



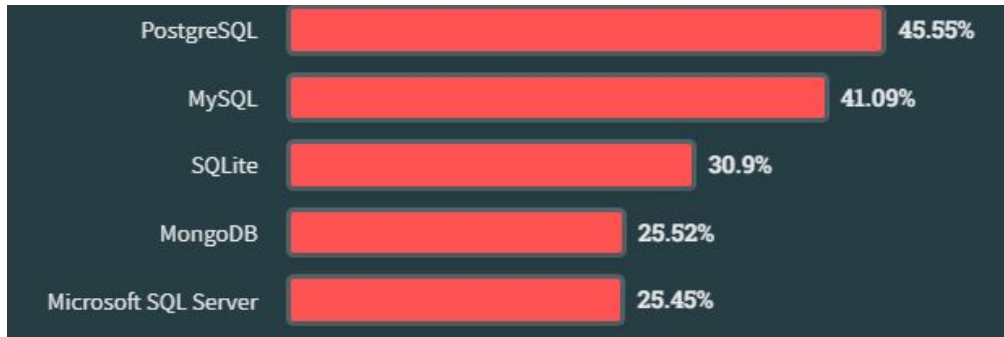
From MongoDB to Postgres: building an Open Standard for Document Databases

Peter Farkas, CEO, Percona
SFSCON 2025, Bolzano

Agenda

- Why do we care? Why move MongoDB to Postgres?
- MongoDB's licensing
- History of SQL as an Open Standard
- MongoDB Query Language as an Open Standard?
- Architecture of FerretDB
- Conclusion

One outlier on this graph



Source: StackExchange Developer Survey, 2024 (excerpt)

“Which database environments have you done extensive development work in over the past year, and which do you want to work in over the next year?”

“Which database environments have you done extensive development work in over the past year, and which do you want to work in over the next year?” 76k responses

SSPL

MongoDB – since 2018, released under the Server Side Public Licence (SSPL).

If MongoDB is used as part of a Cloud Service...

... source code of everything you use to provide that service needs to be licensed under SSPL.

Or you pay a licence fee.

MongoDB's approval (and license) is needed. This **kills innovation and creates vendor lock-in.**

More info:

www.ssplisbad.com

Also: Peter Zaitsev and Matt Yonkovit's articles on Percona Blog

Why not just use Postgres and JSON?

MongoDB has a large ecosystem. Many JS frameworks, stacks (MERN, MEAN, MEVN...) and other tools include or depend on MongoDB compatibility to work.

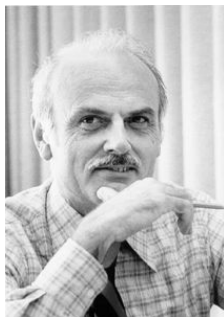






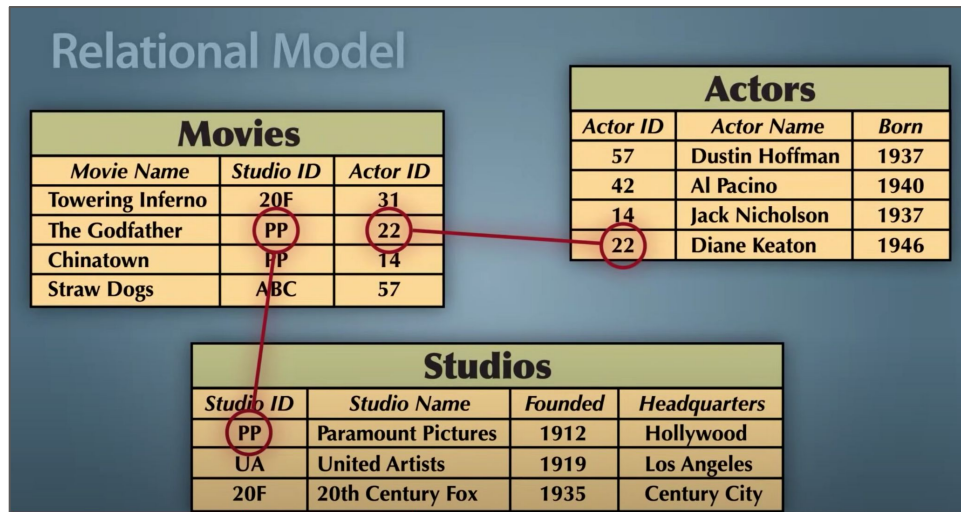

The short story of SQL

The short history of SQL

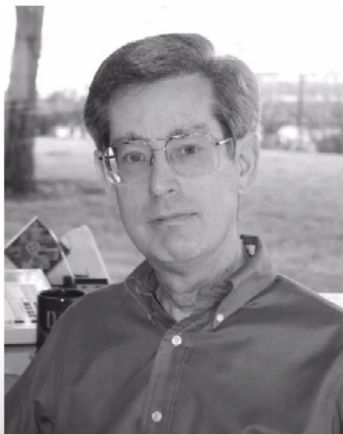


1.2.3. *Access Path Dependence*. Many of the existing formatted data systems provide users with tree-structured files or slightly more general network models of the data. Application programs developed to work with these systems tend to be logically impaired if the trees or networks are changed in structure. A simple example follows.

Muthukkaruppan, K. (2023) "Evolution of transactional databases," *Postgres Conf. SV. Postgres Conf. SV*, San Jose: Hilton, 20 April 2023.



The short history of SQL



Don Chamberlin



Ray Boyce († 1974)



At IBM, in the early 70s, Don Chamberlin and Ray Boyce laid down the foundations of the SQL query language for relational databases, which even non-developers could use.

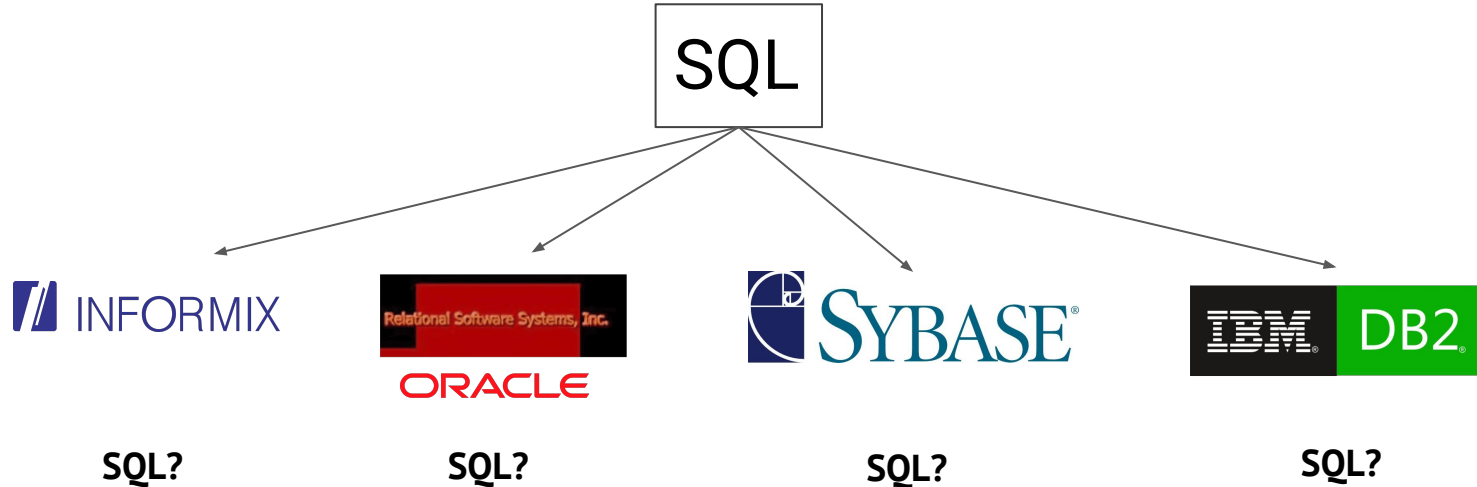
The short history of SQL - late 70s

- **IBM releases multiple relational database products speaking SQL**
- **Vendor lock-in, no other commercially available SQL database**
- **Like SQL? Use IBM.**



The evolution of SQL as an Open Standard – 80s

SQL is great, let's implement it in dozens of different ways!



Many different dialects, no conformity between products.

The evolution of SQL as an Open Standard – 86–87

- SQL Becomes an ANSI and later ISO Open Standard called SQL86.
- Anyone can implement them
- Features can be added on top (standard extension)

All vendors were proprietary, this still meant vendor lock-in.



SQL, an Open Standard. Here comes Open Source!

Mid 90s, early 2000s: Open Source projects started adopting SQL, partial implementations of the standard.

SQL is available to be used by anyone.



PostgreSQL



MySQL



SQLite

Since then: hundreds of derivatives

History repeats itself...sort of

Is this just the thing of the past?

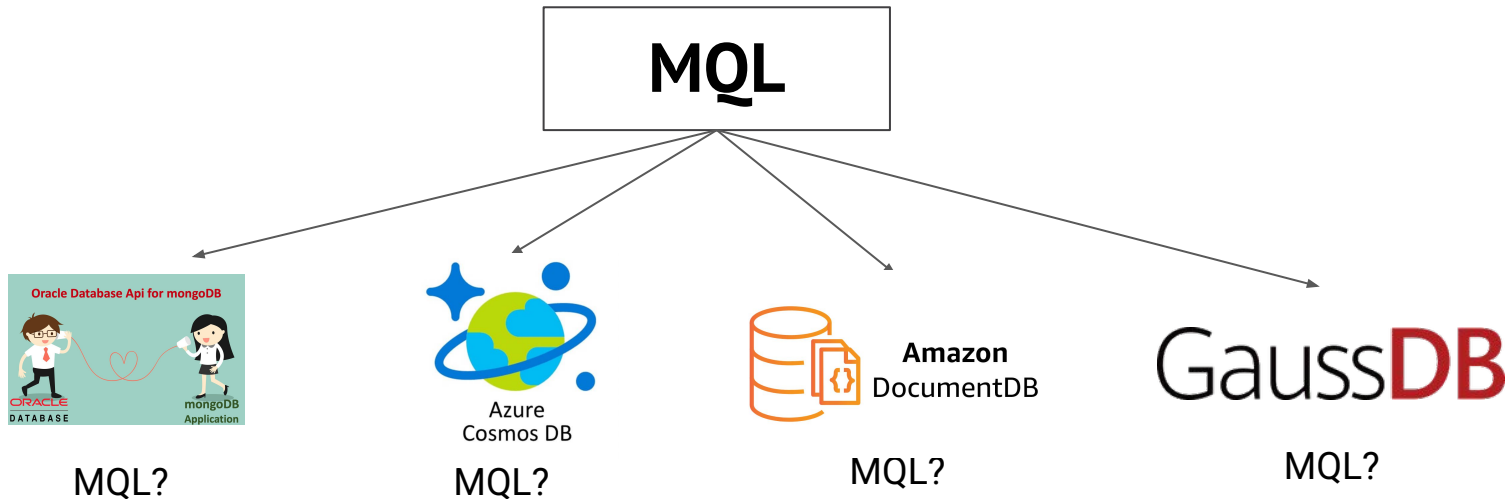
No. Look at Document Databases!

- MongoDB, a then open source database, develops MQL – the MongoDB Query Language.
- “Don’t need to be a DBA to run a database”
- Achieves market dominance in certain segments
- Goes proprietary in 2018
- Attempts to redefine the meaning of open source – SSPL license



A familiar situation

MQL is great, let's implement it in dozens of different ways!



All proprietary. Products look similar, but incompatible with each other. Once you choose one, you may be stuck with that.

The issue with most MongoDB Alternatives

All great products!

Similar query language, but

- vastly different feature set
- different degree of compatibility
- no chance of migration between them
- all proprietary
- Most tied to cloud vendors

For a “MongoDB alternative” – MongoDB sets the pace.

We are in need of a new Open Standard

MQL, an Open Standard?

- A standardized, core feature set based on MongoDB
- A JSON query language
- Can be extended at the expense of portability

Overwhelming interest from vendors and developers in the industry.

It will:

- Ensure portability between products
- Can be extended, just like SQL
- Stimulates innovation, increase competition
- Be very good for users

Steps taken + current progress

FerretDB

- A MongoDB compatibility layer
- Written in Go
- Uses PostgreSQL as backend
- Usable on-prem or in the cloud
- Released under Apache 2.0
- Utilizes Microsoft DocumentDB, a Postgres extension

MongoDB compatibility means that existing elements of the MongoDB Ecosystem (tools, frameworks and applications) are possible to use with FerretDB

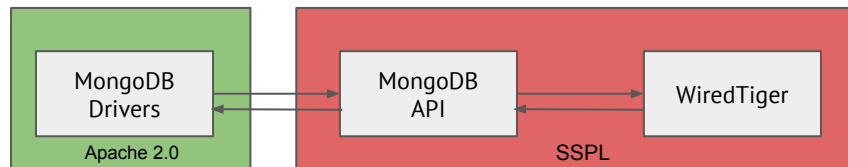
FerretDB vs. MongoDB architecture and licensing

MongoDB Drivers

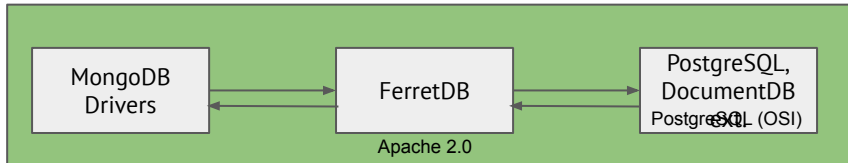
- Main reason behind high adoption of MongoDB
- Provides unmatched Developer Experience
- Free to use under Apache 2.0

MongoDB Backend

- Licensed under SSPL
- Proprietary vendors (Amazon, IBM, Oracle, etc.) replaced it in their own implementations of a compatible product



FerretDB and DocumentDB replaces the MongoDB Backend with PostgreSQL



QUESTIONS

www.ferretdb.com

www.documentdb.io

Grazie!