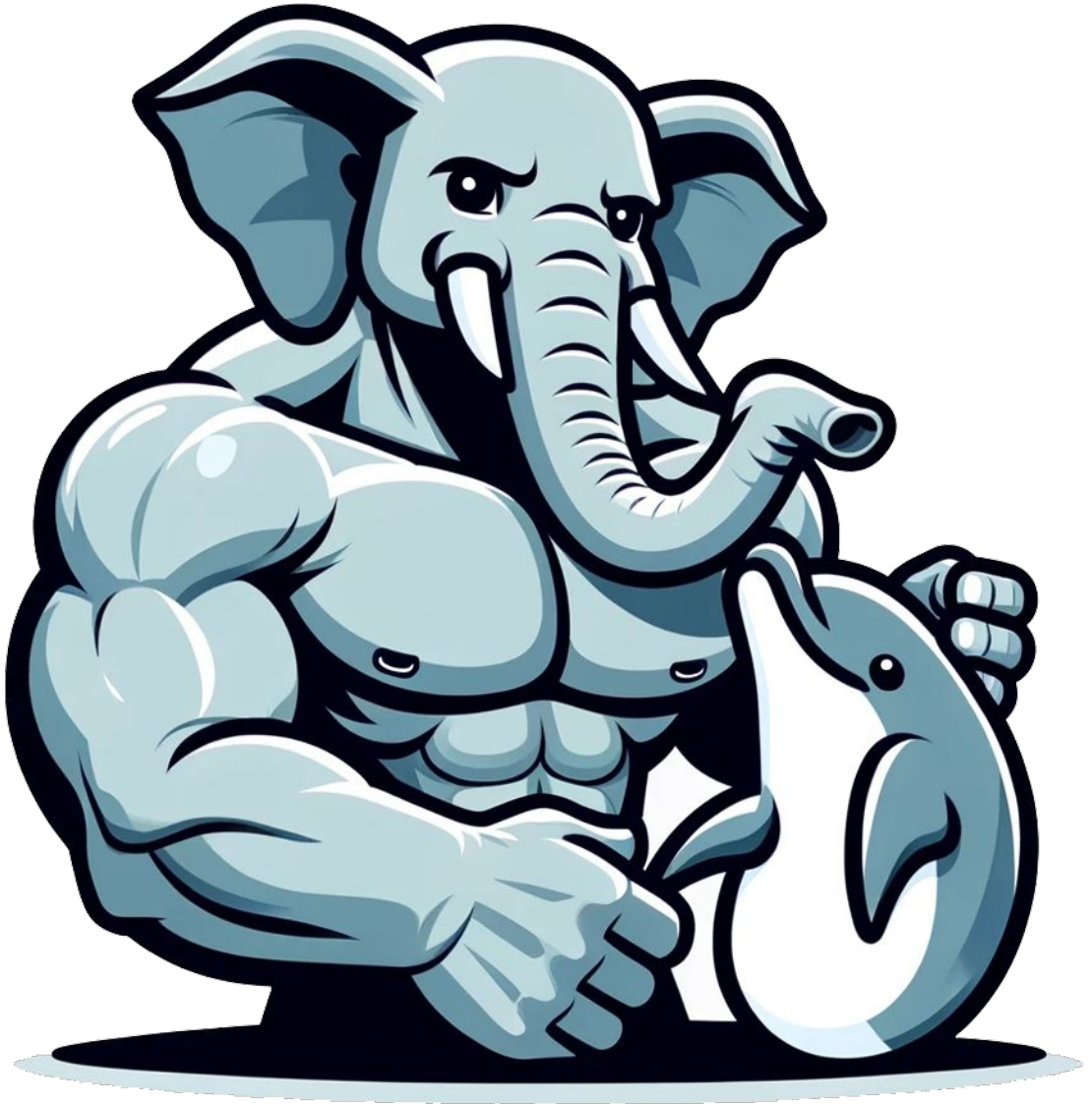
The Future of MySQL is Postgres

Breathing New Life into MySQL Apps With Advanced Postgres Capabilities



Jonah Harris & Mason Sharp

Postgres Conference 2024

nextgres.com
info@nextgres.com

NEXTGRES

We've been in data management for a while

Jonah Harris
Co-Founder, CEO

CTO @ MariaDB (NYSE)
CTO & Dir. AI/ML @ The Meet Group (NASDAQ)
Founding Engineer @ EDB
Recognized Oracle Database Expert
Longstanding PostgreSQL Contributor

Mason Sharp
Co-Founder, CTO

Two-Time Distributed Database Entrepreneur
VP of Engineering @ MariaDB
Sr. Architect & Engineer @ Immuta, FutureWei, TransLattice, EDB
PostgreSQL NYC and SV Meetup Co-organizer

NEXTGRES

For the last 10 years, Postgres grew at a steady rate of about 10% per year.

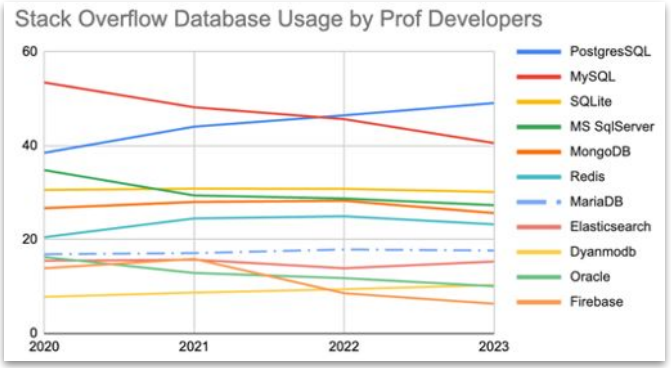
Source: DB-Engines PostgreSQL Trend (2013–2023)

NEXTGRES

Postgres is winning hearts and minds

It's Loved By Developers

Developers love PostgreSQL for its stability, advanced features, and strong community support, making it their go-to database for modern applications.



It's Trusted By Businesses

Businesses trust PostgreSQL for its proven reliability, security, and high-performance data management, making it a go-to solution for enterprise needs.

It's Backed By Investors

Investors back PostgreSQL as the optimal foundation for high-growth database ventures, prizing it for its robustness, extensibility, and vibrant ecosystem.

NEON Timescale HYDRA
tembo Nile EDB
yugabyteDB supabase

NEXTGRES

Despite Postgres' impressive growth, MySQL remains the most popular open source database.

Source: DB-Engines Complete Ranking

NEXTGRES

MySQL remains a top choice for web dev

It's the #1 Open Source Database

Rank			DBMS
Apr 2024	Mar 2024	Apr 2023	
1.	1.	1.	Oracle
2.	2.	2.	MySQL
3.	3.	3.	Microsoft SQL Server
4.	4.	4.	PostgreSQL
5.	5.	5.	MongoDB

It's Essential To Large Businesses

It's Heavily Used By Open Source Apps

NEXTGRES

The market supports MySQL's large user base

Official

Non-Official On-Prem

Cloud

NEXTGRES

While this success was hard-won,
work remains to be done.

NEXTGRES

Postgres has advanced features, but needs to improve some of the basics

Storage	Wraparound	Replication & High Availability	Internal Instrumentation
Secondary Lookups	Vacuuming	Limited Built-in Options	Limited Visibility
Large Objects	Perf Degradation	Complex Setup	Tuning Difficulty
Bloat	System Stalls	Automated Failover	External Dependencies

A Land of Commercial Opportunities

NEXTGRES

MySQL gets ease of use right, but is hampered by its architecture

Extensibility	Mixed Workloads	Advanced Features	Large Databases
Limited Plug-ins	Query Optimization	Limited Indexing	Large Table Performance
Storage Engine Support	Resource Contention	Basic Partitioning	Single-Threaded Design
Sparse Extension Docs	Limited Concurrency	Analytical Functions	Materialized Views

NEXTGRES

Each database has unique strengths and can learn from the other.

NEXTGRES

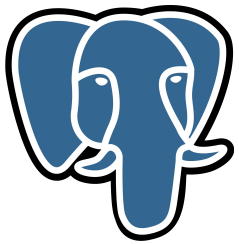
Each database prioritizes different things

Beginner Friendly & Extensive Documentation

Read-Mostly Operations & Simple Transactions

Web-Based Applications & Simpler Workloads

90% of the world's top websites
are powered by MySQL



Advanced Features & SQL Compliance

Complex Read/Write Operations & Large Transactions

Enterprise Applications & Complex Workloads

Postgres is the primary database enterprises
migrate to from commercial systems

NEXTGRES

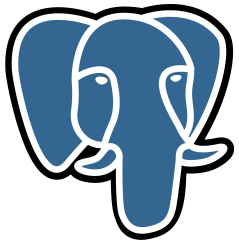
Their ecosystems are quite different

Entrenched Applications

In Decline

No Development Community

Recent Advancements Only In Non-OSS & Cloud (HeatWave)



Larger Ecosystem

On The Rise

Large Multi-Source Development Community

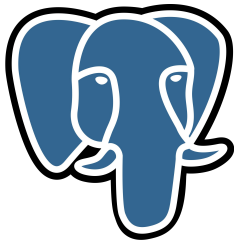
Large Advancements

NEXTGRES

There are big differences on ownership and licensing

IP Owned by Oracle

Licensed Commercially & Under the GPL



IP Owned by Community

Licensed Under the BSD

NEXTGRES

Oracle's focus on proprietary MySQL offerings pushes users to alternatives.

NEXTGRES

The path forward for MySQL users is limited

MariaDB

- Benefits
  - Actively Developed
  - Strong User Community
  - Supported by Linux Distros
  - Backed by a Foundation
  - GPL (if OSS matters to you)
- Disadvantages
  - MySQL 8+ Incompatibilities
  - GPL (if distribution matters)
  - Mostly Single Source

MySQL Clones & Compatibles

- Benefits
  - Actively Developed
  - Mostly Open Source
- Disadvantages
  - Single Source
  - Small User Community

- Benefits
  - Actively Developed
  - Strong User & Developer Community
  - Supported by Linux Distros
  - Backed by a Foundation
  - BSD License (best for all)
- Disadvantages
  - Migration Difficulties

NEXTGRES

This is Postgres Conference, damn it!

Let’s talk about migrating to Postgres.

NEXTGRES

Understanding MySQL vs Postgres data types

- Numeric
  - INT, SMALLINT, BIGINT
    - Maps directly
    - If AUTONUMBER is used, use SERIAL, SMALLSERIAL, BIGSERIAL
  - DECIMAL, NUMERIC
    - Maps directly
  - TINYINT -> SMALLINT
  - FLOAT -> REAL
  - DOUBLE -> DOUBLE PRECISION

NEXTGRES

Understanding MySQL vs Postgres data types

Continued

- String
  - CHAR, VARCHAR
    - Maps directly
  - TEXT, MEDIUMTEXT, LONGTEXT, VARCHAR(max) -> TEXT

NEXTGRES

Understanding MySQL vs Postgres data types

Continued

- Date & Time
  - DATE, TIME
    - Maps directly
  - TIMESTAMP
    - Maps directly
    - MySQL range: '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07'
    - Postgres: 4713 BC to 294276 AD
    - Beware of 0000-00-0000
  - DATETIME -> TIMESTAMP
    - MySQL range: '1000-01-01 00:00:00' to '9999-12-31 23:59:59'
  - YEAR -> SMALLINT, INT

NEXTGRES

Understanding MySQL vs Postgres data types

Continued

- Binary / BLOB types
  - BLOB (16 k), MEDIUMBLOB (16 MB), LONGBLOB (4 GB) -> BYTEA (1 GB)
- SET
  - Use VARCHAR column with a CHECK constraint that enforces the allowed set of values
- ENUM
  - Maps directly, or use VARCHAR with a constraint

NEXTGRES

Choosing a migration approach depends on size & availability

- Offline: have maintenance window, applications inaccessible
- Online: minimize downtime
  - Capture changes to source database
  - Backup database
  - Transform DDL and data
  - Create target database
  - Load data
  - Apply captured changes to source database
    - May need to repeat
  - Pause writes/access source database
  - Cut over

Application user pattern:

- Have one read-write user connection pool and one read-only user connection pool.
- Load balance reads.
- REVOKE write permission for read-write user in source database

NEXTGRES

How to Migrate – Offline: Manual

- **`mysqldump`**
  - **`-compatible=ansi`** will get Postgres-friendlier output
- Edit dumped DDL definitions and data.
  - 3rd party tools
    - FromMySqlToPostgreSql
    - mysql-postgresql-converter
  - May encounter backtick ` issues. In PostgreSQL `` escapes them
- Load into Postgres

NEXTGRES

Performing offline migration with pgloader

- pgloader
  - Many options, such as doing subset
  - Can use MySQL as a source database in a single, easy step! (thank you, Dimitri Fountaine)

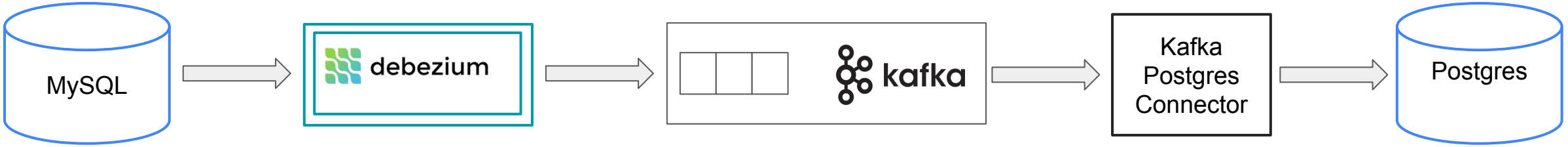
```
pgloader mysql://root:rootpw@localhost/wordpress
         pgsql://postgres:postgres@localhost:5432/wordpress
```

Produces error report of failures

NEXTGRES

Performing online migration with change data capture (CDC)

- Triggers
- Debezium



NEXTGRES

Testing your migration

- Validate Data
  - Validate that the source and destination table data match

- Don’t forget to test performance. Ideally, have production-like workload to test against, including database size
  - Using only a subset of data may not uncover
    - Bad query plans
    - Indexing issues including missing indexes
    - Bloat / issues with vacuum requiring tuning

NEXTGRES

Testing your migration

Continued

- Tune postgresql.conf
  - Here, too, the test server should be similar to the production to make sure it is tuned properly

NEXTGRES

Testing your migration with target architecture

- If using MySQL replication with something like ProxySQL, test a similar configuration with Postgres replication and load balancer like pgCat
- Test HA (e.g., patroni)
- Test backups

NEXTGRES

We've already spent too much time discussing data migration without addressing application changes.

NEXTGRES

That's why we're building a solution that tackles MySQL data and apps for you, and gives your apps additional benefits.

NEXTGRES

We've been managing, developing against, and developing for Postgres for 20+ years each

Started with PostgreSQL 6.5

Community Contributions & EDB Compatibility/Perf Features

Led the Database Team @ The Meet Group

MariaDB PostgreSQL With Xpand (Xgres)

Started with PostgreSQL 8.0

ExtenDB/GridSQL & StormDB/Postgres-XL

Distributed Consistent Database @ FutureWei

Immuta FDW

NEXTGRES

We've also been doing database compatibility for 20+ years

1999

The first version of NEXTGRES was created, compatible with Oracle Database.

2004

Presented a demo at The Ingres Conference of PL/SQL support added to Ingres r3 with a front-end compatible with SQL*Net.

2005

Joined EDB, licensing some NEXTGRES components, adding Oracle Database compatibility features, and working on joint project to add PL/SQL to IBM DB2.

2009

The first version of MySQL compatibility was written as a man-in-the-middle proxy, in front of an Oracle Database or Postgres backend.

2019

The proxy-based MySQL was replaced with a version integrated within Postgres.

2024

Major clean-ups and rewriting background worker listeners to use AWS Babelfish hooks.

NEXTGRES

We want to bring three major MySQL benefits to the Postgres world

Monty's 15 Minute Rule

Many users abandon software that takes too long to set up and become operational.

Managing databases himself and caring about the user experience, Monty implemented a "15 Minute Rule" to ensure installation, configuration, and key features were usable within 15 minutes.

This necessitates items such as better built-in replication options.

High Performance for Read Workloads

Postgres struggles with performance under heavy read loads and simple transactions.

MySQL is renowned for its fast read operations, making it more suitable for web applications that require quick data retrieval.

Development & Admin Tools

Postgres lacks comprehensive tools for development and administration, limiting productivity and ease of use.

The MySQL ecosystem offers a wide array of development and administrative tools, enhancing productivity and simplifying database management.

NEXTGRES

We equally want to bring three major Postgres benefits to the MySQL world

Support for Multiple Procedural Langs

MySQL was limited to SQL/PSM, which didn't gain much adoption, and there's only a simple preview of JavaScript.

Postgres allows substantial work to be done inside the database using almost any language you prefer, keeping the data and business logic close together. This could make MySQL apps even faster.

Large Database Support

MySQL lacks advanced indexing, partitioning, and analytical functions, materialized views and parallel query.

Postgres supports all of these things out of the box, with additional improvements from extensions in the ecosystem, meaning more can be done than was possible inside OSS MySQL.

Access to Extensions

MySQL primarily has a user, not developer, community, resulting in few external advancements to its core.

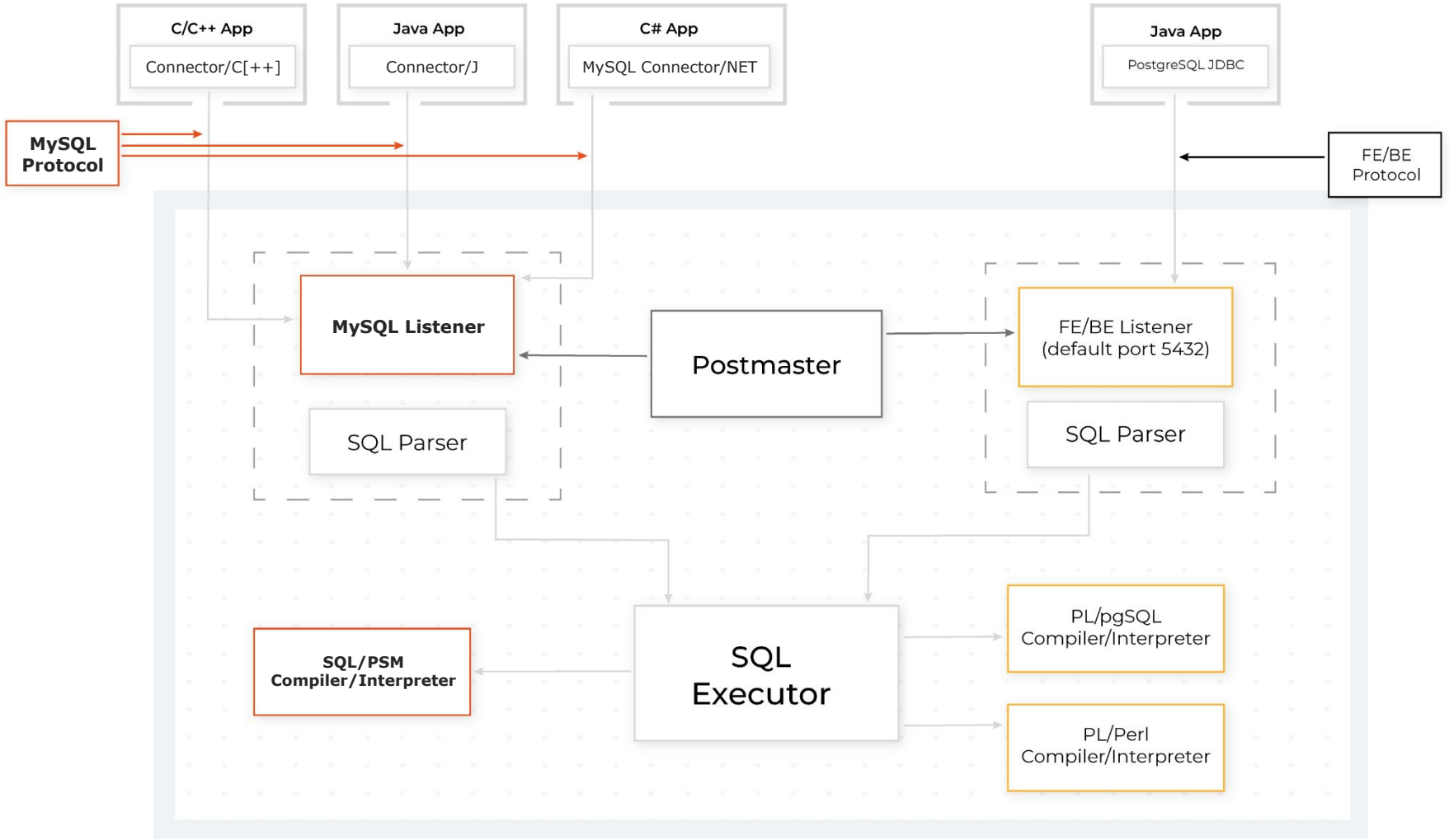
Postgres has extensions for practically everything, created by users and developers working with it and solving real-world problems that are not addressed by the official project.

NEXTGRES

We want to do this with zero application changes

NEXTGRES

We're going to do this with a Postgres extension



We're replacing our background worker listener with AWS' Babelfish Hooks

NEXTGRES

What does development of the extension look like?

We Require A Patched Postgres

Because we're swapping out our old background worker-based listener approach to use the AWS Babelfish hooks, we require the same patched version of Postgres they have.

Having proven their value to multiple Postgres users and developers, our goal is to work with AWS and the community to get their hooks accepted into baseline Postgres.

Docker-Based Development

We're making heavy use of Docker for development, so everyone can get the exact same environment. This also allows us, and others, to connect the most commonly used open source MySQL apps to it, as they too have mature Docker containers.

Developing On GitHub

Just Created Public Repo

Follow Along or Contribute

Star & Watch

https://github.com/nextgres/nextgres

NEXTGRES

The main benefits

MySQL Compatibility

Wire Protocol

SQL Syntax

Procedural Languages

Advanced Capabilities

Columnar Support

Vector Search & AI/ML

Extensions

Database Consolidation

One System to Manage

Improved Productivity

Simplified Operations

Active Community

Multi-Source

Open Source

Growing Market

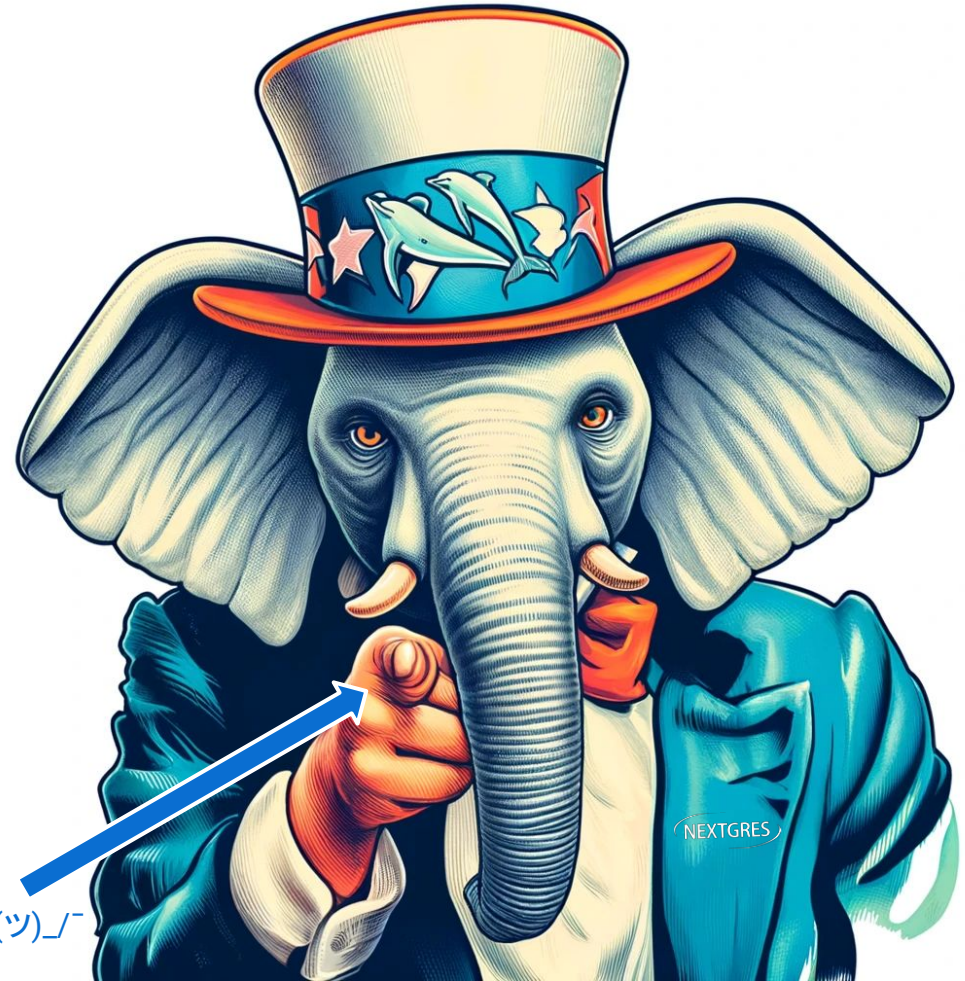
NEXTGRES

Our Timeline

NEXTGRES

We need you (or your team's) help!

- We want to help companies move from MySQL to Postgres quickly, easily, and cost-effectively.
- We have an aggressive timeline, testing with major MySQL open source packages, but we need more real-world use-cases.
- ~75% of companies running Postgres also use MySQL, which means you (more likely than not) have some systems that could move.
- If you'd like to work with us on the Alpha/Beta, reach out to **info@nextgres.com**



Pretty sweet generation, except those AI hands ¯_(ツ)_/¯

...and they lived
happily ever after.

THE END

DEMO

WordPress
mysqldump from MySQL to NEXTGRES
ProxySQL Server Switchover

NEXTGRES

jonah@nextgres.com
mason@nextgres.com