Building a SQL editor around “fast”

Rill Developer’s no-run-button auto-EDA modeling tool

Hamilton Ulmer

Rill Data

@hamiltonulmer on Twitter
🤔 what if your SQL editor didn’t need this button?
Rill's Auto-Profiler; spending the user’s attention budget on EDA

Compactness

x-pixel direction is most costly
should feel like a simple receipt

Easy scanning

group by type, then nullity / uniques
distribution plots enable gut checks to be explored further

Provides details on demand

Richer diagnostics when you click on a profile

Rendered iteratively

As DuckDB returns results, we update the interface
Reduces the cognitive burden of waiting
SELECT
SALE_DATE AS offering_saledate,
INDUSTRYGROUPTYPE AS offering_IndustryGroupType,
INVESTMENTFUNDTYPE AS offering_InvestmentFundType,
ISSUERS.ENTITYNAME AS issuer_EntityName,
RECIPIENTS.RECIPIENTNAME AS recipient_name,
CAST(
  WHEN TOTALOFFERINGAMOUNT = 'Infinite' THEN 0
  ELSE TOTALOFFERINGAMOUNT
END
AS BIGINT) AS TotalOfferingAmount,
FROM OFFERINGS
LEFT JOIN ISSUERS ON OFFERINGS.ACCESSIONNUMBER = ISSUERS.ACCESSIONNUMBER
LEFT JOIN RECIPIENTS ON OFFERINGS.ACCESSIONNUMBER = RECIPIENTS.ACCESSIONNUMBER
WHERE SALE_DATE BETWEEN DATE '2020-01-01' AND current_date()
Problems

Profiling scales poorly in column space. And you can’t sample columns.

Queries are multithreaded, but sequences of queries are serial. So queries need to be fast.

Rendering hundreds of profiles will weigh down the DOM and make the app janky.

A single sparkline is cheap, but hundreds or thousands of sparklines could make the browser janky.

Most profiles don’t actually need to be re-calculated if the query DAG hasn’t changed for that column.

Solutions

query priority queue – only prioritize running queries the user needs to see.

approximation methods – humans can only perceive so much visual accuracy anyway, so don’t be overly precise.

shape-preserving dimensionality reduction methods – M4, AM4, etc

virtualization & tanstack-style queries – only query / render what’s needed in the viewport.

aggressive caching – cache things the user has already seen.

AST-aware profiling

*Coming Soon*
Thanks for listening!

1. follow along at @hamiltonulmer & @rilldata on twitter
2. talk to me @ the Motherduck afterparty
3. visit rilldata.com/careers