



ULTIMATE INDEX and MATCH

CHIN CHEE CHONG
ccchin.training@gmail.com

Contents

Using Index & Match Functions Combo in Excel (10 Easy Examples).....	2
INDEX Function: Finds the Value Based on Coordinates	2
MATCH Function: Finds the Position based on a Lookup Value	4
Let's Combine Them to Create a Powerhouse (INDEX + MATCH)	6
Example 1: A simple Lookup Using INDEX MATCH Combo.....	7
Example 2: Lookup to the Left	10
Example 3: Two Way Lookup	11
Example 4: Lookup Value From Multiple Column/Criteria	14
Example 5: Get Values from Entire Row/Column	16
Example 6: Find the Student's Grade (Approximate Match Technique)	19
Example 7: Case Sensitive Lookups.....	22
Example 8: Find the Closest Match.....	24
Example 9: Use INDEX MATCH with Wildcard Characters.....	26
Example 10: Three Way Lookup	29
Why is INDEX/MATCH Better than VLOOKUP?.....	33
INDEX/MATCH can look to the Left (as well as to the right) of the lookup value	33
INDEX/MATCH can work with vertical and horizontal ranges	33
VLOOKUP cannot work with descending data.....	34
INDEX/MATCH can be slightly faster	34
INDEX/MATCH is Independent of the Actual Column Position	34
VLOOKUP is easier to use.....	36
Bonus Section.....	37
Method 1: Using helper cells	38
Method 2: Using CTRL + SHIFT + ENTER (CSE)	39
Method 3: Using two INDEX() functions	42

Using Index & Match Functions Combo in Excel (10 Easy Examples)

Excel has a lot of functions – about 450+ of them.

And many of these are simply awesome. The amount of work you can get done with a few formulas still surprises me (even after having used Excel for 10+ years).

And among all these amazing functions, the INDEX MATCH functions combo stands out.

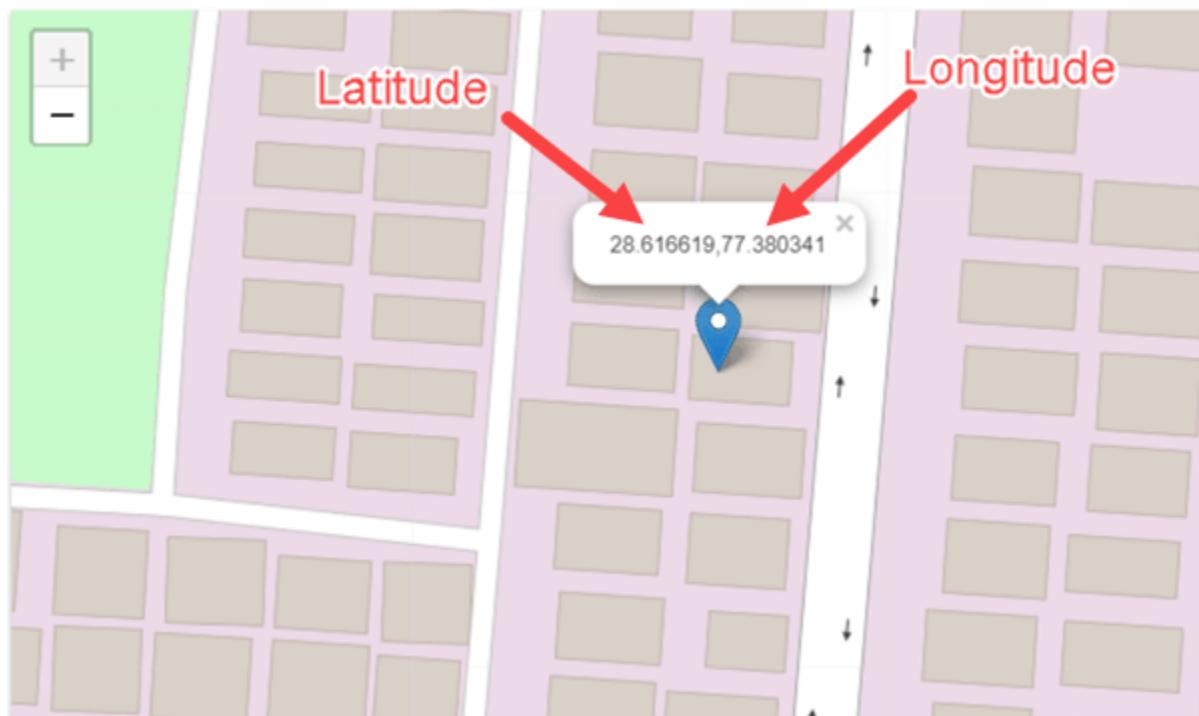
INDEX Function: Finds the Value Based on Coordinates

The easiest way to understand how Index function works is by thinking of it as a GPS satellite.

As soon as you tell the satellite the latitude and longitude coordinates, it will know exactly where to go and find that location.

So despite having a mind-boggling number of lat-long combinations, the satellite would know exactly where to look.

I quickly did a search for my work location and this is what I got.



Anyway, enough of geography.

Just like a satellite needs latitude and longitude coordinates, the INDEX function in Excel would need the row and column number to know what cell you're referring to.

And that's Excel INDEX function in a nut-shell.

So let me define it in simple words for you.

The INDEX function will use the row number and column number to find a cell in the given range and return the value in it.

	A	B	C	D	E	F	G	H	I
1	70	33	72	91	67	42	54	68	74
2	90	25	57	44	20	29	90	47	54
3	77	32	92	88	70	30	23	8	59
4	19	57	86	8	94	83	21	26	95
5	39	73	8	13	92	53	77	42	4
6	73	80	18	92	1	80	19	27	84
7	31	88	12	93	88				77
8	49	44	94	37	43	55	17	94	16
9	96	30	79	70	99	18	55	68	70
10	62	6	57	88	56	69	70	61	42
11	88	28	41	90	67	16	94	36	11
12	4	88	82	39	19	87	64	75	58
13	48	79	29	30	40	52	87	24	92
14	10	21	44	19	14	76	92	11	43
15	55	92	88	14	100	10	26	52	52
16	32	5	35	30	25	37	81	99	96
17	75	51	16	32	23	43	59	91	3
18	98	64	86	65	61	89	85	17	12

Row 9, Column 5

All by itself, INDEX is a very simple function, with no utility. After all, in most cases, you are not likely to know the row and column numbers.

But...

The fact that you can use it with other functions (hint: MATCH) that can find the row number and the column number makes INDEX an extremely powerful Excel function.

Below is the syntax of the INDEX function:

```
=INDEX (array, row_num, [col_num])
=INDEX (array, row_num, [col_num], [area_num])
```

- array – a range of cells or an array constant.
- row_num – the row number from which the value is to be fetched.
- [col_num] – the column number from which the value is to be fetched. Although this is an optional argument, but if row_num is not provided, it needs to be given.
- [area_num] – (Optional) If array argument is made up of multiple ranges, this number would be used to select the reference from all the ranges.

INDEX function has 2 syntaxes (just FYI).

The first one is used in most cases. The second one is used in advanced cases only (such as doing a three-way lookup) which we will cover in one of the examples later in this tutorial.

But if you're new to this function, just remember the first syntax.

MATCH Function: Finds the Position based on a Lookup Value

Going back to my previous example of longitude and latitude, MATCH is the function that can find these positions (in the Excel spreadsheet world).

In simple language, the Excel MATCH function can find the position of a cell in a range.

And on what basis would it find a cell's position?

Based on the lookup value.

For example, if you have a list as shown below and you want to find the position of the name 'Mark' in it, then you can use the MATCH function.

C2	A	B	C	D
			=MATCH("Mark", \$A\$1:\$A\$9, 0)	
1	John			
2	Jenny			
3	Mark			
4	Jim			
5	Michael			
6	Hannah			
7	Kane			
8	Alan			
9	Angela			

The function returns 3, as that's the position of the cell with the name Mark in it.

MATCH function starts looking from top to bottom for the lookup value (which is 'Mark') in the specified range (which is A1:A9 in this example). As soon as it finds the name, it returns the position in that specific range.

Below is the syntax of the MATCH function in Excel.

```
=MATCH(lookup_value, lookup_array, [match_type])
```

- `lookup_value` – The value for which you are looking for a match in the `lookup_array`.
- `lookup_array` – The range of cells in which you are searching for the `lookup_value`.
- `[match_type]` – (Optional) This specifies how excel should look for a matching value. It can take three values -1, 0 , or 1.

Understanding Match Type Argument in MATCH Function

There is one additional thing you need to know about the MATCH function, and it's about how it goes through the data and finds the cell position.

The third argument of the MATCH function can be 0, 1 or -1.

Below is an explanation of how these arguments work:

- 0 – this will look for an exact match of the value. If an exact match is found, the MATCH function will return the cell position. Else, it will return an error.
- 1 – this finds the largest value that is less than or equal to the lookup value. For this to work, your data range needs to be sorted in ascending order.
- -1 – this finds the smallest value that is greater than or equal to the lookup value. For this to work, your data range needs to be sorted in descending order.

To summarize and put it in simple words:

- INDEX needs the cell position (row and column number) and gives the cell value.
- MATCH finds the position by using a lookup value.

Let's Combine Them to Create a Powerhouse (INDEX + MATCH)

Now that you have a basic understanding of how INDEX and MATCH functions work individually, let's combine these two and learn about all the wonderful things it can do.

To understand this better, I have a few examples that use the INDEX MATCH combination.

I will start with a simple example and then show you some advanced use cases as well.

Example 1: A simple Lookup Using INDEX MATCH Combo

Let's do a simple lookup with INDEX/MATCH.

Below is a table where I have the marks for ten students.

	A	B
1	Name	Marks
2	John	93
3	Jenny	88
4	Mark	47
5	Jim	65
6	Michael	41
7	Hannah	47
8	Kane	32
9	Alan	57
10	Angela	89
11	Chris	25

From this table, I want to find the marks for Jim.

Below is the formula that can easily do this:

```
=INDEX($A$2:$B$11,MATCH("Jim",$A$2:$A$11,0),2)
```

The screenshot shows a Microsoft Excel spreadsheet. The formula bar at the top contains the formula `=INDEX(A2:B11,MATCH("Jim",A2:A11,0),2)`. The main area displays a table with columns 'Name' and 'Marks'. A red arrow points from the formula bar to the value '65' in cell D3, which is highlighted with a green border. The cell D3 also has a callout bubble labeled 'Jim's Marks'.

	A	B	C	D	E
1	Name	Marks			
2	John	93			
3	Jenny	88			
4	Mark	47			
5	Jim	65			
6	Michael	41			
7	Hannah	47			
8	Kane	32			
9	Alan	57			
10	Angela	89			
11	Chris	25			

Now, if you're thinking this can easily be done using a VLOOKUP function, you're right! This is not the best use of INDEX MATCH awesomeness. Despite the fact that I am a fan of INDEX MATCH, it is a little more difficult than VLOOKUP. If fetching data from a column on the right is all you want to do, I recommend you use VLOOKUP.

The reason I have shown this example, which can also easily be done with VLOOKUP is to show you how INDEX MATCH works in a simple setting.

Now let me show a benefit of INDEX MATCH.

Suppose you have the same data, but instead of having it in columns, you have it in rows (as shown below).

	A	B	C	D	E	F	G	H	I	J	K
1	Name	John	Jenny	Mark	Jim	Michael	Hannah	Kane	Alan	Angela	Chris
2	Marks	93	88	47	65	41	47	32	57	89	25

You know what, you can still use INDEX MATCH combo to get Jim's marks.

Below is the formula that will give you the result:

```
=INDEX($B$1:$K$2,2,MATCH("Jim",$B$1:$K$1,0))
```

	A	B	C	D	E	F	G	H	I	J	K
1	Name	John	Jenny	Mark	Jim	Michael	Hannah	Kane	Alan	Angela	Chris
2	Marks	93	88	47	65	41	47	32	57	89	25
3											
4											
5											

Jim's Marks

65

Note that you need to change the range and switch the row/column parts to make this formula work for horizontal data as well.

This can't be done with VLOOKUP, but you can still do this easily with HLOOKUP.
INDEX MATCH combination can easily handle horizontal as well as vertical data.

Example 2: Lookup to the Left

It's more common than you think.

A lot of times, you may be required to fetch the data from a column which is to the left of the column that has the lookup value.

Something as shown below:

	A	B	C
1	Store	Sales	Manager
2	1	679	John
3	2	568	Jenny
4	3	534	Mark
5	4	679	Jim
6	5	757	Michael
7	6	798	Hannah
8	7	544	Kane
9	8	751	Alan
10	9	598	Angela
11	10	571	Chris

To find out Michael's sales, you will have to do a lookup on the left.

If you're thinking VLOOKUP, let me stop your right there.

VLOOKUP is not made to look for and fetch the values on the left.

Can you still do it using VLOOKUP?

Yes, you can!

But that can turn into a long and ugly formula.

So if you want to do a lookup and fetch data from the columns on the left, you are better off using INDEX MATCH combo.

Below is the formula that will get Michael's sales number:

```
=INDEX($A$2:$C$11,MATCH("Michael",C2:C11,0),2)
```

The screenshot shows a Microsoft Excel spreadsheet. The formula bar at the top contains the formula `=INDEX(A2:C11,MATCH("Michael",C2:C11,0),2)`. Below the formula bar is a table with four columns: Store, Sales, and Manager. The Manager column contains names: John, Jenny, Mark, Jim, Michael, Hannah, Kane, Alan, Angela, and Chris. The cell E4 contains the formula `=INDEX(A2:C11,MATCH("Michael",C2:C11,0),2)`. The cell E5 contains the value `757`, which is the sales figure for Michael. A red box highlights the formula in the formula bar.

	A	B	C	D	E	F	G
1	Store	Sales	Manager		Michael's Sales		
2	1	679	John				
3	2	568	Jenny				
4	3	534	Mark				
5	4	679	Jim				
6	5	757	Michael				
7	6	798	Hannah				
8	7	544	Kane				
9	8	751	Alan				
10	9	598	Angela				
11	10	571	Chris				

Another point here for INDEX MATCH. VLOOKUP can fetch the data only from the columns that are to the right of the column that has the lookup value.

Example 3: Two Way Lookup

So far, we have seen the examples where we wanted to fetch the data from the column adjacent to the column that has the lookup value.

But in real life, the data often spans through multiple columns.

INDEX MATCH can easily handle a two-way lookup.

Below is a dataset of the student's marks in three different subjects.

	A	B	C	D
1	Name	Math	Physics	Chemistry
2	John	71	53	89
3	Jenny	88	45	55
4	Mark	44	28	54
5	Jim	97	70	73
6	Michael	47	34	38
7	Hannah	88	75	90
8	Kane	87	42	54
9	Alan	27	37	56
10	Angela	35	89	57
11	Chris	67	54	82

If you want to quickly fetch the marks of a student in all three subjects, you can do that with INDEX MATCH.

The below formula will give you the marks for Jim for all the three subjects (copy and paste in one cell and drag to fill other cells or copy and paste on other cells).

```
=INDEX($B$2:$D$11,MATCH($F$3,$A$2:$A$11,0),MATCH(G$2,$B$1:$D$1,0))
```

	A	B	C	D	E	F	G	H	I
1	Name	Math	Physics	Chemistry					
2	John	71	53	89					
3	Jenny	88	45	55					
4	Mark	44	28	54					
5	Jim	97	70	73					
6	Michael	47	34	38					
7	Hannah	88	75	90					
8	Kane	87	42	54					
9	Alan	27	37	56					
10	Angela	35	89	57					
11	Chris	67	54	82					

Let me quickly also explain this formula.

INDEX formula uses B2:D11 as the range.

The first MATCH uses the name (Jim in cell F3) and fetches the position of it in the names column (A2:A11). This becomes the row number from which the data needs to be fetched.

The second MATCH formula uses the subject name (in cell G2) to get the position of that specific subject name in B1:D1. For example, Math is 1, Physics is 2 and Chemistry is 3.

Since these MATCH positions are fed into the INDEX function, it returns the score based on the student name and subject name.

This formula is dynamic, which means that if you change the student name or the subject names, it would still work and fetch the correct data.

One great thing about using INDEX/MATCH is that even if you interchange the names of the subjects, it will continue to give you the correct result.

Example 4: Lookup Value From Multiple Column/Criteria

Suppose you have a dataset as shown below and you want to fetch the marks for 'Mark Long'.

	A	B	C
1	First Name	Last Name	Math
2	John	Haynes	71
3	Mark	Frost	88
4	Mark	Long	44
5	Jim	Hansen	97
6	Michael	Rubio	47
7	Hannah	Bass	88
8	Kane	Becker	87
9	Alan	Lockard	27
10	Angela	Lott	35
11	Chris	Whit	67

Since the data is in two columns, I can't do a lookup for Mark and get the data.

If I do it that way, I am going to get the marks data for Mark Frost and not Mark Long (because the MATCH function will give me the result for the MARK it meets).

One way of doing this is to create a helper column and combine the names. Once you have the helper column, you can use VLOOKUP and get the marks data.

But with INDEX/MATCH combo, you don't need a helper column. You can create a formula that handles multiple criteria in the formula itself.

The below formula will give the result.

```
=INDEX($C$2:$C$11,MATCH($E$3&"|"&$F$3,$A$2:A11&"|"&$B$2:$B$11,0))
```

	A	B	C	D	E	F	G
1	First Name	Last Name	Math				
2	John	Haynes	71				
3	Mark	Frost	88				
4	Mark	Long	44				
5	Jim	Hansen	97				
6	Michael	Rubio	47				
7	Hannah	Bass	88				
8	Kane	Becker	87				
9	Alan	Lockard	27				
10	Angela	Lott	35				
11	Chris	Whit	67				

First Name	Last Name	Score
Mark	Long	44

Let me quickly explain what this formula does.

The MATCH part of the formula combines the lookup value (Mark and Long) as well as the entire lookup array. When \$A\$2:A11&"|"&\$B\$2:\$B\$11 is used as the lookup array, it actually checks the lookup value against the combined string of first and last name (separated by the pipe symbol).

This ensures that you get the right result without using any helper columns.

You can do this kind of lookup (where there are multiple columns/criteria) with VLOOKUP as well, but you need to use a helper column. INDEX MATCH combo makes it slightly easy to do this without any helper columns.

Example 5: Get Values from Entire Row/Column

In the examples above, we have used the INDEX function to get value from a specific cell. You provide the row and column number, and it returns the value in that specific cell.

But you can do more.

You can also use the INDEX function to get the values from an entire row or column.

And how can this be useful you ask!

Suppose you want to know the total score of Jim in all the three subjects.

You can use the INDEX function to first get all the marks of Jim and then use the SUM function to get a total.

Let's see how to do this.

Below I have the scores of all the students in three subjects.

	A	B	C	D
1	Name	Math	Physics	Chemistry
2	John	71	53	89
3	Jenny	88	45	55
4	Mark	44	28	54
5	Jim	97	70	73
6	Michael	47	34	38
7	Hannah	88	75	90
8	Kane	87	42	54
9	Alan	27	37	56
10	Angela	35	89	57
11	Chris	67	54	82

The below formula will give me the total score of Jim in all the three subjects.

```
=SUM(INDEX($B$2:$D$11,MATCH($F$4,$A$2:$A$11,0),0))
```

The screenshot shows a Microsoft Excel spreadsheet. The formula bar at the top contains the formula `=SUM(INDEX(B2:D11,MATCH(F4,A2:A11,0),0))`. Below the formula bar is a table with columns labeled A through J. The first row contains column headers: Name, Math, Physics, and Chemistry. The second row contains data for John: 71, 53, 89. The third row contains data for Jenny: 88, 45, 55. The fourth row contains data for Mark: 44, 28, 54. The fifth row contains data for Jim: 97, 70, 73. The sixth row contains data for Michael: 47, 34, 38. The seventh row contains data for Hannah: 88, 75, 90. The eighth row contains data for Kane: 87, 42, 54. The ninth row contains data for Alan: 27, 37, 56. The tenth row contains data for Angela: 35, 89, 57. The eleventh row contains data for Chris: 67, 54, 82. To the right of the table, there is a small inset showing a row labeled "Total" with "Jim" in the first cell and "240" in the second cell, with a red arrow pointing from the cell containing Jim's name to the "Jim" in the inset.

	A	B	C	D	E	F	G	H	I	J
1	Name	Math	Physics	Chemistry						
2	John	71	53	89						
3	Jenny	88	45	55						
4	Mark	44	28	54						
5	Jim	97	70	73						
6	Michael	47	34	38						
7	Hannah	88	75	90						
8	Kane	87	42	54						
9	Alan	27	37	56						
10	Angela	35	89	57						
11	Chris	67	54	82						

Let me explain how this formula works.

The trick here is to use 0 as the column number.

When you use 0 as the column number in the INDEX function, it will return all the row values.

Similarly, if you use 0 as the row number, it will return all the values in the column.

So the below part of the formula returns an array of values – {97, 70, 73}

```
INDEX($B$2:$D$11,MATCH($F$4,$A$2:$A$11,0),0)
```

If you just enter this above formula in a cell in Excel and hit enter, you will see a #VALUE! error. This is because it's not returning a single value, but an array of value.

But don't worry, the array of values are still there. You can check this by selecting the formula and press the F9 key. It will show you the result of the formula which in this case is an array of three value – {97, 70, 73}

Now, if you wrap this INDEX formula in the SUM function, it will give you the sum of all the marks scored by Jim.

You can also use the same to get the highest, lowest and average marks of Jim.

Just like we have done this for a student, you can also do this for a subject. For example, if you want the average score in a subject, you can keep the row number as 0 in the INDEX formula and it will give you all the column values of that subject.

Example 6: Find the Student's Grade (Approximate Match Technique)

So far, we have used the MATCH formula to get the exact match of the lookup value.

But you can also use it to do an approximate match.

Now, what the hell is Approximate Match?

Let me explain.

When you're looking for stuff such as names or ids, you're looking for an exact match. But sometimes, you need to know the range in which your lookup values lie. This is usually the case with numbers.

For example, as a class teacher, you may want to know what's the grade of each student in a subject, and the grade is decided based on the score.

Below is an example, where I want the grade for all the students and the grading is decided based on the table on the right.

	A	B	C	D	E	F
1	Name	Math	Grade		Marks Range	Grade
2	John	71			0	F
3	Jenny	88			33	E
4	Mark	44			50	D
5	Jim	97			70	C
6	Michael	47			80	B
7	Hannah	88			90	A
8	Kane	87				
9	Alan	27				
10	Angela	35				
11	Chris	67				

So if a student gets less than 33, the grade is F and if he/she gets less than 50 but more than 33, it's E, and so on.

Below is the formula that will do this.

```
=INDEX($F$3:$F$8,MATCH(B2,$E$3:$E$8,1),1)
```

The screenshot shows an Excel interface. The formula bar at the top contains the formula `=INDEX(F3:F8,MATCH(B2,E3:E8,1),1)`. Below the formula bar is a table with two parts. The left part is a student marks table with columns for Name, Math, and Grade. The right part is a grade conversion table with columns for Marks Range and Grade.

	A	B	C	D	E	F	G	H
1	Name	Math	Grade					
2	John	71	C					
3	Jenny	88	B					
4	Mark	44	E					
5	Jim	97	A					
6	Michael	47	E					
7	Hannah	88	B					
8	Kane	87	B					
9	Alan	27	F					
10	Angela	35	E					
11	Chris	67	D					

Marks Range	Grade
0	F
33	E
50	D
70	C
80	B
90	A

Let me explain how this formula works.

In the MATCH function, we have used 1 as the [match_type] argument. This argument will return the largest value that is less than or equal to the lookup value.

This means that the MATCH formula goes through the marks range, and as soon as it finds a marks range that is equal to or less than the lookup marks value, it will stop there and return its position.

So if the lookup mark value is 20, the MATCH function would return 1 and if it's 85, it would return 5.

And the INDEX function uses this position value to get the grade.

IMPORTANT: For this to work, your data needs to be sorted in ascending order. If it's not, you can get wrong results.

Note that the above can also be done using below VLOOKUP formula:

```
=VLOOKUP(B2,$E$3:$F$8,2,TRUE)
```

But MATCH function can go a step further when it comes to approximate match.

You can also have a descending data and can use INDEX MATCH combo to find the result. For example, if I change the order of the grade table (as shown below), I can still find the grades of the students.

The screenshot shows two tables in Excel. The first table (A1:C11) has columns A (Name), B (Math), and C (Grade). The second table (F1:F7) has columns Marks Range and Grade. The formula bar shows =INDEX(\$F\$3:\$F\$8,MATCH(B2,\$E\$3:\$E\$8,-1),1).

	A	B	C	D	E	F	G	H
1	Name	Math	Grade					
2	John	71	C					
3	Jenny	88	B					
4	Mark	44	E					
5	Jim	97	A					
6	Michael	47	E					
7	Hannah	88	B					
8	Kane	87	B					
9	Alan	27	F					
10	Angela	35	E					
11	Chris	67	D					

Marks Range	Grade
100	A
90	B
80	C
70	D
50	E
33	F

To do this, all I have to do is change the [match_type] argument to -1.

Below is the formula that I have used:

```
=INDEX($F$3:$F$8,MATCH(B2,$E$3:$E$8,-1),1)
```

VLOOKUP can also do an approximate match but only when data is sorted in ascending order (but it doesn't work if the data is sorted in descending order).

Example 7: Case Sensitive Lookups

So far all the lookups we have done have been case insensitive.

This means that whether the lookup value was Jim or JIM or jim, it didn't matter. You'll get the same result.

But what if you want the lookup to be case sensitive.

This is usually the case when you have large data sets and a possibility of repetition or distinct names/ids (with the only difference being the case)

For example, suppose I have the following data set of students where there are two students with the name Jim (the only difference being that one is entered as Jim and another one as jim).

	A	B
1	Name	Marks
2	Jim	93
3	Jenny	88
4	Mark	47
5	jim	65
6	Michael	41
7	Hannah	47
8	Kane	32
9	Alan	57
10	Angela	89
11	Chris	25

Note that there are two students with the same name – Jim (cell A2 and A5).

Since a normal lookup wouldn't work, you need to do a case sensitive lookup.

Below is the formula that will give you the right result. Since this is an array formula, you need to use Control + Shift + Enter.

```
=INDEX($B$2:$B$11,MATCH(TRUE,EXACT(D3,A2:A11),0),1)
```

	A	B	C	D	E	F	G
1	Name	Marks					
2	Jim	93					
3	Jenny	88					
4	Mark	47					
5	jim	65					
6	Michael	41					
7	Hannah	47					
8	Kane	32					
9	Alan	57					
10	Angela	89					
11	Chris	25					

Let me explain how this formula works.

The EXACT function checks for an exact match of the lookup value (which is 'jim' in this case). It goes through all the names and returns FALSE if it isn't a match and TRUE if it's a match.

So the output of the EXACT function in this example is –

```
{FALSE;FALSE;FALSE;TRUE;FALSE;FALSE;FALSE;FALSE;FALSE}
```

Note that there is only one TRUE, which is when the EXACT function found a perfect match.

The MATCH function then finds the position of TRUE in the array returned by the EXACT function, which is 4 in this example.

Once we have the position, the INDEX function uses it to find the marks.

Example 8: Find the Closest Match

Let's get a little advanced now.

Suppose you have a dataset where you want to find the person who has the work experience closest to the required experience (mentioned in cell D2).

	A	B	C	D	E
1	Name	Experience (Years)		Required Experience	Right Person for the Job
2	John	9.3		2.5	??
3	Tom	1.7			
4	Arjun	22			
5	Greg	24.3			
6	Martha	17			
7	Xi	13.7			
8	Jenny	2.2			
9	Bob	11.7			
10	Marie	4.5			
11	Amy	12.3			
12	Charlie	17.3			
13	Sam	2.1			
14	Bruce	26.3			
15	Jamie	5.4			

While lookup formulas are not made to do this, you can combine it with other functions (such as MIN and ABS) to get this done.

Below is the formula that will find the person with the experience closest to the required one and return the name of the person. Note that the experience needs to be closest (which can be either less or more).

```
=INDEX($A$2:$A$15,MATCH(MIN(ABS(D2-B2:B15)),ABS(D2-$B$2:$B$15),0))
```

Since this is an array formula, you need to use Control + Shift + Enter.

	A	B	C	D	E
1	Name	Experience (Years)		Required Experience	Right Person for the Job
2	John	9.3			
3	Tom	1.7			
4	Arjun	22			
5	Greg	24.3			
6	Martha	17			
7	Xi	13.7			
8	Jenny	2.2			
9	Bob	11.7			
10	Marie	4.5			
11	Amy	12.3			
12	Charlie	17.3			
13	Sam	2.1			
14	Bruce	26.3			
15	Jamie	5.4			

The trick in this formula is to change the lookup value and lookup array to find the minimum experience difference in required and actual values.

Before I explain the formula, let's understand how you would do it manually.

You will go through each cell in column B and find the difference in the experience between what is required and the one that a person has. Once you have all the differences, you will find the one which is minimum and fetch the name of that person.

This is exactly what we are doing with this formula.

Let me explain.

The lookup value in the MATCH formula is MIN(ABS(D2-B2:B15)).

This part gives you the minimum difference between the given experience (which is 2.5 years) and all the other experiences. In this example, it returns 0.3

Note that I have used ABS to make sure I am looking for the closest (which can be more or less than the given experience).

Now, this minimum value becomes our lookup value.

The lookup array in the MATCH function is ABS(D2-\$B\$2:\$B\$15).

This gives us an array of numbers from which 2.5 (the required experience) has been subtracted.

So now we have a lookup value (0.3) and a lookup array

{6.8;0.8;19.5;21.8;14.5;11.2;0.3;9.2;2;9.8;14.8;0.4;23.8;2.9}

MATCH function finds the position of 0.3 in this array, which is also the position of the person's name who has the closest experience.

This position number is then used by the INDEX function to return the name of the person.

Related Read: Find the Closest Match in Excel (examples using lookup formulas)

Example 9: Use INDEX MATCH with Wildcard Characters

If you want to look up a value when there is a partial match, then you need to use wildcard characters.

For example, below is a dataset of company name and their market capitalizations and you want to want to get the market cap. data for the three companies on the right.

	A	B	C	D	E
1	Name	Mkt Cap		Name	Mkt Cap
2	Apple Inc.	926.9		Apple	?
3	Amazon.com, Inc.	777.8		Microsoft	?
4	Alphabet Alphabet Inc.	766.4		Alibaba	?
5	Microsoft Corporation	750.6			
6	Facebook Inc.	541.5			
7	Alibaba Group Holding Limited	499.4			
8	Berkshire Hathaway Inc	491.9			
9	Tencent Holdings Limited	491.3			
10	J.P. Morgan Chase & Co.	387.7			

Since these are not exact matches, you can't do a regular lookup in this case.

But you can still get the right data by using an asterisk (*), which is a wildcard character.

Below is the formula that will give you the data by matching the company names from the main column and fetching the market cap figure for it.

```
=INDEX($B$2:$B$10,MATCH(D2&"*",$A$2:$A$10,0),1)
```

	A	B	C	D	E	F	G
1	Name	Mkt Cap		Name	Mkt Cap		
2	Apple Inc.	926.9		Apple	926.9		
3	Amazon.com, Inc.	777.8		Microsoft	750.6		
4	Alphabet Alphabet Inc.	766.4		Alibaba	499.4		
5	Microsoft Corporation	750.6					
6	Facebook Inc.	541.5					
7	Alibaba Group Holding Limited	499.4					
8	Berkshire Hathaway Inc	491.9					
9	Tencent Holdings Limited	491.3					
10	J.P. Morgan Chase & Co.	387.7					

Let me explain how this formula works.

Since there is no exact match of lookup values, I have used D2&"*" as the lookup value in the MATCH function.

An asterisk is a wildcard character that represents any number of characters. This means that Apple* in the formula would mean any text string that starts with the word Apple and can have any number of characters after it.

So when Apple* is used as the lookup value and the MATCH formula looks for it in column A, it returns the position of 'Apple Inc.', as it starts with the word Apple.

You can also use wildcard characters to find text strings where the lookup value is in between. For example, if you use *Apple* as the lookup value, it will find any string that has the word apple anywhere in it.

Note: This technique works well when you only have one instance of matching. But if you have multiple instances of matching (for example Apple Inc and Apple Corporation, then the MATCH function would return the position of the first matching instance only.

Example 10: Three Way Lookup

This is an advanced use of INDEX MATCH, but I will still cover it to show you the power of this combination.

Remember I said that INDEX function has two syntaxes:

```
=INDEX (array, row_num, [col_num])  
=INDEX (array, row_num, [col_num], [area_num])
```

So far in all our examples, we have only used the first one.

But for a three-way lookup, you need to use the second syntax.

Let me first explain what a three-way look means.

In a two-way lookup, we use the INDEX MATCH formula to get the marks when we have the student's name and the subject name. For example, fetching the marks of Jim in Math is a two-way lookup.

A three-way look would add another dimension to it. For example, suppose you have a dataset as shown below and you want to know the score of Jim in Math in Mid-term exam, then this would be three-way lookup.

	A	B	C	D	E	F	G	H	I
1	Unit Test								
2	Name	Math	Physics	Chemistry		Unit Test	Mid Term	Final	
3	John	71	53	89					
4	Jenny	88	45	55					
5	Mark	44	28	54					
6	Jim	97	70	73					
7	Michael	47	34	38					
8									
9	Mid Term					Math	Math	Physics	
10	Name	Math	Physics	Chemistry		Jim	?	?	?
11	John	79	69	63					
12	Jenny	47	71	70					
13	Mark	95	63	37					
14	Jim	98	89	41					
15	Michael	71	60	88					
16									
17	Final								
18	Name	Math	Physics	Chemistry					
19	John	70	44	66					
20	Jenny	51	56	71					
21	Mark	51	93	55					
22	Jim	46	40	64					
23	Michael	50	90	85					

Below is the formula that will give the result.

```
=INDEX($B$3:$D$7,$B$11:$D$15,$B$19:$D$23),MATCH($F$5,$A$3:$A$7,0),MATCH(G$4,$B$2:$D$2,0),(IF(G$3="Unit Test",1,IF(G$3="Mid Term",2,3))))
```

	A	B	C	D	E	F	G	H	I
1	Unit Test								
2	Name	Math	Physics	Chemistry			Unit Test	Mid Term	Final
3	John	71	53	89			Math	Math	Physics
4	Jenny	88	45	55					
5	Mark	44	28	54		Jim	97	98	40
6	Jim	97	70	73					
7	Michael	47	34	38					
8									
9	Mid Term								
10	Name	Math	Physics	Chemistry					
11	John	79	69	63					
12	Jenny	47	71	70					
13	Mark	95	63	37					
14	Jim	98	89	41					
15	Michael	71	60	88					
16									
17	Final								
18	Name	Math	Physics	Chemistry					
19	John	70	44	66					
20	Jenny	51	56	71					
21	Mark	51	93	55					
22	Jim	46	40	64					
23	Michael	50	90	85					

The above formula checked for three things – the name of the student, the subject, and the exam.

After it finds the right value, it returns it in the cell.

Let me explain how this formula works by breaking down the formula into parts.

- array – (\$B\$3:\$D\$7,\$B\$11:\$D\$15,\$B\$19:\$D\$23): Instead of using a single array, in this case, I have used three arrays within parenthesis.
- row_num – MATCH(\$F\$5,\$A\$3:\$A\$7,0): MATCH function is used to find the position of the student's name in cell \$F\$5 in the list of student's name.
- col_num – MATCH(G\$4,\$B\$2:\$D\$2,0): MATCH function is used to find the position of the subject name in cell \$B\$2 in the list of subject's name.
- [area_num] – IF(G\$3="Unit Test",1,IF(G\$3="Mid Term",2,3)): The area number value tells the INDEX function which of the three arrays to use to fetch the value. If the exam is Unit Term, the IF function would return 1 and the INDEX function would use the first array to fetch the value. If the exam is Mid-term, the IF formula would return 2, else it will return 3.

This is an advanced example of using INDEX MATCH, and you're unlikely to find a situation when you have to use this. But it's still good to know what Excel formulas can do.

Why is INDEX/MATCH Better than VLOOKUP?

OR Is it?

Yes, it is – in most cases.

But before I do that, let me say this – VLOOKUP is an extremely useful function and I love it. It can do a lot of things in Excel and I use it every now and then myself. Having said that, it doesn't mean that there can't be anything better, and INDEX/MATCH (with more flexibility and functionalities) is better.

So if you want to do some basic lookup, you're better off using VLOOKUP.

INDEX/MATCH is VLOOKUP on steroids. And once you learn INDEX/MATCH, you might always prefer using it (especially because of the flexibility it has).

Without stretching it too far, let me quickly give you the reasons why INDEX/MATCH is better than VLOOKUP.

INDEX/MATCH can look to the Left (as well as to the right) of the lookup value

I covered it in one of the example above.

If you have a value which is on the left of the lookup value, you can't do that with VLOOKUP

At least not with just VLOOKUP.

Yes, you can combine VLOOKUP with other formulas and get it done, but it gets complicated and messy.

INDEX/MATCH, on the other hand, is made to lookup everywhere (be it left, right, up, or down)

INDEX/MATCH can work with vertical and horizontal ranges

Again, with full respect to VLOOKUP, it's not made to do this.

After all, the V in VLOOKUP stands for vertical.

VLOOKUP can only go through data that is vertical, while INDEX/MATCH can go through data vertically as well horizontally.

Of course, there is the HLOOKUP function to take care of horizontal lookup, but it isn't VLOOKUP then.. right?

I like the fact that INDEX MATCH combo is flexible enough to work with both vertical and horizontal data.

VLOOKUP cannot work with descending data

When it comes to the approximate match, VLOOKUP and INDEX/MATCH are at the same level.

But INDEX MATCH takes the point as it can also handle data that is in descending order.

I show this in one of the examples in this tutorial where we have to find the grade of students based on the grading table. If the table is sorted in descending order, VLOOKUP would not work (but INDEX MATCH would).

INDEX/MATCH can be slightly faster

The difference in speed in VLOOKUP and INDEX/MATCH is hardly noticeable when you have small data sets. But if you have thousands of rows and many columns, this can be a deciding factor.

INDEX/MATCH is Independent of the Actual Column Position

If you have a dataset as shown below as you're fetching the score of Jim in Physics, you can do that using VLOOKUP.

	A	B	C	D
1	Name	Math	Physics	Chemistry
2	John	71	53	89
3	Jenny	88	45	55
4	Mark	44	28	54
5	Jim	97	70	73
6	Michael	47	34	38
7	Hannah	88	75	90
8	Kane	87	42	54
9	Alan	27	37	56
10	Angela	35	89	57
11	Chris	67	54	82

And to do that, you can specify the column number as 3 in VLOOKUP.

All is fine.

But what if I delete the Math column.

In that case, the VLOOKUP formula will break.

Why? – Because it was hardcoded to use the third column, and when I delete a column in between, the third column becomes the second column.

Using INDEX/MATCH, in this case, is better as you can make the column number dynamic by using MATCH. So instead of a column number, it checks for the subject name and uses that to return the column number.

Surely you can do that by combining VLOOKUP with MATCH, but if you combining anyway, why not do it with INDEX which is a lot more flexible.

When using INDEX/MATCH, you can safely insert/delete columns in your dataset.

Despite all these factors, there is a reason VLOOKUP is so popular.

And it's a big reason.

VLOOKUP is easier to use

VLOOKUP only takes a maximum of four arguments. If you can wrap your head around these four, you're good to go.

And since most of the basic lookup cases are handled by VLOOKUP as well, it has quickly become the most popular Excel function.

INDEX/MATCH, on the other hand, is a little more difficult to use. You may get a hang if it when you start using it, but for a beginner, VLOOKUP is far more easy to explain and learn.

And this is not a zero-sum game.

So, if you're new to the lookup world and don't know how to use VLOOKUP, better learn that.

Bonus Section

Lookup multiple criteria in rows or columns

Some reports involve the need to find a value from a source table using multiple criteria in rows and columns. In this example, we have a table containing both the actual and budget revenues and profits for each application as shown below:

Division	Apps	Actual	Actual	Budget	Budget
		Revenue	Profit	Revenue	Profit
Game	Fightrr	11,649	802	10,593	554
Game	Kryptis	7,718	876	6,409	654
Game	Perino	15,033	469	12,724	530
Game	Hackrr	18,701	985	19,102	1,302
Productivity	WenCaL	14,432	240	15,113	363
Productivity	Blend	17,990	1,166	18,181	1,223
Productivity	Sleops	11,022	550	13,112	474
Utility	Accord	17,760	800	16,854	572
Utility	Misty Wash	30,400	787	30,237	932
Utility	Twenty20	20,400	614	18,477	1,120

From this data, you need to create a report that returns the value corresponding to three criteria that the user selects:

- Actual or Budget
- Revenue or Profit
- App name

Select KPI:	Actual	
	Profit	
WenCaL		

This becomes the matrix lookup, however, unlike the regular cases, this has more than one header (Row 20: Actual or Budget, Row 21: Revenue or Profit).

	A	B	C	D	E	F
17	More Complex					
18						
19						
20			Actual	Actual	Budget	Budget
21	Division	Apps	Revenue	Profit	Revenue	Profit
22	Game	Fightrr	11,649	802	10,593	554
23	Game	Kryptis	7,718	876	6,409	654
24	Game	Perino	15,033	469	12,724	530
25	Game	Hackrr	18,701	985	19,102	1,302
26	Productivity	WenCaL	14,432	240	15,113	363
27	Productivity	Blend	17,990	1,166	18,181	1,223
28	Productivity	Sleops	11,022	550	13,112	474
29	Utility	Accord	17,760	800	16,854	572
30	Utility	Misty Wash	30,400	787	30,237	932
31	Utility	Twenty20	20,400	614	18,477	1,120

Method 1: Using helper cells

Since the MATCH() function can only handle single rows and columns, the simplest way to resolve the problem we had earlier is to use helper cells that combine the values in rows 20 and 21 into one row instead of two. This combination becomes the unique identifier of each column.

Cell C19 = C20&C21

You will see that C19 now contains ActualRevenue. Drag this formula to the right until cell F19.

	A	B	C	D	E	F
17	More Complex					
18			ActualRevenue	ActualProfit	BudgetRevenue	BudgetProfit
19			Actual	Actual	Budget	Budget
20			Revenue	Profit	Revenue	Profit
21	Division	Apps	Revenue	Profit	Revenue	Profit
22	Game	Fightrr	11,649	802	10,593	554
23	Game	Kryptis	7,718	876	6,409	654
24	Game	Perino	15,033	469	12,724	530
25	Game	Hackrr	18,701	985	19,102	1,302
26	Productivity	WenCaL	14,432	240	15,113	363
27	Productivity	Blend	17,990	1,166	18,181	1,223
28	Productivity	Sleops	11,022	550	13,112	474
29	Utility	Accord	17,760	800	16,854	572
30	Utility	Misty Wash	30,400	787	30,237	932
31	Utility	Twenty20	20,400	614	18,477	1,120

Go back to the last MATCH() function that used cells I20 and I21 as the lookup value. Instead of using C20:F21 as the lookup array, you can now use the new helper cells C19:F19.

Cell I22 = INDEX(C22:F31,MATCH(H22,B22:B31,0),MATCH(I20&I21,C19:F19,0))

This formula displays the value that corresponds to the selected criteria in cells I20, I21 and H22.

Method 2: Using CTRL + SHIFT + ENTER (CSE)

There is another approach that eliminates the use of helper cells. This involves generating an array for the MATCH() function by pressing the keys CTRL + SHIFT + ENTER (CSE).

The MATCH() function is not meant to handle array functions, rather, it looks at things one cell at a time instead of holding things in memory and handle them. Some functions that can handle arrays are:

INDEX()

SUMPRODUCT()

VLOOKUP()

HLOOKUP()

AGGREGATE()

However, clicking on CSE on a MATCH() function enables it to handle arrays. Write the same function as above, only instead of using the helper cells, revise the last MATCH() function to combine cells C20:F20 and C21:F21 using the & symbol.

Cell I26 = INDEX(C22:F31,MATCH(H22,B22:B31,0),MATCH(I20&I21,C20:F20&C21:F21,0))

After pressing **ENTER**, you will notice that it results to an error because there is no instruction telling the **MATCH()** function to keep the values in memory. To see the step-by-step calculation of the function resulting to the error:

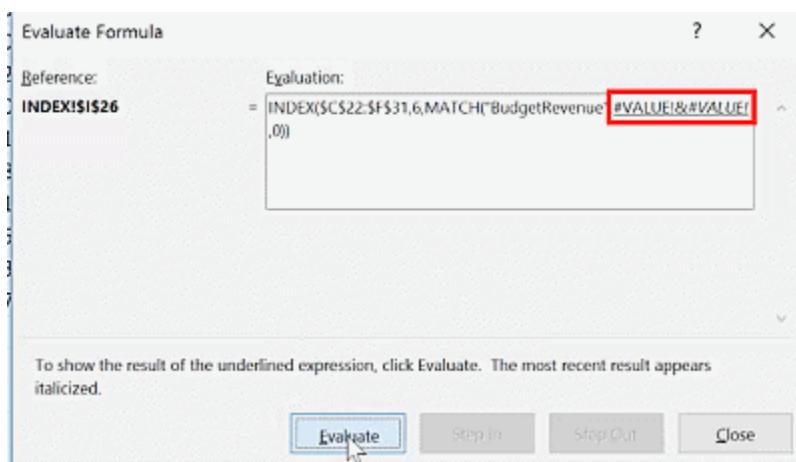
Click on cell I26.

Go to the **FORMULAS**

Click on **EVALUATE FORMULA**.

Division	App	Actual Revenue	Actual Profit	Budget Revenue	Budget Profit	Select KPI:	Budget Revenue
Game	Riottr	11,649	803	10,593	554	Blend	\$8383
Game	Kryptik	7,738	876	6,409	694		
Game	Perino	15,033	469	12,724	530		
Game	Hacker	18,701	985	19,102	1,302	Array Solution with CSE	#VALUE!
Productivity	WrenGal	14,412	240	15,113	363		
Productivity	Blind	17,990	1,166	18,181	1,223	Array Solution without CSE	
Sleeps		11,011	550	13,112	474		
Utility	Accord	17,160	800	16,854	542		
Utility	Moxy Wash	30,400	787	30,237	932		
Utility	Twenty29	20,406	614	18,477	3,120		

You will see the function written on the white space. Each click on the **EVALUATE** button will show you the calculation step-by-step. It first finds the value of cell H22, and finds the row containing that App, followed by finding the values in cells I20 and I21 and combining them. You will notice that the value returned by C20:F20 and C21:F21 are **#VALUE!** The function notices that you are trying to combine things and it is confused because it does not know how to handle such instances.



Click on **CLOSE** to exit the window. Go back to cell I26. Click on the function in the formula bar and instead of pressing the **ENTER** key, click **CTRL + SHIFT + ENTER**.

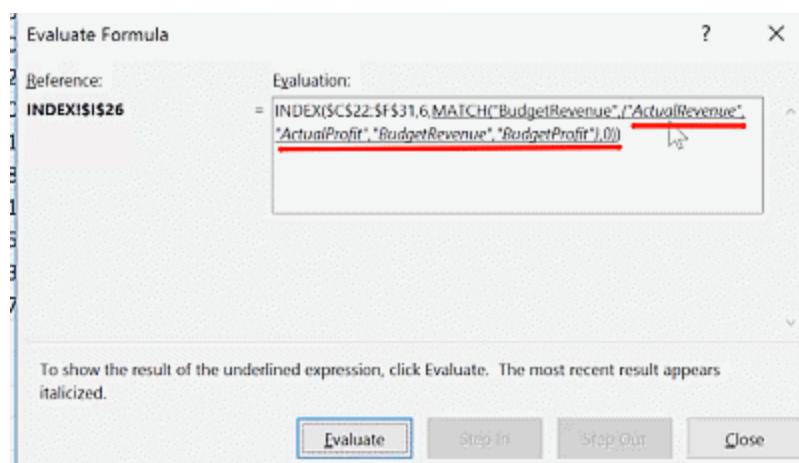
Actual Revenue	Actual Profit	Budget Revenue	Budget Profit	Select KPI:	Budget Revenue
Actual Revenue	Actual Profit	Budget Revenue	Budget Profit	Blend	18181
11,649	802	10,593	554		
7,718	876	6,409	654		
15,033	469	12,724	530		
18,701	985	19,102	1,302	Array Solution with CSE	18181
14,432	240	15,113	363		
17,990	1,166	18,181	1,223	Array Solution without CSE	
11,022	550	13,112	474		
17,760	800	16,854	572		
30,400	787	30,237	932		
20,400	614	18,477	1,120		

Notice that it places {} at the start and end of the function and becomes:

Cell I26 {=INDEX(C22:F31,MATCH(H22,B22:B31,0),MATCH(I20&I21,C20:F20&C21:F21,0))}

It now displays the value that corresponds to the three criteria selected.

Click on cell I26 again and go to **EVALUATE FORMULA** to see how this differs from the previous one. Instead of returning #VALUE! for both C20:F20 and C21:F21, it is now able to find the values in those cells and combine them. The {} gives an instruction to the MATCH() function to keep the values in memory, which makes it easier for it to take C20 and combine it with C21 to give ActualRevenue, D20 with D21 to give ActualProfit, and so on.



It then finds the column number that corresponds to the criteria, “BudgetRevenue”, which is 3. The INDEX() function is now able to find the value using the numbers from the row and column arguments.

This approach is a simple way of writing but it can be confusing for a lot of people. It is only suitable when you are writing the formulas for your own use. Otherwise, if you have other users, it is very likely that those users do not know array functions. They might see the array, click on the formula, inspect it a bit, and press **ENTER**. Since they press **ENTER** instead of **CSE**, the formula will then result to an error again. In such cases, it would be best to avoid this approach.

Method 3: Using two INDEX() functions

The third approach does not require helper cells nor CSE, but replaces the last MATCH() function with an INDIRECT() function.

Start off using the same functions for the most part except for the last MATCH() function:

Cell I29 = INDEX(C22:F31,MATCH(H22,B22:B31,0),MATCH(I20&I21...)

In order to avoid the use of CSE, we need to use a function that can handle arrays. The INDEX() function can be used by putting the entire lookup area inside the INDEX() function. To do this, replace the last MATCH() function in Method 2 with:

INDEX(C20&F20&C21:F21,...

It now becomes:

Cell I29 = INDEX(C22:F31,MATCH(H22,B22:B31,0),MATCH(I20&I21,INDEX(C20:F20&C21:F21...)

It is important to comply with the syntax of the INDEX() function. The `row_num` argument is mandatory. Instead of leaving the formula up until C21:F21, you need to specify a row number. In this case, you want to tell the function to take every single row. There are two ways to do that:

Use the Excel separator (,) and leave it empty

Cell I29 = INDEX(C22:F31,MATCH(H22,B22:B31,0),MATCH(I20&I21,INDEX(C20:F20&C21:F21,...

Use 0 as the row number

Cell I29 = INDEX(C22:F31,MATCH(H22,B22:B31,0),MATCH(I20&I21,INDEX(C20:F20&C21:F21,0)...

Next, comply with the syntax of the MATCH() function where you need to specify the match type.

Again, use 0 for an exact match. The final formula becomes:

Cell I29 = INDEX(C22:F31,MATCH(H22,B22:B31,0),MATCH(I20&I21,INDEX(C20:F20&C21:F21,0),0))

The screenshot shows an Excel spreadsheet with a formula bar containing the formula `=INDEX(C22:F31,MATCH(H22,B22:B31,0),MATCH(I20&I21,INDEX(C20:F20&C21:F21,0),0))`. The status bar at the bottom right indicates "Array Solution with CSE" and "18181". The spreadsheet has a header row and data starting from row 22. A dropdown menu is open over the formula bar, showing options like "Select KPI:", "Budget", "Revenue", "Blend", and "Actual".

		Actual	Revenue	Actual	Profit	Budget	Revenue	Budget	Profit
	Division	Apps	Revenue	Profit	Budget	Revenue	Budget	Revenue	Budget
22	Game	Fighter	11,649	802	10,593	554			
23	Game	Kryptis	7,718	876	6,409	654			
24	Game	Perino	15,033	469	12,724	530			
25	Game	Hackrr	18,701	985	19,102	1,302			
26	Productivity	WenCaL	14,432	240	15,113	363			
27	Productivity	Blend	17,990	1,166	18,181	1,223			
28	Productivity	Sleeps	11,022	550	13,112	474			
29	Utility	Accord	17,760	800	16,854	572			
30	Utility	Misty Wash	30,400	787	30,237	932			
31	Utility	Twenty20	20,400	614	18,477	1,120			

This gives you the same result as the first two methods. There are other ways to resolve this problem but I find these three to be the simplest ways.