

Excel's Best New Features

With recent Excel updates provided through Microsoft 365/Office 365 subscriptions and the release of Office 2021, Microsoft continues to pack new features into the popular spreadsheet tool. Enhancements to data analysis tools, new functions, and improved collaboration opportunities are available in more recent Excel versions. 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 you can put Excel's best new features to work right away!

Introduction

With recent Excel updates provided through Microsoft 365/Office 365 subscriptions and the release of Office 2021, Microsoft continues to pack new features into the popular spreadsheet tool. Enhancements to data analysis tools, new functions, and improved collaboration opportunities are available in more recent Excel versions. 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 you can put Excel's best new features to work right away!

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 upon what version of Excel you have and how often you receive updates.

Perpetual Licenses

If you use a perpetual license of Excel – such as Excel 2016 or Excel 2019 – the concept of “new” is easy to understand. You receive new features whenever you upgrade to the next version of Excel. Thus, if you recently upgraded to Excel 2019, then all the new features included in that release are indeed “new” to you. You will not receive additional new features until you upgrade to the newest version of Excel, found in the recently released Office 2021 suite.

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 upon what **channel** you or your IT staff elect. In this environment, three primary channels are available: 1) **Current Channel**,

2) **Monthly Enterprise Channel**, and 3) **Semi-Annual Enterprise Channel**. As detailed in **Table 1**, feature updates occur on an unscheduled basis, 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 with an “early-look” at new and updated features that will soon debut in the Current Channel. So, effectively there are four primary update channels 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, Options** to display the window pictured in **Figure 1**.

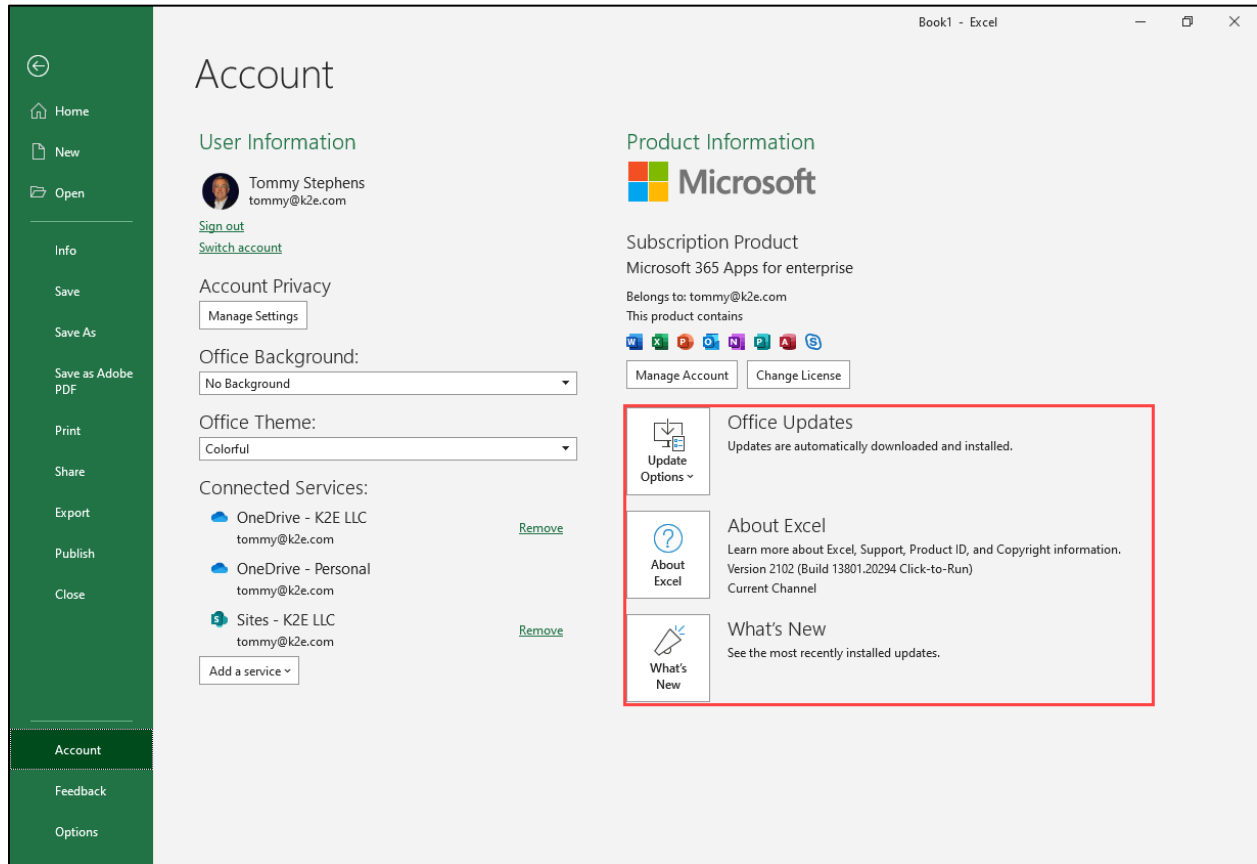


Figure 1 - Identifying Channel and Version

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, “2102” translates into the February 2021 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.

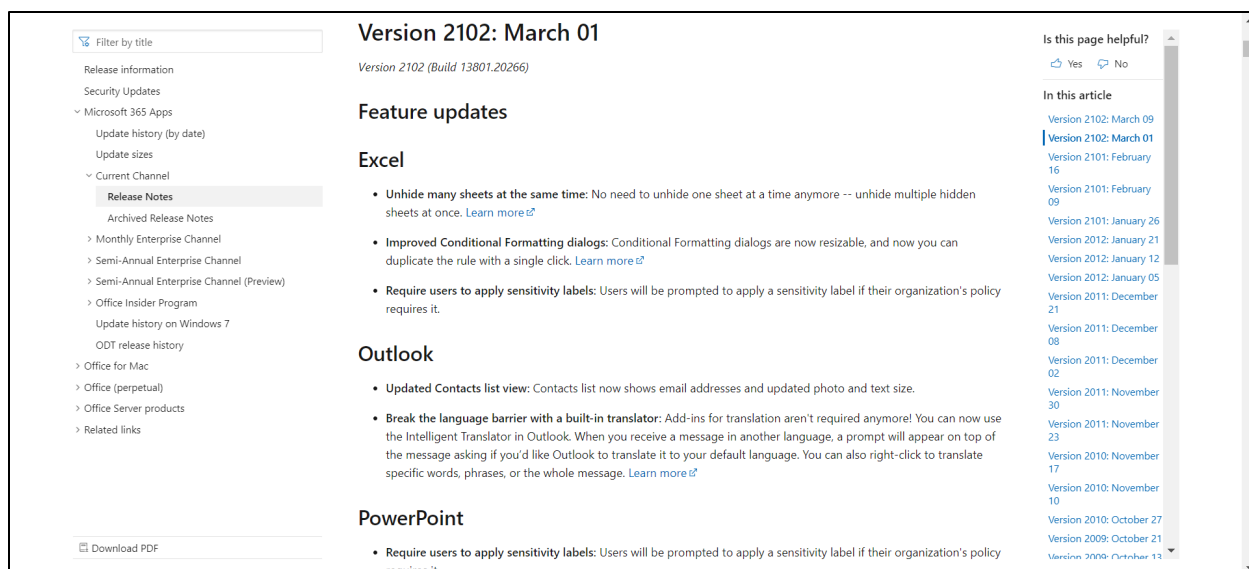


Figure 2 - Identifying New Features in Available in Microsoft 365 Apps

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 date and closing price by default. However, you can optionally choose to show *opening price*, *high price*, *low price*, and *volume*, if desired.

Using STOCKHISTORY

The syntax for using the STOCKHISTORY function can be relatively simple, as indicated below. However, note that of the arguments available, only the *stock* and *start_date* are required. 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 formula shown above.

	A	B	C	D	E	F	G	H
1	Date	Close						
2	1/2/2020	\$ 160.62						
3	1/3/2020	\$ 158.62						
253	12/30/2020	\$ 221.68						
254	12/31/2020	\$ 222.42						
255								

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 the *Open price*, *High price*, *Low price*, and *Volume* for each trading interval.

=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 amount of 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.

To utilize this feature, first ensure that you sign in to your organizational Power BI account. 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 just as if the data were "native" to Excel.

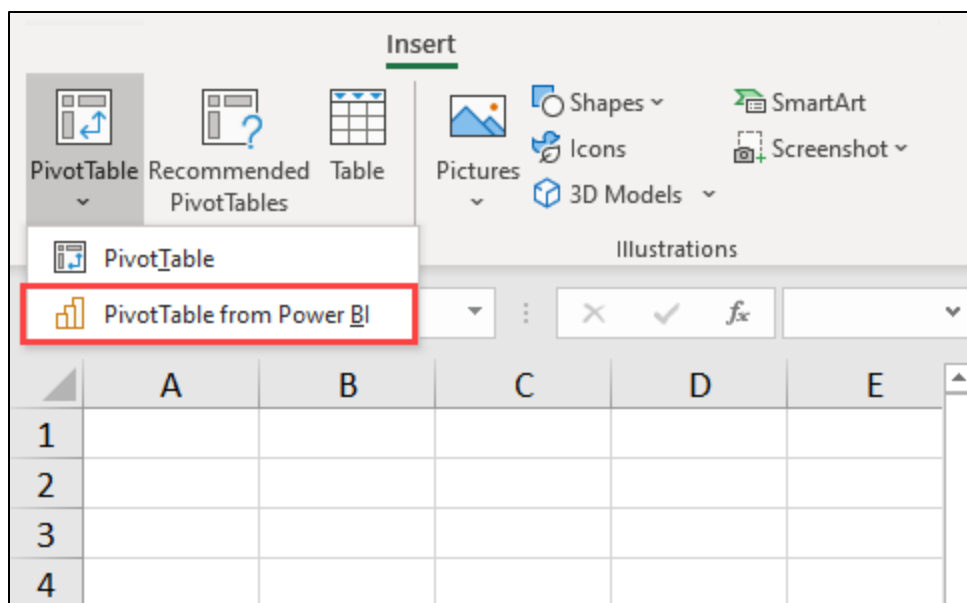


Figure 5 - Creating a PivotTable from a Power BI Dataset

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 space or comma, between each field that it joins together. **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 together data from other rows.

	A	B	C	D	E	F	G
1	Address 1	Address 2	City	State	Zip		
2	101 Peachtree St.	Apt. 2714	Atlanta	GA	30303	101 Peachtree St.,Apt. 2714,Atlanta,GA,30303	
3	2147 Palm Drive		Bakersfield	CA	93301	2147 Palm Drive,Bakersfield,CA,93301	
4	1999 Wacker Dr	Suite 846	Chicago	IL	60290	1999 Wacker Dr,Suite 846,Chicago,IL,60290	
5	2000 Lincoln Parkway		Dallas	TX	75201	2000 Lincoln Parkway,Dallas,TX,75201	
6	2219 Byrum Lane	Suite 1	Eugene	OR	97401	2219 Byrum Lane,Suite 1,Eugene,OR,97401	
7	8744 Beach Blvd		Ft Lauderdale	FL	33301	8744 Beach Blvd,Ft Lauderdale,FL,33301	
8							

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 prove beneficial when some addresses contain apartment or suite numbers, and some 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 simply 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.

	A	B	C	D	E	F
1	Address 1	City	State	Zip		
2	101 Peachtree St.	Atlanta	GA	30303		101 Peachtree St., Atlanta, GA, 30303
3	2147 Palm Drive	Bakersfield	CA	93301		2147 Palm Drive, Bakersfield, CA, 93301
4	1999 Wacker Dr	Chicago	IL	60290		1999 Wacker Dr, Chicago, IL, 60290
5	2000 Lincoln Parkway	Dallas	TX	75201		2000 Lincoln Parkway, Dallas, TX, 75201
6	2219 Byrum Lane	Eugene	OR	97401		2219 Byrum Lane, Eugene, OR, 97401

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 to see 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*, and so forth. 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 that 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,

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

criteria 1, 2, etc. is 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.

	A	B	C	D	E
1	Salesperson	Transaction Amount			
2	Smith	2,576.53			
3	Munoz	3,040.69			
4	Harris	3,080.61			
5	Yee	1,751.59			
24	Burkholtz	2,160.14			
25	Jackson	1,966.50			

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

A MAXIFS function can evaluate up to 127 conditions. In **Figure 11**, MAXIFS is calculating the result by examining two criteria: *Salesperson* and *Month*.

	A	B	C	D	E	F
1	Salesperson	Month	Transaction Amount			
2	Smith	January	2,576.53			
3	Munoz	January	3,040.69			
4	Harris	January	3,080.61			
5	Yee	January	1,751.59			
287	Yee	December	3,449.39			
288	Burkholtz	December	3,511.37			
289	Jackson	December	3,539.27			

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

MINIFS

The **MINIFS** function is the opposite of MAXIFS – you can use MINIFS 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. 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 from the top down, and the lookup column or row no longer needs to be sorted for approximate matches. 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. This feature allows XLOOKUP to look to the lookup column's left – something VLOOKUP cannot do.

The syntax of XLOOKUP is:

**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; 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 next larger item
2	Wildcard character match

Search mode (optional) determines the search order, 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 be used to 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 for 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	Pric	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					
L001	N0019B					
L002	N00116B					
L003	N00124B					
L201	708901CF					
L202	N0029B					
L203	N00216B					
M101	N0039B					
M102	FA550009					
M201	FA550016					
M202	FA550024					

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 then is used as 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.

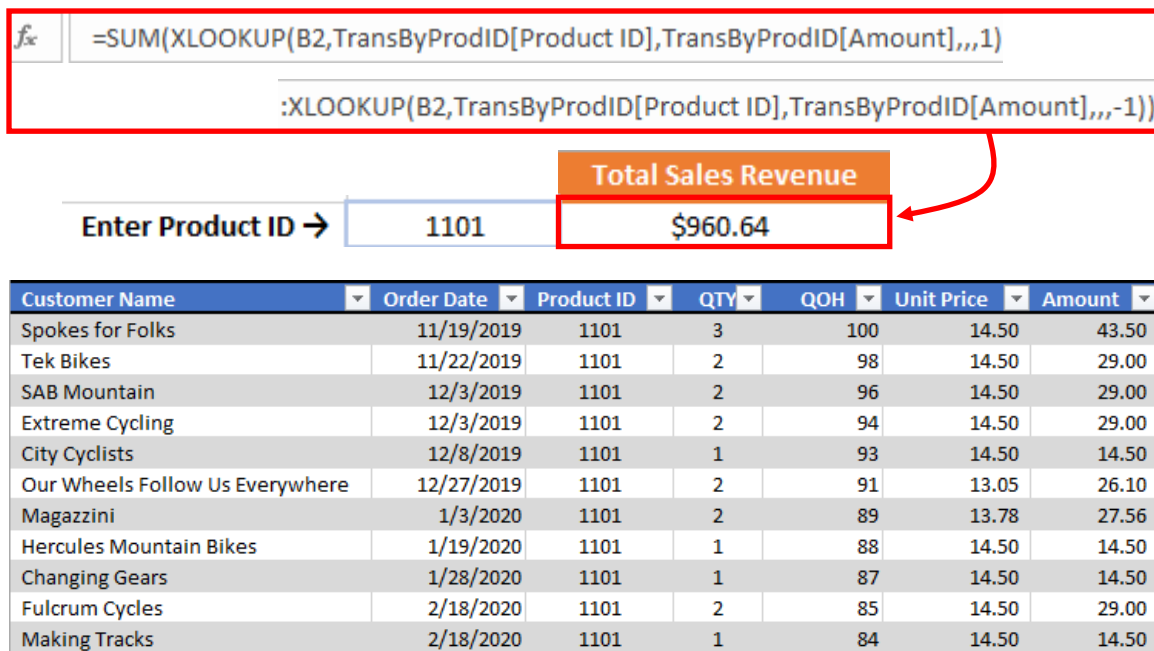


Figure 13 – Using XLOOKUP to Sum Sales of a Specified Product ID

XLOOKUP is here, and it can do everything VLOOKUP can do and more. XLOOKUP and its companion **XMATCH**, with similar functionality advances, will revolutionize how you use lookup functions. They are easy to use, intuitive, and packed with 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 also 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 in 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 workbook's author and the external accountant could then review and edit the workbook together and in real-time.

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

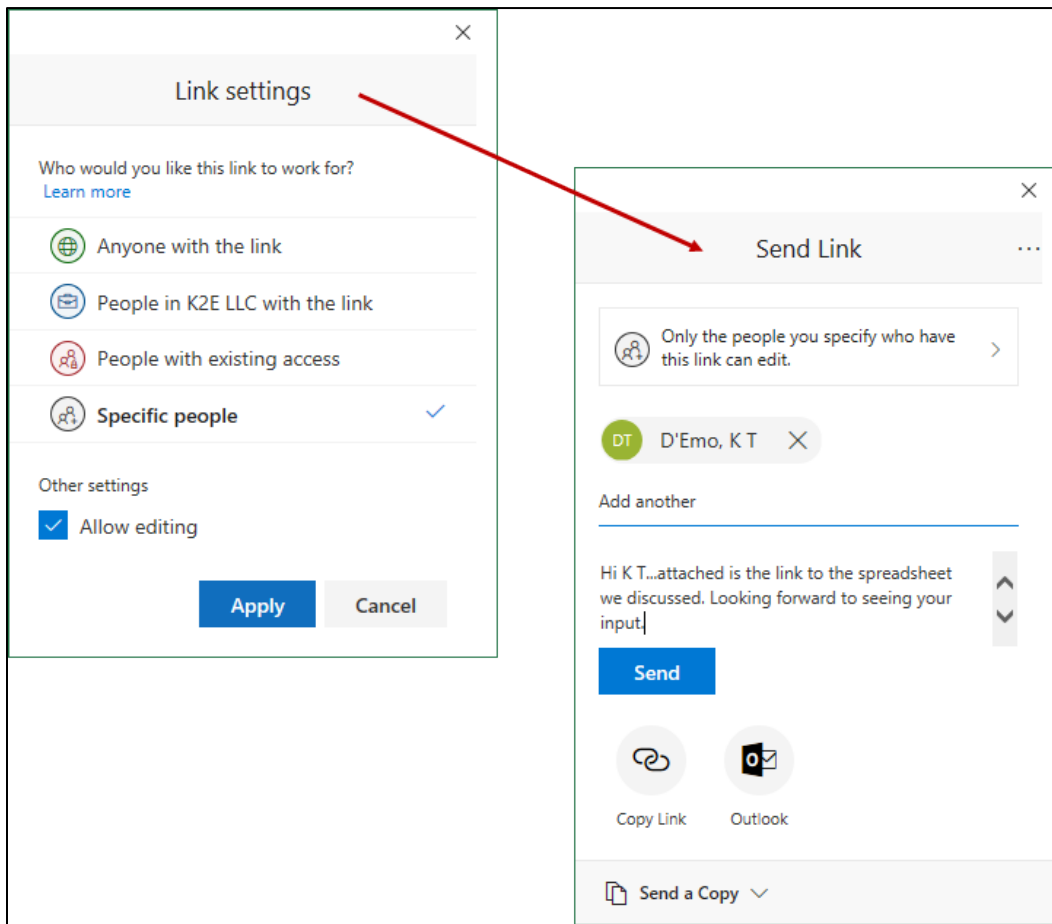


Figure 14 - Setting Up an Excel Workbook for Co-Authoring

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 even need to have Excel installed on their device 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 at the same time.” 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, keep in mind that if you store the workbook in OneDrive, OneDrive for Business, or SharePoint Online, you can access previous versions of the workbook 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 some 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 and SharePoint Online is Cloud-based. Functionally, the two platforms are otherwise nearly identical. Excel can use either SharePoint platform as a means of facilitating collaboration using the co-authoring experience discussed previously. To take advantage of this capability, begin by saving 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 documents and edit documents. If necessary, these permissions can be quite granular, 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 after-thought, but 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 also have the same workbook open, all users will be able to edit the workbook and see each others' 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.

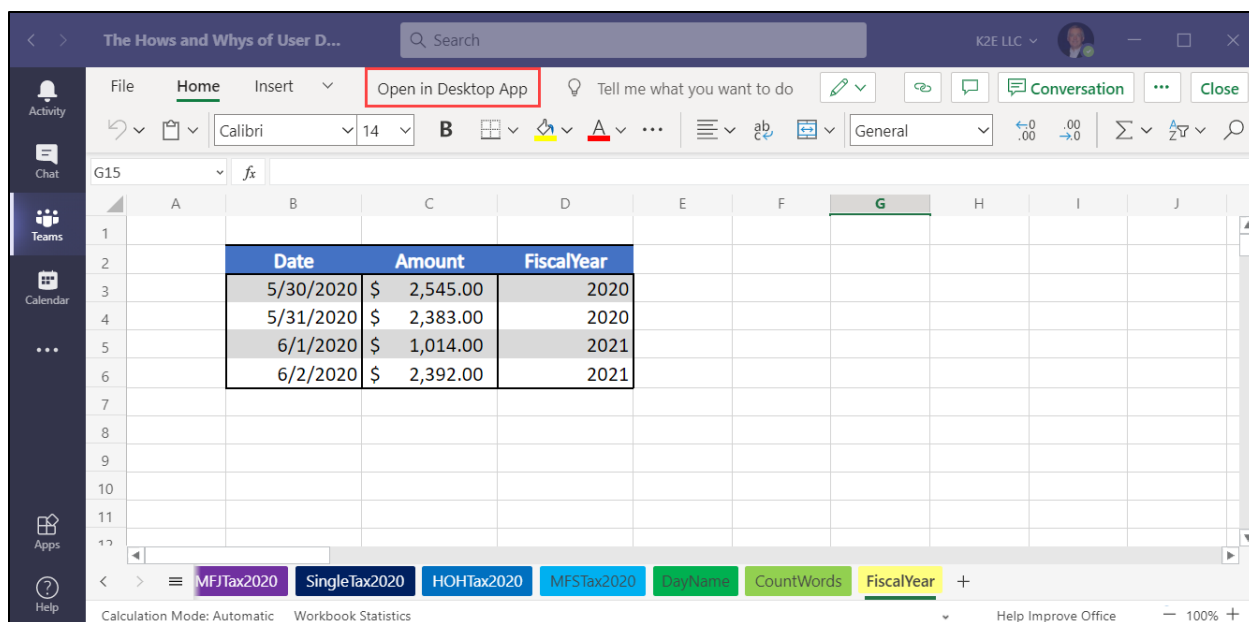


Figure 15 - Opening a Workbook in Teams Using the Excel Web App

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

If you use Excel version 1907 or newer through a Microsoft 365/Office 365 subscription, you can access even more powerful features related to arrays. More 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 simply press the **Enter** key upon entering an array formula and not worry about the **CTRL + SHIFT + ENTER** keyboard sequence.

Loosely speaking, 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, a task that 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.

	A	B	C	D	E	F	G
1							
2		Apples		Apples			
3		Bananas		Bananas			
4		Cherries		Cherries			
5		Apples		Dates			
6		Dates		Elderberries			
7		Elderberries					
8		Apples					
9		Apples					
10		Bananas					
11		Cherries					
12							

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 you can use in a dynamic array aware version 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. Notably, neither **FILTER** nor any of the other five dynamic array formulas alter the source data at all. Instead, each formula displays its results as a separate output range.
2. You can use **SORT** to arrange data in a dynamic array in ascending or descending order, again 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 needs to 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. To achieve the objective, she enters the following simple formula into the worksheet.

=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.

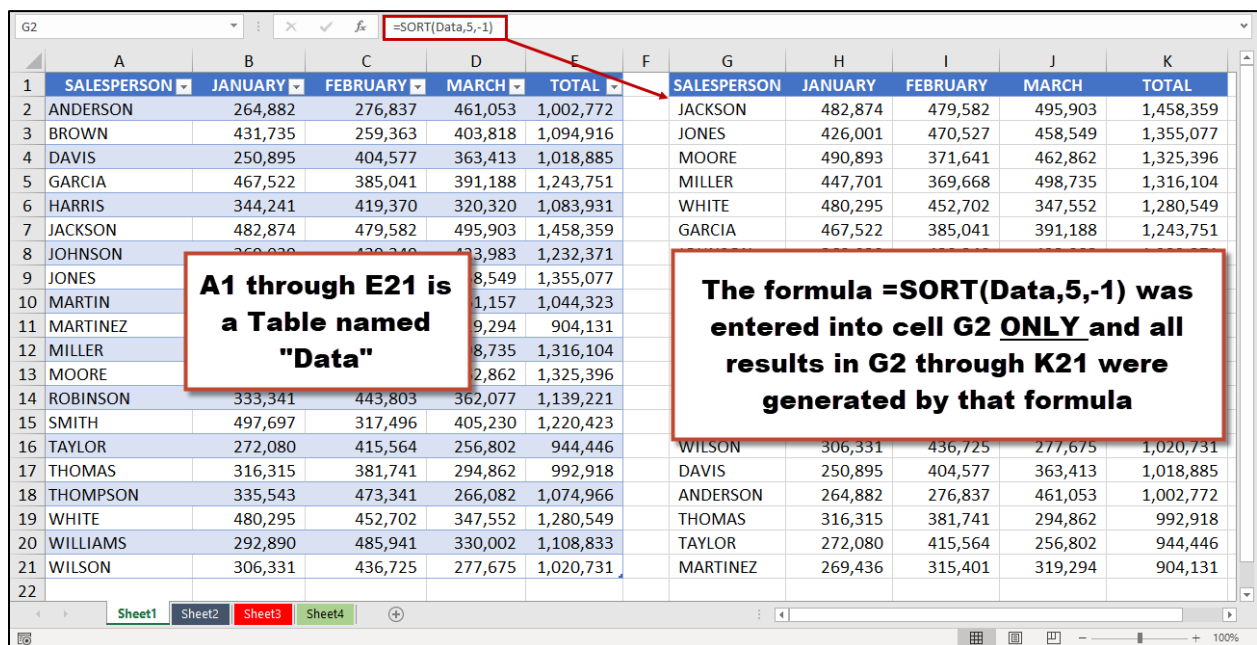


Figure 17 – Sorting a Dynamic Array with the SORT Function

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, we expect their popularity to skyrocket. Moreover, suppose you are using a dynamic array aware version of Excel. In that case, you can work with legacy array formulas without worrying about the **CTRL + SHIFT + ENTER** keystroke sequence to enter your array formulas.

Power Query’s Continued Evolution

Excel’s Power Query continues to evolve, and Microsoft adds new features to this tool in subscription-based licenses of Excel seemingly every month. One of the more compelling options that became available beginning with Current Channel’s 2007 version is the ability to extract information from PDF documents directly into Excel.

Consider the report pictured in **Figure 18**. It is a PDF document that contains operational data from a small chain of restaurants. For presentation purposes, we have hidden six months of data from the display.

Month Ending Date	Restaurant Number	Sales	Payroll Expense
1/31/2022	101	1,172,329	215,709
1/31/2022	102	960,554	157,531
1/31/2022	103	1,169,956	180,173
1/31/2022	104	989,238	212,686
2/28/2022	101	1,170,269	227,032
2/28/2022	102	973,438	205,395
2/28/2022	103	1,116,058	179,685
2/28/2022	104	1,051,097	230,190
3/31/2022	101	1,191,158	200,115
3/31/2022	102	950,959	165,467
3/31/2022	103	1,180,638	204,250
3/31/2022	104	1,189,884	188,002
10/31/2022	101	1,071,390	162,851
10/31/2022	102	1,095,797	168,753
10/31/2022	103	938,998	205,641
10/31/2022	104	1,105,036	212,167

Figure 18 - Sample Sales Report for Importing into Excel Using Power Query

Suppose you need the data from the import available to you in an Excel workbook so that you can analyze operations and results in Excel. You certainly do not want to spend valuable time manually entering 160 cells into a workbook! Fortunately, you don’t have to because Power Query can extract the data from the PDF document and place it into the spreadsheet for you.

In Excel, to take advantage of this feature, click **Data** from the Ribbon’s **Home** tab, followed by **Get Data, From File, and From PDF**. In the dialog box that follows, select the PDF document that contains the data you need to extract and then click **Import** to open Power Query’s **Navigator** shown in **Figure 19**.

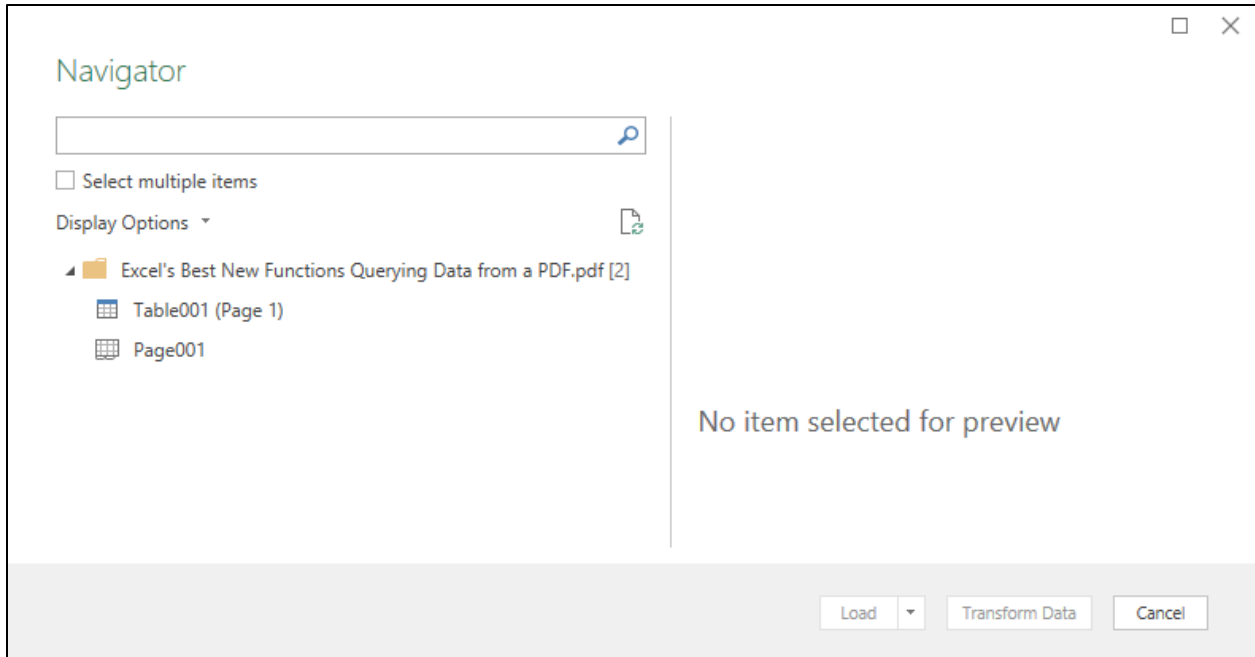


Figure 19 - Opening Power Query's Navigator

Next, choose the table or range you wish to import and select the **Load To...** option, as shown in **Figure 20**.

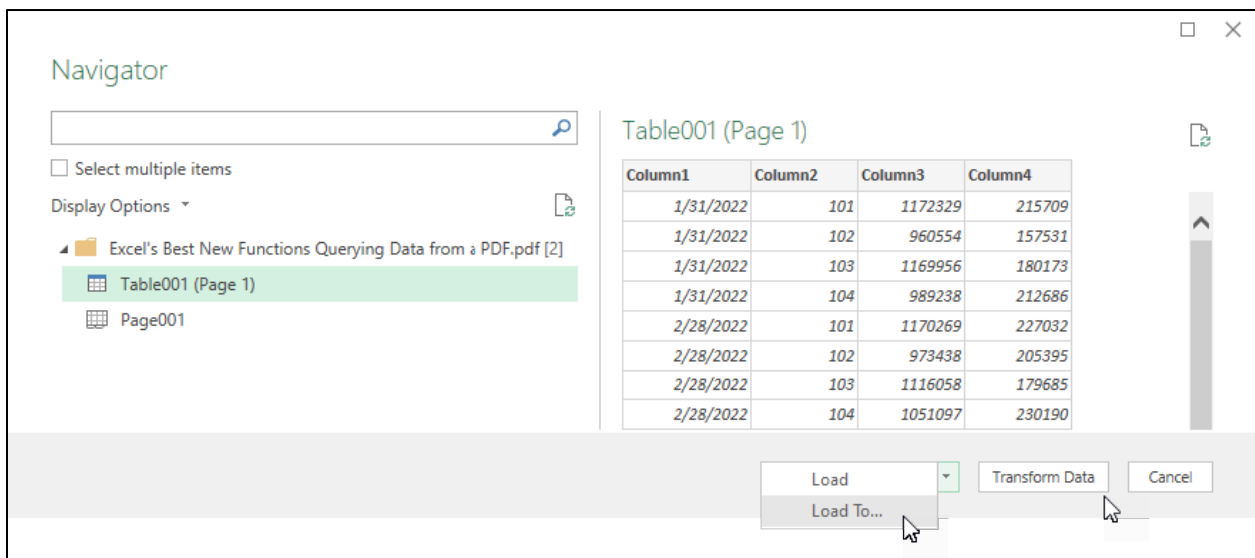


Figure 20 - Previewing the Data to Load into Excel

In the **Import Data** dialog box pictured in **Figure 21**, choose to import the data into a **Table** and place the table in a **New worksheet**.

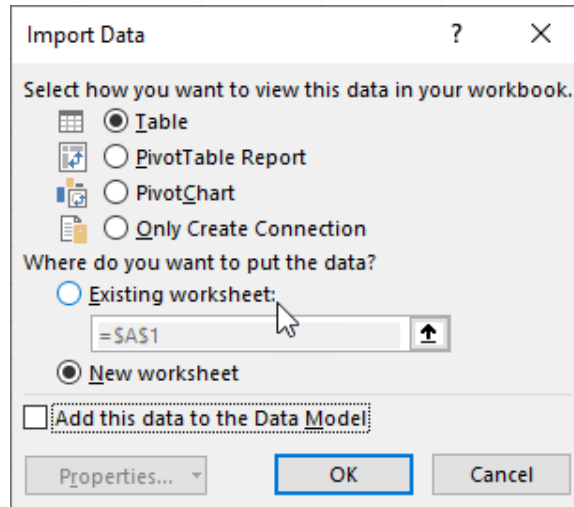


Figure 21 - Excel's Import Data Dialog Box

Figure 22 presents the results of the query. Power Query's ability to extract and import data into Excel from PDF documents should benefit almost all business professionals by reducing the amount of time spent on unnecessary data entry.

	A	B	C	D
1	Column1	Column2	Column3	Column4
2	1/31/2022	101	1172329	215709
3	1/31/2022	102	960554	157531
4	1/31/2022	103	1169956	180173
5	1/31/2022	104	989238	212686
6	2/28/2022	101	1170269	227032
7	2/28/2022	102	973438	205395
8	2/28/2022	103	1116058	179685
9	2/28/2022	104	1051097	230190
10	3/31/2022	101	1191158	200115

Figure 22 - Results of Querying a PDF's Data into Excel Using Power Query

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.