



POWERQUERY.SK

Professional Power Query & The M Language

Michal Chmelár

February 2024

ISBN: 978-80-973078-7-5

EAN: 9788097307875

© 2023-2024 Smart People, spol. s r.o. All rights reserved.

Reproduction, transmission, dissemination or storage of parts or the entire content of this publication in any form, without the prior written consent of the publisher, is prohibited. No part of this publication may be published or distributed in any way or in any form without the written permission of the publisher. All names, brands and designations used are or may be trademarks of their respective owners.

This is an independent publication and is neither affiliated with, nor authorized, sponsored, or approved by, Microsoft Corporation.

All information in this publication is provided “as is”. Neither the author nor the publisher guarantee their 100% correctness, and are not responsible for any damages that could arise from their use. You bear the risk of using them yourself. While the information in this book is believed to be true and accurate at the date of publication, neither the author nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Publisher: Smart People, spol. s r.o., Slovak Republic, EU

Year: 2024

Published & printed in Slovak Republic, EU

First translated edition

Translated from the Slovak original: *Profesionálne Power Query a jazyk M*

First published in Slovak Republic in 2023

This work is subject to copyright. All rights are reserved by the publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the publisher's location, in its current version, and permissions for use must always be obtained from Smart People, spol. s r.o. company. Violations are liable to prosecution under the respective Copyright Law. Trademarked names, logos and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit the trademark owner, with no intention of infringement of the trademark. The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Acknowledgment

Writing a book is not easy. It requires a lot of time, effort and discipline. Therefore, I would like to thank the people who made this book possible.

First of all, I would like to thank my wife for her support during the writing and publishing of the book. If it weren't for her, this book would probably never have been created. I also thank my clients for giving me many challenges over the years, on which I constantly improved. And finally, I thank God for giving me this idea, the space and the patience to bring it all to a successful conclusion again.

And of course, I thank the entire community around the PowerPivot.sk blog for their ideas, observations and comments. It is thanks to them that this book is built much more practically than if I had created it myself.

It belongs to all of you

BIG THANKS

Table of Contents

INTRODUCTION	9
HOW TO READ THIS BOOK – METHODOICAL INSTRUCTIONS FOR THE BOOK	10
BASICS OF WORKING WITH POWER QUERY	13
INTRODUCTION TO POWER QUERY	14
HOW TO GET INTO POWER QUERY & GET DATA.....	17
POWER QUERY EDITOR & WORKING WITH QUERIES.....	27
TRANSFORMING DATA WITH THE POWER QUERY EDITOR.....	35
FILTERING, SORTING, GROUPING & SUMMARIZING DATA.....	51
CREATING CALCULATED COLUMNS & EXTRACTING VALUES FROM COLUMNS	60
COLUMN FROM EXAMPLES – FAST TEXT TRANSFORMATIONS.....	66
MANUALLY ENTERED TABLES.....	71
AUTHENTICATING & CHANGING DATA SOURCE SETTINGS	73
GETTING DATA FROM CSV, XML AND JSON FILES	76
COMBINING TABLES – MERGE QUERIES, APPEND QUERIES	82
PIVOTING AND UNPIVOTING COLUMNS	96
FINDING, HANDLING AND CORRECTING ERRORS.....	104
ADVANCED EDITOR.....	113
TABLES FROM EXAMPLE.....	117
POWER QUERY PARAMETERS	127
TRANSFORMING DATA USING PYTHON AND R SCRIPT	132
TRANSFORMING DATA WITH CHATGPT.....	136
THE M LANGUAGE, EXPRESSIONS AND SCRIPTS.....	143
INTRODUCTION TO THE M LANGUAGE	144
QUERIES, VALUES AND SYNTAX OF THE M LANGUAGE.....	146
WORKING WITH NUMERIC AND TEXT VALUES.....	160
WORKING WITH DATES AND TIMES.....	168
CONVERTING TEXT INTO DATE AND NUMERIC VALUES AND BACK	178
LISTS.....	187
RECORDS	197
IDENTIFIERS IN THE M LANGUAGE.....	211
LET..IN BLOCK	214
COMMENTS	222
THE IF..THEN..ELSE STATEMENT	226
WORKING WITH BLANK AND NULL VALUES	230

FUNCTIONS & CUSTOM FUNCTIONS.....	236
LOOPING CONSTRUCTS IN THE M LANGUAGE.....	259
TABLES & THEIR PROPERTIES.....	268
METADATA	283
HANDLING ERRORS & THE TRY STATEMENT	289
WORKING WITH FILES AND BINARY VALUES.....	304
SECTIONS & ENVIRONMENTS.....	309
TYPES, CUSTOM TYPES AND TYPE SYSTEM OF THE M LANGUAGE.....	318
POWER QUERY FIREWALL	341
ULTIMATE CALENDAR TABLE FOR POWER QUERY	352
CONCLUSION	353
PRACTICAL DATA PROCESSING TECHNIQUES IN POWER QUERY	355
BEFORE YOU START	356
HOW TO ADD ROWS TO AN EXISTING TABLE.....	356
HOW TO COMBINE DATA FROM SEVERAL IDENTICAL EXCEL SHEETS.....	358
HOW TO COMBINE DATA FROM MULTIPLE FILES	362
HOW TO COMBINE DATA FROM MULTIPLE IDENTICAL DATA SOURCES.....	365
DETECTING FIRST & LAST ROWS WITH DATA	371
FINDING NEW AND MISSING ROWS IN 2 VERSIONS OF THE SAME TABLE.....	376
JOINING ROWS VIA CONDITION AND CUSTOM DATA CATEGORIZATION	378
WORKING WITH THE PREVIOUS & NEXT ROW	381
CALCULATING RUNNING TOTALS	385
WORKING & NON-WORKING DAYS, AND CALCULATIONS WITH THEM.....	387
MAPPING TABLES.....	391
CROSS JOIN AND COMBINATION TABLES	397
CALCULATIONS WITHIN A GROUP, WEIGHTED AVERAGE & SUMIF	401
WORKING WITH DUPLICATE ROWS	406
DYNAMIC FILTERING BY USING THE CONFIGURATION TABLE	414
CALCULATING THE NUMBER OF CONSECUTIVE DAYS.....	420
NATURALIZATION OF PARENT-CHILD HIERARCHIES.....	426
VARIOUS TINY PRACTICAL TECHNIQUES	431
KEYBOARD SHORTCUTS TO INCREASE PRODUCTIVITY IN POWER QUERY	438
GETTING DATA FROM APIS.....	443
INTRODUCTION TO API, PRINCIPLES & API TERMINOLOGY.....	444
INITIAL TESTING AND DEBUGGING OF API CALLS	452
GETTING DATA FROM REST APIS & THE GET REQUEST	454
DYNAMIC DATA SOURCES & AUTOMATIC DATA REFRESH IN POWER BI	465
CALLING A POST REQUEST	471

CALLING OTHER REQUEST TYPES	475
ADVANCED OPTIONS OF THE WEB.CONTENTS FUNCTION.....	476
BASIC AUTHENTICATION, BEARER TOKEN & OAUTH 2.0 AUTHENTICATION	479
USING TEXT ENCODINGS WHEN WORKING WITH APIS	481
GETTING DATA FROM SOAP APIS	486
GETTING DATA FROM ODATA APIS	488
GETTING DATA FROM GRAPHQL APIS	495
PAGING RESULTS FROM APIS	499
OTHER TECHNIQUES USED WHEN WORKING WITH APIS	503
POWER QUERY CUSTOM CONNECTORS	515
POWER QUERY PERFORMANCE OPTIMIZATION.....	519
BEFORE YOU START	520
POWER QUERY PHILOSOPHY & OPERATING PRINCIPLES.....	521
STREAMING, RE-STREAMING & BUFFERING	534
QUERY FOLDING – PUSHING OPERATIONS TO THE DATA SOURCE.....	550
IMPLEMENTING CUSTOM QUERY FOLDING	565
QUERY PLAN	573
CACHES & EVALUATION OF REFERENCED QUERIES.....	577
DIAGNOSTICS OF THE CAUSES OF SLOW PROCESSING IN POWER QUERY	589
CORRECT MEASUREMENT OF QUERY SPEED & DURATION.....	595
BACKGROUND DOWNLOADS OF DATA PREVIEWS & OTHER DECELERATORS.....	604
SPEEDING UP THE SLOW POWER QUERY EDITOR.....	608
OPTIMIZING TABLE JOINS / MERGE QUERIES	611
OPTIMIZING CROSS JOINS & COMBINATION TABLES.....	630
OPTIMIZING JOINING OF ROWS VIA A CONDITION / CONDITIONAL JOINS	633
OPTIMIZING CUMULATIVE FUNCTIONS – GROUP BY, DISTINCT, PIVOT	639
OPTIMIZING THE SPEED OF EXPANDING COLUMNS.....	651
OPTIMIZING LARGE NUMBER OF ITERATIONS	661
OPTIMIZING NUMEROUS TABLE JOINS & ITERATIONS	667
OPTIMIZING COUNTING OF WORKING DAYS.....	678
OPTIMIZING ITERATIONS USING THE DIVIDE & CONQUER TECHNIQUE	685
OPTIMIZING RUNNING TOTALS & THE CUMULATIVE COLUMN TECHNIQUE.....	691
REPLACING INEFFICIENT OPERATIONS & HIDDEN OPTIMIZATIONS	698
REMOVING UNNECESSARY & MERGING REPEATED OPERATIONS	706
OPTIMIZING SPEED USING THE EARLY TERMINATION TECHNIQUE	712
OPTIMIZING THE SPEED OF ERROR HANDLING.....	718
OPTIMIZING CALCULATIONS USING A COMPLETELY DIFFERENT APPROACH.....	720
OPTIMIZING TABLE SORTING & THE QUERY EVALUATION PHASE.....	728

INCREMENTAL DATA REFRESH.....	733
VARIOUS TIPS FOR OPTIMIZING POWER QUERY PERFORMANCE	742
CONCLUSION	745
EPILOGUE	747
WHAT TO DO NEXT.....	748
ABOUT THE AUTHORS BLOGS	749
DOWNLOAD LINKS.....	750
INDEX	751
NOTES	757

Introduction

How to Read This Book – Methodical Instructions for The Book

This book is written to help you learn Power Query and the M language step by step from the very basics, and to learn commonly used procedures and techniques, which you will encounter in practice when using this technology. It is best to read this book in the order it is written, although you may find the later chapters more appealing. Everything has its own time and logical sequence, in which order to approach it. This is also why this book is written and organized into a specific sequence of chapters and sections, which gradually build on each other, so that you not only learn Power Query well, but also handle increasingly difficult problems in Power Query and the M language. Therefore, go also through the things that you think you won't need. In many cases, you will find that the techniques and procedures used there might also be needed in your case, or that they will clarify a problem that you could not solve before.

All presented solutions and examples were tested on datasets of up to 65 million of rows, so that you can be sure that they work fast also on large volumes of data. If, for any reason, they work slower, then there's a reference provided for the optimization technique, pointing to the specific section of the book, which can be used to rapidly speed up its execution.

This book contains knowledge, solutions and techniques from authors commercial work & personal research, that emerged during work on extra demanding Power Query optimization projects, and were not published anywhere until now. Many times, they will challenge your thinking patterns to think about those problems in a completely different way, in order to deal with them quickly and efficiently.

The first chapter deals with the basics of Power Query. It guides the reader from the beginning of how to get data into Power Query, perform basic transformations on it, and load it successfully into the target applications. It covers basic techniques of filtering, grouping, summarizing, sorting and combining data, as well as creating calculated columns and basic error handling. It also covers how to leverage artificial intelligence (AI) to get the job done.

The second chapter deals with writing the scripts and expressions in the M language. It covers the entire M language, from values, types, and conversions between them, through working with lists, records, functions, tables, errors and metadata, up to the advanced topics like environments and the Power Query firewall.

The third chapter covers many practical techniques for data processing in Power Query. It covers multiple techniques how to access and combine data in different ways, how to perform common calculations and transformation patterns on data, and how to prepare data for analytical tools for specific scenarios.

The fourth chapter deals with getting data from various APIs, to get data from a data source that is not supported in your Power Query version. After explaining the concepts and vocabulary used for APIs, it guides you through how to use today's most common standard APIs like REST, SOAP, OData and GraphQL APIs. And deals also with the most common problems encountered when using APIs, including how to make them work with Power BI automatic data refresh features.

The fifth chapter extensively covers Power Query performance optimization. It contains in-depth explanation how various parts of Power Query internally work, and how to leverage that to speed up the scripts rapidly. It also contains dozens of strategies, how to approach specific performance problems, and multiple ways how to solve them. So that you can transform large volumes of data in Power Query in a fast and efficient way. This way you will know not only how to do it, but also why it has to be done that way. And, most importantly, know how to apply and modify them so that they can also be used in similar situations.

However, not everything in this book is flawless. Despite the consistent efforts of the author, minor typos may occur in the book. It is also possible that some screenshots or instructions will not accurately reflect your version of Power Query. Power Query exists in many implementations, and each of them exist in many versions, while new versions are released twice a month. In addition, from 2020, there's a trend where Microsoft needs to more or less change the appearance of some Power Queries at least once a year. And writing a universal guide that will last 5 to 10 years is almost impossible. So, when you can't find a specific button, tab or window, first try to look in the application for a while to see if it's somewhere around or if it is named somehow differently. Alternatively, think about why it is so. In some cases, you may have to look for a solution on the web, or you may find that it needs to be done differently. All this happens from time to time. Therefore, please take these factors into account when something doesn't work as it should. Because despite the best efforts of the author of the book, due to the nature of frequent changes in the software vs. static printed books, it cannot be completely ruled out. And we're all only humans, so even the instructions in this book can contain mistakes. Therefore, arm yourself with patience and perseverance in the long run. In this way, you will increase the chance that you will succeed in finding a solution to the given problem. And sometimes take a look at the [PowerQuery.sk](https://powerquery.sk) blog. Maybe there will already be a ready-made solution.

All instructions are written for the English version of Power Query, in versions of Power BI Desktop and Excel from October 2023. In other Power Query implementations, such as Power Query Online, most of the functionality covered in this book should work for you as well. It's just that some of it won't work, or it will be named something else there. However, the book does not deal with versions

of Power Query older than 10 years, since they are no longer supported even by their manufacturer, and only a marginal number of users use these versions today.

There are also many footnotes in the book. They are there so that you can quickly find a topic that is currently being mentioned in the text, or additional information on that topic that did not fit in the book.

One last thing is that while this book is aimed in the beginning at a beginner audience, it also targets an advanced audience towards the end. However, it is not an exhaustive resource that covers this technology down to the last detail. This book would have to have several thousand pages. And it would be just as suitable as a paperweight under the door. Instead, the book focuses on the most essential things you need to know about Power Query in order to be able to independently process data in Power Query and prepare it for analytical and reporting tools. And, to use some more advanced techniques that may come in handy in practice. So, if you are just starting to work with Power Query, or have been working in it for several months, this book will be for you. However, if you have been working with it for several years, then probably only the last chapters will be useful for you from this book. In such a case, rather consider one of our advanced Power Query courses.

Basics of Working with Power Query

Introduction to Power Query

Power Query is a tool for advanced preparation and merging of data from different sources. Its advantage is that it is very easy to use and does not impose any limits on the user's creativity. It can be connected to approx. 30-160 different data sources, depending on the version you have. It can load data not only from common databases such as SQL Server, Oracle, PostgreSQL, Access or MySQL, but also from Excel, CSV, JSON, XML and PDF files, SharePoint, OneDrive, MS Teams, MS Dynamics, Microsoft 365 services, SAP, OLAP cubes, Google Analytics, or Salesforce. In addition, it can be connected to almost any system or data source via API or ODBC interfaces and extract data from it. And in some of its versions, this list can also be extended via custom connectors and usage of APIs.

But it doesn't stop at retrieving data. The resulting data can then be transformed in Power Query and combined in any way. For example, you can add missing data, correct existing data, combine data from multiple files or databases, combine data from all data sources (e.g. data from Excel with data from SQL Server), or calculate other statistics as needed. And all this using simple steps and without any restrictions on where these data came from. After they are retrieved, all data in Power Query look like tables, and from that moment on it doesn't matter where they came from – all operations work the same way on all of them.

Power Query uses its own M language for data transformations, which currently contains over 1,000 functions and scripting elements for data transformation and work – e.g. grouping and summarizing data, transposing data, merging and appending tables, filtering data, creating conditional columns, intelligently dividing data into multiple columns, date and time calculations, mathematical, statistical and scientific operations, row-by-row processing, and a bunch of other operations. And if these operations are not enough for you, you have the option of creating your own functions, with the help of which even complex transformations can be performed later in a few steps. However, you won't need to know this language at the beginning, because a large part of the functionality is built into the user interface, where most common transformations can be applied to data using interactive menus and buttons.

Another huge benefit of using Power Query is that it remembers all the steps you apply to your data as a sequence of steps. And you can subsequently change this sequence at any time – you can add more steps, change their order, delete and modify existing steps, or insert additional intermediate steps. And all this for only one purpose – so that everything continues to work, even if the data or perhaps its structure in the data source changes. Or if you later remember that you should have used a different sequence of steps, so that you don't have to redo everything from

scratch. All these steps are then applied automatically not only to the existing data, but also to the new data that comes with further data refreshes.

Thanks to this, you can extremely simplify data preparation in some cases, so much so that you will not have to laboriously build a data warehouse. Or, wait 3 weeks for the IT department, which under normal circumstances prepares this data, and due to being overwhelmed with other tasks, cannot do it anyway.

Power Query has a flexible and extensible architecture that allows users to connect to and transform data from a wide variety of sources. The Power Query architecture can be divided into three main components:

- 1) **Data Source Connectors:** Power Query includes built-in connectors for various data sources such as Excel workbooks, CSV files, databases (SQL Server, Oracle, PostgreSQL, MySQL, etc.), web services, and other data sources. These connectors provide a standard way to access and get data from various sources,
- 2) **Transformation Engine:** Once the data is retrieved, the transformation engine allows users to manipulate the data and transform it using a wide variety of built-in transformations such as filtering, sorting, merging, pivoting and more. Users can also create custom transformations using M, the functional scripting language built into Power Query,
- 3) **User interface and output:** Power Query provides an intuitive user interface that allows users to visually interact with data and preview data transformations as they are used. Users can then load the transformed data into Excel, Power BI or other applications for further analysis.

In addition to these core components, Power Query also includes a number of other features such as data profiling, error handling, and caching that increase the performance and reliability of data transformations and analysis. Overall, the Power Query architecture is designed to provide a flexible and powerful data transformation tool for users with diverse data requirements, and to handle billions of rows.

Power Query was originally introduced in 2012 as an Excel add-in called “Data Explorer”, later renamed by Microsoft to its current name of “Power Query”. In the following years, due to its popularity, this technology was also added into other products from Microsoft. At the time of writing, Power Query is included in the following Microsoft products, services, and components:

- 1) Power Query in Excel 365 and Excel 2010/2013/2016/2019/2021,
- 2) Power Query in Excel 365 for Mac,
- 3) Power BI Desktop (for cloud service, and also for Power BI Report Server),
- 4) Power BI Service,
- 5) Power BI Report Server,

- 6) Power BI / Power Platform Data Gateway,
- 7) Power BI Dataflows,
- 8) Power BI Datamarts,
- 9) Dataverse Database (found in Power Apps, MS Teams, MS Dynamics, etc.),
- 10) Power Automate,
- 11) SQL Server Analysis Services (SSAS Tabular),
- 12) Azure Analysis Services,
- 13) SQL Server Integration Services (SSIS) – Power Query Source,
- 14) Azure Data Factory (ADF).

Implementations listed in points #7 to 10 are also collectively referred to as “Power Query Online” because it is really just online version of Power Query built into these products and services.

Power Query does not have the same capabilities in all these implementations. In some of them, for example, a smaller number of connectors is available, in others more of them. Some of them have support for artificial intelligence, some don't. Some of them have more capabilities, some less. Some of them have a newer user interface, some older, and some none. And to make matters worse, each of these Power Queries has its own number of versions, because in most of the products above, Power Query is regularly updated and expanded. So outwardly, individual implementations can differ quite significantly for an uninitiated person. In reality, however, the capabilities of all of them are relatively balanced, and even the most functionally poor Power Query implementation has around 85-90% of the capabilities of the best implementation. Therefore, most of the information in this book will be applicable in most cases to every single Power Query, even if the user interface or the names of some buttons may look different. For space reasons, this book is written for the current versions included in Excel 365, Power BI Desktop, and Power Query Online. In other versions, you will probably have to do a bit of research on how to implement those procedures there. However, it shouldn't be difficult after a little practice.

Power Query is simply the ultimate data preparation tool. And at the same time, it is created in the way that most users imagine working with data during their preparation. In fact, this tool should be mandatory for everyone who needs to prepare data for their analyzes or reports. Because if you master it, it will save you a lot of time and trouble compared to classic data preparation techniques, and at the same time you will be well prepared for possible changes in the future. The whole tool is so flexible that there is probably no situation that it cannot handle. And if you don't want to learn it by yourself, you just found out where you can learn it 😊 Because not everyone has time to try by trial and error how to do what they need. Whatever you decide, be sure to at least try this tool. After playing with it for a while,

you'll be surprised what you can do with it. And we're going to show you some sample uses in this chapter so you know how to start working with it.

How to Get into Power Query & Get Data

To get started, let's take a look how to get into Power Query. Due to the fragmentation of the user interface not only in one Microsoft product, but sometimes across different versions of Power Query over time in the same product, it may not look the same way. So, we'll first take a look at how to get data with Power Query in some of the most commonly used Power Query implementations. So that you not only know how to get started, but so that we don't have to repeat these starting instructions over and over again for other Power Query implementations. Thanks to this, we only need to provide a simple instruction in the following sections as e.g. "get data from such and such an Excel file into Power Query", and it will be immediately clear to you how to do it in your version of Power Query. At the same time, this left more room in the book for more interesting information and procedures. If you are also considering working with Power Query Online, it is recommended that you first master at least basic work within Power Query in Power BI Desktop or Excel. This will make it much easier for you to work with the online versions of Power Query.

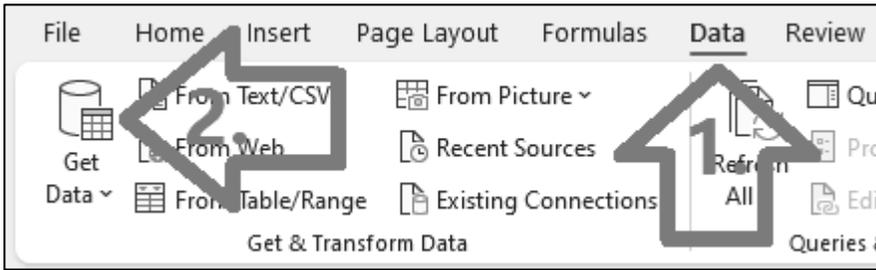
So now let's take a look at how to get data from an Excel file in multiple Power Query implementations. We will get the data from the Excel file that contains our sample database that we will use throughout this book. It can be downloaded from the link mentioned at the end of the book¹.

In this file, among other things, there are holidays for the year 2002, in the table named "Holidays". For the sake of simplicity, we will consider getting the holidays in one year for the time being. In this example, we will use Czech holidays. It will work similarly for holidays in other countries. We have chosen this variant on purpose so that we can show several things in Power Query in the following examples. These holidays are not in a suitable format for later analyses, but in the next part we will show you how to transform them into the desired format.

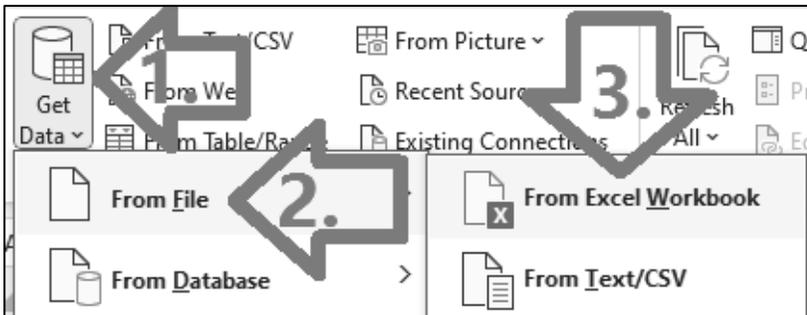
Power Query in Excel 365 and Excel 2019/2021

If we want to get data into Power Query in the latest versions of Excel, we need to go to the "Data" tab in the main menu of Excel, where we click on the "Get Data" button:

¹ Page 750



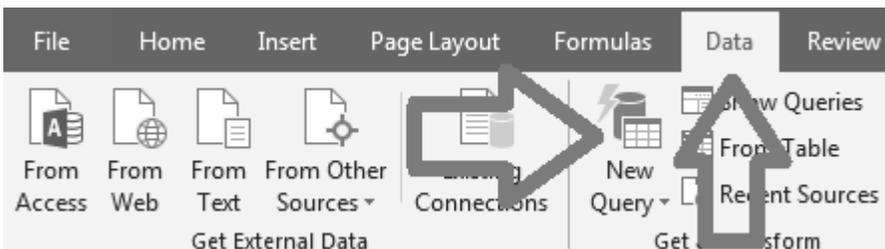
A menu opens where individual categories of supported data sources are displayed. In this menu, you usually find your data source, select it, and connect to it. We now need to connect to the Excel file, so we click on “From File” => “From Excel Workbook”:



After this step, a file selection window will appear. From this step, the rest of the procedure is more or less the same for all versions of Power Query. Therefore, proceed to the section named as “Common Continuation for Desktop Versions”².

Power Query in Excel 2016

In Excel 2016, the procedure is almost exactly the same as in the latest versions of Excel mentioned above. The only difference is that you go to the “Data” tab in Excel's main menu, where you click on the “New Query” button:

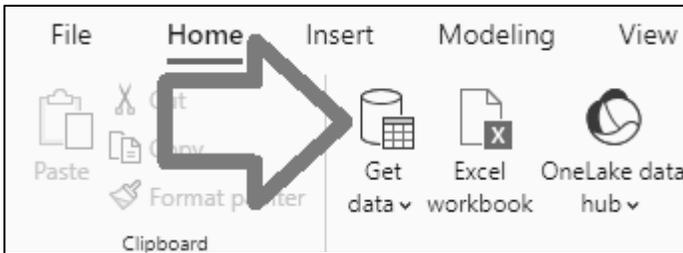


Everything else is then the same as in newer versions of Excel.

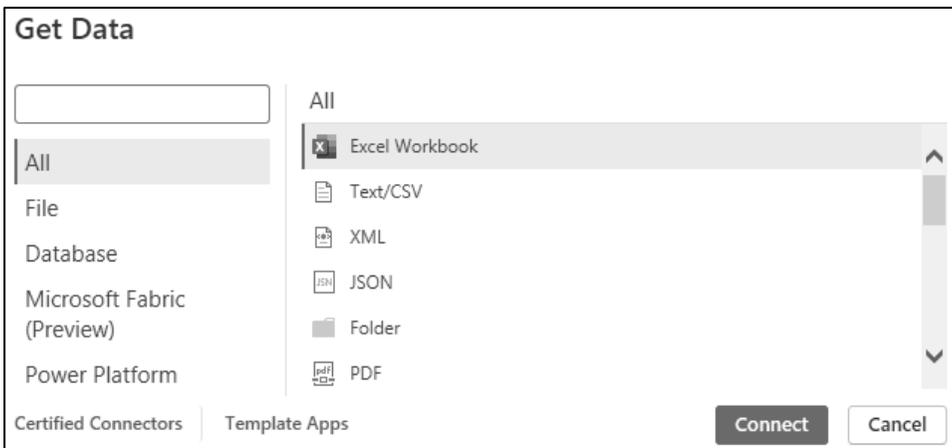
² Page 24

Power Query in Power BI Desktop

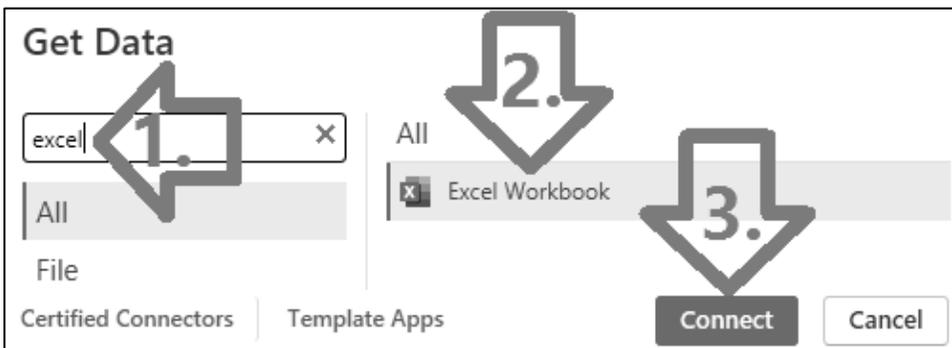
In Power BI Desktop, click on the “Get data” icon in the main menu:



A long list opens showing all supported data sources:



You usually find your data source in this list, select it, and connect to it. We now need to connect to an Excel file, so in this list we will look for a connector named as “Excel Workbook”. If it wasn't located right at the beginning, you don't need to search for it manually. Just type the word “excel” in the top left field, and it should find all the connectors that contain this word in their name. Then click on the connector named as “Excel Workbook”, and then press the “Connect” button at the bottom right:

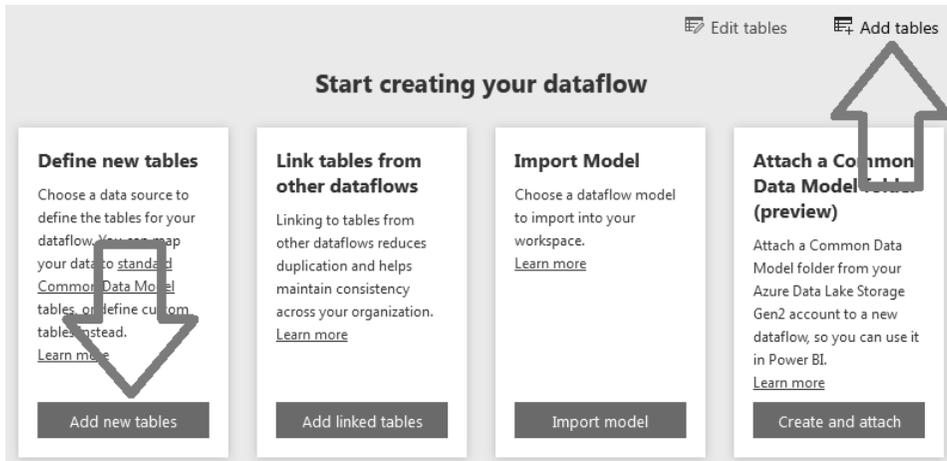


After this step, a file selection window will appear. From this step, the rest of the procedure is more or less the same for all versions of Power Query. Therefore, proceed to the section named as “Common Continuation for Desktop Versions”³.

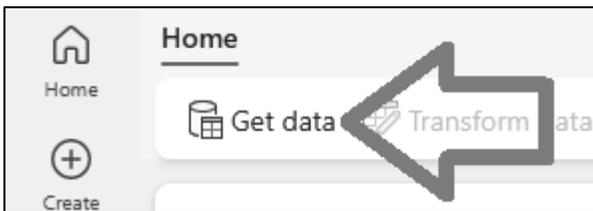
Power Query Online

Power Query Online is implemented in Power BI Dataflows, Power BI Datamarts and in the Dataverse database. The procedure for getting data in them differs only in the first step, and the rest of the procedure is more or less the same for all online implementations.

In Power BI Dataflows, click either on the “Add New Tables” link when you're creating a new dataflow, or click on the “Add Tables” button at the top right if you've already created a dataflow:

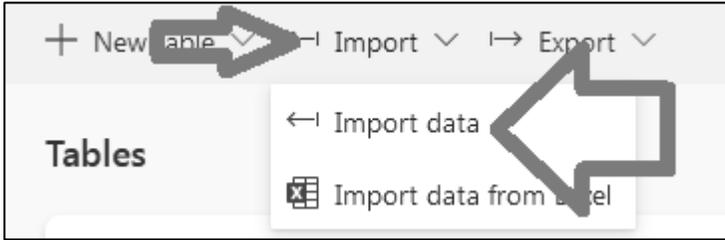


In Power BI Datamarts, click on the “Get Data” link in the main menu:

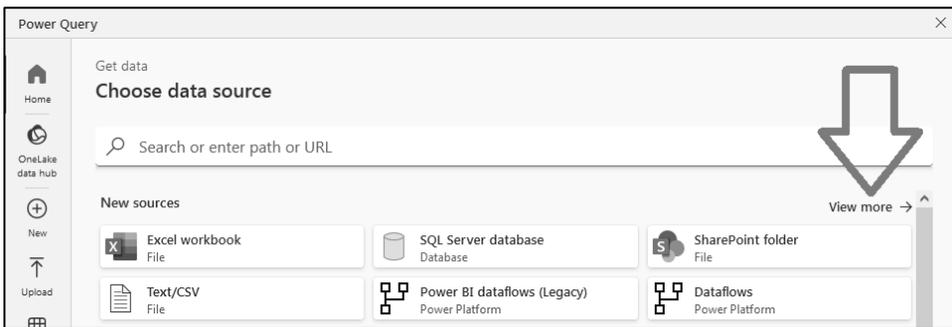


In Power Apps or other services using the Dataverse database, first go to the list of tables in the Dataverse database, and then click on “Import” => “Import data” in the main menu:

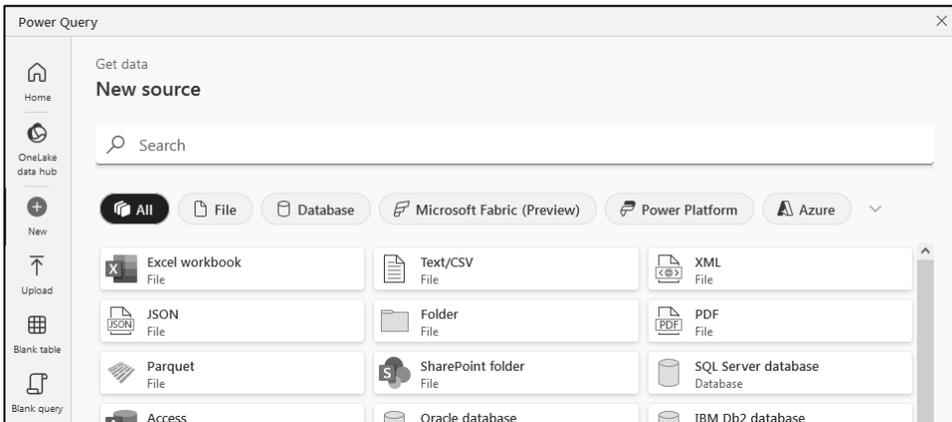
³ Page 24



In all three cases, another page will then open, displaying a selection screen for connectors. This screen only displays a few of supported data sources by default. In order to see them all, click on the “View more” link to display them:



Then, it displays all data sources supported by that version of Power Query:



You usually find your data source in this list, select it, and connect to it. We now need to connect to the Excel file, so we will look for the connector named as “Excel workbook” in this list. If it isn't displayed right at the beginning, then you don't need to search for it manually. Just type the word “excel” in the search box at the top, and it should find all the connectors that contain this word in their name. Then, select the connector named as “Excel workbook”:

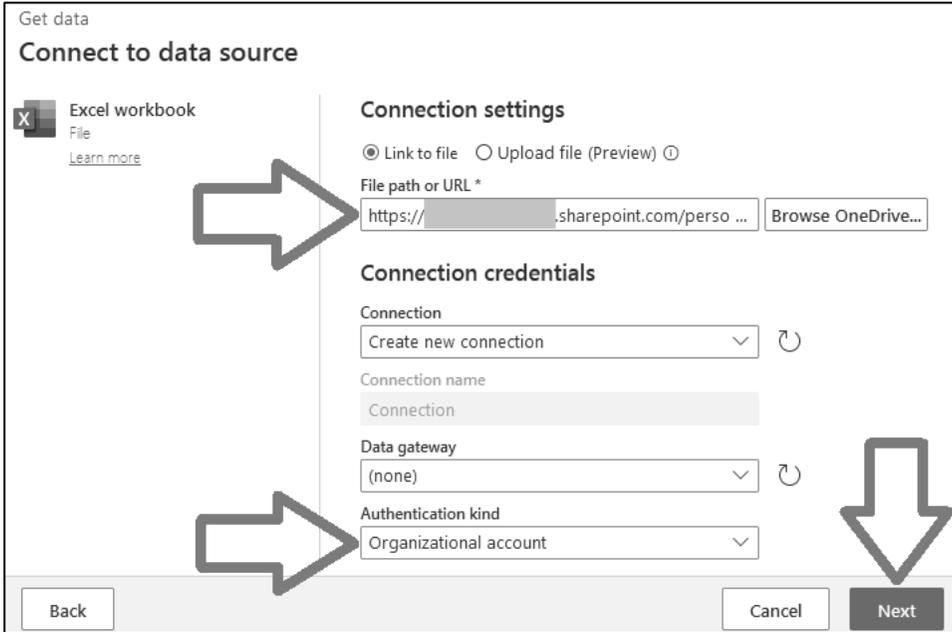


Another page with connection settings will appear (see screenshot displayed below). Such a page is usually displayed for most data sources. Here it is necessary to fill at least some of the listed fields in order to connect to the given data source. In this case, it is necessary to select the Excel file from which we want to get the data. It must be uploaded either to OneDrive, SharePoint, or some other server. In the case of a file on OneDrive, we select it by pressing the “Browse OneDrive...” button, or we can upload it directly to OneDrive and select it directly from there via the “Upload file” option. If the file is located on another server or in another service, for example in MS Teams, it is necessary to enter the URL address of that file.

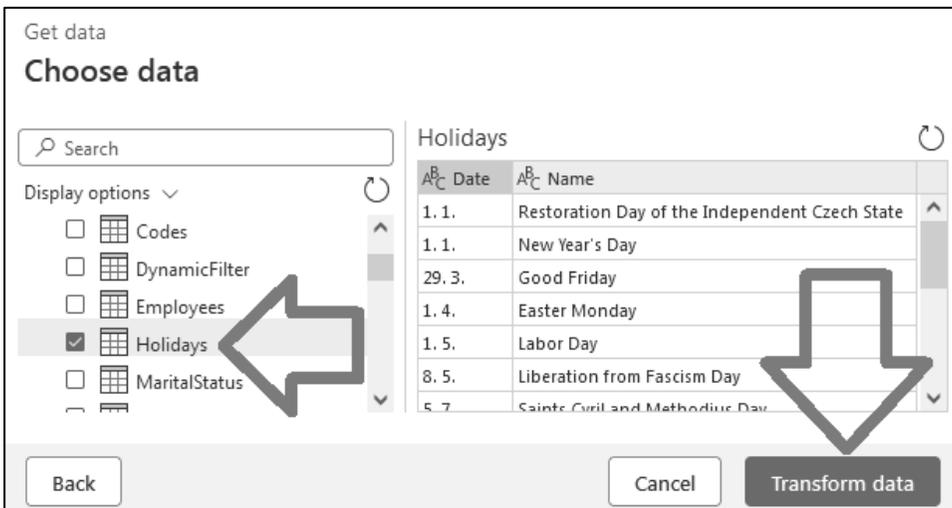
Then in the “Authentication type” field, which is used to specify the type of credentials, we need to select “Organizational account” and then log in to that data source. There are 3 options in this field:

- 1) **Anonymous** – no authentication / no login. Used for publicly available data sources,
- 2) **Basic** – authentication by using name and password (“Basic authentication”). An outdated and not very secure authentication method that should no longer be used today,
- 3) **Organizational account** – authentication with a cloud account to Microsoft cloud services, such as SharePoint, OneDrive, MS Teams, Power BI, etc. You usually log in using your company email address.

When you have filled in all the fields, press the “Next” button:



After successfully connecting to the data source, from this point on, the procedure for all data sources is exactly the same. It is almost identical to the Navigator window in the desktop versions of Power Query. A window will appear where all the tables found in that data source will be displayed on the left. Clicking on the table name will show you a preview of the first few rows from that table:

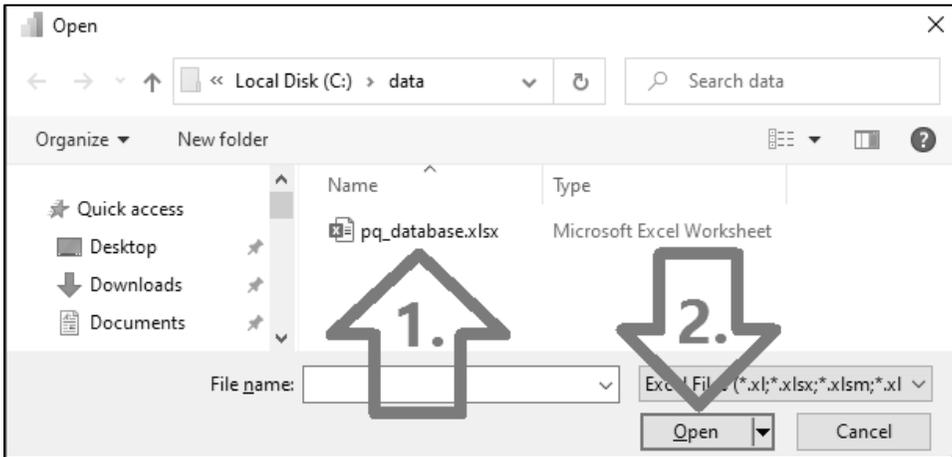


Thanks to this, you can see what data is currently in the table before you get data from it, and you don't need any special tools for that. This is a big advantage, because even if you don't know the database or data source you are connecting to, you can

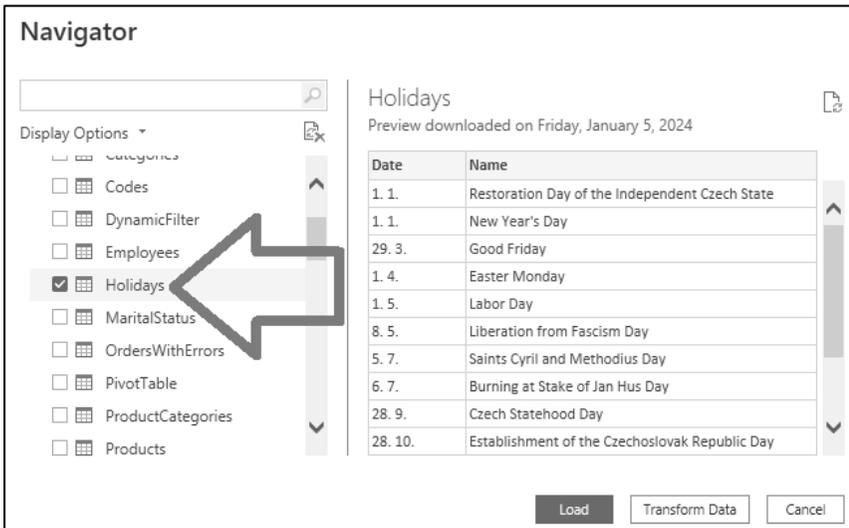
quickly find the tables you need this way. Then, select these tables using the checkboxes next to them, and press the “Transform Data” or the “Next” button at the bottom of the screen, which will take you to the online version of Power Query Editor. From here, working with Power Query Editor is almost identical to working with its desktop counterpart.

Common Continuation for Desktop Versions

When the file selection window is displayed, select the Excel file mentioned above and press the “Open” button:



After successfully connecting to the data source, the procedure for all data sources is exactly the same from this point on. A window named as “Navigator” will appear where all the tables found in that data source will be displayed on the left. Clicking on that table name will show you a preview of the first few rows from that table:



Thanks to this, you can see what data is in the table before you get data from it, and you don't need any special tools for that. This is a great advantage, because even if you do not know the database or data source you are connecting to, you can quickly find the tables you need.

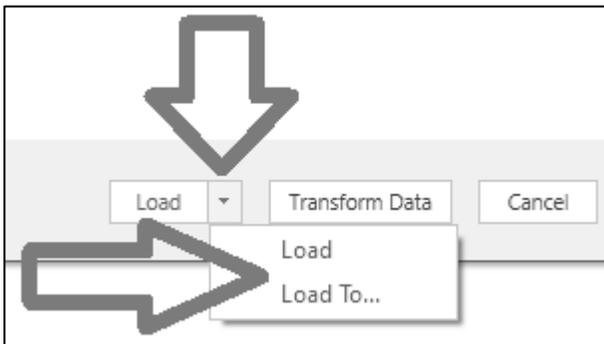
We ended up in a window where the tables are displayed which it found in the Excel file. When you have selected a table or tables (Power BI allows you to select more of them at once), you basically have 2 options, for which you have buttons in the bottom right of that window:

- 1) **“Load”** – loads data into Excel or the Power BI data model,
- 2) **“Transform Data”** or **“Edit”** – launches the Power Query Editor so that you can transform the data before loading. This step can also be done later. The next part deals more with it.

The “Load” button loads the data into the target application as new tables. Therefore, select the “Holidays” table and press the “Load” button to load that table into your target application.

However, in Excel you have 2 options that appear when you click on that little arrow on that “Load” button:

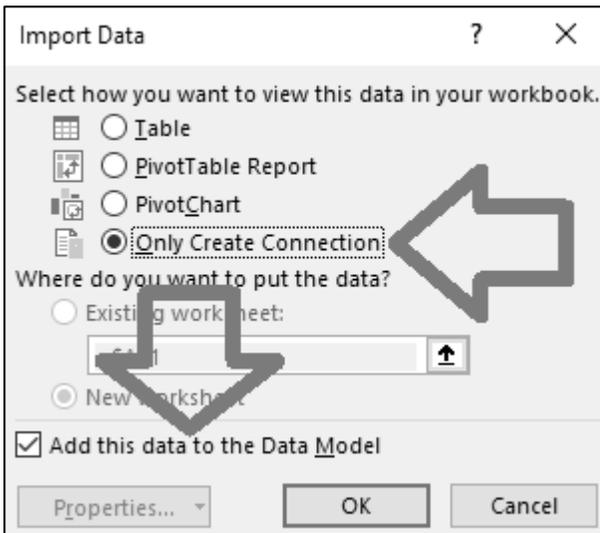
- 1) “Load” – loads data into a new Excel sheet,
- 2) “Load into...” – loads data into another destination, for example into a PowerPivot data model.



And especially the second option is interesting. You use the first one when you only want to load data into Excel. Or also when you want to add them to PowerPivot from there, but that is quite impractical. Especially with millions of rows, you will run into Excel's limitations, and it's also impractical to drag the data into Excel and then have the same copy of the data again in PowerPivot. Plus, such a solution will not later migrate to Power BI as you would expect if you decide to switch to Power BI in the future.

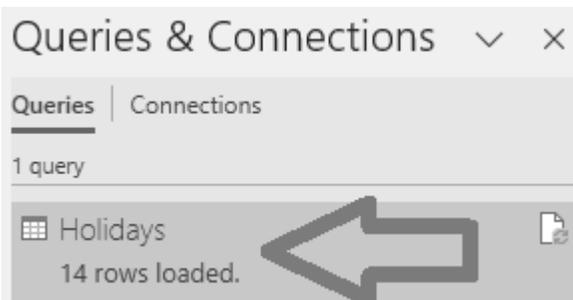
Therefore, we will now take a look at the second option, and specifically how to use it to load data directly into the PowerPivot data model. Such a solution is not only more memory efficient than the previous option. But even in the future, when switching to Power BI, you will appreciate the fact that when you migrate to Power BI, the queries from Power Query, the data model from PowerPivot, and also the linking between them will be correctly imported.

So, when you want to load data directly into the PowerPivot data model, select the “Load to...” option. A window like this will appear:



There, select the 2 settings that are marked in the screenshot – in the upper part of the window, select “Only Create Connection”, and in the lower part of the window, check the “Add this data to the Data Model” checkbox. Then press the “Load” button and wait for your data to be loaded into PowerPivot.

When the data is loaded into Excel or directly into PowerPivot, the “Queries and Connections” panel will appear on the right side of Excel, which is the panel that lists queries in the workbook:



Each loading of data through Power Query creates a **new query** in Power Query, for each table loaded. Because Power Query works with **queries** that usually result in tables. This is where the name Power **Query** comes from. You can later change the query settings at any time by right-clicking on the given query in this panel and selecting the “Edit” option.

If you loaded data into **Power BI Desktop**, it also created a query in Power Query for each table loaded. However, Power BI Desktop doesn't have a dedicated panel for this, and the tables simply appear in the menu bar on the right. Behind each table there is a Power Query query, and it works exactly the same as if you loaded the data directly into the data model in Excel. If you would later like to edit the query, you can access the query editing by clicking the right mouse button on the table name and selecting the “Edit query” option. The following section deals with this in more detail.

The destination where the data is loaded will be called the “**host application**” from now on. Whether it's Excel, Power BI, a Dataverse database, Power BI Dataflows, or any other place that Power Query data is loaded into. This will simplify the following explanations for us, and we will not have to list all such targets separately each time in such a case.

This is how you can get data easily from any data source that Power Query supports. However, that is not all that can be done with it. In the next section, we will demonstrate how queries can be changed and modified so that you can adjust the data to your format. Because only after that you will start seeing what a huge treasure is hidden behind that one innocent button 😊

Power Query Editor & Working with Queries

Now that you know how to basically use Power Query to load data into Excel or a data model, you need to know about another option. Power Query not only allows you to simply retrieve data from a lot of data sources, but it also allows you to transform them into any form and combine them as you wish. Therefore, let's take a look at the real purpose of Power Query, and in particular what it's primarily built for – **transforming data**.

As we mentioned earlier, it often happens that you don't have all the data available in the format you need. If you currently have them, they will not be enough for regular ad hoc analysis anyway, because later you will need to combine these data, e.g. with some handmade Excel spreadsheet. Or with some table from the web, from SAP, etc.

That's why we're now going to see how you can transform the data either before loading it, or even after loading it into the host application. The beginning

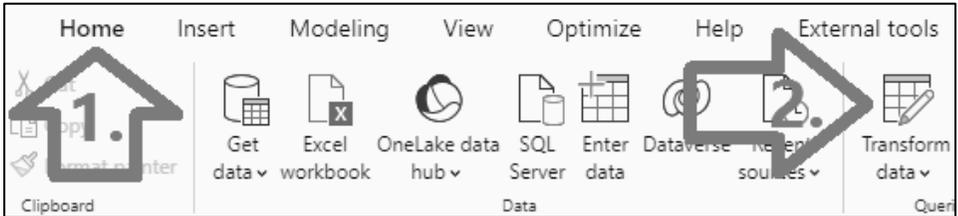
of the procedure will again differ cosmetically depending on which environment you are in. After opening the Power Query window, the entire procedure will be completely identical in all the cases.

So, if you want to transform the loaded data, you must first open the Power Query window, which is correctly called the “**Power Query Editor**”. We will show you the procedure first for Power BI, then for Excel and Power Query Online. In other Power Query implementations, the procedure is similar, that is, if the Power Query Editor is there at all. For example, in Power Query for SQL Server Integration Services, there is only a window for entering a script. Then we will show you how to work with this Editor.

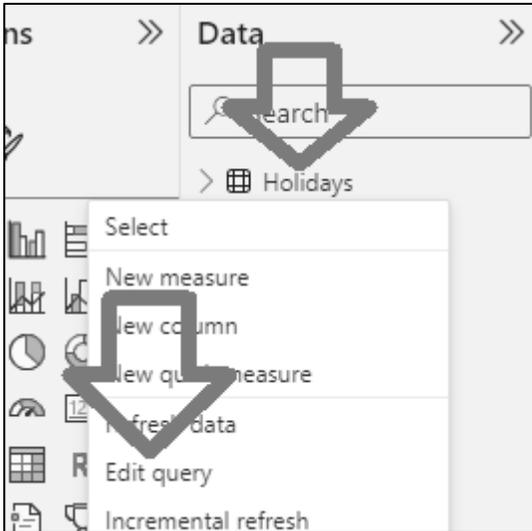
How to Open the Power Query Editor in Power BI Desktop

There are three ways to open the Power Query Editor in Power BI Desktop:

- 1) In the main menu, on the “Home” tab, click on the “Transform data” button:



- 2) Right-click on the table name in the data model menu in the panel on the right, and click on the “Edit query” option:



- When getting new data, click on the “Transform data” button next to the “Load” button:

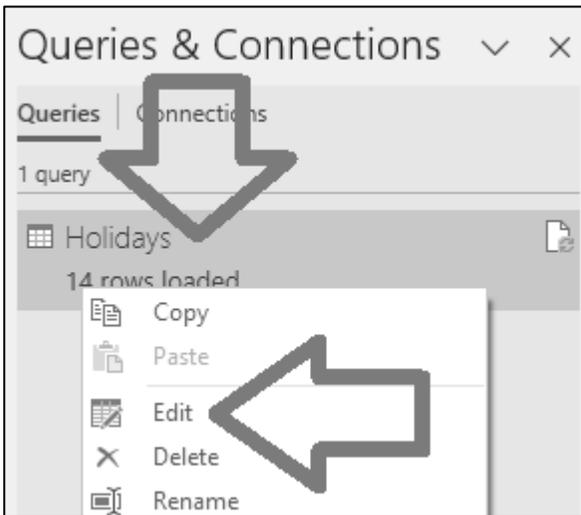


Immediately after that, the Power Query Editor window opens.

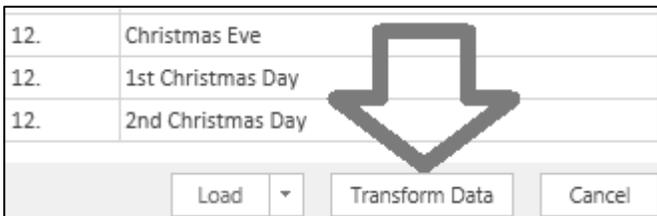
How to Open the Power Query Editor in Excel

There are three ways to open the Power Query Editor in Excel:

- Right-click on the table name in the “Queries and Connections” panel on the right, and click on “Edit”, or simply double-click on the name of the given table/query:



- When getting new data, click on the “Transform data” button next to the “Load” button:

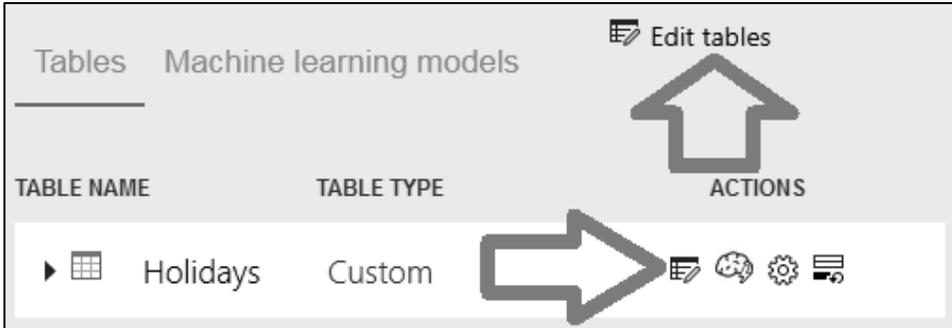


- In the main menu of Excel, click on “Data” => “Get Data” => “Launch Power Query Editor...”.

Immediately after that, the Power Query Editor window opens.

How to Open the Power Query Editor in Power Query Online

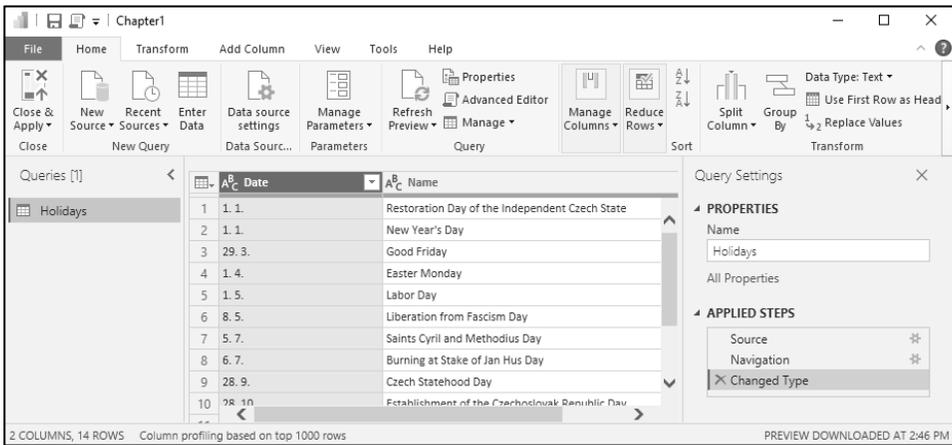
In Power Query Online, you open the Power Query Editor by clicking on the edit table icon next to the given table, or by clicking on the “Edit Tables” button at the top right:



Immediately after that, the online version of Power Query Editor window opens.

Orientation in the Power Query Editor

When the Power Query Editor window opens, it will look something like this:



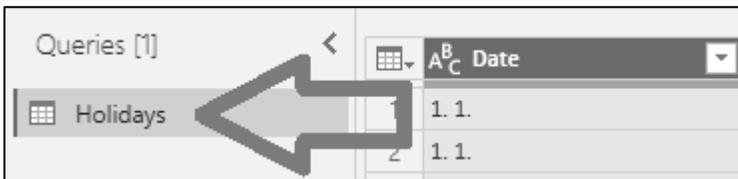
This window looks more or less the same in all Power Query implementations. It differs only in the color of the window, the color and number of buttons, and sometimes also the arrangement of the buttons. Plus, by default, the “Queries” panel on the left is hidden in the Excel version, and displayed in the Power BI version, but you can also display it in Excel as well.

At the top of the Power Query Editor, there are tabs like “Home”, “Transform”, “Add Column” and so on, which are used to transform data. On the left is

the “Queries” panel, which lists all the queries in Power Query and is used to work with them.

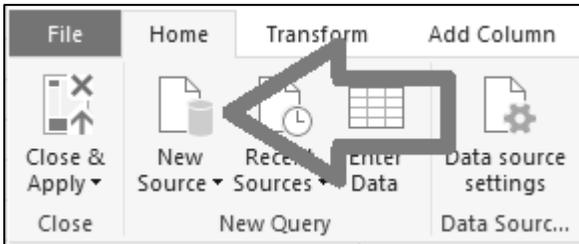
Note that the Power Query Editor also displays a **preview of the data**. And on the right side of the screen, in the “Query Settings” panel, there are some operations in the “Applied Steps” section. This panel is used to manage operations, or transformations, which you later gradually apply to your data in order to use them, step by step, to modify the data into the desired format. And Power Query always shows you a preview of what the data will look like after each such step.

As we mentioned earlier, Power Query works with **queries**. For each table you get with Power Query, a separate **query** is created in Power Query. This query appears in the “Queries” panel on the right, and usually has the same name as the table in the data source:

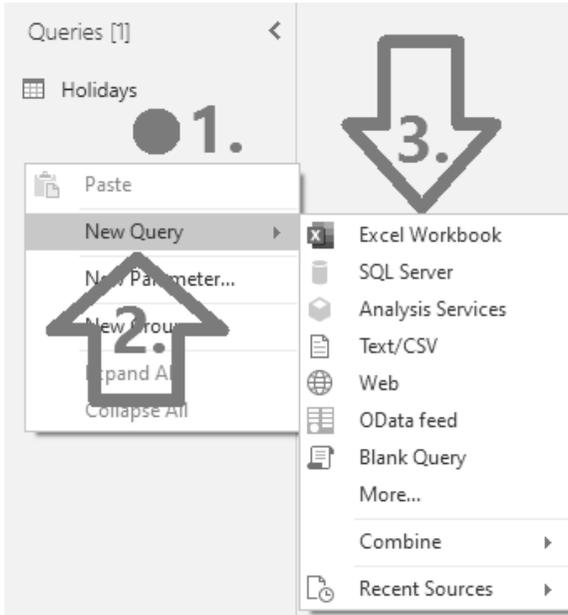


You can also add other queries to this panel without having to go back to the host application. This can be done in two basic ways:

- 1) In the main menu, by clicking on the “New source” or the “Get data” button:



- 2) In the “Queries” panel on the right, by right-clicking on an empty space in this panel, and selecting the “New Query” option, and then a specific data source:



Note: Both options for adding a query are equivalent, and there is no difference between them, even from a technical point of view, despite the different names of the buttons. The different button and menu item names are just a result of continuous renaming of names in Power Query and surrounding software, and the unsystematic implementation of those changes when they didn't rename it consistently everywhere. That's also why there are names from the years of 2013 or 2016, which is on the one hand confusing for the beginners in Power Query, but on the other hand understandable when you already know what's behind it.

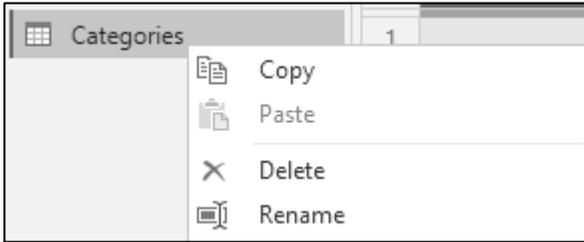
After selecting a specific data source (e.g. Excel) and getting a table from it (e.g. the “Categories” table), this table is added as a new query to that “Queries” panel:



Tables added in this way will also be loaded into the host application later, just like any other tables.

You can then do various activities with these queries. You can, for example, move them to another position with the mouse, or rename the query by double-clicking

on the query name. You can also right-click on the query, and the following menu will appear (only the first items are shown in the picture):

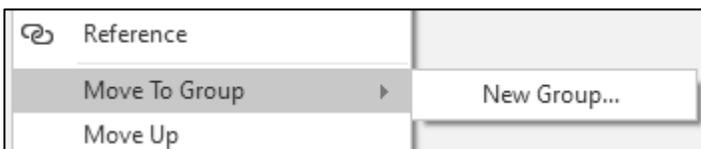


Here are some items from this menu:

- **Rename** – allows you to rename the query,
- **Delete** – deletes the query,
- **Properties...** – allows you to document the query by entering its description,
- **Duplicate** – duplicates the query by duplicating the M script of the original query, making the duplicate independent of the original query (if the original query changes, it will not affect the duplicated query),
- **Reference** – duplicates the query by referencing the original query, thus creating a duplicate dependent on the original query (if the original query changes, it will also affect the results in the duplicated query),
- **Copy** – allows you to copy a query to another Power Query,
- **Advanced Editor** – opens the Advanced Editor for editing the query script,
- **Move To Group** – allows you to group queries into groups and subgroups.

You can use “Move to Group” to **organize your queries into groups** and subgroups. This comes in handy when you have a large number of queries, and you need to group them into groups according to logical units. Or when you want to separate helper queries into a separate group from other queries.

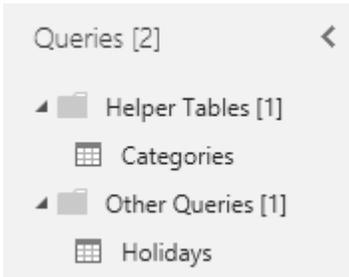
So, if you want to include one or more queries in a specific group, mark these queries (by clicking on them, holding down the Ctrl key if there are more of them), then right-click one of the marked queries and select the “Move To Group” option. If you haven’t created any groups yet, only the “New Group...” option will be available there:



By clicking on this option, a window for creating a new group will appear. There, enter the name of the new group, for example “Helper Tables”, and press the “OK” button:



Then a group with that name will be created, and the selected queries will be moved to that group. At the same time, all other non-grouped queries will be moved to the automatically created “Other Queries” group:



Note: *These groups have no effect on Power Query functionality, such as data transformation or its speed. They only serve to organize more queries to make it easier for you to work with them.*

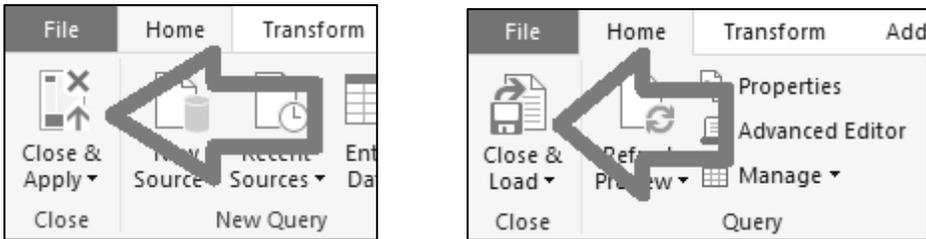
If you have groups already created, and you want to move one query or several queries at once from one group to another, right-click on that query and select “Move To Group”. The group names will be displayed in the sub-menu. By clicking on the name of the given group, the selected queries will be moved to that group. However, there is also an easier way to do this. Just mark queries with the Ctrl key, and then drag the marked queries with your mouse and drop them to the target group.

Subgroups can then be created in the groups. You can do this by right-clicking on the name of a specific group and selecting the “New Group” option. There you

enter the name of the new subgroup, and after confirmation it will create this subgroup for you.

These groups can then be expanded and collapsed using the arrows to the left of their name. This way you can save a lot of vertical space on the screen. Groups can of course also be deleted by right-clicking on the name of a specific group and selecting the “Delete Group” option. However, be aware that this action will delete the group and all queries and subgroups in it. If you want to keep the queries, either move them to another group beforehand, or click on “Ungroup” option instead of the “Delete Group” option. This will move those queries to the “Other queries” group and delete that original group.

After you've finished your work in the Power Query Editor, there's one last step to take – **load this data into the host application**. Because without this, all changes will remain only in Power Query, but won't be reflected in the host application. You can apply all changes easily, in the main menu on the “Home” tab, by clicking on the “Close & Apply” button (in Power BI), or “Close & Load” (in Excel):



In Power Query Online, this button is located at the bottom right and is named as “Save & close”. In other Power Query implementations, this button may be named differently.

Clicking this button will start the **process of loading and transforming all changed tables**. After its successful completion, you will have the data loaded in the host application.

Transforming Data with the Power Query Editor

We are now going to edit the Holidays table from the previous section, using the Power Query Editor. That table is currently not in a usable format for data analysis. Mainly because the dates in the first column of the table are not complete, as they are missing a year. And they are also stored as text, and we need to have them stored as a computer date so that we can easily merge this table with other tables on that date. For example, with a calendar table when analyzing sales data by working vs. non-working days. Therefore, we need to transform these textual semi-dates to reliable computer dates. There's also another problem – the Czechs have two

End of Book Sample

Did you like this sample of the **Professional Power Query & The M Language** book? Would you like to learn much more?

If yes, then you can order it right way on this website:

<https://www.powerquery.sk/professional-power-query-the-m-language-book/>

Thank you.



POWERQUERY.SK