

# Benchmarks, myths, experience and engineering



The “secret” science behind  
choosing the correct DB engine

# Benchmarks, myths, experience and engineering



The “~~secret~~” science behind choosing the ~~correct~~ DB engines

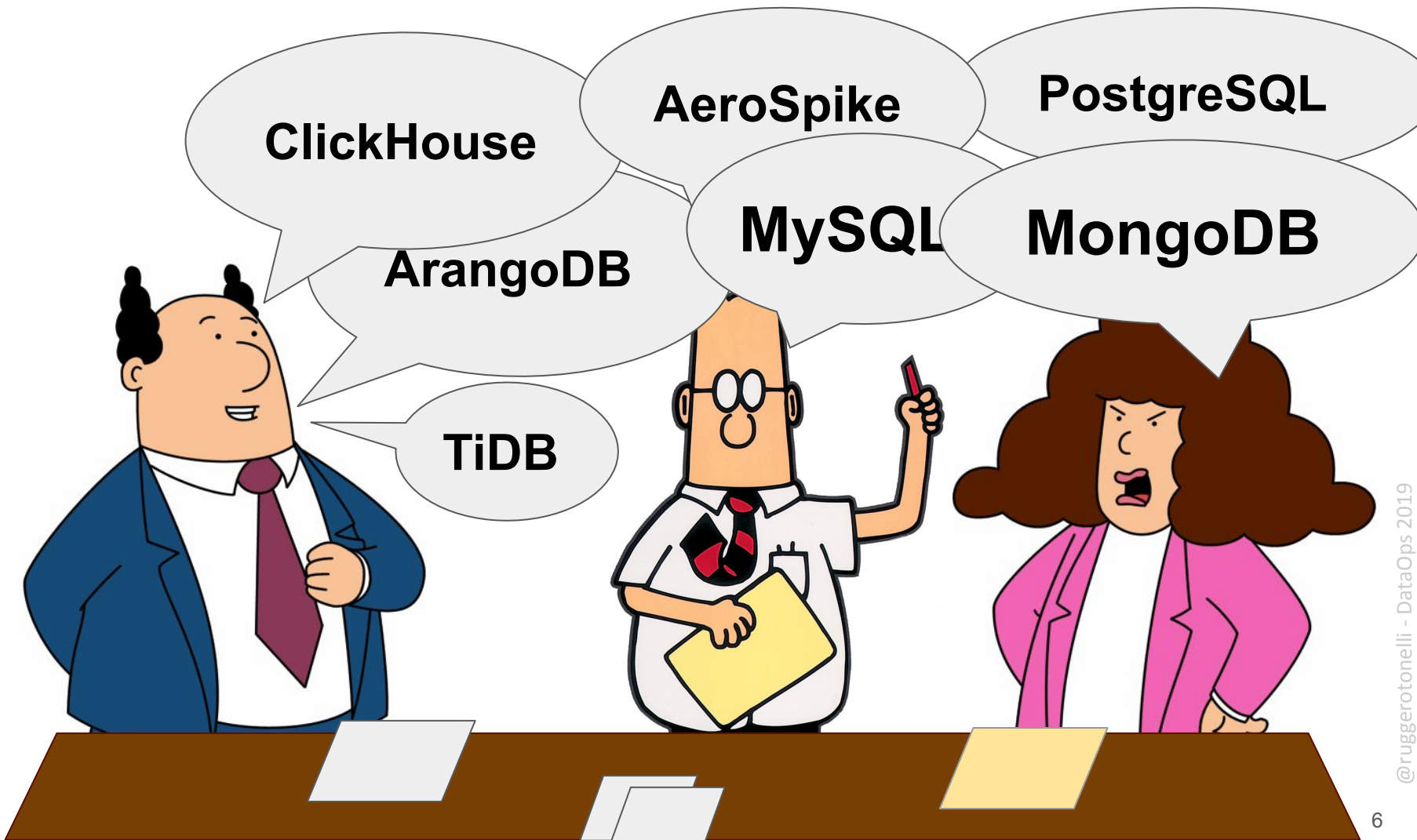
# Benchmarks, experience and engineering



Being pragmatic  
when choosing DB engines

**Intro**

**a Software Engineer, a DBA  
and a SRE enter a bar  
discussing about...**



**ClickHouse**

**AeroSpike**

**PostgreSQL**

**ArangoDB**

**MySQL**

**MongoDB**

**TiDB**

**ClickHouse is faster than MySQL in OLAP**

**Aurora is better!**

**Our workload  
MySQL handles everything**

**PostgreSQL**

**MongoDB can do SQL!**

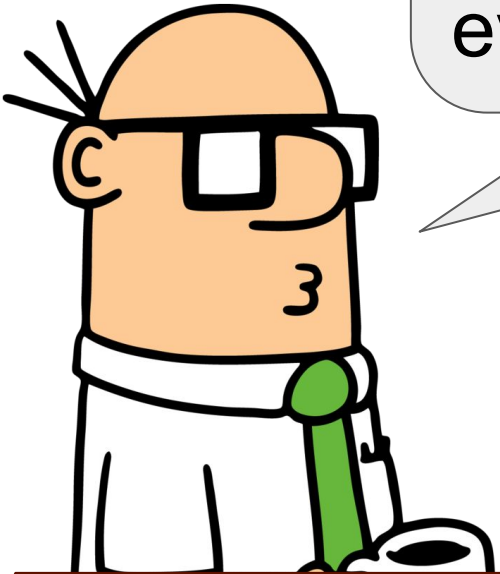


**RUN A BENCHMARK !!!**



**please?..**





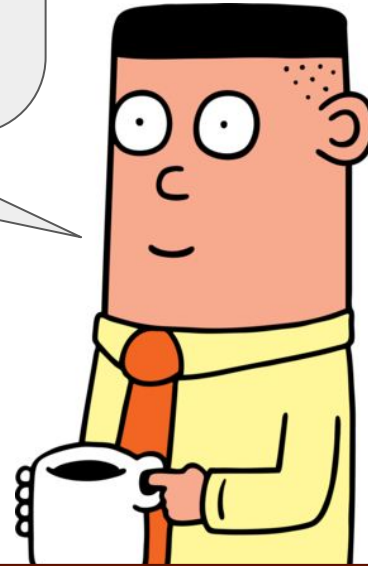
Shouldn't we get all the **requirements, constraints** and **restrictions** before even start?

Do we know what's the **expected load** and **performance**?



So... are you telling me that choosing a DB is not about faith, dogmas or bullying the others?

**...we should run Vitesse!**



**Humans are so boring...  
the correct answer is  
ORA \* \* E, always!**



# Part One

Of requirements, constraints and restrictions

# Of requirements, constraints and restrictions

# Budget

# Time to Market, MVP or PoC

# Internal know-how

# Coding languages (support maturity)

# Of requirements, constraints and restrictions

# Paid support

# Adoption level (maturity)

# Software licensing

# Workload types

# Of requirements, constraints and restrictions

# Resiliency

# Scalability

# Performance

# Encryption at rest and on-the-fly

# Of requirements, constraints and restrictions

# Vendor lock-in

# Mind the Cloud

# SW/HW “limitations”

# Eventual migration path



# Of requirements, constraints and restrictions

# Ease of management

# Documentation

# Known users and specific cases

# Maturity ....did we say maturity enough?

# Part Two

Benchmarking definition, criteria and tools

# Benchmarking definition, criteria and tools

# Essential requirements for “experiments”

# Product’s best practices

# “Coding” your own (benchmark)

# Open Source benchmarks

# Benchmarking definition, criteria and tools

# SysBench

# YCSB

# Your own workload

# Your peers connections

# Benchmarking definition, criteria and tools

# Benchmarks you find in the Internet

# Researching “matching” issues

# Drawing your own conclusions

# Document processes and trade-offs

# Part Three

Data Engineering and Experience

# Data Engineering and Experience

# Know your enemies or RTFM

# Capacity planning & forecasting

# Plan for the worst

# Monitoring and Observability

# Data Engineering and Experience

# Polyglot Persistence

# Multi-model DBs

# Data integration

# Multiverse databases!



**Recap**

# Recap

Requirements and restriction are not that hard.

Benchmarking is difficult, you better have an objective and consistent results.

Reaching consensus on choosing a DB Engine is better when you have numbers.

# Thank You

@ruggeronelli

# Q&A

Images	
Page 1-3	123RF
Page 6 - 11	Dilbert by Scott Adams