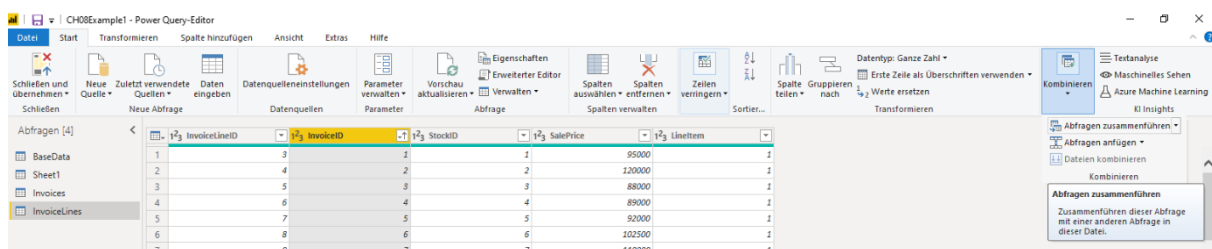


Abbildung 193

Danach wählen Sie die zu verknüpfenden Spalten aus:

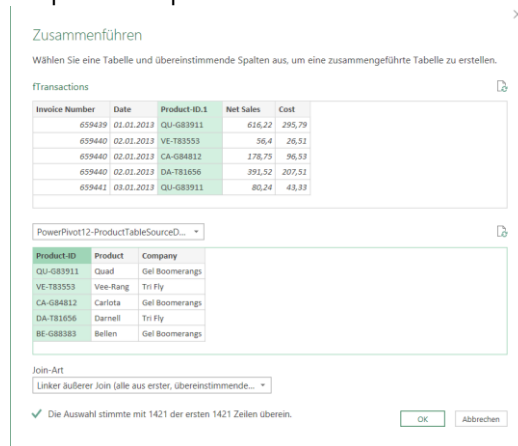


Abbildung 194

Danach erfolgt die Auswahl, wie die Daten weiterverarbeitet werden sollen:

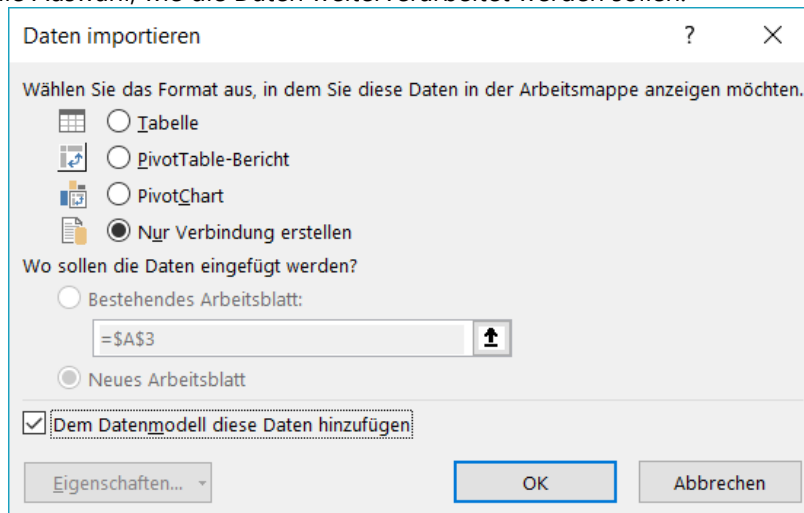


Abbildung 195

Hinweis: Werden über die Checkbox die Daten dem Datenmodell hinzugefügt, steht diese Datenquelle zur weiteren Verarbeitung in Powerpivot zur Verfügung.

Im Spaltenkopf kann jetzt ausgewählt werden, welche Spalten in der Abfrage angezeigt werden:

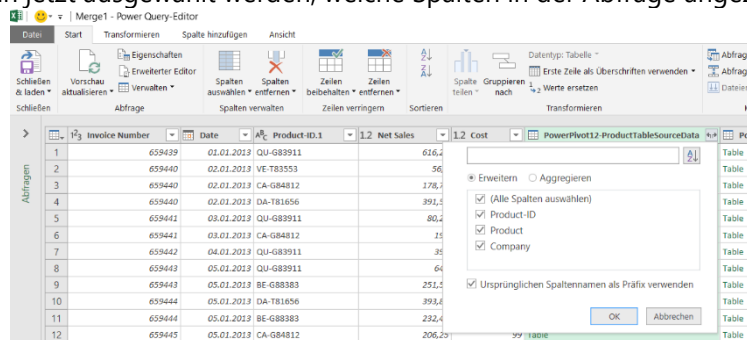


Abbildung 196

### 7.5.3 Pivot-Tabelle anlegen

Über rechte Maustaste auf die Datenquelle, *Laden in* kann aus den Daten z.B. einen neue Pivot-Tabelle angelegt werden.

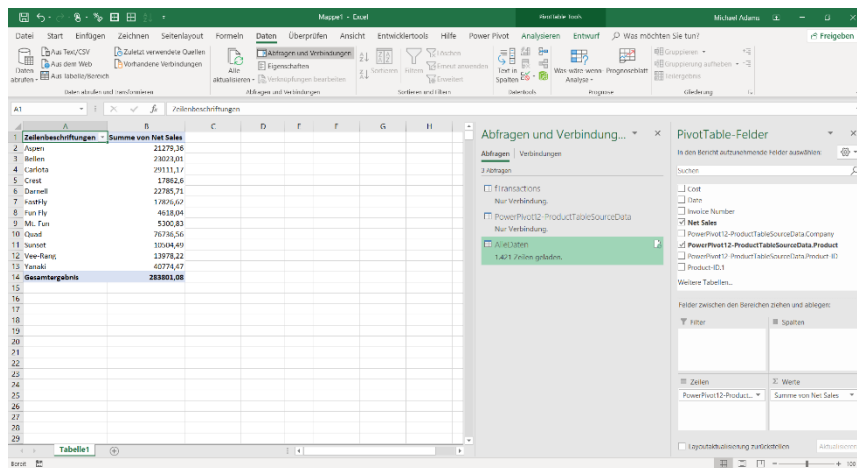


Abbildung 197

### 7.6 Unterschiedliche Tabellen zusammenfügen

Ordner: Append

- Neue Arbeitsmappe anlegen
- Alle drei Tabellen laden und über *Laden in* lediglich Verbindung herstellen

Da die Ausgangsabfragen unterschiedliche Spalten haben, müssen diese vereinheitlicht werden:

- In der Tabelle *Sales2017* müssen *Units*, *NetCostEquivalent* und *Price* multipliziert werden. Dazu markieren Sie die zu multiplizierenden Spalten und wählen den Befehl *Spalte hinzufügen/Multiplizieren*:

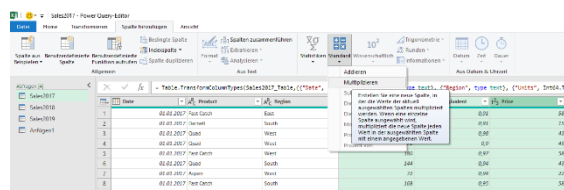


Abbildung 198

- Die Ergebnisspalte benennen wir in *Revenue* um, damit die Spaltennamen über alle Tabellen identisch sind.
- Jetzt können *Units*, *NetCostEquivalent* und *Price* entfernt werden.
- In der Tabelle *Sales2019* muss die überschüssige Spalte *Color* entfernt werden
- Befehl *Anfügen* wählen und alle drei Abfragen auswählen:

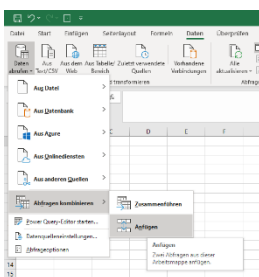


Abbildung 199

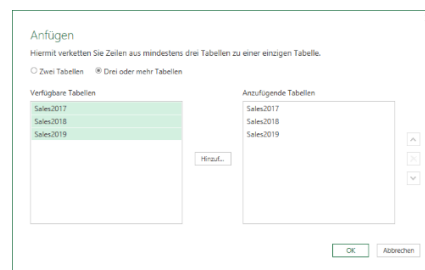


Abbildung 200

**Hinweis/Tipp:** Die Spalten werden nach dem **Spaltennamen** und nicht nach Inhalt angefügt! Möchte man unterschiedliche Spalten in einer Spalte zusammenbringen, benennt man sie einfach gleich.

## 7.7 Spaltenoperationen

### 7.7.1 Duplikate entfernen

Ordner: Duplikate

Zunächst wird der gesamte Ordner importiert und zu einer Liste kombiniert.

Danach die Spalten markieren, die als Kriterium für Duplikate fungieren und über Rechtsklick den Befehl *Duplikate entfernen* wählen.

### 7.7.2 Duplizieren und Rechnen mit Datum

Ordner: CSV-Multiimport

Um z.B. aus Datum bestimmte Informationen zu berechnen, empfiehlt es sich zunächst eine Kopie der Datumsspalte zu erstellen und dann die Berechnung anzuwenden:

- Spalte hinzufügen/Spalte duplizieren
- In der Kopie: Transformieren/Datum z.B.:
  - Datum/Alter
  - Danach: Dauer/Jahre

#### 7.7.2.1 Datumsdifferenz ermitteln:

Duration.Days([Ende]-[Start])

### 7.7.3 Lücken auffüllen

Datei: Misc/CarMakeAndModelMatrix.xlsx und HierarchieSAP.txt

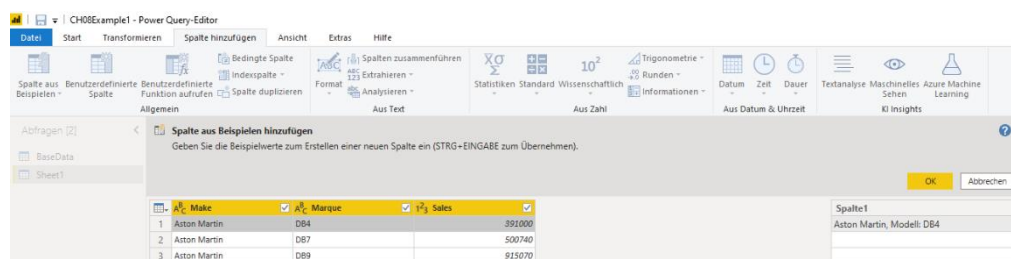
Um Lücken in Zeilendaten aufzufüllen, um z.B. eine eindeutige Zuordnung zu restlichen Daten zu ermöglichen, können leere Zellen über *Transformieren/Ausfüllen* mit zu wiederholenden Werten aufgefüllt werden.

### 7.7.4 Spalten aus Beispielen

Datei: Misc/CarMakeAndModelMatrix.xlsx

Diese Funktion ermöglicht aus vorgegebenen Daten eine Verkettung zu erzeugen, ohne Funktionen und Befehle eingeben zu müssen:

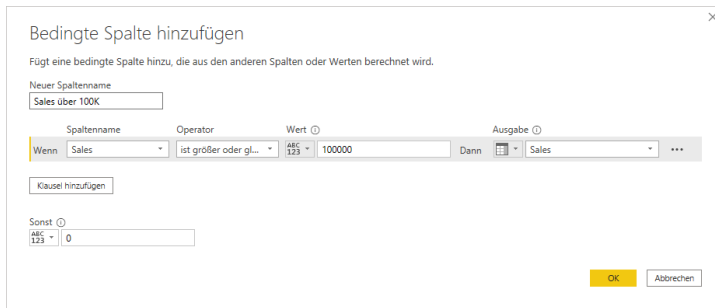
- *Spalte Hinzufügen/Spalte aus Beispielen*
- Hier gewünschte Beispieldaten eingeben:



### 7.7.5 Bedingte Spalte

Datei: Misc/CarMakeAndModelMatrix.xlsx

Um mittels einer Wenn-Dann Logik eine neue Spalte zu erzeugen, wählen Sie den Befehl *Spalte hinzufügen/bedingte Spalte* und stellen dann die gewünschte(n) Bedingung(en) ein:



## 7.8 Pivot Transformationen

### 7.8.1 Daten depivotieren

Ordner: *Entpivotieren*

- Start/Zeilen entfernen/Oberste (2) Zeilen
- Start/Erste Zeile als Überschriften
- Leerzeilen (*Null*) wegfiltern
- Spaltenüberschriften löschen
- Monatsspalten markieren und Transformieren/Entpivotieren:

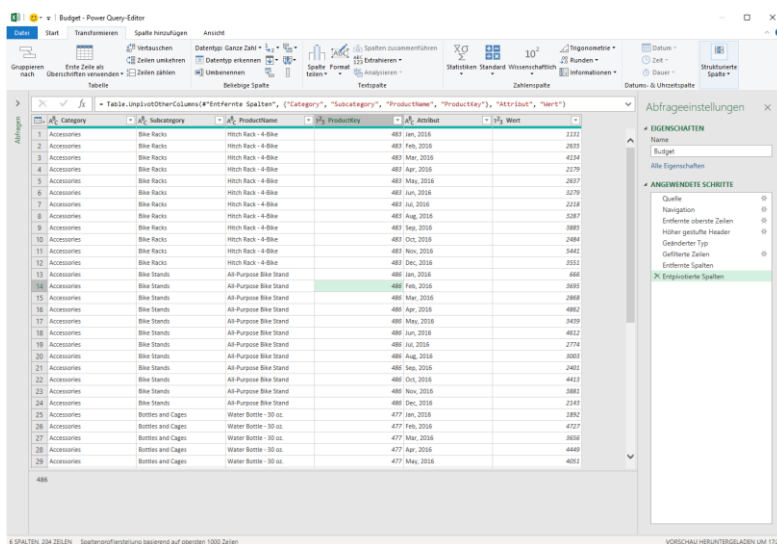


Abbildung 201

### 7.8.2 Daten pivotieren

Ordner: *Pivotieren*

Zunächst klicken Sie in die Spalte die pivotiert werden soll – in unserem Beispiel Regionen - und wählen *Transformieren/Pivotieren*. Nachfolgend wählen Sie die Zahlenspalte die aggregiert werden soll, hier *Umsatz*:

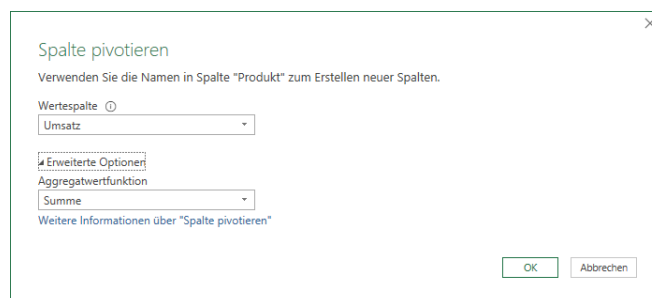


Abbildung 202

Ergebnis:

Produkt	1.2 West	1.2 Süd	1.2 Nord	1.2 Ost
1 Außervielzahn	7767	13723,2	2118,03	null
2 Inbus	22648,12	1434,65	5201,83	null
3 Innensechskant	12042,56	19386,8	16295,65	7264,24
4 Innensechskant-TR	5910,87	6410,88	801,53	1108,46
5 Innenvielzahn	13810,58	2459,5	3847,07	null
6 Phillips-Kreuzschlitz	785,52	null	13382,19	10012,69
7 Pozidriv-Kreuzschlitz	14272,17	2655,1	8082,3	5410,96
8 Rändelkopf	9630,17	9007,17	null	3503,89
9 Schlitz	8210,62	10921,05	15889,13	null
10 Sechskant	6314,17	10881,86	24676,85	11374,19
11 TorxSet	4504,58	4341,72	12333,07	9393,69
12 Torx	5028,85	11654,29	null	9004
13 Torx-TR	12844,36	null	20626,51	9069,81
14 Tri-Wing	12928,52	1786,76	3963,07	6500,52
15 Vierkant	10012,31	6116,29	null	null

Abbildung 203

**Tipp:** Transformieren/Vertauschen ändert die Spalten-/Zeilenanordnung und entspricht dem Transponieren.

### 7.9 Datenqualität anzeigen

Ordner: Pivot-Rohdaten

Verschiedene Einstellungen zu Ansichten und Anzeige der Datenqualität befinden sich auf dem Tab Ansicht:

- Bearbeitungsleiste zum Bearbeiten von Ausdrücken
- Spaltenqualität zeigt fehlerhafte oder leere Daten
- Spaltenverteilung zeigt die Datenverteilung
- Spaltenprofil zeigt im unteren Bildschirmbereich Details zur markierten Spalte

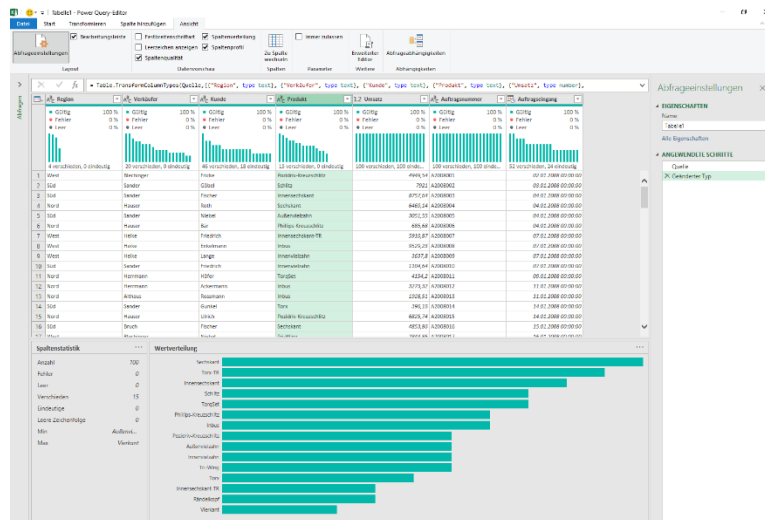


Abbildung 204

#### 7.9.1 Daten bereinigen

Wenn Sie mit der Maus auf das Spaltenprofil oben zeigen, können grundsätzliche Bereinigungen durchgeführt werden:

- Duplikate entfernen
- Leere entfernen
- Fehler entfernen

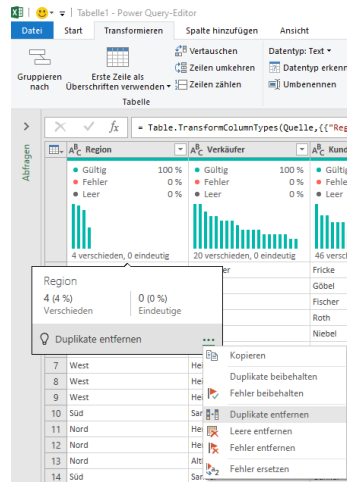


Abbildung 205

## 7.10 Datenimporte

### 7.10.1 Mappen und Tabellen Multiimport

Ordner: Mappen und Tabellen Multiimport\Data

Hier werden mehrere Arbeitsmappen und Tabellenblätter in eine Excel-Tabelle konsolidiert.

Wählen Sie zunächst den Ordner in dem sich die zu importierenden Dateien befinden und laden diesen als Verbindung in PowerQuery. (Siehe auch Kap. 7.10)

Entfernen Sie anschließend alle anderen Spalten, außer der ersten:

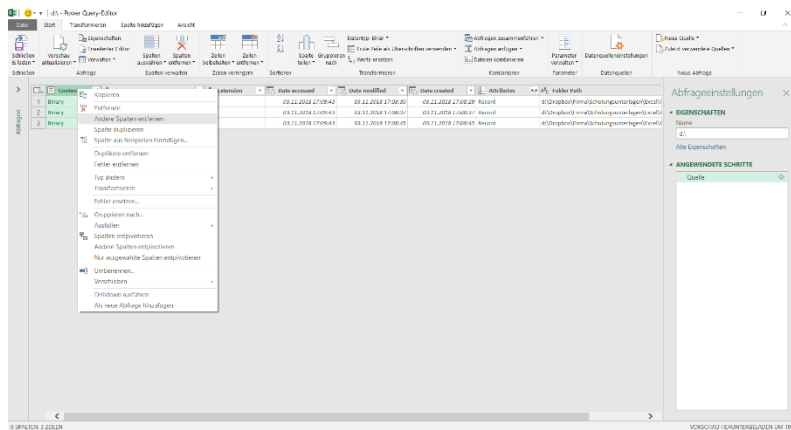


Abbildung 206

Neue Spalte *ExcelData* hinzufügen:

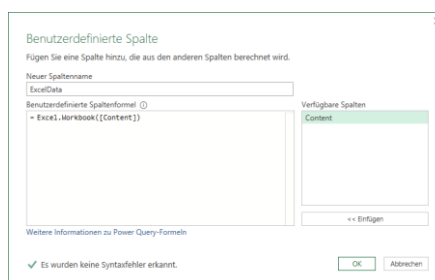


Abbildung 207

Spalte mit Inhalt *Binary* entfernen und Inhalte expandieren. Das Ergebnis:

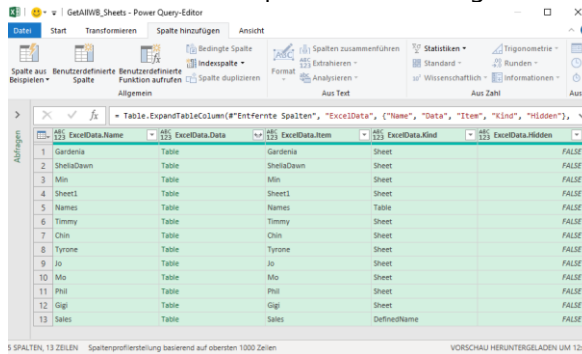


Abbildung 208

Dann:

- *Kind*: filtern nach *Sheet*
- *Item*: <>Sheet
- *Kind*, *Name*, *Hidden* entfernen

Klick auf ein *Table*-Eintrag zeigt, dass die Spaltenüberschriften nicht korrekt sind. Korrektur:

- Benutzerdef. Spalte namens *PromoteHeaders* zufügen:  
Table.PromoteHeaders ([ExcelData.Data])
- *PromoteHeaders* expandieren

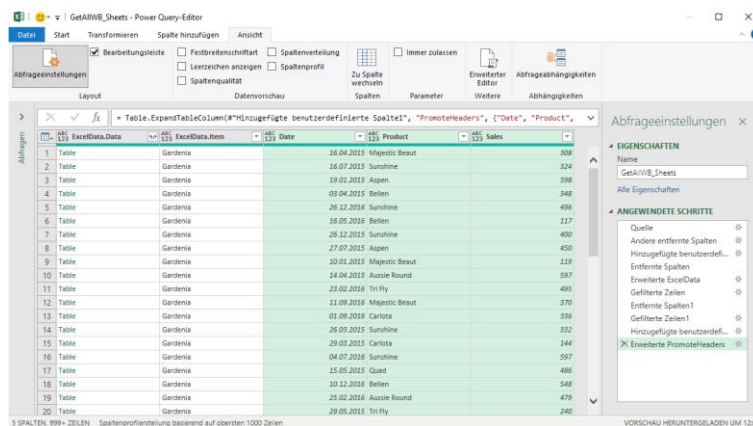


Abbildung 209

*ExcelData.Item* in z.B. SalesRep umbenennen.

**Hinweis:** Falls das Datum in einer Pivot-Tabelle gruppiert werden soll, muss in Powerquery der Typ auf *Datum* gesetzt werden.

### 7.10.2 Importieren und parsen von XML und JSON-Daten

JSON: JavaScript object notation

XML: Extensible markup language

#### 7.10.2.1 Daten aus Spalte extrahieren

Ordner: XML\_JSON

Zunächst bitte Verbindung zu *XMLInColumn.xlsx* herstellen. Dann Spalte mit XML-Daten markieren (AvailableColors). Danach *Spalte hinzufügen/Analysieren/XML*

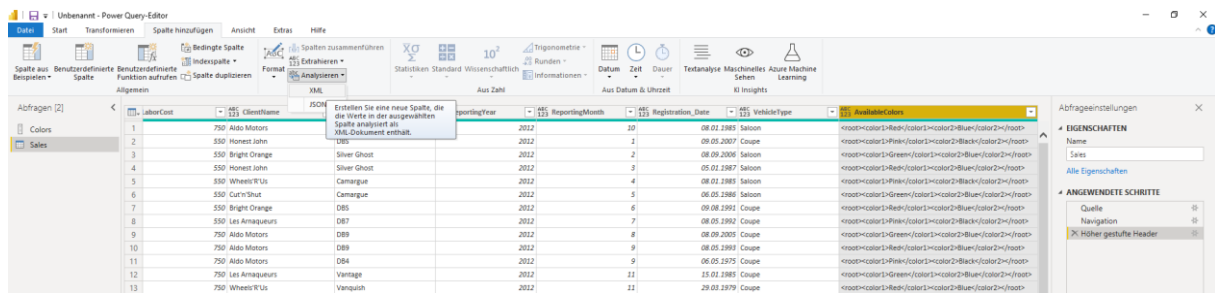


Abbildung 210

7.10.2.2 Komplexe JSON-Files laden und parsen

Ordner: XML\_JSON

- Zunächst bitte Verbindung zu CarSales/JSON\_Complex.Json herstellen.
- Danach die Liste in Tabelle umwandeln
- Die Liste mit dem Befehl ‚auf neue Zeilen ausweiten‘ expandieren (Siehe Abbildung 211)
- Zum Schluss nach und nach die gewünschten Records expandieren

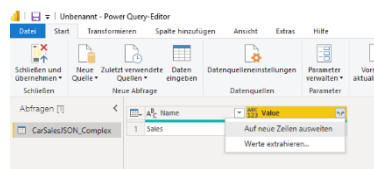


Abbildung 211

**Unterschied Liste und Tabelle:** Eine Liste enthält zeilenweise Daten die nicht auf Spaltenebene ausgewertet werden können. Eine Tabelle beinhaltet Spalten, so dass es häufig Sinn macht eine Liste in eine Tabelle zu konvertieren.

7.11 SQL-Abfragen

Über Daten abrufen/Aus Datenbank/Aus SQL kann eine Verbindung zum SQL-Server initiiert werden:

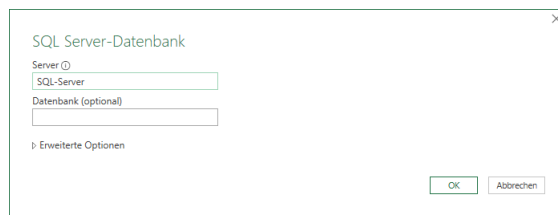


Abbildung 212

Hierbei kann es sich um einen lokalen, oder einen aus dem Web erreichbaren Server handeln.

Sofern der Server erreichbar ist, werden Sie nach Ihren Log-in Daten gefragt...

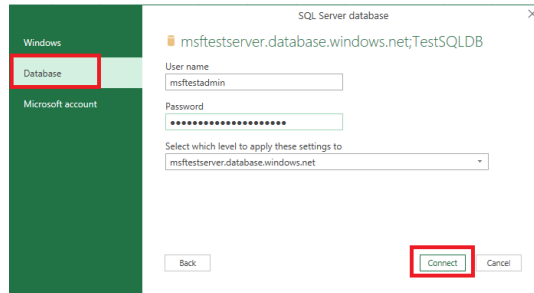


Abbildung 213

...und die Verbindung zu den für Sie verfügbaren Datenbanken wird aufgebaut:

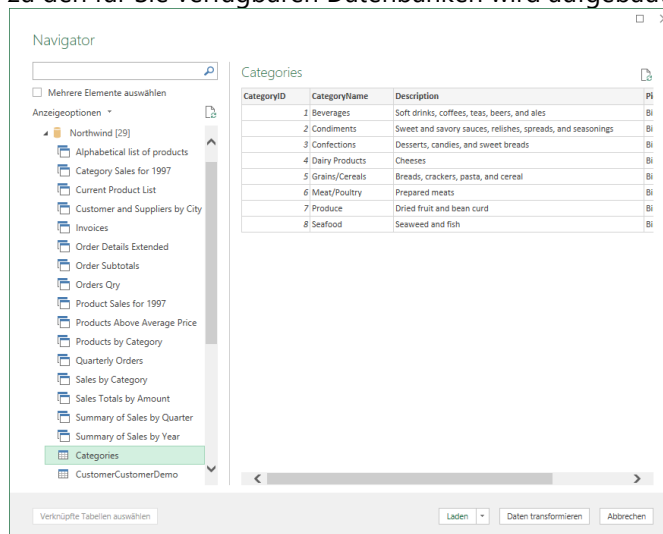


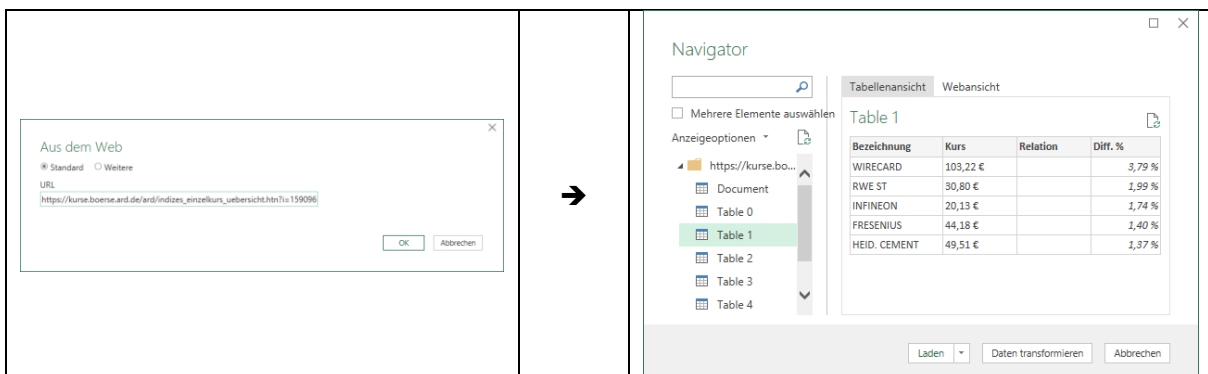
Abbildung 214

### 7.12 Web-Abfragen

z.B.:

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

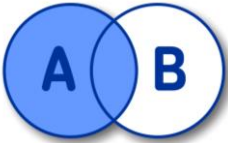
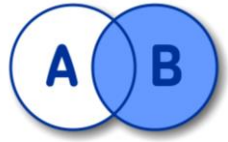
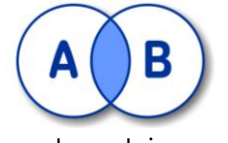
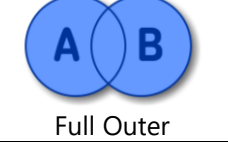
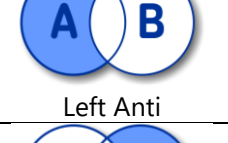

Über den Befehl *Daten abrufen/aus anderen Quellen/aus dem Web* können Daten aus Internetseiten abgerufen werden. Dazu braucht lediglich die Webadresse eingegeben zu werden und Power Query zeigt alle unter dieser Adresse gefunden Quellen an:



### 7.13 Join-Arten

Ordner: *Join-Typen*

Bei der Verknüpfung von Tabellen können verschiedene Join-Arten verwendet werden:

Art	Effekt
 Left Outer	Alle Datensätze aus linker Tabelle und übereinstimmende aus der rechten. (SVerweis)
 Right Outer	Alle Datensätze aus rechter Tabelle
 Inner Join	Zeigt nur Datensätze an, die in beiden Tabellen vorkommen (AND)
 Full Outer	Zeigt alle Datensätze, egal ob es eine Entsprechung in der anderen Tabelle gibt
 Left Anti	In Tabelle 1 JA, in Tabelle 2 NEIN
 Right Anti	In Tabelle 1 NEIN, in Tabelle 2 JA

**Hinweis:** Über die *FUZZY-Option* lässt sich einstellen wie genau die Verknüpfungsfelder übereinstimmen müssen

### 7.13.1 Beispiele für Joins

#### 7.13.1.1 Left Outer einfach

Zu jedem Umsatz aus der linken Tabelle soll der Produktpreis aus der rechten Tabelle bezogen werden. Dies entspricht der Funktion SVerweis() oder XVerweis().

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		

Abbildung 215

7.13.1.2 Left Outer mehrfach

Wenn mehrere Schlüsselkriterien gleichzeitig zum Einsatz kommen sollen (Abbildung 217), z.B. Produkt **und** Farbe, können bei Erstellung der Verknüpfung in Powerquery bei gedrückter STRG Taste mehrere Spalten markiert werden:

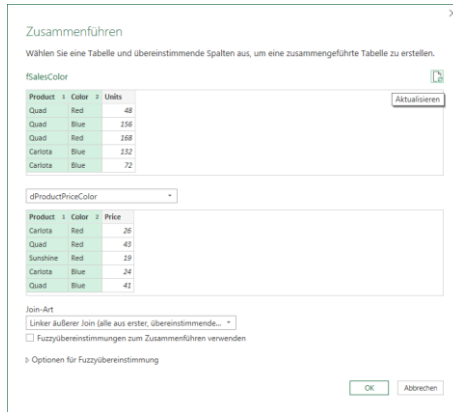


Abbildung 216

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			

Abbildung 217

7.13.1.3 Left Outer mit Mehrfachtreffer in Zieltabelle

Mehrfachtreffer können in einer Zieltabelle per SVerweis nicht nachgeschlagen werden:

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

Abbildung 218

Über einen Join in Powerquery können auch Mehrfachtreffer ausgewertet werden:

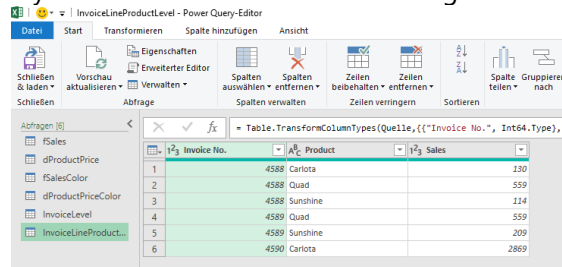


Abbildung 219

### 7.13.2 Inner Join

Welche Teilnehmer waren in beiden Städten?

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		Beulah Wenger
Reyna Luke		Malvina Hamer
Beulah Wenger		Bernita Crutcher
Malvina Hamer		Shiela Anaya
Vivan Keeney		Yolonda Armstead

Abbildung 220

### 7.13.3 Left Anti Join

Welche Teilnehmer waren nur 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		Beulah Wenger
Reyna Luke		Malvina Hamer
Beulah Wenger		Bernita Crutcher
Malvina Hamer		Shiela Anaya
Vivan Keeney		Yolonda Armstead

Abbildung 221

## 7.14 Gruppierungen

Ordner: *Gruppieren*, Datei, PQ -GroupBy:

- Tabelle in PowerQuery übernehmen über *Daten/Aus Tabelle Bereich*
- PowerQuery: *Transformieren/Gruppieren*:

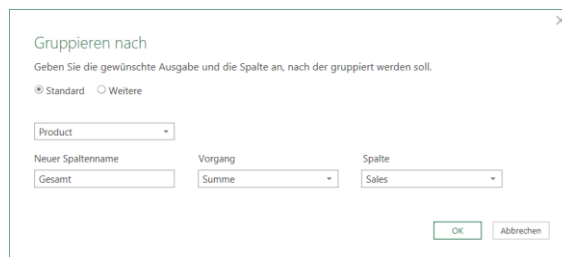


Abbildung 222

### Hinweise:

- Es kann auch nach mehreren Spalten gruppiert werden. Dazu müssen zuvor die zu gruppierenden Spalten markiert werden und anschließend gruppiert werden.
- Über *Weitere* kann nach weiteren Kriterien und Aggregationen gruppiert werden

### 7.14.1 Gruppieren nach mehreren Kriterien

Ordner: *Gruppieren*, Datei, PQ\_GroupBy\_MoreCriteria.xlsx

Folgende Mehrfachgruppierung zeigt an, ob ein Produkt von einem SalesRep mehrmals verkauft wurde und mit welchem Gesamtumsatz:

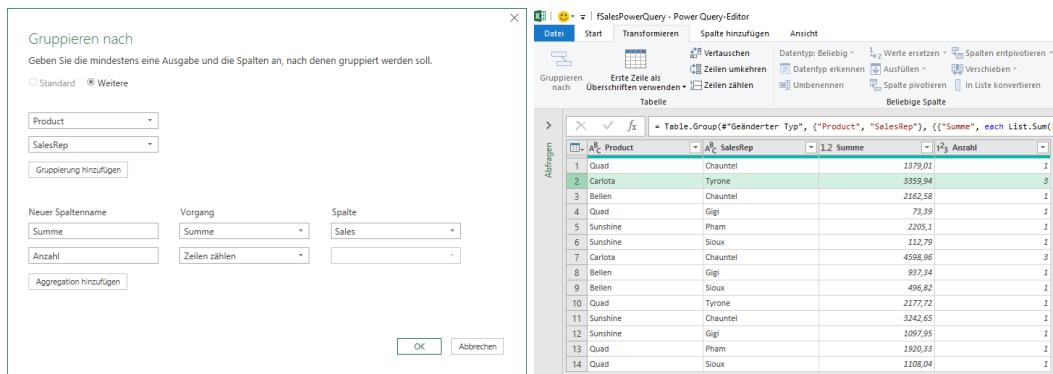


Abbildung 223

### 7.14.2 Gruppierungen mit Untertabellen

**Hinweis:** Zusatzthema außerhalb des Seminars

Um die Detaildaten (Untertabelle) einer Gruppe verfügbar zu machen kann über die Option *Weitere* eine Aggregation für *Alle Zeilen* durchgeführt werden:

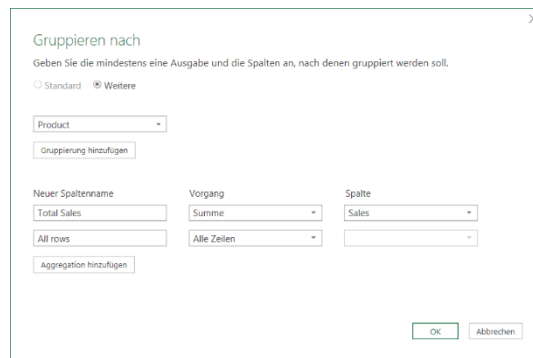


Abbildung 224

Die Untertabelle kann jetzt eingesehen und für weitere Auswertungen verwendet werden:

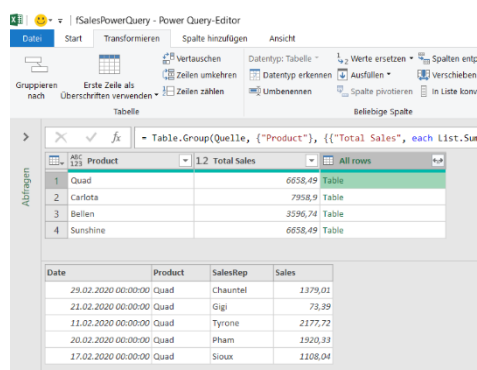


Abbildung 225

Aus dieser Untertabelle kann jetzt eine Liste mit den gewünschten Werten bezogen werden. Dazu erstellen Sie eine neue benutzerdef. Spalte und fügen die folgende Formel ein:

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

Anschließend kann diese Liste in kommasetrennte Werte umgewandelt werden:

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

## 7.15 Beispiele für Ausdrücke

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

## 8 Power Query advanced

### 8.1 Komplexe Excel-Dateien auswerten

#### 8.1.1 Einfacher Mittelwert mit List()

Ordner: Mittelwert einer Spalte

Spalte hinzufügen, dann:

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

z.B.:

```
let
    Quelle = Excel.CurrentWorkbook(){[Name="Tabelle1"]}[Content],
    #"Geänderter Typ" = Table.TransformColumnTypes(Quelle,{{"UMSATZ", Int64.Type}}),
    #"Hinzugefügte benutzerdefinierte Spalte" = Table.AddColumn(#"Geänderter Typ",
"Benutzerdefiniert", each List.Average(#"Geänderter Typ" [UMSATZ]))
in
    #"Hinzugefügte benutzerdefinierte Spalte"
```

#### 8.1.2 Dateinamen und Blattnamen in eine Liste übernehmen

Ordner: MultipleExcelImport-FileNames

- Alles aus Ordner importieren
- Rechtsklick *Dateiname/Transformieren/Kleinbuchstaben*
- Nach .xls filtern
- Dateiname: Rechtsklick oder Tab *Transformieren, Spalte Teilen*
  - Tipp: Optionen/Daten laden/(automatische) Typerkennung abschalten
- Nur *Content* und *City* Spalte behalten
- *Spalte hinzufügen/Benutzerdefiniert*
  - `=Excel.Workbook([Content],true)` holt die Tabellennamen  
true: erste Zeile der jeweiligen Tabelle = Überschrift
- Tabelle mit allen Items expandieren
- Filtern
  - `SalesRep.Name` nach beginnt nicht mit: *Sheet* filtern (Groß-Kleinschreibung!)
  - `SalesRep.Kind = Sheet`
- Nicht benötigte Spalten entfernen, übrig bleibt:

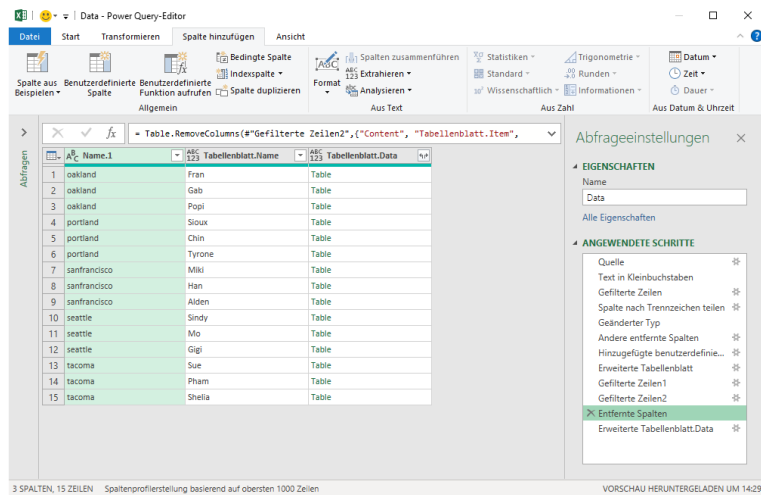


Abbildung 226

Zum Schluss die Tabellenspalte (*Data*) expandieren:

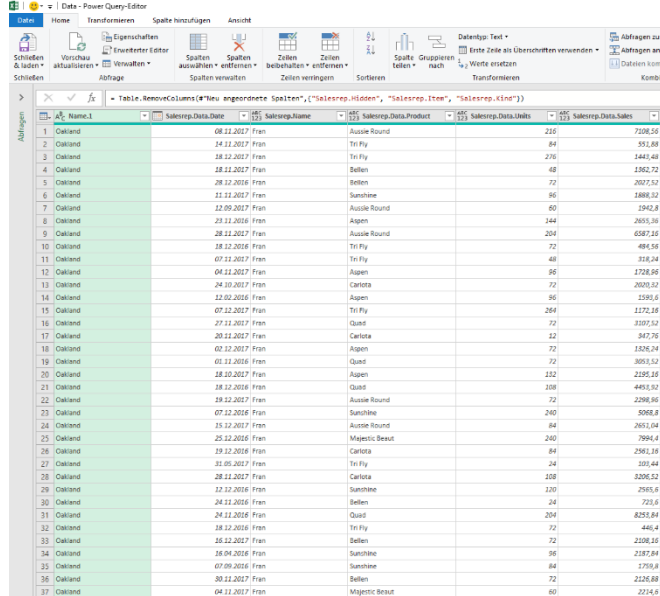


Abbildung 227

### 8.1.3 Ältere Duplikate löschen

Ordner: Duplikate/ÄltereDuplikateLöschen

- Beide Tabellen Laden und in einer neuen Abfrage hintereinander hängen.
- Nach *Datum* aufsteigend sortieren, Nach *Brand* aufsteigend, nach *Impressions* absteigend.
- Spalten Date bis Campaign markieren und Duplikate löschen

Hinweis: Gelöscht wird immer die zweite Duplikatzeile, also die mit der kleineren Anzahl von *Impressions*.

### 8.1.4 Datumsliste aus von..bis erzeugen

Hier sollen aus von...bis Intervallen, einzelne Zeilen erzeugt werden:

	A	B	C	D	E	F	G	H
1	ProductFamli	From	To	ID		ProductFamli	Date	ID
2	Fiction	01.11.2021	04.11.2023	A		Fiction	01.11.2021	A
3	Sport	03.11.2021	08.11.2021	B		Fiction	02.11.2021	A
4	News	02.11.2021	06.11.2021	C		Fiction	03.11.2021	A
5						Fiction	04.11.2021	A
6						Sport	03.11.2021	B
7						Sport	04.11.2021	B
8						Sport	05.11.2021	B
9						Sport	06.11.2021	B
10						Sport	07.11.2021	B
11						Sport	08.11.2021	B
12						News	02.11.2021	C
13						News	03.11.2021	C
14						News	04.11.2021	C
15						News	05.11.2021	C
16						News	06.11.2021	C
17								

Abbildung 228

Dies kann über die List-Funktion erreicht werden:

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

### 8.1.5 Vorperioden mittels SelfJoin ermitteln

Um mit Werten aus vorherigen Zeilen zu rechnen, können zwei jeweils um 1 versetzte Indexspalten angelegt werden, die mittels LeftOuter-SelfJoin verknüpft werden:

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

Abbildung 229

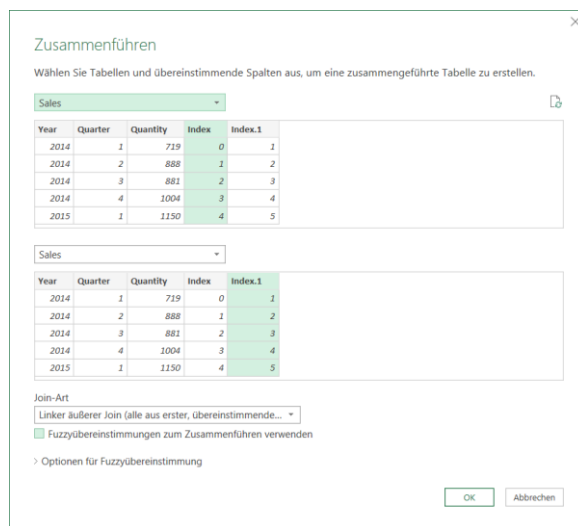


Abbildung 230

Damit wird für den Index 0 in der „verknüpften Tabelle“ keine Entsprechung gefunden (null), für den Index 1 wird der erste Wert aus der „verknüpften Tabelle“ zurückgeliefert usw.

Anschließend können Differenzen und %-Unterschiede ermittelt werden:

Differenzen:

```
[Quantity Current] - [Quantity Prev]
```

%-Unterschiede:

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

## 8.1.6 Webabfrage Aktien, Einstieg M

## ExamplefileLoop\_Work

[https://edvadamsde-my.sharepoint.com/:x/g/person/adams\\_edv-adams\\_de/EQVNBdHokB1NhtUj\\_7h9A8kB5OPldIYmSWCOtJO-sh0kAA?e=IGB2f3](https://edvadamsde-my.sharepoint.com/:x/g/person/adams_edv-adams_de/EQVNBdHokB1NhtUj_7h9A8kB5OPldIYmSWCOtJO-sh0kAA?e=IGB2f3)

## 8.2 M-Code

## 8.2.1 Aufbau M-Code

**Bearbeitung:**

- Angewendete Schritte
- Bearbeitungsleiste
- Erweiterter Editor

**TIPP:** Zoom im Editor: Strg + Shift + +/-

**Aufbau:**

Let

    #"Identifizier"=Expression

In

    Ergebnis



Abbildung 231

*StepName1* wird auch als Identifier, Variable oder Transformationsschritt bezeichnet. Über  $f_x$  können weitere Steps hinzugefügt werden.

**Beispiel:**

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

**Kommentare** werden wie in DAX notiert: // oder /\*...\*/

**TIPP:** ctrl+k+c | ctrl+k+u //Block kommentieren/auskommentieren

## 8.2.1.1 Notation Bezeichner (Identifier)

Ohne Leerzeichen: EntfernteSpalten = ...

Mit Leerzeichen: #"Geänderter Typ" = ...

Abbildung 232

### 8.2.2 Hilfen

=Funktionsname

zeigt Hilfe zu einer Funktion.

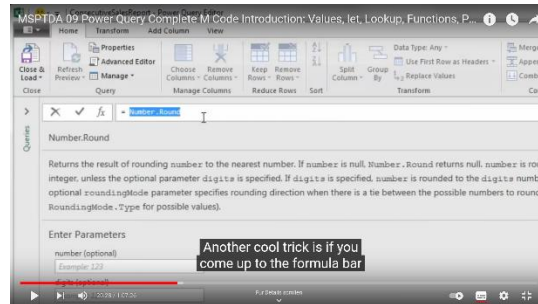


Abbildung 233

= #shared

zeigt alle Funktionen:

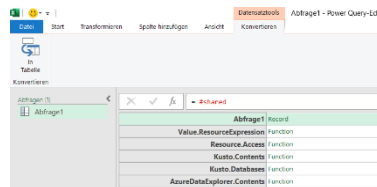


Abbildung 234

Danach einfach in Tabelle konvertieren und nach der gewünschten Funktion filtern.

### 8.2.3 Datentypen

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 ]

Abbildung 235

### 8.2.4 Arten von Werten



Abbildung 236

**Primitive Werte zufügen:**

$f_x$  klicken und Wert (Text, Zahl etc.) eingeben.

**Listen**

= {1,3, "Hallo Welt"}

= {1..50}

Listen können verschachtelt sein

= {1,3,{10,20},"Hallo Welt"}

und manche Funktionen erwarten verschachtelte Listen als Argumente.

**Records**

Hard kodierter Record (Literal): =[Product Name="Quad", Sales=20]

**Hinweis:** Product Name ist ein generalisierter Bezeichner, daher keine „ "

## 8.2.5 Tabelle erstellen (Tabellenkonstruktor)

#table({"First Column", "Second Column"}, {{1,2},{3,4}})

Siehe auch: [#table - PowerQuery M | Microsoft Learn](#)

## 8.2.6 Referenzen

**Mappe:** References

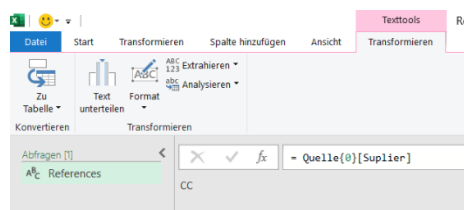
Projektion: Spalte(n)

Selektion: Zeile

Basis in Power Query ist 0 für die erste Zeile

= Quelle{0}[Supplier]

Verweis auf Wert (Drilldown): = Quelle{0}[Produkt] => Quelle/Zeile/Spalte



..Abbildung 237

**Einzelwerte referenzieren**

Row Index Lookup: = Quelle{2}[Products]

Diese Referenzierung ist von der Sortierung abhängig

Key Match Lookup: = Quelle{[Products="Carlota"]}[Price] => Voraussetzung: Products sind eindeutig

Diese Referenzierung ist von der Sortierung unabhängig

### Zeilen referenzieren

=Quelle{3}

Rückgabe: Record

### Spalten referenzieren

= Quelle[Products]

Rückgabe: Liste

### Mehrere Spalten referenzieren

= Quelle{[Products],[],[],[...]}

Rückgabe: Tabelle

#### 8.2.7 Each und \_

Der Unterstrich repräsentiert bei Zeilenweisen Kontext einen Datensatz (Record).

So wird bei diesem Beispiel aus jedem Datensatz die Spalte *Products* angezeigt:

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

Der \_ arbeitet kontextabhängig, im nachfolgenden Beispiel repräsentiert der \_ die jeweilige Liste. Hier wird aus der jeweiligen Liste im Feld *Quali* immer das 0te Element (Approbation/A) angezeigt:

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		Approbation
0	List		A

Abbildung 238

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

**Hinweis:** der \_ ist die Vereinfachte Schreibweise einer Funktion/Parameterübergabe:

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

Hier steht die Variable LE für ein Listenelement. Dabei wird LE (die Liste) an die Funktion **LE**[Quali]{0} übergeben, welche dann das 0te Element der Liste holt/zeigt.

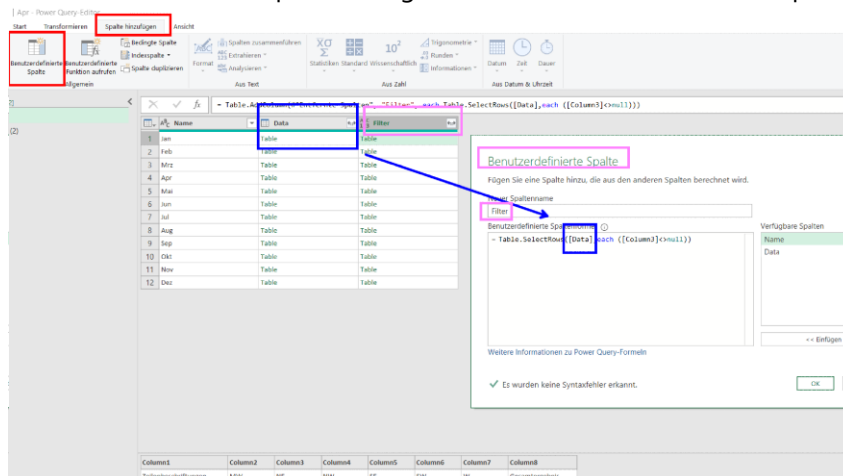
## 8.3 Fragmentierte Daten (Mappen/Tabellen) importieren

### 8.3.1 EineMappe, alle Blätter/Spalten gleich, pivottiert, Lösung mit Hilfsspalten

Ordner: Workfiles-Advanced\MultiTransform-Unpivot\Data\Reporting2020\_SameColumns.xlsx"

- Excel Einzeldatei importieren
- Eine Tabelle (Abfrage) behalten und duplizieren
- Im Duplikat Transformieren, und M-Funktionen kopieren

- Dann in benutzerdef. Spalte einfügen und Tabellennamen der Vorspalte verwenden



- Zum Schluss Hilfspalten löschen und letzte Spalte expandieren

### 8.3.2 Mehrere Mappen aus Ordner, alle Blätter/Spalten gleich, pivottiert, Lösung mit Hilfspalten

Ordner: Workfiles-Advanced\MultiTransform-Unpivot\Data\Reporting2020\_SameColumns.xlsx"

- Ordnerimport Excel, nur Tabellen mit SameColumns behalten
- Nur Spalte *Content* behalten
- Benutzerdef. Spalte: Excel.Workbook([Content])
- Spalte expandieren
- Nur Spalte *Data* behalten
- Jetzt kann die Hilfspaltenlösung aus 8.3.1 verwendet werden

### 8.3.3 Mehrere Mappen aus Ordner, alle unterschiedliche Spalten, pivottiert, Lösung mit M und Function

- Ordnerimport Excel, nur Tabellen mit DiffColumns behalten
- Nur Spalte *Content* behalten
- Benutzerdef. Spalte: Excel.Workbook([Content])
- Neue Spalte expandieren

- Danach nur Dateinamen, Blattnamen und Tabellendaten behalten:

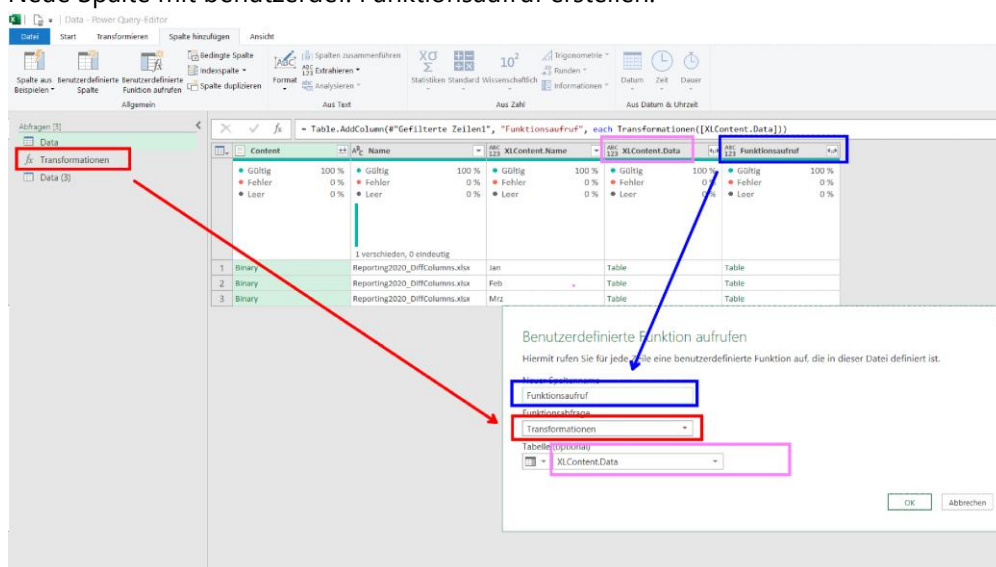
	Name	Name.1	Data
1	Reporting2019_DiffColumns.xlsx	Jan	Table
2	Reporting2019_DiffColumns.xlsx	Feb	Table
3	Reporting2019_DiffColumns.xlsx	Mrz	Table
4	Reporting2019_DiffColumns.xlsx	Apr	Table
5	Reporting2019_DiffColumns.xlsx	Mai	Table
6	Reporting2019_DiffColumns.xlsx	Jun	Table
7	Reporting2019_DiffColumns.xlsx	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

- Abfrage duplizieren und eine Tabelle aus Spalte *Data* anzeigen
- Hier gewünschte Transformationen vornehmen und im erweiterten Editor kopieren
- Leere Abfrage anlegen und eine Function daraus erstellen und zwischen *let* und *in* die im vorherigen Schritt kopierten Trafos einfügen. Danach die Benennung der Quellen anpassen:

```
(Tabelle) =>
let
    Quelle = Tabelle,

    #"Gefilterte Zeilen1" = Table.SelectRows(Quelle, each ([Column3] <> null) and ([Column1] <>
"Gesamtergebnis")),
    #"Entfernte Spalten" = Table.RemoveColumns(#"Gefilterte Zeilen1",{"Column8"}),
    #"Höher gestufte Header" = Table.PromoteHeaders(#"Entfernte Spalten", [PromoteAllScalars=true]),
    #"Entpivotierte Spalten" = Table.UnpivotOtherColumns(#"Höher gestufte Header", {"Name",
"XLContent.Name", "Data (2).Zeilenbeschriftungen"}, "Attribut", "Wert")
in
    #"Entpivotierte Spalten"
```

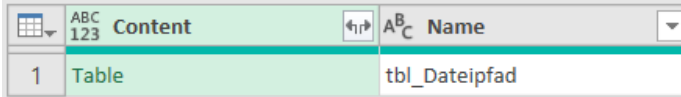
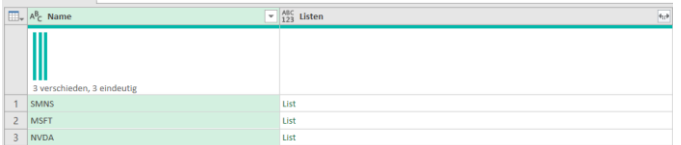
- Neue Spalte mit benutzerdef. Funktionsaufruf erstellen:



- Nicht benötigte Spalten entfernen

- Daten in Spalte Funktionsaufruf expandieren

### 8.4 Nützliche M-Funktionen

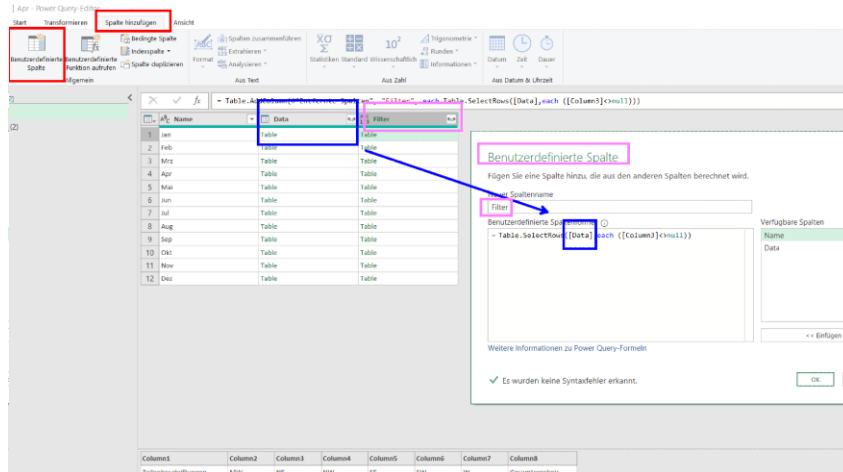
Excel.CurrentWorkbook()	Gibt Excelobjekte zurück außer Tabellenblätter 
Folder.Files(Dateipfad)	Importiert alle Dateien eines Ordners
Table.ColumnNames(table)	Holt alle Spaltennamen einer Tabelle als Liste
Table.ToList(Table)	Erstellt List aus Tabelle
Table.FromList(List)	Erstellt Tabelle aus Liste
Table.ColumnNames	Holt Spaltennamen
Table.SelectColumns	Wählt Spalten aus
Table.SelectRows	Filtert
Table.RenameColumns	Table.RenameColumns(Table, { {"CustomerNum", "CustomerID"}, {"PhoneNum", "Phone"} })
List.Zip	Liste 1: {1, 2, 3} Liste 2: {'a', 'b', 'c'} Ergebnis: {1, 'a'}, {2, 'b'}, {3, 'c'}
Table.RenameColumns UND List.Zip	Table.RenameColumns(Tabelle, List.Zip({CurrentColumnNames, NewColumnNames}))
Table.Group(table as table, key as any, aggregatedColumns as list, optional groupKind as nullable number, optional comparer as nullable function) as table	Gruppiert Tabellen <b>key:</b> Nach welchen Spalten? <b>aggregatedColumns:</b> Berechnete Spalten <b>groupKind:</b> bei GroupKind.Local werden nur aufeinanderfolgende Zeilen als Gruppe betrachtet und nicht die gesamte Tabelle.
Table.ExpandList	Expandiert alle Listen in einer Listenspalte Table.ExpandListColumn(Tabelle, Spaltenname) 

## 8.5 Importe in der Praxis

### 8.5.1 EineMappe, alle Blätter/Spalten gleich, pivottiert, Lösung mit Hilfsspalten

Ordner: Workfiles-Advanced\MultiTransform-Unpivot\Data\Reporting2020\_SameColumns.xlsx"

- Excel Einzeldatei importieren
- Eine Tabelle (Abfrage) behalten und duplizieren
- Im Duplikat Transformieren, und M-Funktionen kopieren
- Dann in benutzerdef. Spalte einfügen und Tabellennamen der Vorspalte verwenden



- Zum Schluss Hilfsspalten löschen und letzte Spalte expandieren

### 8.5.2 Mehrere Mappen aus Ordner, alle Blätter/Spalten gleich, pivottiert, Lösung mit Hilfsspalten

Ordner: Workfiles-Advanced\MultiTransform-Unpivot\Data\Reporting2020\_SameColumns.xlsx"

- Ordnerimport Excel, nur Tabellen mit SameColumns behalten
- Nur Spalte *Content* behalten
- Benutzerdef. Spalte: Excel.Workbook([Content])
- Spalte expandieren
- Nur Spalte *Data* behalten

## 8.6 Functions und Parameter

### 8.6.1 Functions definieren

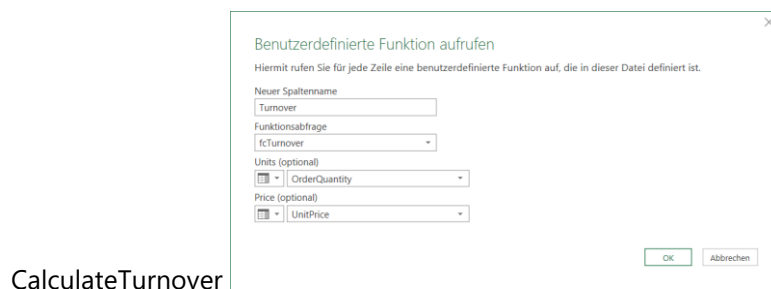
Ordner: Functions

Leere Abfrage erstellen, dann erweiterter Editor:

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

#### Function benutzen:

Spalte hinzufügen/benutzerdefinierte Funktion aufrufen:



CalculateTurnover

Abbildung 239

## 8.6.2 Parameterabfragen definieren (Viele Dateien auf einmal transformieren)

Ordner: Parameter Queries

Um viele Dateien auf einmal zu transformieren und zusammenzufügen, erstellt man zunächst eine eigene Parameterabfrage. Diese empfängt den Namen der jeweiligen Tabelle und transformiert diese, sowie alle weiteren.

### Parameterabfrage anlegen:

- Eine beliebige aus den zu transformierenden Dateien öffnen und nach Bedarf transformieren z.B.: (automatischen Schritt *geänderter Typ* am besten immer entfernen)
  - Transponieren
  - Überschriften hochsetzen
  - Depivotieren
- Erweiterter Editor, ersten Schritt *Quelle=...* entfernen
- Neuen Parameter z.B. *TableName* erstellen und bei Tabellennamen einsetzen:

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"
```

Abbildung 240

- Abfrage abspeichern, z.B. unter dem Namen *TransformTable*

### Parameterabfrage einsetzen

- Ordnerimport durchführen und transformieren
- Nur Content behalten
- Benutzerdef. Spalte, z.B. *GetWorkbookContent* erstellen und den Inhalte der Arbeitsmappe holen:

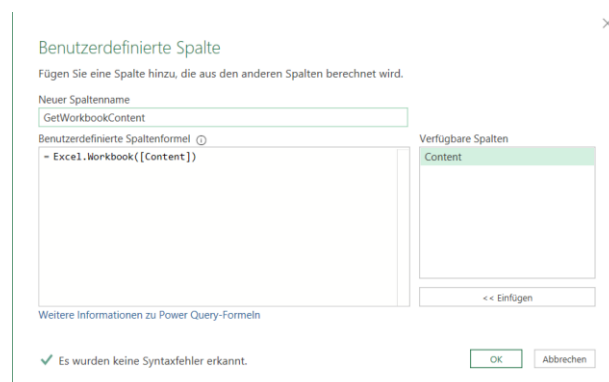


Abbildung 241

- Spalte Content (Binary) entfernen
- Tabelle expandieren und nur die Spalte *Data* behalten, denn diese beinhaltet die Tabellenblattinhalte
- Spalte hinzufügen/Benutzerdef. Funktion:

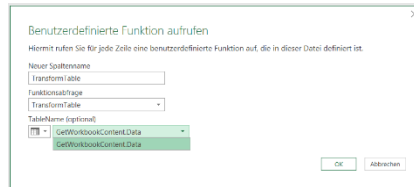


Abbildung 242

- jetzt *GetWorkbookContent.Data* an unsere neue Funktion *TransformTable* übergeben.
- Zum Schluss erste Spalte entfernen und Daten expandieren

### 8.6.3 Daten aus Excel an Abfrage übergeben

Ordner: Parameter Queries

In diesem Beispiel soll der Ordner-Importpfad an die Parameterabfrage aus Kap. 8.6.2 aus Excel übergeben werden:

- Excel: Tabelle anlegen und Pfad dort eingeben
- Tabelle benennen, z.B. tPath
- Neue Abfrage aus dieser Tabelle anlegen
- Drilldown in die erste Zelle:

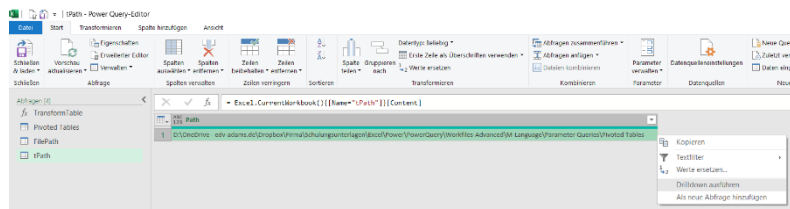


Abbildung 243

- Im erweiterten Editor neuen Schritt einfügen der den vorherigen an Folder.Files übergibt:

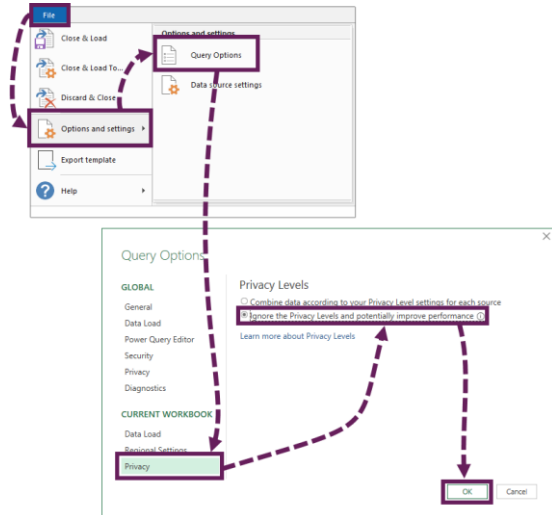
```
let
    Quelle = Excel.CurrentWorkbook() { [Name="tPath"] } [Content],
    IPATH = Quelle{0}[Path], //Hole aus Tabelle tPath Zeile 0, Spalte Path
    FilesPath=Folder.Files(IPATH)
in
    FilesPath
```

Zum Schluss das Ergebnis dieser Abfrage an die gewünschte Abfrage als „Dateipfad“ übergeben:

```
let
    Quelle = FilesPath,
    ...
    ...
```

Notiz: Excel-Mappenobjekte holen: Excel.Workbook(), Excel.CurrentWorkbook()

## 8.6.4 Bei Formula Firewall Fehlermeldung



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

## 8.7 Techniken

### 8.7.1 Parameter

**DB:** AdventureWorks\_Database.xlsx

Es soll ein Dropdown erstellt werden, das als Filterkriterium für eine Abfrage (Territories) dient:

- Tabelle Territories importieren
- **TIPP:** Rechtsklick auf Spalte/*Als neue Abfrage hinzufügen* erzeugt eine Liste aus der Spalte. Hier bitte aus Spalte *Country* anwenden
- Duplikate in dieser Liste entfernen
- *Start/Parameter/Neuer Parameter:*

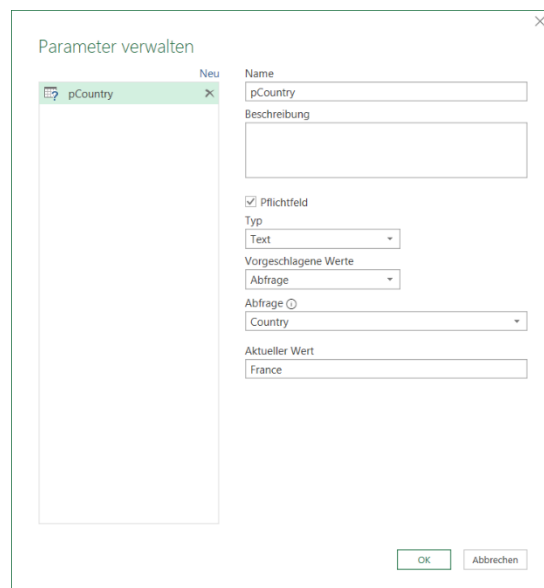


Abbildung 244

- Zielabfrage in die der Parameter als Filterkriterium einfließen soll aufrufen und beliebig an *Country* filtern
- Nun den M-Code ändern von  
`= Table.SelectRows("#"Geänderter Typ", each ([Country] = "Australia"))`

nach:

```
= Table.SelectRows("#Geänderter Typ", each ([Country] = pCountry))
```

(pCountry ist der Name des Parameters)

### Parameter in Power Query benutzen:

*Start/Parameter verwalten/Parameter bearbeiten:*

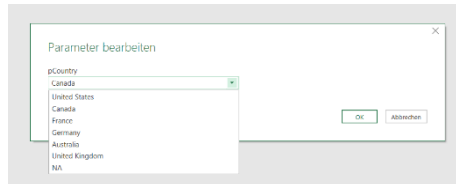


Abbildung 245

Zum Schluss die gefilterte Abfrage aktualisieren.

### Parameter in Power BI Desktop verwenden:

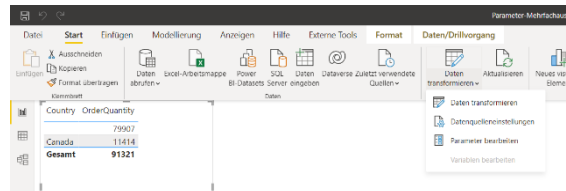


Abbildung 246

## 8.8 Gemischtes

### 8.8.1 Zeilen einfügen

#"Insert Row" = Table.InsertRows("#Hinzugefügte benutzerdefinierte Spalte", 0, {[From= 1, To= "data2", Currency="AA", Benutzerdefiniert="dd"]})

### 8.8.2 Vortag berechnen

Date.AddDays([Auftragseingang],-1) (Nur bei durchgängigen, lückenlosen Daten!)

### 8.8.3 Datediff

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

### 8.8.4 Daten von...bis in Zeilen auflisten

Ordner: Time interval FROM TO

- Tabelle nach Query übernehmen
- Uhrzeit aus Datum entfernen
- Neue Spalte einfügen: =List.Dates([From],Duration.Days([To]-[From])+1,#duration(1,0,0,0))

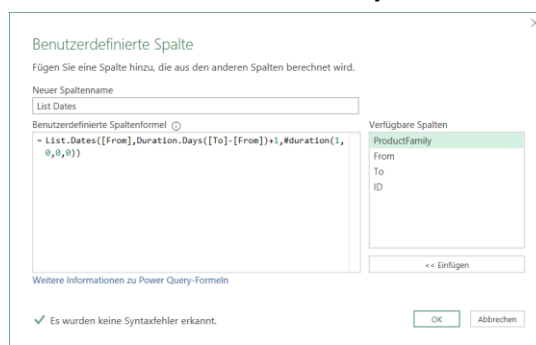


Abbildung 247

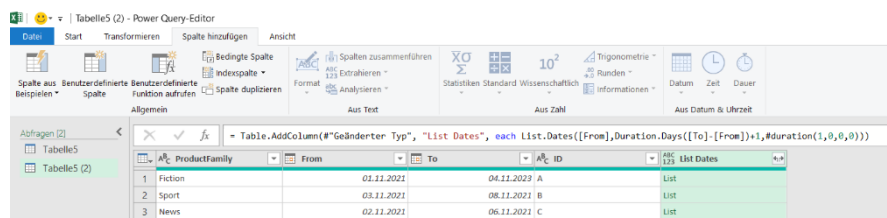


Abbildung 248

- Spalte *List Dates* über das Doppelpfeilsymbol *in Zeilen* expandieren
- Spalten *From/To* entfernen:



```
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

Abbildung 249