Table of Contents

1

In this first section we will see how to transform a monthly report from being updated manually to automatically. Along the way, we'll see how classic Excel features like tables and SUMIFS can help somewhat, and then how modern Excel features like Power Query and the data model can help even more. Each post in the series provides the next step in the automation sequence.

I named this series "Stop Wasting Time" because when we spend more time than is necessary to do a task, we are wasting time. We tend to update our Excel workbooks using manual steps, just like last month. And we feel too busy to learn a more efficient way. So we just keep doing the same manual steps every month. Meanwhile, Excel is just sitting there, waiting to help!

If you like sports, imagine you are the coach and Excel is sitting there on the bench. Excel is like your star player, and you aren't putting Excel in the game. Excel is sitting on the bench, waiving its arms saying "coach, coach ... please put me in the game!"

This series is designed to demonstrate how to get Excel into the game. Throughout the series, we'll discover Excel knowledge that will help us move our workbooks from manual to automated. After all, if we are busy, we certainly wouldn't want to spend more time doing something than is needed. So, that is the big picture for the series, and I hope you enjoy it!

Report

Before we get too far, let's take a look at our sample report. Since I have an accounting background, I'll illustrate this process with a classic financial statement. However, the steps and process can be applied broadly to many other types of reports.

Here is the report that we'll improve over the next few posts:

Report	
ASSETS	**Amount**
Cash and Cash Equivalents	89,882
Prepaids and Deposits	61,452
Accounts Receivable	11,655
Inventory	23,984
Fixed Assets	79,340
TOTAL ASSETS	266,313
LIABILITIES AND EQUITY	
Accounts Payable	19,426
Accrued Expenses	74,854
Loans	36,145
TOTAL LIABILITIES	130,425
Retained Earnings	45,501
Net Income	90,387
TOTAL EQUITY	135,888
TOTAL LIABILITIES AND EQUITY	266,313

And, here is the source data, which changes every month:

AcctName	FS Line	Amount
Checking account	Cash and Cash Equivalents	25,573
Money market	Cash and Cash Equivalents	39,441
Savings	Cash and Cash Equivalents	24,868
Prepaids	Prepaids and Deposits	47,562
Deposits	Prepaids and Deposits	13,890
Accounts Receivable	Accounts Receivable	11,655
Inventory	Inventory	23,984
Furniture and Fixtures	Fixed Assets	11,778
Computer Equipment	Fixed Assets	19,393
Machinery	Fixed Assets	24,014
Automobiles	Fixed Assets	24,155
Accounts Payable	Accounts Payable	19,426
Payroll liabilities	Accrued Expenses	42,663
Taxes	Accrued Expenses	32,191
Loan from officer	Loans	36,145
Retained earnings	Retained Earnings	45,501
Current year earnings	Net Income	90,387

Our goal is to automatically get the new monthly data into the report.

We'll start off easy, and assume the data is already cleaned up and in Excel. But as the series progresses, I'll increase the complexity of the illustration so we can learn just how powerful the modern Excel tools are.

Since we are going to show the full progression of this report, from manual to automated, we'll start by reviewing the manual update process.

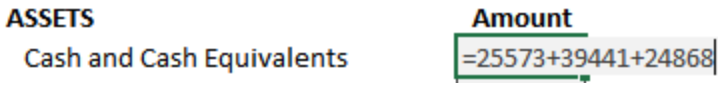
Manual Updates

All Excel users are on a journey, and we are at different places along the path. But, we all start at the beginning. We all open Excel for the very first time and see the grid. We get in and start playing around and figuring things out. Early on in our journey, we end up doing a lot of things manually. Why? Just because we haven't learned more efficient options. As we learn more, we are able to automate more. And our journey continues like this ... learning and improving as we go.

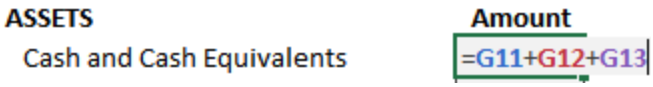
But, sometimes, we can get to a certain point and feel comfortable enough to stop improving our workbooks. Perhaps we get busy at work and decide the workbooks are good enough. We know enough to update the monthly reports. Even though it takes many manual steps, at least we know how to get them done. Sure, there may be a bunch of manual steps, and sure we may have to work after hours and weekends. But, at least we are able to get them done.

But here's the thing: we can delegate A LOT of manual tasks to Excel. That is, we can automate a lot of stuff with Excel. I like to think about it like this. At the very beginning of our Excel journey, we do things manually. But our goal is to get to the place where our workbooks are automated. And the thing that moves us from "manual" to "automated" is Excel knowledge. I visualize it like this:

Along our journey of Excel knowledge, we learn lots of cool things that help us work faster. At the beginning of our journey, we may update our report by using Excel like a digital 10-key. Perhaps we'd use formulas like this:



And that approach allows us to finish the report. So, we use it ... for a while. Until we discover that each time the data changes, we have to literally rewrite every formula. Trying to figure out a better way, we decide on a different approach. A more efficient approach. One that doesn't require us to rewrite every formula when the values change. We use direct-cell references, like this:



This is an improvement because as the cell values change, they flow into the report automatically and we don't have the rewrite formulas.

But, then we discover that this approach is fragile, and breaks easily. For example, our formulas break when the data values occupy different cells, the data is sorted in a different order, or a new account is added. When the formulas break, we have to rewrite them...manually.

So, we set out to improve the workbook with some classic Excel features.

Classic Excel to the Rescue

Our current approach has some issues ... basically, changing the sort order breaks the formulas and adding new accounts requires us to rewrite formulas. So, let's take them one at a time. We'll address the sort order with SUMIFS and adding new accounts with Tables. Let's start by converting our ordinary data range to a Table.

Tables

When we have a data range that may expand, for example new accounts or transactions being added, it is a good idea to store that data in a Table rather than an ordinary range. Why? Because tables auto-expand to include the new transactions. And we can refer to the data in a table with structured table references rather than A1-style range references.

To convert our ordinary range into a table, start by clicking any cell in the data range, like this:

AcctName	FS Line	Amount
Checking account	Cash and Cash Equivalents	25,573
Money market	Cash and Cash Equivalents	39,441
Savings	Cash and Cash Equivalents	24,868
Prepaids	Prepaids and Deposits	47,562
Deposits	Prepaids and Deposits	13,890
Accounts Receivable	Accounts Receivable	11,655
Inventory	Inventory	23,984
Furniture and Fixtures	Fixed Assets	11,778
Computer Equipment	Fixed Assets	19,393
Machinery	Fixed Assets	24,014
Automobiles	Fixed Assets	24,155
Accounts Payable	Accounts Payable	19,426
Payroll liabilities	Accrued Expenses	42,663
Taxes	Accrued Expenses	32,191
Loan from officer	Loans	36,145
Retained earnings	Retained Earnings	45,501
Current year earnings	Net Income	90,387

Then select the **Insert > Table** command. Excel displays a confirmation dialog. We click OK and our table is created:

AcctName	FS Line	Amount
Checking account	Cash and Cash Equivalents	25,573
Money market	Cash and Cash Equivalents	39,441
Savings	Cash and Cash Equivalents	24,868
Prepaids	Prepaids and Deposits	47,562
Deposits	Prepaids and Deposits	13,890
Accounts Receivable	Accounts Receivable	11,655
Inventory	Inventory	23,984
Furniture and Fixtures	Fixed Assets	11,778
Computer Equipment	Fixed Assets	19,393
Machinery	Fixed Assets	24,014
Automobiles	Fixed Assets	24,155
Accounts Payable	Accounts Payable	19,426
Payroll liabilities	Accrued Expenses	42,663
Taxes	Accrued Expenses	32,191
Loan from officer	Loans	36,145
Retained earnings	Retained Earnings	45,501
Current year earnings	Net Income	90,387

Tables have many advantages. One advantage is that they auto-expand. When you type or paste new values immediately under (or right), the table will automatically expand to include it. This means if we enter a new account, for example Payroll Checking which maps to the Cash and Cash Equivalents FS Line, it will be automatically included in the table. But, we need to understand how to reference table ranges in our formulas. So, let's talk about names.

Tables have names. You can view/set the name with the **TableTools > Table Name** field. Our table is named **Table1**, and we can use that name in our formulas to reference the table data. But, we can also reference a single table column by using a structured table reference. To reference a specific column, we use the table's name and then the column name in [square brackets], like this: **Table1[Amount]** or **Table1[FS Line]**.

Now, let's see how we can use SUMIFS to summarize the data in our new table.

SUMIFS

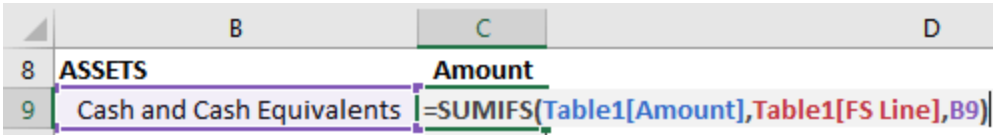
SUMIFS is a wonderful function, and it is designed to add up numbers in a column based on matching row labels. The first argument identifies the column of numbers to add, and the remaining arguments come in pairs. Each pair defines a condition. Officially, it looks like this:

=SUMIFS(sum_range, criteria_range1, criteria_value1, ...)

But, I've come up with a narrative I use to help me remember the arguments. Each **bold** word in the narrative represents an argument:

"Add up **this** column of numbers, but only include those rows where **this** column is equal to **this** value."

For example, if we wanted to add up the table's **amount** column, but only include those rows where the **FS Line** column matches our report's **label**, we could use something like this:



We fill the formula down to compute the remaining report values:

Report	
ASSETS	**Amount**
Cash and Cash Equivalents	89,882
Prepaids and Deposits	61,452
Accounts Receivable	11,655
Inventory	23,984
Fixed Assets	79,340
TOTAL ASSETS	266,313
LIABILITIES AND EQUITY	
Accounts Payable	19,426
Accrued Expenses	74,854
Loans	36,145
TOTAL LIABILITIES	130,425
Retained Earnings	45,501
Net Income	90,387
TOTAL EQUITY	135,888
TOTAL LIABILITIES AND EQUITY	266,313

Same report as before, but this approach is more efficient to update each month because we aren't rewriting formulas when the sort order changes. Yay!

Recap

We started our journey by performing manual updates. To improve our report, we added a table so that as new rows are added to the table they are automatically referenced by the formulas. We then used the SUMIFS function to compute the report values.

We moved the report to the right a bit ... from being manually updated to being more automated:

We still have many improvements to make, and we'll cover them in the upcoming posts in the series ... stay tuned!

Additional Resources

- Sample file: StopWasting1.xlsx
- Learn more about SUMIFS
- Learn more about using Tables with SUMIFS

2

This is the second part in the Stop Wasting Time series. In this section, we learn how Power Query can help us update our report in less time. After all, why would we want to spend more time than necessary updating it? That is called "wasting time" and we just don't like wasting such a precious resource. So, let's get to it.

Quick Recap

As a quick recap, we started with a report like this:

Report	
ASSETS	**Amount**
Cash and Cash Equivalents	89,882
Prepaids and Deposits	61,452
Accounts Receivable	11,655
Inventory	23,984
Fixed Assets	79,340
TOTAL ASSETS	266,313
LIABILITIES AND EQUITY	
Accounts Payable	19,426
Accrued Expenses	74,854
Loans	36,145
TOTAL LIABILITIES	130,425
Retained Earnings	45,501
Net Income	90,387
TOTAL EQUITY	135,888
TOTAL LIABILITIES AND EQUITY	266,313

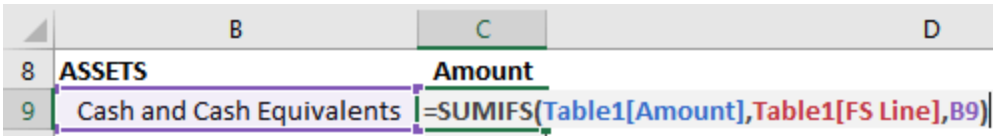
It started off being manually updated with a formula like this:

ASSETS	**Amount**
Cash and Cash Equivalents	=25573+39441+24868

But, we had to re-write all of the formulas each month as the values changed. So, we improved it with formulas that referenced the data cells, like this:

ASSETS	Amount
Cash and Cash Equivalents	=G11+G12+G13

That made it better, but our formulas broke when the values moved cells. For example, when the data was sorted. Plus, if new rows were added to the data range we would need to rewrite formulas. So, we improved the workbook by storing the data in a table and using SUMIFS, like this:



	B	C	D
8	ASSETS	Amount	
9	Cash and Cash Equivalents	=SUMIFS(Table1[Amount],Table1[FS Line],B9)	

And that is about as far as we'll go using "classic" Excel items. It is time to move to some "modern" Excel tools.

The terms Classic Excel and Modern Excel were used by my Excel-friend Rob Collie. Essentially, Classic Excel represents the Excel stuff we've been using for a long time ... formulas, functions, features, PivotTables, VLOOKUP, and so on. Modern Excel has been here for the past couple of Excel versions and include the power tools like Power Query, Power Pivot, the data model, and so on.

Power Query

Now, we'll take the next step on our journey and use Power Query. In the first post of the series, I simplified the data. As a reminder, it looked like this:

AcctName	FS Line	Amount
Checking account	Cash and Cash Equivalents	25,573
Money market	Cash and Cash Equivalents	39,441
Savings	Cash and Cash Equivalents	24,868
Prepaids	Prepaids and Deposits	47,562
Deposits	Prepaids and Deposits	13,890
Accounts Receivable	Accounts Receivable	11,655
Inventory	Inventory	23,984
Furniture and Fixtures	Fixed Assets	11,778
Computer Equipment	Fixed Assets	19,393
Machinery	Fixed Assets	24,014
Automobiles	Fixed Assets	24,155
Accounts Payable	Accounts Payable	19,426
Payroll liabilities	Accrued Expenses	42,663
Taxes	Accrued Expenses	32,191
Loan from officer	Loans	36,145
Retained earnings	Retained Earnings	45,501
Current year earnings	Net Income	90,387

But in practice, our data is often split into multiple tables. For example, a data table that stores the values and a lookup table that stores the account names.

Here is an example of our **Data** table:

AcctID	Amount
1000	25,573
1001	39,441
1002	24,868
1010	47,562
1011	13,890
1012	11,655
1020	23,984
1030	11,778
1032	19,393
1034	24,014
1036	24,155
2000	19,426
2001	42,663
2002	32,191
2010	36,145
3000	45,501
3001	90,387

And here is an example of our **Lookup** table:

AcctID	AcctName	FS Line
1000	Checking account	Cash and Cash Equivalents
1001	Money market	Cash and Cash Equivalents
1002	Savings	Cash and Cash Equivalents
1010	Prepaids	Prepaids and Deposits
1011	Deposits	Prepaids and Deposits
1012	Accounts Receivable	Accounts Receivable
1020	Inventory	Inventory
1030	Furniture and Fixtures	Fixed Assets
1032	Computer Equipment	Fixed Assets
1034	Machinery	Fixed Assets
1036	Automobiles	Fixed Assets
2000	Accounts Payable	Accounts Payable
2001	Payroll liabilities	Accrued Expenses
2002	Taxes	Accrued Expenses
2010	Loan from officer	Loans
3000	Retained earnings	Retained Earnings
3001	Current year earnings	Net Income

So, before we can prepare our report, we need to first combine these tables. Now, how would we tackle this with classic Excel features? Hmmm. Perhaps we could use VLOOKUP to retrieve the account names and FS Lines. And that would work just fine ... but ... when we write a formula, we end up "babysitting" it each month. In other words, if there are more rows than last month, we need to confirm that the formulas are filled down into the new rows. If the data source changes, we need to confirm the formulas reference the new range. So, yes, VLOOKUP would work here ... but ... let's try Power Query instead and see if it offers additional advantages.

Hint: it does. At first, we'll use Power Query to do what we could have done with VLOOKUP, but as the series continues, we'll discover how Power Query provides benefits far beyond what VLOOKUP can do.

We'll use the following three steps to combine our tables with Power Query:

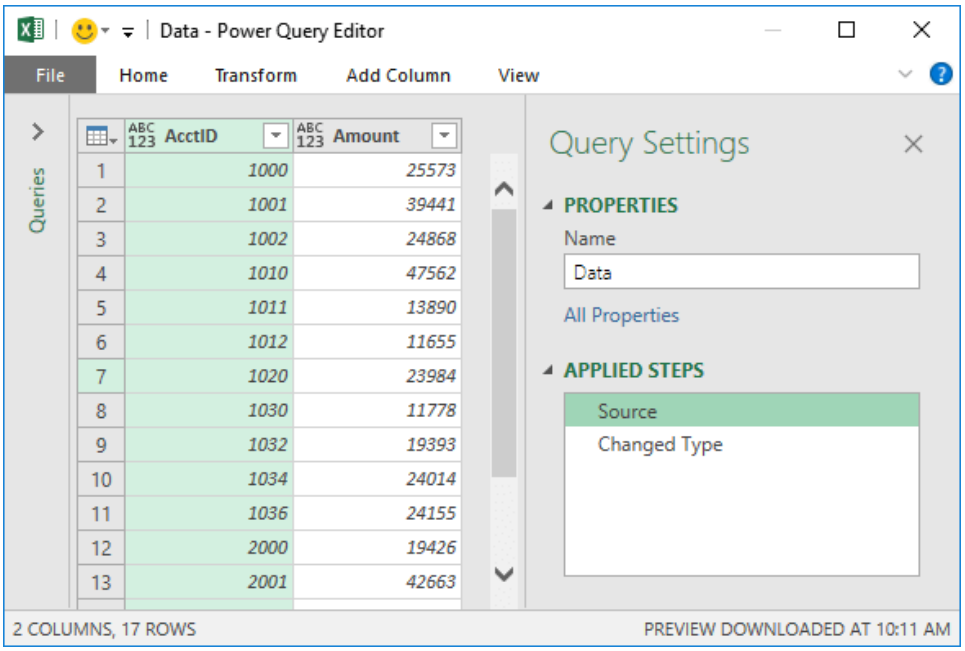
- Get data table
- Get lookup table
- Combine them

Let's take them one at a time.

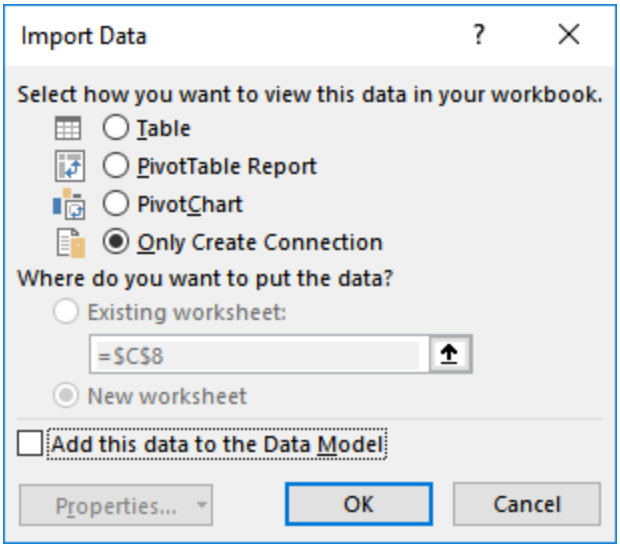
Note: depending on your version of Excel, you may or may not have Power Query or the same commands, screens, and options as my screenshots below which were created with Excel O365 for Windows.

Get data table

To get the data table into Power Query, we select any cell in the data table and click **Data > From Table/Range**. The data table is loaded into Power Query as shown below:



Now, we click **Home > Close & Load To ...** to display the dialog below:



We select **Only Create Connection** and click OK.

And we do the same thing for the next table.

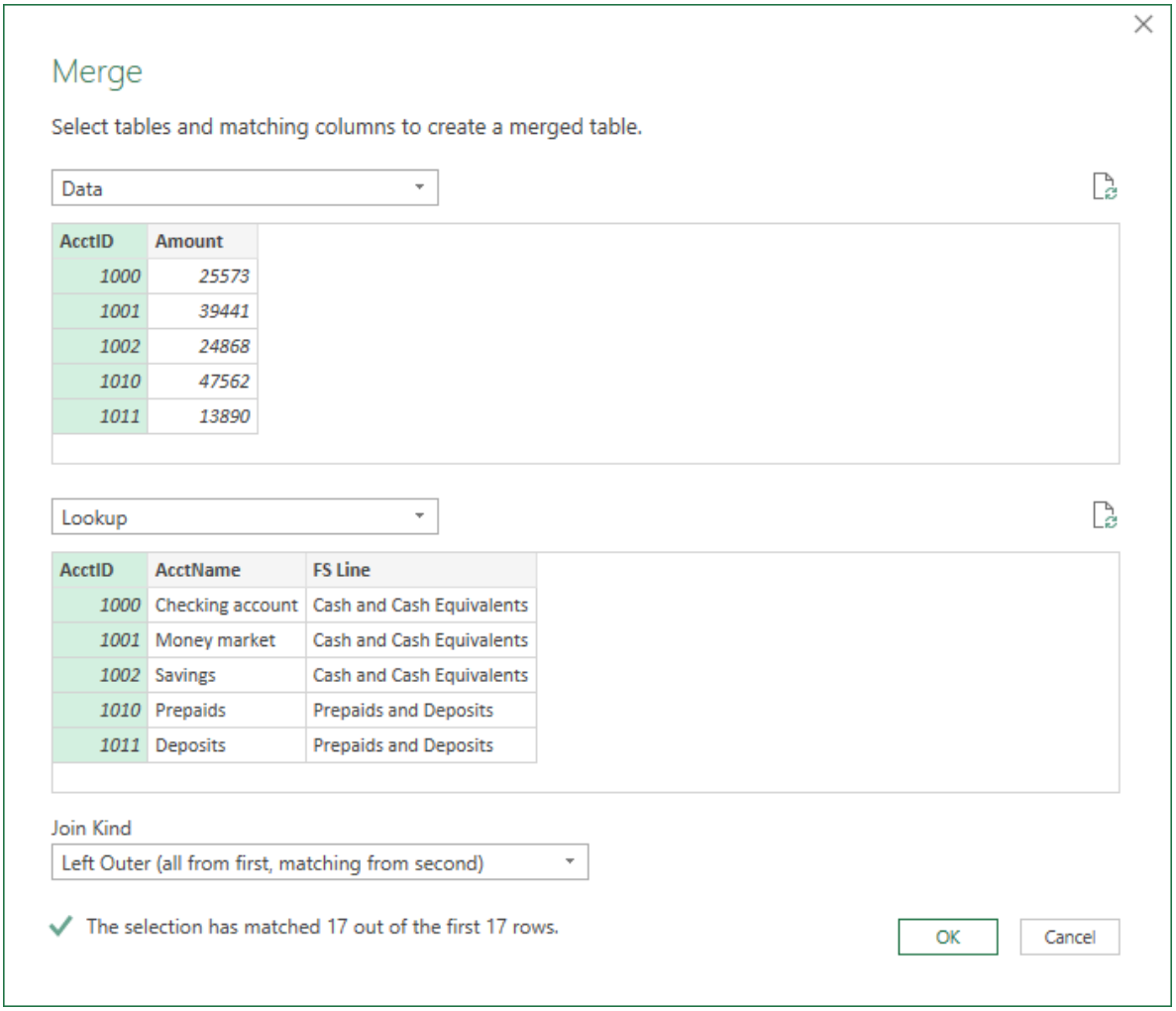
Get lookup table

We select any cell in the lookup table and click **Data > From Table/Range**. We **Close & Load To ...** and **Only Create Connection**.

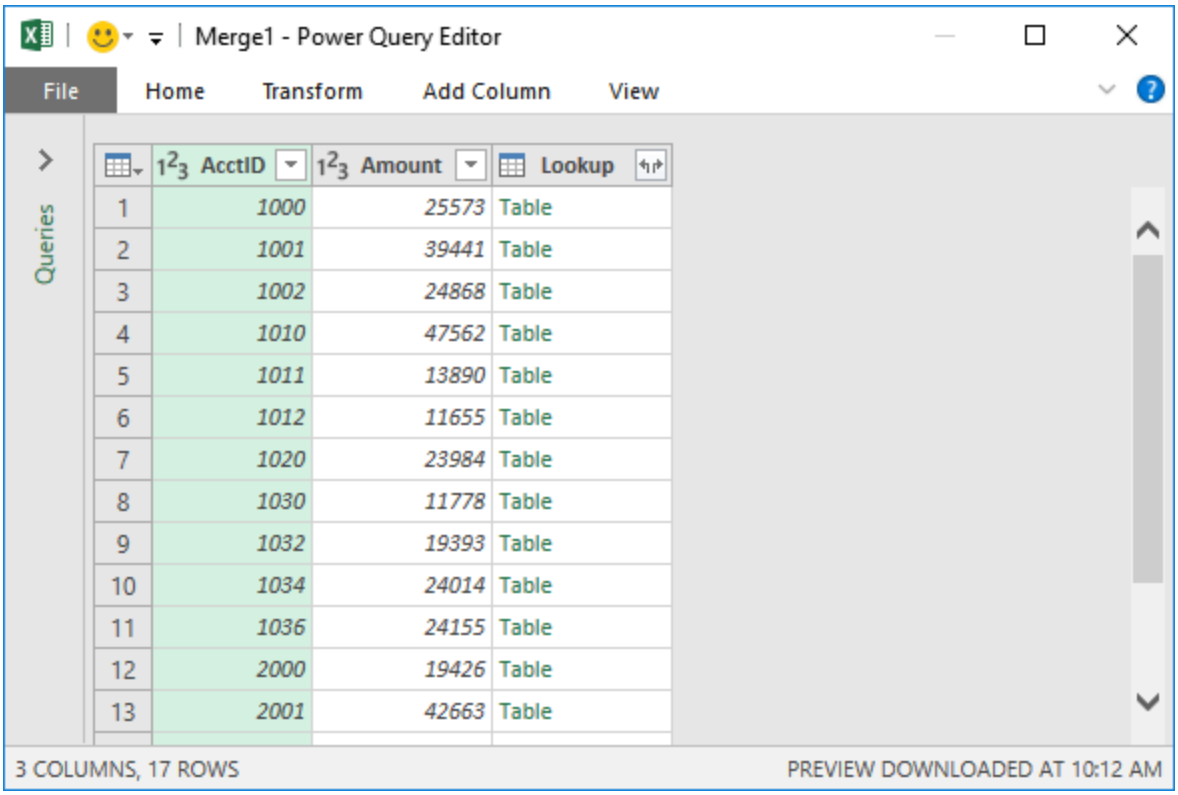
With our data and lookup tables inside Power Query, we can now combine them.

Combine them

We need to combine the data and lookup tables. In classic Excel, we'd probably use some type of lookup function, such as VLOOKUP, to do this. But here, we'll use Power Query. We click **Data > Get Data > Combine Queries > Merge**. In the resulting Merge dialog, we pick our **Data** table from the first drop-down and our **Lookup** table from the second drop-down. Then, we need to identify the shared, or lookup column, in both tables. To do this, we just use the mouse to select the **AcctID** columns from both tables, as shown below.

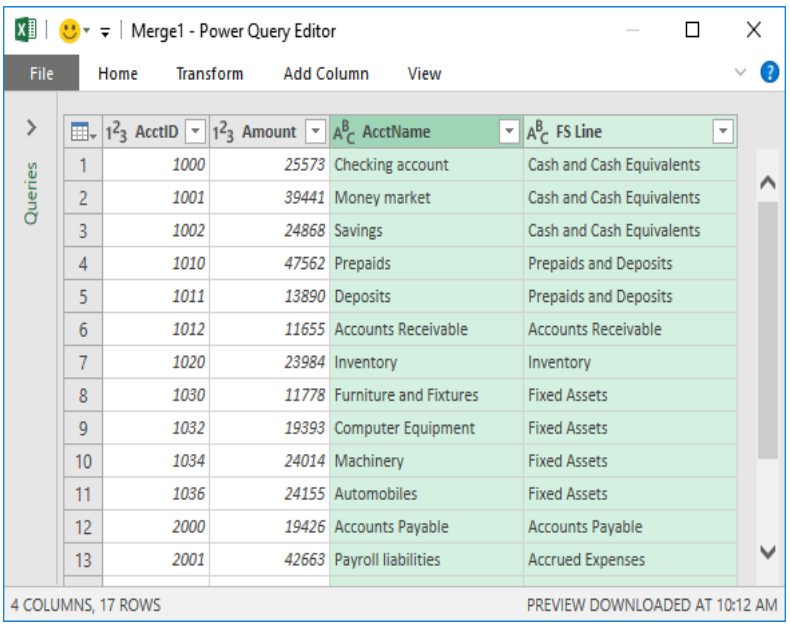


In this case, the default Join Kind (Left Outer) is perfect since we want all rows from the data table, and any matching rows from the lookup table. We click OK and the Power Query Editor window is opened:



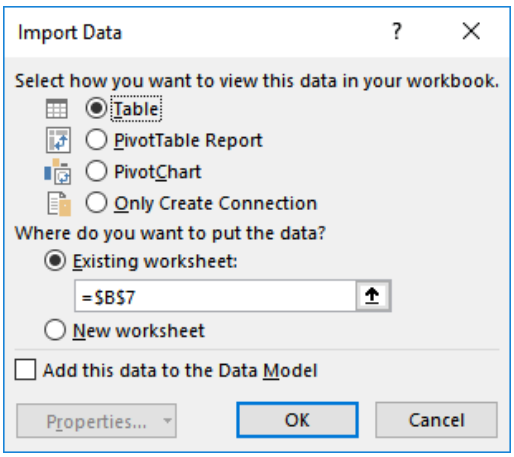
We click the Expand icon in the upper-right of the Lookup column label, and pick the columns from the Lookup table we want to see in our results. In this case, we want to see the AcctName and FS Line columns, as shown below.

We click OK ... and bam:



At this point, we are looking good. We've combined the data and lookup tables, and are ready to return the results to a worksheet.

We click **Home > Close & Load To ...** and decide to send the results to a **Table** in an **Existing worksheet**, as shown below:



We click OK and the results are sent into our Excel worksheet:

At this point, we can refresh this table by clicking the **Data > Refresh All** command. Power Query will import any changes, including new data and lookup rows, combine them, and update the results table ... without us needing to babysit formulas!

For now, we'll pull the values into our report by using SUMIFS as we did before.

Report	
ASSETS	**Amount**
Cash and Cash Equivalents	89,882
Prepaids and Deposits	61,452
Accounts Receivable	11,655
Inventory	23,984
Fixed Assets	79,340
TOTAL ASSETS	266,313
LIABILITIES AND EQUITY	
Accounts Payable	19,426
Accrued Expenses	74,854
Loans	36,145
TOTAL LIABILITIES	130,425
Retained Earnings	45,501
Net Income	90,387
TOTAL EQUITY	135,888
TOTAL LIABILITIES AND EQUITY	266,313

But, by using Power Query to combine these tables instead of VLOOKUP, we have additional options and benefits. And, we will take our next step towards automation in the next post in the series.

Recap

We started our journey with manual steps. As we learned more, we were able to move to the right ... towards our goal of automation.

We learned how tables address new data rows and the SUMIFS function addresses the sort order. In this post, we learned how Power Query can import and merge data and lookup tables without formulas. And, that going forward, we can just click the Refresh All command as table values change or new rows are added.

We'll take another step to the right in our next section. :-)

Additional Resources

- Sample file: StopWasting2.xlsx
- Prior posts about Power Query

3

In the beginning of our journey, we were updating the report manually. Now, as we've learned more, we've been able to automate parts of it. That is, we've been able to delegate stuff we were doing manually to Excel. The net effect is that we've been able to do the same work in less time. We've stopped spending more time than necessary ... we've stopped wasting time. But, we still have more improvements to make! So let's get to it.

Background

Just to refresh our memory, let's take a look at our report:

Report	
ASSETS	**Amount**
Cash and Cash Equivalents	89,882
Prepaids and Deposits	61,452
Accounts Receivable	11,655
Inventory	23,984
Fixed Assets	79,340
TOTAL ASSETS	266,313
LIABILITIES AND EQUITY	
Accounts Payable	19,426
Accrued Expenses	74,854
Loans	36,145
TOTAL LIABILITIES	130,425
Retained Earnings	45,501
Net Income	90,387
TOTAL EQUITY	135,888
TOTAL LIABILITIES AND EQUITY	266,313

In the first post, we started with formulas that basically used Excel as a digital 10-key:

ASSETS	**Amount**
Cash and Cash Equivalents	=25573+39441+24868

Since that approach forced us to rewrite formulas as the cell values changed, we upgraded the formulas to use cell references like this:

ASSETS	Amount
Cash and Cash Equivalents	=G11+G12+G13

But, since that approach broke if we sorted the data table and values moved around, we upgraded our workbook to use tables and SUMIFS, like this:

	B	C	D
8	ASSETS	Amount	
9	Cash and Cash Equivalents	=SUMIFS(Table1[Amount],Table1[FS Line],B9)	

In the second post, our data came in two separate tables, a data table and a lookup table. So, we learned how to use Power Query to join them (instead of VLOOKUP). Power Query produced the results table:

AcctID	Amount	AcctName	FS Line
1000	25,573	Checking account	Cash and Cash Equivalents
1001	39,441	Money market	Cash and Cash Equivalents
1002	24,868	Savings	Cash and Cash Equivalents
1010	47,562	Prepaids	Prepaids and Deposits
1011	13,890	Deposits	Prepaids and Deposits
1012	11,655	Accounts Receivable	Accounts Receivable
1020	23,984	Inventory	Inventory
1030	11,778	Furniture and Fixtures	Fixed Assets
1032	19,393	Computer Equipment	Fixed Assets
1034	24,014	Machinery	Fixed Assets
1036	24,155	Automobiles	Fixed Assets
2000	19,426	Accounts Payable	Accounts Payable
2001	42,663	Payroll liabilities	Accrued Expenses
2002	32,191	Taxes	Accrued Expenses
2010	36,145	Loan from officer	Loans
3000	45,501	Retained earnings	Retained Earnings
3001	90,387	Current year earnings	Net Income

Now, in this part, we'll take our next steps with Power Query.

Observation

The previous posts had an underlying assumption. We assumed the data was already in Excel. This is so subtle that Excel users often don't even consider it. We know that in order to use SUMIFS, or Tables, or VLOOKUP, the data has to be in Excel. But, how did it get there? Probably with a manual step ... like copy/paste.

We are so accustomed to this manual copy/paste step that we don't even think about it. We just do it. Each month. Again and again. But, here's the thing. This is a manual step that Power Query may be able to eliminate.

In this post, we'll make the following improvements to our workflow:

- Get data with Power Query
- Aggregate in Power Query

Let's get to it.

Get Data with Power Query

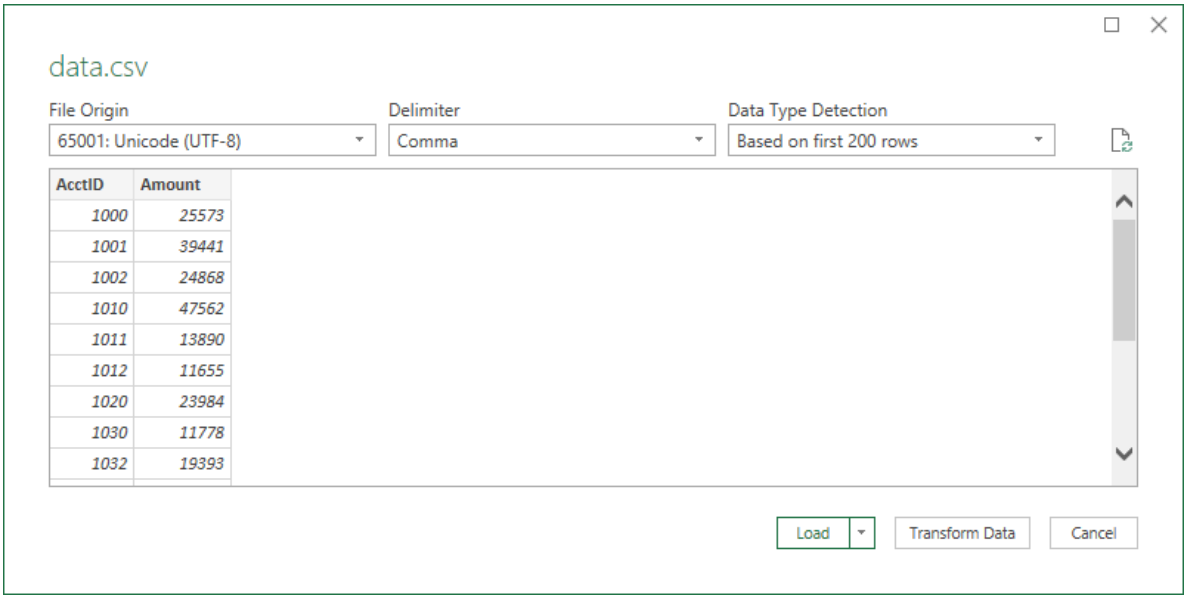
Most often, our data doesn't originate in Excel ... it is from another system or application. Of course, there are times when users manually type data into a workbook. But often, the transactions we analyze are exported from somewhere and copy/pasted into an Excel workbook. Pasting data into Excel is so common in fact, that Paste is the #1 most-used Excel command!

So, rather than copy/paste, we may be able to eliminate that step with Power Query. It depends on the location of the data, but Microsoft has done a great job of enabling Power Query to connect to all sorts of data sources, including files, databases, cloud services, and more.

In our illustration, we'll import our data from CSV files. We'll start with getting our data table.

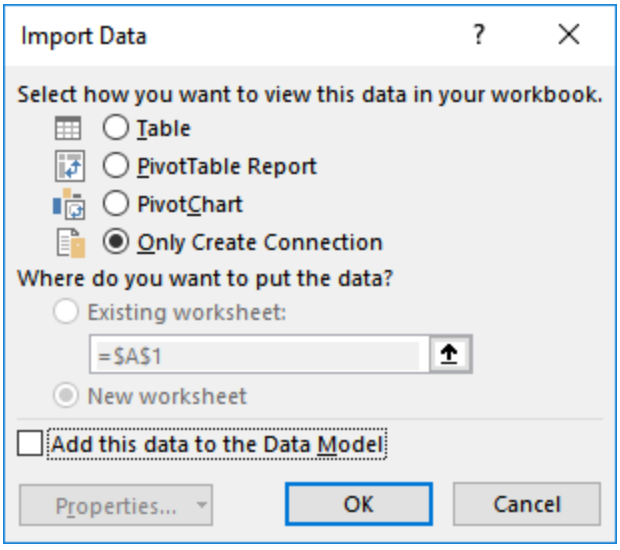
Get Data Table

In Excel, we click the **Data > From Text/CSV** command. In the resulting dialog box, we simply navigate to the csv file and click OK and receive the following preview dialog:



If we wanted to perform any transformations (edits), we could click the Transform Data button ... but since our data is clean we expand the Load button and select **Load To ...**

We select **Only Create Connection** as shown below:



Now that we have imported the data table, we basically perform the same steps to retrieve our lookup table.

Get Lookup Table

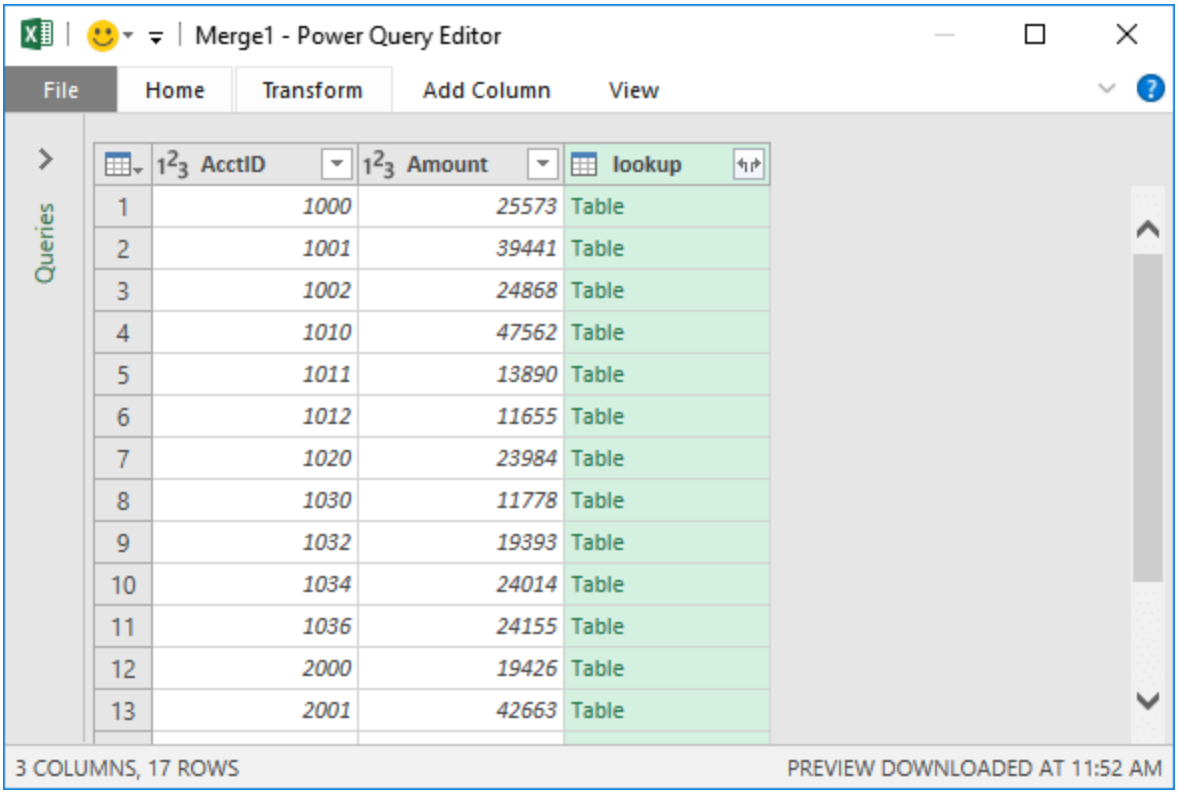
We select the **Data > From Text/CSV** command and browse to the csv file. Once again, we **Load To ...** and select **Only Create Connection**.

Now, we just need to combine the two tables.

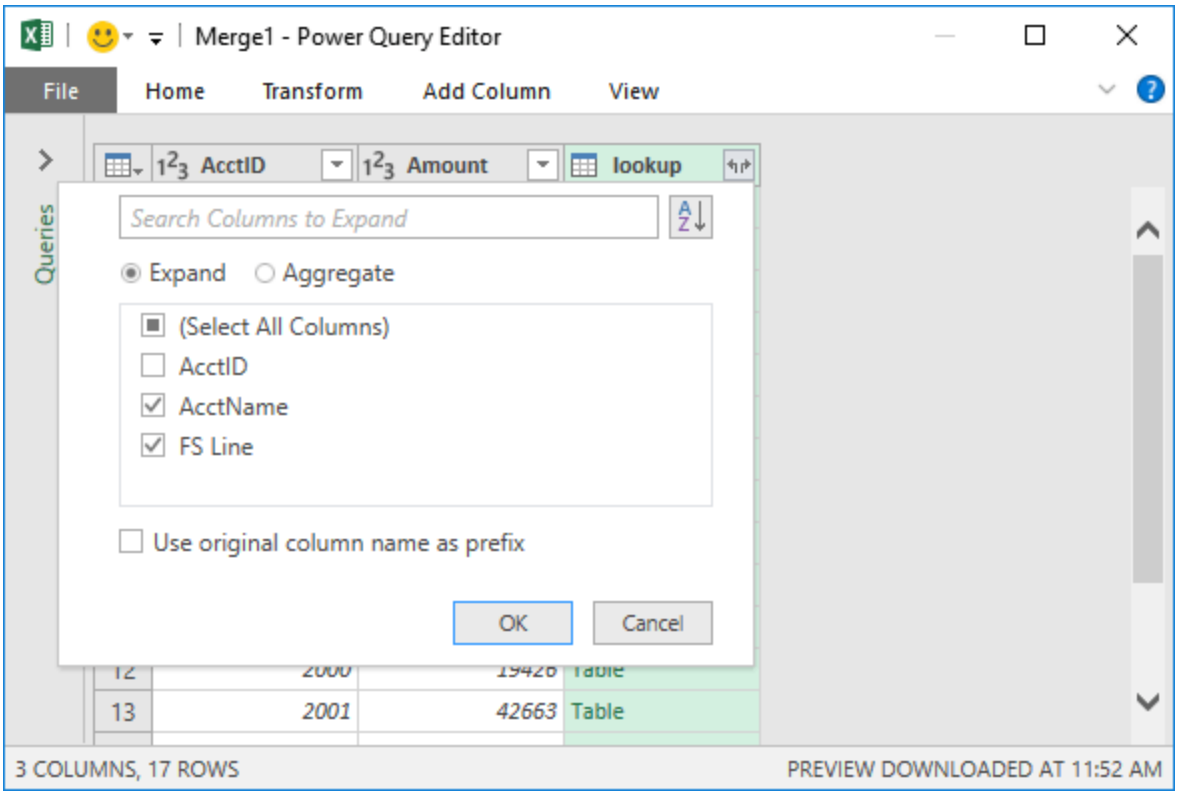
Combine Tables

We select **Data > Get Data > Combine Queries > Merge**. We identify the **data** table in the first drop-down, the **lookup** table in the second drop-down, and select the **AcctID** columns from both tables as shown below.

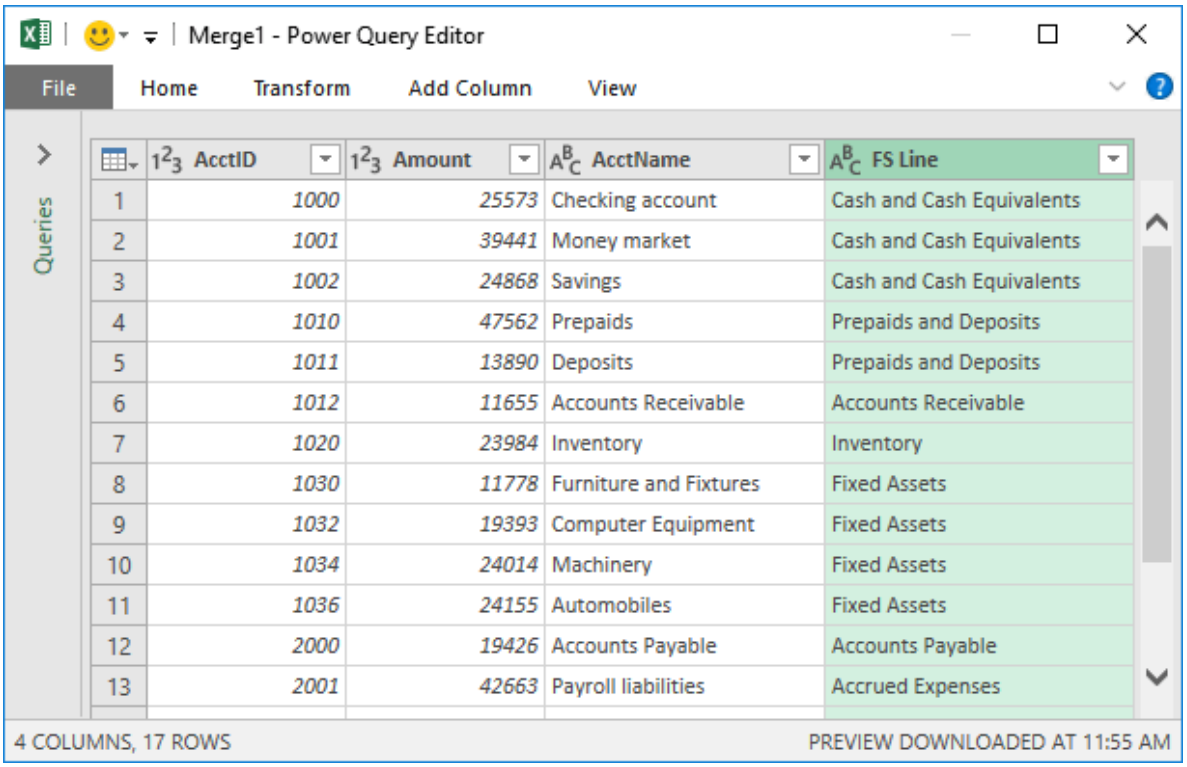
We click OK and the results are displayed in the Power Query editor as shown below.



We click the expand icon in the upper-right corner of the lookup column header and pick **AcctName** and **FS Line**, as shown below.



We click OK and bam:

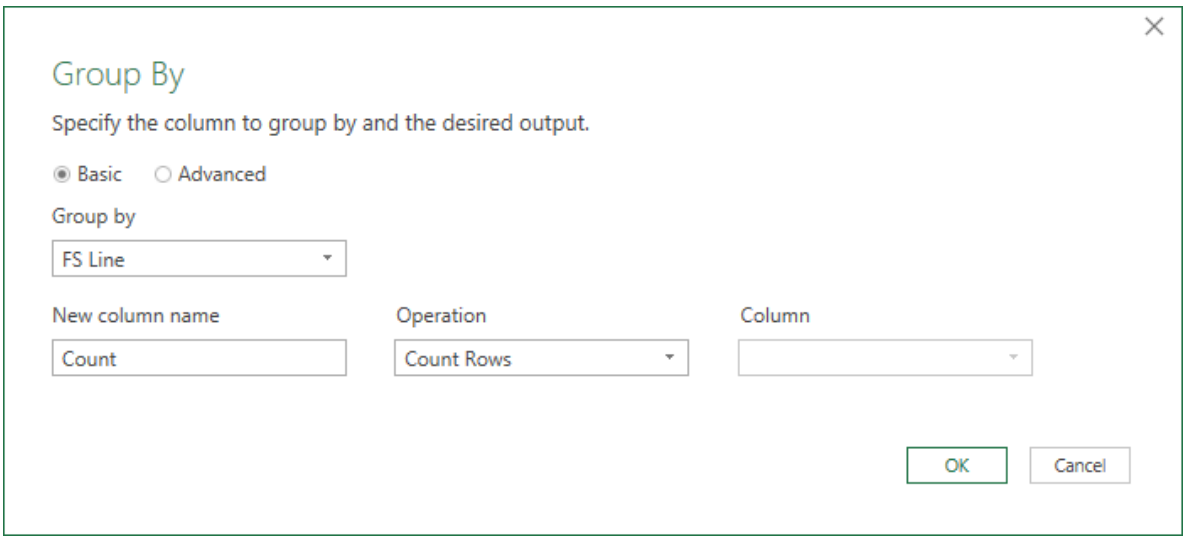


Now, we could send these results back to a worksheet like we did in the previous post. But, we can also do the aggregation inside of Power Query!

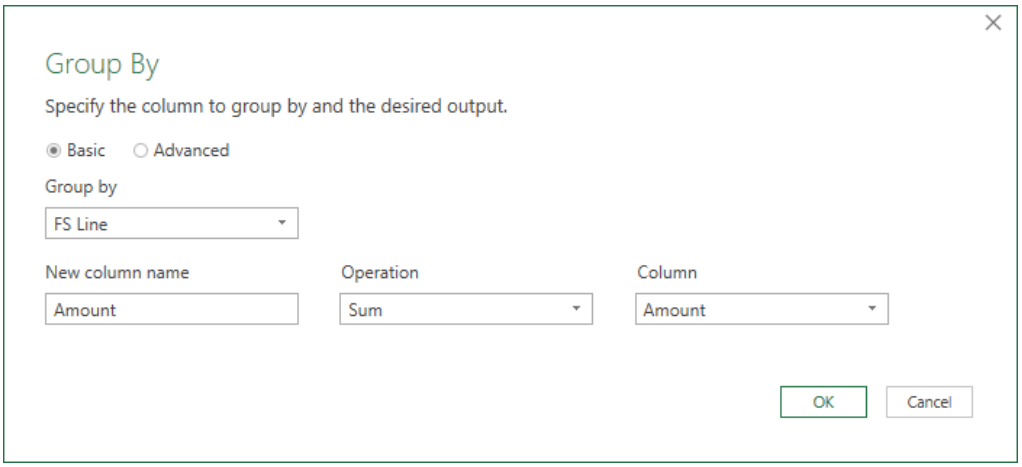
So, if you are keeping track, this means that Power Query can replace the copy/paste, the VLOOKUP to join the tables, and the SUMIFS to aggregate the values. Awesome ... let's do this thing!

Aggregate in Power Query

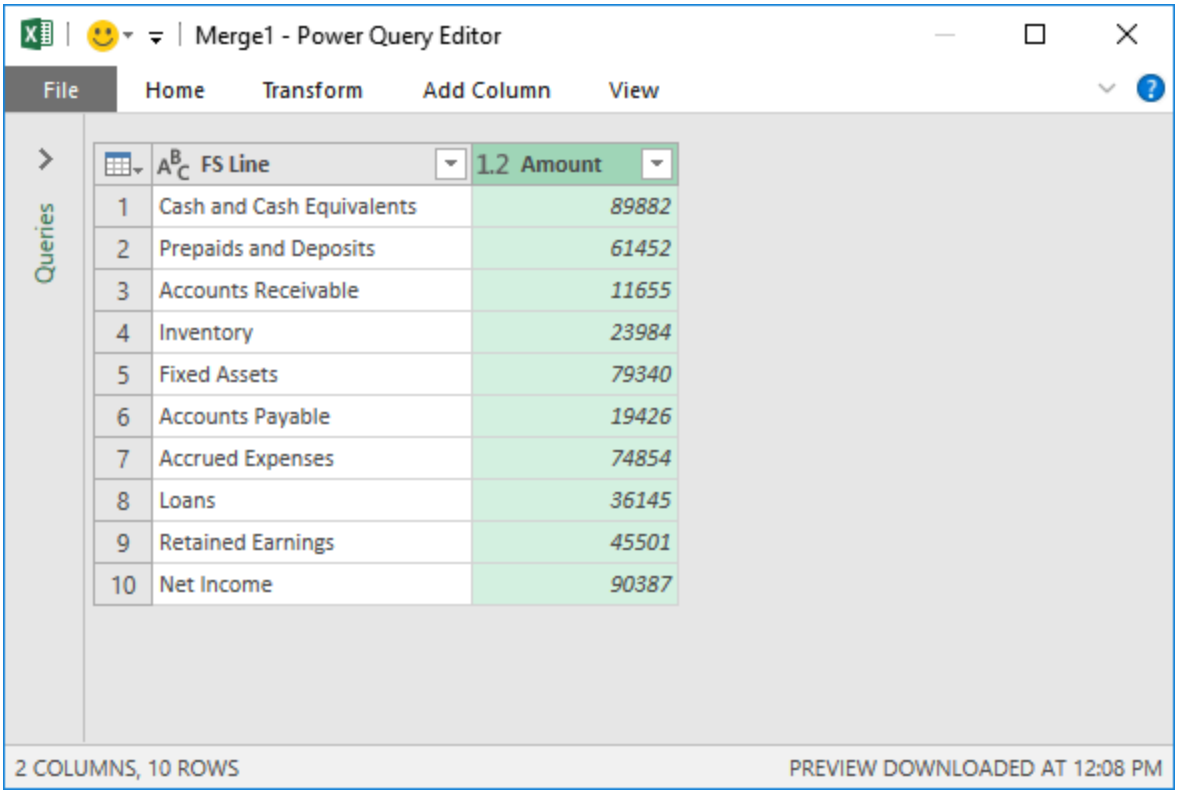
To aggregate the amount values by the FS Line column, we select the **FS Line** column (as shown in the screenshot above) and then click Power Query's **Transform > Group By** command. The resulting dialog is shown below.



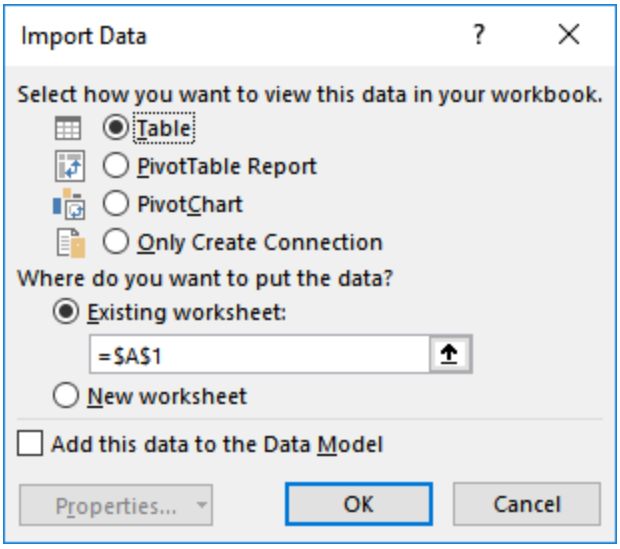
The Group by field should be **FS Line** so that it creates one row for each unique value in the FS Line column. Then, we want a new column called **Amount** that is basically the **Sum** of the **Amount** column. So we update the dialog as follows:



We click OK and bam:



We can **Close & Load To ...** a **Table** in an **Existing worksheet**:



We click OK and the results are loaded into Excel:

FS Line	Amount
Cash and Cash Equivalents	89882
Prepaids and Deposits	61452
Accounts Receivable	11655
Inventory	23984
Fixed Assets	79340
Accounts Payable	19426
Accrued Expenses	74854
Loans	36145
Retained Earnings	45501
Net Income	90387

At this point, we have the basic values that we need in our report. But, Power Query provides a table format ... not really a report format.

So, in the last section, we'll address this by using the data model :-)

Additional Resources

- Sample data file: data.csv
- Sample lookup file: lookup.csv
- Excel file you can use to import the data and lookup files: StopWasting3.xlsx
- Read previous Power Query posts

4

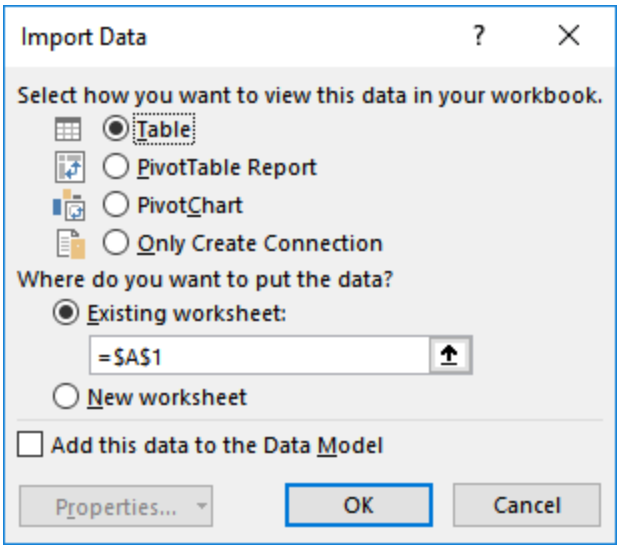
In this last part, we are going to use the data model to improve our reporting process. This is where everything we've learned comes together. By the time we have finished this section, updating our report will be extremely easy. In fact, it will only require a single click ... or, no clicks if you prefer :-)

Review

As a quick review, here is the report we are trying to automate:

Report	
ASSETS	**Amount**
Cash and Cash Equivalents	89,882
Prepaids and Deposits	61,452
Accounts Receivable	11,655
Inventory	23,984
Fixed Assets	79,340
TOTAL ASSETS	266,313
LIABILITIES AND EQUITY	
Accounts Payable	19,426
Accrued Expenses	74,854
Loans	36,145
TOTAL LIABILITIES	130,425
Retained Earnings	45,501
Net Income	90,387
TOTAL EQUITY	135,888
TOTAL LIABILITIES AND EQUITY	266,313

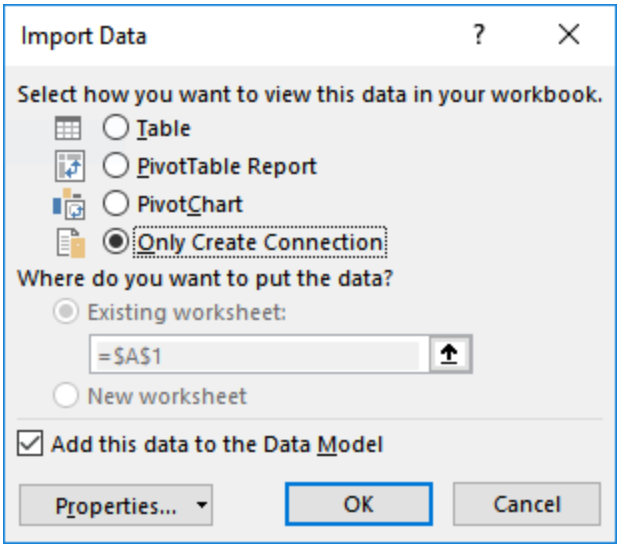
In the first part, we learned how to improve our report with tables and SUMIFS. In the second part, we learned how to use Power Query to combine tables. In the third part, we learned how to use Power Query to import the data and lookup csv files, combine them without VLOOKUP, and aggregate the values without SUMIFS. Then, we loaded the combined data to a **Table** in an **Existing worksheet** with the following Import Data options:



However, in this section, we'll load the data to the **Data Model** instead.

Load to Data Model

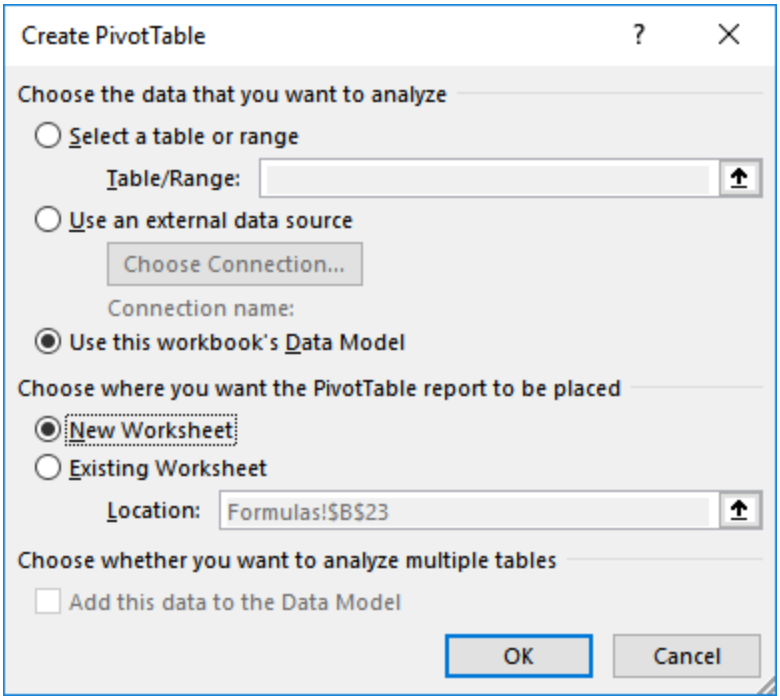
So, instead of using the options in the screenshot above, we **Only Create Connection** and load to the **Data Model** by using the options below:



At this point, the data is stored in the data model instead of being displayed in the worksheet. So, hmmm ... if we can't see the data in the worksheet, how do we access it? Well, one easy way is by building a PivotTable.

PivotTable

To build a PivotTable with data from the data model, we use the **Insert > PivotTable** command. In the resulting dialog, we opt to **Use this workbook's Data Model** as the data source:

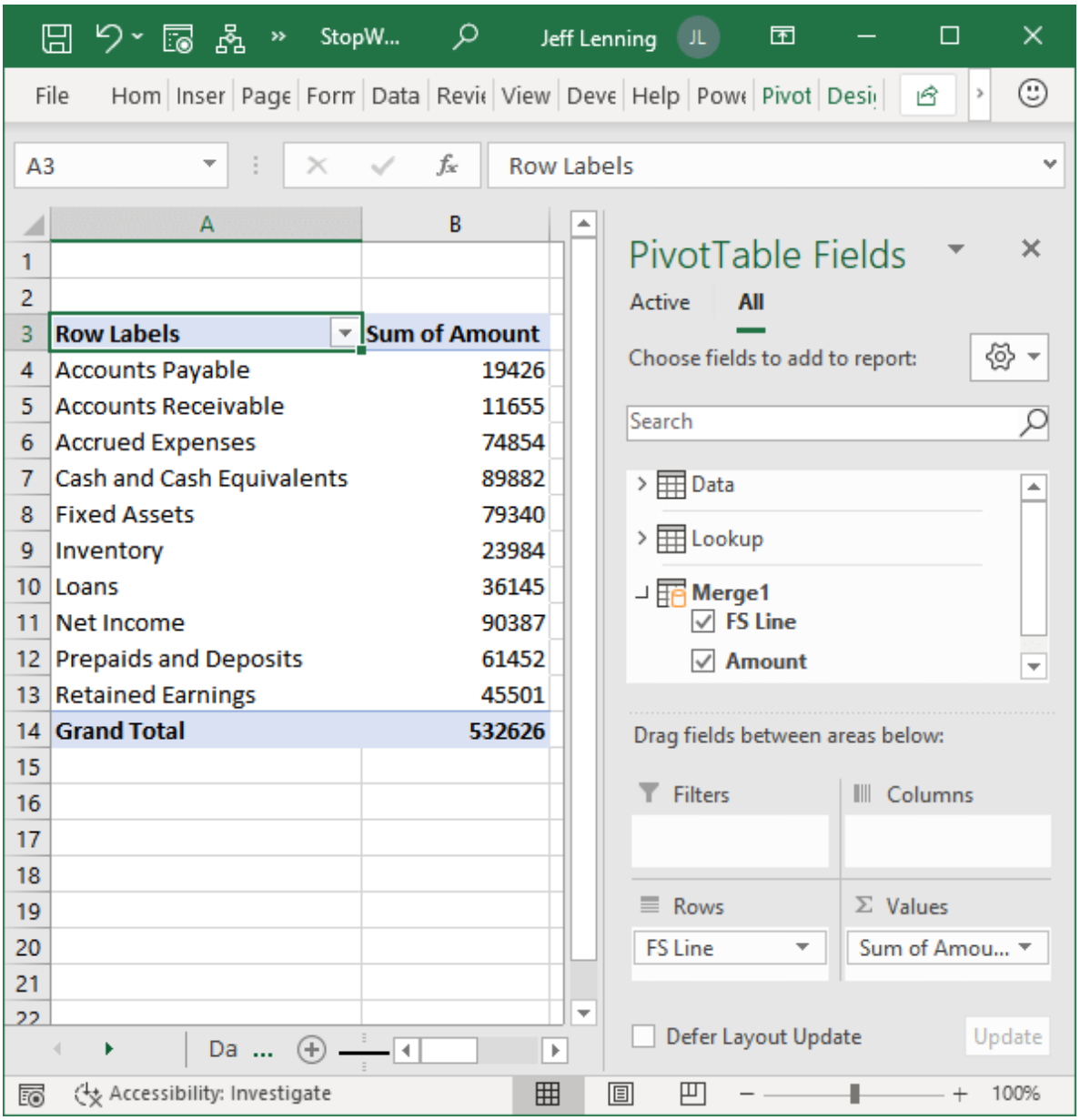


We click OK and now we just need to build the PivotTable.

We do so by inserting fields from the combined table (the one we merged in the previous post). We insert the **FS Line** field into the **Rows** area, and the **Amount** field into the **Values** area. We can click-and-drag or just check each checkbox.

Note: you'll see that the name of the field in the table is "Amount" but the name in the Values area is "Sum of Amount." When we inserted the Amount field into the report, Excel automatically created the "Sum of Amount" implicit measure (which we'll reference shortly).

The results are shown below:



And at this point, we are super close. The only issue that remains is the structure or format of the report. Our PivotTable above looks nothing like our target report.

So ... hmmm. It is like we need a PivotTable report to pull values from the data model ... but we need a formula-based report to get it into the exact format we need.

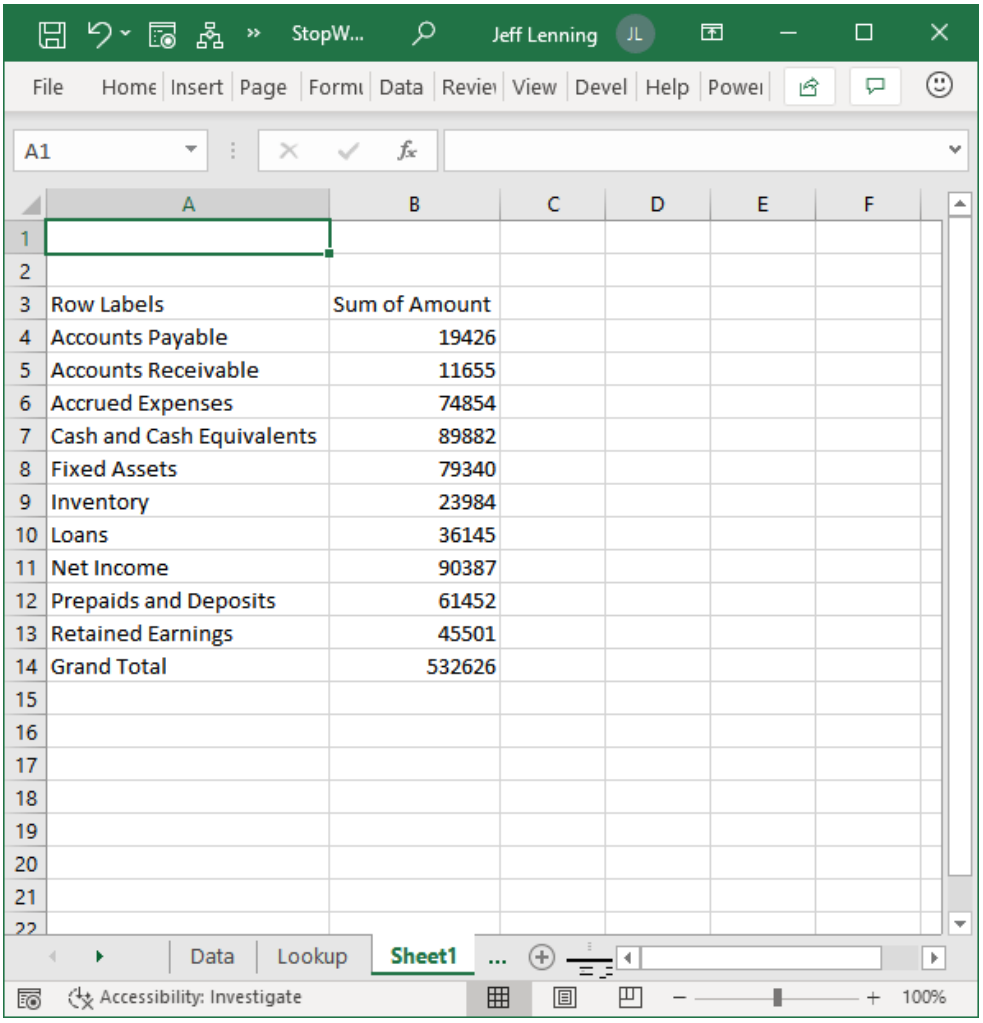
A PivotTable retrieves values from the data model. That is good. But, it doesn't provide the format that we need. But, formulas do ... because we can place formulas into any cells desired, insert rows anywhere we need, and apply whatever formatting we need.

Is there somehow a way to have the best of both worlds? To somehow use formulas to retrieve values from the data model? Or better yet … is there way to just convert the PivotTable into formulas? The answers are: yes, yes, and yes!

Formulas

My Excel-friend Rob Collie taught me this trick … thanks Rob!!!!!

Since we built our PivotTable with the Data Model, we have a command available to us that will convert the PivotTable to formulas. It is called, appropriately, Convert to Formulas. To use it, we select any cell in the PivotTable, and then select **PivotTable Tools > OLAP Tools > Convert to Formulas**. Bam:



Note: If the PivotTable uses an ordinary range or table as the data source, the Convert to Formulas command will be disabled.

The PivotTable is gone and in it's place are a bunch of formulas ... wow! These formulas use cube functions to retrieve values from the data model.

At this point, we are in good shape and have a couple of options. We could simply format this report as desired by inserting rows and cutting and pasting report lines into the correct order. And that would work just fine.

But, since we already have the basic report set up, let's just modify the formulas so we can paste them into our formatted report below:

Report	
ASSETS	**Amount**
Cash and Cash Equivalents	
Prepaids and Deposits	
Accounts Receivable	
Inventory	
Fixed Assets	
TOTAL ASSETS	0
LIABILITIES AND EQUITY	
Accounts Payable	
Accrued Expenses	
Loans	
TOTAL LIABILITIES	0
Retained Earnings	
Net Income	
TOTAL EQUITY	0
TOTAL LIABILITIES AND EQUITY	0

When we inspect the first amount formula, we see a CUBEVALUE function like this:

=CUBEVALUE("ThisWorkbookDataModel",$A4,B$3)

It is displayed in context below.

	A	B	C	D	E	F
3	Row Labels	Sum of Amount				
4	Accounts Payable	=CUBEVALUE("ThisWorkbookDataModel",$A4,B$3)				
5	Accounts Receivable	11655				
6	Accrued Expenses	74854				

The first argument "ThisWorkbookDataModel" tells the function to retrieve values from the data model. That makes sense. Then, we see a reference to the row label "Accounts Payable" in A4 and the implicit measure "Sum of Amount" in B3. We can tweak these arguments so that we can paste the formula into our basic report structure.

Let's start by replacing the reference to **B$3** with this: **"[Sum of Amount]"** ... and note the "quotes" and [square brackets] are both needed. The updated formula looks like this:

=CUBEVALUE("ThisWorkbookDataModel",$A4,"[Sum of Amount]")

Now, we just need to do the same type of thing for the row label, and enclose the label in quotes and square brackets. We can do this with the concatenation operator **&** as follows:

=CUBEVALUE("ThisWorkbookDataModel","["&$A4&"]","[Sum of Amount]")

Also, since we want to be able to paste the formula into any column, we'll remove the dollar sign $ in front of the A4 reference to make it relative, as shown below:

=CUBEVALUE("ThisWorkbookDataModel","["&A4&"]","[Sum of Amount]")

With these updates made, we can now copy that formula and paste it into our report structure.

Report	
ASSETS	**Amount**
Cash and Cash Equivalents	89,882
Prepaids and Deposits	
Accounts Receivable	
Inventory	
Fixed Assets	
TOTAL ASSETS	89,882

Yes ... it works! And we can paste it down into the remaining cells:

Report	
ASSETS	**Amount**
Cash and Cash Equivalents	89,882
Prepaids and Deposits	61,452
Accounts Receivable	11,655
Inventory	23,984
Fixed Assets	79,340
TOTAL ASSETS	266,313
LIABILITIES AND EQUITY	
Accounts Payable	19,426
Accrued Expenses	74,854
Loans	36,145
TOTAL LIABILITIES	130,425
Retained Earnings	45,501
Net Income	90,387
TOTAL EQUITY	135,888
TOTAL LIABILITIES AND EQUITY	266,313

And we got it!

The report looks the same as the one from the very first post. But the original version of this report was very manual to update each month. This one, which looks identical, is easier to update and maintain each month. We just need to click the **Data > Refresh All** command. That initiates the update sequence ... Power Query retrieves any updated data, merges the data and lookup tables, aggregates the values, sends it into the data model, and our formulas retrieve the updated values.

So, our monthly update process went from manual to automated. That is how you stop wasting time my friend!

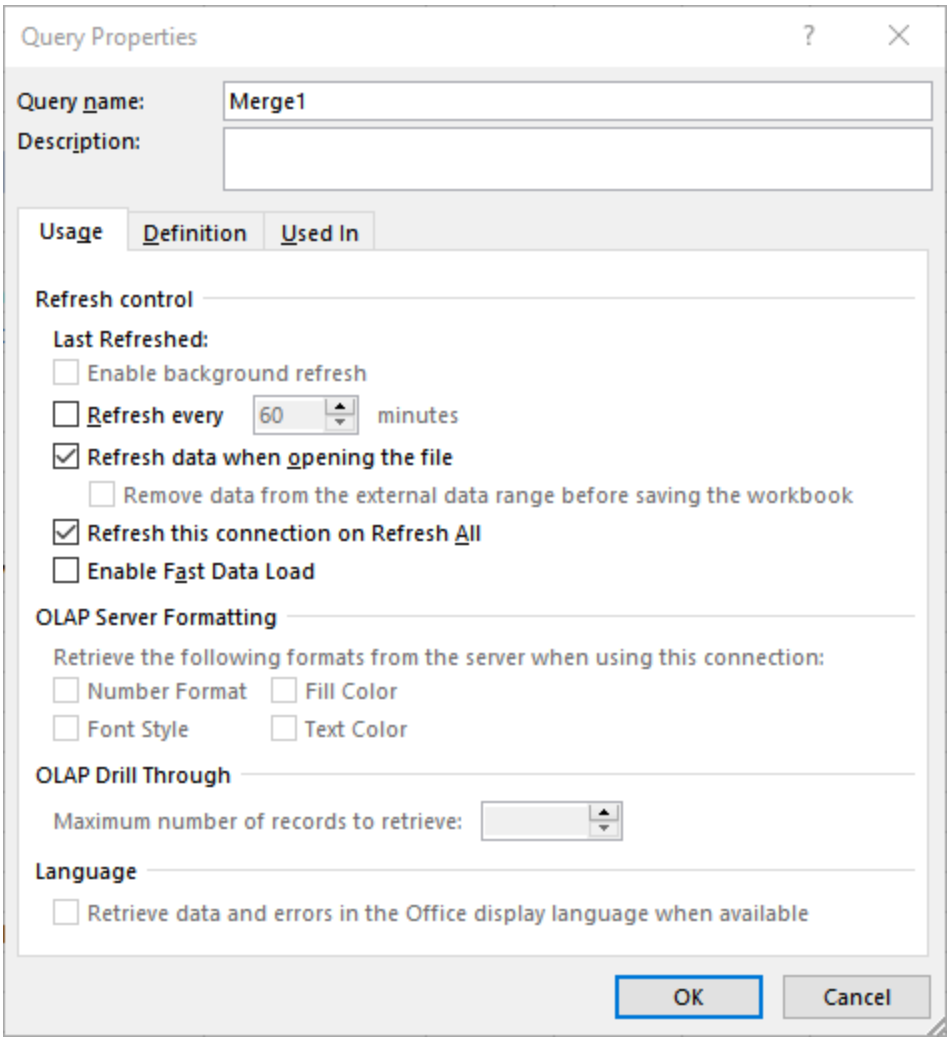
But, hang on Jeff, we still have a manual step. I mean, after you open the workbook, you still have to manually click the Refresh All button. Come on ... can't we fully automate this? Well, sure. If you don't want to have to click the Refresh All command, you can tell Power Query to refresh when you open the workbook.

Update on open

To have the merge query refresh automatically when you open the workbook, right-click the query name in the **Queries & Connections** pane and select **Properties**.

Note: you can toggle the display of the Queries & Connection panel by clicking Data > Queries & Connections.

In the resulting **Query Properties** dialog, check the **Refresh data when opening the file** as shown below.

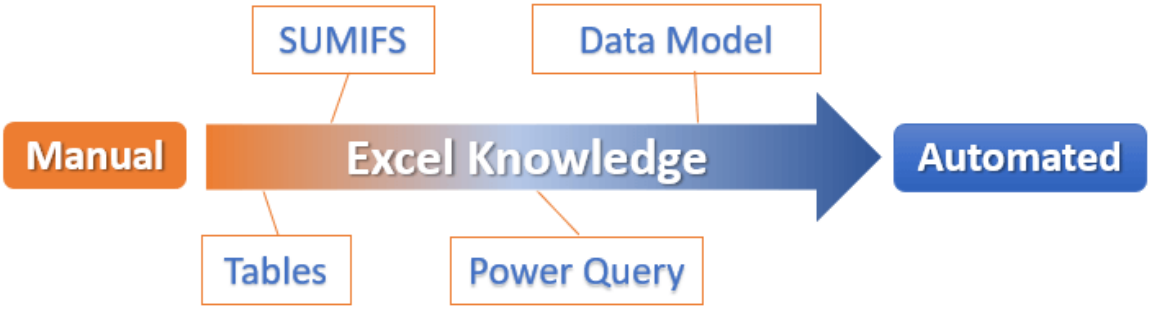


Now, when you open the workbook, the query will refresh automatically.

And that is how we take a report from manual to automatic :-)

Recap

At the end of the day, I hope that the things we covered in this series help you move to the right on the scale below:



We Excel users are all at different places on our Excel journey. Regardless of where you are now, if you'd like to move up on the scale, we offer on-demand training courses that can help.

We have two tracks: Undergraduate and Masters. The Undergraduate courses cover "classic" Excel features ... things like functions, formulas, PivotTables, conditional formatting, VLOOKUP, SUMIFS, INDEX/MATCH, and so on. The Masters courses cover more technical topics, including Power Query, Power Pivot, Power BI, and VBA/Macros.

You can jump in wherever you are at without having to start at the beginning. For example, if you are already comfortable with the classic Excel features, you can jump into the Masters track.

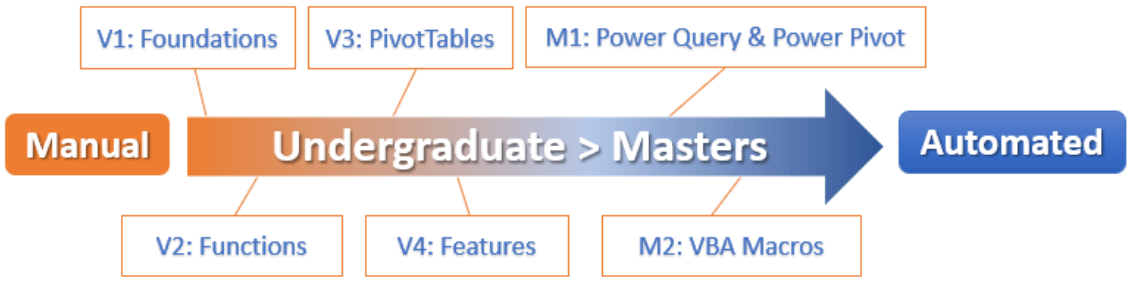
- Course 1: the foundations
- Course 2: formulas and functions
- Course 3: PivotTables
- Course 4: internal-use workpapers and related features

View the undergraduate courses, topics, and details: Undergraduate Courses

- Course 1: Power Query, Power Pivot, Power BI, table and graph design
- Course 2: VBA and Macros

View masters courses, topics, and details: Masters Courses

Here is a visual that helps you see the progression and how the courses fit together:



If we can provide any additional information about our training courses, please contact us ... we are happy to help!

And thanks for checking out the Stop Wasting Time e-book. I hope the things we discussed will help you get your work done faster!

Additional Resources

Sample file: StopWasting4.xlsx

Want to learn even more ways to save yourself your time with Excel? Check out the wide range of courses at Excel University today!

Enroll Today!

Excel University Copyright - 2019