

KAPITEL 8

Power Query



Power Query, das BI-Tool, das sowohl in Excel als auch für Power BI Desktop zur Verfügung steht, ist das Standardwerkzeug für die Automatisierung von ETL-Prozessen. Das Extrahieren, Transformieren und Laden externer Daten wird über die Skriptsprache M aufgezeichnet, und der Code steht während des Prozesses und nach Abschluss der letzten Aktion zur Verfügung. Ähnlich wie beim VBA-Makrorekorder zeichnet Power Query alle Zugriffe auf Pfade, Dateien oder Excel-Objekte absolut auf. Die Skriptsprache M bietet aber die Möglichkeit, variable Inhalte zu erstellen und mit Parametern zu arbeiten.

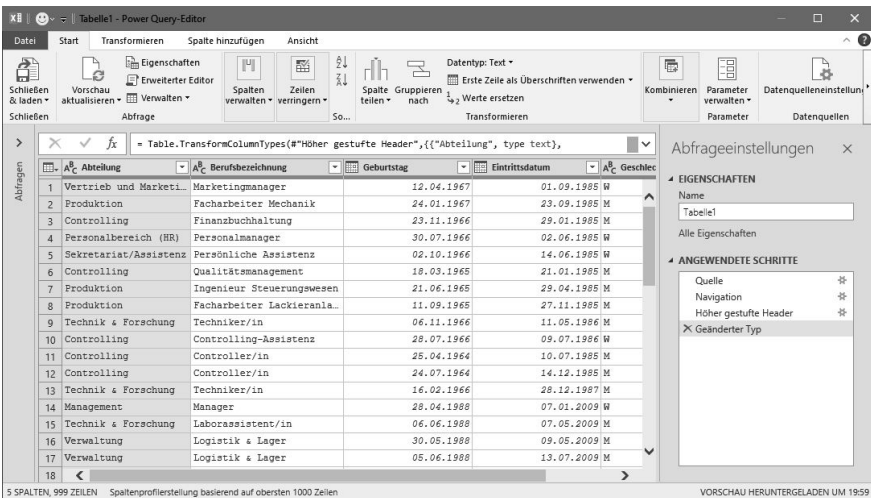


Abbildung 8.1: Der Power-Query-Editor

Power Query ersetzt den mühsamen und unflexiblen Text-Assistenten, der für die Transformation von Excel- oder CSV-Daten zuständig war. Der Editor kann aber weit mehr als Trennzeichen erkennen und Daten in Spalten aufteilen. Viele Textfunktionen wie GLÄTTEN und ERSETZEN haben Einzug gehalten, Funktionen wie *Nach unten ausfüllen* oder *Entpivotieren* machen das Werkzeug besonders flexibel.

8.1 Mit Parametertabellen in Power Query arbeiten

Nach dem ETL-Prinzip werden die Daten zuerst aus einer externen oder auch internen Quelle (Tabellen, Tabellenbereiche) geholt (extrahiert) und dann transferiert. Power Query zeichnet dabei bei externen Daten den Pfad und den Dateinamen auf. Namen von Tabellenblättern werden ebenfalls in Textform in den M-Code geschrieben.



Abbildung 8.2: Pfade und Dateinamen werden im M-Code fixiert.

Versuchen Sie, solche fixen Verweise auf Pfade, Dateien und Tabellenblätter aus dem M-Code herauszuhalten. Mit einer Parametertabelle bleibt der Zugriff auf externe Daten mit Power Query variabel, und mit dem Einsatz einer Datenüberprüfungsliste lässt er sich sogar gezielt steuern.

8.1.1 Das Beispiel: Lagerbestand

In diesem Praxisbeispiel liefert das ERP-System Artikellisten für Produkte, die das Unternehmen im Angebot hat. Jede einzelne Artikelkategorie steht in einem eigenen Tabellenblatt, die Arbeitsmappe heißt *Lagerbestand.xlsx*.

Art.Nr.	Bezeichnung	Kategorie
X101-001	OfficeConnect Fast Ethernet NIC	Netzwerk
X101-002	Fast Ethernet XL 10/100 PCI RJ45 boot	Netzwerk
X101-003	3Com USB Networking Interface 10/100	Netzwerk
X101-004	Fast Ethernet XL10/100 PCI NM-Bulk	Netzwerk
X101-005	ACER PQ	
X101-006	ACER NIC	
X101-007	ArLancer	
X101-008	ArLancer	
X101-009	HP 10/100	
X101-010	HP 3COM	
X101-011	IBM Ether	
X101-012	IBM Ether	

Art.Nr.	Bezeichnung	Kategorie
X101-013	CPU AMD Duron 700 MHz	Bauteile
X101-014	CPU AMD Duron 750 MHz	Bauteile
X101-015	CPU AMD Duron 800 MHz	Bauteile
X101-016	CPU AMD Duron 850 MHz	
X101-017	Celeron 766 Box FC-PGA/12	
X101-018	Celeron 800 Box FC-PGA/12	
X101-019	Celeron 850 Box FC-PGA/12	
X101-020	Pentium III 933 Box FC-PGA	

Art.Nr.	Bezeichnung	Kategorie
X101-021	ACER TM 1 HDD Wechselsplatte 12GB TM 600	Komponenten
X101-022	ACER TM Wechselsplatte 20GB f. TM 350	Komponenten
X101-023	HP HD 6.4GB IDE Disk Drive Ultra ATA/66	Komponenten
X101-024	HP TOP HD 9GB Ultra3 HDD 7.2K	Komponenten
X101-025	HD AL-6LE MAJ3091MP 9.1 GB Ultra160 68P	Komponenten
X101-026	DeLuxe Access Keyboard dt. PS2 W32	Komponenten
X101-027	Internet Keyboard dt. PS2 W32	Komponenten
X101-028	Cordless Desktop dt. PS2 W32	Komponenten
X101-029	Cordless Maus/Tastatur Mitsumi USB	Komponenten
X101-030	IBM Tastatur Preferred USB weiß deutsch	Komponenten
X101-031	EPSON Stylus Color 3000 400/600 Z/s A2	Komponenten
X101-032	EPSON Stylus Color 1160	Komponenten
X101-033	EPSON Stylus Photo 870	Komponenten
X101-034	CANON BJC-5500	Komponenten

Abbildung 8.3: Lagerbestand, verteilt auf mehrere Standorte

Erstellen Sie eine Power-Query-Abfrage auf die externe Quelle und holen Sie die Umsätze aus der ersten Region:

Drücken Sie **(Strg)+N** für eine neue Mappe und wählen Sie *Daten/Daten abrufen und transformieren/Daten abrufen/Aus Datei/Aus Arbeitsmappe*. Markieren Sie die Datei *Lagerbestand.xlsx* und wählen Sie *Importieren*.

Markieren Sie im Navigator das erste Tabellenblatt *Lager Nord* und wählen Sie *Daten transformieren*. Schalten Sie im Power-Query-Editor unter *Ansicht* die Abfrageeinstellungen und die Bearbeitungsleiste ein. Im Schritt-Editor werden die Schritte angezeigt, die Power Query verwendet, um die Daten zu extrahieren und zu transformieren.

Im letzten Schritt wandelt Power Query die Spaltentypen automatisch um, findet dabei aber nicht immer den richtigen Typ. Ändern Sie die Spaltentypen über das Typensymbol oder im Kontextmenü der rechten Maustaste mit *Typ ändern*. Markieren Sie die Schritte einzeln und sehen Sie sich den M-Code in der Bearbeitungsleiste an.

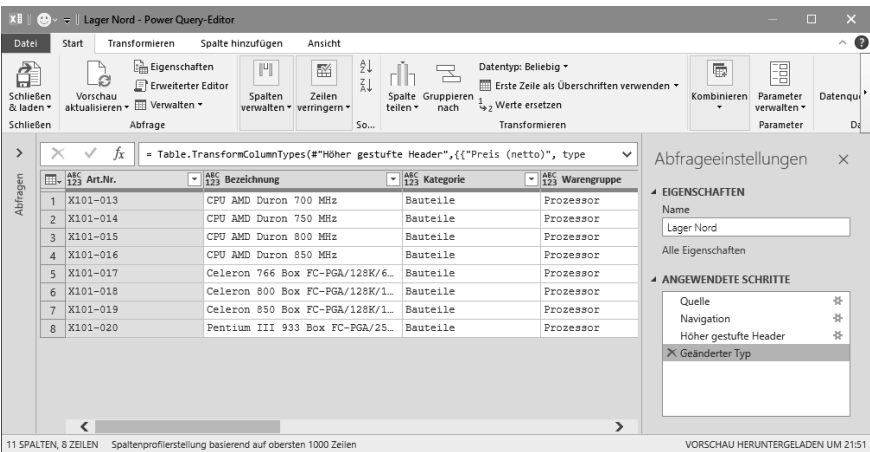


Abbildung 8.4: Die erste Abfrage auf den Lagerbestand

8.1.2 Parametertabelle anlegen

Der M-Code der Power-Query-Abfrage speichert, wie oben gezeigt, sowohl den Pfad als auch den Dateinamen und das Tabellenblatt in Textform. Wird die Quelle umbenannt oder an einen anderen Speicherort verschoben, bleibt das zuletzt abgefragte Ergebnis zwar erhalten, eine Aktualisierung der Abfrage führt aber zur Fehlermeldung.

Holen Sie das Abfrageergebnis in ein leeres Tabellenblatt, setzen Sie es in Zeile 5 ab. Legen Sie dann eine Parametertabelle an:

Geben Sie in Zeile 1 die Spaltenüberschriften *Parameter* und *Wert* ein und drücken Sie mit dem Zellzeiger in A1 **[Strg]+[T]**.

Die Daten für die Tabelle befinden sich in $\$A\$1:\$B\1 , markieren Sie die Option *Tabelle hat Überschriften*.

Tragen Sie in Spalte A die Parameter *Dateipfad*, *Dateiname* und *Tabellenblatt* ein.

Tragen Sie in B2 den Pfad *C:\Daten* ein, oder verwenden Sie den Pfad, in dem sich Ihre Auswertungsdatei befindet. Wenn die externe Datei im gleichen Pfad wie die Auswertungsdatei steht, können Sie diesen berechnen:

=LINKS(ZELLE("Dateiname";\$A\$1);FINDEN("[";ZELLE("Dateiname";\$A\$1))-1)

Schreiben Sie den Dateinamen *Lagerbestand.xlsx* in die Zelle B3.

Für die Auswahl des Tabellenblatts verwenden Sie eine Datenüberprüfungsliste. Wählen Sie *Daten/Datentools/Datenüberprüfung*. Schalten Sie unter *Zulassen* um auf *Liste* und geben Sie diese Quelle ein:

Lager Süd;Lager Nord;Zentrallager

	A	B	C	D	E
1	Parameter	Wert			
2	Dateipfad	C:\Users\ischels\Documents\MUT\Excel Zauberbuch\			
3	Dateiname	Lagerbestand.xlsx			
4	Tabellenblatt				
5		Lager Süd			
6		Lager Nord			
6		Zentrallager			
7	Art.Nr.	Bezeichnung	Kategorie	Warengruppe	Hersteller
8	X101-013	CPU AMD Duron 700 MHz	Bauteile	Prozessor	AMD
9	X101-014	CPU AMD Duron 750 MHz	Bauteile	Prozessor	AMD
10	X101-015	CPU AMD Duron 800 MHz	Bauteile	Prozessor	AMD
11	X101-016	CPU AMD Duron 850 MHz	Bauteile	Prozessor	AMD
12	X101-017	Celeron 766 Box FC-PGA/128K/66FSB	Bauteile	Prozessor	INTEL
13	X101-018	Celeron 800 Box FC-PGA/128K/100FSB	Bauteile	Prozessor	INTEL
14	X101-019	Celeron 850 Box FC-PGA/128K/100FSB	Bauteile	Prozessor	INTEL
15	X101-020	Pentium III 933 Box FC-PGA/256K/133FSB	Bauteile	Prozessor	INTEL

Abbildung 8.5: Parametertabelle mit Datenüberprüfungsliste für das Tabellenblatt

Größere Auswahllisten schreiben Sie in eine freie Spalte, weisen den Daten den Bereichsnamen *Regionsliste* zu und verwenden diese als Quelle:

=Regionsliste

Damit ist die Parameterliste erstellt, weisen Sie der Tabelle noch über den Tabellenentwurf den Tabellennamen *tbl_Parameter* zu. Speichern Sie die Auswertungsmappe unter dem Pfad, in dem sich Ihre Datenquelle für die Power-Query-Abfrage befindet.

8.1.3 Parametertabelle in Power Query einbinden

Im nächsten Schritt holen Sie die neue Parametertabelle in den Power-Query-Editor und verknüpfen Dateipfad, Dateiname und Tabellenblatt mit den Quellangaben für die Umsatzabfrage.

Setzen Sie den Zellzeiger in die Parametertabelle und wählen Sie *Daten/Daten abrufen und transformieren/Aus Tabelle/Bereich*.

Weisen Sie den beiden Spalten den Datentyp *Text* zu. Filtern Sie über den Textfilter den Dateipfad. Klicken Sie auf den angezeigten Wert und wählen Sie im Kontextmenü der rechten Maustaste *Drilldown ausführen*. Geben Sie der Abfrage die Bezeichnung *Dateipfad*.

Duplizieren Sie in der linken Abfrageliste die Abfrage (im Kontextmenü). Geben Sie dieser Abfrage die Bezeichnung *Dateiname*. Suchen Sie unter *Angewendete Schritte* den Schritt *Gefilterte Zeilen*. Ändern Sie das Filterkriterium von *Dateipfad* auf *Dateiname*.

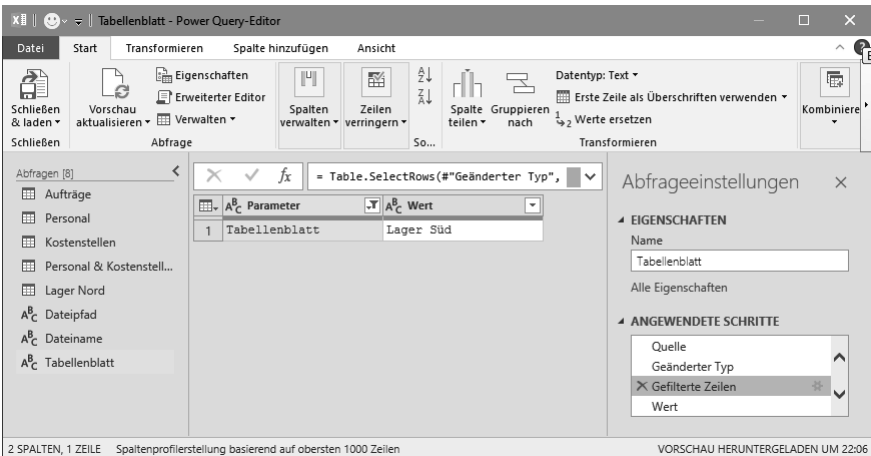


Abbildung 8.6: Variablen für Dateipfad, Dateiname und Tabellenblatt als Abfragen

Duplizieren Sie die Abfrage noch einmal, nennen Sie das Duplikat *Tabellenblatt*, und ändern Sie den Filter von *Dateiname* in *Tabellenblatt*. Jetzt sollte der Abfrage-Editor die drei Variablen *Dateipfad*, *Dateiname* und *Tabellenblatt* enthalten.

Achten Sie auf die Schreibweise der Variablen. Power Query unterscheidet zwischen Groß- und Kleinschreibung. Wenn Sie die Abfrage *DateiPfad* nennen, muss sie in dieser Schreibweise auch im M-Code verwendet werden.

8.1.4 Abfragevariablen in M-Code einbinden

Holen Sie die per Filter und Drilldown erstellten Variablen jetzt in die zuvor erstellte Abfrage. Geben Sie dieser die Bezeichnung *Lagerabfrage* (vorher: *Lager Nord*). Für die Änderungen können Sie die ersten beiden Schritte markieren und den Code in der Bearbeitungsleiste ändern oder unter *Ansicht/Weitere/Erweiterter Editor* den gesamten M-Code der Abfrage öffnen und editieren.

Ersetzen Sie im Schritt *Quelle* den Dateipfad und den Dateinamen durch eine Verkettung der beiden Variablen mit dem &-Zeichen:

Alt: = Excel.Workbook(File.Contents("C:\Daten\Lagerbestand.xlsx"), null, true)
 Neu: = Excel.Workbook(File.Contents(Dateipfad&Dateiname), null, true)

Ersetzen Sie im Schritt *Navigation* den Namen des Tabellenblatts durch die Variable:

Alt: = Quelle[[Item="Lager Nord",Kind="Sheet"]][Data]
 Neu: = Quelle[[Item=Tabellenblatt , Kind="Sheet"]][Data]



Abbildung 8.7: Die Parameter im M-Code

Wählen Sie *Schließen und Laden* im ersten Register (*Home* oder *Start*), um alle Abfrageergebnisse nach Excel zu holen.

Die Tabellenblätter, die damit für die drei Variablen angelegt werden, können Sie löschen, dadurch werden diese Abfragen auf den Status *Nur Verbindung* gesetzt.

Damit haben Sie eine funktionelle Parametertabelle für die Power-Query-Abfrage. Mit der Datenüberprüfungsliste stellen Sie das passende Tabellenblatt ein, ein Klick auf *Daten/Alle aktualisieren* holt das aktuelle Abfrageergebnis aus der Datenquelle.

8.1.5 Daten aktualisieren per Makro

Mit einem kleinen VBA-Makro automatisieren Sie auch den letzten Schritt noch. Schreiben Sie es im VBA-Editor. Achten Sie darauf, dass die Auswertungsmappe anschließend mit dem Dateityp XLSM gespeichert werden muss (Excel-Arbeitsmappe mit Makros).

Aktivieren Sie mit **[Alt]+[F11]** den VBA-Editor. Schalten Sie im *Ansicht*-Menü den Projekt-Explorer ein, und öffnen Sie unter *Microsoft Excel-Objekt* das Tabellenblatt mit der Umsatzauswertung (und der Parametertabelle).

Geben Sie dieses Makro ein. Die absolute Adresse `$$$4` ist das »target«, wenn diese Zelle verändert wird, startet das Makro die Aktualisierung aller Abfragen.

```
Private Sub Worksheet_Change(ByVal Target As Range)
If Target.Address = "$$$4" Then ActiveWorkbook.RefreshAll
End Sub
```

Speichern Sie die Mappe als Makro-Arbeitsmappe ab. Testen Sie das Makro, holen Sie über die Datenüberprüfungsliste einen anderen Eintrag in die Zelle `$$$4`, wird die Power-Query-Abfrage automatisch aktualisiert.

8.2 Fuzzy-Übereinstimmung (ungefähre Übereinstimmung)

Das Problem kennen Sie sicher vom SVERWEIS: Der Suchbegriff muss exakt mit den Suchdaten übereinstimmen, sonst findet die Formel nichts. Ein Leerzeichen oder ein überflüssiger Punkt genügt schon, um die Suche zu torpedieren.

Power Query bietet in Microsoft 365 ein nützliches Werkzeug, das beim Zusammenführen von Daten bis zu einem bestimmten Grad auch ungefähre Übereinstimmung zulässt. Die Fuzzy-Übereinstimmung akzeptiert kleine Abweichungen zwischen Suchbegriff und Suchbereich. Wikipedia definiert das Ganze als unscharfe Suche, einem Informatik-Algorithmus für Zeichenketten:

https://de.wikipedia.org/wiki/Unscharfe_Suche

Erstellen Sie im Power-Query-Editor eine erste Abfrage. Wählen Sie *Start/Kombinieren/Abfragen zusammenführen*. Holen Sie eine zweite Abfrage mit einer identischen Spalte in das Dialogfeld. Markieren Sie in beiden Vorschaulisten die Spalte mit den gemeinsamen Schlüsselbegriffen.

Schalten Sie *Fuzzyübereinstimmung zum Zusammenführen verwenden* ein, um das Ergebnis zu verbessern. Power Query sucht damit auch Werte in der zweiten Abfrage, die mit denen der ersten ungefähr übereinstimmen.

Über die *Optionen* lässt sich das Ergebnis noch steuern:

- *Schwellenwert für die Ähnlichkeit*: Gibt an, wie ähnlich zwei Werte sein müssen, damit sie als übereinstimmend erkannt werden. Mit dem Mindestwert 0 stimmen alle überein, beim Höchstwert 1 nur die mit exakt gleicher Schreibweise. Nehmen Sie einen Standardwert von 0,8, und variieren Sie ihn, falls zu viele oder zu wenig Treffer gelandet werden.
- *Groß-/Kleinschreibung ignorieren*: Standardmäßig wird Groß-/Kleinschreibung beim Vergleich ignoriert. Schalten Sie die Option aus, wenn die Schreibweise eine Rolle spielt.
- *Durch Kombination von Textteilen abgleichen*: Findet Übereinstimmungen, indem Texte zusammengefügt werden (z. B. Sportverein und Sport Verein).
- *Maximale Zahl Übereinstimmungen*: Geben Sie an, wie viele Übereinstimmungen Sie für jede einzelne Zeile haben wollen (Standard: *alle Werte*). Mit 1 findet Fuzzy nur jeweils eine Übereinstimmung.
- *Transformationstabelle*: Tragen Sie den Namen einer zweiseitigen Zuordnungstabelle ein.

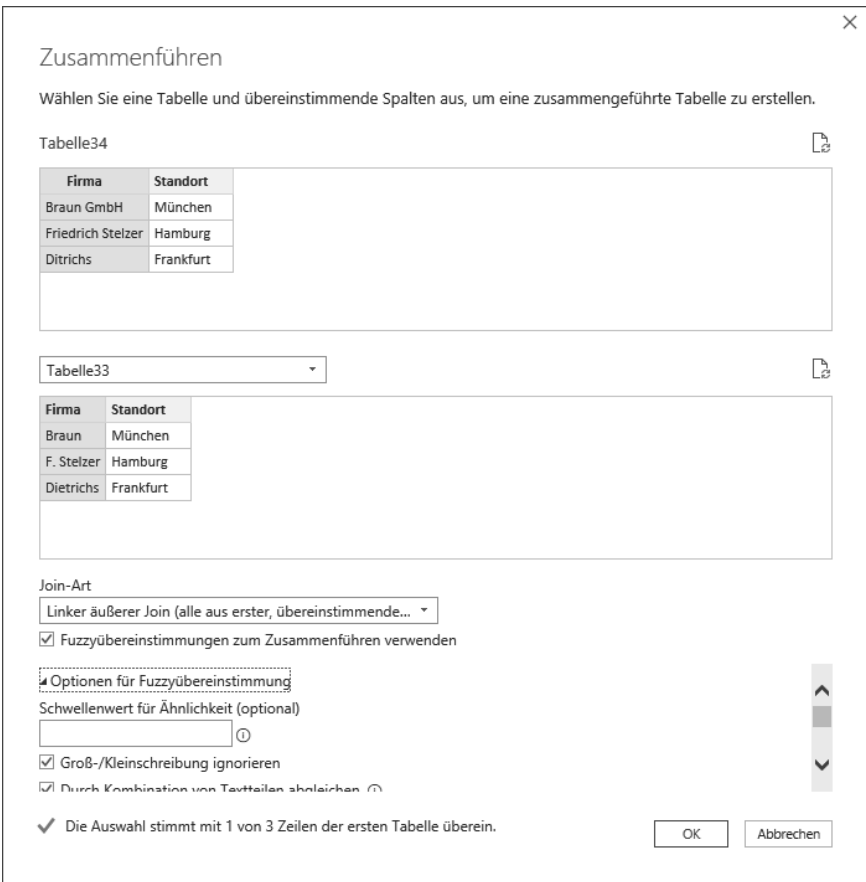


Abbildung 8.8: Mit Fuzzy-Logik sucht Power Query mit künstlicher Intelligenz.