DATA CHANGE RECONCILIATION

Building a Reconciliation Report for Data Changes



BEFORE / AFTER

AI & DATA

Building a Reconciliation Report for Data Changes

Learning outcomes

- Reconcile row count, unique keys, totals and exceptions across stages.
- Find three unexplained changes and locate their first appearance.
- Distinguish permitted formatting changes from data loss or duplication.
- Write a stage-specific conservation contract with exact comparison units and authorized differences.
- Reconcile full key multiplicities, field associations and exception identities rather than relying on matching totals.
- Separate missing, zero and negative amounts and report mixed currencies without unsupported conversion.
- Build a reproducible evidence report that locates first divergence and supports a bounded release decision.
- Reconcile permitted routing and aggregation using complete membership lineage.

Original teaching material. All organizations, people, records and operational rules in exercises are synthetic. Use only the supplied non-sensitive inputs. This is a practice course, not an accredited or regulated certification.

Study the foundations, work through the applied units, complete the practice tasks, and then attempt the assessment. Use only the supplied fictional practice data. Keep your original inputs and your revised work separate.

Core concepts

Controls express what should be conserved

A reconciliation report compares known quantities before and after a transformation. Useful controls include row count, unique-key count, numeric total and exception count. Not every transformation preserves every metric: aggregation intentionally changes row count, while trimming spaces should not. State the permitted change for each step before inspecting results. In this exercise trimming and category mapping must preserve records and amounts; a unique lookup join must not multiply rows; export must preserve all values. Held exceptions must remain visible.

Compare adjacent stages as well as endpoints

An end-to-end difference shows that something changed, but adjacent comparisons show where the change first appeared. Calculate deltas fromS 0 toS 1, S1 toS 2 and so on. If one step loses a row and the next duplicates another, the final row count may return to its original value while the dataset remains wrong. Therefore an endpoint row count of 10 is not enough. The unique-key count and amount total help reveal the offsetting errors. Preserve each stage snapshot or control log so the analysis can be repeated.

Use multiple independent controls

Row count detects some losses and multiplications; unique-key count distinguishes repeated identifiers; amount totals test value conservation; exception count checks whether unresolved cases were silently removed. A matching total cannot prove all values stayed attached to the right records, so row-level checks remain useful. Controls are diagnostic evidence, not a complete proof of correctness. When a discrepancy appears, identify the first affected stage and inspect the relevant source records instead of adjusting the report total to match a desired number.

Explain normal changes and stop unexplained ones

Trimming surrounding spaces can legitimately alter text while leaving all four controls unchanged. A category mapping may change labels but must not drop held records under this policy. The supplied stage log contains three error-producing transitions. Describe the likely mechanism supported by the row-level notes: deletion, multiplication and numeric alteration. Do not claim an exact software bug without evidence. The release recommendation should hold the final output until these discrepancies are resolved and controls rerun from the preserved source.

Worked example

Demonstration: a pipeline starts with 5 rows and ends with 5, but stage 2 drops one and stage 3 duplicates another. Equal endpoints conceal two errors. An adjacent-stage report records -1 then +1 rows, while unique keys remain reduced, revealing that the original set was not preserved.

Foundation practice

Stage controls

stage	rows	unique ids	amount total	exceptions
S0 raw	10	10	550	2
S1 trim spaces	10	10	550	2
S2 category map	9	9	490	1
S3 join lookup	10	9	540	1

stage	rows	unique ids	amount total	exceptions
S4 export	10	9	545	1

Permitted changes

S1: Trim spaces only; preserve all controls

S2: Map labels; preserve rows, keys, amounts and held exceptions

S3: Attach unique lookup attributes; preserve rows, keys, amounts

S4: Export; preserve all values

Row level notes

- S2 omitted ID 06 with amount 60 and unresolved category.
- S3 produced two rows for ID 05 amount 50 because lookup key repeated.
- S4 changed one amount from 20 to 25.

Foundation tasks

Guided preparation

Instructions

Read the complete input pack. Create the output columns specified in the capstone, keep the original IDs, and state the unit of work. Complete the first two records manually before using an optional AI assistant. Record any assumption that could change the result.

Independent capstone

Instructions

Create a before/after reconciliation with deltas for all four controls at every transition. Find the three unexplained changes, link them to the first stage and source note, and distinguish the normal trim step. Recommend whether the final output may be released.

Peer verification and revision

Instructions

Exchange only your student work with a peer. Ask the peer to trace two ordinary results and every exception back to the supplied input. Record the exact item corrected, the original value, revised value and reason. Submit the revised artifact plus a short explanation of the most important remaining limitation.

01 Turn “nothing important changed” into a testable contract

A reconciliation report is an argument about a transformation, not a decorative table of totals. The argument needs a starting population, an intended operation, a comparison rule and evidence. Without those four pieces, the statement “the files agree” is incomplete. Two files may agree on a grand total while disagreeing on which event owns each amount. They may also disagree in bytes while representing the same permitted information. Decide what agreement means before deciding which differences matter.

Our extension concerns one synthetic event per event_id. Every event has an amount in USD, an event date, a category and a disposition. The pipeline is allowed to trim category spaces, map category codes and attach one reference tag. It is not allowed to discard an event, multiply one, move an amount to another ID or resolve a held exception without evidence. Export is a preservation operation. These are local exercise rules, not universal rules for every data pipeline. A legitimate aggregate would need a different contract.

A useful contract names both invariants and authorized changes. “Preserve amounts” means more than preserve their sum: each event must retain its own amount. “Preserve exceptions” means retain the held identities and dispositions, not merely finish with the same count. “Attach a lookup” requires one authorized tag per event, not any nonblank text. State the field, the allowed transformation and the population to which it applies. This makes the control falsifiable: a reviewer can exhibit a particular event that breaks it.

Do not create permissions after seeing an inconvenient result. If a transformation loses a held record, calling that loss “cleaning” does not make it authorized. A change request can propose a new rule, but the current report must evaluate the current rule. Record unresolved differences as unresolved. A technically reconciled dataset may still be unsuitable for release because its retained HOLD records need review. Conversely, a pipeline may have no HOLD rows only because it erased them. Distinguish correctness of transfer from readiness for use.

Think of the report as a chain of claims. A source inventory supports the population claim; adjacent-stage comparisons support conservation; key and field evidence support association; exception history supports disposition; and an explicit gate supports the final decision. Each claim should identify a check capable of disproving it. A report that lists only favorable metrics has not established the chain, even if every listed number is accurate.

E01-X1 Write the extension contract

Use assets/contract.json. State the grain, amount unit, four permitted operations and two distinct reasons a fully reconstructed dataset can still be held.

Submit: A compact contract with six named controls and a release distinction.

02 Inventory snapshots before comparing them

The extension package contains five complete snapshots, R0_raw.csv through R4_export.csv. R0 is the starting population; R1 is the trim result; R2 is the category-map result; R3 is the lookup result; and R4 is the export result. Their common column order is event_id, event_date, category, amount, currency, status and lookup_tag. Blank lookup tags are expected before the lookup step. They are not missing amounts. Every amount in this main case is supplied and must satisfy the exact two-decimal grammar.

Treat these files as preserved evidence, not as sheets to edit until they agree. A source row can be examined many times without being changed. Keep your report separately. The complete executable assets/reconcile.py reads the files and prints JSON; it does not repair them. If you later create a repaired candidate, give it a separate identity and compare it again. Retaining a failed snapshot is valuable because it records where the problem actually appeared.

The reference_lookup.csv file is the authorized reference for the exercise. lookup_used.csv records the lookup input associated with the faulty R3 snapshot. They are intentionally different. The first supplies one tag per event; the second supplies two entries for E05. The report should distinguish these roles instead of silently treating whichever file was loaded last as authoritative. File names alone are weak evidence in real work, so the manifest and contract explicitly name each role here.

The manifest's dates are supplied scenario metadata. The events occurred on 2026-02-28; the source message was sent on 2026-03-01 at 17:00 UTC; the five processing times are on 2026-03-02. Those are three different clocks. None is the time you execute a check. If you run the program today, record your actual run separately. Do not replace the business date with the processing date or claim that a supplied scenario timestamp proves an execution you performed.

An inventory also prevents scope drift. This report covers batch REC-26-02 and the supplied snapshots, not every event in the organization's history. The synthetic reference table may contain an event absent from an intermediate snapshot; that does not authorize inventing the missing fact row from the reference table. Facts come from the preserved source. A reviewer should be able to locate the source, identify the intended destination and repeat your comparison without guessing which versions you used.

Evidence	Role
contract.json	Local transformation and release rules
manifest.json	Snapshot order and scenario timestamps
R0_raw.csv … R4_export.csv	Observed stage records supplied for analysis
reference_lookup.csv	Authorized tag for each event
lookup_used.csv	Lookup input associated with faulty R3

E02-X1 Label the evidence clocks

Read the manifest and identify the business event date, source-message timestamp, R4 scenario processing timestamp and the status of your own execution timestamp before you run anything.

Submit: Four labeled time statements and the correct authoritative lookup file.

03 Define controls so two reviewers get the same numbers

A row count includes every data record after the header. It includes held records, zero amounts, negative amounts and repeated IDs. A unique-ID count counts distinct event_id strings. In this case IDs are exact, nonblank strings, so no normalization is authorized. Duplicate excess is the sum of count minus one over each observed ID. With twelve rows and eleven unique IDs, duplicate excess is one; this does not tell you which ID repeated or whether a different ID disappeared.

The amount total is the signed sum of all rows, including repeated rows. Do not silently deduplicate before calculating the observed total: that would turn a diagnostic into a repair. A credit of -8.80 contributes minus 880 cents; a zero contributes zero but remains a record. The main amount grammar permits an optional minus, ASCII digits, a decimal point and exactly two decimal digits. It excludes commas, surrounding spaces, exponent notation and blanks. A failure to parse must block a supposedly complete total rather than disappear inside it.

The exception count is the number of rows whose status is exactly HOLD. The HOLD identity set is a separate control. A duplicated held event could increase the row count while leaving the held identity set unchanged. A lost held event and a newly held different event could preserve the count while changing the set. Both the count and identities matter. CLEAR is not proof that a prior HOLD was legitimately resolved; the transition must supply authority for that change.

Two supplementary controls count zero amounts and negative amounts. These help diagnose selective losses that a sum alone may miss. Dropping a zero row leaves the total unchanged. Reversing the sign of a credit changes the total and can change the negative count. Neither supplementary measure proves full conservation, but each adds a different view. The point is not to collect every possible statistic; it is to select controls sensitive to the errors the contract forbids.

Write each measure's population and unit next to its name. “Amount 39723” is ambiguous until the report says integer cents, USD, all observed data rows. When displaying dollars, use 397.23 without changing the comparison unit. A percentage difference is optional context, not an alternative to the exact difference. A very small unauthorized change still fails this exercise because the policy has no tolerance. Never infer a tolerance merely because a printed report rounds numbers for readability.

E03-X1 Define and contrast the controls

For the hypothetical ID sequence X1, X1, X2 and amounts 10.00, 10.00, 0.00, statuses HOLD, HOLD, CLEAR, calculate rows, unique IDs, duplicate excess, cents total, HOLD rows, HOLD IDs and zero rows.

Submit: Seven named values with the distinct-row versus distinct-ID distinction.

04 Establish an independently checked starting population

A baseline is a measured starting point, not a number copied from yesterday's report. Read R0_raw.csv and calculate the controls directly. Its twelve event IDs run from E01 through E12. The original source contains two held events, E06 and E11. It also contains a zero-valued event, E09, and a negative event, E08. Each is intentional evidence. Treating these rows as unusable would change the starting population before the transformation under review even begins.

A useful manual cross-check separates positive, zero and negative contributions. The positive values sum to 406.03 USD. E08 contributes -8.80 USD and E09 contributes 0.00 USD, producing a net of 397.23 USD. This decomposition is not a second independent dataset, but it is a different arithmetic path that can catch a dropped minus sign. The positive total alone is not the required net. Record both the signed total and the convention used to obtain it.

Baseline key coverage should be explicit. Listing E01 through E12 provides stronger evidence than saying “twelve distinct IDs,” because a later file can have twelve different IDs and still omit one expected event. Record exact identity strings rather than numeric suffixes. E01 is not the same stored value as 1. The event_date and currency are invariant fields in this case. A complete record comparison must retain them even though all supplied dates and currencies currently agree.

The source category codes are intentionally unevenly padded. Spaces around E01's H, E03's H, E04's O and E06's question mark are the only category characters removed by R1. These visible differences are a worked example of permissible change. They do not permit trimming other fields, changing the date or replacing the unknown category with a guessed known category. An unchanged aggregate across R0 and R1 is necessary but not sufficient; verify that the correct four text edits, and no others, occurred.

Checking the baseline first also guards against stale control logs. If a summary says twelve rows but the current file contains thirteen, do not choose the summary because it makes later arithmetic easier. Identify the disagreement, preserve the current evidence and stop the dependent conclusion. In this self-contained exercise the files and rules are supplied together, so you can calculate the baseline without external access. Your report should say which actual file supports each reported quantity.

Event	Amount USD	Status
E06	60.60	HOLD

Event	Amount USD	Status
E08	-8.80	CLEAR
E09	0.00	CLEAR
E11	11.11	HOLD

E04-X1 Build the R0 control line

Compute every control defined in the previous unit from R0_raw.csv. Show positive and negative amount components and identify the four permitted R1 category edits.

Submit: One full control row, arithmetic decomposition and four before/after pairs.

05 Use adjacent deltas to locate first divergence

Calculate a delta as after minus before. Use this sign convention everywhere, including in narrative findings. A negative row delta means fewer observed records at the later stage; a positive amount delta means the later signed total is larger. Label cents separately from dollars. A delta of -6060 cents is -60.60 USD, not -6060 USD. One mislabeled unit can make an otherwise correct reconciliation unusable.

Compare each adjacent pair, not only R0 with R4. The first pair tests trimming, the second tests category mapping, the third tests the lookup attachment and the fourth tests export. This localizes a discrepancy to the earliest supplied snapshot where it is observable. It does not prove the exact implementation bug or the precise instant during a transformation when it occurred. Report the stage boundary and the evidence, and reserve causal claims about code for code evidence.

A difference can persist without being newly introduced. E06 is absent from R2, R3 and R4, but its first observed absence is R2. Listing three independent deletion incidents would overstate the evidence. Likewise, a repeated E05 remains repeated after export even if export adds no further copy. Carry unresolved findings forward, but retain their original first-divergence stage. This separates incident creation from incident persistence.

The endpoint row count is twelve at both R0 and R4. That fact does not cancel the intermediate loss and duplication. Adjacent changes telescope arithmetically: the sum of step deltas equals the endpoint delta for the same metric and population definition. Here the row deltas sum to zero, while the unique-ID delta remains negative. Telescoping is a useful arithmetic check on the report, not proof that compensating errors are acceptable.

Observe what a control does not reveal. The export total changes by five dollars, but that net does not tell you whether exactly one amount changed. Multiple alterations can sum to five. An exception count falling to zero might indicate legitimate resolution in another pipeline, but the current export contract permits no such change. Keep the numerical observation separate from its interpretation. The next step is to inspect identities and fields, not to guess the entire mechanism from a summary line.

E05-X1 Complete the stage and delta matrices

Read all five R snapshots. Report the seven numeric controls for each and all four adjacent deltas. Check the endpoint deltas by summing the adjacent deltas.

Submit: Five stage rows, four delta rows and one endpoint vector.

06 Reconcile identity multiplicities, not just sets

A set answers whether an identity is present. A multiset, also called a bag, answers how many times it is present. Both are needed here. Comparing the R1 and R2 ID sets finds that E06 disappeared. Comparing R2 and R3 sets finds no newly added event identity, even though the row count increased. A multiplicity table identifies E05 as the event whose count grew from one to two. A set-only comparison would miss that multiplication.

Build a frequency table keyed by event_id before attempting field-level pairing. For each ID, show its count in the earlier and later stage. An ID present only before is missing; one present only after is unexpected; unequal positive counts indicate multiplication or contraction. Equal counts above one remain an identity problem when the declared grain requires uniqueness. Do not choose the first duplicate row merely because a spreadsheet lookup or dictionary happens to return it.

E05 illustrates why duplicate evidence needs complete rows. The two R3 rows carry tags ref-E05a and ref-E05b. Their amounts are both 50.50. These are not identical full rows, although they represent the same forbidden repeated event identity. Removing “exact duplicate rows” would not fix the problem. The authorized reference supplies ref-E05, so selecting either observed suffix would also invent a resolution. The correct reconciliation finding identifies the conflict and points to the declared reference.

Use a counterexample to test your own method. Suppose a before file contains Q1 and Q2, and an after file contains Q1 twice. The row count is unchanged; the unique count falls; the ID set loses Q2; the multiplicity of Q1 rises. If both source amounts were equal, the total could remain unchanged too. This is why key coverage and multiplicities are not redundant extras. They test conservation of identity rather than conservation of a numeric summary.

A practical frequency ledger also gives reviewers a compact route back to evidence. Keep the source stage, destination stage, ID, expected count, actual count and relevant full records. When counts differ, mark the group unpairable for ordinary one-to-one field comparison. You can still compare whole-row bags, but do not fabricate an individual row pairing. An honest unresolved duplicate group is more useful than a neat but unsupported list of matched records.

E06-X1 Trace the missing and repeated identities

Create count vectors for every ID E01–E12 across R0–R4. For E06 and E05, name the first divergence and report the observed R3 E05 tags together with the authorized tag.

Submit: Twelve count vectors and two evidence findings.

07 Add a field ledger to catch value swaps

Once an event has exactly one row on both sides of a transition, compare its fields under the permitted transformation. A field ledger names the event, field, expected value, observed value and first affected transition. The expected value may be a permitted category mapping rather than literal source text. For export, however, every field must be preserved exactly. An amount should not move to another event merely because the overall sum stays stable.

R3 to R4 contains three amount alterations. E02 changes from 20.20 to 25.20. E03 changes from 30.30 to 40.40. E04 changes from 40.40 to 30.30. Their signed deltas are +5.00, +10.10 and -10.10 USD. The last two cancel in the grand total. A report that mentions only the five-dollar net difference is numerically correct but substantively incomplete: it misses two wrong event-amount associations.

The same transition changes E11's status from HOLD to CLEAR. That change does not affect the amount total or key coverage. It violates a different invariant: unresolved exception disposition must remain visible. A complete field comparison checks all protected fields, not just measures. The event date and currency happen to remain stable, but they are still in scope. A field that does not change in this dataset can still be a required control.

For E05, the one-to-one precondition is false because both snapshots contain two rows. The full-row bags are unchanged across export, so export introduces no additional E05 difference. The upstream duplicate remains unresolved. State both facts. Avoid treating the stable duplicate bag as evidence that E05 is valid, and avoid attributing the original duplication to export. Correctly assigning first divergence makes the eventual repair narrower and easier to verify.

Do not describe an amount swap as harmless reordering. Row order may change without violating this contract, but the mapping from event ID to amount may not. Sort for presentation if useful, while preserving the identity-based comparison. In a report intended for action, list the affected IDs explicitly. “Some values changed” gives an engineer little to reproduce; four exact field differences with before and after values provide a concrete test target.

E07-X1 Write the complete export field ledger

Compare R3_lookup.csv and R4_export.csv. List every changed field, amount deltas in cents, and the reason E05 receives no new one-to-one field finding.

Submit: Four field rows, a net amount delta and a duplicate-group statement.

08 Reconcile the life of an exception

An exception is a data state that needs explanation, not a row to hide from the report. The main source contains HOLD events E06 and E11. Both have known amounts. Their category is unknown, but their existence and financial contribution are not unknown. A category-map step can label the unresolved category UNMAPPED while preserving the event and its HOLD status. The mapping does not resolve the reason for holding the event.

Track an exception by identity through every stage. E06 is HOLD at R0 and R1, then absent. E11 remains HOLD through R3, then appears as CLEAR at R4. These are different failure mechanisms: disappearance and unauthorized status change. The final exception count of zero is therefore not a success signal. It conceals two unresolved cases in two different ways. Report the transitions at which each loss of exception visibility first occurs.

Exception reconciliation has a useful conservation equation when a workflow genuinely permits resolutions: opening exceptions plus new exceptions minus authorized resolutions equals closing exceptions, with identities supporting every term. A count-only equation is still insufficient because a resolution for one ID cannot explain another ID's disappearance. In this exercise there are no authorized resolutions. Consequently, both opening held identities must remain visible throughout the pipeline, and the expected closing exception population is exactly E06 and E11.

Consider why a zero amount does not remove an exception's importance. A held event could concern category, eligibility, lineage or completeness rather than money. Dropping a zero-valued held record would leave the amount total unchanged but still destroy required evidence. The independent capstone later uses this trap. Design exception controls around the actual reason for holding, not just around whether the event changes a grand total.

A useful report gives each exception an observed state and a decision state. “E11 observed CLEAR in export” is an observation; “E11 remains unresolved under the contract” is the decision supported by missing resolution authority. Keeping both prevents the report from laundering an erroneous source status into an approved outcome. If a real reviewer later supplies valid resolution evidence, record that new authority separately and rerun the relevant controls. Do not rewrite the original evidence to make the history look cleaner.

E08-X1 Build exception histories

For E06 and E11, list the observed status or ABSENT at each stage. Then state the expected closing held identities, their combined original amount and why the observed zero is not an authorized resolution.

Submit: Two five-stage histories and a closing-exception reconciliation.

09 Compare against the authorized result, not a convenient approximation

A literal text comparison can overreport valid changes. R0 category “ H ” should become “H” at R1, and R1 category “H” should become “Hardware” at R2. Comparing the raw source string to the later label without applying the declared rule would flag both as errors. The solution is not to ignore categories. Construct the expected result for that specific transition and compare the observed result against it.

Apply only the authorized projection. For trimming, remove ASCII spaces from the two ends of category; do not strip internal spaces or modify every string column. For mapping, use the complete supplied dictionary: H to Hardware, O to Office and question mark to UNMAPPED. For lookup attachment, use the one tag in reference_lookup.csv for each event ID. For export, use the prior full record unchanged. These projections are small, explicit and testable.

Keep cardinality in the projection. Applying the mapping to R1 should produce twelve expected rows, including E06 with category UNMAPPED and status HOLD. If an implementation maps only known categories and filters unknown ones, it has changed the population, not merely the label. Similarly, attaching a reference tag should produce one expected output per input event. Constructing expected rows before comparing them makes these conservation requirements concrete.

Beware over-normalization in the comparator. If you trim every field on both sides, you can hide an unauthorized change to an ID. If you round every amount, you can hide a forbidden fractional-cent alteration. If you drop repeated IDs before comparing, you can hide multiplication. A comparator is itself a transformation and needs a contract. Normalize only the representations declared equivalent; preserve differences that the policy treats as meaningful.

For this main exercise, export preserves exact field strings as well as amounts. Therefore changing 20.20 to 020.20 would preserve its numeric value but still violate the export representation rule. In a different exercise that permits amount reformatting, you would define numeric equality separately from text equality. Do not mix those policies silently. Good reports identify the level of comparison, explain authorized differences and leave unexpected differences visible for investigation.

E09-X1 Construct three expected records

Using R1 and the reference table, write E06 after a correct category map, E05 after a correct lookup attachment, and E02 after a correct export. Use all seven columns in their declared order; preserve the correct prior amount and status.

Submit: Three complete expected rows with the operation that justifies each change.

10 Read a reproducible reconciliation program critically

The supplied program is deliberately small enough to inspect. It uses the Python standard library and requires no spreadsheet application. From the folder containing assets, run python -B assets/reconcile.py. The program reads the manifest and contract, validates the main fields, calculates controls, constructs authorized transition expectations and compares them to actual snapshots. It prints one JSON report. Redirecting standard output to a new file is optional; do not overwrite a source snapshot.

The amount parser converts accepted text directly to integer cents. It does not pass through binary floating point, so 0.10 becomes exactly 10 cents and -8.80 becomes exactly -880 cents. Grammar validation precedes conversion. This is a design choice for the supplied two-decimal inputs, not a general currency library. Values with three decimal places fail rather than being rounded. If a different currency or precision is required, revise the contract and implementation together.

The comparison routine groups rows by event ID before checking fields. Unequal group sizes produce a multiplicity finding with complete expected and actual rows. Singleton groups receive a field comparison. Equal-sized duplicate groups are compared as whole-row bags

rather than paired by position. The report also lists unresolved duplicate identities even when those bags remain unchanged. This is why R4 can introduce no new E05 change while still failing the overall identity requirement.

The report's all_listed_checks_executed flag refers to the checks in this program when a complete report is returned. It is not a claim that every imaginable quality check was performed, nor that a person reviewed the findings. A parse error prevents a normal report; capture that failure honestly rather than labeling the absent report as a pass. Your own worksheet should distinguish expected program output from output actually observed after running the command.

A runnable program is not self-validating evidence. Inspect its definitions, manually trace selected rows and test an input that should fail. A tool can faithfully implement the wrong comparison rule. For example, a set-based ID check could miss duplicate excess, and a sum-only amount check could miss the E03/E04 swap. Understand which part of the program detects each error before relying on its final decision. Reproducibility means another reviewer can repeat the same bounded procedure, not that the procedure proves everything about the data.

```
# The complete implementation is assets/reconcile.py.
# This self-contained miniature demonstrates the exact-cent rule.
import re

def exact_cents(text):
    if not re.fullmatch(r"-?[0-9]+\.[0-9]{2}", text):
        raise ValueError("Expected a signed two-decimal amount")
    sign = -1 if text.startswith("-") else 1
    whole, fraction = text.lstrip("-").split(".")
    return sign * (100 * int(whole) + int(fraction))

print([exact_cents(x) for x in ["0.10", "0.20", "-8.80"]])
print(sum(exact_cents(x) for x in ["0.10", "0.20"]))
```

E10-X1 Run or trace the report

Predict both printed lines from the miniature. Run assets/reconcile.py if Python is available and retain its entire output; otherwise label the report not run and supply the predicted control and finding contents from your manual work. Explain the scope of the execution flag.

Submit: Two miniature output lines and a complete report, clearly marked observed or expected.

11 Write findings that lead to a specific repair

A useful report has layers. Begin with the decision and the bounded population. Then show a compact control matrix, followed by the detailed findings that explain the differences. Attach exact record evidence and the comparison rules so a reviewer can reproduce each finding. This ordering lets a decision-maker see the consequence quickly without forcing an engineer to reverse-engineer a vague summary. Concision should remove repetition, not evidence.

Give every finding a stable label. In the extension, F1 can name E06's disappearance, F2 E05's multiplication, F3 E02's amount alteration, F4 the E03/E04 association swap and F5

E11's unauthorized status change. F4 contains two field differences but one paired mechanism description. The report may instead assign separate labels to those two fields, provided it does not lose either row. Counts of findings depend on grouping; counts of changed fields and affected identities must remain explicit.

Use language proportional to evidence. “E06 is present before mapping and absent afterward” is directly observed. “The mapping process filtered unknown categories” is a plausible explanation, but the snapshots alone do not prove the implementation path. The lookup_used.csv evidence supports a more specific multiplication explanation for E05 because it contains two rows for that key. Still, avoid naming a particular software defect unless you have inspected the relevant code or execution trace.

Quantify impact without confusing net and gross effects. The endpoint amount is 5.10 USD lower than the source: loss of 60.60, multiplication of 50.50 and net export addition of 5.00. The export swap contributes zero to the net but corrupts two associations. A report should not rank that swap as harmless solely because its net is zero. Similarly, the final disappearance of held statuses is a visibility failure even though it does not change the final amount sum.

Every finding should end with a verification condition. For E06, restore the event from the preserved source and verify one row with the original amount and HOLD status. For E05, use the authorized unique reference and verify one row with ref-E05. For export, verify every field association again. These are conditions for demonstrating a repair, not instructions to patch the summary totals. The release decision remains HOLD until the transformation errors are repaired, all controls rerun and unresolved business exceptions handled under actual authority.

E11-X1 Produce the findings page

Write five findings F1–F5 using the grouping above. Include first transition, exact evidence, affected amount or status, uncertainty and a retest condition. Give the endpoint amount bridge and decision.

Submit: A five-finding page plus a signed cents bridge.

12 Reconcile an intentionally changed grain

Not every row-count change is an error. The separate partition_source.csv case authorizes routing records by the supplied route field, then aggregating ACCEPT records by group. It contains eight source records. ACCEPT, HOLD and REJECT are disjoint destinations in this local case. REJECT means retained in a rejection ledger, not deleted without trace. The report must reconcile the union of the three destinations back to the original population.

This case changes the conservation equation. The source row count should equal accepted detail rows plus held rows plus rejected rows. The same identity must appear in exactly one route, and every source identity must appear somewhere. Matching the sum of route counts is insufficient if one ID is duplicated into two routes and another is omitted. Check the membership union and intersections, then compare each routed record's original group and amount.

The amount bridge uses signed amounts. T6 is an accepted credit of -5.00, so it reduces its group's accepted amount. T7 is a held zero, so it contributes no amount but remains in the held count and identity list. The rejection ledger's T5 amount of 50.00 remains part of the source reconciliation even though it is not part of the accepted output. “Excluded from accepted output” and “excluded from accounting for the source” are different statements.

After routing, accepted detail has five rows. Grouping by A and B produces two summary rows. This smaller count is expected because the grain changes from record to group. Each summary needs lineage listing the contributing accepted IDs. Group A's lineage is T1, T2, T6 and T8; group B's is T4. Compare the membership multiset against accepted detail, not against all source records. Otherwise held and rejected records would be incorrectly demanded in the accepted aggregate.

A complete reconciliation therefore contains two linked proofs: source-to-routes preserves every source record exactly once, and accepted-detail-to-groups preserves all accepted amounts and membership under the new grain. A matching final aggregate total alone proves neither proof. If the report says “eight rows became two,” it must also explain where the nonaccepted records remain and how the five accepted details are represented. This is how a justified row-count change differs from the unauthorized losses in the main case.

E12-X1 Reconcile routes and groups

Using partition_source.csv, calculate each route’s ID list, row count and cents total. Produce the two accepted group rows with exact member IDs and prove both count and amount bridges.

Submit: Three route rows, two group rows and complete lineage coverage.

13 Protect exactness, missingness and units

Exact arithmetic begins with a declared representation. The main case uses integer cents after validating two-decimal text. Python also offers Decimal for decimal arithmetic; constructing Decimal from decimal strings avoids first introducing a binary floating-point approximation. Neither technique decides your rounding policy for you. If a source supplies 1.005 under a two-decimal contract, the correct response here is a validation failure, not an automatic choice of 1.00 or 1.01.

The supplementary precision_cases.json contains a deliberately different missing-value condition: V3 has a blank USD amount. For this small exercise only, report missing amounts separately and total the known amounts. Do not feed that blank into the main pipeline, whose contract forbids missing amounts. This distinction demonstrates that a reusable technique must still honor the local policy. A known subtotal is not a complete total when a member's amount is unknown.

Currency is part of the unit, not decorative metadata. V4 is 5.00 EUR, while the other records are USD. Without an exchange rate and conversion rule, there is no supported combined monetary total. Report one control line per currency. A USD net of zero can coexist with a missing USD amount; it is a zero known subtotal, not evidence that every USD observation is zero or that the unknown amount is irrelevant.

Separate zero, missing and invalid. V6 is a supplied zero and must be counted as present. V3 is missing. The tokens NaN and Infinity are invalid under this contract even though some numeric libraries can represent special values. Likewise, commas, leading spaces, a leading plus and exponent notation are rejected by the declared grammar. This is not a universal claim that such notations can never represent numbers; it is the explicit input rule for this exercise.

Precision controls should also be tested at boundaries. Include a negative amount, zero, leading zero digits if permitted, and a forbidden fractional cent. Record whether the comparison is exact text, exact typed value or rounded presentation. A display such as 0.00 does not prove that an underlying unknown or invalid value was handled correctly. By making missing counts and validation failures visible, the reconciliation report prevents apparently tidy totals from concealing incomplete evidence.

```
from decimal import Decimal
values = [Decimal("0.10"), Decimal("0.20"), Decimal("-0.30")]
print(sum(values, Decimal("0.00")))
```

E13-X1 Reconcile supplementary precision cases

Use precision_cases.json. Give USD and EUR row counts, known-amount counts, missing counts, known subtotals, zero counts and negative counts. State the miniature output and disposition of all seven invalid tokens.

Submit: Two currency control lines, seven rejection decisions and a missing-total caveat.

14 Test whether the controls can fail for the right reasons

A control that always passes is not evidence of correctness. Test sensitivity with a deliberate, bounded fault whose expected effect you understand. Make any trial change in a separate in-memory object or disposable copy, never in the preserved snapshots. A negative control is useful only when it targets a specific claim. For example, deleting a zero-valued row tests identity and row coverage while deliberately leaving the amount total unchanged.

The cancellation_case.json supplies a small independent counterexample. Before and after each contain C1, C2 and C3, once each, with a total of 60.00. C1 and C2 exchange their amounts. Row counts, unique counts, multiplicities and grand totals all pass. Per-ID amount comparison fails twice. This case demonstrates the boundary of the aggregate controls without requiring a large dataset or an obscure software behavior.

Different faults should activate different evidence. Duplicating a positive row raises rows and total but not unique IDs. Duplicating a zero raises rows without changing total. Changing HOLD to CLEAR affects disposition controls without changing amounts. Renaming one ID may preserve rows, unique count and total while changing key coverage. Reordering rows should not fail an identity-based preservation contract. A good test suite includes both forbidden changes that must fail and allowed changes that must pass.

Be explicit about the comparator's assumptions. If IDs are unique, field pairing by ID is meaningful. If they are duplicated, pairing by position can create false differences or hide

real ones. Whole-row bag comparison can determine whether a duplicate group changed, but it cannot make an invalid duplicate identity valid. Include a duplicate group whose rows are reordered to confirm that your method separates order from multiplicity and content.

Test results need provenance too. Write the input mutation, expected detection, actual observation and interpretation. If you have not executed a test, label it expected, not passed. A failed negative control is a failure of the checking method when the forbidden change escapes detection. Repair that method before relying on a green summary. The purpose is not to make tests fail theatrically; it is to demonstrate that the report's claims are supported by controls with the necessary sensitivity and appropriate limits.

E14-X1 Design a sensitivity matrix

Calculate the cancellation case’s before/after controls and exact per-ID amount differences. For deleting a zero row, duplicating a zero row, swapping two amounts, changing HOLD to CLEAR, renaming one unique ID and reordering rows, state which controls detect the change and whether the contract should fail.

Submit: The exact swap ledger and six predicted test outcomes, labeled expected until run.

15 Make evidence repeatable without overstating fingerprints

A report is reproducible when another reviewer can identify the same inputs, apply the same rules and obtain the same observations. Store the batch identifier, ordered stage names, file names, comparison definitions and tool version or command used. Keep the output separate from its sources. If a later source file changes, the earlier report should still identify the version it examined rather than silently appearing to describe the new file.

The supplied program includes a SHA-256 fingerprint of each snapshot's bytes. A fingerprint is a practical way to notice a byte change and bind a report to the file it read. It is not proof of who authored the file, whether the file is correct or whether a business action was authorized. An untrusted source can supply both a bad file and its matching fingerprint. Provenance requires relationships and authority, not just a long hexadecimal string.

Byte identity and semantic identity differ. A permitted line-ending change can alter a file fingerprint without changing parsed records. Reordering rows can change the bytes while preserving an order-insensitive row bag. Conversely, unchanged bytes merely mean the same recorded evidence was reused; they do not validate the values inside. Use fingerprints for evidence identity and schema-aware comparisons for the actual preservation claim. State which kind of agreement each check establishes.

A lightweight provenance record can describe entities, activities and responsibility without adopting a complex platform. R1 is a snapshot entity derived from R0 through trimming; the contract describes the authorized operation; your reconciliation run produces another entity, the report. The W3C PROV family provides a formal vocabulary for such relationships, but the exercise does not require implementing that vocabulary. The practical lesson is to retain enough lineage to distinguish inputs, transformations, outputs and observations.

Time labels deserve the same discipline. The business event date belongs to the data, the message timestamp belongs to the supplied communication history, the scenario processing timestamp belongs to the stated pipeline run, and your observed run time belongs to your check. Do not overwrite one with another. If you did not record an execution time, say it was not recorded. A missing timestamp is a documentation limitation; inventing one is false evidence. Preserve the distinction even when every date falls within the same calendar week.

E15-X1 Specify an evidence envelope

For the main report, list the batch ID, five file names in order, three kinds of supplied time evidence, your own run-status field and the limited claim supported by a matching fingerprint. Explain what a reordered CSV can do to byte and row-bag comparisons.

Submit: A concrete evidence envelope and two bounded fingerprint statements.

16 Repair the transformation and distinguish two release gates

A reconciliation finding is not repaired by changing the reported total. Reconstruct the affected output from the preserved source and the authorized rules. For the main case, a correct pipeline trims the declared category spaces, maps all categories including unknown ones, attaches the unique reference tags and exports without altering protected fields. Each of the twelve original events must remain exactly once. Correctness should be demonstrated by rerunning the same controls, not assumed from the intention to fix.

Rebuild rather than selectively patch when patches would obscure provenance. Restoring E06 by copying a reference tag alone would not recover its original amount, date and disposition. Removing one E05 row arbitrarily would leave the wrong observed tag. Fixing only the five-dollar total would leave the E03/E04 swap and E11 status alteration. The preserved source and reference table provide enough evidence to reconstruct all twelve complete expected records without guessing.

Retest all adjacent boundaries in the repaired candidate, not only the final output. Each preservation step should have zero numeric deltas and no unauthorized field differences after its permitted projection. Check the endpoint as well, including the complete per-ID record mapping. A correction in one stage can create a new error later, and an accidentally correct endpoint can conceal intermediate damage. The same controls that found the problem become acceptance criteria for the repair.

There are two gates. The technical transfer gate asks whether the reconstructed candidate preserves the contracted data. The business disposition gate asks whether unresolved HOLD records may be released. The repaired main candidate passes the transfer checks but still contains E06 and E11 as HOLD. Under this exercise's release policy, it remains HOLD. The reconciliation must not invent a category resolution merely to produce a green final decision.

If valid resolution authority later arrives, it represents a new permitted change that needs its own evidence and comparison. Record who or what authorized the disposition, the affected identities and the subsequent checks. Until then, a truthful outcome is “reconciled; not releasable because two exceptions remain.” This is more informative than labeling the entire

process either successful or failed. It tells the next reviewer exactly which problem has been solved and which decision still lacks authority.

E16-X1 Specify the repaired candidate

Write all twelve expected final records in the seven-column schema. State repaired controls, adjacent numeric deltas, remaining HOLD IDs and separate technical and release decisions. Do not create or overwrite a source file.

Submit: A full twelve-row expected candidate and two-gate decision.

17 Reconcile the preserved foundation without inventing records

The original practice case is preserved in assets/foundation_controls.json. It supplies five stage-control rows and three row-level notes, rather than complete row snapshots. Its stages are S0 raw, S1 trim spaces, S2 category map, S3 join lookup and S4 export. Keep this S-series distinct from the extension's R-series. Similar operations do not make their populations interchangeable: the foundation begins with ten rows and amount 550, while the extension begins with twelve events and 397.23 USD.

The foundation supports exact adjacent deltas for rows, unique IDs, amount total and exceptions. It also supports three specific findings from the supplied notes: S2 omitted ID 06 with amount 60, S3 produced two rows for ID 05 with amount 50, and S4 changed one amount from 20 to 25. The last note does not identify the affected ID. Do not borrow the extension's E02 to fill that gap. Report the amount alteration with identity unspecified in the foundation evidence.

This distinction is an important reconciliation skill. A summary log can locate the first stage where an aggregate changes, but it cannot recreate every source record. The foundation's notes narrow the interpretation, yet they do not supply a complete ten-row source table. You can produce a valid bounded report while explicitly stating that full field-level verification is not possible from this pack alone. Honest limits are preferable to invented precision.

The S0-to-S1 transition is the normal step: all four controls remain stable under the trim-only policy. S1-to-S2 first reveals a loss, S2-to-S3 reveals multiplication and S3-to-S4 reveals a value alteration. The final row count returns to ten, but unique IDs remain nine, the amount ends at 545 and exceptions fall to one. The endpoint differences therefore remain material to the declared preservation rules even though one headline count matches.

Use the foundation to practice reporting at the evidence level available. State the decision, show all four transitions, link each unexplained change to its supplied note and explain the limitation. A reasonable next evidence request would be the preserved row snapshots and the unidentified altered record's lineage. That request is not permission to fabricate missing rows now. The current release recommendation is HOLD because the supplied evidence already establishes unauthorized differences, even without the additional records.

E17-X1 Complete the foundation reconciliation

Use foundation_controls.json only. Produce every adjacent delta, the endpoint vector, three first-divergence findings, the normal transition and a release decision. State exactly what is unknown about the S4 alteration.

Submit: Four delta rows, three findings and a bounded-evidence statement.

18 Independent case: a lost zero and a multiplied credit

The final case uses four separate files, capstone_C0.csv through capstone_C3.csv. Each row has id, amount and status. The grain is one record per ID. All amounts are exact two-decimal values in one common synthetic unit; no currency conversion is involved. Every transition is required to preserve all three fields and identity multiplicity. Row order is irrelevant. There are no authorized deletions, duplicate expansions, amount changes or status resolutions.

C0 contains six records: A1 has 12.00, A2 has 0.00 and HOLD, A3 has -4.00, A4 has 18.00, A5 has 7.00 and A6 has 9.00 and HOLD. Other statuses are CLEAR. The later files are the evidence to investigate, not repaired outputs. This case deliberately includes a zero and a negative amount so that a familiar “lost rows reduce total, duplicate rows increase total” shortcut is unreliable. A lost zero need not change the total, and duplicating a credit makes the signed total smaller.

Build the report in a disciplined order. First state the population and invariants. Then calculate controls from each file, including rows, unique IDs, duplicate excess, signed cents, HOLD rows, zero rows and negative rows. Next produce adjacent deltas and per-ID multiplicity histories. Only then compare fields for uniquely paired IDs and whole-row bags for duplicate groups. Record first divergence separately from later persistence, and trace held identities even when their observed status changes.

Do not stop when you can explain the final amount difference. The case also tests association preservation: two amounts can exchange IDs without changing their combined contribution. A complete report should name every altered field and every lost or multiplied identity. It should also show the full expected repaired candidate from C0, not merely a target total. This gives a reviewer concrete acceptance evidence and prevents a correction that happens to restore one summary while leaving another error.

Finish with separate conclusions about the observed final file and the reconstructed candidate. The observed file cannot be released under the stated rules. A faithful reconstruction can restore transfer correctness, but the two original HOLD records remain unresolved. State which checks you actually ran, which you calculated manually and which are only proposed. Your report should be understandable without a conversation with its author: inputs, rules, observations, findings, uncertainties and decision must all be present in the artifact itself.

E18-X1 Deliver the independent reconciliation report

Use the four capstone files. Supply complete controls and deltas, six ID-count histories, all first-divergence findings and field changes, the endpoint amount bridge, the six repaired records and separate transfer/release decisions. Treat every HOLD as unresolved because no resolution evidence is supplied.

Submit: A self-contained report with exact tables and record-level evidence.

Assessment

Choose one best answer per question. Questions 1–10 check the foundation concepts; questions 11–20 apply the extended study units. Complete and submit the practical tasks separately. A strong quiz score does not replace correct practical work.

1. A preservation pipeline begins and ends with 12 rows. One intermediate stage loses an event and another duplicates a different event. What does the endpoint count establish?

A. All source events remain because the final count recovered.

B. Only amount values require checking because identities are conserved.

C. The intermediate loss is resolved by the later additional row.

D. Only row-count equality; it does not establish identity conservation.

2. R1 includes E06 and R2 omits it; R3 and R4 also omit it. Why should the report compare adjacent snapshots?

A. To prove the exact line of code that deleted E06.

B. To locate R1→R2 as the first supplied boundary showing the loss.

C. To count the missing E06 as three separate deletion incidents.

D. To replace the need for an end-to-end population comparison.

3. A source has 10 rows and 10 unique IDs. After processing it has 10 rows but 9 unique IDs. Which explanation is consistent with these controls under a preservation contract?

A. One source identity may have been lost and another repeated.

B. Only text formatting changed while every identity stayed unique.

C. A new unique identity was added without any other changes.

D. A zero-valued amount was changed while all identities stayed unchanged.

4. R1 is authorized only to trim category spaces. Its total differs from R0 by 1 cent while row and unique-ID counts match. Which conclusion follows?

A. The difference is permitted because it is smaller than one dollar.

B. The difference is permitted if all category labels look correct.

C. The amount difference violates the declared trim-only contract.

D. The difference proves a record was deleted.

5. E06 has amount 60.60, category ? and status HOLD. Mapping permits ?→UNMAPPED but no resolution. What is the correct expected mapping output?

A. Retain E06, 60.60 and HOLD, with category UNMAPPED.

B. Remove E06 from the population until a known category is supplied.

C. Retain E06 and its amount but change HOLD to CLEAR.

D. Retain E06 as HOLD but replace its known amount with zero.

6. One uniquely identified record changes from 20.00 to 25.00; all other fields, rows and IDs are unchanged. Which aggregate control directly reveals this isolated change?

A. The row count, which increases by one.

B. The unique-ID count, which decreases by one.

C. The HOLD count, which necessarily becomes zero.

D. The signed amount total, which increases by 5.00.

7. A lookup attachment produces two rows for a positive 50.50 event instead of one. No other records change. What is the observed effect?

A. Unique IDs rise by one and amount total stays unchanged.

B. Rows stay unchanged because only lookup attributes were requested.

C. Rows rise by one and the signed total rises by 50.50.

D. Only the tag field changes; the repeated amount should be excluded from controls.

8. The final report is short by 5.10 USD. A colleague proposes adding 5.10 to its displayed total without changing the data. What should the reviewer require?

A. Accept the patch if it makes the endpoint amount match exactly.

B. Investigate and repair the transformations, then rerun the evidence checks.

C. Accept the patch only when the row count also matches.

D. Apply the patch to a random record instead of the report.

9. Eight source records are routed completely and disjointly into five ACCEPT, two HOLD and one REJECT records. The five accepted records become two group summaries. When is the row-count change justified?

A. Whenever the two summary rows have a plausible grand total.

B. Only if the final summary still contains eight rows.

C. When held and rejected records are omitted from all evidence.

D. When route coverage and accepted membership and amounts are fully reconciled.

10. A final file has matching row counts but unexplained identity and amount differences. The release policy blocks unauthorized changes. What decision is supported?

A. HOLD until the differences are repaired and required checks rerun.

B. Release because row count is the primary preservation measure.

C. Release if the unexplained net amount is small enough to display as zero.

D. Release after adding a note but without repairing the known defects.

11. E03 gains 10.10 while E04 loses 10.10. Rows, IDs and total are unchanged. Which additional check detects the problem?

A. Compare only the grand total with additional decimal places.

B. Compare only the set of amount values without IDs.

C. Compare each event ID’s expected amount with its observed amount.

D. Compare only the final row count with the source row count.

12. A corrected main candidate has all 12 exact source events, authorized labels and tags, and E06/E11 still HOLD. The policy blocks unresolved HOLD records. Which conclusion is correct?

A. Both gates pass because no transformation errors remain.

B. Transfer can be reconciled while the release decision remains HOLD.

C. Transfer must fail because retaining HOLD is itself data corruption.

D. Change both HOLD statuses to CLEAR to complete the technical repair.

13. The supplied USD values 0.10, 0.20 and -0.30 are known, one other USD value is blank, and a fifth USD value is 0.00. What should the supplementary report say?

A. Complete total 0.00 because the known values cancel.

B. Known subtotal 0.00, one missing amount, complete total unknown.

C. Known subtotal is missing because one member is missing.

D. Treat the blank as another supplied zero and report no missing values.

14. The input contract requires exactly two decimal places and authorizes no rounding. Which treatment of 1.005 is correct?

A. Round to 1.01 because all currency inputs must round upward.

B. Round to 1.00 because matching totals matters more than original precision.

C. Accept it unchanged because Decimal can represent it exactly.

D. Reject it as outside the declared input grammar.

15. Two R3 E05 rows have different tags; the two-row bag is unchanged at R4. How should export reconciliation treat E05?

A. No new bag change at export, but the upstream duplicate remains unresolved.

B. Pair each duplicate by arbitrary position and declare both independently valid.

C. Attribute the original multiplication to export because duplicates are visible there.

D. Delete one observed tag before computing the export comparison.

16. A report records the same SHA-256 fingerprint for a snapshot as a prior report. What limited conclusion is appropriate in this evidence workflow?

A. The snapshot’s amounts and exception statuses must be correct.

B. The source author necessarily approved the later release.

C. The fingerprint supports identifying the same snapshot bytes, not correctness or authority.

D. All semantically equivalent exports must have the same fingerprint.

17. In the capstone, A2 with amount 0.00 and HOLD disappears. Which observation is consistent with that deletion alone?

A. Rows and total both remain unchanged because the amount was zero.

B. The amount total must fall by one because one record was removed.

C. Only the negative-amount count falls.

D. Rows, unique IDs and HOLD count fall while the amount total stays unchanged.

18. A pipeline duplicates a -4.00 record and changes nothing else. What is the amount delta using after minus before?

A. +4.00, because every duplicate increases the total.

B. -4.00, because the additional credit reduces the signed total.

C. 0.00, because repeated IDs should be counted only once in observed totals.

D. -8.00, because there are now two copies in the final file.

19. The foundation note says “one amount changed from 20 to 25” but supplies no affected ID. What should the report state?

A. ID E02 changed because the extension uses that ID for a similar alteration.

B. ID 05 changed because it was duplicated in the preceding stage.

C. S3→S4 has a +5 change; the affected foundation ID is not supplied.

D. No amount finding can be made until every original row is supplied.

20. A learner has read the code and predicted the report but has not executed it. Which execution statement is accurate?

A. Expected output prepared; program execution not yet observed.

B. All checks passed because the expected values are known.

C. The supplied scenario processing time is the learner’s execution time.

D. Execution status is irrelevant if the release decision is HOLD.

Sources and further reading

W3C Model for Tabular Data
https://www.w3.org/TR/tabular-data-model/

W3C PROV Overview: provenance and derivation
https://www.w3.org/TR/prov-overview/

Python decimal documentation
https://docs.python.org/3/library/decimal.html

Python collections.Counter documentation
https://docs.python.org/3/library/collections.html#collections.Counter