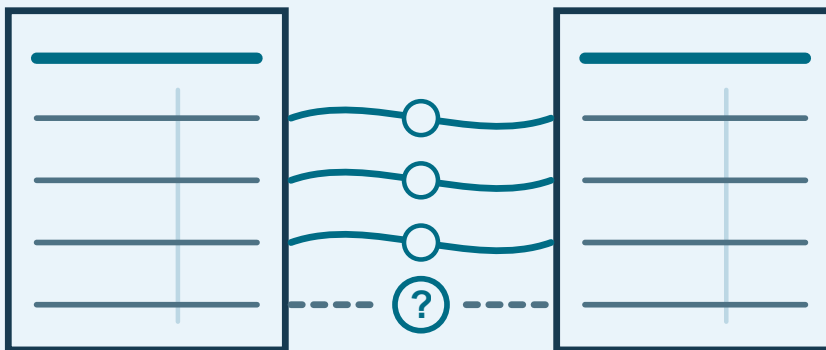




PYTHON AND DATA WORKFLOWS

Reconcile Business Data with SQL

Explain invoice and receipt differences without hiding duplicates or missing records



Sanjaay Gowda T S

FREE SAMPLE / INTRODUCTION + CHAPTER 1 / EDITION 1.0.0

sanjaaygowdats.ai

About this free sample

Reconcile Business Data with SQL

Sanjaay Gowda T S

First edition 1.0.0, 2026-10-13

Copyright 2026 Sanjaay Gowda T S. This is a labelled free sample, not the complete ebook. You may share this sample unchanged. It contains the cover, sample permissions, contents, introduction, Chapter 1 and a closing product link. The remaining chapters, worked solutions and companion files are not included. The full ebook text, diagrams and artwork have separate single-reader terms; this sample permission does not authorize sharing the full purchased PDF or EPUB.

The companion source code, tests and fictional data are separately MIT licensed. You may adapt or share that project with its included licence notice. MIT does not apply to the ebook text or book artwork. The books and companion projects have no technical activation requirement or buyer-specific watermark.

This first edition was prepared with AI assistance. The case studies are fictional teaching examples. Any measured outputs come from the supplied examples, not production customer data. The teaching scope, assumptions and limitations are explained in the chapters. Independent human technical, editorial and assistive-technology reader review is pending. Automated tests and accessibility checks do not establish certification or suitability for every use.

Run examples in a separate working folder using the supplied synthetic data. Review inputs and calculations before adapting the techniques to real operations. Model software, service access, subscriptions, ongoing updates and consulting are not included.

Author website: sanjaaygowdats.ai

Ebook terms: sanjaaygowdats.ai/ebooks/license

Inside this free sample

Start here.....	4
1. Define the reconciliation contract	5
Continue with the full guide	6

This free sample contains the introduction and Chapter 1 only. The closing link opens the complete guide. The companion project and worked solutions are supplied with the complete ebook.

Start here

Reconciliation asks whether two representations of activity agree under a stated set of rules. A query that returns no differences is useful only when you can explain which records it compared, which records it excluded and why the comparison was valid. Joining two extracts and subtracting their amounts is often the shortest part of the work.

This book follows Cedar Events, a fictional organization that issues invoices for room bookings and records receipts separately. All customers, identifiers and amounts are invented. The example is an educational data comparison, not financial advice, a ledger implementation or a claim of accounting compliance. It compares integer amounts in one teaching currency, without fees, credits, taxes or exchange rates.

The supplied companion creates an in-memory SQLite database using Python's standard library. It stages two JSON arrays, aggregates receipts by invoice identity, classifies invoice differences and lists receipts whose referenced invoice is missing. Nothing connects to a live system or changes a business record. Run the fixture and tests from the project folder.

```
python main.py --input example-input.json
python -m unittest discover -s tests -v
```

Read the resulting classifications first. C101 has two receipts that together match its invoice. C102 is underpaid, C103 is overpaid and C104 has no receipt. R5 references C999, which is absent from the invoice extract. That last exception must remain visible even though a query starting only from invoices would not encounter it.

The nine chapters build an evidence trail around these cases. You will define grain, constrain staging, understand join multiplication, distinguish missing values from zero and prove control equations. The exercises change one premise at a time. Worked solutions show what the change establishes and what it still cannot prove. Use a supported Python version; the code requires only interfaces available in Python 3.10 and later.

1. Define the reconciliation contract

Decide what agreement means

Cedar's invoice extract has one row per invoice identity. Its receipt extract has one row per receipt identity and may contain several receipts for one invoice. Agreement means that the sum of linked receipt amounts equals the invoice amount exactly. That definition is an original teaching policy. A real organization might reconcile allocations, settlements or outstanding balances using different events and cutoffs.

Write the comparison date and source coverage before inspecting mismatches. An invoice exported at noon and receipts exported at the end of the day can disagree for a legitimate timing reason. The companion assumes both supplied arrays form one fixed comparison snapshot. It cannot establish that the source systems exported every relevant record.

Contract item	Cedar example	Consequence
Invoice grain	One invoice identity	Duplicate identity blocks the run
Receipt grain	One receipt identity	Split payment is allowed
Amount unit	Integer cents	No binary float tolerance needed
Difference sign	Paid minus invoice	Negative means underpaid

Do not infer identity from customer and amount. Two customers can share a name, and one customer can have several equal invoices. A stable invoice identity connects the records. If a receipt allocates across several invoices, the correct receipt-side grain becomes an allocation identity rather than a raw bank receipt. The current program deliberately does not invent that allocation.

Review lab: false agreement

Two wrong datasets can have equal grand totals. Suppose a receipt is linked to C102 instead of C101 and another mistake offsets the total. A total-only check may pass while both invoices are wrong. Retain identity-level comparisons and a global control. These checks complement each other: the global control detects lost amounts, while the identity comparison detects misallocation that preserves the grand total. Neither proves that the original invoices represent legitimate transactions.

Exercise: define a timing exception

Describe a receipt recorded one day after an invoice snapshot. State what additional timestamp and cutoff evidence you need before calling it a timing difference. Avoid classifying every unpaid invoice as a timing issue. The explanation should be supported by a specific later event and a policy that defines whether that event belongs to this reconciliation.

Worked scope decision: cancellation

Cedar cancels a booking after issuing an invoice. One export still contains the invoice while another system records a cancellation elsewhere. The current inputs have no cancellation field, so the program cannot infer that the invoice should be excluded. Deleting the invoice from a spreadsheet may make receipt totals look closer, but it changes the population without retaining the reason. A revised contract could include invoice status and a cancellation event reference, with an agreed cutoff. It would then report excluded cancellations as a separate controlled population. Construct a cancelled invoice with a prior receipt as well as one without a receipt; the two cases may need different review actions. Until that requirement is implemented, the correct response is to identify the unsupported event and ask for authoritative evidence. This is a limitation of source scope, not a defect that can be fixed by changing a subtraction sign.

Continue with the full guide

This free sample ends after the introduction and Chapter 1. The complete guide contains nine teaching chapters, worked exercise solutions, primary references and a separate Python companion project. The full PDF, reflowable EPUB and project files are supplied together.

Use the complete guide to work through the remaining exercises and inspect the supplied examples. The companion code and fictional data are separately MIT licensed; the complete ebook has the single-reader terms described on its product page.

The examples are teaching fixtures, not production customer results. Independent human technical, editorial and assistive-technology reader review remains pending. Software access, subscriptions, consulting and guaranteed results are not included.

[View the complete guide and purchase options.](#)

You may share this labelled free sample unchanged. Keep the full purchased PDF and EPUB private. Thank you for reading.