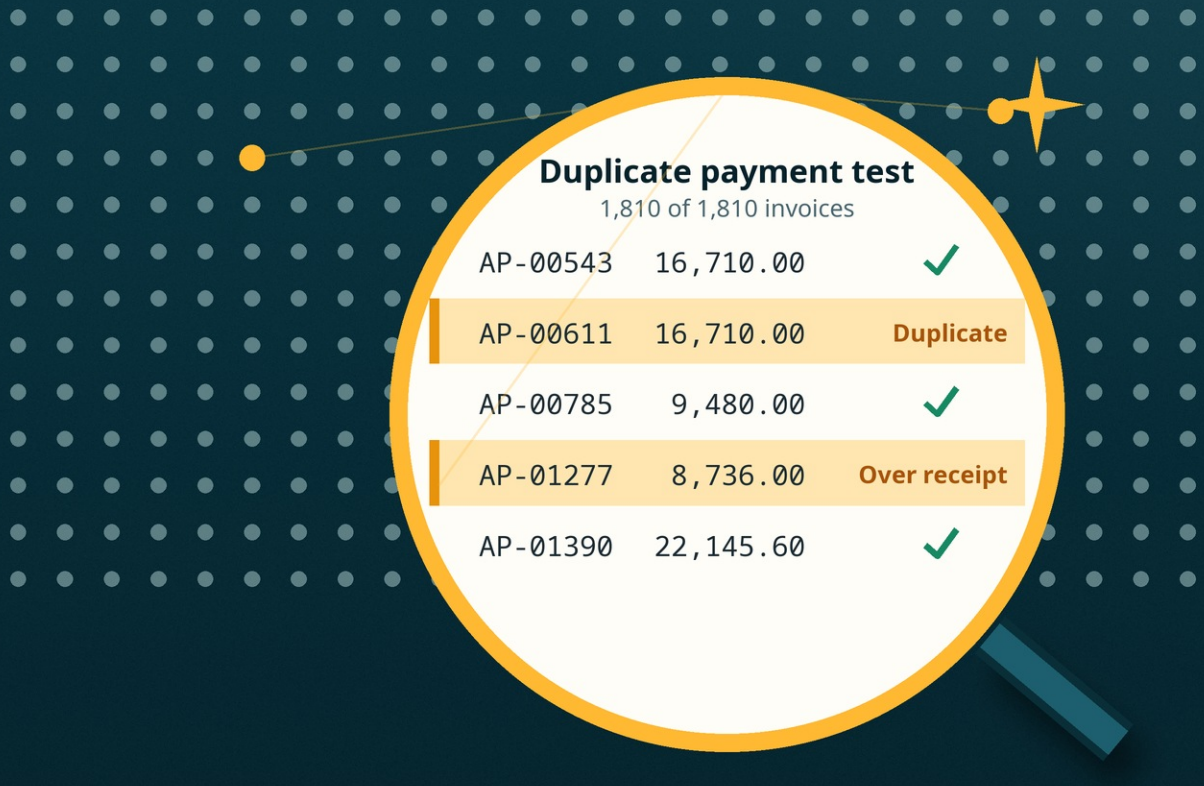
THE 'AI FOR' SERIES

CA-LED · AI-ASSISTED

FOR INTERNAL AUDITORS AND SOX TEAMS

AI for Internal Auditors

A practical guide to Claude Code for risk-based audits, testing and reporting



CHANDAN GUPTA

100+ READY PROMPTS

FOR INTERNAL AUDITORS AND SOX TEAMS

AI for Internal Auditors

A practical guide to Claude Code for risk-based audits, control testing, data analytics and reporting

Free sample: the introduction and Chapter 1

CHANDAN GUPTA, CA, CIA

First edition, November 2026

AI for Internal Auditors: A practical guide to Claude Code for risk-based audits, control testing, data analytics and reporting

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 The IIA. Claude and Claude Code are products and trademarks of Anthropic, PBC. This book is independent and is not sponsored, endorsed or reviewed by Anthropic. It is also not published, endorsed or reviewed by The Institute of Internal Auditors (The IIA) or COSO; CIA is a certification of The IIA. Product features, plans, privacy settings and terms change; check Anthropic's current documentation before relying on them.

Not professional advice. This book is for education. Standards, frameworks, regulations and your organization's methodology are summarized here only to illustrate how to use the tool. Always check the current standards and your own internal audit methodology, apply your professional judgment, and review every output before relying on it or sharing it. The author accepts no liability for decisions made using this book.

All companies, people, systems 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 professional standards 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 18

IN THE FULL BOOK

How to Use This Book

PART 1: FOUNDATIONS

1 What Leaves Your Laptop
2 Get Approved and Set Up

PART 2: THROUGH THE AUDIT LIFECYCLE

3 Risk Assessment and the Audit Universe
4 Planning and Scoping an Engagement
5 Walkthroughs, Narratives and Flowcharts
6 Control Testing and Sampling
7 Full-Population Analytics: Purchasing, Sales and Payroll
8 Full-Population Analytics: Journals, Expenses and Access
9 Fraud Red Flags
10 Working Papers and Evidence
11 Writing Observations That Get Fixed
12 Audit Reports and Audit Committee Summaries
13 Follow-Up and Issue Tracking

PART 3: WORKING WELL

14 Checking What It Gives You
15 Your Department's Playbook
16 Limits, and When Not to Use It

PART 4: TRY IT YOURSELF

17 Walkthrough: Duplicate Payments and the Three-Way Match
18 Walkthrough: User Access and Segregation of Duties

APPENDICES

Appendix A: The Prompt Library
Appendix B: The Audit Data Checklist
Appendix C: Official Sources
About the author

BEFORE YOU START

How to Use This Book

This book shows internal auditors how to hand the mechanical half of an engagement to an AI tool that runs on their own laptop, send it as little company data as the task needs, and check everything it gives back.

> SCENARIO · CALDER & VOSS INTERNAL AUDIT · OFF-SITE · SIX WEEKS TO DECIDE
>
> At the department off-site, Ruth Okafor learns that two of her staff have been pasting spreadsheets into chat tools on their phones to “tidy them up”. Nobody can say what was sent or where it went. Banning AI outright would only push it further out of sight. Ruth gives Priya Raman six weeks to work out what is safe to use on audit work, and what is not.

Who this book is for

You work in internal audit. Perhaps you are a staff auditor two years out of university, a SOX tester in the middle of the testing season, an IT auditor who reads access extracts for a living, an audit manager who reviews other people’s working papers, or a chief audit executive who answers to an audit committee. You know risk-based auditing, COSO, sampling and report writing far better than any software does.

What you lack is hours. Too much of each engagement goes on work that is necessary but not judgment: reformatting ERP extracts, matching one list against another, re-typing walkthrough notes into a narrative, chasing the same number through three spreadsheets, rewriting an observation for the fourth time so the audit committee will read past its first line.

You are probably also wary of AI, and you are right to be. The data you handle is confidential: payroll, vendor bank details, unpublished results, the occasional investigation. Your audit committee expects discretion. Your IT security team may not have approved any AI tool yet. And a tool that sounds certain when it is wrong is a real risk in a profession built on evidence. This book does not ask you to trust the tool. It shows you how to set it up so that it sees as little

company data as the task allows, where that protection stops, and how to review its work the way you would review a new auditor's working paper.

You do not need to know how to program. You write requests in plain English. The tool writes any code that is needed, and, once Chapter 2's settings are in place, you approve each step before it runs.

What Claude Code is, in plain English

Claude is an AI model made by a company called Anthropic. Most people meet it in a chat window: you type a question, paste some text or attach a file, and it answers.

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, create new ones, and write and run small programs on your laptop. Recent versions start in auto mode, where a background check approves most actions for you. Chapter 2 switches that to Manual mode, in which it asks your permission before it changes anything or runs most commands.

It works in a terminal (a plain text window where you type), and also in Anthropic's desktop app and in some code editors. This book uses the terminal, because it shows most clearly what the tool is doing, and covers both Windows and Mac.

	Chat window	**Claude Code**
Where it works	In a browser or app	On your laptop, inside one folder you choose
What it sees	Whatever you paste or attach, in full	Only what it opens, and what its scripts print
What it can do	Answer and draft	Answer, draft, create files, and run scripts on your files
On a 60,000-row AP extract	Reads what fits, and may skip or guess rows	Writes a script that tests every row and reports counts you can tie out
Who is in control	You decide what to paste	You decide the folder, and in Manual mode approve each action

That difference matters for two reasons.

The first is **confidentiality**. Attach an AP extract to a chat and every vendor, bank account and amount goes to the provider. With Claude Code you can ask it to look only at the column headers, write a short program (a *script*) that runs the test, and run that script on your laptop. Only what the script prints, such as counts and totals, goes back. That sharply reduces what leaves your machine. It is a habit, not a guarantee, and Chapter 1 sets out exactly where it stops.

The second is **evidence**. A chat model reading a large file can lose or invent rows without telling you. A script applies the same rule to every row, every time, and produces counts that must add

up. It can be filed, re-run and re-performed by a reviewer. In internal audit, the safe way and the defensible way turn out to be the same way.

How the book is organized

Read Part 1 first, in order. After that, use the lifecycle chapters in Part 2 as you reach that stage of an engagement.

Part	Chapters	What you get
Part 1: Foundations	1 What Leaves Your Laptop; 2 Get Approved and Set Up	What Claude Code sends and keeps, the routine that keeps company data local, and a setup your IT security team can sign off.
Part 2: Through the audit lifecycle	3 Risk Assessment; 4 Planning; 5 Walkthroughs; 6 Control Testing; 7 and 8 Full-Population Analytics; 9 Fraud Red Flags; 10 Working Papers; 11 Observations; 12 Reports and Audit Committee Summaries; 13 Follow-Up	For each stage: where it helps, worked examples with copy-ready prompts, more prompts to try, and the traps specific to that stage.
Part 3: Working well	14 Checking What It Gives You; 15 Your Department's Playbook; 16 Limits	How to review its work, how to make a whole team use it the same way, and where it should not be used at all.
Part 4: Try it yourself	17 Duplicate Payments and the Three-Way Match; 18 User Access and Segregation of Duties	Two end-to-end walkthroughs on practice files with planted exceptions, which you can download, run and check against an answer key.

> **Tip.** If you have one hour, read Chapter 1, then try one worked example from the chapter closest to your current engagement, on a test you have already done by hand. Comparing its answer with yours is the quickest way to learn what it is good at and where it needs watching.

> **From the desk.** Before you start, list five tasks from your last engagement that were mostly matching, reformatting or re-typing: the vendor master clean-up, the sample selection, the narrative from walkthrough notes, the issues tracker update. That list is your first month's practice. Start with the smallest.

How to read a worked example

The lifecycle chapters are built around worked examples. They all follow the same pattern, so you can scan them quickly and adapt them to your own engagements.

1 **The situation**
A short, realistic setting: who is doing the work, which entity, which 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 names, columns and periods to match yours. Keep the instruction to use a script that runs on your computer whenever the data belongs to the company.

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

4 **What to check**
A checklist of what to verify before you rely on the result: counts that should tie out, records to trace to the source, and judgments that remain yours.

Dark terminal panels show what a session looks like on screen. Lines starting with > are what you type.

> **WHAT A PROMPT BOX LOOKS LIKE**
> Look only at the column headers of AP_Invoices_H1.xlsx. Do not open the full file. Write a script that finds invoices with the same vendor, invoice number and amount, run it here, and show me only the number of duplicate groups and their total value. Save the detail to Duplicates.xlsx.

> **Read this once.** **Illustrative or tested?** The 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 two walkthroughs in Part 4 are labelled *tested*: each was run on the practice files that come with this book, and its "How this was tested" note says what was run and when.

The people in this book

Each chapter opens with a short scenario from the internal audit department of Calder & Voss Group, a made-up packaging and materials company listed in the US, with operations in the UK, Germany, Dubai and Singapore. The department has fourteen people. These are the ones you will meet.

Who	Role	What to watch for
Ruth Okafor	Chief Audit Executive, Charlotte. Reports to the audit committee.	Skeptical and precise. Asks whether a reviewer could re-perform it.
Daniel Marsh	Audit manager, financial audits and SOX.	The early adopter. Quick to try things, sometimes quick to trust them.

Who	Role	What to watch for
Priya Raman	Senior auditor, data analytics, Singapore.	Methodical. Becomes the department's Claude Code person.
Tomás Herrera	Staff auditor in his first year.	Bright and overloaded. Probably the closest to you.
Leila Haddad	IT auditor, Dubai. ITGCs and user access.	Fast and practical. Not yet convinced that documentation matters.
Graham Pike	Audit senior, operational audits, Manchester.	A good interviewer who writes very long notes.

Conventions used in this book

- Every company, person, vendor and figure in the examples is made up. Any resemblance to a real business is accidental.
- File names look like `AP_Invoices_H1.xlsx`. Commands you type look the same.
- Claude Code changes often. Menus, messages and option labels may differ from what is shown here. Where a setting matters for confidentiality, the book tells you how to confirm it yourself.
- The IIA's Global Internal Audit Standards, COSO's 2013 Internal Control - Integrated Framework and SOX are referred to only in general terms. Your own methodology, and the current text of the standards, are the authority.

A note before you start

> **Independence and approval.** This book is independent. It is not affiliated with, sponsored by or endorsed by Anthropic, The IIA or COSO. Anthropic's own documentation, terms and privacy policy are the authority on how Claude Code works. Nothing in this book is legal or professional advice, and nothing in it replaces your organization's policies on data, AI tools and audit methodology. Before you use Claude Code on company data, make sure your organization has approved it for that use (Chapter 2).

CHAPTER

1

What Leaves Your Laptop

Before an AI tool touches company data, you should know what it sends, where, and how to keep that to the minimum the task needs. This chapter gives you the rule, the working pattern that follows from it, and a routine for every engagement.

> SCENARIO · CALDER & VOSS PACKAGING · VENDOR MASTER · FIRST WEEK OF THE PILOT
>
> Tomás Herrera has to find duplicate vendors in the US vendor master: 2,300 records with addresses, tax IDs and bank accounts. At 7 pm he opens a public chat tool, pastes the first 500 rows, then stops. Ruth's rule for the pilot: nothing goes into any AI tool until the department can say what leaves the laptop. "If we can't re-perform it, we didn't do it."

This chapter is what Priya showed Tomás the next morning: what Claude Code sends, what it does not, and the one working pattern that keeps a vendor master on the laptop it belongs on.

The one rule

Claude Code runs on your laptop, but the thinking does not. The AI model that reads your request and decides what to do next runs on Anthropic's servers. Claude Code is the part on your machine that carries messages to the model and carries out its instructions: list this folder, open that file, run this script.

So the rule is short. **Whatever Claude Code reads, it sends.**

If it opens a file, the file's contents go to Anthropic as part of the conversation. If it runs a command, whatever the command prints goes too. If nothing in the session opens a file and shows its contents, those contents are not sent to the model.

You cannot stop the model seeing what Claude Code reads. You can decide what Claude Code reads. The rest of this chapter follows from that.

What is sent, and what is not

A session is a loop. You type a request; Claude Code sends it to the model with some standing context; the model replies, often by asking to use a tool; Claude Code runs the tool on your laptop and sends back the result. The loop repeats until the job is done. This is what travels in it.

Item	Sent to Anthropic?	Notes
What you type	Yes	Including any figures, names or rows you paste in.
Instruction files such as `CLAUDE.md`	Yes	Loaded at the start of each session. Keep company data out of them.
A file Claude Code opens to read	Yes, its contents	Excel, CSV, Word, PDF, images. A scanned PDF goes as page images.
Output of a command or script	Yes	Whatever is printed. This is the part you shape.
File and folder names it lists	Yes	A folder called `Investigation_Dubai_Vendor` tells the model what it is.
Files a script reads but does not print	No	The script runs on your laptop. Only what it prints is sent.
Files it never touches	No	Other folders stay put, as long as nothing opens them.
Output files a script writes	Not unless opened	A workbook saved to the folder stays local until something reads it.

Two more points are easy to miss.

Your laptop keeps a copy. Claude Code saves each session as a plain-text transcript in a `.claude` folder under your user profile, so you can resume later. By default these are deleted after a set period, which the `cleanupPeriodDays` setting controls; check the current default in the documentation. A locked screen, disk encryption and an account nobody else uses therefore matter as much as any setting inside the tool. Corporate laptops usually have the first two; confirm it.

There is also a cloud version. Anthropic offers Claude Code sessions that run on its own machines, on copies of your files or code held there. This book assumes Claude Code running on

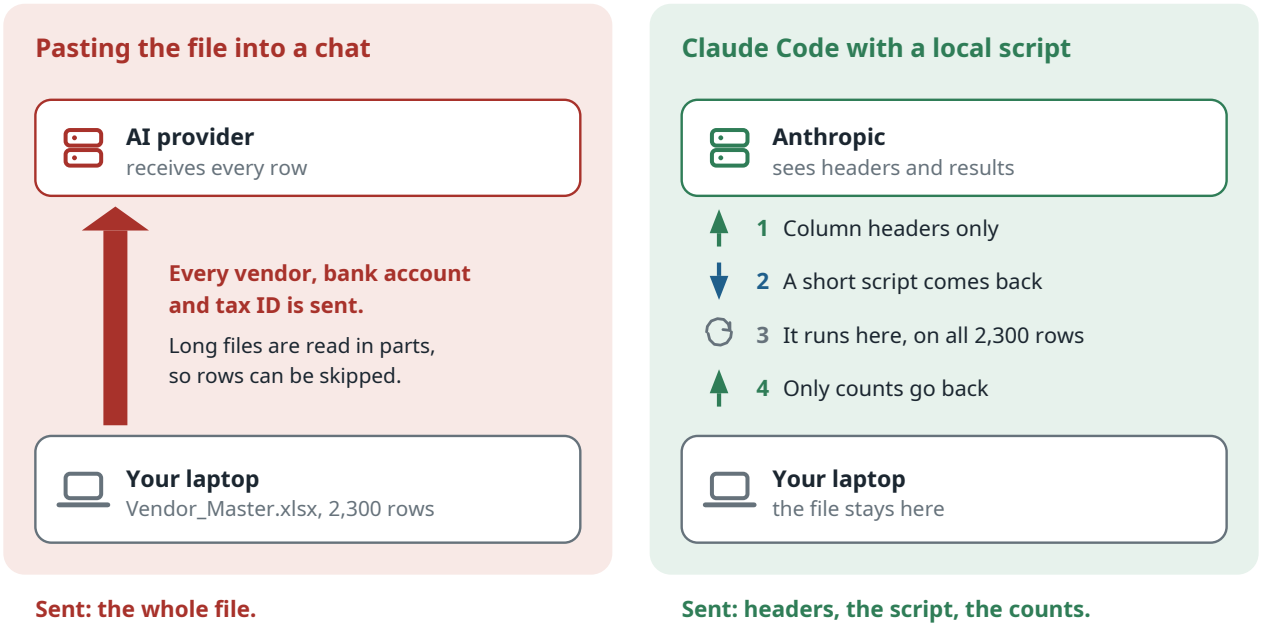
your own laptop. Do not use cloud sessions for audit data unless your organization has approved exactly that.

The script pattern

Priya started with Tomás's own task: which vendor records look like the same supplier set up twice? There are two ways to get the answer.

The chat way: paste the file, ask the question. Every vendor name, address, tax ID and bank account goes to the provider. The model then tries to compare thousands of rows inside a working memory that cannot hold them all at once, and it can miss pairs without saying so.

The script way is how you should use Claude Code on any company data.



The same question, asked two ways. On the right, the vendor master never leaves the laptop; only its headers, the script and the counts do. Illustrative.

1 **Look at the shape, not the data**
Claude Code learns the layout from the column headers: which column holds the tax ID, how bank accounts are stored, which date format is used. If you allow it, it may also look at a few sample rows, and those rows are sent.

2 **Write**
It writes a short script, usually in Python, that does the test. You do not have to read the code, but you can ask for a plain-English explanation of every rule it applies.

3 **Run locally**
With your permission, it runs the script on your laptop. The script reads every row here. None of those rows is sent anywhere.

4 **Report**
The script prints a summary: rows read, groups found, rows saved to a file. That printout goes back to the model, and to you. The detail stays in a workbook on your disk.

Notice what this does for evidence as well as privacy. A script treats row 2,000 exactly like row 2. It produces counts that must reconcile: rows in should equal rows matched plus rows not matched. You can file it, re-run it next quarter and hand it to a reviewer. **On company data, the safe way and the defensible way are the same way.** The rest of the book calls this the script pattern, and every worked example uses it.

WORKED EXAMPLE · ILLUSTRATIVE

Duplicate vendors without sending a single bank account

Priya sits with Tomás. He copies one file into a new folder for this job: `Vendor_Master_US.xlsx`, exported from Oracle that afternoon, 2,300 rows with vendor ID, name, address, tax ID, bank account and routing number, created date and status. “Ask for a script,” she says, “and tell it exactly what it may print.”

> **TYPE THIS**
>
> Vendor_Master_US.xlsx is company data. Read only its column headers; do not open the data rows and do not show me any values.
>
> Write a Python script that finds possible duplicate vendors: (a) the same tax ID, (b) the same bank account and routing number, (c) names that match after removing punctuation, case and words like Inc, LLC, Co and Corp.
>
> Run it here. Print only: rows read, the number of groups found by each test, and the number of vendors in more than one group.
>
> Save the groups to Vendor_Duplicates.xlsx, one sheet per test, with a Parameters sheet (file name, row count, rules used, date run). Do not print the groups.

What Claude Code does

1 **Reads the headers**
It opens the file just far enough to list the column names, and tells you which columns it will use for each test.

2 **Writes the script**
It creates `vendor_dupes.py` in the folder and asks permission to run it.

3 **Runs it on your laptop**
You approve. The script reads all 2,300 rows locally.

4 **Reports counts only**
On screen: something like 2,300 rows read; 14 groups on tax ID, 9 on bank account, 31 on normalized name; 22 vendors in more than one group.

5 **Leaves the detail with you**
`Vendor_Duplicates.xlsx` is saved in the folder. You open it in Excel. Claude Code has not read it, so its contents were not sent.

BEFORE YOU RELY ON IT, CHECK

- [ ] Rows read matches the row count you see in Excel (and the count in the Oracle report).
- [ ] Open three groups from each sheet and confirm by eye that they really are the same supplier, or explain why not (a parent and a subsidiary can share a tax ID legitimately).
- [ ] Ask Claude Code to explain the name rule in plain English, and check it would not merge two genuinely different vendors with similar names.

Note. If you then ask "why might these 9 vendors share bank accounts?", Claude Code needs to read the 9 groups to answer, and they will be sent. That may be acceptable. The point is that you decide, row by row, what goes out.

Leaks the script pattern does not stop

- **The peek.** Headers are rarely sensitive, but sample rows are real data. For sensitive files, say "headers only, no values", or give it a dummy file with the same headers and made-up rows (the second example below).
- **What you ask it to print.** "Show me the vendors paid over $250,000" prints them, and printed rows are sent. Ask for counts and totals on screen and lists saved to a file.
- **Error messages.** When a script fails, the error sometimes quotes the value that caused it. Usually that is one cell, but know that it happens.
- **Names of files and folders.** They are sent whenever Claude Code lists the folder. Name engagement folders by code (`IA-2026-14_AP`) rather than by the subject of a sensitive review.

ADD THIS LINE TO ANY REQUEST ON COMPANY DATA

Use a script for this and run it here. Do not open or print the data rows. Show me only counts, totals and the names of any files you save.

Know the limits. **The script pattern controls what goes to the model. It is a working habit, not a security boundary. Scripts and commands you approve run as you.** They can open any file your account can open, not just the engagement folder. Claude Code's permission settings govern its own read and edit tools, not the programs it runs. Before you approve a script, check which files it opens and what it prints. **Commands can reach the internet.** An install, a download or an upload can send data off your laptop with no involvement from the model. Decline any command that posts, uploads or contacts an address you did not expect, and ask what it is for. **Connected tools have their own providers.** Web search, web fetch and any MCP servers or plugins pass what you give them to third parties. Claude Code offers an optional sandbox that limits what commands can read and where they can connect. At the time of writing it runs on macOS, Linux and WSL 2 but not on native Windows; check the current documentation. Your organization's IT controls, data classification and approval decide the rest (Chapter 2).

Match the data to the method

Internal audit sees more kinds of sensitive data than almost any other function, often in the same week. Not all of it should be handled the same way. Agree a simple scheme with your CAE and IT security before the first engagement, and write it into your department's `CLAUDE.md` (Chapter 15). Here is a starting point; your organization's own data classification policy takes precedence.

Kind of data	Examples	How to use it with Claude Code
Methodology and public material	Audit manual, RCM templates, published standards you hold a copy of, your own notes with no company specifics	Can be read directly.
Internal business data	AP, AR and GL extracts, PO and receipt data, policies, process notes with roles instead of names	Script pattern: headers only, results printed as counts, detail saved to files.
Personal data	Payroll, HR rosters, T&E claims, user access lists with names	Script pattern with no sample rows, or a dummy file; mask or code names where you can. Data protection laws such as GDPR and UK GDPR may apply; ask your privacy team.
Highly restricted	Investigations, whistleblower reports, privileged legal advice, unpublished results, board papers, passwords and keys	Not in Claude Code unless legal and the CAE approve the specific use in writing. Some of it, never (Chapter 16).

WORKED EXAMPLE · ILLUSTRATIVE

Build on a fake file, run on the real one

Priya needs a payroll test for C&V Asia: employees paid after their termination date, and two employees sharing one bank account. The file, `Payroll_Register_Q2.xlsx`, holds names, salaries and bank details for 1,150 people. She will not let even a sample row out, so she has Claude Code build the test on a made-up copy, then runs the finished script on the real file herself.

TYPE THIS

I will give you only the column headers of a payroll register, not the file: Employee_ID, Name, Pay_Date, Gross_Pay, Net_Pay, Bank_Account, Bank_Code, Termination_Date, Cost_Center.

1. Create Dummy_Payroll.xlsx with 200 made-up rows using exactly these headers, and plant three cases: one employee paid after their termination date, two employees with the same bank account, and one normal leaver paid before leaving.

2. Write payroll_tests.py that takes a file name as an argument and writes Payroll_Exceptions.xlsx with one sheet per test and a Parameters sheet.

3. Run it on the dummy file only and show me that it catches the planted cases and not the normal leaver.

Do not look for or open any other file in this folder.

What Claude Code does

1. **Makes the fake file**
   It writes a small generator and creates `Dummy_Payroll.xlsx`. Every value is invented.

2. **Writes the test**
   It writes `payroll_tests.py`, reading the file name from the command line.

3. **Proves it on the fake**
   It runs the script on the dummy file and reports that the three planted cases are caught and the normal leaver is not.

4. **Hands over**
   Priya closes the session, opens a plain terminal (not Claude Code), and runs `python payroll_tests.py Payroll_Register_Q2.xlsx` herself. Nothing from the real file is sent anywhere.

BEFORE YOU RELY ON IT, CHECK

- [ ] Read the script's rules (or ask for them in plain English) before you run it on real data. A test proven on a fake file is only as good as the fake.
- [ ] On the real run, tie the rows read to the payroll register's control total.
- [ ] Keep the real file outside the folder Claude Code was started in, so it cannot be opened by mistake.

> **From the desk.** Run Claude Code in a folder that holds only the copies this task needs, never in your Documents folder, your desktop or a shared drive root. Name the folder by engagement code, keep the original extracts read-only somewhere else, and delete scratch copies when the working papers are filed.

Training and privacy settings, by plan

Two separate questions matter. Can your sessions be used to train future models? And how long does Anthropic keep them? The answers depend on the kind of plan.

Plan type	Used for training?	What to do
Individual plans (such as Pro and Max)	Only if the model-improvement setting is on. The user controls it.	If policy allows an individual plan at all, switch the setting off in claude.ai under Settings, then Privacy, and recheck it after any change to the consumer terms.
Commercial plans (Team, Enterprise, API keys from the Claude Console)	Not by default under Anthropic's commercial terms, unless the organization opts in to a program that allows it.	Ask the account administrator to confirm no such opt-in is active. Keep a copy of the terms with the pilot approval.
Through a cloud provider (for example Amazon Bedrock or Google Cloud)	Governed by that provider's terms and your company's agreement with it.	IT and legal confirm the terms as part of the approval (Chapter 2).

Retention, meaning how long conversations are stored, differs by plan and setting, and it changes from time to time. This book does not quote durations. Read the current figures in Anthropic's Privacy Center and the Claude Code data-usage documentation before the pilot starts, file a dated copy with the approval, and check again once a year.

For a department, a commercial plan is the natural starting point: training is off by default, an administrator can manage settings for everyone, and you have terms you can show the audit committee if asked.

> **Confidentiality.** Some actions share a full transcript with Anthropic on purpose: the `/feedback` command, and a session survey that may ask to see your transcript. Never share a transcript from an audit session. Chapter 2 shows the settings that switch both off.

The routine for every session

BEFORE YOU TYPE THE FIRST PROMPT

- [ ] The use is approved: your organization allows Claude Code for this kind of data, on this plan.
- [ ] You started Claude Code in **one engagement folder** holding only the copies this task needs.
- [ ] The status line shows Manual mode, not auto mode (Chapter 2).
- [ ] Nothing highly restricted is in that folder or below it.
- [ ] Names are replaced by roles or codes wherever the test does not need them.
- [ ] Your first prompt says: headers only, script runs here, print counts, save detail to a file.
- [ ] You will read each permission prompt before you approve it.

Where to be careful

Watch out. Do not paste rows into the prompt "just to show it the format". Pasted text is sent like everything else. Describe the format in words, or point it at the headers.

Don't. If a request would need Claude Code to read a sensitive document in full (an investigation memo, a board paper, a legal letter), stop. Ask whether the task needs the tool at all, and if it does, get the approval first.

Key takeaways

- Whatever Claude Code reads, it sends. You control what it reads.
- For company data, use the script pattern: headers in, script runs locally, counts out, detail saved to a file you open yourself.
- The pattern limits what reaches the model; it does not limit what an approved script can do. Read before you approve.
- Match the method to the data: direct reading for methodology, scripts for business data, dummy files for personal data, and nothing for the highly restricted.
- Prefer a commercial plan, check training and retention settings yourself, and never share audit transcripts.

ABOUT THE AUTHOR

Chandan Gupta

CA, CIA. An auditor who builds the tools the audit needs.

Chandan is a Chartered Accountant and a Certified Internal Auditor (The Institute of Internal Auditors). His work spans internal audit, internal controls over financial reporting and statutory audit.

I learned auditing the slow way: vouching, ticking, and copying numbers from one schedule to the next. Over the years the files moved from paper to spreadsheets to data extracts, but one thing stayed the same. Too much of the day went on work a machine should have done, while the part that needed an auditor waited.

So I started building my own tools. Among them:

- a desktop workbench that loads a year's ledger from an ERP export, proves it is complete, and runs journal-entry tests on every entry
- internal audit analytics that turn untidy exports into draft observations, each traceable back to the source row
- reconciliation utilities, and a set of Claude Code skills that carry a controls engagement from the request list to the closing report

Nearly all of them run offline. I am not comfortable with company data leaving the machines it belongs on, and I suspect most internal auditors feel the same.

I design each tool the way I would design an audit procedure: what must be proved, from which data, and how a reviewer will re-perform it. Claude Code writes the code; I test and review the result as I would a junior auditor's working paper. Some tools worked first time. Many needed several rounds. This book comes out of those rounds, including the mistakes.

It is written for colleagues who keep asking the same question: how do you use AI on real audit work without giving up confidentiality or professional skepticism? My answer is less about clever prompts than about habits. Keep the data local, let the tool do the mechanical work, and keep the judgment and the check for yourself.

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

The same is true of AI. An output is evidence that something produced an output. Whether it is right is still the auditor's call. The routines in this book are there to help you make that call faster, not to skip it.

If the book saves you an evening, or falls short on your files, I would like to hear about it, and about what you want in the next edition. You can find me on LinkedIn: https://www.linkedin.com/in/chandan-gupta-b8a108192



Get the full book

This sample has the introduction and Chapter 1. The full **AI for Internal Auditors** adds setup and approval, eleven chapters that follow the audit lifecycle (risk assessment, planning, walkthroughs, control testing and sampling, full-population analytics, fraud red flags, working papers, observations, reporting and follow-up) with worked examples, chapters on checking output and building your department's playbook, two tested walkthroughs on downloadable practice files, and an appendix of every prompt in the book.

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