FOR ACCOUNTANTS, CONTROLLERS AND FP&A

Excel + AI for Finance Teams

Use Claude for Excel and Claude Code to clean, reconcile, model and report

D6 | *fx* =IFERROR((C6-B6)/B6, 0)

	A	B	C	D	E
1	**Account**	**Budget**	**Actual**	**Var %**	**Note**
2	Revenue	4,820	5,104	+5.9%	
3	Cost of sales	2,910	3,288	+13.0%	*explain*
4	Payroll	1,140	1,128	-1.1%	
5	Rent	210	210	0.0%	
6	Marketing	180	246	+36.7%	*explain*
7	Freight	96	131	+36.5%	
8	IT services	74	71	-4.1%	
9	Travel	52	48	-7.7%	

CHANDAN GUPTA

100+ READY PROMPTS

FOR ACCOUNTANTS, CONTROLLERS AND FP&A TEAMS

Excel + AI for Finance Teams

Use Claude for Excel and Claude Code to clean, reconcile, model and report, and keep your spreadsheets auditable

Free sample: the introduction and Chapter 1

CHANDAN GUPTA, CA, CIA

First edition, November 2026

Excel + AI for Finance Teams: Use Claude for Excel and Claude Code to clean, reconcile, model and report, and keep your spreadsheets auditable

Copyright © 2026 Chandan Gupta. All rights reserved. No part of this book may be reproduced, shared or resold in any form without the author's written permission, except for short quotations in reviews. The prompts in this book may be used freely in your own work.

First edition, November 2026.

Not affiliated with Anthropic or Microsoft. Claude and Claude Code are products and trademarks of Anthropic, PBC. Microsoft Excel is a trademark of the Microsoft group of companies; "Excel" is used in this book only to describe the software the reader uses. SAP, Oracle, NetSuite, QuickBooks, Xero and other product names are trademarks of their owners and are mentioned for description only. This book is independent and is not sponsored, endorsed or reviewed by Anthropic, Microsoft or any software vendor. Product features, plans, privacy settings and terms change; check the vendor's current documentation before relying on them.

Not professional advice. This book is for education. Accounting standards, tax rules, banking terms and company policies vary and change, and are mentioned here only to illustrate how to use the tool. Apply your own professional judgment and your company's policies, and review every output before you rely on it, post it or share it. The author accepts no liability for decisions made using this book.

All companies, people, banks, vendors and figures in the examples are made up. Any resemblance to real entities is coincidental.

How this book was made. The author produced it with the same AI-assisted workflow it teaches, and reviewed, edited and checked every chapter. Worked examples in Parts 1 to 3 are illustrative; the walkthroughs in Part 4 were run on the practice files supplied with the book, as each chapter's testing note describes.

Updates and corrections. Claude Code, Anthropic's terms and the software around it all change. If you find an error or something that no longer works, please tell the author (the contact is on the About the author page). Corrections go into the next update and are listed by date on this page.

Contents

How to Use This Book 4

PART 1 · FOUNDATIONS

1 What Leaves Your Laptop 9

About the author 16

IN THE FULL BOOK

How to Use This Book

PART 1: FOUNDATIONS

1 What Leaves Your Laptop
2 Set Up on Windows or Mac
3 Ground Rules for Workbooks
4 Claude Inside Excel, or Claude Code?

PART 2: FINANCE WORK

5 Cleaning Messy Exports
6 Reconciliations
7 Month-End Close
8 Budgets, Forecasts and Cash
9 Auditing an Inherited Model
10 Management Reporting Packs
11 Macros, Power Query or Python

PART 3: WORKING WELL

12 Checking the Output
13 Your Team Playbook
14 Limits, and When to Stay in Excel

PART 4: TRY IT YOURSELF

15 Walkthrough: A Month-End Bank Reconciliation
16 Walkthrough: A Quarterly Budget Versus Actual Pack
17 Walkthrough: Reviewing an Inherited Forecast Model

APPENDICES

Appendix A: The Prompt Library
Appendix B: The Company-Data Checklist
Appendix C: Glossary
Appendix D: Official Sources
About the author

BEFORE YOU START

How to Use This Book

This book shows you how to hand the mechanical part of finance work in Excel to an AI tool that runs on your own computer, keep the company data it sees to a minimum, and get back workbooks you can trace, check and defend.

> SCENARIO · LARKSPUR · FINANCE TEAM · THIRTY DAYS TO DECIDE
>
> An expense review at Larkspur Industrial Supply turns up three different AI chat tools on finance team credit cards. Nobody knows what was pasted into them. Dana Whitfield, the CFO, does not ban them. She gives Priya Raman, the FP&A manager, thirty days to propose a way of working that IT, the auditors and Dana herself can live with.

Who this book is for

You work in corporate finance. You might be a staff or senior accountant, a controller, an FP&A analyst, part of an AP or AR team, or the finance manager who signs off on all of it. You know the general ledger, the close calendar and Excel far better than any software vendor does.

What you lack is time. Too much of the month goes on cleaning exports, matching one list to another, rebuilding the same pack, and checking someone else's spreadsheet before you dare use it. You have heard that AI can help. You have also heard about people pasting payroll files into chatbots, and about confident answers that turned out to be wrong.

This book takes both seriously. It shows you a way of using AI on spreadsheets that keeps most of the data on your laptop, produces new workbooks with real formulas instead of numbers in a chat window, and leaves a trail a reviewer or auditor can follow.

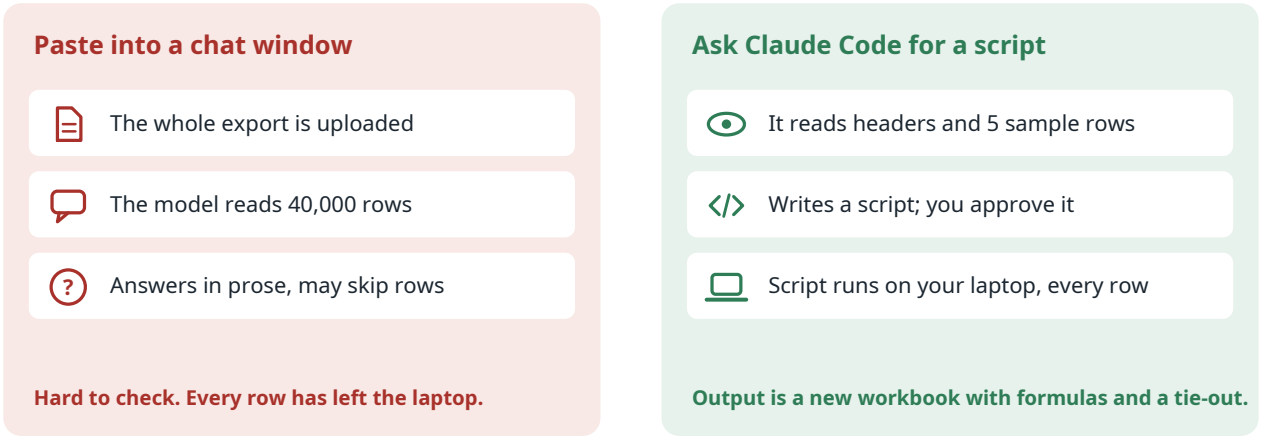
You do not need to code. You describe what you want in plain English. The tool writes the code, shows it to you, and, set up as Chapter 2 shows, asks before it edits a file or runs a script.

What Claude Code is, in plain English

Claude is an AI model made by Anthropic. Most people meet it in a chat window: type a question, attach a file, get an answer.

Claude Code uses the same model differently. It is a program you install on your own computer and start inside a folder you choose. From there it can look at the files in that folder, write new files, and run small programs on your machine, asking your permission as it goes if you set it to (Chapter 2). It runs in a terminal (a plain text window where you type), and also in Anthropic's desktop app.

For spreadsheet work, the key move is this. Instead of reading your 40,000-row export itself, Claude Code looks at the column headers and a few sample rows, then writes a short program, a *script*, that does the cleaning or matching. The script runs on your laptop and writes a new workbook. Only what the script prints, such as counts, totals and a list of exceptions, goes back to the model.



The same job done two ways. The right-hand way sends far less data, checks every row, and leaves you a workbook and a script you can review.

That pattern matters for two reasons. The first is **confidentiality**: customer names, bank details and unreleased results mostly stay on your machine. Chapter 1 is precise about what is sent and where this protection stops. The second is **accuracy**: a script treats row 39,812 exactly like row 12, and reports counts you can tie out. A model reading a huge file in a chat can lose rows or guess. On finance work, the careful way and the accurate way turn out to be the same way.

And Claude inside Excel?

Anthropic also makes **Claude for Excel**, an add-in that opens as a panel inside Excel and works on the workbook in front of you: it explains formulas with cell references, traces errors, changes assumptions while keeping the formulas linked, and tidies sheets. It is a good tool, and many readers will use both. The difference is where the work happens and what is sent. The add-in reads the open workbook's cells, and what it reads goes to Anthropic. Claude Code works on files in a folder by writing scripts, so for big, repeatable or sensitive jobs most of the data stays on your laptop and the script becomes the audit trail.

Reach for	When the job is	For example
Claude in Excel (the add-in)	One workbook you trust, open in front of you; explaining, tracing, small edits, quick what-ifs	"Why is this cell #REF!?" "Walk me through how EBITDA is built."
Claude Code (this book's main tool)	Many files, the same job every month, anything that needs an audit trail, or data too sensitive to send whole	Stacking three ERP exports, a 1,400-line bank rec, checking every formula in an inherited model

Chapter 4 covers the add-in properly: what Anthropic documents it can do, what it sends, and a rule for choosing between the two, job by job.

How the book is organized

The book has four parts. Read Part 1 in order. After that, go straight to the chapters for the work you do.

Part	Chapters	What you get
Part 1: Foundations	1 What Leaves Your Laptop; 2 Set Up on Windows or Mac; 3 Ground Rules for Workbooks; 4 Claude Inside Excel, or Claude Code?	What is sent and what isn't, a working setup, the rules that keep every output checkable, and when to use the Excel add-in instead.
Part 2: Finance work	5 Cleaning Messy Exports; 6 Reconciliations; 7 Month-End Close; 8 Budgets, Forecasts and Cash; 9 Auditing an Inherited Model; 10 Management Reporting Packs; 11 Macros, Power Query or Python	Tasks where it helps, three worked examples each with copy-ready prompts, more prompts to try, and the traps specific to each area.
Part 3: Working well	12 Checking the Output; 13 Your Team Playbook; 14 Limits, and When to Stay in Excel	How to review its work, how to make the whole team work the same way, and where not to use it.
Part 4: Try it yourself	15 Bank Reconciliation; 16 Budget vs Actual Pack; 17 Auditing an Inherited Model	Three end-to-end walkthroughs on practice files with planted traps, an answer key and a self-grading test.

> **Tip.** If you have one hour: read Chapter 1, do Chapter 2, then repeat one worked example from Part 2 on a job you already finished by hand last month. Comparing its answer with yours teaches you more about what it is good at than any amount of reading.

> **From the desk.** Before you start, write down five jobs from last month that were mostly cleaning, matching, merging or re-keying: the bank rec, the intercompany tie-out, the BvA refresh, the vendor statement that never agrees. That list is your practice plan. Start with the smallest.

How to read a worked example

Every Part 2 chapter is built around worked examples, all in the same shape:

1. **The situation**
   Who is doing the job, the entity and period, the files, and what has to come out.

2. **The prompt**
   The exact request to type into Claude Code, in a box labelled **Type this**. Change the file, sheet and column names to match yours. Keep the instruction to use a script that runs locally whenever the data belongs to your company.

3. **What Claude Code does**
   Three to five steps describing what you should see: what it looks at first, what it asks permission for, and what it produces. Your screen will not match word for word; it phrases things differently each time.

4. **What to check**
   A tick-box list of the checks to do before you rely on the result: totals that must tie, rows to find in the source file, rules to confirm in the script.

Grey terminal panels show what a session looks like. Lines starting with `>` are what you type.

> **WHAT A PROMPT BOX LOOKS LIKE**
>
> Compare `Bank_Jun26.csv` with `NS_Cash_GL_Jun26.xlsx` on date, amount and check number. Look only at the headers and a few rows first. Then write a script, run it here, and show me only the counts and the unmatched items.

> **Read this once.** **Illustrative or tested?** Worked examples in Parts 1 to 3 are labelled *illustrative*. Their prompts are written to work, but the files, figures and on-screen results are representative; they were not produced by running that exact prompt on real data. The three walkthroughs in Part 4 are labelled *tested*: each was run on the practice files that come with the book, and its "How this was tested" note says exactly what was run, when, and how it scored.

The people in this book

Each chapter opens with a short scenario from Larkspur Industrial Supply, a made-up distributor of industrial parts based in Columbus, Ohio, with about $140 million in revenue and subsidiaries in Canada and the UK. Its finance team spends the year learning to use Claude Code. You will meet:

Who	Role	What to watch for
Dana Whitfield	CFO	Precise and skeptical. Protects the company's credibility with the bank and the auditors.
Marcus Bell	Controller. Owns the close.	The early adopter. Quick to try things, sometimes quick to trust them.
Priya Raman	FP&A Manager	Methodical. Becomes the team's Claude Code person.
Jake Ortiz	Staff Accountant: bank recs, AP, fixed assets	Bright and overloaded. Probably the closest to you.
Lena Kowalski	Senior Accountant: AR and intercompany	The Excel expert with the macro nobody else understands. Not sold on Python.
Sam Okafor	IT Manager	Wants to know exactly what leaves each laptop.

Conventions

- Every company, person, bank, vendor and figure in the examples is made up.
- File names look like `Budget_FY26.xlsx`. Commands you type look the same.
- Paths are shown for Windows (`C:\Finance\Jobs\...`). On a Mac, use a folder in your home directory such as `~/Finance/Jobs/...`; Chapter 2 covers both.
- Claude Code changes often. Menus, messages and option names may differ slightly from what is shown. Where a setting matters for confidentiality, the book tells you how to confirm it yourself.
- Accounting policies, thresholds and approval rules are your company's. The book uses examples such as a $25,000 flux threshold to show the method, not to recommend a number.

A note before you start

> **Independence and advice.** This book is independent. It is not affiliated with, sponsored by or endorsed by Anthropic or Microsoft. Claude and Claude Code are Anthropic's products, and Anthropic's own documentation, terms and privacy policy are the authority on how they work. Nothing here is accounting, tax, legal or audit advice. It describes ways of working that you remain responsible for, under your company's policies and controls.

CHAPTER

1

What Leaves Your Laptop

Before Claude Code touches a company workbook, you should know what it sends, where it goes, and how to keep it to the minimum the job needs. This chapter gives you the facts and a routine you can explain to IT and to your auditors.

> SCENARIO · LARKSPUR · JUNE AR AGING · COLLECTIONS CALL AT 9 AM
>
> Jake Ortiz needs past-due customers from the June AR aging (2,400 customers, names and balances) for tomorrow's collections call. At 9 pm he nearly pastes the whole sheet into a free chatbot, then stops. Next morning Sam Okafor from IT asks the question nobody on the team can answer: what actually leaves the laptop?

Jake did the right thing by stopping, but he could not have said why. Most finance people can't. AI tools are sold on what they can do, not on where your data goes when they do it.

This chapter answers Sam's question for Claude Code. It explains what the tool sends to Anthropic, what stays on your machine, and the one habit that keeps the amount sent small: asking Claude Code to write a script that runs on your laptop instead of reading your file. It ends with a routine you can follow on every job, and the conversation to have with IT and your auditors before you start.

What counts as company data

In a finance team, nearly everything in a working folder is confidential to someone. It helps to name the categories, because they carry different risks:

Customer and vendor data
Names, contacts, balances, pricing, credit limits. Often covered by contracts as well as your own policy.

Payroll and people
Salaries, bonuses, tax IDs, bank details, performance notes. Personal data, with privacy law behind it.

Bank and treasury
Account numbers, routing details, statements, payment files. A fraud risk if they travel.

Unreleased results
This month's numbers before the board, the lender or the owners have seen them.

Deals and plans
Acquisitions, financing, restructuring, pricing changes. Often under an NDA.

Credentials
Passwords, API keys, ERP logins, saved in a text file "for now". These never belong in a working folder.

None of this means you can't use AI on finance work. It means you decide, job by job, what the tool needs to see, and you arrange the work so it sees that and little else.

What Claude Code sends, and what it doesn't

Claude Code is a program on your computer, but the thinking happens on Anthropic's servers. Each time you type a request, Claude Code sends the conversation so far to Anthropic and gets the next step back. The conversation includes:

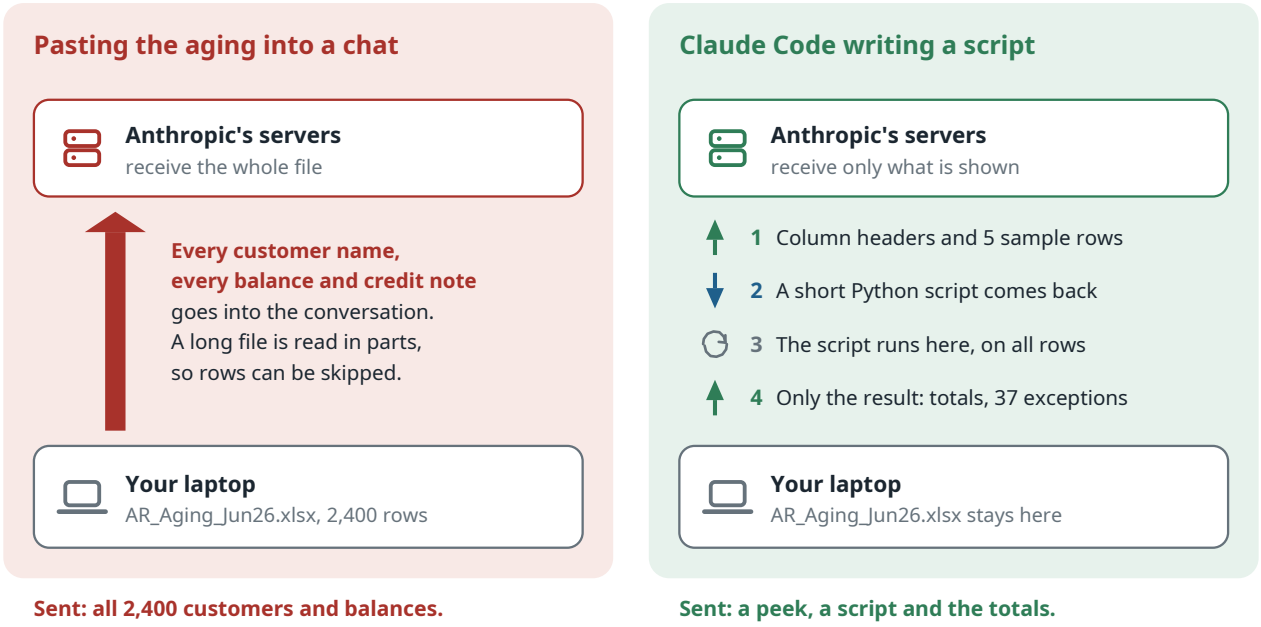
Your prompts, everything you type, including anything you paste.

Anything it reads. If it opens a file, the part of the file it read goes into the conversation. If it runs a command or a script, whatever that prints goes into the conversation too.

Its own replies, including the scripts it writes.

The traffic is encrypted on the way. What does *not* travel is everything in the folder that Claude Code never opens, and the full data that a script processes on your machine without printing. That gap is what you use.

One more point: Anthropic also offers Claude Code sessions that run on its own machines, on copies of your files held there. This book always means Claude Code running on your own computer. Do not use the cloud version for company data.



The same task, two ways. Pasting sends every row. A script sends a peek, the script itself, and only what the script prints.

The local-script pattern

Here is the habit the whole book builds on. Instead of asking Claude Code to read your file and answer, you ask it to do three things:

1 **Look at the shape, not the data**
Print the sheet names, the column headers and a handful of rows, so it can see how the file is laid out.

2 **Write a script**
A short Python program that does the whole job: filters, matches, totals, builds the output workbook.

3 **Run it here and report the result**
The script runs on your laptop over every row. Claude Code sees only what the script prints: counts, totals and the exceptions you asked for.

For Jake's aging, the conversation would carry a header row, five sample customers, a 40-line script and a summary such as "2,400 customers, $18.2 million open, 37 accounts over 90 days, list saved to `Past_Due_Jun26.xlsx`". The other 2,363 customers never leave the laptop.

The pattern has a second benefit that matters as much as the first. A script processes every row the same way and can prove it did, with control totals. A chat model reading a long file in its head can skip or misread rows without telling you. On finance work, the careful way and the accurate way are the same way.

> **THE PATTERN IN ONE PROMPT**
>
> AR_Aging_Jun26.xlsx is our June AR aging. Don't read the file into our chat. First, write and run a short script that prints only the sheet names, the header row and the first 5 rows of the main sheet. Then

> write a Python script that runs here and saves Past_Due_Jun26.xlsx with every customer whose balance over 90 days is above $5,000: customer, total balance, over 90, last payment date. Show me only the count of customers, the total open balance and the total over 90 days, so I can tie them to the aging's grand total.

Watch out. The script pattern reduces what is sent. It does not make it zero. The sample rows are real data, an exception list can name customers, and if Claude Code decides to open a file to debug something, that content goes too. Read what it is about to do before you approve it, and mask names in the peek when the task doesn't need them.

Plans and settings that matter

How Anthropic treats what you send depends on your plan. As of this writing, Anthropic's documentation describes two groups:

	Consumer plans (Pro, Max, Ultra)	Commercial plans (Team, Enterprise, API)
Training on your data	Your choice. A privacy setting controls whether your chats and sessions may be used to improve models. Turn it off for work use.	Not used for training by default, under commercial terms.
Retention	Differs by setting. Check the current privacy terms.	Differs by plan; zero data retention is available to qualified Enterprise accounts.
Admin control	You control your own account.	An admin can manage users and settings for the whole team.
Fits	One person trying it on made-up data.	A finance team using it on company files.

Plan names, terms and retention periods change, so treat the table as a map and check the current documentation before you decide. Three settings are worth knowing on any plan:

- **The training setting** on consumer plans, at claude.ai under Settings, then Privacy (the "Help improve" model-training toggle). Make sure it says what your company has decided.
- **Feedback.** The `/feedback` command sends a copy of the conversation to Anthropic. Turn it off with the `DISABLE_FEEDBACK_COMMAND` setting (Chapter 2) so nobody sends a session full of company data by accident.
- **Local transcripts.** Claude Code keeps session transcripts on your own computer, in plain text, for a period you can change. They include everything the session saw. Treat that folder like the working files themselves.

What never goes in the folder

Claude Code can read any file in the folder where you start it, and the folders below it. So the folder is your main boundary, though not a sealed one: by default some simple read-only commands can still look outside it without asking, and Chapter 2 shows the setting that stops that. Keep the folder small and boring:

Keep in the job folder	Keep out of it
Copies of the exports this job needs, for this period	The whole shared drive, or your Documents or Desktop folder
Scripts and output workbooks for this job	Other entities' or other months' files you don't need
A short `CLAUDE.md` with job notes (Chapter 13)	Passwords, API keys, ERP logins, bank login details
Masked or trimmed extracts where the job allows	Full payroll with tax IDs and bank accounts when you only need department totals

Don't. Never start Claude Code in your home folder, on a mapped network drive, or in the folder that syncs the whole finance share. Make a job folder, copy in what the job needs, and start there. Chapter 2 shows the layout.

Talking to IT and the auditors

Sam's question deserves an answer in writing before the team starts. A one-page note covers it, and it saves you from explaining the same thing in every audit.

DRAFT THE NOTE FOR IT AND THE AUDITORS

Draft a one-page note for our IT manager and our external auditors on how the finance team will use Claude Code. Cover: which plan we use and its training and retention terms (leave placeholders for me to fill from Anthropic's current documentation), the job-folder rule, the local-script pattern, which data categories we exclude, who reviews outputs, and what we keep as support for each job. Plain English, no marketing tone. Save as AI_Use_Note_Draft.docx.

Expect three questions back. IT will ask about the plan and the settings, and whether they can enforce them centrally (on Team and Enterprise they largely can). The auditors will ask how you know the output is right, which is what Chapter 12 is for, and what evidence you keep. Your answer to the second: the inputs as received, the script, the output workbook with its tie-out sheet, and the reviewer's initials. Dana Whitfield, Larkspur's CFO, put the standard more briefly when she signed off the team's note: "If it doesn't tie, it doesn't leave the building."

A different tool. Anthropic also offers Claude for Excel, an add-in that works inside an open workbook. It is a different tool with different trade-offs: it reads the workbook content it works on and sends that to Anthropic, and its own documentation lists limits such as no macro support. This book is about Claude Code and the script pattern; Chapter 4 explains the add-in and when to use which. Apply the same thinking about what a workbook contains before you open it in either.

A routine for every job

BEFORE, DURING AND AFTER

- [ ] Your plan's training and retention settings are what your company has decided, and you checked them this quarter.
- [ ] You start Claude Code in a job folder that holds only this job's files.
- [ ] Credentials, full payroll and anything under an NDA the job doesn't need are not in the folder.
- [ ] Your first prompt asks for headers and a few rows, and a script that runs here.
- [ ] You read each permission prompt before saying yes, and you know which file it is about.
- [ ] The script's totals tie to the source before you use any output.
- [ ] When the job is done, the support is filed and scratch copies are deleted.

CHECK THE FOLDER BEFORE YOU START

Before we start: list every file in this folder and below it, with size and type, but don't open any of them. Tell me if anything looks like it shouldn't be here for a bank reconciliation (payroll, credentials, other entities, other periods).

SUMMARIZE WITHOUT EXPOSING PEOPLE

Payroll_Jun26.xlsx has one row per employee. I only need department totals. Write a script that runs here and saves Payroll_Dept_Jun26.xlsx with department, headcount, gross pay, employer taxes and benefits. Don't print any employee-level rows to our chat, and don't copy the names or tax IDs into the output.

From the desk. Keep a sanitized copy of a typical export, with made-up names and rounded amounts, in a practice folder. When you want to try a new prompt, try it there first. You learn how Claude Code behaves without any real data at stake.

Key takeaways

- Claude Code sends your prompts, its replies, and anything it reads or a command prints. Files it never opens stay on your laptop.
- Ask for headers and a few rows, then a script that runs locally and prints only totals and exceptions. That cuts what is sent and checks every row.
- Your plan decides training and retention terms. Check the current settings; don't rely on memory or on this book.
- The job folder is your boundary. Put in only what the job needs; keep credentials out always.
- Write a one-page note for IT and the auditors before the team starts, and keep support for every job.

ABOUT THE AUTHOR

Chandan Gupta

CA, CIA. An auditor who has spent years inside other people's spreadsheets, and now builds tools to check them.

Chandan is a Chartered Accountant (India) and a Certified Internal Auditor (The Institute of Internal Auditors), with experience in statutory audit, internal audit and internal controls over financial reporting.

Most of my working life has been spent on the receiving end of finance teams' spreadsheets: the reconciliation that supports a balance, the model behind a forecast, the export that is supposed to be complete. I have seen how much good work goes into them, and how often one hardcoded number or one range that stops a row short undoes it.

So I started building tools that do the mechanical checking for me: loading a year's ledger from an ERP export and proving it is complete, testing every journal entry instead of a sample, turning messy exports into lists a reviewer can act on, matching vendor balances. Almost all of it runs offline, because I am not comfortable with company data leaving the machines it belongs on.

Claude Code is how I build them now. I describe what has to be proved and from which data, it writes the code, and I test and review the result the way I would review a junior's work. Some tools worked the first time. Many took a third or fourth round. This book comes out of those rounds, mistakes included, rewritten for the people who prepare the spreadsheets rather than the people who audit them.

> **From my weekly series on internal audit.** A signature is evidence of a signature, not automatically evidence of a control.

I wrote that about controls, and it applies to AI just as well. An AI output is evidence that something produced an output. Whether it is right is still your call, and the routines in these pages exist to help you make that call quickly, not to skip it.

If this book saves you a late night at month end, or falls short on your files, I would like to hear about it, and what you would want in the next edition. I'm on LinkedIn: https://www.linkedin.com/in/chandan-gupta-b8a108192

Get the full book

This sample has the introduction and Chapter 1. The full **Excel + AI for Finance Teams** adds setup on Windows and Mac, the ground rules for workbooks, seven chapters of finance work (cleaning exports, reconciliations, month-end close, budgets and cash forecasts, auditing inherited models, reporting packs, and macros versus Power Query versus Python) with worked examples, chapters on checking output and building your team playbook, three tested walkthroughs on downloadable practice files, and an appendix of every prompt in the book.

> ✓ **Get it.** Available on Amazon Kindle.