



From Zero to Excel Master at RPIT

40-question practice course · companion to the 1,000-Student Practice Workbook

rpit.in
youtube.com/@RPITtech
we@rpit.in

This is the same 40-question course RPIT hands every new trainee before their first client visit — built entirely on one linked, three-sheet, 1,000-student dataset. It runs basic to advanced, in order, with a real answer key at the end so you can check your own work as you go. Work through it alongside `Student_Practice_Workbook_1000_Students.xlsx`, open on your own screen.

Concept coverage map

Every core Excel skill this course touches, and exactly which question teaches it — use this table to jump straight to a topic instead of reading start to finish.

CONCEPT	COVERED BY
Sorting, filtering, custom number formats	Chapter 1 (Q1–Q6)
Core functions — SUM, AVERAGE, COUNT, MAX, MIN	Chapter 1 (Q7)
Relative vs. absolute references (\$)	Chapter 1 (Q8)
Basic charts	Chapter 1 (Q10)
Conditional aggregation — COUNTIF/S, SUMIF/S, AVERAGEIF/S	Chapter 2 (Q11–Q16)
Conditional formatting	Chapter 2 (Q17)
Error handling — IFERROR	Chapter 2 (Q18)
Named ranges	Chapter 2 (Q19)
Lookups — VLOOKUP → INDEX/MATCH → XLOOKUP	Chapter 3 (Q21) / Chapter 4 (Q37)
Array logic — SUMPRODUCT	Chapter 3 (Q24)
Data validation with a custom formula	Chapter 3 (Q25)
What-if analysis — Goal Seek, Data Table, Scenario Manager	Chapter 3 (Q26, Q28, Q29)
Text functions — TEXT, LEFT/RIGHT/MID	Chapter 1 (Q9) / Chapter 3 (Q27)
PivotTables, Pivot Charts, Slicers	Chapter 4 (Q31–Q34)
Dashboards, Power Query, Macros	Chapter 4 (Q35–Q38)
Sheet/cell protection	Chapter 3 (Q30)
Cross-sheet, no-helper-column formulas	Chapter 5 (Q39–Q40)

PROLOGUE

0Day 1, 9:30 AM — Orientation

Aditi joins RPIT Computers as a trainee on a Monday morning, one folder of tools and one laptop to her name. No client has asked her for anything yet — that’s the point. Before RPIT sends anyone out to fix a school’s attendance sheet or a shop’s billing register, they hand the trainee the same thing: a 1,000-row student dataset, three linked sheets, and forty questions that get harder one at a time.

“This isn’t a textbook,” her trainer tells her, sliding the workbook across the desk. “Every one of these forty is something a real client has actually

asked us for — sorted, phrased so you can't just wing it, and checkable, so you'll know the moment you get one wrong instead of finding out three weeks later at a client's office.”

The workbook has three sheets that already talk to each other — Marks, Fees and Contact, all keyed to the same Student ID — plus a ReadMe tab explaining the data types, and a hidden Lists sheet holding the dropdowns. Aditi's job for the week: work through all forty questions, in order, without skipping ahead. By Friday, she should be able to open any spreadsheet a client hands her and know exactly where to start.

1 Learning to see the data

“Before you write a single formula, I want you to be able to answer questions about this data using nothing but your eyes and the ribbon,” Aditi’s trainer says. Most client calls RPIT gets aren’t about formulas at all — they’re “my columns won’t line up,” “can you find me the students in Class 10,” or “the pincode keeps dropping its zero.”

Questions 1 through 10 don’t touch a single function. They’re about navigating, filtering, sorting and formatting a 1,000-row sheet the way a client would actually ask for it — plus the basic building blocks (SUM, AVERAGE, COUNT, MAX, MIN) and the one habit almost every beginner skips: locking a cell reference with \$ before it copies somewhere it shouldn’t.

Q1 Freeze the header, then unfreeze it

Freeze Panes AutoFilter Marks

A client once told RPIT their spreadsheet “loses itself” every time they scroll — the header row was frozen and they didn’t know what that meant or how to undo it. Aditi’s first job is to make sure she never sounds confused about that call.

Your task — Open Marks. Row 1 is already frozen (View → Freeze Panes). Unfreeze it, scroll down, notice the headers disappear — then re-freeze it yourself (click any cell in row 2 first, then View → Freeze Panes → Freeze Top Row). Separately, confirm AutoFilter is switched on for row 1; if it isn’t, turn it on from Data → Filter.

Why a client asks this: Freeze Panes and AutoFilter are the two settings every client expects by default and almost never asks for by name — they just say “I can’t see the headers” or “I can’t find anything.”

Check your answer → A1

Q2 Find every Class 10 student

AutoFilter Marks

“How many kids do we have in Class 10?” is the single most common question a school office asks RPIT, usually over the phone, usually while Aditi is mid-lunch.

Your task — Using the AutoFilter arrow on the Class column, filter Marks to show only rows where Class = 10. Read the row count Excel shows at the bottom-left status bar (or in the filter dropdown itself).

Why a client asks this: Filtering answers a one-off question in five seconds without writing a single formula — the fastest tool always wins for a one-time ask.

Check your answer → A2

Q3 Who owes the most?

Sort Fees

Before RPIT builds anyone a dashboard, they ask the same rough-and-ready question a school accountant asks first thing every morning: who’s furthest behind on fees, right now, today.

Your task — On Fees, select the table and sort by column G (Balance Due (Rs.)) from largest to smallest (Data → Sort, or the Z-to-A button with Balance Due selected). Read off the student in row 2 after sorting.

Why a client asks this: Sorting is a one-time snapshot — useful for a quick “top of the list” answer, but the moment someone pays, the sort is stale. That’s the seed for the dashboard Aditi builds in Chapter 4.

Check your answer → A3

Q4 Never lose a leading zero again

Custom Number Format

Contact

A parent's pincode starts with a 0 in some states. The moment that cell is treated as a number instead of protected text, Excel silently eats the zero — and RPIT has had more than one client mail a form back to the wrong region because of it.

Your task — On Contact, select the Pincode column (P2:P1001). Right-click → Format Cells → Custom, and enter the format code 000000 (six zeros). Test it by typing a pincode that starts with 0, like 012345, into a spare cell with the same format applied — it should display all six digits.

Why a client asks this: A custom format changes only how the number displays, not the value stored — which matters, because it means SUMs and lookups on that column still work normally.

Check your answer → A4

Q5 Add a student without breaking the links

Data Entry

Referential Integrity

Marks, Fees, Contact

A new admission walks in mid-term. The school wants that student in the system today, everywhere the existing 1,000 already are.

Your task — Manually add one new row at the bottom of all three sheets (row 1002). Use the same new Student ID (e.g. STU1001) in all three, and fill in realistic values for every column — remember that Fees and Contact pull Student Name and Class from Marks via VLOOKUP, so type the name and class into Marks first, then confirm they appear automatically in the other two sheets once the Student ID matches.

Why a client asks this: This is the exact failure mode that breaks real client workbooks: someone adds a row to one sheet and forgets the other two, and every formula referencing that Student ID downstream returns #N/A.

Check your answer → A5

Q6 Rename a city — and watch something break

Find & Replace

VLOOKUP dependency

Contact

“Can you just change every ‘Bengaluru’ to ‘Bangalore’?” is a completely reasonable client request. What they don't know is that another column is silently depending on the old spelling.

Your task — On Contact, use Find & Replace (Ctrl+H) to change every “Bengaluru” to “Bangalore” in the City column. Then look at the State column for those same rows — does it still show a state, or does it break?

Why a client asks this: Every VLOOKUP in this workbook is only as reliable as an exact text match. This question exists to make that failure visible on purpose, in a safe practice file, instead of in front of a client.

Check your answer → A6

Q7 The five functions everything else is built on

SUM

AVERAGE

COUNT

MAX

MIN

Marks

Before COUNTIFS and SUMPRODUCT, every RPIT trainee has to be fast and correct with the five plain functions — because half of every advanced formula is one of these wearing a condition.

Your task — In a spare cell, try each of these against Marks and record what each returns: =SUM(F2:J2) · =AVERAGE(L2:L1001) · =COUNT(A2:A1001) · =MAX(L2:L1001) · =MIN(T2:T1001).

Why a client asks this: One of these five is a deliberate trap — see below — and it's the single most common beginner mistake RPIT sees in client-built sheets.

Check your answer → A7

Q8 The \$ that saves every formula below it

Absolute Reference

Marks

The single most-repeated correction RPIT's trainer gives new hires: “you copied that formula down and it moved.” This exercise is designed to make that mistake happen once, on purpose, so it never happens on a client file.

Your task — In a blank cell, e.g. W2, type =L2*100 and copy it down a few rows. Watch how the reference shifts to L3, L4... as you go — which is usually exactly what you want. Now, in a different cell, type a formula that should always multiply against the same fixed cell (for example a single tax-rate cell) using \$L\$2, and copy that down instead — confirm it stays locked on L2 the whole way down.

Why a client asks this: Relative references (L2) shift when copied; absolute references (\$L\$2) don't. Most real-world formula bugs are one of these used in the wrong place.

Check your answer → A8

Q9 Pull a region code out of a pincode

LEFT

MID

RIGHT

Contact

A logistics client once wanted a quick “region code” for routing without adding a whole new lookup system — just the first three digits of the pincode, which roughly map to a postal region in India.

Your task — In a spare column on Contact, use =LEFT(P2,3) to pull the first three digits of the Pincode as a “region code”. Then try =MID(P2,4,3) to see the middle three digits, and =RIGHT(P2,2) for the last two, so you can feel the difference between the three functions on the same cell.

Why a client asks this: LEFT/MID/RIGHT are the cheapest way to split apart a piece of structured text (a pincode, an ID, a product code) without a full text-to-columns operation.

Check your answer → A9

Q10 One chart, no PivotTable yet

Column Chart

Marks

A teacher asked RPIT for “just a quick picture” comparing each class's average — not a dashboard, not a report, just something she could screenshot into a WhatsApp message to the principal.

Your task — Without building a PivotTable yet, select the Class and Percentage columns (or a small summary you build with AVERAGEIF per class), and Insert → Recommended Charts → Clustered Column to show average Percentage per Class.

Why a client asks this: Not every chart request needs a PivotTable behind it — sometimes the fastest honest answer is a plain column chart off a small summary table.

Check your answer → A10

2 Making the sheet answer back

By Tuesday, Aditi has stopped scrolling through 1,000 rows to answer a question — she's started making the sheet answer it for her. This is the chapter where COUNTIF, SUMIF and their multi-condition cousins (COUNTIFS, SUMIFS, AVERAGEIF) turn a static table into something a client can actually interrogate: how many failed, how much is overdue, what came in through UPI.

It's also where conditional formatting stops being decoration and starts being an early-warning system, and where IFERROR and named ranges quietly make a workbook safe to hand to someone else.

Q11 How many students failed?

COUNTIF

Marks

The first question every school principal asks after exam season, every single time, without fail.

Your task — In a spare cell: =COUNTIF(Marks!N:N,"FAIL")

Why a client asks this: COUNTIF is the single most-used function in every RPIT client file — one condition, one range, no array thinking required.

[Check your answer](#) → [A11](#)

Q12 How many Class 10 students passed?

COUNTIFS

Marks

A parent-teacher meeting is scheduled for Class 10 only, and the coordinator wants the pass count before she walks in.

Your task — =COUNTIFS(Marks!C:C,10,Marks!N:N,"PASS")

Why a client asks this: The moment a question has two conditions joined by “and”, COUNTIF alone can't do it — that's the signal to reach for COUNTIFS.

[Check your answer](#) → [A12](#)

Q13 Class 8's average percentage

AVERAGEIF

Marks

A Class 8 parent asks how their child compares to the class average — RPIT's client (the school) wants that number ready before they call the parent back.

Your task — =AVERAGEIF(Marks!C:C,8,Marks!L:L). Remember Percentage is stored as a fraction (0.70, not 70), so format the result cell as a percentage to read it correctly.

Why a client asks this: AVERAGEIF is COUNTIF's cousin for averages — same single-condition shape, different aggregation.

[Check your answer](#) → [A13](#)

Q14 Total overdue across the whole school

SUMIF

Fees

The accountant's month-end question: how much money is currently overdue, full stop, before we start breaking it down by class or payment mode.

Your task — =SUMIF(Fees!O:O,"Overdue",Fees!G:G)

Why a client asks this: SUMIF answers exactly the kind of “how much, total” question that shows up in every finance conversation RPIT has with a client.

[Check your answer](#) → [A14](#)

Q15 How much came in through UPI?

SUMIFS

Fees

The office wants to know how UPI collections stack up against cash and cards before deciding whether to push parents toward UPI harder next term.

Your task — =SUMIFS(Fees!F:F,Fees!J:J,"UPI")

Why a client asks this: SUMIFS is SUMIF with room for more than one condition — here there's only one, but the syntax (sum range first, then condition pairs) is what carries forward to every multi-condition total.

[Check your answer](#) → [A15](#)

Q16 Blood group headcount, for the school nurse

COUNTIF

Contact

A genuine, unglamorous client request: the school nurse wants a blood-group headcount on file before the annual health camp.

Your task — For each of the eight blood groups (A+, A-, B+, B-, AB+, AB-, O+, O-), write =COUNTIF(Contact!H:H,"A+") and so on, or build one small summary table covering all eight at once.

Why a client asks this: This is COUNTIF used the way it shows up most often in real client work: not once, but stamped down a short list of categories.

[Check your answer](#) → [A16](#)

Q17 A third early-warning rule

Conditional Formatting

Marks

The workbook already highlights low Percentage and FAIL results — the trainer's brief is simple: “add one more, for attendance, and don't touch the two that are already there.”

Your task — Select the Attendance % column (T2:T1001). Home → Conditional Formatting → New Rule → “Format cells that contain” → Cell Value less than 0.75, orange fill. Confirm it doesn't interfere with the workbook's existing rules on Percentage (a colour scale) and on FAIL rows (a whole-row highlight).

Why a client asks this: Conditional formatting turns a static table into something that flags problems on its own, the moment someone opens the file — no formula-reading required from the client.

[Check your answer](#) → [A17](#)

Q18 Make a lookup fail politely

IFERROR

Fees

A client typed a Student ID with a typo, and the workbook they'd been handed showed a wall of #N/A across the row — which looked broken, even though nothing was actually wrong with the formulas.

Your task — Take one of your Level 2 lookups (or any existing VLOOKUP on Fees, e.g. column B's Student Name pull) and wrap it: =IFERROR(VLOOKUP(...), "Not Found"). Test it by typing a fake ID like STU9999 into a spare row and confirming the wrapped formula shows “Not Found” instead of #N/A.

Why a client asks this: IFERROR doesn't fix a wrong lookup — it makes a correct lookup's failure state readable by someone who isn't you.

[Check your answer](#) → [A18](#)

Q19 Name a range, then use its name

Named Range

Marks

RPIT hands over a lot of workbooks to people who will never learn what $\$L\$2:\$L\1001 means, but who can absolutely read a formula that says `AllPercentages`.

Your task — Select L2:L1001 on Marks, then Formulas → Define Name → call it AllPercentages. Then, instead of the Result column's FAIL count, write a percentage-based version of the same idea using the named range: `=COUNTIF(AllPercentages,"<0.33")` — this counts every student below the 33% pass line directly off the Percentage column, the same threshold the workbook's own Result formula already uses.

Why a client asks this: A named range doesn't change what the formula computes — it changes whether the next person who opens the file understands it at a glance.

[Check your answer](#) → [A19](#)

Q20 Freeze today's numbers before they change

Paste Special — Values

Fees

“What did the overdue balance look like on the 1st of the month?” is a question a school accountant will ask eventually — and `TODAY()`-based formulas can't answer it retroactively, because they only ever show today.

Your task — Copy the Balance Due (Rs.) column from Fees. Paste it into a new sheet using Paste Special → Values only (Ctrl+Alt+V, then Values). Confirm the pasted column is now plain numbers with no live formula behind them.

Why a client asks this: This is how you turn a moving, live number into a permanent “as of today” snapshot for reporting — the pasted values won't change even if the source data does tomorrow.

[Check your answer](#) → [A20](#)

3 Judgment calls

“Two formulas can both return the right answer today and only one of them is still right in six months,” her trainer warns before Chapter 3. This is where Aditi replaces a working VLOOKUP with INDEX/MATCH not because the VLOOKUP was wrong, but because it breaks the moment someone inserts a column in front of it.

It's also where the workbook starts asking her for judgment instead of just syntax: which student to flag, what a struggling student would actually need to pass, and how to stop a colleague from typing an impossible number into the Fee Paid column in the first place.

Q21 Replace a VLOOKUP that a column insert would break

INDEX

MATCH

Fees

“VLOOKUP works until someone inserts a column,” Aditi’s trainer says, “and someone always inserts a column.” This is where she learns the fix that survives it.

Your task — Fees column B currently uses `=VLOOKUP(A2,Marks!$A$2:$B$1001,2,FALSE())` to pull Student Name. Rewrite it as an INDEX/MATCH that returns the exact same result: `=INDEX(Marks!$B$2:$B$1001,MATCH(A2,Marks!$A$2:$A$1001,0))`

Why a client asks this: VLOOKUP's third argument (2) is a hardcoded column position — insert a column into Marks between A and B and every VLOOKUP silently starts pulling the wrong column. INDEX/MATCH points at the Student Name column directly, so it survives that exact kind of edit.

[Check your answer](#) → [A21](#)

Q22 The top student in every class

MAXIFS

Array formula

Marks

Prize-giving season. RPIT's client wants one name per class, and wants it fast, without twelve separate manual lookups.

Your task — Find the highest-Percentage student in each of the 12 classes, either with `=MAXIFS(Marks!$L$2:$L$1001,Marks!$C$2:$C$1001,{class})` for the value, or an array formula combining MAX and IF for the same result on older Excel versions.

Why a client asks this: MAXIFS is the direct, modern answer; understanding the equivalent MAX+IF array logic matters because not every client is on a current Excel build.

[Check your answer](#) → [A22](#)

Q23 Flag students who are older than expected for their class

Logical formula

Contact

A school admin office wants a quiet, non-judgmental flag on students whose age looks out of step with their class — a signal to check the record, not an accusation.

Your task — On Contact, write `=G2>(C2+7)` (Age greater than Class plus 7) and copy it down. See how many students it flags.

Why a client asks this: This kind of “flag anything unusual” formula is common in RPIT's admin-office work — but it only earns its keep if the threshold is actually calibrated to the real data, which this question deliberately tests.

[Check your answer](#) → [A23](#)

Q24 Failing and skipping class — is there a pattern?

SUMPRODUCT

Marks

The vice-principal has a hunch: students who fail are also the ones with poor attendance. RPIT's job is to confirm it with a number, not a hunch.

Your task — =SUMPRODUCT((Marks!N2:N1001="FAIL")*(Marks!T2:T1001<0.75))

Why a client asks this: SUMPRODUCT multiplies two TRUE/FALSE arrays together (TRUE=1, FALSE=0) and sums the result — it's the classic way to count rows matching two conditions across arrays without COUNTIFS' more rigid syntax, and it's the natural bridge into array thinking.

[Check your answer](#) → [A24](#)

Q25 Stop an impossible fee entry before it happens

Data Validation — custom formula

Fees

Someone on the accounts team once typed a Fee Paid amount larger than the Annual Fee itself — a typo, not fraud, but it took a week to notice. RPIT's fix: don't let it happen at all.

Your task — Select Fees!F2:F1001 (Fee Paid). Data → Data Validation → Allow: Custom → Formula: =F2<=\$E2 — this rejects any Fee Paid value greater than that same row's Annual Fee. Add an error message so the accounts team knows why it was rejected. Test it by trying to type a value larger than the Annual Fee into any row.

Why a client asks this: Data Validation with a custom formula checks a rule against other cells, not just a fixed range or list — this is the difference between a dropdown and real business logic.

[Check your answer](#) → [A25](#)

Q26 What score gets a struggling student to 60%?

Goal Seek

Marks

A parent asks the single most common question a teacher gets: “what does my child actually need to score to reach 60%?” Goal Seek answers it without a single manual guess-and-check.

Your task — Pick STU0070 — Trisha Naidu, Class 9 (Math 55, Science 2, English 66, Hindi 75, Social Science 72, currently 54.0% overall). Data → What-If Analysis → Goal Seek: Set cell = her Percentage cell (L), To value = 0.6, By changing cell = her Science cell (G).

Why a client asks this: Goal Seek works backwards from a target instead of forwards from an input — exactly the shape of “what would it take” questions clients ask constantly, and exactly the kind of question a plain formula can't answer directly.

[Check your answer](#) → [A26](#)

Q27 One merged sentence, for a WhatsApp message

TEXT / string concatenation

Contact

RPIT built a client a bulk-WhatsApp reminder tool once, and every message needed to read like a person wrote it — not a mail-merge. It starts with a single concatenated sentence, per row.

Your task — ="Dear Parent, "&D2&", your ward "&B2&" is in Class "&C2

Why a client asks this: String concatenation (&) is what turns columns of raw data back into a sentence a human being will actually read — the last step before any real communication goes out.

[Check your answer](#) → [A27](#)

Q28 What if every student got a discount?

Data Table — What-If Analysis

Fees

The management committee is debating a blanket discount to boost enrollment next year — they want to see the revenue impact at a few different discount levels before they decide.

Your task — Build a one-variable Data Table (What-If Analysis → Data Table) modelling total fee collection (assuming every student paid in full) at discount levels of 0%, 10%, 20% and 30% off the Annual Fee.

Why a client asks this: A Data Table recalculates a formula across a whole list of “what if” inputs in one shot — far faster than manually changing one cell and re-checking the total four separate times.

[Check your answer](#) → [A28](#)

Q29 Best case vs. current reality

Scenario Manager

Fees

The same committee wants a second comparison: not a discount hypothetical, but the gap between what's actually been collected so far and what would happen if every single overdue balance got paid tomorrow.

Your task — Set up two named scenarios in Scenario Manager: “Current” (today's actual Fee Paid total) and “Best Case” (Current + every Overdue balance recovered in full).

Why a client asks this: Scenario Manager, unlike a Data Table, is built for comparing a small number of named, saved states side by side — useful when the “what if” isn't one sliding variable but a specific situation.

[Check your answer](#) → [A29](#)

Q30 Lock the formulas, leave the input open

Sheet Protection

Cell Locking

Marks

A client's data-entry clerk once overtyped a formula cell by accident and didn't notice for two weeks. RPIT's answer: protect the sheet so formula columns can't be touched, but leave the marks-entry columns wide open.

Your task — Select columns K through U (the formula columns: Total Marks through Remarks) and confirm Format Cells → Protection → Locked is checked (it is, by default). Select columns F through J (the marks-entry columns) and uncheck Locked for those. Then Review → Protect Sheet.

Why a client asks this: By default, every cell in Excel is “Locked” — but locking only takes effect once sheet protection is switched on. The real skill is unlocking the specific cells that should stay editable before protecting the rest.

[Check your answer](#) → [A30](#)

4 The client walk-through

Thursday is presentation day. RPIT’s clients — school administrators, coaching-class owners, shop managers — don’t want to see 1,000 rows. They want one sheet that tells them what’s going on, and they want to be able to click a class name and watch it update, without calling Aditi to do it for them.

This chapter is PivotTables, Pivot Charts, Slicers, a real dashboard sheet built from formulas — not typed numbers — and a first, careful look at Power Query and macros: the tools that turn “I know Excel” into “clients trust me with their spreadsheets.”

Q31 The class-by-section performance grid

PivotTable Marks

This is the exact table Aditi’s trainer says every school administrator eventually asks for, in almost these exact words: “can I see average performance by class and section, one table, one screen.”

Your task — Insert → PivotTable from the Marks table. Rows = Class, Columns = Section, Values = Average of Percentage. Format the value area as a percentage.

Why a client asks this: A PivotTable answers a two-dimensional question (class × section) that would need 48 separate AVERAGEIFS formulas to replicate manually — and updates itself the instant the source data changes.

Check your answer → A31

Q32 Which payment mode actually works?

PivotTable Fees

RPIT’s client is deciding whether to keep offering six payment modes or trim the list — they need to know both how much money and how many parents use each one, side by side.

Your task — Insert a second PivotTable from Fees: Rows = Payment Mode, Values = Sum of Fee Paid and Count of Student ID.

Why a client asks this: The most-used payment mode and the highest-earning payment mode aren’t always the same one — this table exists specifically to check whether they line up.

Check your answer → A32

Q33 One chart, four fee statuses

Pivot Chart Fees

Same committee, next slide: they want the Paid / Partial / Pending / Overdue split as one glance-able chart, not a table of numbers.

Your task — From your Fees PivotTable (or a fresh one), add a Pivot Chart (clustered column) showing the count of students in each Fee Status.

Why a client asks this: A Pivot Chart is a normal chart wired directly to a PivotTable — it inherits the same filters and refreshes with the same data, so it’s never out of sync with the numbers behind it.

Check your answer → A33

Q34 Let the client click through classes themselves

Slicer

Fees

The whole point of a dashboard is that the client doesn't call RPIT every time they want to check a different class — they click a button themselves.

Your task — On your Fees PivotTable, Insert → Slicer → Class. Click through a few different classes and watch the Payment Mode summary (or Fee Status chart) update to match just that class.

Why a client asks this: A Slicer is a visible, clickable filter control — the difference between a report someone reads once and a tool someone actually uses on their own.

[Check your answer](#) → [A34](#)

Q35 The one-screen dashboard

Dashboard

Live formulas

New sheet, pulling from Marks + Fees

This is the sheet RPIT actually hands the client. Everything before this question was practice for this one moment: one screen, no scrolling, every number live.

Your task — Build a new Dashboard sheet with, at minimum: total students, overall pass %, total fees outstanding, and a chart of grade distribution — every number driven by a formula referencing Marks/Fees, never typed in by hand.

Why a client asks this: “Driven by formulas, not typed numbers” is the whole test — a dashboard that needs to be manually updated every month isn't a dashboard, it's a chore with extra steps.

[Check your answer](#) → [A35](#)

Q36 The same calculated column, done two ways

Power Query

Marks

RPIT's more technical clients have started asking for Power Query specifically — it's worth showing Aditi the difference in approach, not just the destination.

Your task — Data → Get & Transform → From Table/Range to import the Marks table into Power Query. Add a calculated column there (e.g. a simple Total Marks recomputation) using Power Query's Add Column → Custom Column, instead of a worksheet formula.

Why a client asks this: A worksheet formula recalculates live, cell by cell; a Power Query step is a repeatable, auditable transformation applied once and then refreshed on demand — the right tool changes once a workbook is doing real data-cleaning rather than live calculation.

[Check your answer](#) → [A36](#)

Q37 VLOOKUP's modern replacement, typed fresh

XLOOKUP

Fees or Contact

A client on a current Microsoft 365 subscription asked why RPIT still writes VLOOKUP when their own version of Excel has something newer — fair question, worth answering hands-on.

Your task — If you have Microsoft 365, type a fresh XLOOKUP (don't edit an existing VLOOKUP) that reproduces one of this workbook's lookups — for example, Student Name on Fees: =XLOOKUP(A2,Marks!\$A\$2:\$A\$1001,Marks!\$B\$2:\$B\$1001)

Why a client asks this: XLOOKUP drops the fragile “column index number” VLOOKUP relies on, can search in either direction, and returns a clean, readable #N/A-free result by default with an optional fourth argument — but it's genuinely newer than a lot of installed Excel versions.

[Check your answer](#) → [A37](#)

Q38 Record a one-click Overdue filter

Macro

Fees

A client asked, half-joking, whether Excel could “just remember what I click every morning.” It can — that’s exactly what a macro is.

Your task — Developer → Record Macro. Filter Fees to show only “Overdue” rows (using the Fee Status column). Stop recording. Clear the filter, then run your recorded macro and confirm it reapplies the same Overdue filter automatically.

Why a client asks this: This is the smallest possible macro — one filter action — chosen deliberately, because the value of macros isn’t complexity, it’s turning a repeated manual action into one click.

Check your answer → **A38**

5 The certification challenge

The last two questions get no scaffolding. No column-by-column instructions, no “use this function.” Just a real business question — which struggling students have already paid their full fee, and can we hand a parent one clean printout of their child's record — and three linked sheets to answer it from.

This is the test RPIT actually cares about: not whether Aditi can follow forty sets of instructions, but whether she can look at a messy real question and know which of the last four chapters to reach for.

Q39 Struggling, but already paid in full

Cross-sheet formula

Marks + Fees (no helper columns)

This is the question with no scaffolding: RPIT's client wants a list of students who are failing academically but whose fees are fully paid — good candidates for free extra academic support, since the school isn't chasing them for money. No helper columns allowed, on either sheet.

Your task — On a new sheet, build one formula (or one small formula-driven block) that identifies every student where Marks!Result = "FAIL" AND Fees!Fee Status = "Paid", without adding any helper columns to Marks or Fees themselves.

Why a client asks this: This is what “cross-sheet, no helper columns” actually tests: whether you can combine a condition that lives on one sheet with a condition that lives on another, inside a single formula, instead of reaching for a staging column as a crutch.

Check your answer → A39

Q40 One printout, one student, three sheets

Report layout

VLOOKUP

Marks + Fees + Contact

The final ask, and the one most schools actually pay RPIT for: a parent walks in and wants everything about their child — marks, fees, contact details — on one clean sheet of paper, not three spreadsheet tabs.

Your task — Design a one-page printable report-card layout on a new sheet: one input cell for a Student ID, and every other field pulled automatically from Marks, Fees and Contact using the same VLOOKUP pattern the workbook already uses between sheets — just applied to one student ID at a time instead of a whole column.

Why a client asks this: This closes the loop on everything in Chapters 1–4: it's the same VLOOKUP logic from Question 21, the same string-building from Question 27, applied as a single-record report instead of a bulk table — which is what “knowing Excel” actually looks like in front of a client.

Check your answer → A40

EPILOGUE

✓ Friday, 6 PM — Certified

Forty questions in, Aditi's workbook looks nothing like the clean file she started with — it has her own PivotTables, a dashboard sheet, a protected Marks tab, and a cross-sheet formula she had to think about twice before it worked. That's deliberate. The workbook a trainee ends the week with is never the one they were handed.

This is the same test every RPIT trainee works through before they're sent to a client site. If you're working through it on your own, hold yourself to the same rule Aditi did: don't move to the next question until the answer key confirms the one you're on. Excel punishes shortcuts quietly — a wrong range or a relative reference that should've been absolute won't throw an error, it'll just be wrong, silently, until someone downstream notices.

Finished all forty and checked them against the answer key? You're ready for the next workbook in the RPIT training series — or ready to stop guessing your way through Excel at work.



Answer key — all 40 questions

Every number below was checked against the actual 1,000-row workbook (recalculated, not estimated) — if your result doesn't match, the mismatch is worth chasing down rather than dismissing, since it usually means a range is off by one row or a reference wasn't locked with \$.

Learning to see the data

A1 Freeze the header, then unfreeze it

- Expected result** — Row 1 frozen + filter arrows visible on every column header.
- There's no single "correct value" here — this is a settings check. Confirm both of these before moving on:
 - Freeze Panes: with a cell in row 2 selected, View → Freeze Panes → Freeze Top Row. Correct result: row 1 stays visible no matter how far down you scroll; a thin dark line appears under row 1.
 - AutoFilter: click any cell inside the table, then Data → Filter (or Ctrl+Shift+L). Correct result: every header in row 1 shows a small dropdown arrow.
- ← back to Q1

A2 Find every Class 10 student

- Expected result** — 69 students are in Class 10.
- Filter Class = 10 and read the status bar count (it will say something like "69 of 1000 records found").
- ← back to Q2

A3 Who owes the most?

- Expected result** — STU0401 — Ayaan Thakur owes the most, with a Balance Due of ₹45,000.
- Sort Fees by Balance Due (Rs.) descending. The top row is the highest-balance student in the entire 1,000-row dataset.
 - Remember to re-sort by Student ID (or undo) afterwards if this sheet feeds VLOOKUPs elsewhere that expect the original row order — sorting doesn't break VLOOKUP with an exact-match FALSE argument, but it's good practice to know what moved.
- ← back to Q3

A4 Never lose a leading zero again

- Expected result** — Format code 000000 applied; a typed 012345 stays 012345.
- Format code: 000000
 - None of the 1,000 pincodes in this dataset actually start with a 0 (checked across all rows) — so you won't see a visible change on the existing data, and that's fine. The point of this exercise is that the format is now defensive: type a test value like 012345 into any Pincode cell and it will now display as 012345 instead of silently becoming 12345.
 - Common mistake: leaving Pincode as General format and typing pincodes as numbers. The zero is gone the instant you press Enter, and there's no warning.
- ← back to Q4

A5 Add a student without breaking the links

- Expected result** — STU1001 present and resolving correctly on all three sheets — no #N/A anywhere.
- Correctness check, not a fixed value: after adding STU1001 to Marks with a name and class, go to Fees and Contact, add STU1001 in the next empty row, and copy the VLOOKUP formulas down from row 1001 into row 1002 for the Student Name / Class columns.
 - Confirm the Student Name shown in Fees and Contact for STU1001 matches exactly what you typed into Marks — if it shows #N/A, the Student ID doesn't match character-for-character (a stray space is the usual cause).
 - This is also why VLOOKUP ranges in this workbook are written as \$A\$2:\$B\$1001, not a full-column A:B — a hardcoded row limit like that needs to be extended to row 1002 too, or the new row won't be found by anyone else's formulas.
- ← back to Q5

A6 Rename a city — and watch something break

Expected result — 26 rows switch to #N/A in State after the rename — until Lists is updated too.

- 26 rows in Contact have City = “Bengaluru” before the replace.
- After Find & Replace, all 26 become “Bangalore” — and their State column immediately shows #N/A. The State formula is `=VLOOKUP(N2,Lists!$A$2:$B$36,2,FALSE())`, and the hidden Lists sheet's lookup table still says “Bengaluru”, not “Bangalore” — so the exact match fails.
- The fix: either rename the city back, or open the (hidden) Lists sheet and update its City column to match — a real reminder that Find & Replace on one sheet doesn't know or care what other sheets depend on that exact spelling.

← back to Q6

A7 The five functions everything else is built on

Expected result — SUM=339 · AVERAGE=70.21% · COUNT=0 (use COUNTA for 1000) · MAX=98.2% · MIN=55.8%.

- `=SUM(F2:J2)` → 339 (row 2's Math+Science+English+Hindi+Social Science – this matches the workbook's own Total Marks column exactly, since K2 is built with the same formula).
- `=AVERAGE(L2:L1001)` → 0.7021, i.e. 70.21% across all 1,000 students (format the cell as a percentage to read it properly – Percentage is stored as a fraction, not a whole number).
- `=COUNT(A2:A1001)` → 0. This is the trap: COUNT only counts cells holding numbers, and Student ID (“STU0001” etc.) is text. To count how many student rows exist, use `=COUNTA(A2:A1001)`, which returns 1000.
- `=MAX(L2:L1001)` → 0.982 – the school's top overall percentage is 98.2%.
- `=MIN(T2:T1001)` → 0.558 – the lowest Attendance % in the dataset is 55.8%.

← back to Q7

A8 The \$ that saves every formula below it

Expected result — Relative reference shifts per row; `$L$2` stays locked on row 2 no matter how far you copy it.

- `=L2*100` copied down row-by-row becomes `=L3*100`, `=L4*100`, etc. – each row correctly multiplies its own Percentage, which is what you want when every row needs its own value.
- `=$L$2*100` copied down stays `=$L$2*100` in every row – every row now multiplies the same row 2 value, which is what you want only when you deliberately want one fixed reference (a single rate, a single total) to anchor every row below it.
- The test to internalize: before copying any formula down or across, ask “should this reference move with the copy, or stay put?” — and lock the ones that should stay put with \$.

← back to Q8

A9 Pull a region code out of a pincode

Expected result — `LEFT(P2,3)` = “639” for STU0001's pincode 639904.

- Worked example on row 2, where Pincode (P2) = 639904:
 - `=LEFT(P2,3)` → “639”
 - `=MID(P2,4,3)` → “904”
 - `=RIGHT(P2,2)` → “04”
- Note these all return text, not numbers — if you need to do arithmetic on the result, wrap it in `VALUE()`, e.g. `=VALUE(LEFT(P2,3))`.

← back to Q9

A10 One chart, no PivotTable yet

Expected result — A 12-bar column chart, one bar per class, showing average Percentage.

- Build a small 12-row helper table (Class 1 through 12, each with `=AVERAGEIF(Marks!$C$2:$C$1001,{class}, Marks!$L$2:$L$1001))`), then select it and insert a Clustered Column chart.
- Correct result: 12 bars, one per class, ranging roughly between 61% and 76% average percentage (Class 3 and Class 10 have the widest section-to-section spread — see Question 31's pivot table for the exact numbers by class and section).

[← back to Q10](#)

Making the sheet answer back

A11 How many students failed?

Expected result — 16 students failed (out of 1,000).

- `=COUNTIF(Marks!N:N, "FAIL")`

← back to Q11

A12 How many Class 10 students passed?

Expected result — 66 of the 69 Class 10 students passed (from Question 2, Class 10 has 69 students total).

- `=COUNTIFS(Marks!C:C, 10, Marks!N:N, "PASS")`

← back to Q12

A13 Class 8's average percentage

Expected result — 70.16% — Class 8's average percentage.

- `=AVERAGEIF(Marks!C:C, 8, Marks!L:L)`

← back to Q13

A14 Total overdue across the whole school

Expected result — ₹53,59,900 (Rs. 53,59,900) is currently overdue across the school.

- `=SUMIF(Fees!O:O, "Overdue", Fees!G:G)`

← back to Q14

A15 How much came in through UPI?

Expected result — ₹28,06,800 collected specifically via UPI.

- `=SUMIFS(Fees!F:F, Fees!J:J, "UPI")`

← back to Q15

A16 Blood group headcount, for the school nurse

Expected result — AB+ is the most common (141); B- the least common (112).

- AB+ → 141
- O- → 135
- A+ → 132
- A- → 122
- B+ → 122
- AB- → 119
- O+ → 117
- B- → 112
- Total: 1,000 — confirms every student has exactly one blood group recorded, nothing missing.

← back to Q16

A17 A third early-warning rule

Expected result — 57 rows turn orange (Attendance % below 75%).

- New Rule → Format only cells that contain → Cell Value → less than → 0.75 → orange fill → applies to \$T\$2:\$T\$1001.
- 57 students currently have Attendance % below 75% and will turn orange the moment the rule is applied.
- Careful check: the two existing rules are a colour scale on Percentage (L2:L1001) and a whole-row highlight where Result = "FAIL" (applied to A2:U1001) — your new rule is a separate range (T2:T1001 only), so it will not conflict with either.

← back to Q17

A18 Make a lookup fail politely

Expected result — A missing Student ID shows "Not Found" instead of #N/A.

- `=IFERROR(VLOOKUP(A2,Marks!$A$2:$B$1001,2,FALSE()),"Not Found")`
- Test with A2 = "STU9999" (an ID that doesn't exist in Marks): the plain VLOOKUP returns #N/A; the IFERROR-wrapped version returns the text "Not Found".

← back to Q18

A19 Name a range, then use its name

Expected result — 16 — matches the FAIL count from Question 11 exactly.

- `=COUNTIF(AllPercentages,"<0.33")`
- This returns 16 — exactly matching Question 11's `=COUNTIF(Marks!N:N,"FAIL")` count, because the workbook's Result column is itself built from `=IF(L2>=0.33,"PASS","FAIL")`. Two different formulas, same 16 students, which is a good way to double-check your own work: if these two numbers ever disagreed, one of your formulas would be wrong.

← back to Q19

A20 Freeze today's numbers before they change

Expected result — A frozen, formula-free copy of today's Balance Due column.

- Correctness check: click any cell in the pasted column on the new sheet and look at the formula bar. It should show a plain number (e.g. 19500), not a formula. If it still shows =G2 or similar, you did a regular paste instead of Paste Special → Values.
- This snapshot will now stay exactly as it is even if Balance Due later changes on Fees — that's the entire point.

[← back to Q20](#)

Judgment calls

A21 Replace a VLOOKUP that a column insert would break

- Expected result — Same output as the VLOOKUP it replaced — but immune to a column insert in Marks.
- `=INDEX(Marks!$B$2:$B$1001,MATCH(A2,Marks!$A$2:$A$1001,0))`
 - Confirms identical output to the existing VLOOKUP for every one of the 1,000 rows — this is a like-for-like replacement, not a different calculation.
 - The real-world win only shows up if you test it: insert a blank column into Marks between A and B. The old VLOOKUP (hardcoded to column 2) now returns the wrong data. The INDEX/MATCH version, which points directly at the Student Name column, keeps working. (Undo the column insert afterwards — the rest of the workbook expects Marks' original layout.)

← back to Q21

A22 The top student in every class

- Expected result — 12 names, one per class — school-wide topper is STU0193 Aakash Srivastava (Class 7, 98.2%).
- Class 1: STU0819 — Dhruv Bansal — 94.0%
 - Class 2: STU0820 — Siya Srivastava — 93.2%
 - Class 3: STU0833 — Kartik Ahluwalia — 95.4%
 - Class 4: STU0245 — Dhruv Rawat — 94.8%
 - Class 5: STU0472 — Atharv Nair — 97.6%
 - Class 6: STU0397 — Yash Saxena — 95.8%
 - Class 7: STU0193 — Aakash Srivastava — 98.2% (also the school-wide topper from Question 7)
 - Class 8: STU0320 — Krishna Sen — 93.0%
 - Class 9: STU0418 — Ojasvi Naidu — 95.0%
 - Class 10: STU0303 — Ayaan Joshi — 95.6%
 - Class 11: STU0228 — Ishita Pathak — 93.6%
 - Class 12: STU0753 — Pranav Mishra — 93.4%

← back to Q22

A23 Flag students who are older than expected for their class

- Expected result — Age > Class+7 flags nobody here; Age > Class+5 flags 185 students.
- `=G2>(C2+7)` flags 0 students in this dataset — not a formula error, just a threshold that's too loose for how this data was generated. The largest Age-Class gap in the whole dataset is +6 (Age is at most Class+6 for every one of the 1,000 rows), so a +7 threshold can never trigger here.
 - Try `=G2>(C2+5)` instead, and it flags 185 students — proving the formula logic itself is correct; the first threshold just didn't match this dataset's actual spread.
 - The lesson: a logical formula returning zero matches isn't automatically broken — check whether the threshold makes sense for the data before assuming the formula is wrong.

← back to Q23

A24 Failing and skipping class — is there a pattern?

Expected result — 7 students are both failing and below 75% attendance — a real correlation, not coincidence.

- `=SUMPRODUCT((Marks!N2:N1001="FAIL")*(Marks!T2:T1001<0.75))`
- Result: 7 students are both failing and below 75% attendance.
- Context for that number: 16 students fail overall (Question 11), and 57 have attendance below 75% (Question 17). If the two problems were unrelated, you'd expect the overlap to be small by chance (roughly $16 \times 57 \div 1000 \approx 0.9$ students). Seeing 7 — about eight times that — is a real signal, not noise: it genuinely supports the vice-principal's hunch.

← back to Q24

A25 Stop an impossible fee entry before it happens

Expected result — Rule `=F2<=$E2` blocks any Fee Paid greater than that row's Annual Fee; no existing data violates it.

- Custom formula: `=F2<=$E2` (note the mixed reference — the row number moves with each cell, but the column stays anchored to E so every row checks against its own Annual Fee).
- Checked against all 1,000 existing rows: none of them currently violate this rule — every existing Fee Paid value is already less than or equal to its Annual Fee, so applying this validation won't flag any current data, only future typos.
- Test: try typing 99999 into any Fee Paid cell — Excel should block the entry and show your custom error message.

← back to Q25

A26 What score gets a struggling student to 60%?

Expected result — Science score needs to rise from 2 to 32 for Trisha Naidu to reach 60% overall.

- Trisha's current total is 270/500 (54.0%). To reach 60% overall (300/500), her Science score alone would need to rise from 2 to 32 — a jump of 30 marks, with every other subject held constant.
- This is a genuinely achievable target (32/100 in Science), which is why this particular student was chosen for the exercise — for most of the school's actual FAIL students, reaching 60% would require a Science score over 100 (impossible), because they're behind across every subject, not just one. Try Goal Seek on a FAIL student like STU0198 and Excel will show you exactly that: no valid solution within the 0–100 range.

← back to Q26

A27 One merged sentence, for a WhatsApp message

Expected result — “Dear Parent, Krishna Rawat, your ward Vivaan Rawat is in Class 5”

- `=&"Dear Parent, "&D2&", your ward "&B2&" is in Class "&C2`
- For row 2 (D2 = Father's Name “Krishna Rawat”, B2 = Student Name “Vivaan Rawat”, C2 = Class 5), the result reads: “Dear Parent, Krishna Rawat, your ward Vivaan Rawat is in Class 5”

← back to Q27

A28 What if every student got a discount?

Expected result — Full collection ranges from ₹2,83,86,000 (0% discount) down to ₹1,98,70,200 (30% discount).

- Base figure: SUM of Annual Fee across all 1,000 students = ₹2,83,86,000.
- 0% discount → ₹2,83,86,000
- 10% discount → ₹2,55,47,400
- 20% discount → ₹2,27,08,800
- 30% discount → ₹1,98,70,200

← back to Q28

A29 Best case vs. current reality

Expected result — Current collection: ₹1,81,69,900. Best case (all overdue recovered): ₹2,35,29,800.

- Current: SUM of Fee Paid (Rs.) across all 1,000 students = ₹1,81,69,900.
- Overdue balance that could still be recovered: ₹53,59,900 (this is the same figure from Question 14).
- Best Case: ₹1,81,69,900 + ₹53,59,900 = ₹2,35,29,800.

← back to Q29

A30 Lock the formulas, leave the input open

Expected result — K:U formula columns locked and blocked from editing; F:J input columns stay editable.

- Step order matters: unlock F2:J1001 first (Format Cells → Protection → uncheck Locked), then turn on Review → Protect Sheet. If you protect the sheet before unlocking F:J, every cell — including the ones you need editable — stays locked.
- Correctness check after protecting: try typing into K2 (a formula cell) — Excel should block it with a warning. Try typing into F2 (Math marks) — it should accept the edit normally.

← back to Q30

The client walk-through

A31 The class-by-section performance grid

Expected result — 48 cells (12 classes × 4 sections) — lowest is Class 2 Section B (63.7%), highest is Class 7 Section D (76.6%).

- Average Percentage by Class (rows) × Section (columns) — all 12 classes:
- Class 1: A 67.7% · B 69.9% · C 71.4% · D 69.8%
- Class 2: A 68.0% · B 63.7% · C 71.4% · D 68.8%
- Class 3: A 61.2% · B 74.3% · C 70.5% · D 71.8%
- Class 4: A 69.1% · B 69.4% · C 70.1% · D 67.1%
- Class 5: A 71.9% · B 73.9% · C 66.7% · D 67.2%
- Class 6: A 72.5% · B 72.2% · C 72.9% · D 72.2%
- Class 7: A 74.3% · B 71.2% · C 76.1% · D 76.6%
- Class 8: A 75.1% · B 68.0% · C 72.2% · D 66.0%
- Class 9: A 70.7% · B 71.2% · C 67.6% · D 68.1%
- Class 10: A 68.2% · B 67.3% · C 76.1% · D 65.4%
- Class 11: A 71.8% · B 70.7% · C 69.8% · D 73.8%
- Class 12: A 70.2% · B 66.7% · C 70.1% · D 73.7%

← back to Q31

A32 Which payment mode actually works?

Expected result — Cash is both the most-used mode (189 students) and the highest-collecting (₹34,81,100).

- Cash: ₹34,81,100 collected · 189 students
- Online Transfer: ₹31,00,700 · 175 students
- Debit Card: ₹30,15,000 · 170 students
- Cheque: ₹29,76,200 · 158 students
- UPI: ₹28,06,800 · 163 students
- Credit Card: ₹27,90,100 · 145 students

← back to Q32

A33 One chart, four fee statuses

Expected result — Paid (429) is the largest group; Pending (67) is the smallest.

- Paid: 429 students
- Overdue: 292 students
- Partial: 212 students
- Pending: 67 students
- (Total: 1,000 — every student falls into exactly one status.)

← back to Q33

A34 Let the client click through classes themselves

Expected result — Clicking a class in the Slicer filters every connected table and chart to just that class.

- Correctness check: after inserting the Slicer and clicking, say, “10”, every number in the connected PivotTable (and any Pivot Chart built from it) should recalculate to reflect only Class 10's 69 students — not the full 1,000.
- If the numbers don't change when you click a class, the Slicer likely isn't connected to that PivotTable — right-click the Slicer → Report Connections and confirm the right PivotTable is checked.

← back to Q34

A35 The one-screen dashboard

Expected result — 1,000 students · 98.4% pass rate · ₹1,02,16,100 outstanding · grade split F16/D105/C354/B350/A92/A+83.

- Total students: =COUNTA(Marks!A2:A1001) → 1,000
- Overall pass %: =COUNTIF(Marks!N2:N1001,"PASS")/COUNTA(Marks!A2:A1001) → 98.4%
- Total fees outstanding: =SUM(Fees!G2:G1001) → ₹1,02,16,100
- Grade distribution for the chart: F 16 · D 105 · C 354 · B 350 · A 92 · A+ 83 (totals 1,000).

← back to Q35

A36 The same calculated column, done two ways

Expected result — Power Query's custom column reproduces 339 for row 1, matching Marks!K2.

- Correctness check: in the Power Query editor, Add Column → Custom Column, formula [Math]+[Science]+[English]+[Hindi]+[Social Science], should reproduce the same 339 for row 1 as the worksheet's own Total Marks column (K2).
- The conceptual difference to walk away with: the worksheet formula lives in the cell and recalculates instantly on every change; the Power Query column lives in a query and only updates when you click Refresh — useful when the source data is large, external, or needs several cleaning steps applied in a fixed order.

← back to Q36

A37 VLOOKUP's modern replacement, typed fresh

Expected result — Same Student Name output as the VLOOKUP and INDEX/MATCH versions — but only on Excel 2021+/365.

- =XLOOKUP(A2,Marks!\$A\$2:\$A\$1001,Marks!\$B\$2:\$B\$1001) reproduces the same Student Name result as the workbook's existing VLOOKUP and INDEX/MATCH versions.
- A practical caution worth carrying forward from RPIT's own client work: not every installed copy of Excel supports XLOOKUP (it needs Microsoft 365 or Excel 2021+), so a workbook meant to travel to older installations should stick with INDEX/MATCH from Question 21, even though XLOOKUP is arguably the cleaner formula to write today.

← back to Q37

A38 Record a one-click Overdue filter

Expected result — Running the macro reproduces exactly the 292 Overdue rows, on one click.

- Correctness check: after clearing the filter and running the macro, Fees should show exactly the 292 Overdue rows from Question 33 — the same result as manually filtering, just automated.
- If the macro filters the wrong column or shows an unexpected row count, the most common cause is recording started before clicking inside the actual data table, or the filter was applied to the wrong column header during recording.

← back to Q38

The certification challenge

A39 Struggling, but already paid in full

- Expected result** — Exactly 9 students are failing academically while fully paid up — listed above by ID and name.
- One clean approach, entered as an array formula (Ctrl+Shift+Enter on older Excel, or spills automatically in Microsoft 365), listing every matching Student ID and name into one cell: =TEXTJOIN(", ",TRUE,IF((Marks!N2:N1001="FAIL")*(COUNTIFS(Fees!\$A\$2:\$A\$1001,Marks!A2:A1001,Fees!\$O\$2:\$O\$1001,"Paid")>0),Marks!A2:A1001&" - "&Marks!B2:B1001,""))
 - The full, verified list of matches (9 students, out of 1,000):
 - STU0066 — Imran Reddy (Class 11) · STU0148 — Sai Kulkarni (Class 5) · STU0311 — Karan Jain (Class 5) · STU0336 — Dhruv Khanna (Class 8) · STU0440 — Ishaan Mishra (Class 2) · STU0528 — Ira Mehta (Class 2) · STU0709 — Siddharth Trivedi (Class 10) · STU0715 — Sana Sethi (Class 11) · STU0984 — Zoya Desai (Class 1)
 - That 9-student list is the actual thing to check your formula against — whatever formula shape you chose, it should surface exactly these nine Student IDs and no others.

← back to Q39

A40 One printout, one student, three sheets

- Expected result** — A one-page report that repopulates completely from a single Student ID typed into one input cell.
- Worked example for Student ID STU0001, pulled from all three sheets:
 - Name: Vivaan Rawat · Class: 5-B · Result: PASS (Grade C, 67.8%) · Attendance: 80.4%
 - Annual Fee: ₹26,000 · Fee Paid: ₹6,500 · Balance Due: ₹19,500 · Fee Status: Partial
 - Father's Name: Krishna Rawat · Blood Group: O- · City: Udaipur, Rajasthan
 - Change the input cell to any other valid Student ID (STU0002–STU1000) and the entire report should repopulate automatically — that single-cell-in, full-record-out behaviour is the correctness check for this exercise.

← back to Q40