

The Future of Fast Databases: Lessons from a Decade of QuestDB

Steve Sklar

Senior Cloud Engineer,, QuestDB



Intro



Team with decades of experience building low-latency trading software and large-scale databases



Fastest growing open source time-series database on Github



Starred

14.4k

Contributors

146

Raised funding from top US Venture Capital firms who invested in:



redis



docker

stripe



GitLab



MuleSoft

coinbase



My fast database definition*:

Designed for performant frequent **multi million record ingestion** and performant frequent **queries over multi billion record datasets**.

* My own biased definition. But this is my talk, so just setting the context

10 years ago the fast data was:



- 1TB RAM, 128-core boxes
- Sending code to the data (stored procedures)
- Sharing by extracting data out of the big box
- Vendor locked language, tools, storage



Fast data now:

- Python, Pandas, Polars, RStudio, Spark, DuckDB
- Open data formats - Parquet
- Freedom to use data locally, local compute
- Leverage scalable, infinite and inexpensive storage infrastructure, such as S3, GCS, Azure Blob

Type III database



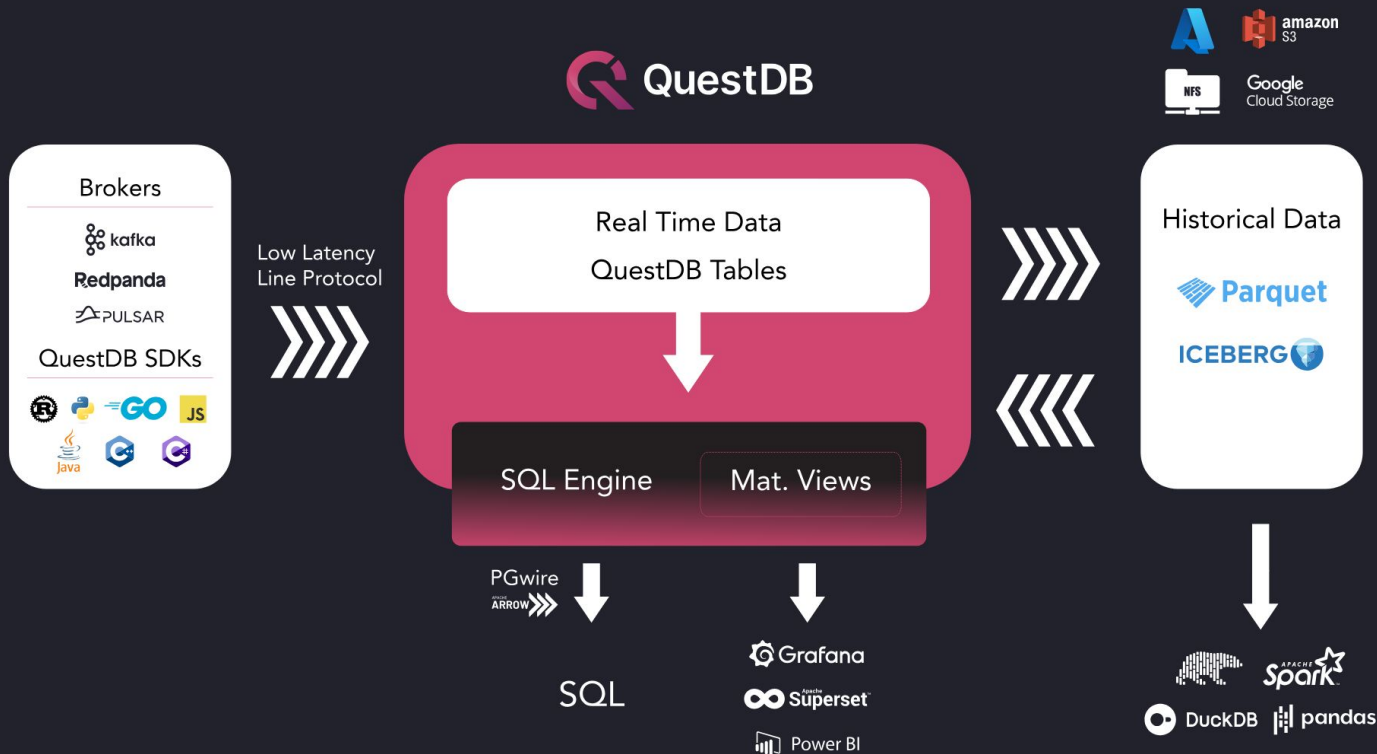
- Does not own data
- Does not force every workload through its APIs
- Does not force all logic to run in one place
- Does speak SQL

Type III database - problems



- Object store is high latency, coarse-grained API storage, consistency issues
- Maintaining Parquet files on object store is clunky and high latency
- Querying Parquet optimally is complex

Introducing QuestDB



questdb / questdb

Q Type ↵ to search

636

35

4

Insights

questdbPublic

Edit Pins

Watch134

Fork1.2k

Starred15k

master81 Branches104 Tags

Go to file

Add file

<> Code

ideoma

feat(sql): change column type SQL to support changing symbol col...ad39e42 · yesterday5,168 Commits

.github	test(pgwire): add nodejs test runner (#5404)	2 weeks ago
.idea	fix(sql): bug in SAMPLE BY queries after a query cache hi...	6 months ago
artifacts	ci(build): azure github release pipeline to work with drafts...	6 months ago
benchmarks	feat(core): speed up WAL small transaction apply 100x (#...	2 weeks ago
ci	chore(core): new qdb-core crate with support for column ...	4 days ago
compat	fix(pgwire): fix handling of sparse bind variables (#5514)	last week
core	feat(sql): change column type SQL to support changing s...	yesterday
examples	build: 8.2.3 (#5499)	2 weeks ago
i18n	docs(core): [i18n] Update README.zh-cn.md (#4967)	5 months ago
pkg/ami/marketplace	perf(sql): stricter column pre-touch in parallel filter querie...	2 weeks ago
utils	build: 8.2.3 (#5499)	2 weeks ago
win64svc	feat(core): configure to roll log files daily by default when ...	9 months ago
.all-contributorsrc	chore(docs): Vietnamese README translation (#4358)	last year

About

QuestDB is a high performance, open-source, time-series database

[questdb.io](#)

java

sql

database

time-series

cpp

grafana

postgresql

simd

olap

market-data

low-latency

financial-analysis

sensor-data

tsdb

real-time-analytics

time-series-database

tick-data

capital-markets

questdb

Readme

Apache-2.0 license

Code of conduct

Security policy

Activity

Custom properties

15k stars

134 watching

1.2k forks

Report repository

<https://github.com/questdb/questdb>

Database popularity



Rank			DBMS	Database Model	Score		
Apr 2025	Mar 2025	Apr 2024			Apr 2025	Mar 2025	Apr 2024
1.	1.	1.	InfluxDB	Time Series, Multi-model	21.54	+0.04	-5.03
2.	2.	↑ 3.	Kdb	Multi-model	6.49	-0.61	-1.21
3.	3.	↓ 2.	Prometheus	Time Series	6.48	+0.10	-1.44
4.	4.	↑ 5.	Graphite	Time Series	4.26	-0.32	-0.50
5.	5.	↓ 4.	TimescaleDB	Time Series, Multi-model	3.59	+0.11	-1.28
6.	6.	↑ 9.	QuestDB	Time Series, Multi-model	3.12	+0.03	+0.64
7.	7.	7.	Apache Druid	Multi-model	3.05	+0.26	-0.24
8.	8.	↓ 6.	DolphinDB	Multi-model	2.20	-0.09	-2.02
9.	9.	↑ 11.	GridDB	Time Series, Multi-model	2.00	+0.01	-0.02
10.	10.	↓ 8.	TDengine	Time Series, Multi-model	1.66	-0.09	-1.01
11.	11.	↓ 10.	RRDtool	Time Series	1.55	+0.02	-0.58
12.	12.	↑ 13.	Fauna	Multi-model	1.46	-0.06	-0.09
13.	↑ 14.	↓ 12.	OpenTSDB	Time Series	1.41	0.00	-0.32
14.	↓ 13.	↑ 15.	Apache IoTDB	Time Series	1.40	-0.04	+0.20
15.	15.	↓ 14.	VictoriaMetrics	Time Series	1.34	-0.04	-0.08

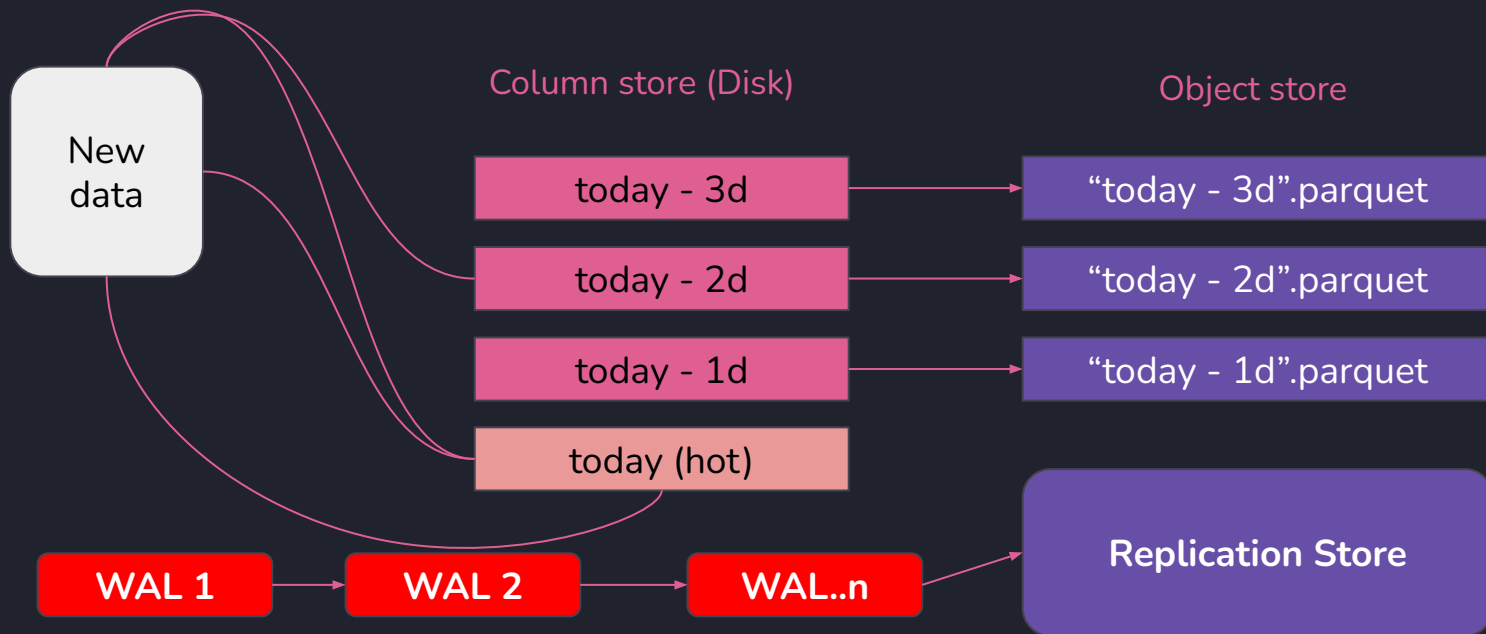
QuestDB is the fastest-growing time-series database

QuestDB - data flow



- High performance, low-latency ingress sink. In-order, out-of-order, dedup, small batches, big batches.
- Direct access to Parquet partitions on object store
- High performance data egress via Apache Arrow

Balancing hot and cold data: the data first mile

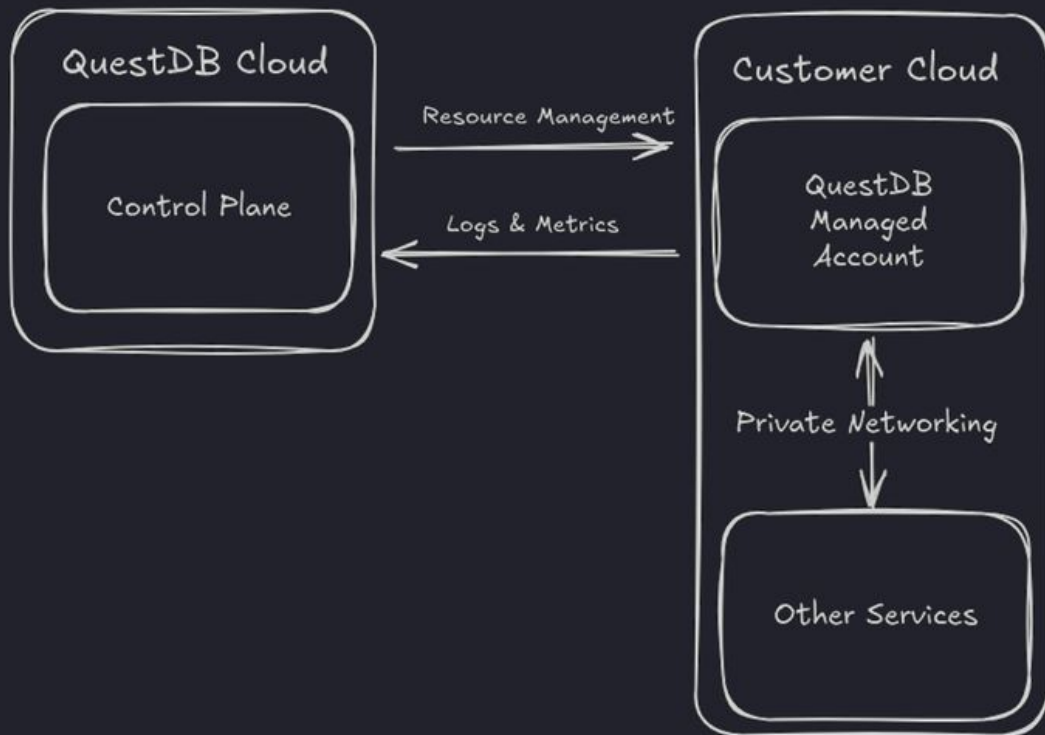


QuestDB - SQL engine



- Parallel, vectorized SQL execution, leverages JIT and data layout
- Low-latency SQL execution, for high-QPS applications
- Materialized views for OHLC and indicators (VWAP, MAVG etc.)
- Spans multiple data formats and storages

QuestDB - Bring Your Own Cloud



QuestDB - built for finance



- Optimised data re-sampling for charts, and mat views
- Coming soon - array data types to support Order Books
- Window functions for finance indicators
- ASOF joins optimised for data troubleshooting

Customer base focused on Finance



Global banks



Aerospace



Hedge funds

BREXAN HOWARD

WORLDQUANT.



Space



Exchanges



Telco



ONE TRADING

Tech



Central banks



Banco de la República
We are the Central Bank of Colombia

Energy



Check our live demo



Web Console

+ Example queries

Run Search docs

History

Tables

- Filter...
- AAPL_orderbook
- ethblocks_json
- trades
- trips
- *weather

SQL

```
1
2 /* This query finds each buy event, and joins with the most
3    recent sell event for the same symbol.
4    Note that we are filtering buys since 2024 and sells since
5    Dec 31st 2023. This is to make sure we always have a previous
6    sell row for each buy event */
7
8 WITH trade_buys AS (
9   SELECT timestamp, symbol, price, side
10  FROM trades
11  WHERE side = 'buy' AND timestamp IN '2024'
12 ), trade_sells AS (
13   SELECT timestamp, symbol, price, side
14  FROM trades
15  WHERE side = 'sell' AND timestamp > '2023-12-31'
16 )
17 SELECT * from trade_buys ASOF JOIN trade_sells ON (symbol);
```

Log [4:38:01 PM GMT+01:00] ✓ 239,442,009 rows in 384ms Execute: 281.98ms Network: 102.02ms Total: 384ms Count: 250.61ms Authentication: 680ns Compile: 2ms

239,442,009 rows

timestamp	symbol	price	side	timestamp1	symbol1	price1	side1
timestamp	symbol	double	symbol	timestamp	symbol	double	symbol
2024-01-01T00:00:00.314299Z	AVAX-USD	38.56	buy	2023-12-31T23:59:46.690587Z	AVAX-USD	38.51	sell
2024-01-01T00:00:00.314299Z	AVAX-USD	38.56	buy	2023-12-31T23:59:46.690587Z	AVAX-USD	38.51	sell
2024-01-01T00:00:00.315867Z	AVAX-USD	38.56	buy	2023-12-31T23:59:46.690587Z	AVAX-USD	38.51	sell
2024-01-01T00:00:00.315867Z	AVAX-USD	38.56	buy	2023-12-31T23:59:46.690587Z	AVAX-USD	38.51	sell

Copyright © 2024 QuestDB

Connected QuestDB 8.1.2

<https://demo.questdb.io/>

<https://github.com/questdb>