

SESSION 2 - CLASS NOTES

Advanced Excel: Repair Layouts, Verify Formulas, Summarise by Criteria, and Master Dates

These notes cover the six topics demonstrated in Session 2, in the order taught. The sequence continues Session 1's method: first repair the layout (Topics 1-2), then verify what the cells really contain (Topic 3), then summarise by conditions (Topic 4), combine text (Topic 5), and finally command Excel's date engine (Topic 6) - the topic we spent the most time on, and the one this document treats in the most depth. Each topic follows the familiar pattern: what it is, why it matters in audit, step-by-step method, a worked example, and a takeaway.

1. Selecting Every Merged Cell at Once (Ctrl+F Method)

What it is

Session 1 explained why merged cells are dangerous: the value lives only in the top-left cell, and the rest are truly empty, so SUMIF, VLOOKUP, sorting and filtering all misbehave. Session 2 completed the cure: a way to find and select every merged cell in a sheet in one action, however many hundreds there are.

How to do it

1. Press Ctrl+F and click Options to expand the dialog.
2. Click the Format button, open the Alignment tab, tick Merge cells, and click OK.
3. Leave the 'Find what' box empty and click Find All. Every merged range in the sheet is listed.
4. Click inside the results list and press Ctrl+A - all merged cells are now selected together in the sheet.
5. Close the dialog and click Home > Merge & Center once: every merged cell in the sheet unmerges in a single action.

Remember to clear the search format afterwards (**Ctrl+F > Format dropdown arrow > Clear Find Format**), otherwise your next ordinary Find will mysteriously return nothing.

Worked example

A client's fixed asset register arrives with the asset category merged across each block of assets - 47 merged ranges scattered over 600 rows. Unmerging them one by one would take twenty minutes and you would miss some. The Ctrl+F format search finds all 47, Ctrl+A selects them together, and one click on Merge & Center dissolves every one. Total time: under thirty seconds.

Takeaway: *Ctrl+F searches by FORMAT, not just by content. Merge cells is a format - so Find All plus Ctrl+A turns a sheet-wide cleanup into one action.*

2. Filling the Blanks Left Behind (F5 > Blanks + Ctrl+Enter)

The problem unmerging creates

Unmerging solves the structural danger but leaves holes: the category name now sits only in the first row of each block, with genuine blanks below it. Formulas still cannot group by category until every row carries its own label. Typing them back manually is out of the question on 600 rows.

How to do it - the two-keystroke fill

1. Select the column range containing the gaps (say A2:A600).
2. Press F5 > Special > Blanks > OK. Only the empty cells are now selected; the first blank is the active cell.
3. Without clicking anywhere, type = and press the Up arrow once. The formula reads, for example, =A2 - 'take the value above me'.
4. Press Ctrl+Enter (not Enter). The same relative formula lands in every selected blank simultaneously, and each one pulls the label down from above.
5. Finally, select the column, copy, and Paste Special > Values, so the labels become permanent and survive sorting.

Why Ctrl+Enter is the hero

Enter fills one cell and moves on. Ctrl+Enter fills the ENTIRE current selection with the entry, adjusting relative references cell by cell. It works far beyond this trick: type any value or formula once and stamp it into a thousand selected cells at once.

Worked example

The unmerged fixed asset register has category names only on the first row of each block. F5 > Blanks selects the 553 empty cells, =Up-arrow + Ctrl+Enter fills all of them in one keystroke, and Paste Special > Values locks them in. The register is now a clean database: every row self-describing, ready for SUMIFS (Topic 4) and PivotTables.

Takeaway: *Unmerge, F5 > Blanks, =above, Ctrl+Enter, then paste as values. Five moves that convert a formatted report back into usable data - learn the sequence as one reflex.*

3. T(), N() and FORMULATEXT() - Interrogating What a Cell Really Holds

What they are

Session 1's ISTEXT and ISNUMBER answer TRUE or FALSE. These three go further - they show you the contents from a particular angle:

Function	What it returns	Audit meaning
T(A2)	The value if it is text; empty ("") if it is not	Anything visible in the result is text - a one-glance text detector
N(A2)	The value if a number; a date's serial number; TRUE as 1; text as 0	Shows the numeric substance behind a cell; text-numbers expose themselves by returning 0
FORMULATEXT(A2)	The cell's formula as a text string; an error if the cell holds no formula	Reveals whether a figure is CALCULATED or TYPED - and exactly how it is calculated

The formula-consistency audit

A schedule may look uniform while one cell in the middle hides a manual override or an edited formula. Put FORMULATEXT to work beside the column:

=FORMULATEXT(D5)	shows D5's formula as text
=FORMULATEXT(D5)=FORMULATEXT(D4)	FALSE flags a break in the pattern

Because relative references change row by row (=B5*C5 becomes =B6*C6), comparing raw formula text of relative formulas flags every row. The practical audit method: apply the comparison down the column and investigate only where the STRUCTURE changes - a hardcoded number, an added '+5000', a reference pointing at the wrong sheet. Where formulas use consistent structure, the eye scans the FORMULATEXT column and the odd one out jumps off the screen.

Worked example

In a depreciation schedule of 80 assets, the total will not reconcile with the client's books by exactly Rs 12,500. A FORMULATEXT column beside the depreciation figures shows 79 rows reading =C15*\$E\$2 style - and one row reading =C47*\$E\$2+12500. Someone 'adjusted' a single asset by hand. T() and N() close the loop on data: a WDV column that looks numeric but where N() returns 0 is text masquerading as numbers.

Takeaway: Figures can be typed, calculated, or text in disguise - and all three look identical on screen. T(), N() and FORMULATEXT() are the interrogation tools that make a cell confess which one it is.

4. SUMIFS and Family - Conditional Totals, Counts and Extremes

The family at a glance

Function	Question it answers	Syntax shape
SUMIF / SUMIFS	Total of amounts meeting condition(s)	SUMIFS(sum_range, range1, criteria1, range2, criteria2, ...)
COUNTIF / COUNTIFS	How many rows meet the condition(s)	COUNTIFS(range1, criteria1, range2, criteria2, ...)
AVERAGEIF / AVERAGEIFS	Average of values meeting condition(s)	AVERAGEIFS(avg_range, range1, criteria1, ...)
MAXIFS / MINIFS	Largest / smallest value meeting condition(s)	MAXIFS(max_range, range1, criteria1, ...) (Excel 2016 and later)

One trap to memorise: SUMIF puts the sum range LAST - SUMIF(criteria_range, criteria, sum_range) - while SUMIFS puts it FIRST. Mixing the two orders is the single most common error with this family. The professional habit: use the plural -IFS versions everywhere, even for one condition, so the syntax is always the same.

Wildcards in criteria

Text criteria accept * (any number of characters) and ? (exactly one character). To search for a literal * or ?, prefix it with a tilde ~.

=SUMIFS(C:C, A:A, "Rent*")	every ledger STARTING WITH Rent (Rent - Office, Rent - Godown)
=SUMIFS(C:C, A:A, "*GST*")	every ledger CONTAINING GST anywhere
=COUNTIFS(B:B, "??-2026")	invoice refs of exactly two characters before - 2026

Logical operators in criteria - and the & rule

Comparisons go inside quotes. The moment the comparison value sits in a cell or comes from a function, the operator stays in quotes and the value is joined with & - this join is the part every beginner misses:

=SUMIFS(C:C, C:C, ">=100000")	hardcoded threshold
=SUMIFS(C:C, C:C, ">="&F1)	threshold kept in cell F1
=SUMIFS(C:C, B:B, ">="&DATE(2025,4,1), B:B, "<="&DATE(2026,3,31))	FY 2025-26 only
=COUNTIFS(A:A, "<>")	non-blank cells

Worked example

From a 4,000-row ledger dump, one formula each answers: total rent paid across all rent ledgers (SUMIFS with "Rent*"), the number of transactions above the Rs 10,000 cash-payment scrutiny threshold in Q4 (COUNTIFS with ">="&10000 and two date bands), the average professional fee per bill (AVERAGEIFS), and the single largest payment to any party containing 'Infra' (MAXIFS with "*Infra*"). Four questions a partner asks daily - four one-line answers, no sorting, no filtering, no helper columns.

Takeaway: Learn one syntax - SUMIFS(answer_range, question_range, question, ...) - and you own eight functions. Wildcards widen the question; ">="&cell connects it to your workings.

5. Four Ways to Join Text - and When Each One Wins

The contenders

Method	Strength	Limitation
A2&" "&B2 (the & operator)	Fastest to type; total control; works everywhere	Every space and separator typed by hand; tedious beyond a few pieces
CONCATENATE(A2," ",B2)	Familiar from old workbooks	Legacy only - cannot accept ranges; kept for compatibility; avoid in new work
CONCAT(A2:D2)	Accepts whole RANGES, not just single cells	No delimiter option - values run together with nothing between them
TEXTJOIN(" ",TRUE,A2:D2)	Ranges PLUS a delimiter PLUS the option to skip empty cells	Needs Excel 2016/365; older machines will not have it

The formatting trap that catches everyone

Joining pulls the UNDERLYING value, not the displayed one: a date joins as its serial number and 1,25,000 joins as 125000. Wrap numbers and dates in TEXT() with the format code you want (Topic 6 lists the codes):

```
= "Reminder: invoice "&A2&" dated "&TEXT(B2,"dd-mm-yyyy")&" for Rs  
"&TEXT(C2,"#,##0")&" is overdue."
```

Worked examples

Building a reconciliation key is the daily use of &: joining ledger code and month as
=A2&" "&TEXT(B2,"mm-yyyy") gives both sides of a books-vs-26AS comparison a common handle for COUNTIFS or lookup matching. TEXTJOIN shines in reporting: with party names in a column,
=TEXTJOIN(" ",TRUE,A2:A40) produces the sentence-ready list 'parties with balances above Rs 5 lakhs' for an audit observation - and the TRUE argument quietly skips every blank row, which CONCAT cannot do.

Takeaway: Use & for keys and short joins, TEXTJOIN for lists with separators, CONCAT only when a bare range join is truly wanted - and wrap every date or amount in TEXT() before it travels into a sentence.

6. The Date Engine - Serial Numbers, Extractors and Working-Day Arithmetic

We spent the most classroom time here, deliberately: almost every professional computation - ageing, interest, due dates, eligibility, delay days - is date arithmetic underneath. Master this topic and the rest of Excel gets easier.

6.1 The concept: dates are numbers, time is the decimal

Excel stores 01-01-1900 as the number 1 and simply counts forward: every date is a serial number, which is why dates can be subtracted, sorted, and averaged. Time is the decimal fraction of one day: 0.25 is 6:00 AM, 0.5 is noon, 0.75 is 6:00 PM. So 46113.5 is a date-time - the integer part is the date, the decimal is the time (recall INT() and MOD(,1) from the pivot practice file). The instant test of any 'date': =ISNUMBER(A2). TRUE means a genuine date; FALSE means text wearing a costume.

6.2 The extractors

=YEAR(A2)	=MONTH(A2)	=DAY(A2)	pull the parts as numbers
=WEEKDAY(A2,2)	1=Monday ... 7=Sunday (type 2 is the intuitive one)		

YEAR, MONTH and DAY return numbers - ideal as helper columns for SUMIFS and pivots.

WEEKDAY's second argument chooses the numbering scheme; type 2 (Monday=1 to Sunday=7) reads most naturally, so =WEEKDAY(A2,2)>5 neatly flags weekend-dated vouchers - a classic audit red flag.

6.3 TEXT() - the date's wardrobe

Code	Shows	Code	Shows
d / dd	5 / 05	m / mm	7 / 07
ddd	Sun	mmm	Jul
dddd	Sunday	mmmm	July
yy / yyyy	26 / 2026	mmm-yy	Jul-26
hh:mm	14:30	hh:mm AM/PM	02:30 PM

=TEXT(A2,"dddd")	the day name - instantly groupable
=TEXT(A2,"mmm-yy")	the month label every MIS report needs

Remember the distinction from Session 1: TEXT() produces TEXT output - perfect for labels, keys and joined sentences, but no longer a date. For calculations keep the real date; for display, either format the cell or dress a copy with TEXT().

6.4 DATEDIF() - the hidden gap-measurer

DATEDIF computes the completed gap between two dates. It is a hidden function - Excel will not autocomplete or tooltip it; type it fully by hand - yet it is indispensable:

=DATEDIF(start, end, "Y")	completed years	"M" completed months
"D" days		
=DATEDIF(start, end, "YM")	months left over after the years	

Coordinator: CA Saran Kumar U (+91 7022345678)

ICAI Hyderabad Branch

ABCD of Microsoft Tools - 6-Day Workshop

=DATEDIF(A2, TODAY(), "Y")&"y "&DATEDIF(A2, TODAY(), "YM")&"m" reads like
7y 4m

Professional uses arrive daily: completed years of service for gratuity eligibility (five completed years - exactly what "Y" measures), asset age for useful-life review, a director's tenure, the age of a receivable in whole months. The start date must not be later than the end date, or DATEDIF returns an error.

6.5 NETWORKDAYS.INTL() and WORKDAY.INTL() - the working-day pair

Two directions of the same idea. NETWORKDAYS.INTL COUNTS the working days between two dates; WORKDAY.INTL FINDS the date a given number of working days away. The .INTL versions let you declare which days are your weekend - essential in India where many offices close only on Sunday:

=NETWORKDAYS.INTL(A2, B2, 11, Holidays) count working days; 11 = Sunday-only weekend
=WORKDAY.INTL(A2, 15, 11, Holidays) the date 15 working days after A2

Weekend code	Meaning
1 (default)	Saturday and Sunday
11	Sunday only
7	Friday and Saturday
"0000011" string	Craft your own: seven digits Mon-Sun, 1 = off-day (0000011 = Sat-Sun off)

The optional last argument accepts a range of holiday dates - keep a named range Holidays listing the year's public holidays once, and every deadline formula in the workbook respects it automatically.

Worked example - the compliance calendar

A due-date tracker built in four columns: notice received date; response due
=WORKDAY.INTL(A2,30,11,Holidays) - thirty working days, Sundays and gazetted holidays excluded;
days remaining =NETWORKDAYS.INTL(TODAY(),B2,11,Holidays); and a status flag that turns red inside five days via conditional formatting. Add =TEXT(B2,"dddd") beside the due date so nobody schedules a filing for a Sunday. Every element of Topic 6 - serials, TEXT codes, and both .INTL functions - working in one small, genuinely useful tool.

Takeaway: A date is a number, time is its decimal, TEXT() is its wardrobe, DATEDIF measures the gap, and the .INTL pair does working-day arithmetic your compliance practice runs on. This topic repays every minute spent on it.

ICAI Hyderabad Branch

ABCD of Microsoft Tools - 6-Day Workshop

Quick Reference Card

#	Feature / Function	Where / How	One-line professional use
1	Select all merged cells	Ctrl+F > Options > Format > Alignment > Merge cells > Find All > Ctrl+A	Unmerge a whole sheet in one action
2	Fill blanks from above	F5 > Special > Blanks, type =[cell above], Ctrl+Enter	Rebuild labels after unmerging; then paste as values
3	T() / N() / FORMULATEXT()	=T(A2), =N(A2), =FORMULATEXT(A2)	Expose text-in-disguise and hand-edited formulas
4	SUMIFS family	SUMIFS(sum_range, range1, crit1, ...); wildcards * ?; ">="&F1	Conditional totals, counts, averages, extremes - one syntax
5	Joining text	& for keys; TEXTJOIN(", ", TRUE, range) for lists; TEXT() inside joins	Reconciliation keys and sentence-ready audit lists
6a	Date extractors	YEAR, MONTH, DAY, WEEKDAY(A2,2), TEXT(A2,"mmm-yy")	Helper columns that feed SUMIFS and pivots
6b	DATEDIF (type by hand)	=DATEDIF(start, end, "Y"/"M"/"D"/"YM")	Gratuity years, asset age, tenure, receivable age
6c	Working-day pair	NETWORKDAYS.INTL / WORKDAY.INTL, weekend code 11, Holidays range	Compliance due dates that skip Sundays and holidays

Practice before Session 3: rebuild the compliance calendar from Topic 6 with your own office's holiday list, and answer three partner-style questions on any ledger dump using only SUMIFS-family formulas. Session 3 moves to Power Query - where everything you repaired by hand in Sessions 1 and 2 (unmerging, filling blanks, fixing text-numbers and text-dates) becomes a one-click refresh. Knowing the manual method first is exactly what will make the automation feel earned.