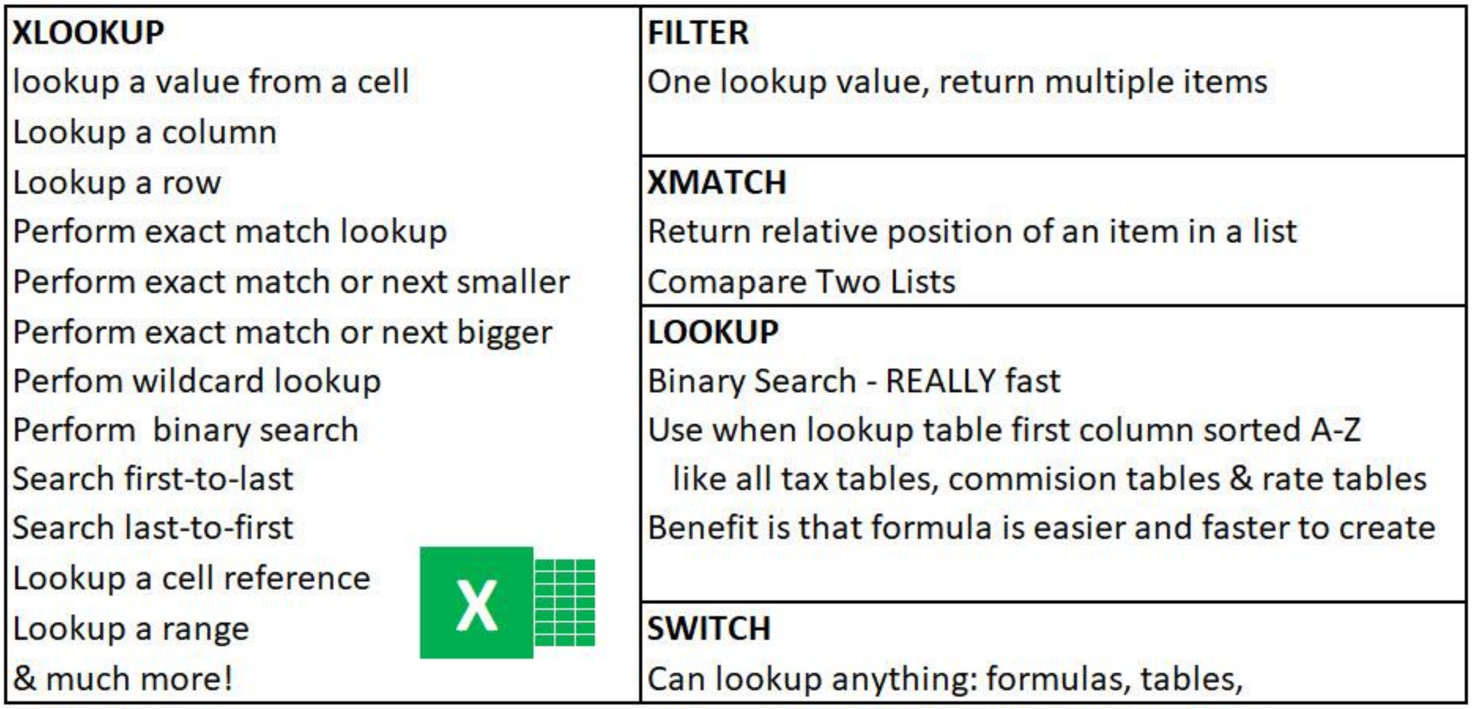
MS 365 Excel Basics 06: Mastering Excel Lookup: XLOOKUP, FILTER, XMATCH, LOOKUP, SWITCH Functions!

Table of Contents

Important LOOKUP Functions 2
You must know Excel well before using AI........ 4
Example of AI provided formulas and then better formulas that are not provided: 5
XLOOKUP function arguments........ 6
No duplicates in lookup_array argument........ 7
Duplicates in lookup_array argument........ 8
Exact match lookup 9
Exact match lookup to a lookup product price........ 9
Data Validation dropdown list........10
Lookup an employee record 11
Lookup a column of day's sales and then add 11
Exact match or next smaller lookup 12
Lookup a tax rate: XLOOKUP or LOOKUP function?........13
LOOKUP function arguments........14
Why the Binary Search in the LOOKUP function is fast 15
Timing Results for XLOOKUP and LOOKUP functions 16
How XLOOKUP Binary may yield different answers than LOOKUP Binary when there are duplicates?........17
How can XLOOKUP yield same answer as LOOKUP, even with duplicate values? 18
When to use LOOKUP function rather than XLOOKUP function?........19
XLOOKUP can do vertical or horizontal lookup 19
XLOOKUP to perform Exact Match or Next Bigger lookup: lookup pipe size 19
XLOOKUP with wild card lookup: "Coca Cola" matches "Coca Cola Inc." 20
Lookup first when there are duplicates: lookup first day worked........21
Lookup last when there are duplicates: lookup last sale for employee........21
Two lookup values: lookup product price for a specified city........22
Two-Way lookup: lookup tax based on weekly pay and allowances........23
FILTER array function arguments 24
FILTER function to return multiple items from one lookup value 26
XMATCH function arguments........27
XMATCH lookup function to compare two lists 27
SWITCH function arguments 28
SWITCH to lookup the correct pricing lookup table based on product name 29
LOOKUP function to create formula much more quickly for tax calculations........30

Important LOOKUP Functions



XLOOKUP

lookup a value from a cell
Lookup a column
Lookup a row
Perform exact match lookup
Perform exact match or next smaller
Perform exact match or next bigger
Perfom wildcard lookup
Perform binary search
Search first-to-last
Search last-to-first
Lookup a cell reference
Lookup a range
& much more!

FILTER

One lookup value, return multiple items

XMATCH

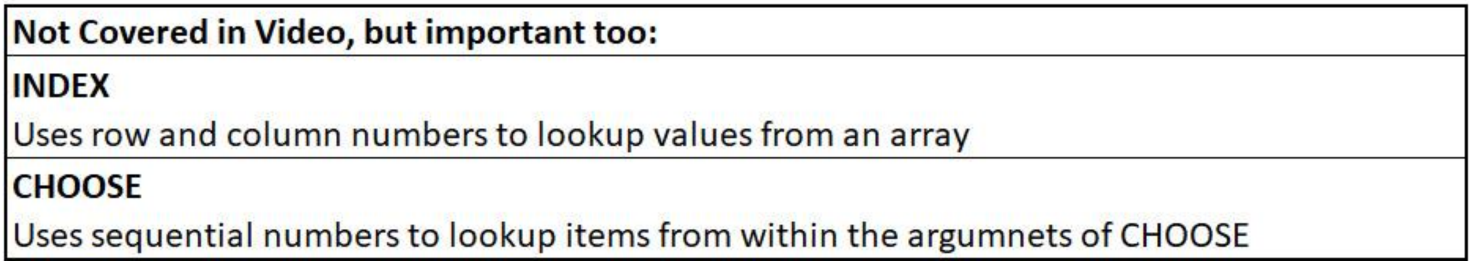
Return relative position of an item in a list
Comapare Two Lists

LOOKUP

Binary Search - REALLY fast
Use when lookup table first column sorted A-Z
like all tax tables, commision tables & rate tables
Benefit is that formula is easier and faster to create

SWITCH

Can lookup anything: formulas, tables,



Not Covered in Video, but important too:

INDEX

Uses row and column numbers to lookup values from an array

CHOOSE

Uses sequential numbers to lookup items from within the argumnets of CHOOSE

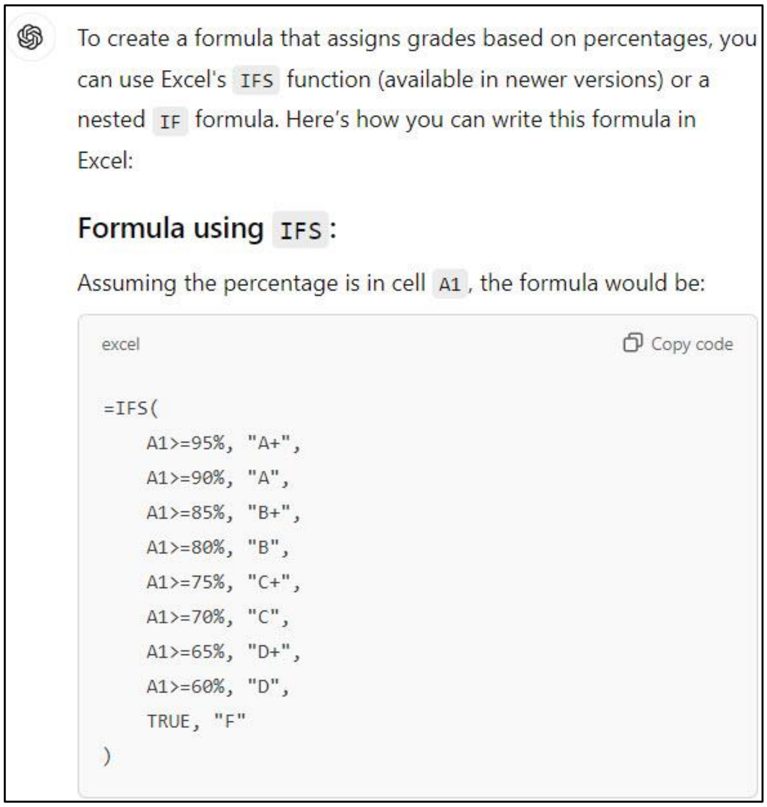
You must know Excel well before using AI

When you ask AI a question about building a solution in Excel, AI WILL give you an answer, but the answer may not always be a very good answer.

For example, if you ask a reasonable question like this:

> I would like an Excel formula that can assign an "A+" grade to 95% or higher, an "A" grade to less than 95% and greater than or equal to 90%, a "B+" grade to less than 90% and greater than or equal to 85%, a "B" grade to less than 85% and greater than or equal to 80%, a "C+" grade to less than 80% and greater than or equal to 75%, a "C" grade to less than 75% and greater than or equal to 70%, a "D+" grade to less than 70% and greater than or equal to 65%, a "D+" grade to less than 65% and greater than or equal to 60%,and a "F" grade to all other scores that are less than 60%.

This is what Chat GPT 4 returns as an answer:

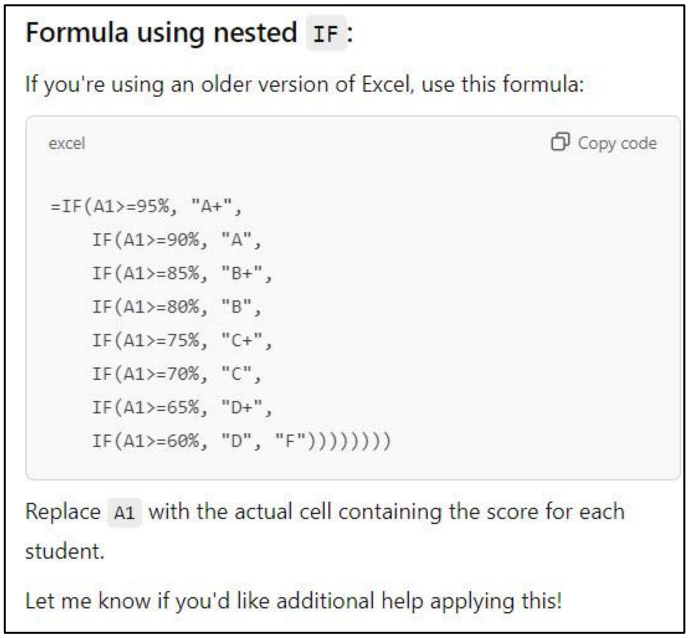


To create a formula that assigns grades based on percentages, you can use Excel's `IFS` function (available in newer versions) or a nested `IF` formula. Here's how you can write this formula in Excel:

Formula using `IFS`:

Assuming the percentage is in cell `A1`, the formula would be:

```
=IFS(
    A1>=95%, "A+",
    A1>=90%, "A",
    A1>=85%, "B+",
    A1>=80%, "B",
    A1>=75%, "C+",
    A1>=70%, "C",
    A1>=65%, "D+",
    A1>=60%, "D",
    TRUE, "F"
)
```



Formula using nested `IF`:

If you're using an older version of Excel, use this formula:

```
=IF(A1>=95%, "A+",
    IF(A1>=90%, "A",
    IF(A1>=85%, "B+",
    IF(A1>=80%, "B",
    IF(A1>=75%, "C+",
    IF(A1>=70%, "C",
    IF(A1>=65%, "D+",
    IF(A1>=60%, "D", "F"))))))))
```

Replace `A1` with the actual cell containing the score for each student.

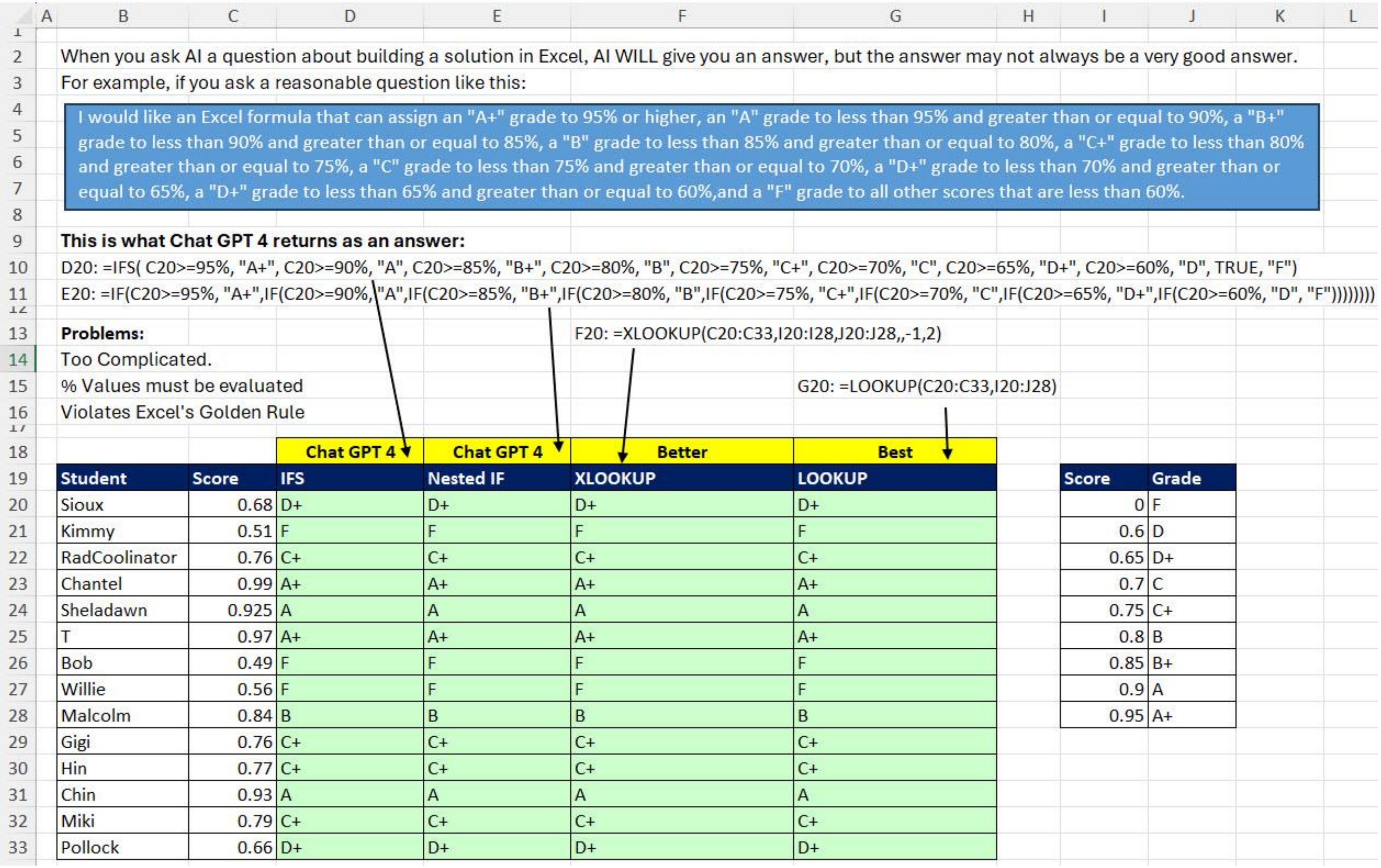
Let me know if you'd like additional help applying this!

The problems with this are:

1. Too Complicated.
2. % Values must be evaluated inside formula to get the decimal amount.
3. Violates Excel's Golden Rule: If a formula input can change, put it into a cell, label it and refer to it in the formula with a cell reference.

This means you must learn Excel well if you want to us AI to find Excel solutions – this way, when AI gives you a bad answer, you can decipher that and not use the answer.

Example of AI provided formulas and then better formulas that are not provided:



When you ask AI a question about building a solution in Excel, AI WILL give you an answer, but the answer may not always be a very good answer.

For example, if you ask a reasonable question like this:

I would like an Excel formula that can assign an "A+" grade to 95% or higher, an "A" grade to less than 95% and greater than or equal to 90%, a "B+" grade to less than 90% and greater than or equal to 85%, a "B" grade to less than 85% and greater than or equal to 80%, a "C+" grade to less than 80% and greater than or equal to 75%, a "C" grade to less than 75% and greater than or equal to 70%, a "D+" grade to less than 70% and greater than or equal to 65%, a "D+" grade to less than 65% and greater than or equal to 60%,and a "F" grade to all other scores that are less than 60%.

This is what Chat GPT 4 returns as an answer:

D20: =IFS(C20>=95%, "A+", C20>=90%, "A", C20>=85%, "B+", C20>=80%, "B", C20>=75%, "C+", C20>=70%, "C", C20>=65%, "D+", C20>=60%, "D", TRUE, "F")

E20: =IF(C20>=95%, "A+",IF(C20>=90%, "A",IF(C20>=85%, "B+",IF(C20>=80%, "B",IF(C20>=75%, "C+",IF(C20>=70%, "C",IF(C20>=65%, "D+",IF(C20>=60%, "D", "F"))))))))

Problems:

Too Complicated.

% Values must be evaluated

Violates Excel's Golden Rule

F20: =XLOOKUP(C20:C33,I20:I28,J20:J28,,-1,2)

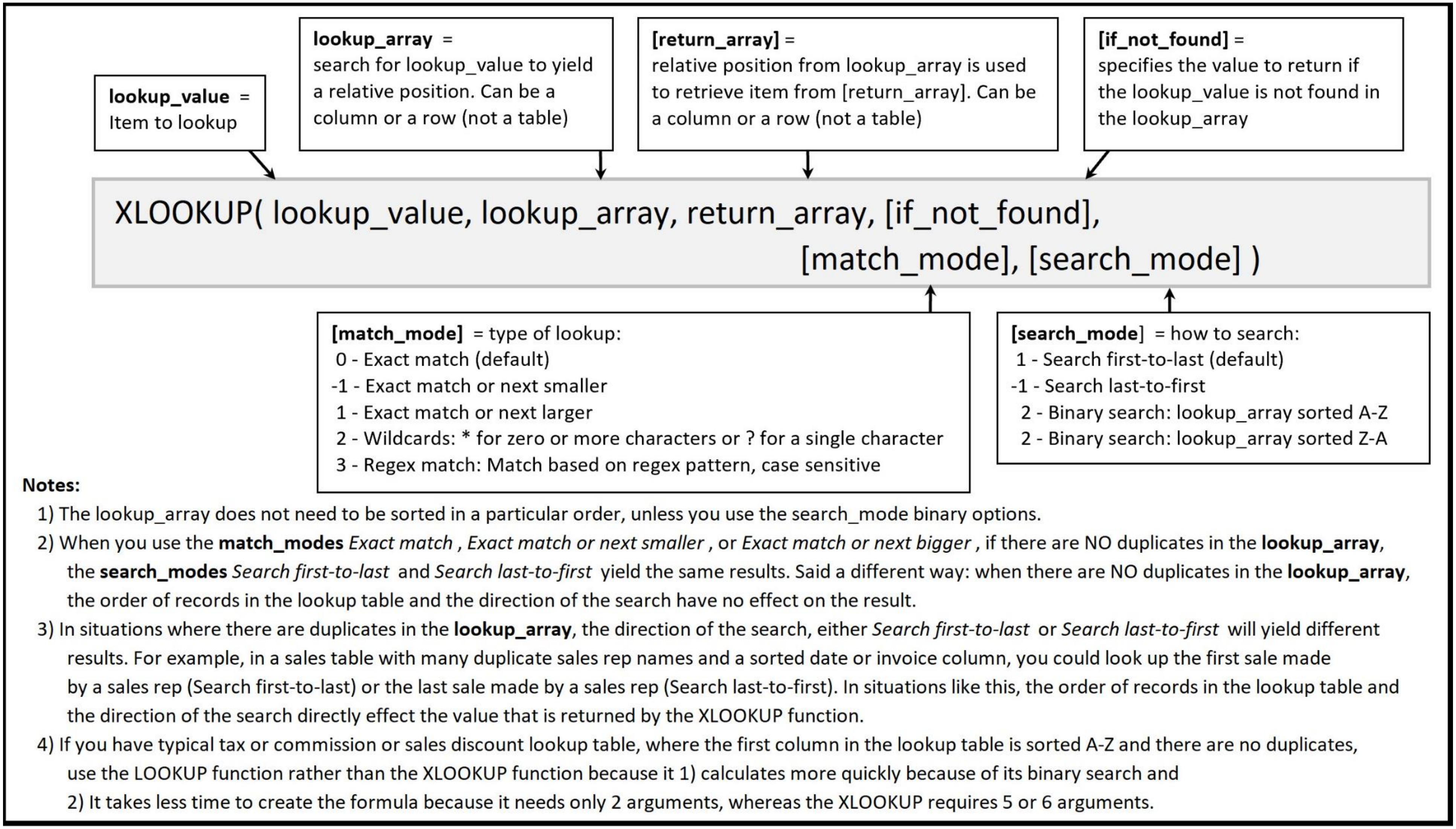
G20: =LOOKUP(C20:C33,I20:J28)

		Chat GPT 4	**Chat GPT 4**	**Better**	**Best**
Student	**Score**	**IFS**	**Nested IF**	**XLOOKUP**	**LOOKUP**
Sioux	0.68	D+	D+	D+	D+
Kimmy	0.51	F	F	F	F
RadCoolinator	0.76	C+	C+	C+	C+
Chantel	0.99	A+	A+	A+	A+
Sheladawn	0.925	A	A	A	A
T	0.97	A+	A+	A+	A+
Bob	0.49	F	F	F	F
Willie	0.56	F	F	F	F
Malcolm	0.84	B	B	B	B
Gigi	0.76	C+	C+	C+	C+
Hin	0.77	C+	C+	C+	C+
Chin	0.93	A	A	A	A
Miki	0.79	C+	C+	C+	C+
Pollock	0.66	D+	D+	D+	D+

Score	**Grade**
0	F
0.6	D
0.65	D+
0.7	C
0.75	C+
0.8	B
0.85	B+
0.9	A
0.95	A+

XLOOKUP function arguments

The XLOOKUP function was introduced in 2019 and has replaced older lookup functions such as VLOOKUP, HLOOKUP and INDEX. The XLOOKUP function can lookup a cell, a column, a row, a cell reference, a range, and it can also perform various types of lookup such as: exact match, exact match or next smaller, exact match or next bigger, wildcard lookup, binary search, search first-to-last, search last-to-first and other types of lookups. There are a few other lookup functions that we still use because they can do things that XLOOKUP cannot do, or do things more efficiently than XLOOKUP can, such as the lookup functions: FILTER, XMATCH, LOOKUP, and SWITCH. We will see examples of each of these lookup functions also.



lookup_value =
Item to lookup

lookup_array =
search for lookup_value to yield a relative position. Can be a column or a row (not a table)

[return_array] =
relative position from lookup_array is used to retrieve item from [return_array]. Can be a column or a row (not a table)

[if_not_found] =
specifies the value to return if the lookup_value is not found in the lookup_array

XLOOKUP(lookup_value, lookup_array, return_array, [if_not_found],
[match_mode], [search_mode])

[match_mode] = type of lookup:
0 - Exact match (default)
-1 - Exact match or next smaller
1 - Exact match or next larger
2 - Wildcards: * for zero or more characters or ? for a single character
3 - Regex match: Match based on regex pattern, case sensitive

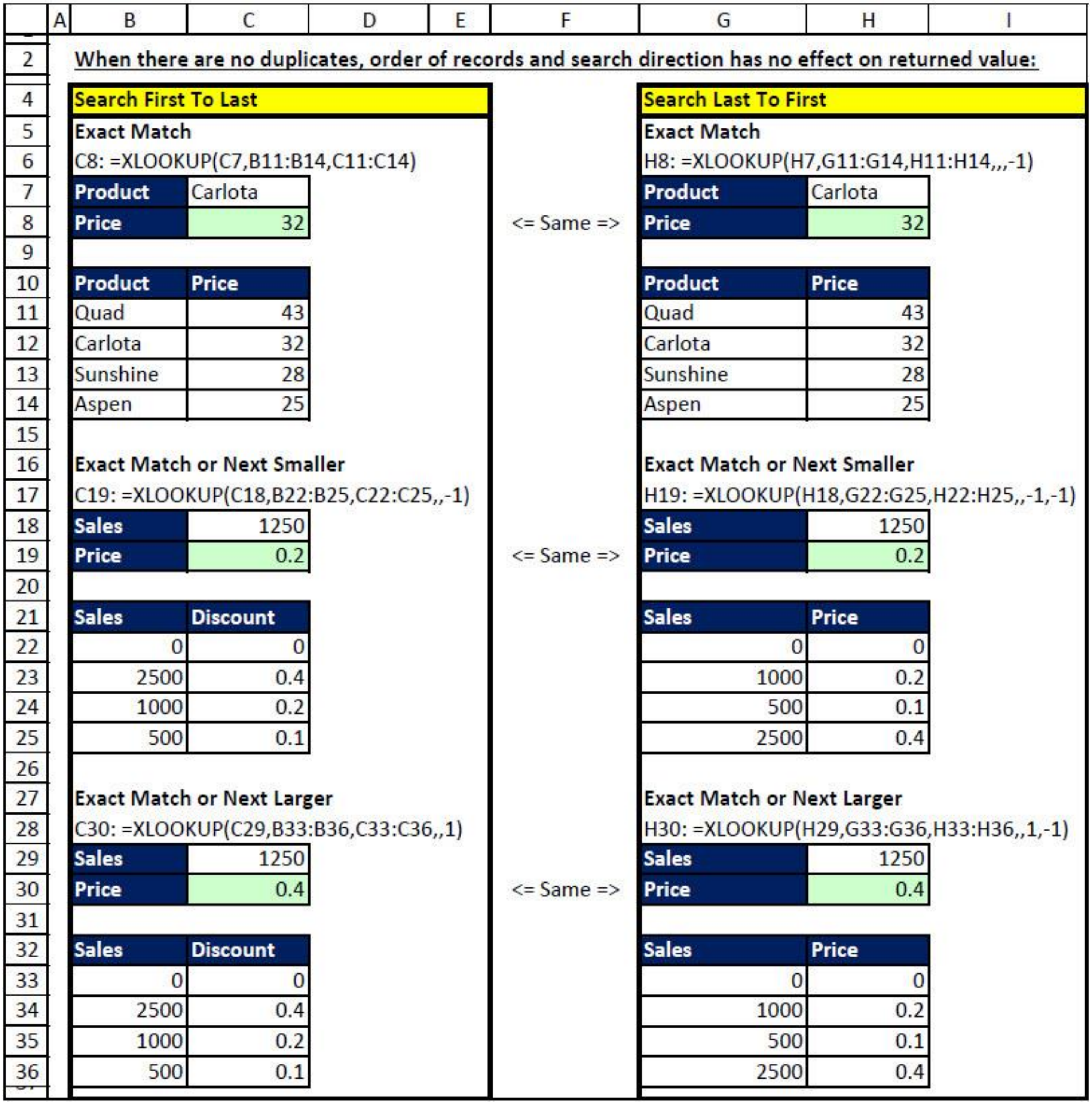
[search_mode] = how to search:
1 - Search first-to-last (default)
-1 - Search last-to-first
2 - Binary search: lookup_array sorted A-Z
2 - Binary search: lookup_array sorted Z-A

Notes:

1) The lookup_array does not need to be sorted in a particular order, unless you use the search_mode binary options.
2) When you use the **match_modes** *Exact match* , *Exact match or next smaller* , or *Exact match or next bigger* , if there are NO duplicates in the **lookup_array**, the **search_modes** *Search first-to-last* and *Search last-to-first* yield the same results. Said a different way: when there are NO duplicates in the **lookup_array**, the order of records in the lookup table and the direction of the search have no effect on the result.
3) In situations where there are duplicates in the **lookup_array**, the direction of the search, either *Search first-to-last* or *Search last-to-first* will yield different results. For example, in a sales table with many duplicate sales rep names and a sorted date or invoice column, you could look up the first sale made by a sales rep (Search first-to-last) or the last sale made by a sales rep (Search last-to-first). In situations like this, the order of records in the lookup table and the direction of the search directly effect the value that is returned by the XLOOKUP function.
4) If you have typical tax or commission or sales discount lookup table, where the first column in the lookup table is sorted A-Z and there are no duplicates, use the LOOKUP function rather than the XLOOKUP function because it 1) calculates more quickly because of its binary search and 2) It takes less time to create the formula because it needs only 2 arguments, whereas the XLOOKUP requires 5 or 6 arguments.

No duplicates in lookup_array argument

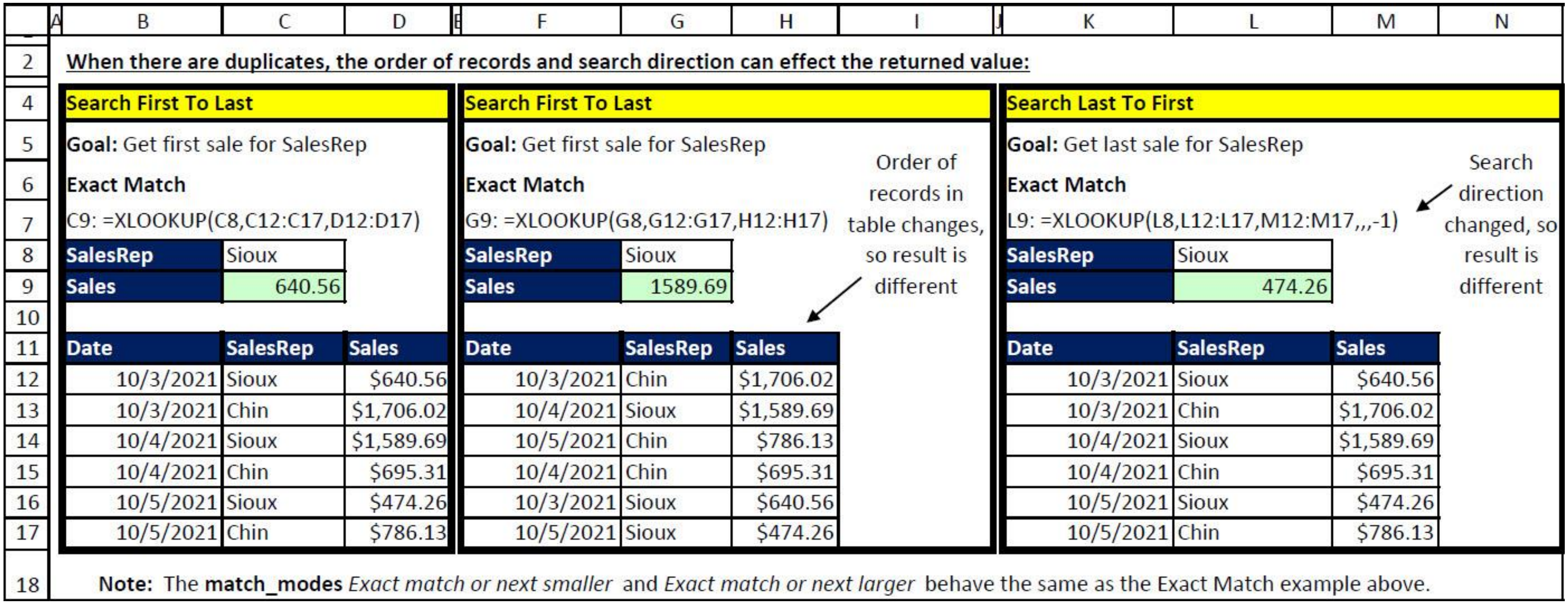
When there are no duplicates in the lookup_array argument , the order of records and search direction has no effect on returned value:



	A	B	C	D	E	F	G	H	I
2		**When there are no duplicates, order of records and search direction has no effect on returned value:**							
4		**Search First To Last**					**Search Last To First**		
5		**Exact Match**					**Exact Match**		
6		C8: =XLOOKUP(C7,B11:B14,C11:C14)					H8: =XLOOKUP(H7,G11:G14,H11:H14,,,-1)		
7		**Product**	Carlota				**Product**	Carlota	
8		**Price**	32			<= Same =>	**Price**	32	
9									
10		**Product**	**Price**				**Product**	**Price**	
11		Quad	43				Quad	43	
12		Carlota	32				Carlota	32	
13		Sunshine	28				Sunshine	28	
14		Aspen	25				Aspen	25	
15									
16		**Exact Match or Next Smaller**					**Exact Match or Next Smaller**		
17		C19: =XLOOKUP(C18,B22:B25,C22:C25,,-1)					H19: =XLOOKUP(H18,G22:G25,H22:H25,,-1,-1)		
18		**Sales**	1250				**Sales**	1250	
19		**Price**	0.2			<= Same =>	**Price**	0.2	
20									
21		**Sales**	**Discount**				**Sales**	**Price**	
22		0	0				0	0	
23		2500	0.4				1000	0.2	
24		1000	0.2				500	0.1	
25		500	0.1				2500	0.4	
26									
27		**Exact Match or Next Larger**					**Exact Match or Next Larger**		
28		C30: =XLOOKUP(C29,B33:B36,C33:C36,,1)					H30: =XLOOKUP(H29,G33:G36,H33:H36,,1,-1)		
29		**Sales**	1250				**Sales**	1250	
30		**Price**	0.4			<= Same =>	**Price**	0.4	
31									
32		**Sales**	**Discount**				**Sales**	**Price**	
33		0	0				0	0	
34		2500	0.4				1000	0.2	
35		1000	0.2				500	0.1	
36		500	0.1				2500	0.4	

Duplicates in lookup_array argument

When there are duplicates, the order of records and search direction can affect the returned value:



When there are duplicates, the order of records and search direction can effect the returned value:

Search First To Last

Goal: Get first sale for SalesRep

Exact Match

C9: =XLOOKUP(C8,C12:C17,D12:D17)

SalesRep	Sioux
Sales	640.56

Date	SalesRep	Sales
10/3/2021	Sioux	$640.56
10/3/2021	Chin	$1,706.02
10/4/2021	Sioux	$1,589.69
10/4/2021	Chin	$695.31
10/5/2021	Sioux	$474.26
10/5/2021	Chin	$786.13

Search First To Last

Goal: Get first sale for SalesRep

Exact Match

G9: =XLOOKUP(G8,G12:G17,H12:H17)

SalesRep	Sioux
Sales	1589.69

Order of records in table changes, so result is different

Date	SalesRep	Sales
10/3/2021	Chin	$1,706.02
10/4/2021	Sioux	$1,589.69
10/5/2021	Chin	$786.13
10/4/2021	Chin	$695.31
10/3/2021	Sioux	$640.56
10/5/2021	Sioux	$474.26

Search Last To First

Goal: Get last sale for SalesRep

Exact Match

L9: =XLOOKUP(L8,L12:L17,M12:M17,,,-1)

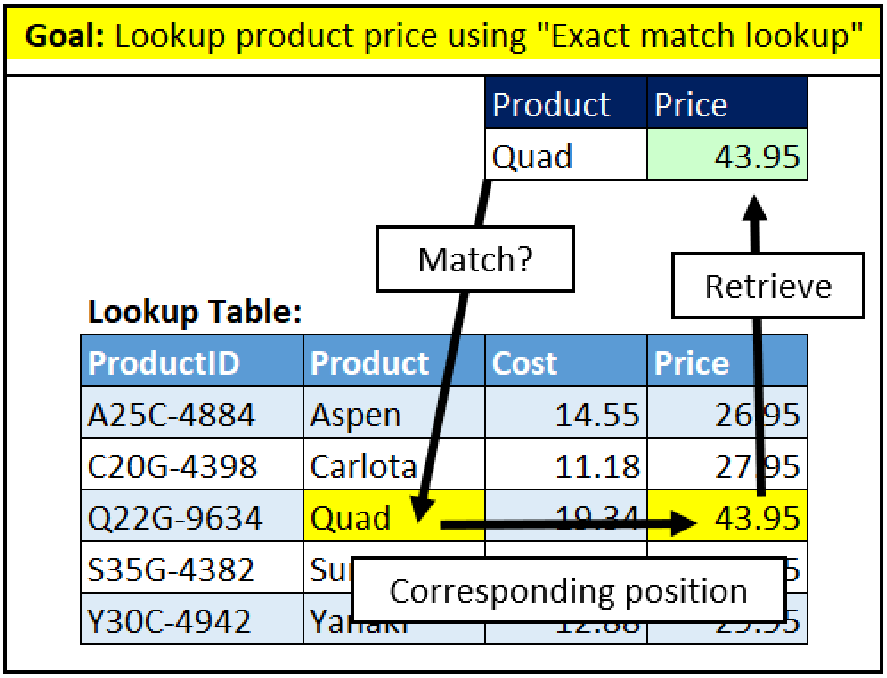
SalesRep	Sioux
Sales	474.26

Search direction changed, so result is different

Date	SalesRep	Sales
10/3/2021	Sioux	$640.56
10/3/2021	Chin	$1,706.02
10/4/2021	Sioux	$1,589.69
10/4/2021	Chin	$695.31
10/5/2021	Sioux	$474.26
10/5/2021	Chin	$786.13

Note: The **match_modes** *Exact match or next smaller* and *Exact match or next larger* behave the same as the Exact Match example above.

Exact match lookup



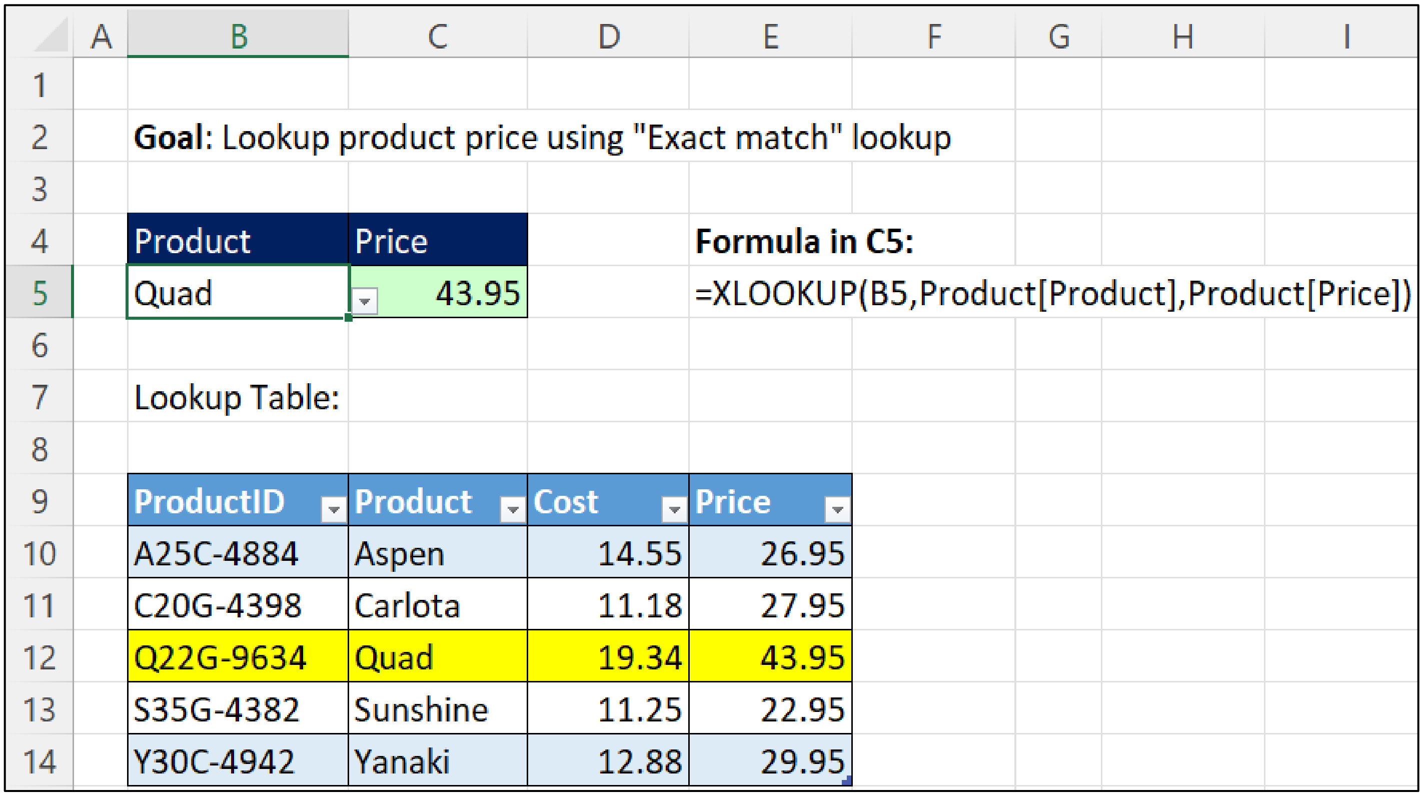
Goal: Lookup product price using "Exact match lookup"

Product	Price
Quad	43.95

Lookup Table:

ProductID	Product	Cost	Price
A25C-4884	Aspen	14.55	26.95
C20G-4398	Carlota	11.18	27.95
Q22G-9634	Quad	19.34	43.95
S35G-4382	Su		5
Y30C-4942	Ya		

Exact match lookup to a lookup product price



Goal: Lookup product price using "Exact match" lookup

Product	Price
Quad	43.95

Formula in C5:

=XLOOKUP(B5,Product[Product],Product[Price])

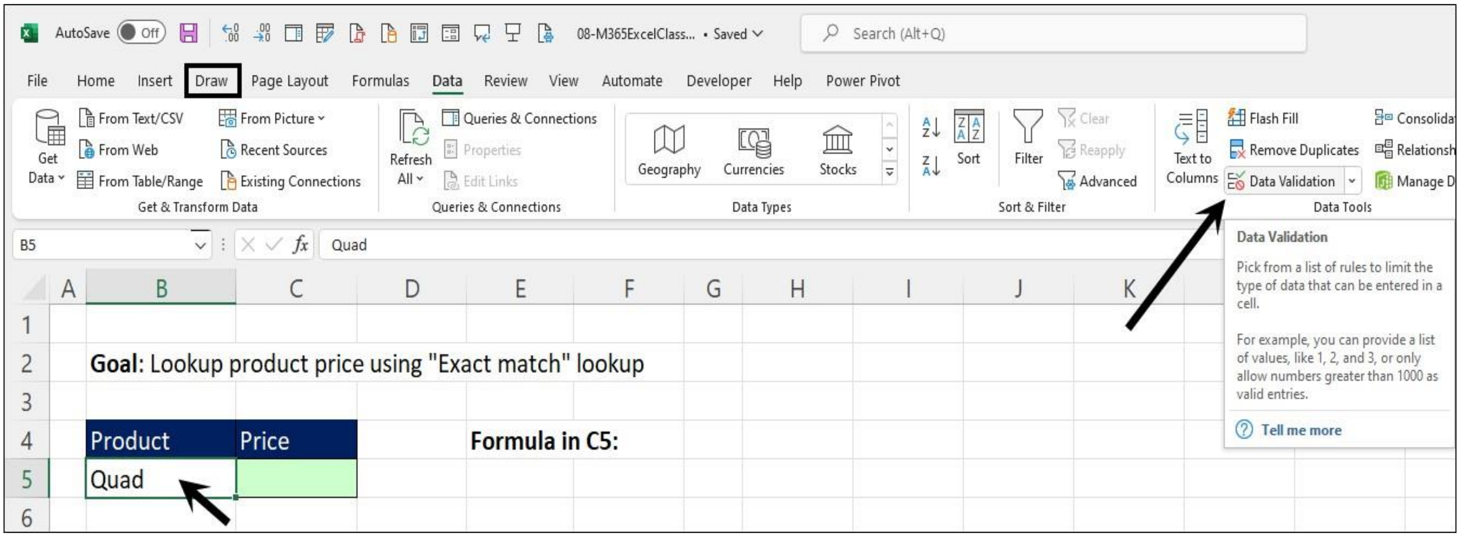
Lookup Table:

ProductID	Product	Cost	Price
A25C-4884	Aspen	14.55	26.95
C20G-4398	Carlota	11.18	27.95
Q22G-9634	Quad	19.34	43.95
S35G-4382	Sunshine	11.25	22.95
Y30C-4942	Yanaki	12.88	29.95

Data Validation dropdown list

When you have a lookup table and you are doing Exact Match Lookup, you can use the unique list from the first column of the lookup table to help add a dropdown list of values to the cell that that contains the lookup value for the XLOOKUP function. Here are the steps:

1. Select cell that contains the lookup value
2. Click the Data tab, then in the Data Tools group click the Data Validation button



Goal: Lookup product price using "Exact match" lookup

Product	Price
Quad	

Formula in C5:

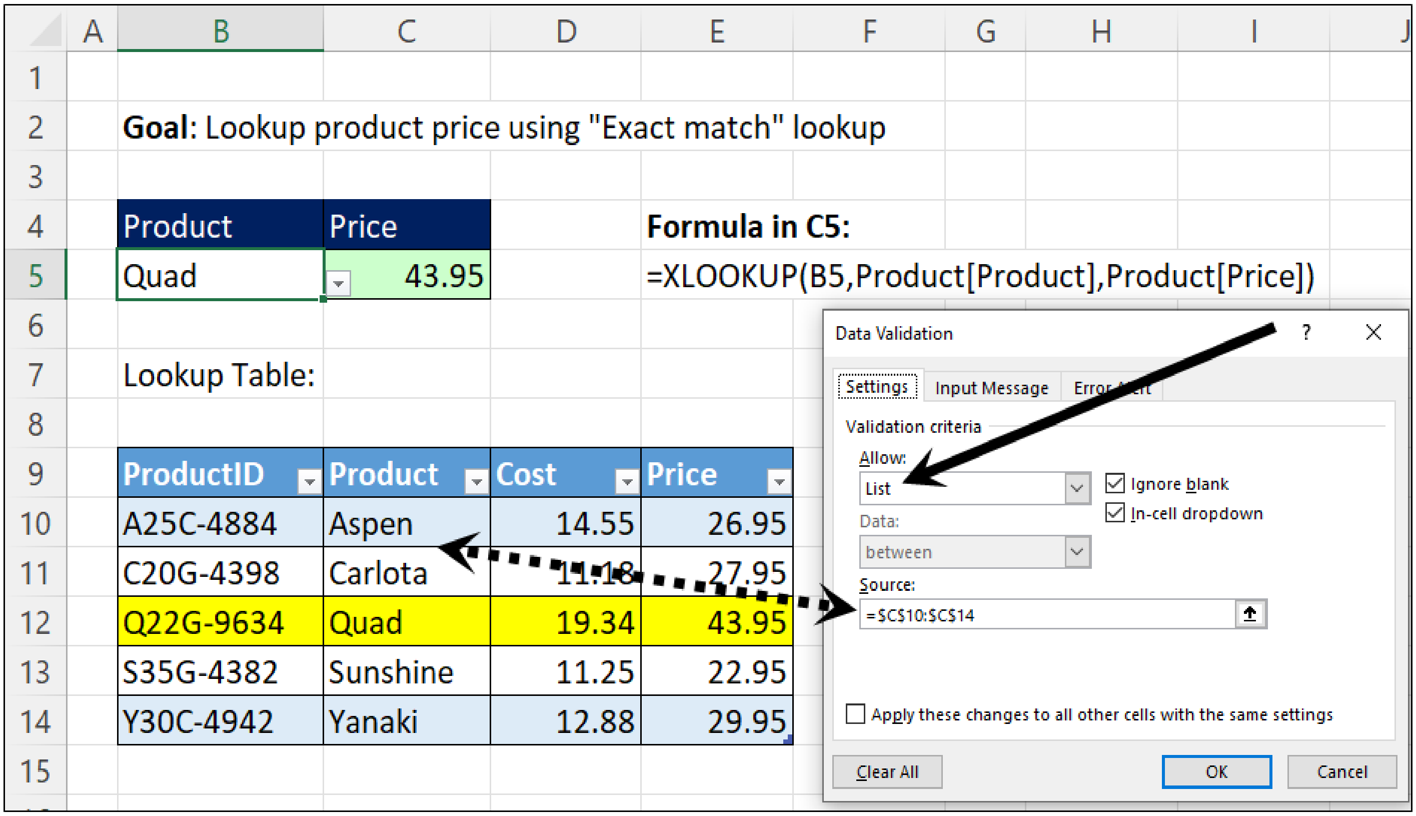
Data Validation

Pick from a list of rules to limit the type of data that can be entered in a cell.

For example, you can provide a list of values, like 1, 2, and 3, or only allow numbers greater than 1000 as valid entries.

Tell me more

3. In the Data Validation dialog box, select "List" from the Allow textbox dropdown
4. Click in the Source textbox and select the unique list from the first column of the lookup table.
5. Click OK



Goal: Lookup product price using "Exact match" lookup

Product	Price
Quad	43.95

Formula in C5:
=XLOOKUP(B5,Product[Product],Product[Price])

Lookup Table:

ProductID	Product	Cost	Price
A25C-4884	Aspen	14.55	26.95
C20G-4398	Carlota	11.18	27.95
Q22G-9634	Quad	19.34	43.95
S35G-4382	Sunshine	11.25	22.95
Y30C-4942	Yanaki	12.88	29.95

Data Validation

Settings | Input Message | Error Alert

Validation criteria

Allow: List

☑ Ignore blank

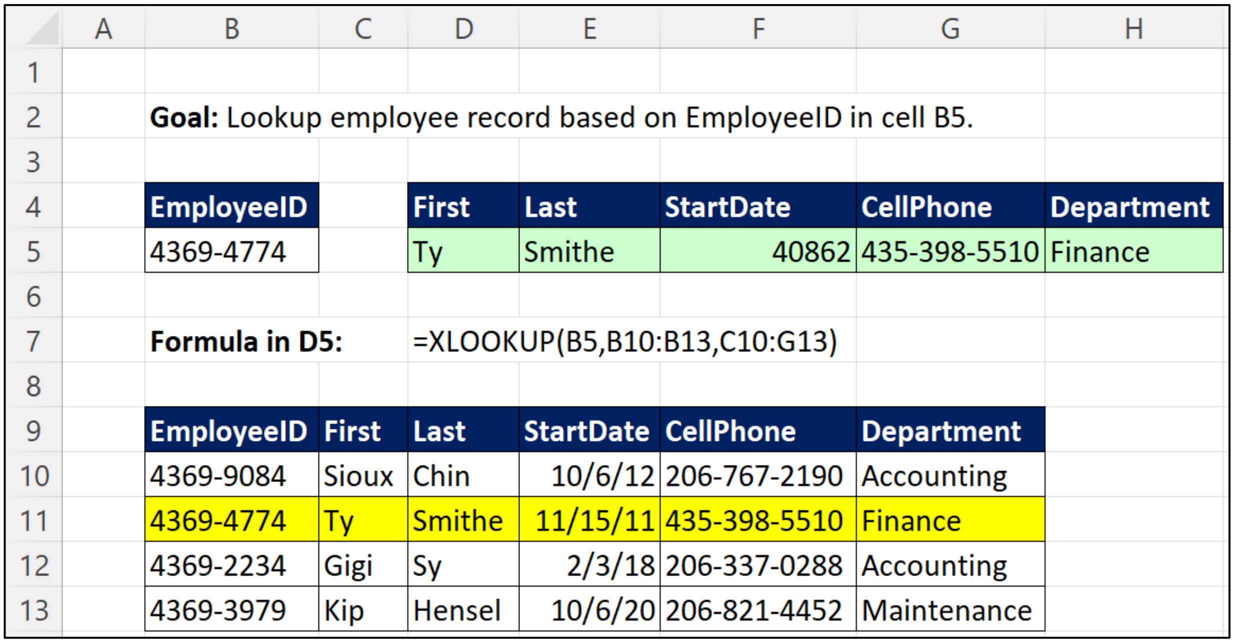
☑ In-cell dropdown

Data: between

Source: =$C$10:$C$14

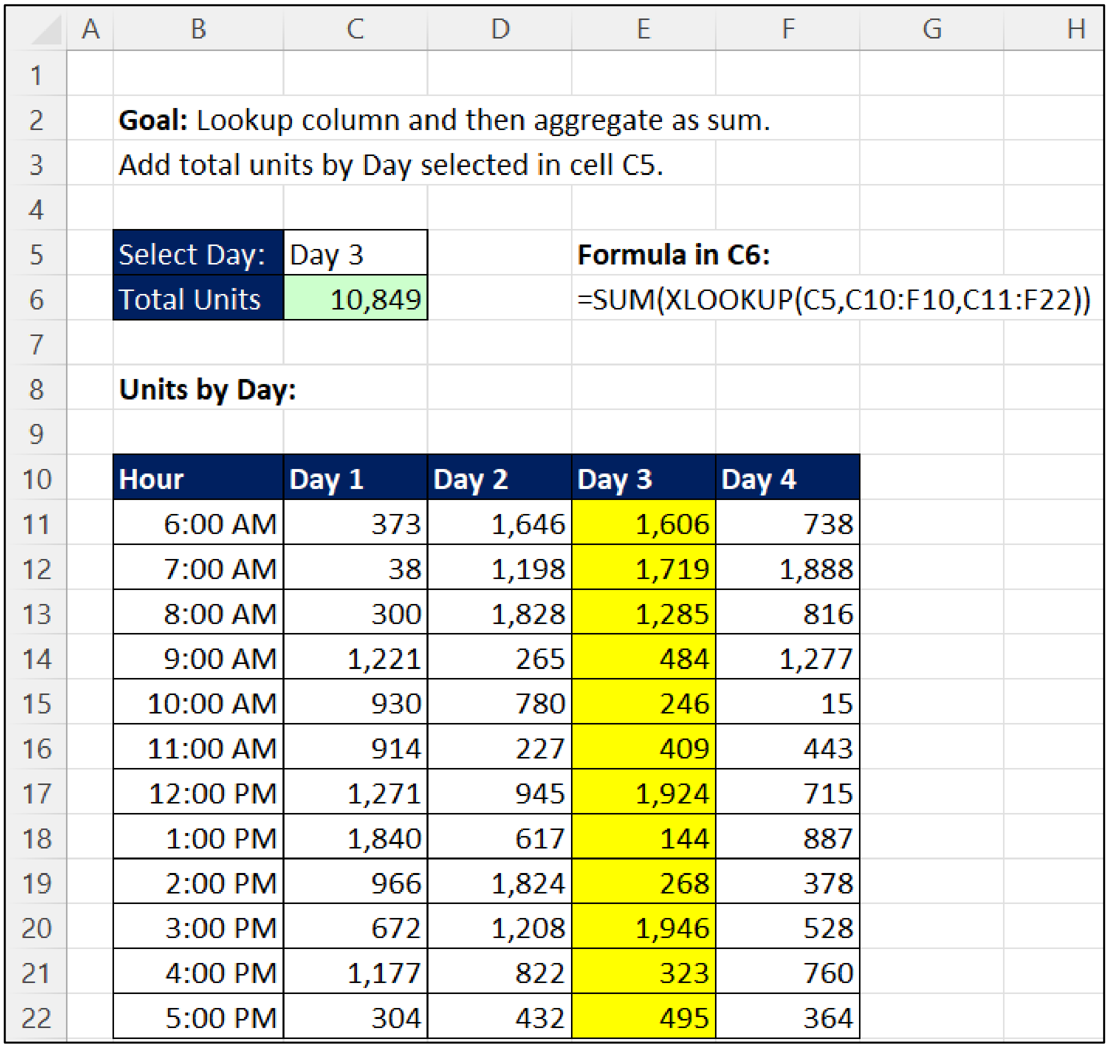
☐ Apply these changes to all other cells with the same settings

Lookup an employee record



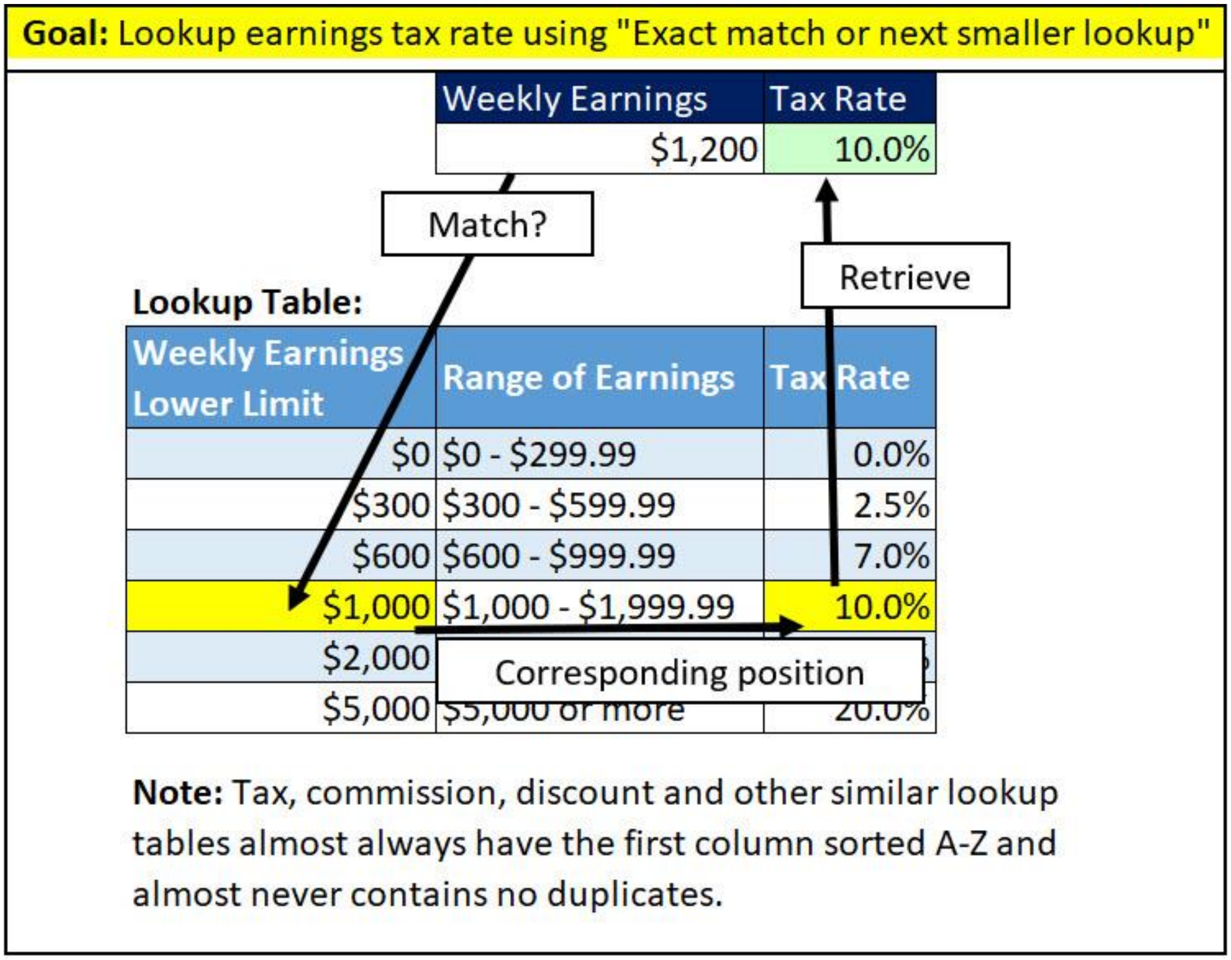
	A	B	C	D	E	F	G	H
1								
2		**Goal:** Lookup employee record based on EmployeeID in cell B5.						
3								
4		**EmployeeID**		**First**	**Last**	**StartDate**	**CellPhone**	**Department**
5		4369-4774		Ty	Smithe	40862	435-398-5510	Finance
6								
7		**Formula in D5:**		=XLOOKUP(B5,B10:B13,C10:G13)				
8								
9		**EmployeeID**	**First**	**Last**	**StartDate**	**CellPhone**	**Department**	
10		4369-9084	Sioux	Chin	10/6/12	206-767-2190	Accounting	
11		4369-4774	Ty	Smithe	11/15/11	435-398-5510	Finance	
12		4369-2234	Gigi	Sy	2/3/18	206-337-0288	Accounting	
13		4369-3979	Kip	Hensel	10/6/20	206-821-4452	Maintenance	

Lookup a column of day's sales and then add



	A	B	C	D	E	F	G	H
1								
2		**Goal:** Lookup column and then aggregate as sum.						
3		Add total units by Day selected in cell C5.						
4								
5		Select Day:	Day 3		**Formula in C6:**			
6		Total Units	10,849		=SUM(XLOOKUP(C5,C10:F10,C11:F22))			
7								
8		**Units by Day:**						
9								
10		**Hour**	**Day 1**	**Day 2**	**Day 3**	**Day 4**		
11		6:00 AM	373	1,646	1,606	738		
12		7:00 AM	38	1,198	1,719	1,888		
13		8:00 AM	300	1,828	1,285	816		
14		9:00 AM	1,221	265	484	1,277		
15		10:00 AM	930	780	246	15		
16		11:00 AM	914	227	409	443		
17		12:00 PM	1,271	945	1,924	715		
18		1:00 PM	1,840	617	144	887		
19		2:00 PM	966	1,824	268	378		
20		3:00 PM	672	1,208	1,946	528		
21		4:00 PM	1,177	822	323	760		
22		5:00 PM	304	432	495	364		

Exact match or next smaller lookup



Goal: Lookup earnings tax rate using "Exact match or next smaller lookup"

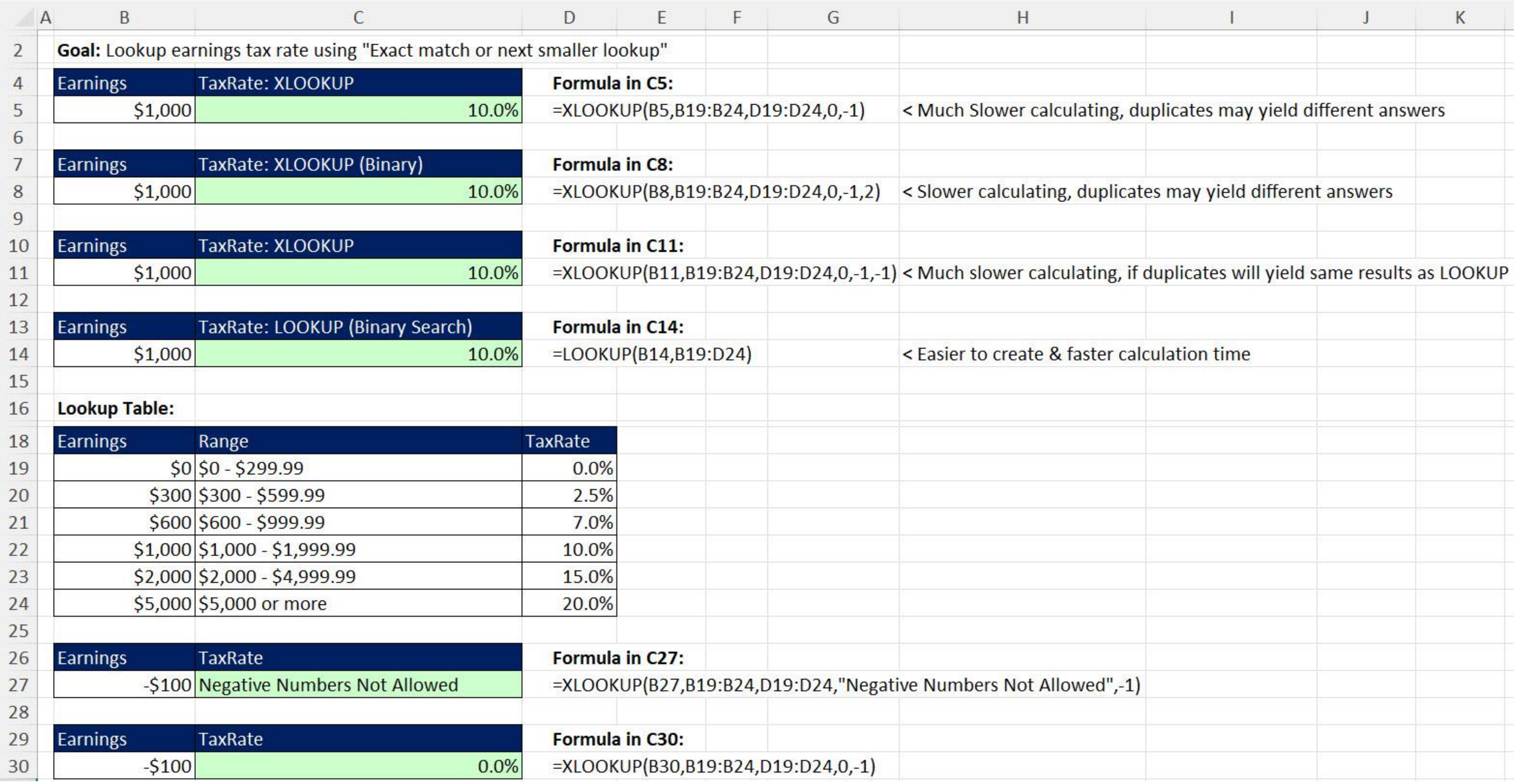
Weekly Earnings	Tax Rate
$1,200	10.0%

Lookup Table:

Weekly Earnings Lower Limit	Range of Earnings	Tax Rate
$0	$0 - $299.99	0.0%
$300	$300 - $599.99	2.5%
$600	$600 - $999.99	7.0%
$1,000	$1,000 - $1,999.99	10.0%
$2,000		
$5,000	$5,000 or more	20.0%

Note: Tax, commission, discount and other similar lookup tables almost always have the first column sorted A-Z and almost never contains no duplicates.

Lookup a tax rate: XLOOKUP or LOOKUP function?

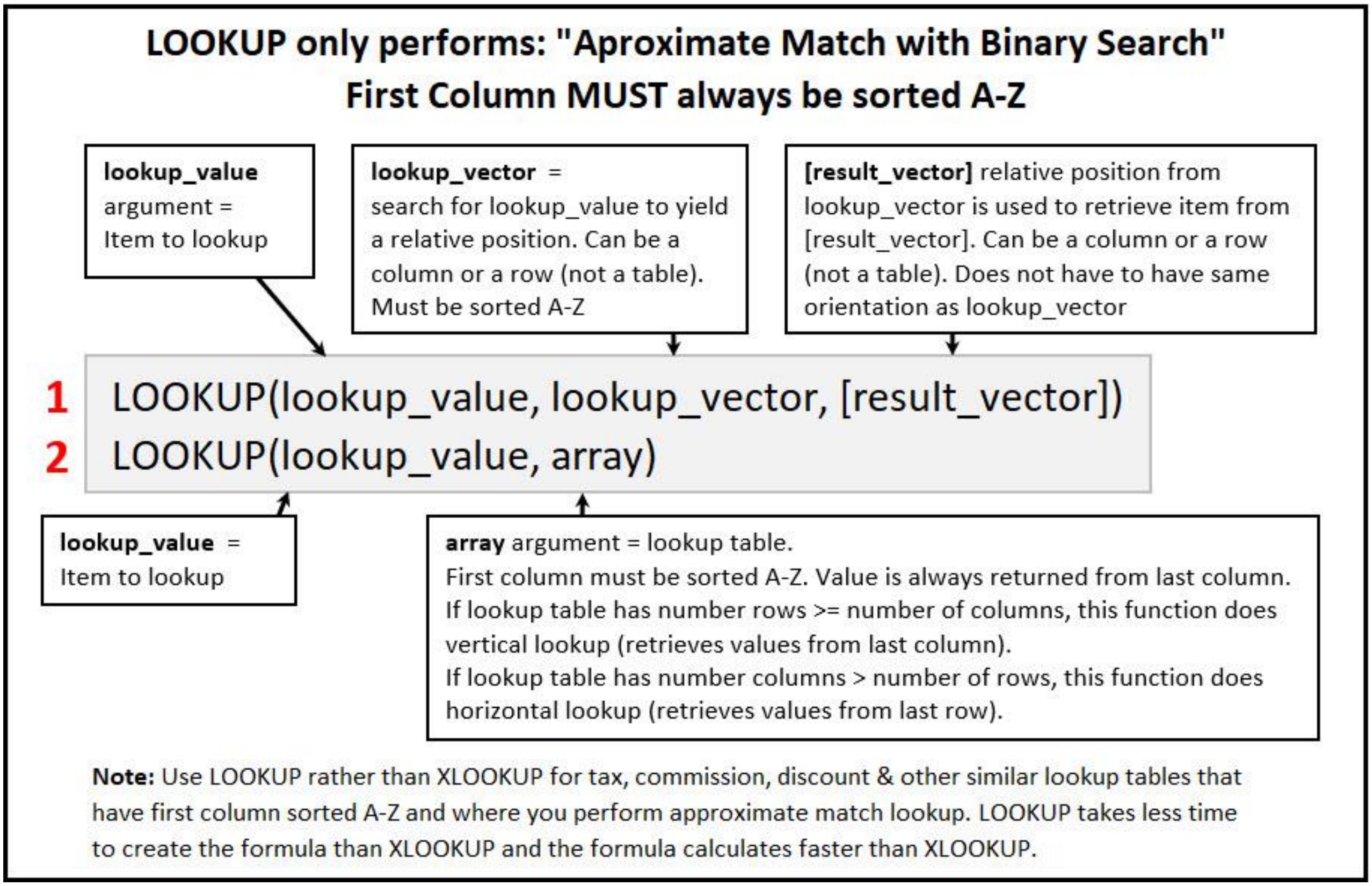


	A	B	C	D	E	F	G	H	I	J	K
2		**Goal:** Lookup earnings tax rate using "Exact match or next smaller lookup"									
4		Earnings	TaxRate: XLOOKUP	**Formula in C5:**							
5		$1,000	10.0%	=XLOOKUP(B5,B19:B24,D19:D24,0,-1)				< Much Slower calculating, duplicates may yield different answers			
6											
7		Earnings	TaxRate: XLOOKUP (Binary)	**Formula in C8:**							
8		$1,000	10.0%	=XLOOKUP(B8,B19:B24,D19:D24,0,-1,2)				< Slower calculating, duplicates may yield different answers			
9											
10		Earnings	TaxRate: XLOOKUP	**Formula in C11:**							
11		$1,000	10.0%	=XLOOKUP(B11,B19:B24,D19:D24,0,-1,-1)				< Much slower calculating, if duplicates will yield same results as LOOKUP			
12											
13		Earnings	TaxRate: LOOKUP (Binary Search)	**Formula in C14:**							
14		$1,000	10.0%	=LOOKUP(B14,B19:D24)				< Easier to create & faster calculation time			
15											
16		**Lookup Table:**									
18		Earnings	Range	TaxRate							
19		$0	$0 - $299.99	0.0%							
20		$300	$300 - $599.99	2.5%							
21		$600	$600 - $999.99	7.0%							
22		$1,000	$1,000 - $1,999.99	10.0%							
23		$2,000	$2,000 - $4,999.99	15.0%							
24		$5,000	$5,000 or more	20.0%							
25											
26		Earnings	TaxRate	**Formula in C27:**							
27		-$100	Negative Numbers Not Allowed	=XLOOKUP(B27,B19:B24,D19:D24,"Negative Numbers Not Allowed",-1)							
28											
29		Earnings	TaxRate	**Formula in C30:**							
30		-$100	0.0%	=XLOOKUP(B30,B19:B24,D19:D24,0,-1)							

LOOKUP function arguments

The LOOKUP is the first lookup function in spreadsheet history. The LOOKUP function was one of the original 21 functions in the first spreadsheet in history introduced in 1979: VisiCalc.

The first thing to note about the LOOKUP function is that it has two options—in essence giving you two different functions in one function (see Figure below):



LOOKUP only performs: "Aproximate Match with Binary Search"
First Column MUST always be sorted A-Z

lookup_value argument = Item to lookup

lookup_vector = search for lookup_value to yield a relative position. Can be a column or a row (not a table). Must be sorted A-Z

[result_vector] relative position from lookup_vector is used to retrieve item from [result_vector]. Can be a column or a row (not a table). Does not have to have same orientation as lookup_vector

1 LOOKUP(lookup_value, lookup_vector, [result_vector])
2 LOOKUP(lookup_value, array)

lookup_value = Item to lookup

array argument = lookup table.
First column must be sorted A-Z. Value is always returned from last column.
If lookup table has number rows >= number of columns, this function does vertical lookup (retrieves values from last column).
If lookup table has number columns > number of rows, this function does horizontal lookup (retrieves values from last row).

Note: Use LOOKUP rather than XLOOKUP for tax, commission, discount & other similar lookup tables that have first column sorted A-Z and where you perform approximate match lookup. LOOKUP takes less time to create the formula than XLOOKUP and the formula calculates faster than XLOOKUP.

The second argument of LOOKUP, ***lookup_vector*** or ***array***, is the trigger for the function to determine whether to run Option 1 or Option 2. If you put a one-way array (column or row, but not table) into the second argument, Option 1 is executed; if you put a table (rows and columns) into the second argument, Option 2 is executed.

Consider the following information about Option 1, **LOOKUP(*lookup_value*, *lookup_vector*, [*result_vector*])**:

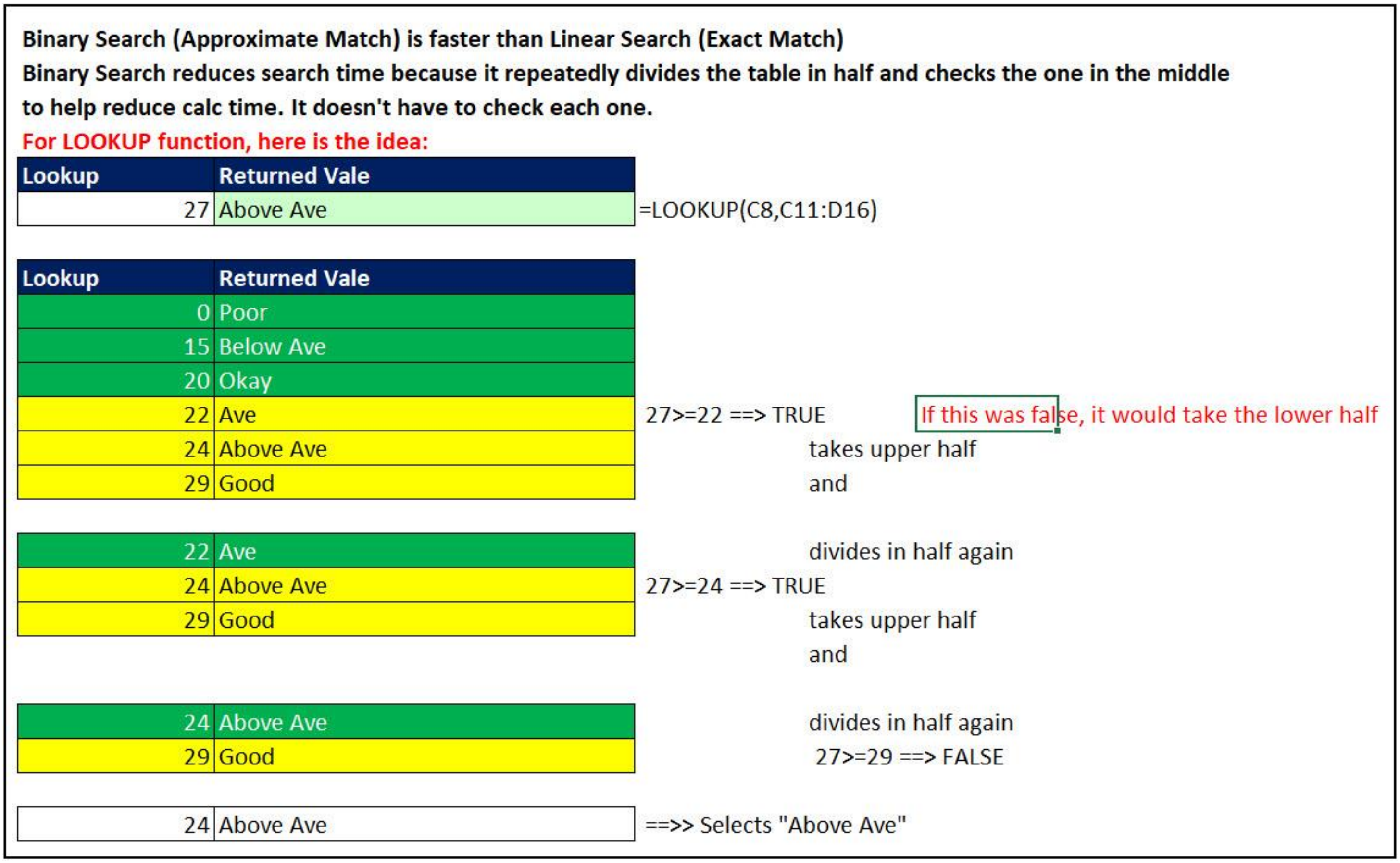
- This option searches for the lookup value in the ascending-sorted **lookup_vector** to find a match and yield a relative position that it uses to then retrieve the corresponding element in the argument **[result_vector]**.
- The **lookup_vector** must be sorted ascending, A–Z.
- The **lookup_vector** and **[return_vector]** argument values must be one-way arrays. These one-way arrays do not have to have the same orientation or the same number of elements. This is different from how the second and third arguments in XLOOKUP work.

Consider the following information about Option 2, **LOOKUP(*lookup_value*, *array*)**:

- This option allows the user to search for the lookup_value in the first column of the lookup table (array argument) and then return an item in the same corresponding position from the last column of the lookup table (array argument).

- The array argument allows you to either use a vertical lookup table or a horizontal lookup table.
    - If the number of rows in the lookup table is greater than or equal to the number of columns, LOOKUP does a vertical lookup. With a vertical lookup, the first column of the table must be sorted ascending, A–Z. The LOOKUP function tries to find a match and yields a relative position that it then uses to retrieve the corresponding element from the last column in the vertical lookup table.
    - If the number of columns in the lookup table is greater than the number of rows, LOOKUP does a horizontal lookup. With a horizontal lookup, the first row of the table must be sorted ascending, A–Z. The LOOKUP function tries to find a match and yields a relative position that it then uses to retrieve the corresponding element from the last row in the horizontal lookup table.
- The advantage that this option has over the XLOOKUP function is that if you have a table that is sorted correctly (like most tax tables, commission tables, and other tables), you only have to enter two arguments when using the LOOKUP function rather than enter five with the XLOOKUP function.

Why the Binary Search in the LOOKUP function is fast



Binary Search (Approximate Match) is faster than Linear Search (Exact Match)

Binary Search reduces search time because it repeatedly divides the table in half and checks the one in the middle to help reduce calc time. It doesn't have to check each one.

For LOOKUP function, here is the idea:

Lookup	Returned Vale	
27	Above Ave	=LOOKUP(C8,C11:D16)

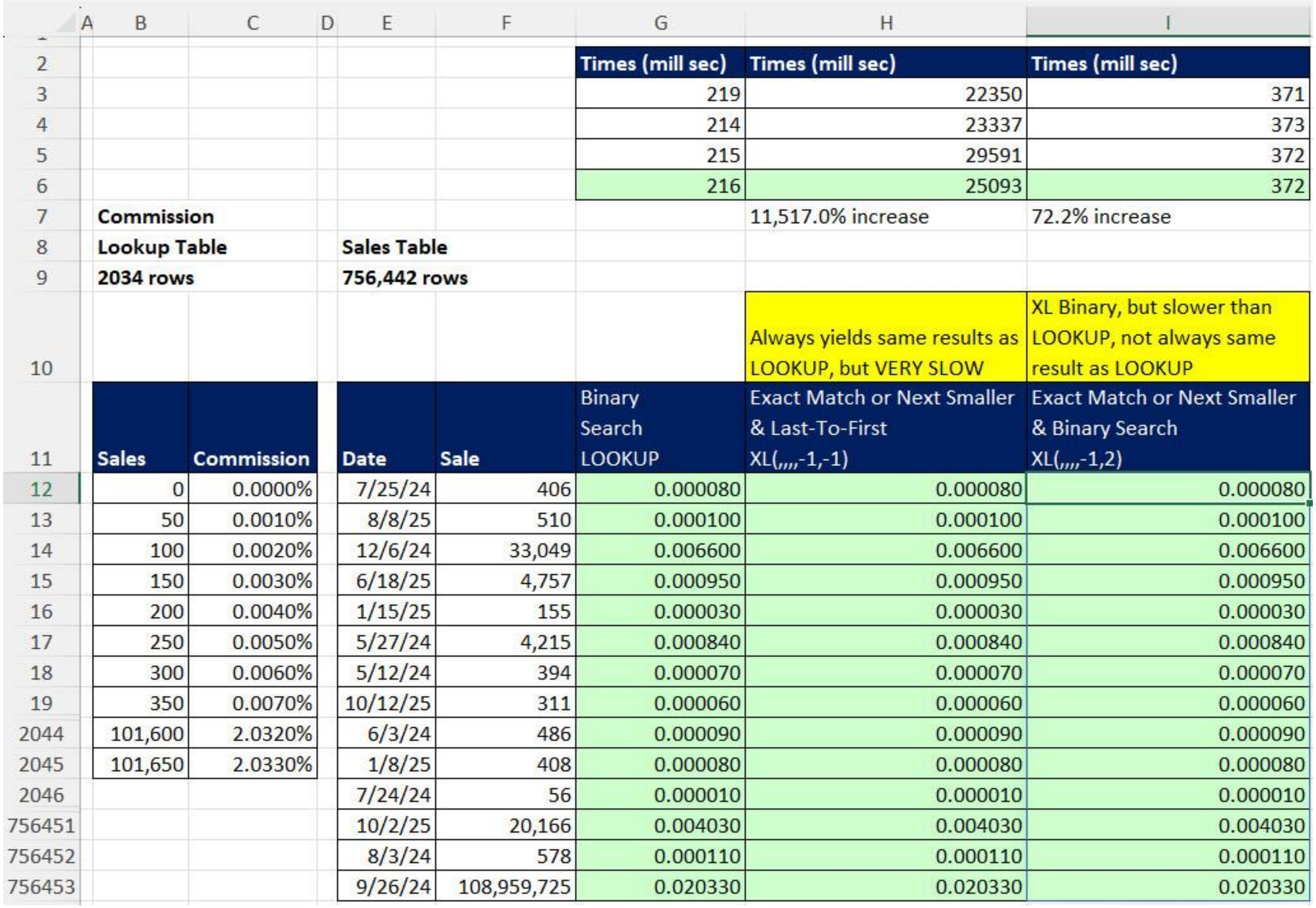
Lookup	Returned Vale	
0	Poor	
15	Below Ave	
20	Okay	
22	Ave	27>=22 ==> TRUE If this was false, it would take the lower half
24	Above Ave	takes upper half
29	Good	and

22	Ave	divides in half again
24	Above Ave	27>=24 ==> TRUE
29	Good	takes upper half
		and

24	Above Ave	divides in half again
29	Good	27>=29 ==> FALSE

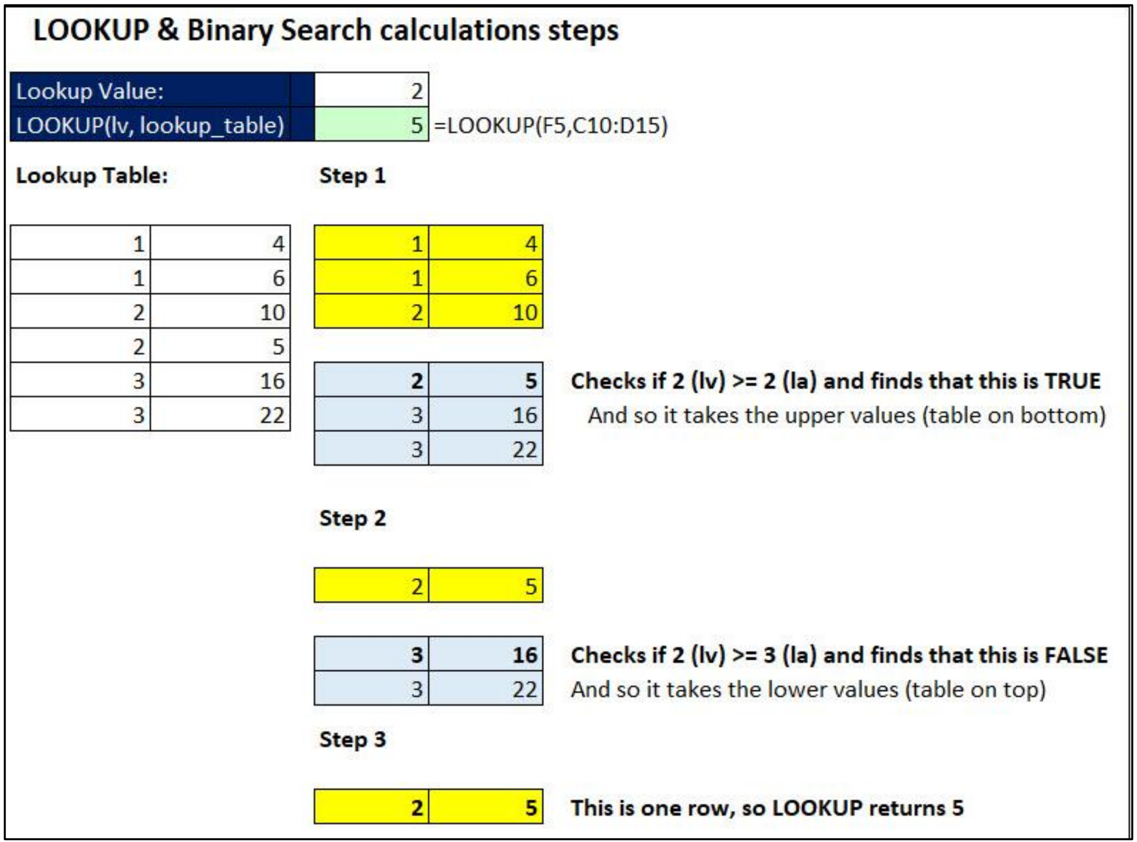
24	Above Ave	==>> Selects "Above Ave"

Timing Results for XLOOKUP and LOOKUP functions



	A	B	C	D	E	F	G	H	I
2							Times (mill sec)	Times (mill sec)	Times (mill sec)
3							219	22350	371
4							214	23337	373
5							215	29591	372
6							216	25093	372
7		Commission						11,517.0% increase	72.2% increase
8		Lookup Table			Sales Table				
9		2034 rows			756,442 rows				
10								Always yields same results as LOOKUP, but VERY SLOW	XL Binary, but slower than LOOKUP, not always same result as LOOKUP
11		Sales	Commission		Date	Sale	Binary Search LOOKUP	Exact Match or Next Smaller & Last-To-First XL(,,,,-1,-1)	Exact Match or Next Smaller & Binary Search XL(,,,,-1,2)
12		0	0.0000%		7/25/24	406	0.000080	0.000080	0.000080
13		50	0.0010%		8/8/25	510	0.000100	0.000100	0.000100
14		100	0.0020%		12/6/24	33,049	0.006600	0.006600	0.006600
15		150	0.0030%		6/18/25	4,757	0.000950	0.000950	0.000950
16		200	0.0040%		1/15/25	155	0.000030	0.000030	0.000030
17		250	0.0050%		5/27/24	4,215	0.000840	0.000840	0.000840
18		300	0.0060%		5/12/24	394	0.000070	0.000070	0.000070
19		350	0.0070%		10/12/25	311	0.000060	0.000060	0.000060
2044		101,600	2.0320%		6/3/24	486	0.000090	0.000090	0.000090
2045		101,650	2.0330%		1/8/25	408	0.000080	0.000080	0.000080
2046					7/24/24	56	0.000010	0.000010	0.000010
756451					10/2/25	20,166	0.004030	0.004030	0.004030
756452					8/3/24	578	0.000110	0.000110	0.000110
756453					9/26/24	108,959,725	0.020330	0.020330	0.020330

How XLOOKUP Binary may yield different answers than LOOKUP Binary when there are duplicates?



LOOKUP & Binary Search calculations steps

Lookup Value:	2
LOOKUP(lv, lookup_table)	5

=LOOKUP(F5,C10:D15)

Lookup Table:

1	4
1	6
2	10
2	5
3	16
3	22

Step 1

1	4
1	6
2	10

2	**5**
3	16
3	22

Checks if 2 (lv) >= 2 (la) and finds that this is TRUE

And so it takes the upper values (table on bottom)

Step 2

2	5

3	**16**
3	22

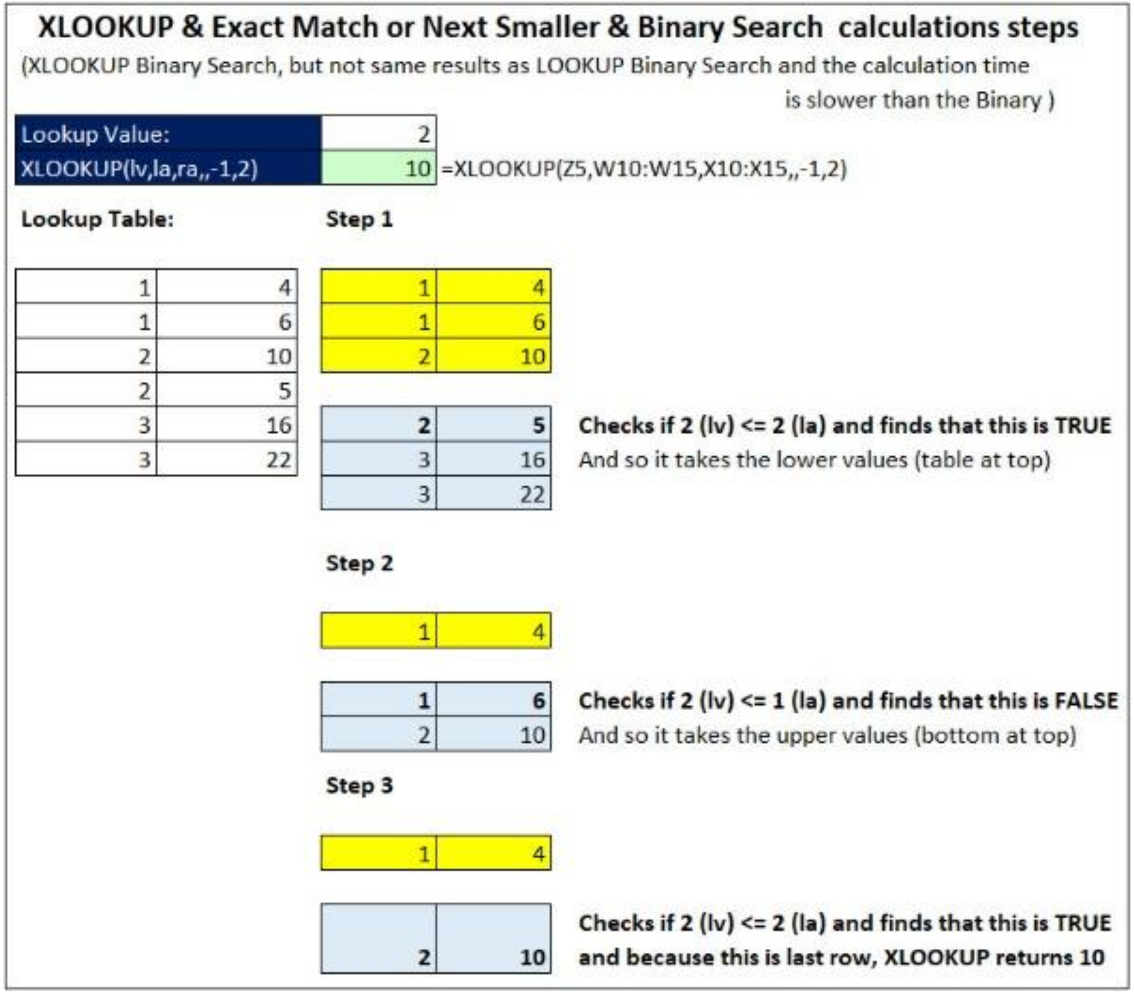
Checks if 2 (lv) >= 3 (la) and finds that this is FALSE

And so it takes the lower values (table on top)

Step 3

2	**5**

This is one row, so LOOKUP returns 5



XLOOKUP & Exact Match or Next Smaller & Binary Search calculations steps

(XLOOKUP Binary Search, but not same results as LOOKUP Binary Search and the calculation time is slower than the Binary)

Lookup Value:	2
XLOOKUP(lv,la,ra,,-1,2)	10

=XLOOKUP(Z5,W10:W15,X10:X15,,-1,2)

Lookup Table:

1	4
1	6
2	10
2	5
3	16
3	22

Step 1

1	4
1	6
2	10

2	**5**
3	16
3	22

Checks if 2 (lv) <= 2 (la) and finds that this is TRUE

And so it takes the lower values (table at top)

Step 2

1	4

1	**6**
2	10

Checks if 2 (lv) <= 1 (la) and finds that this is FALSE

And so it takes the upper values (bottom at top)

Step 3

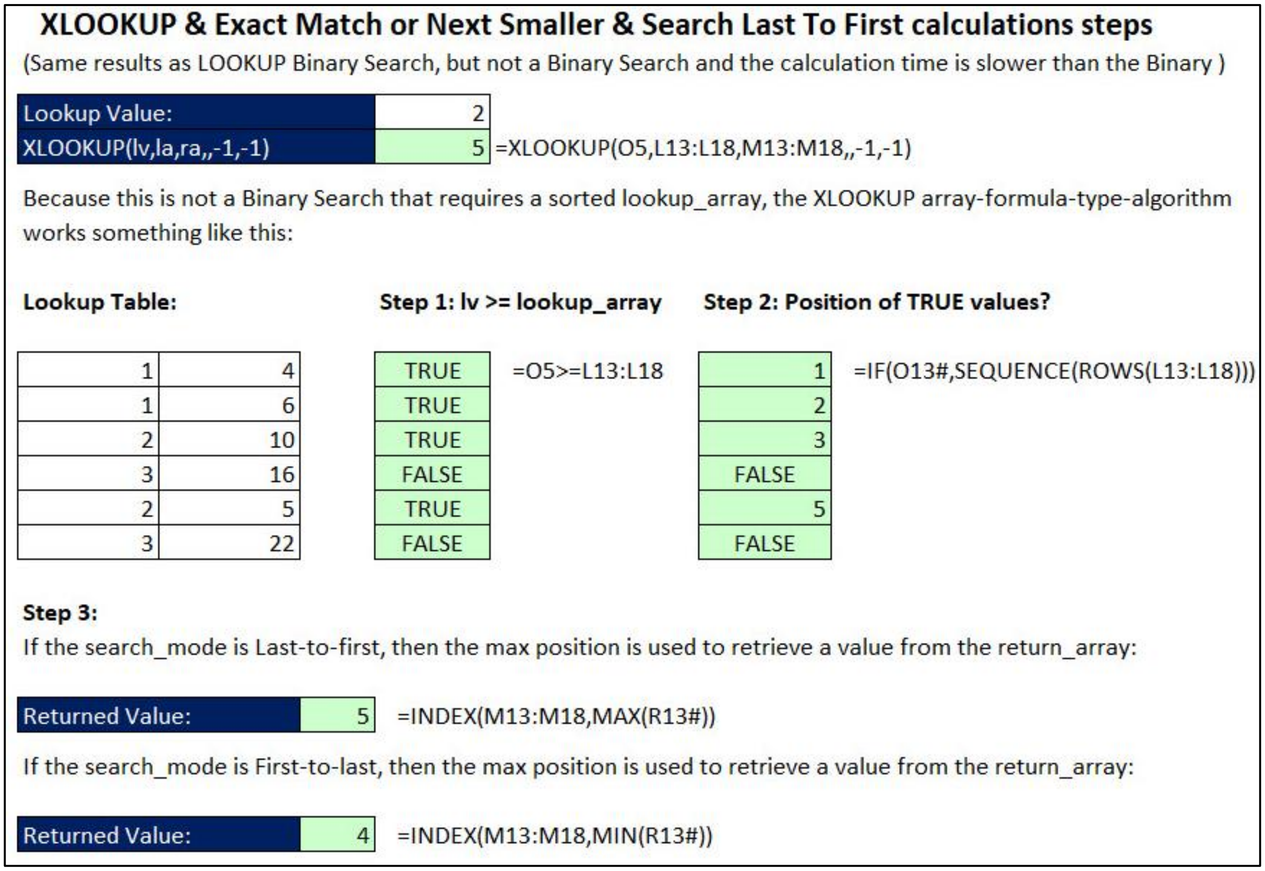
1	4

2	**10**

Checks if 2 (lv) <= 2 (la) and finds that this is TRUE

and because this is last row, XLOOKUP returns 10

How can XLOOKUP yield same answer as LOOKUP, even with duplicate values?



XLOOKUP & Exact Match or Next Smaller & Search Last To First calculations steps

(Same results as LOOKUP Binary Search, but not a Binary Search and the calculation time is slower than the Binary)

Lookup Value:	2	
XLOOKUP(lv,la,ra,,-1,-1)	5	=XLOOKUP(O5,L13:L18,M13:M18,,-1,-1)

Because this is not a Binary Search that requires a sorted lookup_array, the XLOOKUP array-formula-type-algorithm works something like this:

Lookup Table:		**Step 1: lv >= lookup_array**		**Step 2: Position of TRUE values?**	
1	4	TRUE	=O5>=L13:L18	1	=IF(O13#,SEQUENCE(ROWS(L13:L18)))
1	6	TRUE		2	
2	10	TRUE		3	
3	16	FALSE		FALSE	
2	5	TRUE		5	
3	22	FALSE		FALSE	

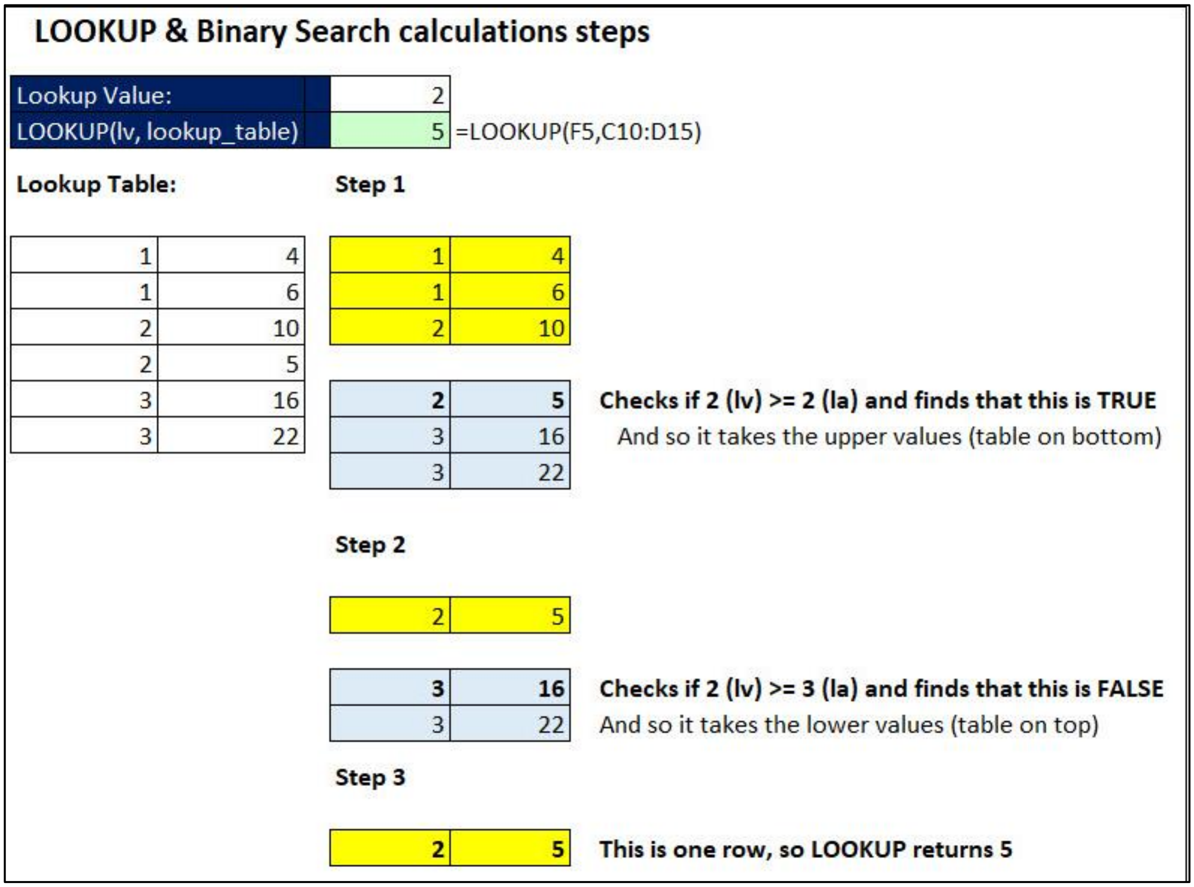
Step 3:

If the search_mode is Last-to-first, then the max position is used to retrieve a value from the return_array:

Returned Value:	5	=INDEX(M13:M18,MAX(R13#))

If the search_mode is First-to-last, then the max position is used to retrieve a value from the return_array:

Returned Value:	4	=INDEX(M13:M18,MIN(R13#))



LOOKUP & Binary Search calculations steps

Lookup Value:	2	
LOOKUP(lv, lookup_table)	5	=LOOKUP(F5,C10:D15)

Lookup Table:

1	4
1	6
2	10
2	5
3	16
3	22

Step 1

1	4
1	6
2	10

2	**5**	**Checks if 2 (lv) >= 2 (la) and finds that this is TRUE**
3	16	And so it takes the upper values (table on bottom)
3	22	

Step 2

2	5

3	**16**	**Checks if 2 (lv) >= 3 (la) and finds that this is FALSE**
3	22	And so it takes the lower values (table on top)

Step 3

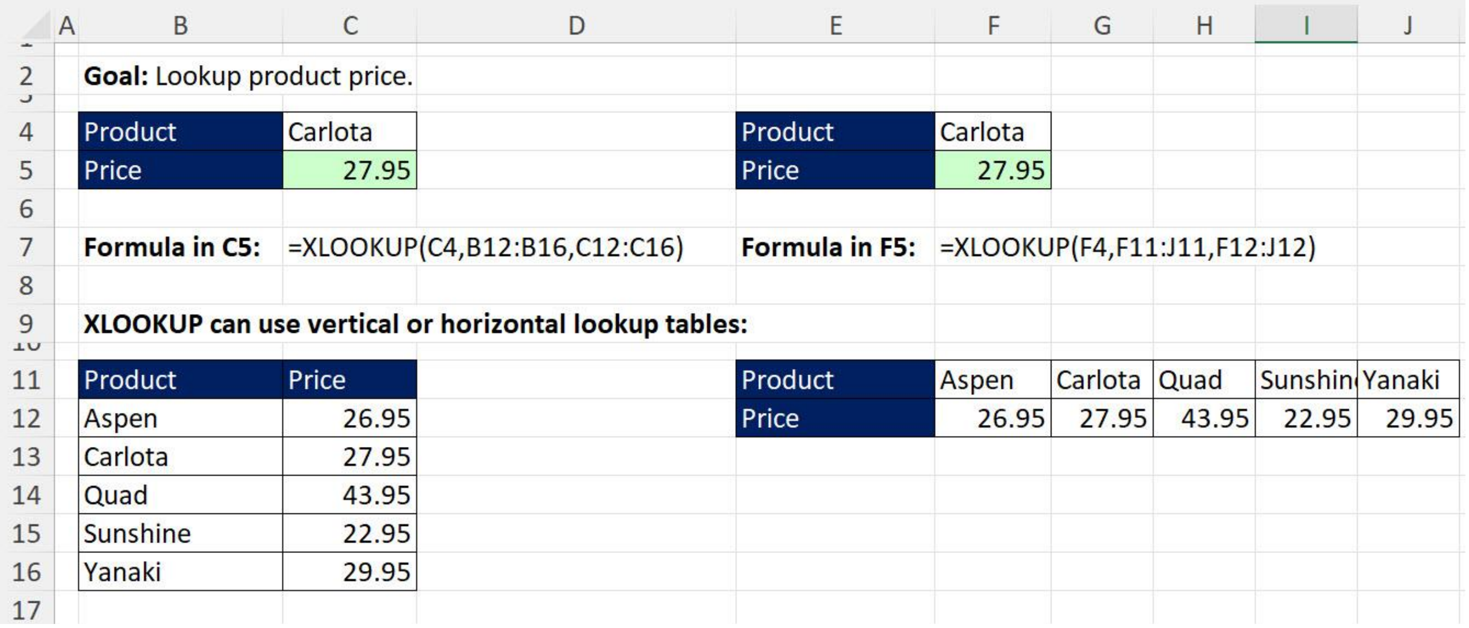
2	**5**	**This is one row, so LOOKUP returns 5**

When to use LOOKUP function rather than XLOOKUP function?

If you need to do “Exact Match or Next Smaller” lookup, also known as “Approximate Match Lookup” and you are using a typical tax, commission, sales discount, or other similar tables that have the first column sorted A-Z, the first column has no duplicates and the last column has the values you want to return, use the LOOKUP function rather than the XLOOKUP function because of these benefits:

1. It takes less time to create the lookup formula with the LOOKUP function because you only have to enter two arguments rather than five or six.
2. The LOOKUP function uses a binary search process that allows the formula to calculate faster than the XLOOKUP function with a binary search option selected.

XLOOKUP can do vertical or horizontal lookup



Goal: Lookup product price.

Product	Carlota
Price	27.95

Product	Carlota
Price	27.95

Formula in C5: =XLOOKUP(C4,B12:B16,C12:C16)

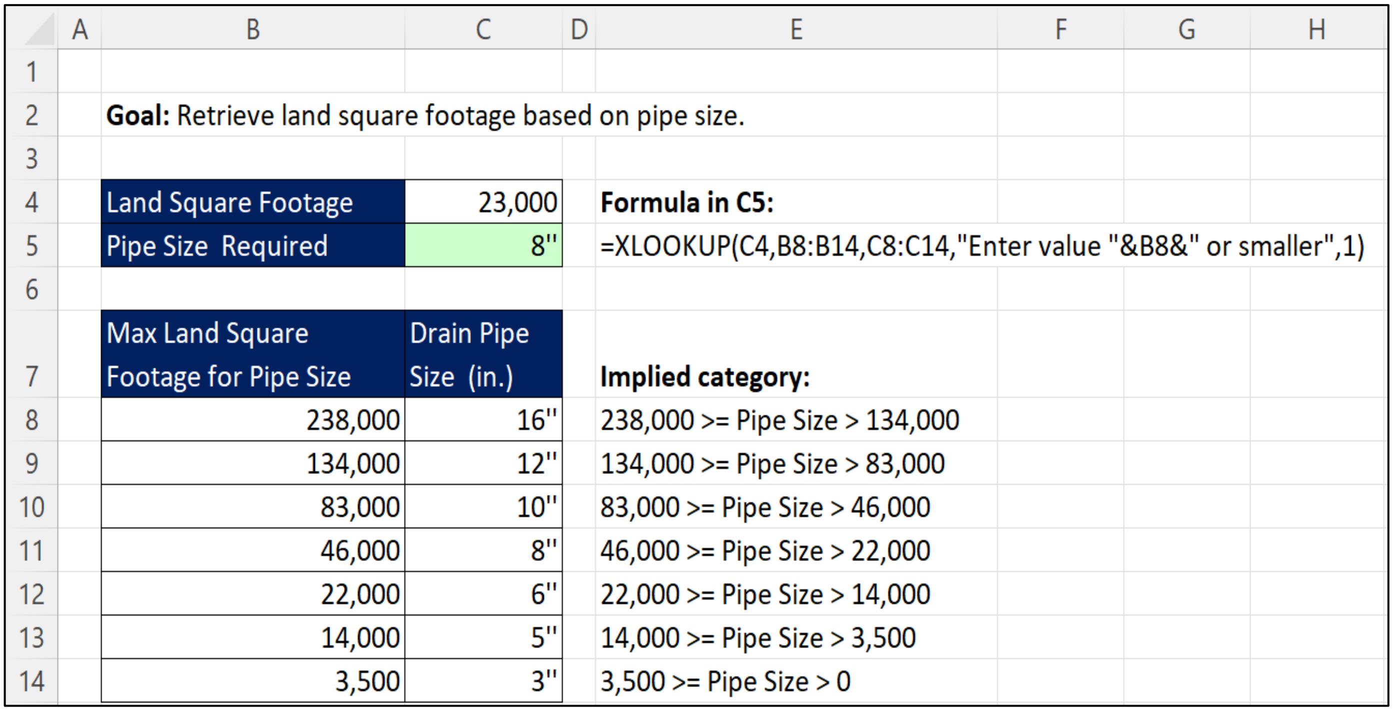
Formula in F5: =XLOOKUP(F4,F11:J11,F12:J12)

XLOOKUP can use vertical or horizontal lookup tables:

Product	Price
Aspen	26.95
Carlota	27.95
Quad	43.95
Sunshine	22.95
Yanaki	29.95

Product	Aspen	Carlota	Quad	Sunshin	Yanaki
Price	26.95	27.95	43.95	22.95	29.95

XLOOKUP to perform Exact Match or Next Bigger lookup: lookup pipe size



Goal: Retrieve land square footage based on pipe size.

Land Square Footage	23,000
Pipe Size Required	8"

Formula in C5:

=XLOOKUP(C4,B8:B14,C8:C14,"Enter value "&B8&" or smaller",1)

Max Land Square Footage for Pipe Size	Drain Pipe Size (in.)	**Implied category:**
238,000	16"	238,000 >= Pipe Size > 134,000
134,000	12"	134,000 >= Pipe Size > 83,000
83,000	10"	83,000 >= Pipe Size > 46,000
46,000	8"	46,000 >= Pipe Size > 22,000
22,000	6"	22,000 >= Pipe Size > 14,000
14,000	5"	14,000 >= Pipe Size > 3,500
3,500	3"	3,500 >= Pipe Size > 0

XLOOKUP with wild card lookup: “Coca Cola” matches “Coca Cola Inc.”

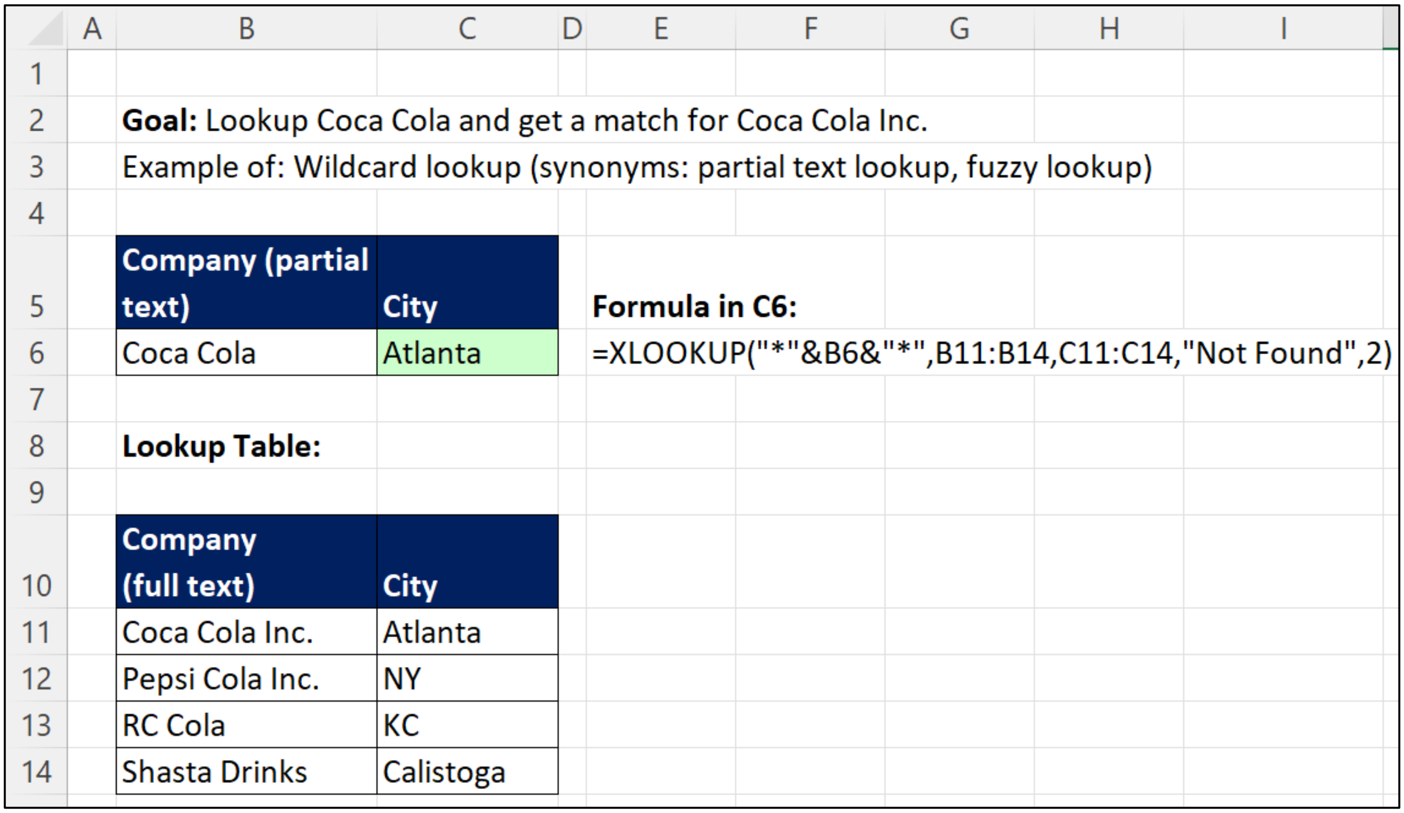
A few special characters can help you conduct searches in an Excel worksheet:

Asterisk (*): The * is a wildcard character that represents any set of zero or more characters.

Question mark (?): The ? is a wildcard character that represents exactly any one single character.

Tilde (~): The ~ is used to convert the wildcards * and ? to actual characters.

The figure below shows a lookup situation where you use a 2 in the [*match_mode*] argument so you can use wildcards with the lookup value. The goal is to type just part of the company name, such as *Coca Cola*, and get a match with the full company name *Coca Cola Inc*. The wildcard asterisk (*) represents zero or more characters. When you use the join operator (&) to join an asterisk in double quotes to the front and back of the cell reference B6, you are asking to make a match with any text in the *lookup_array* argument that contains *Coca Cola*. The formula also includes the text “Not Found” in the [*if_not_found*] argument so that XLOOKUP can return that message if the user types some text that is not contained in any of the values in the *lookup_array* argument.



Goal: Lookup Coca Cola and get a match for Coca Cola Inc.

Example of: Wildcard lookup (synonyms: partial text lookup, fuzzy lookup)

Company (partial text)	City
Coca Cola	Atlanta

Formula in C6:

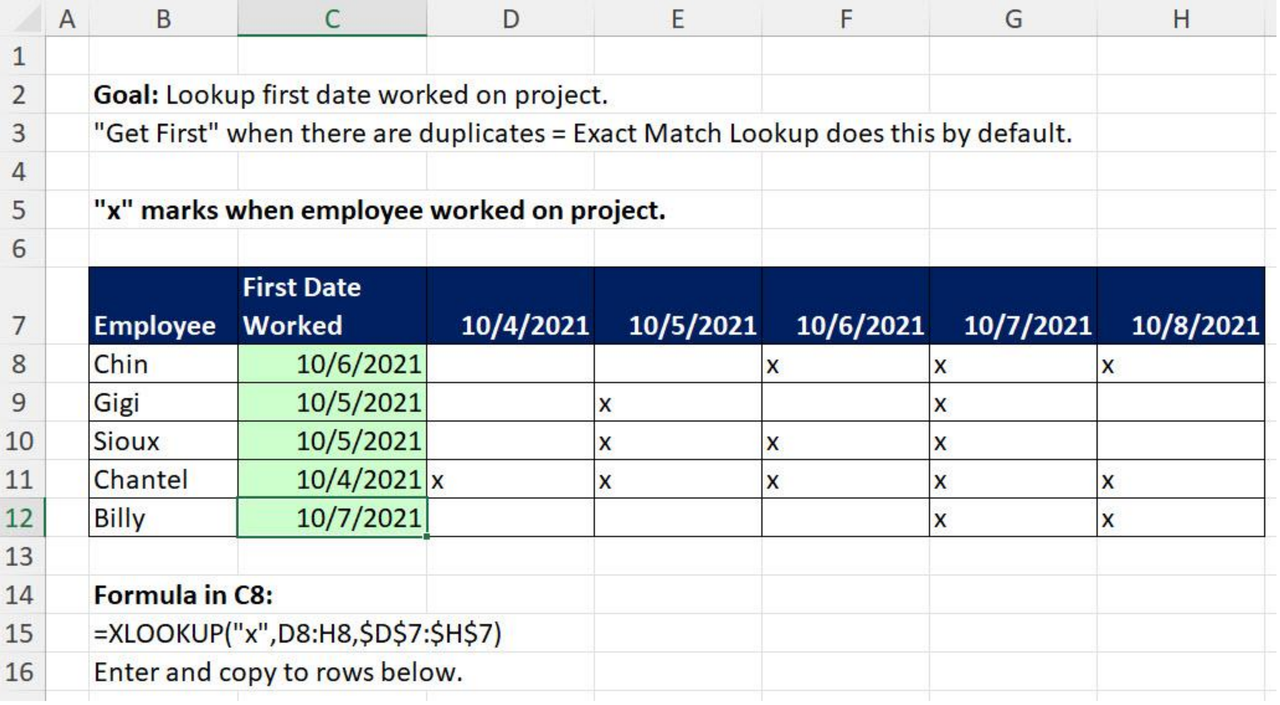
```
=XLOOKUP("*"&B6&"*",B11:B14,C11:C14,"Not Found",2)
```

Lookup Table:

Company (full text)	City
Coca Cola Inc.	Atlanta
Pepsi Cola Inc.	NY
RC Cola	KC
Shasta Drinks	Calistoga

The drawback to using wildcards is that it is not guaranteed that the formula can always return the correct value. For example, if you type Cola into cell B6, the formula will not know whether you want Coca Cola Inc., Pepsi Cola Inc., or RC Cola. However, because the default is to return the first item when there are duplicates, if you enter Cola, the formula will return the city of Atlanta because Coca Cola Inc. is the first company name that contains Cola. This sort of lookup is never 100% guaranteed to work.

Lookup first when there are duplicates: lookup first day worked



Goal: Lookup first date worked on project.

"Get First" when there are duplicates = Exact Match Lookup does this by default.

"x" marks when employee worked on project.

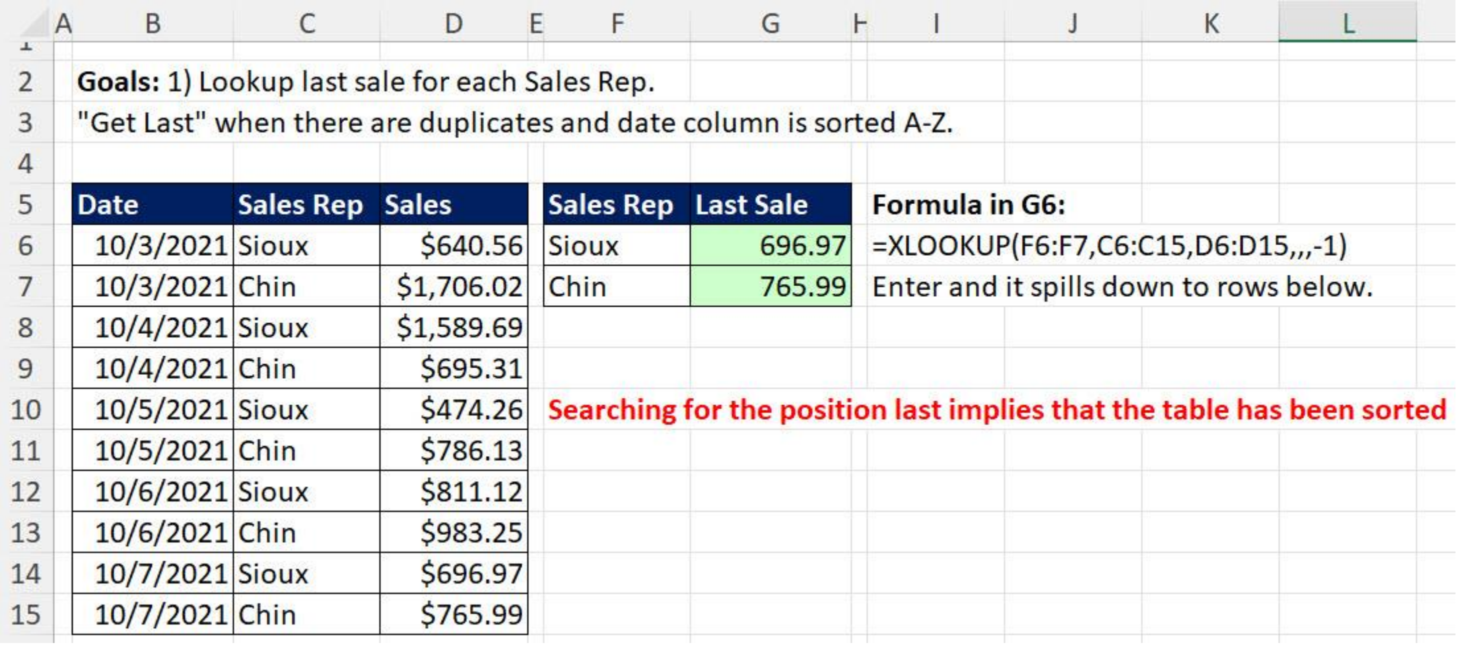
Employee	First Date Worked	10/4/2021	10/5/2021	10/6/2021	10/7/2021	10/8/2021
Chin	10/6/2021			x	x	x
Gigi	10/5/2021		x		x	
Sioux	10/5/2021		x	x	x	
Chantel	10/4/2021	x	x	x	x	x
Billy	10/7/2021				x	x

Formula in C8:

=XLOOKUP("x",D8:H8,$D$7:$H$7)

Enter and copy to rows below.

Lookup last when there are duplicates: lookup last sale for employee



Goals: 1) Lookup last sale for each Sales Rep.

"Get Last" when there are duplicates and date column is sorted A-Z.

Date	Sales Rep	Sales
10/3/2021	Sioux	$640.56
10/3/2021	Chin	$1,706.02
10/4/2021	Sioux	$1,589.69
10/4/2021	Chin	$695.31
10/5/2021	Sioux	$474.26
10/5/2021	Chin	$786.13
10/6/2021	Sioux	$811.12
10/6/2021	Chin	$983.25
10/7/2021	Sioux	$696.97
10/7/2021	Chin	$765.99

Sales Rep	Last Sale
Sioux	696.97
Chin	765.99

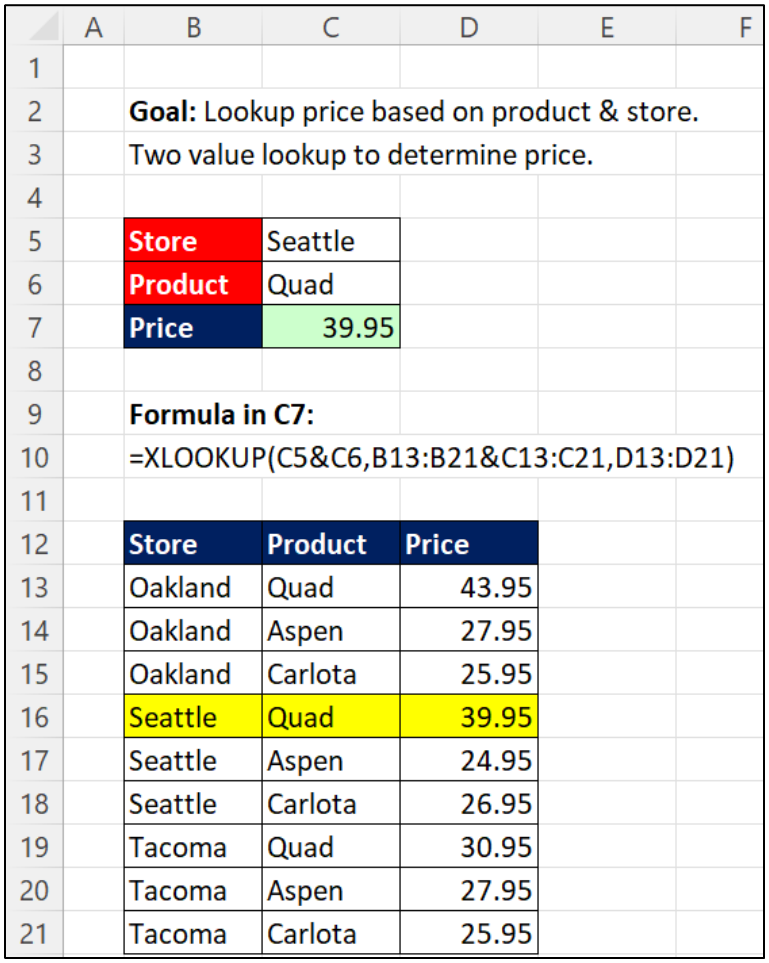
Formula in G6:

=XLOOKUP(F6:F7,C6:C15,D6:D15,,,-1)

Enter and it spills down to rows below.

Searching for the position last implies that the table has been sorted

Two lookup values: lookup product price for a specified city



Goal: Lookup price based on product & store.

Two value lookup to determine price.

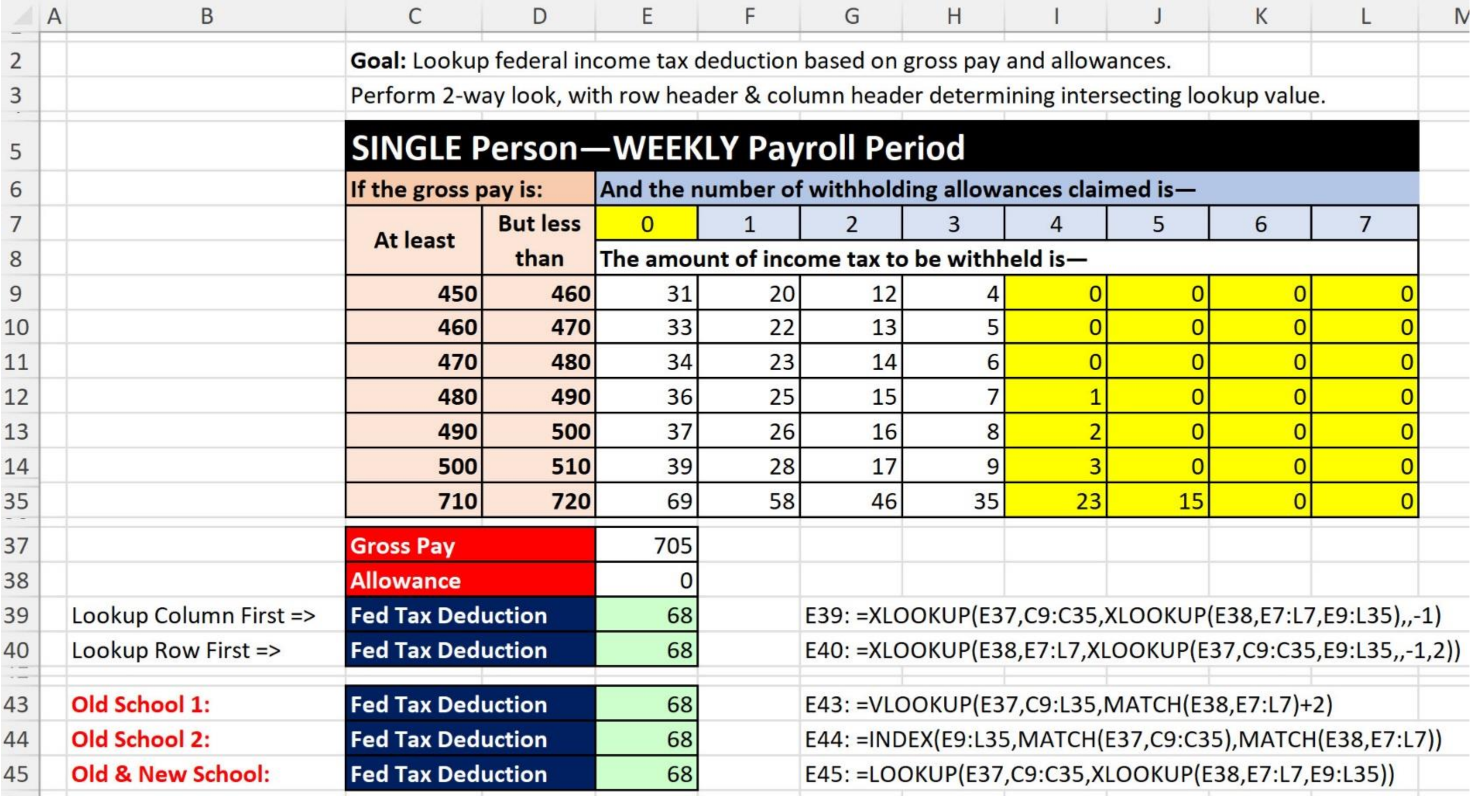
Store	Seattle
Product	Quad
Price	39.95

Formula in C7:

`=XLOOKUP(C5&C6,B13:B21&C13:C21,D13:D21)`

Store	Product	Price
Oakland	Quad	43.95
Oakland	Aspen	27.95
Oakland	Carlota	25.95
Seattle	Quad	39.95
Seattle	Aspen	24.95
Seattle	Carlota	26.95
Tacoma	Quad	30.95
Tacoma	Aspen	27.95
Tacoma	Carlota	25.95

Two-Way lookup: lookup tax based on weekly pay and allowances



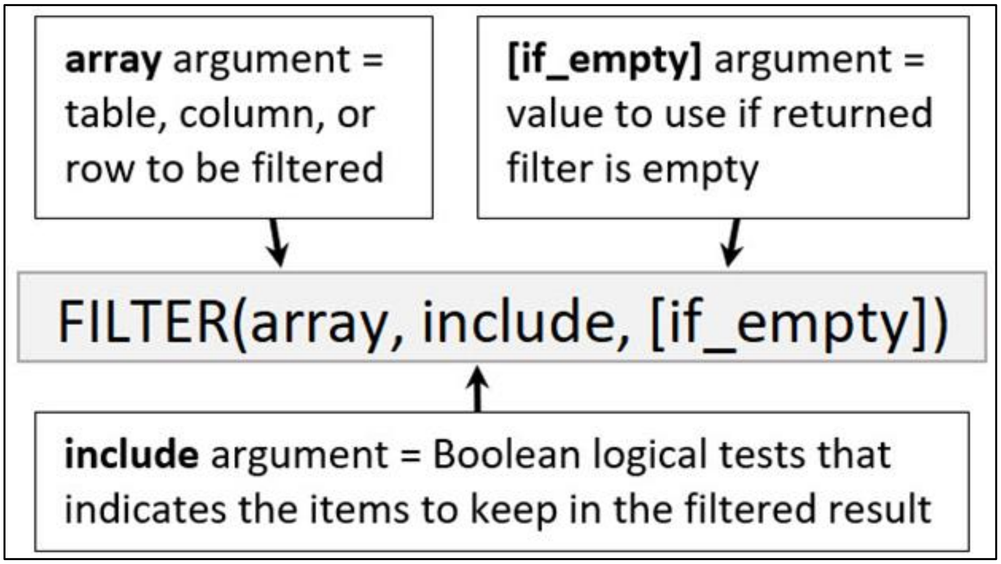
	A	B	C	D	E	F	G	H	I	J	K	L
2			**Goal:** Lookup federal income tax deduction based on gross pay and allowances.									
3			Perform 2-way look, with row header & column header determining intersecting lookup value.									
5			**SINGLE Person—WEEKLY Payroll Period**									
6			**If the gross pay is:**		**And the number of withholding allowances claimed is—**							
7			**At least**	**But less than**	0	1	2	3	4	5	6	7
8					**The amount of income tax to be withheld is—**							
9			**450**	**460**	31	20	12	4	0	0	0	0
10			**460**	**470**	33	22	13	5	0	0	0	0
11			**470**	**480**	34	23	14	6	0	0	0	0
12			**480**	**490**	36	25	15	7	1	0	0	0
13			**490**	**500**	37	26	16	8	2	0	0	0
14			**500**	**510**	39	28	17	9	3	0	0	0
35			**710**	**720**	69	58	46	35	23	15	0	0
37			**Gross Pay**		705							
38			**Allowance**		0							
39		Lookup Column First =>	**Fed Tax Deduction**		68		E39: =XLOOKUP(E37,C9:C35,XLOOKUP(E38,E7:L7,E9:L35),,-1)					
40		Lookup Row First =>	**Fed Tax Deduction**		68		E40: =XLOOKUP(E38,E7:L7,XLOOKUP(E37,C9:C35,E9:L35,,-1,2))					
43		**Old School 1:**	**Fed Tax Deduction**		68		E43: =VLOOKUP(E37,C9:L35,MATCH(E38,E7:L7)+2)					
44		**Old School 2:**	**Fed Tax Deduction**		68		E44: =INDEX(E9:L35,MATCH(E37,C9:C35),MATCH(E38,E7:L7))					
45		**Old & New School:**	**Fed Tax Deduction**		68		E45: =LOOKUP(E37,C9:C35,XLOOKUP(E38,E7:L7,E9:L35))					

FILTER array function arguments

Note: In Excel Basics video #2, we called a formula that makes a logical test on a column a “Array Logical Test”. In this section of the notes, I am referring to an “Array Logical Test” as “Boolean Array Logical Test” in reference to the great 19th century mathematician George Boole.

Note: For filtering data sets in place, you can use the filter feature (Data tab, Sort & Filter group). In addition, if you want to complete a quick filter on a data set and then copy and paste the records to a new location, the filter feature is easier to use than the FILTER function. However, the results from the filter feature do not instantly update when source data changes. When you use the FILTER function, or any worksheet formula, when the source data changes, the results from the FILTER function update instantly. This is the main advantage of the FILTER function over the filter feature.

The FILTER array function filters a dataset based on conditions and criteria that you specify in a Boolean Array Logical Test to deliver a resultant array of answers. The FILTER array function has the following syntax:


array argument = table, column, or row to be filtered
[if_empty] argument = value to use if returned filter is empty
FILTER(array, include, [if_empty])
include argument = Boolean logical tests that indicates the items to keep in the filtered result

The arguments of the FILTER function are as follows:

array**:** This argument contains the table, column, or row that you want to filter.

include**:** This argument contains a Boolean Array Logical Test that indicates which items to include and which items to not include in the resultant array of answers. This array must have the same number of rows or columns as the array that is being filtered.

[*if_empty*]: This argument contains the value you want the formula to return when the result of the filter is an empty filter (that is, with no records). Here are two examples.

- Single-cell message: =FILTER(B5:D11,(B5:B11=F5)*(C5:C11=F8),"NA")
- Three-column message: =FILTER(B5:D11,(B5:B11=F5)*(C5:C11=F8),{"None,"-","-"})

A Boolean Array Logical Test must evaluate to an array of TRUE, FALSE, or number values. If there are other values in the logical test resultant array, such as text values or errors, the FILTER function delivers a #VALUE! error. A TRUE value or any nonzero number means that the item should be included in the filtered array of answers. A FALSE value or zero means that the item should not be included in the filtered array of answers. Consider these examples (which can be found on the worksheet 3.4):

=FILTER({"Q";"C";"Q";"A"},{TRUE;FALSE;TRUE;FALSE}) = {"Q";"Q"}

=FILTER({"Q";"C";"Q";"A"},{1;0;1;0}) = {"Q";"Q"}

=FILTER({"Q";"C";"Q";"A"},{43;0;-2;0}) = {"Q";"Q"}

=FILTER({"Q";"C";"Q";"A"},{TRUE;0;-2;0}) = {"Q";"Q"}

=FILTER({"Q";"C";"Q";"A"},{TRUE;"";-2;"Text"}) = #VALUE!

If you are filtering by rows, a Boolean Array Logical Test must contain a single column that contains the same number of rows as there are in the item being filtered in the *array* argument. If the numbers of rows in the *array* and *include* arguments are not equivalent, the result is a #VALUE! error. Consider these examples (which can be found on the worksheet 3.4):

=FILTER({"Q";"C";"Q";"A"},{TRUE;FALSE;TRUE;FALSE}) = {"Q";"Q"}

=FILTER({"Q";"C";"Q";"A"},{TRUE;FALSE;TRUE}) = #VALUE!

If you are filtering by columns, a Boolean Array Logical Test must contain a single row that contains the same number of columns as there are in the item being filtered in the *array* argument. If the number of columns in the array and *include* arguments are not equivalent, the result is a #VALUE! error. Consider these examples (which can be found on the worksheet 3.4):

=FILTER({"Q","C","Q","A"},{TRUE,FALSE,TRUE,FALSE}) = {"Q","Q"}

=FILTER({"Q","C","Q","A"},{TRUE,FALSE,TRUE}) = #VALUE!

The include argument in the FILTER function can contain a Single-Condition Logical Test, an AND Logical Test, an OR Logical Test, a NOT Logical Test, and many other types of logical tests. Here is a list of examples of each of these different logical tests:

Single-condition logical test: =FILTER(B5:D11,B5:B11=F5)

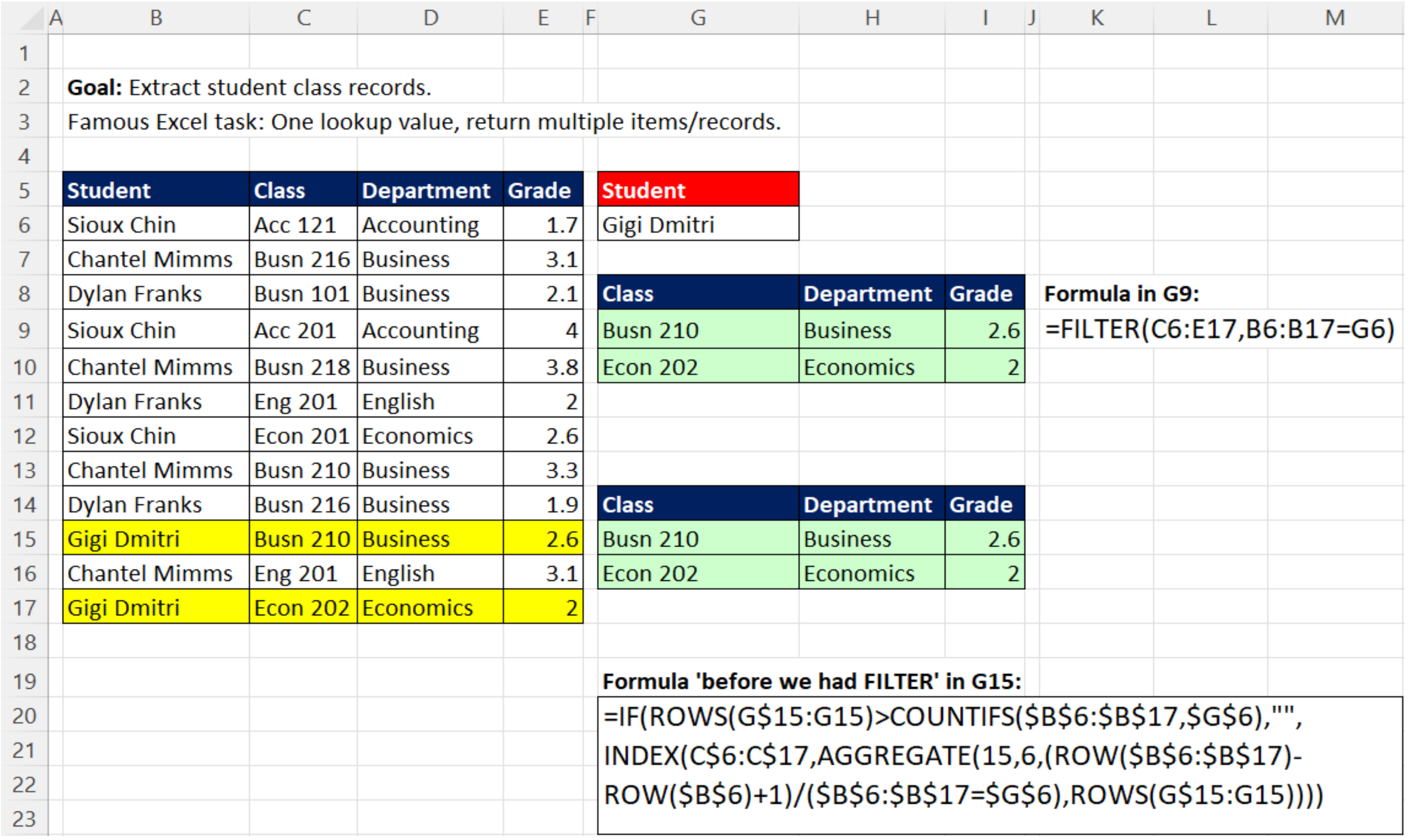
AND logical test using multiplication: =FILTER(B5:D11,(B5:B11=F5)*(C5:C11=F8))

OR logical test using addition: =FILTER(B5:D11,(B5:B11=F5)+(C5:C11=F8))

NOT logical test: =FILTER(B5:D11,B5:B11<>F5)

Use an AND and OR logical test together (complex filter): =FILTER(B11:B16,(C11:C16>C7)*(D11:D16>D7)*(E11:E16<E7)*((F11:F16>F7)+(G11:G16>G7)))

FILTER function to return multiple items from one lookup value



Goal: Extract student class records.

Famous Excel task: One lookup value, return multiple items/records.

Student	Class	Department	Grade
Sioux Chin	Acc 121	Accounting	1.7
Chantel Mimms	Busn 216	Business	3.1
Dylan Franks	Busn 101	Business	2.1
Sioux Chin	Acc 201	Accounting	4
Chantel Mimms	Busn 218	Business	3.8
Dylan Franks	Eng 201	English	2
Sioux Chin	Econ 201	Economics	2.6
Chantel Mimms	Busn 210	Business	3.3
Dylan Franks	Busn 216	Business	1.9
Gigi Dmitri	Busn 210	Business	2.6
Chantel Mimms	Eng 201	English	3.1
Gigi Dmitri	Econ 202	Economics	2

Student
Gigi Dmitri

Class	Department	Grade
Busn 210	Business	2.6
Econ 202	Economics	2

Formula in G9:

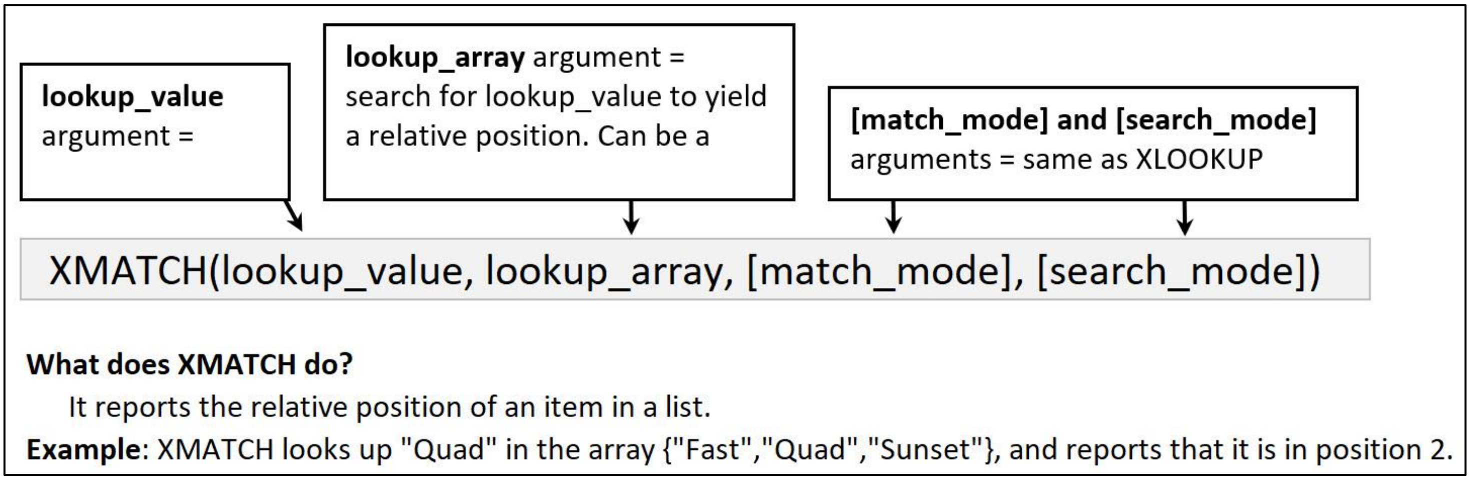
=FILTER(C6:E17,B6:B17=G6)

Class	Department	Grade
Busn 210	Business	2.6
Econ 202	Economics	2

Formula 'before we had FILTER' in G15:

```
=IF(ROWS(G$15:G15)>COUNTIFS($B$6:$B$17,$G$6),"",
INDEX(C$6:C$17,AGGREGATE(15,6,(ROW($B$6:$B$17)-
ROW($B$6)+1)/($B$6:$B$17=$G$6),ROWS(G$15:G15))))
```

XMATCH function arguments



lookup_value argument =

lookup_array argument = search for lookup_value to yield a relative position. Can be a

[match_mode] and [search_mode] arguments = same as XLOOKUP

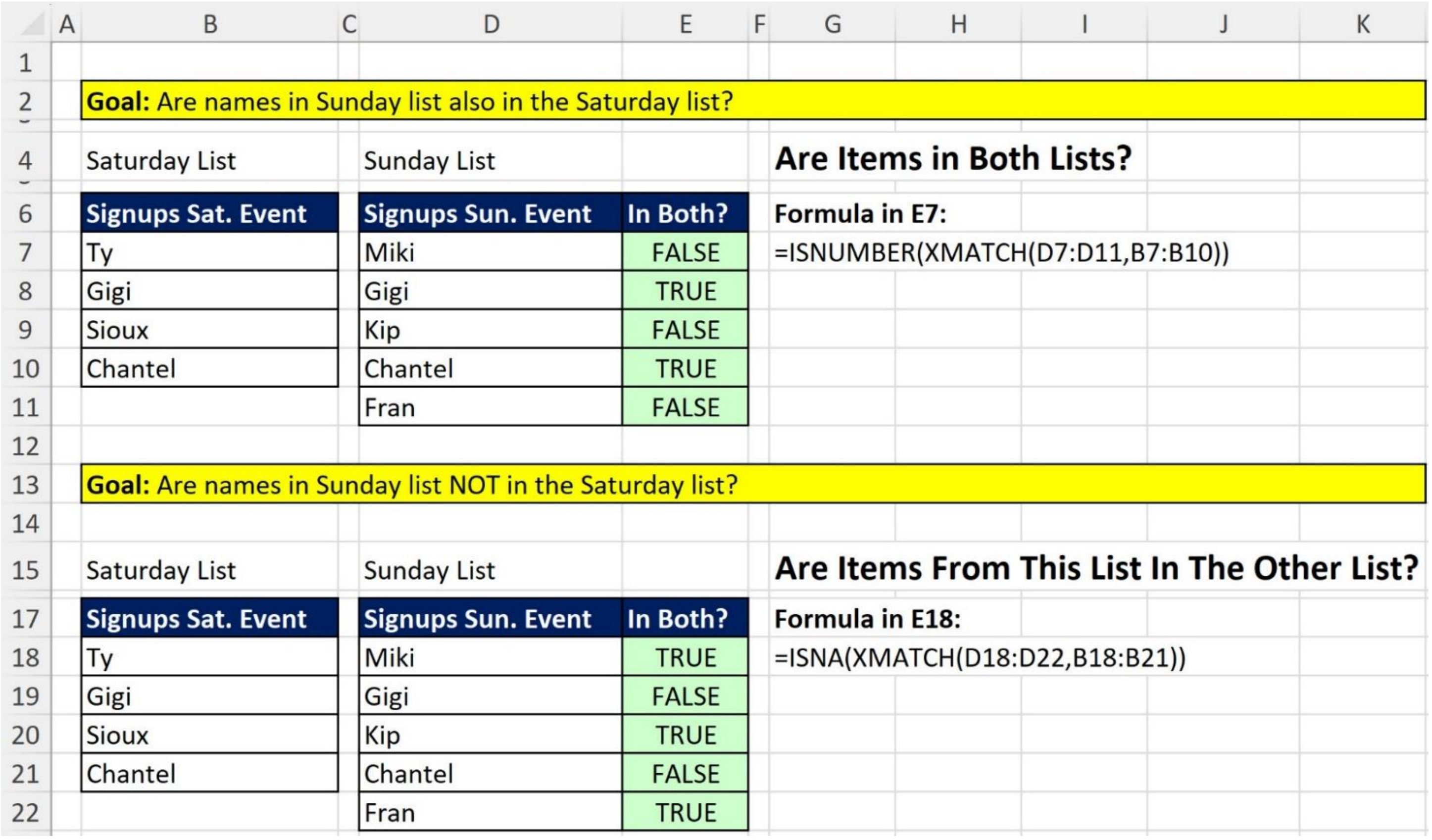
XMATCH(lookup_value, lookup_array, [match_mode], [search_mode])

What does XMATCH do?

It reports the relative position of an item in a list.

Example: XMATCH looks up "Quad" in the array {"Fast","Quad","Sunset"}, and reports that it is in position 2.

XMATCH lookup function to compare two lists



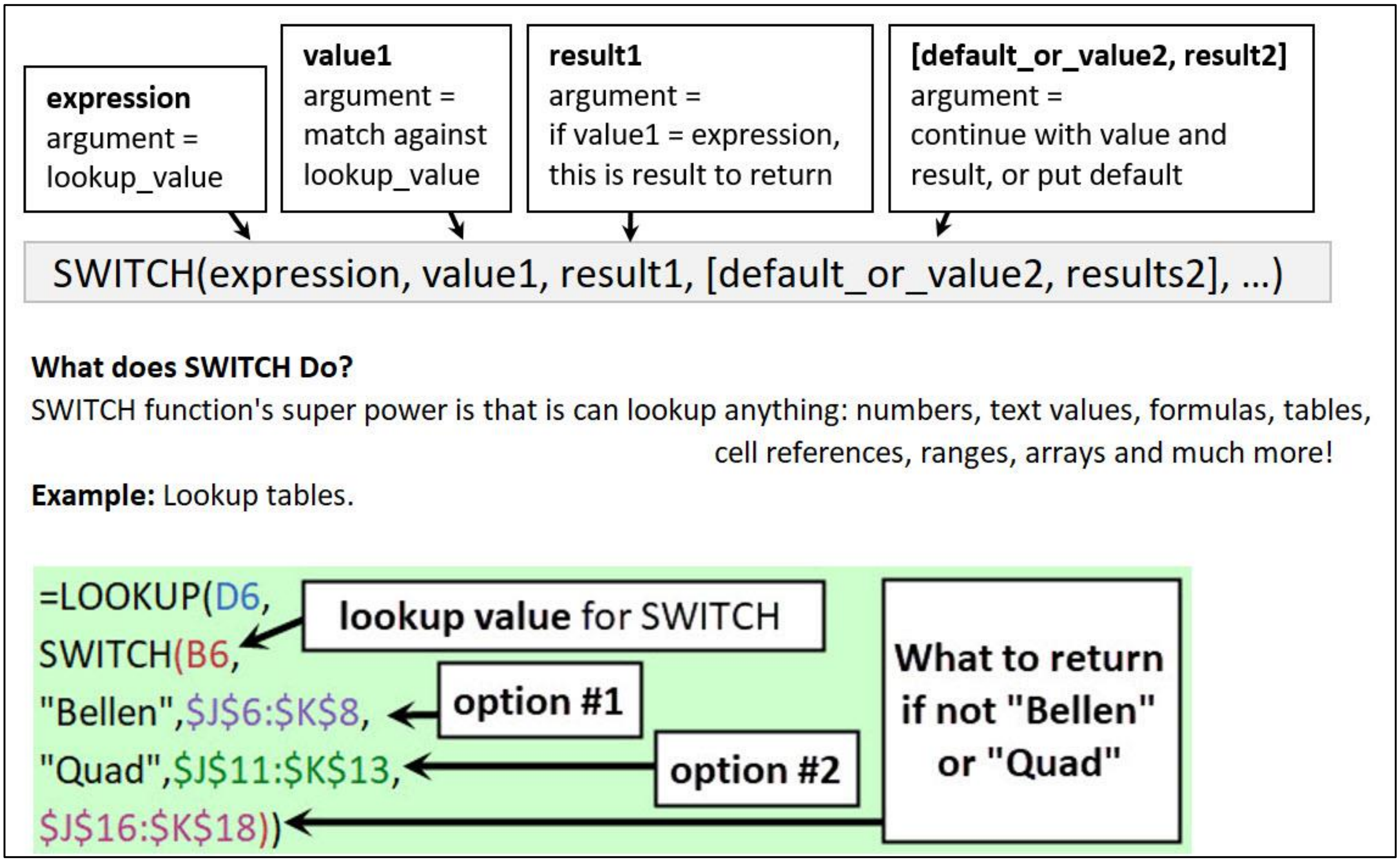
	A	B	C	D	E	F	G	H	I	J	K
1											
2		**Goal:** Are names in Sunday list also in the Saturday list?									
4		Saturday List		Sunday List			**Are Items in Both Lists?**				
6		**Signups Sat. Event**		**Signups Sun. Event**	**In Both?**		**Formula in E7:**				
7		Ty		Miki	FALSE		=ISNUMBER(XMATCH(D7:D11,B7:B10))				
8		Gigi		Gigi	TRUE						
9		Sioux		Kip	FALSE						
10		Chantel		Chantel	TRUE						
11				Fran	FALSE						
12											
13		**Goal:** Are names in Sunday list NOT in the Saturday list?									
14											
15		Saturday List		Sunday List			**Are Items From This List In The Other List?**				
17		**Signups Sat. Event**		**Signups Sun. Event**	**In Both?**		**Formula in E18:**				
18		Ty		Miki	TRUE		=ISNA(XMATCH(D18:D22,B18:B21))				
19		Gigi		Gigi	FALSE						
20		Sioux		Kip	TRUE						
21		Chantel		Chantel	FALSE						
22				Fran	TRUE						

SWITCH function arguments

The SWITCH lookup function is the Swiss Army knife of the lookup function family because it is not limited to values in cells, ranges, or arrays (as the XLOOKUP and LOOKUP functions are). With the SWITCH function, you can look up tables, formulas, functions, text values, number values, values in cells, and any other type of formula element that you can dream up. Because it is a function that can look up any type of formula element or value, SWITCH is also an efficient substitute for the IF or IFS functions in some situations.

The arguments of the SWITCH function are as follows:

- expression: This argument specifies the lookup value.
- value1, value2, etc.: These arguments specify the values to match.
- result1, result2, etc.: These arguments specify the values returned.
- default: This argument is delivered when no matches are found in the value arguments.


expression argument = lookup_value
value1 argument = match against lookup_value
result1 argument = if value1 = expression, this is result to return
[default_or_value2, result2] argument = continue with value and result, or put default
lookup value for SWITCH
option #1
option #2
What to return if not "Bellen" or "Quad"

```
SWITCH(expression, value1, result1, [default_or_value2, results2], ...)
```

What does SWITCH Do?

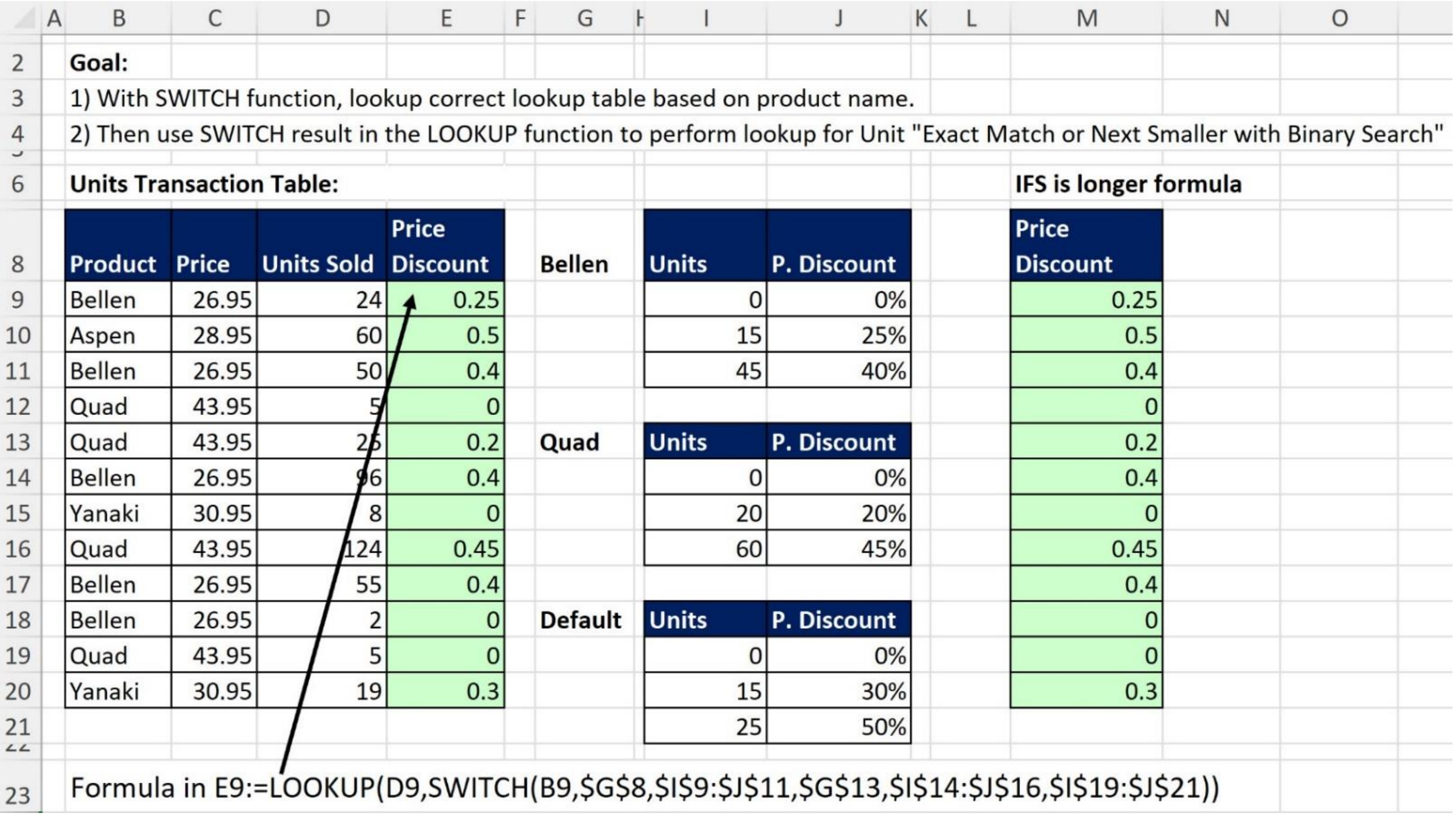
SWITCH function's super power is that is can lookup anything: numbers, text values, formulas, tables, cell references, ranges, arrays and much more!

Example: Lookup tables.

```
=LOOKUP(D6,
SWITCH(B6,
"Bellen",$J$6:$K$8,
"Quad",$J$11:$K$13,
$J$16:$K$18))
```

SWITCH to lookup the correct pricing lookup table based on product name

The Figure above shows a close-up of the SWITCH function used in next example, where you have to look up different lookup tables based on a product name. The relative cell reference B6 contains the name of the product sold. If the product in cell B6 is Bellen, then the table in the range J6:K8 is used; if the product is Quad, the table in the range J11:K13 is used; finally, if the product is not Bellen or Quad, the default table from the range J16:K18 is used.



Goal:

1) With SWITCH function, lookup correct lookup table based on product name.

2) Then use SWITCH result in the LOOKUP function to perform lookup for Unit "Exact Match or Next Smaller with Binary Search"

Units Transaction Table:

Product	Price	Units Sold	Price Discount
Bellen	26.95	24	0.25
Aspen	28.95	60	0.5
Bellen	26.95	50	0.4
Quad	43.95	5	0
Quad	43.95	25	0.2
Bellen	26.95	96	0.4
Yanaki	30.95	8	0
Quad	43.95	124	0.45
Bellen	26.95	55	0.4
Bellen	26.95	2	0
Quad	43.95	5	0
Yanaki	30.95	19	0.3

Bellen

Units	P. Discount
0	0%
15	25%
45	40%

Quad

Units	P. Discount
0	0%
20	20%
60	45%

Default

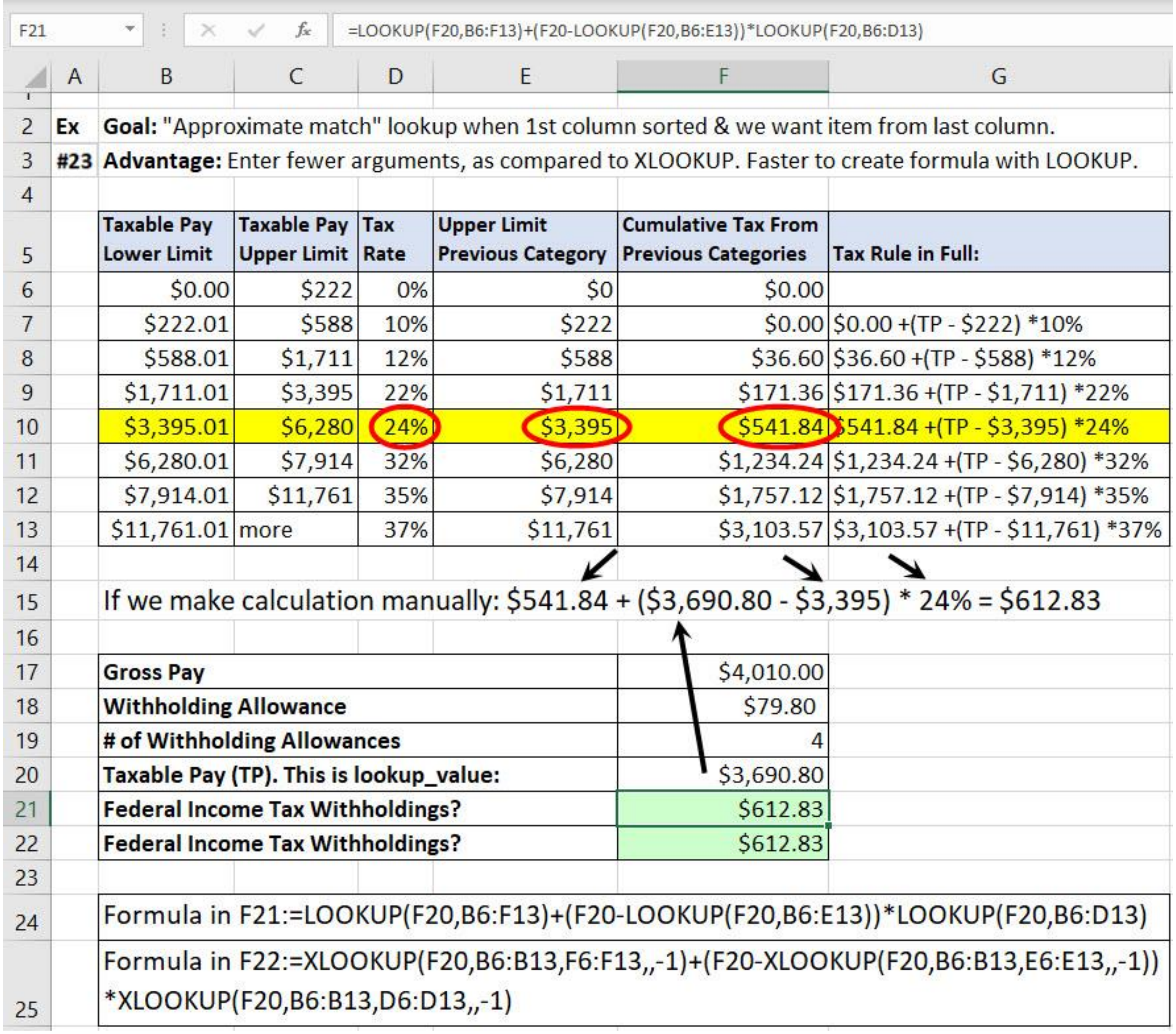
Units	P. Discount
0	0%
15	30%
25	50%

IFS is longer formula

Price Discount
0.25
0.5
0.4
0
0.2
0.4
0
0.45
0.4
0
0
0.3

Formula in E9:=LOOKUP(D9,SWITCH(B9,$G$8,$I$9:$J$11,$G$13,$I$14:$J$16,$I$19:$J$21))

LOOKUP function to create formula much more quickly for tax calculations



F21 =LOOKUP(F20,B6:F13)+(F20-LOOKUP(F20,B6:E13))*LOOKUP(F20,B6:D13)

Ex #23 **Goal:** "Approximate match" lookup when 1st column sorted & we want item from last column.

Advantage: Enter fewer arguments, as compared to XLOOKUP. Faster to create formula with LOOKUP.

Taxable Pay Lower Limit	Taxable Pay Upper Limit	Tax Rate	Upper Limit Previous Category	Cumulative Tax From Previous Categories	Tax Rule in Full:
$0.00	$222	0%	$0	$0.00	
$222.01	$588	10%	$222	$0.00	$0.00 +(TP - $222) *10%
$588.01	$1,711	12%	$588	$36.60	$36.60 +(TP - $588) *12%
$1,711.01	$3,395	22%	$1,711	$171.36	$171.36 +(TP - $1,711) *22%
$3,395.01	$6,280	24%	$3,395	$541.84	$541.84 +(TP - $3,395) *24%
$6,280.01	$7,914	32%	$6,280	$1,234.24	$1,234.24 +(TP - $6,280) *32%
$7,914.01	$11,761	35%	$7,914	$1,757.12	$1,757.12 +(TP - $7,914) *35%
$11,761.01	more	37%	$11,761	$3,103.57	$3,103.57 +(TP - $11,761) *37%

If we make calculation manually: $541.84 + ($3,690.80 - $3,395) * 24% = $612.83

Gross Pay	$4,010.00
Withholding Allowance	$79.80
# of Withholding Allowances	4
Taxable Pay (TP). This is lookup_value:	$3,690.80
Federal Income Tax Withholdings?	$612.83
Federal Income Tax Withholdings?	$612.83

Formula in F21:=LOOKUP(F20,B6:F13)+(F20-LOOKUP(F20,B6:E13))*LOOKUP(F20,B6:D13)

Formula in F22:=XLOOKUP(F20,B6:B13,F6:F13,,-1)+(F20-XLOOKUP(F20,B6:B13,E6:E13,,-1))*XLOOKUP(F20,B6:B13,D6:D13,,-1)