Realy

The accurate, auditable retrieval layer for any agent that needs the right structured data

Realy is a Code-of-Thought, tool-calling agent that retrieves structured data from natural-language requests. It sits between your database and any LLM agent or workflow, precisely and safely finding the exact data an agent requests, with a full record of how it got there.

The problem

Any agent built on top of a database faces the same tradeoff in how data reaches the model:

- **Hard-coded retrieval** (fixed API calls, pre-built views) is fast and predictable, but rigid. It requires anticipating every possible request in advance, breaks when the schema changes, and simply cannot answer anything no one built a view for.
- **Free-form code generation** (an agent writing SQL on the fly) is flexible, but risky. It can silently return subtly wrong data, exposes an arbitrary-SQL attack surface, and adds latency and cost to every request.

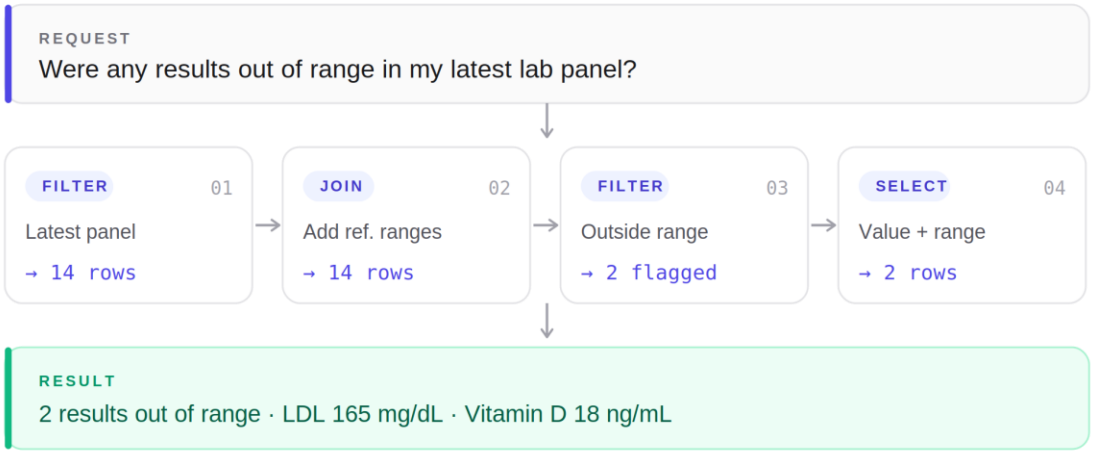
You end up choosing between **control** and **flexibility**. In any setting where accuracy, auditability, and safety matter, neither is good enough.

How Realy works

Realy's core idea is simple: **constrain the action space, not the question space.**

Instead of generating and running free-form code, Realy composes a small, curated catalog of validated table operations to find the right data from a natural language request. For each query, a Really proposes a sequence of operations and then validates each step with code, backtracking when one fails, until it lands on the data that best answers the request. This sequence, or "Code-of-Thought," yields results with greater accuracy compared to competing methods and creates a persistent artifact for each intermediate step that can be audited.

The result is the best of both approaches: the **precision, reliability, and auditability** of engineered code, with the **flexibility** to answer arbitrary data requests over any existing database.



Realy resolves each request as a validated sequence of known operations, or "Code-of-Thought." Every step is a persistent, auditable artifact, run on your existing row-level access. No arbitrary SQL touches the database.

Why it's better

- **Precise context.** Returns only the data a task needs in a standard format for your agent
- **Answers questions no one built for.** A general operation catalog removes the need to hand-build views; novel requests are answered by composing existing operations, even as schemas evolve.
- **Resilient to messy, changing data.** Built for ambiguous column names, mixed types, and evolving schemas, with typed comparators and null-safe semantics.

- **No arbitrary SQL.** A bounded operation catalog removes the security and uncertainty of open-ended generation, while running on top of your existing access controls.
- **Glass-box by design.** Every plan produces a reviewable trace of intermediate results, so anyone can inspect exactly what was fetched and how.
- **Fast and economical.** Caches validated query sequences for common questions, answering them with no LLM in the retrieval path and cutting latency and token cost.