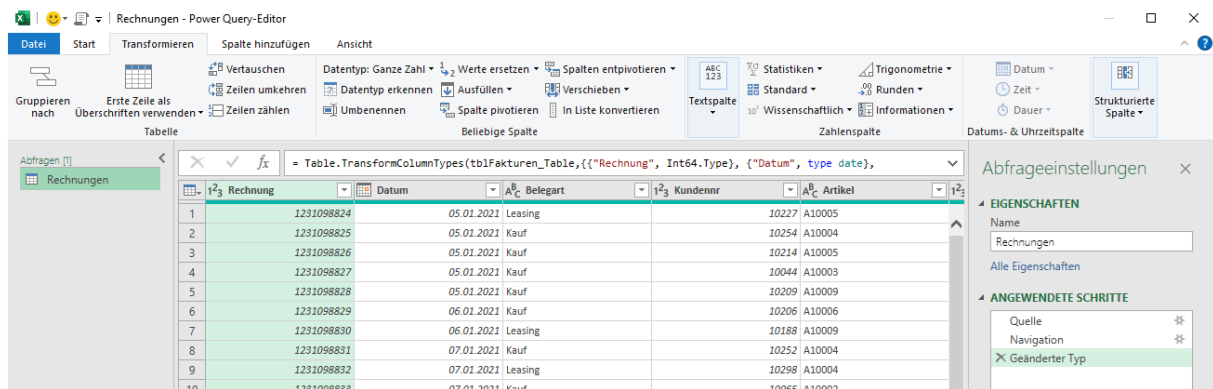
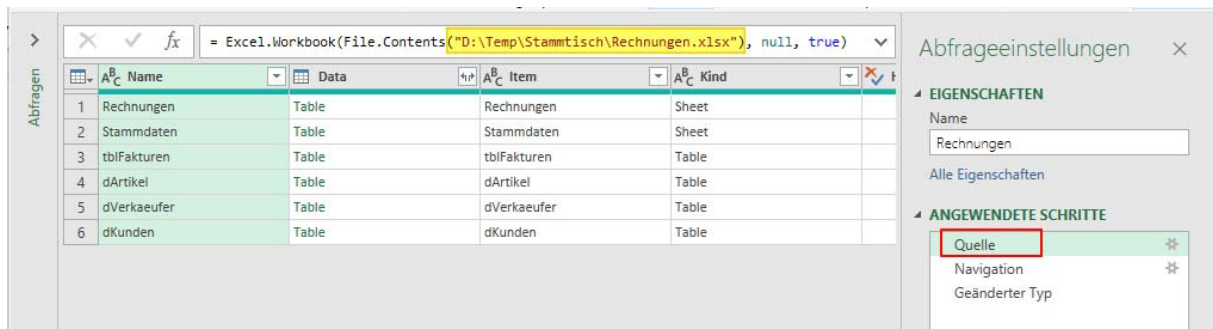


## Excel-Zelle als Parameter an Power Query übergeben

Ich habe aus einer externen Arbeitsmappe eine Tabelle mit Rechnungsdaten importiert:

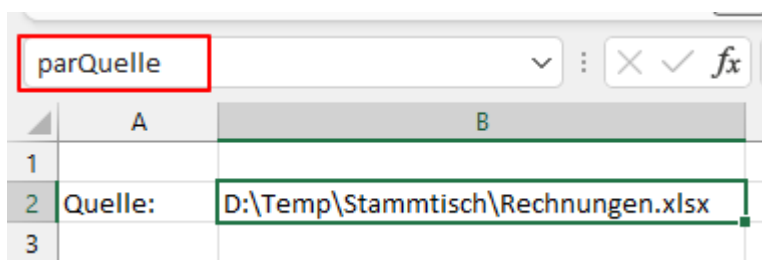


Der Pfad zu der Datei ist im Moment im Schritt Quelle noch fest hinterlegt:



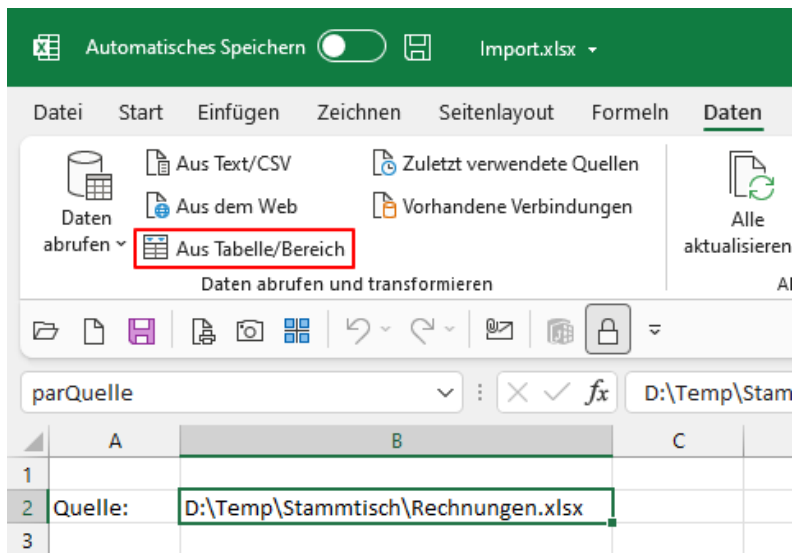
Dieser Pfad soll zukünftig anwenderfreundlich in Excel in einer definierten Zelle verändert und an Power Query übergeben werden.

Um eine einzelne Zelle aus Excel in PQ einzulesen, muss für die betreffende Zelle ein Name festgelegt werden. Am schnellsten geht das, wenn man den gewünschten Namen direkt in das Namensfeld links oben eintippt, in meinem Beispiel also „parQuelle“:

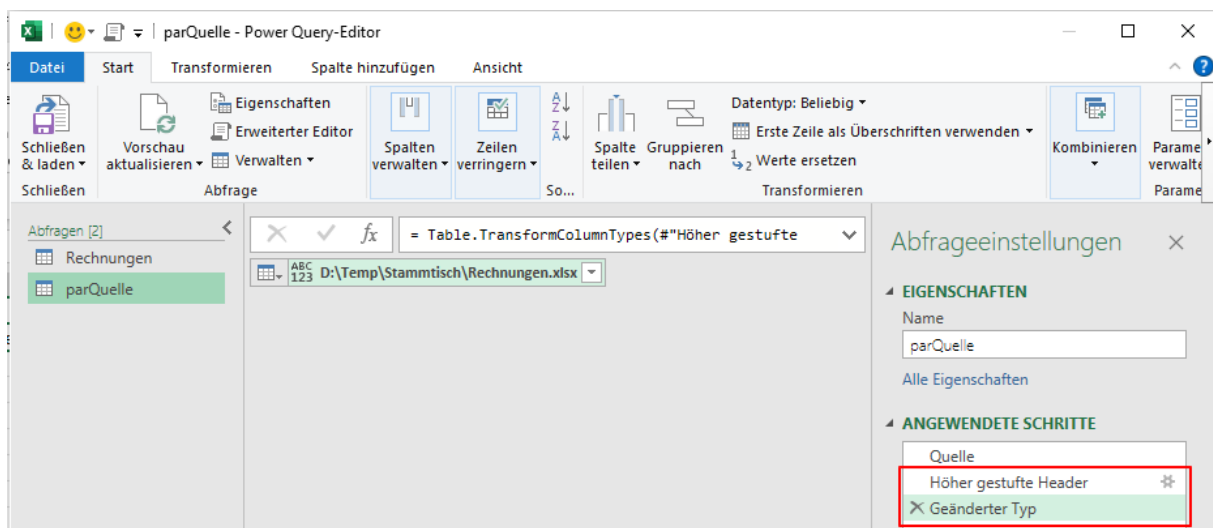


Dann kann diese einzelne Zelle importiert werden.

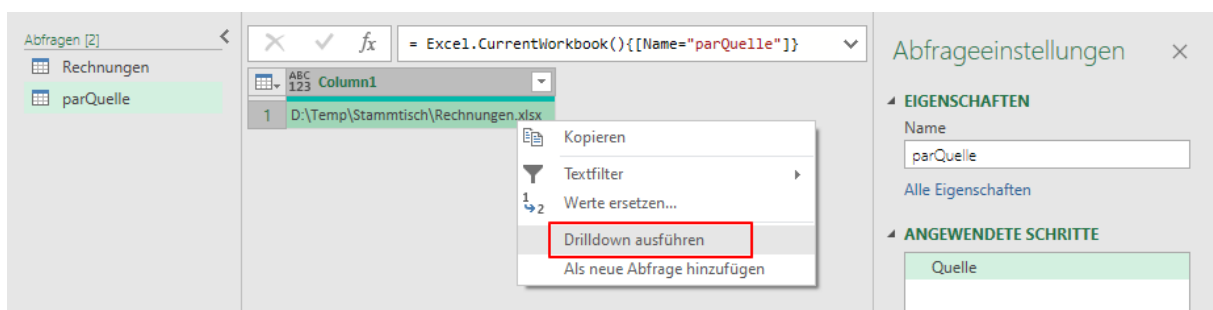
Daten | Aus Tabelle/Bereich:



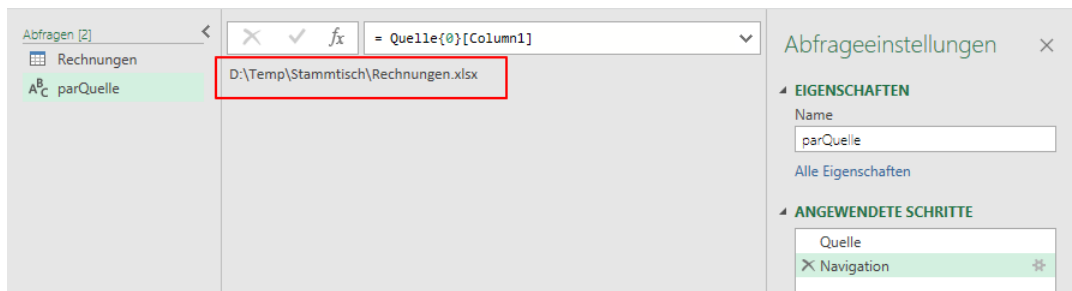
Power Query führt normalerweise gleich an paar zusätzliche Schritte aus, die nicht benötigt werden und daher gelöscht werden können:



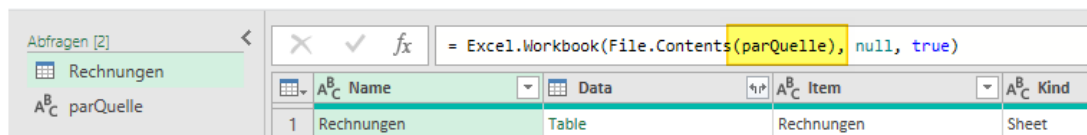
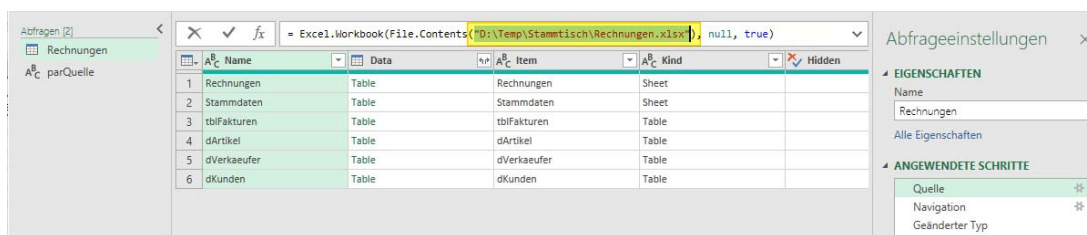
Bleibt also nur noch der erste Schritt „Quelle“. Über einen Rechtsklick in die angezeigte Zelle wird der Menüpunkt „Drilldown“ aufgerufen:



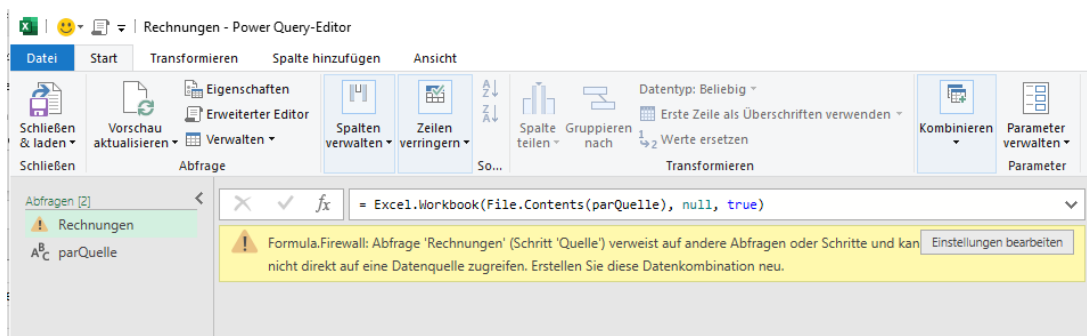
Am Ende haben wir damit die Pfadangabe als reinen Text und können diesen anschließend in der Rechnungsabfrage einbauen:



Dazu wird in der Abfrage „Rechnungen“ der erste Schritt „Quelle“ angepasst. Die in Anführungszeichen gesetzte Pfadangabe wird ersetzt durch den Namen der Parameter-Abfrage „parQuelle“:

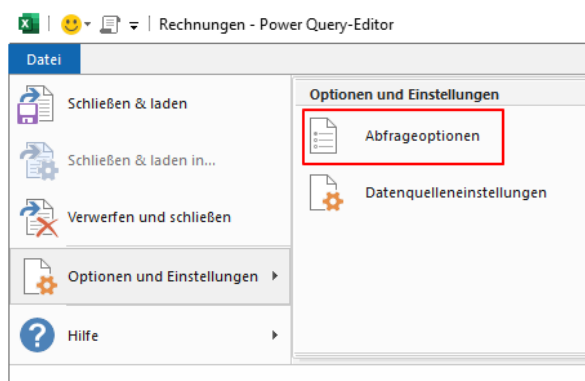


Diese Anpassung führt erst einmal zu einer Fehlermeldung:

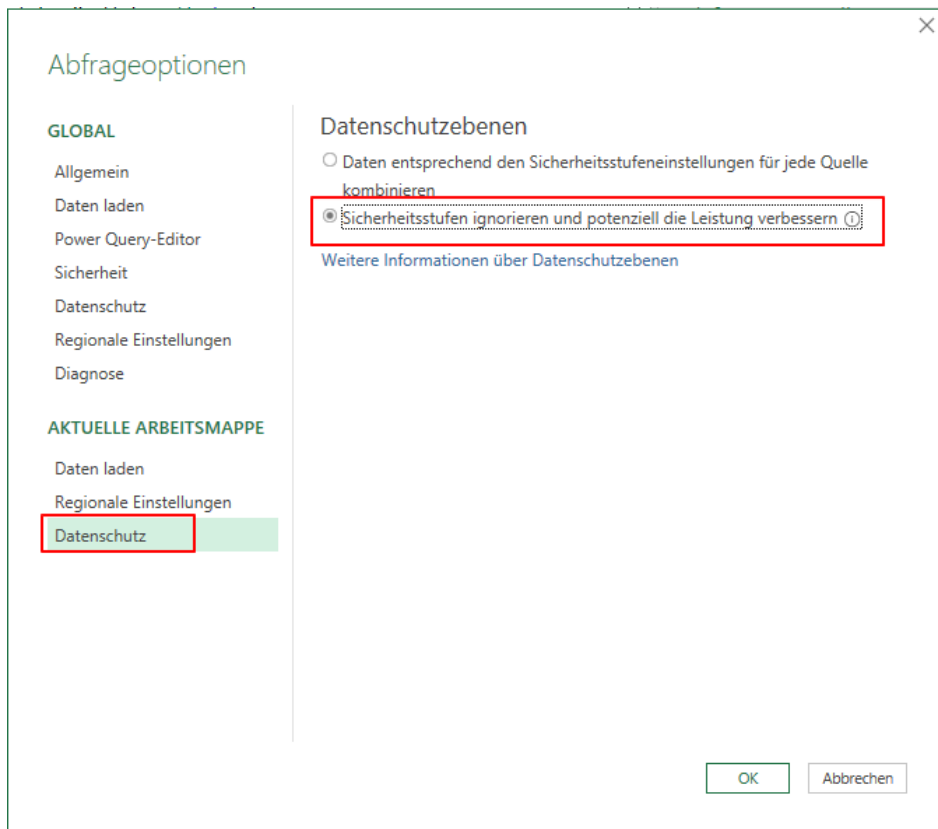


Dies hängt mit den Sicherheitseinstellungen zusammen, die in Power Query vorgabemäßig aktiviert sind. Die Meldung lässt sich leicht beseitigen über das Menü

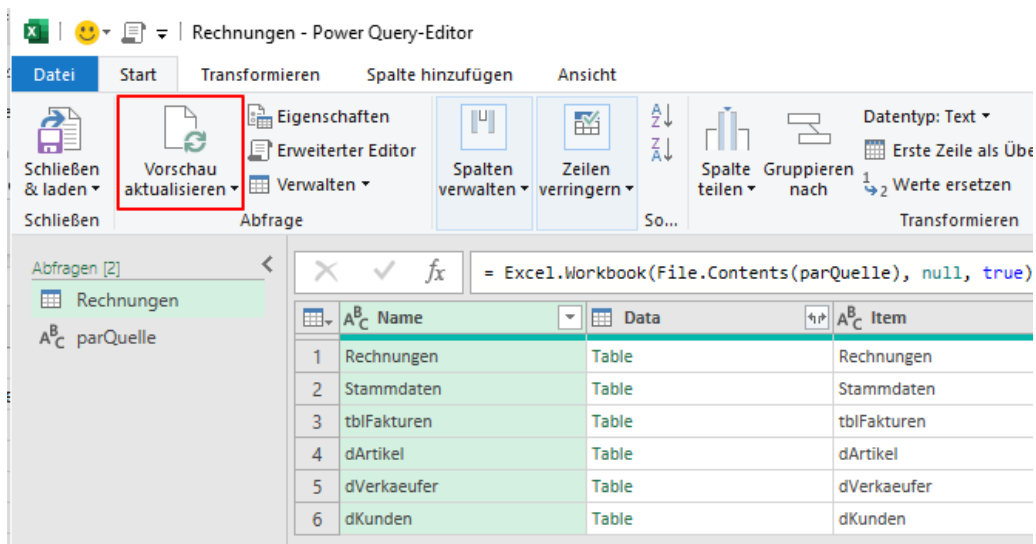
Datei | Optionen | Abfrageoptionen



Hier wird unter dem Punkt „Datenschutz“ für die aktuelle Arbeitsmappe die Datenschutzebene angepasst auf „Sicherheitsstufen ignorieren...“:



Nach einem Klick auf das Menü Start | Vorschau aktualisieren ist der Fehler verschwunden und die Abfrage liefert wieder das gewünschte Ergebnis:



Jetzt kann der Pfad oder der Dateiname jederzeit in der entsprechenden Excel-Zelle verändert werden und wird beim Aktualisieren der Rechnungsabfrage berücksichtigt.

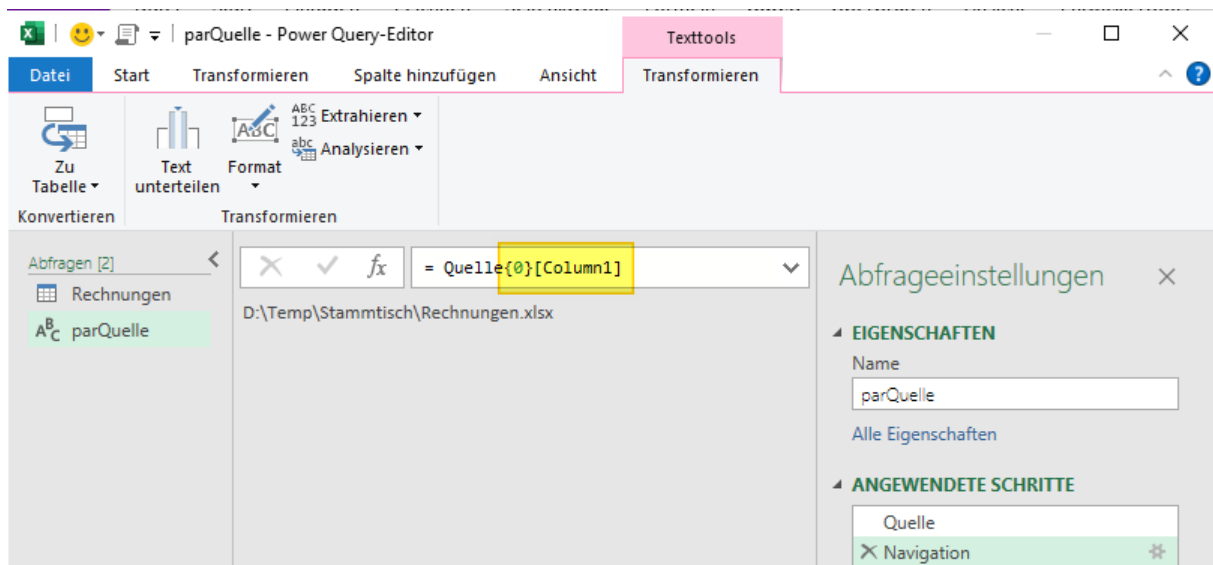
## Formula.Firewall-Fehler vermeiden

Wenn dieser Fehler wie oben beschrieben in den Abfrageoptionen beseitigt wurde, dann tritt das Problem auf dem aktuellen PC nicht mehr auf. Wird die Datei aber an einen anderen Anwender weitergegeben, wird dieser beim Aktualisieren wieder mit dem Formula.Firewall-Fehler konfrontiert und muss seinerseits die Abfrageoptionen anpassen.

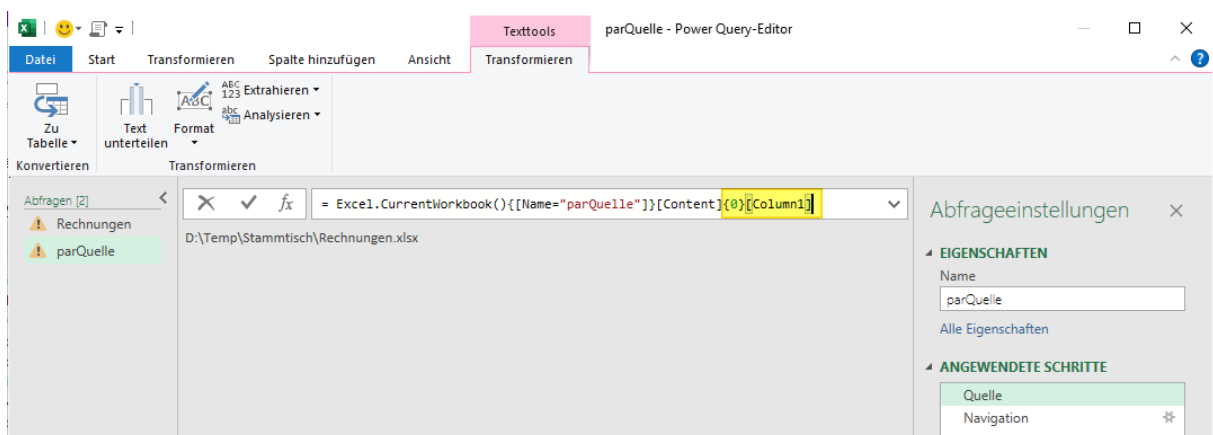
Um das zu vermeiden, müssen die Abfragen angepasst werden.

Im Power Query-Editor bearbeite ich zuerst die Parameter-Abfrage „parQuelle“ und markiere dazu den letzten Schritt „Navigation“. In der Bearbeitungsleiste markiere ich den Teil nach „Quelle“ und kopiere ihn mit Strg+V in die Zwischenablage:

{0}[Column1]



Dann markiere ich den ersten Schritt „Quelle“ und füge in der Bearbeitungsleiste ganz am Ende den zuvor kopierten Teil aus der Zwischenablage wieder ein:



Der Schritt „Navigation“ ist damit überflüssig geworden und kann gelöscht werden.

Als nächstes kopiere ich jetzt den gesamten Inhalt des verbliebenen Schrittes „Quelle“ in die Zwischenablage (ohne das Gleichheitszeichen am Anfang!):

Excel.CurrentWorkbook(){[Name="parQuelle"]}.Content{0}[Column1]

Als nächstes öffne ich die Abfrage „Rechnungen“ und markiere dort den Schritt „Quelle“

In der Bearbeitungsleiste muss nun „parQuelle“ mit dem kopierten Inhalt der Zwischenablage ersetzt werden. Aus

```
= Excel.Workbook(File.Contents(parQuelle), null, true)
```

wird damit

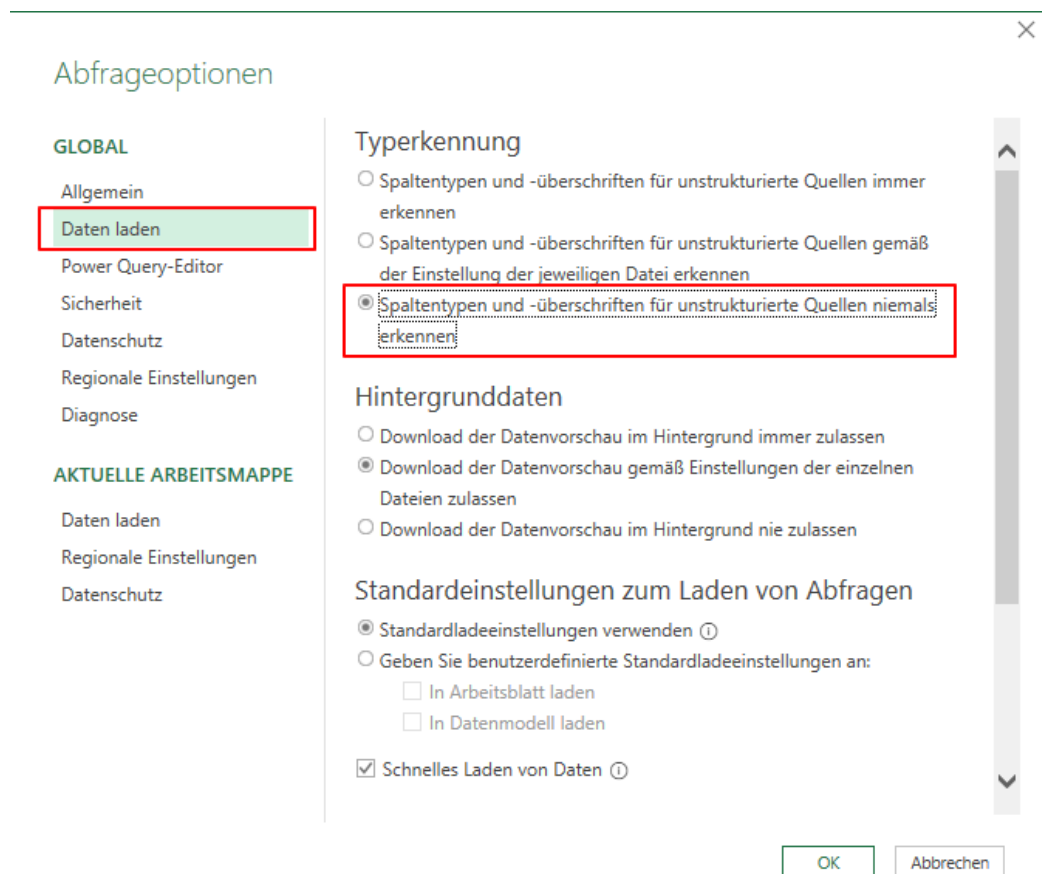
```
= Excel.Workbook(File.Contents(Excel.CurrentWorkbook()[[Name="parQuelle"]][Content][0][Column1]), null, true)
```

Das war's auch schon. Die ursprüngliche Parameter-Abfrage parQuelle ist damit komplett überflüssig geworden und kann gelöscht werden, denn alle Schritte sind jetzt direkt in der Rechnungen-Abfrage enthalten.

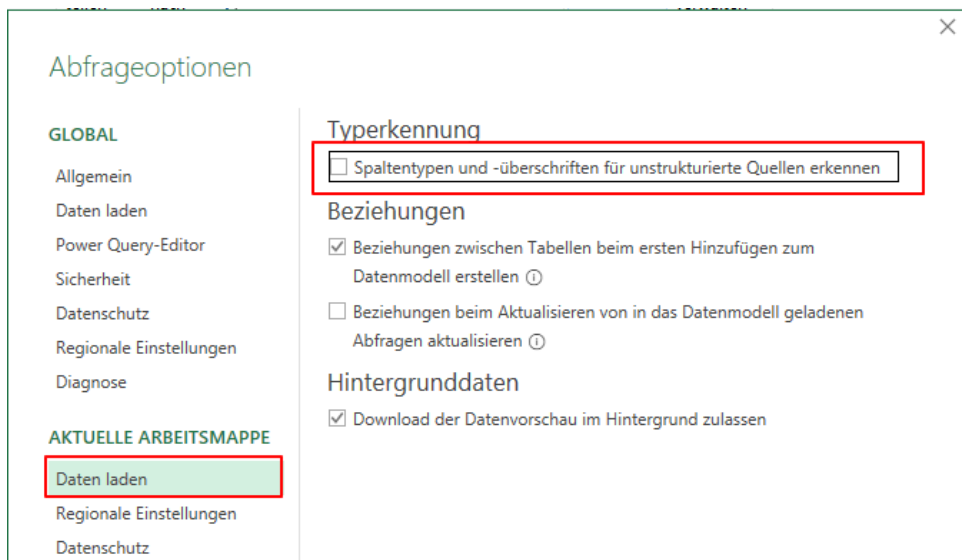
## Abfrageoptionen anpassen

Um die automatische Typenerkennung in Power Query abzuschalten, gibt es eine Einstellung in den Abfrageoptionen.

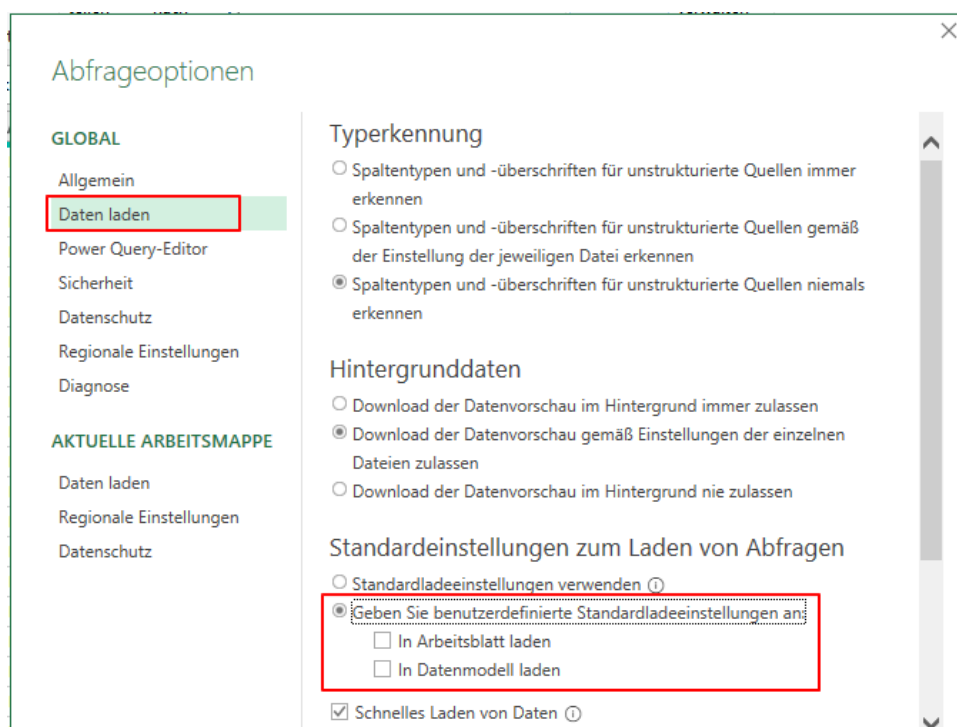
Die globale Einstellung gilt für alle Arbeitsmappen:



Oder alternativ nur für die aktuelle Mappe:



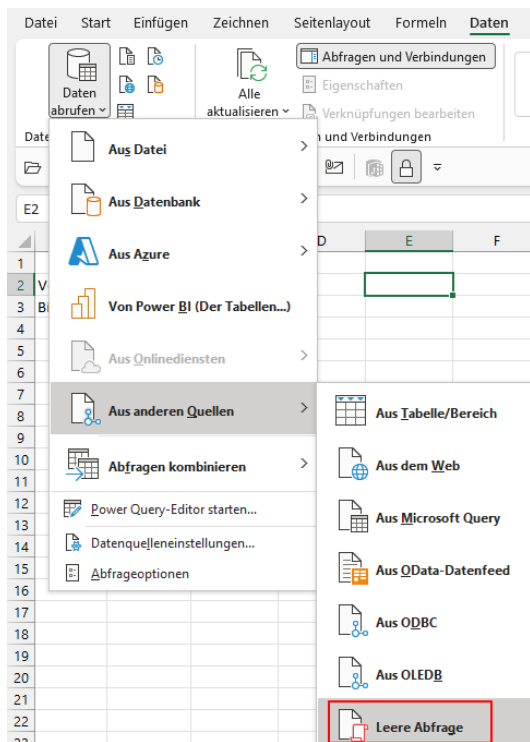
Sollen die Abfragen nicht – wie es in den Standardeinstellungen der Fall ist – automatisch nach Excel geladen, sondern nur eine Verbindung erstellt werden, ist das über folgende globale Einstellung möglich:



## Dynamische Kalendertabelle erzeugen

Über Zeile M-Code lässt sich ganz leicht eine Kalendertabelle erzeugen. Dazu wird erst eine neue, leere Abfrage erstellt:

Daten | Daten abrufen | Aus anderen Quellen | Leere Abfrage



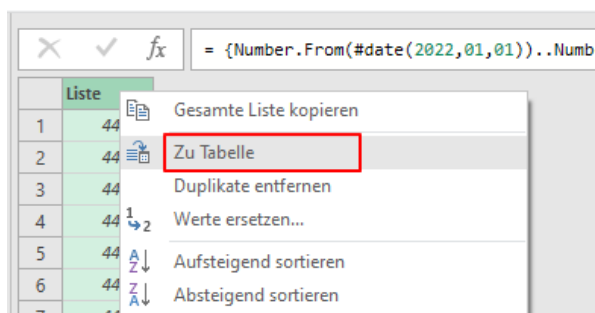
In der Bearbeitungszeile wird folgender M-Code eingegeben:

```
= {Number.From(#date(2022,01,01))..Number.From(#date(2022,12,31))}
```

Damit wird eine Liste erstellt, die alle Datumswerte zwischen den angegebenen Anfangs- und Endterminen enthält:

	Liste
1	44562
2	44563
3	44564
4	44565
5	44566
6	44567
7	44568
8	44569
9	44570

Über einen Rechtsklick auf den Listenkopf wird das Ganze in eine Tabelle umgewandelt:



Das nächste Eingabefenster wird unverändert mit OK übernommen:



Zu Tabelle

Erstellen Sie eine Tabelle aus einer Liste von Werten.

Trennzeichen eingeben oder auswählen

Keine

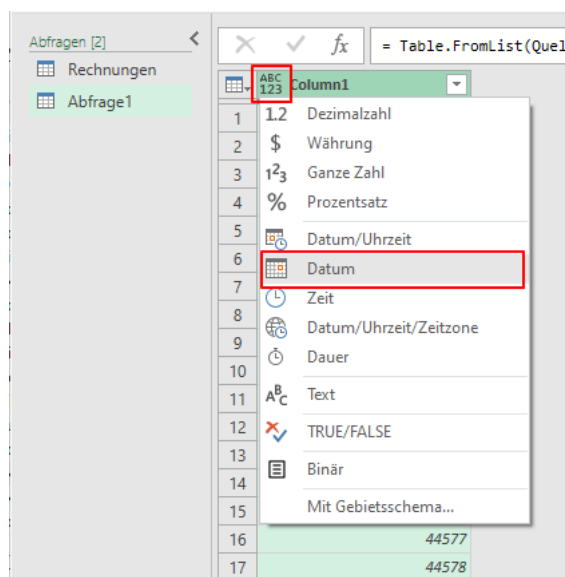
Behandlung zusätzlicher Spalten

Als Fehler anzeigen

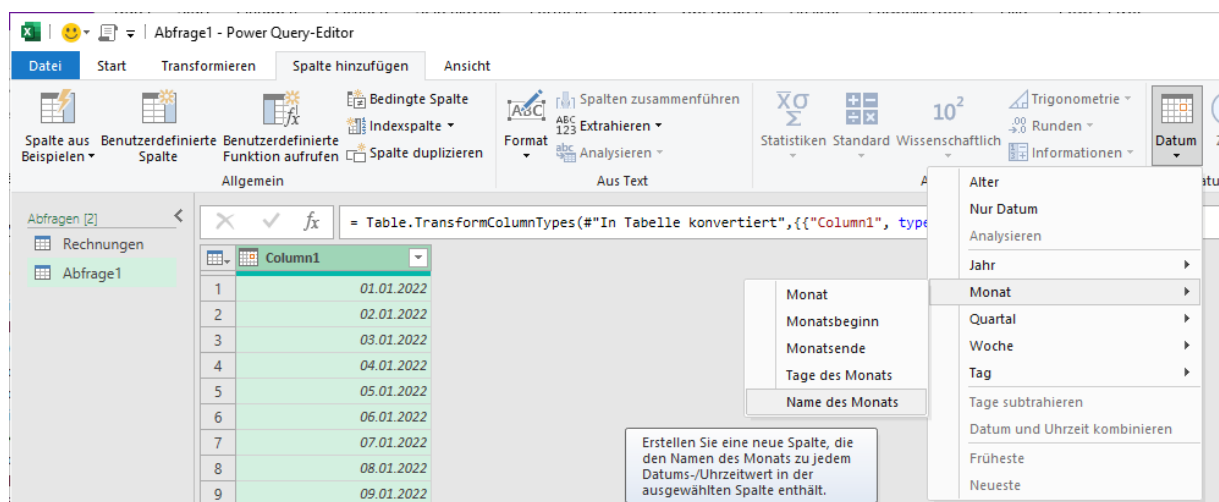
OK

Abbrechen

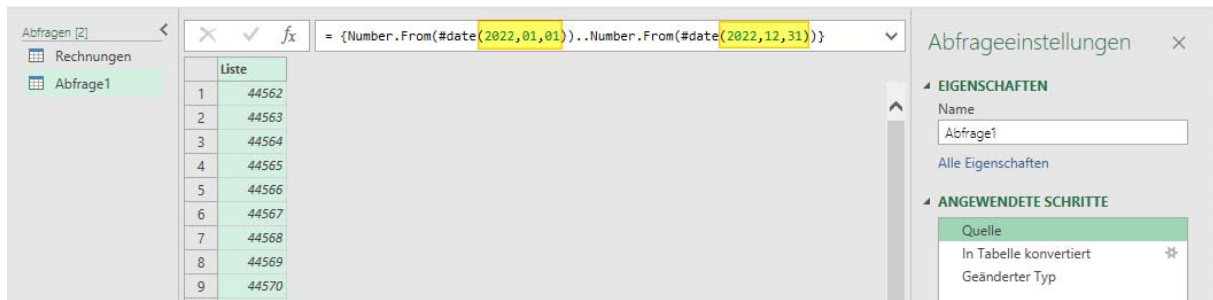
Nun wird aus der Liste eine Tabelle, für die der Datentyp noch auf „Datum“ geändert wird:



Anschließend kann über das Menü „Spalte hinzufügen | Datum“ alle gewünschten Datumsinformationen ergänzt werden, z.B. Monatsname, Wochentag etc:



Für einen neuen Kalender muss nur der Datumsbereich im ersten Schritt „Quelle“ angepasst werden:



Noch dynamischer geht es, wenn wie im ersten Beispiel (mit dem Pfad zur Rechnungsdatei) zwei Parameter-Felder im Arbeitsblatt definiert, in Power Query importiert und wieder in den Quelle-Schritt eingefügt werden.

Unter der Annahme, dass die beiden Parameter-Zellen im Arbeitsblatt mit „parVon“ und „parBis“ benannt wurden, sollte der erste Schritt „Quelle“ in der Kalenderabfrage am Ende wie folgt aussehen:

```
=
{Number.From(Table.TransformColumnTypes(Excel.CurrentWorkbook(){[Name="parVon"]}[Content],
{"Column1", type
date})){0}[Column1])..Number.From(Table.TransformColumnTypes(Excel.CurrentWorkbook(){[Name
="parBis"]}[Content],{"Column1", type date})){0}[Column1])}
```