

Lookups in Excel

Brendan Clarke, NHS Education for Scotland, brendan.clarke2@nhs.scot

01/08/2024




Welcome

- this session is for 🎒🎒 intermediate Excel users
- we'll get going properly at 13.05
- you'll need Excel M365 Desktop and [this sample Excel workbook](#) to follow along, and you'll need to be comfortable working with tables, references, and formulas to get the most out of the session
- if you can't access the chat, you might need to join our Teams channel:
tinyurl.com/kindnetwork
- you can find session materials at tinyurl.com/kindtrp

The KIND network

- a social learning space for staff working with knowledge, information, and data across health, social care, and housing in Scotland
- we offer social support, free training, mentoring, community events, ...
- Teams channel / mailing list

Excel training sessions

Session	Date	Area	Level
Relative, absolute, mixed, structured, and R1C1 references in Excel	15:00-16:00 Thu 8th August 2024	Excel	 : intermediate-level
Excel first steps	09:30-10:30 Tue 3rd September 2024	Excel	 : beginner-level
References and names in Excel	09:30-10:30 Tue 10th September 2024	Excel	 : beginner-level

Session outline

- introduction to lookups
- three methods for lookups
 - INDEX() + MATCH()
 - VLOOKUP()
 - XLOOKUP()
- partial matching

Getting started

- files for today
 - `s03_exercises.xlsx` is a starting-point for the exercises today
 - `s03_exercises_final.xlsx` is the end-point for the exercises today - it's there to help if you get stuck or lost

Task

1. open the sample spreadsheet `s03_exercises.xlsx` and have a look around

Introduction to lookups

- imagine that you have a table like this:

name	age
Jalycia	63
Dorethia	32
Broward	53
Shacorey	65
Melvinia	24
Atravion	32
Hayvin	23
Eunika	41
Conwell	39
Ritha	61

- how would you find someone's age from their name?
 1. take the relevant name from somewhere
 2. look up the name in the table
 3. return the age corresponding to that name

INDEX() and MATCH()

- older way of doing lookups
 - think *timeless classic* rather than *old banger*
- simple and flexible
- great way to understand and practice the logic of lookups

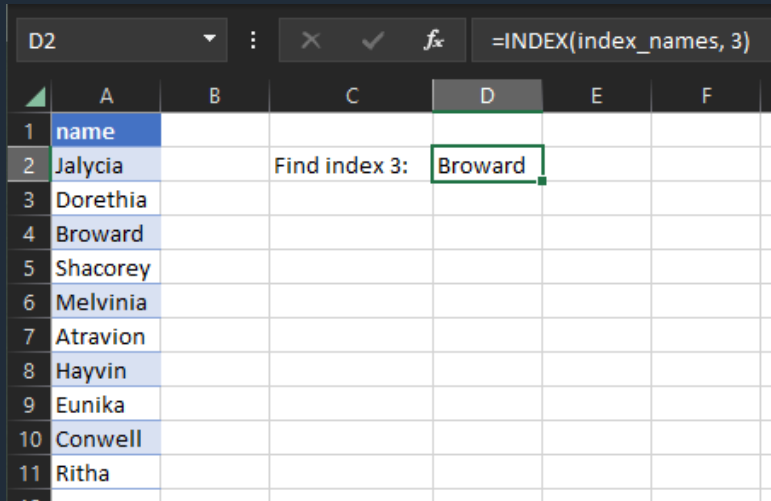
INDEX()

- **INDEX()** retrieves data from cells. You supply two arguments:
 - an array (range) of cells
 - a row number
- **INDEX(A1:A10, 3)** gives you the third row of the range A1:A10
- general syntax: **INDEX(array, row_num, [column_num])**
 - **array** is the range we want to look in
 - **row_num** and the optional **column_num** is the number of the row (\pm column) that you want

INDEX()

Task

1. Find the **INDEX** worksheet of the `s03_exercises.xlsx` workbook
2. We have a one-column table containing names in **A2:A11**
3. Use **INDEX()** to find the value in the third row



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	name					
2	Jalycia		Find index 3:	Broward		
3	Dorethia					
4	Broward					
5	Shacorey					
6	Melvinia					
7	Atravion					
8	Hayvin					
9	Eunika					
10	Conwell					
11	Ritha					

The formula bar at the top shows the formula: `=INDEX(index_names, 3)`. The cell D2 contains the result "Broward".

MATCH()

- **MATCH()** retrieves the row number of a cell. You supply two arguments:
 - a lookup value
 - an array of cells
- **MATCH(lookup_value, lookup_array, [match_type])**

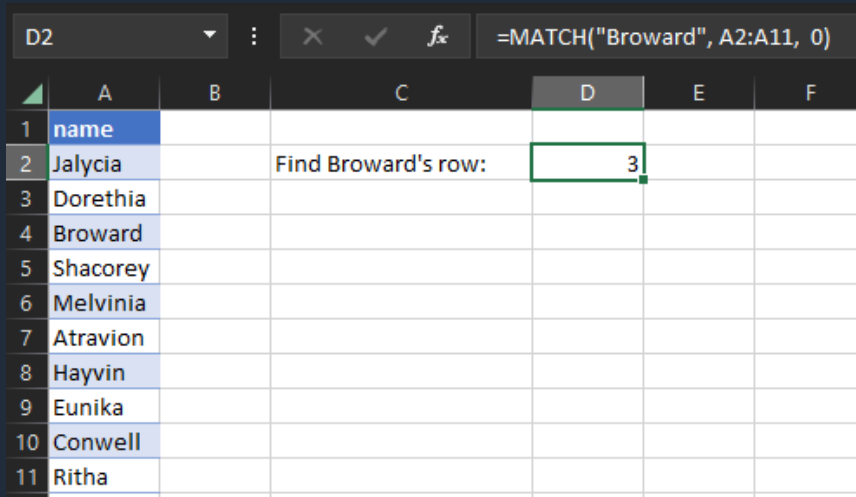
Warning

- the optional **match_type** argument is deathly important:
- **1 (or omitted)** finds the largest value that is less than or equal to the lookup
- **0** finds the first value that is exactly equal to the lookup
- **-1** finds the smallest value that is greater than or equal to the lookup

MATCH()

Task

1. Go to the **MATCH** worksheet
2. We have (the same) one-column table of names in **A2:A11**
3. Use **MATCH()** to find the row containing “Broward” - being careful to set **match_type** appropriately



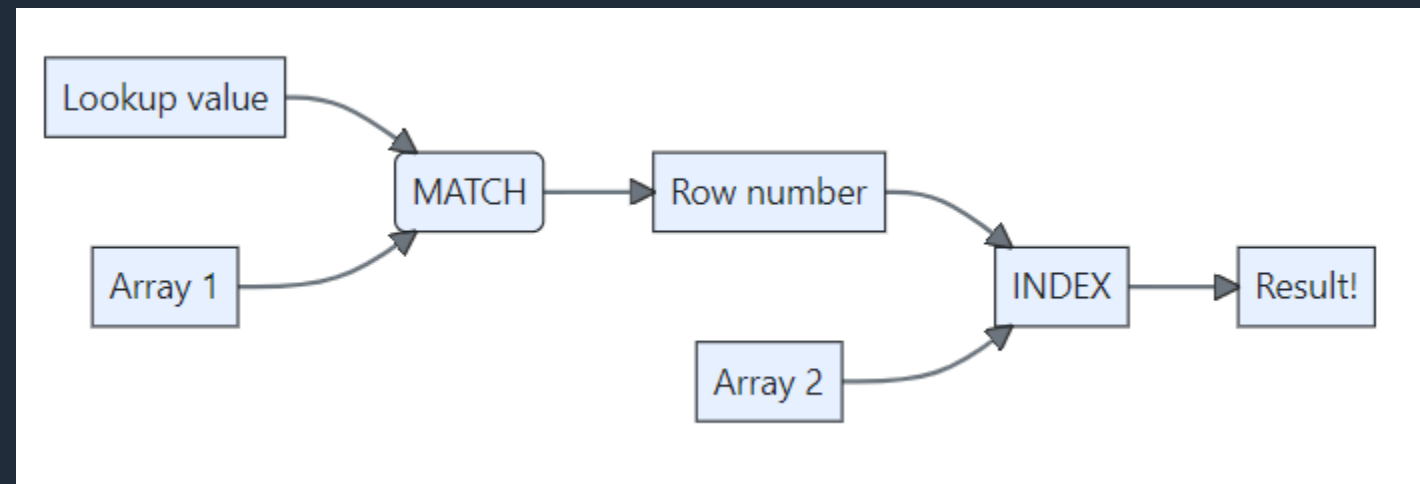
The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	name					
2	Jalycia		Find Broward's row:	3		
3	Dorethia					
4	Broward					
5	Shacorey					
6	Melvinia					
7	Atravion					
8	Hayvin					
9	Eunika					
10	Conwell					
11	Ritha					

The formula bar at the top shows the formula: `=MATCH("Broward", A2:A11, 0)`

INDEX() + MATCH()

- given that match gives us the row number, and index takes a row number and returns a value, we can link them up nicely



- first search through our data to find a **MATCH()**ing row
- then use **INDEX()** to retrieve some other column from that row

INDEX() + MATCH()

- the general syntax will be `=INDEX(range, MATCH("Name", range, 0))`

Task

1. Go to the **INDEX + MATCH (1)** worksheet
2. We have a two column table, and we want a user to be able to supply a number in **E2**, and receive the appropriate word in **E3**
3. First, write an appropriate **MATCH()** formula to find the row matching the input
4. Next, write an **INDEX()** to take that **MATCH()** value and return the appropriate word
5. Finally, try joining both formulas together in one to give an **INDEX() + MATCH()**

INDEX() + MATCH()

	A	B	C	D	E
1	digit	word			
2	1	one		Input:	6
3	2	two		MATCH()	=MATCH(\$E\$2, A2:A10)
4	3	three		INDEX()	=INDEX(B2:B10, E3)
5	4	four		INDEX() and MATCH()	=INDEX(B2:B10, MATCH(E2, A2:A10))
6	5	five			
7	6	six			
8	7	seven			
9	8	eight			
10	9	nine			

INDEX() + MATCH()

Task

1. Go to the **INDEX + MATCH (2)** worksheet. We have several formulas about Eunika. Can you fix them to give the correct result?

VLOOKUP()

- roughly INDEX() + MATCH() in one
- VLOOKUP returns along rows, allowing us to look up a value in one column, and returns a corresponding value from another column
- that sounds ideal, but beware: VLOOKUP() is more quirky than INDEX() + MATCH()

VLOOKUP()

- `VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])`
- `lookup_value` is what we want to look up
- `table_array` is the table that we're looking up in
- `col_index_num` is the number of the column that we want to return from
- `range_lookup` allows either approximate matching (the default) or exact matching (by specifying FALSE)
 - The major source of oddity. Make sure you **check the manual** before doing anything serious with `VLOOKUP()`

VLOOKUP()

Task

1. Go to the **VLOOKUP** worksheet
2. There are two requests for lookups. Can you populate them with appropriate **VLOOKUP()** formulas?

Warning

- **VLOOKUP()** generally requires the lookup value to appear in the left-most column of your range
- as the second part of the task shows, this requirement can manifest in lots of odd ways

XLOOKUP()

- semi-new (any versions after Excel 2019)
- similar to **INDEX()** + **MATCH()**. Major differences:
 - it **spills**, so ideal for returning unpredictable quantities of cells
 - provides exact matching by default
- the basic syntax is: = **XLOOKUP(lookup_value, lookup_array, return_array)**

XLOOKUP()

Task

1. Go to the **XLOOKUP (1)** worksheet
2. There are two requests for lookups. Can you populate them with appropriate XLOOKUP() formulas?

Returning several cells with XLOOKUP()

Task

1. Go to the **XLOOKUP (2)** worksheet
2. Set up an **XLOOKUP()** supplying a range of columns as the **return_array** value

Fighting with tables

- note that returning several cells into a table often causes errors
- you may need to work with a range instead if planning to use **XLOOKUP()** to return an unpredictable amount of data

Partial text matching

- Excel is not brilliant for partial text matching
- there are a number of commercial add-ins and VBA-based solutions
 - those come with risks
 - we'll concentrate on base-Excel here
- three main options:
 1. `XLOOKUP()` with `match_mode = 2` + wildcard
 2. `VLOOKUP()` with `range_lookup = FALSE` + wildcards
 3. `MATCH()` with `match_type = 0` + wildcards

XLOOKUP() with match_mode = 2 + wildcard

- XLOOKUP() allows partial matching via the “*” wildcard
- add “*” to the end of the partial string you want to match
- however, please be aware that:
 1. this will only reliably match sub-strings at the start of strings
 1. (so “ow*” will match “owl” but not “towel”)
 2. adding extra wildcards interacted unpredictably with match_mode in testing
- =XLOOKUP("*substring*", search_col, return_col, ,2)

VLOOKUP() with range_lookup = FALSE + wildcards

- as with XLOOKUP, adding "*" allows wildcard matching
- but you can do better with the wildcard sandwich: "*substring*"
 - or, if you're bringing in a reference "*" & cell_ref & "*"
- ensure you set the (confusingly named) range_lookup argument to FALSE to allow approximate matching
- VLOOKUP("*substring*", table, column, FALSE)

MATCH() with match_type = 0 + wildcards

- same wildcard sandwich strategy, with the flexibility of INDEX() + MATCH()
- my preferred solution, largely because the other options are harder to debug
- =INDEX(array, MATCH("*substring*", array, 0))

Task

1. Go to the **Partial matching** worksheet
2. Try implementing these three ways of partial matching

Feedback and resources

- please can I ask for some feedback - takes less than a minute, completely anonymous, helps people like you find the right training for them

Session	Date	Area	Level
Relative, absolute, mixed, structured, and R1C1 references in Excel	15:00-16:00 Thu 8th August 2024	Excel	📌📌 : intermediate-level
Excel first steps	09:30-10:30 Tue 3rd September 2024	Excel	📌 : beginner-level

Session

Date


Area

Level

References and
names in Excel

09:30-10:30 Tue 10th
September 2024

Excel

 :beginner-level