

PostgreSQL

Fanavaran Anisa

Iran Linux House

Linux & Open Source Training Center

www.anisa.co.ir



About Me

- BigData.ir (since 1392)
- UT PHD Student & Instructor
- Big Data & Data Engineering Lecturer
- CTO of a private AI powered retailer company
- CDO of Saba Tamin ...
- CTO & Data Architect & Developer in many projects



Course Overview



Introduction to PostgreSQL Course

Getting Started and Basic SQL

Introduction and Installing PostgreSQL

Entity-Relationship (ER) Design

Data Definition Language (DDL)

Data Manipulation Language (DML)

SELECT Query Fundamentals

Joins and CTE

Advanced SQL Techniques

Subqueries and CASE WHEN

Window Functions

Practical Queries(Exercise)

Working With JSON Data

Recursive queries for tree structures/Arrays

Popular SQL/PG functions

Full-Text Search, Vectorization, and

Columnar Storage

Views and Materialized Views

Index Types and Usage Guide

Backup and Recovery

Log Management and Replication

Explain and Query Monitoring

User Access, Security, and Programming

User Access Management

Security and Encryption

Writing PL/PgSQL Code (IF, FOR, WHILE)

Functions and Stored Procedures

Triggers

Lateral Join and Flattening

Internal PostgreSQL Tables

Foreign Data Wrappers (FDW) and Cross-Database
Queries

Postgres Extensions and PostGIS

Distributed PG Using Citus

High Availability

Query Parallelism

Managing Large Databases

modern Online PG Tools

(ParadeDB/RisingWave/QuestDB)



Target Audiences



Who should learn PostgreSQL?

- Data Analyzers
- Software Eng. Student
- Backend Developers
- Data Engineers
- Postgres Fans
-



Why Postgres?



Why Postgres

- Open-source nature
- ACID compliance
- Support for advanced data types
- Extensibility & It's Ecosystem
-

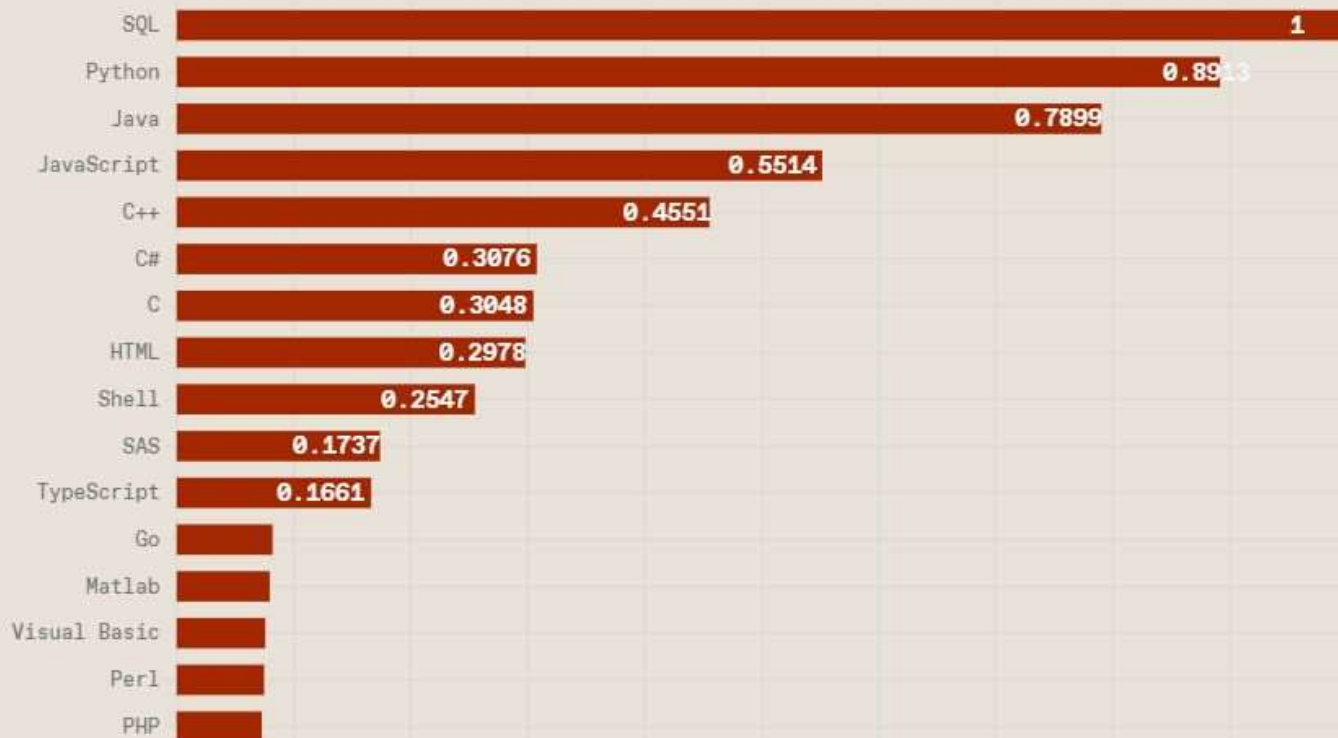


Why Postgres

Top Programming Languages 2023

Click a button to see a differently weighted ranking

Spectrum **Jobs** Trending

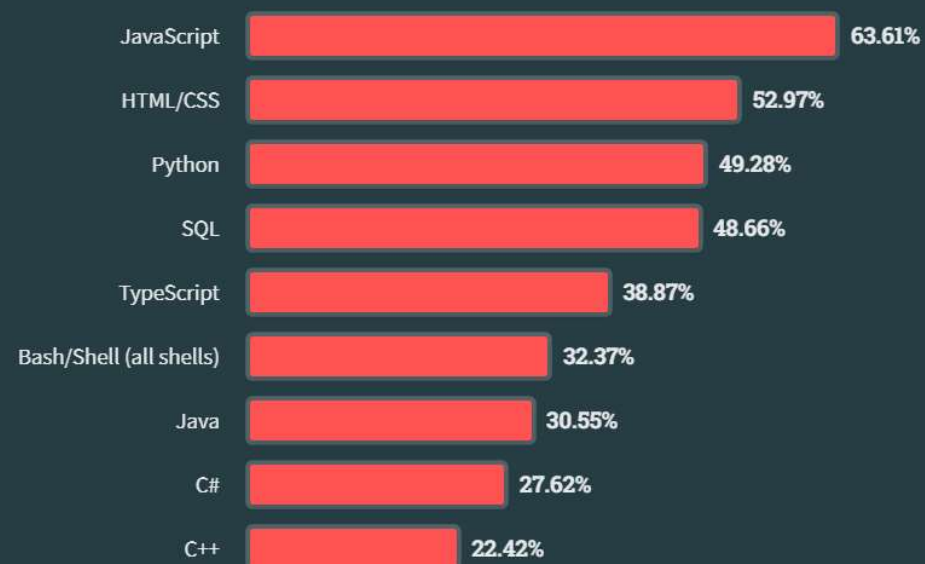


Why Postgres : DB-Engines Db of the Year

Year	Database
2022	Snowflake
2021	Snowflake
2020	PostgreSQL
2019	MySQL
2018	PostgreSQL
2017	PostgreSQL
2016	Microsoft SQL Server
2015	Oracle
2014	MongoDB
2013	MongoDB

2023 -> Postgres!

Why Postgres – StackOverFlow Survey



Overview

Developer Profile

Technology

Most popular technologies

Admired and Desired

Worked with vs. want to work with

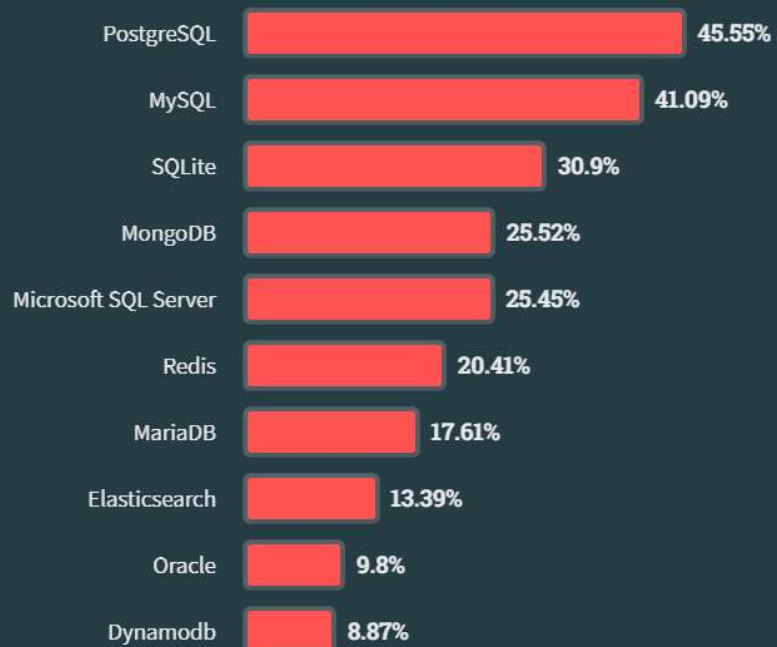
Top paying technologies

AI

Work



Why Postgres – StackOverFlow Survey



Overview

Developer Profile

Technology

Most popular technologies

Admired and Desired

Worked with vs. want to work with
Top paying technologies

AI

Work

Community



Postgres Ecosystem



Is Postgres a One-Size-Fits-ALL?

Postgres is eating the database world

By [Ruohang Feng \(@Vonng\)](#) | [WeChat](#) | [Medium](#) | 2024-03-04

Module: PGSQL

Categories: PostgreSQL

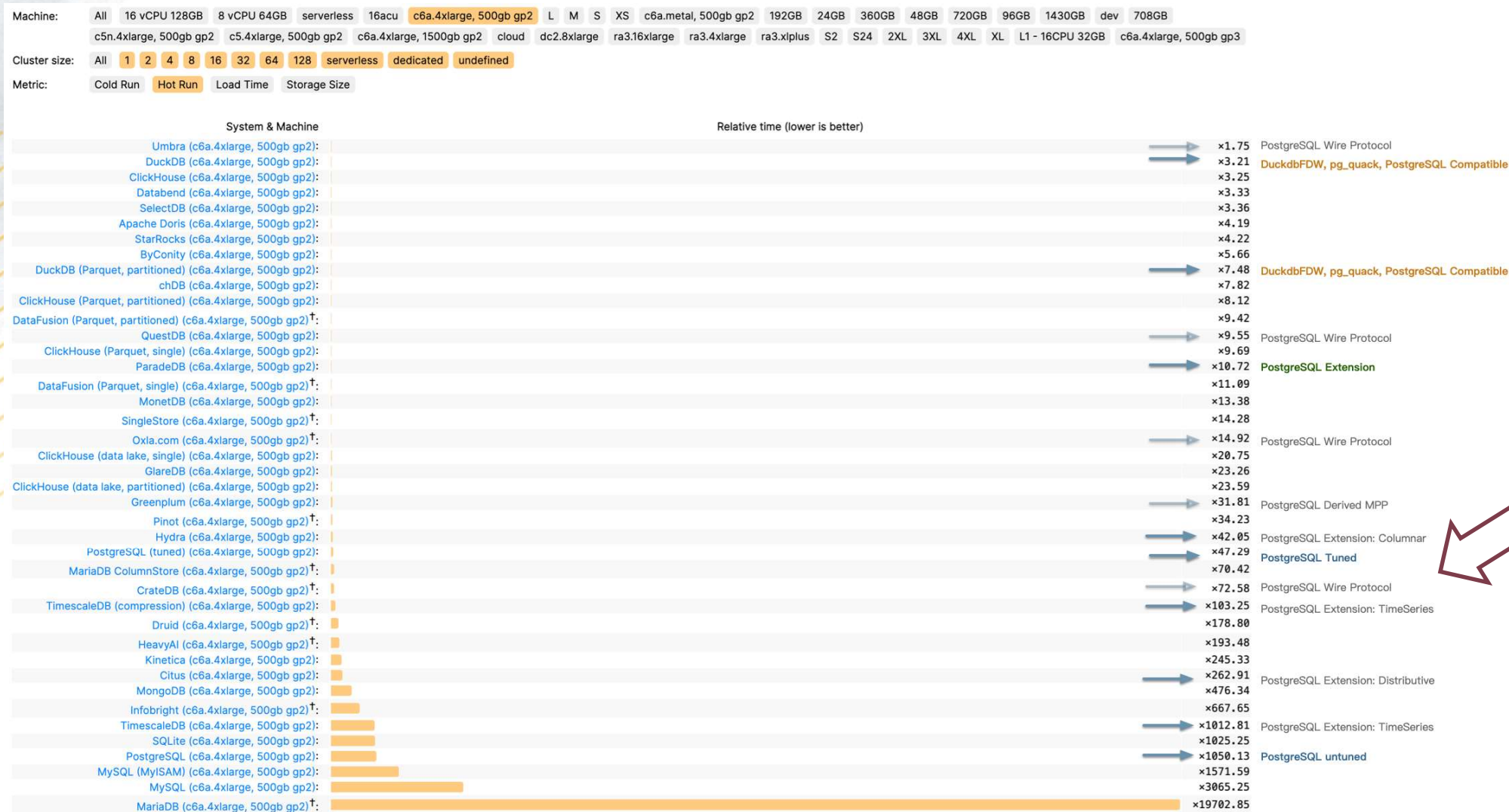
Tags: Ecosystem

PostgreSQL isn't just a simple relational database; it's a data management framework with the potential to engulf the entire database realm. The trend of "Using Postgres for Everything" is no longer limited to a few elite teams but is becoming a mainstream best practice.

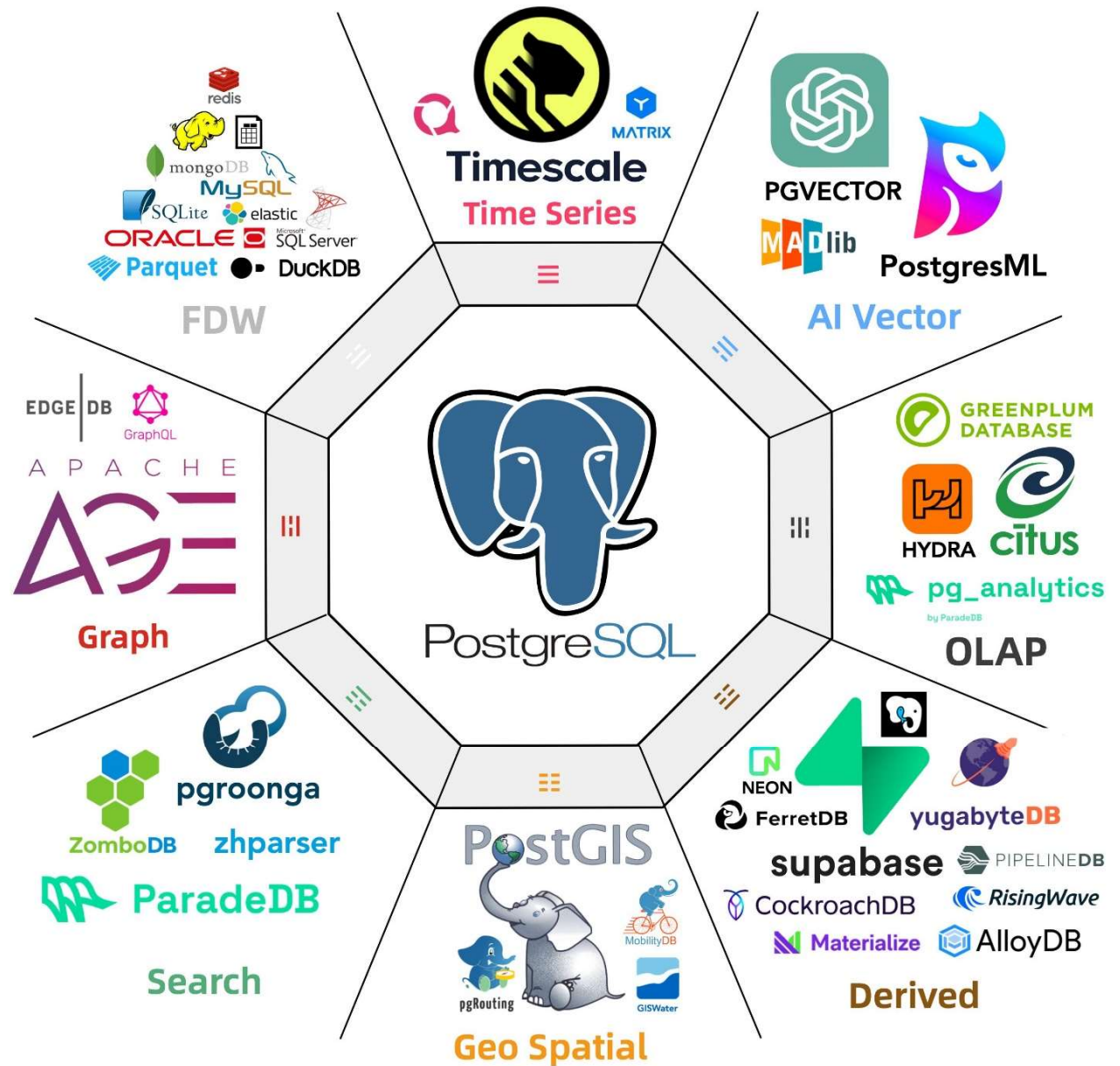
OLAP's New Challenger

In a 2016 database meetup, I argued that a significant gap in the PostgreSQL ecosystem was the lack of a **sufficiently good** columnar storage engine for OLAP workloads. While PostgreSQL itself offers lots of analysis features, its performance in full-scale analysis on larger datasets doesn't quite measure up to dedicated real-time data warehouses.

ClickBench Results – How Many PG is there?



Almost for every Need!



PostgreSQL Extensions



PGXMAN – An Extension Manager

Discover, install, and upgrade
with confidence

terminal

```
> curl -sfl https://install.pgx.sh | sh -
```

```
👉🏻🎉 pgxman successfully installed
```

```
> pgxman install pgvector
```

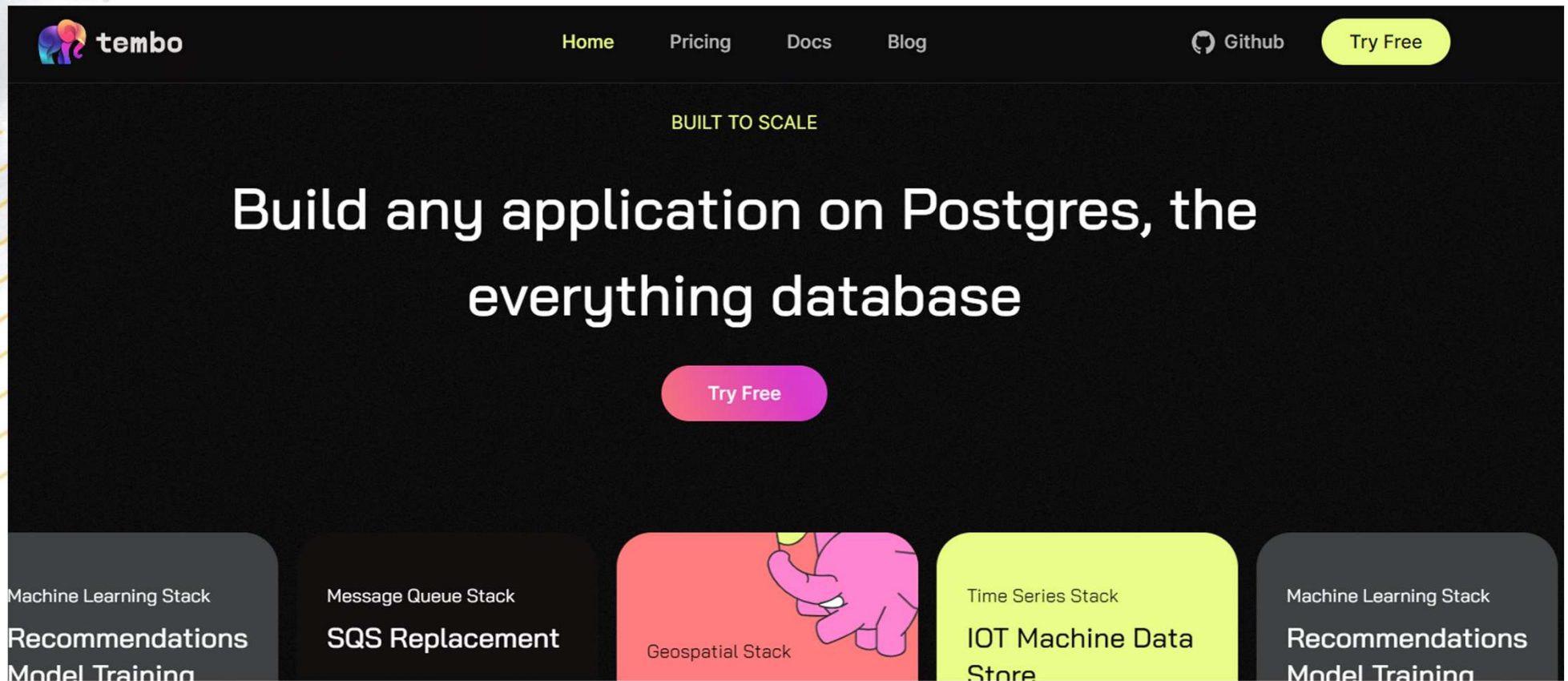
```
The following Debian packages will be installed:
```

```
postgresql-14-pgxman-pgvector=0.5.1
```

```
> Do you want to continue? [Y/n] y
```

```
pgvector has been successfully installed.
```

Tembo

A screenshot of the Tembo website. The header is dark with the Tembo logo (a colorful elephant head) on the left and navigation links (Home, Pricing, Docs, Blog) in the center. On the right, there are links to Github and a 'Try Free' button. The main content area has a dark background with the text 'BUILT TO SCALE' in yellow, followed by 'Build any application on Postgres, the everything database' in white. Below this is a pink 'Try Free' button. At the bottom, there are five colored boxes representing different stacks: Machine Learning Stack (grey), Message Queue Stack (dark grey), Geospatial Stack (pink with a hand holding a pencil), Time Series Stack (yellow), and another Machine Learning Stack (grey). Each box lists specific use cases like Recommendations, Model Training, SQS Replacement, IOT Machine Data Store, etc.

tembo

Home Pricing Docs Blog

Github Try Free

BUILT TO SCALE

Build any application on Postgres, the everything database

Try Free

Machine Learning Stack
Recommendations
Model Training

Message Queue Stack
SQS Replacement

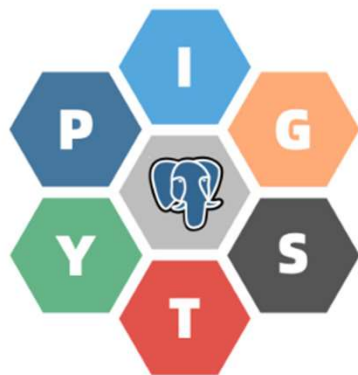
Geospatial Stack

Time Series Stack
IOT Machine Data Store

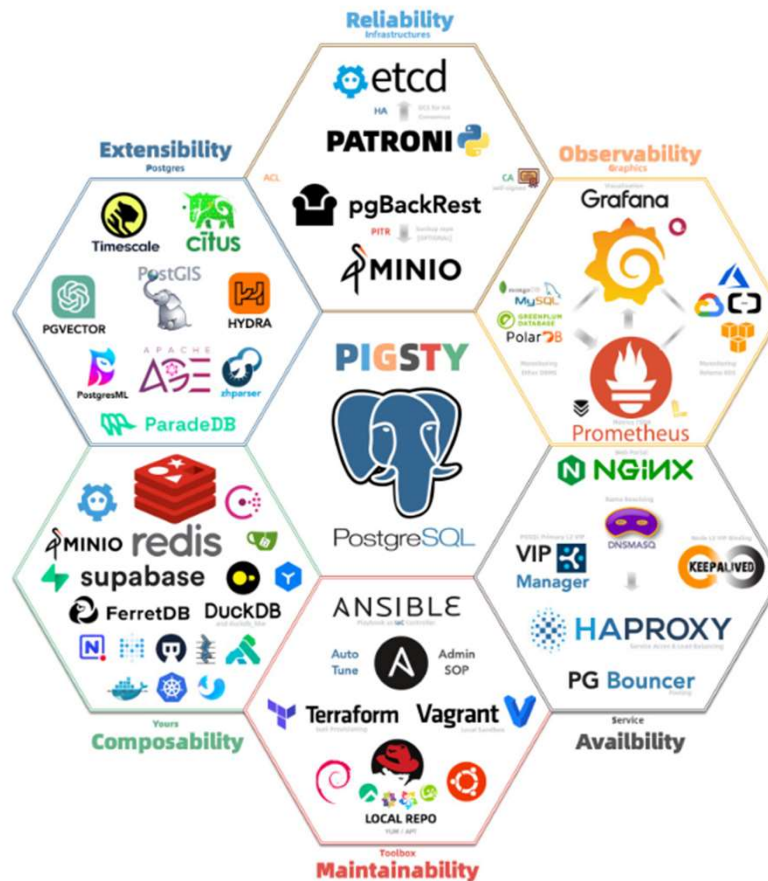
Machine Learning Stack
Recommendations
Model Training

Tembo Stacks for modern data needs

PIGSTY: Battery-Included Postgres



Battery-Included, Local-First
PostgreSQL Distribution as an
Open-Source RDS Alternative



P Extensible
ostgres

I Reliable
nfras

G Observable
raphics

S Available
ervice

T Maintainable
oolbox

Y Composable
ours



Citus : Distributed Postgres

Why Shard Postgres? Performance

See how Citus gives this application ~20X faster transactions and 300X – 150,000X faster analytics queries.



A side-by-side comparison of Citus vs. single-node Postgres, comparing the performance of transactions, analytical queries, and analytical queries with rollups.

ParadeDB : Columnar, Search & Vector Optimized DB

Analytical Performance vs. Postgres

Scores measured by Clickbench



94X Faster than Postgres

ParadeDB brings column-oriented storage and vectorized query execution to Postgres tables. Users can choose between row and column-oriented storage at table creation time.

PG Wire Protocol

A **wire protocol** is the format for interactions between a database server and its clients. It encompasses authentication, sending queries, receiving responses, and so on

Every database needs a protocol to standardize communication with the outside world. PostgreSQL has done a great job of implementing and documenting its own Postgres Frontend/Backend Network Protocol. The openness of its protocol, and the popularity of Postgres itself, led to a flourishing ecosystem of drivers, client libraries and tools that can manage, write to, and read from PostgreSQL databases. The integration ecosystem itself now acts as a strong incentive for new databases to adopt Postgres wire protocol.

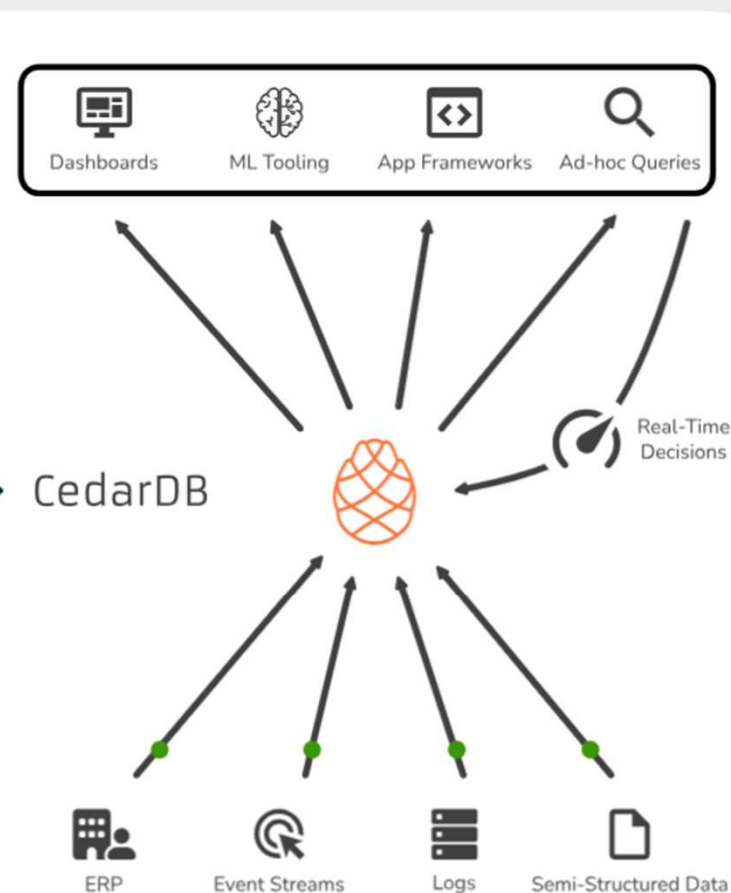


PG Wire Protocol Enabled DBs - CedarDB

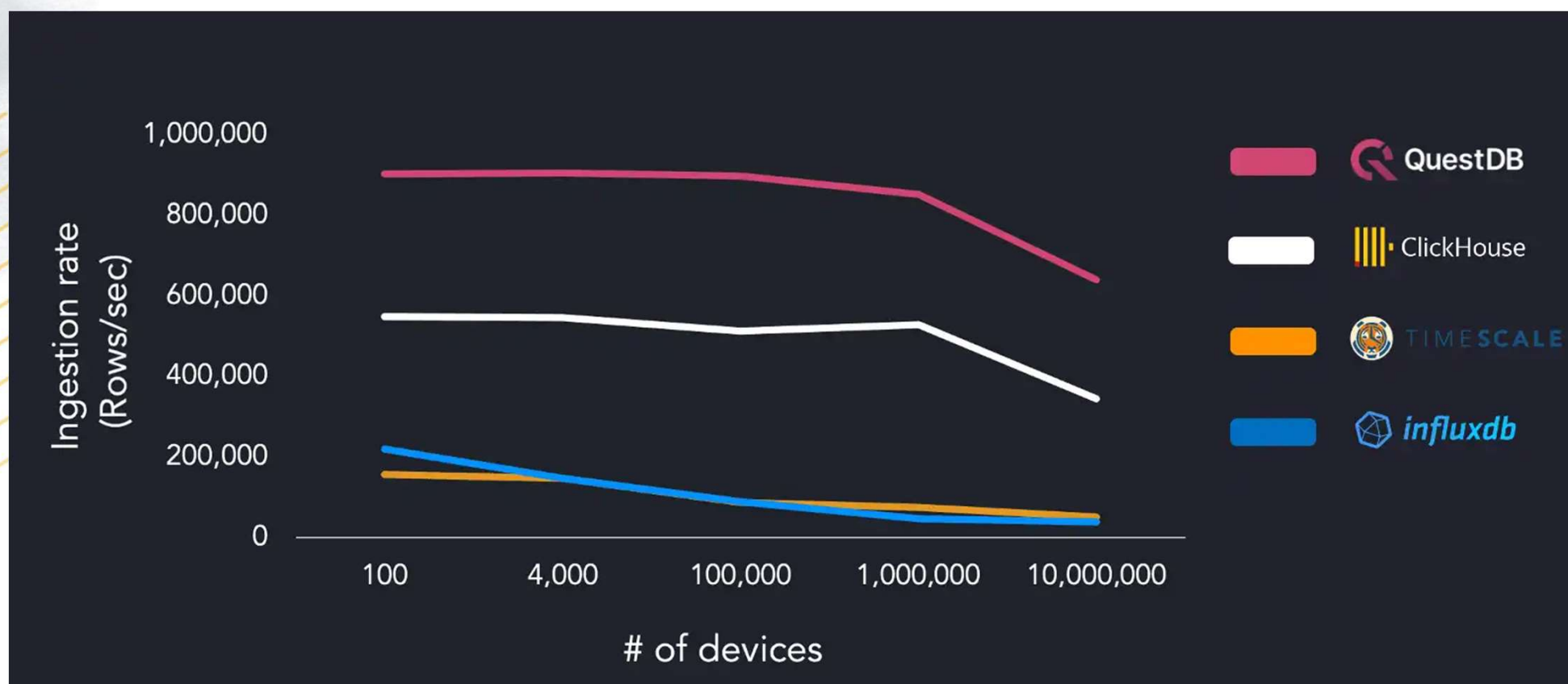


Product

Compa

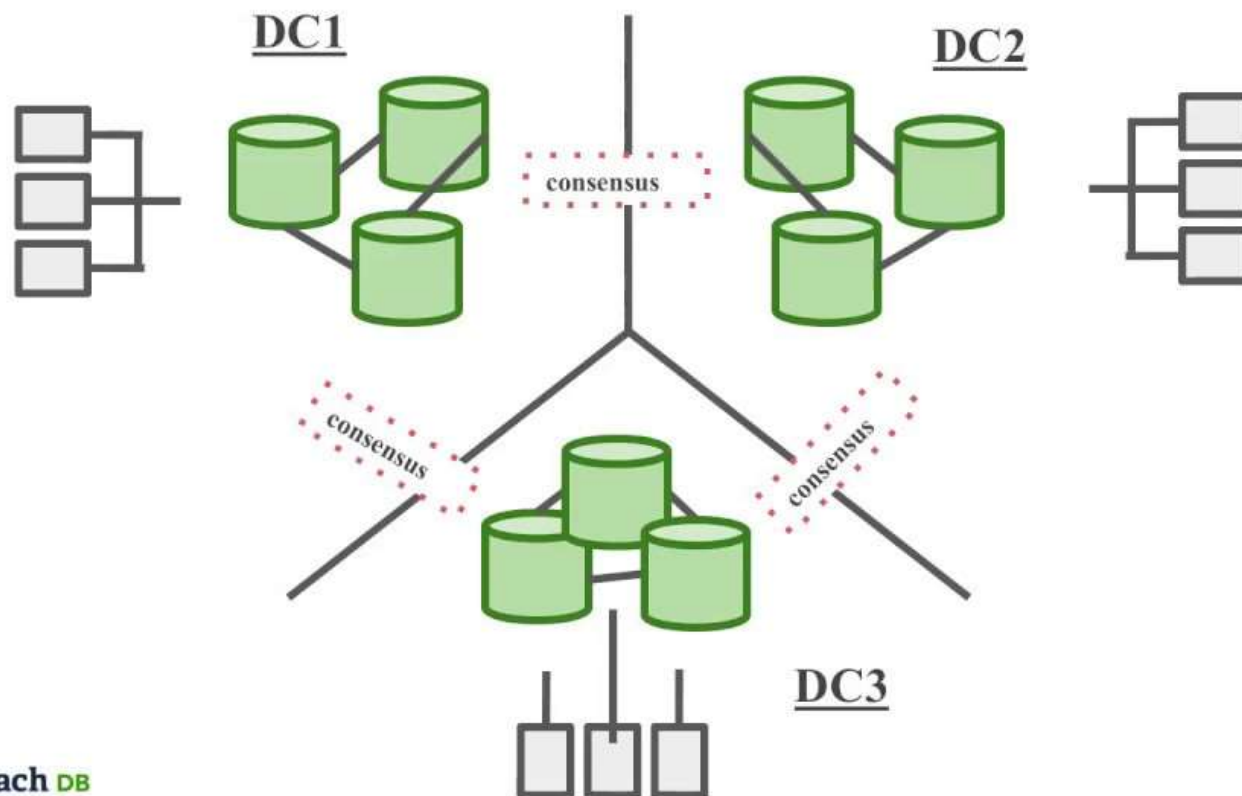


PG Wire Protocol Enabled DBs – QuestDB (a Time Series DB)



CockroachDB – A Distributed RDBS

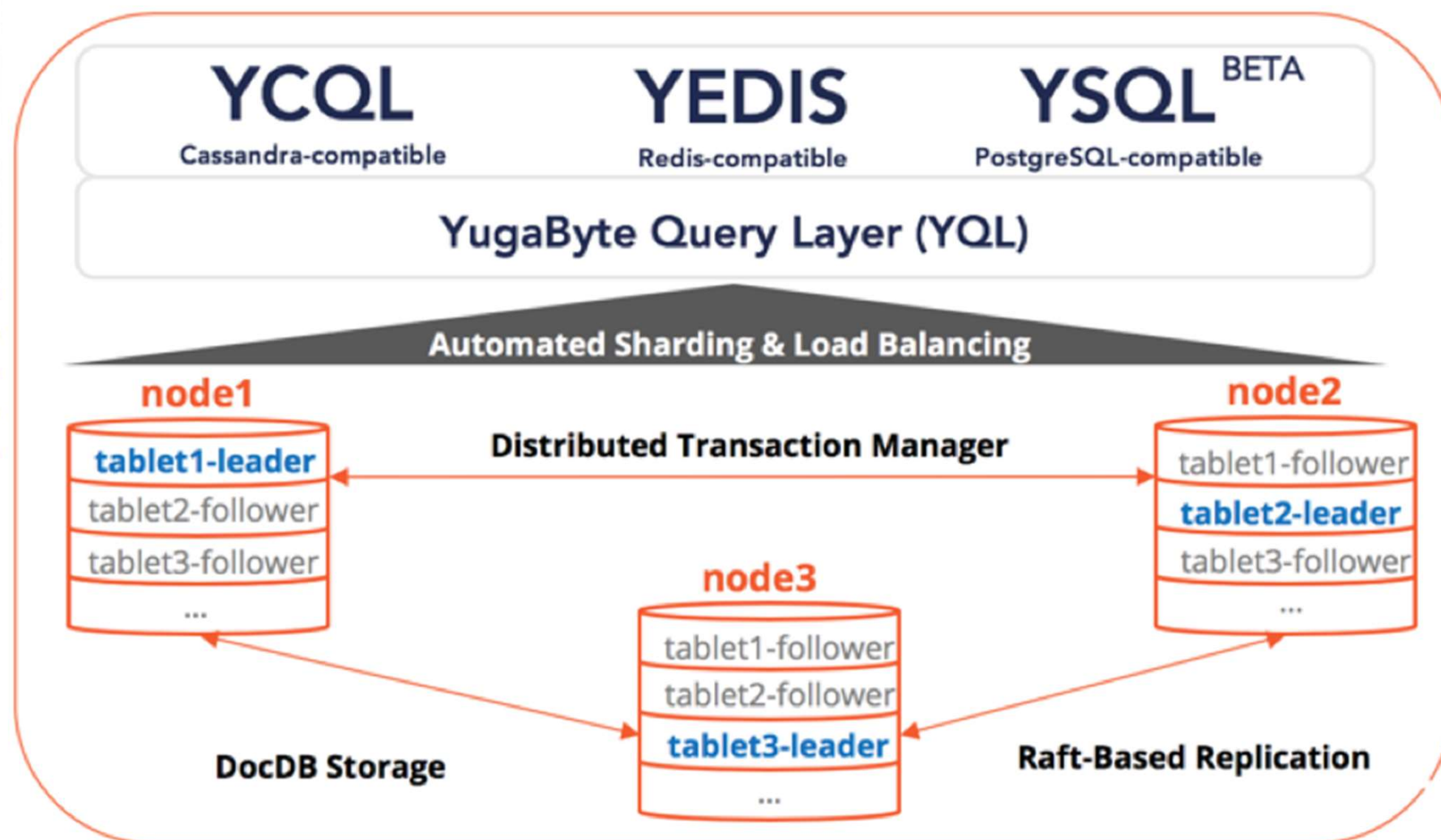
Data Integrity at Scale: Multi-Active Availability



@cockroachdb

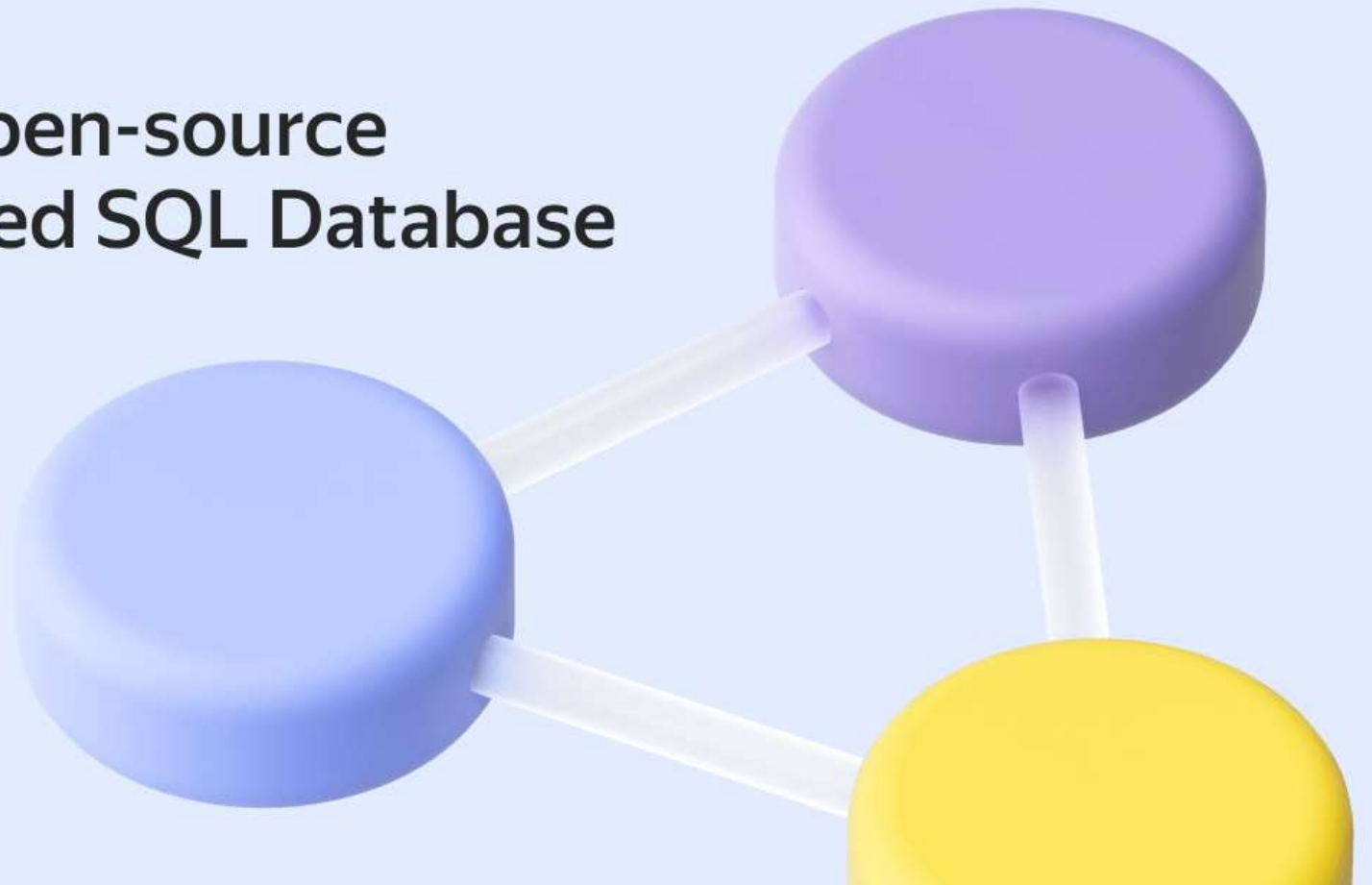


YugaByte – A Multi-Model Distributed DB



YDB : an open source Distributed SQL Database

**YDB — open-source
Distributed SQL Database**



PostgreSQL Extensions or Protocols

PostgreSQL Extensions or Protocols: Architecture Roulette

Should new infrastructure use PostgreSQL as a query engine, or aim for protocol compatibility instead?



CHRIS RICCOMINI
APR 15, 2024



Share

PostgreSQL has a lot of momentum right now. Nearly every startup I talk to is using PostgreSQL or a PostgreSQL-compatible database. Most database vendors offer some form of PostgreSQL compatibility through extensions, protocol compatibility, or PostgreSQL SQL dialect support.

A big part of PostgreSQL's success is its extension ecosystem. Users no longer need to adopt Elasticsearch for search, Pinecone for vector search, Neo4J for graph operations, and Snowflake for online analytical processing (OLAP). Instead, users are opting for PostgreSQL extensions like pg_bm25, pgvector, postgis, pg_analytics, hydra_columnar,



Serverless Postgres Solutions

Serverless Postgres

The fully managed serverless Postgres with a generous free tier. We separate storage and compute to offer autoscaling, branching, and bottomless storage.

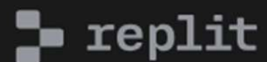
Supabase
Neon.Tech

Start Free

View Pricing

...

Powering 500,000+ databases for developers and partners

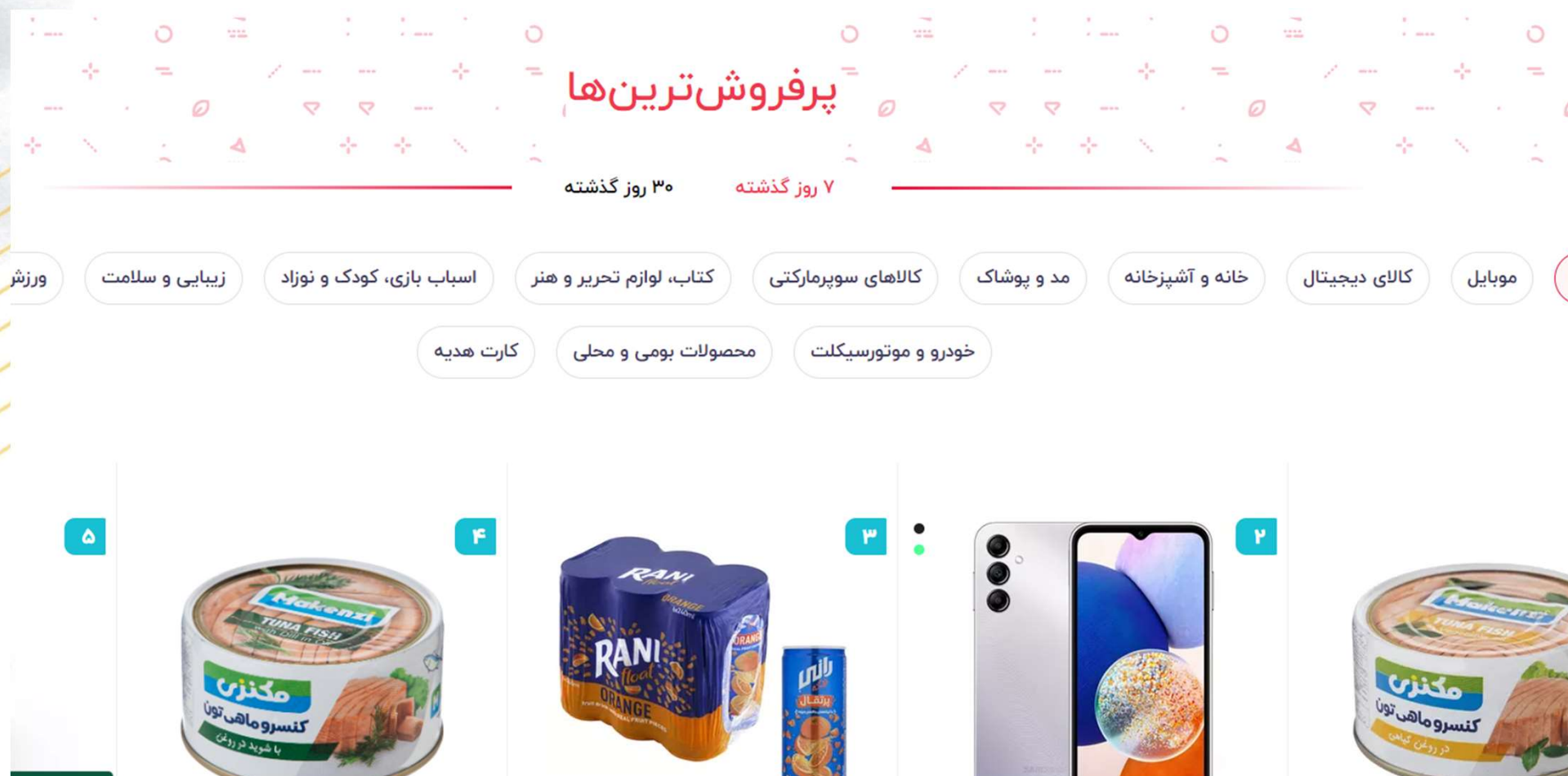


Is Postgres a One-Size-Fits-ALL?

البته این موضوع به این معنی نیست که در طراحی سامانه‌های پیچیده امروزی، تنها به پستگرس اکتفا شود. تجربه و توصیه فعالان این حوزه، استفاده از معماری‌های ترکیبی است که بسته به نیاز از تمامی بانک‌های اطلاعاتی نوین به عنوان اجزای یک سامانه اطلاعاتی بزرگ، استفاده شود. آنچه مدنظر ماست این است که برای هسته اصلی سامانه به شرطی که ماهیت داده‌های آن تراکنشی باشد مثل اکثر سامانه‌های تجاری که نیاز به به ذخیره، به روزرسانی و حذف داده‌های کاربران، محصولات، سفارش‌ها و مانند آن را دارند، از پستگرس در کنار سایر بانک‌های اطلاعاتی غیر رابطه‌ای استفاده شود.



A Practical Polyglot Sample - Digikala



History



Origin & History

- The first implementation of [POSTGRES] began back in 1986 and was put into production in 1988.
 - PostgreSQL evolved from the **Ingres project** at the **University of California, Berkeley**. In 1982, the leader of the Ingres team, **Michael Stonebraker**, left Berkeley to make a proprietary version of Ingres. He returned to Berkeley in 1985, and began a post-Ingres project to address the problems with contemporary database systems that had become increasingly clear during the early 1980s. He won the Turing Award in 2014 for these and other projects, and techniques pioneered in them.
 - After the user community and demands doubled in size in the early 90s, the POSTGRES Project ended and Postgres95, an open-source SQL language interpreter, was launched.
 - Since then, Postgres has continued to receive widespread adoption, especially with the introduction of the public cloud. With each release, there are significant enhancements and improvements providing more functionality and scalability for customer data.
- (<https://www.postgresql.org/docs/current/history.html>)




Origin & History

Version	Year	Key Features
PostgreSQL 6.0	1996	First official PostgreSQL release, Open-source licensing
PostgreSQL 7.1	2001	Introduction of Write-Ahead Logging (WAL), Enhanced query optimizer
PostgreSQL 9.0	2010	Streaming replication, Hot standby
PostgreSQL 12	2019	Advanced indexing, Improved partitioning support
PostgreSQL 13	2020	Enhanced partitioning and indexing, Improved query performance
PostgreSQL 14	2021	Better performance and usability for logical replication and connection handling
PostgreSQL 15	2022	Improved sort performance, JSON enhancements, Incremental sorting
PostgreSQL 16	2023	expanded SQL/JSON syntax , advanced monitoring statistics, and refined access control mechanisms, ensuring efficient policy management across extensive deployments.



ORDBMS vs DBMS

Feature	ORDBMS (Object-Relational DBMS)	RDBMS (Relational DBMS)
Data Model	Extends relational model with object-oriented features.	Purely relational model.
Complexity	More complex, handles complex data types.	Simpler, primarily for structured data.
Use Case	Suitable for applications requiring complex data representation (like CAD, multimedia).	Ideal for transactional and operational databases with structured data.
Query Language	Extensions to SQL for object-oriented features.	Standard SQL.
Performance	Can be slower due to complexity.	Generally faster for simple queries.
Example	PostgreSQL, Oracle. 	MySQL, SQLite.



Section Overview



Any Question ?

