Excel's Best New Features

With recent Excel updates provided through Microsoft 365/Office 365 subscriptions, Microsoft continues to pack new features into the popular spreadsheet tool. Recent Excel versions offer enhancements to data analysis tools, new functions, and improved collaboration opportunities. For those who know about these new features and apply them, exciting opportunities for improved productivity await. Take advantage of this opportunity to learn how to put Excel's best new features to work immediately!

Introduction

With recent Excel updates provided through Microsoft 365/Office 365 subscriptions, Microsoft continues to pack new features into the popular spreadsheet tool. Recent Excel versions offer enhancements to data analysis tools, new functions, and improved collaboration opportunities. For those who know about these new features and apply them, exciting opportunities for improved productivity await. Take advantage of this opportunity to learn how to put Excel's best new features to work immediately!

Learning Objectives

Upon completing this session, you should be able to:

- Apply primary new functions available in Excel, including STOCKHISTORY, SORT, and FILTER;
- Employ Excel's co-authoring feature to collaborate in real-time with other users;
- List at least three new functions that capitalize on Excel's Dynamic Arrays feature; and
- Utilize Excel's XLOOKUP and XMATCH functions to locate data in spreadsheets.

"New" is a Relative Term

Before detailing specific new Excel features and functions, let us quickly describe what "new" means in the context of this session. This conversation is necessary because "new" is a relative term, depending on your Excel version and how often you receive updates.

Perpetual Licenses

If you use a perpetual license of Excel, the "new" concept is easy to understand. You receive new features whenever you upgrade to the next version of Excel. Thus, if you recently upgraded to Excel 2021, all the new features in that release are indeed "new" to you. However, you will not receive additional new features until you upgrade to the next version of Excel.

Subscription Licenses

On the other hand, if you run a version of Excel provided through a Microsoft 365 or Office 365 subscription plan, you receive periodic updates to your instance of Excel (and other Office applications). These updates occur at different intervals, depending on what ***channel*** you or your IT staff elect. Three primary channels are available in this environment: 1) **Current Channel**, 2)

Monthly Enterprise Channel, and 3) **Semi-Annual Enterprise Channel**. As detailed in **Table 1**, feature updates occur unscheduled, monthly, or semi-annually.[1]

COMPARING THE PRIMARY UPDATE CHANNELS FOR MICROSOFT 365 APPS			
	Current Channel	**Monthly Enterprise Channel**	**Semi-Annual Enterprise Channel**
Recommended use	Provide your users with new Office features as soon as they are ready but on no set schedule.	Provide your users with new Office features only once a month and on a predictable schedule.	For select devices in your organization, where extensive testing is needed before rolling out new Office features. For example, to comply with regulatory, governmental, or other organizational requirements.
Release frequency	At least once a month (likely more often), but on no set schedule	Once a month, on the second Tuesday of the month	Once a month, on the second Tuesday of the month
Feature updates	As soon as they're ready (usually once a month), but on no set schedule	Once a month, on the second Tuesday of the month	Twice a year (in January and July), on the second Tuesday of the month
Security updates (if needed)	Once a month, on the second Tuesday of the month	Once a month, on the second Tuesday of the month	Once a month, on the second Tuesday of the month
Non-security updates (if needed)	Usually, at least once a month (possibly more often), but no set schedule	Once a month, on the second Tuesday of the month	Once a month, on the second Tuesday of the month
Support duration for a given version	Until the next version is released with new features, which is usually about one month	Two months	Fourteen months

Table 1 - Comparing Microsoft 365 Apps Update Channels

Of note is the option to enroll in the Current Channel (Preview). This channel provides users an "early look" at new and updated features that will soon debut in the Current Channel. So, four primary update channels are effectively available for Microsoft 365 Apps. If team members within an organization are enrolled in different channels, you potentially have four different sets of features and functions in use within the organization simultaneously. Thus, what a user in the Semi-Annual Enterprise channel considers a "new" feature might be old-hat for someone in the Current Channel.

[1] See the full discussion at https://docs.microsoft.com/en-us/deployoffice/overview-update-channels.

You can see which channel you are in by clicking **File**, **Account**, and **Options** to display the window pictured in **Figure 1**.

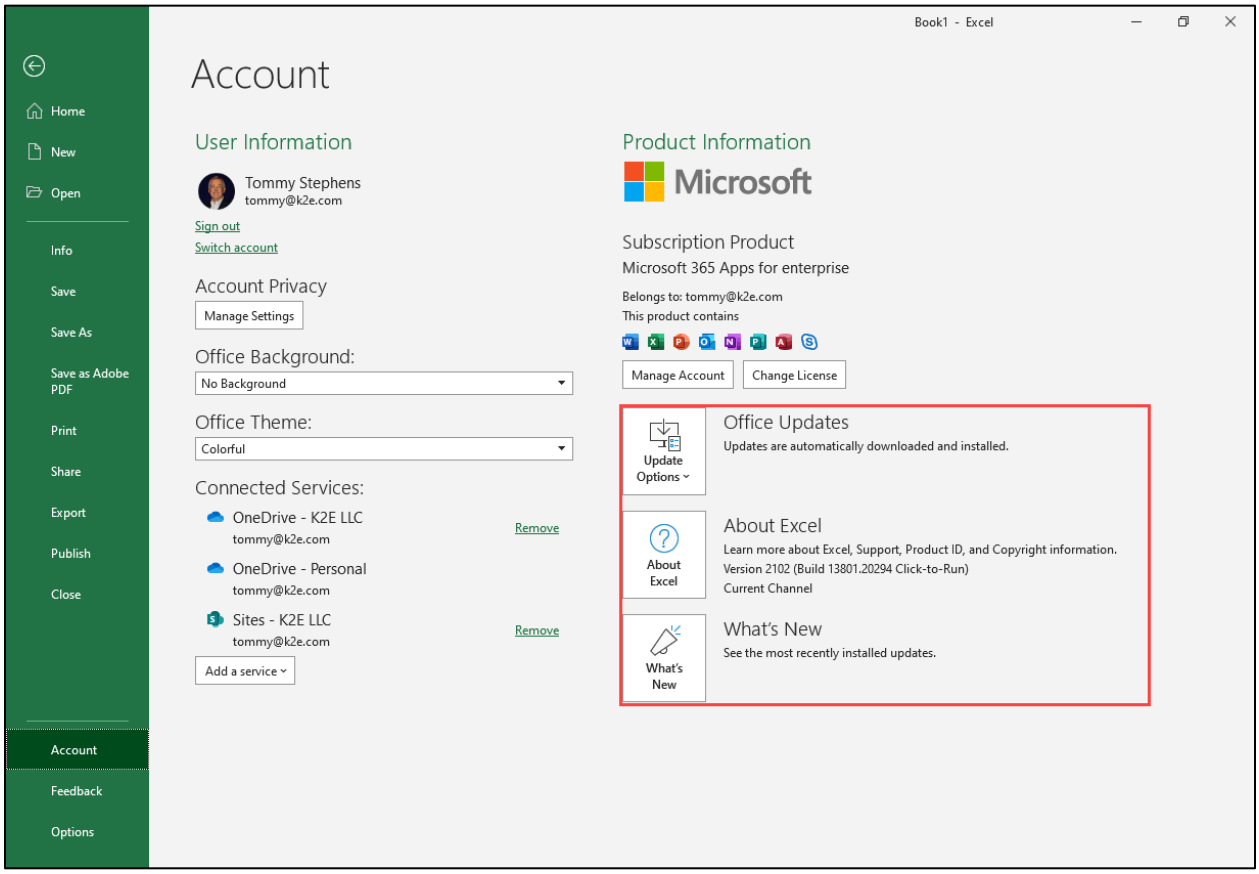


Figure 1 - Identifying Channel and Version

Also identified in Figure 1 is the *version number*; this, too, is necessary to understand new features in the subscription environment. The first two digits of the version number represent the last two digits of the year Microsoft published it. Likewise, the second two digits of the version number represent the month when Microsoft issued the version. Thus, "2502" translates into the February 2025 version.

Knowing the channel and the version number allows you to identify which features became available in that release. To do so, visit the following website:

https://docs.microsoft.com/en-us/officeupdates/update-history-microsoft365-apps-by-date

Next, select your channel from the window's left side, followed by **Release Notes**. Then scroll through the releases until you locate your version number. Upon doing so, as shown in **Figure 2**, you can see what new features became available in that version of Excel and other apps.

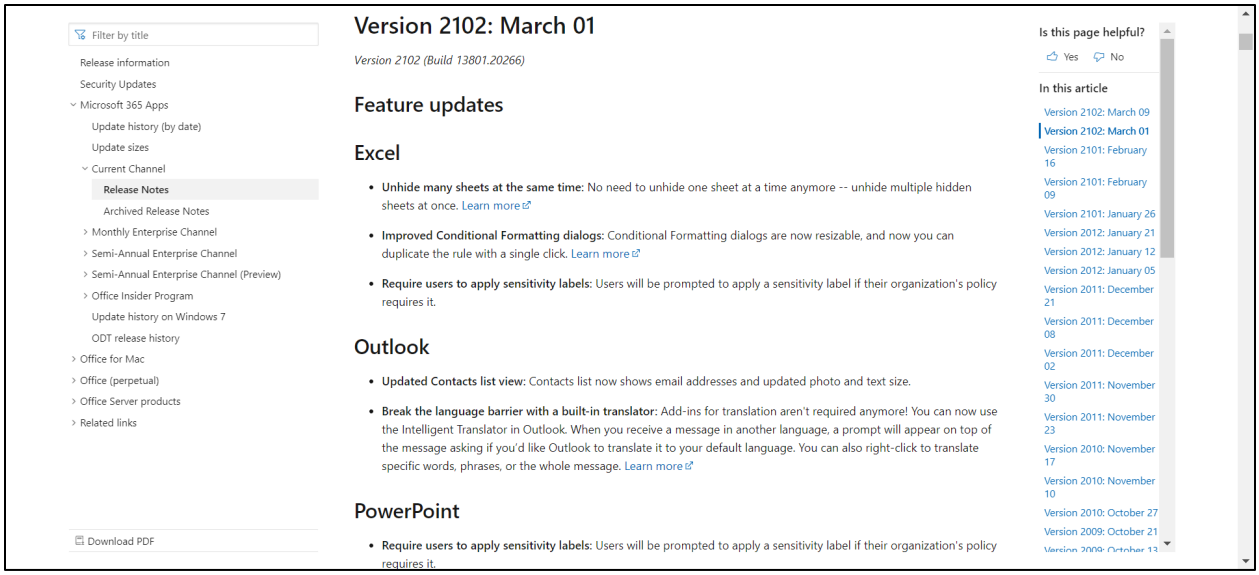


Figure 2 - Identifying New Features in Available in Microsoft 365 Apps

Essential New Functions and Features in Excel

STOCKHISTORY

Excel's long-awaited **STOCKHISTORY** function is now available in subscription-based versions of Excel. Unfortunately, this function is not available in Excel 2016 and Excel 2019.

With this feature, you can retrieve historical stock prices by entering a few variables into a formula. Moreover, you can retrieve values for a single date or a range of dates. Further, if you choose a range of dates, you can designate *daily*, *weekly*, or *monthly* intervals. STOCKHISTORY displays the date and closing price by default. However, if desired, you can choose to show the opening price, high price, low price, and volume.

Using STOCKHISTORY

The syntax for using the STOCKHISTORY function can be relatively simple, as indicated below. However, note that only the stock and start_date are required of the arguments available. Thus, a formula using STOCKHISTORY could be as simple as **=STOCKHISTORY("MSFT","1/29/2021")**. Of course, this formula returns the closing price for a share of Microsoft stock on January 29, 2021.

Additionally, you can create more sophisticated formulas using STOCKHISTORY if your needs require additional information. Specifically, the full syntax of a formula can include all the following items.

STOCKHISTORY(stock, start_date, [end_date],[interval],[headers], [property0], [property1] [property2], [property3], [property4], [property5])

- **stock**: The identifier for the financial instrument targeted. This reference can be a ticker symbol or a Stocks data type.
- **start_date**: The earliest date for which you want information.
- **end_date** (optional): The latest date for which you want information.
- **interval** (optional): Daily (0), Weekly (1), or Monthly (2) interval options for data
- **headers** (optional): Specifies if the formula returns additional header rows with the array.
- **property0 – property5** (optional): Specifies which information to include in the result, Date (0), Close (1), Open (2), High (3), Low (4), Volume (5).

Extending the previous example, we can create more powerful formulas that use the function. For instance, we can use the following formula to generate a listing of closing prices for a range of dates:

=STOCKHISTORY("MSFT","1/1/2020","12/31/2020")

To illustrate, **Figure 3** below provides an abbreviated set of results from the abovementioned formula.

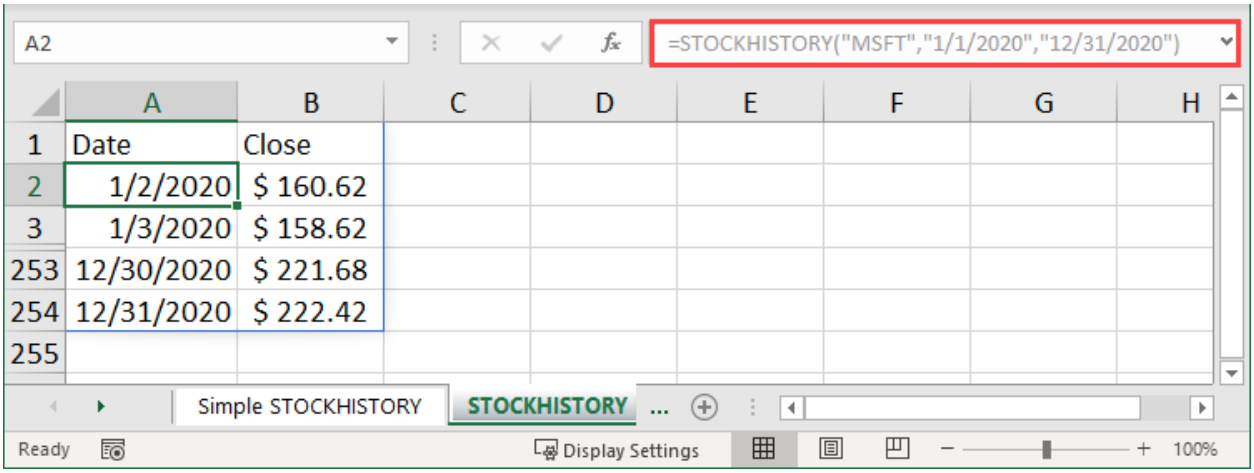


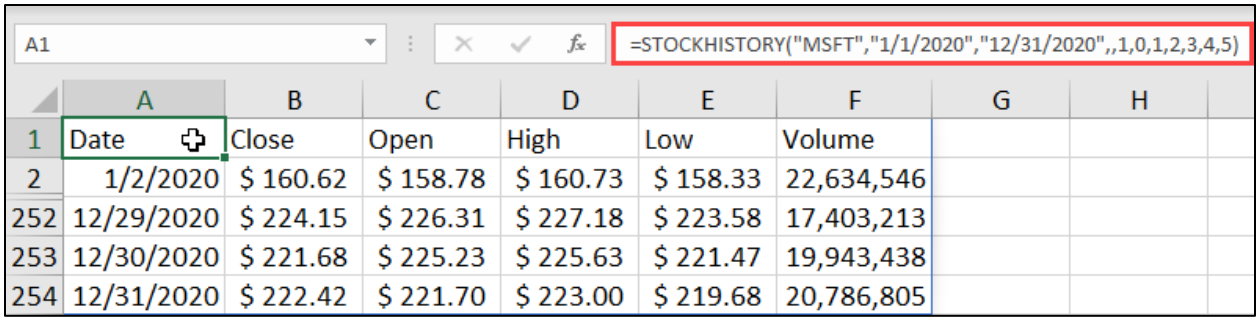
Figure 3 - Sample Results Using STOCKHISTORY

A Deeper Dive into STOCKHISTORY

The following example of the new function incorporates optional arguments to add columns for each trading interval for the Open price, High price, Low price, and Volume.

=STOCKHISTORY("MSFT", "1/1/2020", "12/31/2020",,1,0,1,2,3,4,5)

For example, **Figure 4** illustrates an abbreviated set of results using this formula.



	A	B	C	D	E	F	G	H
1	Date	Close	Open	High	Low	Volume		
2	1/2/2020	$ 160.62	$ 158.78	$ 160.73	$ 158.33	22,634,546		
252	12/29/2020	$ 224.15	$ 226.31	$ 227.18	$ 223.58	17,403,213		
253	12/30/2020	$ 221.68	$ 225.23	$ 225.63	$ 221.47	19,943,438		
254	12/31/2020	$ 222.42	$ 221.70	$ 223.00	$ 219.68	20,786,805		

Figure 4 - STOCKHISTORY Example with Optional Arguments

Of course, you can use STOCKHISTORY results in the same fashion as if you entered the data manually.

Summarizing STOCKHISTORY

In short, STOCKHISTORY is one of the most widely-anticipated functions added to Excel in recent years. If you have a subscription-based version of Excel, you should already have access to this feature or receive access soon. Importantly, you can use STOCKHISTORY to retrieve stocks' historical prices and incorporate them into other calculations in your spreadsheets. Therefore, the next time you need to perform research to obtain historical data about a stock, consider using STOCKHISTORY. Most importantly, if you do, you will reduce the time you spend retrieving data.

Create PivotTables from Datasets in Power BI

(Available in the Current Channel of a subscription license beginning with Version 2011)

Another new feature currently only available in subscription-based versions of Excel is the opportunity to create PivotTables in Excel from datasets created in Power BI. With more organizations using Power BI as part of their reporting and business intelligence efforts, many Power BI datasets probably contain data that Excel users would like to summarize using PivotTables. That is now possible with this enhancement to Excel.

First, ensure that you sign in to your organizational Power BI account to utilize this feature. Once you sign in, open a blank workbook in Excel. Then, from the Ribbon's **Insert** tab, click the drop-down arrow under the **PivotTable** icon and choose **PivotTable from Power BI**, as shown in **Figure 5**. This action connects the Power BI-based data to Excel, and once connected, you can generate your PivotTable as if the data were "native" to Excel.

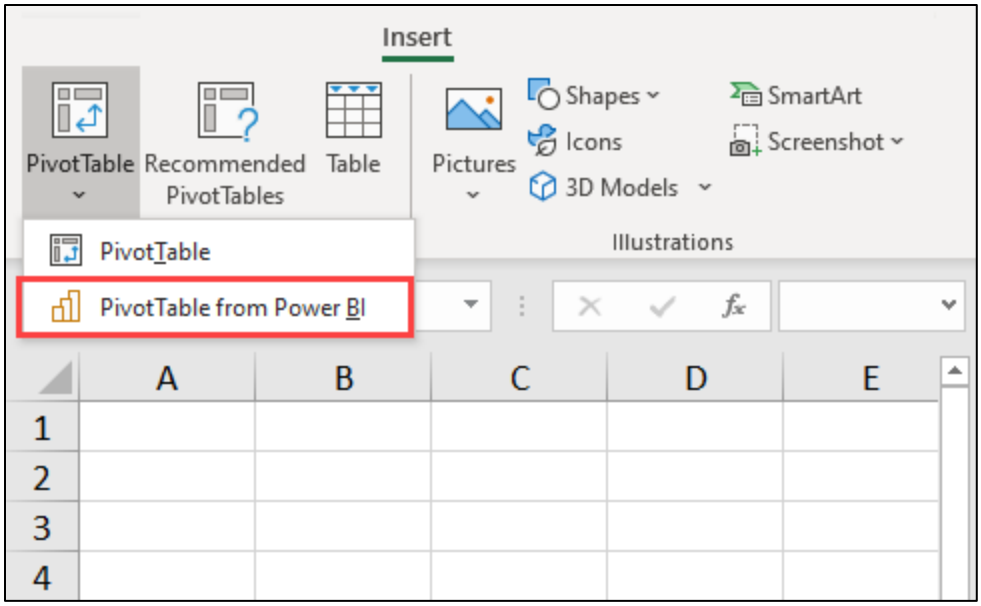


Figure 5 - Creating a PivotTable from a Power BI Dataset

Six New Functions Added to Excel 2019

Added to subscription-based versions of Excel beginning with the 1601 release and now available to those running Excel 2019 are six new functions that many users will find helpful.

1. TEXTJOIN
2. CONCAT
3. IFS
4. SWITCH
5. MAXIFS
6. MINIFS

We discuss and illustrate each of these below.

TEXTJOIN

You can use Excel's new **TEXTJOIN** function to concatenate a list or range of text, which places a delimiting character, such as a space or comma, between each field it joins. **Figure 6** provides a simple illustration of where TEXTJOIN is used to concatenate five data columns into one. Notably, because the formula uses relative referencing, you can copy it to join data from other rows.

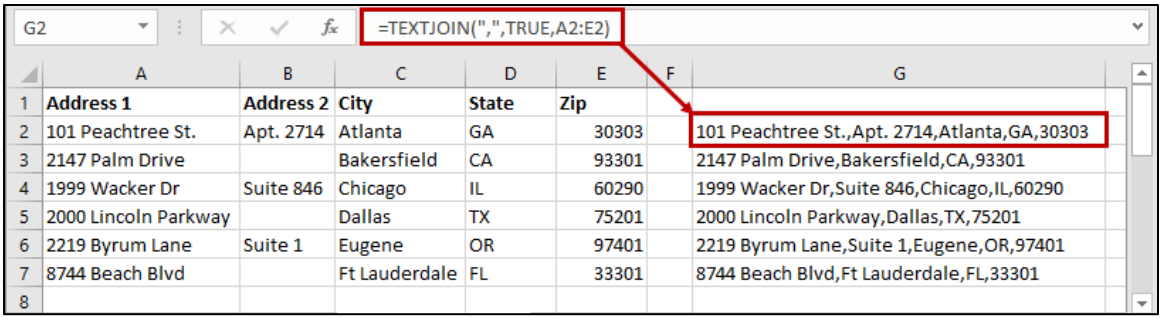


Figure 6 - Using TEXTJOIN to Concatenate Three Columns of Data

The syntax of a formula that uses the TEXTJOIN function appears below.

=TEXTJOIN(delimiter, ignore_empty, text1, [text2], …)

Of note, you must specify the delimiting character(s); this can be a cell reference to a valid text string. You also can instruct TEXTJOIN on how to handle empty cells. If the formula's **ignore_empty** argument is blank or **TRUE** (as shown in Figure 6), TEXTJOIN ignores empty cells. This treatment can be beneficial when some addresses contain apartment or suite numbers and others do not. On the other hand, if the ignore_empty argument is **FALSE**, TEXTJOIN includes the blank cells resulting from the formula.

CONCAT

The new **CONCAT** function replaces the previous CONCATENATE function and is used to join multiple text strings. (For backward compatibility with prior versions of Excel, CONCATENATE will continue to be available.) **Figure 7** presents an example of using CONCAT to join multiple text strings into one. Compare the formula in Figure 6 to the one in Figure 7 and notice the relative brevity and simplicity of the TEXTJOIN function, particularly regarding how it inserts commas into the concatenated test string.

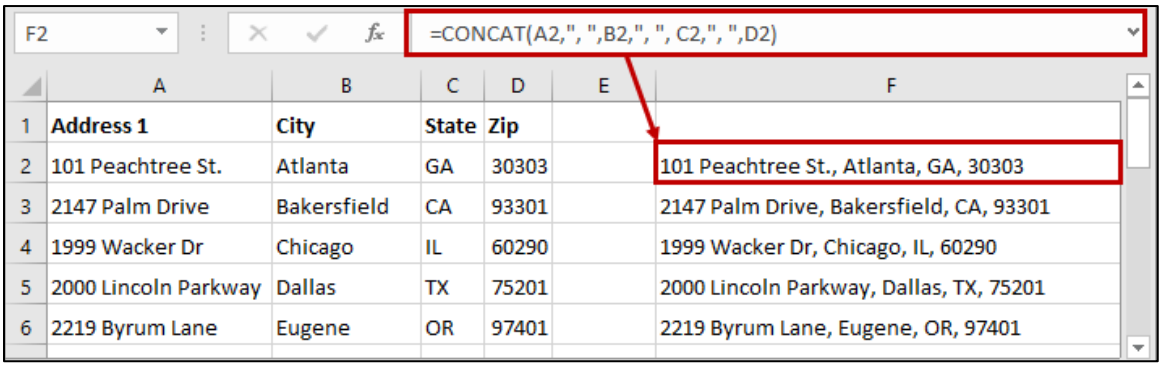
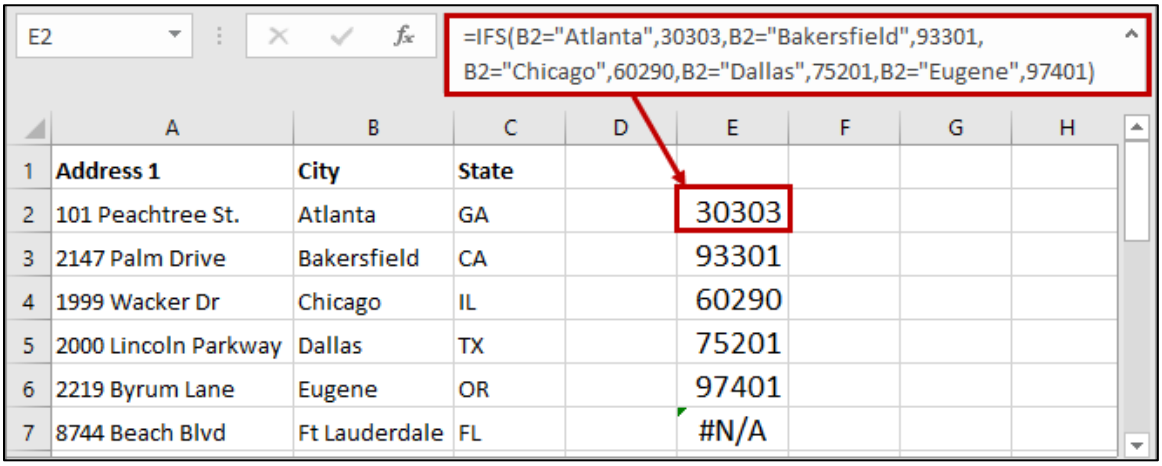


Figure 7 - Using CONCAT in Excel 2016 to Join Text Strings

IFS

Like Excel's **IF** function, the new **IFS** function allows you to perform conditional calculations. If one or more conditions evaluate as "true," the IFS function returns a value corresponding to the first condition satisfied. The IFS function returns **#N/A!** if none of the tested conditions are "true." **Figure 8** provides an example of how you can use the IFS function to simplify a process where you might have used nested IF functions in the same formula.

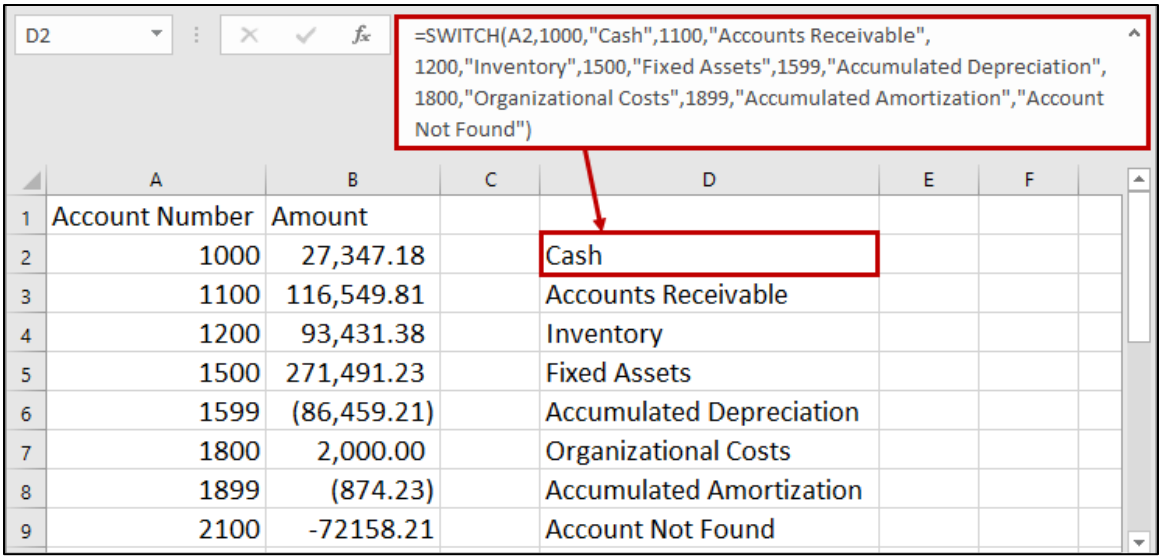


	A	B	C	D	E	F	G	H
1	**Address 1**	**City**	**State**					
2	101 Peachtree St.	Atlanta	GA		30303			
3	2147 Palm Drive	Bakersfield	CA		93301			
4	1999 Wacker Dr	Chicago	IL		60290			
5	2000 Lincoln Parkway	Dallas	TX		75201			
6	2219 Byrum Lane	Eugene	OR		97401			
7	8744 Beach Blvd	Ft Lauderdale	FL		#N/A			

Figure 8 - Testing Multiple Conditions Using Excel's IFS Function

SWITCH

You can use Excel's **SWITCH** function to evaluate a single expression against a list of up to 126 values and return the result corresponding to the first matching value. If SWITCH does not find a match, you can choose to return an optional default value. To illustrate, consider the worksheet and formula based on the SWITCH function pictured in **Figure 9**. In this example, the SWITCH function checks whether the value in cell A2 matches any of the following account numbers: 1000, 1100, 1200, 1500, 1599, 1800, or 1899. If SWITCH finds an exact match, it returns the corresponding account name – *Cash*, *Accounts Receivable*, etc. If SWITCH does not find an exact match, it returns *Account Not Found*. Like IFS, SWITCH can reduce your need for nested IF functions in a formula.



	A	B	C	D	E	F
1	Account Number	Amount				
2	1000	27,347.18		Cash		
3	1100	116,549.81		Accounts Receivable		
4	1200	93,431.38		Inventory		
5	1500	271,491.23		Fixed Assets		
6	1599	(86,459.21)		Accumulated Depreciation		
7	1800	2,000.00		Organizational Costs		
8	1899	(874.23)		Accumulated Amortization		
9	2100	-72158.21		Account Not Found		

Figure 9 - Using Excel's SWITCH Function Instead of Nested IF Functions

MAXIFS

The **MAXIFS** function allows you to identify and return the maximum value in a range of cells specified by a set of conditions or criteria you specify. Similar to **SUMIFS**, **COUNTIFS**, and **AVERAGEIFS**, the syntax for this function is

=MAXIFS(max_range, criteria_range1, criteria1, [criteria_range2, criteria2], ...)

where

max_range is the actual range of cells in which the maximum will be determined,

critera_range 1, **2, etc.** is the set of cells to evaluate with the criteria, and

criteria 1, 2, etc., are the criteria to evaluate.

To illustrate, consider the example pictured in **Figure 10**, with multiple rows hidden for presentation purposes. In this illustration, MAXIFS identifies the maximum value in the range of **B2:B25** for *Munoz*, the specified salesperson.

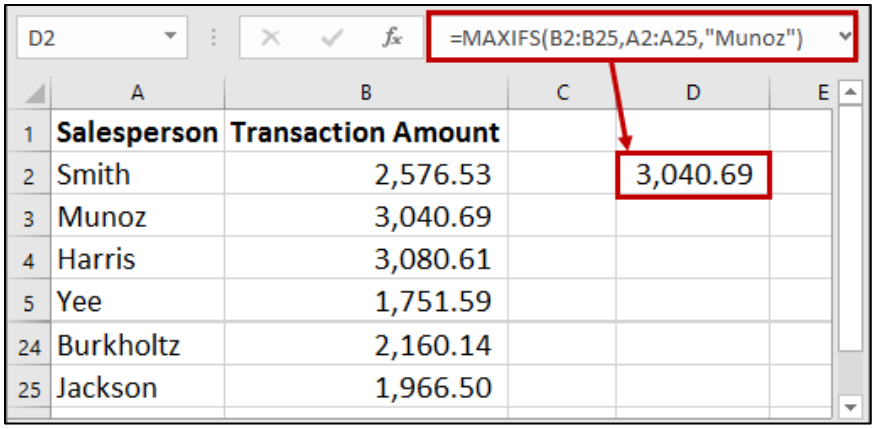


Figure 10 - Using MAXIFS to Identify the Largest Transaction for a Salesperson

A MAXIFS function can evaluate up to 127 conditions. For example, in **Figure 11**, MAXIFS calculates the result by examining two criteria: *Salesperson* and *Month*.

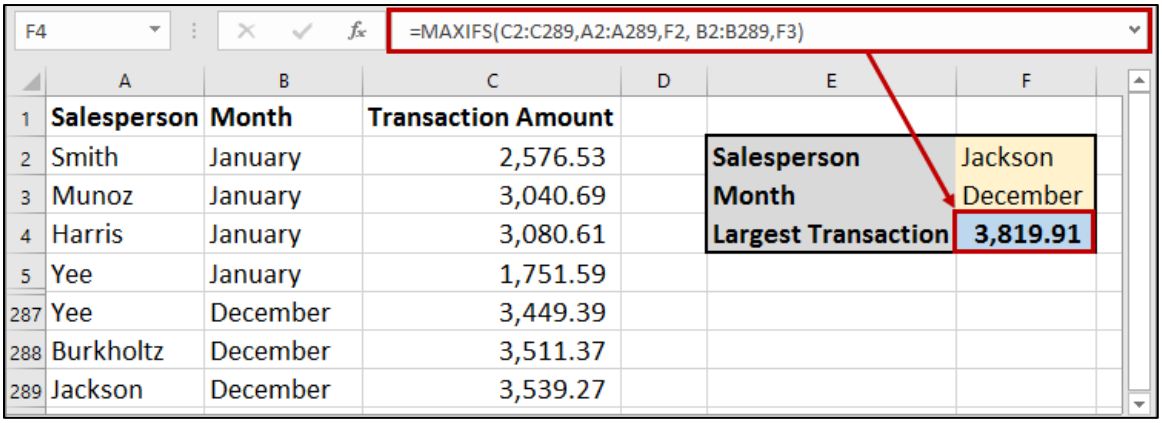


Figure 11 - MAXIFS Function Identifying the Largest Value Based on Multiple Conditions

MINIFS

The **MINIFS** function is the opposite of MAXIFS – you can use MINFS to identify the smallest value in a range based on up to 127 criteria. The syntax for MINIFS is similar to MAXIFS, with the apparent substitution of *MINIFS* for *MAXIFS* in the formula.

XLOOKUP – A Superior Alternative to VLOOKUP, HLOOKUP, and INDEX

Microsoft 365 subscribers gain access to powerful new functionality with continuous updates. Among the most useful new functions is **XLOOKUP**. XLOOKUP is an alternative to VLOOKUP, HLOOKUP, and INDEX. XLOOKUP works with traditional ranges or tables of data, and you can use it across worksheets or workbooks. In addition, XLOOKUP provides greater functionality and overcomes many of the limitations of the previous functions. For example, XLOOKUP defaults to an exact match, can perform lookups from the bottom up and top down, and sorting the lookup column or row for approximate matches is unnecessary. It also allows you to specify a range of

cells on which to perform the lookup instead of a column number in a lookup table. Hence, the order of the table columns in a lookup table does not matter. In addition, this feature allows XLOOKUP to look to the lookup column's left – something VLOOKUP cannot do.

The syntax of XLOOKUP is as follows:

XLOOKUP(lookup value, lookup array, return array, [if not found], [match mode], [search mode])

where:

Lookup value represents the value to be looked up.

Lookup array is the range or table column in which to search.

Return array is the range or table column from which to return a related value.

If not found (optional) represents what to return if the lookup value is not found; defaults to #NA.

Match mode (optional) determines how to perform the search; it defaults to an exact match.

Option Number	Option Behavior
0 or Omitted	Exact match (default)
-1	Exact match or next smaller item (default for VLOOKUP)
1	Exact match or the next larger item
2	Wildcard character match

The **Search mode** (optional) determines the search order and defaults to first-to-last.

Option Number	Option Behavior
1	Search first-to-last (default)
-1	Search last-to-first
2	Binary search sorted in ascending order
-2	Binary search sorted in descending order

In this first example, XLOOKUP will perform a reverse lookup. Since XLOOKUP uses a lookup array (column or row) instead of a lookup table, you can execute the lookup on any column. XLOOKUP can return the result from any column, even those to the left of the lookup column.

Nature's Softness carries an extensive inventory of quality facial products. Owner Debbie Nelson often looks up a vendor's part number when restocking a product. She uses a simple VLOOKUP formula to accomplish this task. However, she often needs to look up an internal SKU from a vendor's part number. Since the SKUs are in a column to the left of the column containing vendor part numbers, VLOOKUP cannot perform this task. While you can use the MATCH and INDEX functions to perform this task, Debbie has limited knowledge of their use. Her assistant pointed out that the XLOOKUP function is a Swiss Army knife for data lookups and built a simple formula to retrieve the desired information.

The following formula performs a reverse lookup using a vendor's part number to retrieve a SKU from the Inventory List.

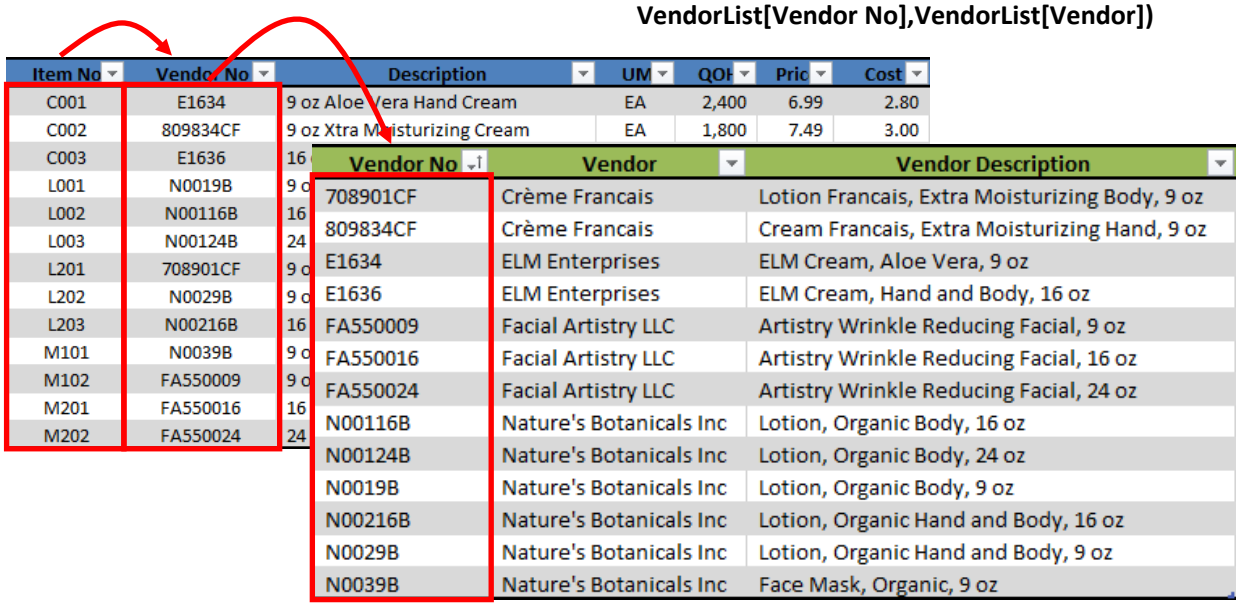
=XLOOKUP(C5,InventoryList[Vendor No],InventoryList[SKU])

However, when the formula does not identify a vendor's part number in the lookup array, XLOOKUP returns **#NA,** just like VLOOKUP. A minor change to the formula cures that shortcoming without using IFERROR. The following formula adds the fourth optional argument when it does not find a matching item. The argument can be a value, formula, cell reference, defined name, or text (enclosed in quotation marks). In this case, the formula displays "Item Not Found" when it does not find a match for a vendor part number in the lookup array.

=XLOOKUP(C6,InventoryList[Vendor No],InventoryList[SKU],"Item Not Found")

Debbie needs to retrieve the name of the vendor when reordering a product. Unfortunately, the vendor list is in another table. XLOOKUP overcomes this problem because it works across multiple worksheets and workbooks.

=XLOOKUP(XLOOKUP(B2,InventoryList[Item No],InventoryList[Vendor No]),
VendorList[Vendor No],VendorList[Vendor])



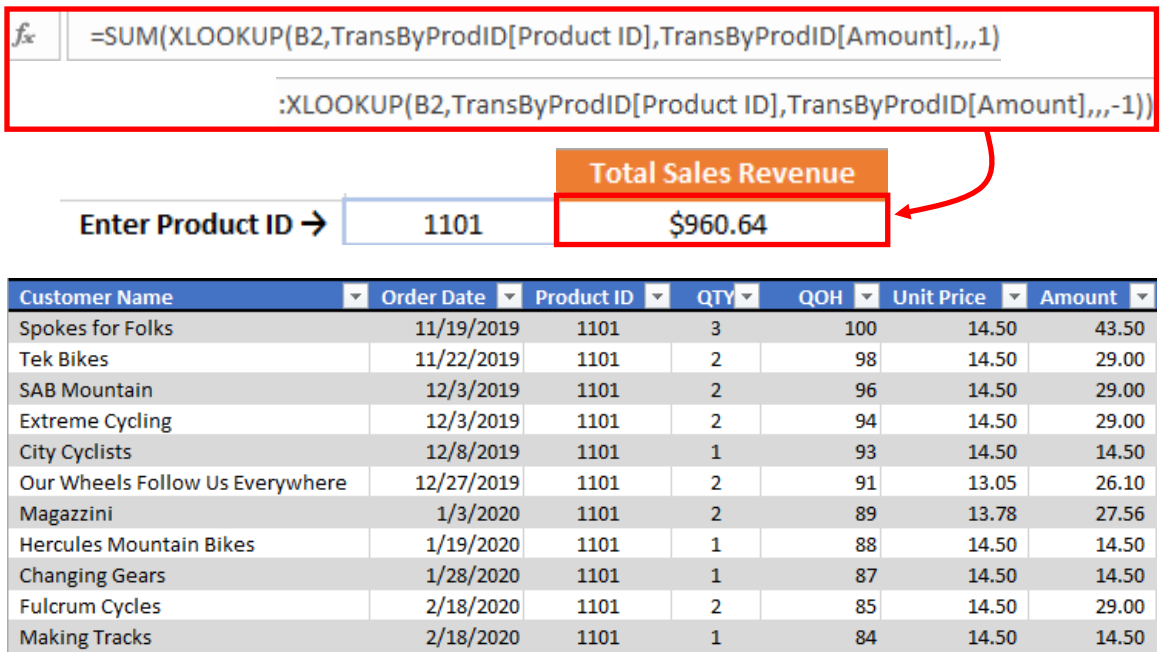
Item No	Vendor No	Description	UM	QOH	Price	Cost
C001	E1634	9 oz Aloe Vera Hand Cream	EA	2,400	6.99	2.80
C002	809834CF	9 oz Xtra Moisturizing Cream	EA	1,800	7.49	3.00
C003	E1636	16				
L001	N0019B	9 o				
L002	N00116B	16				
L003	N00124B	24				
L201	708901CF	9 o				
L202	N0029B	9 o				
L203	N00216B	16				
M101	N0039B	9 o				
M102	FA550009	9 o				
M201	FA550016	16				
M202	FA550024	24				

Vendor No	Vendor	Vendor Description
708901CF	Crème Francais	Lotion Francais, Extra Moisturizing Body, 9 oz
809834CF	Crème Francais	Cream Francais, Extra Moisturizing Hand, 9 oz
E1634	ELM Enterprises	ELM Cream, Aloe Vera, 9 oz
E1636	ELM Enterprises	ELM Cream, Hand and Body, 16 oz
FA550009	Facial Artistry LLC	Artistry Wrinkle Reducing Facial, 9 oz
FA550016	Facial Artistry LLC	Artistry Wrinkle Reducing Facial, 16 oz
FA550024	Facial Artistry LLC	Artistry Wrinkle Reducing Facial, 24 oz
N00116B	Nature's Botanicals Inc	Lotion, Organic Body, 16 oz
N00124B	Nature's Botanicals Inc	Lotion, Organic Body, 24 oz
N0019B	Nature's Botanicals Inc	Lotion, Organic Body, 9 oz
N00216B	Nature's Botanicals Inc	Lotion, Organic Hand and Body, 16 oz
N0029B	Nature's Botanicals Inc	Lotion, Organic Hand and Body, 9 oz
N0039B	Nature's Botanicals Inc	Face Mask, Organic, 9 oz

Figure 12 – XLOOKUP Can Look Across Worksheets or Workbooks

The formula shown in **Figure 12** performs a double lookup to return the value from the vendor list. It first uses XLOOKUP to find the vendor product number in the Inventory List, which is an input to XLOOKUP to return the vendor name from the Vendor List on another worksheet.

XLOOKUP can return values (as in the previous formula), arrays (ranges of values), or references (a reference to a cell or range of cells, such as B3 or A1:A10). Because XLOOKUP can search first-to-last or last-to-first, you can use it to identify the cell references of the first and last sales transactions of a specific product, which you then can use as inputs to other functions, such as SUM, COUNT, AVERAGE, MIN, or MAX. Figure 13 displays a formula to sum sales revenue by product ID from a transaction table.



Customer Name	Order Date	Product ID	QTY	QOH	Unit Price	Amount
Spokes for Folks	11/19/2019	1101	3	100	14.50	43.50
Tek Bikes	11/22/2019	1101	2	98	14.50	29.00
SAB Mountain	12/3/2019	1101	2	96	14.50	29.00
Extreme Cycling	12/3/2019	1101	2	94	14.50	29.00
City Cyclists	12/8/2019	1101	1	93	14.50	14.50
Our Wheels Follow Us Everywhere	12/27/2019	1101	2	91	13.05	26.10
Magazzini	1/3/2020	1101	2	89	13.78	27.56
Hercules Mountain Bikes	1/19/2020	1101	1	88	14.50	14.50
Changing Gears	1/28/2020	1101	1	87	14.50	14.50
Fulcrum Cycles	2/18/2020	1101	2	85	14.50	29.00
Making Tracks	2/18/2020	1101	1	84	14.50	14.50

Figure 13 – Using XLOOKUP to Sum Sales of a Specified Product ID

XLOOKUP is here, and it can do everything VLOOKUP can and more. With similar functionality advances, XLOOKUP and its companion XMATCH will revolutionize how you use lookup functions. In addition, they are easy to use, intuitive, and have productivity-enhancing analytical power.

Collaborating More Effectively in Excel

Multiple newer options exist for multi-user collaboration in Excel, and they are all superior to legacy techniques used for this purpose. This section focuses on more recent and better possibilities for collaborating more effectively with others when working in Excel.

Co-Authoring Excel Workbooks

Instead of emailing copies of workbooks to others, you may be able to send a ***link*** to the document instead of the document itself. This feature is contingent upon which version of Excel you use (2016 and newer, including subscription-based licenses) and whether you store your documents in a cloud-based repository such as **OneDrive for Business**. You can send a link by choosing the **Share with People** option shown in **Figure 15**. Sending a link can facilitate real-time collaboration because all users will access and edit the same workbook in real-time.

Co-authoring allows multiple users to access and edit Excel workbooks simultaneously, using the Internet as the "backbone" of the process. In addition to co-authoring in Excel, you can co-author in Word, PowerPoint, and OneNote. With co-authoring, any user with appropriate rights can access and edit documents, and you can make these edits simultaneously with other users working on the same file. Thus, the author of an Excel workbook used to calculate the tax accrual

for a company could share that workbook with its external accountant in public practice. The author and the external accountant could then review and edit the workbook in real-time.

To set up co-authoring for a workbook, you must have a Microsoft 365 subscription and store your document in a cloud-based location such as OneDrive, OneDrive for Business, or SharePoint Online. After you do so, click **Share** near the upper right corner of the workbook. Next, you can enter the **Link Settings** for the collaboration and send links to the document to all invited parties. **Figure 14** illustrates that process.

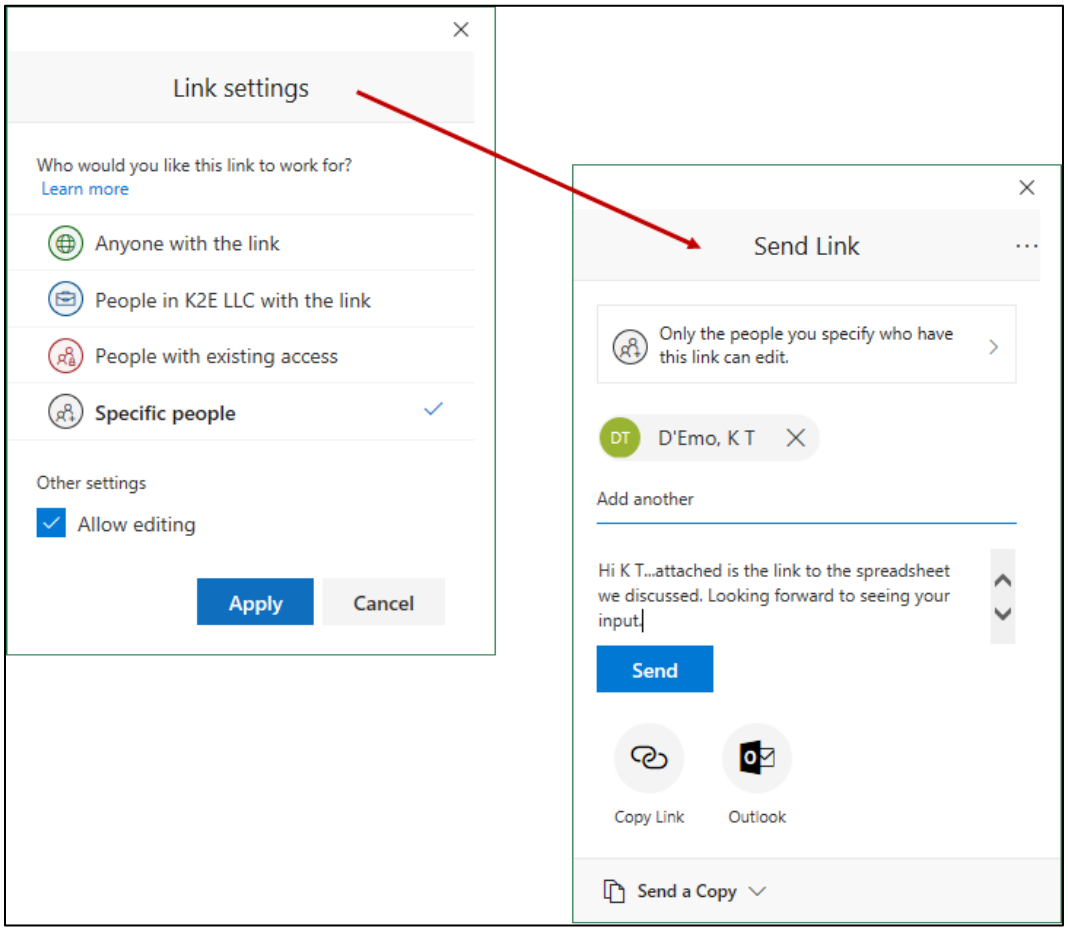


Figure 14 - Setting Up an Excel Workbook for Co-Authoring

Once you share the document, you and other users can simultaneously access and edit it. Further, changes from different users will appear almost instantaneously. Note that other users do not need Excel installed on their devices to access and edit the workbook. If they do not have Excel installed on their device, the workbook will open in **Excel Online**.

A common question regarding co-authoring is, “What happens when multiple users attempt to edit the same cell simultaneously.” In this case, Excel uses “cell locking” to allow only one person

access to a cell at any point in time. For example, if Sara clicks a cell to edit it, Juan cannot select that same cell until Sara moves on to another. The last change made to a cell is the one that saves when the workbook saves, either through a manual save process or by Excel's **AutoSave** feature. However, remember that if you store the workbook in OneDrive, OneDrive for Business, or SharePoint Online, you can access previous workbook versions and "roll back" the workbook to a prior version.

Collaborating Using SharePoint Server, SharePoint Online, and Teams

SharePoint Server, **SharePoint Online**, and **Teams** provide three additional opportunities to collaborate on Excel and other Microsoft Office documents. If you have access to one or more of these tools – and most Microsoft 365/Office 365 subscribers will – then you have terrific options for collaborating with other users – both inside and outside your organization – to get better results in less time.

Publishing to SharePoint Server and SharePoint Online

From a practical perspective, the most significant difference between SharePoint Server and SharePoint Online is that SharePoint Server is a local resource while SharePoint Online is Cloud-based. Functionally, the two platforms are otherwise nearly identical. Excel can use either SharePoint platform to facilitate collaboration using the co-authoring experience discussed previously. To take advantage of this capability, save your workbook to your SharePoint environment. After that, multiple users can access the workbook simultaneously and co-author the document.

You can co-author using the Excel Web App only if everyone uses the Excel Web App to access the workbook in this environment. If anyone uses the desktop client application of Excel to access the workbook, co-authoring in Excel Web App is disabled for that workbook as long as it is open in the client application.

Given the similarity of this process to that previously discussed when saving the workbook in OneDrive, you might ask, "*What is an advantage of using SharePoint for Excel co-authoring*?" SharePoint provides several options not otherwise available if you choose to co-author through OneDrive.

1. **Permissions**. With SharePoint, you can establish customized permissions at the user level to control which users can access and edit documents. These permissions can be very granular if necessary, including providing access at the site, library, folder, or document level.

2. **Versioning**. SharePoint offers the option of creating versions of your documents, so you can refer to a prior version, if necessary, as you review a workbook for completeness and accuracy.

3. **Number of versions**. If you enable versions in SharePoint, you can also establish the number of versions to retain. This feature helps manage the amount of storage space used by large documents, with potentially numerous revisions.

4. **Check out**. Finally, SharePoint supports document check-out. When you check out a workbook, SharePoint locks it for editing until you check it back into the library.

Co-Authoring Using Microsoft Teams

The newest and perhaps best form of collaborating involves using **Microsoft Teams** as the platform for co-authoring. Microsoft created Teams around the idea that the most significant needs of information workers are communication and collaboration. In other words, these capabilities are not an afterthought; instead, Microsoft incorporated them into the fundamental design of Teams.

The Teams platform takes advantage of the functionality provided by SharePoint Online to facilitate co-authoring but in a more user-friendly environment. More to the point, Microsoft enables co-authoring by default in Teams. Thus, if you open a workbook stored in Teams, and other users have the same workbook open, all users can edit the workbook and see each other's changes in real-time. Notably, as shown in **Figure 15**, when you open a workbook in Teams, it opens in the Excel Web App. However, you can click **Open in Desktop App** if you require access to features not available in the Web App.

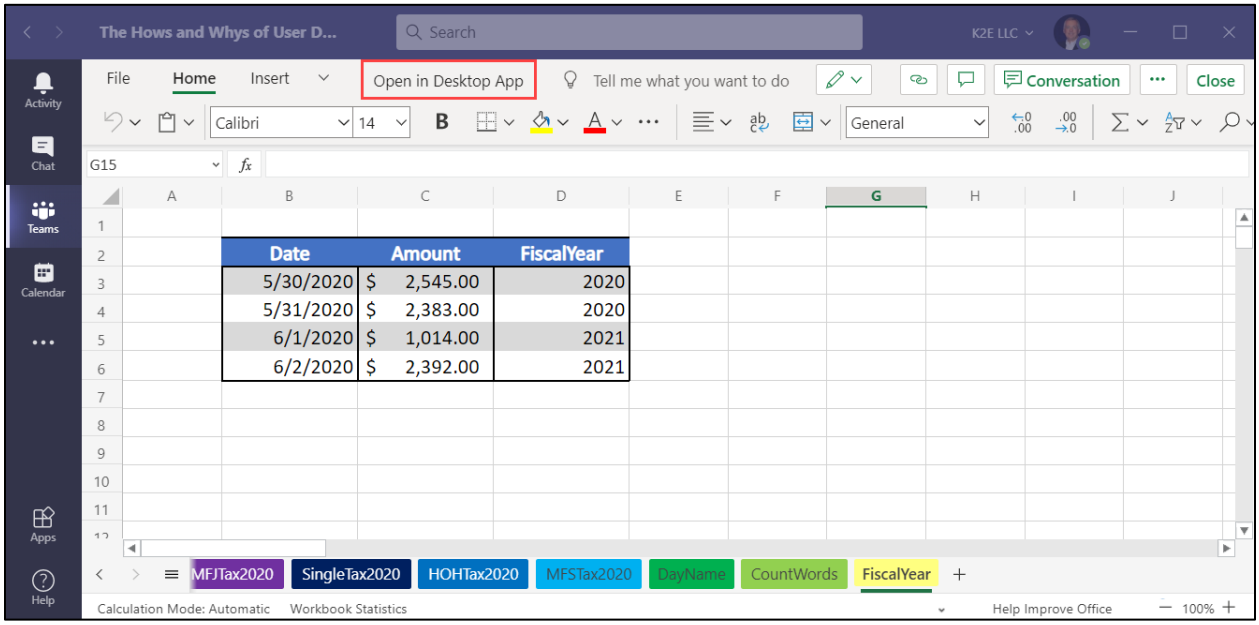


Figure 15 - Opening a Workbook in Teams Using the Excel Web App

In sum, as more users take advantage of the Teams platform, expect to see significant increases in collaboration because of the ease of doing so in Teams.

Dynamic Arrays – New Forms of Array Power

You can access more powerful array-related features using Excel version 1907 or newer through a Microsoft 365/Office 365 subscription. Specifically, you can use *dynamic arrays* and six new functions to get more done in less time. Further, if you are in a dynamic array-aware Excel version, you no longer need to ensure braces surround your array formulas. Instead, you can press the **Enter** key upon entering an array formula and not worry about the **CTRL** + **SHIFT** + **ENTER** keyboard sequence.

A dynamic array is a range of data that automatically resizes as users add or delete data. But do not confuse a dynamic array with a table, for these two features are decidedly different. Whereas a table can serve as a dynamically resizing range of data, it cannot do everything a dynamic array can. For example, you can use dynamic arrays to sort or filter data without disturbing the original data, which is impossible with tables. On the other hand, dynamic arrays cannot do everything tables can. To illustrate, you cannot create a data model from multiple dynamic arrays.

As shown below, dynamic arrays allow us to break free from the "one-cell, one formula" mentality that has always existed in spreadsheets. Before dynamic arrays, we had to enter formulas into every cell where we wanted calculation results to display. Dynamic arrays change that by allowing us to create a single formula, and its results will appear in as many cells as necessary, given the volume of data under consideration. To illustrate, consider the example provided in **Figure 16**. In this example, the user entered the formula shown in the formula bar *only into cell D2*. However, the formula *dynamically copied* itself so that its results list all the unique values in the range.

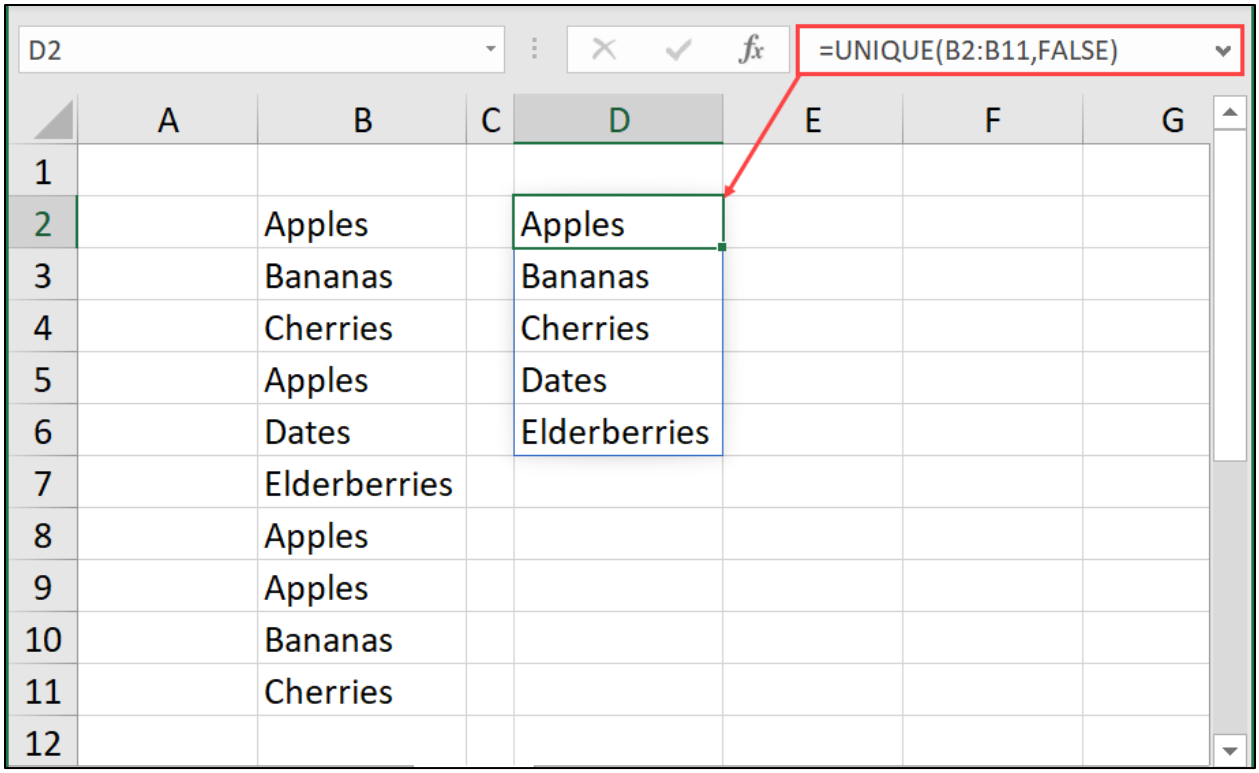


Figure 16 – First Example of a Formula Based on a Dynamic Array

In the example presented, note that cells B2 through B11 are a traditional range of data. However, we could have used the dynamic array formula with a table supplying the data. The **UNIQUE** function illustrated in Figure 16 is one of six new functions in dynamic array-aware versions of Excel. As its name implies, UNIQUE extracts all the unique entries from an array. Following is a listing of the other five functions.

1. **FILTER** filters a dynamic array based on criteria specified in the formula. Neither FILTER nor the other five dynamic array formulas alter the source data. Instead, each formula displays its results as a separate output range.

2. You can again use SORT to arrange data in a dynamic array in ascending or descending order without disturbing the source data's arrangement.

3. **SORTBY** sorts based on values in a corresponding range or array.

4. **RANDARRAY** returns an array of random numbers.

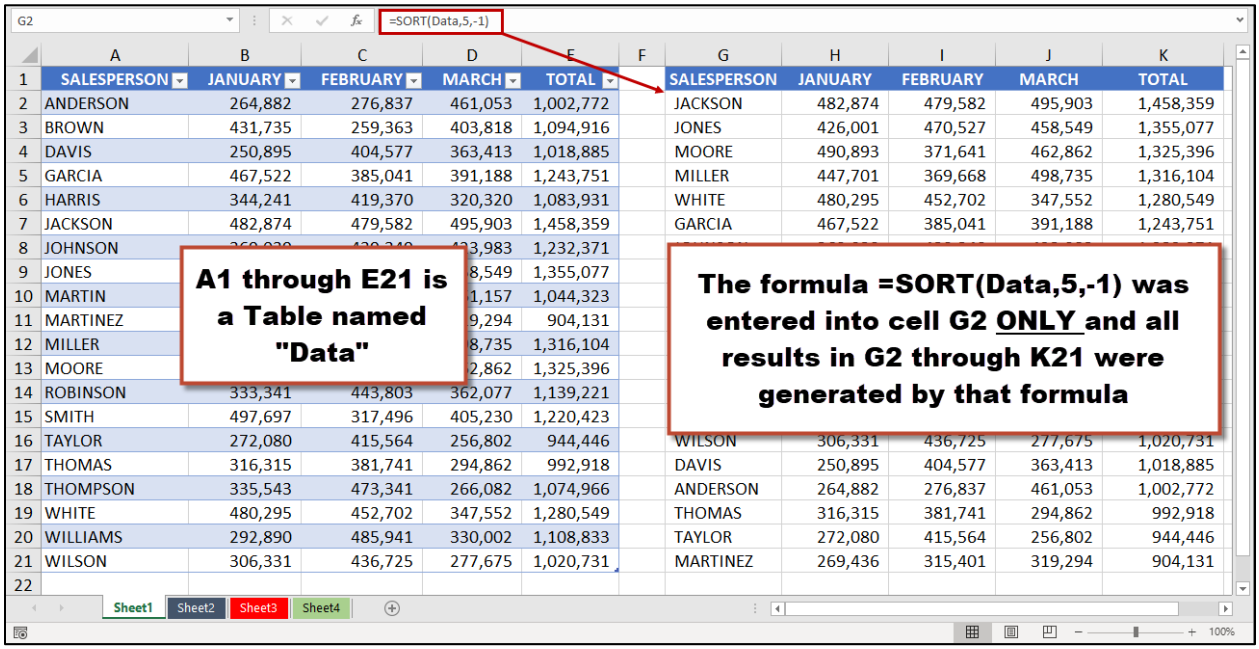
5. You can use the **SEQUENCE** function to generate a listing of sequential numbers displayed in an array.

Note that each of the six functions outlined above works only in the context of dynamic arrays.

To provide an example of how you can use one of the new functions on a dynamic array, consider the example in **Figure 17**. The left side of the image shows a table named "Data." A user must sort the data in that table in descending order based on the "Total" column values. However, she does not want to change the data's sort order in the source table. Instead, she enters the following simple formula into the worksheet to achieve the objective.

=SORT(Data,5,-1)

The formula sorts the dynamic array, known as *Data*, on the *fifth* column, in descending order (*-1*). Equally important, it left the original data set unchanged.



	A	B	C	D	E	F	G	H	I	J	K
1	SALESPERSON	JANUARY	FEBRUARY	MARCH	TOTAL		SALESPERSON	JANUARY	FEBRUARY	MARCH	TOTAL
2	ANDERSON	264,882	276,837	461,053	1,002,772		JACKSON	482,874	479,582	495,903	1,458,359
3	BROWN	431,735	259,363	403,818	1,094,916		JONES	426,001	470,527	458,549	1,355,077
4	DAVIS	250,895	404,577	363,413	1,018,885		MOORE	490,893	371,641	462,862	1,325,396
5	GARCIA	467,522	385,041	391,188	1,243,751		MILLER	447,701	369,668	498,735	1,316,104
6	HARRIS	344,241	419,370	320,320	1,083,931		WHITE	480,295	452,702	347,552	1,280,549
7	JACKSON	482,874	479,582	495,903	1,458,359		GARCIA	467,522	385,041	391,188	1,243,751
8	JOHNSON			3,983	1,232,371						
9	JONES			8,549	1,355,077						
10	MARTIN			1,157	1,044,323						
11	MARTINEZ			9,294	904,131						
12	MILLER			8,735	1,316,104						
13	MOORE			2,862	1,325,396						
14	ROBINSON	333,341	443,803	362,077	1,139,221						
15	SMITH	497,697	317,496	405,230	1,220,423						
16	TAYLOR	272,080	415,564	256,802	944,446		WILSON	306,331	436,725	277,675	1,020,731
17	THOMAS	316,315	381,741	294,862	992,918		DAVIS	250,895	404,577	363,413	1,018,885
18	THOMPSON	335,543	473,341	266,082	1,074,966		ANDERSON	264,882	276,837	461,053	1,002,772
19	WHITE	480,295	452,702	347,552	1,280,549		THOMAS	316,315	381,741	294,862	992,918
20	WILLIAMS	292,890	485,941	330,002	1,108,833		TAYLOR	272,080	415,564	256,802	944,446
21	WILSON	306,331	436,725	277,675	1,020,731		MARTINEZ	269,436	315,401	319,294	904,131
22											

Figure 17 – Sorting a Dynamic Array with the SORT Function

Dynamic array formulas remain a relatively new and obscure feature in Excel. However, as more users become aware of their power and ease of use, their popularity will skyrocket. Moreover, suppose you are using a dynamic array-aware version of Excel. In that case, you can use legacy array formulas without worrying about the **CTRL** + **SHIFT** + **ENTER** keystroke sequence to enter your array formulas.

Excel's GROUPBY And PIVOTBY Functions

Formula-Based Alternatives To PivotTables

In 1994, Microsoft added PivotTables to Excel. Since then, countless business professionals have adopted this feature as their *de facto* way of summarizing and analyzing information – for good

reasons! A PivotTable offers powerful data summarization options without using time-consuming and error-prone formulas.

However, PivotTables have limitations. One such restriction is that they are rigid, with few options for manipulating and presenting their summarized data. To address that issue, Microsoft recently added **GROUPBY** and **PIVOTBY** – two functions in Excel that offer the computational advantages of traditional PivotTables while at the same time providing formula-based flexibility.

Getting Started With GROUPBY

Like Excel's **SUM**, **SUBTOTAL**, and other functions, GROUPBY is an Excel function you can use to summarize information quickly, easily, and accurately. However, GROUPBY differs from other Excel features as it allows you to quickly and easily aggregate data by categories, such as product line, region, and customer. In other words, you can use GROUPBY to generate a data summary filtered to specific criteria.

Although up to eight arguments can be present in the formula, only three are necessary when creating a formula based on the GROUPBY function.

1. First, indicate the row(s) in a table or range by which you will summarize your data; in the formula below, the **(FinancialData[Segment])** argument serves that purpose. Note that you can specify multiple rows; if you do, you will have numerous groupings in your formula results.

2. Second, specify the column(s) that serves as the numeric values you wish to summarize; in the example below, the **(FinancialData[Net Sales])** argument serves that purpose. Like rows, you can specify multiple columns; if you do, your report will have multiple aggregations.

3. Finally, specify the aggregation function; in the example below, that is the **SUM** argument.

To illustrate, consider the formula shown below.

=GROUPBY(FinancialData[Segment],FinancialData[Net Sales],SUM)

You could use this formula to summarize the amounts in FinancialData's "Net Sales" column by segment. Upon creating the formula, your output might resemble that shown in **Figure 18**.

Channel Partners	$	1,800,593.64
Enterprise	$	19,611,694.38
Government	$	52,504,260.67
Midmarket	$	2,381,883.08
Small Business	$	42,427,918.50
Total	$	118,726,350.26

Figure 18 - Output Generated By Simple GROUPBY Formula

In this simple example, the GROUPBY function creates summaries of the sales data for each of the specified business segments. You need not sort, filter, or otherwise manipulate or rearrange the data to use GROUPBY.

Multiple Grouping Levels

Next, assume you need to summarize your data not only by **Segment** but also by **Country**. In that case, you can modify your formula to include **Country** as a second summarizing criterion. The modified formula appears below.

=GROUPBY(FinancialData[[Country]:[Segment]],FinancialData[Net Sales],SUM)

Upon entering the formula outlined above, the output changes to that pictured in **Figure 19**.

Channel Partners	Canada	$	491,164.14
Channel Partners	France	$	372,090.36
Channel Partners	Germany	$	336,425.88
Channel Partners	Mexico	$	234,379.08
Channel Partners	United States of America	$	366,534.18
Enterprise	Canada	$	3,967,491.25
Enterprise	France	$	3,890,890.63
Enterprise	Germany	$	4,086,826.25
Enterprise	Mexico	$	3,315,881.25
Enterprise	United States of America	$	4,350,605.00
Government	Canada	$	10,741,236.52
Government	France	$	12,127,782.72
Government	Germany	$	11,452,895.94
Government	Mexico	$	9,791,599.38
Government	United States of America	$	8,390,746.11
Midmarket	Canada	$	510,213.98
Midmarket	France	$	593,802.08
Midmarket	Germany	$	301,344.75
Midmarket	Mexico	$	511,136.40
Midmarket	United States of America	$	465,385.88
Small Business	Canada	$	9,177,549.00
Small Business	France	$	7,369,606.50
Small Business	Germany	$	7,327,848.00
Small Business	Mexico	$	7,096,356.00
Small Business	United States of America	$	11,456,559.00
Total		$	118,726,350.26

Figure 19 - GROUPBY Formula With Two Summarizing Criteria

As with the example in Figure 18, you can quickly summarize data and arrange it vertically without creating a PivotTable.

Summarizing Multiple Columns Of Data By Multiple Criteria Using GROUPBY

Continuing with the same data, let us use GROUPBY to summarize multiple columns of data using various criteria. This illustration will summarize **Net Sales** and **Gross Profit** by **Segment** and **Country**. Sometimes, it may be necessary to rearrange your data when summarizing multiple columns using GROUPBY. Specifically, the summarized columns must be contiguous when using this function. Therefore, if the summarizing columns are not contiguous, rearrange them in that order. Alternatively, you could insert a new “helper” column contiguous to the other columns of interest and use a formula to link the value from the original column into the helper column. We

can use the following formula to summarize multiple columns by multiple criteria – precisely, Segment followed by Country, and Net Sales followed by Gross Profit. **Figure 20** illustrates the formula's results.

=GROUPBY(FinancialData[[Country]:[Segment]],
FinancialData[[Net Sales]:[Gross Profit]],SUM)

Channel Partners	Canada	$ 491,164.14	$ 358,978.14
Channel Partners	France	$ 372,090.36	$ 271,581.36
Channel Partners	Germany	$ 336,425.88	$ 247,358.88
Channel Partners	Mexico	$ 234,379.08	$ 170,890.08
Channel Partners	United States of America	$ 366,534.18	$ 267,994.68
Enterprise	Canada	$ 3,967,491.25	$ (121,508.75)
Enterprise	France	$ 3,890,890.63	$ (95,749.38)
Enterprise	Germany	$ 4,086,826.25	$ (101,473.75)
Enterprise	Mexico	$ 3,315,881.25	$ (120,678.75)
Enterprise	United States of America	$ 4,350,605.00	$ (175,135.00)
Government	Canada	$ 10,741,236.52	$ 2,258,471.52
Government	France	$ 12,127,782.72	$ 2,709,915.22
Government	Germany	$ 11,452,895.94	$ 2,677,175.94
Government	Mexico	$ 9,791,599.38	$ 2,039,159.38
Government	United States of America	$ 8,390,746.11	$ 1,703,451.11
Midmarket	Canada	$ 510,213.98	$ 132,488.98
Midmarket	France	$ 593,802.08	$ 164,542.08
Midmarket	Germany	$ 301,344.75	$ 85,354.75
Midmarket	Mexico	$ 511,136.40	$ 150,546.40
Midmarket	United States of America	$ 465,385.88	$ 127,170.88
Small Business	Canada	$ 9,177,549.00	$ 900,799.00
Small Business	France	$ 7,369,606.50	$ 730,731.50
Small Business	Germany	$ 7,327,848.00	$ 771,973.00
Small Business	Mexico	$ 7,096,356.00	$ 667,606.00
Small Business	United States of America	$ 11,456,559.00	$ 1,072,059.00
Total		**$ 118,726,350.26**	**$ 16,893,702.26**

Figure 20 - Summarizing Multiple Columns Of Data Using GROUPBY

Optional Arguments For GROUPBY Summarizations

As indicated earlier, there are only three mandated arguments for using the GROUPBY function: 1) row fields, 2) values, and 3) the summarizing function, such as SUM, COUNT, and AVERAGE. However, you can use up to five optional arguments to enhance your formula's results. We discuss each of these discussed below.

1. The *field_headers* argument allows you to specify whether your *row_fields* and *values* have headers and whether those headers should appear in your formula's results.
2. The *total_depth* argument determines whether row headers should contain totals.
3. You can use the *sort_order* argument to determine how Excel should sort rows in your formula's results.
4. A *filter_array* indicates whether to consider a corresponding row of data when filtering.
5. The *field_relationship* argument specifies the relationship fields when multiple columns are provided to row fields.

Although a complete discussion of each of the five options is beyond the scope of this session, a simple example can help you to understand the power of these options. For instance, suppose you want the output shown in Figure 18 to display sorted in *descending* order based on the summarized values. To achieve that result, you could modify the formula as shown below to indicate that you want the data sorted on the second column (thus, the entry of "2" in the formula) and, because you prefaced your entry of "2" with a minus sign, Excel knows to sort the data in *descending* order.

=GROUPBY(FinancialData[Segment],FinancialData[Net Sales],SUM,,,-2)

You can learn more about GROUPBY's optional arguments at https://k2e.fyi/groupby.

PIVOTBY – An Even More Flexible And Powerful Tool

Excel's **PIVOTBY** function provides even more flexibility and utility than the GROUPBY function. Specifically, PIVOTBY allows you to create data summaries and reports using *any* Excel function. Fortunately, the process for using PIVOTBY is like that of using GROUPBY, so if you know how to work with GROUPBY, the learning curve to adapt to PIVOTBY is short.

When using PIVOTBY, Excel requires you to specify four arguments:

1. **Row fields**, to describe which column of data to use as row labels,
2. **Column fields**, to identify which column of data to use as column headers,
3. **Values**, to indicate which column of data contains the data to summarize, and
4. **Function**, the summarizing function you want to use to analyze the data.

Using the data from the previous example, we could build the following formula to summarize our sales data by channel and country. The formula to accommodate that need appears below, along with the report generated by the formula.

=PIVOTBY(FinancialData[Segment],FinancialData[Country],FinancialData[Net Sales],SUM)

	Canada	France	Germany	Mexico	United States of America	Total
Channel Partners	491,164.14	372,090.36	336,425.88	234,379.08	366,534.18	1,800,593.64
Enterprise	3,967,491.25	3,890,890.63	4,086,826.25	3,315,881.25	4,350,605.00	19,611,694.38
Government	10,741,236.52	12,127,782.72	11,452,895.94	9,791,599.38	8,390,746.11	52,504,260.67
Midmarket	510,213.98	593,802.08	301,344.75	511,136.40	465,385.88	2,381,883.08
Small Business	9,177,549.00	7,369,606.50	7,327,848.00	7,096,356.00	11,456,559.00	42,427,918.50
Total	**24,887,654.89**	**24,354,172.28**	**23,505,340.82**	**20,949,352.11**	**25,029,830.17**	**118,726,350.26**

Figure 21 - Sample Report Generated With Excel's PIVOTBY Function

If necessary, you could use any of the seven available optional arguments to modify your PIVOTBY formula so that the results meet your needs better. Several of these are similar to the optional arguments available with GROUPBY, and you can learn more about PIVOTBY's optional arguments at https://k2e.fyi/pivotby.

For example, suppose we modify the original formula to that shown below.

=PIVOTBY(FinancialData[Segment],FinancialData[Country],FinancialData[Net Sales],SUM,,-1)

In this case, we have added two commas and a "-1" to the end of the PIVOTBY formula. That modification causes Excel to display the PivotTable with the Grand Total as the first numerical row of the report instead of the last.

	Canada	France	Germany	Mexico	United States of America	Total
Total	24,887,654.89	24,354,172.28	23,505,340.82	20,949,352.11	25,029,830.17	118,726,350.26
Channel Partners	491,164.14	372,090.36	336,425.88	234,379.08	366,534.18	1,800,593.64
Enterprise	3,967,491.25	3,890,890.63	4,086,826.25	3,315,881.25	4,350,605.00	19,611,694.38
Government	10,741,236.52	12,127,782.72	11,452,895.94	9,791,599.38	8,390,746.11	52,504,260.67
Midmarket	510,213.98	593,802.08	301,344.75	511,136.40	465,385.88	2,381,883.08
Small Business	**9,177,549.00**	**7,369,606.50**	**7,327,848.00**	**7,096,356.00**	**11,456,559.00**	**42,427,918.50**

Figure 22 - Sample PIVOTBY-Based Formula With Optional Argument

Although PivotTables remain entrenched firmly into many business professionals' Excel repertoire, GROUPBY and PIVOTBY are welcome additions to the lineup. Notably, because they are both formula-based, these functions update dynamically as your data changes – there is no need to initiate a refresh, such as with a PivotTable.

Building Charts From Using GROUPBY And PIVOTBY

One ancillary advantage of using the GROUPBY and PIVOTBY functions is creating dynamic charts linked to the aggregations produced using these functions. Specifically, you can take the values

produced by GROUPBY and PIVOTBY and link them directly into an Excel chart. This may not seem significant initially because of the presence of PivotCharts and the ability to generate PivotCharts from PivotTables. However, there is a considerable limitation associated with PivotCharts. Specifically, Excel does not provide access to all chart types if you attempt to create a PivotChart. For example, suppose you created a PivotTable and wanted to create a corresponding PivotChart in the form of a Treemap. If you attempt to do so, Excel displays the message in **Figure 23**.

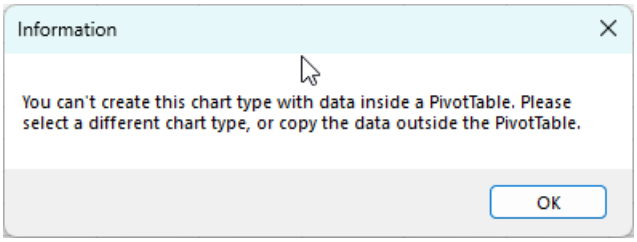


Figure 23 - Error Message Associated With Attempting To Create Certain Types Of Charts As PivotCharts

This message arises because Excel excludes the following chart types from the PivotChart library.

- Box and Whisker
- Funnel
- Histograms
- Map
- Stock
- Sunburst
- Treemaps

Fortunately, there is a simple solution to this issue. Specifically, you can use GROUPBY or PIVOTBY to create your aggregations and then build your chart from those aggregations. When you use this method, you avoid the limitations imposed by the absence of the seven chart types listed above.

To illustrate, let us return to the **FinancialData** table previously used in this session; the PivotTable pictured in **Figure 24** summarizes the data from that table.

Sum of Net Sales	Column Labels					
Row Labels	Canada	France	Germany	Mexico	United States of America	Grand Total
Government	$ 10,741,236.52	$ 12,127,782.72	$ 11,452,895.94	$ 9,791,599.38	$ 8,390,746.11	$ 52,504,260.67
Small Business	$ 9,177,549.00	$ 7,369,606.50	$ 7,327,848.00	$ 7,096,356.00	$ 11,456,559.00	$ 42,427,918.50
Enterprise	$ 3,967,491.25	$ 3,890,890.63	$ 4,086,826.25	$ 3,315,881.25	$ 4,350,605.00	$ 19,611,694.38
Midmarket	$ 510,213.98	$ 593,802.08	$ 301,344.75	$ 511,136.40	$ 465,385.88	$ 2,381,883.08
Channel Partners	$ 491,164.14	$ 372,090.36	$ 336,425.88	$ 234,379.08	$ 366,534.18	$ 1,800,593.64
Grand Total	**$ 24,887,654.89**	**$ 24,354,172.28**	**$ 23,505,340.82**	**$ 20,949,352.11**	**$ 25,029,830.17**	**$ 118,726,350.26**

Figure 24 - PivotTable Created From "FinancialData" Table

If we wanted the data displayed as a Treemap chart, we could not complete that task because, as shown below, Treemaps are unavailable as PivotCharts.

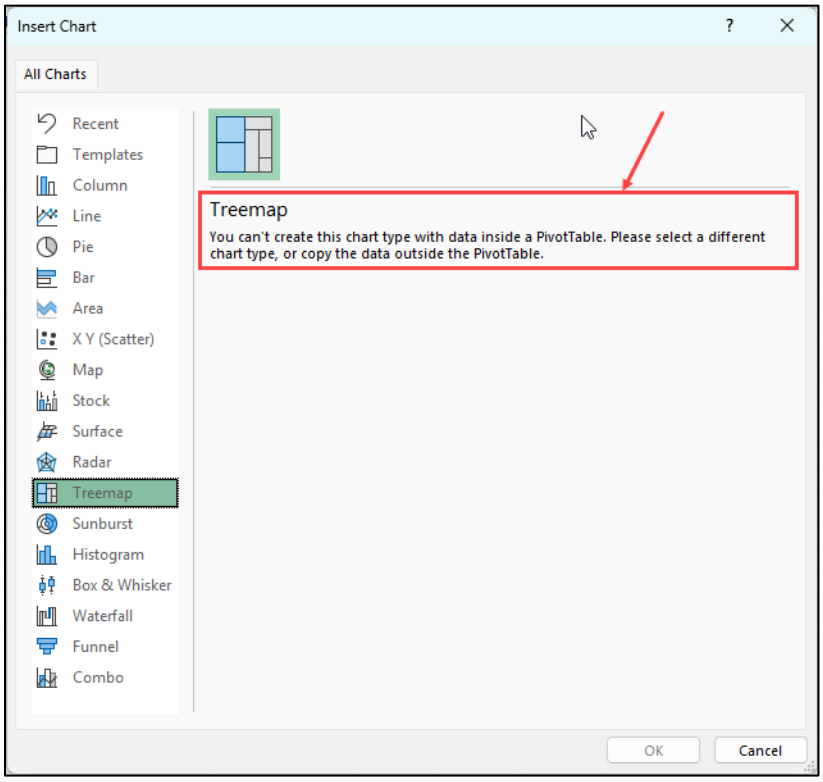


Figure 25 - Sample "Blocked" Treemap Chart

However, we could change our approach and use the results of a GROUPBY-based formula to achieve our desired results. Specifically, we could create the following formula:

GROUPBY(FinancialData[Segment],FinancialData[Net Sales],SUM,,0,-2)

The formula above creates the data summary pictured in **Figure 26**.

Government	52,504,260.67
Small Business	42,427,918.50
Enterprise	19,611,694.38
Midmarket	2,381,883.08
Channel Partners	1,800,593.64

Figure 26 - Data Summary Produced By GROUPBY-based Formula

With the tabulation complete, we could quickly generate the chart pictured in **Figure 27** to provide a graphical representation of the data. Again, note that this would not be possible with a PivotChart.

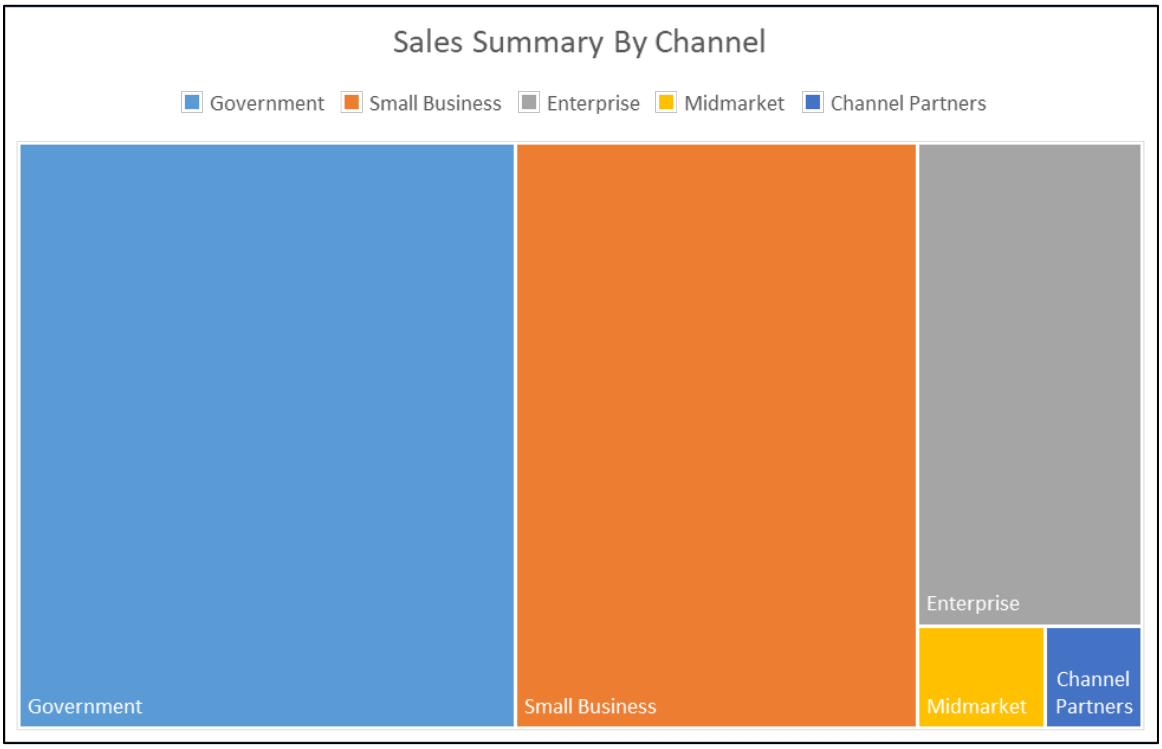


Figure 27 - Using Excel's GROUPBY Function To Bypass PivotChart Limitations

Summary

Excel continues to evolve rapidly, particularly for those who use a subscription-based license. Far from an exhaustive list of all new features, what you have learned in this session should put you well on your way to improved productivity, efficiency, and accuracy when working in Excel.