

Data Aggregation: Groups, Summaries, and Granularity



- Transform raw row-level records into grouped summaries that answer real business questions



- Aggregation bridges transaction tables and decision-ready insights



- Core tension: more detail vs. clearer insight — the granularity trade-off



- Outcomes: reports, dashboards, KPIs, and choosing the right grain for every task



What One Row Truly Represents



- The foundational question: what does one row represent—an entity, event, or snapshot?



- Defining grain: the business meaning of a single record shapes all valid analysis.



- Example: one row = one order vs. one order line vs. one daily store total.



- Misreading grain causes silent errors: double counting and inflated averages.

DATABASE TABLE

Column A	Column B	Column C	Column D	...
_____	_____	_____	_____	...
_____	_____	_____	_____	...
_____	_____	_____	_____	...
_____	_____	_____	_____	...

ONE ROW = ITS BUSINESS MEANING



Who/What

What entity or thing does this row describe?



When

What point in time does this row capture?



What

What event, transaction, or snapshot is it?



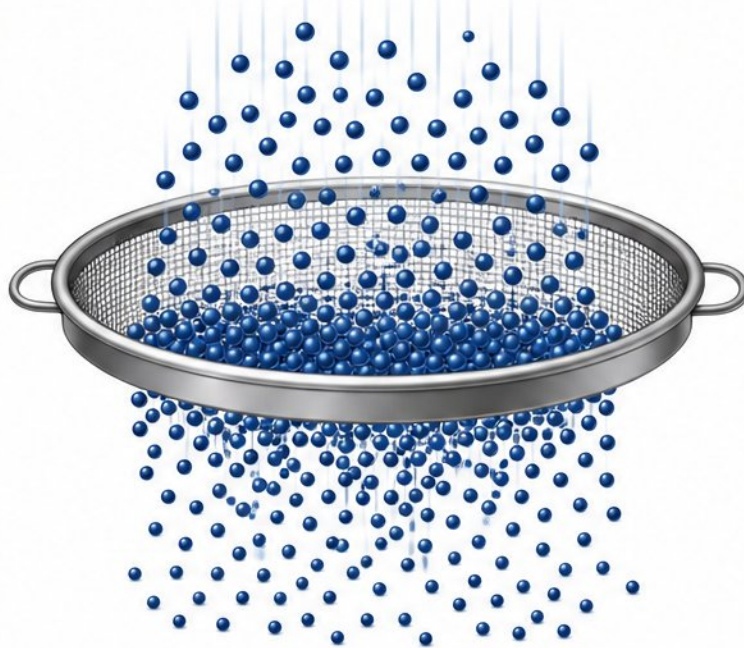
At What Level

What is the granularity or level of detail?

Atomic vs. Aggregated: The Grain Spectrum

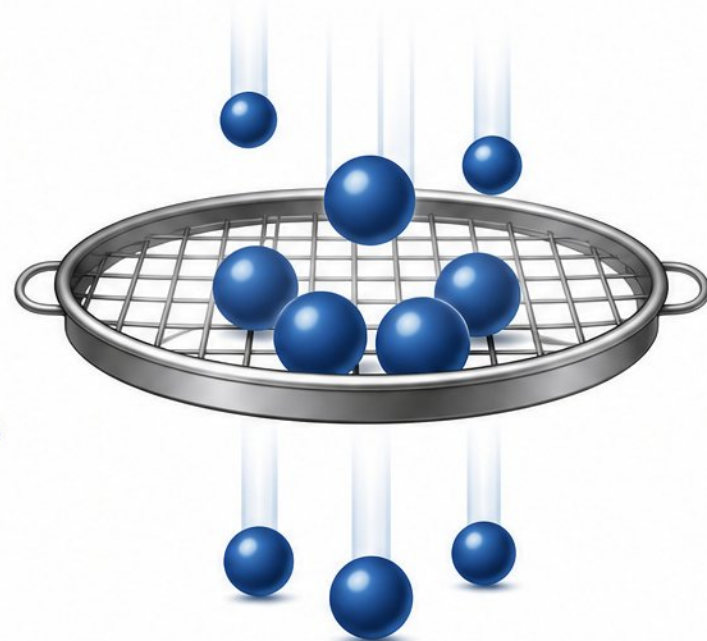
ATOMIC GRAIN

Fine detail: every raw event



AGGREGATED GRAIN

Summarized view: totals or averages



Roll up:
from fine events
to summaries



Atomic grain: the scanner beep or single click—every raw event captured once



Aggregated grain: daily totals or monthly averages—simpler but less detail

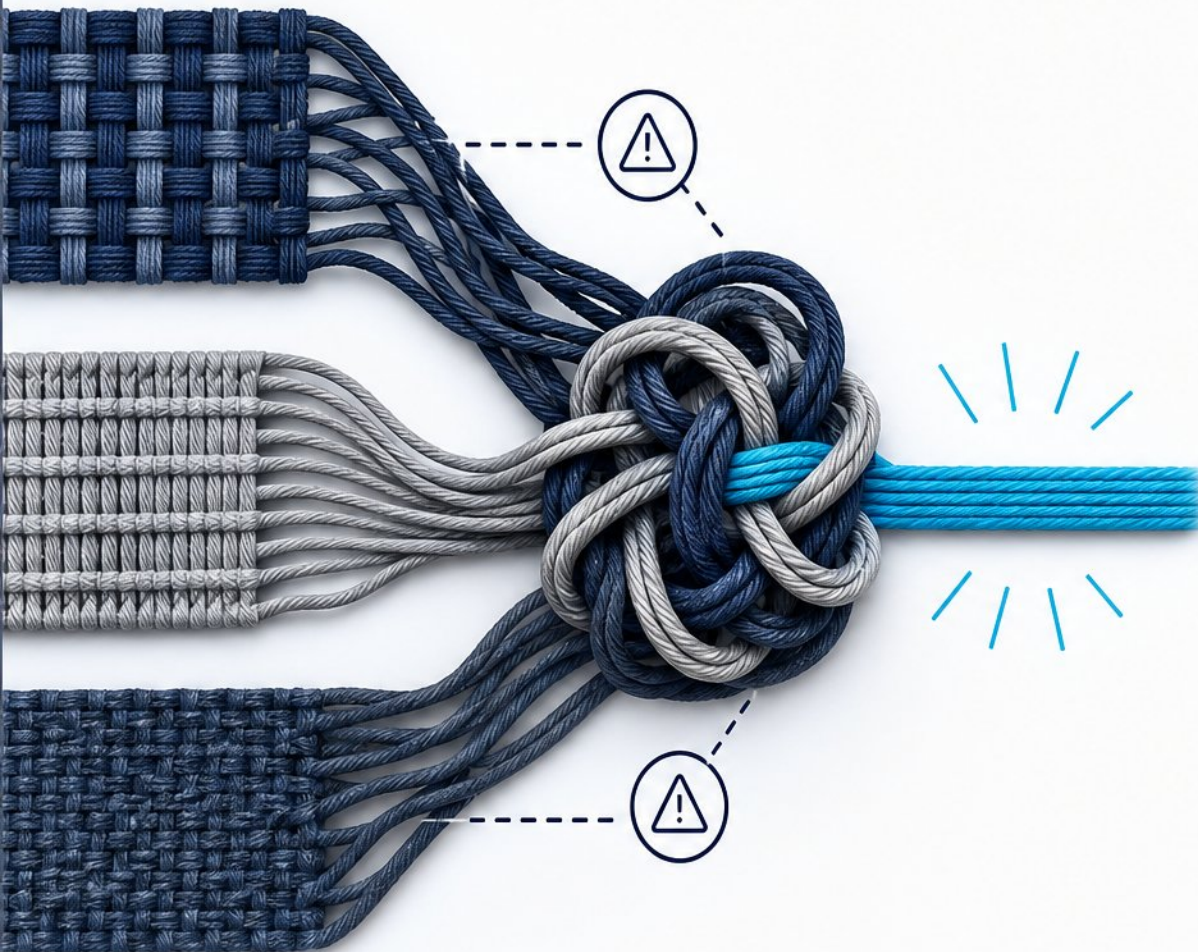


You can always roll up atomic data; lost detail never comes back from aggregates



Start from the finest grain available for flexible, future-proof analysis

Spotting and Avoiding Mismatched Grain



- **Fan-out trap:**
joining without aggregating
multiplies attributes, inflates
metrics



- **Mixed-grain pitfall:**
daily totals in transaction
tables cause double counting



- **Incompatible grains:**
daily summary and raw
transactions can't join
meaningfully



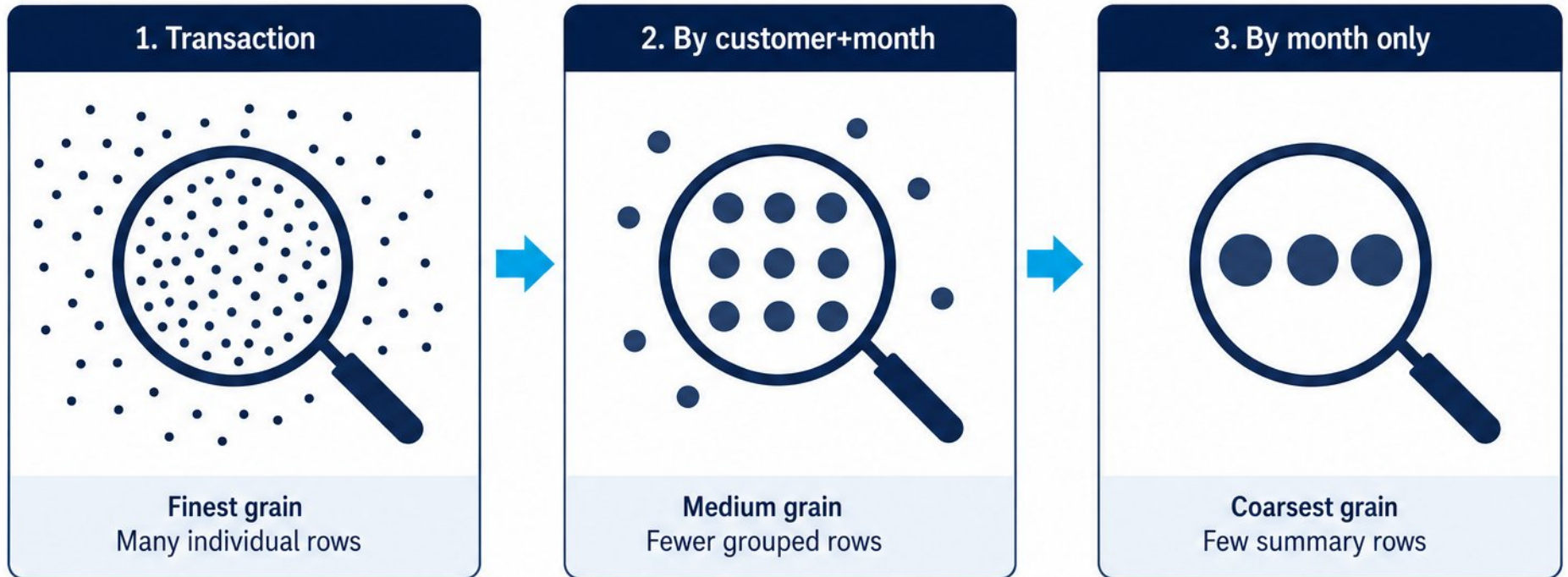
- **Verification habit:**
confirm what one row
means after every join
or grouping

Identifying the Question Behind the Data



- ✓ Begin with decisions needed, not the tool or table
- ✓ 'By region,' 'over time' = specific GROUP BY keys
- ✓ Choose COUNT, SUM, AVG based on what to measure
- ✓ State target grain and operator before touching data

How Groups Reshape Granularity



- Grouping collapses many fine-grained rows into one coarser summary row.



- Group keys define what each output row represents—change key, change story.



- Visual walkthrough: Transaction → by customer+month → by month only.



- Drill-down: add group key for finer grain; roll-up: remove key for broader view.

Summaries That Answer Questions (and Those That Mislead)



- **COUNT** for 'how many',
SUM for 'how much',
AVG for typical



- Same average can hide
very different customer
behaviors



- High total revenue may
only reflect large
population size



- Never average averages
or sum ratios—recompute
from components



Picking the Right Summary for the Job

ADDITIVE



Additive Metrics

(revenue, count)



SUMS SAFELY

across any dimension
without distortion.

SEMI-ADDITIVE



Semi-Additive Metrics

(balances, inventory)



CANNOT BE SUMMED OVER TIME—

use snapshots.

NON-ADDITIVE



Non-Additive Metrics

(rates, percentages)



MUST BE RECALCULATED FROM COMPONENTS

at each grain.



- Additive metrics (revenue, count) sum safely across any dimension without distortion.
- Semi-additive metrics (balances, inventory) cannot be summed over time—use snapshots.
- Non-additive metrics (rates, percentages) must be recalculated from components at each grain.
- If metric type and grain misalign, the number is silently wrong—verify with a small sample.



Composite Summaries: Beyond Single Aggregates



- Real questions need combined aggregates: conversion rates, per-capita sales, average order value



- Safe pattern: calculate numerator and denominator at the same grain, then divide



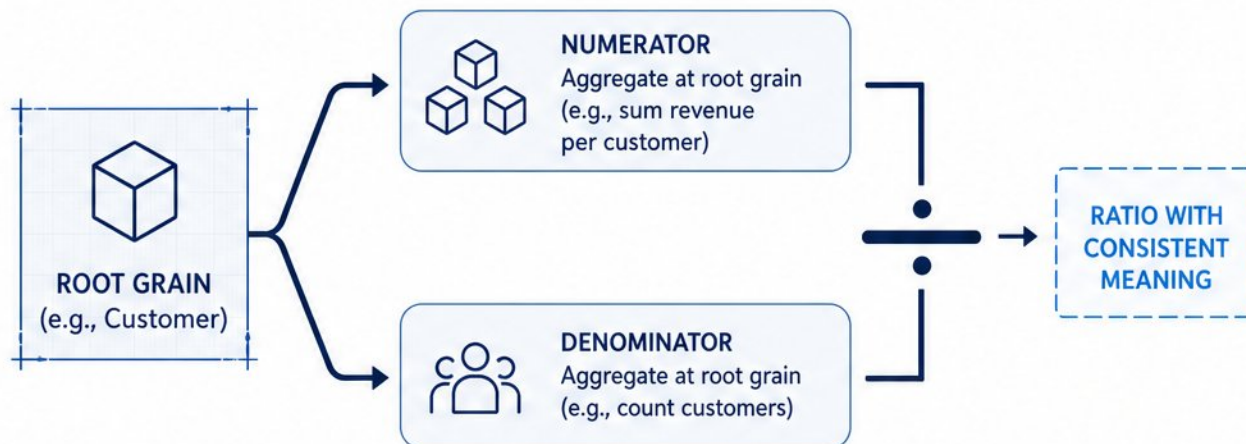
- Common trap: mismatched grain creates ratios with no consistent meaning



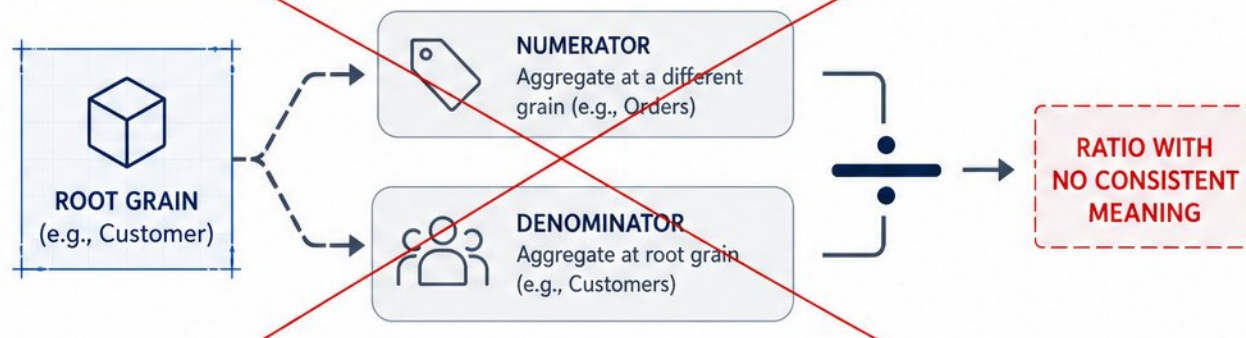
- Example — Customer LTV: sum revenue per customer first, then average those totals



SAFE PATTERN: SAME GRAIN

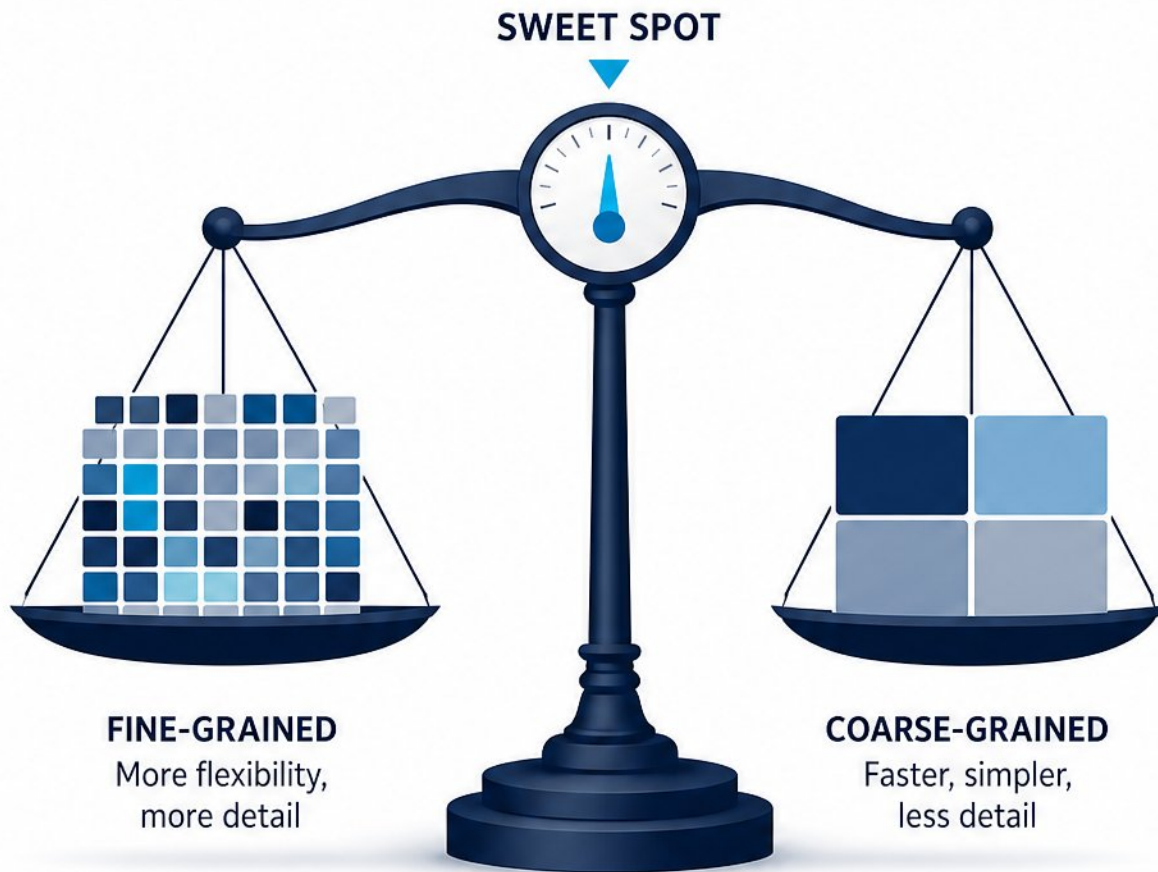


COMMON TRAP: MISMATCHED GRAIN



EXAMPLE — CUSTOMER LTV: sum revenue per customer first, then average those totals

Choosing the Right Granularity: A Strategic Decision



Aim for the sweet spot:

enough detail to act, not so much that it slows you down.



- Finer grain means more flexibility but more rows to manage



- Coarser grain is faster but limits future drill-down questions



- Stop grouping when segments need different business responses



- If you can't act on the difference, merging is likely safe

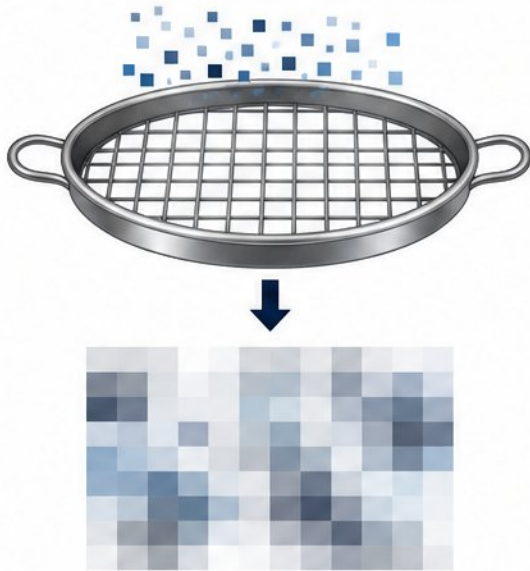


- Granularity is a business decision, not a technical setting

Over-Aggregation and Under-Aggregation: Two Sides of One Problem

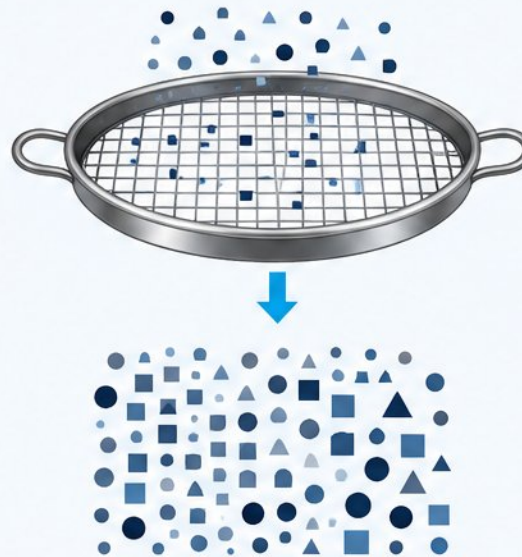
OVER-AGGREGATION

Too coarse: losing detail that matters



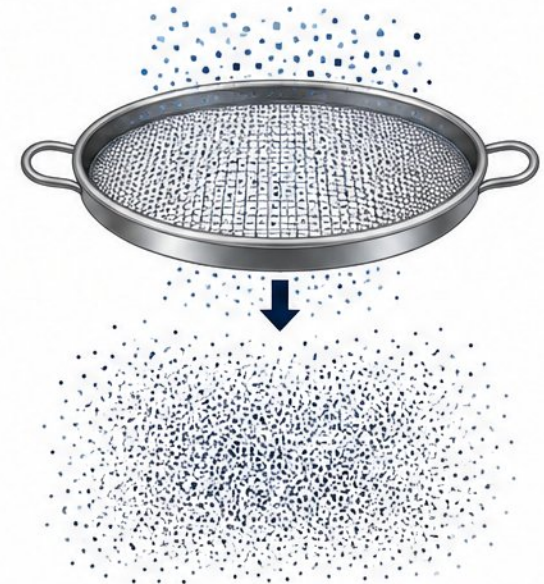
JUST RIGHT

Right grain: revealing meaningful patterns



UNDER-AGGREGATION

Too fine: so much detail that patterns drown in noise



- **Over-aggregation:** losing detail that matters—like averaging across different customer segments



- **Under-aggregation:** so much detail that patterns drown in noise



- **Neither extreme** is universally right; the correct grain depends on the question



- **Litmus test:** can stakeholders answer their question from the output?

Handling Dirty Data and Edge Cases Before Grouping

1



NULL groups hide patterns; decide on showing or excluding them.

2



Duplicate rows silently inflate counts and totals.

3



Unexpected totals?
Trace back to the source row meaning.

4



Verify duplicates, NULLs, and grain before any GROUP BY.

PRE-FLIGHT INSPECTION BEFORE GROUP BY

—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—



NULL values create a “NULL group” and can hide patterns.



Duplicate rows inflate counts and totals.

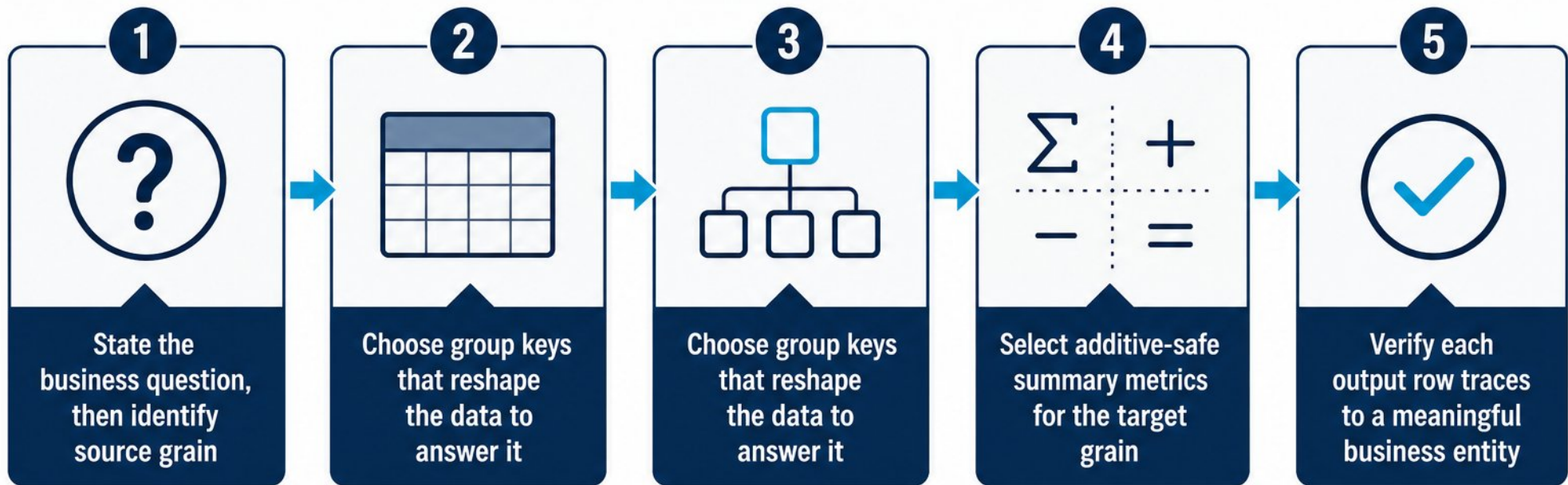


Inconsistent grain leads to mixed meanings.



Clean data in → reliable groups out.

Answering Questions with Group-by-Group Logic



Worked example: sales transactions → monthly revenue by category

SOURCE GRAIN

○
e.g., transaction
(row-level event)

GROUP-BY TRANSFORMATION

TARGET GRAIN

○
e.g., monthly
by category

Communicating Aggregated Results Without Confusion



- Label outputs with explicit grain: "One row = one customer's monthly spend"



- Use line/bar charts for trends; tables for exact numbers and comparisons



- Never mix aggregation levels in one total—label subtotals clearly



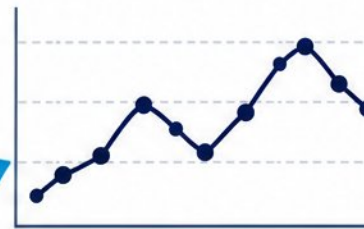
- Pick one aggregated number per insight to make the point clearest

KEY INSIGHT

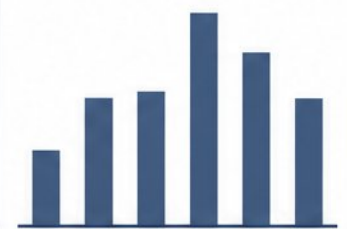


One row = one customer's monthly spend

TREND (LINE CHART)



COMPARISON (BAR CHART)



EXACT NUMBERS (TABLE)

AMBIGUOUS LABELS (AVOID)

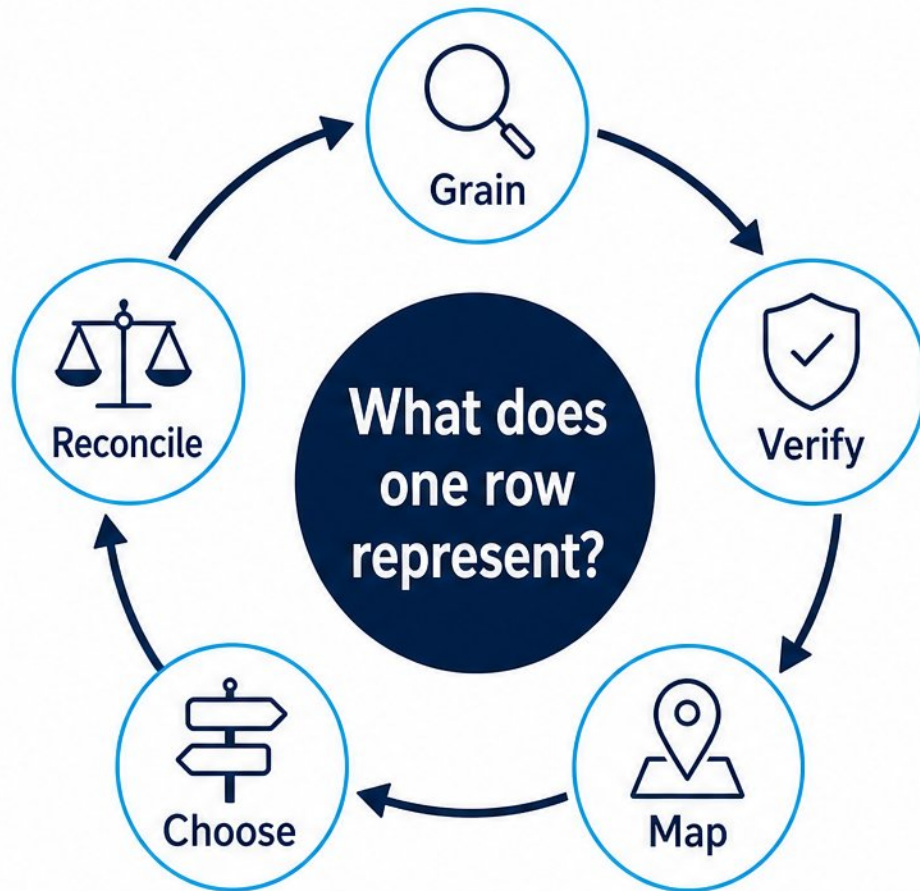
~~Total Spend~~

~~Monthly Total~~

DETAIL

INSIGHT

From Manual Thinking to Habit: Your Aggregation Reflex



- First reflex: 'What does one row represent?'



- Practice checklist: grain, verify, map, choose, reconcile



- Same logic: SQL, Excel, BI tools, pandas



- Groups serve decisions—the right grain makes action obvious



Raw Data (Unstructured)



Grouped Summary (Meaningful)

PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/6a100048/public