Click to verify

We have to combine several data files, coming from our time & attendance system. The exported files are saved in a SharePoint- or OneDrive-folder, and contain starting rows 'describing' the selection criteria. Is there a possibility to combine these files and automatically remove the starting 12 rows of each file? Kind regards, Maarten Posted : 01/12/2020 5:27 am Hi Maarten, If you get files from a SharePoint list you can then combine them and then in the Query Editor go to the 'Transform Sample File' query and on the Home tab use Remove Rows > Remove Top Rows to remove the top 12 rows. Then in the Query under 'Other Queries' you'll see the consolidated files with the first 12 rows removed from each file. Mynda Posted : 01/12/2020 6:34 am Hi Mynda, Thanks. Unfortunately in a Dutch environment all these names are different. So I had to search some time getting the right place to correct, but I managed. Regards, Maarten Posted : 01/12/2020 9:18 am Requests: Server: route: excel-forum use_home_url: url: pageid:87422 default_groupid: 3 2022 The easiest way to add a row to a manually created table in Power BI is to click the Gear icon next to Source under Applied Steps in the the Power Query Editor. The following example shows how to do so in practice. Example: How Manually Add Row to Table in Power BI Suppose we would like to manually create a table in Power BI. To do so, click the Home tab along the top ribbon, then click the Transform data icon: This will bring up the Power Query Editor. Next, click the Enter Data icon under the New Query group on the Home tab: Next, we will enter the following data that contains information about various basketball players: Once we click OK, the table will be created: Now suppose that we would like to add more rows to the table. To do so, click the Gear icon next to Source under Applied Steps in the the Power Query Editor. This will bring up the Create Table window once again where you can manually add a new row to the table. For example, we may decide to add one new row that contains information about a basketball player on the Magic team: Once we click OK, this new row will be added to the table: Feel free to add as many new rows as you would like. Additional Resources The following tutorials explain how to perform other common tasks in Power BI: How to Add Index Column to Table in Power BI How to Count Distinct Values in Column in Power BI How to Count Number of Occurrences in Power BI I received a question from Sam, who asked how to insert a variable number of rows (X rows) into a table based on a column value. That is, by the number specified in a column. In this post, we'll accomplish this request with Power Query. Before we jump into Excel, let's confirm our objective. Let's say we have a table that contains a company name and the number of passes issued to each, like this: Sam wants to have Excel insert rows based on the number of passes, like this: We'll use Power Query to accomplish this. We'll accomplish this by performing the following steps: Get source into Power Query Create a list Expand to new rows Load results into table Note: this tutorial uses Power Query. You can quickly determine if your version of Excel includes Power Query by clicking the Data tab, and looking for the "Get & Transform Data" group of commands. Time needed: 10 minutesLet's work through the details of each step. Get source into Power Query To get the source table into Power Query, select any cell in the table and click the Data > From Table/Range command. The data preview will appear in the Power Query editor: Create a list Next, we need to create a new column that contains a list for each company. Each list will start at 1 and end at the number of passes. To create such a list, we select Add Column > Custom Column. In the resulting Custom Column dialog, we enter a column name, such as List, and the formula ={1..[Passes]} as shown below:The {curly brackets} tell Power Query we are creating a list. The 1 will be the first value in each list. The two dots .. tell Power Query to fill in the values between 1 and the last value, which in our case is stored in the [Passes] field. If your data has a different field name, you would modify the formula accordingly.Once we hit OK, Power Query displays a new column with a List data type. We no longer need the original Passes column, so we select it and then use the Home > Remove Columns command to delete it. Expand to new rows Select the List column, and then click Transform > Expand. (Or, you can use the expand icon in the upper-right corner of the column label and select Expand to New Rows.)Power Query expands the table by inserting the desired number of rows:All that remains now is to send the results to an Excel table. Load results into table To send these results back to Excel, we click Home > Close & Load To …. Select Table and send to a new or existing sheet as desired. Click OK and … mission accomplished: The nice part about this approach is that it is easy to refresh next period. For example, if we add a bunch of new rows in the source table, all we need to do is right-click the results table and select Refresh. What do you think about this approach? Do you have any alternate ways to accomplish this or improvements to the query? Let me know by posting a comment below … thanks! I have a table that lists transactions. Among other fields, each record lists a description, an account number and an amount. I want to create a column that gives the sum of all transactions with that particular account number. For example, if I have five records as follows, how do I create the sum column in power query? Description Amount Account Sum Column Rent: Room 103 $ 150.00 1001 $600.00 Rent: Room 104 $ 200.00 1001 $600.00 Rent: Room 105 $ 250.00 1001 $600.00 Supplies: Room 103 $ 67.00 1002 $90.00 Supplies: Room 104 $ 23.00 1002 $90.00 Posted : 04/02/2018 3:37 pm Hi Chris, There is a very simple technique you can use for this. First, here is the query I used, I will explain it: let Source = Excel.CurrentWorkbook(){[Name="Table1"]}[Content], #"Changed Type" = Table.TransformColumnTypes(Source,{{"Description", type text}, {"Amount", Int64.Type}, {"Account", Int64.Type}}), #"Grouped Rows" = Table.Group(#"Changed Type", {"Account"}, {{"Sum", each List.Sum([Amount]), type number}}), #"Filtered Rows" = Table.SelectRows(#"Grouped Rows", each ([Account] = 1001)), #"1001" = #"Filtered Rows"{[Account=1001]}[Sum], #"Added Custom" = Table.AddColumn(#"Changed Type", "Sum Column", each Table.SelectRows(#"Grouped Rows", each ([Account] = [Account]))[Account=[Account]]}[Sum]) in #"Added Custom" All I did is to group rows by account (this will provide the sum), applied a filter on Account column to display only one account, and in the Sum column-right click on that value and choose Drill Down, this will create the step #"1001" = #"Filtered Rows"{[Account=1001]}[Sum], I did these steps just to get the right syntax I need to use in the last step, when I add a new column with our sum. In the formula for the new column, I simply replaced the parameter I used for the filter with a dynamic reference to the Account column, combining it with the next step(drill down): Table.SelectRows(#"Grouped Rows", each ([Account] = 1001)){[Account=1001]}[Sum] Table.SelectRows(#"Grouped Rows", each ([Account] = [Account])){[Account=[Account]]}[Sum] Note that in the last #"Added Custom" step, I referred to the #"Changed Type" step, skipping the steps: #"Filtered Rows" and #"1001", which can even be deleted, As I already said I needed those steps just to obtain the correct syntax for my formula. Hope the colors I used will help you understanding what I did. Posted : 05/02/2018 1:06 am Hi Catalin, Thanks for such a detailed answer. I appreciate the work you put into responding to my question. The color coding and attached spreadsheet are most helpful. Posted : 05/02/2018 7:56 pm Awesome. Thanks. To assist I made a 'dummy' table and renamed the approriate columns & the table to match your code. I then copied the code across and bingo it worked. Last thing to do was convert it back to my actual column names etc Posted : 21/03/2019 9:06 pm Requests: Server: route: excel-forum use_home_url: url: pageid:87422 default_groupid: 3 2022 Hello! I would like to know if it is possible through PowerQuery to insert rows in a table based on certain criteria/condition In my case I have the initial table: PS.: Here in Brazil our date format is dd.mm.yy Date CustomerName NumContract DurationContract ValueContract Status TotalPoints 9/7/14 Carlos 110 3 anos 9000 Ativo 30.000 15/7/14 Ricardo 111 3 anos 12000 Cancelado 60.000 My goal is to achieve the following table: Date CustomerName NumContract DurationContract ValueContract Status TotalPoints TotalYears YearValue YearPoints CurrentYear StartDate EndDate 9/7/14 Carlos 110 3 anos 9000 Ativo 30.000 3 3.000 10.000 1 9/7/14 8/7/15 9/7/14 Carlos 110 3 anos 9000 Ativo 30.000 3 3.000 10.000 2 9/7/15 8/7/16 9/7/14 Carlos 110 3 anos 9000 Ativo 30.000 3 3.000 10.000 3 9/7/16 8/7/17 15/7/14 Ricardo 111 3 anos 12000 Cancelado 60.000 3 4.000 20.000 1 15/7/14 14/7/15 15/7/14 Ricardo 111 3 anos 12000 Cancelado 60.000 3 4.000 20.000 2 15/7/15 14/7/16 15/7/14 Ricardo 111 3 anos 12000 Cancelado 60.000 3 4.000 20.000 3 15/8/15 14/7/17 The first step is to insert rows: If PrazoContrato = 3 anos (3 Years) so we have to have 3 rows for that Contract The second step is about to insert new (6) columns on the right based in certain conditions - TotalYears = > Same as DurationContract but as value not - YearValue = > is equal ValueContract divided TotalYears - YearPoints = > is equal TotalPoints divided TotalYears - CurrentYear = > here in each row of the contract we set the current year for that contract - StartDate = > here in each row of the contract we set the start date of the current year - EndDate = > here in each row of the contract we set the end date of the current year Is that possible to transform the original table as mentioneb above through PowerQuery? Any help it would be very appreciated Best Regards Posted : 08/10/2017 1:13 pm Hi Jose, Thanks for sharing a sample file. In the attached I've inserted the rows for you. Please inspect the 'Applied Steps' to see how it was done as you will no doubt want to add more conditional columns to allow for contracts that span more than 4 years. In regards to the other columns, I think adding these should be fairly straight forward for you now, but if you have any troubles please come back. Mynda Posted : 10/10/2017 7:44 am There is a nice trick I use to create new rows, which may be very useful in case your data does not have an equal amount of rows to be added. This way you can add a variable amount of rows to each existing rows. You can simply add a column with this formula: = Table.AddColumn(Source, "Custom", each Text.Repeat("a",[Year]-1)&"a") Where the Year column holds the number of rows needed for each existing row. All you have to do now is to Split this new column by comma delimiter, the trick is to split into rows, not into columns. This action will also copy all other columns, no need to fill down. See the attached file for an example. Posted : 11/10/2017 12:01 am Requests: Server: route: excel-forum use_home_url: url: pageid:87422 default_groupid: 3 2022 I was looking for a simple way in Power BI to manually add a row to an existing table using Power Query, which is not something that is supported through the menu of Power Query, but is something you can accomplish through the Advanced Editor. For this example I want to add a row to a dimension table "Ticket priority" to identify unknown or null rows in my fact table "Tickets". The ticket priority table looks like this. I want to add a new row with the values 0 for priority_id and Unknown for priority. To do this, click the Advanced Editor button in the menu. The button is on the Home tab and on the View tab. After clicking on the button, the Advanced Editor window will show you Power Query M code. If you're unexperienced in this area, I will not in depth in explaining how Power Query M works. There are enough resources to find for this. But I will explain how the lines of code that you see correspond to the applied steps that you might know. In the code you see, every line corresponds to an applied step that you see in the Power Query editor in the right side. In Power Query M and in this example, everything between the let and in statement will define a step. Each line of code starts with the name of the step, followed by a equal sign and then followed by a Power Query M function. Each step line ends with a comma. Each line will generate a table as output. The line after the in statement defines which step name will provide the table output. In this case the step name of the last step #"Changed Type". Now lets add the new step using a line of Power Query M code. To start, add a comma after the #"Changed Type" step line and press the Enter button to add a new line. Then, copy and paste the following line: #"Add Unknown row" = Table.InsertRows(#"Changed Type", 0, {[priority_id=0, Priority="Unknown"]}) With this line we create a new step named "Add Unknown row" and use the function Table.InsertRows to add a new row to the table. To find out how the function Table.InsertRows works, look at the documentation from Microsoft. The syntax explanation is as follows: Table.InsertRows(table as table, offset as number, rows as list) as table So Table.InsertRows will take a table, an offset and a list of rows as parameters. In our case the table is the output table of the previous step #"Changed Type", the offset is 0 (zero) to put the new row at the top of the table. Between the brackets the new row is defined. We'll set the value of the priority_id column to 0 (zero) and the Priority column to "Unknown". (When you have more columns, just add these columns between the brackets separated by a comma and define the column name and value as in the example.) And lastly we change the line after the in statement and change it to #"Add unknown row", to use the output table of the step we just added. After clicking the Done button you will see under the applied steps, our new step has been added and in the table you see that the first row is our new Unknown row. Table.InsertRows is a Power Query M function that returns a table with a list of rows inserted into the table at a given position. The function returns a table with the new rows inserted at the specified offset. Compatible with: Power BI Service Power BI Desktop Excel Microsoft 365 Table.InsertRows(table as table, offset as number, rows as list,) as table The Table.InsertRows function allows you to insert one or more rows into an existing table at a specified position. The offset parameter specifies how many rows to skip before inserting rows into the table. Note that the inserted rows must match the column types of the table. Let's look at some examples of the Table.InsertRows function in Power Query. We'll start with a simple dataset that will be used in our examples. Below is a table named Source: Sometimes, you may notice missing rows in your data. In this case, the Index column is missing a value of 4. To fix this, you can use the Table.InsertRows function to insert a new row with the appropriate values after the third row. Here's how you can do it: Table.InsertRows(Source, 3, { [Index = 4, Name = "Paul", Age = 19] }) In this expression: Source refers to the original table. 3 specifies the position in the table where the new row will be inserted (in this case, after the third row). The list { [Index = 4, Name = "Paul", Age = 19] } contains a single record, which holds the values for the new row. After running this function, the table will look like this: Now, the Index column contains a consecutive sequence from 1 to 5, and the missing row has been successfully inserted. You can also insert multiple rows at once by providing a list of records to the Table.InsertRows function. This is useful when you need to add several rows in one go. Here's how you add multiple rows at the same position in the table: Table.InsertRows(Source, 3, { [Index = 4, Name = "Paul", Age = 19], [Index = 4, Name = "Paul", Age = 19] }) In this example, we add the same record twice. They are both added after the third row in the table, with the following table as outcome: Both new rows are inserted after the third row, and the table now contains the added data. To try the full M code, you can paste the below query into the advanced editor: let Source = #table(type table [Index = Int64.Type, Name = text, Age = Int64.Type], { { 1, "Rick", 32 }, { 2, "Tom", 59 }, { 3, "Tanya", 27 }, { 5, "Melanie", 61 } }), InsertSingleRow = Table.InsertRows(Source, 3, { [Index = 4, Name = "Paul", Age = 19] }), InsertMultipleRows = Table.InsertRows(Source, 3, { [Index = 4, Name = "Paul", Age = 19], [Index = 4, Name = "Paul", Age = 19] }) in InsertMultipleRows Learn more about Table.InsertRows in the following articles: Manually Insert Rows to a Table in Power QueryLearn how to insert rows into a table in Power Query. Useful for creating journal entries, or adding additional information. » Read more Other functions related to Table.InsertRows are: 2023-2025 © BI Gorilla. All rights are reserved. Information from Microsoft docs is property of Microsoft Corp. | Privacy Policy There are situations where you may manually want to add rows to a table. For example, when creating the bookings for journal entries using Power Query. In this post, you learn how to create records, sum columns, and add rows to tables using Power Query. Table of contents A client once asked me if we could create a journal entry in Power Query. Each month, they needed to upload bonus reservations in their accounting system. And the software allowed them to upload a CSV file, as long as it was in the required format. Since this was such a repetitive process, it was the perfect candidate for automation. After all, the only parameters it had were the bonus amount-bookings on the one hand and the counter-booking of the reservation on the other hand. In accounting, a journal entry needs to have the same debit as a credit amount. For the purpose of this solution, it has to meet the following requirements: The bonus reservations need to be specified on the employee level on account 1620 The total reservation for the month is booked on General Ledger 1420. The journal entry needs a booking Period The dataset originated from an Excel worksheet. The bonus amount an employee receives is calculated and imported into Power Query. The result of that is the below picture: From the starting position, you will now learn how to apply the relevant transformations. The visible lines amount to the total Bonus reservation and are booked on the Debit side of the journal entry. You can add a custom column, name it General Ledger, and enter the account number in the formula box = 1620. This will result in the following formula: = Table.AddColumn(#"Changed Type", "General Ledger", each 1620) And your dataset changes to: We don't need to add any parameters here, because the bonus reservation is always booked on this account. A journal entry has the same amount of debit as it has credit. Our next step is to calculate the credit amount of the journal entry. You will add a new record to the dataset, with the credit booking of the total bonus amount. But how do you calculate the total sum of the Amount column? The easiest way is to select the Amount column. Then navigate to the Transform tab, click on Statistics and select the Sum operation. A new step will be created with the below code: = List.Sum(AddGeneralLedger[Amount]) We can use this code when generating a new record in the dataset. Before moving on, what does the above code do exactly? In Power Query, referencing a column results in a list containing all the values from the column. You can do that by writing the Previous Stepname and putting the Column Name between square brackets: = AddGeneralLedger[Amount] If you add a new step (by clicking the Fx button) and paste the above code, the result will be a list like below: Since the previous result is a list, you can wrap it around the List.Sum function, to Sum up all the values in the list. We now know the total bonus amount. What comes next is to create a new record from scratch. You will now learn how to create a record, with a column for each of the existing columns in the table. In this example, you have the columns Name, Amount and General Ledger. But first, how do you create a record? Creating a record in Power Query is easy once you know it. You can add a record with a single column by pasting the below code in the formula bar: = [Column1 = "A"] When you require multiple columns, you can expand the formula with the below syntax. = [Column1 = "A", Column2 = "B", Column3 = "C"] The general syntax for creating a record is: Start and end the formula with square brackets Between the square brackets, provide the column names without any quotations, followed by an equal sign and a value. Separate column names by a comma. The column value can be anything. It could be a number, text, list, record, table, binary item, etc. Anything that can be put into a regular table cell is valid as a column value. For our exercise, you can create the necessary record as follows: =[Name = "Bonus Reservation", Amount = - List.Sum(AddGeneralLedger[Amount]), General Ledger = 1420] In the previous section, we created a formula for the Amount value. The only change now is that it is displayed negatively to represent the credit amount. We renamed the step to CreditRecord, and the query now appears as follows: So far, we have prepared our data for the debit and credit side of the journal entry. We will now combine the data. There are several functions that can do that. Power Query has a function called Table.InsertRows. This function takes an existing table and adds rows to it. The three arguments for the function are: The table to add rows to The (index) position where the new rows will be inserted The new rows to add to the dataset, are input as a list With this function, we can easily combine our previous steps. Start by writing the following code: = Table.InsertRows(AddGeneralLedger, 0, CreditRecord) AddGeneralLedger is the step that contains all the debit journal entries, and CreditRecord contains the rows we want to insert. You will notice that Power Query throws an error saying: "Expression.Error: We cannot convert a value of type Record to type List." When you look closely at the third argument, it indicates that the new rows added, need to be provided in list form. With a simple adjustment, you can fix the code. Wrapping the CreditRecord between curly brackets turns it into a list. = Table.InsertRows(AddGeneralLedger, 0, { CreditRecord }) And with that, the data combines as in the below picture: Allow me to show you another way of combining the data. The default function Power Query uses to combine tables is Table.Combine. This function is used when you apply the Append Queries operation. Table.Combine takes a list of tables as its main argument. A wrong solution would be to write: = Table.Combine({ AddGeneralLedger, CreditRecord }) The error you get is: "Expression.Error: We cannot convert a value of type Record to type Table. The reason is that CreditRecord is provided as 'Record' object, and the function Table.Combine expects a table. What you could do instead is: = Table.Combine({ AddGeneralLedger, Table.FromRecords({ CreditRecord }) }) What's different from the wrong solution is: The CreditRecord is transformed to a table using the Table.FromRecords function. Both tables are wrapped between curly brackets and separated by a comma. As you can tell, the data types provided to function are very important in Power Query. Getting comfortable manipulating your data types, allows you to use different ways to get to the same end result. Either one of the two methods will combine your data into a single dataset. Now, there is only one thing left to do: add a book period. Our last requirement is to add a booking period to the journal entry. This value needs to change for each journal entry. To make things easy for your user, you can create a parameter for the booking period. To do that: Right-click in the query pane and select new parameter. Name your parameter BookingPeriod. Tip: not having any spaces in the name makes it easy to reference the parameter in your code. Change the type to Date and enter a current value. In this example, I used 31/01/2021. With the parameter in place, you can now add a custom column. Name this new column Booking Period, and add the formula: = BookingPeriod Change the new column Booking Period to a date type. With all those steps combined, you know how to create a journal entry in Power Query And that's how you can create journal entries using Power Query. If you want to have a look at the file itself, you can find it here: Enjoy Power Query!