

Excel Lesson: **More Advanced Table Lookup Functions**

Topics Covered:

MATCH()

INDEX()

2-Way lookups (find data in a table based on entry in one column and one row)

2-Column lookups (find data based on results of lookups done in two columns.)

Real World Example Application:

While Excel provides some very strong built-in functions for doing lookups of data within tables, those are mostly built around the lookup seeking a match in a single column/row of information and then returning related information about that match from within the table. For more on those features read the [ExcelLookupsLesson.pdf](#)¹ Lesson.

But sometimes those functions just don't do the job. Sometimes you may need to find a value based upon a match in the values in the top row of a table combined with a match of the values in the first column of the table. Or you may need to find a value based on the values in pairs of column entries. In these cases, **VLOOKUP()**, **HLOOKUP()** and **LOOKUP()** just won't do the job.

Consider a sales chart – across the top are the months of the year, and down the left side are various products sold. At the intersection of these column/row combinations are quantities sold or total dollars for the sales in that month for that item. To find out how many widgets were sold in February, you need to be able to do a 2-Way lookup.

Or maybe you have a list of merchandise with a few options available that cause changes in the cost of the merchandise (as models of vehicles or clothing items with optional accessories). In this case you may need to use a 2-column lookup.

To implement a 2-way or 2-column lookup, you need to use two built in functions from Excel:

MATCH() - Returns the relative position of an item in an array that matches a specified value in a specified order. Use **MATCH** instead of one of the **LOOKUP** functions when you need the position of an item in a range instead of the item itself.

INDEX() - Returns a value or the reference to a value from within a table or range.

Specific information on how both the **MATCH()** and **INDEX()** functions work, their syntax, and basic use can be found in [ExcelMatchAndIndexLesson.pdf](#)².

And before continuing, if you are not familiar with setting up equations in Excel, addressing and Named Range definitions and use, please read [ExcelBasicsLesson.pdf](#)³ before continuing.

¹ <http://www.jlathamsite.com/teach/ExcelLookupsLesson.pdf>

² <http://www.jlathamsite.com/teach/ExcelMatchAndIndexLesson.pdf>

³ <http://www.jlathamsite.com/teach/ExcelBasicsLesson.pdf>

The **two-way lookup** analyzed. For this lesson you'll need to look at the [ExcelMoreLookupsLesson.xls](http://www.jlathamsite.com/teach/ExcelMoreLookupsLesson.xls)⁴ workbook. The TwoWayLookups worksheet looks like this:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		January	February	March	April	May	June	July	August	September	October	November	December
2	Widgets	7	3	5	10	29	4	18	3	10	9	19	34
3	Gadgets	12	11	10	9	8	7	6	5	4	3	8	9
4	Turtles	0	0	0	4	6	8	20	8	7	3	0	0
5	Girdles	18	5	7	4	10	4	12	8	6	4	4	18
6	YoYos	3	3	3	3	3	4	3	4	2	4	3	6
7	Coal	5	5	4	3	0	0	0	0	3	4	5	109
8													
9													
10		Month:	December			Named Ranges:							
11		Product:	YoYos			MonthList	A1:M1						
12		Sales	6			ProductList	A1:A7						
13						SalesTable	A1:M7						
14	Indexes:	month	13			TheMonth	C10						
15		product	6			TheProduct	C11						

I've laid this out all in final fashion with a few added items to help explain things. For instance, the list of Named Ranges and the two Indexes displays aren't actually needed for use, they are there for explanation.

A couple of things to notice: Because we have defined our SalesTable range to extend from A1 to M7, we also have to include cell A1 in both our MonthList and ProductList so that the MATCH() functions used with them return the proper index values into the table.

If you click in either cell C10 (TheMonth) or C11 (TheProduct) you will see a pull-down arrow appear at the right edge of the cell. Clicking on it gives you a list of valid entries. If you have not seen/done this before, it was done by clicking in each cell and then choosing Data / Validation and working from there. If you wish to see the setup for those cells, click in them and choose Data / Validation to see their setup.

A couple of quick notes about Data / Validation using a list as was done here:

Advantage: gives your users a complete list of properly spelled and formatted items to pick from. This increases accuracy and reduces the possibility of error.

Disadvantage: the list used for selection must be on the same sheet with the cell using the list. This can cause you to have to create worksheets with extra areas on them to hold the lists and have to set the sheets up for printing with this thought in mind (think of restricting the Print Area to exclude the source of the lists).

Maintenance: by using a defined range with absolute addressing like \$B\$1:\$M\$1 you have to watch out for additions/deletions from the list and/or movement of the list around on the worksheet. Better to use Named Ranges for this kind of thing.

⁴ <http://www.jlathamsite.com/teach/ExcelMoreLookupsLesson.xls>

The real workhorse cell on the sheet is at C12. The formula for that cell looks like this:
=INDEX(SalesTable,MATCH(TheProduct,ProductList,0),MATCH(TheMonth,MonthList,0))

It looks a bit complicated, but really isn't. Remember that the INDEX() function needs three parameters to do its job, and those are (as they relate to this example)

Parameter #1: The address reference to the table where the looked-up information is to come from.

Parameter #2: The row offset value into the table specified by Parameter #1.

Parameter #3: The column offset value into the table specified by Parameter #1.

Since the MATCH() function gives us values we can apply as index values, we just use that function to determine those in the proper order "on the fly" so to speak. If you look down where I've placed the Indexes information, you can see the actual values they return for any given choice of either product or month. Since those values are there, we could have written our 2-way lookup formula like this:

=INDEX(A1:M7,C15,C14)

Or using the data table's name,

=INDEX(SalesTable,C15,C14)

But by including the MATCH() functions as part of the formula in cell C12, we don't have to even have anything like cells C14 and C15 on our worksheet – saving worksheet space, system resources, and keeping our sheet neater.

Two-Column Lookup

We will begin by taking a quick look at the worksheet that's been set up to help with this:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Uses an Array Formula												
2	When creating or after editing an Array Formula use [Shift]+[Ctrl]+[Enter] to exit formula entry/editing.												
3													
4	Model	Westerner			Model	Bedrooms	Model#	Baths					
5	Bedrooms	4			Colonial	2	C2	1				Bridgeport	
6	Model#	W4			Colonial	3	C3	1.5				Colonial	
7					Colonial	4	C4	2				Palace	
8					Bridgeport	3	B3	1.5				Tudor	
9					Bridgeport	4	B4	2				Westerner	
10					Tudor	2	T2	1					2
11					Tudor	3	T3	2					3
12					Westerner	2	W2	1					4
13					Westerner	3	W3	1.5					5
14					Westerner	4	W4	2					
15					Palace	4	P4	2.5				Named Ranges	
16					Palace	5	P5	4				Models	E5:E16
17												Bedrooms	F5:F16
18												ModelCode	G5:G16

When would you need such a lookup? You might need a 2-column lookup in cases where none of the VLOOKUP(), HLOOKUP(), LOOKUP() or even 2-way lookups works. With all of those the intersection of rows and columns forms results in a single value. If you look at the table of information above you see that each Model and # of bedroom combinations has 2 more things associated with them: model number and number of baths. You could stretch that on out to include pricing, square footage and many other items that simply wouldn't fit into a single cell reasonably.

I've set up a couple of selection cells there at B4 and B5 so you can choose items from the two columns we'll be using to finish out our lookup. Cell B6 shows the results (note that it will display #N/A if no lookup match is found – as with choosing any Model other than the 'Palace' with 5 bedrooms). The stuff out in columns L and M is mostly informational, although the lists used for the pull-down validation in cells B4 and B5 are out there also.

The heart of this whole thing is an odd looking formula in cell B6:
`{=INDEX(ModelCode,MATCH(B4&B5,Models&Bedrooms,0))}`

The first thing you probably notice is that it doesn't start with an = symbol, but with a left curly brace {. How'd that get there?

It got there because after typing in the formula in the normal fashion (including starting it with an = symbol) we ended it not with just the [Enter] key, but with a key combination: [Shift]+[Ctrl]+[Enter]. That has to be the terminator for entry/editing of what are known as *array formulas* in Excel.

What/how array formulas are is a difficult concept to explain in a few words. Try to look at them as being formulas that don't just look at one cell or range of cells, but rather they look at ranges of cells and evaluate all cells within that range at once as individuals. Told you it wasn't easy to explain. I'll try again: in this case **Models** refers to the range E5:E16 and the range

Bedrooms refers to F5:F16, and the way we've used the ampersand (&) symbol tells Excel to pair up the contents of each pair of cells in those ranges one at a time so that internally it sees "Colonial2", "Colonial3" ... "Palace5" as it works through the list of items in those ranges. Just before doing that it had pulled the contents of cells B4 and B5 together in similar fashion for the comparison. Excel is smart enough to figure out which are individual cell references and which ones it needs to use to perform the added evaluations. Array formulas are extremely powerful, allowing you to do things that cannot be done otherwise outside of a Visual Basic for Applications (VBA) code module – and generally the array function works faster than the VBA code to do the same thing.

Could we have done this any other way, without resorting to an array formula? The answer is an unqualified YES.

You could have set up a column left of the model numbers column and set it up as is shown down in the lower area of our sheet:

Non-Two-Column Setup				Model	Bedrooms	Added	Model#	Baths
Model	Westerner			Colonial	2	Colonial2	C2	1
Bedrooms	4	Westerner4		Colonial	3	Colonial3	C3	1.5
Model#	W4			Colonial	4	Colonial4	C4	2
				Bridgeport	3	Bridgeport3	B3	1.5
				Bridgeport	4	Bridgeport4	B4	2
				Tudor	2	Tudor2	T2	1
				Tudor	3	Tudor3	T3	2
				Westerner	2	Westerner2	W2	1
				Westerner	3	Westerner3	W3	1.5
				Westerner	4	Westerner4	W4	2
				Palace	4	Palace4	P4	2.5
				Palace	5	Palace5	P5	4

We added one formula in Cell C25 there next to the cell where we choose # of bedrooms:
 =B24&B25

That gives us our first parameter for the VLOOKUP() formula we set up in the results cell there next to the Model # phrase:
 =VLOOKUP(C25,G24:I35,2,FALSE)

Cells G24:I35 include all the data under the columns labeled "Added", "Model#" and "Baths" in our data table. Notice in the "Added" column, we've used a formula like
 = En & Fn

Where "n" is the row number. This does the concatenation (pulling together) of the two pieces of information we need to match to our value used in the VLOOKUP() and let it replace the fancy-schmancy Array Formula. Obviously the array formula method is a bit cleaner with regards to worksheet layout, at the expense of perhaps losing some of the understandability, and adding a little confusion during editing operations if they are required later (remembering to terminate the edit with that 3-key combination to keep it defined as an array formula).

ENJOY!