

# Power Query und M

## Excelstammtisch vom 07.05.2019

(zusammengefasst von René Martin)

### 1. Kurzer Blick zurück

Mit Office 2010 hat Microsoft für Excel mehrere Werkzeuge veröffentlicht, die einen erweiterten Datenbankzugriff und Möglichkeiten eines Reportings eröffnen: Power Query, PowerPivot, Power Maps und PowerView. Man musste sie in der damaligen Excel-Version aus dem Internet herunterladen und in Excel installieren. Ab der Version 2013 ist Power Query fester Bestandteil von Excel. In der Version 2016 und 2019 signiert dieses Programm in Excel unter *Daten abrufen und transformieren* im Register *Daten*.

Parallel zu Excel wurde dieses Werkzeug in Power BI integriert, das kostenlos heruntergeladen werden kann. Nicht nur, dass Microsoft ein mächtiges Werkzeug für Excel und Power BI geschaffen hat; im Zusammenspiel mit Visual Studio wird es zu einer cleveren Hilfe für den Zugriff auf Daten und deren Verarbeitung.

Zudem ist Power Query auch vorhanden in:

- SQL Server Analysis Services (SSAS) ab Version 2017,
- Common Data Service (CDS),
- Common Data Service for Apps (CDS-A)
- Den neuen Power BI Dataflows

### 2. Beispiel 1: Die Oberfläche von und das Arbeiten mit Power Query

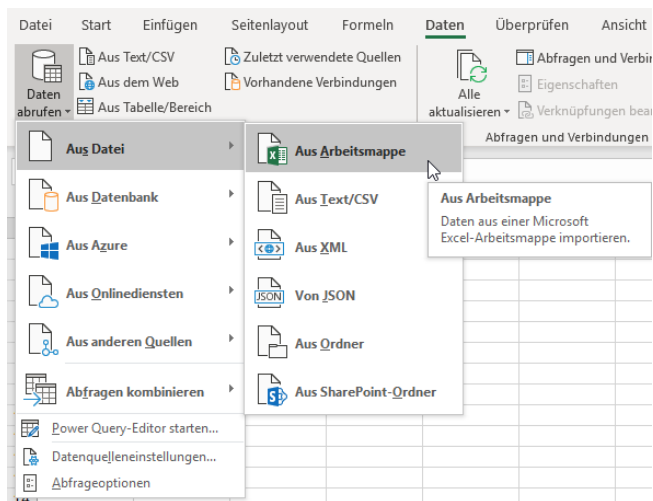
Angenommen Sie erhalten in regelmäßigen Abständen eine Liste aus einem System, das folgendermaßen aussieht:

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Verkäufe									
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											

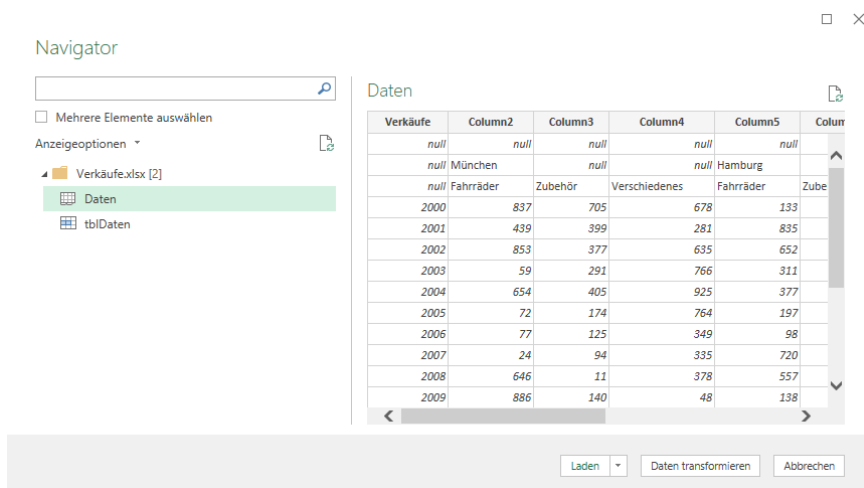
		München			Hamburg			Berlin		
		Fahrräder	Zubehör	Verschiedenes	Fahrräder	Zubehör	Verschiedenes	Fahrräder	Zubehör	Verschiedenes
2000		837	705	678	133	593	115	302	175	631
2001		439	399	281	835	58	346	788	46	415
2002		853	377	635	652	494	238	939	670	936
2003		59	291	766	311	210	788	207	374	466
2004		654	405	925	377	786	661	859	251	49
2005		72	174	764	197	125	781	732	583	683
2006		77	125	349	98	700	337	800	871	332
2007		24	94	335	720	172	643	201	870	230
2008		646	11	378	557	1000	743	992	40	145
2009		886	140	48	138	536	559	810	870	13
2010		608	264	27	546	247	630	297	73	586
2011		961	899	752	944	822	122	394	401	694
2012		161	25	753	592	624	706	719	521	15
2013		448	788	213	434	738	876	652	55	19
2014		980	577	899	690	672	698	510	185	681
2015		107	869	418	995	539	570	40	388	921
2016		837	705	678	133	593	115	302	175	631
2017		439	399	281	835	58	346	788	46	415
2018		853	377	635	652	494	238	939	670	936

Diese Liste soll aufbereitet werden, damit man sie als Pivottable nach Orten und/oder Jahr und/oder Kategorie zusammenfassen kann.

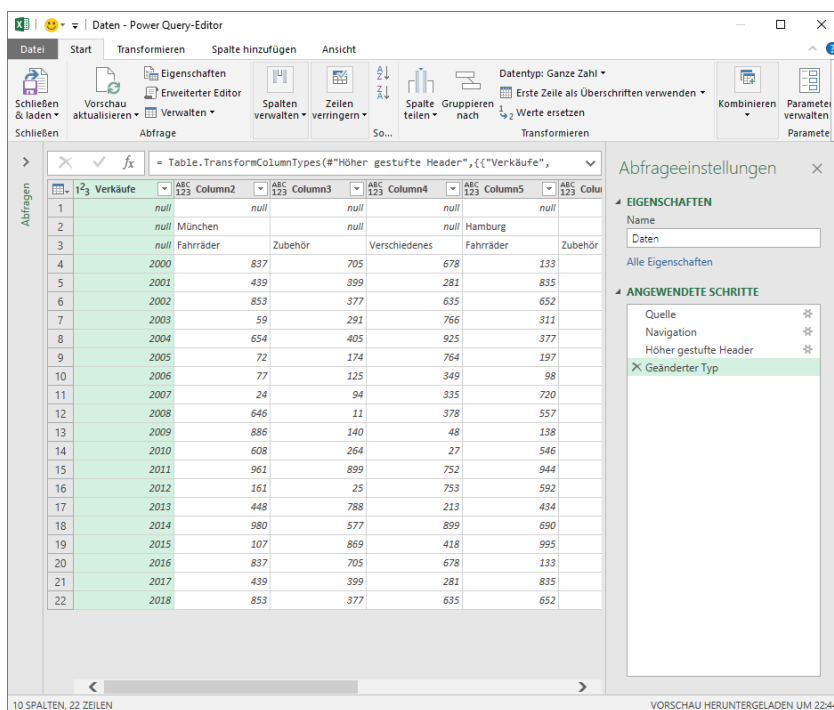
In einer neuen leeren Datei wird Bezug auf diese Datenquelle genommen:



Der Assistent zeigt die Liste der Tabellenblätter, Namen und intelligenten Tabellen an:



Mit der Schaltfläche *Daten transformieren* wird der Power Query-Editor geöffnet:



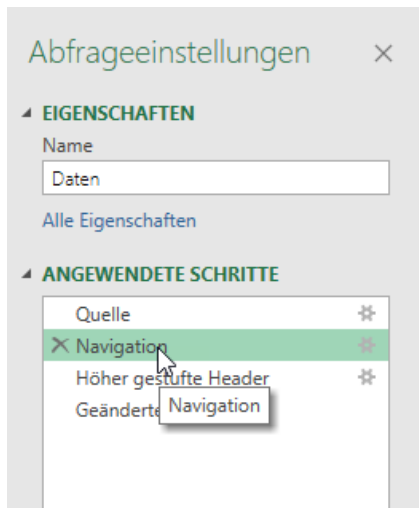
## Tipp

Die Bearbeitungsleiste ist wichtig und notwendig. Sie sollte über die Registerkarte Ansicht eingeschaltet werden.

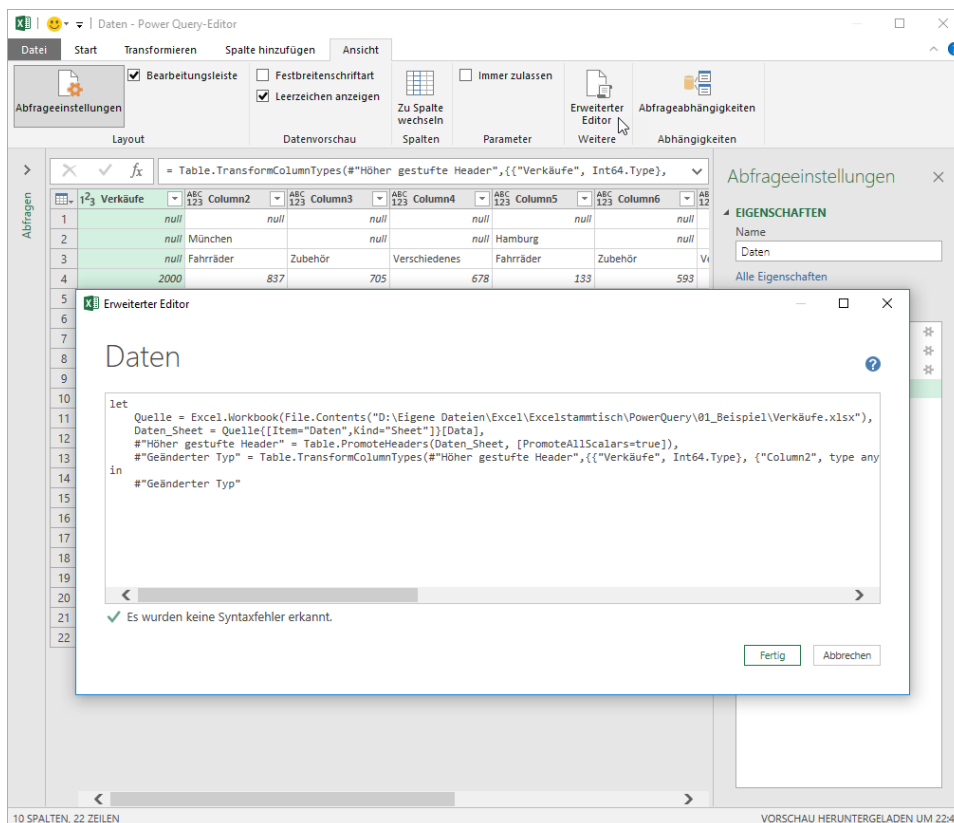
Alle Schritte werden im Aufgabenbereich *Abfrageeinstellungen* aufgelistet. Man kann einzelne Schritte zurückgehen und sehen, wie die Daten zu diesem Zeitpunkt aussahen. Man kann einzelne Schritte dazwischen löschen und auch einfügen.

## Achtung

Da man Schritt löschen und „zurückgehen“ kann, gibt es keine Rückgängigfunktion!

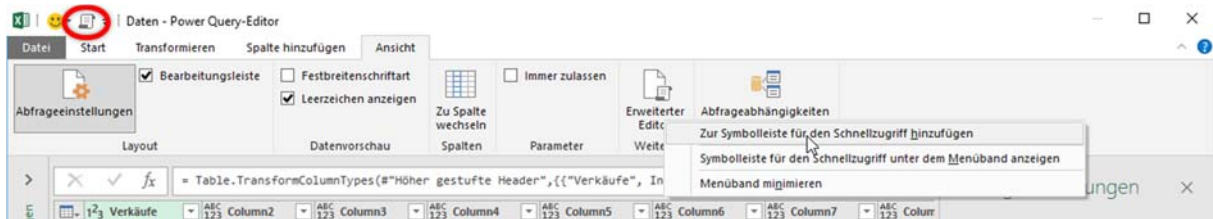


Jeder Schritt wird in der Sprache M in der Bearbeitungsleiste angezeigt. Alle Schritte können über den *Erweiterten Editor* (Registerkarte *Ansicht*) eingesehen werden.



## Tipp

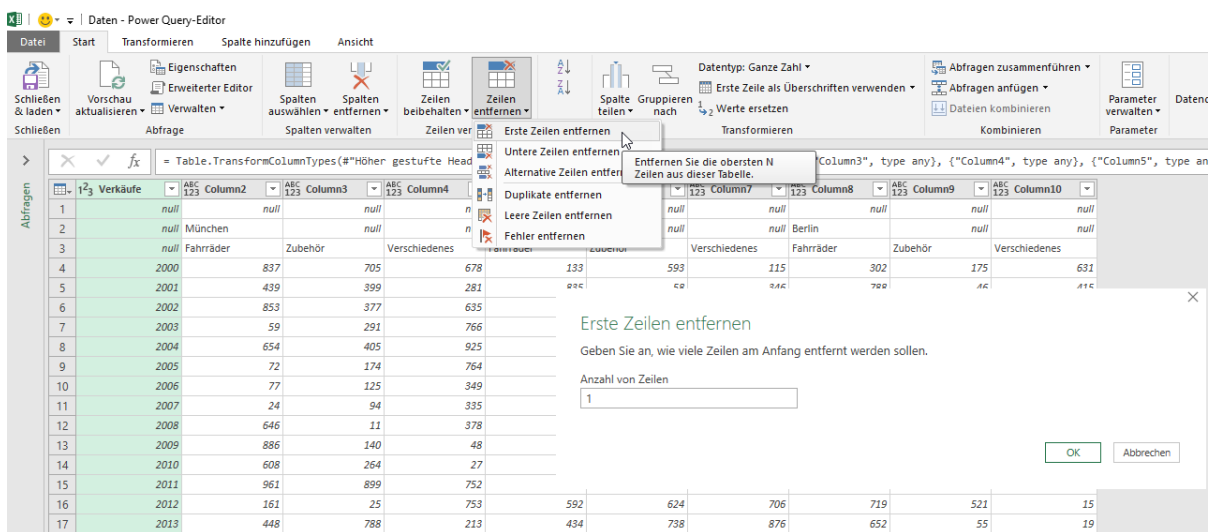
Füge den Editor als Symbol in der Symbolleiste für den Schnellzugriff hinzu!



Und nun kann die Liste bearbeitet werden.

## Schritt 1: Die erste Zeile wird entfernt.

*Start / Zeilen verringern / Zeilen entfernen / Erste Zeilen entfernen*

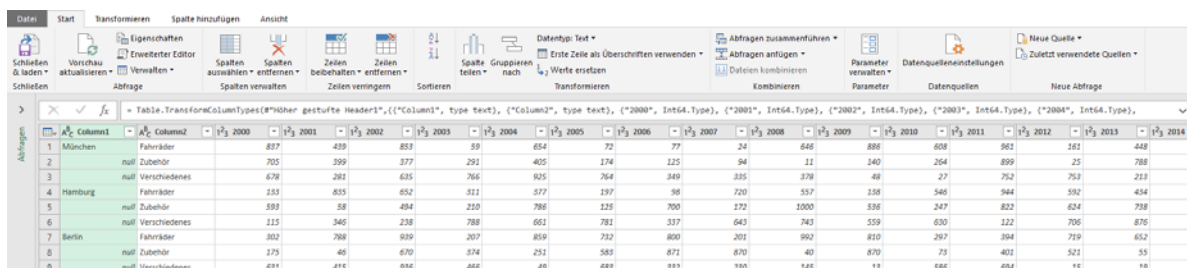


## Schritt 2: Die Tabelle transponieren

*Transformieren / Tabelle / Vertauschen* (entspricht dem Excel-Befehl transponieren)

## Schritt 3: Die erste Zeile wird zur Überschrift

*Start / Transformieren / Erste Zeile als Überschrift verwenden.* Das Ergebnis sieht folgendermaßen aus:

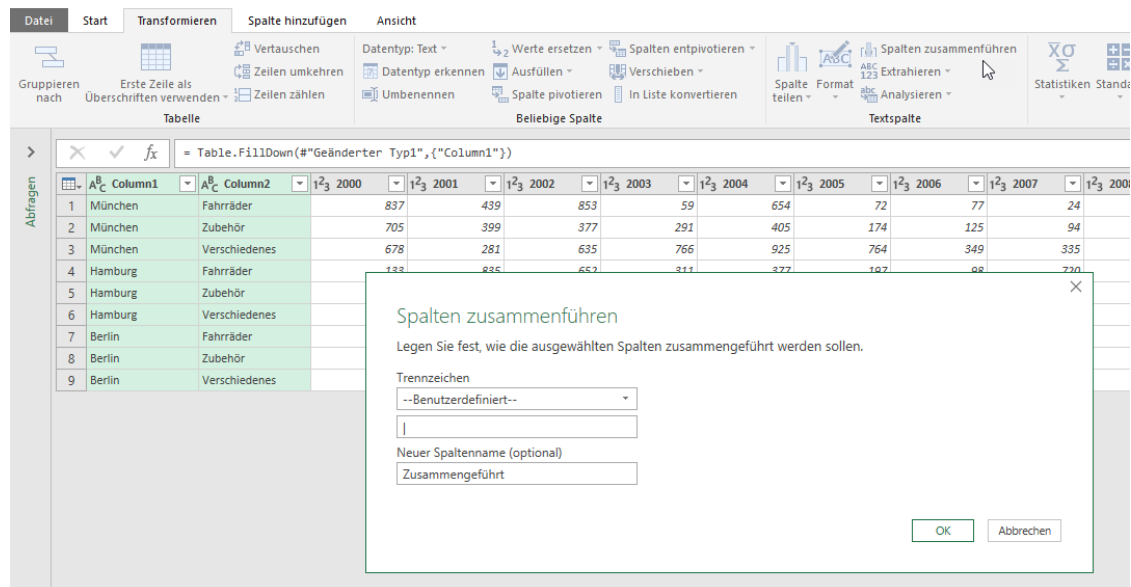


## Schritt 4: Erste Spalte ausfüllen

Der Befehl *Transformieren / Beliebige Spalte / Ausfüllen / Nach unten* füllt die Lücken der ersten Spalte auf.

## Schritt 5: Spalten verknüpfen

Nun können die ersten beiden Spalten verbunden werden. Dies leistet Transformieren / Textspalte / Spalten zusammenführen. Im Dialog kann das Trennzeichen und der Name der neuen Spalte ausgewählt werden.



## Schritt 6: Spalten entpivotieren

Nun können die übrigen Spalten entpivotiert werden. Dieser Befehl findet sich im Kontextmenü oder in Transformieren / Beliebige Spalte. Das Ergebnis sieht wie folgt aus:

	Zusammengeführt	Attribut	Wert
1	München/Fahrräder	2000	837
2	München/Fahrräder	2001	439
3	München/Fahrräder	2002	853
4	München/Fahrräder	2003	59
5	München/Fahrräder	2004	654
6	München/Fahrräder	2005	72
7	München/Fahrräder	2006	77
8	München/Fahrräder	2007	24
9	München/Fahrräder	2008	646
10	München/Fahrräder	2009	886
11	München/Fahrräder	2010	608
12	München/Fahrräder	2011	962
13	München/Fahrräder	2012	342
14	München/Fahrräder	2013	448
15	München/Fahrräder	2014	980
16	München/Fahrräder	2015	107
17	München/Fahrräder	2016	837
18	München/Fahrräder	2017	439
19	München/Fahrräder	2018	853
20	München/Zubehör	2000	705
21	München/Zubehör	2001	399

## Schritt 7: Erste Spalte trennen

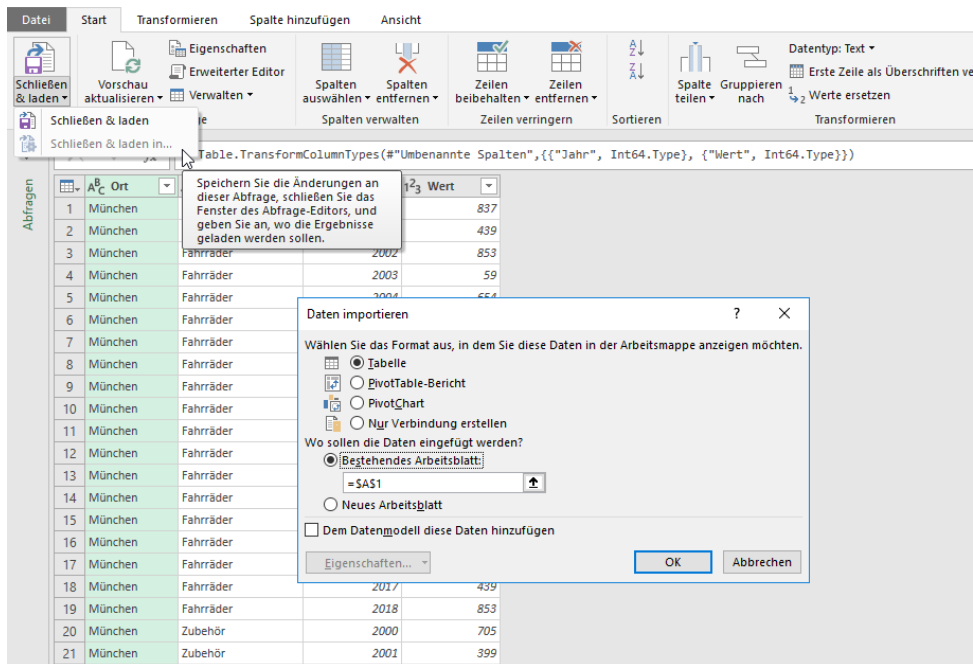
Nun kann die erste Spalte wieder aufgesplittet werden (Start / Transformieren / Spalte teilen / Nach Trennzeichen).

## Schritt 8: Überschrift und Datentyp

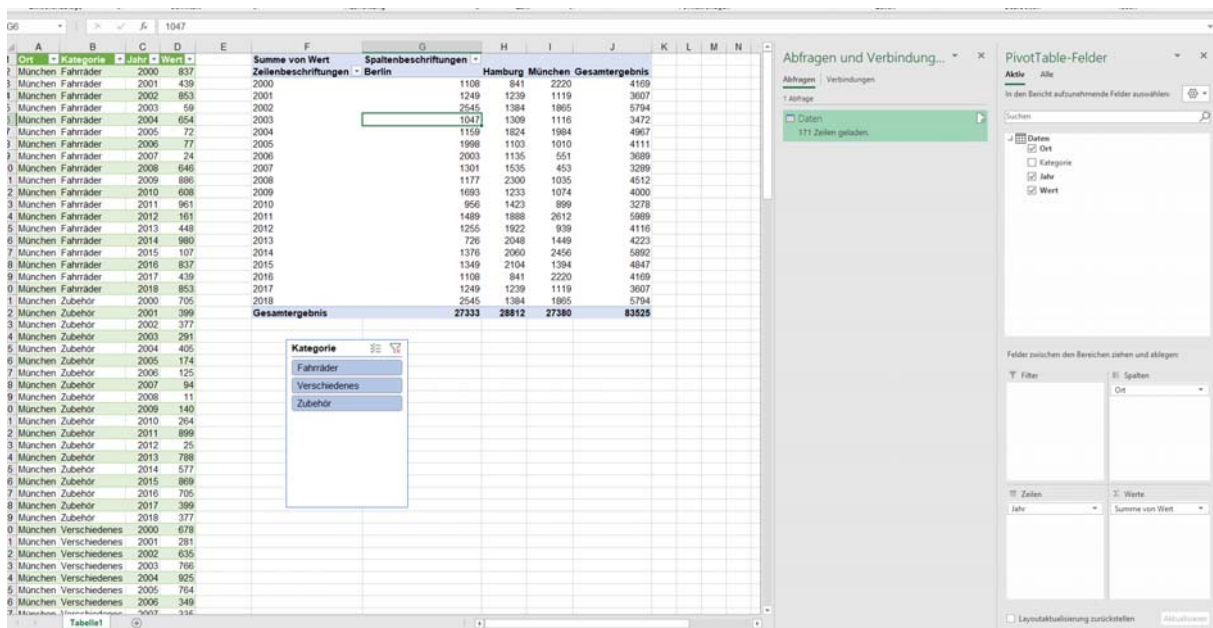
Nun kann die Überschrift verändert werden und der Datentyp für die Spalten Jahr und Wert in Zahl geändert werden.

## Schritt 9: Nach Excel

Anschließend werden die Daten nach Excel geschrieben – entweder als Verknüpfung oder sie werden in die Tabelle eingefügt: Start / Schließen & Laden / Schließen & laden in.



Auf Basis dieser intelligenten Tabelle kann nun eine Pivottable erstellt werden.



## Hinweis

Den Aufgabenbereich Abfragen und Verbindungen erhält man über Daten / Abfragen und Verbindungen. Über diesen Aufgabenbereich gelangt man wieder zurück zum Power Query-Editor oder kann aus einer Verbindung eine Tabelle erzeugen.

Dieses Beispiel zeigt die Mächtigkeit von Power Query. Es verdeutlicht, dass die einzelnen Assistenten leichter zu bedienen sind als die Funktionen oder Befehle von Excel und zum Teil mehr Möglichkeiten verfügen als Excel (Entpivotieren, erste Zeile wird zur Überschrift, Daten trennen, Spalten zusammenfügen, ...)

### 3. M

Ein Blick in den Erweiterten Editor des vorigen Beispiels zeigt den Code:

```
let
    Quelle = Excel.Workbook(File.Contents("D:\Eigene
Dateien\Excel\Excelstammtisch\PowerQuery\01_Beiispiel\Verkäufe.xlsx"), null,
true),
    Daten_Sheet = Quelle{[Item="Daten",Kind="Sheet"]}[Data],
    #"Höher gestufte Header" = Table.PromoteHeaders(Daten_Sheet,
[PromoteAllScalars=true]),
    #"Geänderter Typ" = Table.TransformColumnTypes(#"Höher gestufte
Header",{{"Verkäufe", Int64.Type}, {"Column2", type any}, {"Column3", type
any}, {"Column4", type any}, {"Column5", type any}, {"Column6", type any},
{"Column7", type any}, {"Column8", type any}, {"Column9", type any},
{"Column10", type any}}),
    #"Entfernte oberste Zeilen" = Table.Skip(#"Geänderter Typ",1),
    #"Transponierte Tabelle" = Table.Transpose(#"Entfernte oberste
Zeilen"),
    #"Höher gestufte Header1" = Table.PromoteHeaders(#"Transponierte
Tabelle", [PromoteAllScalars=true]),
    #"Geänderter Typ1" = Table.TransformColumnTypes(#"Höher gestufte
Header1",{{"Column1", type text}, {"Column2", type text}, {"2000",
Int64.Type}, {"2001", Int64.Type}, {"2002", Int64.Type}, {"2003",
Int64.Type}, {"2004", Int64.Type}, {"2005", Int64.Type}, {"2006",
Int64.Type}, {"2007", Int64.Type}, {"2008", Int64.Type}, {"2009",
Int64.Type}, {"2010", Int64.Type}, {"2011", Int64.Type}, {"2012",
Int64.Type}, {"2013", Int64.Type}, {"2014", Int64.Type}, {"2015",
Int64.Type}, {"2016", Int64.Type}, {"2017", Int64.Type}, {"2018",
Int64.Type}}),
    #"Nach unten gefüllt" = Table.FillDown(#"Geänderter Typ1",{ "Column1"}),
    #"Zusammengeführte Spalten" = Table.CombineColumns(#"Nach unten
gefüllt",{ "Column1", "Column2"},Combiner.CombineTextByDelimiter("|",
QuoteStyle.None),"Zusammengeführt"),
    #"Entpivotierte andere Spalten" =
Table.UnpivotOtherColumns(#"Zusammengeführte Spalten", {"Zusammengeführt"},
"Attribut", "Wert"),
    #"Spalte nach Trennzeichen teilen" = Table.SplitColumn(#"Entpivotierte
andere Spalten", "Zusammengeführt", Splitter.SplitTextByDelimiter("|",
QuoteStyle.Csv), {"Zusammengeführt.1", "Zusammengeführt.2"}),
    #"Geänderter Typ2" = Table.TransformColumnTypes(#"Spalte nach
Trennzeichen teilen",{{"Zusammengeführt.1", type text},
{"Zusammengeführt.2", type text}}),
    #"Umbenannte Spalten" = Table.RenameColumns(#"Geänderter
Typ2",{{"Zusammengeführt.1", "Ort"}, {"Zusammengeführt.2", "Kategorie"},
{"Attribut", "Jahr"}}),
    #"Geänderter Typ3" = Table.TransformColumnTypes(#"Umbenannte
Spalten",{{"Jahr", Int64.Type}, {"Wert", Int64.Type}})
```

in

```
"#Geänderter Typ3"
```

Die angewendeten Schritte werden zeilenweise mitprotokolliert. Jeder Befehl (außer der letzten Zeile) endet mit einem Komma. Die dahinter liegende Sprache heißt M. Der gesamte Code ist in zwei `let ... in`-Blöcken eingeschlossen. Im `let`-Teil stehen die einzelnen Zeilen, die von einem Namen eingeleitet werden (als Quasivariablen), der der Wert einer Funktion übergeben wird. Die Zeilen werden linear von oben nach unten abgearbeitet – Sprünge sind natürlich möglich. Im `in`-Teil befindet sich das Ergebnis der Transformation. Der von Power Query erzeugte Code kann modifiziert werden.

### Hinweis

Die Befehle sind Case sensitive. Das bedeutet: ein Befehl *Text.StartsWith* wird nicht erkannt wenn er als *Text.Startswith* geschrieben wird.

Es stehen die klassischen Funktionen zur Verfügung:

Textfunktionen, beispielsweise: *Text.Length*, *Text.StartsWith*, *Text.Middle*, *Text.PositionOf*, ...

Datumsfunktionen, beispielsweise: *Date.Day*, *Date.DayOfWeek*, *Date.QuarterOfYear*, *DateTime.LocalNow*, ...

Zahlenfunktionen, beispielsweise: *Number.Exp*, *Number.Sin*, *Number.Factorial*, ...

Verzweigungen wie *If*

Interessanter sind natürlich die Befehle der Tabellen (*Table*), die eine Manipulation der Tabelle bewirken: *AddColumns*, *CombineColumns*, *DuplicateColumn*, *FillDown*, *RemoveColumns*, *RenameColumns*, *RemoveFirstN*, *RemoveLastN*, *Sort*, *SplitColumn*, ...

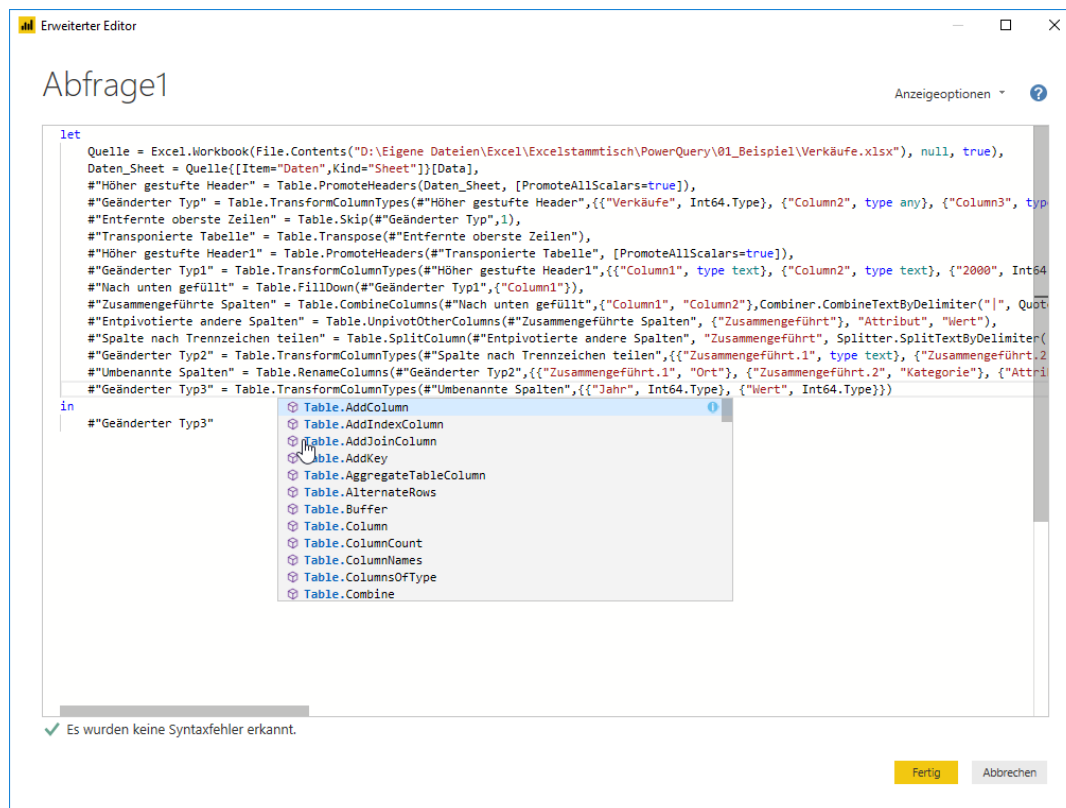
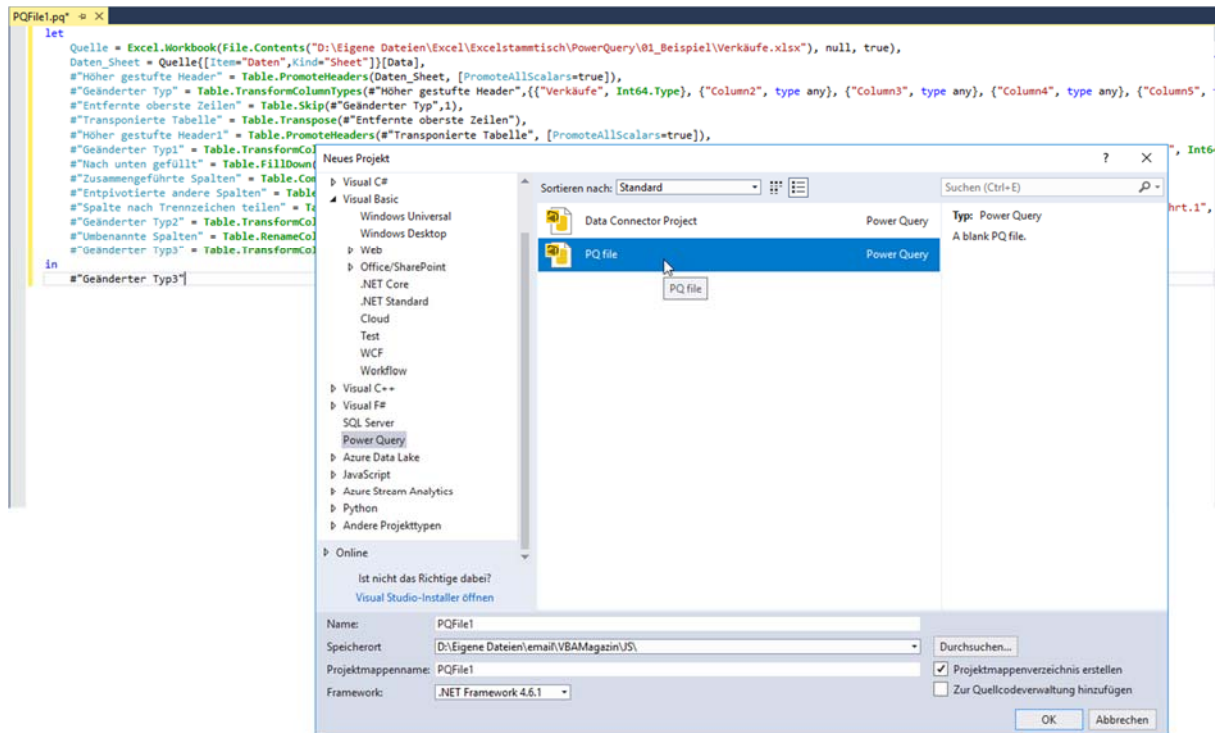
Und schließlich die Funktionen, die einen Zugriff auf Excelmappen, auf XML- oder JSON-Dateien, auf Datenbanken, ... ermöglichen:

*Excel.Workbook*, *Xml.Document*, *Sql.Database*, *OleDb.Query*, ...

Während PowerBI IntelliSense zur Verfügung stellt, während es für Visual Studio ein Tool gibt, das die Codeeingabe erleichtert:

<https://marketplace.visualstudio.com/items?itemName=Dakahn.PowerQuerySDK>

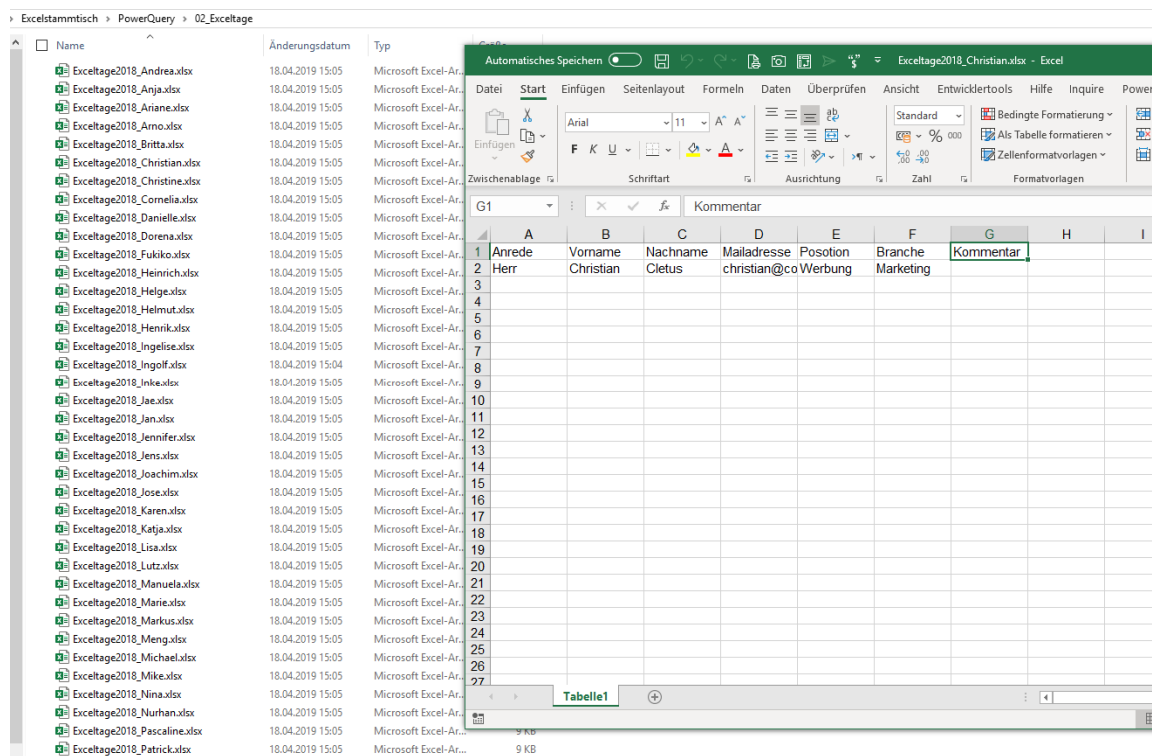




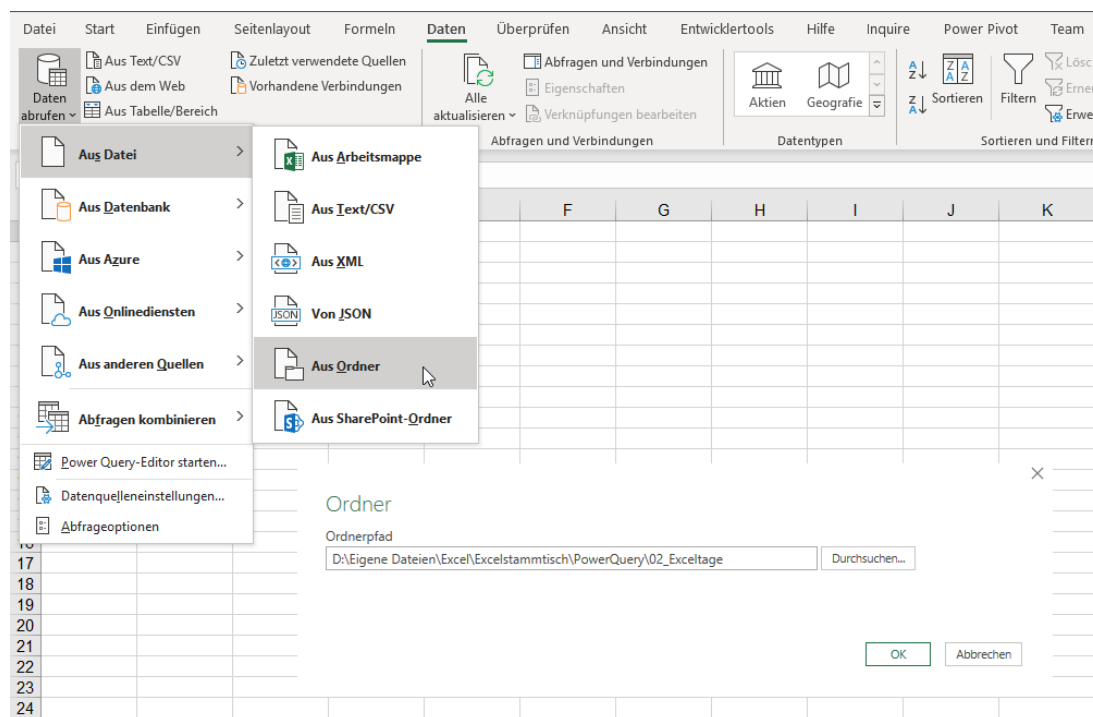
## 4. Beispiel 2: Zugriff auf Ordner

Power Query kann auch auf sämtliche Dateien eines Ordners zugreifen.

In einem Ordner liegen mehrere Arbeitsmappen, die alle den gleichen Aufbau besitzen:



Auf sie wird zugegriffen:



Da die Dateien die gleiche Struktur haben, kann der Content ermittelt werden:

Folder.Files("D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\02\_Exceltage")

**Content**

	Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
1	Binary	Exceltage2018_Andrea.xlsx	.xlsx	01.05.2019 21:48:20	18.04.2019 15:05:07	01.05.2019 21:48:20	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuer
2	Binary	Exceltage2018_Anja.xlsx	.xlsx	01.05.2019 21:48:20	18.04.2019 15:05:08	01.05.2019 21:48:20	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuer
3	Binary	Exceltage2018						Excelstammtisch\PowerQuer
4	Binary	Exceltage2018						Excelstammtisch\PowerQuer
5	Binary	Exceltage2018						Excelstammtisch\PowerQuer
6	Binary	Exceltage2018						Excelstammtisch\PowerQuer
7	Binary	Exceltage2018						Excelstammtisch\PowerQuer
8	Binary	Exceltage2018						Excelstammtisch\PowerQuer
9	Binary	Exceltage2018						Excelstammtisch\PowerQuer
10	Binary	Exceltage2018						Excelstammtisch\PowerQuer
11	Binary	Exceltage2018						Excelstammtisch\PowerQuer
12	Binary	Exceltage2018						Excelstammtisch\PowerQuer
13	Binary	Exceltage2018						Excelstammtisch\PowerQuer
14	Binary	Exceltage2018						Excelstammtisch\PowerQuer
15	Binary	Exceltage2018						Excelstammtisch\PowerQuer
16	Binary	Exceltage2018						Excelstammtisch\PowerQuer
17	Binary	Exceltage2018						Excelstammtisch\PowerQuer
18	Binary	Exceltage2018						Excelstammtisch\PowerQuer
19	Binary	Exceltage2018						Excelstammtisch\PowerQuer
20	Binary	Exceltage2018						Excelstammtisch\PowerQuer
21	Binary	Exceltage2018						Excelstammtisch\PowerQuer
22	Binary	Exceltage2018						Excelstammtisch\PowerQuer
23	Binary	Exceltage2018						Excelstammtisch\PowerQuer
24	Binary	Exceltage2018						Excelstammtisch\PowerQuer
25	Binary	Exceltage2018						Excelstammtisch\PowerQuer
26	Binary	Exceltage2018						Excelstammtisch\PowerQuer
27	Binary	Exceltage2018						Excelstammtisch\PowerQuer
28	Binary	Exceltage2018						Excelstammtisch\PowerQuer
29	Binary	Exceltage2018						Excelstammtisch\PowerQuer
30	Binary	Exceltage2018						Excelstammtisch\PowerQuer
31	Binary	Exceltage2018						Excelstammtisch\PowerQuer
32	Binary	Exceltage2018						Excelstammtisch\PowerQuer
33	Binary	Exceltage2018						Excelstammtisch\PowerQuer
34	Binary	Exceltage2018						Excelstammtisch\PowerQuer
35	Binary	Exceltage2018						Excelstammtisch\PowerQuer
36	Binary	Exceltage2018_Nurhan.xlsx	.xlsx	01.05.2019 21:48:20	18.04.2019 15:05:09	01.05.2019 21:48:20	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuer
37	Binary	Exceltage2018_Pascaline.xlsx	.xlsx	01.05.2019 21:48:20	18.04.2019 15:05:08	01.05.2019 21:48:20	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuer
38	Binary	Exceltage2018_Patrick.xlsx	.xlsx	01.05.2019 21:48:20	18.04.2019 15:05:09	01.05.2019 21:48:20	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuer

**Daten kombinieren**

Wählen Sie das aus jeder Datei zu extrahierende Objekt aus. Weitere Informationen

Beispieldatei: Erste Datei

Anzeigeoptionen \*

Beispieldateiparameter2 [1]

Tabelle1

Anrede	Vorname	Nachname	Mailadresse	Posotion	Branche
Frau	Andrea	Dunker	andrea@contoso.com	COO	Betrieb

☐ Dateien mit Fehlern überspringen

OK Abbrechen

Die erste Spalte wird gelöscht, die Daten zurück nach Excel geschrieben. Werden nun weitere Dateien eingefügt, kann die Tabelle aktualisiert werden:

	A	B	C	D	E	F	G	H
	Anrede	Vorname	Nachname	Mailadresse	Posotion	Branche	Kommentar	
2	Frau	Andrea	Dunker	andrea@contoso.com	COO	Betrieb		
3	Frau	Anja	Richter	ania@contoso.com	Techniker	Forschung u. Entwicklung		
4	Frau	Ariane			Geschäftsführungsassistent	Vertrieb		
5	Herr	Arno			Leiter Dateneingabe	Forschung u. Entwicklung		
6	Frau	Britta			Medizinische Beratung	Marketing		
7	Herr	Christian			Werbung	Marketing		
8	Frau	Christine			VP Verkauf	Vertrieb		
9	Frau	Cornelia			Direktor klinische Entwicklung	Forschung u. Entwicklung		
10	Frau	Danielle			Phase I-Versuche	Forschung u. Entwicklung		
11	Frau	Dorena			IIS-Manager	Forschung u. Entwicklung		
12	Herr	Fukiko			Konstruktion	Forschung u. Entwicklung		
13	Herr	Heinrich			Senior VP Sales & Marketing	Vertrieb		
14	Herr	Helge			Entwicklungsleiter	Forschung		
15	Herr	Helmut			VP Neuentwicklungen Pharmazie	Forschung u. Entwicklung		
16	Herr	Henrik			Forschungsplanung	Forschung		
17	Frau	Ingelise			Marketingdirektor	Marketing		
18	Herr	Ingolf			Vorsitzender & Geschäftsführer	Büro des Vorsitzenden		
19	Frau	Inke			Finanzdirektor	Finanzen		
20	Frau	Jae			Databankmanagement	Forschung u. Entwicklung		
21	Herr	Jan			Verkauf SA	Vertrieb		
22	Frau	Jennifer			Produktverwaltung	Forschung u. Entwicklung		
23	Herr	Jens			Fertigungsstrategie	Forschung u. Entwicklung		
24	Herr	Joachim			Verkauf - Europa	Vertrieb		
25	Herr	Jose			Phase III-Versuche	Forschung u. Entwicklung		
26	Frau	Karen			Qualitätssicherung	Forschung u. Entwicklung		
27	Frau	Katja			Phase IV -Versuche	Forschung u. Entwicklung		
28	Frau	Lisa			Produktmanagement	Marketing		
29	Herr	Lutz			Entwicklungsleiter	Forschung		
30	Frau	Manuela			Senior VP Forschung u. Entwicklung	Forschung u. Entwicklung		
31	Frau	Marie			Forscher	Forschung		
32	Herr	Markus			Forscher	Forschung		
33	Frau	Meng			Forscher	Forschung		
34	Herr	Michael			Pre-Market-Reporting	Forschung u. Entwicklung		
35	Herr	Mike			VP Fertigung	Forschung u. Entwicklung		

## 5. Beispiel 3: Zugriff auf Ordner -Teil II

Schwieriger gestaltet sich der Zugriff allerdings, wenn die Arbeitsmappen nicht gleichmäßig geformt sind, beispielsweise, wenn sie unterschiedliche Tabellenblattnamen tragen:

	Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
1	Binary	Callcenter_Auenland.xlsx	.xlsx	05.05.2019 23:32:22	05.05.2019 23:32:22	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\
2	Binary	Callcenter_Entenhausen.xlsx	.xlsx	05.05.2019 23:33:14	05.05.2019 23:33:14	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\
3	Binary	Callcenter_Fantasia.xlsx	.xlsx	05.05.2019 23:31:43	05.05.2019 23:31:43	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\
4	Binary	Callcenter_Lummerland.xlsx	.xlsx	05.05.2019 23:31:11	05.05.2019 23:31:11	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\
5	Binary	Kleinbonum.xlsx	.xlsx	03.05.2019 18:17:02	18.11.2018 16:38:05	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\
6	Binary	rene.jpg	.jpg	03.05.2019 18:17:02	18.11.2018 17:38:29	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\
7	Binary	ServiceID_CallCenter.txt	.txt	03.05.2019 18:17:02	18.11.2018 16:38:05	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\
8	Binary	tasten_Excel_Rene.pdf	.pdf	03.05.2019 18:17:02	06.02.2015 17:25:11	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\
9	Binary	Callcenter_Elysum.xlsx	.xlsx	05.05.2019 23:35:12	05.05.2019 23:35:12	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\Unite...

### Schritt 1: Die Endung in Großbuchstaben transformieren und filtern

*Transformieren / Textspalte / Format / GROSSBUCHSTABEN*

### Schritt 2: Der Unterordner wird ausgeschlossen

### Schritt 3: Nur die Dateien, die mit dem Text „Callcenter“ beginnen werden gefiltert.

	Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
1	Binary	Callcenter_Auenland.xlsx	.XLSX	05.05.2019 23:32:22	05.05.2019 23:32:22	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\
2	Binary	Callcenter_Entenhausen.xlsx	.XLSX	05.05.2019 23:33:14	05.05.2019 23:33:14	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\
3	Binary	Callcenter_Fantasia.xlsx	.XLSX	05.05.2019 23:31:43	05.05.2019 23:31:43	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\
4	Binary	Callcenter_Lummerland.xlsx	.XLSX	05.05.2019 23:31:11	05.05.2019 23:31:11	03.05.2019 18:17:02	Record	D:\Eigene Dateien\Excel\Excelstammtisch\PowerQuery\03_Orte\

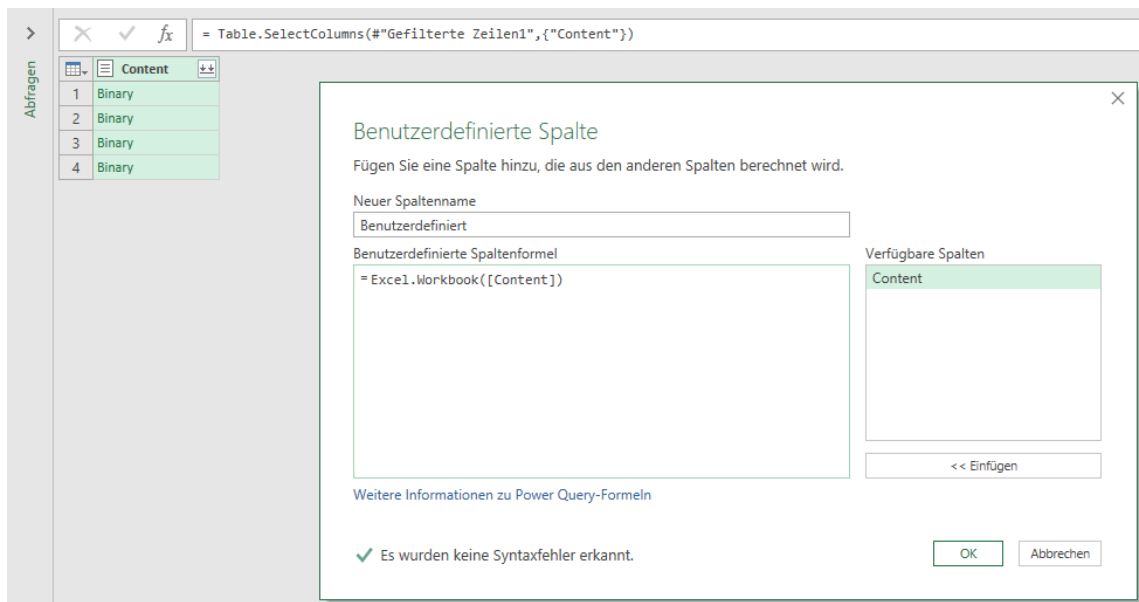
### Schritt 4: Die Spalten, außer „Content“ werden entfernt

### Schritt 5: Die Daten „herauslösen“

Der Befehl

```
=Excel.Workbook([Content])
```

extrahiert die Daten



## Schritt 6: Die Tabelle wird extrahiert

Das Ergebnis:

Table.ExpandTableColumn(#"Erweiterte Benutzerdefiniert", "Data", {"Column1", "Column2", "Column3", "Column4", "Column5", "Column6", "Column7"})

Content	Name	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Item	Kind	Hidden
1	Auenland	CallCenter Auenland - Anzahl Kundenkontakte KW 21 / 2018								Auenland	Sheet
2	Auenland									Auenland	Sheet
3	Auenland	ServiceID	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag		Auenland	Sheet
4	Auenland		1	11		11	15	14		Auenland	Sheet
5	Auenland		2	11		11	14	17		Auenland	Sheet
6	Auenland		8	11		17	14	11		Auenland	Sheet
7	Auenland		7	11		12	11	12		Auenland	Sheet
8	Auenland		11	3		1	1	2		Auenland	Sheet
9	Auenland		3	46		10	10	13		Auenland	Sheet
10	Auenland		4	11		11	13	19		Auenland	Sheet
11	Auenland		4	19		10	12	12		Auenland	Sheet
12	Auenland		9	19		15	19	10		Auenland	Sheet
13	Auenland		10	12		14	12	12		Auenland	Sheet
14	Auenland		12	10		11	11	15		Auenland	Sheet
15	Entenhausen	CallCenter Entenhausen - Anzahl Kundenkontakte KW 21 / 2018								Entenhausen	Sheet
16	Entenhausen									Entenhausen	Sheet
17	Entenhausen	ServiceID	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag		Entenhausen	Sheet
18	Entenhausen		1			10	17	14		Entenhausen	Sheet
19	Entenhausen		2			18	10	12		Entenhausen	Sheet
20	Entenhausen		8			16	15	19		Entenhausen	Sheet
21	Entenhausen		7			15	18	17		Entenhausen	Sheet
22	Entenhausen		11			4	6	2		Entenhausen	Sheet
23	Entenhausen		3			27	17	10		Entenhausen	Sheet
24	Entenhausen		4			10	17	11		Entenhausen	Sheet
25	Entenhausen		5			12	19	18		Entenhausen	Sheet
26	Entenhausen		6			16	11	12		Entenhausen	Sheet
27	Entenhausen		9			18	10	17		Entenhausen	Sheet
28	Entenhausen		10			11	18	19		Entenhausen	Sheet
29	Entenhausen		12			13	17	19		Entenhausen	Sheet
30	Fantasia	CallCenter Fantasia - Anzahl Kundenkontakte KW 21 / 2018								Fantasia	Sheet
31	Fantasia									Fantasia	Sheet

## Schritt 7: Weitere Transformationsschritte

Die nicht benötigte Spalte „Content“ wird gelöscht, ebenso die ersten beiden Zeilen. Die erste Zeile wird zur Überschrift umgewandelt, die erste Spalte wird in „Orte“ umbenannt. Nicht benötigte Zeilen werden gelöscht.

Table.TransformColumnTypes(#"Gefilterte Zeilen4",{{"ServiceID", Int64.Type}})

Orte	ServiceID	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Auenland_1
Auenland	1	11			11	15	14	
Auenland	2	11			14	11	17	
Auenland	8	11			17	14	11	
Auenland	7	11			12	11	12	
Auenland	11	3			1	1	2	
Auenland	3	46			10	10	13	
Auenland	4	11			11	13	19	
Auenland	4	19			10	12	12	
Auenland	9	19			15	19	10	
Auenland	10	12			14	12	12	
Auenland	12	10			11	11	15	
Entenhausen	1				10	17	14	
Entenhausen	2				18	10	12	
Entenhausen	8				16	15	19	
Entenhausen	7				15	18	17	
Entenhausen	11				4	6	2	
Entenhausen	3				27	17	10	
Entenhausen	4				10	17	11	
Entenhausen	5				12	19	18	
Entenhausen	6				16	11	12	
Entenhausen	9				18	10	17	
Entenhausen	10				11	18	19	
Entenhausen	12				13	17	19	
Fantasia	1				11	15	14	
Fantasia	2				14	11	17	
Fantasia	8				16	15	19	
Fantasia	7				15	18	17	
Fantasia	4				10	17	11	
Fantasia	5				12	19	18	
Fantasia	6				16	11	12	
Fantasia	9				13	10	17	
Fantasia	10				11	18	19	
Fantasia	12				11	11	15	
Lummerland	1	70	80	60	119	140	90	
Lummerland	2	55	140	162	90	60	60	
Lummerland	8	99	66	96	120	171	99	
Lummerland	7	70	190	135	180	170	72	

Die Spalten werden entpivotiert und die Informationen nach Excel geschrieben.

ABC 123	Orte	123 ServiceID	ABC Attribut	ABC 123 Wert
1	Auenland	1	Montag	11
2	Auenland	1	Mittwoch	11
3	Auenland	1	Donnerstag	15
4	Auenland	1	Freitag	14
5	Auenland	2	Montag	11
6	Auenland	2	Mittwoch	14
7	Auenland	2	Donnerstag	11
8	Auenland	2	Freitag	17
9	Auenland	8	Montag	11
10	Auenland	8	Mittwoch	17
11	Auenland	8	Donnerstag	14
12	Auenland	8	Freitag	11
13	Auenland	7	Montag	11
14	Auenland	7	Mittwoch	12
15	Auenland	7	Donnerstag	11
16	Auenland	7	Freitag	12
17	Auenland	11	Montag	3
18	Auenland	11	Mittwoch	1
19	Auenland	11	Donnerstag	1
20	Auenland	11	Freitag	2
21	Auenland	3	Montag	46
22	Auenland	3	Mittwoch	10
23	Auenland	3	Donnerstag	10
24	Auenland	3	Freitag	13
25	Auenland	4	Montag	11
26	Auenland	4	Mittwoch	11
27	Auenland	4	Donnerstag	13
28	Auenland	4	Freitag	19
29	Auenland	4	Montag	19
30	Auenland	4	Mittwoch	10
31	Auenland	4	Donnerstag	12
32	Auenland	4	Freitag	12
33	Auenland	9	Montag	19
34	Auenland	9	Mittwoch	15
35	Auenland	9	Donnerstag	19

Problem: Die Aktualisierung schlägt fehl, wenn die „oberste“ Datei gelöscht oder umbenannt wurde:

The screenshot shows an Excel spreadsheet with a table named 'Orte' containing columns 'ServiceID', 'Attribut', and 'Wert'. A right-click context menu is open over the table, with the 'Aktualisieren' (Refresh) option highlighted. An error dialog box from Microsoft Excel is displayed, stating: '[Expression.Error] Die Spalte "Auenland" der Tabelle wurde nicht gefunden.' (The column 'Auenland' of the table was not found.)

Der Grund: In der Zeile

```
= Table.RenameColumns("#Höher gestufte Header",{{"Auenland", "Orte"}})
```

steht hart codiert der Name der Tabelle. Diese Zeile muss ersetzt werden durch:



```
#"Umbenannte Spalten" = Table.RenameColumns("#Höher gestufte Header", {{Table.ColumnNames("#Höher gestufte Header") {0}, "Orte"}}),
```

Dann funktioniert die Aktualisierung – unabhängig von den Namen der Dateien.

## 6. Beispiel 4: Zugriff auf SAP-Daten oder falsches Minuszeichen

Ein Download aus einer Datenbank liefert Daten, bei denen das Minuszeichen hinter der Zahl steht:

Betrag in €	Währg	Betr. in H'	HWähr	Text
6.761,73	EUR	6.761,73	EUR	RECHNUNG ABC 812500
212,33-	EUR	212,33-	EUR	ABC Rep.
202,60-	EUR	202,60-	EUR	ABC Rep.
305,84-	EUR	305,84-	EUR	ABC Rep.
479,22-	EUR	479,22-	EUR	ABC Rep.
418,05-	EUR	418,05-	EUR	ABC Rep.
34.984,00	EUR	34.984,00	EUR	WM-Tickets Coop
416,88-	EUR	416,88-	EUR	ABC Rep.
286,36-	EUR	286,36-	EUR	ABC Rep.
253,25-	EUR	253,25-	EUR	ABC Rep.
171,43-	EUR	171,43-	EUR	ABC Rep.
253,25-	EUR	253,25-	EUR	ABC Rep.
405,19-	EUR	405,19-	EUR	ABC Rep.
279,74-	EUR	279,74-	EUR	ABC Rep.
93,50-	EUR	93,50-	EUR	ABC Rep.

Das kann schnell mit Power Query bereinigt werden:

Der Import sieht folgendermaßen aus:

Die bereinigten Daten (Zeilen und Spalten wurden entfernt, die Zahlen in Text transformiert):

= Table.TransformColumnTypes("#Entfernte Spalten",{{" Betrag in BW", type text}})						
ABC 123 Zuordnung	Belegdatum	Buch.dat.	A <sup>B</sup> <sub>C</sub> Betrag in BW	A <sup>B</sup> <sub>C</sub> Währg	A <sup>B</sup> <sub>C</sub> Text	
1 E9409332	29.05.2016	15.06.2016	6761,73	EUR	RECHNUNG ABC 812500	
2 811879	01.06.2016	30.06.2016	212,33-	EUR	ABC Rep.	
3 811875	01.06.2016	30.06.2016	202,60-	EUR	ABC Rep.	
4 811874	01.06.2016	30.06.2016	305,84-	EUR	ABC Rep.	
5 811890	01.06.2016	30.06.2016	479,22-	EUR	ABC Rep.	
6 811867	01.06.2016	30.06.2016	418,05-	EUR	ABC Rep.	
7 101768	01.06.2016	30.06.2016	34984	EUR	WM-Tickets Coop	
8 811868	01.06.2016	30.06.2016	416,88-	EUR	ABC Rep.	
9 811893	01.06.2016	30.06.2016	286,36-	EUR	ABC Rep.	
10 811917	01.06.2016	30.06.2016	253,25-	EUR	ABC Rep.	
11 811918	01.06.2016	30.06.2016	171,43-	EUR	ABC Rep.	
12 811920	01.06.2016	30.06.2016	253,25-	EUR	ABC Rep.	
13 811934	01.06.2016	30.06.2016	405,19-	EUR	ABC Rep.	
14 811956	01.06.2016	30.06.2016	279,74-	EUR	ABC Rep.	
15 811892	01.06.2016	30.06.2016	93,50-	EUR	ABC Rep.	
16 811869	01.06.2016	30.06.2016	284,42-	EUR	ABC Rep.	
17 811870	01.06.2016	30.06.2016	307,79-	EUR	ABC Rep.	
18 811872	01.06.2016	30.06.2016	214,28-	EUR	ABC Rep.	
19 811880	01.06.2016	30.06.2016	401,30-	EUR	ABC Rep.	
20 811873	01.06.2016	30.06.2016	54,55-	EUR	ABC Rep.	
21 811882	01.06.2016	30.06.2016	208,44-	EUR	ABC Rep.	
22 811891	01.06.2016	30.06.2016	299,99-	EUR	ABC Rep.	
23 811889	01.06.2016	30.06.2016	227,92-	EUR	ABC Rep.	
24 E9409159	01.06.2016	12.06.2016	35,72-	EUR	RECHNUNG ABC 811886	
25 E9409158	01.06.2016	12.06.2016	45,45-	EUR	RECHNUNG ABC 811884	
26 E9409157	01.06.2016	12.06.2016	113,64-	EUR	RECHNUNG ABC 811883	
27 E9409156	01.06.2016	12.06.2016	353,89-	EUR	RECHNUNG ABC 811878	
28 E9409155	01.06.2016	12.06.2016	107,15-	EUR	RECHNUNG ABC 811876	

Eine WENN-Funktion hilft beim Erzeugen des Minus am Anfang in einer neuen, benutzerdefinierten Spalte:

```
= if Text.EndsWith([#" Betrag in BW"], "-") then
  "-" & Text.Replace([#" Betrag in BW"], "-", "")
else
  [#" Betrag in BW"]
```

Die Texte werden in Dezimalzahlen transformiert:

A <sup>B</sup> <sub>C</sub> Betrag in BW	A <sup>B</sup> <sub>C</sub> Währg	A <sup>B</sup> <sub>C</sub> Text	1.2 Währung_korrekt
6761,73	EUR	RECHNUNG ABC 812500	6761,73
212,33-	EUR	ABC Rep.	-212,33
202,60-	EUR	ABC Rep.	-202,6
305,84-	EUR	ABC Rep.	-305,84
479,22-	EUR	ABC Rep.	-479,22
418,05-	EUR	ABC Rep.	-418,05
34984	EUR	WM-Tickets Coop	34984
416,88-	EUR	ABC Rep.	-416,88
286,36-	EUR	ABC Rep.	-286,36
253,25-	EUR	ABC Rep.	-253,25
171,43-	EUR	ABC Rep.	-171,43
253,25-	EUR	ABC Rep.	-253,25
405,19-	EUR	ABC Rep.	-405,19
279,74-	EUR	ABC Rep.	-279,74
93,50-	EUR	ABC Rep.	-93,5
284,42-	EUR	ABC Rep.	-284,42



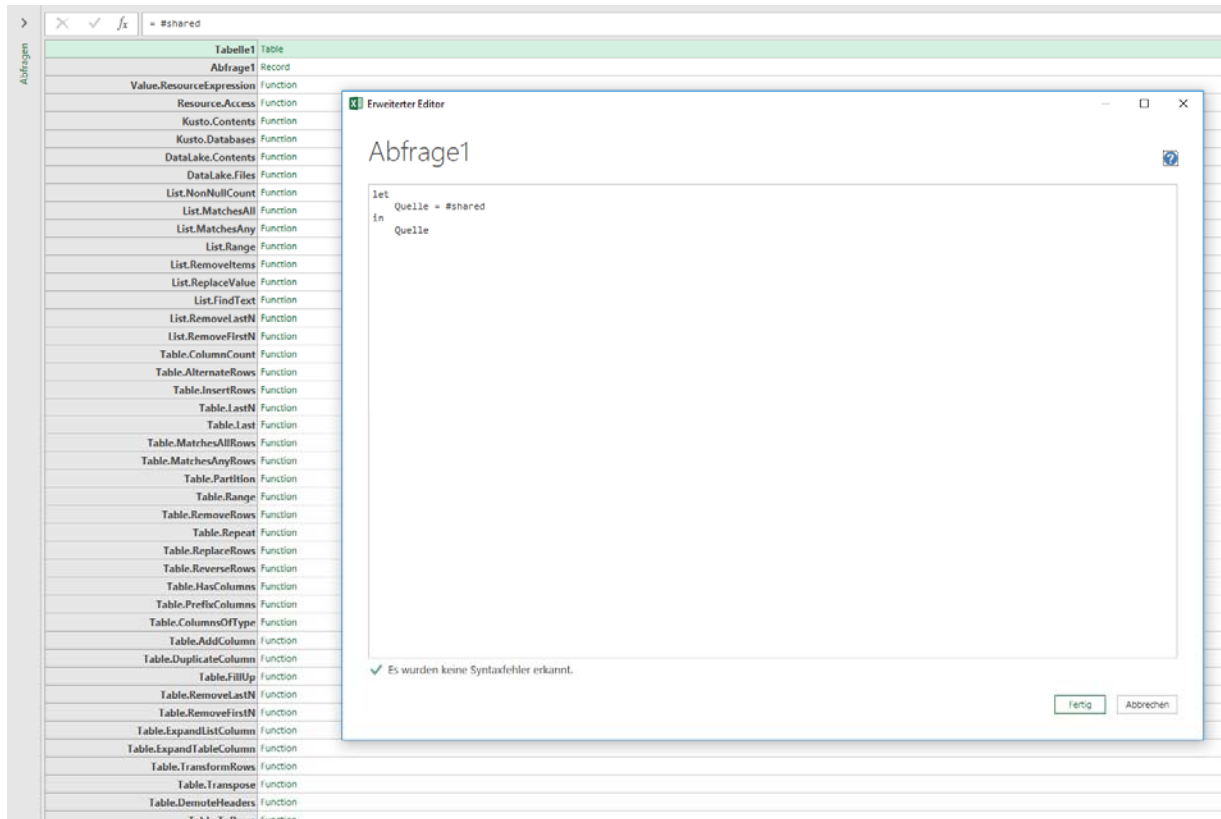
Ebenso kann in der ersten Spalte ein „E vor alle Zahlen hinzugefügt werden, die kein „E“ aufweisen:

```
if Text.Start([Zuordnung], 1) = "E" then
[Zuordnung]
else
"E" & [Zuordnung])
```

	A	B	C	D	E	F	G
1	ZuordnungE	Belegdatum	Buch.dat.	Währung_korrekt	Währg	Text	
2	E9409332	29.05.2016	15.06.2016	6761,73 EUR		RECHNUNG ABC 812500	
3	E811879	01.06.2016	30.06.2016	-212,33 EUR		ABC Rep.	
4	E811875	01.06.2016	30.06.2016	-202,6 EUR		ABC Rep.	
5	E811874	01.06.2016	30.06.2016	-305,84 EUR		ABC Rep.	
6	E811890	01.06.2016	30.06.2016	-479,22 EUR		ABC Rep.	
7	E811867	01.06.2016	30.06.2016	-418,05 EUR		ABC Rep.	
8	E101768	01.06.2016	30.06.2016	34984 EUR		WM-Tickets Coop	
9	E811868	01.06.2016	30.06.2016	-416,88 EUR		ABC Rep.	
10	E811893	01.06.2016	30.06.2016	-286,36 EUR		ABC Rep.	
11	E811917	01.06.2016	30.06.2016	-253,25 EUR		ABC Rep.	
12	E811918	01.06.2016	30.06.2016	-171,43 EUR		ABC Rep.	
13	E811920	01.06.2016	30.06.2016	-253,25 EUR		ABC Rep.	
14	E811934	01.06.2016	30.06.2016	-405,19 EUR		ABC Rep.	
15	E811956	01.06.2016	30.06.2016	-279,74 EUR		ABC Rep.	
16	E811892	01.06.2016	30.06.2016	-93,5 EUR		ABC Rep.	
17	E811869	01.06.2016	30.06.2016	-284,42 EUR		ABC Rep.	
18	E811870	01.06.2016	30.06.2016	-307,79 EUR		ABC Rep.	
19	E811872	01.06.2016	30.06.2016	-214,28 EUR		ABC Rep.	
20	E811880	01.06.2016	30.06.2016	-401,3 EUR		ABC Rep.	
21	E811873	01.06.2016	30.06.2016	-54,55 EUR		ABC Rep.	
22	E811882	01.06.2016	30.06.2016	-208,44 EUR		ABC Rep.	
23	E811891	01.06.2016	30.06.2016	-299,99 EUR		ABC Rep.	
24	E811889	01.06.2016	30.06.2016	-227,92 EUR		ABC Rep.	
25	E9409159	01.06.2016	12.06.2016	-35,72 EUR		RECHNUNG ABC 811886	
26	E9409158	01.06.2016	12.06.2016	-45,45 EUR		RECHNUNG ABC 811884	

## 7. Alle Funktionen von M

Die Liste sämtlicher Funktionen kann man sich mit dem Befehl #shared erzeugen:



## 8. Beispiel 5: Zugriff auf eine Internetseite

Am Beispiel der Seite

<https://www.consorsbank.de/ev/Wertpapierhandel/Kurse-Maerkte/Waehrungen>

The screenshot shows the Consorsbank website with two tables of exchange rates. The first table is titled 'Europa' and lists exchange rates for Denmark, Great Britain, Croatia, Norway, Sweden, Switzerland, and Turkey. The second table is titled 'Osteuropa' and lists exchange rates for Bulgaria, Latvia, Lithuania, Poland, Romania, Russia, Czech Republic, and Hungary. Each table has columns for Land, Kürzel, Kurs, Veränderung, Zeit, and Heute. The 'Veränderung' column shows percentage changes, and the 'Heute' column shows a bar chart representing the daily change.

Land	Kürzel	Kurs	Veränderung	Zeit	Heute
► Dänemark	EUR/DKK	7,4609	0,00 %	17:41	
▼ Großbritannien	EUR/GBP	0,8495	-0,92 %	03.05.	
► Kroatien	EUR/HRK	7,4153	0,00 %	17:58	
► Norwegen	EUR/NOK	9,7402	0,00 %	17:59	
► Schweden	EUR/SEK	10,6779	0,00 %	17:59	
▼ Schweiz	EUR/CHF	1,1374	-0,14 %	03.05.	
▼ Türkei	EUR/TRY	6,6645	-0,02 %	03.05.	

Land	Kürzel	Kurs	Veränderung	Zeit	Heute
► Bulgarien	EUR/BGN	1,9558	0,00 %	17:39	
► Lettland	EUR/LVL	0,7028	0,00 %	17:59	
► Litauen	EUR/LTL	3,4528	0,00 %	17:59	
▼ Polen	EUR/PLN	4,2734	-0,29 %	03.05.	
► Rumänien	EUR/RON	4,7415	0,00 %	17:59	
► Russland	EUR/RUB	72,8640	0,00 %	17:59	
► Tschechien	EUR/CZK	25,6830	0,00 %	17:34	
▼ Ungarn	EUR/HUF	322,6600	-0,46 %	03.05.	

soll exemplarisch der Zugriff auf eine Internetseite gezeigt werden:

Da sich auf der Internetseite mehrere Tabellen befinden, müssen diese ausgewählt werden:

Navigator

☒ Mehrere Elemente auswählen

Anzeigeoptionen ▾

📁 https://www.consorsbank.de/ev/Wertpapierha...

- ☐ Document
- ☒ Table 0
- ☒ Table 1
- ☒ Table 2
- ☒ Table 3
- ☒ Table 4
- ☒ Table 5

Tabellenansicht Webansicht

Table 0

Vorschau heruntergeladen am Mittwoch

Land	Kürzel	Kurs
Dänemark	Dänemark	EUR/DKK EURO / DÄNISCHE KRONE (EUR/DKK)
Großbritannien	Großb...annien	EUR/GBP EURO / BRITISCHES PFUND (EUR/GBP)
Kroatien	Kroatien	EUR/HRK EURO / KUNA (EUR/HRK)
Norwegen	Norwegen	EUR/NOK EURO / NORWEGISCHE KRONE (EUR/NOK)
Schweden	Schweden	EUR/SEK EURO / SCHWEDISCHE KRONE (EUR/SEK)
Schweiz	Schweiz	EUR/CHF EURO / SCHWEIZER FRANKEN (EUR/CHF)
Türkei	Türkei	EUR/TRY EURO / TÜRKISCHE LIRA (EUR/TRY)

Laden ▾ Daten transformieren Abbrechen

Table 0

Table 1

Table 2

Table 3

Table 4

Table 5

fx = Table.TransformColumnTypes(Data0,{{"Land", type text}, {"Kürzel", type text}, {"Kurs", type text}, {"in %", type number

	Land	Kürzel	Kurs	1.2 in %	% Veränderung	Zeit	Heute	
1	Dänemark	Dänemark	EUR/DKK	7,4657	0,00 %	21:41:00		Währung handeln
2	Großbritannien	Großb...annien	EUR/GBP	0,858	-0,24 %	22:07:00		Währung handeln
3	Kroatien	Kroatien	EUR/HRK	7,4153	0,00 %	21:58:00		Währung handeln
4	Norwegen	Norwegen	EUR/NOK	9,7102	0,31 %	21:59:00		Währung handeln
5	Schweden	Schweden	EUR/SEK	10,6757	0,26 %	21:59:00		Währung handeln
6	Schweiz	Schweiz	EUR/CHF	1,1395	-0,31 %	22:07:00		Währung handeln
7	Türkei	Türkei	EUR/TRY	6,6758	-0,22 %	22:06:00		Währung handeln

Die nicht benötigten Spalten werden gelöscht. Die Spalte, in der sich die Information der Währung befindet, wird aufgesplittet (lf = line feed = Zeilenschaltung):

Spalte nach Trennzeichen teilen

Geben Sie das Trennzeichen an, das zum Teilen der Textspalte verwendet werden soll.

Trennzeichen eingeben oder auswählen  
--Benutzerdefiniert--

#(if)

Aufteilen:

- ☐ Beim äußersten linken Trennzeichen
- ☐ Beim äußersten rechten Trennzeichen
- ☒ Bei jedem Vorkommen des Trennzeichens

Erweiterte Optionen

Aufteilen in

- ☒ Spalten
- ☐ Zeilen

Spaltenanzahl für Teilung  
7

Anführungszeichen  
"

☒ Mithilfe von Sonderzeichen teilen

Sonderzeichen einfügen

OK Abbrechen

Und erneut wird der Schrägstrich verwendet, um die Spalte zu teilen (trennen):

	Data.Land	Data.Kurs.1.2	Data.in %
1	Dänemark	DKK	7,4657
2	Großbritannien	GBP	0,8580
3	Kroatien	HRK	7,4153
4	Norwegen	NOK	9,7102
5	Schweden	SEK	10,6757
6	Schweiz	CHF	1,1395
7	Türkei	TRY	6,6758
8	Bulgarien	BGN	1,9558
9	Lettland	LVL	0,7028
10	Litauen	LTL	3,4528
11	Polen	PLN	4,2821
12	Rumänien	RON	4,7587
13	Russland	RUB	72,5679

Das Land wird sortiert. Da PowerQuery case sensitive ist wird die USA zwischen die Türkei und Ungarn einsortiert: Also: Türkei < USA < Ungarn.

Eine Lösung: Man ersetzt „USA“ durch „Usa“, sortiert und ersetzt anschließend „Usa“ in „USA“. Oder man ändert in M (im erweiterten Editor) den Sortierbefehl:

```
#"Sortierte Zeilen" = Table.Sort(#"Umbenannte Spalten",{ {each Text.Upper([Land]), Order.Ascending} })
```

Oder man transformiert alle Ländernamen in Großbuchstaben.

Ebenso kann nun der EUR als erste Zeile eingefügt werden:

```
= Table.InsertRows(#"Geänderter Typ2", 0, { [Land = "EURO-Länder", Währung = "EUR", Kurs = 1] })
```







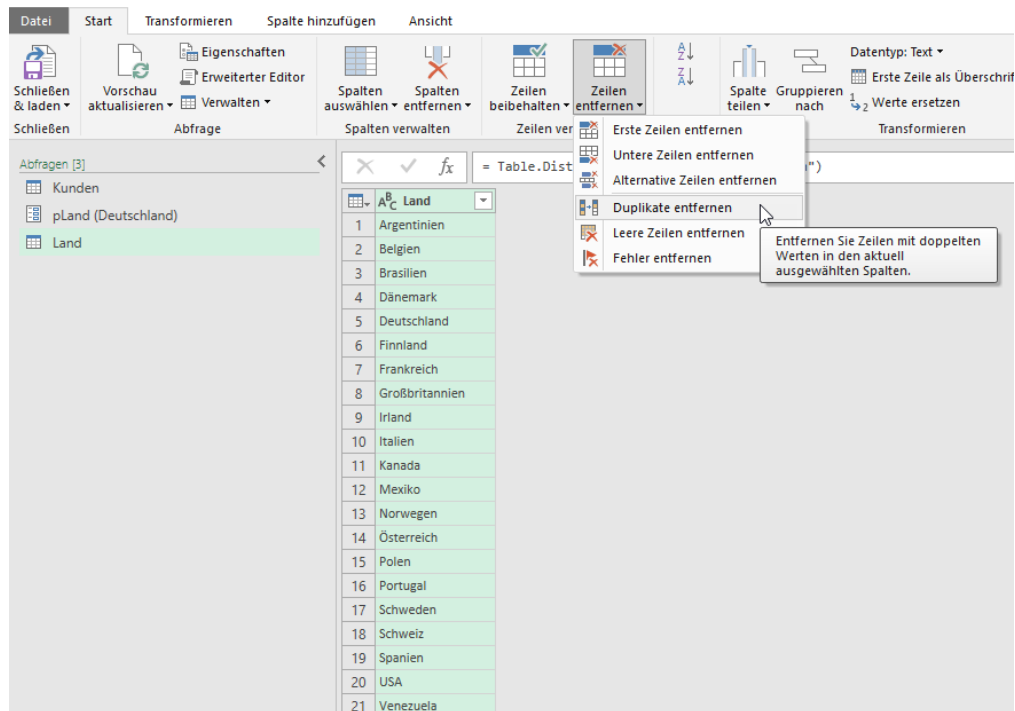
intaktperson	A <sup>0</sup> Position	A <sup>0</sup> Straße	A <sup>0</sup> Ort	A <sup>0</sup> Region	A <sup>0</sup> PLZ	A <sup>0</sup> Land	A <sup>0</sup> Telefon	A <sup>0</sup> Telefax
Anders	Vertriebsmitarbeiterin	Obere Str. 57	Berlin		12209	Deutschland	030-0074321	030-0076545
ujillo					05021	Mexiko	(5) 555-4729	(5) 555-3745
o Moreno					05023	Mexiko	(5) 555-3932	
is Hardy					WA1 1DP	Großbritannien	(71) 555-7788	(71) 555-6750
na Berglun					S-958 22	Schweden	0921-12 34 65	0921-12 34 67
Moos					68306	Deutschland	0621-08460	0621-08924
ique Citea					67000	Frankreich	88.60.15.31	88.60.15.32
i Sommer					28023	Spanien	(91) 555 22 82	(91) 555 91 99
ice Lebihar					13008	Frankreich	91.24.45.40	91.24.45.41
th Lincoln					T2F 8M4	Kanada	(604) 555-4729	(604) 555-3745
a Ashwort					EC2 SNT	Großbritannien	(71) 555-1212	
o Simpson					1010	Argentinien	(1) 135-5555	(1) 135-4892
ico Chang					05022	Mexiko	(5) 555-3392	(5) 555-7293
Vang					3012	Schweiz	0452-076545	
Afonso					05432-043	Brasilien	(11) 555-7647	
th Brown					WX1 6LT	Großbritannien	(71) 555-2282	(71) 555-9199
ittlieb					52066	Deutschland	0241-039123	0241-059428
Labruno					44000	Frankreich	40.67.88.88	40.67.89.89
svon					WX3 6FW	Großbritannien	(71) 555-0297	(71) 555-3373
I Mendel					8010	Österreich	7675-3425	7675-3426
uz					05442-030	Brasilien	(11) 555-9857	
Roel					28034	Spanien	(91) 555 94 44	(91) 555 55 93
e Rancé					59000	Frankreich	20.16.10.16	20.16.10.17
Larsson					S-844 67	Schweden	0695-34 67 21	
ranken					80805	Deutschland	089-0877310	089-0877451
Schmitt					44000	Frankreich	40.32.21.21	40.32.21.20
Accorti					10100	Italien	011-4988260	011-4988261
idriguez					1675	Portugal	(1) 354-2534	(1) 354-2535
lo Saavedr					08022	Spanien	(93) 203 4560	(93) 203 4561
sdro Freyr					41101	Spanien	(95) 555 82 82	
Fonseca					04876-786	Brasilien	(11) 555-9482	
d Snyder	Marketingmanager	2732 Baker Blvd.	Eugene	OR	97403	USA	(503) 555-7555	
l Pereira	Inhaber	5F Ave. Los Palos Grandes	Caracas	DF	1081	Venezuela	(2) 283-2951	(2) 283-3397
Pontes	Buchhalter	Rua do Paço, 67	Rio de Janeiro	RJ	05454-876	Brasilien	(21) 555-0091	(21) 555-8765
Hernández	Vertriebsmitarbeiter	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	5022	Venezuela	(5) 555-1340	(5) 555-1948
atimer	Vertriebsmitarbeiter	City Center Plaza 516 Main St.	Elgin	OR	97827	USA	(503) 555-6874	(503) 555-2376

Diesen kann beim Filtern verwendet werden:

Das bedeutet: Würde man den die Abfrage im Power Query-Editor öffnen, hätte man die Möglichkeit, den Wert des Parameters zu ändern. Eine gute, aber nicht praktikable Lösung.

Besser ist Folgende: In einer zweiten Abfragen werden eindeutige Werte erzeugt, indem die die Duplikate entfernt werden (*Start / Zeilen verringern / Zeilen entfernen / Duplikate entfernen*)





Diese Liste wird in Excel als Liste für ein Dropdownfeld in einer Datenüberprüfung verwendet:

N	O	P	Q
Land			Land
Argentinien			Argentinien
Belgien			Belgien
Brasilien			Brasilien
Dänemark			Dänemark
Deutschland			Deutschland
Finnland			Finnland
Frankreich			Frankreich
Großbritannien			Großbritannien
Irland			
Italien			
Kanada			
Mexiko			
Norwegen			
Österreich			
Polen			
Portugal			
Schweden			
Schweiz			
Spanien			
USA			
Venezuela			

Daraus wird eine intelligente Tabelle erstellt, die als weitere Abfrage verwendet wird. Diese wird als Filterkriterium verwendet:

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Position	Straße	Ort	Region	PLZ	Land	Telefon	Telefax			Land			Land
2	Marketingmanager	24, place Kléber	Strasbourg	67000	Frankreich	88 80 15 31	88 80 15 32				Argentinien			Frankreich
3	Inhaber	12, rue des Bouchers	Marseille	13008	Frankreich	91 24 45 40	91 24 45 41				Belgien			
4	Inhaberin	67, rue des Cinquante Otages	Nantes	44000	Frankreich	40 67 88 88	40 67 88 89				Brasilien			
5	Vertriebsagentassistent	184, chaussée de Tournai	Lille	59000	Frankreich	20 16 10 16	20 16 10 17				Dänemark			
6	Marketingmanager	54, rue Royale	Nantes	44000	Frankreich	40 32 21 21	40 32 21 20				Deutschland			
7	Vertriebsmitarbeiter	67, avenue de l'Europe	Versailles	78000	Frankreich	30 59 84 10	30 59 85 11				Finnland			
8	Vertriebsmanager	1 rue Alsace-Lorraine	Toulouse	31000	Frankreich	61 77 61 10	61 77 61 11				Frankreich			
9	Inhaberin	265, boulevard Charonne	Paris	75012	Frankreich	(1) 42 34 22 66	(1) 42 34 22 77				Großbritannien			
10	Marketingmanager	25, rue Lauriston	Paris	75016	Frankreich	(1) 47 55 60 10	(1) 47 55 60 20				Irland			
11	Vertriebsagent	2, rue du Commerce	Lyon	69004	Frankreich	78 32 54 88	78 32 54 87				Italien			
12	Buchhalter	59 rue de l'Abbaye	Reims	51100	Frankreich	26 47 15 10	26 47 15 11				Kanada			
13											Mexiko			

## 11. Beispiel 8: „Relativer“ Pfad

Beim Import aus einer Datei wird stets der Pfad absolut angegeben, also beispielsweise:

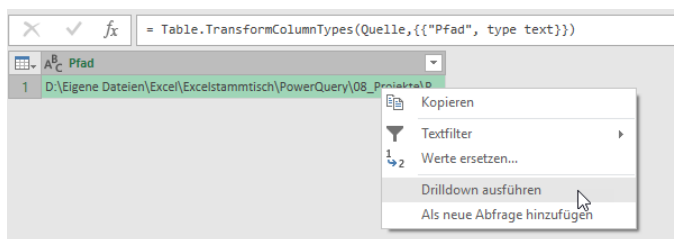
```
Quelle = Csv.Document(File.Contents("D:\Eigene
Dateien\Excel\Excelstammtisch\PowerQuery\08_Projekte\Projekt
Frankfurt\Daten\Kunden.csv"),[Delimiter=";", Columns=14,
Encoding=1252, QuoteStyle=QuoteStyle.None]),
```

Abhilfe schafft eine Excel-Funktion:

```
=LINKS(ZELLE("dateiname");FINDEN("[ ";ZELLE("dateiname"))-1)
```

Sie liefert den Pfad der aktuellen Datei. Wird aus dieser Information eine intelligente Tabelle erzeugt, kann diese in Power Query eingebunden werden.

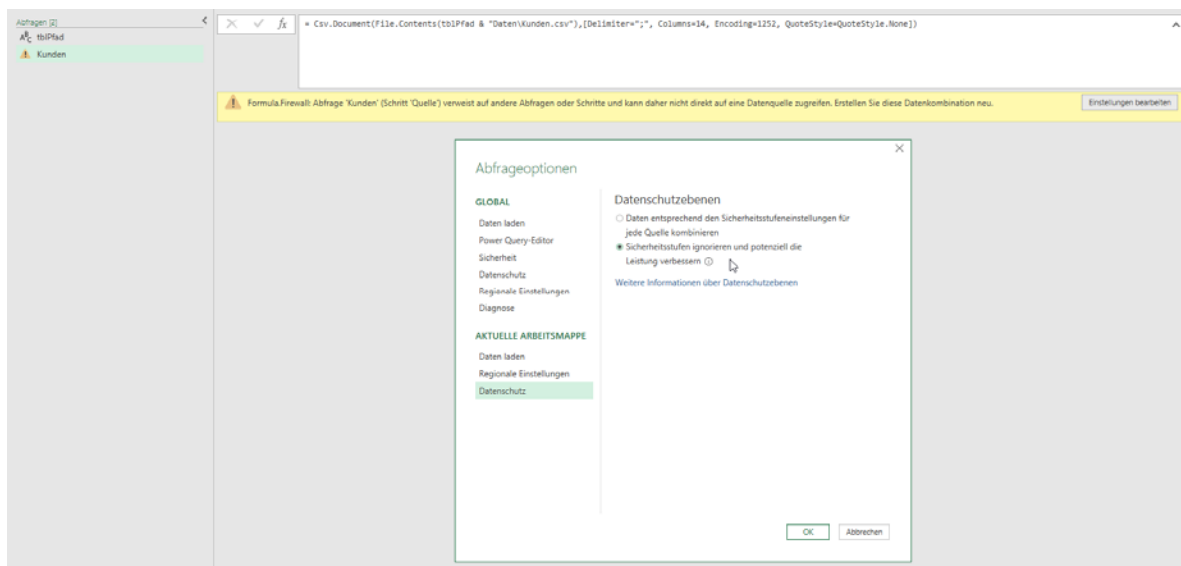
Auf diese Abfragetabelle wird ein Drilldown ausgeführt, so es sich um Text handelt:



Dieser kann nun in den Code eingefügt werden:

```
= Csv.Document(File.Contents(tblPfad &
"Daten\Kunden.csv"),[Delimiter=";", Columns=14, Encoding=1252,
QuoteStyle=QuoteStyle.None])
```

Allerdings müssen noch die Sicherheitseinstellungen geändert werden:



Damit ist die Datei nun „ordnerunabhängig“:

