Introduction to Excel Dynamic Arrays

By Neale Blackwood

www.a4accounting.com.au

www.excelyourself.com.au

a4@iinet.net.au

A4 ACCOUNTING

EXCEL CONSULTING, TRAINING AND WEBINARS

Introduction

Subscription version

The subscription version of Excel is now regularly updated with new features and functions. In 2020 a game changing update was made. Dynamic arrays were introduced which allow you to work with a range in the same way you used to work with a cell.

This update has changed how you can build spreadsheets and enables many new possibilities for solutions that previously were complex, difficult, or impossible in Excel.

Version Issues

Excel has always had some version issues when new features and functions are released, and they can't be used in previous versions. That divide has expanded with dynamic arrays.

Old vs New

I will refer to the pre-dynamic arrays as old Excel or "in the old days" and the subscription version as new Excel.

New terminology, symbols, and errors

Dynamic arrays use new terminology. There is also a new formula symbol and a new use of an existing formula symbol. Plus a few new errors have been added.

A whole new way of creating formulas describes dynamic arrays.

This is the biggest change to the way Excel calculates since it was released.

Existing Functions

Most existing functions will work with dynamic arrays.

New Functions

New functions have been added to Excel since 2020 and more are added regularly. This session will cover some of the new functions.

This session

This session is designed to get you started using dynamic arrays.

Dynamic Arrays

The dynamic array changes have far-reaching implications in how you create formulas in Excel. Let's start with a simple example.

In the image on the right, we have a list of 5 consecutive numbers. This is in the DA sheet.

We want to link to cells A1:A5 in the range B1 to B5. In the old days there were at least two options.

	A	B
1	1	
2	2	
3	3	
4	4	
5	5	

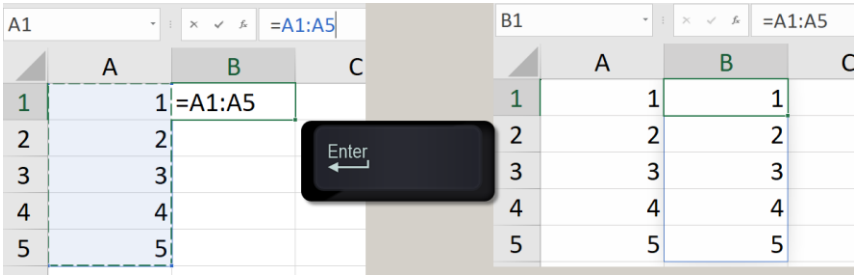
1. Create a formula in cell B1 =A1 and copy it down.
2. Select the range B1:B5 enter the formula =A1 and press Ctrl + Enter.

Both these techniques enter a formula in each cell in the range B1:B5.

The New Way

In cell B1 type = and then use the mouse to select the range A1:A5 and press Enter.

The formula entered in cell B1 is.



=A1:A5

You could have entered that formula in old Excel in a single formula and it would have returned 1, but it would only be in cell B1. If you selected B1:B5 and used Ctrl + Enter it would have entered a formula in each cell in the range. And the formula would have worked correctly.

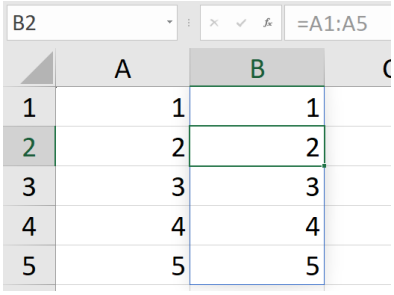
In the above image the formula is **only** in cell B1. The formula has ***Spilled*** down. This is a new term in Excel. Dynamic arrays can spill down and spill to the right and sometimes spill in both directions.

New Terms – Spill and Spill Range

When formulas spill, there is a single formula in the top, left cell of the range and there are values displayed in the other cells in the range. That range is called the ***Spill Range.*** In the above image the formula is in cell B1, and the spill range is B1:B5. The spill range has a thin blue line around the range whenever you select a cell in the range. There are no formulas in the range B2:B5.

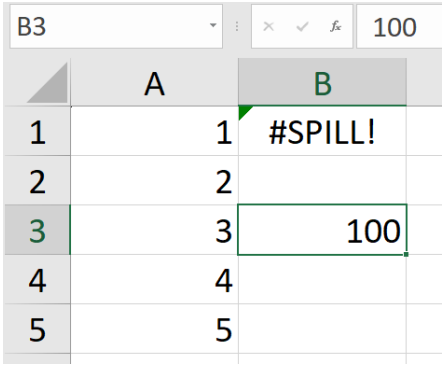
If you select a cell in the spill range the formula in the formula bar is greyed out – see image on right cell B2.

If you enter a value in a cell beneath the top, left cell where the formula is, you will see a new error.



The #SPILL! error

I have entered 100 in cell B3. This caused the new **#SPILL!** error in the formula cell B1.

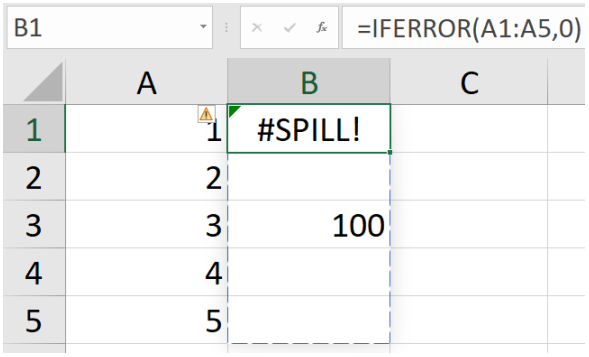


The #SPILL! error is generated whenever you have an entry in a cell that will be part of the spill range.

Remember the spill range can go down or across or in both directions to create a two-dimensional range.

This means you need to be careful when designing your spreadsheets and allow enough room if your ranges could expand. We will see examples of expanding ranges later in the session.

Note: The IFERROR function does not handle the #SPILL! error. See image on right.



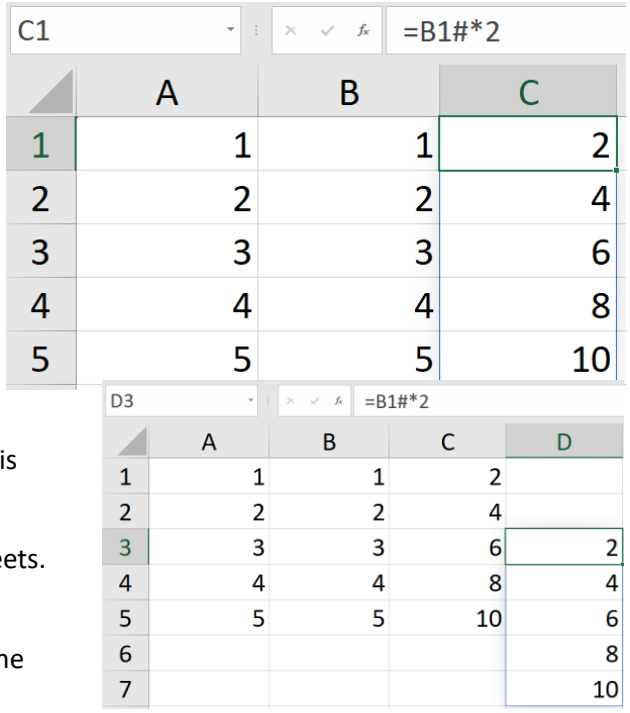
To remove the error, you must remove the entry in cell B3.

Note the dashed thin blue line around the spill range when there is an error. This shows you where the spill area is, and it needs to be empty.

Once you have a spill range, you can refer to it using a new formula symbol.

The new # (hash) symbol

Let's say we want to double the values in our spill range. In cell C1 we can enter the formula.



=B1#*2

The # symbol follows the cell reference.

Always refer to the top, left cell of the spill range.

Try entering that same formula in cell D3.

That formula will work the same throughout this sheet.

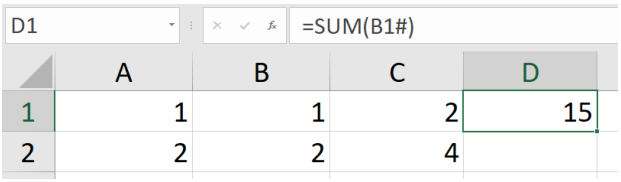
You can also refer to spill ranges from other sheets.

If you select the spill range with the mouse or keyboard, Excel will enter the B1# notation in the formula.

Summing spill ranges

You can also use the # notation to refer to a spill range within a function.

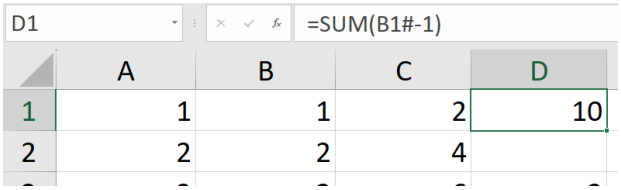
In cell D1 enter the following formula to add up all the cells in our original spill range.



=SUM(B1#)

The dynamic array notation is versatile.

We want to deduct 1 from each cell in the spill range and add that adjusted value up. That should be 5 less than the above calculation as there are 5 cells in the range. The following formula will do just that.



=SUM(B1#-1)

I keyed in the -1, but we could have used a cell reference to handle the adjustment value.

With two spill ranges in columns B and C, here are some other useful SUM functions.

Adds up both ranges – returns 45.

=SUM(B1#,C1#)

Add up both ranges and also returns 45, If a column was inserted between B and C the values would be included in rows 1 through 5.

=SUM(B1#:C1#)

Divides column B by C and returns 2.5. Each row equates to 0.5 hence 5 x 0.5 = 2.5.

=SUM(B1#/C1#)

Multiplies the two spill ranges together – returns 110.

=SUM(B1#*C1#)

Another new formula symbol

In old Excel you could enter =A1:A5 in cell B1 and press Enter, and it would enter a single formula. To replicate that in the new Excel you use.

=@A1:A5

The @ symbol instructs Excel to treat the formula as an old version. You may see this symbol added to your existing formulas when you open them in the new Excel. The @ goes at the front of the reference or function.

New Dynamic Array Functions

This is the most exciting part of dynamic arrays. As we saw, existing functions can work with dynamic arrays. There are new functions that offer new capabilities with many opportunities to automate tasks that previously were manual.

These new functions offer the ability to create ranges that expand, and contract based on their ranges and in some cases the conditions you can set within the functions.

Optional arguments

As we examine these new functions, we will cover each function's syntax. Any argument (the part of a function, between the brackets and separated by commas) that is in square brackets is optional and can be omitted. In the UNIQUE function below there are three arguments and the last two are optional. In general, all optional arguments are listed last in the argument sequence.

In these new functions many of the optional arguments can be omitted as their default setting is the most common usage.

array – in many functions that follow the first argument is **array**. This means a range reference eg A1:A10 or A1:B10. Fewer $ signs are required when referring to cells and ranges in dynamic array formulas.

UNIQUE function

It is worth noting that there are two types of unique that can be returned by this function. This function allows you to create automated drop-down lists and automated reports. Combined with the other dynamic array functions you can create some amazing and dynamic reports.

This function is useful for creating dynamic drop-down lists based on existing data sets.

Syntax

UNIQUE(array,[by-col],[exactly_once])

array – the range to extract the unique entries from. It is typically a vertical list. It could be more than one column. It could also be a horizontal list with one or more rows. Horizontal ranges require a 1 in the second argument.

by_col – **optional** – enter a 1 or TRUE to specify a horizontal range. If omitted the default is FALSE meaning a vertical range.

exactly_once – **optional** – enter a 1 or TRUE to specify you want a list of the entries that appear only once in the list. If omitted the default is FALSE which removes duplicates.

Example:

=UNIQUE(A2:A8)

Removing duplicates

In the Unique sheet the formula in cell C2 on the right has the standard meaning of unique. A list of all the entries in a list without any duplicates.

The sequence of the entries in the original list is maintained in the list returned by UNIQUE.

	A	B	C	D	E	F
1	**Codes**		Unique 1	Unique 2	Formula in C2	Formula in D2
2	ABC123		ABC123	ABC321	=UNIQUE(A2:A8)	=UNIQUE(A2:A8,,1)
3	XYZ321		XYZ321			
4	BCD231		BCD231			
5	ABC123		ABC321			
6	XYZ321					
7	BCD231					
8	ABC321					

We will examine the new SORT function later which enables you to sort lists.

Only once

Cell D2 shows an example of using the exactly_once argument. The last code in the list is not duplicated. I can't think of many times you need a list of items that only a appear once. Maybe if you were expecting duplicate codes and needed a list of items that weren't duplicated.

Changing entries in the list in column A automatically updates the result of the UNIQUE function.

Multiple columns

If you include a multiple column reference in the UNIQUE function it checks for uniqueness across all columns – see example on right that is in the Unique2 sheet.

Because we selected two columns the results spills to the right, and down.

D2 =UNIQUE(A2:B7)

	A	B	C	D	E	F
1	**First**	**Last**		UNIQUE		
2	John	Jones		John	Jones	
3	Sue	Smith		Sue	Smith	
4	John	Jones		Sue	Ng	
5	Sue	Ng		George	Ng	
6	George	Ng				
7	Sue	Smith				

Distinct Count

A useful calculation that the UNIQUE function simplifies is a distinct count.

This is the number of unique items that appear in a list.

In cell E2 we have the following formula.

E2 =COUNTA(UNIQUE(A2:A9))

	A	B	C	D	E
1	**Departments**		Unique		Distinct Count
2	Admin		Admin		5
3	Sales		Sales		
4	Marketing		Marketing		
5	Production		Production		
6	Admin		Human Resources		
7	Production				
8	Sales				
9	Human Resources				

```
=COUNTA(UNIQUE(A2:A9))
```

In this formula the COUNTA function counts all the entries returned by the UNIQUE function.

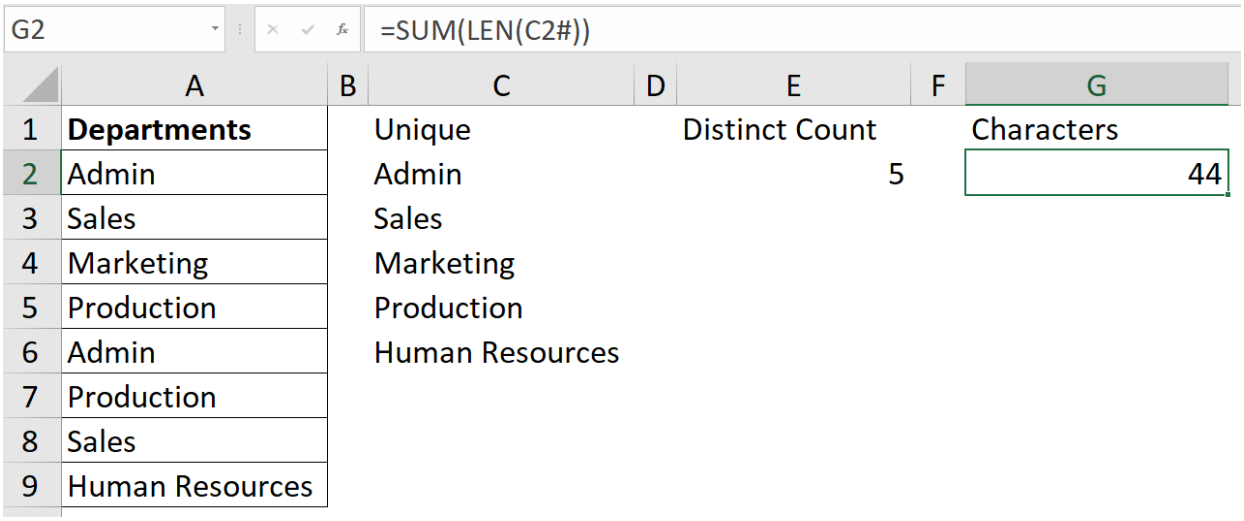
Note: the COUNT function only counts numbers and dates. COUNTA counts all entries.

Automagical

With dynamic arrays, existing functions can be combined to perform seemingly magical new calculations.

In the Unique3 sheet let's say we wanted to count how many characters were in the spill range. The formula in cell G2 below will do that for you.

=SUM(LEN(C2#))



	A	C	E	G
1	**Departments**	Unique	Distinct Count	Characters
2	Admin	Admin	5	44
3	Sales	Sales		
4	Marketing	Marketing		
5	Production	Production		
6	Admin	Human Resources		
7	Production			
8	Sales			
9	Human Resources			

The LEN function normally counts the number of characters in a single cell.

When you refer to the spill C2# range the LEN function calculates the lengths for each of the individual cells in the spill range. Then the SUM function adds them all up, returning the total number of characters in the spill range.

Warning - zeroes and empty cells

Be warned that the UNIQUE function doesn't handle ranges with both blank cells and zeroes very well.

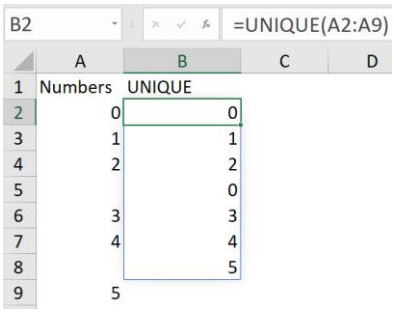
In the Unique4 sheet the list on the left has a zero and two blank cells.

Note that 0 is shown twice in the UNIQUE results.

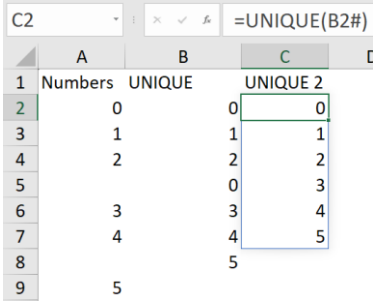
Unfortunately, UNIQUE treats a blank cell like a zero, **BUT** if shows it separately to the zero.

The workaround is to use a separate UNIQUE.

If you want to show a blank cell as well as a zero – it is a little more complicated and not covered in this session.



	A	B
1	Numbers	UNIQUE
2	0	0
3	1	1
4	2	2
5		0
6	3	3
7	4	4
8		5
9	5	



	A	B	C
1	Numbers	UNIQUE	UNIQUE 2
2	0	0	0
3	1	1	1
4	2	2	2
5		0	3
6	3	3	4
7	4	4	5
8		5	
9	5		

FILTER function

This function is a major game changer. In the past applying filters was always a manual process. This function changes filtering into a formula-based, automatic process.

Syntax

=FILTER(array,include,[if_empty])

array – the range to filter. This can be a single or multiple column range. It could be another function that returns an array (range).

include – the condition that determines what to display. The condition must return TRUE or a non-zero number for the row(s) to display. The condition doesn't have to be within the array range. You can have more than one condition.

i**f-empty** – **optional** - if the filter returns no rows, then you can specify a text string or a number to display. If you want to return text, it needs to be enclosed in quotation marks.

Example:

=FILTER(A2:C9,B2:B9=E2)

In the example on the right the formula is in cell G2, and it spills across three columns and down two rows.

The condition is column B equals Orange.

It is not case sensitive. Changing cell E2 updates the FILTER, see image below.

The dynamic nature of the filtering makes this especially useful for reports and dashboards where the user can change cell selections and automatically update the report.

G2 =FILTER(A2:C9,B2:B9=E2)

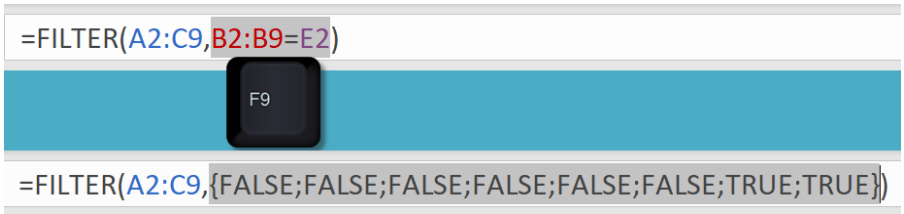
	A	B	C	D	E	F	G	H	I
1	**Product**	**Colour**	**Quantity**		**Filter by**		**Product**	**Colour**	**Quantity**
2	Gadget	Red	1,718		Orange		Gadget	Orange	1466
3	Widget	Red	1,032				Widget	Orange	1654
4	Gadget	Blue	1,438						
5	Widget	Blue	1,458						
6	Gadget	White	1,870						
7	Widget	White	1,094						
8	Gadget	Orange	1,466						
9	Widget	Orange	1,654						

G2 =FILTER(A2:C9,B2:B9=E2)

	A	B	C	D	E	F	G	H	I
1	**Product**	**Colour**	**Quantity**		**Filter by**		**Product**	**Colour**	**Quantity**
2	Gadget	Red	1,718		Blue		Gadget	Blue	1438
3	Widget	Red	1,032				Widget	Blue	1458

How this works

To see how this works see the image below.



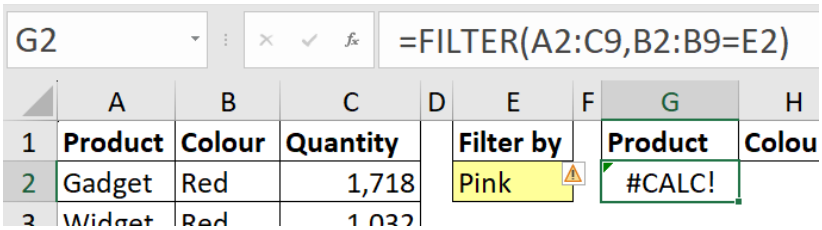
If we select part of a formula in the Formula Bar and press F9 function key, Excel will show the result of that part. You must select a self-contained section that can be calculated by itself.

As you can see there is a list of TRUE and FALSE results based on the colour being orange in column

B. When a TRUE is encountered that row is displayed. In this case, the last two rows.

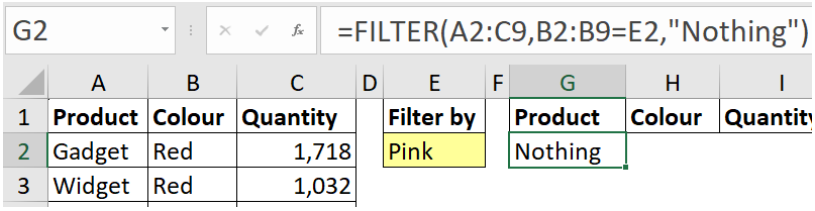
When the FILTER returns nothing

If the entry in cell E2 isn't in the table, then the new #CALC! error is returned. The optional third argument allows you to handle that.



=FILTER(A2:C9,B2:B9=E2,"Nothing")

The word Nothing is displayed if the filter returns nothing. If you are returning text, always enclose the text within quotation marks.



More than one condition

If you want to apply more than one criteria for your filter you have two options.

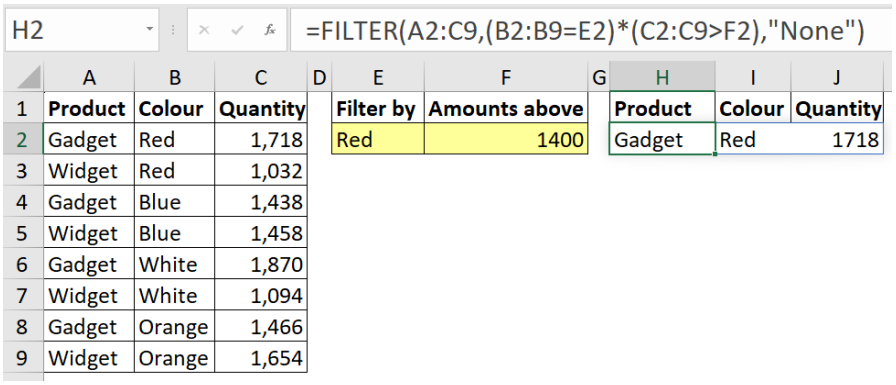
AND Comparison

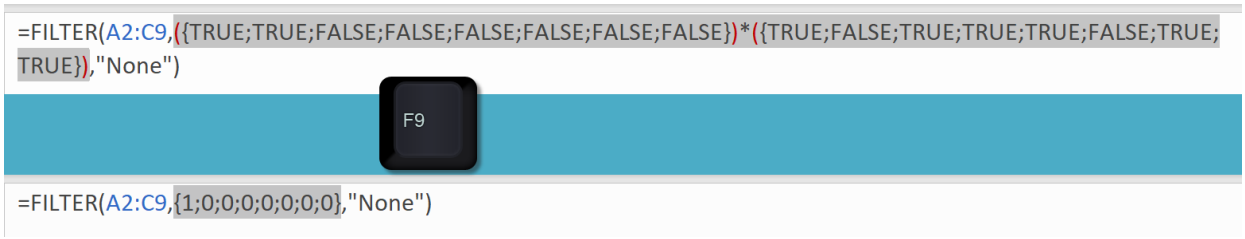
This means you want one condition and another condition to apply. Both conditions must be met for the row to display. In the image below we want the red colour and a quantity above 1,400. We need to enclose each separate condition in brackets and place the multiplication symbol between them - see below.

=FILTER(A2:C9,(B2:B9=E2)*(C2:C9>F2),"None")

Row 2 is the only row that meets both criteria. The way it calculates is shown on the following page.

Note: TRUE = 1 and FALSE = 0. When you multiply by FALSE it zeroes the value.



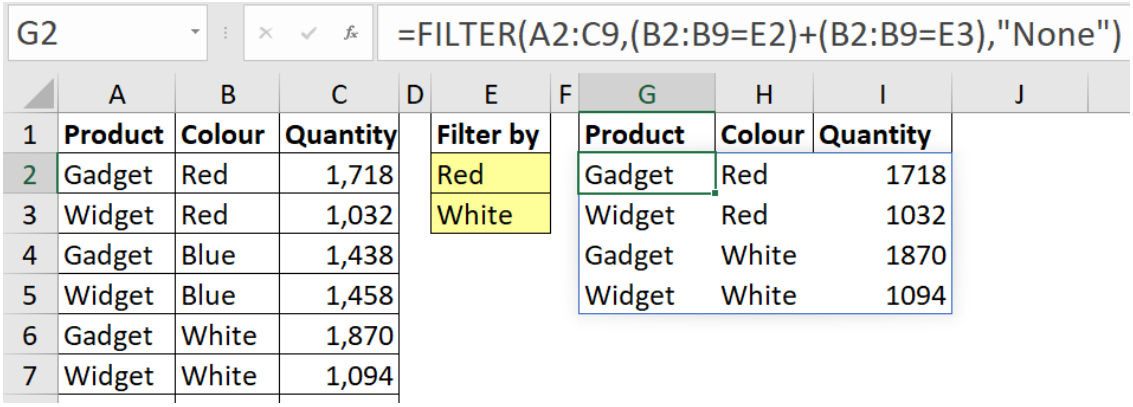


In the above example the TRUEs and FALSEs are converted to 1's and 0's because we multiply the tow condition results together.

1 is treated as TRUE and 0 is treated as FALSE to work out which row(s) to display.

OR Comparison

This means you want one condition or another condition to be met. Either condition can be met for the row to display. Instead of using the multiplication symbol between the conditions you use the plus sign. See the example below.



	A	B	C	D	E	F	G	H	I	J
1	**Product**	**Colour**	**Quantity**		**Filter by**		**Product**	**Colour**	**Quantity**	
2	Gadget	Red	1,718		Red		Gadget	Red	1718	
3	Widget	Red	1,032		White		Widget	Red	1032	
4	Gadget	Blue	1,438				Gadget	White	1870	
5	Widget	Blue	1,458				Widget	White	1094	
6	Gadget	White	1,870							
7	Widget	White	1,094							

The formula is.

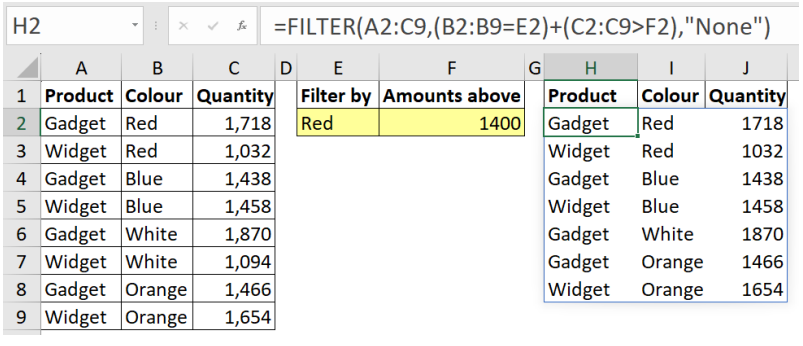
=FILTER(A2:C9,(B2:B9=E2)+(B2:B9=E3),"None")

The example above has both conditions relating to the same column. This means they are mutually exclusive. If one condition is met, it means the other condition can't me met. The colour cannot be both red and white. The result of adding the TRUE's (1) and FALSE's (0) together cannot be more than 1.

OR Conditions across columns

You may have conditions that relate to separate columns. From a previous example (FILTER_2 sheet) we might want red or a quantity above 1,400.

This is interesting as the result using + could create a result of more than 1. A row could be both red and above 1,400, as in row 2. Let's see how this works. The example below uses the + sign between the two conditions.



	A	B	C	D	E	F	G	H	I	J
1	**Product**	**Colour**	**Quantity**		**Filter by**	**Amounts above**		**Product**	**Colour**	**Quantity**
2	Gadget	Red	1,718		Red	1400		Gadget	Red	1718
3	Widget	Red	1,032					Widget	Red	1032
4	Gadget	Blue	1,438					Gadget	Blue	1438
5	Widget	Blue	1,458					Widget	Blue	1458
6	Gadget	White	1,870					Gadget	White	1870
7	Widget	White	1,094					Gadget	Orange	1466
8	Gadget	Orange	1,466					Widget	Orange	1654
9	Widget	Orange	1,654							

The formula in cell H2 is.

```
=FILTER(A2:C9,(B2:B9=E2)+(C2:C9>F2),"None")
```

Let's examine the formula's criteria using the F9 function key.

```
=FILTER(A2:C9,{2;1;1;1;1;0;1;1},"None")
```

The first result returns 2, this is row 2. It is both Red and above 1,400.

The FILTER function will display any number above zero that the added criteria returns. Numbers will always be positive and always be a whole number.

The FILTER function will hide/ignore the rows that add to zero as this means no conditions have been met. Row 7 (White and 1094) is the only row that meets neither condition, and as such is omitted from the returned range.

Show all option

Sometimes you may want to show all the entries in the filter. We can modify the FILTER_1 sheet formulas to achieve that.

Here are two formulas to achieve that.

The first one displays all entries if cell E2 is blank.

The second displays all entries if cell E2 contains the * symbol. The * symbol is a universal wildcard character representing all entries.

```
=FILTER(A2:C9,(E2="")+(B2:B9=E2),"Nothing")
```

Filter by	Product	Colour	Quantity
	Gadget	Red	1718
	Widget	Red	1032
	Gadget	Blue	1438
	Widget	Blue	1458
	Gadget	White	1870
	Widget	White	1094
	Gadget	Orange	1466
	Widget	Orange	1654

```
=FILTER(A2:C9,(E2="*")+(B2:B9=E2),"Nothing")
```

Filter by	Product	Colour	Quantity
*	Gadget	Red	1718
	Widget	Red	1032
	Gadget	Blue	1438
	Widget	Blue	1458
	Gadget	White	1870
	Widget	White	1094
	Gadget	Orange	1466
	Widget	Orange	1654

SORT function

In the same way that we can now FILTER via a function, Excel has also added a SORT function. Plus, you can use them together. The SORT function is used when the column to be sorted by is also to be displayed. If you want to sort by a column, but not display the column, use the SORTBY function, which will be covered next. The examples are in the SORT + SORTBY sheet.

Syntax

SORT(array,[sort_index],[sort_order],[by_col])

array – the range to sort. It can be a single or a multiple column range. This can also be another function that returns a range like FILTER or UNIQUE.

sort_index – optional – this is the column number within the array to use to sort the range by. If omitted the default is 1, the first column is used to sort the array.

sort_order - optional – 1 is ascending and -1 for descending. If omitted the default is 1, ascending.

by_col - optional – FALSE sets the direction as a vertical sort by rows. TRUE sets a horizontal sort by columns. If omitted FALSE is used. Most sorts are on vertical ranges so this argument can be omitted.

The example on the right is sorting the table by the (3) third column (Salary), in descending order using -1 as the third argument.

The last argument is omitted as it defaults to a vertical table. The formula in cell E2 is.

E2 =SORT(A2:C7,3,-1)

	A	B	C	D	E	F	G
1	First Name	Last Name	Salary		First Name	Last Name	Salary
2	Sally	de Silva	86,050		Keith	Ng	86599
3	George	Smith	85,402		Sally	va	86050
4	Pamela	Jones	79,573		George	Smith	85402
5	Keith	Ng	86,599		Kelly	Chen	84948
6	Kelly	Chen	84,948		Pamela	Jones	79573
7	Sue	Tan	76,847		Sue	Tan	76847

=SORT(A2:C7,3,-1)

If you omit all the optional arguments it will sort by the first column in the range, in ascending order. See image on right.

The formula in cell E2 is.

=SORT(A2:C7)

E2 =SORT(A2:C7)

	A	B	C	D	E	F	G
1	First Name	Last Name	Salary		First Name	Last Name	Salary
2	Sally	de Silva	86,050		George	Smith	85402
3	George	Smith	85,402		Keith	Ng	86599
4	Pamela	Jones	79,573		Kelly	Chen	84948
5	Keith	Ng	86,599		Pamela	Jones	79573
6	Kelly	Chen	84,948		Sally	de Silva	86050
7	Sue	Tan	76,847		Sue	Tan	76847

SORTBY function

We can also sort by a column that we don't display. This is when you use the SORTBY function instead of the SORT function. From our previous example, maybe we want to sort by the Salary column, but we don't want to display the values. It can also be used to perform sorts based on multiple columns.

Syntax

SORTBY(array,by_array1,[sort_order1],[by_array2],[sort_order2],…)

array – the range to sort. It can be a single or a multiple column range. This can also be another function that return a range like FILTER or UNIQUE.

by_array1 – this is the range to base the sort on – it won't display. The first by_array is required, but you can add more columns to sort by, these are optional.

sort_order1 – optional - 1 is ascending and -1 for descending. If omitted is default to 1, ascending.

The example on the right has the two name columns displayed based on a sort of the Salary column.

The sort order is determined by the default value for the third argument which was omitted and is 1, or ascending. The formula in cell E2.

E2 =SORTBY(A2:B7,C2:C7)

	A	B	C	D	E	F	G
1	**First Name**	**Last Name**	**Salary**		**First Name**	**Last Name**	**Salary**
2	Sally	de Silva	86,050		Sue	Tan	
3	George	Smith	85,402		Pamela	Jones	
4	Pamela	Jones	79,573		Kelly	Chen	
5	Keith	Ng	86,599		George	Smith	
6	Kelly	Chen	84,948		Sally	de Silva	
7	Sue	Tan	76,847		Keith	Ng	

=SORTBY(A2:B7,C2:C7)

Top three

To show only the first three results we can use another function that will be covered in more detail in a future session. The TAKE function.

E2 =TAKE(SORTBY(A2:B7,C2:C7),J1)

	A	B	C	D	E	F	G	H	I	J
1	**First Name**	**Last Name**	**Salary**		**First Name**	**Last Name**	**Salary**		**Top**	3
2	Sally	de Silva	86,050		Sue	Tan				
3	George	Smith	85,402		Pamela	Jones				
4	Pamela	Jones	79,573		Kelly	Chen				
5	Keith	Ng	86,599							
6	Kelly	Chen	84,948							
7	Sue	Tan	76,847							

The formula in cell E2 is.

=TAKE(SORTBY(A2:B7,C2:C7),J1)

This extracts the first 3 rows from the sorted list.

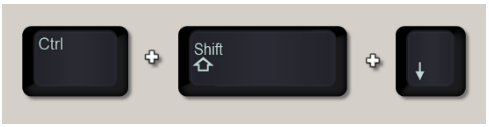
Worked Example

In the Data sheet we have a sales data table with over 8,000 rows. We want to create a two-condition filter based on State and Source.

Steps

1. In the Data sheet in cell R6 type =UNIQUE(

   Use the mouse to click on cell D2 and hold down the Ctrl + Shift keys and press the down arrow. This selects the range below. Press Enter.



   The final formula is.

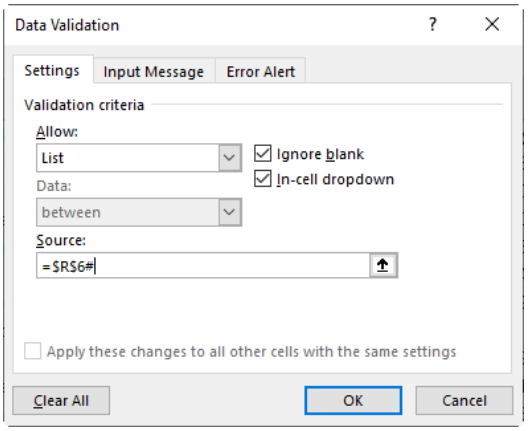
   =UNIQUE(D2:D8250)

   Add the SORT function to the formula as below.

   =SORT(UNIQUE(D2:D8250))

2. Copy the R6 formula to cell S6.

3. Select cell S2 and press **Alt D L** in sequence (don't hold the keys down) this opens the Data Validation dialog - image on right.



   In the Allow drop down select List.

   Click in the Source box and use the mouse to select cell R6 and then add the # to the end of the reference and click OK.

4. In cell S3 repeat the previous step but select cell S6 in the Source box and add # and click OK.

5. In cell U1 type = use the mouse to select the range A1:P1 and press Enter. This brings across the data headings. The formula is.

   =A1:P1

6. In cell U2 enter the following formula.

   =FILTER(A2:P8250,(D2:D8250=S2)*(E2:E8250=S3),"none")

   Tip:
   Select the first cell and use Ctrl + Shift + Down arrow to select the long ranges.

   Select a State (cell S2) and Source (cell S3).