

Nobody tests the layer everyone actually looks at.

Why your dashboard can be wrong even when every test passes.

THE STAKES

Pipeline tests catch a broken join.

They don't catch a wrong metric definition sitting one layer up — in the semantic model your dashboards read from.

WHAT IT IS

The semantic layer is where "revenue," "active user," and "churn" get defined once — and reused across every dashboard, model, and report.

WHAT'S ALREADY COVERED

- **dbt tests**
- **Data contracts**
- **CI gates**

Real coverage — but all of it stops at the pipeline. None of it validates the layer on top.

THE GAP

Row-level traceability:

You can see the final number. You often can't see which source rows actually built it.

I tried AtScale for semantic modeling.

The modeling held up. Row-level visibility into an aggregated number did not.

WORKED EXAMPLE

Hypothetical illustration — not a real reported figure

Semantic layer defines "active user" as: logged in AND completed one core action, built by joining a sessions table to an events table.

	Rows	Active users
Dashboard (semantic layer output)	—	42,300
Traced to source rows	40,500 unique	40,500
Gap	~1,800 duplicate rows	~4.3% overcount

Cause: a join fanout — one session matched multiple events — duplicated ~1,800 session rows before the aggregate ran. Every dbt test on the base tables still passed; nothing tested the aggregated output.

WHY IT MATTERS

No trace path turns debugging into guesswork.

Re-run the transformation. Eyeball the SQL. Hope you spot the bad join. In the example above, that's exactly how the fanout would surface — only by tracing back to rows, not by rerunning a pipeline test.

FRAMEWORK — 3 QUESTIONS TO ASK

- 1 Can I trace this number to its source rows?**
- 2 Who owns validating this definition — not just this pipeline?**
- 3 What test would catch a wrong definition, not just a broken join?**

SYNTHESIS

Tests passing tells you the rows moved correctly.

It doesn't tell you the number they add up to is right.

LET'S TALK

**If you're running a semantic layer —
what's actually closing your row-level visibility
gap?**

Comment — genuinely curious what's working.