pganalyze

One Billion Row Challenge: Postgres, DuckDB, and extensions

Ryan Booz

Ryan Booz

Solutions Engineer
pganalyze

𝕏 @ryanbooz

in /in/ryanbooz

www.softwareandbooz.com

▶ youtube.com/@ryanbooz

Agenda

01 One Billion Row Challenge

02 Analytical Workloads and Big Data

03 Postgres

04 DuckDB

05 Why not both??

06 Demo

pganalyze

01/06

One Billion Row Challenge

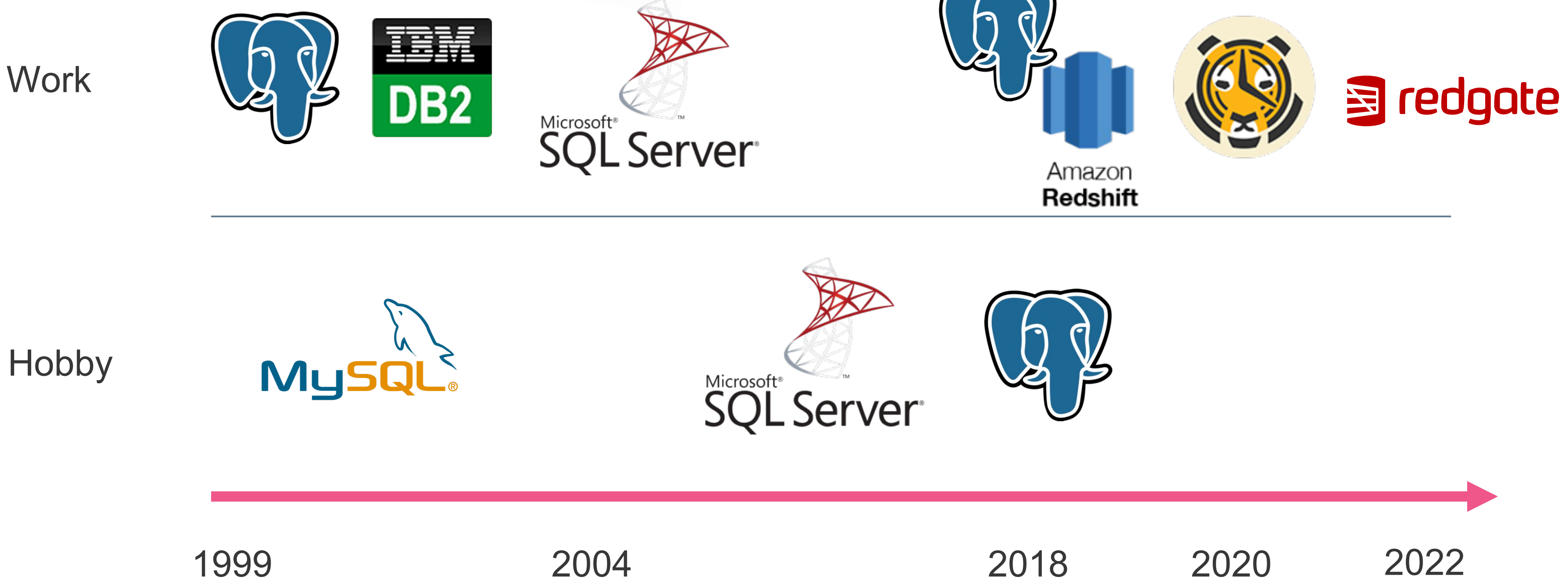
1️⃣ 🐝 🏎️

One Billion Row Challenge

- Gunnar Morling
    - https://github.com/gunnarmorling/1brc
- 1 Billion temperature readings
- Java program to read and aggregate data
- Time process

```
Sochi;27.3
Da Nang;36.3
Kinshasa;29.5
Guatemala City;26.6
Darwin;38.6
Upington;24.5
Ségou;33.6
Nouakchott;19.8
Birao;18.4
Wichita;31.7
Dolisie;25.1
Napier;3.2
Heraklion;23.2
Moscow;13.0
Dili;34.1
Adelaide;2.1
Villahermosa;33.9
```

My Journey



Work

Hobby

1999

2004

2018

2020

2022

Advent of Code
2023

WORDLE

A DAILY WORD GAME

1️⃣ 🐝 🏎️

Postgres?

pganalyze

02/06

Analytical Workloads and Big Data

Big Data?

- Data that can't be normally processed given current, "standard" methods
- Changes over time
- The five V's:
  - `Volume, Velocity, Variety, Veracity, Value`

Big Data?

- The way we process data has changed
  - New storage techniques
  - Primarily aggregated results
  - Older data is processed less and stored more efficiently
  - Storage, processing, and memory are constantly growing
  - Databases software capabilities continue improving

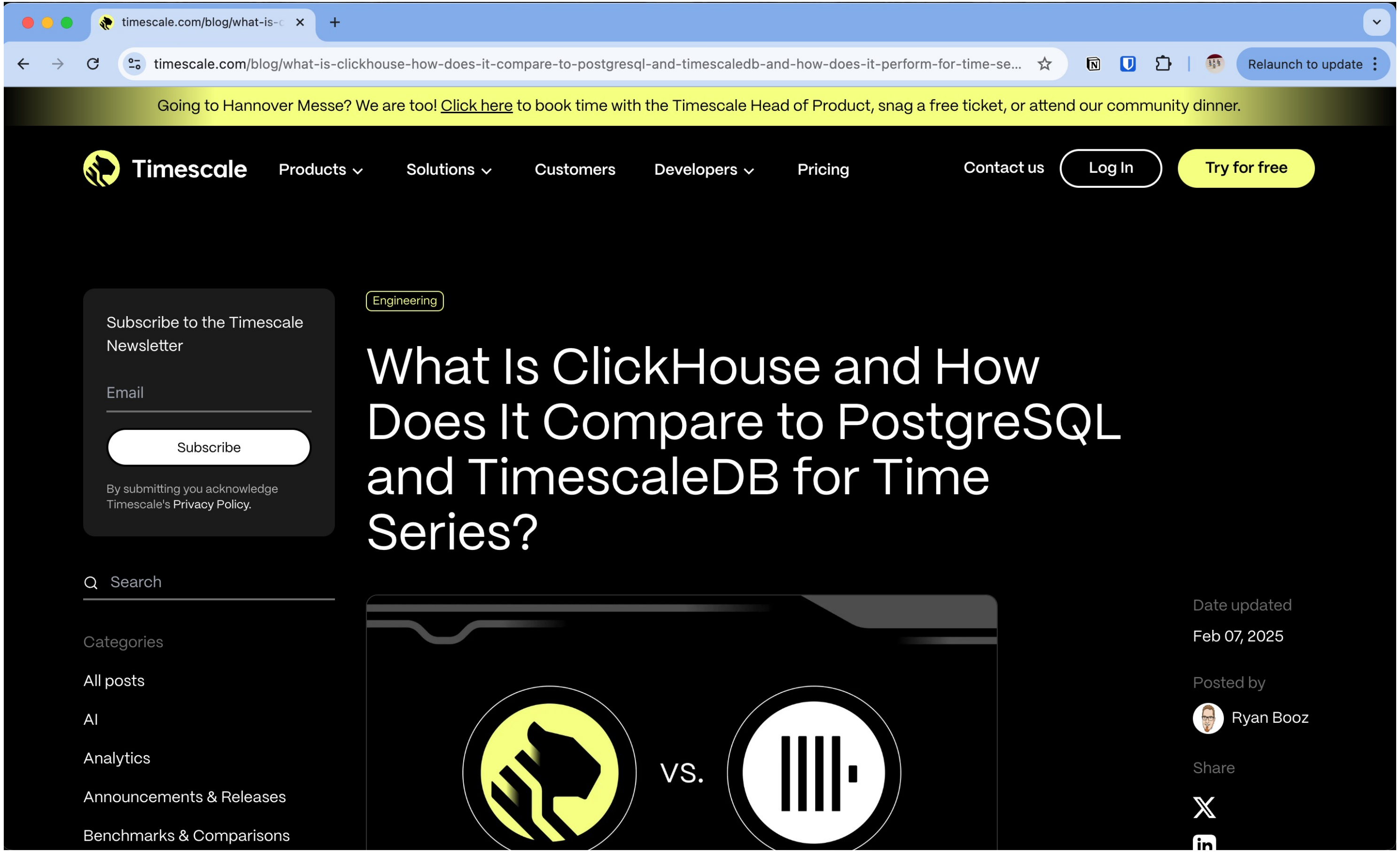
Big Data?

- Data lakes and supporting technology have provided more storage options
    - `Parquet`
    - `Iceburg`
    - `Avro`
    - `Etc.`

pganalyze

03/06

Postgres


timescale.com/blog/what-is-c
timescale.com/blog/what-is-clickhouse-how-does-it-compare-to-postgresql-and-timescaledb-and-how-does-it-perform-for-time-se...
Relaunch to update
Timescale
Products
Solutions
Customers
Developers
Pricing
Contact us
Log In
Try for free
Subscribe
Search
Share
VS.

Going to Hannover Messe? We are too! Click here to book time with the Timescale Head of Product, snag a free ticket, or attend our community dinner.

Subscribe to the Timescale Newsletter

Email

By submitting you acknowledge Timescale's Privacy Policy.

Categories

All posts

AI

Analytics

Announcements & Releases

Benchmarks & Comparisons

Engineering

What Is ClickHouse and How Does It Compare to PostgreSQL and TimescaleDB for Time Series?

Date updated

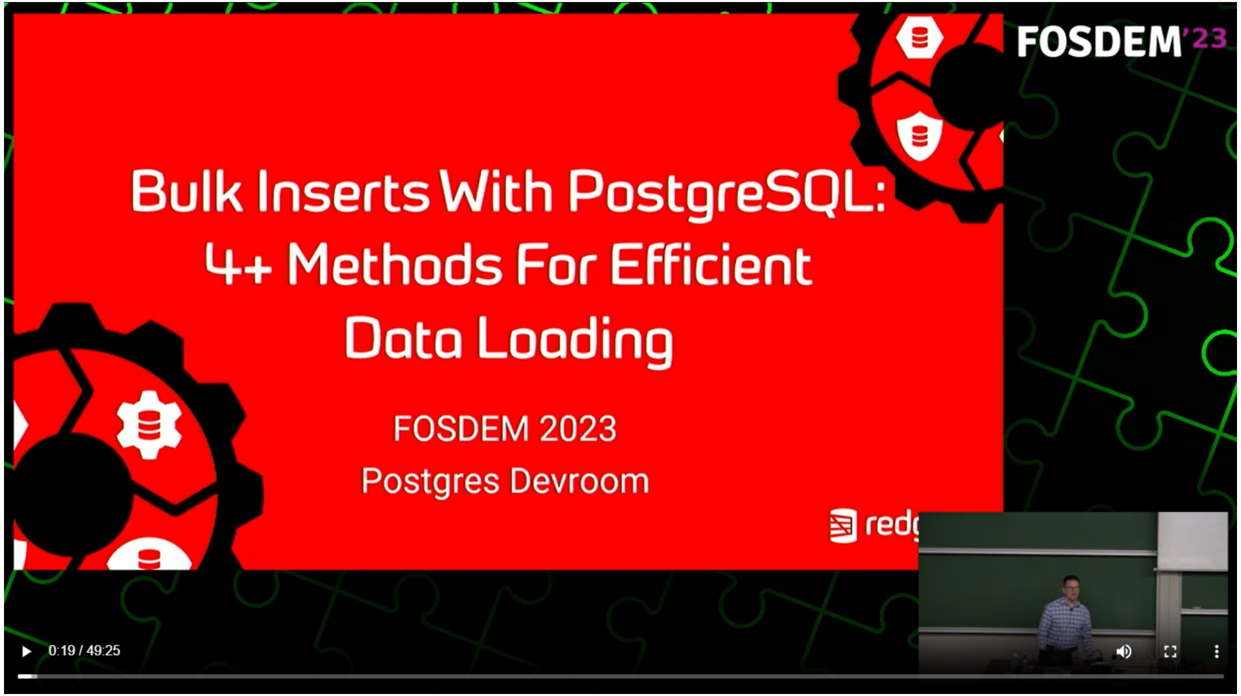
Feb 07, 2025

Posted by

Ryan Booz

Postgres Ingest

- Ingest of large files is challenging
- COPY is single threaded
- Transaction overhead requires batching
- Unlogged tables improve performance at the expense of data safety



FOSDEM'23

Bulk Inserts With PostgreSQL: 4+ Methods For Efficient Data Loading

FOSDEM 2023

Postgres Devroom

https://bit.ly/ryan-booz-2023-talks

Query Processing

- Lots of tuning opportunities available
    - `work_mem`
    - `shared_buffers`
    - `random_page_cost`
    - `max_workers_processes`
    - `max_parallel_workers`
    - `max_parallel_workers_per_gather`

Query Processing

- Still row based
- No batch row retrieval methods in core
- Accessing data (from heap) still requires reading and projecting the entire row
  - `Minus non-queried columns in TOAST`

It’s not columnar

Columnar Data

- Stored as sets of rows per column (vectors)
- Compression within vectors
- Compatible with newer analytic file formats

It’s not vectorized

Vectorized processing

- Process batches of column data with one operation
- Take advantage of technologies like SIMD
- Specialized functions like aggregates to handle vectors

04/06

DuckDB

What is DuckDB?

- Open-source, MIT licensed analytics database
- Developed by National Research Institute in Amsterdam
- Columnar, in-memory, relational database
- Easily embedded
- No dependencies

What is DuckDB?

- Columnar-vectorized query processing
- PostgreSQL parser (libpg_query from pganalyze)
- Extensible
- MVCC
- No concept of roles

It is columnar

It is vectorized

Data Processing

- Natively reads/writes multiple file formats
- Natively connects to block storage like S3
- In memory processing
- Can persist as a table during processing
- In-memory data dies with process

pganalyze

05/06

Why not both?

pg_duckdb

pg_mooncake

pg_duckdb

- Use DuckDB data access functionality
- Use DuckDB execution engine to read data in Postgres tables
- Limited Postgres database write capabilities

pg_mooncake

- Use DuckDB data access functionality
- V2 uses Iceberg storage format for “shadow” table

pganalyze

06/06

DEMO

Conclusions

pganalyze

- Postgres can be tuned to improve native performance
- For the right workloads and query patterns, partitioning is worth serious consideration
- Columnar storage is required for efficient analytical processing
- Vectorized functions are needed to take advantage of columnar data

Conclusions

- DuckDB has a great toolset
- Helpful ETL/ELT and Data Lake functionality
- Integrations with Postgres are still not production ready without more safeguards, particularly around memory management

The future of PostgreSQL analytics is 🎉🔥🦾

🎉 THANK YOU! 🎉

pganalyze

What Questions do you have?

pganalyze

pganalyze

One Billion Row Challenge: Postgres, DuckDB, and extensions

Ryan Booz