

Lookup Functions: VLOOKUP, XLOOKUP and INDEX-MATCH

Bring data together from different tables. Master VLOOKUP and HLOOKUP (exact and approximate match), the modern XLOOKUP, the flexible INDEX + MATCH combination (including two-way and reverse lookups), OFFSET, consolidating several sheets, CHOOSE, and how to troubleshoot

● Live classes 4-5

✦ Excel for Microsoft 365, 2026

Intermediate

🕒 120 min

By the end of this lesson you can

- Explain why lookups are the Excel equivalent of a SQL join
- Use VLOOKUP and HLOOKUP with exact and approximate match, and know their limits
- Use XLOOKUP for left lookups, default values, last-match search and multiple return columns
- Combine INDEX and MATCH for reverse (left) and two-way (matrix) lookups
- Use CHOOSE, and fix the common causes of

Datasets used

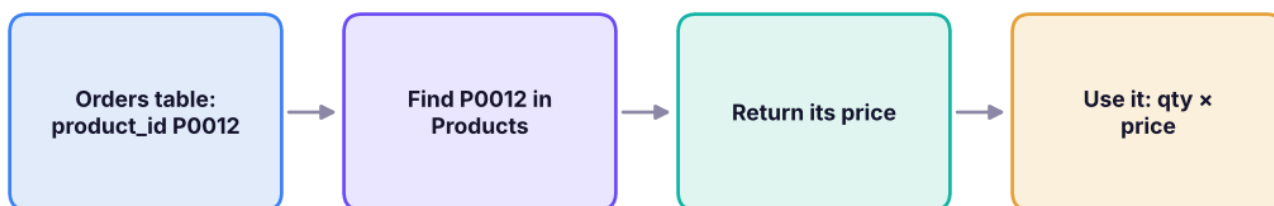
products.csv

superstore_sales.csv

Why lookups matter

Business data is split across tables: orders in one, products in another, customers in a third. A **lookup** fetches a value from one table using a key that both tables share, such as a product ID or employee ID.

How every lookup works



A lookup takes a key from one table, finds it in another table, and returns a value from the same row.

TIP You are already learning SQL joins

A lookup is a **left join** for one column at a time: keep every row of the main table and bring in matching information from the other table. You will see the same idea as `JOIN` in SQL (SQL Lesson 8), `merge()` in pandas, and relationships in Power BI.

We will use this lookup table on a sheet called `Prod` :

X Prod: product lookup table (A1:E7)

	A	B	C	D	E
1	Product ID	Product	Category	Price	Stock
2	P101	Office Chair	Furniture	4500	32
3	P102	Standing Desk	Furniture	18500	8
4	P103	Laser Printer	Technology	12900	15
5	P104	Notebook Pack	Office Supplies	240	540
6	P105	Smartphone	Technology	21999	27
7	P106	Bookcase	Furniture	7600	0

VLOOKUP

`=VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])`

- **lookup_value**: what you are looking for (the key).
- **table_array**: the table to search; the key **must be in its first column**.
- **col_index_num**: which column of that table to return (1 = first column).
- **range_lookup**: FALSE (or 0) for an **exact** match. Use TRUE only for approximate matching on sorted bands.

X Order lines looking up product details with VLOOKUP

	A	B	C	D	E	F
1	Order	Product ID	Qty	Product =VLOOKUP(B2,Prod!\$A\$2:\$E\$7,2,FALSE)	Price	Amount
2	1001	P103	2	Laser Printer	12,900	25,800
3	1002	P101	6	Office Chair	4,500	27,000
4	1003	P105	1	Smartphone	21,999	21,999
5	1004	P104	20	Notebook Pack	240	4,800
6	1005	P109	3	#N/A	#N/A	#N/A

D2 `=VLOOKUP(B2,Prod!$A$2:$E$7,2,FALSE)`

F2 `=C2*E2`

Each pattern is copied down/across to the other cells in its column.

Row 6 shows #N/A because P109 does not exist in the product table. That is the correct answer: the key is not there.

COMMON MISTAKES The four classic VLOOKUP mistakes

1. **Forgetting FALSE:** the default is approximate match, which silently returns a *wrong* row for unsorted data.
2. **Not locking the table:** A2:E7 copied down becomes A3:E8, so the last rows are missed. Use \$A\$2:\$E\$7 or a Table name.
3. **Hard-coded column numbers:** inserting a column inside the table changes which column 4 means. Use MATCH for the column number, or XLOOKUP.
4. **Key not in the first column:** VLOOKUP cannot look to the left (use INDEX + MATCH or XLOOKUP).

Approximate match: bands and slabs

With TRUE, VLOOKUP finds the **largest value less than or equal to** the lookup value. The first column must be sorted ascending. This is perfect for tax slabs, commission tiers, discount bands and grades.

X Approximate match for commission tiers

	A	B	C	D	E	F	G
1	Sales from	Commission		Rep	Sales	Rate	Commission
2	0	0%		Asha	85,000	0%	0
3	100000	3%		Ravi	240,000	3%	7,200
4	250000	5%		Meera	250,000	5%	12,500
5	500000	7%		Kabir	610,000	7%	42,700
6	1000000	10%		Zoya	1,200,000	10%	120,000

F2 =VLOOKUP(E2,\$A\$2:\$B\$6,2,TRUE)

G2 =E2*F2

Each pattern is copied down/across to the other cells in its column.

Ravi's 2,40,000 falls between 1,00,000 and 2,50,000, so he gets the 1,00,000 row: 3%. Meera's 2,50,000 matches the 2,50,000 row exactly: 5%.

HLOOKUP

HLOOKUP is VLOOKUP turned sideways: it searches the **first row** and returns a value from a row below. Use it when data runs across columns (for example months as headers).

X HLOOKUP across a horizontal table

	A	B	C	D	E	F	G	H
1	Month	Jan	Feb	Mar	Apr		Find month	Mar
2	Target	120	130	145	150		Target →	145
3	Actual	118	136	139	162		Actual →	139

H2 =HLOOKUP(H1,\$A\$1:\$E\$3,2,FALSE)

H3 =HLOOKUP(H1,\$A\$1:\$E\$3,3,FALSE)

XLOOKUP: the modern lookup (Microsoft 365 / Excel 2021+)

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

XLOOKUP fixes every VLOOKUP limitation:

Feature	VLOOKUP	XLOOKUP
Default match	Approximate (risky)	Exact
Look to the left	No	Yes: lookup and return arrays are separate
Column inserted in the table	Breaks the index number	Safe
Custom "not found" message	Needs IFERROR	Built-in 4th argument
Return several columns	One at a time	Spills many columns
Search last to first	No	<code>search_mode = -1</code>
Wildcards	Yes	<code>match_mode = 2</code>
Next smaller / larger match	Sorted data only	<code>match_mode = -1 or 1, any order</code>

XLOOKUP examples (lookup table on sheet Prod)

	A	B	C
1	Need	Formula	Result
2	Price of P105	<code>=XLOOKUP("P105",Prod!A2:A7,Prod!D2:D7)</code>	21,999
3	ID of "Bookcase" (left lookup)	<code>=XLOOKUP("Bookcase",Prod!B2:B7,Prod!A2:A7)</code>	P106
4	Missing ID with a message	<code>=XLOOKUP("P109",Prod!A2:A7,Prod!B2:B7,"Not found")</code>	Not found
5	Product, category and price at once	<code>=XLOOKUP("P103",Prod!A2:A7,Prod!B2:D7)</code>	Laser Printer
6	Last Furniture item in the list	<code>=XLOOKUP("Furniture",Prod!C2:C7,Prod!B2:B7,,0,-1)</code>	Bookcase
7	Product name containing "Desk"	<code>=XLOOKUP("*Desk*",Prod!B2:B7,Prod!D2:D7,,2)</code>	18,500

C2 `=XLOOKUP("P105",Prod!A2:A7,Prod!D2:D7)`
C3 `=XLOOKUP("Bookcase",Prod!B2:B7,Prod!A2:A7)`
C4 `=XLOOKUP("P109",Prod!A2:A7,Prod!B2:B7,"Not found")`
C5 `=XLOOKUP("P103",Prod!A2:A7,Prod!B2:D7)`
C6 `=XLOOKUP("Furniture",Prod!C2:C7,Prod!B2:B7,,0,-1)`
C7 `=XLOOKUP("*Desk*",Prod!B2:B7,Prod!D2:D7,,2)`

The fourth row returns three columns at once: the result spills across C5:E5.

XLOOKUP for bands (no sorting needed)

`match_mode = -1` returns the exact match or the **next smaller** item, like VLOOKUP's TRUE, but the table does not need to be sorted:

`=XLOOKUP(E2, $A$2:$A$6, $B$2:$B$6, , -1)` gives the same commission rates as the approximate VLOOKUP above.

TIP XMATCH

XMATCH is the modern version of MATCH with the same match and search modes as XLOOKUP.

=XMATCH("P105",Prod!A2:A7) → 5.

INDEX and MATCH

Before XLOOKUP, professionals used **INDEX + MATCH**, and you will still see it everywhere. It works in every Excel version and is very flexible.

- =INDEX(array, row_num, [col_num]) returns the value **at a position**.
- =MATCH(lookup_value, lookup_array, [match_type]) returns the **position** of a value. Use 0 for exact match.

X INDEX returns a value at a position; MATCH returns the position of a value

	A	B	C	D	E	F
1	State	City	Population 2011 (mn)		Formula	Result
2	Maharashtra	Mumbai	18.4		=INDEX(A2:C6,2,1)	Delhi
3	Delhi	Delhi	16.3		=MATCH("Ahmedabad",B2:B6,0)	3
4	Gujarat	Ahmedabad	6.4		=MATCH("Population 2011 (mn)",A1:C1,0)	3
5	West Bengal	Kolkata	14.1		=INDEX(C2:C6,MATCH("Kolkata",B2:B6,0))	14.1
6	Karnataka	Bengaluru	8.5		=INDEX(A2:A6,MATCH("Ahmedabad",B2:B6,0))	Gujarat

F2 =INDEX(A2:C6,2,1)

F3 =MATCH("Ahmedabad",B2:B6,0)

F4 =MATCH("Population 2011 (mn)",A1:C1,0)

F5 =INDEX(C2:C6,MATCH("Kolkata",B2:B6,0))

F6 =INDEX(A2:A6,MATCH("Ahmedabad",B2:B6,0))

Reverse (left) lookup

In the table above, City is in column B and State is in column A. VLOOKUP cannot return a column to the **left** of the key, but INDEX + MATCH can: =INDEX(A2:A6, MATCH("Ahmedabad", B2:B6, 0)) → **Gujarat**.

	A: State	B: City
1	Maharashtra	Mumbai
2	Delhi	Delhi
3	Gujarat	Ahmedabad
4	West Bengal	Kolkata
5	Karnataka	Bengaluru

1. `MATCH("Ahmedabad", B2:B6, 0) = 3`

2. `INDEX(A2:A6, 3) = "Gujarat"`

INDEX + MATCH: MATCH finds the row number of the city; INDEX returns the state on that row.

Two-way (matrix) lookup

INDEX can take a row **and** a column number. Use one MATCH for each to look up the intersection of a row and a column, for example an employee's salary from a table where both the employee and the field are chosen in drop-downs.

X Two-way lookup: <code>=INDEX(\$B\$2:\$E\$6, MATCH(\$G3,\$A\$2:\$A\$6,0), MATCH(H\$2,\$B\$1:\$E\$1,0))</code>									
	A	B	C	D	E	F	G	H	I
1	EmpName	DepartmentID	JobTitle	Salary	City				
2	Employee_1	D10	Developer	82000	Pune		Employee	JobTitle	Salary
3	Employee_2	D20	Analyst	68000	Delhi		Employee_4	Manager	125000
4	Employee_3	D10	Tester	54000	Pune		Employee_2	Analyst	68000
5	Employee_4	D30	Manager	125000	Mumbai				
6	Employee_5	D20	Developer	79000	Delhi				

H3 `=INDEX($B$2:$E$6, MATCH($G3,$A$2:$A$6,0), MATCH(H$2,$B$1:$E$1,0))`
 I3 `=INDEX($B$2:$E$6, MATCH($G3,$A$2:$A$6,0), MATCH(I$2,$B$1:$E$1,0))`
 H4 `=INDEX($B$2:$E$6, MATCH($G4,$A$2:$A$6,0), MATCH(H$2,$B$1:$E$1,0))`
 I4 `=INDEX($B$2:$E$6, MATCH($G4,$A$2:$A$6,0), MATCH(I$2,$B$1:$E$1,0))`

Because the headers in row 2 (JobTitle, Salary) are matched by MATCH, you can change a header to City and the formula returns cities without editing it. The XLOOKUP version nests one XLOOKUP inside another:
`=XLOOKUP($G3,$A$2:$A$6,XLOOKUP(H$2,$B$1:$E$1,$B$2:$E$6)).`

INDEX + MATCH with multiple conditions

To match on two columns at once (for example Region **and** Product), multiply the conditions inside MATCH:

`=INDEX(F2:F9, MATCH(1, (B2:B9="North")*(C2:C9="Tablet"), 0))` → returns the revenue for the North + Tablet row. In Microsoft 365 this works as a normal formula; in older Excel press Ctrl + Shift + Enter.
 With XLOOKUP: `=XLOOKUP(1, (B2:B9="North")*(C2:C9="Tablet"), F2:F9).`

OFFSET and consolidating worksheets

In class we also used lookups to **consolidate** data: pull values from several tables or sheets into one summary. Three tools do this job.

OFFSET: a value or range by position

=OFFSET(reference, rows, cols, [height], [width]) starts at reference, moves down rows and right cols, and returns that cell (or a range of height × width). Combined with MATCH it does the same two-way lookup as INDEX + MATCH. The table below holds month-end share prices (illustrative numbers for fictional companies).

X

Two-way lookup of a share price with INDEX + MATCH and with OFFSET + MATCH

	A	B	C	D	E	F	G	H	
1	Date	Asha Foods	Kiran Motors	Meru Steel		Company	Date	INDEX + MATCH	
2	31-01-2026	412	1180	96		Kiran Motors	31-03-2026	1262	
3	28-02-2026	398	1215	101		Meru Steel	31-05-2026	112	
4	31-03-2026	425	1262	99		Asha Foods	28-02-2026	398	
5	30-04-2026	440	1240	108				Last 3 months avg (Asha)	
6	31-05-2026	436	1298	112					

H2

=INDEX(\$B\$2:\$D\$6,MATCH(G2,\$A\$2:\$A\$6,0),MATCH(F2,\$B\$1:\$D\$1,0))

I2

=OFFSET(\$A\$1,MATCH(G2,\$A\$2:\$A\$6,0),MATCH(F2,\$B\$1:\$D\$1,0))

I5

=AVERAGE(OFFSET(\$B\$1,COUNT(\$B\$2:\$B\$6)-2,0,3,1))

Each pattern is copied down/across to the other cells in its column.

The last formula shows OFFSET's real strength: OFFSET(\$B\$1, COUNT(B2:B6)-2, 0, 3, 1) is a range of the **last three** prices, so the average updates automatically when a new month is added. The same idea builds dynamic named ranges and charts (Lesson 9).

COMMON MISTAKES

OFFSET is volatile

OFFSET (like INDIRECT, TODAY and RAND) recalculates every time anything in the workbook changes. A few hundred OFFSETs are fine; tens of thousands slow a workbook down. For lookups prefer INDEX + MATCH or XLOOKUP, and for growing ranges prefer an Excel Table.

Consolidating several sheets into one

Situation	Best tool	How
Same layout on every sheet (Jan, Feb, Mar...) and you need totals	3-D reference	=SUM(Jan:Mar!B2) adds cell B2 on every sheet from Jan to Mar (Lesson 4)
Same kind of data, but rows in a different order on each sheet	Data → Consolidate	Choose Sum, add each sheet's range, tick <i>Top row</i> and <i>Left column</i> , optionally <i>Create links to source data</i>
Different tables that share a key (city code, product ID)	Lookups	Bring each column in with XLOOKUP or INDEX + MATCH, as in the class tasks below
Many files every month	Power Query	Get Data → From Folder, then Append (Lesson 12)

Class practice: city tables

These are the consolidation tasks from the live class. Table 1 maps cities to states and population; Table 2 maps city codes to cities.

X Consolidation tasks: state from city, population from city code, grade from marks							
	A	B	C	D	E	F	G
1	City Code	City	State	Population 2011 (mn)		Task	Input
2	C1	Mumbai	Maharashtra	12.48		State of this city	Pune
3	C2	Delhi	Delhi	11.01		Population of city code	C31
4	C3	Ahmedabad	Gujarat	5.57		Population of city code	C4
5	C4	Surat	Gujarat	4.47		City code of this city	Allahabad
6	C31	Allahabad	Uttar Pradesh	1.11			
7	C21	Pune	Maharashtra	3.12		Marks	Grade (approx. match)
8						0	D
9						58	C
10						91	A
11						Min marks	Grade
12						0	D
13						40	C
14						60	B
15						80	A

H2 =IFNA(INDEX(\$C\$2:\$C\$7,MATCH(G2,\$B\$2:\$B\$7,0)),"Not found")

H3 =IFNA(INDEX(\$D\$2:\$D\$7,MATCH(G3,\$A\$2:\$A\$7,0)),"Not found")

H5 =IFNA(INDEX(\$A\$2:\$A\$7,MATCH(G5,\$B\$2:\$B\$7,0)),"Not found")

G8 =VLOOKUP(F8,\$F\$12:\$G\$15,2,TRUE)

Each pattern is copied down/across to the other cells in its column.

The state and city-code answers use INDEX + MATCH wrapped in IFNA; =XLOOKUP(G2,B2:B7,C2:C7,"Not found") gives the same result in Microsoft 365. The grade lookup uses **approximate match** (TRUE): VLOOKUP finds the largest "Min marks" that is less than or equal to the marks, so the grade table must be sorted ascending. This is the classic use of approximate match: tax slabs, commission bands and grades.

CHOOSE, and VLOOKUP with CHOOSE

=CHOOSE(index_num, value1, value2, ...) returns the item at a position.

X CHOOSE examples			
	A	B	C
1	Input	Formula	Result
2	2	=CHOOSE(A2,"Bronze","Silver","Gold")	Silver
3	27-04-2026	=CHOOSE(WEEKDAY(A3,2),"Mon","Tue","Wed","Thu","Fri","Sat","Sun")	Mon
4	15-08-2026	= "Q"&CHOOSE(MONTH(A4),4,4,4,1,1,1,2,2,3,3,3)	Q2

C2 =CHOOSE(A2,"Bronze","Silver","Gold")

C3 =CHOOSE(WEEKDAY(A3,2),"Mon","Tue","Wed","Thu","Fri","Sat","Sun")

```
C4 ="Q"&CHOOSE(MONTH(A4),4,4,4,1,1,1,2,2,2,3,3,3)
```

The last example gives the **Indian financial-year quarter** (April–June = Q1), which a plain `ROUNDUP(MONTH())/3` cannot do.

VLOOKUP + CHOOSE was the old trick for a left lookup: `CHOOSE({1,2}, B2:B6, A2:A6)` builds a virtual two-column table with City first and State second, so VLOOKUP can search it:

```
=VLOOKUP("Ahmedabad", CHOOSE({1,2}, B2:B6, A2:A6), 2, FALSE) → Gujarat.
```

Today, prefer XLOOKUP or INDEX + MATCH: they are easier to read and faster.

Troubleshooting #N/A

Cause	How to check	Fix
The key really is missing	Search the lookup column with Ctrl + F	Return a message: <code>IFNA(..., "Not found")</code> or XLOOKUP's 4th argument
Extra spaces	<code>=LEN(A2)</code> is longer than expected	TRIM both sides, or <code>XLOOKUP(TRIM(B2), TRIM(Prod!A2:A7), ...)</code>
Number vs text (101 vs "101")	<code>=ISTEXT(B2)</code> differs from the lookup column	Convert one side: <code>VALUE()</code> or <code>TEXT(..., "0")</code>
Approximate match on unsorted data	Wrong but no error	Use exact match (<code>FALSE / 0</code>)
Range not locked	Errors appear only in lower rows	Add \$ or use a Table
Invisible characters	<code>=CODE(RIGHT(B2))</code> returns 160 or 10	SUBSTITUTE and CLEAN

2026 UPDATE Lookups in 2026

- **XLOOKUP** and **XMATCH** are now the standard; use VLOOKUP only when sharing files with very old Excel versions.
- **FILTER** returns **all** matching rows, not just the first (Lesson 6).
- For joining large tables regularly, **Power Query → Merge Queries** (Lesson 12) is faster and repeatable, and it is exactly a SQL join.

Case: pricing the Superstore order lines

Suppose the order lines arrive without a price and you must look it up from the product master (`products.csv`). This is a lookup on 6,323 rows. pandas does the same as XLOOKUP with merge:

Python

```
import pandas as pd
orders = pd.read_csv("superstore_sales.csv")[["order_id", "product_id", "quantity"]]
products = pd.read_csv("products.csv")
priced = orders.merge(products[["product_id", "product_name", "list_price"]], on="product_id",
how="left") # = XLOOKUP for every row
priced["gross_amount"] = priced.quantity * priced.list_price
print("Rows:", len(priced), "| unmatched product IDs:", priced.list_price.isna().sum())
priced.head(5)
```

OUTPUT

Rows: 6323 | unmatched product IDs: 0

	order_id	product_id	quantity	product_name	list_price	gross_amount
0	IN-2023-00001	P0059	2	Nova Phone 223	18,000	36,000
1	IN-2023-00001	P0050	1	Orbit Appliance 798	3,850	3,850
2	IN-2023-00001	P0011	4	Orbit Table 248	15,520	62,080
3	IN-2023-00002	P0037	2	Crest Storage 946	3,260	6,520
4	IN-2023-00003	P0018	3	Crest Bookcase 170	11,710	35,130

5 rows × 6 columns

Zero unmatched IDs tells you every order line found its product: the same check you should do in Excel with =COUNTIF(price_column, "#N/A") or =SUM(--ISNA(price_column)).

Summary

KEY TAKEAWAYS

- Lookups bring columns from one table into another through a shared key, like a SQL left join.
- VLOOKUP: key in the first column, column number counted from 1, **FALSE for exact match**, lock the table with \$.
- Approximate match (TRUE / match_mode -1) is for sorted bands: tax slabs, commissions, grades.
- XLOOKUP is exact by default, looks left, has a built-in "not found" value, returns several columns and can search from the end.
- INDEX returns a value at a position; MATCH finds the position. Together they do reverse and two-way lookups in any Excel version.
- Most #N/A errors come from missing keys, spaces, or number-vs-text mismatches.

Quick quiz

6 questions · tap an answer to check it

1. =VLOOKUP("P103",A2:E7,4,FALSE) returns a value from which column of the range?

A A

B C

C D

D E

Answer: C. Column index 4 counts from the first column of the table array: A=1, B=2, C=3, D=4.

2. Why does VLOOKUP return the wrong price (not #N/A) for unsorted data?

A The fourth argument was omitted, so it used approximate match

B The table was not locked with \$

C The column number was too large

D VLOOKUP cannot handle text keys

Answer: A. Without FALSE, VLOOKUP uses approximate match, which assumes sorted data.

3. Which formula returns the State for the city in H2, where State is column A and City is column B?

A =VLOOKUP(H2,A2:B6,1,FALSE)

B =INDEX(A2:A6,MATCH(H2,B2:B6,0))

C =MATCH(H2,A2:A6,0)

D =HLOOKUP(H2,A2:B6,1,FALSE)

Answer: B. MATCH finds the row of the city in column B; INDEX returns column A on that row.

4. Sales of 2,40,000 with tiers at 0, 1,00,000, 2,50,000 and 5,00,000 (approximate match). Which tier is used?

A 0

B 1,00,000

C 2,50,000

D 5,00,000

Answer: B. Approximate match returns the largest tier less than or equal to the value.

5. What does XLOOKUP's 4th argument do?

A Sets the match mode

B Returns a value when nothing is found

C Searches from last to first

D Returns several columns

Answer: B. if_not_found replaces #N/A with your own value or message.

6. A lookup returns #N/A even though the ID is visible in both tables. Which is NOT a likely cause?

A One side has trailing spaces

B One side is a number and the other is text

C The workbook is saved as .xlsx

D There is an invisible non-breaking space

Answer: C. File format does not affect lookups. Spaces, invisible characters and type mismatches are the usual culprits.

EXERCISE Grade lookup

Create a grade table: 0 → F, 40 → D, 55 → C, 70 → B, 85 → A. Write a formula that returns the grade for a mark in B2 using (a) VLOOKUP, (b) XLOOKUP and (c) INDEX + MATCH.

Show solution (worked answer)

With the table in E2:F6 (marks ascending): (a) =VLOOKUP(B2,\$E\$2:\$F\$6,2,TRUE) (b) =XLOOKUP(B2,\$E\$2:\$E\$6,\$F\$2:\$F\$6,, -1) (c) =INDEX(\$F\$2:\$F\$6,MATCH(B2,\$E\$2:\$E\$6,1)) (match_type 1 = largest value ≤ B2, needs ascending order).

EXERCISE Two-way lookup

Make a table of 4 regions (rows) × 4 quarters (columns) with sales figures. Add drop-downs for region (H1) and quarter (H2) and return the matching sales value.

Show solution (worked answer)

With regions in A2:A5, quarters in B1:E1 and values in B2:E5:
=INDEX(\$B\$2:\$E\$5,MATCH(\$H\$1,\$A\$2:\$A\$5,0),MATCH(\$H\$2,\$B\$1:\$E\$1,0)) or
=XLOOKUP(\$H\$1,\$A\$2:\$A\$5,XLOOKUP(\$H\$2,\$B\$1:\$E\$1,\$B\$2:\$E\$5)). Make the drop-downs with Data Validation → List, source \$A\$2:\$A\$5 and \$B\$1:\$E\$1.

INTERVIEW CORNER Questions you may be asked

Q: What are the limitations of VLOOKUP and how do you overcome them? It cannot look left, it defaults to approximate match, it breaks when columns are inserted (hard-coded index), and it returns only the first match. Use XLOOKUP or INDEX + MATCH, and FILTER for multiple matches.

Q: Why is INDEX + MATCH considered more flexible than VLOOKUP? The lookup and return ranges are separate, so it can look left, it is not affected by inserted columns, it supports two-way lookups, and it can be faster on large sheets because it only reads the columns it needs.

Q: How would you look up a value using two criteria? XLOOKUP(1,(range1=x)*(range2=y),return_range), INDEX/MATCH with the same multiplied condition, or a helper column that concatenates the two keys.

Practise more online

Free, trusted places to test yourself on this topic. Links open the original site in a new tab.