

Open Source Databases

For Python Devs - pyDay 2024 Barcelona



Dave Pitts - Database Engineer - Adyen (Madrid)



engineered
for ambition

Open Source Databases

For Python Devs - pyDay 2024 Barcelona

Dave Pitts - Database Engineer - Adyen (Madrid)



engineered
for ambition



Open Source Databases

For Python Devs - pyDay 2024 Barcelona

Dave Pitts - Database Engineer - Adyen (Madrid)



engineered
for ambition



Open Source Databases

For Python Devs - pyDay 2024 Barcelona

Dave Pitts - Database Engineer - Adyen (Madrid)



engineered
for ambition

Redis



MySQL™



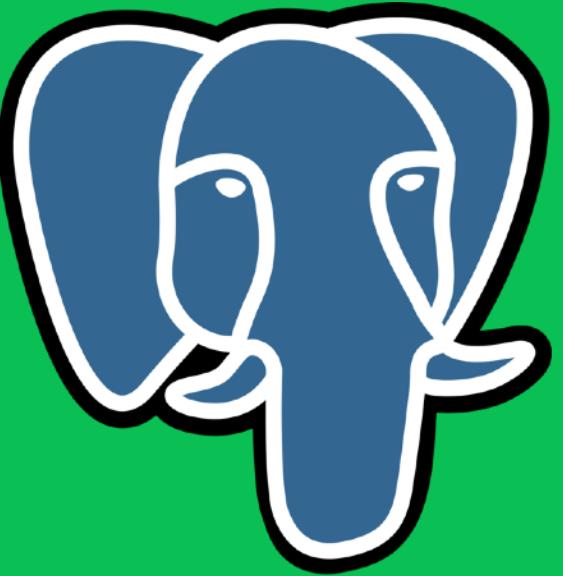
Open Source Databases

For Python Devs - pyDay 2024 Barcelona

Dave Pitts - Database Engineer - Adyen (Madrid)



engineered
for ambition



Open Source Databases

For Python Devs - pyDay 2024 Barcelona

Dave Pitts - Database Engineer - Adyen (Madrid)



Open Source Databases

For Python Devs - pyDay 2024 Barcelona

Dave Pitts - Database Engineer - Adyen (Madrid)



engineered
for ambition

Open Source Databases

For Python Devs - pyDay 2024 Barcelona

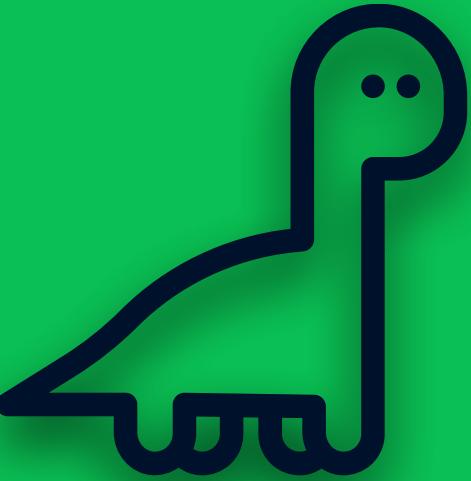
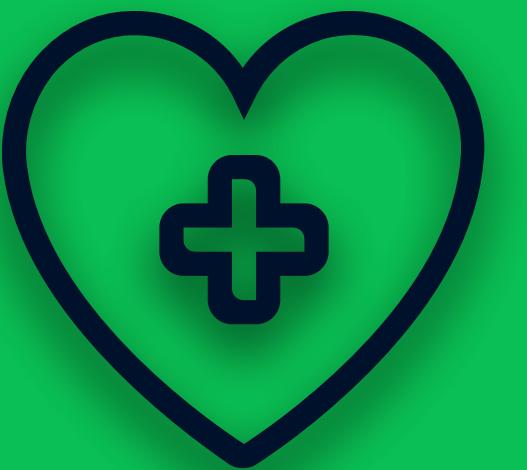
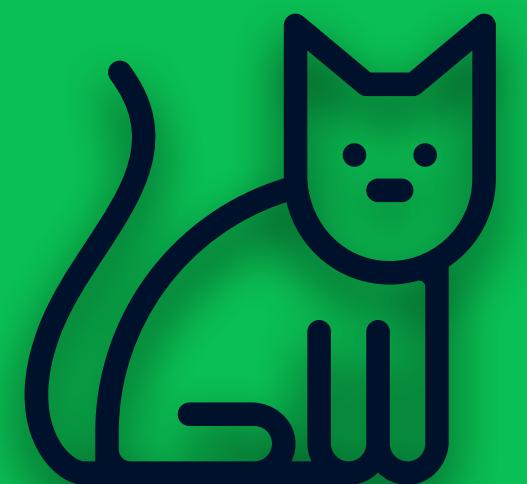
Dave Pitts - Database Engineer - Adyen (Madrid)



Open Source Databases

For Python Devs - pyDay 2024 Barcelona

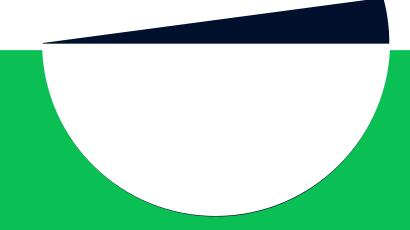
Dave Pitts - Database Engineer - Adyen (Madrid)



adyen

engineered
for ambition

The Adyen Formula



We include
different people
to sharpen our ideas

Neurodiversity ERG (Madrid co-Lead)



What, Why, When for DBs?

adyen

engineered
for ambition

Hey **Dave** this spreadsheet has a problem...



London Whitehall, Research Assistant (1989)

Hey **Dave** this spreadsheet has a problem...

it doesn't add up!?



London Whitehall, Research Assistant (1989)

148508
982459
165395
719690
177238
901601
208382
521095
601030
167732
269259
990883
239158
860832
268194
882809
721983
825738
843093
650053
980110
366120
42208
198027
51309
263461
545427
98790
271648
80043

Hey **Dave** this spreadsheet has a problem...
it doesn't add up!?



London Whitehall, Research Assistant (1989)

Hey **Dave** this spreadsheet has a problem...

it doesn't add up!?



London Whitehall, Research Assistant (1989)

148508	148508
982459	982459
165395	165395
719690	719690
177238	177238
901601	901601
208382	208382
521095	521095
601030	601030
167732	167732
269259	269259
990883	990883
239158	239158
860832	860832
268194	268194
882809	882809
721983	721983
825738	825738
843093	843093
650053	650053
980110	980110
366120	366120
42208	42208
198027	198027
51309	51309
263461	263461
545427	545427
98790	98790
271648	271648
80043	80043

Hey **Dave** this spreadsheet has a problem...

it doesn't add up!?



London Whitehall, Research Assistant (1989)

148508	148508
982459	982459
165395	165395
719690	719690
177238	177238
901601	901601
208382	208382
521095	521095
601030	601030
167732	167732
269259	269259
990883	990883
239158	239158
860832	860832
268194	268194
882809	882809
721983	721983
825738	825738
843093	843093
650053	650053
980110	980110
366120	366120
42208	42208
198027	198027
51309	51309
263461	263461
545427	545427
98790	98790
271648	271648
80043	80043

Hey **Dave** this spreadsheet has a problem...

it doesn't add up!?



London Whitehall, Research Assistant (1989)

148508	148508
982459	982459
165395	165395
719690	719690
177238	177238
901601	901601
208382	208382
521095	521095
601030	601030
167732	167732
269259	269259
990883	990883
239158	239158
860832	860832
268194	268194
882809	882809
721983	721983
825738	825738
843093	843093
650053	650053
980110	980110
366120	366120
42208	42208
198027	198027
51309	51309
263461	263461
545427	545427
98790	98790
271648	271648
80043	80043

What the PIVOT table?

Trying to join to spreadsheet and/or aggregating data



Out in Tech, Oct 2023

What the PIVOT table?

Trying to join to spreadsheet and/or aggregating data

Tables, SQL and relational model (1970)



Out in Tech, last month

Young People Centre (Brighton, UK)

- Services (Internet, Tea, Games, Counseling...)



Young People Centre (Brighton, UK)

- Services (Internet, Tea, Games, Counseling...)
- Users (Age, Gender, PostCode, Dyslexia, ADHD ...)



Young People Centre (Brighton, UK)

- Services (Internet, Tea, Games, Counseling...)
- Users (Age, Gender, PostCode, Dyslexia, ADHD ...)
- Visits (Date, UserID, ServiceIDs)





<https://en.wikipedia.org/wiki/SQLite#/media/File:SQLite370.svg>

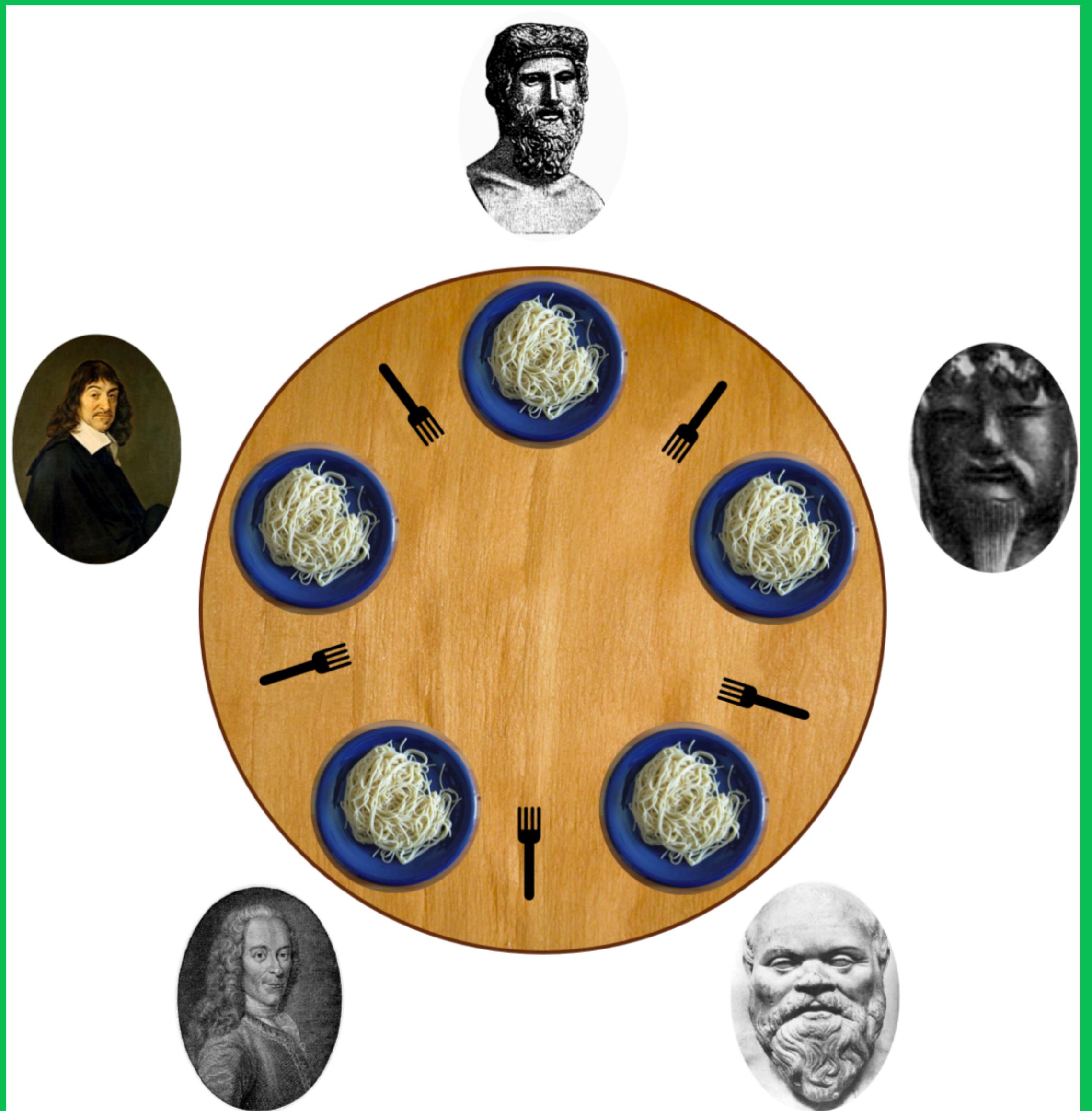
Let's Talk About Concurrency



[https://en.wikipedia.org/wiki/
Dining_philosophers_problem#/media/
File:An_illustration_of_the_dining_philosophers_problem.png](https://en.wikipedia.org/wiki/Dining_philosophers_problem#/media/File:An_illustration_of_the_dining_philosophers_problem.png)

Let's Talk About Concurrency

Philosophers clockwise from top -
Plato, Konfuzius, Socrates, Voltaire and Descartes.



[https://en.wikipedia.org/wiki/
Dining_philosophers_problem#/media/
File:An_illustration_of_the_dining_philosophers_problem.png](https://en.wikipedia.org/wiki/Dining_philosophers_problem#/media/File:An_illustration_of_the_dining_philosophers_problem.png)

Let's Talk About Concurrency

Philosophers clockwise from top -
Plato, Konfuzius, Socrates, Voltaire and Descartes.

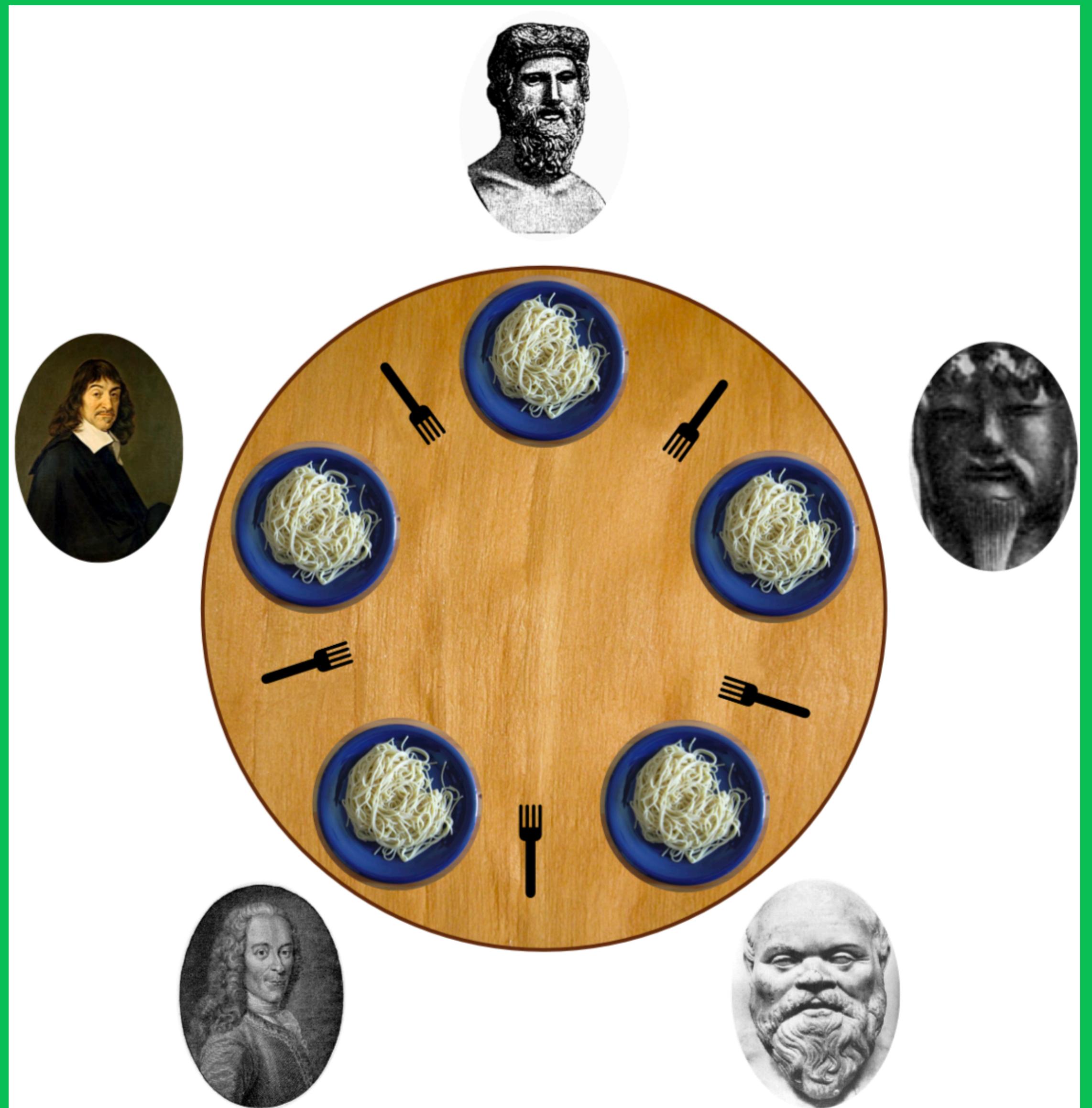
*The dish served is a kind of spaghetti which has to be eaten with **two** forks.*

[https://en.wikipedia.org/wiki/
Dining_philosophers_problem#/media/
File:An_illustration_of_the_dining_philosophers_problem.png](https://en.wikipedia.org/wiki/Dining_philosophers_problem#/media/File:An_illustration_of_the_dining_philosophers_problem.png)



Let's Talk About Concurrency

Philosophers clockwise from top -
Plato, Konfuzius, Socrates, Voltaire and Descartes.



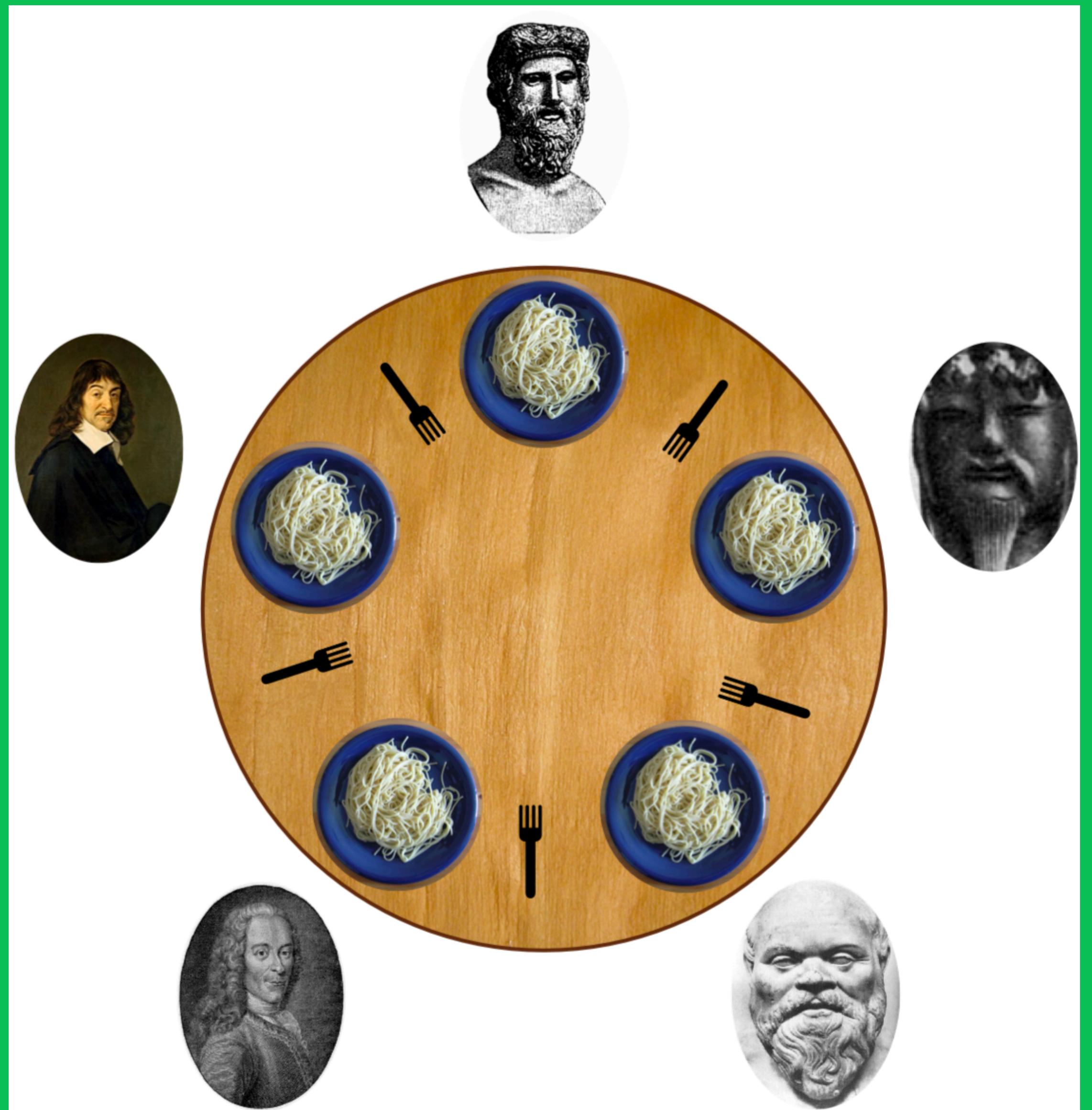
*The dish served is a kind of spaghetti which has to be eaten with **two** forks.*

*Each philosopher can only **alternately** think and eat.*

https://en.wikipedia.org/wiki/Dining_philosophers_problem#/media/File:An_illustration_of_the_dining_philosophers_problem.png

Let's Talk About Concurrency

Philosophers clockwise from top -
Plato, Konfuzius, Socrates, Voltaire and Descartes.



*The dish served is a kind of spaghetti which has to be eaten with **two** forks.*

*Each philosopher can only **alternately** think and eat.
Moreover, a philosopher can only eat his spaghetti when he has **both** a left and right fork....*

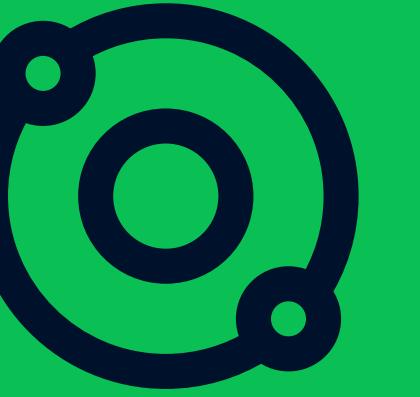
[https://en.wikipedia.org/wiki/
Dining_philosophers_problem#/media/
File:An_illustration_of_the_dining_philosophers_problem.png](https://en.wikipedia.org/wiki/Dining_philosophers_problem#/media/File:An_illustration_of_the_dining_philosophers_problem.png)



https://en.wikipedia.org/wiki/MySQL#/media/File:MySQL_logo.svg

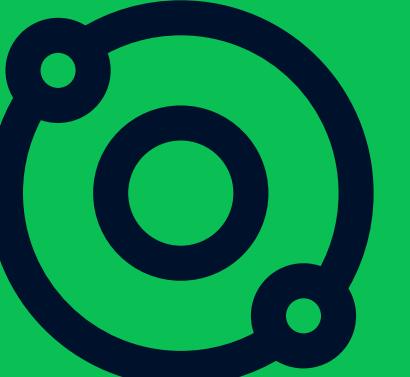






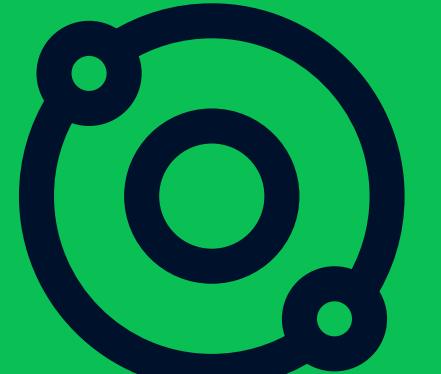
Un-split brain (aka Move Back in Time) MySQL

- DBA Nightmare scenario
- #kudos to Shlomi Noach & Github
- Very painful & funny (YouTube)



Un-split brain (aka Move Back in Time) MySQL

- DBA Nightmare scenario
- #kudos to Shlomi Noach & Github
- Very painful & funny (YouTube)



Source: Catdyen

Demo One: MySQL Secondary index gotcha

so we have direct access to the "index orientated table" and lower "read_cost": "20.35"

```
"read_cost": "20.35",
"eval_cost": "20.00",
"prefix_cost": "40.35",
"data_read_per_join": "6K"
},
```

vs the almost identical unique index `id2` we see x3.5 times higher read cost

```
"cost_info": {
    "read_cost": "70.26",
    "eval_cost": "20.00",
    "prefix_cost": "90.26",
    "data_read_per_join": "6K"
},
```

Secondary indexes in InnoDB are also B-trees, but they do not store the actual row data. Instead, they store the indexed column(s) and a reference to the corresponding primary key value.

<https://github.com/dgapitts/learning-mysql/blob/main/docs/020-mysql-secondary-index.md>

What's the Difference Between MySQL and PostgreSQL?



[https://aws.amazon.com/compare/
the-difference-between-mysql-vs-postgresql/](https://aws.amazon.com/compare/the-difference-between-mysql-vs-postgresql/)

What's the Difference Between MySQL and PostgreSQL?

MySQL has improved performance for high-frequency read operations.



[https://aws.amazon.com/compare/
the-difference-between-mysql-vs-postgresql/](https://aws.amazon.com/compare/the-difference-between-mysql-vs-postgresql/)

What's the Difference Between MySQL and PostgreSQL?



MySQL has improved performance for high-frequency read operations.

MySQL databases do not offer Multiversion concurrency control (MVCC), but PostgreSQL supports this feature.



[https://aws.amazon.com/compare/
the-difference-between-mysql-vs-postgresql/](https://aws.amazon.com/compare/the-difference-between-mysql-vs-postgresql/)

What's the Difference Between MySQL and PostgreSQL?



MySQL has improved performance for high-frequency read operations.

MySQL databases do not offer Multiversion concurrency control (MVCC), but PostgreSQL supports this feature.



[https://aws.amazon.com/compare/
the-difference-between-mysql-vs-postgresql/](https://aws.amazon.com/compare/the-difference-between-mysql-vs-postgresql/)

Why does MVCC matter?



Nice 👍

Mum has posted a new cat..

Why does MVCC matter?



- Hey ... Now it gone ...

Why does MVCC matter?



Nice 👍

The cat photo is back ❤️

What's the Difference Between MySQL and PostgreSQL?



MySQL has improved performance for high-frequency read operations.

MySQL databases do not offer Multiversion concurrency control (MVCC), but PostgreSQL supports this feature.

PostgreSQL is better suited for enterprise-level applications with frequent write operations and complex queries.



[https://aws.amazon.com/compare/
the-difference-between-mysql-vs-postgresql/](https://aws.amazon.com/compare/the-difference-between-mysql-vs-postgresql/)

Talking Postgres is my favorite pg podcast

[https://aws.amazon.com/compare/
the-difference-between-mysql-vs-postgresql/](https://aws.amazon.com/compare/the-difference-between-mysql-vs-postgresql/)

Demo Two: OLAP Cube and PG Workmem Parameter

Citus Simple OLAP Cube

OLAP Table with 100 different columns (all integer)

```
# CREATE TABLE perf_row(  
    c00 int8, c01 int8, c02 int8, c03 int8, c04 int8, c05 int8, c06 int8, c07 int8, c08 int8, c09 int8,  
    c10 int8, c11 int8, c12 int8, c13 int8, c14 int8, c15 int8, c16 int8, c17 int8, c18 int8, c19 int8,  
    c20 int8, c21 int8, c22 int8, c23 int8, c24 int8, c25 int8, c26 int8, c27 int8, c28 int8, c29 int8,  
    c30 int8, c31 int8, c32 int8, c33 int8, c34 int8, c35 int8, c36 int8, c37 int8, c38 int8, c39 int8,  
    c40 int8, c41 int8, c42 int8, c43 int8, c44 int8, c45 int8, c46 int8, c47 int8, c48 int8, c49 int8,  
    c50 int8, c51 int8, c52 int8, c53 int8, c54 int8, c55 int8, c56 int8, c57 int8, c58 int8, c59 int8,  
    c60 int8, c61 int8, c62 int8, c63 int8, c64 int8, c65 int8, c66 int8, c67 int8, c68 int8, c69 int8,  
    c70 int8, c71 int8, c72 int8, c73 int8, c74 int8, c75 int8, c76 int8, c77 int8, c78 int8, c79 int8,  
    c80 int8, c81 int8, c82 int8, c83 int8, c84 int8, c85 int8, c86 int8, c87 int8, c88 int8, c89 int8,  
    c90 int8, c91 int8, c92 int8, c93 int8, c94 int8, c95 int8, c96 int8, c97 int8, c98 int8, c99 int8  
);
```

Citus Simple OLAP Cube

Distinct values C00:500, C70:35500, C99:50000

```
# INSERT INTO perf_row
SELECT
    g % 00500, g % 01000, g % 01500, g % 02000, g % 02500, g % 03000, g % 03500, g % 04000, g % 04500, g % 05000,
    g % 05500, g % 06000, g % 06500, g % 07000, g % 07500, g % 08000, g % 08500, g % 09000, g % 09500, g % 10000,
    g % 10500, g % 11000, g % 11500, g % 12000, g % 12500, g % 13000, g % 13500, g % 14000, g % 14500, g % 15000,
    g % 15500, g % 16000, g % 16500, g % 17000, g % 17500, g % 18000, g % 18500, g % 19000, g % 19500, g % 20000,
    g % 20500, g % 21000, g % 21500, g % 22000, g % 22500, g % 23000, g % 23500, g % 24000, g % 24500, g % 25000,
    g % 25500, g % 26000, g % 26500, g % 27000, g % 27500, g % 28000, g % 28500, g % 29000, g % 29500, g % 30000,
    g % 30500, g % 31000, g % 31500, g % 32000, g % 32500, g % 33000, g % 33500, g % 34000, g % 34500, g % 35000,
    g % 35500, g % 36000, g % 36500, g % 37000, g % 37500, g % 38000, g % 38500, g % 39000, g % 39500, g % 40000,
    g % 40500, g % 41000, g % 41500, g % 42000, g % 42500, g % 43000, g % 43500, g % 44000, g % 44500, g % 45000,
    g % 45500, g % 46000, g % 46500, g % 47000, g % 47500, g % 48000, g % 48500, g % 49000, g % 49500, g % 50000
FROM generate_series(1,500000) g;
```

Before pg15 - hash_mem_multiplier=1

Aggr/Group 50K values: very high Disc Usage and high IOPs

```
# EXPLAIN (ANALYZE, BUFFERS) SELECT c99, SUM(c29), AVG(c71) FROM perf_row GROUP BY c99;  
          QUERY PLAN
```

```
-----  
HashAggregate (cost=97743.84..104357.10 rows=50256 width=72) (actual time=1697.361..2050.372 rows=50000 loops=1)  
  Group Key: c99  
  Planned Partitions: 4 Batches: 5 Memory Usage: 4145kB Disk Usage: 23496kB  
  Buffers: shared hit=15688 read=39868, temp read=2641 written=4909  
    -> Seq Scan on perf_row (cost=0.00..60556.04 rows=500004 width=24) (actual time=2.007..833.247 rows=500000 loops=1)  
      Buffers: shared hit=15688 read=39868
```

pg15+ hash_mem_multiplier=2

Aggr/Group 50K values: high Disc Usage and high IOPs

```
# EXPLAIN (ANALYZE, BUFFERS) SELECT c99, SUM(c29), AVG(c71) FROM perf_row GROUP BY c99;  
QUERY PLAN
```

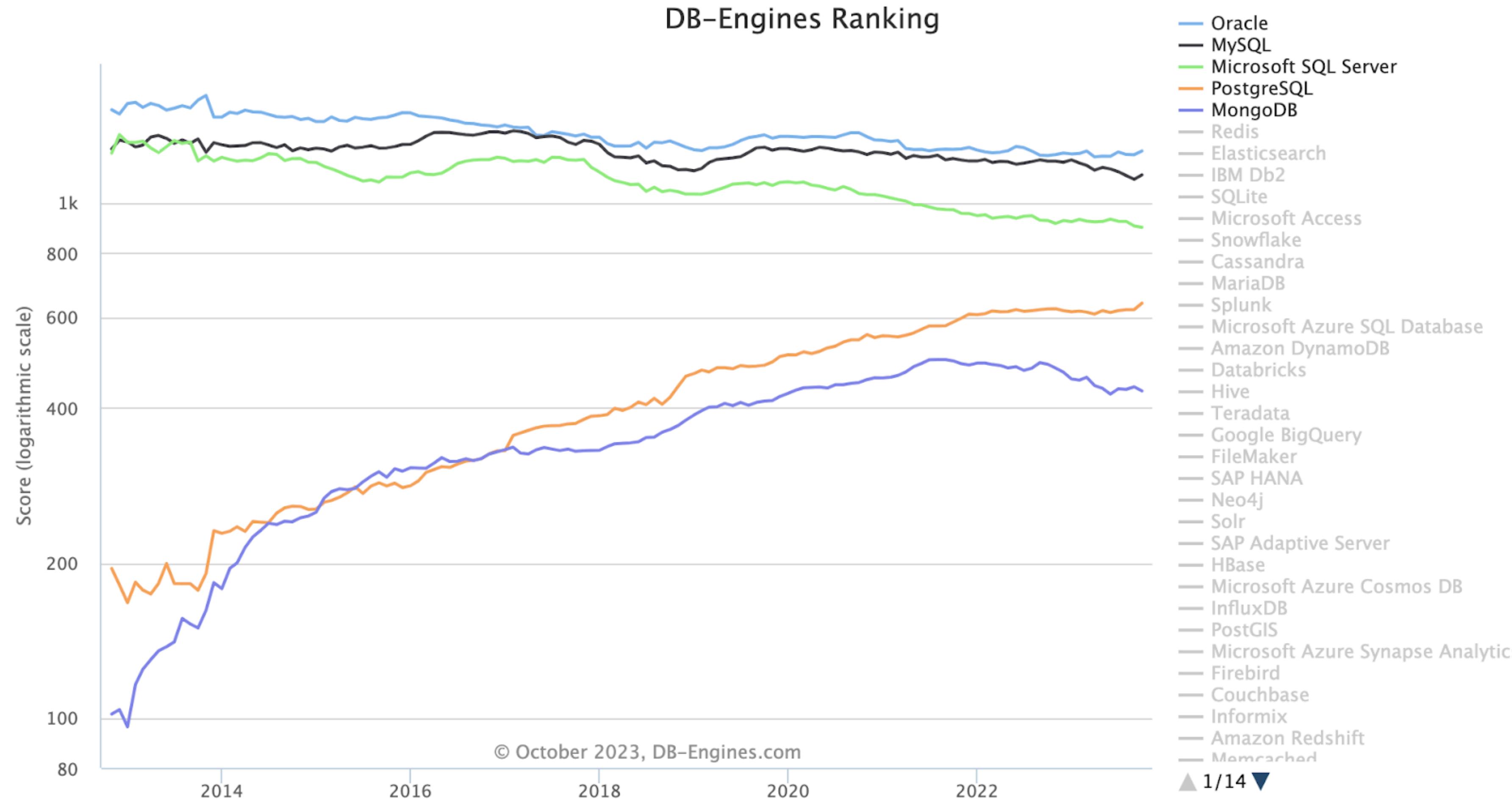
```
-----  
HashAggregate (cost=97743.84..104357.10 rows=50256 width=72) (actual time=1480.648..1689.421 rows=50000 loops=1)  
  Group Key: c99  
  Planned Partitions: 4 Batches: 5 Memory Usage: 8241kB Disk Usage: 14104kB  
  Buffers: shared hit=15688 read=39868, temp read=1525 written=2896  
    -> Seq Scan on perf_row (cost=0.00..60556.04 rows=500004 width=24) (actual time=0.700..825.849 rows=500000 loops=1)  
      Buffers: shared hit=15688 read=39868
```

Custom hash_mem_multiplier=4

Aggr/Group 50K values - with custom setting (pg13+)

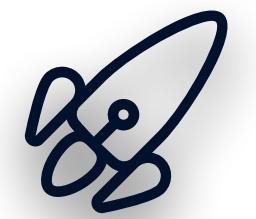
```
# set hash_mem_multiplier=4;
SET
# EXPLAIN (ANALYZE, BUFFERS) SELECT c99, SUM(c29), AVG(c71) FROM perf_row GROUP BY c99;
          QUERY PLAN
-----
HashAggregate (cost=64306.07..65059.91 rows=50256 width=72) (actual time=247.716..257.823 rows=50000 loops=1)
  Group Key: c99
  Batches: 1  Memory Usage: 12561kB
  Buffers: shared hit=15822 read=39734
    -> Seq Scan on perf_row (cost=0.00..60556.04 rows=500004 width=24) (actual time=0.226..88.872 rows=500000 loops=1)
        Buffers: shared hit=15822 read=39734
```

October

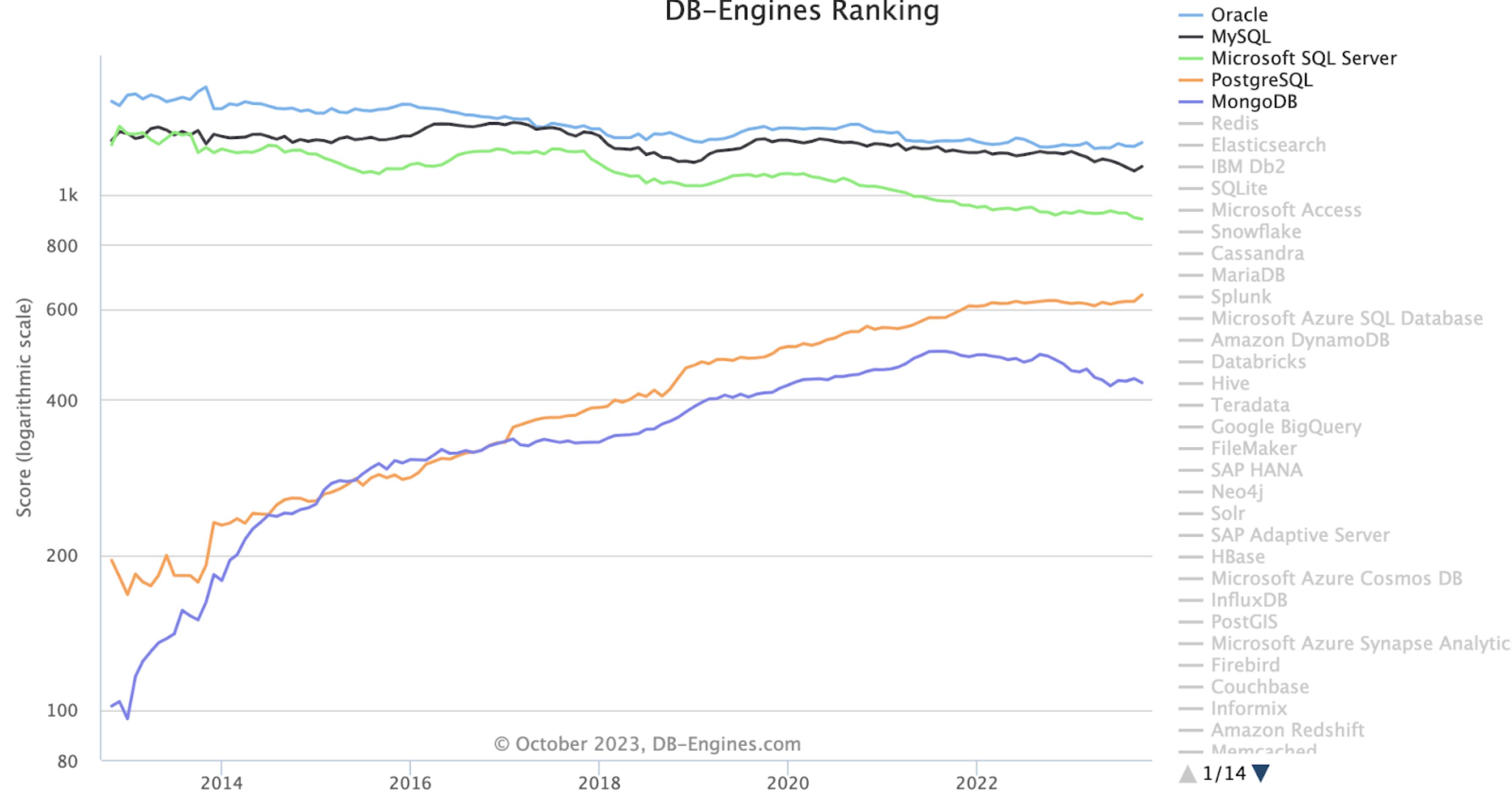


[https://db-engines.com/en/ranking_trend \(top five\)](https://db-engines.com/en/ranking_trend (top five))

October



DB-Engines Ranking



[https://db-engines.com/en/ranking_trend \(top five\)](https://db-engines.com/en/ranking_trend (top five))

Adyen ahead of the curve ...



3000+ employees & 115+ nationalities



27 offices around the world



**€767.5 BN billion in processed volume
in 2022**

adyen

engineered
for ambition

LANCASTER

VALVE

Aēsop.

Booking.com

BOSE®

foodora

LVMH

TIFFANY & Co.

FACEBOOK

ZARA

BLIZZARD
ENTERTAINMENT

Gap Inc.

Cartier

RITUALS...

Spotify

M

PRADA

citizen
M
hotels

HMS
HOST

DUNKIN'

adidas

Microsoft

SUBWAY

Hello
FRESH

freelancer

asics

ebay

Uber

zalando

L'Occitane
EN PROVENCE

LANCEL
PARIS 1876

wagamama

H&M

Alibaba Group
阿里巴巴集团

SINGAPORE
AIRLINES

JOE & THE JUICE

a alza.cz

patagonia®

MERLIN
ENTERTAINMENTS GROUP®

FANATEC®

Foot Locker.

HAKKASAN
GROUP

UNDER ARMOUR

de Bijenkorf

BONOBOS

tinder

HappySocks®

Grab

So, you hit 2147483647...

- *Welcome to the club. You'll hate it... (great article on LinkedIn)*
- An interesting MySQL edge case, I saw recently (not Adyen)
- 2147483647 8th Mersenne prime ($M_p = 2^p - 1$ p=2,3,5,7,13,17,19,31, ...)



Every database engine has its edge cases and limits

Tuning Postgres Internals

- “*This is going to blow your mind*”
- *Bruce Momjian on Multi-Version Concurrency Control in Postgres (MVCC)*
- *co-founder and core team member of the PostgreSQL Global Dev Group, and worked on PostgreSQL since 1996.*

Every database engine has it's edge cases and limits



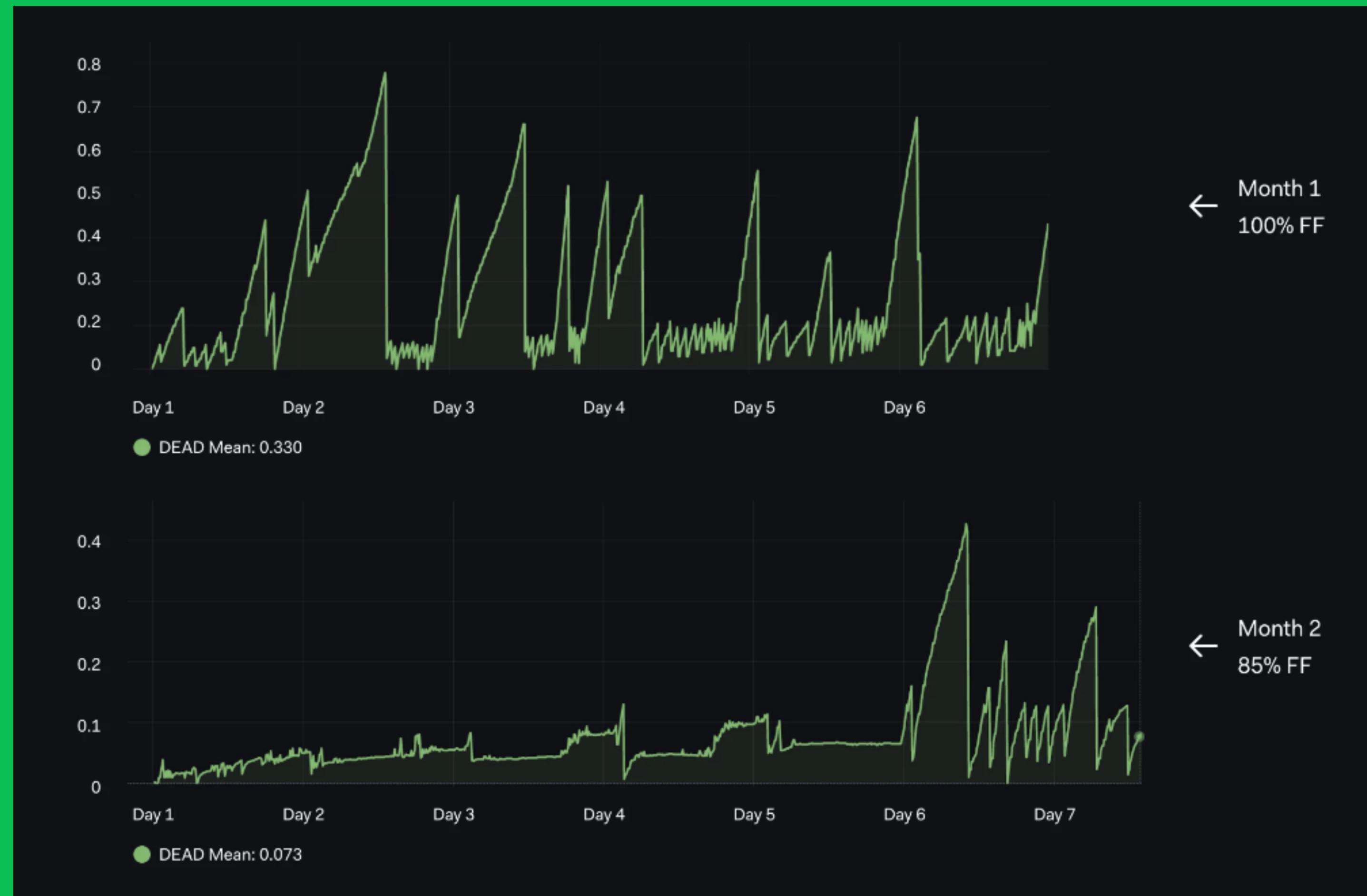
Tuning Postgres Internals and Prometheus Dashbaords

- Dave and Derk presenting Adyen's tuning at scale in Berlin (pgconf.eu 2022)



Tuning Postgres Internals and Prometheus Dashbaords

- <https://www.adyen.com/knowledge-hub/postgresql-hot-updates-part2>



Tuning Postgres Internals

- “*Kudos to the team at Adyen*” (*Bruce Momjian*)



How Open is OpenSource?

How Open is OpenSource?



How Open is OpenSource?



“Beware of false prophets, which come to you in sheep's clothing, but inwardly they are ravening wolves.”

King James, 1611

Open Source vs BSL



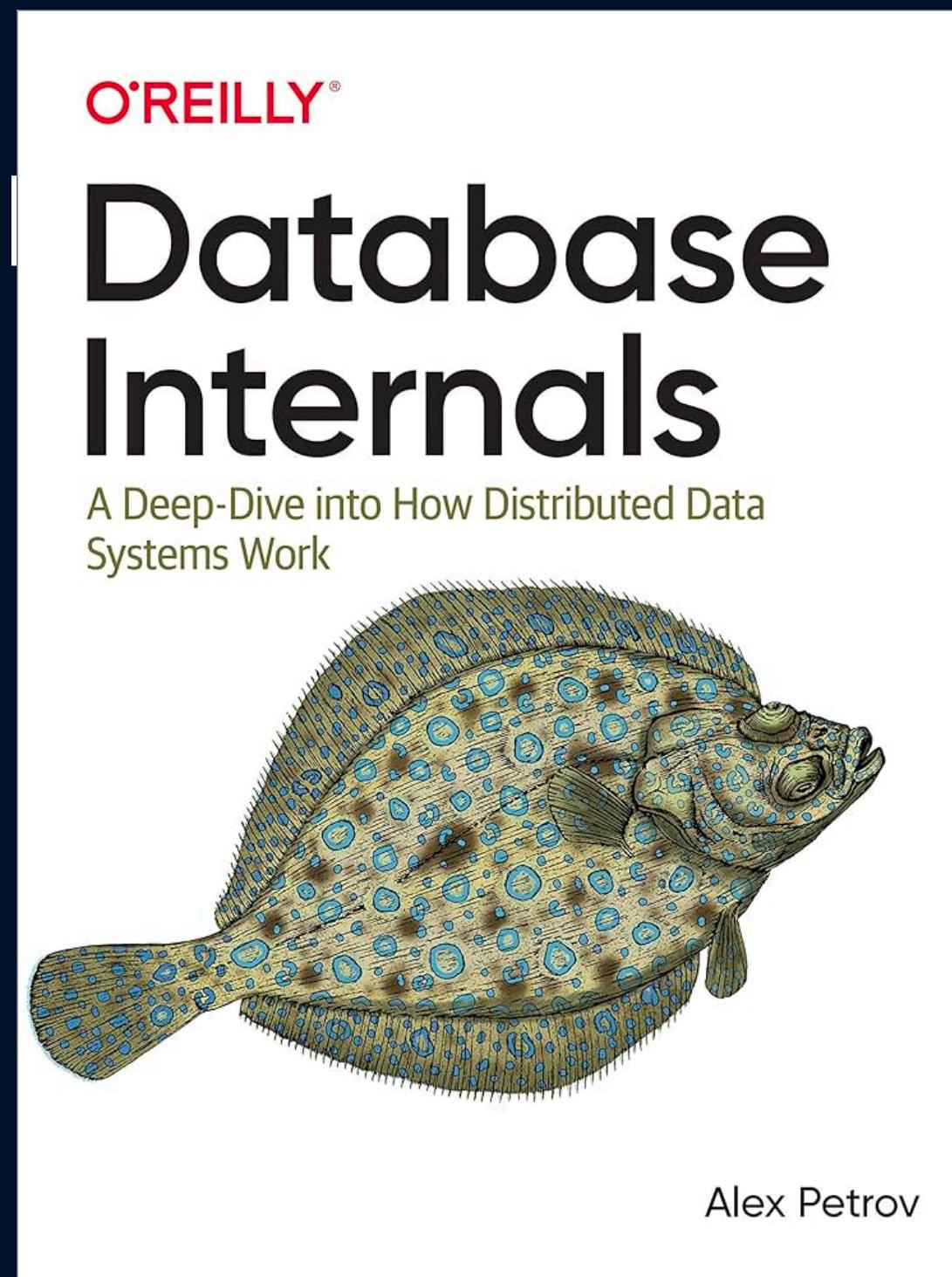
Definition: BSL is a licensing model where the software is initially restricted for commercial use but becomes fully Open Source after a certain period, often 2-4 years.

Examples: MariaDB and CockroachDB use BSL for some of their offerings.

Database Internals

Database Internals

"A very fine line between indexes and tables"



- SE Radio 417: Alex Petrov: Distributed

Implementations - old & new

Paged/Heap (older)



Log Structured (newer)

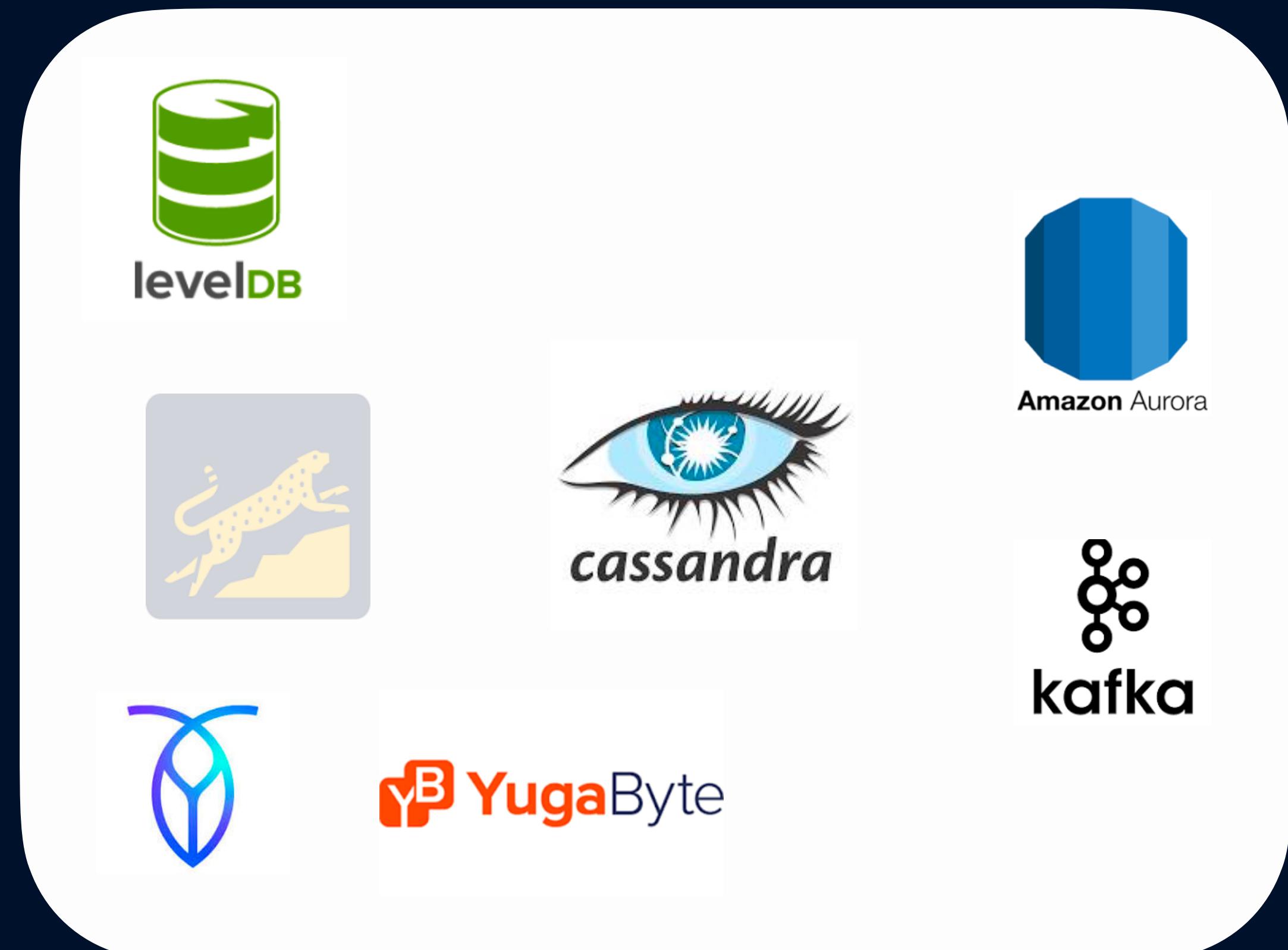


Implementations - old & new

Paged/Heap (older)



Log Structured (newer)



Drop Column - pgrocks (fdw)

Session One - pgbench

pgbench -P 1 -T 300

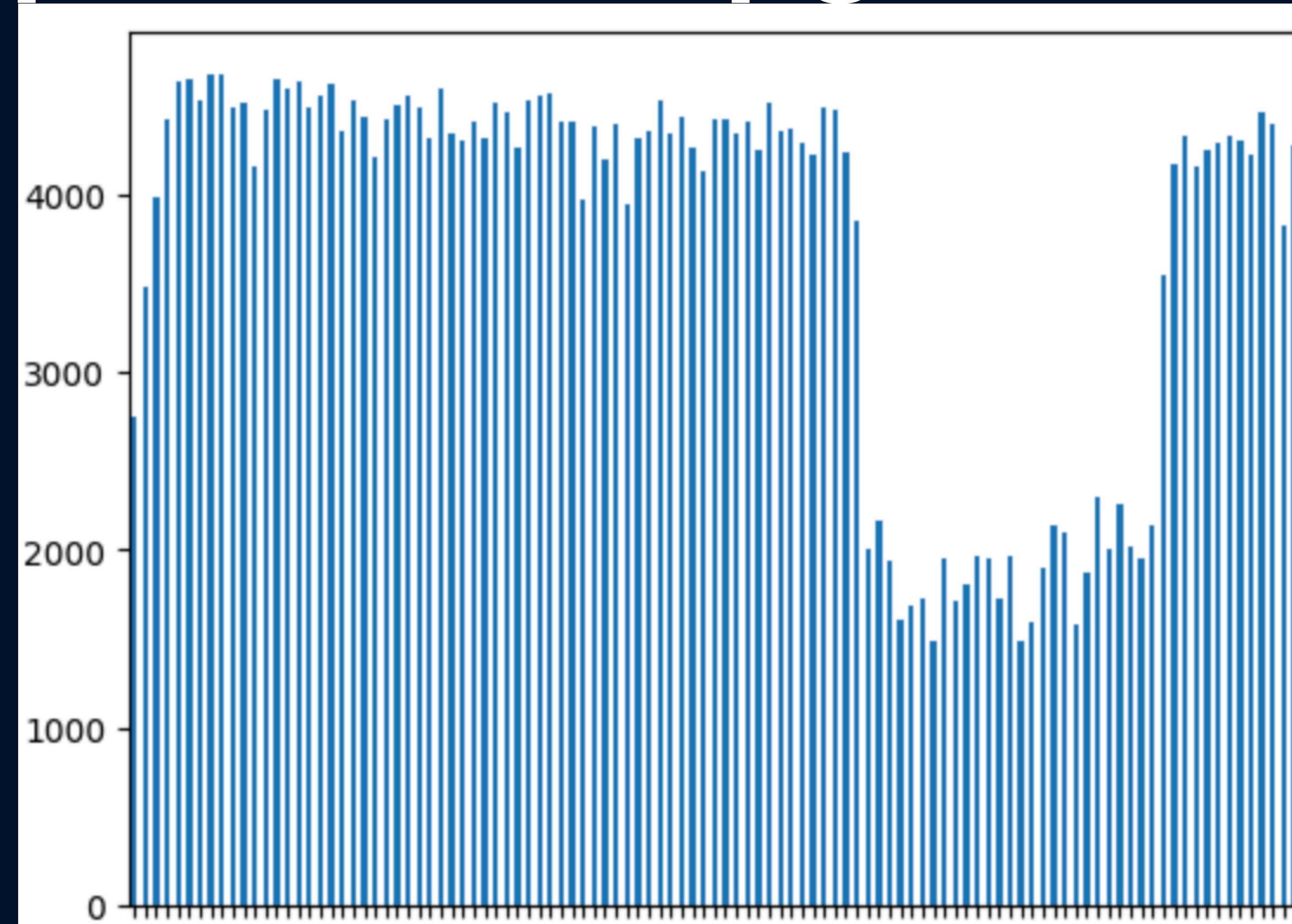
Session Two - Long TXN

```
# alter table big_table drop  
column fil1;
```

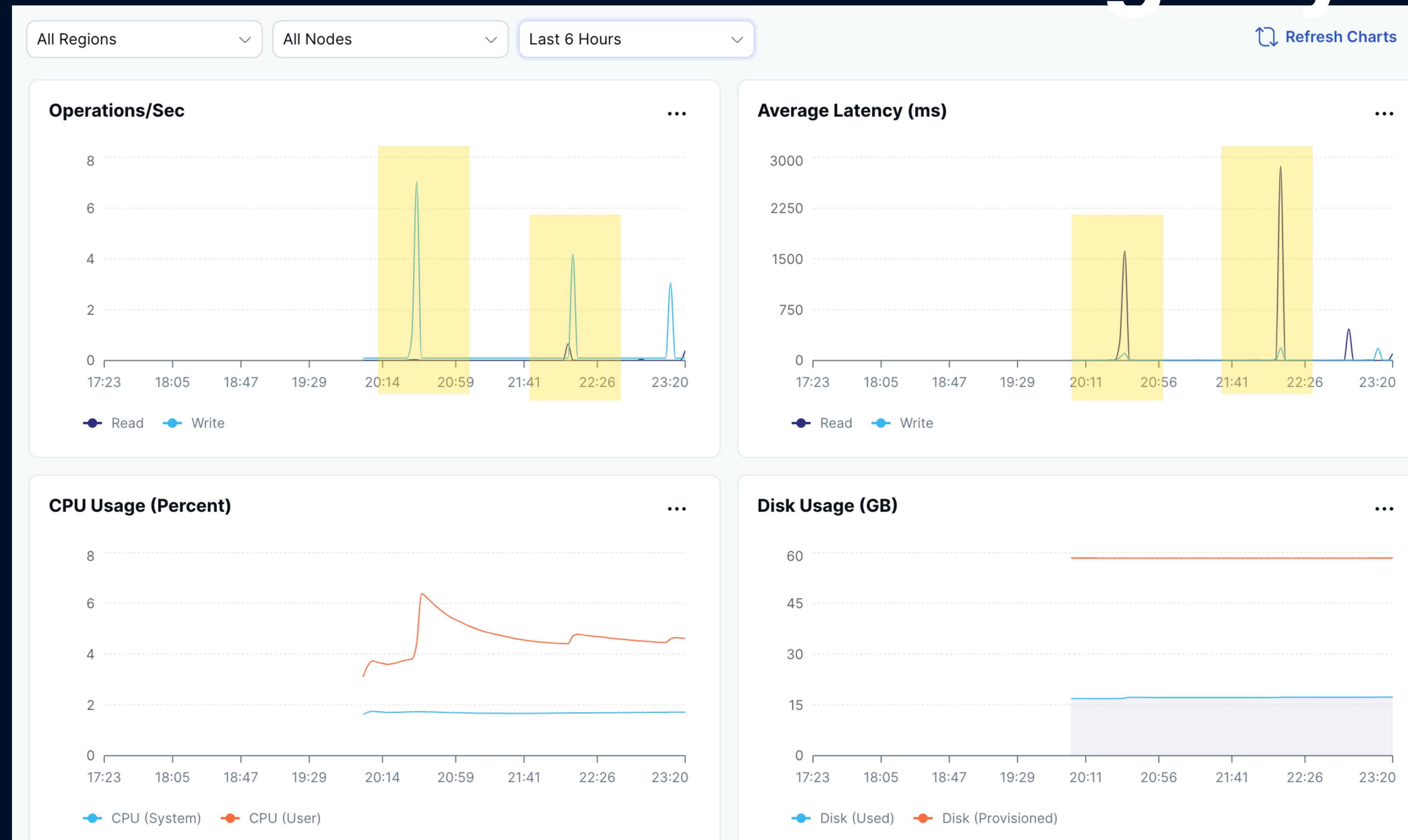
ALTER TABLE

Time: 15.719 ms

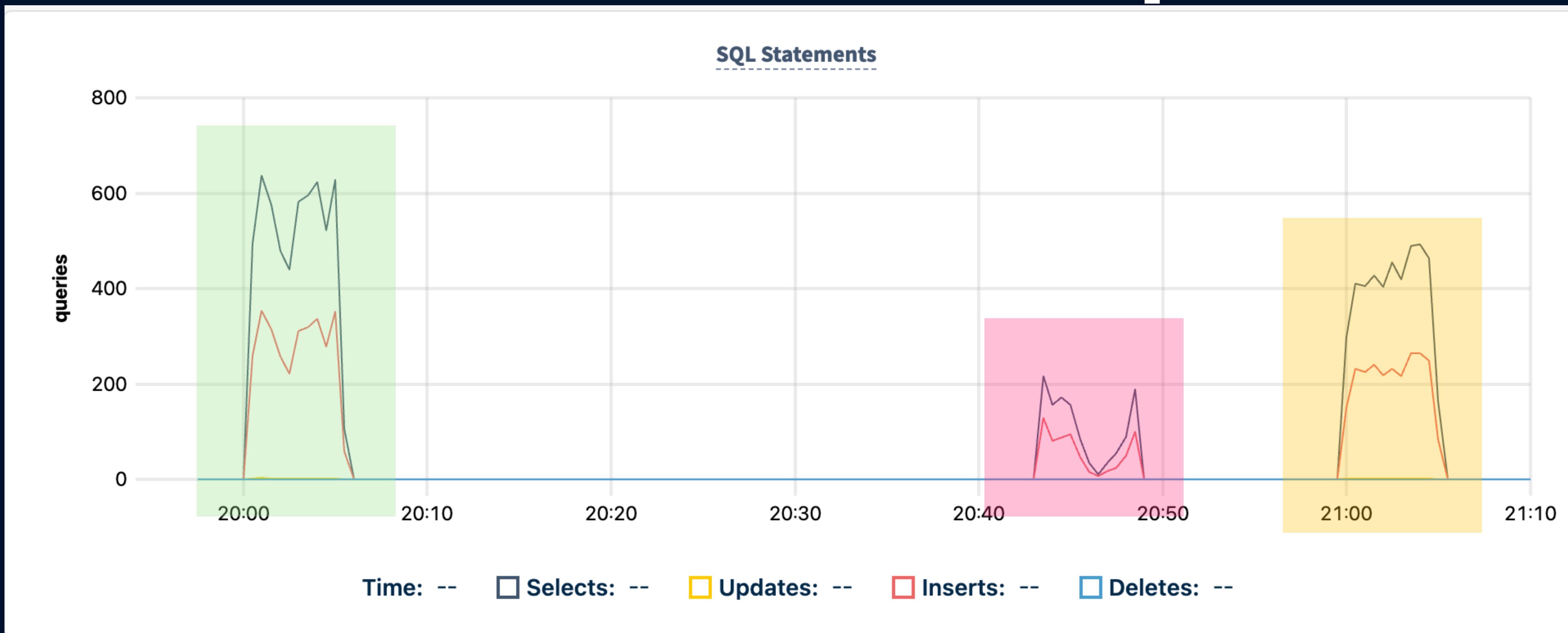
Drop Column - pgrocks (fdw)



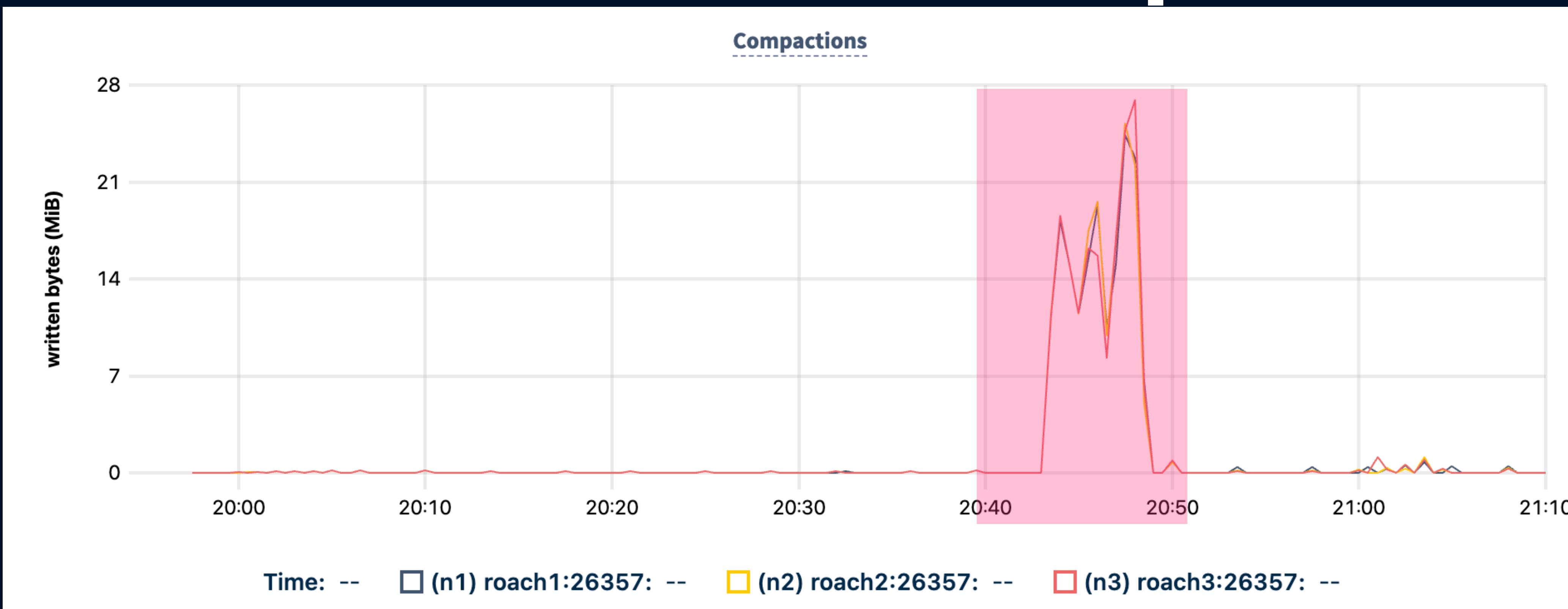
Nice Dashboards ... YugabyteDB



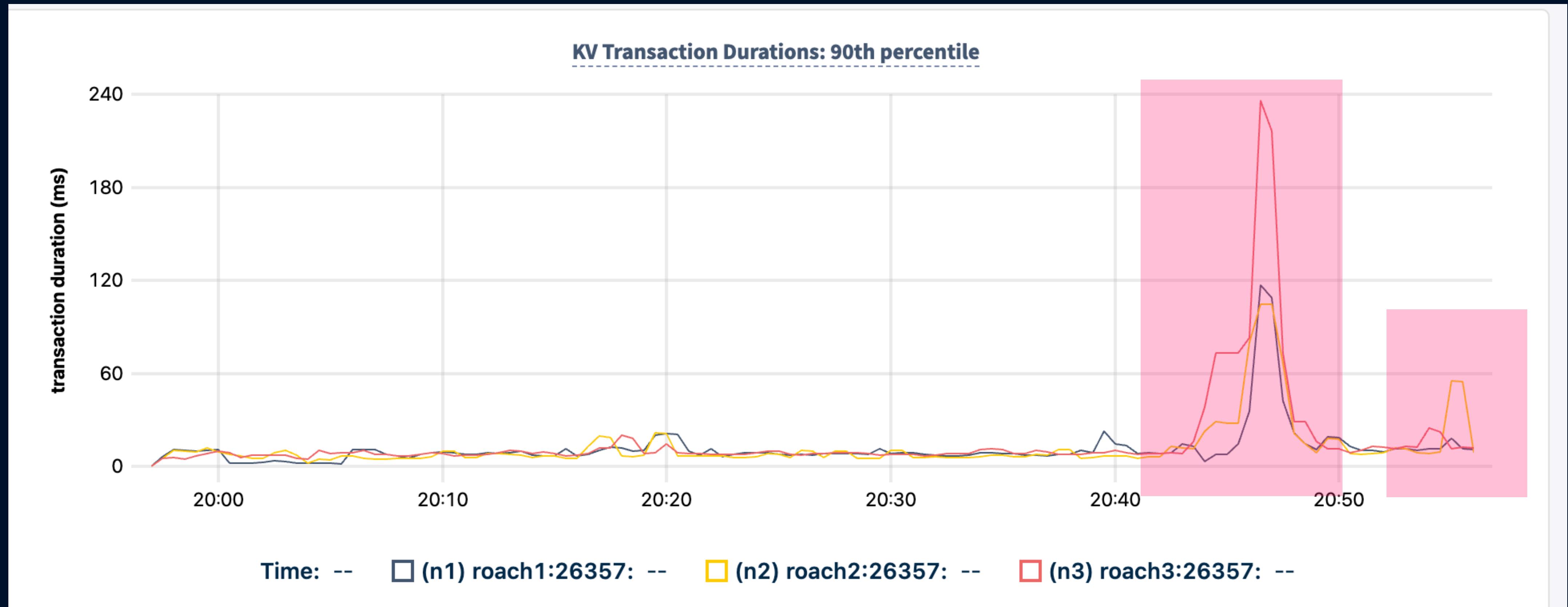
CockroachDB Introspection



CockroachDB Introspection



Time to play ... CockroachDB



Truncate Table - pgrocks (fdw)

Session One - pgbench

```
pgbench -P 1 -T 300
```

Session Two - Truncate

```
# truncate table big_table;
```

```
TRUNCATE TABLE
```

```
Time: 75.829 ms
```

Truncate Table - pgrocks (fdw)

Session One - pgbench

```
pgbench -P 1 -T 300
```

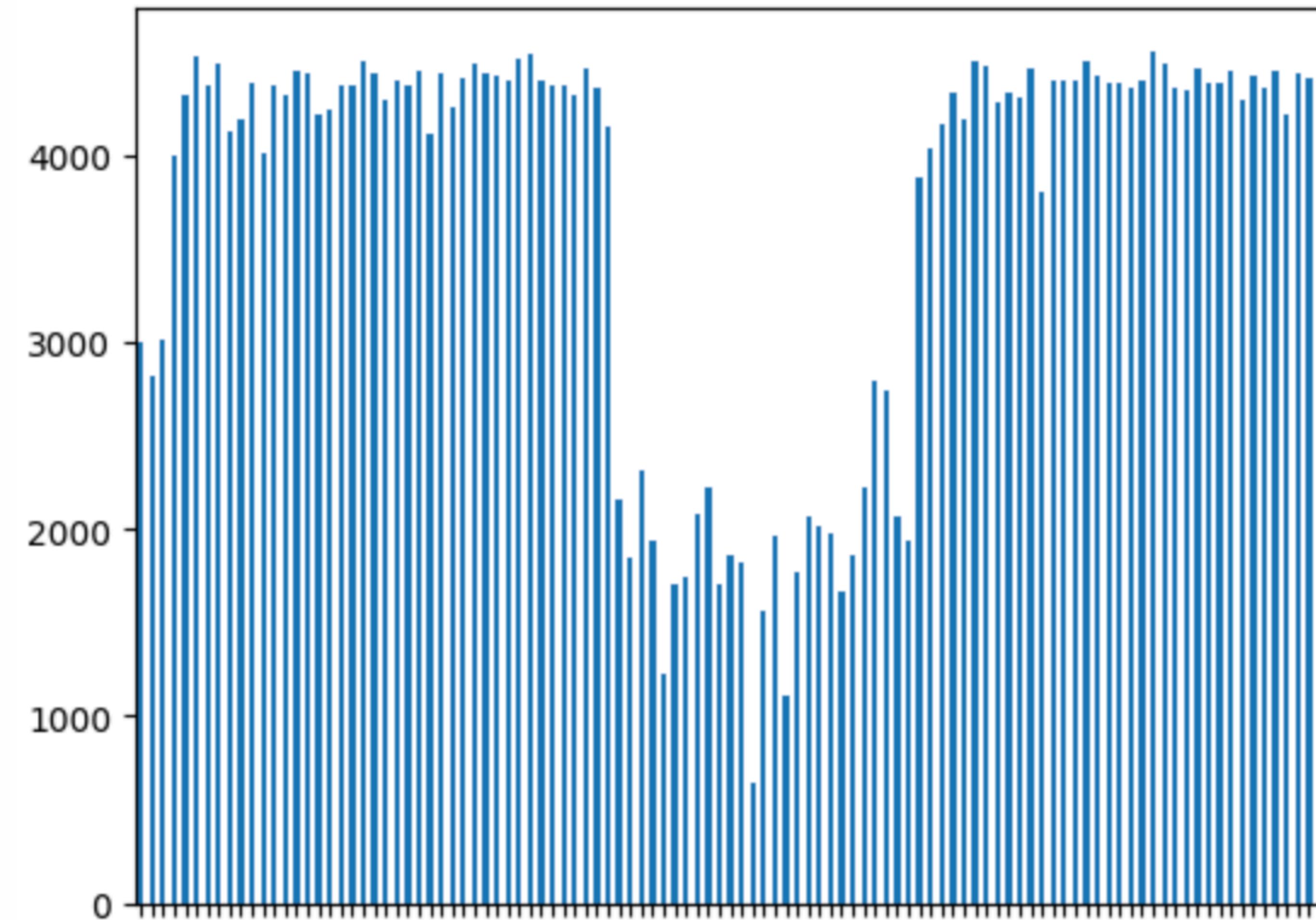
Session Two - Truncate

```
# truncate table big_table;
```

```
TRUNCATE TABLE
```

```
Time: 75.829 ms
```

Truncate Table - pgrocks (fdw)



Conclusion

LSM trees are going to be very useful for **certain workloads** and **distributed databases**

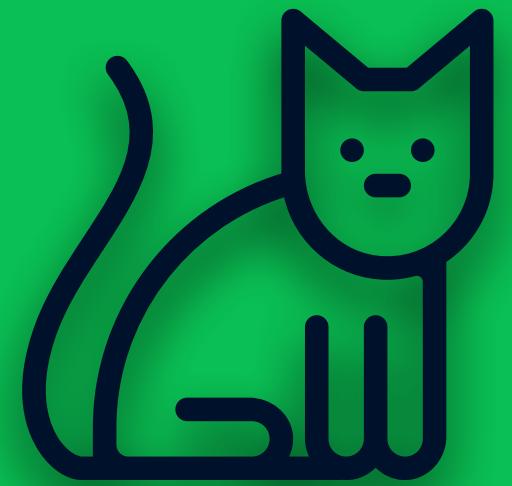
There are some interesting **edge cases** (e.g. drop column, truncate, long transactions, read amplification, distributed reads ..)

Beware being sold easy fixes to problems like traditional vacuum

It may look like the Postgres we know and love, but under the hood it **quite different**

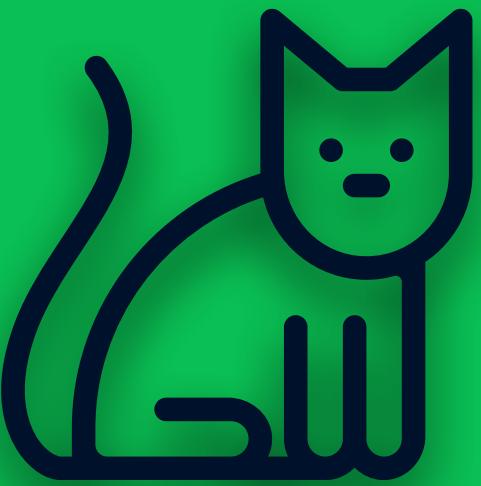
Tips for Python Devs

1) Embrace SQLAlchemy - it is really good



Tips for Python Devs

- 1) Embrace SQLAlchemy - it is really good
- 2) But SQLAlchemy does not scale for free either (100 tps)



Tips for Python Devs

- 1) Embrace SQLAlchemy - it is really good**
- 2) But SQLAlchemy does not scale for free either**
- 3) AND SQLAlchemy is getting easier to support**
- 4) ... oh yes watch out for auto commit madness in SQLAlchemy**



Tips for Python Devs

- 1) Embrace SQLAlchemy - it is really good**
- 2) But SQLAlchemy does not scale for free either**
- 3) AND SQLAlchemy is getting easier to support**
- 4) ... oh yes watch out for auto commit madness in SQLAlchemy**



Links

[Barcelona & Madrid PostgreSQL User Groups - LinkedIn](#)

[Barcelona PostgreSQL User Groups - Meetup](#)

[Madrid PostgreSQL User Groups - Meetup](#)

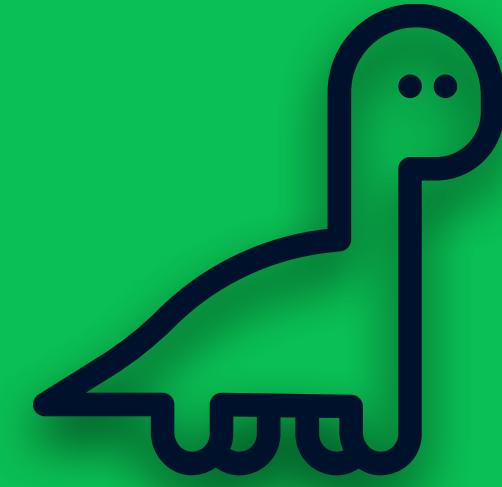
[Talking PostgreSQL Podcast - Tom Lane episode](#)



Q & A ?

Maybe over a beer ...

Dave Pitts - Database Engineer - Adyen (Madrid)



adyen

engineered
for ambition