



Same rows. Faster. Or the doctor *won't sign.*

Executes the original and the rewrite on Zetaris; signs only when the rows match. Hero: **2.11×** over 10 paired runs.



landing + this deck
querydoc.edycu.dev

Case report hero-r1

ACCEPTED

SPECIMEN · THE REWRITE DROPS THE STRUCK LINE

```
SELECT COUNT(1) AS n
FROM   TPCDS_DB.store_sales ss
LEFT JOIN TPCDS_DB.promotion p ON ss.ss_promo_sk = p.p_promo_sk
WHERE  ss.ss_customer_sk IS NULL
```

RESULTS VS REFERENCE · PER-SIDE MEDIANS OF 3 ALTERNATING PAIRS

original		H	29.3 s
rewrite			13.2 s
original	87cd00c5		
rewrite	87cd00c5		

ROW-IDENTICAL · 87CD00C5

Verified by **Verifier** — code, no model · 2.1× = median pair ratio, 3/3 pairs faster

PROBLEM · WHO SIGNS OFF

Faster can still be *wrong*.

```
SELECT COUNT(1) AS n
FROM   TPCDS_DB.store_sales ss
- LEFT JOIN TPCDS_DB.promotion p
-   ON ss.ss_item_sk = p.p_item_sk
WHERE  ss.ss_customer_sk IS NULL
```

original count

129,776

after the edit

129,752

Reviewers see one deleted join. Users: whoever reviews a rewrite before it merges — the diff is all they see.

SOLUTION

A model proposes. *Code decides.*

CODE

Verifier

Executes both, compares rows, times pairs.

- 1 original runs twice – deterministic?
- 2 same columns and types
- 3 0 rows differ (or equal row hash)
- 4 $\geq 1.2\times$, saving $\geq 1,782.9$ ms, most pairs won

MODEL

Diagnoser

Reads the plan and the join key.

unused_outer_join, node 7

MODEL

Rewriter

One rewrite per hypothesis.

drops the unused join

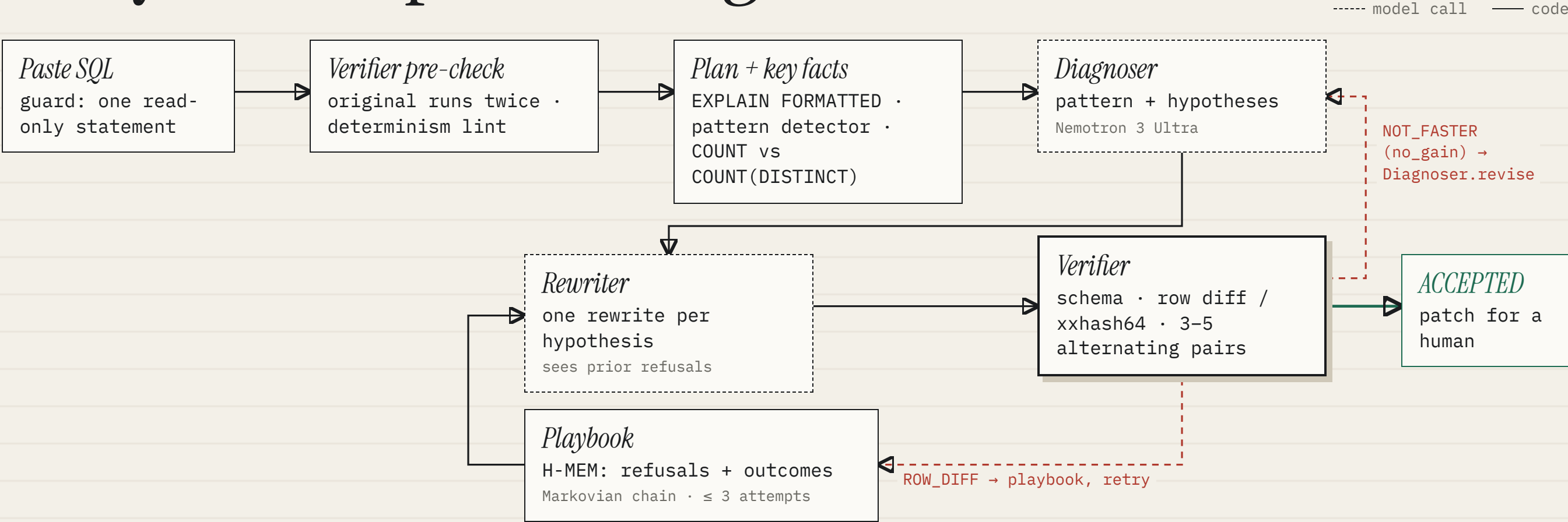
CODE

Orchestrator

Sequences, retries, remembers refusals.

1 attempt → ACCEPTED

Every rewrite passes *one gate*.



<i>Engines</i>	Zetaris · local platform over a Java JDBC sidecar (live)	DuckDB · committed NYC TLC subset (sample, keyless)	model: Nightshift default · NIM fallback (recorded)
----------------	--	---	---

PRODUCT

The report a reviewer *signs*.

○ ○ ○ localhost:8000/#/recorded/hero-r1

3

Recorded live run on Zetaris — hero-r1 · 2026-10-25T12:58:44.862Z. This is evidence from an earlier run, not a new execution.

Case report — Accepted

recorded live run · 2026-10-25 · specimen 87cd00c5 · engine zetaris · model text:
nim

2.1×

median over 3 alternating pairs · saves 16.9 s · 3/3 pairs faster · same 1 row(s)

original

rewrite

H

29.3 s

13.2 s

1

ROW-IDENTICAL · 87CD00C5

1 Verdict and stamp, first.

2 Measured bars: original vs rewrite.

3 Labelled: evidence from a recorded run.

real screen of the app · evidence from the recorded run, not a new execution

v0.1.0 · code 7ac4af9 · 05 / 11

THE MOMENT

One edit. Two keys. *Two verdicts.*

Case report hero-r1

ACCEPTED

LEFT JOIN TPCDS_DB.promotion p
ON ss.ss_promo_sk = p.p_promo_sk

ON p_promo_sk 300 distinct of 300 non-null

fingerprint 87cd00c5 = 87cd00c5
2.1× median pair ratio · 3/3 faster

ROW-IDENTICAL · 87CD00C5

Case report hero-trap-r1

REFUSED

LEFT JOIN TPCDS_DB.promotion p
ON ss.ss_item_sk = p.p_item_sk

ON p_item_sk 291 distinct of 293 non-null

attempt 1 fingerprint e51f9da7 ≠ 87cd00c5
first differing row 129752

REFUSED · ROW_DIFF

EVIDENCE

Measured on the engine, *not estimated*.

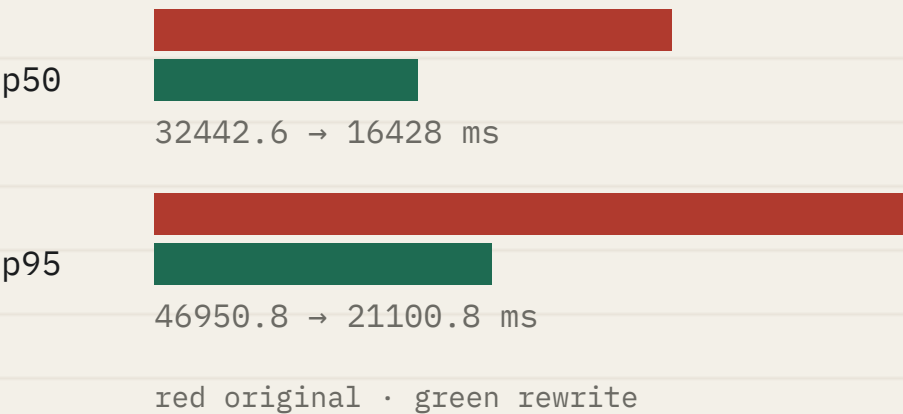
2.1×

live verdict · 3 pairs

hero-r1 · 3/3 pairs
hero-r2 · 2.15×
fingerprint 87cd00c5

2.11×

bench, 10 alternating pairs



6 / 15

baseline changed rows

one model call, same Verifier,
DuckDB
Query Doctor shipped: 0

Δ

Δ

Δ

Δ

Δ

Δ

=

=

=

=

=

.

.

.

.

INSTRUMENTS

Zetaris runs it. *Meterless remembers it.*

Zetaris

11 calls

9 Verifier runs

1 key profile

1 EXPLAIN FORMATTED

NVIDIA Nemotron · 2 calls

Meterless

3 calls

2 H-MEM open, query

1 Markovian chunk

Nightshift: 0 live runs, mock-tested

STACK

Small, tested, *runs keyless*.

```
querydoc/
├─ src/agents/      diagnoser · rewriter · verifier · patterns · profile
├─ src/verify/      canonical · diff · fingerprint · timing · order · determinism
├─ src/reasoning/   chain.ts — Markovian
├─ src/memory/      playbook.ts — H-MEM
├─ src/adapters/    nightshift · nim · llm
├─ engine/zetaris-jdbc/ QuerySidecar.java — one warm JDBC connection
├─ vendor/meterless/ upstream engines, unmodified
├─ test/            34 files
└─ Dockerfile       SAMPLE_MODE=true, no keys
```

447

tests, no keys needed

TypeScript · Node 22 ·
DuckDB sample engine

Cursor wrote scaffold
d3e823c; Claude Code the
rest.

WHAT WAS UNIQUE

It keeps the *wrong answers*.

ON THE PAGE

Refused rewrites stay, struck through, with the differing row.

IN MEMORY

A repeated refusal is read from the ledger.

IN THE REPORT

"identical on the current data" — not a proof.

- 1

hypothesis h1 · REJECT (ROW_DIFF)

Rewriter's note — model text

~~Removed unnecessary LEFT JOIN to promotion table since no promotion columns are referenced and the join cannot affect the row count of a LEFT JOIN with COUNT(1).~~

rows differ: 1 vs 1 — first differing row ["129752"] (rewrite only)

►SQL — diff against the original
- 2

hypothesis h1 · REJECT (ROW_DIFF)

Rewriter's note — model text

~~Replace LEFT OUTER JOIN with LEFT SEMI JOIN to preserve row count semantics while avoiding broadcast hash join overhead by eliminating duplicate matches from non-unique p_item_sk.~~

rows differ: 1 vs 1 — first differing row ["2853"] (rewrite only)

►SQL — diff against the original
- 3

hypothesis h2 · REJECT (ROW_DIFF)

Rewriter's note — model text

~~Removed the redundant LEFT JOIN to promotion table since no promotion columns are used and the join cannot affect COUNT(1) result~~

refusal reused from ledger mem-11 — same original-result fingerprint; not re-run

►SQL — diff against the original

FUTURE

Next: queries *we didn't write*.

-
- ❑ Queries not designed by us, baseline on Zetaris
 - ❑ Re-record the hero through a live Nightshift runtime
 - ❑ Zetaris MCP intake (get_slow_queries)
 - ❑ Write-back of an approved rewrite
-



querydoc.edycu.dev
landing · deck · recorded runs

Verified by the Verifier.