Skill Navora

TRACK 1

Advanced Excel

From the Excel window to dashboards: formatting, 100+ functions, lookups, pivot tables, advanced charts, Power Query, Power Pivot and AI in Excel.

12 lessons | Version 2026.10 | Updated October 2026

Created by **Varun Kumar**
www.skillnavora.in

Contents

Every entry is a clickable link. The PDF also has bookmarks for each lesson and section.

01 Getting Started with Excel for Data Analysis
02 Data Entry, Data Validation & Data Cleaning
03 Number, Custom & Conditional Formatting
04 Formulas, Cell References & Aggregate Functions
05 Text Functions, Logical Functions & IF
06 Sorting, Filtering & Dynamic Arrays
07 Lookup Functions: VLOOKUP, XLOOKUP and INDEX-MATCH
08 Pivot Tables from Basic to Advanced
09 Date, Time & Number Functions and Named Ranges
10 Charts and Advanced Charts
11 Interactive Dashboard: Superstore Case Study
12 Power Query, Power Pivot, Analysis ToolPak & What-If Analysis

ADVANCED EXCEL · LESSON 1

Getting Started with Excel for Data Analysis

Learn the Excel window, the words analysts use every day, how to lay out data so it can be analysed, the five data types Excel understands, and the shortcuts that save hours.

Live classes 1–2 · ✦ Excel for Microsoft 365, 2026 · Beginner · ⏱ 90 min

By the end of this lesson you can

- Name every part of the Excel window and explain what it does
- Use the core vocabulary (workbook, worksheet, cell, range, reference, formula, function) correctly
- Recognise structured data and fix a badly laid-out sheet
- Identify the five data types Excel stores, and spot text that only looks like a number or date
- Navigate and select data quickly with keyboard shortcuts

Why Excel still matters in 2026

Even with Python, SQL, Power BI and AI assistants, Excel is still the first tool almost every analyst opens. It is where business teams keep budgets, targets, trackers and exports from other systems. If you can clean, analyse and present data in Excel quickly, you can add value from day one in any data role.

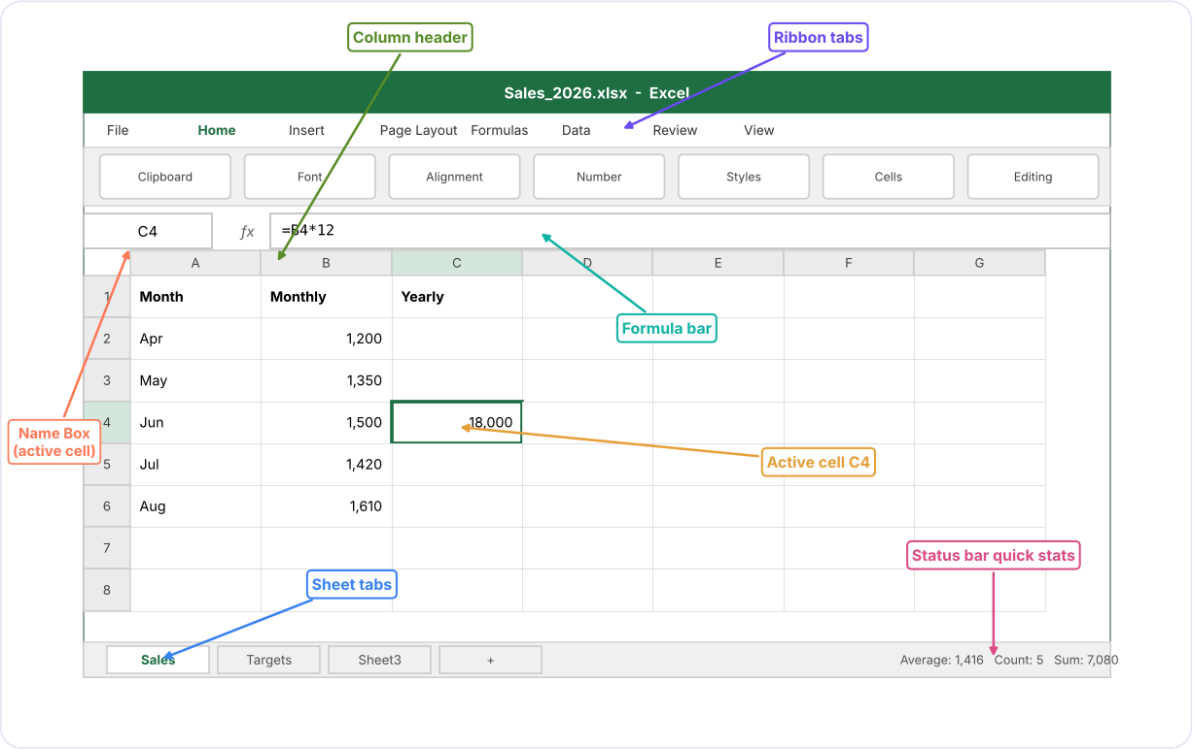
Excel is also the easiest place to *learn* data thinking. Every idea you meet here comes back later:

Idea in Excel	Same idea later in the course
A table with one header row	A SQL table, a pandas DataFrame, a Power BI table
SUMIFS, COUNTIFS	`GROUP BY` in SQL, `groupby()` in pandas, DAX measures
VLOOKUP / XLOOKUP	SQL `JOIN`, pandas `merge()`, Power BI relationships
Pivot table	SQL aggregation, `pivot_table()` in pandas, Power BI matrix
Conditional formatting	Highlighting in dashboards

LIVE CLASS **What was covered in class**

Live class 1 was the programme introduction. Live class 2 covered the basics of Excel, terminology, shortcuts, Excel properties, data types, data validation and formatting, and we started working on the first sheet. This lesson covers the first half; data validation and formatting continue in Lessons 2 and 3.

The Excel window, part by part



The main parts of the Excel window. The active cell is C4, and the formula bar shows the formula inside it.

Part	What it does	Tip
Ribbon	Commands grouped into tabs: Home, Insert, Formulas, Data, Review, View	Press `Alt` to see a key letter on every command
Name Box	Shows the address of the active cell or the name of a selected range	Type `D500` and press Enter to jump straight there
Formula bar	Shows what is really inside the cell: a value or a formula	The cell shows the *result*; the bar shows the *recipe*
Column headers	Letters A, B, C … XFD (16,384 columns)	Double-click the border between headers to auto-fit width
Row headers	Numbers 1 to 1,048,576	Click a row number to select the whole row
Sheet tabs	Each tab is one worksheet inside the workbook	Right-click a tab to rename, colour, move or copy it
Status bar	Quick sum, average and count of the selected cells	Right-click it to add Min, Max and Numerical Count

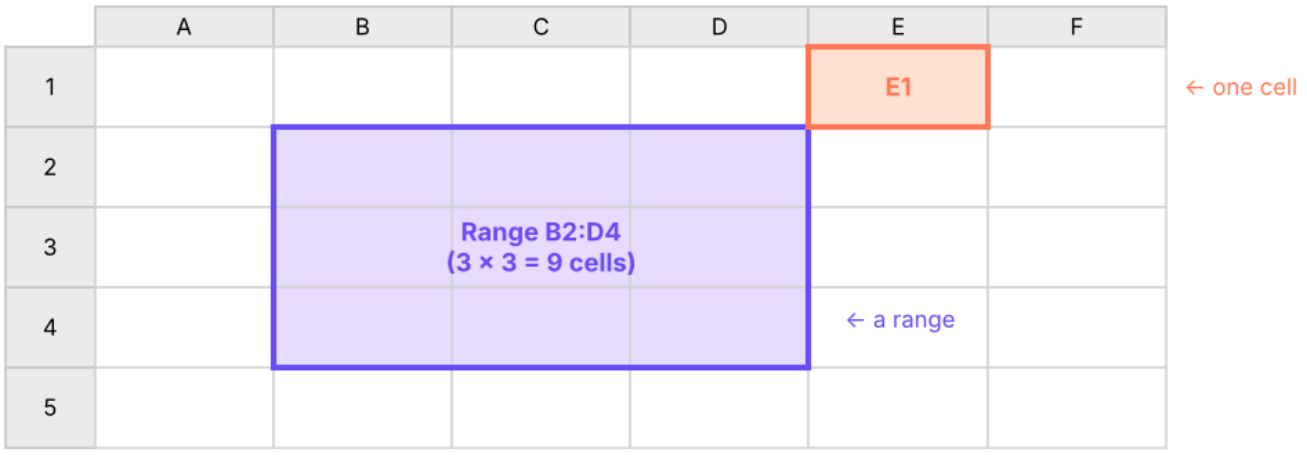
TIP **The status bar is a free calculator**

Select any numbers and look at the bottom-right corner. Excel shows the Average, Count and Sum without you writing a formula. It is the fastest way to sanity-check a total someone sends you.

The vocabulary every analyst uses

DEFINITION **Core terms**

Workbook: the file (`.xlsx`). One workbook holds many worksheets. **Worksheet (sheet)**: one grid of rows and columns inside a workbook. **Cell**: one box where a row and a column meet, for example `C4` . **Cell reference (address)**: column letter + row number, such as `C4` . Used inside formulas to point at cells. **Range**: a rectangle of cells written as `first:last` , for example `A1:C10` (3 columns × 10 rows = 30 cells). **Formula**: anything that starts with `=` and calculates a result, such as `=B4*12` . **Function**: a built-in, named formula that takes inputs (arguments), such as `=SUM(B2:B6)` . **Argument**: an input you give a function, separated by commas: `=ROUND(B2, 1)` has two arguments.



How cells, ranges and references relate. A range is always written from its top-left cell to its bottom-right cell.

Excel limits worth knowing

Item	Limit in Excel 365
Rows per sheet	1,048,576
Columns per sheet	16,384 (A to XFD)
Characters in one cell	32,767
Characters in a formula	8,192
Arguments in one function	255
Decimal precision	15 significant digits

WATCH OUT **The 15-digit trap**

Excel keeps only 15 significant digits. If you type a 16-digit card number or a long ID, the last digits turn into zeros. Store long IDs, phone numbers, PIN codes and account numbers as **text** (format the column as Text first, or start the entry with an apostrophe `'`).

Structured data: the single most important habit

Excel will do almost anything, but it can only *analyse* data that is laid out as a clean table. Analysts call this **structured** or **tidy** data.

STEP BY STEP **The six rules of a structured table**

1. **One header row** at the top, with a unique, short name for every column.
2. **One record per row**, for example one order, one employee or one day.
3. **One kind of information per column**: dates only in the date column, numbers only in the amount column.
4. **No blank rows or blank columns** inside the table.
5. **No merged cells** and no totals typed in the middle of the data.
6. **Raw values, not decorations**: write `Delhi`, not `Delhi ✔ (checked)`.

Here is a typical "report-style" sheet that looks fine to a human but is hard to analyse:

Before: a report, not a table

	A	B	C	D
1	**Sales Report – Q1**			
2	*Region: North*			
3	Name	Jan	Feb	Mar
4	Asha Rao	12,000	15,500	n/a
5	Ravi Verma	9,800		11,250
6				
7	*Region: South*			
8	Name	Jan	Feb	Mar
9	Meera Iyer	14,250	13,900	16,100
10	**TOTAL**	36,050	29,400	27,350

Problems: a title row above the headers, the region written as a heading instead of a column, a blank row, the header row repeated, numbers stored as text with commas, `n/a` mixed into a number column, and a total row inside the data.

The same information as a structured table:

After: a structured table (long format)

	A	B	C	D
1	**Region**	**Name**	**Month**	**Sales**
2	North	Asha Rao	Jan	12000
3	North	Asha Rao	Feb	15500
4	North	Ravi Verma	Jan	9800
5	North	Ravi Verma	Mar	11250
6	South	Meera Iyer	Jan	14250
7	South	Meera Iyer	Feb	13900
8	South	Meera Iyer	Mar	16100
9	**Total**			92800

```
D9   =SUM(D2:D8)
```

Now a pivot table, a SUMIFS, a chart or Power BI can use the data immediately. The total sits **below** the table, not inside it, and is calculated, not typed.

> **TIP** **Long format beats wide format**
>
> In the "before" sheet each month is a separate column (wide format). In the "after" sheet the month is a value in one `Month` column (long format). Long format is what pivot tables, SQL and Power BI expect. In Lesson 12 you will see how Power Query's **Unpivot** turns wide data into long data in one click.

The five data types Excel understands

Every cell holds exactly one type of value. Knowing the type explains most "my formula doesn't work" problems.

Type	Examples	Default alignment	Notes
Number	`42`, `3.14`, `-500`, `18%`	Right	Percentages and currency are numbers with a format
Text (string)	`Delhi`, `E101`, `'00451`	Left	Anything Excel cannot read as a number
Date / time	`27-04-2026`, `14:30`	Right	Stored as a serial number: days since 1 Jan 1900
Logical (Boolean)	`TRUE`, `FALSE`	Centre	Result of comparisons like `=B2>100`
Error	`#N/A`, `#DIV/0!`, `#VALUE!`	Centre	Tells you a formula has a problem

Excel's own functions can tell you the type of a value. Each formula in column B below checks the value in column A:

Checking data types with functions

	A	B	C	D
1	Value	ISNUMBER	ISTEXT	TYPE (1=number, 2=text, 4=logical)
2	42	TRUE	FALSE	1
3	Delhi	FALSE	TRUE	2
4	00451	FALSE	TRUE	2
5	27-04-2026	TRUE	FALSE	1
6	TRUE	FALSE	FALSE	4
7	1500	FALSE	TRUE	2

```
B2   =ISNUMBER(A2)
C2   =ISTEXT(A2)
D2   =TYPE(A2)
```

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

Notice two important results:

- The **date** in row 5 is a number. Excel stores 27-04-2026 as the serial number 46139, which is why you can subtract one date from another to get days.
- `'1500` in row 7 is **text**, even though it looks like a number. `SUM` ignores it. Imported data often has "numbers stored as text"; Excel flags them with a small green triangle.

Dates are numbers underneath

	A	B	C
1	Date	Same cell shown as a number	Days until 31-Dec-2026
2	27-04-2026	46139	248
3	01-01-2026	46023	364

```
B2   =A2
C2   =DATE(2026,12,31)-A2
B3   =A3
C3   =DATE(2026,12,31)-A3
```

COMMON MISTAKES **Fixing numbers stored as text**

- Select the cells, click the yellow warning icon, choose **Convert to Number**.
- Or use **Data → Text to Columns → Finish** on the column.
- Or in a formula: `=VALUE(A7)` or `=A7*1`.
- Remove hidden spaces first with `=TRIM(A7)` (and `=CLEAN()` for line breaks from web exports).

Error values and what they mean

Error	Meaning	Typical cause
`#DIV/0!`	Division by zero	`=A2/B2` where B2 is empty or 0
`#N/A`	Value not available	A lookup could not find the value
`#VALUE!`	Wrong type of argument	Doing maths with text, such as `="abc"*2`
`#REF!`	Invalid reference	You deleted a row or column a formula used
`#NAME?`	Excel does not recognise a name	A typo in a function name, or text without quotes
`#NUM!`	Invalid number	`=SQRT(-1)`
`#SPILL!`	A dynamic array has no room	Cells below the formula are not empty (Lesson 6)
`#####`	Not an error	The column is too narrow to show the value

Excel properties: what makes a workbook "smart"

These behaviours make Excel more than a calculator:

1. **Automatic recalculation**: change an input and every dependent formula updates instantly (Formulas → Calculation Options can switch this to Manual for huge files; press F9 to recalculate).
2. **Relative references**: copy `=B2*12` down and it becomes `=B3*12`, `=B4*12` automatically (Lesson 4 covers `$` to lock references).
3. **AutoFill**: drag the small square (fill handle) at the bottom-right of a selection to continue patterns: `Jan` → `Feb` → `Mar`, `1, 2, 3`, `Mon, Tue`, dates and formulas.
4. **Flash Fill** (`Ctrl + E`): type one or two examples of a pattern (such as first names taken from full names) and Excel fills the rest.
5. **Excel Tables** (`Ctrl + T`): turn a range into a named table that expands automatically, keeps formatting, and allows readable formulas like `=SUM(Sales[Amount])`.
6. **File properties**: File → Info shows the author, size, last-modified date and lets you inspect the workbook for hidden data before sharing.

2026 UPDATE **What's new in Excel (2025–2026)**

- **Copilot in Excel** can explain formulas, suggest formula columns, highlight insights and create charts from a plain-English request (Microsoft 365 Copilot licence required).
- **Python in Excel** lets you type `=PY(` and run pandas code inside a cell; the code runs in Microsoft's cloud.
- **Dynamic array functions** such as `UNIQUE`, `SORT`, `FILTER`, `XLOOKUP`, `TEXTSPLIT`, `GROUPBY` and `PIVOTBY` are now standard in Microsoft 365.
- **Checkbox cells** (Insert → Checkbox) store TRUE/FALSE and are useful for trackers. You will use each of these later in the Excel track.

Keyboard shortcuts that save hours

You do not need to memorise 100 shortcuts. Learn these 25 and you will already be faster than most office users. (Mac users: replace `Ctrl` with `Cmd` for most of them.)

Task	Shortcut
Jump to the edge of the data	`Ctrl + Arrow key`
Select to the edge of the data	`Ctrl + Shift + Arrow key`
Select the whole current table	`Ctrl + A` (inside the data)
Go to cell A1 / last used cell	`Ctrl + Home` / `Ctrl + End`
Select a whole column / row	`Ctrl + Space` / `Shift + Space`
Insert / delete rows or columns	`Ctrl + Shift + +` / `Ctrl + -`
Edit the active cell	`F2`
Lock a reference ($A$1 → A$1 → $A1)	`F4` while editing a formula
Fill down / fill right	`Ctrl + D` / `Ctrl + R`
Enter the same value in all selected cells	type, then `Ctrl + Enter`
AutoSum	`Alt + =`
Today's date / current time	`Ctrl + ;` / `Ctrl + Shift + ;`
Create an Excel Table	`Ctrl + T`
Toggle filters	`Ctrl + Shift + L`
Format Cells dialog	`Ctrl + 1`
Show formulas instead of results	`Ctrl + ` `
Flash Fill	`Ctrl + E`
Paste Special	`Ctrl + Alt + V`
Find / Replace	`Ctrl + F` / `Ctrl + H`
New sheet	`Shift + F11`
Move between sheets	`Ctrl + Page Up / Page Down`
Hide rows / columns	`Ctrl + 9` / `Ctrl + 0`
Undo / Redo	`Ctrl + Z` / `Ctrl + Y`
Repeat last action	`F4` (outside a formula)
Show key tips on the ribbon	`Alt`

TIP **Practise with real data**

Open `employees_small.csv` (download it from the top of this lesson). Click any cell inside the data, press `Ctrl + A`, then `Ctrl + T` to make it a Table. Press `Ctrl + Down` to jump to the last row and `Ctrl + Home` to come back. Select the salary column and read the Sum and Average in the status bar.

Your first worksheet, step by step

Let's build a small salary sheet with the 20-row employee file. The first rows look like this:

Python

```
import pandas as pd
emp = pd.read_csv("employees_small.csv")
emp.head(6)
```

OUTPUT

	emp_id	name	department	city	join_date	salary	rating
0	E101	Kavya Iyer	HR	Mumbai	2016-02-28	97000	4
1	E102	Imran Reddy	Finance	Delhi	2019-12-14	157000	3
2	E103	Harpreet Kulkarni	Finance	Mumbai	2024-10-05	131000	4
3	E104	Diya Iyer	HR	Bengaluru	2024-01-02	90000	4
4	E105	Simran Iyer	HR	Bengaluru	2017-08-03	139000	4
5	E106	Saanvi Rao	HR	Delhi	2025-02-18	83000	4

6 rows × 7 columns

STEP BY STEP **Build it**

1. **Open** the CSV in Excel (File → Open, or Data → From Text/CSV).
2. Press `Ctrl + T` to convert it to a Table, and name it `Emp` (Table Design → Table Name).
3. In a new column `annual_salary`, type = then click the salary cell and type *12. Because it is a Table, Excel fills the whole column with `=[@salary]*12`.
4. Below the data, add summary formulas: total salary, average salary, number of employees.
5. Format the salary columns as `#,##0` with `Ctrl + 1` (Lesson 3 explains every format).
6. Freeze the header row: View → Freeze Panes → Freeze Top Row.

First worksheet: summary formulas (data on sheet Emp)

	A	B	C
1	Measure	Formula result	What the formula does
2	Employees	20	Counts non-empty IDs
3	Total monthly salary	1,978,000	Adds the salary column
4	Average monthly salary	98,900	Total ÷ count
5	Highest salary	157,000	Largest value
6	Lowest salary	45,000	Smallest value

```
B2   =COUNTA(Emp!A2:A21)
B3   =SUM(Emp!F2:F21)
B4   =AVERAGE(Emp!F2:F21)
```

```
B5   =MAX(Emp!F2:F21)
B6   =MIN(Emp!F2:F21)
```

The results above were calculated from the real file. Check them yourself in Python:

Python

```
s = emp["salary"]
print("Employees:", s.count())
print("Total:", f"{s.sum():,}", "| Average:", f"{s.mean():,.0f}", "| Max:", f"{s.max():,}", "| Min:",
f"{s.min():,}")
```

OUTPUT

```
Employees: 20
Total: 1,978,000 | Average: 98,900 | Max: 157,000 | Min: 45,000
```

Saving and sharing the right way

Format	Use it when
`.xlsx`	Normal workbooks (default)
`.xlsm`	Workbooks that contain macros (VBA)
`.xlsb`	Very large workbooks; binary format opens and saves faster
`.csv`	Plain data to move between systems (Python, SQL, Power BI). Keeps only one sheet and no formatting or formulas
`.pdf`	Read-only reports for people who should not edit

WATCH OUT **CSV and leading zeros**

When you open a CSV directly, Excel guesses types and may drop leading zeros (`00451` → `451`) or turn codes like `1-2` into dates. Use **Data → From Text/CSV** and set the column type to Text in the preview to keep values exactly as they are.

Summary

KEY TAKEAWAYS

- The **cell** shows the result; the **formula bar** shows what is really inside.
- A **range** is written top-left to bottom-right, for example `B2:D4`.
- Analyse only **structured data**: one header row, one record per row, one type per column, no blanks, no merged cells, no totals inside.
- Excel has five value types: **number, text, date/time (a number underneath), logical, error**.
- Long IDs and codes must be stored as **text** to avoid the 15-digit and leading-zero traps.
- Learn the navigation shortcuts first: `Ctrl + Arrow`, `Ctrl + Shift + Arrow`, `Ctrl + T`, `Ctrl + 1`, `F2`, `F4`.

Quick quiz

6 questions · tap an answer to check it

1. A cell shows `18,000`, but the formula bar shows `=B4*12`. What is inside the cell?

A The number 18,000

B A formula whose result is 18,000

C Text that looks like a number

D An error

Answer: B. The cell displays the result; the formula bar shows what is stored. This cell stores a formula.

2. How many cells are in the range `B2:D5`?

A 8

B 9

C 12

D 15

Answer: C. Columns B, C, D = 3 columns; rows 2 to 5 = 4 rows; 3 × 4 = 12 cells.

3. Which of these breaks the rules of structured data?

- **A** One header row with unique names
- **B** A merged "Region" heading above each group of rows
- **C** One record per row
- **D** A date column that contains only dates

Answer: B. Region should be its own column, with the value repeated on every row. Merged headings make the data impossible to sort, filter or pivot.

4. Why does `SUM` ignore some numbers imported from a CSV file?

- **A** SUM only works on Excel Tables
- **B** The values are stored as text
- **C** The numbers have too many decimals
- **D** The column is too narrow

Answer: B. Numbers stored as text are skipped by SUM. Convert them with Text to Columns, VALUE() or Convert to Number.

5. What does Excel store when you type the date 27-04-2026?

- **A** The text "27-04-2026"
- **B** A serial number counting days from 1 January 1900
- **C** Three separate numbers for day, month and year
- **D** A logical value

Answer: B. Dates are numbers formatted to look like dates. That is why date arithmetic works.

6. Which shortcut turns a range into an Excel Table?

- **A** Ctrl + E
- **B** Ctrl + 1
- **C** Ctrl + T
- **D** Ctrl + Shift + L

Answer: C. Ctrl + T creates a Table. Ctrl + E is Flash Fill, Ctrl + 1 opens Format Cells, Ctrl + Shift + L toggles filters.

EXERCISE **Clean a messy sheet**

Look at the "Before" report in this lesson. Write down, in order, the steps you would take in Excel to turn it into the "After" table.

Show solution (worked answer)

1. Delete the title row and the blank row.
2. Insert a new first column called `Region` and fill it with North or South for each person; delete the "Region:" heading rows and the repeated header row.
3. Delete the TOTAL row (totals belong below or in a pivot table).
4. Replace `n/a` with an empty cell (Ctrl + H, replace `n/a` with nothing).
5. Convert the Jan–Mar columns to real numbers (Text to Columns or Convert to Number).
6. Unpivot the month columns into `Month` and `Sales` (Power Query → Unpivot Other Columns, Lesson 12) so each row is one person-month.
7. Press Ctrl + T to make it a Table and add the total with `=SUM()` below it.

EXERCISE **Spot the data type**

For each entry, say which data type Excel stores: `₹1,250`, `'2026`, `15:45`, `=5>3`, `#N/A`, `Pune`.

Show solution (worked answer)

`₹1,250` is a number (currency format). `'2026` is text because of the apostrophe. `15:45` is a time, which is a number (0.65625 of a day). `=5>3` returns the logical value TRUE. `#N/A` is an error. `Pune` is text.

INTERVIEW CORNER

Questions you may be asked

Q: What is the difference between a formula and a function? A formula is any expression that starts with = (for example `=A2*B2`). A function is a named, built-in calculation used inside formulas (for example `SUM`, `VLOOKUP`). Every function is used in a formula, but not every formula uses a function.

Q: How does Excel store dates, and why does it matter? As serial numbers (days since 1-Jan-1900; times are fractions of a day). It means you can subtract dates to get durations, add days to a date, and sort dates correctly. If a date is stored as text, none of that works.

Q: What makes data "analysis-ready"? One header row, one record per row, one data type per column, no blank rows or merged cells, no subtotals mixed into the data, and consistent spelling of categories.

ADVANCED EXCEL · LESSON 2

Data Entry, Data Validation & Data Cleaning

Stop bad data at the door with data validation and drop-down lists, then clean what already exists with Flash Fill, Text to Columns, Find & Replace, Go To Special, Remove Duplicates, Paste Special and Excel Tables.

Live class 2 · ✦ Excel for Microsoft 365, 2026 · Beginner · ⏱ 90 min

By the end of this lesson you can

- Enter data quickly and consistently with AutoFill, Flash Fill and custom lists
- Build drop-down lists, number and date limits and custom-formula rules with Data Validation
- Create dependent (cascading) drop-downs
- Clean messy text, split and combine columns, and remove duplicates safely
- Use Paste Special and Excel Tables as everyday cleaning tools

Garbage in, garbage out

Most time in real analytics is not spent on fancy charts; it is spent fixing data: extra spaces, inconsistent spellings (`Bangalore`, `Bengaluru`, `bengaluru`), numbers stored as text, duplicates and blanks. Surveys of data teams regularly report that cleaning and preparing data takes well over half of an analyst's time.

There are two ways to win this fight:

Prevent problems at entry time, and clean whatever slips through. Both are needed.

Fast, consistent data entry

AutoFill and the fill handle

Select a cell and drag the small square at its bottom-right corner (the **fill handle**). Excel continues the pattern it recognises.

You type	Drag gives you
`Jan`	Feb, Mar, Apr ...
`Monday`	Tuesday, Wednesday ...
`1` , `2` (select both)	3, 4, 5 ...
`Q1`	Q2, Q3, Q4, Q1 ...
`01-04-2026`	02-04-2026, 03-04-2026 ... (right-drag for months or years)
`Item 1`	Item 2, Item 3 ...
`=B2*12`	`=B3*12` , `=B4*12` ...

Double-click the fill handle to fill a formula down as far as the column next to it has data. This is the fastest way to copy a formula to 10,000 rows.

> **TIP** **Custom lists**
>
> File → Options → Advanced → **Edit Custom Lists** lets you add your own sequences, for example `North, South, East, West` or your company's department order. AutoFill and custom sort then follow that order.

Flash Fill (Ctrl + E)

Flash Fill learns a pattern from one or two examples you type. It is perfect for splitting names, extracting codes or reformatting text.

Flash Fill: type the first answer in each column, then press Ctrl + E

	A	B	C	D	E
1	**Full name**	**Email**	**First name**	**Last name**	**Username**
2	Kavya Iyer	kavya.iyer@abc.in	Kavya	Iyer	kiyer
3	Imran Reddy	imran.reddy@abc.in	Imran	Reddy	ireddy
4	Harpreet Kulkarni	harpreet.k@abc.in	Harpreet	Kulkarni	hkulkarni
5	Diya Iyer	diya.iyer@abc.in	Diya	Iyer	diyer

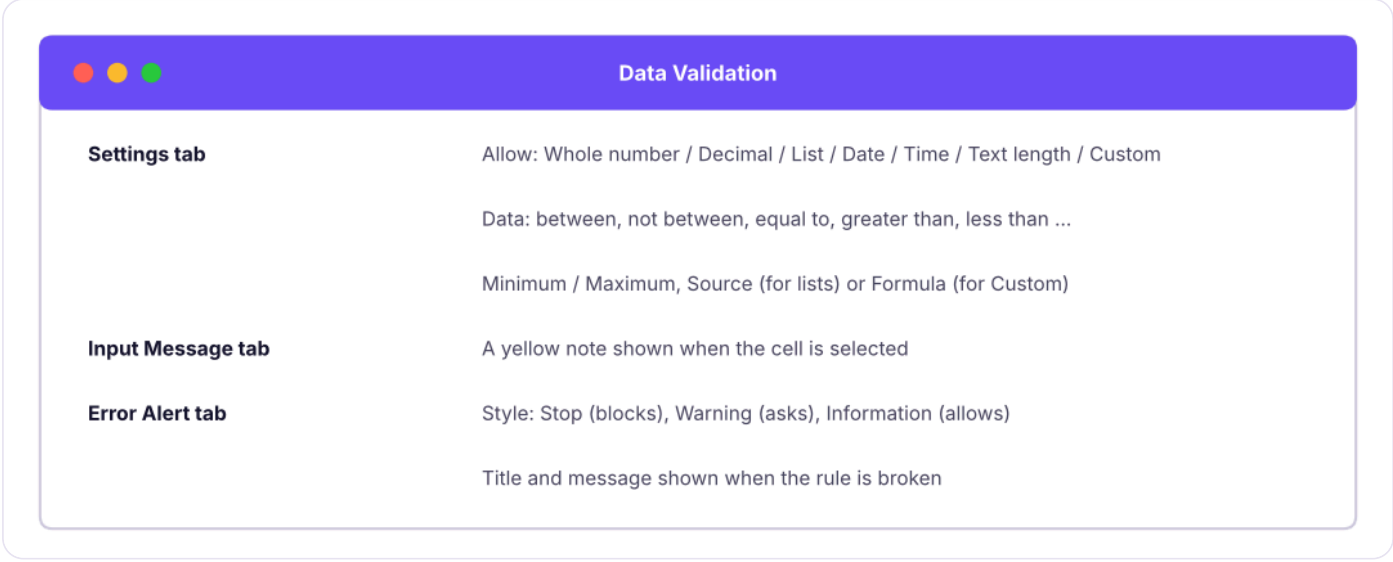
Purple cells are typed by you; green cells are filled by Flash Fill.

WATCH OUT **Flash Fill is not a formula**

Flash Fill writes **static values**. If the source changes, the results do not update. Use formulas (Lesson 5) or Power Query (Lesson 12) when the data will be refreshed.

Data Validation: rules for what can be typed

Data → Data Validation (shortcut `Alt, A, V, V`) attaches a rule to cells. Anyone who types something that breaks the rule is stopped or warned.



The three tabs of the Data Validation dialog.

The validation types you will use most

Allow	Example rule	Business use
Whole number	between 18 and 65	Age of an employee
Decimal	between 0 and 0.5	Discount rate
List	`Sales,Finance,IT,HR,Operations`	Department drop-down
Date	between `01-04-2026` and `31-03-2027`	Entries inside the financial year
Time	between 09:00 and 18:00	Office hours
Text length	equal to 6	PIN code, employee code
Custom	`=COUNTIF($A$2:$A$500,A2)=1`	No duplicate IDs

Building a drop-down list, step by step

STEP BY STEP **Department drop-down**

1. On a sheet called `Lists`, type the allowed values in a column (`Sales`, `Finance`, `IT`, `HR`, `Operations`) and convert it to a Table (`Ctrl + T`) named `tblDept`. A Table grows automatically when you add a department.
2. Select the cells where people will enter a department (for example `C2:C500`).
3. Data → Data Validation → Allow: **List**.
4. Source: `=INDIRECT("tblDept[Department]")` or simply select the range `=Lists!$A$2:$A$6`.
5. Keep **In-cell dropdown** ticked. Add an Input Message ("Choose a department") and an Error Alert (Stop, "Please pick a department from the list").

Data entry sheet with validation rules (rules shown in row 2)

	A	B	C	D	E	F	G
1	**Emp ID**	**Name**	**Department**	**Age**	**Join date**	**Discount**	**PIN code**
2	*Text length = 4*	*Any text*	*List: tblDept*	*Whole 18–65*	*Date ≥ 01-01-2000*	*Decimal 0–0.5*	*Text length = 6*
3	E101	Kavya Iyer	HR	29	28-02-2016	0.1	400001
4	E102	Imran Reddy	Fin	17	14-12-2019	0.75	11002

Row 3 passes every rule (green). In row 4, `Fin` is not in the list, 17 is under the minimum age, 0.75 is above the maximum discount and `11002` has only 5 characters, so Excel would block each of these entries (red).

Custom formula rules

The **Custom** option accepts any formula that returns TRUE (allow) or FALSE (block). Write the formula for the **first** selected cell; Excel adjusts it for the others.

Rule	Custom formula (cells A2:A500 selected)
No duplicates	`=COUNTIF($A$2:$A$500,A2)=1`
Must start with "E" and be 4 characters	`=AND(LEFT(A2,1)="E",LEN(A2)=4)`
Numbers only (for a text-formatted column)	`=ISNUMBER(--A2)`
No spaces	`=ISERROR(FIND(" ",A2))`
Weekdays only (dates)	`=WEEKDAY(A2,2)<6`
Total of a budget column cannot exceed 1,00,000	`=SUM($B$2:$B$50)⩽100000`
Email must contain "@" and "."	`=AND(ISNUMBER(FIND("@",A2)),ISNUMBER(FIND(".",A2)))`

Dependent (cascading) drop-downs

A second list that changes with the first: choose a **Category**, and the **Sub-category** list shows only matching items.

Lists sheet: each column is a named range (Formulas → Create from Selection → Top row)

	A	B	C
1	**Furniture**	**Office_Supplies**	**Technology**
2	Chairs	Binders	Phones
3	Tables	Paper	Accessories
4	Bookcases	Storage	Machines
5	Furnishings	Art	Copiers
6		Appliances	

STEP BY STEP **Make it work**

1. Select the lists (with headers) → Formulas → **Create from Selection** → Top row. Excel creates the names `Furniture`, `Office_Supplies` and `Technology`.
2. Category cell `A2`: Data Validation → List → Source: `Furniture,Office_Supplies,Technology`.
3. Sub-category cell `B2`: Data Validation → List → Source: `=INDIRECT(A2)`.
4. `INDIRECT` turns the text in A2 into a reference to the named range with that name.

2026 UPDATE **Dynamic-array version (Microsoft 365)**

With a two-column mapping table (`Category`, `SubCategory`), put `=FILTER(Map[SubCategory],Map[Category]=A2)` in a helper cell and point the validation source to it with the spill operator, for example `=$H$2#`. No named ranges are needed, and new items appear automatically.

Finding data that already breaks the rules

Validation only checks **new** typing. Data pasted in, or entered before the rule existed, is not checked. Use **Data → Data Validation → Circle Invalid Data** to draw red circles around existing violations.

COMMON MISTAKES **Validation is not security**

Pasting over a validated cell replaces the rule. Use Paste Special → Values when pasting into validated areas, and protect the sheet (Lesson 11) if people must not change rules.

Cleaning text with formulas

These functions fix most text problems. Each formula below is written for row 2 and copied down.

Common text-cleaning formulas

	A	B	C	D	E
1	Raw value	=TRIM(A2)	=PROPER(TRIM(A2))	=SUBSTITUTE(A2,"-","")	=UPPER(A2)
2	bengaluru	bengaluru	Bengaluru	bengaluru	BENGALURU
3	NEW DELHI	NEW DELHI	New Delhi	NEW DELHI	NEW DELHI
4	pune-411001	pune-411001	Pune-411001	pune411001	PUNE-411001
5	mUMBAI	mUMBAI	Mumbai	mUMBAI	MUMBAI

```
B2   =TRIM(A2)
C2   =PROPER(TRIM(A2))
D2   =SUBSTITUTE(A2,"-","")
E2   =UPPER(A2)
```

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

- `TRIM` removes spaces at the start and end and squeezes repeated inner spaces to one.
- `PROPER` capitalises the first letter of each word; `UPPER` and `LOWER` change the case of everything.
- `SUBSTITUTE(text, old, new)` replaces every occurrence of `old`.
- `CLEAN` removes non-printing characters such as line breaks from copied web data.
- The non-breaking space (character 160) from websites is not removed by TRIM. Use `=TRIM(SUBSTITUTE(A2,CHAR(160)," "))`.

Text to Columns

Data → Text to Columns splits one column into several, either by a **delimiter** (comma, space, `-`, `|`) or by **fixed width**.

Before: one column with a delimiter

	A	B	C
1	Order info		
2	IN-2025-00042	Pune	Technology
3	IN-2025-00043	Delhi	Furniture

STEP BY STEP **Split by a delimiter**

1. Select the column. Data → Text to Columns.
2. Choose **Delimited** → Next → tick **Other** and type `|` (or comma, tab, space).
3. In step 3, set each column's type: keep IDs and codes as **Text** to protect leading zeros; set dates to **Date: DMY**.
4. Choose the destination cell (so the original is not overwritten) → Finish.

Text to Columns is also a quick fix for numbers stored as text: select the column, Text to Columns, and click **Finish** immediately. Excel re-reads every value as a number.

2026 UPDATE **TEXTSPLIT, TEXTBEFORE, TEXTAFTER (Microsoft 365)**

Formula versions that update automatically: `=TEXTSPLIT(A2,"|")` spills the parts across columns; `=TEXTBEFORE(A2,"-")` and `=TEXTAFTER(A2,"-",-1)` return the text before the first or after the last hyphen. See Lesson 5.

Find & Replace and wildcards

`Ctrl + H` replaces text in the selection (or the whole sheet). Click **Options** to match case, match the entire cell, or search in formulas versus values.

Wildcard	Meaning	Example
`*`	Any number of characters	`Bang*` finds Bangalore, Bangaluru
`?`	Exactly one character	`E10?` finds E101 to E109
`~`	Treat the next character literally	`~*` finds a real asterisk

Example: replace `Bang*` with `Bengaluru` (with **Match entire cell contents** ticked) to standardise every spelling in one step.

Go To Special: select blanks, constants, formulas, errors

`F5 → Special` (or `Ctrl + G → Special`) selects cells by type. The most useful trick in data cleaning is **filling blanks with the value above**:

Before: category written only once per block

	A	B	C
1	**Category**	**Product**	**Sales**
2	Furniture	Chair A	12000
3		Table B	18500
4		Bookcase C	9400
5	Technology	Phone X	25200
6		Router Y	4300

STEP BY STEP **Fill blanks with the value above**

1. Select `A2:A6`. Press `F5 → Special → Blanks → OK`. Only the empty cells are selected (yellow).
2. Type = and press the **Up arrow** (the formula becomes =A2).
3. Press `Ctrl + Enter` to put the formula in every selected blank at once.
4. Select the column, copy, and Paste Special → **Values** to freeze the result.

Other Go To Special options: **Constants** (typed values only, great for clearing inputs but keeping formulas), **Formulas** (to check or protect them), **Errors**, **Visible cells only** (copy a filtered list without hidden rows; shortcut `Alt + ;`), **Row differences**.

Removing duplicates

Data → Remove Duplicates deletes rows that repeat in the columns you choose and keeps the **first** occurrence.

Duplicate check before removing

	A	B	C	D
1	Emp ID	Name	City	Is duplicate?
2	E101	Kavya Iyer	Mumbai	First
3	E102	Imran Reddy	Delhi	First
4	E101	Kavya Iyer	Mumbai	Duplicate
5	E103	Harpreet Kulkarni	Mumbai	First
6	E102	Imran Reddy	Delhi	Duplicate

```
D2   =IF(COUNTIF($A$2:A2,A2)>1,"Duplicate","First")
D3   =IF(COUNTIF($A$2:A3,A3)>1,"Duplicate","First")
D4   =IF(COUNTIF($A$2:A4,A4)>1,"Duplicate","First")
D5   =IF(COUNTIF($A$2:A5,A5)>1,"Duplicate","First")
D6   =IF(COUNTIF($A$2:A6,A6)>1,"Duplicate","First")
```

The expanding range `$A$2:A2` counts how many times the ID has appeared **so far**, so the first appearance says "First" and later ones say "Duplicate". This lets you review duplicates before deleting anything.

> **WATCH OUT** **Always work on a copy**
>
> Remove Duplicates permanently deletes rows. Copy the sheet first, or use the formula above, conditional formatting (Home → Conditional Formatting → Highlight Duplicate Values) or `=UNIQUE()` (Lesson 6) to inspect duplicates without losing data.

Paste Special: the cleaning Swiss-army knife

`Ctrl + Alt + V` opens Paste Special.

Option	What it does	Typical use
Values (V)	Pastes results, not formulas	Freeze formula results; break links to other files
Formats (T)	Copies only formatting	Apply a header style to another table
Transpose (E)	Rows become columns	Turn a wide header row into a list
Multiply / Divide / Add / Subtract	Combines the copied number with each target cell	Convert rupees to lakhs: copy `100000`, select the amounts, Paste Special → Divide
Skip blanks	Empty copied cells do not overwrite targets	Merge two partially filled columns
Paste Link	Creates `=Sheet1!A1` references	Live link to a source cell

Excel Tables: structure that maintains itself

Convert any clean range with `Ctrl + T`. Tables are the best habit you can build in Excel:

- **Auto-expand**: new rows and columns join the table, so formulas, charts, pivot tables and validation lists include them automatically.
- **Structured references**: formulas read like English, for example `=SUM(tblSales[Amount])` or `=[@Price]*[@Qty]`.
- **Calculated columns**: type a formula once and it fills the whole column.
- **Total Row**: Table Design → Total Row adds a row with drop-down summaries that respect filters (they use `SUBTOTAL`).
- **Banded rows, filter buttons and a frozen header** come for free.

An Excel Table named tblOrders with a calculated column and Total Row

	A	B	C	D	E
1	OrderID	Product	Price	Qty	Amount
2	1001	Chair	3500	4	14,000
3	1002	Desk	12500	1	12,500
4	1003	Lamp	1200	6	7,200
5	1004	Shelf	5400	2	10,800
6	Total			13	44,500

```
E2   =C2*D2
E3   =C3*D3
E4   =C4*D4
E5   =C5*D5
D6   =SUBTOTAL(109,D2:D5)
E6   =SUBTOTAL(109,E2:E5)
```

In a real Table, the Amount formula appears as `=[@Price]*[@Qty]`. The Total Row uses `SUBTOTAL(109, …)`, which sums only visible rows, so totals change when you filter.

DO IT WITH AI **Do it with AI**

Copilot in Excel or any chat assistant can draft cleaning steps. A good prompt: *"I have a column of Indian city names with inconsistent spelling, extra spaces and mixed case (examples: ' bengaluru', 'BANGALORE', 'Bengaluru '). Give me an Excel formula to standardise them to proper case, and a list of Find & Replace steps to map old names to new ones."* Always test the answer on a copy of your data.

Summary

KEY TAKEAWAYS

- **Prevent** bad data with Data Validation (lists, limits, custom formulas) and **cure** the rest with cleaning tools.
- Validation lists should point to a Table so they grow automatically; use `INDIRECT` or `FILTER` for dependent drop-downs.
- Validation does not check pasted or old data: use **Circle Invalid Data**.
- `TRIM`, `CLEAN`, `PROPER`, `SUBSTITUTE` fix most text; Text to Columns and Flash Fill split data.
- **Go To Special → Blanks** then `=↑` and `Ctrl + Enter` fills blank cells with the value above.
- Inspect duplicates with `COUNTIF` before using Remove Duplicates; always keep a copy.
- Turn every dataset into an **Excel Table** (`Ctrl + T`).

Quick quiz

6 questions · tap an answer to check it

1. Which validation setting stops users entering a duplicate employee ID in A2:A500?

A List with source A2:A500

B Custom: `=COUNTIF($A$2:$A$500,A2)=1`

C Text length equal to 4

D Whole number between 1 and 500

Answer: B. COUNTIF counts how many times the value appears; allowing only a count of 1 blocks duplicates.

2. You pasted 2,000 rows into a validated column. How do you find values that break the rule?

A Re-create the validation rule

B Data Validation → Circle Invalid Data

C Press F9

D Use Remove Duplicates

Answer: B. Validation only checks typing. Circle Invalid Data marks existing violations.

3. What does `=TRIM(" New Delhi ")` return?

A "New Delhi"

B "New Delhi"

C "NewDelhi"

D " New Delhi "

Answer: B. TRIM removes leading and trailing spaces and reduces repeated inner spaces to one.

4. Which feature fills blank cells in a column with the value from the cell above in a few keystrokes?

A Flash Fill

B Remove Duplicates

C Go To Special → Blanks, then = + Up arrow + Ctrl + Enter

D Text to Columns

Answer: C. Go To Special selects only the blanks; Ctrl + Enter writes the relative formula into all of them.

5. What is the main downside of Flash Fill?

A It only works on numbers

B Results are static values that do not update when the source changes

C It needs a macro

D It cannot split text

Answer: B. Use formulas or Power Query for refreshable cleaning.

6. Why should a drop-down list's source be an Excel Table?

- **A** The list grows automatically when new items are added
- **B** Tables are required for validation
- **C** It makes the drop-down faster
- **D** It prevents duplicates

Answer: A. Tables expand automatically, so new departments appear in the drop-down without editing the rule.

EXERCISE **Validation set-up**

Using `employees_small.csv`, add validation so that: department must be one of Sales, Finance, IT, HR, Operations; rating must be a whole number from 1 to 5; join date cannot be in the future; and employee IDs are unique.

Show solution (worked answer)

- Department (C2:C21): List, source `Sales,Finance,IT,HR,Operations`.
- Rating (G2:G21): Whole number, between 1 and 5.
- Join date (E2:E21): Date, less than or equal to `=TODAY()`.
- Emp ID (A2:A21): Custom, `=COUNTIF($A$2:$A$21,A2)=1`. Add an Error Alert (Stop) for each with a helpful message, then run Circle Invalid Data to check the existing rows.

EXERCISE **Clean the city column**

A column contains: `mumbai`, `MUMBAI`, `Bombay`, `Mumbai`, `mumbai city`. Produce one clean value, `Mumbai`, for all five.

Show solution (worked answer)

1. Helper formula `=PROPER(TRIM(A2))` gives `Mumbai`, `Mumbai`, `Bombay`, `Mumbai`, `Mumbai City`.
2. Paste Special → Values over the original.
3. Find & Replace with Match entire cell contents: `Bombay` → `Mumbai`, `Mumbai City` → `Mumbai`. For data that refreshes every week, build the same steps in Power Query so they repeat automatically.

INTERVIEW CORNER

Questions you may be asked

Q: How would you stop users from entering invalid data in a shared tracker? Use Data Validation (drop-down lists from a Table, number and date limits, custom formulas for uniqueness), add input messages, set error alerts to Stop, and protect the sheet so the rules cannot be removed. Check older data with Circle Invalid Data.

Q: How do you remove duplicates without losing information? First flag them with `COUNTIF` or conditional formatting and review which record to keep; decide the key columns that define a duplicate; work on a copy; then use Remove Duplicates (or `UNIQUE`) on those columns.

Q: What is the difference between TRIM and CLEAN? TRIM removes extra spaces; CLEAN removes non-printing characters such as line breaks. Web data often needs both, plus `SUBSTITUTE(…,CHAR(160),"")` for non-breaking spaces.

ADVANCED EXCEL · LESSON 3

Number, Custom & Conditional Formatting

Make numbers readable and insights jump out. Learn cell and number formats, write your own custom number formats (including lakhs and crores, K and M, and colour codes), and use conditional formatting with built-in rules and your own formulas.

Live class 2 | ✦ Excel for Microsoft 365, 2026 | Beginner | ⏱ 100 min

By the end of this lesson you can

- Apply cell formatting (fonts, alignment, borders, wrap, merge alternatives) the professional way
- Choose the right built-in number format and know that formatting never changes the stored value
- Write custom number formats with sections, conditions, colours, text and scaling
- Highlight insights with conditional formatting: rules, Top/Bottom, data bars, colour scales and icon sets
- Write custom-formula conditional formatting, such as highlighting whole rows

Formatting changes how a value looks, not what it is

This is the most important idea in this lesson. When you format `0.256` as `26%`, the cell still contains `0.256`. Every formula that uses the cell sees `0.256`.

X Same stored value, different formats

	A	B	C
1	Stored value	Format applied	What you see
2	0.256	Percentage, 0 decimals	26%
3	0.256	Percentage, 1 decimal	25.6%
4	1,234,568	Number with separator	1,234,567.89
5	46139	Date	27-Apr-2026
6	46139	Long date	Monday, 27 April 2026
7	0.6875	Time	16:30

```
C2   =TEXT(A2,"0%")
C3   =TEXT(A3,"0.0%")
C4   =TEXT(A4,"#,##0.00")
C5   =TEXT(A5,"dd-mmm-yyyy")
C6   =TEXT(A6,"dddd, d mmmm yyyy")
C7   =TEXT(A7,"hh:mm")
```

Column C uses the TEXT function to show exactly what each format produces. (TEXT returns text, so use it for labels, not for numbers you still need to calculate with.)

WATCH OUT **Rounding by format vs rounding by formula**

Formatting `2.456` with 1 decimal *shows* `2.5` but keeps `2.456`. Adding three such cells can show `7.4` while the visible numbers add up to `7.5`. If the rounded value must be used in calculations (invoices, tax), round it with a formula: `=ROUND(A2,1)`.

Cell formatting the professional way

Open **Format Cells** with `Ctrl + 1`. It has six tabs: Number, Alignment, Font, Border, Fill, Protection.

Goal	How
Bold header with fill colour	Home → Bold (`Ctrl + B`) and Fill Color, or Cell Styles → Heading
Wrap long text	Home → Wrap Text (`Alt + H, W`)
Centre a title across columns **without merging**	`Ctrl + 1` → Alignment → Horizontal: **Center Across Selection**
Rotate header text	Alignment → Orientation
Thin borders around a table	Home → Borders → All Borders
Copy formatting	Format Painter (double-click it to paint many places)
Clear only formatting	Home → Clear → Clear Formats
Consistent look across a workbook	Page Layout → Themes, and Cell Styles

COMMON MISTAKES **Avoid merged cells**

Merged cells break sorting, filtering, copying, selecting columns and pivot tables. Use **Center Across Selection** instead: it looks the same and keeps every cell independent.

Built-in number formats

Format	Example display of 1234.5	Use for
General	1234.5	Default, no fixed format
Number	1,234.50	Quantities, amounts
Currency	₹1,234.50	Money, symbol next to the number
Accounting	₹ 1,234.50	Money in financial statements: symbol aligned at the left edge, zero shown as a dash
Percentage	123450%	Rates; type 12% or format `0.12`
Fraction	1234 1/2	Measurements
Scientific	1.23E+03	Very large or small numbers
Text	1234.5 (as text)	IDs, codes, phone numbers
Date / Time	Various	Dates and times (stored as serial numbers)

Shortcuts: `Ctrl + Shift + 1` number with 2 decimals, `Ctrl + Shift + 4` currency, `Ctrl + Shift + 5` percentage, `Ctrl + Shift + 3` date.

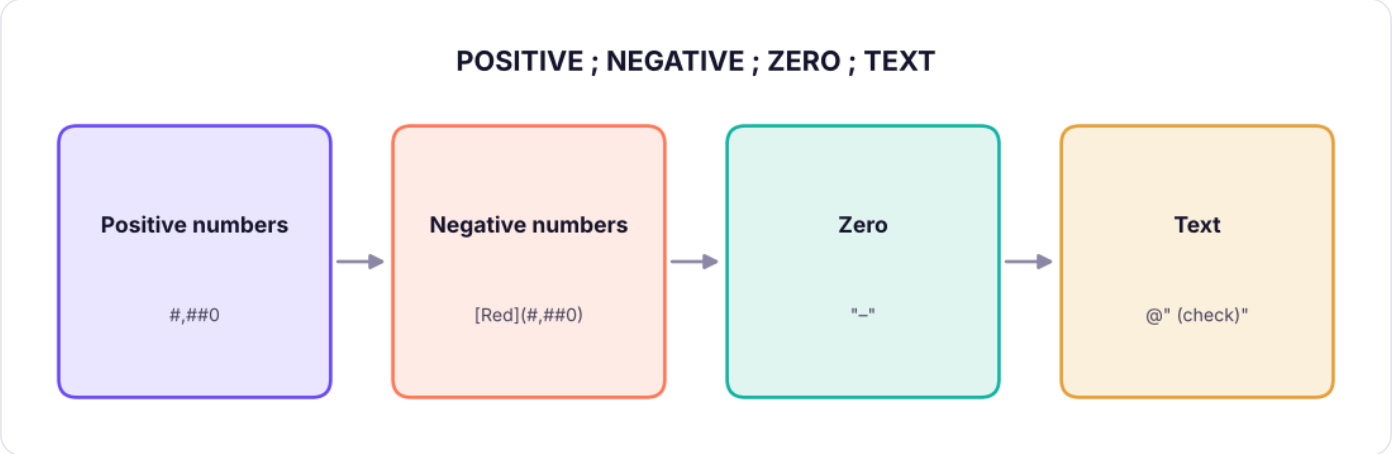
Custom number formats

`Ctrl + 1` → Number → **Custom** lets you write your own format code. Once you understand the structure, you can make dashboards that are much easier to read.

The four sections

A format code can have up to four parts, separated by semicolons:

```text
POSITIVE ; NEGATIVE ; ZERO ; TEXT
#,##0    ; [Red]-#,##0 ; "-" ; @
```



A custom number format has up to four sections separated by semicolons.

Placeholder characters

Code	Meaning	Value 7.5 →
`0`	Digit, shows a zero if none	`000.00` → 007.50
`#`	Digit, shows nothing if none	`#.##` → 7.5
`?`	Digit, shows a space (aligns decimals)	`?.??` → 7.5␣
`,` (in the middle)	Thousands separator	`#,##0`
`,` (at the end)	Divide by 1,000 for each comma	`#,##0,` shows 12,500 as 13
`%`	Multiply by 100 and add %	`0%`
`"text"`	Show literal text	`0" units"`
`@`	The cell's text	`"Dept: "@`
`[Red]`, `[Blue]`, `[Color10]`	Colour	`[Red]-0`
`[⩾1000]`	Condition	`[⩾1000]#,##0,"K";0`

Custom formats you will reuse

Each result in column C was produced by Excel's TEXT function with the format code in column B.

Custom format cookbook

	A	B	C
1	**Value**	**Format code**	**Result**
2	12500	#,##0	12,500
3	12500	#,##0,"K"	13K
4	12500	#,##0.0,"K"	12.5K
5	8450000	#,##0.0,,"M"	8.5M
6	-3200	#,##0;(#,##0)	(3,200)
7	0	#,##0;-#,##0;"–"	–
8	45	0" units"	45 units
9	0.0725	0.0%	7.3%
10	7	000	007
11	9876543210	00000 00000	98765 43210
12	1500	[>=1000]#,##0.0,"K";0	1.5K
13	850	[>=1000]#,##0.0,"K";0	850
14	0.25	▲ 0%;▼ 0%	▲ 25%
15	-0.12	▲ 0%;▼ 0%	▼ 12%

C2 `=TEXT(A2,B2)`

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

TIP **Up and down arrows in KPIs**

`▲ 0%;▼ 0%` (add `[Color10]` for green and `[Red]` for red: `[Color10]▲ 0%;[Red]▼ 0%`) is the classic format for "change versus last month" cells in dashboards. The arrow and colour change automatically with the sign.

Indian number format: lakhs and crores

Indian reports group digits as 12,34,56,789 (crore, lakh, thousand). If your Windows region is set to English (India), the standard `#,##0` already groups this way. Otherwise use a conditional format:

Text

```
[≥10000000]##\,##\,##\,##0;[≥100000]##\,##\,##0;##,##0
```

X Lakhs and crores

	A	B	C	D
1	Value	Indian grouping	In lakhs	In crores
2	1234567	12,34,567	12.35 L	0.12 Cr
3	98765432	9,87,65,432	987.65 L	9.88 Cr
4	450000	4,50,000	4.50 L	0.05 Cr

```
C2   =TEXT(A2/100000,"0.00")&" L"
D2   =TEXT(A2/10000000,"0.00")&" Cr"
C3   =TEXT(A3/100000,"0.00")&" L"
D3   =TEXT(A3/10000000,"0.00")&" Cr"
C4   =TEXT(A4/100000,"0.00")&" L"
D4   =TEXT(A4/10000000,"0.00")&" Cr"
```

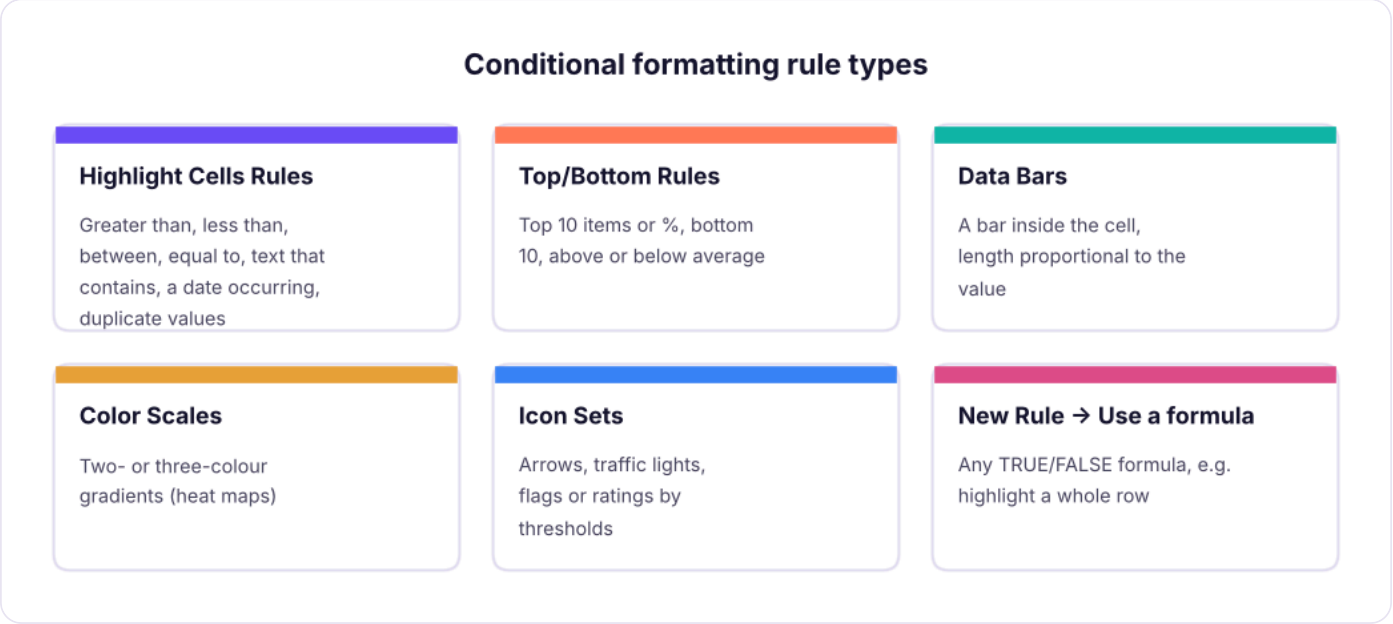
There is no "divide by 1 lakh" comma trick in Excel (each trailing comma divides by 1,000), so for lakhs either divide in a helper column (as above) or use Power BI's display units.

Date and time format codes

Code	Shows (for 27-04-2026 14:05)
`d, dd`	27
`ddd, dddd`	Mon, Monday
`m, mm`	4, 04
`mmm, mmmm`	Apr, April
`yy, yyyy`	26, 2026
`hh:mm`	14:05
`hh:mm AM/PM`	02:05 PM
`[h]:mm`	Total hours beyond 24, for timesheets (e.g. 37:30)
`"Q"` with a formula	Quarter labels need a formula: `="Q"&ROUNDUP(MONTH(A2)/3,0)`

Conditional formatting: let insights jump out

Home → Conditional Formatting changes a cell's look based on its value. It is how you turn a wall of numbers into something a manager understands in five seconds.



The five families of conditional formatting rules.

Highlight rules and Top/Bottom

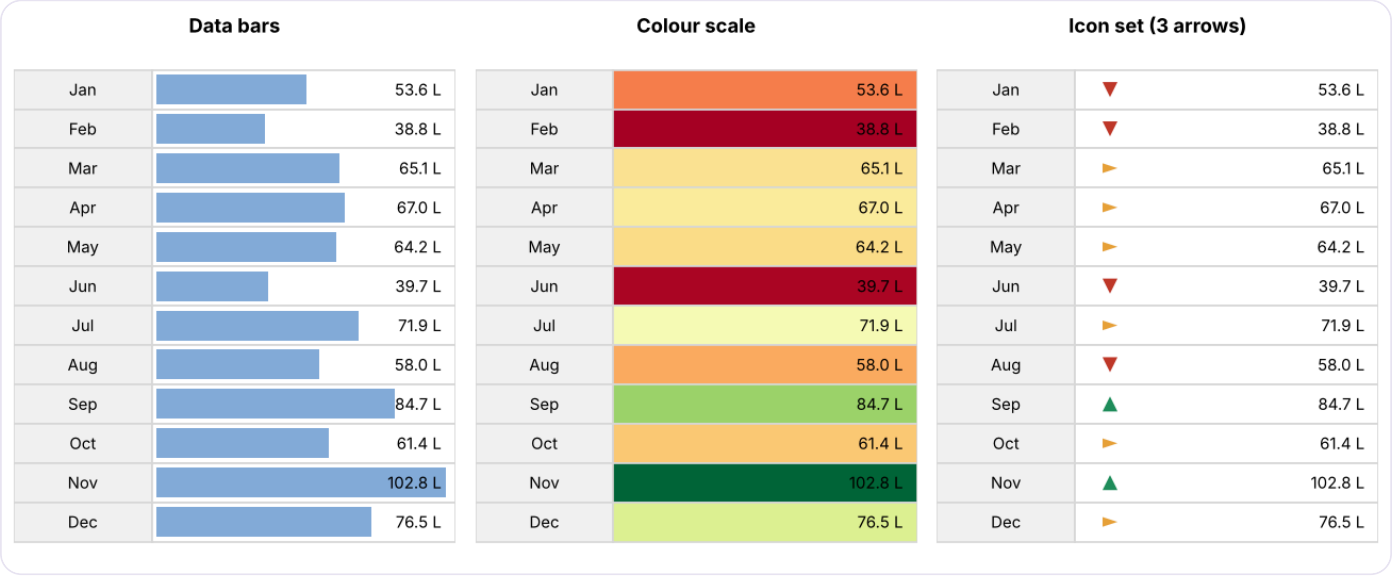
In the table below, green marks sales **above 1,50,000** and red marks **profit below 0** (both "Highlight Cells Rules"):

Highlight rules: sales > 1,50,000 (green), profit < 0 (red)

	A	B	C	D
1	Region	Category	Sales	Profit
2	North	Furniture	128,400	-3,250
3	North	Technology	212,950	28,400
4	South	Office Supplies	96,300	14,820
5	South	Technology	187,200	19,950
6	East	Furniture	74,100	-8,900
7	West	Technology	158,700	22,310

Data bars, colour scales and icon sets

These visual rules are best seen on real data. Here they are applied to monthly sales from the Superstore dataset:



Data bars (left), a green–yellow–red colour scale (middle) and icon sets (right) applied to the same monthly sales.

Rule	When to use	Watch out
Data bars	Compare magnitudes inside a table	Negative values draw leftwards; set a fixed minimum of 0 if needed
Colour scales	Heat maps: sales by region × month, attendance grids	Red/green is hard for colour-blind readers; consider blue–white–orange
Icon sets	Status at a glance: on track, at risk, off track	Default thresholds are percentages of the range (67% / 33%); set them to numbers or percentiles that match your business targets

Under **Manage Rules → Edit Rule** you can change thresholds, choose "Show Bar Only" or "Show Icon Only", and reverse the icon order.

Custom formula conditional formatting

New Rule → Use a formula to determine which cells to format is the most powerful option. Write a formula for the **top-left cell** of your selection that returns TRUE for cells you want formatted. Dollar signs matter:

- $ before the column ($D2) locks the rule to column D, so **every cell in the row** checks column D → the whole row is highlighted.
- No $ before the row number lets the row change as Excel moves down the selection.

Highlight whole rows where Department = Finance AND Salary > 1,00,000 — rule: =AND($C2="Finance",$F2>100000)

	A	B	C	D	E	F	G
1	Emp ID	Name	Department	City	Join date	Salary	Rating
2	E101	Kavya Iyer	HR	Mumbai	28-02-2016	97,000	4
3	E102	Imran Reddy	Finance	Delhi	14-12-2019	157,000	3
4	E103	Harpreet Kulkarni	Finance	Mumbai	05-10-2024	131,000	4
5	E104	Diya Iyer	HR	Bengaluru	02-01-2024	90,000	4
6	E110	Vikram Joshi	Finance	Chennai	19-10-2022	75,000	5
7	E119	Rohan Das	Finance	Delhi	19-07-2018	97,000	5

Rows E102 and E103 meet both conditions. E110 and E119 are in Finance but earn less than 1,00,000, so they stay white.

More custom-formula recipes

Goal	Select	Formula
Zebra stripes (every other row)	the table	`=MOD(ROW(),2)=0`
Overdue tasks (due date before today, status not Done)	the table	`=AND($E2<TODAY(),$F2<>"Done")`
Values above the column average	`C2:C100`	`=C2>AVERAGE($C$2:$C$100)`
Duplicate IDs	`A2:A100`	`=COUNTIF($A$2:$A$100,A2)>1`
Weekend dates	`A2:A100`	`=WEEKDAY(A2,2)>5`
Row of the employee picked in a drop-down (H1)	the table	`=$B2=$H$1`
Missing values	the table	`=ISBLANK(A2)` or `=A2=""`
Top 3 per column	`C2:C100`	`=C2⩾LARGE($C$2:$C$100,3)`

STEP BY STEP **Create a whole-row highlight**

1. Select the data **without** the header (for example `A2:G21`). The active cell must be in row 2.
2. Home → Conditional Formatting → New Rule → **Use a formula to determine which cells to format**.
3. Formula: `=AND($C2="Finance",$F2>100000)`.
4. Click **Format...** → Fill → choose a colour → OK.
5. To edit later: Conditional Formatting → **Manage Rules** (choose "This Worksheet" to see all rules). Rules higher in the list win; tick **Stop If True** to prevent lower rules from also applying.

COMMON MISTAKES **The three most common conditional-formatting bugs**

1. **Wrong anchor**: writing the formula for row 1 while the selection starts at row 2 shifts every result by one row.
2. **Missing $**: `=C2="Finance"` (no $) checks a different column for each cell, so only part of the row lights up.
3. **Copying cells copies rules**: pasting formatted cells many times creates hundreds of duplicate rules and slows the workbook. Clean up in Manage Rules, or paste values only.

Formatting a report: before and after

A clean report follows a few simple rules:

1. **Right-align numbers**, left-align text, and use the same number of decimals in a column.
2. Use **thousands separators**; show large values in K/M or lakh/crore with clear labels.
3. **Bold** headers and totals only; avoid using many colours.
4. Use **one highlight colour** for "needs attention" and keep everything else neutral.
5. **Freeze** the header row; make the column widths fit the content.
6. Add a title, the date range and the data source.

DO IT WITH AI **Do it with AI**

Ask an assistant: *"Write an Excel custom number format that shows values above 1 crore as '1.25 Cr', values above 1 lakh as '4.50 L', and smaller values with Indian comma grouping."* Then test the answer with `=TEXT(value, "format")` on a few numbers before applying it. (A pure format can only scale by thousands, so a good answer will suggest a helper column or Power BI for lakhs and crores.)

Summary

KEY TAKEAWAYS

- Formatting changes the **display**, never the stored value. Round with `ROUND` when it matters.
- Avoid merged cells; use **Center Across Selection**.
- Custom formats have four sections, `positive;negative;zero;text`, and support colours, conditions, text and scaling with trailing commas.
- Conditional formatting families: highlight rules, Top/Bottom, data bars, colour scales and icon sets.
- For whole-row highlights use a formula with $ before the column: `=$C2="Finance"`.
- Manage Rules shows priority; rule order and Stop If True decide which format wins.

Quick quiz

6 questions · tap an answer to check it

1. A cell contains 0.4567 formatted as 0%. What value does `=A1*100` return?

A 46

B 45.67

C 0.46

D 46%

Answer: B. The stored value is still 0.4567, so the formula returns 45.67. Formatting only changes the display.

2. Which custom format shows 8,450,000 as 8.5M?

A `#,##0,"M"`

B `#,##0.0,,"M"`

C `0.0"M"`

D `#,##0.0%`

Answer: B. Two trailing commas divide by 1,000,000, and 0.0 shows one decimal.

3. In the format `#,##0;[Red](#,##0);"-"`, how is -2500 shown?

A -2,500 in black

B (2,500) in red

C –

D 2,500 in red

Answer: B. The second section is for negative numbers: red, in brackets, without a minus sign.

4. You want to highlight entire rows where Status (column F) is "Late". Rows start at 2. Which rule is correct?

A `=F2="Late"`

B `=$F2="Late"`

C `=$F$2="Late"`

D `=F$2="Late"`

Answer: B. Locking the column with $ makes every cell in the row test column F; leaving the row free lets each row test itself.

5. What does "Center Across Selection" replace?

A Wrap Text

B Merge & Center

C Format Painter

D Freeze Panes

Answer: B. It centres text across several cells without merging them, so sorting and selecting still work.

6. Which conditional formatting type is best for a sales heat map of region × month?

A Icon sets

B Data bars

C Colour scales

D Highlight duplicates

Answer: C. Colour scales shade each cell by value, which creates a heat map.

EXERCISE **Dashboard number formats**

Write custom formats for: (a) revenue shown in thousands with "K" and no decimals; (b) month-on-month change shown as a green up arrow with one decimal percentage when positive, and a red down arrow when negative; (c) a 6-digit PIN code that must keep leading zeros.

Show solution (worked answer)

(a) `#,##0,"K"` (b) `[Color10]▲ 0.0%;[Red]▼ 0.0%;"-"` (c) `000000` (or store the PIN as text).

EXERCISE **Conditional formatting practice**

Using `employees_small.csv`: (1) colour-scale the salary column; (2) add an icon set to rating with green for 5, amber for 3–4 and red for 1–2; (3) highlight the whole row of anyone who joined after 1-Jan-2024.

Show solution (worked answer)

(1) Select F2:F21 → Color Scales → Green–Yellow–Red. (2) Select G2:G21 → Icon Sets → 3 traffic lights → Manage Rules → Edit: green when ≥ 5 (Number), amber when ≥ 3 (Number), otherwise red. (3) Select A2:G21 → New Rule → formula `=$E2>DATE(2024,1,1)` → choose a fill.

INTERVIEW CORNER **Questions you may be asked**

Q: What is the difference between formatting a number and rounding it? Formatting changes only how the value looks; the full value is used in calculations. `ROUND` changes the stored value. Use `ROUND` when the rounded figure must flow into other calculations, such as invoices.

Q: How would you highlight the top 10% of customers by revenue? Conditional Formatting → Top/Bottom Rules → Top 10% (change 10 to any value), or a formula rule `=C2 ⩾ PERCENTILE.INC($C$2:$C$500,0.9)` for more control.

Q: Why might a conditional format highlight the wrong rows? The formula was written for a different starting row than the active cell of the selection, or the $ signs are wrong.

ADVANCED EXCEL · LESSON 4

Formulas, Cell References & Aggregate Functions

Write formulas with confidence. Understand operators and their order, relative, absolute and mixed references, and every aggregate function you need, from SUM and COUNTA to SMALL, LARGE, SUMIFS, COUNTIFS, MAXIFS, SUBTOTAL and AGGREGATE.

Live class 3 · ✦ Excel for Microsoft 365, 2026 · Beginner · ⏱ 110 min

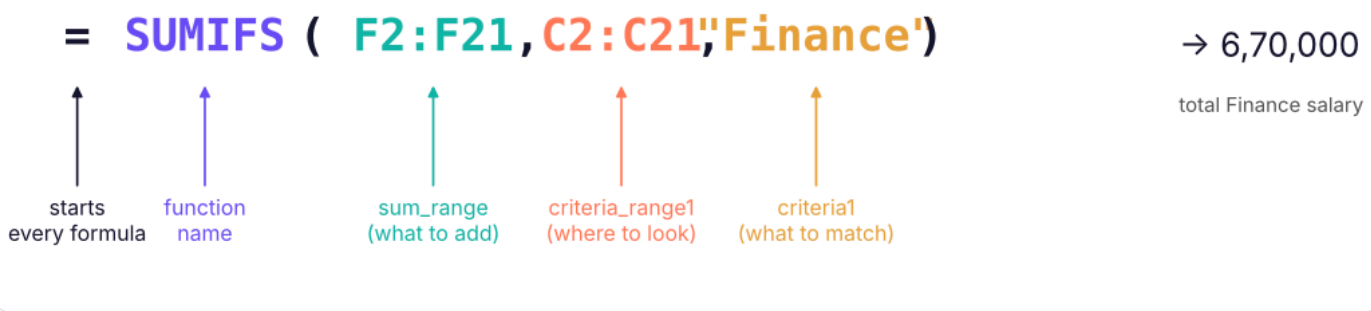
By the end of this lesson you can

- Explain the difference between a formula and a function and read any function's syntax
- Predict how a formula changes when copied, and lock references with $ (F4)
- Summarise data with SUM, COUNT, COUNTA, COUNTBLANK, AVERAGE, MIN, MAX, SMALL, LARGE, MEDIAN and MODE
- Answer conditional questions with COUNTIF(S), SUMIF(S), AVERAGEIF(S), MAXIFS and MINIFS
- Use SUBTOTAL and AGGREGATE for filtered data and data with errors

Formula vs function

DEFINITION **Two words people mix up**

A **formula** is any calculation that starts with `=`. It can use cell references, numbers, operators and functions: `=B2*C2`, `=(B2-C2)/C2`, `=SUM(B2:B10)*1.18`. A **function** is a named, ready-made calculation inside a formula: `SUM`, `AVERAGE`, `IF`. A function has a name, brackets and **arguments** separated by commas.



Anatomy of a function call. Arguments in square brackets are optional.

When you type a function name and `(`, Excel shows a tooltip with its syntax, for example `SUMIFS(sum_range, criteria_range1, criteria1, [criteria_range2, criteria2], …)`. Press `Ctrl + A` after the bracket to open the **Function Arguments** dialog, which explains every argument and previews the result.

Operators and order of calculation

Operator	Meaning	Example	Result
`+ - * /`	Add, subtract, multiply, divide	`=10+4*2`	18
`^`	Power	`=2^10`	1024
`%`	Percent	`=500*18%`	90
`&`	Join text	`="Q"&2`	Q2
`= <> > < ⩾ ⩽`	Comparison, returns TRUE/FALSE	`=B2⩾100`	TRUE or FALSE
`:`	Range	`A1:A10`	10 cells
`,`	Union (inside functions)	`=SUM(A1:A3,C1:C3)`	Adds both ranges

Excel follows the usual maths order, remembered as **BODMAS/PEMDAS**: brackets, then `^`, then `*` and `/`, then `+` and `-`, then `&`, then comparisons.

Order of operations

	A	B	C
1	Formula	Result	Why
2	=10+4*2	18	Multiply first: 10 + 8
3	=(10+4)*2	28	Brackets first: 14 × 2
4	=2^3^2	64	Excel works left to right: (2^3)^2 = 64
5	=-2^2	4	Negation before power in Excel: (-2)^2 = 4
6	=100*(1+18%)	118	Add 18% GST

```
B2    =10+4*2
B3    =(10+4)*2
B4    =2^3^2
B5    =-2^2
B6    =100*(1+18%)
```

WATCH OUT **`-2^2` is 4 in Excel**

In maths -2^2 is −4, but Excel applies the minus sign first. Write `=-(2^2)` if you mean −4. This catches people in finance formulas.

Cell references: relative, absolute and mixed

When you copy a formula, Excel adjusts its references **relative** to where you paste it. Usually that is exactly what you want. Sometimes a reference must stay fixed, for example a tax rate in one cell. A $ sign locks the part after

it.

Reference	Copy down	Copy right	Name
B2	B3, B4 …	C2, D2 …	Relative
$B$2	stays $B$2	stays $B$2	Absolute
B$2	stays row 2	C$2, D$2 …	Mixed (row locked)
$B2	B3, B4 …	stays column B	Mixed (column locked)

Press **F4** while the cursor is on a reference to cycle `B2` → `$B$2` → `B$2` → `$B2` → `B2`.

Absolute reference: one rate for every row

GST rate in one input cell (G2) — formula in D2 is =B2*C2*$G$2, copied down

	A	B	C	D	E	F	G
1	Item	Qty	Price	GST amount	Total		GST rate
2	Notebook	10	45	81.00	531.00		18%
3	Pen pack	25	60	270.00	1,770.00		
4	Stapler	4	180	129.60	849.60		

```
D2   =B2*C2*$G$2
E2   =B2*C2+D2
D3   =B3*C3*$G$2
E3   =B3*C3+D3
D4   =B4*C4*$G$2
E4   =B4*C4+D4
```

The GST rate lives in one input cell (yellow). Because `$G$2` is absolute, every copied formula still points to it. Change the rate once and every total updates.

Mixed references: the multiplication table test

A classic interview question: write **one** formula in B2 that can be copied across and down to fill a multiplication table.

One formula, =$A2*B$1, filled across and down

	A	B	C	D	E	F
1	×	1	2	3	4	5
2	1	1	2	3	4	5
3	2	2	4	6	8	10
4	3	3	6	9	12	15
5	4	4	8	12	16	20
6	5	5	10	15	20	25

```
B2   =$A2*B$1
C2   =$A2*C$1
D2   =$A2*D$1
```

```
E2    =$A2*E$1
F2    =$A2*F$1
```
Each pattern is copied down/across to the other cells in its column.

$A2 always uses column A (the row header) and B$1 always uses row 1 (the column header). The same idea is used for price × discount grids, sales × commission tables, and loan EMI tables.

The same table in formula view

Formula view (Ctrl + `)

	A	B	C	D
1	×	1	2	3
2	1	=$A2*B$1	=$A2*C$1	=$A2*D$1
3	2	=$A3*B$1	=$A3*C$1	=$A3*D$1
4	3	=$A4*B$1	=$A4*C$1	=$A4*D$1

References to other sheets and workbooks

Reference	Meaning
`=Sales!B2`	Cell B2 on the sheet named Sales
`='Sales 2026'!B2`	Quotes are needed when the sheet name has spaces
`=SUM(Jan:Mar!B2)`	A **3D reference**: B2 on every sheet from Jan to Mar
`=[Budget.xlsx]Plan!C5`	Another workbook (an external link; Data → Edit Links manages them)

Aggregate functions

Aggregate functions turn many values into one number. The examples below use the 20-row employee table stored on a sheet named Emp (columns: A emp_id, B name, C department, D city, E join_date, F salary, G rating).

Aggregate functions on Emp (salary in F2:F21, rating in G2:G21)

	A	B	C
1	Question	Formula	Result
2	Total salary	=SUM(Emp!F2:F21)	1,978,000
3	How many numbers in salary	=COUNT(Emp!F2:F21)	20
4	How many non-empty names	=COUNTA(Emp!B2:B21)	20
5	Empty cells in B2:B25	=COUNTBLANK(Emp!B2:B25)	4
6	Average salary	=AVERAGE(Emp!F2:F21)	98,900
7	Median salary	=MEDIAN(Emp!F2:F21)	97,000
8	Most common rating	=MODE.SNGL(Emp!G2:G21)	4
9	Lowest / highest salary	=MIN(…) / =MAX(…)	45,000 / 157,000
10	2nd highest salary	=LARGE(Emp!F2:F21,2)	157,000
11	3rd lowest salary	=SMALL(Emp!F2:F21,3)	57,000
12	Average rating (rounded)	=ROUND(AVERAGE(Emp!G2:G21),2)	3.55

```
C2    =SUM(Emp!F2:F21)
C3    =COUNT(Emp!F2:F21)
```

```
C4   =COUNTA(Emp!B2:B21)
C5   =COUNTBLANK(Emp!B2:B25)
C6   =AVERAGE(Emp!F2:F21)
C7   =MEDIAN(Emp!F2:F21)
C8   =MODE.SNGL(Emp!G2:G21)
C9   =TEXT(MIN(Emp!F2:F21),"#,##0")&" / "&TEXT(MAX(Emp!F2:F21),"#,##0")
C10  =LARGE(Emp!F2:F21,2)
C11  =SMALL(Emp!F2:F21,3)
C12  =ROUND(AVERAGE(Emp!G2:G21),2)
```

Function	Counts or uses	Ignores
`COUNT`	Numbers (and dates)	Text, blanks, logical values typed in cells
`COUNTA`	Anything not empty (numbers, text, errors, `""` from formulas)	Truly empty cells
`COUNTBLANK`	Empty cells and cells with `""`	Everything else
`SUM`, `AVERAGE`, `MIN`, `MAX`	Numbers	Text and blanks in ranges
`MEDIAN`	Middle value of sorted numbers	Text and blanks
`MODE.SNGL`	Most frequent number	Text
`SMALL(range,k)`, `LARGE(range,k)`	k-th smallest or largest	Text and blanks

TIP **AVERAGE ignores blanks but not zeros**

A blank cell is left out of the average; a zero is counted. If "no data" is stored as 0, your average will be too low. Leave missing values blank, or use `AVERAGEIF(range,"<>0")`.

2026 UPDATE **MEDIAN vs AVERAGE in salary data**

Salaries are usually skewed by a few high earners, so the **median** describes a "typical" employee better than the average. You will meet this again in Statistics (ML track, Lesson 1).

Conditional aggregation: the IFS family

These are the most used analysis functions in Excel. They answer questions such as *"total sales in the North region for Technology in 2025"*.

Function	Syntax
COUNTIF	COUNTIF(range, criteria)
COUNTIFS	COUNTIFS(range1, criteria1, range2, criteria2, …)
SUMIF	SUMIF(range, criteria, [sum_range])
SUMIFS	SUMIFS(sum_range, range1, criteria1, range2, criteria2, …)
AVERAGEIF(S)	Same pattern as SUMIF(S)
MAXIFS, MINIFS	MAXIFS(max_range, range1, criteria1, …)

Note that SUMIF puts the sum range **last** while SUMIFS puts it **first**. Learn the SUMIFS pattern and use it even for one condition.

Conditional aggregation on Emp

	A	B	C
1	Question	Result	Formula
2	Employees in Finance	7	=COUNTIF(Emp!C2:C21,"Finance")
3	Finance employees in Delhi	2	=COUNTIFS(Emp!C2:C21,"Finance",Emp!D2:D21,"Delhi")
4	Total Finance salary	670,000	=SUMIFS(Emp!F2:F21,Emp!C2:C21,"Finance")
5	Salary of people rated 4 or more	1,102,000	=SUMIFS(Emp!F2:F21,Emp!G2:G21,">=4")
6	Average salary in HR	96,000	=AVERAGEIFS(Emp!F2:F21,Emp!C2:C21,"HR")
7	Highest salary in Sales	156,000	=MAXIFS(Emp!F2:F21,Emp!C2:C21,"Sales")
8	Joined in 2024 or later	6	=COUNTIFS(Emp!E2:E21,">="&DATE(2024,1,1))
9	Names starting with "A"	4	=COUNTIF(Emp!B2:B21,"A*")
10	Salary between 80k and 1.2 lakh	9	=COUNTIFS(Emp!F2:F21,">=80000",Emp!F2:F21,"<=120000")

```
B2   =COUNTIF(Emp!C2:C21,"Finance")
B3   =COUNTIFS(Emp!C2:C21,"Finance",Emp!D2:D21,"Delhi")
B4   =SUMIFS(Emp!F2:F21,Emp!C2:C21,"Finance")
B5   =SUMIFS(Emp!F2:F21,Emp!G2:G21,"⩾4")
B6   =AVERAGEIFS(Emp!F2:F21,Emp!C2:C21,"HR")
B7   =MAXIFS(Emp!F2:F21,Emp!C2:C21,"Sales")
B8   =COUNTIFS(Emp!E2:E21,"⩾"&DATE(2024,1,1))
B9   =COUNTIF(Emp!B2:B21,"A*")
B10  =COUNTIFS(Emp!F2:F21,"⩾80000",Emp!F2:F21,"⩽120000")
```

Let's confirm two of these with pandas, using the same file:

Python

```python
import pandas as pd
emp = pd.read_csv("employees_small.csv", parse_dates=["join_date"])
print("Finance headcount:", (emp.department == "Finance").sum())
print("Total Finance salary:", f"{emp.loc[emp.department == 'Finance', 'salary'].sum():,}")
print("Joined 2024 or later:", (emp.join_date ⩾ "2024-01-01").sum())
```

OUTPUT

```
Finance headcount: 7
Total Finance salary: 670,000
Joined 2024 or later: 6
```

Writing criteria correctly

Criteria	Meaning
`"Finance"`	Equal to Finance (not case-sensitive)
`"<>Finance"`	Not equal to Finance
`"⩾4"`	Greater than or equal to 4 (operator and number inside quotes)
`"⩾"&H1`	Compare with the value in H1 (join with &)
`"⩾"&DATE(2024,1,1)`	Compare with a date
`"*pune*"`	Contains "pune"
`"A*"` / `"*a"`	Starts with A / ends with a
`"?????"`	Exactly five characters
`""`	Blank cells
`"<>"`	Non-blank cells

TIP **Build a summary table with one formula**

Put departments in A2:A6 and the formula `=SUMIFS(Emp!$F$2:$F$21,Emp!$C$2:$C$21,$A2)` in B2, then copy down. The $ signs keep the data ranges fixed while $A2 moves to each department. In Microsoft 365 you can produce the department list itself with `=UNIQUE(Emp!C2:C21)` (Lesson 6).

X Department summary built with SUMIFS and COUNTIFS

	A	B	C	D	E
1	**Department**	**Headcount**	**Total salary**	**Average salary**	**% of payroll**
2	Finance	7	670,000	95,714	33.9%
3	HR	6	576,000	96,000	29.1%
4	IT	1	112,000	112,000	5.7%
5	Operations	2	261,000	130,500	13.2%
6	Sales	4	359,000	89,750	18.1%
7	**Total**	**20**	**1,978,000**	**98,900**	**100.0%**

```
B2   =COUNTIFS(Emp!$C$2:$C$21,$A2)
C2   =SUMIFS(Emp!$F$2:$F$21,Emp!$C$2:$C$21,$A2)
D2   =C2/B2
E2   =C2/$C$7
B7   =SUM(B2:B6)
C7   =SUM(C2:C6)
E7   =SUM(E2:E6)
```

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

This little table is exactly what a pivot table does (Lesson 8), but with formulas you control every cell, which is useful for fixed report layouts.

SUBTOTAL and AGGREGATE: totals for filtered data

`SUM` adds every row, including rows hidden by a filter. `SUBTOTAL` adds only **visible** rows, so the total changes as you filter.

```
=SUBTOTAL(function_num, range)
```

function_num (ignores filtered rows)	101–111 also ignore manually hidden rows	Function
1	101	AVERAGE
2	102	COUNT
3	103	COUNTA
4	104	MAX
5	105	MIN
9	109	SUM

`AGGREGATE` does the same and can also **ignore error values**: `=AGGREGATE(9, 6, D2:D100)` sums D2:D100 ignoring errors (option 6). `=AGGREGATE(14, 6, D2:D100, 2)` returns the 2nd largest value ignoring errors (14 = LARGE).

AGGREGATE ignores the #DIV/0! error that breaks SUM

	A	B
1	Value	
2	120	
3	#DIV/0!	
4	80	
5	SUM →	#DIV/0!
6	AGGREGATE(9,6,...) →	200

```
A3   =10/0
B5   =SUM(A2:A4)
B6   =AGGREGATE(9,6,A2:A4)
```

Common formula errors and how to fix them

Error	Example	Fix
`#DIV/0!`	`=C2/B2` with B2 = 0	`=IFERROR(C2/B2,0)` or `=IF(B2=0,"",C2/B2)`
`#VALUE!`	`=A2+B2` where A2 is text	Convert text to numbers; check for spaces
`#REF!`	Deleted a referenced row	Undo, or rewrite the formula
`#NAME?`	`=SUMIFS(…,Finance)` without quotes	Put text criteria in quotes
Circular reference	`=SUM(A1:A10)` typed in A10	Move the total outside the range
Wrong total after sorting	Mixed relative/absolute references	Lock ranges with $

Formulas → Formula Auditing helps: **Trace Precedents** draws arrows to the cells a formula uses, **Trace Dependents** to cells that use it, **Show Formulas** (`Ctrl + `` `) shows every formula, and **Evaluate Formula** steps through a calculation (Lesson 11).

DO IT WITH AI

Do it with AI

Paste a formula into Copilot or a chat assistant and ask: *"Explain this Excel formula step by step and tell me what would make it return an error: `=SUMIFS(Sales!$F:$F,Sales!$C:$C,$A2,Sales!$E:$E,"≥"&$B$1)`."* Use AI to explain and draft formulas, then check the result on a few rows you can verify by hand.

Summary

KEY TAKEAWAYS

- Formula = any calculation starting with =; function = a named calculation such as `SUM`.
- Order: brackets, power, multiply/divide, add/subtract, join, compare. In Excel `-2^2 = 4`.
- $ locks a reference; **F4** cycles relative → absolute → mixed. Mixed references build 2-D tables with one formula.
- `COUNT` counts numbers; `COUNTA` counts non-empty; `COUNTBLANK` counts empty.
- `SUMIFS(sum_range, criteria_range, criteria, …)` and `COUNTIFS` answer most business questions.
- Criteria with operators go in quotes and join cell values with &: `"≥"&H1`.
- `SUBTOTAL(109,…)` sums only visible rows; `AGGREGATE` can also ignore errors.

Quick quiz

6 questions · tap an answer to check it

1. You copy `=B2*$E$1` from C2 to C5. What is the formula in C5?

A `=B5*$E$4`

B `=B5*$E$1`

C `=B2*$E$1`

D `=E5*$B$1`

Answer: B. B2 is relative and moves to B5; $E$1 is absolute and stays.

2. Which formula, typed in B2 and copied across and down, builds a multiplication table with headers in row 1 and column A?

A `=A2*B1`

B `=$A$2*$B$1`

C `=$A2*B$1`

D `=A$2*$B1`

Answer: C. Column A is locked for the row header, and row 1 is locked for the column header.

3. A range contains 10, "abc", blank, 25, TRUE (typed). What does COUNT return?

A 2

B 3

C 4

D 5

Answer: A. COUNT counts only numbers in a range: 10 and 25.

4. Which formula totals Sales (D) where Region (B) is "North" AND Category (C) is "Technology"?

A `=SUMIF(B:B,"North",D:D)`

B `=SUMIFS(D:D,B:B,"North",C:C,"Technology")`

C `=SUMIFS(B:B,"North",C:C,"Technology",D:D)`

D `=COUNTIFS(B:B,"North",C:C,"Technology")`

Answer: B. SUMIFS takes the sum range first, then pairs of criteria range and criteria.

5. How do you count values in A2:A100 greater than the value in H1?

A `=COUNTIF(A2:A100,>H1)`

B `=COUNTIF(A2:A100,">H1")`

C `=COUNTIF(A2:A100,">"&H1)`

D `=COUNTIF(A2:A100,H1>)`

Answer: C. The operator is text and is joined to the cell value with &.

6. You filter a table to show only Pune. Which total updates to show only Pune's sales?

A `=SUM(D2:D500)`

B `=SUBTOTAL(109,D2:D500)`

C `=SUMIF(D2:D500,">0")`

D `=AGGREGATE(9,0,D2:D500)` with option 0

Answer: B. SUBTOTAL 9/109 ignores rows hidden by a filter.

EXERCISE **Superstore questions with formulas**

Open `superstore_sales.csv` in Excel. Using only aggregate and IFS functions, find: (1) total sales; (2) number of order lines in the West region; (3) total profit for Technology in 2025; (4) average discount for Furniture; (5) the largest single order-line sale in the South.

Show solution (worked answer)

Assuming columns: J region, L category, O sales, Q discount, R profit and B order_date (check your own column letters): (1) `=SUM(O:O)` (2) `=COUNTIFS(J:J,"West")` (3) `=SUMIFS(R:R,L:L,"Technology",B:B,"≥"&DATE(2025,1,1),B:B,"≤"&DATE(2025,12,31))` (4) `=AVERAGEIFS(Q:Q,L:L,"Furniture")` (5) `=MAXIFS(O:O,J:J,"South")` Check your answers in Python with pandas (Python track, Lesson 22) or with a pivot table (Lesson 8): they must match exactly.

INTERVIEW CORNER **Questions you may be asked**

Q: What is the difference between COUNT, COUNTA and COUNTBLANK? COUNT counts numeric cells, COUNTA counts every non-empty cell (including text and formulas returning ""), COUNTBLANK counts empty cells and cells containing "".

Q: When would you use a mixed reference? When a formula is copied in two directions and must keep either the row or the column fixed, for example a price × quantity grid or a commission table: `=$A2*B$1`.

Q: Why use SUMIFS instead of SUMIF? SUMIFS supports multiple conditions, has a consistent argument order (sum range first), and works for one condition too, so one pattern covers every case.

Q: How do you total only visible rows after filtering? Use `SUBTOTAL(109, range)` or `AGGREGATE(9, 5, range)`; the Table Total Row does this automatically.

ADVANCED EXCEL · LESSON 5

Text Functions, Logical Functions & IF

Extract, clean, combine and reformat text with LEFT, RIGHT, MID, LEN, FIND, SEARCH, CONCAT, TEXTJOIN, TEXT and the new TEXTSPLIT/TEXTBEFORE/TEXTAFTER. Then make decisions with AND, OR, NOT, IF, nested IF, IFS, SWITCH and IFERROR.

Live classes 3–4 | ✦ Excel for Microsoft 365, 2026 | Beginner | ⏱ 120 min

By the end of this lesson you can

- Extract parts of codes, names and emails with LEFT, RIGHT, MID, FIND and SEARCH
- Combine text with &, CONCAT and TEXTJOIN and format numbers inside text with TEXT
- Split text with Text to Columns and the TEXTSPLIT, TEXTBEFORE and TEXTAFTER functions
- Build conditions with AND, OR, NOT and decisions with IF, nested IF, IFS and SWITCH
- Handle errors cleanly with IFERROR and IFNA

Why text functions matter

Real data arrives as text: order IDs like `IN-2025-00042`, product codes, email addresses, "Last, First" names, addresses and free-text comments. Text functions let you pull out the part you need (the year, the region code, the domain) so you can analyse it.

Measuring and extracting text

Function	What it returns	Example	Result
`LEN(text)`	Number of characters	`=LEN("Pune")`	4
`LEFT(text, n)`	First n characters	`=LEFT("IN-2025-00042",2)`	IN
`RIGHT(text, n)`	Last n characters	`=RIGHT("IN-2025-00042",5)`	00042
`MID(text, start, n)`	n characters from position start	`=MID("IN-2025-00042",4,4)`	2025
`FIND(find, text, [start])`	Position of text (case-sensitive, no wildcards)	`=FIND("-","IN-2025-00042")`	3
`SEARCH(find, text, [start])`	Position (not case-sensitive, wildcards allowed)	`=SEARCH("pune","PUNE-411001")`	1

Extracting parts of an order ID

	A	B	C	D	E	F
1	Order ID	Country =LEFT(A2,2)	Year =MID(A2,4,4)	Order no. =RIGHT(A2,5)	Length	Year as number
2	IN-2025-00042	IN	2025	00042	13	2025
3	IN-2023-01519	IN	2023	01519	13	2023
4	IN-2024-02988	IN	2024	02988	13	2024

```
B2   =LEFT(A2,2)
C2   =MID(A2,4,4)
D2   =RIGHT(A2,5)
E2   =LEN(A2)
F2   =VALUE(MID(A2,4,4))
```

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

Text functions always return **text**. `MID(A2,4,4)` returns the text `"2025"`, not the number 2025. Wrap with `VALUE()` (or multiply by 1) to get a number you can compare or add.

When the position changes: FIND + MID

Codes rarely have fixed lengths. Combine `FIND` with `LEFT`, `MID` and `RIGHT` to cut at a delimiter wherever it is.

Splitting emails at the @ sign

	A	B	C	D
1	Email	Position of @	User name	Domain
2	kavya.iyer@abc.in	11	kavya.iyer	abc.in
3	imran@datacorp.com	6	imran	datacorp.com
4	harpreet.k@skillnavora.in	11	harpreet.k	skillnavora.in

```
B2   =FIND("@",A2)
C2   =LEFT(A2,FIND("@",A2)-1)
D2   =MID(A2,FIND("@",A2)+1,100)
```

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

Using a large number like 100 in `MID` simply means "take everything to the end". Alternatively use `=RIGHT(A2,LEN(A2)-FIND("@",A2))`.

First and last name from 'First Last'

	A	B	C	D
1	Full name	First name	Last name	Initials
2	Kavya Iyer	Kavya	Iyer	KI
3	Harpreet Kulkarni	Harpreet	Kulkarni	HK
4	Aarav Kulkarni	Aarav	Kulkarni	AK

```
B2   =LEFT(A2,FIND(" ",A2)-1)
C2   =MID(A2,FIND(" ",A2)+1,50)
D2   =LEFT(A2,1)&LEFT(C2,1)
```

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

FIND vs SEARCH

	FIND	SEARCH
Case-sensitive	Yes: `FIND("a","Apple")` → #VALUE!	No: `SEARCH("a","Apple")` → 1
Wildcards * ?	No	Yes
Not found	`#VALUE!`	`#VALUE!`

A common pattern to test whether text **contains** a word: `=ISNUMBER(SEARCH("chair",B2))` returns TRUE or FALSE.

Changing and cleaning text

Function	Purpose	Example → Result
`UPPER, LOWER, PROPER`	Change case	`=PROPER("new delhi")` → New Delhi
`TRIM`	Remove extra spaces	`=TRIM(" a b ")` → a b
`CLEAN`	Remove non-printing characters	Line breaks from web data
`SUBSTITUTE(text, old, new, [n])`	Replace by content	`=SUBSTITUTE("2025-04-27","-","/")` → 2025/04/27
`REPLACE(text, start, n, new)`	Replace by position	`=REPLACE("E101",1,1,"EMP")` → EMP101
`REPT(text, n)`	Repeat text	`=REPT("★",4)` → ★★★★
`EXACT(a, b)`	Case-sensitive comparison	`=EXACT("abc","ABC")` → FALSE

> **TIP** **Count how many times a character appears**
>
> `=LEN(A2)-LEN(SUBSTITUTE(A2,",",""))` counts commas in A2 (the length difference after removing them). Add 1 to count items in a comma-separated list.

Joining text

Three ways to combine text

	A	B	C	D	E	F
1	First	Last	City	With &	CONCAT	TEXTJOIN
2	Kavya	Iyer	Mumbai	Kavya Iyer (Mumbai)	Kavya Iyer	Kavya, Iyer, Mumbai
3	Imran		Delhi	Imran (Delhi)	Imran	Imran, Delhi

```
D2   =A2&" "&B2&" ("&C2&")"
```

```
E2   =CONCAT(A2," ",B2)
F2   =TEXTJOIN(", ",TRUE,A2:C2)
D3   =A3&" "&B3&" ("&C3&")"
E3   =CONCAT(A3," ",B3)
F3   =TEXTJOIN(", ",TRUE,A3:C3)
```

- & is the quickest way to join a few pieces.
- `CONCAT` joins ranges as well as single cells (it replaces the older `CONCATENATE`).
- `TEXTJOIN(delimiter, ignore_empty, …)` puts a separator between items and can **skip blanks**, which is why row 3 shows "Imran, Delhi" without a double comma.

Numbers and dates inside text: TEXT

Joining a number or date directly shows its raw value: `="Sales: "&A2` gives `Sales: 1234567.891`, and a date shows as `46139`. Use `TEXT(value, format)` to control how it appears.

TEXT() inside labels

	A	B
1	**Value**	**Label formula result**
2	1,234,568	Revenue: ₹1,234,568
3	27-04-2026	Report date: 27 Apr 2026
4	0.1834	Growth: 18.3%
5	27-04-2026	Day: Monday

```
B2   ="Revenue: ₹"&TEXT(A2,"#,##0")
B3   ="Report date: "&TEXT(A3,"dd mmm yyyy")
B4   ="Growth: "&TEXT(A4,"0.0%")
B5   ="Day: "&TEXT(A5,"dddd")
```

Splitting text

Text to Columns (Lesson 2) splits once into static values. Microsoft 365 adds formula versions that update automatically:

Function	Example	Result
`TEXTBEFORE(text, delimiter, [instance])`	`=TEXTBEFORE("kavya.iyer@abc.in","@")`	kavya.iyer
`TEXTAFTER(text, delimiter, [instance])`	`=TEXTAFTER("IN-2025-00042","-",-1)`	00042 (−1 = last hyphen)
`TEXTSPLIT(text, col_delim, [row_delim])`	`=TEXTSPLIT("North,South,East",",")`	spills North / South / East across three cells

TEXTSPLIT spills into neighbouring cells (Microsoft 365)

	A	B	C	D	E
1	**Input**				
2	IN-2025-00042	IN			
3	Furniture	Chairs	Apex Chair 412		
4	Furniture > Chairs > Apex Chair 412				

```
B2   =TEXTSPLIT(A2,"-")
A4   =TEXTJOIN(" > ",TRUE,A3:C3)
```

The blue cells are a **spill range**: one formula fills several cells. If any of those cells already contain something, the formula shows `#SPILL!` (Lesson 6).

2026 UPDATE **Regular expressions in Excel (2025)**

Microsoft 365 has added `REGEXTEST`, `REGEXEXTRACT` and `REGEXREPLACE`. For example `=REGEXEXTRACT(A2,"\d{6}")` pulls a 6-digit PIN code out of an address, and `=REGEXTEST(A2,"^[\w.]+@[\w.]+\.\w+$")` checks that an email looks valid. They are covered with Python regex in the Python track.

Logical functions: TRUE or FALSE

Comparisons such as `=B2>100` return TRUE or FALSE. Logical functions combine them:

Function	Returns TRUE when	Example
`AND(c1, c2, …)`	**All** conditions are TRUE	`=AND(B2⩾4, C2="Sales")`
`OR(c1, c2, …)`	**At least one** condition is TRUE	`=OR(D2="Delhi", D2="Mumbai")`
`NOT(c)`	The condition is FALSE	`=NOT(E2="Done")`
`XOR(c1, c2)`	An odd number of conditions are TRUE	Rarely needed

AND, OR and NOT on employee data

	A	B	C	D	E	F	G	H
1	**Name**	**Department**	**City**	**Rating**	**Salary**	**Sales & rating≥4**	**Metro city**	**Not HR**
2	Kavya Iyer	HR	Mumbai	4	97000	FALSE	TRUE	FALSE
3	Isha Mehta	Sales	Delhi	3	57000	FALSE	TRUE	TRUE
4	Aadhya Nair	Sales	Bengaluru	5	45000	TRUE	FALSE	TRUE
5	Krishna Khan	Sales	Mumbai	3	156000	FALSE	TRUE	TRUE

```
F2   =AND(B2="Sales",D2⩾4)
G2   =OR(C2="Delhi",C2="Mumbai")
H2   =NOT(B2="HR")
```

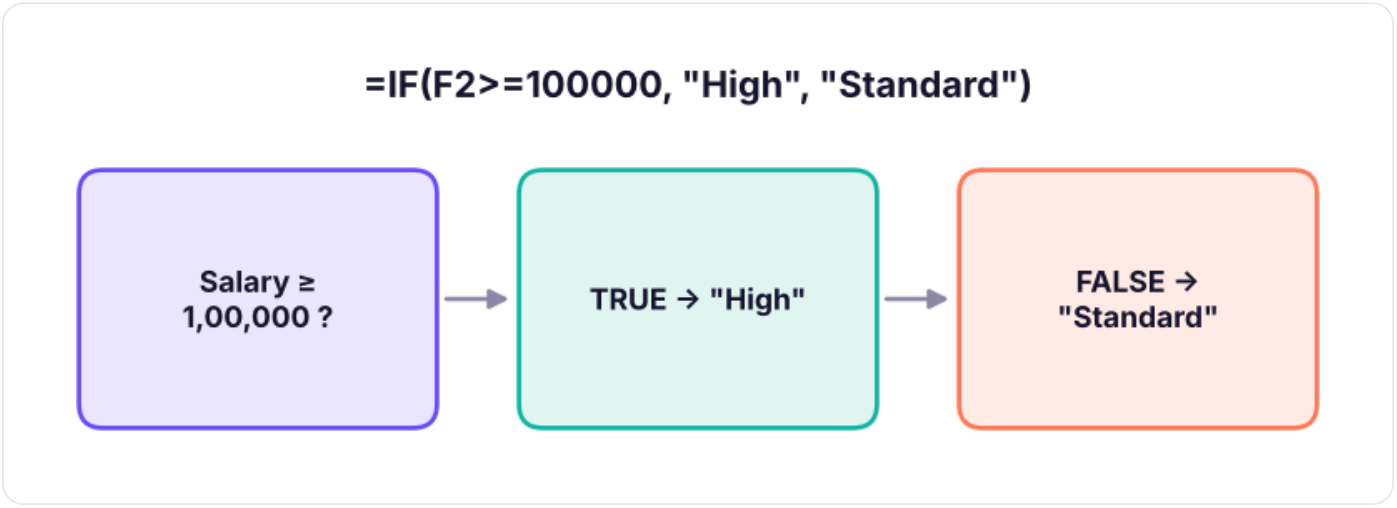
Each pattern is copied down/across to the other cells in its column.

TIP **TRUE = 1, FALSE = 0**

In arithmetic Excel treats TRUE as 1 and FALSE as 0. `=SUM(--(D2:D21≥4))` counts ratings of 4 or more (the double minus converts TRUE/FALSE to 1/0). This trick powers `SUMPRODUCT` in Lesson 6.

IF: making decisions

`=IF(logical_test, value_if_true, value_if_false)`



IF checks a condition and returns one of two results.

IF in practice

	A	B	C	D	E	F
1	Name	Salary	Rating	Band	Bonus (10% if rating ≥ 4)	Pass/Fail target 80k
2	Kavya Iyer	97000	4	Standard	9,700	Pass
3	Imran Reddy	157000	3	High	0	Pass
4	Isha Mehta	57000	3	Standard	0	Fail
5	Vikram Joshi	75000	5	Standard	7,500	Fail

```
D2   =IF(B2≥100000,"High","Standard")
E2   =IF(C2≥4,B2*10%,0)
F2   =IF(B2≥80000,"Pass","Fail")
```

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

IF can return text, numbers, another formula's result, or "" (an empty-looking cell).

Combining IF with AND and OR

IF + AND / OR

	A	B	C	D	E	F
1	Name	Dept	Rating	Years	Promotion: rating ≥4 AND years ≥3	Review: rating ≤2 OR years <1
2	Kavya Iyer	HR	4	10	Eligible	OK
3	Saanvi Rao	HR	4	1	Not yet	OK
4	Aarav Kulkarni	HR	2	1	Not yet	Review
5	Neha Menon	Sales	3	0	Not yet	Review

```
E2   =IF(AND(C2≥4,D2≥3),"Eligible","Not yet")
F2   =IF(OR(C2≤2,D2<1),"Review","OK")
```

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

Nested IF vs IFS

For more than two outcomes, you can put an IF inside another IF (**nested IF**), or use **IFS**, which is easier to read.

Grading marks: nested IF and IFS give the same result

	A	B	C
1	Marks	Nested IF	IFS
2	92	A	A
3	78	B	B
4	60	C	C
5	41	Fail	Fail

```
B2   =IF(A2≥90,"A",IF(A2≥75,"B",IF(A2≥60,"C","Fail")))
C2   =IFS(A2≥90,"A",A2≥75,"B",A2≥60,"C",TRUE,"Fail")
```

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

COMMON MISTAKES **Order matters in nested IF and IFS**

Excel stops at the **first** TRUE condition. If you test `A2≥60` first, a mark of 92 returns "C". Always test from the highest threshold to the lowest (or lowest to highest with <). In IFS, end with `TRUE, "default"` so values that match nothing do not return `#N/A`.

SWITCH: matching exact values

When you compare one value against a list of exact matches, `SWITCH` is cleaner than nested IFs.

SWITCH maps codes to names

	A	B
1	Region code	Region name
2	N	North
3	W	West
4	X	Unknown

```
B2   =SWITCH(A2,"N","North","S","South","E","East","W","West","Unknown")
B3   =SWITCH(A3,"N","North","S","South","E","East","W","West","Unknown")
B4   =SWITCH(A4,"N","North","S","South","E","East","W","West","Unknown")
```

For long mappings, a lookup table with XLOOKUP (Lesson 7) is better than a long formula, because the mapping can be edited without touching formulas.

Handling errors: IFERROR and IFNA

IFERROR replaces any error with a value you choose

	A	B	C	D
1	Sales	Orders	Average order	With IFERROR
2	120000	40	3,000	3,000
3	0	0	#DIV/0!	0
4	95000	25	3,800	3,800

```
C2   =A2/B2
D2   =IFERROR(A2/B2,0)
C3   =A3/B3
D3   =IFERROR(A3/B3,0)
C4   =A4/B4
D4   =IFERROR(A4/B4,0)
```

- `IFERROR(value, value_if_error)` catches **every** error type.
- `IFNA(value, value_if_na)` catches only `#N/A`, typically from lookups, so real problems such as `#REF!` stay visible.

> **WATCH OUT** **Do not hide problems**
>
> Wrapping every formula in IFERROR can hide genuine mistakes (a deleted column, a typo). Use it where an error is expected and meaningful (division by zero, value not found), and prefer IFNA for lookups.

Case: building columns for the Superstore data

Here is how analysts typically enrich the Superstore order table with text and logical formulas before analysis. The same logic in pandas gives us the real results on all 6,323 rows:

Python

```
import pandas as pd
s = pd.read_csv("superstore_sales.csv")
s["order_year"] = s["order_id"].str[3:7].astype(int)            # =VALUE(MID(A2,4,4))
s["product_brand"] = s["product_name"].str.split(" ").str[0]     # =TEXTBEFORE(N2," ")
s["profit_flag"] = s["profit"].apply(lambda p: "Loss" if p < 0 else "Profit")       #
=IF(R2<0,"Loss","Profit")
s["discount_band"] = pd.cut(s["discount"], [-0.01, 0, 0.2, 1], labels=["No discount", "Up to 20%",
"Above 20%"])  # =IFS(...)
s[["order_id", "order_year", "product_name", "product_brand", "discount", "discount_band", "profit",
"profit_flag"]].head(6)
```

OUTPUT

	order_id	order_year	product_name	product_brand	discount	discount_band	profit	pr
0	IN-2023-00001	2023	Nova Phone 223	Nova	0.0	No discount	3,472.55	Pr
1	IN-2023-00001	2023	Orbit Appliance 798	Orbit	0.0	No discount	1,169.54	Pr
2	IN-2023-00001	2023	Orbit Table 248	Orbit	0.0	No discount	6,044.47	Pr
3	IN-2023-00002	2023	Crest Storage 946	Crest	0.2	Up to 20%	450.16	Pr
4	IN-2023-00003	2023	Crest Bookcase 170	Crest	0.3	Above 20%	-5,512.33	Lo
5	IN-2023-00003	2023	Zenith Table 111	Zenith	0.3	Above 20%	-1,598.43	Lo

6 rows × 8 columns

Python

```
pd.crosstab(s["discount_band"], s["profit_flag"], margins=True)
```

OUTPUT

profit_flag	Loss	Profit	All
discount_band			
No discount	8	3472	3480
Up to 20%	776	1167	1943
Above 20%	898	2	900
All	1682	4641	6323

4 rows × 3 columns

The crosstab already tells a business story: almost every order-line with a discount above 20% makes a loss. You will build this same view as a pivot table in Lesson 8.

New column	Excel formula (row 2)
Order year	`=VALUE(MID(B2,4,4))`
Brand	`=TEXTBEFORE(N2," ")` or `=LEFT(N2,FIND(" ",N2)-1)`
Profit flag	`=IF(S2<0,"Loss","Profit")`
Discount band	`=IFS(Q2=0,"No discount",Q2⩽0.2,"Up to 20%",TRUE,"Above 20%")`

DO IT WITH AI

Do it with AI

Prompt: *"Column B has order IDs like IN-2025-00042. Write an Excel formula to extract the year as a number, and another to flag orders where discount (Q) is above 20% and profit (S) is negative as 'Review'. Explain each part."* Expected answer: `=VALUE(MID(B2,4,4))` and `=IF(AND(Q2>0.2,S2<0),"Review","")`.

Summary

KEY TAKEAWAYS

- `LEFT`, `RIGHT`, `MID` extract text; `FIND`/`SEARCH` locate a character so you can cut at a delimiter.
- Text functions return text: wrap with `VALUE()` to get numbers.
- Join with &, `CONCAT` or `TEXTJOIN` (which can skip blanks); format numbers inside labels with `TEXT`.
- `TEXTSPLIT`, `TEXTBEFORE` and `TEXTAFTER` split text with formulas that update.
- `AND` needs all conditions, `OR` needs any; `TRUE` counts as 1 in arithmetic.
- Use `IFS` (with a final `TRUE`) instead of deep nested IFs; order conditions from strictest to loosest.
- Use `IFERROR` for expected errors and `IFNA` for lookups.

Quick quiz

6 questions · tap an answer to check it

1. What does `=MID("SKU-7781-BLU",5,4)` return?

- **A** SKU-
- **B** 7781
- **C** 781-
- **D** BLU

Answer: B. MID starts at position 5 ("7") and takes 4 characters: 7781.

2. A2 contains `ravi.verma@xyz.com`. Which formula returns `ravi.verma`?

- **A** `=LEFT(A2,FIND("@",A2)-1)`
- **B** `=LEFT(A2,FIND("@",A2))`
- **C** `=RIGHT(A2,FIND("@",A2))`
- **D** `=MID(A2,1,FIND(".",A2))`

Answer: A. FIND returns 11 (the @ position); take 10 characters from the left.

3. Which function joins A2:D2 with ", " and skips empty cells?

- **A** `=CONCAT(A2:D2)`
- **B** `=TEXTJOIN(", ",TRUE,A2:D2)`
- **C** `=A2&", "&B2&", "&C2&", "&D2`
- **D** `=TEXT(A2:D2,", ")`

Answer: B. TEXTJOIN's second argument TRUE ignores empty cells.

4. `=IF(A2≥60,"C",IF(A2≥75,"B","A"))` is used to grade a mark of 85. What is the result?

A C

B B

C A

D **VALUE!**

Answer: A. The first test (85 >= 60) is TRUE, so "C" is returned immediately. The conditions are in the wrong order.

5. Why end an IFS formula with `TRUE,"Other"`?

A It makes IFS case-insensitive

B It returns a default value instead of #N/A when no condition matches

C It is required syntax

D It speeds up the calculation

Answer: B. TRUE always matches, so it acts as the "else" branch.

6. What does `="Total: "&TEXT(1234.5,"#,##0")` return?

A Total: 1234.5

B Total: 1,235

C Total: 1,234

D Total: #,##0

Answer: B. TEXT formats the number with separators and no decimals, rounding 1234.5 to 1,235.

EXERCISE **Product codes**

Codes look like `FUR-CH-10001234` (category-subcategory-number). Write formulas for: (a) the category code; (b) the sub-category code; (c) the number as a value; (d) a flag "Furniture" when the category is FUR, otherwise "Other".

Show solution (worked answer)

(a) `=LEFT(A2,3)` or `=TEXTBEFORE(A2,"-")` (b) `=MID(A2,5,2)` or `=TEXTBEFORE(TEXTAFTER(A2,"-"),"-")` (c) `=VALUE(RIGHT(A2,8))` or `=VALUE(TEXTAFTER(A2,"-",-1))` (d) `=IF(LEFT(A2,3)="FUR","Furniture","Other")`

EXERCISE **Commission rules**

Sales reps get: 10% commission if sales ≥ 5,00,000; 7% if ≥ 3,00,000; 4% if ≥ 1,00,000; otherwise 0. Reps in the "Trainee" grade get half. Write one formula (sales in B2, grade in C2).

Show solution (worked answer)

`=B2*IFS(B2≥500000,10%,B2≥300000,7%,B2≥100000,4%,TRUE,0)*IF(C2="Trainee",0.5,1)`

INTERVIEW CORNER **Questions you may be asked**

Q: Difference between FIND and SEARCH? FIND is case-sensitive and does not allow wildcards; SEARCH is not case-sensitive and supports `*` and `?`. Both return `#VALUE!` when the text is not found.

Q: How would you extract the domain from an email address? `=MID(A2,FIND("@",A2)+1,LEN(A2))` or, in Microsoft 365, `=TEXTAFTER(A2,"@")`.

Q: Nested IF or IFS? IFS for readability when testing several conditions in order; SWITCH for exact matches of one value; a lookup table when the mapping is long or changes often.

ADVANCED EXCEL · LESSON 6

Sorting, Filtering & Dynamic Arrays

Organise data with single and multi-level sorts and custom orders, find what matters with AutoFilter and Advanced Filter, and then let formulas do it for you with the dynamic array functions UNIQUE, SORT, SORTBY, FILTER and SEQUENCE, plus the power tool SUMPRODUCT.

Live class 4 · Excel for Microsoft 365, 2026 · Beginner–Intermediate · 110 min

By the end of this lesson you can

- Sort by one or many levels, by custom lists, by colour and left to right
- Filter text, numbers and dates, and use Advanced Filter with AND/OR criteria ranges
- Explain spill ranges, the
- Build live lists with UNIQUE, SORT, SORTBY, FILTER and SEQUENCE
- Answer multi-condition questions with SUMPRODUCT

The sample table

We will use this small sales table (sheet `Reps`, range `A1:F9`) so every result is easy to check by hand. Revenue is `=D2*E2`.

Reps: sample sales table

	A	B	C	D	E	F
1	Rep	Region	Product	Units	Price	Revenue
2	Asha	North	Laptop	5	55,000	275,000
3	Ravi	West	Phone	12	18,000	216,000
4	Meera	South	Laptop	3	55,000	165,000
5	Kabir	North	Tablet	8	25,000	200,000
6	Zoya	East	Phone	15	18,000	270,000
7	Farhan	West	Laptop	4	55,000	220,000
8	Neha	South	Tablet	10	25,000	250,000
9	Rohan	North	Phone	9	18,000	162,000

F2 `=D2*E2`

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

Sorting

Quick sort

Click any cell in a column and use **Data → Sort A to Z** or **Sort Z to A** (or Home → Sort & Filter). Excel detects the table around the cell and keeps each row together.

> **WATCH OUT** **Never sort a single selected column**
>
> If you select only one column and sort, Excel asks whether to "expand the selection". Choosing "Continue with the current selection" sorts that column alone and scrambles your data: names no longer match their revenue. Always click a single cell inside the table, or select the whole table.

Multi-level (level-wise) sort

Data → Sort opens the Sort dialog, where you add levels. Excel sorts by the first level, then breaks ties with the next.

The Sort dialog with two levels: Region A to Z, then Revenue largest to smallest.

Result: Region A→Z, then Revenue high→low

	A	B	C	D
1	**Rep**	**Region**	**Product**	**Revenue**
2	Zoya	East	Phone	270,000
3	Asha	North	Laptop	275,000
4	Kabir	North	Tablet	200,000
5	Rohan	North	Phone	162,000
6	Neha	South	Tablet	250,000
7	Meera	South	Laptop	165,000
8	Farhan	West	Laptop	220,000
9	Ravi	West	Phone	216,000

Other sort options

Option	Where	Example
Custom list	Order → Custom List	Sort months Jan, Feb, Mar instead of alphabetically; sort sizes S, M, L, XL; sort priority High, Medium, Low
Sort by colour	Sort On → Cell Color / Font Color / Conditional Formatting Icon	Put all red-flagged rows on top
Left to right	Options → Sort left to right	Reorder columns by a header row
Case sensitive	Options	Rarely needed
Random order	Add a column `=RAND()` and sort by it	Draw a random sample

> **TIP** **Keep the original order**
>
> Before sorting raw data, add an `Index` column with 1, 2, 3 ... (type 1 and 2, double-click the fill handle). You can always sort by Index to get back to the original order.

Filtering

AutoFilter

Click inside the data and press `Ctrl + Shift + L` (Data → Filter). Every header gets a drop-down arrow.

Filter type	Options
Text filters	Equals, does not equal, begins with, ends with, contains, does not contain, custom (with `*` and `?`)
Number filters	Greater than, between, Top 10 (items or percent), above/below average
Date filters	Today, this week, last month, next quarter, year to date, between, and grouped tick-boxes by year → month → day
By colour	Filter by cell colour, font colour or icon
Search box	Type part of a value to tick only matching items

Filters on several columns combine with **AND**: Region = North **and** Product = Laptop. Within one column, ticking several items is **OR**: North **or** South.

> **TIP** **Filtered totals**
>
> When you filter, the status bar shows "8 of 6,323 records found". Use `SUBTOTAL(109, …)` (Lesson 4) or the Table Total Row for totals that follow the filter. To copy only the visible rows, select them, press `Alt + ;` (visible cells only), then copy.

Here is the same idea on the full Superstore file: filter to Technology orders in the West with a discount above 20%.

Python

```
import pandas as pd
s = pd.read_csv("superstore_sales.csv")
f = s[(s.category == "Technology") & (s.region == "West") & (s.discount > 0.2)]
print(f"{len(f)} of {len(s):,} records found  |  visible sales total: {f.sales.sum():,.0f}  |  visible
profit total: {f.profit.sum():,.0f}")
f[["order_id", "city", "sub_category", "sales", "discount", "profit"]].head(5)
```

OUTPUT

```
36 of 6,323 records found  |  visible sales total: 1,725,761  |  visible profit total: -323,079
```

	order_id	city	sub_category	sales	discount	profit
14	IN-2023-00007	Mumbai	Phones	23,904	0.4	-4,153.37
37	IN-2023-00021	Ahmedabad	Accessories	2,821	0.3	-281.53
76	IN-2023-00037	Ahmedabad	Accessories	4,668	0.4	-1,569.18
77	IN-2023-00038	Ahmedabad	Copiers	53,082	0.4	-11,596.45
134	IN-2023-00066	Pune	Machines	22,155	0.3	-4,687.77

5 rows × 6 columns

Advanced Filter: complex AND/OR in one go

Data → Advanced filters with a **criteria range**: a small table with the same headers as your data.

- Conditions on the **same row** are combined with **AND**.
- Conditions on **different rows** are combined with **OR**.

X Criteria range: (North AND Revenue > 1,80,000) OR (Product = Phone)

	A	B	C
1	**Region**	**Product**	**Revenue**
2	North		>180000
3		Phone	

Rows that match: Asha (North, 2,75,000), Kabir (North, 2,00,000), and every Phone row: Ravi, Zoya and Rohan. Rohan is in the North but has only 1,62,000; he still appears because he sells phones.

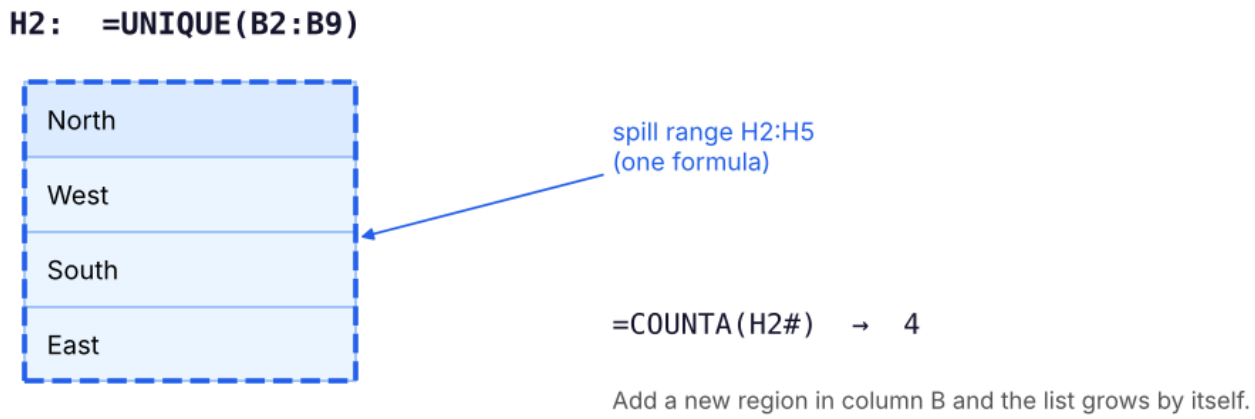
STEP BY STEP **Run an Advanced Filter**

1. Type the criteria range above your data or on another sheet (headers must match exactly).
2. Click inside the data → Data → **Advanced**.
3. List range: your data. Criteria range: the criteria table including headers.
4. Choose **Copy to another location** and pick an output cell to keep the original intact.
5. Tick **Unique records only** to get a list without duplicates (a quick alternative to Remove Duplicates).

Advanced Filter results are static: run it again after the data changes. For live results use `FILTER`.

Dynamic arrays: formulas that spill

In Microsoft 365, a formula can return **many values**. They **spill** into the cells below and to the right. The whole result is called the **spill range**; it has a blue border, and you can refer to it with the # operator.



One formula in H2 spills into H2:H5. Refer to the whole result with H2#.

WATCH OUT **#SPILL!**

If any cell in the area the result needs is not empty, the formula returns `#SPILL!`. Click the error to see which cell is blocking, and clear it. Spilling formulas also cannot be used inside an Excel Table; put them next to the table instead.

UNIQUE

```
=UNIQUE(array, [by_col], [exactly_once])
```

UNIQUE examples (formulas in H2 and J2)

	A	B	C	D	E	F
1				Region list =UNIQUE(B2:B9)		Product list =UNIQUE(C2:C9)
2				North		Laptop
3				West		Phone
4				South		Tablet
5				East		

```
D2   =UNIQUE(Reps!B2:B9)
F2   =UNIQUE(Reps!C2:C9)
```

- `=COUNTA(UNIQUE(C2:C9))` → **3** distinct products. (Before dynamic arrays this needed `=SUMPRODUCT(1/COUNTIF(C2:C9,C2:C9))`.)
- `=UNIQUE(B2:C9)` returns unique **combinations** of Region and Product.
- `=UNIQUE(B2:B9,,TRUE)` returns values that appear **exactly once**: East.

SORT and SORTBY

`=SORT(array, [sort_index], [sort_order], [by_col])` sorts by a column **position** in the array. `-1` means descending. `=SORTBY(array, by_array1, [order1], …)` sorts by **other ranges**, which do not have to be shown.

Top performers: =SORT(A2:F9,6,-1) and =SORTBY(A2:A9,F2:F9,-1)

	A	B	C	D	E	F	G	H
1	Rep	Region	Product	Units	Price	Revenue		SORTBY names
2	Asha	North	Laptop	5	55000	275000		Asha
3	Zoya	East	Phone	15	18000	270000		Zoya
4	Neha	South	Tablet	10	25000	250000		Neha
5	Farhan	West	Laptop	4	55000	220000		Farhan
6	Ravi	West	Phone	12	18000	216000		Ravi
7	Kabir	North	Tablet	8	25000	200000		Kabir
8	Meera	South	Laptop	3	55000	165000		Meera
9	Rohan	North	Phone	9	18000	162000		Rohan

```
A2   =SORT(Reps!A2:F9,6,-1)
H2   =SORTBY(Reps!A2:A9,Reps!F2:F9,-1)
```

Combine functions: `=SORT(UNIQUE(B2:B9))` gives the regions alphabetically: East, North, South, West. `=TAKE(SORT(A2:F9,6,-1),3)` returns only the top 3 rows.

FILTER

`=FILTER(array, include, [if_empty])`

The `include` argument is a TRUE/FALSE test with one result per row. Combine conditions with `*` for AND and `+` for OR.

FILTER examples

	A	B	C	D	E	F	G
1	North reps		Laptops with ≥4 units		North OR Phone		No match
2	Asha		Asha		Asha		None found
3	Kabir		Farhan		Ravi		
4	Rohan				Kabir		
5					Zoya		
6					Rohan		

```
A2  =FILTER(Reps!A2:A9,Reps!B2:B9="North")
C2  =FILTER(Reps!A2:A9,(Reps!C2:C9="Laptop")*(Reps!D2:D9≥4))
E2  =FILTER(Reps!A2:A9,(Reps!B2:B9="North")+(Reps!C2:C9="Phone"))
G2  =FILTER(Reps!A2:A9,Reps!B2:B9="Central","None found")
```

Logic	Write it as	Why it works
AND	`(cond1)*(cond2)`	TRUE×TRUE = 1; anything × FALSE = 0
OR	`(cond1)+(cond2)`	Any TRUE makes the sum ≥ 1
NOT	`(cond)=FALSE` or `NOT(cond)`	Reverses the test

TIP **An interactive report in two cells**

Put a Data Validation drop-down of regions in `H1` (source `=SORT(UNIQUE(B2:B9))`), then `=FILTER(A2:F9,B2:B9=H1)` below it. Choosing a region instantly shows its reps. This is the formula version of a slicer.

SEQUENCE and friends

Function	Example	Result
`SEQUENCE(rows, [cols], [start], [step])`	`=SEQUENCE(5)`	1, 2, 3, 4, 5 down a column
	`=SEQUENCE(1,12,1,1)`	1 to 12 across a row
	`=SEQUENCE(7,1,DATE(2026,4,27),1)`	Seven consecutive dates
`RANDARRAY(rows, cols, min, max, integer)`	`=RANDARRAY(5,1,1,100,TRUE)`	Five random whole numbers
`TAKE(array, rows)` / `DROP(array, rows)`	`=TAKE(SORT(F2:F9,,-1),3)`	Top 3 revenues
`VSTACK` / `HSTACK`	`=VSTACK(Jan!A2:C50,Feb!A2:C50)`	Stack two ranges into one
`CHOOSECOLS(array, 1, 6)`		Only columns 1 and 6
`GROUPBY(row_fields, values, function)`	`=GROUPBY(B2:B9,F2:F9,SUM)`	A summary by region, like a pivot table, in one formula (newer Microsoft 365)

SUMPRODUCT: the original array function

`SUMPRODUCT` multiplies arrays element by element and adds the results. It works in **every** version of Excel and is the classic way to do multi-condition sums before SUMIFS existed, plus things SUMIFS cannot do.

SUMPRODUCT on the Reps table

	A	B	C
1	Question	Formula	Result
2	Total revenue (units × price)	=SUMPRODUCT(D2:D9,E2:E9)	1,758,000
3	Revenue in North	=SUMPRODUCT((B2:B9="North")*D2:D9*E2:E9)	637,000
4	Phone units in West or East	=SUMPRODUCT((C2:C9="Phone")*((B2:B9="West")+(B2:B9="East"))*D2:D9)	27
5	Rows where units > 8	=SUMPRODUCT(--(D2:D9>8))	4
6	Weighted average price	=SUMPRODUCT(D2:D9,E2:E9)/SUM(D2:D9)	26,636.36

```
C2  =SUMPRODUCT(Reps!D2:D9,Reps!E2:E9)
C3  =SUMPRODUCT((Reps!B2:B9="North")*Reps!D2:D9*Reps!E2:E9)
C4  =SUMPRODUCT((Reps!C2:C9="Phone")*((Reps!B2:B9="West")+(Reps!B2:B9="East"))*Reps!D2:D9)
C5  =SUMPRODUCT(--(Reps!D2:D9>8))
C6  =SUMPRODUCT(Reps!D2:D9,Reps!E2:E9)/SUM(Reps!D2:D9)
```

Check: Phone units in West or East = Ravi 12 + Zoya 15 = 27. Rows with more than 8 units: Ravi, Zoya, Neha, Rohan = 4.

NOTE **Why the double minus `--`?**

`(D2:D9>8)` produces TRUE/FALSE values. SUMPRODUCT ignores logical values unless they are converted to numbers. `--` (minus minus) turns TRUE into 1 and FALSE into 0. Multiplying by another array (`*`) also converts them, which is why the other formulas do not need it.

Putting it together: a self-updating region summary

Region summary with UNIQUE + SUMIFS + SORT

	A	B	C
1	Region =SORT(UNIQUE(B2:B9))	Revenue =SUMIFS(F2:F9,B2:B9,H2#)	Share
2	East	270,000	15.4%
3	North	637,000	36.2%
4	South	415,000	23.6%
5	West	436,000	24.8%

```
A2   =SORT(UNIQUE(Reps!B2:B9))
B2   =SUMIFS(Reps!F2:F9,Reps!B2:B9,"East")
C2   =B2/SUM(Reps!F2:F9)
```

In Microsoft 365 the second column is a single formula, `=SUMIFS(F2:F9,B2:B9,H2#)`: because `H2#` refers to the whole spilled list, SUMIFS spills one total per region. Add a new region to the data and the summary grows automatically.

Let's confirm the numbers with pandas:

Python

```
reps = pd.DataFrame({"rep": ["Asha","Ravi","Meera","Kabir","Zoya","Farhan","Neha","Rohan"],
                     "region": ["North","West","South","North","East","West","South","North"],
                     "units": [5,12,3,8,15,4,10,9], "price":
[55000,18000,55000,25000,18000,55000,25000,18000]})
reps["revenue"] = reps.units * reps.price
summary = reps.groupby("region")["revenue"].sum().to_frame()
summary["share"] = (summary.revenue / summary.revenue.sum()).round(3)
summary
```

OUTPUT

	revenue	share
region		
East	270000	0.15
North	637000	0.36
South	415000	0.24
West	436000	0.25

4 rows × 2 columns

Summary

KEY TAKEAWAYS

- Click **one cell** inside the table before sorting; use the Sort dialog for multi-level, custom-list and colour sorts.
- AutoFilter: AND across columns, OR within a column. Advanced Filter: AND on one row, OR across rows of a criteria range.
- Dynamic arrays **spill**; refer to the result with #; clear blocking cells to fix `#SPILL!`.
- `UNIQUE`, `SORT`, `SORTBY`, `FILTER`, `SEQUENCE`, `TAKE`, `VSTACK`, `GROUPBY` replace many manual steps with live formulas.
- In FILTER and SUMPRODUCT, * means AND and + means OR.
- `SUMPRODUCT` handles weighted sums and complex conditions in any Excel version.

Quick quiz

6 questions · tap an answer to check it

1. You select only the Revenue column and sort it, choosing "Continue with the current selection". What happens?

A The whole table is sorted correctly

B Only Revenue is reordered, so revenues no longer match their reps

C Excel refuses to sort

D The table is sorted by Rep name

Answer: B. Sorting a single column separates it from the rest of the row. Always sort the whole table.

2. In an Advanced Filter criteria range, two conditions on the same row mean:

- **A** AND
- **B** OR
- **C** NOT
- **D** XOR

Answer: A. Same row = AND; different rows = OR.

3. `=FILTER(A2:A9,(B2:B9="North")+(C2:C9="Phone"))` returns reps who are:

- **A** In the North and sell phones
- **B** In the North or sell phones (or both)
- **C** Not in the North
- **D** In the North but not selling phones

Answer: B. Adding conditions works as OR.

4. A formula returns #SPILL!. What is the most likely cause?

- **A** A typo in the function name
- **B** A cell in the spill area is not empty
- **C** The data has duplicates
- **D** The workbook is in manual calculation mode

Answer: B. Dynamic arrays need empty cells to spill into.

5. What does `=SUMPRODUCT(--(D2:D9>8))` calculate?

- **A** The sum of values greater than 8
- **B** The number of values greater than 8
- **C** The average of values greater than 8
- **D** The largest value

Answer: B. -- turns TRUE/FALSE into 1/0, and SUMPRODUCT adds them up: a count.

6. Which formula lists each region once, sorted A to Z?

- **A** `=UNIQUE(SORT(B2:B9),TRUE)`
- **B** `=SORT(UNIQUE(B2:B9))`
- **C** `=SORTBY(B2:B9,B2:B9)`
- **D** `=FILTER(B2:B9,B2:B9<>"")`

Answer: B. UNIQUE removes duplicates and SORT orders the result.

EXERCISE **Superstore with dynamic arrays**

In a copy of `superstore_sales.csv` (as a Table named S): (1) list the distinct sub-categories, sorted; (2) show all orders from Kochi with profit below zero; (3) show the 5 largest order lines by sales (order ID, city, sales); (4) count distinct customers.

Show solution (worked answer)

(1) `=SORT(UNIQUE(S[sub_category]))` (2) `=FILTER(S,(S[city]="Kochi")*(S[profit]<0),"None")` (3) `=TAKE(SORTBY(CHOOSECOLS(S,2,9,16),S[sales],-1),5)` (check the column numbers in your file: here 2 = order_id, 9 = city, 16 = sales) (4) `=COUNTA(UNIQUE(S[customer_id]))`

INTERVIEW CORNER

Questions you may be asked

Q: How would you get a list of unique values in Excel? `UNIQUE` in Microsoft 365; otherwise Remove Duplicates on a copy, Advanced Filter with "Unique records only", or a pivot table.

Q: Explain AND/OR logic in FILTER or SUMPRODUCT. Each condition becomes an array of TRUE/FALSE. Multiplying arrays gives 1 only when all are TRUE (AND); adding gives at least 1 when any is TRUE (OR).

Q: How do you sort by a custom order such as High, Medium, Low? Sort dialog → Order → Custom List, and add the list; or use `SORTBY` with `MATCH(range,{"High","Medium","Low"},0)` as the sort key.

ADVANCED EXCEL · LESSON 7

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

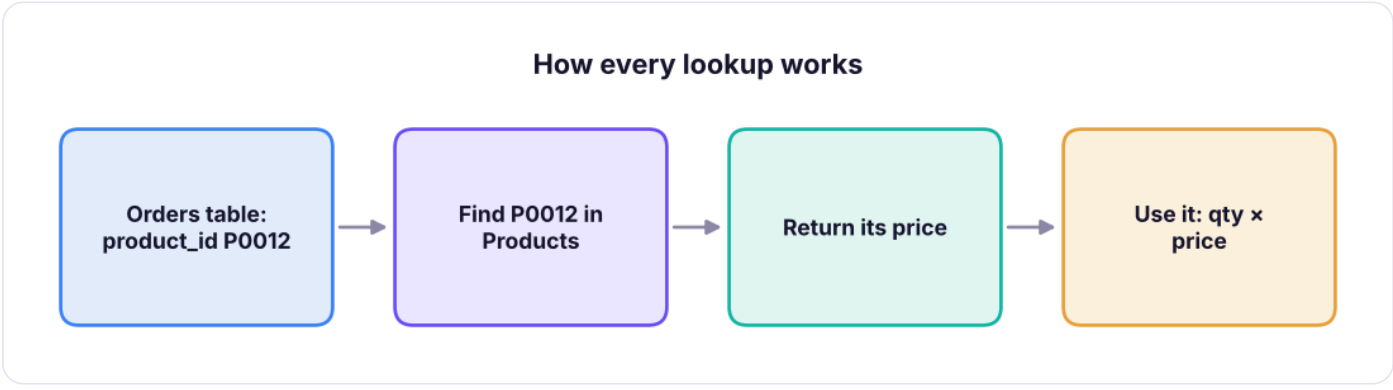
Live classes 4–5 · ✦ Excel for Microsoft 365, 2026 · Intermediate · ⏱ 120 min

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

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.



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`:

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.

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.

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).

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	`search_mode = -1`
Wildcards	Yes	`match_mode = 2`
Next smaller / larger match	Sorted data only	`match_mode = -1` or 1, any order

XLOOKUP examples (lookup table on sheet Prod)

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

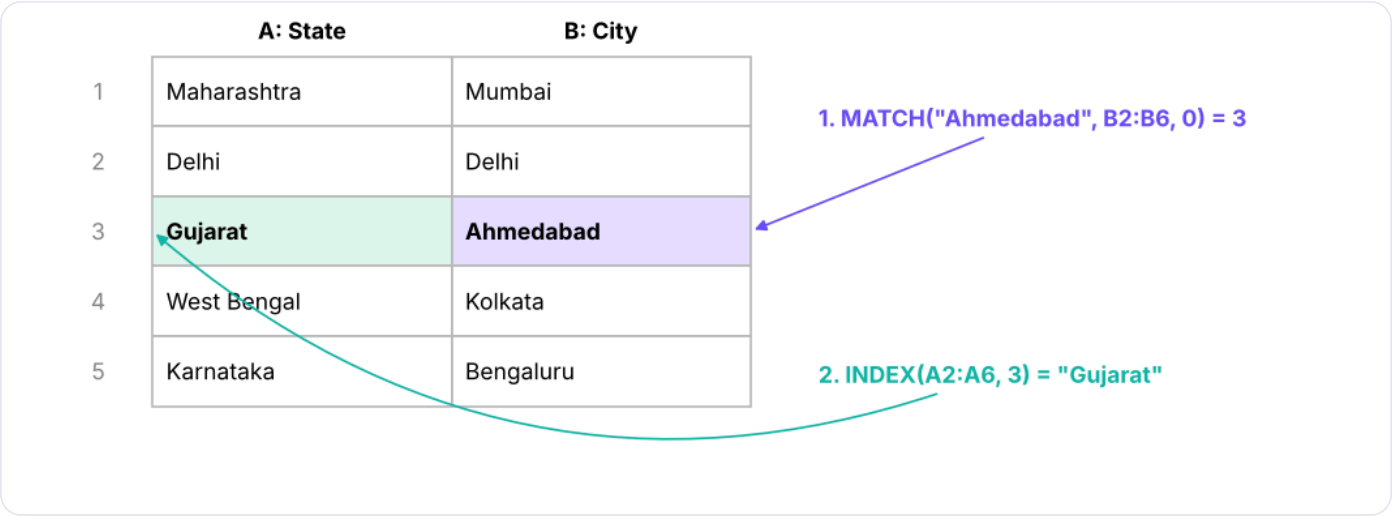
INDEX returns a value at a position; MATCH returns the position of a value

	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**.



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.

Two-way lookup: =INDEX($B$2:$E$6, MATCH($G3,$A$2:$A$6,0), MATCH(H$2,$B$1:$E$1,0))

	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).

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 *Top row* and *Left column*, optionally *Create links to source data*
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.

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.

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),4,4,4,1,1,1,2,2,2,3,3,3)	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: `IFNA(…,"Not found")` or XLOOKUP's 4th argument
Extra spaces	`=LEN(A2)` is longer than expected	`TRIM` both sides, or `XLOOKUP(TRIM(B2),TRIM(Prod!A2:A7),…)`
Number vs text (`101` vs `"101"`)	`=ISTEXT(B2)` differs from the lookup column	Convert one side: `VALUE()` or `TEXT(…,"0")`
Approximate match on unsorted data	Wrong but no error	Use exact match (FALSE / 0)
Range not locked	Errors appear only in lower rows	Add $ or use a Table
Invisible characters	`=CODE(RIGHT(B2))` 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

```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.

ADVANCED EXCEL · LESSON 8

Pivot Tables from Basic to Advanced

Summarise thousands of rows in seconds. Build pivot tables, understand the pivot cache and the four areas, change value field settings, show values as % of total, MoM and YoY change and running totals, group dates and numbers into bins, add calculated fields, slicers and timelines, and solve the Video Game Sales case study.

● Live classes 5–7 ✦ Excel for Microsoft 365, 2026 Intermediate ⏱ 150 min

By the end of this lesson you can

- Build a pivot table and explain the pivot cache, the four areas and Refresh vs Refresh All
- Change the summary function and number format, and show values as % of total, % of row and running totals
- Calculate month-on-month (MoM) and year-on-year (YoY) change
- Group dates by year, quarter and month, group items manually, and group numbers into bins
- Add calculated fields, slicers, timelines and pivot charts, and answer a full case study

What a pivot table does

A pivot table **groups** rows by one or more fields and **summarises** a number for each group (sum, count, average...). It is the fastest way to answer "how much, by what?" questions, with no formulas.

Raw data: 6,323 rows

order_id	region	category	sales
IN-2023-00001	South	Technology	36000.0
IN-2023-00001	South	Office Supplies	3850.0
IN-2023-00001	South	Furniture	62080.0
IN-2023-00002	West	Office Supplies	5216.0
IN-2023-00003	West	Furniture	24591.0
IN-2023-00003	West	Furniture	8288.0
IN-2023-00003	West	Office Supplies	2100.0
IN-2023-00003	West	Technology	69280.0
IN-2023-00004	East	Office Supplies	840.0

Pivot: sum of sales (₹ lakh)

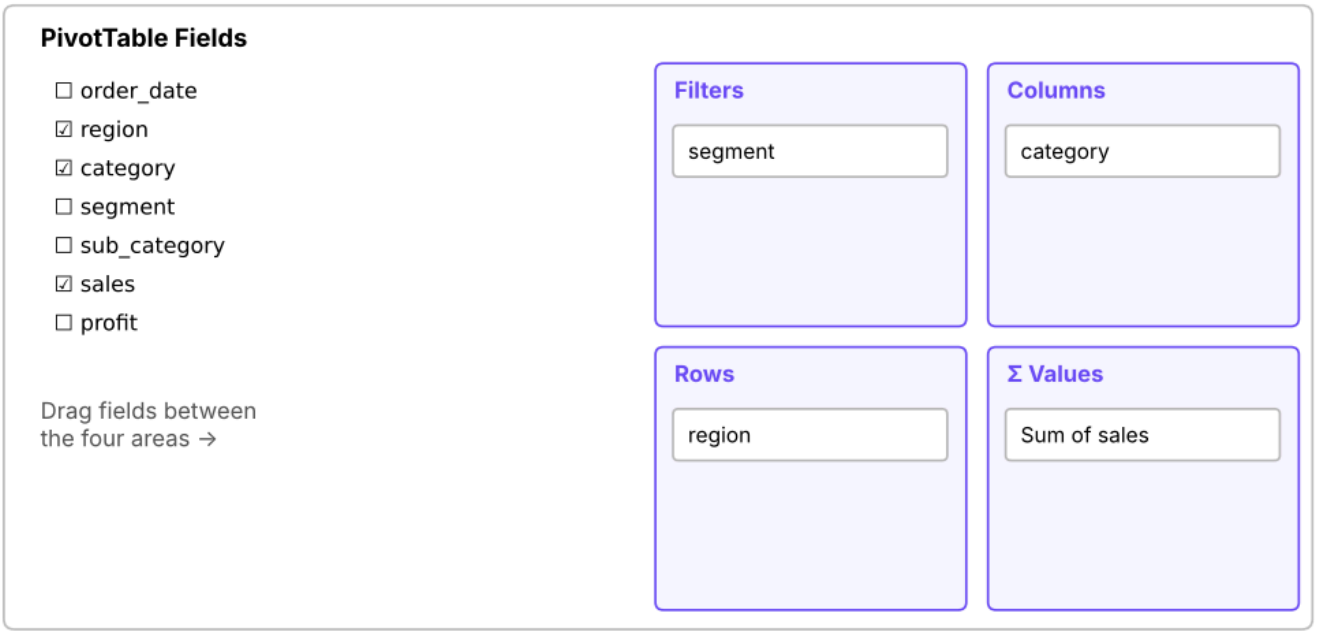
Region	Furniture	Office Supplies	Technology	Total
East	79.4	27.2	358.2	464.8
North	79.8	22.2	388.2	490.2
South	95.9	24.7	418.9	539.5
West	93.3	27.3	399.0	519.6
Grand Total	**348.4**	**101.3**	**1,564.4**	**2,014.1**

A pivot table turns thousands of order lines into a compact summary: rows grouped by Region, columns by Category, values summed.

Creating a pivot table

STEP BY STEP **Step by step**

1. Make sure the data is **structured** (Lesson 1): one header row, no blank rows, no merged cells. Ideally convert it to a Table (`Ctrl + T`) so new rows are included automatically.
2. Click any cell in the data → **Insert → PivotTable** (or `Alt, N, V`).
3. Choose **New Worksheet** → OK. An empty pivot appears with the **PivotTable Fields** pane on the right.
4. Drag fields into the four areas: drag `region` to **Rows**, `category` to **Columns**, `sales` to **Values**.
5. Right-click any number → **Number Format** → `#,##0` so the format sticks when the pivot refreshes.

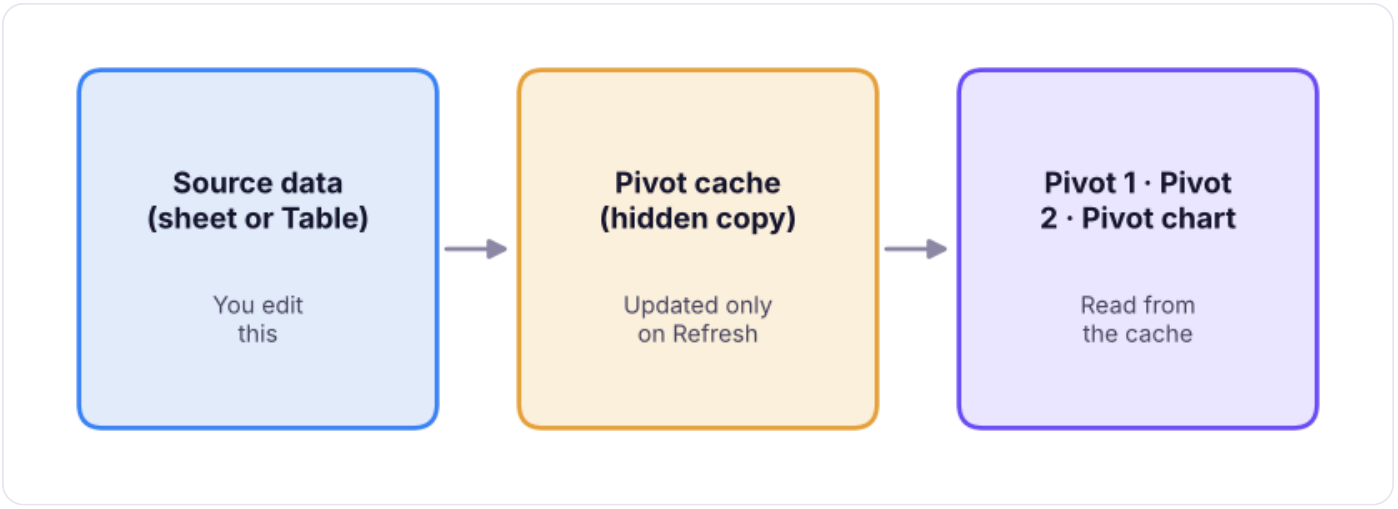


The PivotTable Fields pane: a list of fields at the top, four areas at the bottom.

Area	What it does	Example
Rows	One row per unique value	Region, Category, Order Date
Columns	One column per unique value	Year, Segment
Values	The numbers being summarised	Sum of Sales, Count of Order ID
Filters	Filter the whole pivot by a field	Segment = Corporate

The pivot cache, Refresh and Refresh All

A pivot table does not read your sheet directly. When you create it, Excel copies the data into a hidden **pivot cache**, and the pivot reads from that cache. This is why pivots are fast, and also why they do **not** update automatically when the data changes.



Every pivot reads from a cache. Pivots made from the same data share one cache.

Action	Shortcut	Effect
Refresh	`Alt + F5`	Re-reads the source into this pivot's cache. Every pivot that **shares the same cache** updates too
Refresh All	`Ctrl + Alt + F5`	Refreshes every pivot cache, query and data connection in the workbook
Change Data Source	PivotTable Analyze tab	Point the pivot to a bigger range (unnecessary if the source is a Table)
Refresh on open	PivotTable Options → Data	Refreshes automatically when the file opens

> **TIP** **Use a Table as the source**
>
> If the source is a Table, new rows are included the next time you refresh. If the source is a fixed range like `A1:R6324`, new rows below are ignored until you change the data source.

Value field settings

Click a field in Values → **Value Field Settings** (or right-click a number).

Summarize values by

Sum, Count, Average, Max, Min, Product, Count Numbers, StdDev, Var. Text fields in Values are counted automatically. **Distinct Count** appears only when the pivot is added to the **Data Model** (tick "Add this data to the Data Model" when creating it; Lesson 12).

Show values as

This tab turns a plain sum into an analytical view without writing formulas.

Show values as	Answers	Example
% of Grand Total	What share of the whole?	Furniture = 22% of all sales
% of Column Total	Share within each column	Within 2025, North = 31%
% of Row Total	Mix within each row	In the West, 46% of sales are Technology
% of Parent Row Total	Share within a parent group	Each sub-category's share of its category
Difference From (previous)	Change from the previous item	MoM or YoY change in ₹
% Difference From (previous)	Growth rate	MoM % or YoY %
Running Total In	Cumulative total	Year-to-date sales
Rank Largest to Smallest	Position	Top-selling region = 1
Index	Relative importance	Above 1 = over-represented

Here are three of these views on the Superstore data (computed with pandas to show the exact numbers a pivot would give):

Python

```python
import pandas as pd
s = pd.read_csv("superstore_sales.csv", parse_dates=["order_date"])
pv = s.pivot_table(index="region", columns="category", values="sales", aggfunc="sum")
pct_total = (pv / pv.values.sum() * 100).round(1)
pct_row = (pv.div(pv.sum(axis=1), axis=0) * 100).round(1)
print("Sum of sales (₹):"); print(pv.round(0).astype(int).to_string())
print("\n% of grand total:"); print(pct_total.to_string())
print("\n% of row total (category mix inside each region):"); print(pct_row.to_string())
```

OUTPUT

```
Sum of sales (₹):
category  Furniture  Office Supplies  Technology
region
East        7942092          2719513    35822514
North       7975868          2219542    38822750
South       9593302          2468888    41890579
West        9330590          2725360    39902570

% of grand total:
category  Furniture  Office Supplies  Technology
region
East            3.9              1.4        17.8
North           4.0              1.1        19.3
South           4.8              1.2        20.8
West            4.6              1.4        19.8

% of row total (category mix inside each region):
category  Furniture  Office Supplies  Technology
region
East           17.1              5.9        77.1
```

```
North          16.3            4.5         79.2
South          17.8            4.6         77.6
West           18.0            5.2         76.8
```

MoM and YoY

Month-on-month (MoM) compares each month with the previous month; **year-on-year (YoY)** compares a period with the same period last year (which removes seasonality, such as the festive Q4 peak).

STEP BY STEP **MoM % in a pivot**

1. Put `order_date` in Rows and group it by **Years** and **Months** (see Grouping below).
2. Add `sales` to Values **twice**.
3. On the second copy: Value Field Settings → Show Values As → **% Difference From** → Base field: `order_date` (Months) → Base item: **(previous)**.
4. Rename the field "MoM %".

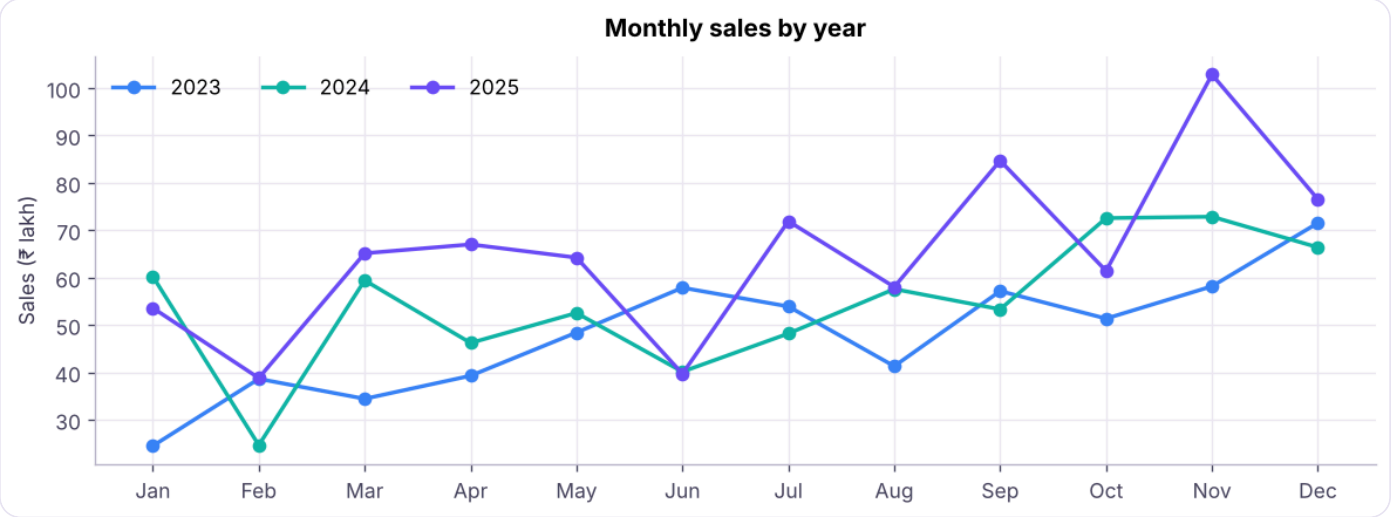
Python

```python
m = s.set_index("order_date").resample("MS")["sales"].sum().to_frame("sales")
m["MoM %"] = (m.sales.pct_change() * 100).round(1)
m["YoY %"] = (m.sales.pct_change(12) * 100).round(1)
m.index = m.index.strftime("%b-%Y")
m.tail(8)
```

OUTPUT

order_date	sales	MoM %	YoY %
May-2025	6,421,274	-4.2	22.2
Jun-2025	3,968,572	-38.2	-1.1
Jul-2025	7,190,087	81.2	49.2
Aug-2025	5,802,184	-19.3	0.8
Sep-2025	8,468,060	45.9	58.9
Oct-2025	6,140,290	-27.5	-15.4
Nov-2025	10,284,516	67.5	41.2
Dec-2025	7,652,266	-25.6	15.3

8 rows × 3 columns



Monthly sales with YoY growth. The festive season (Oct–Dec) peaks every year, so YoY is a fairer comparison than MoM.

In formulas, the same calculations are: MoM % `=(C3-C2)/C2`, YoY % `=(C14-C2)/C2` when months are listed one per row.

Grouping

Group dates

Right-click a date in Rows → **Group** → choose Years, Quarters, Months (and Days, Hours if needed). Excel 365 groups dates automatically when you drag a date field to Rows; you can ungroup with right-click → Ungroup.

> **TIP** **Indian financial year**
>
> Pivot grouping uses calendar quarters (Jan–Mar = Q1). For an April–March financial year, add helper columns to the data first: `FY = YEAR(date)+(MONTH(date)>=4)` and `FQ = "Q"&ROUNDUP(MOD(MONTH(date)-4,12)+1)/3,0)`, or use `CHOOSE` (Lesson 7), and put those in the pivot.

Group items manually

Select several labels (hold `Ctrl`) → right-click → **Group**. Excel creates a new field such as `Product2` with a group called `Group1`; rename it (for example "Low Sales Products"). This is how you combine the three lowest-selling products into one line, or merge small cities into "Others".

Groups vs bins

	Groups	Bins
What	Segregate items into chunks based on a common behaviour or criteria	Split a **numeric** field into equal-width intervals
Size	Can be any size	Every bin has the same width
Works on	Text, dates and numbers	Numbers
Example	"Metro" vs "Non-metro" cities	Order value 0–5,000, 5,000–10,000 …

To bin numbers: put the number field (for example `sales`) in **Rows** → right-click → Group → Starting at 0, Ending at 50,000, By 5,000. Then put `order_id` in Values as a Count.

Python

```
bins = pd.cut(s.sales, range(0, 55000, 5000), right=False)
dist = s.groupby(bins, observed=False).size().to_frame("Count of order lines")
dist.index = [f"{int(i.left):,}-{int(i.right - 1):,}" for i in dist.index]
dist
```

OUTPUT

	Count of order lines
0–4,999	2530
5,000–9,999	829
10,000–14,999	473
15,000–19,999	423
20,000–24,999	238
25,000–29,999	172
30,000–34,999	177
35,000–39,999	202
40,000–44,999	61
45,000–49,999	92

10 rows × 1 columns

Most order lines are small; a few very large ones (machines, copiers) form the long tail. This is a frequency distribution, the pivot-table version of a histogram (Lesson 10).

Layout, subtotals and grand totals

The **Design** tab controls how the pivot looks:

Setting	Options	Tip
Report Layout	Compact (default), Outline, **Tabular**	Tabular + **Repeat All Item Labels** gives a flat table you can copy and reuse
Subtotals	Don't show, at bottom, at top	Turn off for exports
Grand Totals	On for rows/columns, off	Off when sharing a percentage view
Blank Rows	Insert after each item	Easier to read printed reports
Expand / Collapse	+/- buttons, or right-click → Expand/Collapse entire field	Drill from Category down to Sub-category
Show Details	Double-click any number	Opens a new sheet with the rows behind that number: great for auditing

Parent total: when two fields are in Rows (Category → Sub-category), "% of Parent Row Total" shows each sub-category's share of its own category.

Sorting and filtering inside a pivot

- **Sort**: right-click a value → Sort → Largest to Smallest.
- **Label filter**: Row Labels drop-down → Label Filters → Begins with, contains ...
- **Value filter / Top 10**: Row Labels drop-down → Value Filters → **Top 10** → change to Top 5 Items by Sum of sales. This is how you answer "Find the top 5 categories".
- **Slicers**: PivotTable Analyze → **Insert Slicer** → pick fields (Region, Segment). Slicers are clickable buttons that filter the pivot and show what is selected. Right-click a slicer → **Report Connections** to control several pivots with one slicer (they must share the same cache).
- **Timeline**: Insert Timeline on a date field to filter by years, quarters, months or days with a slider.

Calculated fields

A **calculated field** creates a new value field from other fields in the pivot: PivotTable Analyze → Fields, Items & Sets → **Calculated Field**.

Example: `Profit Margin = profit / sales`, formatted as a percentage.

> **WATCH OUT** **Calculated fields work on totals**
>
> Excel calculates the field on the **sums**: Profit Margin for the North = total North profit ÷ total North sales. That is correct for ratios. But a calculated field like `=quantity * price` would multiply the *sum* of quantity by the *sum* of price, which is wrong. Add such row-level calculations as a new column in the source data instead.

Python

```python
pm = s.groupby("category")[["sales", "profit"]].sum()
pm["Profit Margin"] = (pm.profit / pm.sales * 100).round(1).astype(str) + "%"
pm.round(0)
```

OUTPUT

	sales	profit	Profit Margin
category			
Furniture	34,841,853	-818,153	-2.3%
Office Supplies	10,133,304	1,514,794	14.9%
Technology	156,438,412	17,655,524	11.3%

3 rows × 3 columns

GETPIVOTDATA and pivot charts

When you type = and click a cell inside a pivot, Excel writes `GETPIVOTDATA`, for example `=GETPIVOTDATA("sales",$A$3,"region","North")`. It keeps returning North's sales even if the pivot layout changes. Turn it off in PivotTable Analyze → Options → Generate GetPivotData if you prefer normal references.

PivotChart (PivotTable Analyze → PivotChart) creates a chart linked to the pivot: filtering the pivot or clicking a slicer updates the chart. This is the basis of the dashboard in Lesson 11.

Case study: Video Game Sales (regional sales trends)

You are an analyst at a game publisher. The file `video_game_sales.csv` has 2,400 fictional titles with platform, release year, genre, publisher and sales in millions of units by region: North America (`na_sales`), Europe (`eu_sales`), Japan (`jp_sales`), other regions and global. Answer the questions with pivot tables. The answers below were computed from the file so you can check your pivots.

Python

```
g = pd.read_csv("video_game_sales.csv")
print(g.shape); g.head(4)
```

OUTPUT

```
(2400, 11)
```

	rank	name	platform	year	genre	publisher	na_sales	eu_sales	jp_sales	other
0	1	Hidden Kingdom	XSX	2023	Sports	Red Lantern Studios	3.69	2.62	0.29	0.87
1	2	Galactic Kingdom	PS5	2020	Adventure	Saffron Works	3.49	1.79	0.74	0.94
2	3	Galactic League	Mobile	2017	Strategy	Red Lantern Studios	3.08	2.44	0.57	0.67
3	4	Shadow League	Mobile	2025	Role-Playing	Northwind Play	1.66	2.12	2.33	0.63

4 rows × 11 columns

CASE STUDY **Q1. Which genres sell the most worldwide?**

Pivot: Rows = genre, Values = Sum of global_sales, sorted largest to smallest.

Python

```
q1 = g.pivot_table(index="genre", values="global_sales", aggfunc="sum").sort_values("global_sales",
ascending=False)
q1["% of total"] = (q1.global_sales / q1.global_sales.sum() * 100).round(1)
q1.round(1).head(6)
```

OUTPUT

	global_sales	% of total
genre		
Adventure	136.7	10.6
Sports	133.8	10.4
Action	116.3	9.0
Racing	113.6	8.8
Strategy	111.5	8.6
Simulation	108.4	8.4

6 rows × 2 columns

CASE STUDY **Q2. Who are the top 5 publishers, and how many titles did each release?**

Pivot: Rows = publisher, Values = Sum of global_sales and Count of name; Value Filter → Top 5 by global sales.

Python

```
q2 = g.groupby("publisher").agg(global_sales=("global_sales", "sum"), titles=("name",
"count")).sort_values("global_sales", ascending=False).head(5)
q2["sales per title"] = (q2.global_sales / q2.titles).round(2)
q2.round(1)
```

OUTPUT

	global_sales	titles	sales per title
publisher			
Starforge Interactive	168.0	295	0.6
Blue Comet Games	161.4	316	0.5
Northwind Play	144.6	266	0.5
Red Lantern Studios	144.5	268	0.5
Pixel Harbor	143.1	267	0.5

5 rows × 3 columns

CASE STUDY **Q3. How does regional preference differ by genre?**

Pivot: Rows = genre, Values = Sum of na_sales, eu_sales, jp_sales, other_sales; Show Values As → **% of Row Total**.

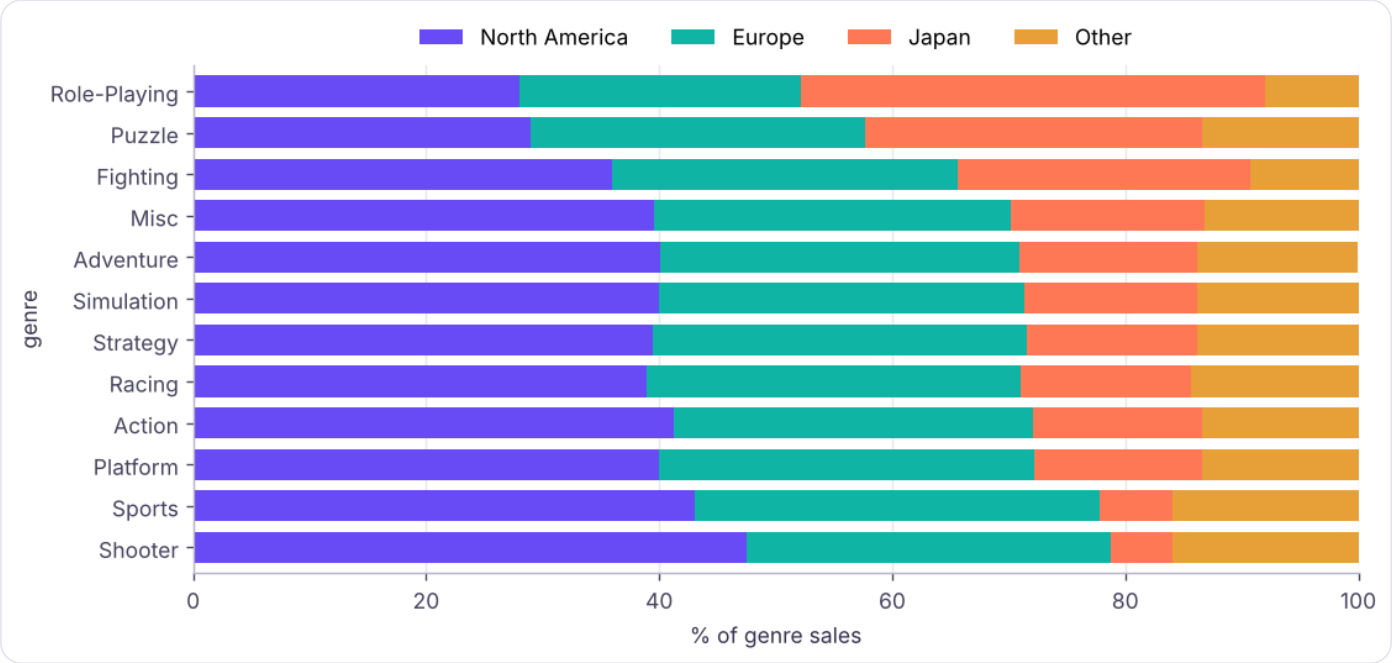
Python

```
reg = ["na_sales", "eu_sales", "jp_sales", "other_sales"]
q3 = g.groupby("genre")[reg].sum()
q3 = (q3.div(q3.sum(axis=1), axis=0) * 100).round(1).sort_values("jp_sales", ascending=False)
q3.head(6)
```

OUTPUT

	na_sales	eu_sales	jp_sales	other_sales
genre				
Role-Playing	28.0	24.2	39.8	8.1
Puzzle	29.0	28.7	28.9	13.4
Fighting	36.0	29.6	25.1	9.3
Misc	39.6	30.6	16.6	13.2
Adventure	40.1	30.8	15.3	13.7
Simulation	40.0	31.3	14.9	13.8

6 rows × 4 columns



Regional mix by genre (% of each genre's sales). Role-Playing and Puzzle games lean much more on Japan than Shooter or Sports titles do.

CASE STUDY **Q4. How have sales changed over the years?**

Pivot: Rows = year, Values = Sum of global_sales, plus a second copy shown as % Difference From (previous) for YoY growth. Insert a PivotChart (line).

Python

```
q4 = g.pivot_table(index="year", values="global_sales", aggfunc="sum")
q4["YoY %"] = (q4.global_sales.pct_change() * 100).round(1)
q4.round(1).tail(8)
```

OUTPUT

year	global_sales	YoY %
2018	87.2	19.0
2019	59.4	-31.9
2020	89.2	50.2
2021	78.0	-12.6
2022	71.9	-7.9
2023	76.4	6.3
2024	80.9	5.9
2025	87.0	7.4

8 rows × 2 columns

CASE STUDY **Q5. Which platform leads in each region?**

Pivot: Rows = platform, Values = the four regional sales columns; sort each column, or use conditional formatting → Top 1 per column.

Python

```
q5 = g.groupby("platform")[reg].sum().round(1)
q5.loc["Leader"] = q5.idxmax()
q5
```

OUTPUT

	na_sales	eu_sales	jp_sales	other_sales
platform				
3DS	45.90	40.40	37.60	17.20
Mobile	63.30	50	22.60	21.10
PC	43.40	34.20	14.90	14.90
PS3	48.10	38.50	18.80	17
PS4	55.40	43.10	16.90	19.20
PS5	58.50	45.40	20.80	19.70
Switch	40	34.40	38	13
X360	42.20	32.60	15.20	14.60
XOne	50.10	39.20	17.70	17.40
XSX	53.80	40.40	18	17.20
Leader	Mobile	Mobile	Switch	Mobile

11 rows × 4 columns

KEY TAKEAWAYS **Insights to present**

Write 3–5 sentences of insight, not just tables. For example: which genres to prioritise for the Japanese market, which publishers are efficient (high sales per title), and how platform transitions show in the year trend. Keep every number traceable to a pivot.

Summary

KEY TAKEAWAYS

- Pivot tables summarise structured data by dragging fields into **Rows, Columns, Values and Filters**.
- They read from a **pivot cache**: Refresh updates one cache; Refresh All updates everything. Use a Table as the source.
- Value Field Settings: change the function (Sum, Count, Average…) and **Show Values As** (% of total, % of row, difference from previous, running total, rank).
- MoM = change from the previous month; YoY = change from the same period last year.
- Group dates (years/quarters/months), group items manually, and group numbers into equal **bins**.
- Calculated fields work on totals, so use them for ratios like margin; add row-level maths to the source data.
- Slicers and timelines make pivots interactive; double-click a number to see the rows behind it.

Quick quiz

6 questions · tap an answer to check it

1. You add 500 rows to the source range, but the pivot does not change. Why?

A Pivot tables only read the first 1,000 rows

B The pivot reads from its cache, which updates only when you refresh (and the range must include the new rows)

C The new rows need a pivot formula

D Pivot tables cannot be updated

Answer: B. Refresh re-reads the source. If the source is a fixed range, also change the data source, or use a Table.

2. Which "Show Values As" option gives month-on-month growth?

A % of Grand Total

B Running Total In

C % Difference From → (previous)

D Rank Largest to Smallest

Answer: C. % Difference From the previous item compares each month with the one before it.

3. What is the main difference between groups and bins?

A Bins are equal-width numeric intervals; groups can be any size and any data type

B Groups only work with dates

C Bins only work with text

D There is no difference

Answer: A. Bins split numbers into equal intervals. Groups are custom collections of items.

4. A calculated field `Revenue = Qty * Price` gives inflated numbers. Why?

- **A** Calculated fields cannot multiply
- **B** It multiplies the sum of Qty by the sum of Price for each group
- **C** The pivot needs refreshing
- **D** Price must be in Rows

Answer: B. Calculated fields work on aggregated values. Compute Qty × Price in the source data instead.

5. How do you see the individual rows behind a number in a pivot table?

- **A** Press F9
- **B** Double-click the number (Show Details)
- **C** Use GETPIVOTDATA
- **D** Insert a slicer

Answer: B. Double-clicking opens a new sheet with the underlying records.

6. Two pivots were built from the same data. You click Refresh on one. What happens to the other?

- **A** It also updates, because they share the same pivot cache
- **B** Nothing; you must refresh it separately
- **C** It is deleted
- **D** It shows an error

Answer: A. Pivots from the same source share a cache; refreshing the cache updates all of them.

EXERCISE **Superstore pivots**

Build pivot tables from `superstore_sales.csv` to answer: (1) sales and profit by category and sub-category, with profit margin; (2) yearly sales by region with YoY %; (3) the 5 cities with the highest profit; (4) the number of order lines by discount (0%, 10%, ... 40%) and their total profit.

Show solution (worked answer)

(1) Rows: category, sub_category; Values: Sum of sales, Sum of profit; Calculated Field `Margin = profit/sales` formatted as %. (2) Rows: region; Columns: order_date grouped by Years; Values: Sum of sales, plus a second copy shown as % Difference From (previous) Years. (3) Rows: city; Values: Sum of profit; Value Filters → Top 10 → Top 5 Items by Sum of profit. (4) Rows: discount; Values: Count of order_id and Sum of profit. You will see profit turn negative at the higher discount levels, matching the Lesson 5 crosstab.

INTERVIEW CORNER **Questions you may be asked**

Q: What is a pivot cache and why does it matter? A hidden copy of the source data that pivots read from. It makes pivots fast and lets several pivots share data, but pivots only update on refresh, and many separate caches make files large.

Q: How do you show each item's share of the total in a pivot? Value Field Settings → Show Values As → % of Grand Total (or % of Column/Row Total).

Q: How would you count unique customers in a pivot? Create the pivot with "Add this data to the Data Model" ticked, then choose Distinct Count in Value Field Settings.

Q: When would you use formulas (SUMIFS) instead of a pivot? When the report needs a fixed layout, mixes data from several sources, or must update without refreshing; pivots are better for exploration and flexible summaries.

ADVANCED EXCEL · LESSON 9

Date, Time & Number Functions and Named Ranges

Work confidently with dates and times (DATE, EOMONTH, EDATE, WORKDAY, NETWORKDAYS, WEEKDAY, WEEKNUM, DATEDIF, fiscal quarters), clean up numbers with INT, MOD, ROUND, CEILING and FLOOR, and make formulas readable with static and dynamic named ranges and LET.

Live classes 7–8 | ✦ Excel for Microsoft 365, 2026 | Intermediate | ⏱ 120 min

By the end of this lesson you can

- Explain how Excel stores dates and times, and do date arithmetic correctly
- Extract parts of dates, find month ends, add months, and count working days with holidays
- Calculate age, tenure and fiscal-year quarters (Indian April–March and US October–September)
- Use INT, MOD, ROUND, ROUNDUP, ROUNDDOWN, MROUND, CEILING.MATH, FLOOR.MATH, SQRT, POWER and ABS
- Create static and dynamic named ranges and simplify long formulas with LET

Dates and times are numbers

Excel stores a date as the number of days since **1 January 1900** (that day is 1). A time is a **fraction of a day**: 12:00 noon is 0.5 and 06:00 is 0.25. A date with a time is simply a whole number plus a fraction.

What is really stored

	A	B	C
1	You see	Stored as	Explanation
2	27-04-2026	46139	Days since 1-Jan-1900
3	12:00	0.5	Half a day
4	06:30	0.2708	6.5 hours ÷ 24
5	27-04-2026 18:00	46139.75	Date + time

```
B2    =A2
B3    =TIME(12,0,0)
B4    =TIME(6,30,0)
B5    =DATE(2026,4,27)+TIME(18,0,0)
```

This is why date arithmetic works:

Date arithmetic

	A	B	C	D	E
1	Order date	Ship date	Days to ship =B2-A2	Due date (+30 days)	Overdue? (as of 15-Jun-2026)
2	27-04-2026	02-05-2026	5	27-05-2026	Overdue
3	20-05-2026	21-05-2026	1	19-06-2026	OK
4	01-06-2026	08-06-2026	7	01-07-2026	OK

C2 `=B2-A2`

D2 `=A2+30`

E2 `=IF(DATE(2026,6,15)>D2,"Overdue","OK")`

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

TIP **TODAY() and NOW()**

`=TODAY()` returns today's date and `=NOW()` the date and time. They are **volatile**: they recalculate every time the workbook changes, so results move every day. For reports that must not change, type the "as of" date into a cell and refer to it, as in the overdue example above. Shortcuts: `Ctrl + ;` types today's date as a fixed value.

Building and taking apart dates

Function	Returns	Example (A2 = 27-04-2026)	Result
`DATE(y, m, d)`	A date from parts	`=DATE(2026,4,27)`	27-04-2026
`YEAR, MONTH, DAY`	Parts of a date	`=MONTH(A2)`	4
`WEEKDAY(date, 2)`	1 = Monday ... 7 = Sunday	`=WEEKDAY(A2,2)`	1
`WEEKNUM(date, 2)`	Week of the year (weeks start Monday)	`=WEEKNUM(A2,2)`	18
`ISOWEEKNUM(date)`	ISO 8601 week number	`=ISOWEEKNUM(A2)`	18
`TEXT(date, "mmm")`	Month name as text	`=TEXT(A2,"mmmm")`	April
`EOMONTH(date, n)`	Last day of the month n months away	`=EOMONTH(A2,0)`	30-04-2026
`EDATE(date, n)`	Same day n months later	`=EDATE(A2,3)`	27-07-2026
`DATEVALUE(text)`	Converts date text to a real date	`=DATEVALUE("27-Apr-2026")`	46139

Date parts and month boundaries

	A	B	C	D	E	F	G	H
1	Date	Year	Month	Day	Weekday (Mon=1)	Month end	First of month	+3 months
2	27-04-2026	2026	4	27	1	30-04-2026	01-04-2026	27-07-2026
3	31-01-2026	2026	1	31	6	31-01-2026	01-01-2026	28-02-2026
4	29-02-2024	2024	2	29	4	29-02-2024	01-02-2024	28-02-2025

B2 `=YEAR(A2)`

```
C2   =MONTH(A2)
D2   =DAY(A2)
E2   =WEEKDAY(A2,2)
F2   =EOMONTH(A2,0)
G2   =EOMONTH(A2,-1)+1
H2   =EDATE(A2,3)
```

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

Notice that EDATE handles month ends correctly: 31-Jan + 1 month = 28-Feb (2026 is not a leap year), and 29-Feb-2024 + 12 months = 28-Feb-2025.

Working days

Function	Use
`NETWORKDAYS(start, end, [holidays])`	Count Mon–Fri days between two dates (inclusive) minus holidays
`NETWORKDAYS.INTL(start, end, weekend, [holidays])`	Choose weekend days: 11 = Sunday only, `"0000011"` = Sat + Sun
`WORKDAY(start, days, [holidays])`	The date after adding n working days
`WORKDAY.INTL(start, days, weekend, [holidays])`	Same with a custom weekend

SLA and delivery planning (holidays in G2:G4)

	A	B	C	D	E	F
1	Ticket	Opened	Closed	Working days =NETWORKDAYS(B2,C2,$G$2:$G$4)	Due (+5 working days)	
2	T-101	10-08-2026	18-08-2026	7	17-08-2026	
3	T-102	16-10-2026	23-10-2026	5	26-10-2026	
4	T-103	06-11-2026	10-11-2026	4	12-11-2026	

```
D2   =NETWORKDAYS(B2,C2,$G$2:$G$4)
E2   =WORKDAY(B2,5,$G$2:$G$4)
D3   =NETWORKDAYS(B3,C3,$G$2:$G$4)
E3   =WORKDAY(B3,5,$G$2:$G$4)
D4   =NETWORKDAYS.INTL(B4,C4,11,$G$2:$G$4)
E4   =WORKDAY.INTL(B4,5,11,$G$2:$G$4)
```

T-103 uses a **Sunday-only weekend** (code 11), common in Indian factories and retail. The holiday on 15-Aug-2026 falls on a Saturday, so it does not change T-101's count of working days.

Age, tenure and durations

`DATEDIF(start, end, unit)` is a hidden function (no tooltip) that returns complete years, months or days between two dates.

Unit	Returns
`"y"`	Complete years
`"m"`	Complete months
`"d"`	Days
`"ym"`	Months after removing complete years
`"md"`	Days after removing complete months

Tenure as of 31-Mar-2026 (in F1)

	A	B	C	D	E	F
1	Employee	Join date	Years	Months after years	Tenure label	31-03-2026
2	Kavya Iyer	28-02-2016	10	1	10 yrs 1 mths	
3	Imran Reddy	14-12-2019	6	3	6 yrs 3 mths	
4	Saanvi Rao	18-02-2025	1	1	1 yrs 1 mths	

```
C2   =DATEDIF(B2,$F$1,"y")
D2   =DATEDIF(B2,$F$1,"ym")
E2   =C2&" yrs "&D2&" mths"
```

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

(Cell F1 holds the date 31-03-2026.) For fractional years use `=YEARFRAC(B2,$F$1)`, for example 10.09 years.

Fiscal years and quarters

Calendar quarter: `="Q"&ROUNDUP(MONTH(A2)/3,0)`. Many organisations use a different financial year:

Year type	Q1	Formula for the quarter number
Calendar	Jan–Mar	`=ROUNDUP(MONTH(A2)/3,0)`
India (FY Apr–Mar)	Apr–Jun	`=ROUNDUP(MOD(MONTH(A2)-4,12)/3+0.01,0)` or `=CHOOSE(MONTH(A2),4,4,4,1,1,1,2,2,2,3,3,3)`
US federal (FY Oct–Sep)	Oct–Dec	`=CHOOSE(MONTH(A2),2,2,2,3,3,3,4,4,4,1,1,1)`

Calendar, Indian FY and US FY quarters

	A	B	C	D	E
1	Date	Calendar Q	India FY quarter	India FY label	US FY quarter
2	10-02-2026	Q1	Q4	FY26	Q2
3	27-04-2026	Q2	Q1	FY27	Q3
4	03-11-2026	Q4	Q3	FY27	Q1

```
B2   ="Q"&ROUNDUP(MONTH(A2)/3,0)
C2   ="Q"&CHOOSE(MONTH(A2),4,4,4,1,1,1,2,2,2,3,3,3)
```

```
D2   ="FY"&(YEAR(A2)+(MONTH(A2)>=4))-2000
```

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

In the Indian convention, 27-Apr-2026 falls in **FY27** (April 2026 – March 2027), quarter Q1.

Times

Function	Example	Result
`TIME(h, m, s)`	`=TIME(14,30,0)`	14:30
`HOUR`, `MINUTE`, `SECOND`	`=HOUR(A2)`	14
Duration in hours	`=(end-start)*24`	8.5
Duration beyond 24 h	Format `[h]:mm`	37:30
Shifts past midnight	`=MOD(end-start,1)`	Works when end is the next day

A very common clean-up task: a system exports times as plain numbers like `1430` meaning 14:30. Convert them with `INT` and `MOD`:

Convert 1430-style numbers into real times

	A	B	C	D	E
1	Raw	Hours =INT(A2/100)	Minutes =MOD(A2,100)	Time =TIME(INT(A2/100),MOD(A2,100),0)	Shift length (to B6)
2	930	9	30	09:30	8.75
3	1430	14	30	14:30	
4	2245	22	45	22:45	
5	5	0	5	00:05	
6	1815	18	15	18:15	

```
B2   =INT(A2/100)
C2   =MOD(A2,100)
D2   =TIME(INT(A2/100),MOD(A2,100),0)
E2   =MOD(D6-D2,1)*24
```

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

Number functions

Function	What it does	Example	Result
`INT(n)`	Rounds **down** to the integer	`=INT(-7.6)`	`-8`
`TRUNC(n, [d])`	Cuts off decimals (towards zero)	`=TRUNC(-7.6)`	`-7`
`ROUND(n, d)`	Normal rounding	`=ROUND(7.456,2)`	7.46
`ROUNDUP(n, d)`	Always away from zero	`=ROUNDUP(7.401,0)`	8
`ROUNDDOWN(n, d)`	Always towards zero	`=ROUNDDOWN(7.99,0)`	7
`ROUND(n, -3)`	Round to thousands	`=ROUND(123456,-3)`	123000
`MROUND(n, m)`	Nearest multiple	`=MROUND(17,5)`	15
`CEILING.MATH(n, m)`	Up to the next multiple	`=CEILING.MATH(23,7)`	28
`FLOOR.MATH(n, m)`	Down to the previous multiple	`=FLOOR.MATH(23,7)`	21
`MOD(n, d)`	Remainder	`=MOD(28,3)`	1
`QUOTIENT(n, d)`	Integer part of a division	`=QUOTIENT(28,3)`	9
`ABS(n)`	Absolute value	`=ABS(-450)`	450
`SIGN(n)`	−1, 0 or 1	`=SIGN(-450)`	`-1`
`SQRT(n)`	Square root	`=SQRT(144)`	12
`POWER(n, p)`	Power (same as ^)	`=POWER(25,2)`	625

Rounding functions side by side

	A	B	C	D	E	F	G	H
1	Value	INT	ROUND 0	ROUNDUP 0	ROUNDDOWN 0	ROUND 2	CEILING.MATH to 5	FLOOR.MATH to 5
2	7.456	7	7	8	7	7.46	10	5
3	-7.456	-8	-7	-8	-7	-7.46	-5	-10
4	23.5	23	24	24	23	23.5	25	20

```
B2   =INT(A2)
C2   =ROUND(A2,0)
D2   =ROUNDUP(A2,0)
E2   =ROUNDDOWN(A2,0)
G2   =CEILING.MATH(A2,5)
H2   =FLOOR.MATH(A2,5)
```

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

Practical uses

- **Packaging**: boxes needed for 523 items at 24 per box = `=CEILING.MATH(523,24)/24` or `=ROUNDUP(523/24,0)` → 22.
- **Price points**: round prices up to the next 99: `=CEILING.MATH(A2,100)-1` turns 1,234 into 1,299.
- **Split profit and loss** into two columns with ABS:

Profit and Loss columns

	A	B	C	D
1	**Store**	**Profit / Loss**	**Profit**	**Loss**
2	Pune	45200	45200	0
3	Delhi	-12800	0	12800
4	Kochi	3300	3300	0
5	Patna	-7600	0	7600
6	**Total**	**28100**	**48500**	**20400**

```
C2   =IF(B2>0,B2,0)
D2   =IF(B2<0,ABS(B2),0)
B6   =SUM(B2:B5)
C6   =SUM(C2:C5)
D6   =SUM(D2:D5)
```

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

Named ranges

A **name** is a label for a cell, a range, a constant or a formula. `=SUM(Sales)` is easier to read and audit than `=SUM(Data!$O$2:$O$6324)`.

Static named ranges

Method	How
Name Box	Select the range, click the Name Box (left of the formula bar), type `GST_Rate`, press Enter
Define Name	Formulas → Define Name: set the name, scope (workbook or one sheet) and comment
Create from Selection	Select a table with headers → Formulas → Create from Selection (`Ctrl + Shift + F3`) → Top row: one name per column
Name Manager	Formulas → Name Manager (`Ctrl + F3`) to edit, delete or find broken names

Rules: names start with a letter or underscore, contain no spaces (use _), cannot look like a cell address (`Q1` or FY26 are not allowed) and are not case-sensitive.

Formulas with names (GST_Rate = F1, Qty and Price are named columns)

	A	B	C	D	E	F
1	**Qty**	**Price**	**Amount =Qty*Price**	**With GST =Qty*Price*(1+GST_Rate)**		18%
2	4	1250	5000	5,900.00		
3	10	299	2990	3,528.20		
4	2	8999	17998	21,237.64		

```
C2   =A2*B2
D2   =A2*B2*(1+$F$1)
C3   =A3*B3
D3   =A3*B3*(1+$F$1)
C4   =A4*B4
```

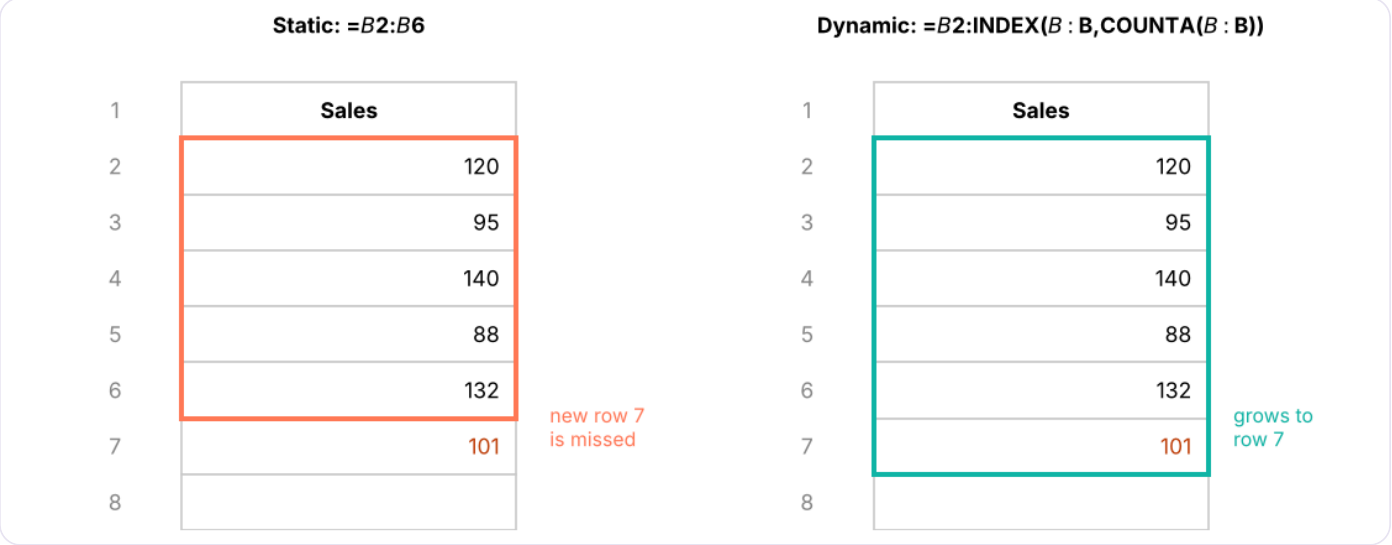
```
D4   =A4*B4*(1+$F$1)
```

(F1, the yellow cell, is named `GST_Rate`. With names defined, the formula in D2 reads `=Qty*Price*(1+GST_Rate)`.)

Dynamic named ranges

A static name like `=$B$2:$B$100` does not grow when you add row 101. A **dynamic** named range expands automatically, which keeps charts, validation lists and formulas up to date.

Method	Refers to	Notes
Excel Table (best)	`=tblSales[Amount]`	Automatic, no formula needed
`OFFSET + COUNTA`	`=OFFSET(Data!$B$2,0,0,COUNTA(Data!$B:$B)-1,1)`	Classic; volatile (recalculates often)
`INDEX + COUNTA`	`=Data!$B$2:INDEX(Data!$B:$B,COUNTA(Data!$B:$B))`	Not volatile; preferred over OFFSET
Spill reference	`=Data!$H$2#`	Points to a dynamic-array result



A static range stays fixed; a dynamic range grows with the data.

LET: names inside one formula

LET (Microsoft 365) assigns names to values **inside** a formula. It makes long formulas readable and faster, because each part is calculated once.

Without LET: `=IF(SUMIFS(F:F,C:C,"Finance")/COUNTIFS(C:C,"Finance")>90000,"High","Normal")`

With LET:

Excel formula

```
=LET(
    total, SUMIFS(F:F, C:C, "Finance"),
    heads, COUNTIFS(C:C, "Finance"),
    avg,   total / heads,
    IF(avg > 90000, "High", "Normal")
)
```

LAMBDA goes one step further and lets you create your own reusable functions; see Lesson 12.

Case: date analysis on Superstore

How long do orders take to ship, by ship mode, and which weekday gets the most orders? In Excel you would add columns `=ship_date-order_date` and `=TEXT(order_date,"ddd")` and pivot them. Here are the real answers:

Python

```
import pandas as pd
s = pd.read_csv("superstore_sales.csv", parse_dates=["order_date", "ship_date"])
orders = s.drop_duplicates("order_id").copy()
orders["days_to_ship"] = (orders.ship_date - orders.order_date).dt.days            # =C2-B2
orders["weekday"] = orders.order_date.dt.day_name().str[:3]                         # =TEXT(B2,"ddd")
orders["fy"] = "FY" + (orders.order_date.dt.year + (orders.order_date.dt.month ≥
4)).astype(str).str[2:]   # Indian FY
print(orders.groupby("ship_mode")["days_to_ship"].agg(["mean", "min", "max"]).round(1).to_string())
print()
print(orders.groupby("fy").size().rename("orders").to_string())
```

OUTPUT

```
                mean  min  max
ship_mode
First Class      1.5    1    2
Same Day         0.0    0    0
Second Class     3.0    2    4
Standard Class   5.5    4    7

fy
FY23     169
FY24     952
FY25    1114
FY26     965
```

Summary

KEY TAKEAWAYS

- Dates are whole numbers (days since 1-Jan-1900); times are fractions of a day. Subtract dates to get days; multiply time differences by 24 for hours.
- `EOMONTH` and `EDATE` handle month ends correctly; `NETWORKDAYS` and `WORKDAY` (with `.INTL` for custom weekends) handle working days and holidays.
- `DATEDIF` gives complete years/months; build fiscal quarters with `CHOOSE(MONTH(…),…)`.
- `INT` rounds down, `TRUNC` cuts towards zero, `ROUNDUP`/`ROUNDDOWN` force direction, `CEILING.MATH`/`FLOOR.MATH` round to multiples, `MOD` gives remainders.
- Named ranges make formulas readable; use Tables or `INDEX`-based names for dynamic ranges, and `LET` for names inside a formula.

Quick quiz

6 questions · tap an answer to check it

1. A1 contains 27-04-2026 and B1 contains 02-05-2026. What does `=B1-A1` return?

A 5 days as text

B 5

C 0.05

D **VALUE!**

Answer: B. Dates are numbers, so subtracting gives the number of days.

2. Which formula returns the last day of the month for the date in A2?

A `=DATE(YEAR(A2),MONTH(A2),31)`

B `=EOMONTH(A2,0)`

C `=EDATE(A2,1)-1`

D `=MONTH(A2)+30`

Answer: B. EOMONTH(date,0) returns the month end; the other options fail for 30-day months or February.

3. What does `=INT(-3.2)` return?

A −3

B −4

C 3

D −3.2

Answer: B. INT always rounds down (towards minus infinity). TRUNC(-3.2) would give −3.

4. Which formula converts the number 1745 into the time 17:45?

A `=TIME(1745)`

B `=1745/24`

C `=TIME(INT(A2/100),MOD(A2,100),0)`

D `=TEXT(1745,"hh:mm")`

Answer: C. INT(1745/100) = 17 hours and MOD(1745,100) = 45 minutes.

5. Why is an INDEX-based dynamic named range often preferred to OFFSET?

A It works only in Excel 365

B OFFSET is volatile and recalculates on every change, which can slow large workbooks

C INDEX cannot refer to whole columns

D OFFSET does not grow with the data

Answer: B. Both grow, but OFFSET is volatile.

6. In the Indian financial year, which quarter is 15-January?

A Q1

B Q2

C Q3

D Q4

Answer: D. FY runs April–March, so January–March is Q4.

EXERCISE **HR date calculations**

Using `employees_small.csv` and an "as of" date of 31-03-2026 in a cell: (1) tenure in complete years; (2) the probation end date (6 months after joining); (3) the first working day after the probation end; (4) the joining quarter in the Indian FY.

Show solution (worked answer)

With join date in E2 and the as-of date in $J$1: (1) `=DATEDIF(E2,$J$1,"y")` (2) `=EDATE(E2,6)` (3) `=WORKDAY(EDATE(E2,6)-1,1)` (the first working day on or after the probation end) (4) `="Q"&CHOOSE(MONTH(E2),4,4,4,1,1,1,2,2,2,3,3,3)`

INTERVIEW CORNER **Questions you may be asked**

Q: How does Excel store dates and times? Dates as serial numbers counting days from 1-Jan-1900; times as fractions of a day. This makes arithmetic, sorting and formatting consistent.

Q: How do you count working days between two dates excluding holidays? `NETWORKDAYS(start, end, holidays)` or `NETWORKDAYS.INTL` with a custom weekend code.

Q: What is a dynamic named range and when do you need one? A name whose range grows with the data, built with an Excel Table, `INDEX/COUNTA` or `OFFSET/COUNTA`. It keeps charts, validation lists and formulas current as data is added.

ADVANCED EXCEL · LESSON 10

Charts and Advanced Charts

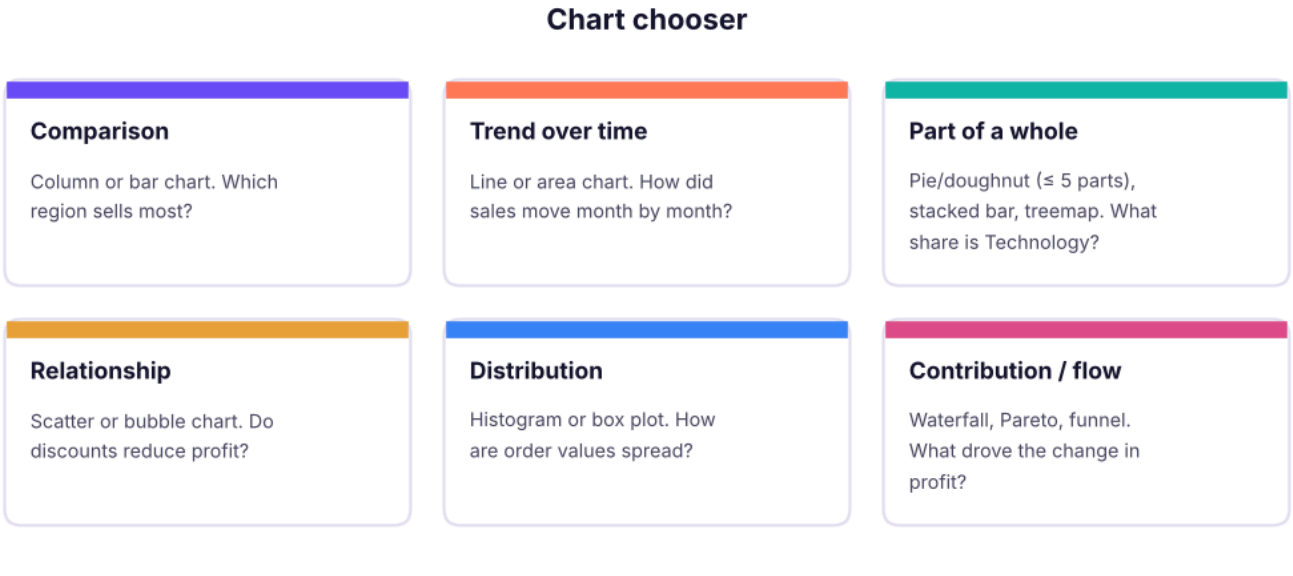
Choose the right chart for every question, build and format column, bar, line, area, pie, scatter and combo charts, add sparklines, and create the advanced business charts taught in class: thermometer, Pareto, population pyramid and waterfall, plus histogram, box plot, treemap and funnel.

Live classes 7–8 · ✦ Excel for Microsoft 365, 2026 · Intermediate · ⏱ 130 min

By the end of this lesson you can

- Pick the right chart type from the question you want to answer
- Insert, edit and format charts quickly (elements, axes, labels, colours, templates)
- Build combo charts with a secondary axis and add sparklines
- Build thermometer, Pareto, population pyramid and waterfall charts step by step
- Use the newer statistical and hierarchy charts and apply chart design best practices

Start with the question, not the chart



Choose a chart by the question you are answering.

Creating and editing a chart

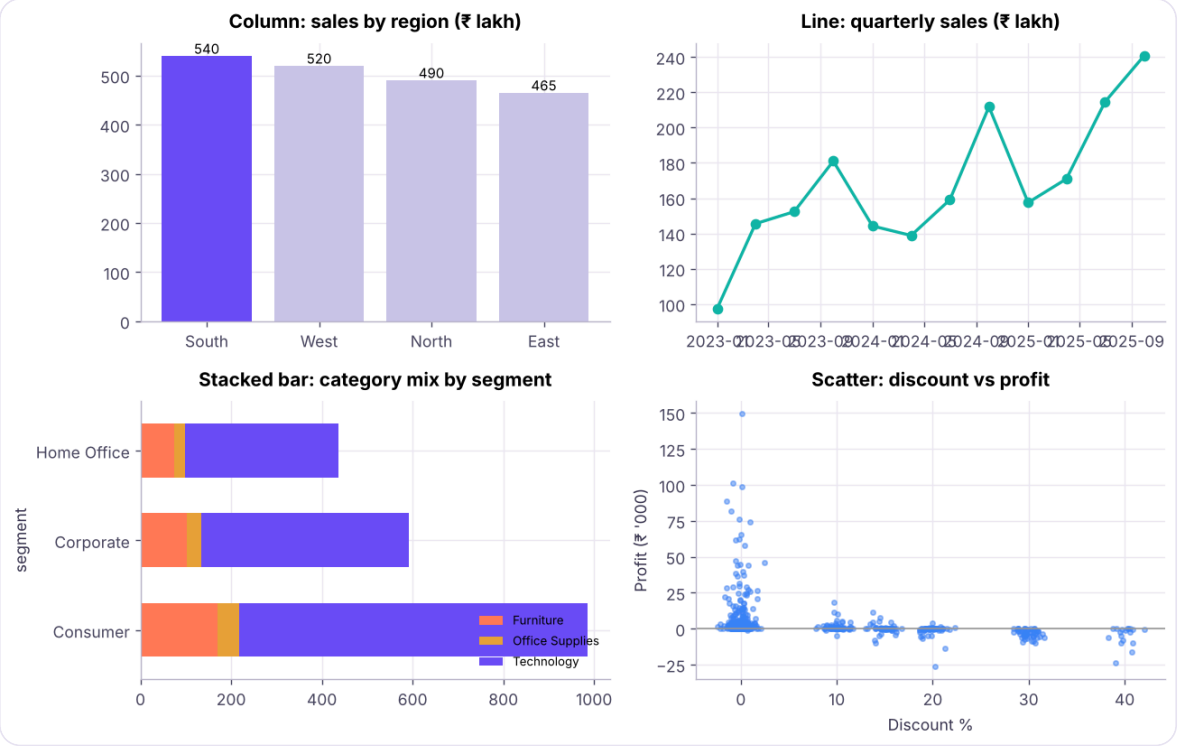
STEP BY STEP **The fast way**

1. Select the data **including headers** (or a summary table / pivot table). For a quick chart press `Alt + F1` (embedded) or `F11` (new chart sheet).
2. Or use **Insert → Recommended Charts** to preview options.
3. Click the chart: three buttons appear at its top-right: **Chart Elements** (+), **Chart Styles** (brush) and **Chart Filters** (funnel).
4. **Chart Design** tab: Switch Row/Column, Select Data, Change Chart Type, Move Chart, Quick Layout.
5. **Format** pane: double-click any element (axis, series, label) or press `Ctrl + 1` with it selected.

Element	Good practice
Title	Say the insight: "Technology drives 77% of sales", not "Sales by Category"
Axes	Start bar/column axes at zero; format numbers (K, L, %); remove unnecessary decimals
Gridlines	Light grey or none
Data labels	Use them instead of an axis when there are few bars
Legend	Only if there is more than one series; better: label lines directly
Colour	One highlight colour for the point you want to make, grey for the rest
Template	Right-click → Save as Template to reuse your style

The core chart types

Here are the core charts built from Superstore data. The Excel steps follow each one.



Column (comparison), line (trend), clustered and stacked bar (comparison of parts), and scatter (relationship).

Chart	Excel path	Use when
Clustered column / bar	Insert → Column or Bar	Comparing categories; use **bar** when labels are long or there are many categories
Stacked / 100% stacked	Same menu	Showing parts of a total across groups
Line	Insert → Line (with markers)	Trends over time; up to ~5 lines
Area	Insert → Line → Area	Cumulative trends; volume over time
Pie / doughnut	Insert → Pie	2–5 parts of a whole that add to 100%
Scatter (XY)	Insert → Scatter	Relationship between two numbers; add a **trendline** (right-click series → Add Trendline, tick Display R^2)
Bubble	Insert → Scatter → Bubble	Scatter with a third measure as bubble size

COMMON MISTAKES

Chart crimes to avoid

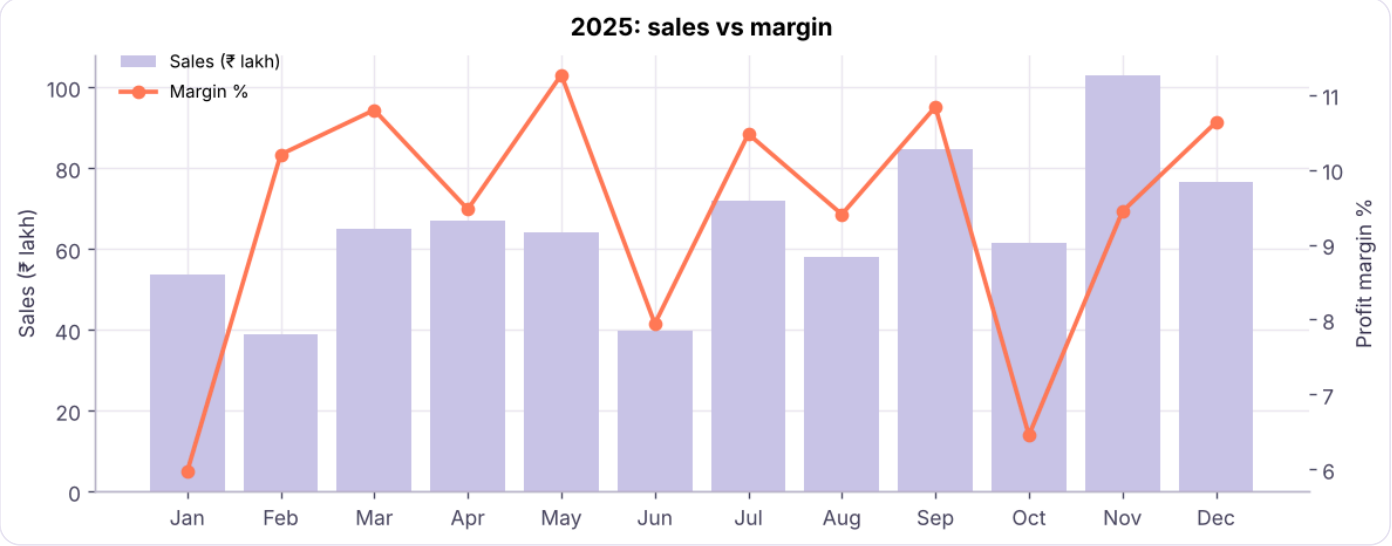
3D charts, pie charts with 10 slices, two different units on one axis, truncated bar axes that exaggerate differences, rainbow colours, and charts without units. If you need to squint, the chart has failed.

Combo charts and the secondary axis

When two measures have very different scales (sales in lakhs and profit margin in %), plot one as columns and the other as a line on a **secondary axis**.

STEP BY STEP **Build a combo chart**

1. Make a summary table: Month, Sales, Margin %.
2. Select it → Insert → **Combo Chart** → Create Custom Combo Chart.
3. Sales: Clustered Column. Margin %: Line with Markers, tick **Secondary Axis**.
4. Add axis titles so readers know which axis belongs to which series.



Combo chart: monthly sales (columns, left axis) and profit margin (line, right axis) for 2025.

Sparklines

Sparklines are tiny charts **inside a cell**: Insert → **Sparklines** → Line, Column or Win/Loss. Select the data range (for example 12 months in a row) and the location cell. Use **Sparkline → Show → High Point / Low Point** to mark extremes. They are ideal in dashboard tables, one row per product or region.

Thermometer chart (target vs actual)

A thermometer shows progress towards a single target, such as "₹ 7.4 crore of the ₹ 9 crore annual target".

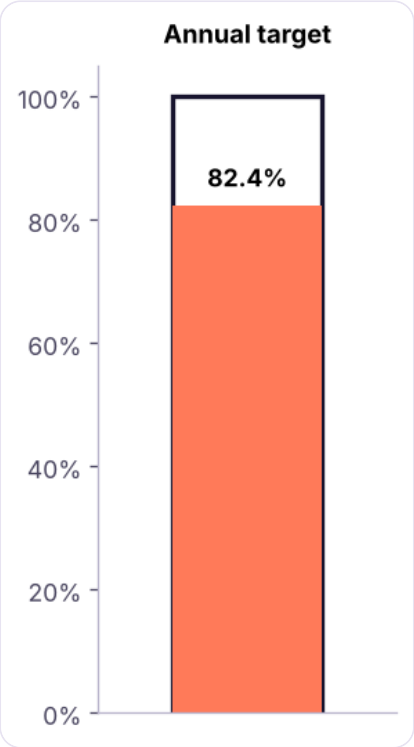
X Thermometer data

	A	B
1	Metric	Value
2	Target	900
3	Achieved	742
4	% achieved	82.4%

```
B4    =B3/B2
```

STEP BY STEP **Build it**

1. Select only the **% achieved** cell (and its label) → Insert → **Clustered Column**.
2. Format the vertical axis: Minimum **0**, Maximum **1** (100%). Delete the horizontal axis and gridlines.
3. Format the series: Gap Width **0%** so the column fills the width; choose a strong fill colour.
4. Optional: add a second series with value 1 (100%) on the secondary axis, no fill and a thick border to draw the "glass" of the thermometer; set the secondary axis to 0–1 as well.
5. Resize the chart to be tall and narrow; add a data label showing 82.4%.



Thermometer chart: 82.4% of the annual target achieved.

Pareto chart (the 80/20 rule)

The Pareto principle says that roughly 80% of effects come from 20% of causes: a few sub-categories drive most sales, a few complaint types cause most tickets. A Pareto chart shows sorted bars plus a **cumulative %** line.

Python

```python
pareto = s.groupby("sub_category")["sales"].sum().sort_values(ascending=False).to_frame()
pareto["share %"] = (pareto.sales / pareto.sales.sum() * 100).round(1)
pareto["cumulative %"] = pareto["share %"].cumsum().round(1)
pareto["sales"] = (pareto.sales / 1e5).round(1)
pareto.rename(columns={"sales": "sales (₹ lakh)"})
```

OUTPUT

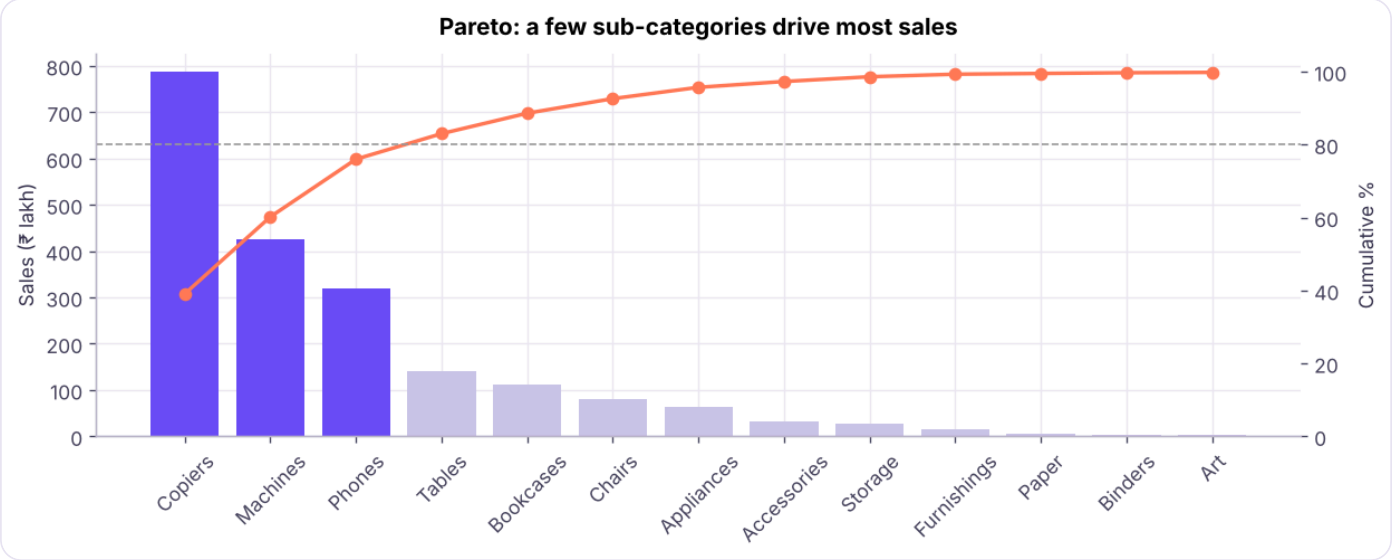
	sales (₹ lakh)	share %	cumulative %
sub_category			
Copiers	787.5	39.1	39.1
Machines	425.9	21.1	60.2
Phones	319.2	15.8	76.0
Tables	140.5	7.0	83.0
Bookcases	112.7	5.6	88.6
Chairs	80.9	4.0	92.6
Appliances	63.3	3.1	95.7
Accessories	31.8	1.6	97.3
Storage	26.6	1.3	98.6
Furnishings	14.3	0.7	99.3
Paper	4.9	0.2	99.5
Binders	4.0	0.2	99.7

13 rows × 3 columns (first 12 shown)

STEP BY STEP

Build it in Excel

1. Summarise sales by sub-category (pivot or SUMIFS) and **sort largest to smallest**.
2. Add `Cumulative %`: in C2 `=SUM($B$2:B2)/SUM($B$2:$B$13)`, copy down.
3. Select the three columns → Insert → **Combo**: Sales = Clustered Column, Cumulative % = Line on the **Secondary Axis**.
4. Set the secondary axis maximum to 100%. Optionally add a horizontal 80% reference line.
5. Excel 2016+ also has a built-in **Insert → Statistical → Pareto** chart that sorts and accumulates for you from raw data.



Pareto chart of sales by sub-category. Copiers, Machines and Phones alone make up 76% of sales; adding Tables passes the 80% line.

Population pyramid

A population pyramid compares two groups (typically male and female) across age bands, with one group's bars going left and the other's going right. In business it works for any two-sided comparison: this year vs last year by product, or two regions by age band.

Python

```
hr = pd.read_csv("hr_employees.csv")
hr["age_band"] = pd.cut(hr.age, [20, 25, 30, 35, 40, 45, 50, 55, 60], labels=["21-25", "26-30", "31-35", "36-40", "41-45", "46-50", "51-55", "56-60"])
pyr = hr.pivot_table(index="age_band", columns="gender", values="emp_id", aggfunc="count", observed=False)
pyr["Male (negative)"] = -pyr["Male"]
pyr
```

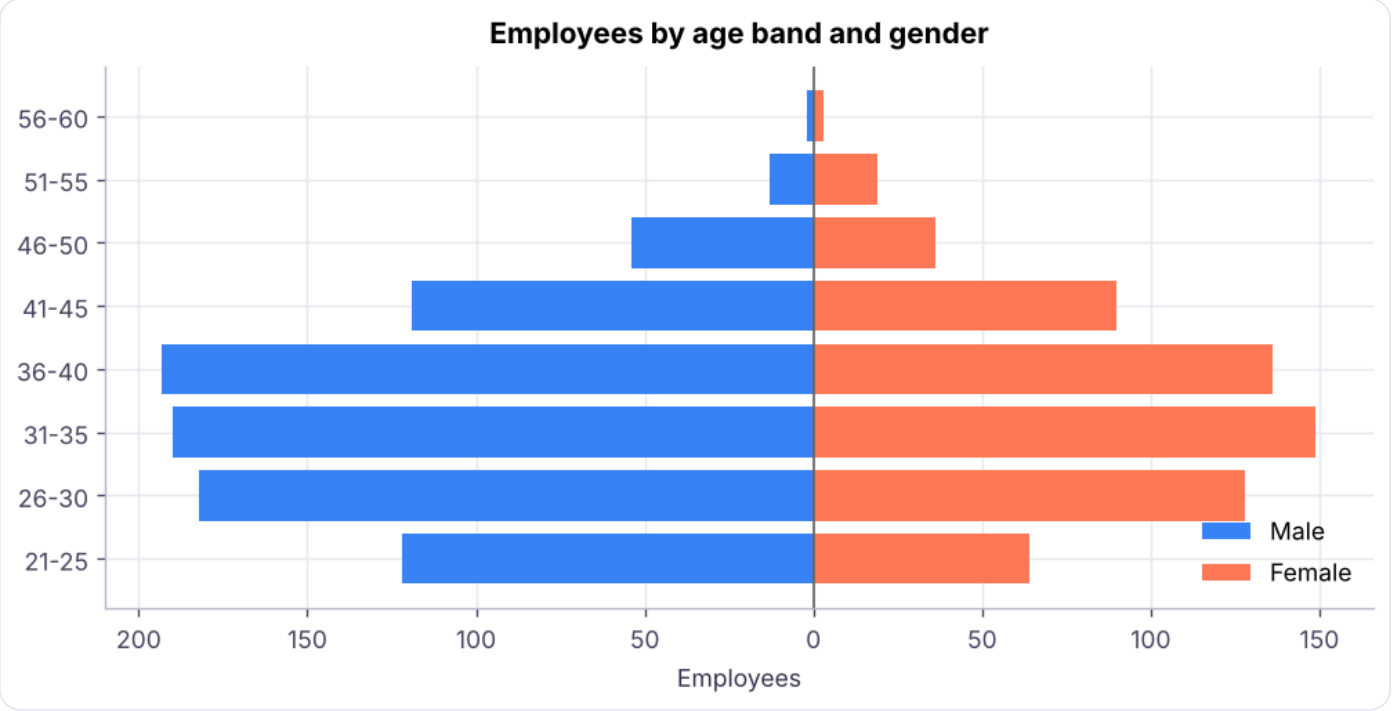
OUTPUT

gender	Female	Male	Male (negative)
age_band			
21-25	64	122	-122
26-30	128	182	-182
31-35	149	190	-190
36-40	136	193	-193
41-45	90	119	-119
46-50	36	54	-54
51-55	19	13	-13
56-60	3	2	-2

8 rows × 3 columns

STEP BY STEP **Build it**

1. Count employees by age band and gender (pivot table or COUNTIFS).
2. Make the male counts **negative** (`=-C2`) in a helper column so they plot to the left.
3. Select age bands, the negative male column and the female column → Insert → **Clustered Bar**.
4. Format series: Series Overlap **100%**, Gap Width **10–30%**.
5. Format the horizontal axis numbers with the custom format `#,##0;#,##0` so negative values show without a minus sign.
6. Vertical axis → Labels → Label Position: **Low** so the age labels sit on the left edge.



Population pyramid of the 1,500 employees in the HR dataset.

Waterfall chart

A waterfall shows how a starting value becomes an ending value through positive and negative steps: revenue to profit, last year's sales to this year's, or budget to actual.

Waterfall data: from revenue to net profit (₹ lakh)

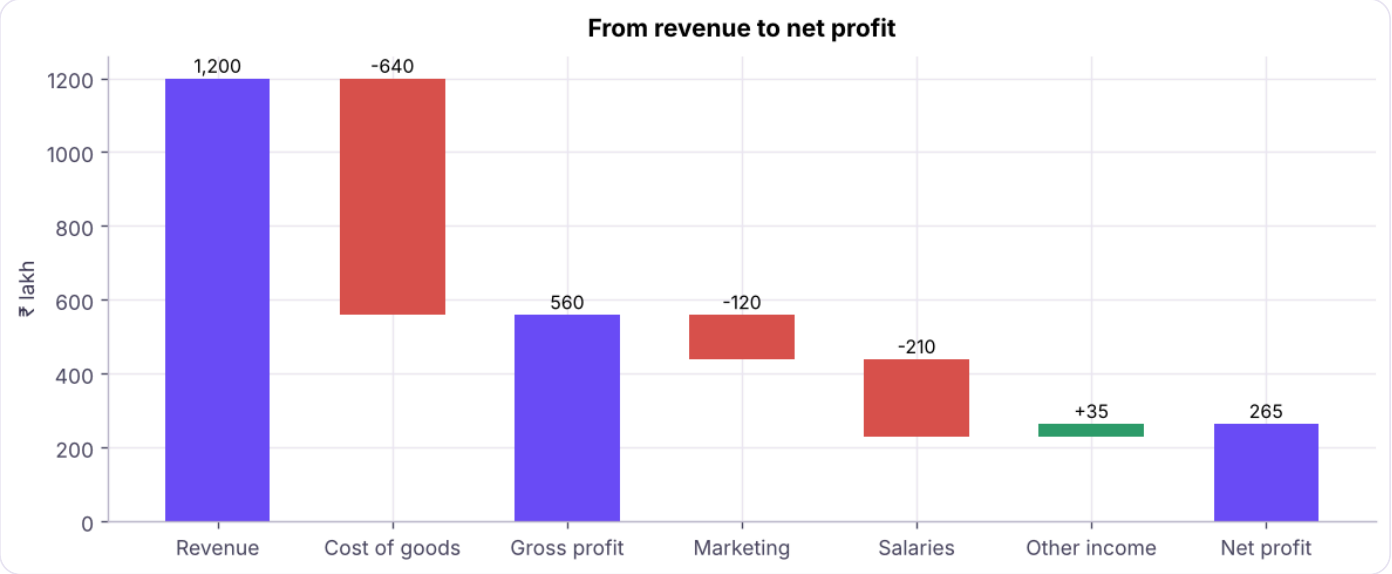
	A	B
1	**Step**	**Amount**
2	Revenue	1200
3	Cost of goods	-640
4	Gross profit	560
5	Marketing	-120
6	Salaries	-210
7	Other income	35
8	Net profit	265

B4 `=B2+B3`
B8 `=B4+B5+B6+B7`

STEP BY STEP **Build it (Excel 2016 and later)**

1. Select the table → Insert → **Waterfall** (Insert → Charts → Waterfall, Funnel...).
2. Click the **Gross profit** bar twice (to select only that point) → right-click → **Set as total**. Do the same for **Net profit**. Totals then start from zero instead of floating.
3. Colour increases, decreases and totals differently (Format Data Point) and add data labels.

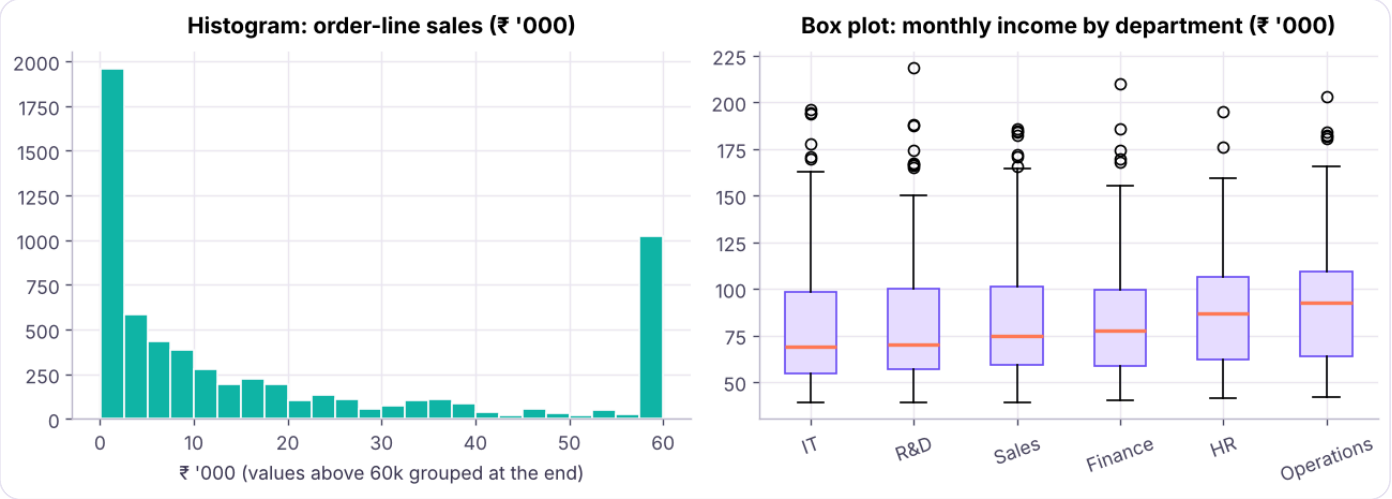
In older Excel versions you build the same chart as a **stacked column** with an invisible "base" series, an "up" series and a "down" series computed with formulas.



Waterfall from revenue to net profit. Totals are anchored at zero; increases are green and decreases are red.

Statistical and hierarchy charts (Excel 2016+)

Chart	Insert path	Shows	Example question
Histogram	Insert → Statistical → Histogram	Frequency distribution; set bin width in Format Axis	How are order values distributed?
Box & Whisker	Insert → Statistical → Box and Whisker	Median, quartiles and outliers by group	Do salaries vary more in Sales than in IT?
Treemap	Insert → Hierarchy → Treemap	Nested rectangles sized by value	Category → sub-category share of sales
Sunburst	Insert → Hierarchy → Sunburst	Rings for hierarchy levels	Region → state → city
Funnel	Insert → Waterfall → Funnel	Drop-off between stages	Visitors → sign-ups → orders
Filled map	Insert → Maps	Values by country or state	Sales by state (needs internet)



Histogram of order-line sales values and box plots of monthly income by department (HR data).

Dynamic charts

Charts become interactive when their source updates:

- **Table source**: new rows are added to the chart automatically.
- **Named range / dynamic named range** (Lesson 9) as the series values: `=Book1.xlsx!SalesLast12`.
- **Drop-down + formulas**: a Data Validation drop-down picks a region, formulas (XLOOKUP/FILTER) pull that region's numbers into a helper table, and the chart reads the helper table.
- **Pivot charts + slicers** (Lesson 8) for click-to-filter dashboards (Lesson 11).

2026 UPDATE **Charts in Excel 365 (2026)**

Copilot can suggest and insert charts from a sentence ("show monthly sales by region as a line chart"), and **Analyze Data** (Home → Analyze Data) proposes ready-made pivot charts with insights. Python in Excel can create seaborn or matplotlib charts inside the grid with `=PY()`.

Summary

KEY TAKEAWAYS

- Choose the chart from the question: comparison (bar/column), trend (line), part-of-whole (pie for ≤5, stacked bar, treemap), relationship (scatter), distribution (histogram, box), contribution (waterfall, Pareto).
- Titles should state the insight; start bar axes at zero; use one highlight colour.
- Combo charts with a secondary axis show two measures of different scales.
- **Thermometer**: one % column, axis 0–100%, gap width 0.
- **Pareto**: sorted bars + cumulative % line on a secondary axis; look for the ~80% point.
- **Population pyramid**: one series negative, overlap 100%, custom format `#,##0;#,##0`.
- **Waterfall**: Insert → Waterfall, then "Set as total" for subtotal bars.

Quick quiz

6 questions · tap an answer to check it

1. Which chart best shows how monthly revenue changed over two years?

- **A** Pie
- **B** Line
- **C** Treemap
- **D** Scatter

Answer: B. Lines show trends over time.

2. In a Pareto chart, what does the line represent?

- **A** The average value
- **B** The cumulative percentage of the total
- **C** The target
- **D** The trend

Answer: B. The line accumulates each bar's share to show where ~80% is reached.

3. How do you make the male bars go to the left in a population pyramid?

- **A** Make the male values negative and format the axis to hide the minus sign
- **B** Use a pie chart
- **C** Reverse the category order
- **D** Put males on a secondary axis

Answer: A. Negative values plot to the left; `#,##0;#,##0` shows them as positive numbers.

4. In Excel's waterfall chart, why must Gross profit be "Set as total"?

A To change its colour

B So it starts from zero instead of floating on the previous step

C To hide it

D To make it a line

Answer: B. Totals should be anchored to the axis; otherwise Excel treats them as another step.

5. When should you use a secondary axis?

A Whenever there are two series

B When two series have very different scales or units, such as sales and margin %

C When the chart is a pie

D Never

Answer: B. A secondary axis lets both series be readable; label both axes clearly.

6. Which chart shows median, quartiles and outliers for several groups?

A Histogram

B Box and whisker

C Funnel

D Sunburst

Answer: B. Box plots summarise distributions and highlight outliers.

EXERCISE **Build four charts from Superstore**

(1) A Pareto chart of profit by sub-category (watch out for negative profit: what does it mean?). (2) A combo chart of quarterly sales (columns) and order count (line, secondary axis). (3) A waterfall from 2024 total sales to 2025 total sales, with one step per region showing the change. (4) A box plot of discount by category.

Show solution (worked answer)

(1) Sort profit descending; negative sub-categories (such as Tables and Bookcases here) reduce the cumulative total, so the cumulative line rises above 100% and then falls; a Pareto chart works best on positive values, so plot losses separately. (2) Pivot by quarter with Sum of sales and Count of order_id → Combo chart with the count on the secondary axis. (3) Table: 2024 total; North change; South change; East change; West change; 2025 total. Insert → Waterfall; set the first and last bars as totals. (4) Select category and discount columns → Insert → Box and Whisker.

INTERVIEW CORNER **Questions you may be asked**

Q: How do you choose a chart type? From the question: comparison, trend, composition, relationship, distribution or contribution. Then simplify: remove clutter, highlight the key point and write an insight title.

Q: What is a Pareto analysis used for? To find the vital few causes (products, customers, complaint types) that produce most of the effect, so that effort is focused where it matters.

Q: How would you show actual vs target for 10 sales reps? A bar chart of actuals with a target marker (a line or a thin overlapping bar per rep), or a bullet chart; a thermometer works for one overall target.

ADVANCED EXCEL · LESSON 11

Interactive Dashboard: Superstore Case Study

Build a complete, interactive sales dashboard from the Superstore data, from business questions and KPIs to pivot charts, slicers, a timeline, KPI cards and a dynamic title. Then protect it (sheet, workbook, allowed ranges), document it with comments and notes, and audit formulas with Evaluate Formula.

● Live classes 8–9 ✦ Excel for Microsoft 365, 2026 Intermediate–Advanced ⏱ 150 min

By the end of this lesson you can

- Turn a business brief into KPIs and questions before building anything
- Prepare data, create one pivot per visual and connect them with slicers and a timeline
- Build KPI cards and a dynamic title with GETPIVOTDATA and TEXT
- Lay out a clean dashboard and apply design rules
- Protect sheets and workbooks, use comments and notes, and audit formulas with Evaluate Formula

The brief

CASE STUDY **Superstore sales review**

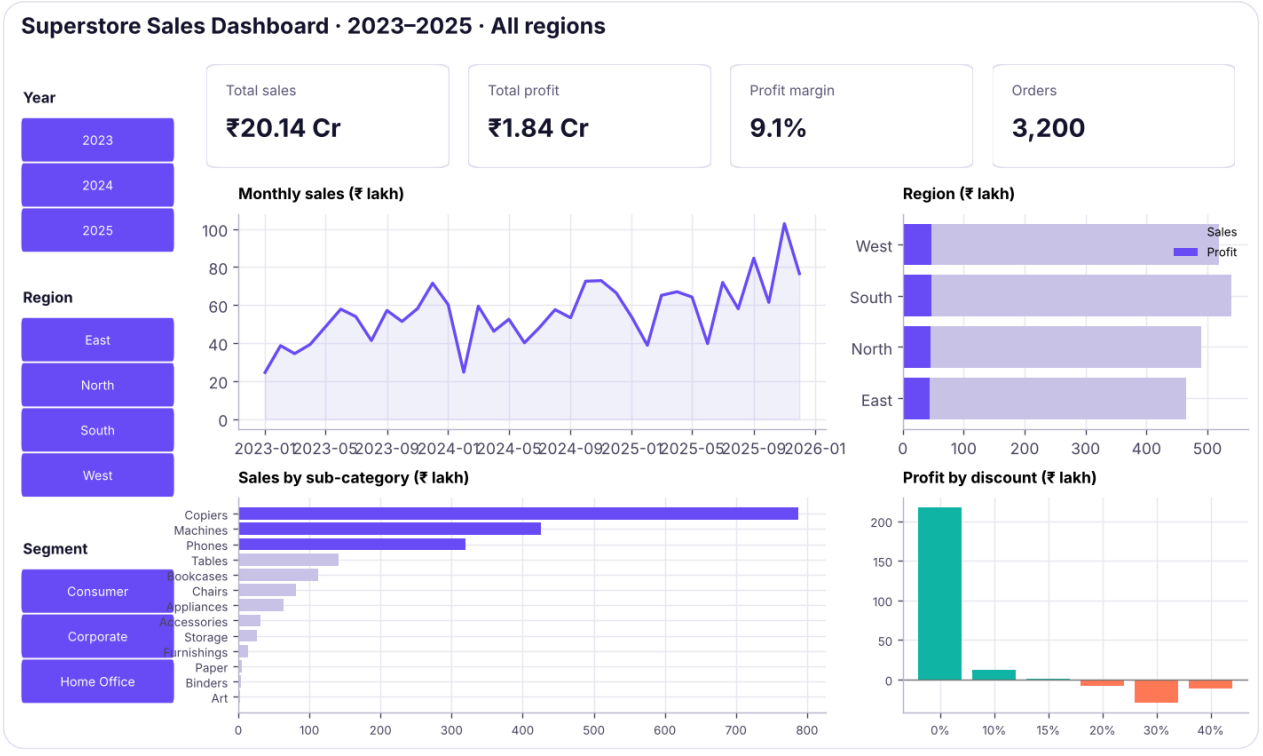
The Head of Sales wants a one-page Excel dashboard to review performance for 2023–2025. They want to see, at a glance: total sales, profit, profit margin and orders; how sales are trending; which regions, categories and sub-categories perform best and worst; and the effect of discounts on profit. They want to filter by year, region and segment. The dashboard will be emailed every month, so it must be protected against accidental changes.

Step 0: questions and KPIs first

KPI / question	Calculation	Visual
Total sales	Sum of sales	KPI card
Total profit and margin	Sum of profit; profit ÷ sales	KPI cards
Number of orders	Distinct count of order_id	KPI card
Sales trend	Sales by month	Line chart
Regional performance	Sales and profit by region	Bar chart
Product mix	Sales by category and sub-category	Bar chart / treemap
Discount impact	Profit by discount level	Column chart
Filters	Year, Region, Segment	Timeline + slicers

Agreeing on this table before building saves hours of rework. It is also your "requirements document" in interviews.

Here is the finished result we are aiming for (drawn from the real data):



The finished Superstore dashboard: KPI cards on top, trend and breakdown charts below, slicers on the left.

Step 1: prepare the data

1. Import `superstore_sales.csv` with **Data → From Text/CSV** (Power Query, Lesson 12), or open it and convert the range to a Table named `tblSales` (`Ctrl + T`).
2. Add helper columns that the pivots will need:

Column	Formula (structured reference)	Why
`year`	`=YEAR([@order_date])`	Year slicer (or use a Timeline)
`month`	`=TEXT([@order_date],"mmm")`	Labels
`margin`	`=[@profit]/[@sales]`	Row-level margin for analysis
`discount_band`	`=IFS([@discount]=0,"0%",[@discount]⩽0.2,"1-20%",TRUE,">20%")`	Simpler grouping
`days_to_ship`	`=[@ship_date]-[@order_date]`	Operations KPI

3. Check data quality: blanks (`COUNTBLANK`), duplicates of `row_id` , negative quantities, dates outside 2023–2025.

Python

```
s.describe(include="all").T[["count", "unique", "min", "max"]].head(12)
```

OUTPUT

	count	unique	min	max
row_id	6,323	NaN	1	6,323
order_id	6323	3200	NaN	NaN
order_date	6323	NaN	2023-01-01	2025-12-31
ship_date	6323	1040	NaN	NaN
ship_mode	6323	4	NaN	NaN
customer_id	6323	785	NaN	NaN
customer_name	6323	424	NaN	NaN
segment	6323	3	NaN	NaN
city	6323	16	NaN	NaN
state	6323	15	NaN	NaN
region	6323	4	NaN	NaN
product_id	6323	78	NaN	NaN

12 rows × 4 columns

Step 2: one pivot table per visual

Create a sheet called **Calc** (it will be hidden later). Insert one pivot table for each chart, all from `tblSales` so they share **one pivot cache**:

Pivot	Rows	Values	Notes
P1 KPIs	(none)	Sum of sales, Sum of profit, Distinct count of order_id	Tick "Add this data to the Data Model" to get Distinct Count
P2 Trend	order_date grouped by Years + Months	Sum of sales	Line chart
P3 Region	region	Sum of sales, Sum of profit	Bar chart
P4 Sub-category	sub_category	Sum of sales (sorted)	Bar chart
P5 Discount	discount	Sum of profit	Column chart

Give each pivot a meaningful name (PivotTable Analyze → PivotTable Name: `ptTrend`, `ptRegion` ...). Names make GETPIVOTDATA formulas and maintenance easier.

Step 3: pivot charts

Click each pivot → PivotTable Analyze → **PivotChart** → choose the type → cut and paste the chart onto a sheet called **Dashboard**.

Clean each chart:

- Right-click the field buttons → **Hide All Field Buttons on Chart**.
- Delete gridlines and legends that are not needed; add data labels where there are few bars.
- Apply the same font and a two-colour palette (one highlight, one neutral).
- Write insight titles, for example "Technology drives most sales" or "Discounts above 20% destroy profit".

Step 4: slicers and a timeline that control everything

1. Select any pivot → PivotTable Analyze → **Insert Slicer** → tick `region` and `segment`.
2. **Insert Timeline** → `order_date` → switch the timeline to **YEARS**.
3. Right-click each slicer → **Report Connections** → tick **all five** pivots. Now one click filters every chart and KPI.
4. Format slicers: Slicer tab → Columns = 1 or 2, a matching style, size them to align with the grid.

> **WATCH OUT** **Report Connections are greyed out?**
>
> Slicers can only connect pivots that share the same pivot cache (or the same Data Model). If you built a pivot from a different range, rebuild it from `tblSales`.

Step 5: KPI cards and a dynamic title

KPI cards are simply cells or shapes linked to the KPI pivot. `GETPIVOTDATA` keeps returning the right number even when slicers change the pivot.

Calc sheet: KPI formulas (values shown for the unfiltered data)

	A	B	C
1	KPI	Formula	Value
2	Total sales	=GETPIVOTDATA("Sum of sales",Calc!$A$3)	201,413,569
3	Total profit	=GETPIVOTDATA("Sum of profit",Calc!$A$3)	18,352,164
4	Profit margin	=C3/C2	9.1%
5	Orders	=GETPIVOTDATA("Distinct Count of order_id",Calc!$A$3)	3,200
6	Sales label	="₹"&TEXT(C2/10^7,"0.00")&" Cr"	₹20.14 Cr

```
C4   =C3/C2
C6   ="₹"&TEXT(C2/10^7,"0.00")&" Cr"
```

Let's verify these numbers with pandas:

Python

```python
print(f"Sales  : {s.sales.sum():,.0f}")
print(f"Profit : {s.profit.sum():,.0f}")
print(f"Margin : {s.profit.sum() / s.sales.sum():.1%}")
print(f"Orders : {s.order_id.nunique():,}")
```

OUTPUT

```
Sales  : 201,413,569
Profit : 18,352,164
Margin : 9.1%
Orders : 3,200
```

To show a KPI card on the dashboard: Insert → Shapes → Rounded Rectangle, click the shape, type `=Calc!C6` in the **formula bar**, and press Enter. The shape now displays the live value.

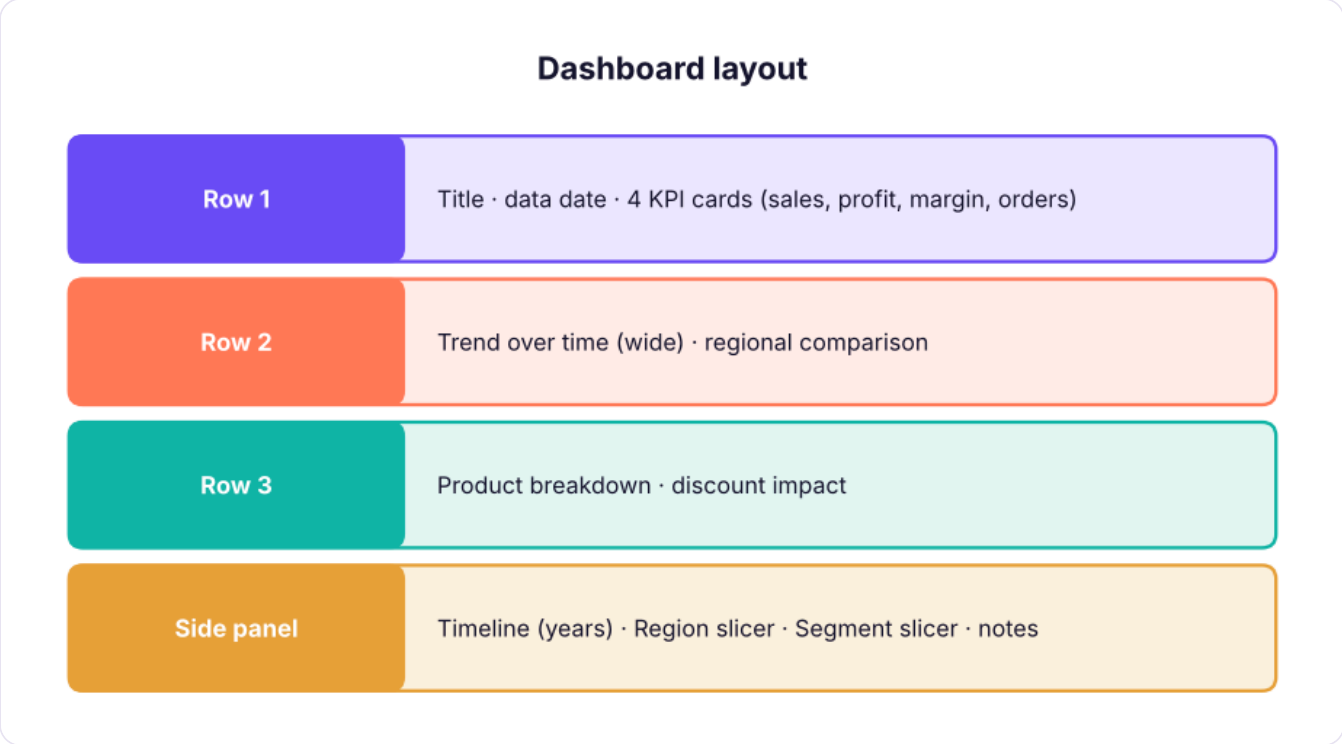
Dynamic title: build the title text in a cell and link a text box to it:

Excel formula

```
="Superstore Sales Dashboard · " & TEXTJOIN(", ", TRUE, Calc!H2:H5)
```

where `H2:H5` is a small pivot with `region` in Rows that is also connected to the slicers, so it lists only the selected regions.

Step 6: layout and design rules



A simple dashboard grid. Most important information top-left; filters on one side; consistent spacing.

1. **Z-pattern**: people read top-left first: put the headline KPIs there.
2. **Align to the grid**: hold `Alt` while dragging or resizing a chart to snap it to cell borders; set every chart to the same height.
3. **Limit colours**: one brand colour, one highlight, red only for bad news.
4. **Remove clutter**: View → untick Gridlines and Headings; hide the Calc sheet (right-click tab → Hide).
5. **Label the date and the source**: "Data: Superstore orders, Jan 2023 – Dec 2025. Refreshed 05-Jan-2026."
6. **Test it**: click every slicer combination; check that totals match the raw data.

Step 7: protect the dashboard

Protection	Where	What it does
Lock / unlock cells	`Ctrl + 1` → Protection → Locked	All cells are locked by default; **unlock** the input cells users may change. Locking only takes effect when the sheet is protected
Hide formulas	`Ctrl + 1` → Protection → Hidden	Formula bar shows nothing for those cells once the sheet is protected
Protect Sheet	Review → Protect Sheet	Prevents editing locked cells; tick what users may still do: **Use PivotTable & PivotChart**, **Edit objects** (needed for slicers), sort, filter, format
Allow Edit Ranges	Review → Allow Edit Ranges	Lets specific ranges be edited, optionally with their own password
Protect Workbook	Review → Protect Workbook	Protects the **structure**: no adding, deleting, renaming, moving or unhiding sheets
Encrypt with password	File → Info → Protect Workbook → Encrypt	Password needed to **open** the file
Mark as Final	File → Info → Protect Workbook	A read-only hint (not security)

STEP BY STEP **Protect so slicers still work**

1. Unlock any cells users need to type in (for example a target input).
2. Review → **Protect Sheet** → tick "Select unlocked cells", "Use PivotTable & PivotChart" and "Edit objects" → set a password.
3. Review → **Protect Workbook** (structure) so the hidden Calc sheet cannot be unhidden.
4. Test with a colleague before emailing.

WATCH OUT **Sheet protection is not strong security**

Sheet and structure passwords stop accidents, not determined users. Never rely on them to protect confidential data: use file encryption, SharePoint permissions or share only what is needed (for example a PDF export).

Comments and notes

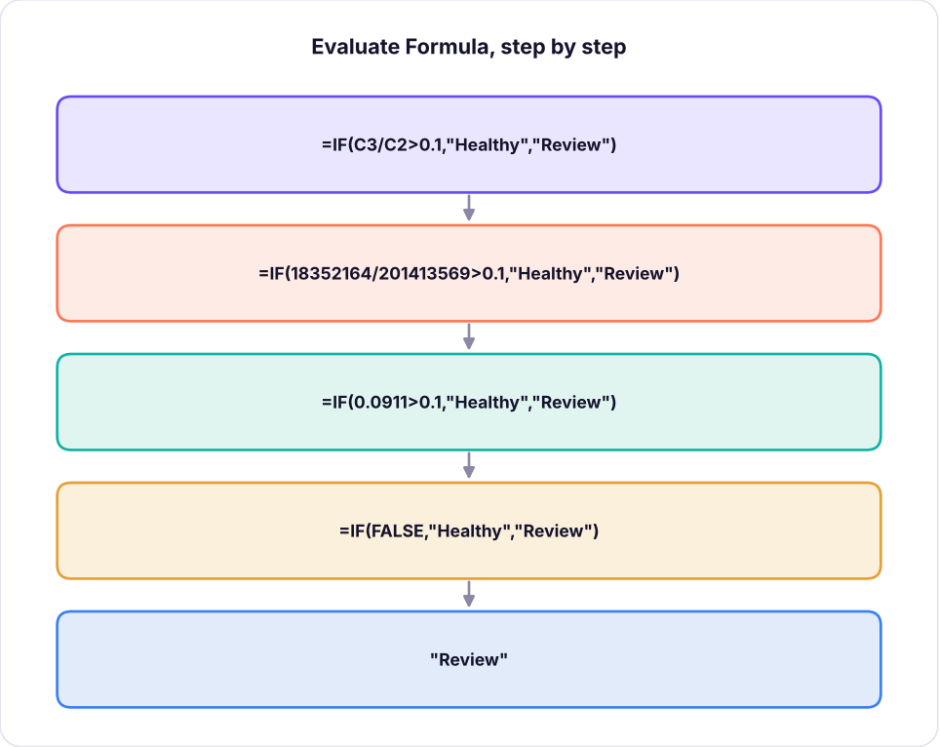
Feature	Insert	Use
Note (old "comment")	Review → Notes → New Note (`Shift + F2`)	A sticky note explaining a cell: "Target approved by CFO on 12-Jan"
Comment (threaded)	Review → New Comment (`Ctrl + Alt + M`)	A conversation: @mention a colleague, reply, resolve

Notes are best for permanent documentation; comments for review discussions. Add a small **"How to use"** note on the dashboard explaining the slicers and data date.

Auditing formulas: Evaluate Formula and friends

When a KPI looks wrong, use **Formulas → Formula Auditing**:

Tool	What it shows
Evaluate Formula	Steps through a formula one calculation at a time: press Evaluate to see each part replaced by its value
Trace Precedents / Dependents	Arrows to the cells a formula uses, or that use it
Show Formulas (`Ctrl + ``)	Every formula instead of its result
Error Checking	Walks through cells with errors and suggests fixes
Watch Window	Keeps an eye on key cells while you work elsewhere
F9 inside the formula bar	Select part of a formula and press F9 to see its value (press Esc to undo)



Evaluate Formula replaces one part at a time until only the result remains.

Delivering the dashboard

- **Refresh**: Data → Refresh All before sending.
- **Export**: File → Export → PDF (select the Dashboard sheet, landscape, fit to one page) for people who only need to read it.
- **Share live**: save to OneDrive/SharePoint and share a link; viewers can use slicers in Excel for the web.
- **Automate**: Power Query + Refresh All, or Office Scripts / Power Automate to refresh and email on a schedule.

DO IT WITH AI **Do it with AI**

Copilot in Excel can draft the analysis: *"Summarise sales and profit by region and category, identify the three biggest loss-making sub-categories, and suggest charts for a dashboard."* Use its suggestions as a starting point, then build the final dashboard deliberately: you are responsible for every number on it.

Insights from the case

Python

```python
sub = s.groupby("sub_category")[["sales", "profit"]].sum()
sub["margin"] = (sub.profit / sub.sales * 100).round(1)
print("Loss-making sub-categories:\n", sub[sub.profit < 0].round(0).to_string())
disc = s.groupby(pd.cut(s.discount, [-0.01, 0, 0.2, 1], labels=["0%", "1-20%", ">20%"]),
observed=False)[["sales", "profit"]].sum()
disc["margin %"] = (disc.profit / disc.sales * 100).round(1)
print("\nMargin by discount band:\n", disc.round(0).to_string())
```

OUTPUT

```
Loss-making sub-categories:
                   sales    profit  margin
sub_category
Bookcases     11274848.0 -422786.0    -4.0
Tables        14046224.0 -816042.0    -6.0

Margin by discount band:
                 sales      profit  margin %
discount
0%        123936290.0  21718963.0      18.0
1-20%      57729097.0    697141.0       1.0
>20%       19748182.0  -4063940.0     -21.0
```

A good dashboard closes with 3–4 written insights, for example:

1. Technology generates about three quarters of sales and most of the profit; furniture is unprofitable overall.
2. Tables and Bookcases lose money; both are heavily discounted.
3. Order lines with discounts above 20% lose money almost every time: review the discount policy.
4. Sales peak every October–December (festive season); plan stock and staffing accordingly.

Summary

KEY TAKEAWAYS

- Start with the brief, KPIs and questions; then data prep; then one pivot per visual.
- Build all pivots from one Table so they share a cache and can be connected to slicers and a timeline (Report Connections).
- KPI cards use GETPIVOTDATA; shapes can display a cell's value by typing `=Cell` in the formula bar.
- Design: top-left headline KPIs, aligned grid, few colours, insight titles, data date and source.
- Protect: unlock inputs, Protect Sheet (allow PivotTables and objects), Protect Workbook structure; use encryption for real security.
- Audit with Evaluate Formula, Trace Precedents/Dependents, Show Formulas and the Watch Window.

Quick quiz

6 questions · tap an answer to check it

1. Why should all dashboard pivots be built from the same Table?

A It makes the file smaller only

B So they share one pivot cache and can all be connected to the same slicers

C Pivot charts require it

D It is needed for conditional formatting

Answer: B. Slicers can only control pivots that share a cache (or data model).

2. After protecting the sheet, slicers stop working. What did you forget?

A To unlock the slicer cells

B To allow "Use PivotTable & PivotChart" and "Edit objects" in Protect Sheet

C To protect the workbook structure

D To refresh the pivots

Answer: B. Slicers are objects that change pivots, so both permissions must be allowed.

3. Which tool lets you see each step of a complex formula's calculation?

- **A** Trace Dependents
- **B** Evaluate Formula
- **C** Watch Window
- **D** Name Manager

Answer: B. Evaluate Formula replaces one part at a time with its value.

4. What does Protect Workbook (structure) prevent?

- **A** Opening the file without a password
- **B** Editing any cell
- **C** Adding, deleting, renaming, moving, hiding or unhiding sheets
- **D** Using slicers

Answer: C. Structure protection guards the sheets, not the cells.

5. How do you count distinct orders in a pivot table?

- **A** Count of order_id
- **B** Add the data to the Data Model and use Distinct Count
- **C** Sum of order_id
- **D** Use a calculated field

Answer: B. Distinct Count is available only for Data Model pivots.

6. What is the difference between a note and a comment in Microsoft 365?

- **A** Notes are simple sticky notes; comments are threaded conversations with replies and mentions
- **B** They are identical
- **C** Comments cannot be printed
- **D** Notes can only be added to formulas

Answer: A. Use notes for documentation and comments for review discussions.

EXERCISE **Build it yourself**

Build the dashboard in this lesson from `superstore_sales.csv`. Then add: (1) a **Ship mode** slicer; (2) a KPI card for **average days to ship**; (3) a chart of the top 10 customers by sales that responds to all slicers; (4) sheet protection that still allows slicers.

Show solution (worked answer)

(1) Insert Slicer → ship_mode → Report Connections → all pivots. (2) Add `days_to_ship` to tblSales; add Average of days_to_ship to the KPI pivot; link a shape to its GETPIVOTDATA cell. (3) New pivot from tblSales: Rows customer_name, Values Sum of sales, Value Filter Top 10; PivotChart (bar); connect to the slicers. (4) Review → Protect Sheet with "Use PivotTable & PivotChart" and "Edit objects" ticked; protect workbook structure.

INTERVIEW CORNER **Questions you may be asked**

Q: Walk me through how you would build an Excel dashboard. Clarify the audience and questions; define KPIs; clean and structure the data as a Table; create one pivot per visual; build pivot charts; connect slicers and a timeline; add KPI cards with GETPIVOTDATA; design the layout; validate numbers against the source; protect and share.

Q: How do you make sure dashboard numbers are correct? Reconcile totals with the raw data (SUM vs pivot), test filters, check distinct counts, use Evaluate Formula and trace precedents for KPI formulas, and have a second person review.

Q: Excel dashboard or Power BI? Excel for small data, quick turnaround and audiences who live in Excel; Power BI for larger data, many users, scheduled refresh, row-level security and richer interactivity.

ADVANCED EXCEL · LESSON 12

Power Query, Power Pivot, Analysis ToolPak & What-If Analysis

Automate cleaning with Power Query (import, transform, unpivot, merge, append, group), model several tables with Power Pivot and DAX, get descriptive statistics and regression from the Analysis ToolPak, plan with Goal Seek, Data Tables, Scenario Manager and Solver, and meet LAMBDA, Python in Excel and macros.

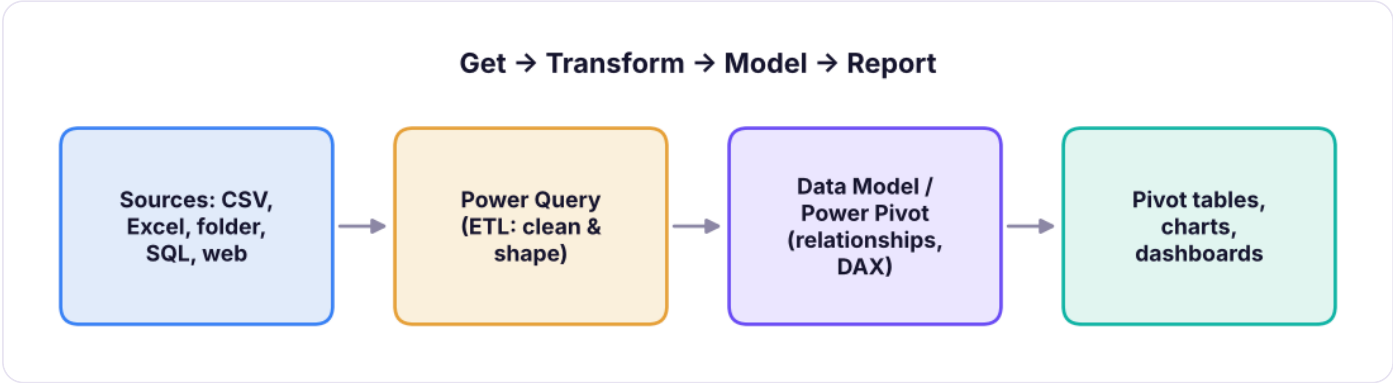
Live class 9 · ✦ Excel for Microsoft 365, 2026 · Advanced · ⏱ 150 min

By the end of this lesson you can

- Build repeatable ETL in Power Query and explain the Applied Steps and M code
- Unpivot wide data, merge (join) and append (union) tables, and group data in Power Query
- Load tables to the Data Model, create relationships and write simple DAX measures in Power Pivot
- Produce descriptive statistics, histograms, correlation and regression with the Analysis ToolPak
- Answer what-if questions with Goal Seek, Data Tables, Scenario Manager and Solver
- Know when to use LAMBDA, Python in Excel, VBA macros and Office Scripts

Why "Power" tools?

Formulas and pivots are great until the same cleaning has to be repeated every month, the data has several related tables, or it grows past a million rows. The Power tools solve exactly these problems, and they are the **same engines used in Power BI**, so everything you learn here transfers directly.



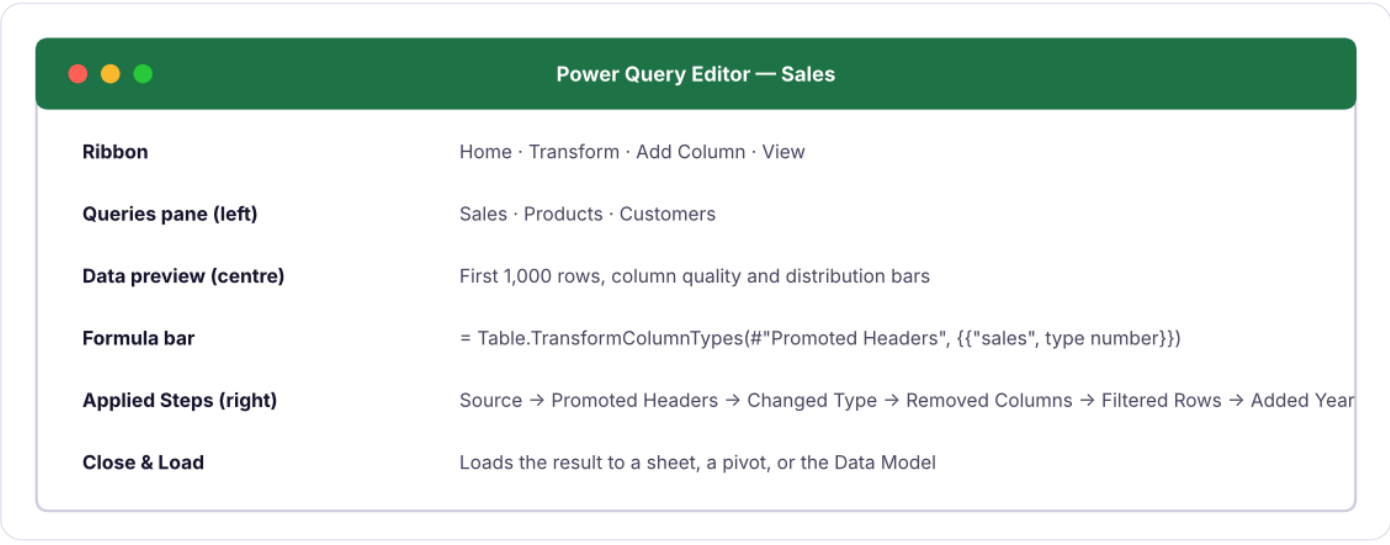
The modern Excel analytics stack. Power Query and the Data Model are shared with Power BI.

Power Query: repeatable cleaning

Power Query (Data → **Get & Transform Data**) imports data and records every cleaning step you click. Next month, put the new file in place and click **Refresh**: all steps run again automatically.

STEP BY STEP **Your first query**

1. Data → **Get Data → From File → From Text/CSV** → choose `superstore_sales.csv` → **Transform Data**. The Power Query Editor opens.
2. Check **data types** in each column header (ABC = text, 123 = whole number, 1.2 = decimal, calendar = date). Fix any wrong ones (right-click header → Change Type).
3. Remove columns you do not need (select → right-click → Remove, or **Choose Columns**).
4. Filter rows with the header drop-downs (for example remove test orders).
5. Add a column: **Add Column → Date → Year**, or **Custom Column** with a formula such as `[sales] - [profit]`.
6. Rename the query (right panel) to `Sales` → **Home → Close & Load To...** → Table, PivotTable report, or **Only Create Connection** + **Add to Data Model**.



The Power Query Editor. Every action you take is recorded in Applied Steps on the right.

Transformations you will use most

Transformation	Where	Example
Remove / keep rows	Home → Remove Rows / Keep Rows	Remove top 3 title rows, remove blank rows, remove duplicates
Use first row as headers	Home → Use First Row as Headers	After importing a report
Split column	Home → Split Column → By Delimiter	`IN-2025-00042` → country, year, number
Replace values	Transform → Replace Values	`Bangalore` → `Bengaluru`
Trim / Clean / Format case	Transform → Format	Same as TRIM, CLEAN, PROPER
Fill down	Transform → Fill → Down	Fill blank category cells (Lesson 2's Go To Special trick, but repeatable)
Conditional column	Add Column → Conditional Column	`if [discount] > 0.2 then "High" else "Normal"`
Column from examples	Add Column → Column From Examples	Like Flash Fill, but repeatable
Group By	Home → Group By	Sales by region (like a pivot)
Unpivot	Transform → Unpivot Columns	Turn month columns into rows
Merge Queries	Home → Merge Queries	A join (lookup) between two tables
Append Queries	Home → Append Queries	Stack tables with the same columns (Jan + Feb + Mar)
From Folder	Get Data → From File → From Folder	Combine every file in a folder automatically

Unpivot: from wide to long

Reports often arrive **wide**: one column per month or per region. Pivot tables, Power BI and Python all prefer **long** data (one row per observation). Unpivot fixes it in one click.

Wide (before): one column per month

	A	B	C	D
1	**Region**	**Jan**	**Feb**	**Mar**
2	North	120	135	150
3	South	98	101	110

STEP BY STEP **Unpivot in Power Query**

1. Load the wide table into Power Query.
2. Select the **Region** column (the one that should stay as it is).
3. Transform → **Unpivot Columns → Unpivot Other Columns**.
4. Rename `Attribute` to `Month` and `Value` to `Sales`.

Long (after): one row per region and month

	A	B	C
1	**Region**	**Month**	**Sales**
2	North	Jan	120
3	North	Feb	135
4	North	Mar	150
5	South	Jan	98
6	South	Feb	101
7	South	Mar	110

"Unpivot **Other** Columns" is better than selecting the month columns, because new months (Apr, May…) added later are unpivoted automatically. The same operation is `melt()` in pandas and `UNPIVOT` in SQL.

Python

```
import pandas as pd
wide = pd.DataFrame({"Region": ["North", "South"], "Jan": [120, 98], "Feb": [135, 101], "Mar": [150,
110]})
wide.melt(id_vars="Region", var_name="Month", value_name="Sales")
```

OUTPUT

	Region	Month	Sales
0	North	Jan	120
1	South	Jan	98
2	North	Feb	135
3	South	Feb	101
4	North	Mar	150
5	South	Mar	110

6 rows × 3 columns

Merge (join) and Append (union)

- **Merge Queries** = a lookup or SQL **JOIN**. Choose the two tables, click the matching key column in each, and choose the join kind: Left Outer (all rows from the first table, matching rows from the second: like

XLOOKUP), Inner (only matches), Full Outer, Left Anti (rows with **no** match: great for finding missing products). Then expand the new column to pick which fields to bring in.

- **Append Queries** = SQL **UNION ALL**: stack January, February and March files that share the same columns.

The M language

Every step is written in **M**, Power Query's formula language. View it in Home → **Advanced Editor**:

Power Query M

```
let
    Source = Csv.Document(File.Contents("C:\Data\superstore_sales.csv"), [Delimiter=",",
Encoding=65001]),
    Promoted = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
    Typed = Table.TransformColumnTypes(Promoted, {{"order_date", type date}, {"sales", type number},
{"profit", type number}}),
    AddedYear = Table.AddColumn(Typed, "Year", each Date.Year([order_date]), Int64.Type),
    HighDiscount = Table.AddColumn(AddedYear, "Discount band", each if [discount] > 0.2 then "High"
else "Normal"),
    Grouped = Table.Group(HighDiscount, {"region", "Year"}, {{"Sales", each List.Sum([sales]), type
number}})
in
    Grouped
```

You rarely type M from scratch, but reading it helps you debug, and you can ask an AI assistant to explain or write a step.

> **TIP** **Change the file path once**
>
> Put the file path in a cell, name it `SourcePath`, and in M use `Excel.CurrentWorkbook(){[Name="SourcePath"]}[Content]{0}[Column1]`. Colleagues can then point the query to their own folder without opening the editor.

Power Pivot and the Data Model

The **Data Model** stores tables inside the workbook in a compressed, columnar engine (VertiPaq, the same engine as Power BI). It handles millions of rows and lets one pivot table use **several related tables**, without VLOOKUPs.

STEP BY STEP **Build a small model**

1. Load `Sales`, `Products` and `Customers` with Power Query → Close & Load To → **Only Create Connection** + **Add this data to the Data Model**.
2. Data → **Manage Data Model** (Power Pivot window) → **Diagram View**.
3. Drag `Sales[product_id]` onto `Products[product_id]` and `Sales[customer_id]` onto `Customers[customer_id]` to create **one-to-many relationships** (Lesson 2 of the SQL track explains why).
4. Insert → PivotTable → **From Data Model**. You can now put `Products[category]` in Rows and `Sales[sales]` in Values: the relationship connects them.

DAX measures in Power Pivot

Measures are formulas that calculate on the fly for whatever is in the pivot (filter context). In Power Pivot → Measures → New Measure (or type in the calculation area):

```DAX
Total Sales := SUM ( Sales[sales] )
Total Profit := SUM ( Sales[profit] )
Profit Margin := DIVIDE ( [Total Profit], [Total Sales] )
Orders := DISTINCTCOUNT ( Sales[order_id] )
Technology Sales := CALCULATE ( [Total Sales], Products[category] = "Technology" )
```

Unlike pivot calculated fields (Lesson 8), `Profit Margin` here is always correct at every level, and `DISTINCTCOUNT` is built in. DAX is covered in depth in the Power BI track (Lesson 4).

Analysis ToolPak: statistics without formulas

The **Analysis ToolPak** is a free add-in: File → Options → Add-ins → Manage: Excel Add-ins → Go → tick **Analysis ToolPak**. It then appears as **Data → Data Analysis**.

Tool	What it produces
Descriptive Statistics	Mean, standard error, median, mode, standard deviation, variance, kurtosis, skewness, range, min, max, sum, count (tick "Summary statistics")
Histogram	Frequency table (and chart) for bins you specify
Correlation	Correlation matrix of several columns
Regression	Linear regression with coefficients, R^2, p-values and residuals
t-Test, z-Test, ANOVA, F-Test	Hypothesis tests (ML track, Lessons 3–5)
Moving Average, Exponential Smoothing	Simple forecasting
Random Number Generation, Sampling	Simulation and sampling

Here is what **Descriptive Statistics** returns for the order-line `sales` column (computed with Python to match the ToolPak's definitions):

Python

```
import numpy as np
from scipy import stats
s = pd.read_csv("superstore_sales.csv")
x = s["sales"]
out = pd.Series({
    "Mean": x.mean(), "Standard Error": x.std() / np.sqrt(len(x)), "Median": x.median(), "Mode":
x.mode().iloc[0],
    "Standard Deviation": x.std(), "Sample Variance": x.var(), "Kurtosis": stats.kurtosis(x,
bias=False), "Skewness": stats.skew(x, bias=False),
    "Range": x.max() - x.min(), "Minimum": x.min(), "Maximum": x.max(), "Sum": x.sum(), "Count":
len(x)}, name="sales")
out.round(2).to_frame()
```

OUTPUT

	sales
Mean	31,854.11
Standard Error	759.76
Median	8,430.00
Mode	420.00
Standard Deviation	60,414.17
Sample Variance	3,649,872,212.91
Kurtosis	16.52
Skewness	3.61
Range	619,248.00
Minimum	42.00
Maximum	619,290.00
Sum	201,413,569.00

13 rows × 1 columns (first 12 shown)

How to read it: the mean (about 31,900) is far above the median (about 8,400), and skewness is strongly positive, so a few very large orders pull the average up. Report the **median** as the typical order line.

The same numbers with Excel functions: `AVERAGE`, `MEDIAN`, `MODE.SNGL`, `STDEV.S`, `VAR.S`, `KURT`, `SKEW`, `MIN`, `MAX`, `SUM`, `COUNT`.

NOTE **ToolPak output is static**

The Analysis ToolPak writes numbers, not formulas. If the data changes, run the tool again. For live results use the equivalent functions.

What-if analysis

Data → What-If Analysis answers "what would happen if..." questions.

Goal Seek: work backwards to a target

"What price gives a profit of exactly ₹ 2,00,000?" Set up the model with formulas, then Data → What-If Analysis → **Goal Seek**: Set cell = profit cell, To value = 200000, By changing cell = price.

X Goal Seek model: which price makes profit = 2,00,000?

	A	B
1	**Input / output**	**Value**
2	Units sold	5000
3	Price	90.40
4	Variable cost per unit	42
5	Fixed costs	42000
6	Profit =B2*(B3-B4)-B5	200,000

```
B3   =50+40.4
B6   =B2*(B3-B4)-B5
```

Goal Seek finds the price ₹90.40: 5,000 × (90.40 − 42) − 42,000 = 2,00,000.

Data Tables: many scenarios at once

A one-variable or two-variable **Data Table** recalculates a model for a list of inputs. Example: profit for prices 80–100 (rows) and units 4,000–6,000 (columns).

STEP BY STEP **Two-variable data table**

1. Put the profit formula in the top-left corner cell of the grid (for example E2: =B6).
2. Prices down the column below it (E3:E7), units across the row to its right (F2:H2).
3. Select the whole grid (E2:H7) → What-If Analysis → **Data Table** → Row input cell = B2 (units), Column input cell = B3 (price).

Python

```python
prices = [80, 85, 90, 95, 100]; units = [4000, 5000, 6000]
grid = pd.DataFrame({u: [u * (p - 42) - 42000 for p in prices] for u in units}, index=prices)
grid.index.name = "Price ↓ / Units →"
grid
```

OUTPUT

	4000	5000	6000
Price ↓ / Units →			
80	110000	148000	186000
85	130000	173000	216000
90	150000	198000	246000
95	170000	223000	276000
100	190000	248000	306000

5 rows × 3 columns

Scenario Manager

Save named sets of inputs (Best case, Base case, Worst case) and switch between them or create a **Scenario Summary** report: What-If Analysis → Scenario Manager → Add.

Solver: optimisation

Solver (another free add-in) finds the best value of a target cell by changing several inputs, subject to constraints: for example, the product mix that maximises profit given limited machine hours and a minimum quantity for each product. Choose Simplex LP for linear problems.

LAMBDA: your own functions

`LAMBDA` (Microsoft 365) turns a formula into a reusable, named function. Define it once in Name Manager:

- Name: `GSTINCL`
- Refers to: `=LAMBDA(amount, rate, amount * (1 + rate))`

Then use it like any built-in function: `=GSTINCL(B2, 18%)`. Helper functions such as `MAP`, `REDUCE`, `SCAN`, `BYROW` and `BYCOL` apply a LAMBDA to every row or element of an array.

Python in Excel

Type `=PY(` in a cell (Microsoft 365, Formulas → Insert Python) and write pandas code. Python runs securely in Microsoft's cloud with Anaconda libraries (pandas, matplotlib, seaborn, scikit-learn, statsmodels):

```python
df = xl("tblSales[#All]", headers=True)
df.groupby("region")["sales"].sum().sort_values(ascending=False)
```

The result can be returned as an Excel value or a Python object, and charts appear as images in the cell. It is a great bridge to the Python track.

Macros, VBA and Office Scripts

Tool	Where it runs	Use for
Macro recorder	View → Macros → Record Macro	Recording repetitive formatting steps; saves as VBA
VBA	Desktop Excel (`Alt + F11`)	Custom automation, user forms; file must be `.xlsm`; macros can carry security risk, so enable only from trusted sources
Office Scripts	Excel for the web (Automate tab), TypeScript	Cloud automation; can be scheduled with **Power Automate**
Power Automate	Cloud	Refresh, email or save reports on a schedule without opening Excel

2026 UPDATE

Excel AI in 2026

Copilot in Excel can generate formulas, highlight patterns, create pivot tables and charts, and run Python analysis from a plain-English prompt. Agentic "office agents" can plan multi-step tasks such as cleaning a file and building a summary. Use them to move faster, and always validate the output (AI for Data Analysis track, Lesson 4).

Summary

KEY TAKEAWAYS

- **Power Query** records cleaning steps and repeats them on Refresh: change types, split, replace, fill down, conditional columns, group by, **unpivot**, **merge** (join) and **append** (union).
- Its language is **M**; read it in the Advanced Editor.
- **Power Pivot / Data Model** stores related tables in a fast columnar engine; relationships replace VLOOKUP; **DAX measures** (SUM, DIVIDE, DISTINCTCOUNT, CALCULATE) compute correctly at every level.
- **Analysis ToolPak**: descriptive statistics, histogram, correlation, regression and tests (static output).
- **What-if**: Goal Seek (one input, target output), Data Tables (grid of scenarios), Scenario Manager (named sets), Solver (optimisation with constraints).
- **LAMBDA** creates custom functions; **Python in Excel** brings pandas into cells; macros/VBA and Office Scripts automate.

Quick quiz

6 questions · tap an answer to check it

1. Which Power Query command turns columns Jan, Feb, Mar into rows with Month and Sales columns?

A Pivot Column

B Unpivot Other Columns (with Region selected)

C Transpose

D Group By

Answer: B. Unpivot converts wide columns into attribute/value rows; "Other Columns" keeps working when new months are added.

2. Merge Queries in Power Query is equivalent to:

A A SQL JOIN or an XLOOKUP for every row

B A SQL UNION

C A pivot table

D Remove Duplicates

Answer: A. Merge joins two tables on matching key columns. Append is the UNION.

3. What is the main benefit of Power Query over manual cleaning?

A It is the only way to remove duplicates

B Steps are recorded and repeated automatically on Refresh

C It uses VBA

D It works only with CSV files

Answer: B. The recorded steps make the process repeatable and auditable.

4. Which tool finds the input value that produces a given result for one formula?

- **A** Goal Seek
- **B** Scenario Manager
- **C** Data Table
- **D** Descriptive Statistics

Answer: A. Goal Seek works backwards from a target output to one input.

5. Why is a DAX measure `DIVIDE([Total Profit],[Total Sales])` better than a pivot calculated field for margin?

- **A** It is faster to type
- **B** It is evaluated correctly in the current filter context at every level and handles division by zero
- **C** Calculated fields cannot divide
- **D** DAX works without a data model

Answer: B. Measures are context-aware and DIVIDE returns blank instead of an error for zero denominators.

6. The Analysis ToolPak reports mean 31,900 and median 8,400 for order values. What does this suggest?

- **A** The data is symmetric
- **B** The distribution is right-skewed by a few large orders
- **C** There is an error in the data
- **D** The mode is larger than the mean

Answer: B. When the mean is much larger than the median, a long right tail is pulling the mean up.

EXERCISE **Power Query practice**

Using `video_game_sales.csv`: (1) load it with Power Query and set the correct types; (2) unpivot the four regional sales columns into `Region` and `Sales`; (3) group by Genre and Region to get total sales; (4) load the result to a pivot table and show % of row total.

Show solution (worked answer)

(1) Data → From Text/CSV → Transform Data; check that year is whole number and sales are decimal. (2) Select every column except na_sales, eu_sales, jp_sales and other_sales (or select those four and choose Unpivot Columns); rename Attribute → Region, Value → Sales; remove the global_sales column first so it is not unpivoted. (3) Home → Group By → Advanced: group by genre and Region, new column Sales = Sum of Sales. (4) Close & Load To → PivotTable Report; Rows genre, Columns Region, Values Sales → Show Values As % of Row Total. You should get the same mix as Lesson 8, Q3.

INTERVIEW CORNER **Questions you may be asked**

Q: What is Power Query and why use it? Excel's (and Power BI's) ETL tool. It connects to many sources, applies recorded, repeatable transformations (types, splits, merges, unpivot, grouping) and refreshes with one click, which reduces manual errors.

Q: What is the difference between Merge and Append? Merge adds columns from another table by matching keys (a join); Append adds rows from tables with the same structure (a union).

Q: What is the Data Model and when do you need it? An in-memory columnar database inside Excel for related tables and large data. Use it when you have several tables, need distinct counts, or exceed normal sheet limits; it also supports DAX measures.

Q: How would you find the break-even price? Build the profit formula and use Goal Seek to set profit to 0 by changing price, or solve algebraically: price = variable cost + fixed costs ÷ units.