GIN, BTREE_GIN, GIST, BTREE_GIST, HASH and BTREE indexes on JSONB data (Josef Machytka: Speaker portfolio)

Josef Machytka
2 min read · Feb 9, 2025

(See my bio, other talks in my portfolio and my speaker experience in the covering article.)

Duration: 45 minutes
Target Audience: Application developers, data analysts

Overview: Talk summarizes several months long and still ongoing internal project testing usage and performance of GIN and BTREE_GIN with different operator classes, GIST and BTREE_GIST indexes for GeoJSON data and also standard HASH and BTREE indexes specifically on JSONB data. Tested on several real life datasets with a total size of dozens of GBs. Also, the influence of TOAST compression algorithms, parallelism, memory settings, table statistics on processing JSONB data was tested. Objective of this project was to gather relevant experience to be able to help our customers with their problems, because the majority of articles on the web about JSONB data in PostgreSQL show only trivial examples without any reasonable value for developers solving multiple performance issues related to JSONB data. The talk also discusses practical limitations developers would face if they try to fully decompose JSONB data into relational tables.

Key Takeaways:

- Understanding of use cases and performance of different types of indexes for JSONB data

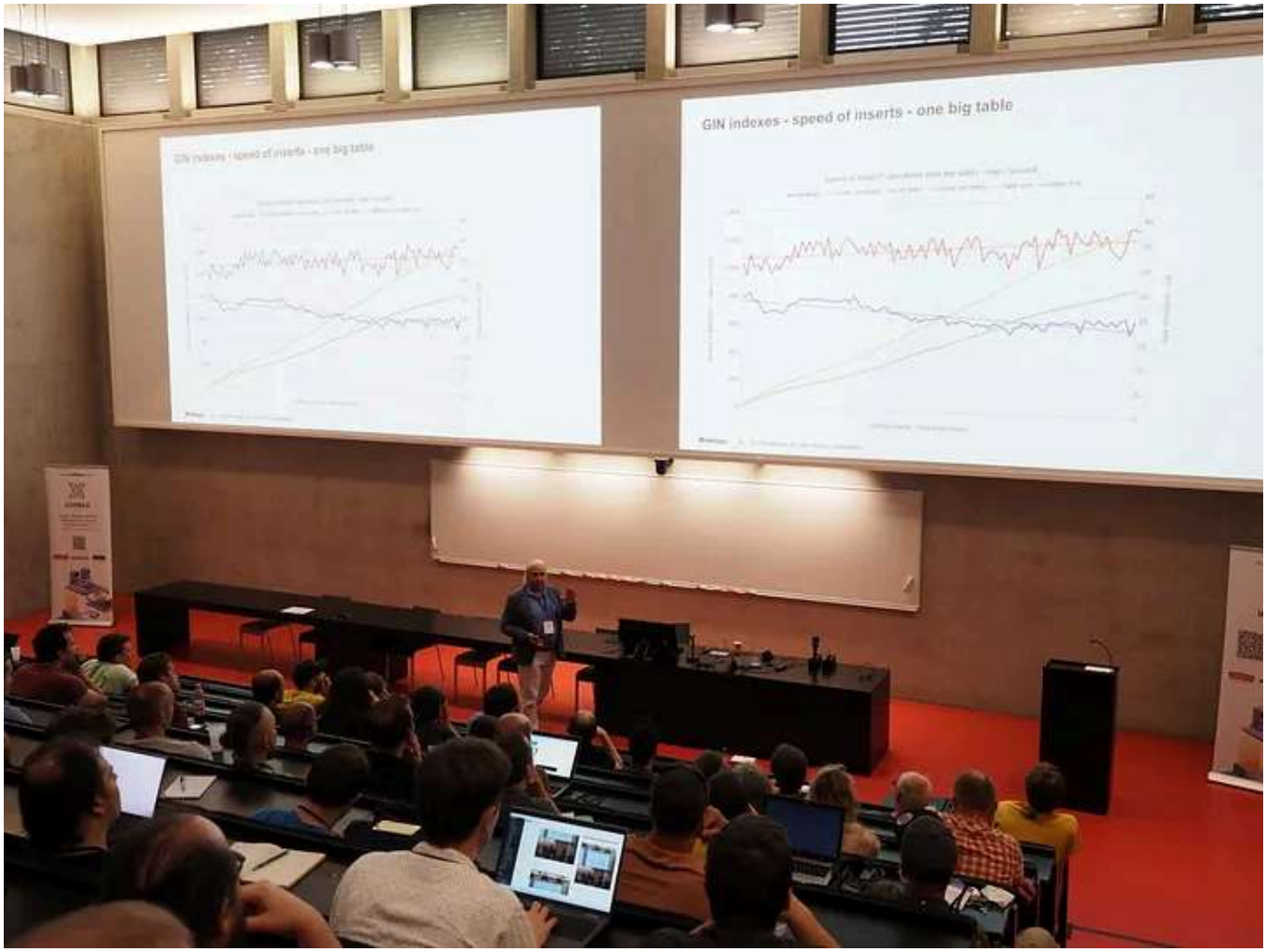
- The impact of system settings like TOAST compression, parallelism, and memory on performance and usage of indexes on JSONB data
- Limitations and considerations for decomposing JSONB data into relational structures
- Insights into internal structure of different types of indexes

Slides: on my GitHub

Presented at:

- Prague PostgreSQL Developer Day 2024 (article on NetApp blog)
- Swiss PG day 2024 (article on NetApp blog)
- Berlin PostgreSQL MeetUp October 2024 (MeetUp entry)

© Tomas Vondra P2D2 — Prague PostgreSQL Developer Day 2024



© Tomas Vondra P2D2 — Prague PostgreSQL Developer Day 2024

© organizers of Swiss PG day — Swiss PG day 2024

Postgresql Speakers Speakerhub Json Jsonb

Edit profile

Written by Josef Machytka

68 Followers · 25 Following

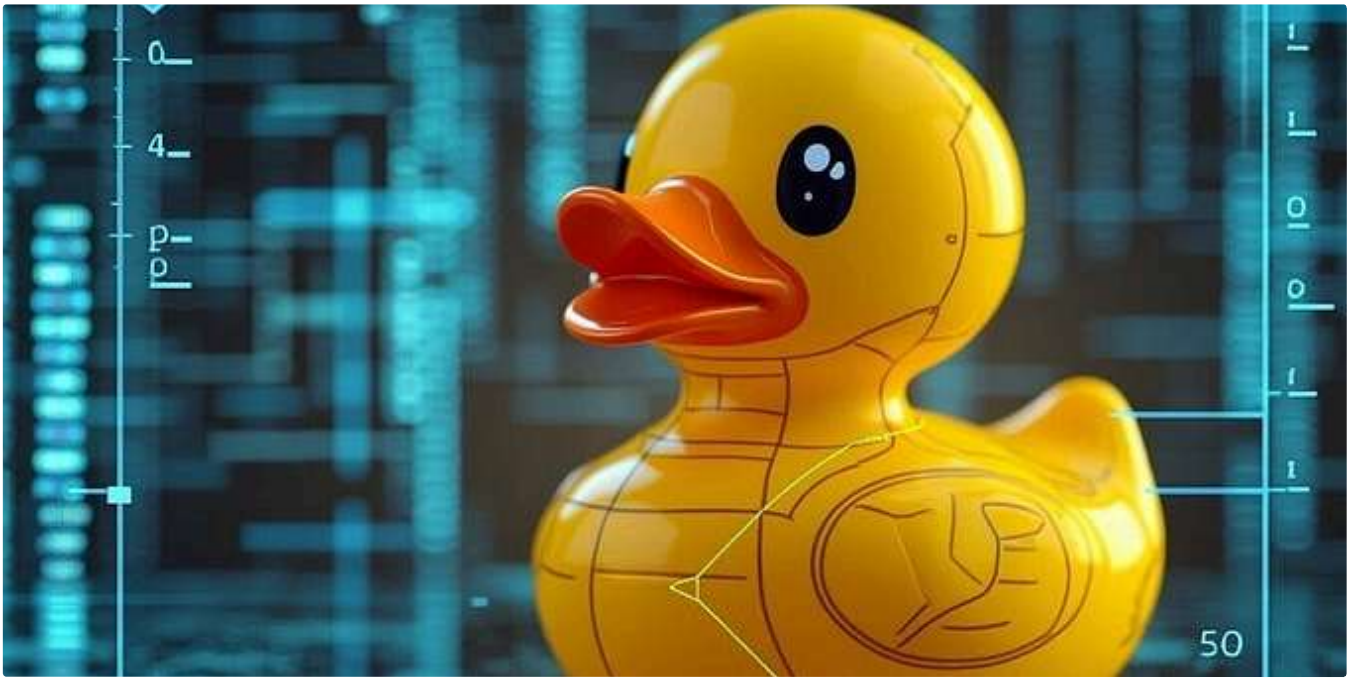
I work as PostgreSQL specialist & database reliability engineer at credativ GmbH.

No responses yet

Josef Machytka he/him

What are your thoughts?

More from Josef Machytka

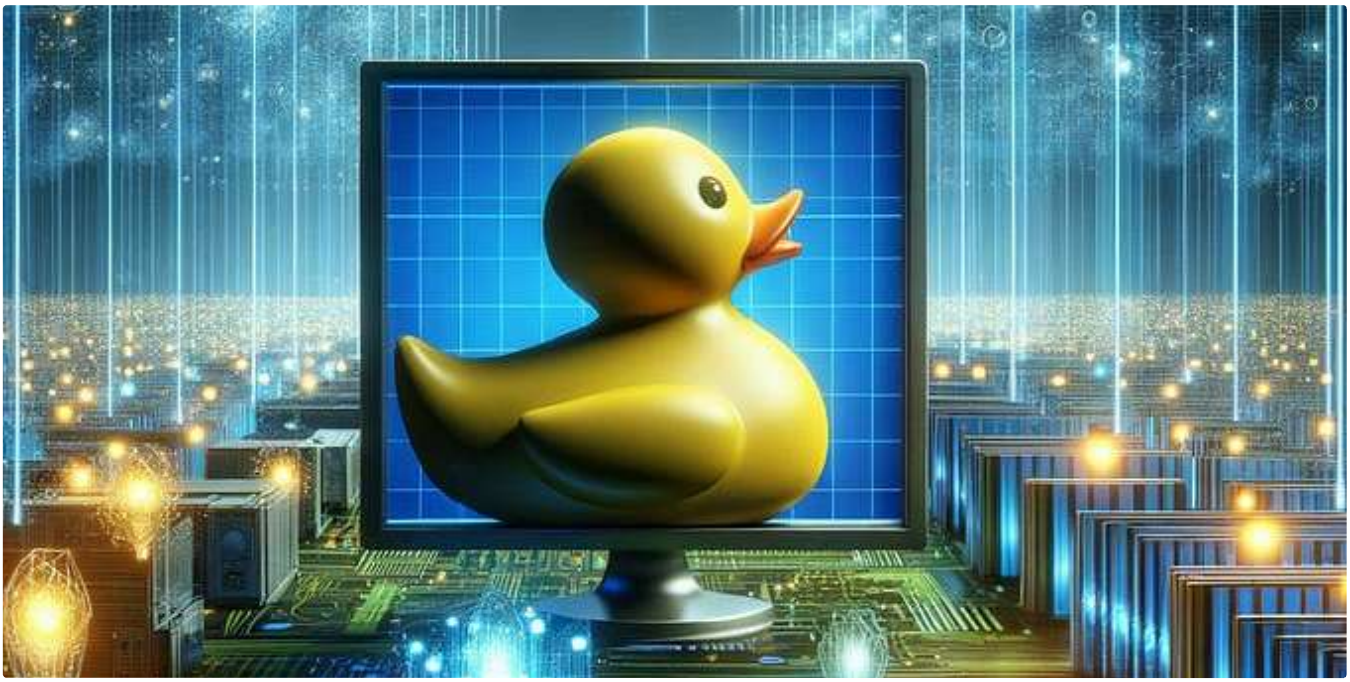


Josef Machytka

DuckDB Database File as a New Standard for Sharing Data?

This is not my original idea; I came across it in an excellent article titled “DuckDB Beyond the Hype” by Alireza Sadeghi. However, it...

Dec 30, 2024 24 3



Josef Machytka

Quick and Easy Statistics and Histograms with DuckDB

DuckDB is an exceptional tool that demonstrates how tasks requiring sometimes considerable manual effort in other tools can be accomplished...

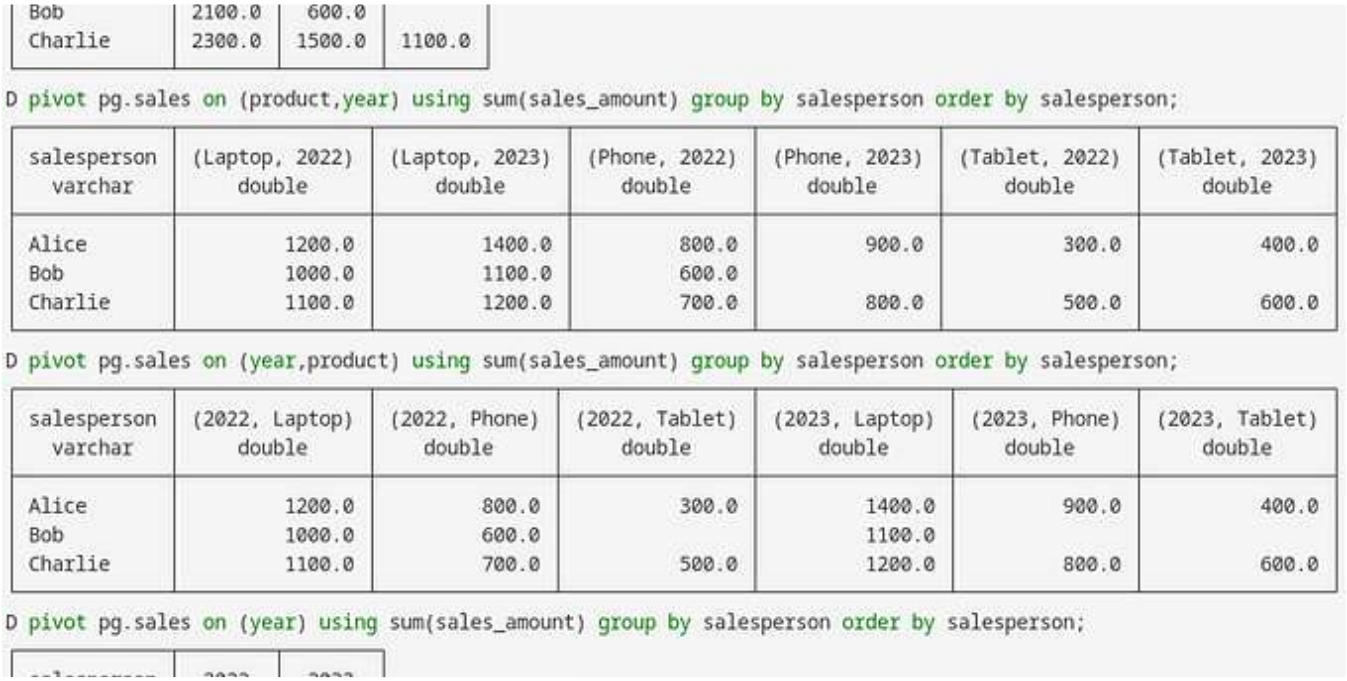
Dec 16, 2024 26

Josef Machytka

PostgreSQL JSONB Operator Classes of GIN Indexes and Their Usage

Throughout 2024, I worked on an internal project exploring the use of JSONB data in PostgreSQL and its various indexing options. During...

Jan 8 5



Bob	2100.0	600.0	
Charlie	2300.0	1500.0	1100.0

```
D pivot pg.sales on (product,year) using sum(sales_amount) group by salesperson order by salesperson;
```

salesperson varchar	(Laptop, 2022) double	(Laptop, 2023) double	(Phone, 2022) double	(Phone, 2023) double	(Tablet, 2022) double	(Tablet, 2023) double
Alice	1200.0	1400.0	800.0	900.0	300.0	400.0
Bob	1000.0	1100.0	600.0			
Charlie	1100.0	1200.0	700.0	800.0	500.0	600.0

```
D pivot pg.sales on (year,product) using sum(sales_amount) group by salesperson order by salesperson;
```

salesperson varchar	(2022, Laptop) double	(2022, Phone) double	(2022, Tablet) double	(2023, Laptop) double	(2023, Phone) double	(2023, Tablet) double
Alice	1200.0	800.0	300.0	1400.0	900.0	400.0
Bob	1000.0	600.0		1100.0		
Charlie	1100.0	700.0	500.0	1200.0	800.0	600.0

```
D pivot pg.sales on (year) using sum(sales_amount) group by salesperson order by salesperson;
```


Josef Machytka

Easy and Intelligent Pivot Tables with DuckDB

After exploring the various capabilities of DuckDB in my earlier articles, I want to focus more on its powerful data analytical...

Dec 4, 2024 7 1

See all from Josef Machytka

Recommended from Medium

Query Optimization Method	Execution Time	Performance Improvement (vs. No Index)
No Index	42,049 ms (≈42 sec)	Baseline
With B-tree Index	9,684 ms (≈9.7 sec)	77% faster
With Chunk-Skipping Index + Columnstore	304 ms (0.3 sec)	99.28% faster

In Timescale by Team Timescale

Handling Billions of Rows in PostgreSQL

Here's how to scale PostgreSQL to handle billions of rows using Timescale compression and chunk-skipping indexes.

Jan 17 72 1

Leapcell

Mastering Go Compiler Optimization for Better Performance

Leapcell: The Next-Gen Serverless Platform for Web Hosting, Async Tasks, and Redis

5d ago 3

Lists

Staff picks

819 stories · 1637 saves

Stories to Help You Level-Up at Work

19 stories · 944 saves

Self-Improvement 101

20 stories · 3324 saves

Productivity 101

20 stories · 2796 saves

Kiran Maan

STOP Using Python Dictionaries Like This!!!

Sometimes…I see that some people use Python dictionaries incorrectly.

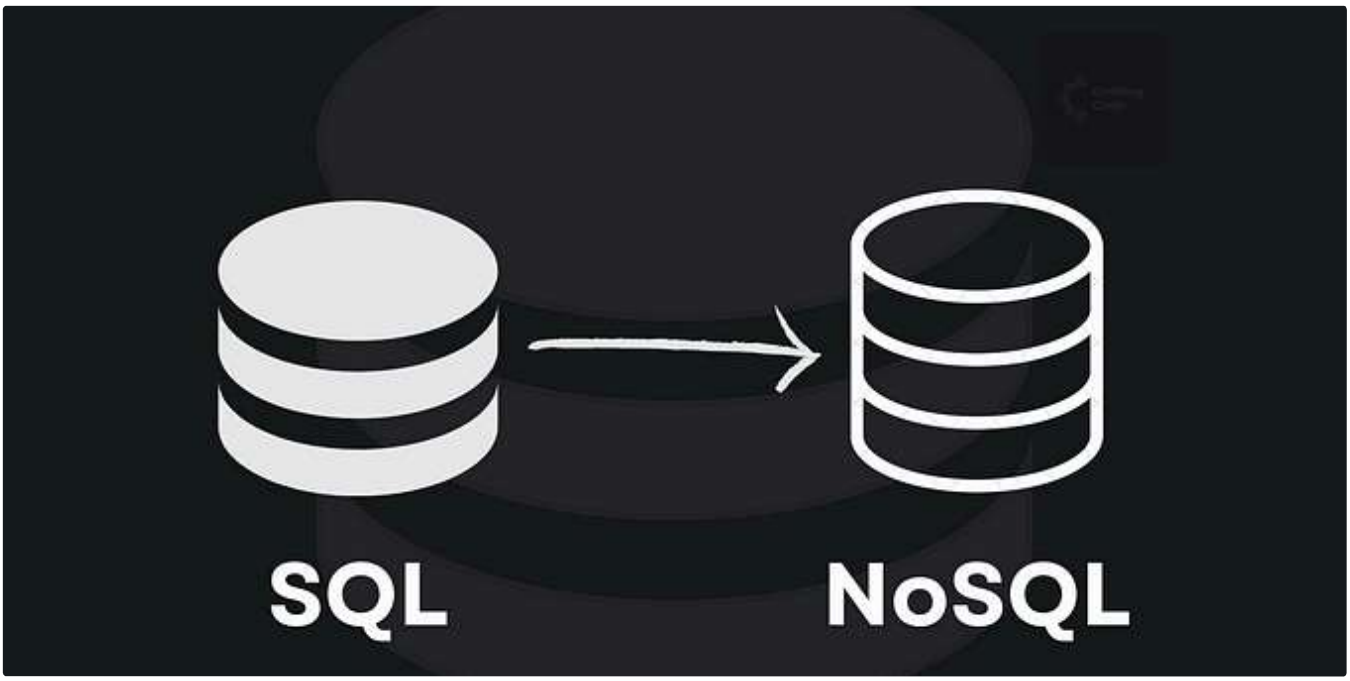
Feb 20 303 6

Adarsh Rajesh

Understanding Collations in PostgreSQL

Sorting, ordering and comparisons are fundamental to any database. However, are they always deterministic and consistent across all...

5d ago 8

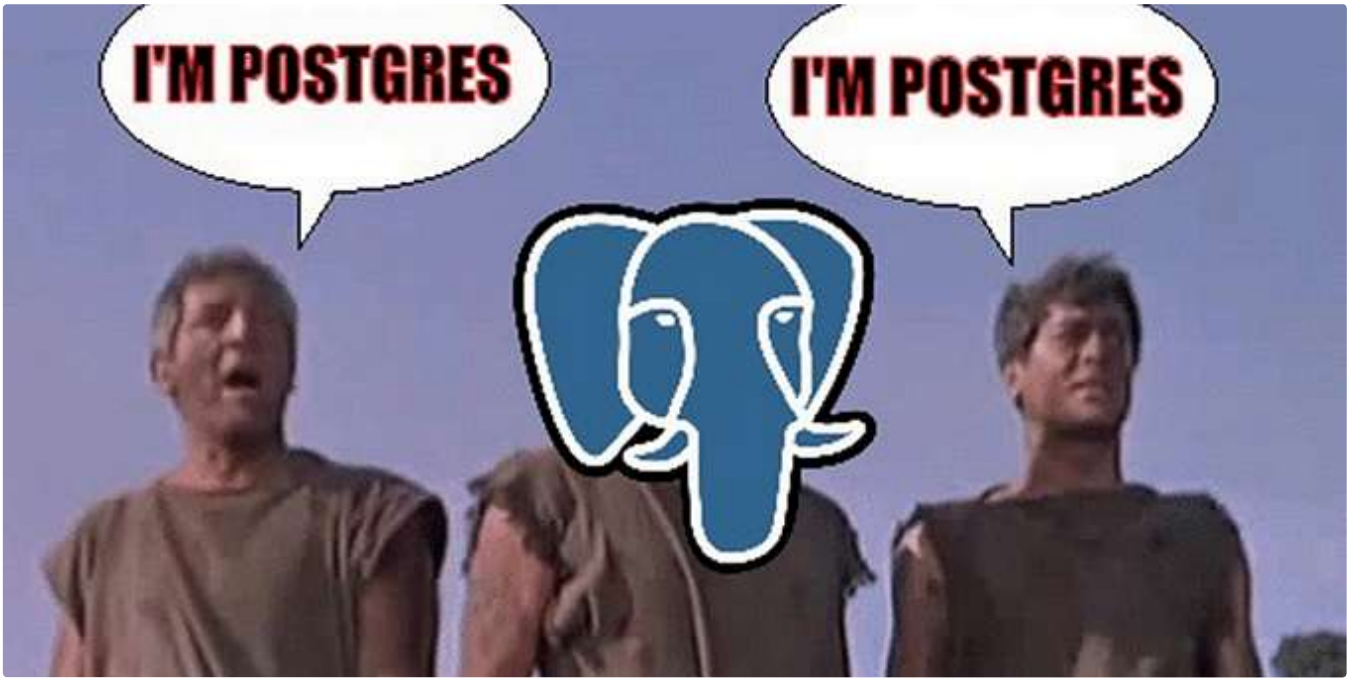


In Stackademic by Crafting-Code

I Dropped SQL for NoSQL. Our App Now Handles 5x the Traffic

The ‘crazy’ database switch that proved our critics wrong

Feb 17 652 22



Mayur (Do not drink & database)

Postgres Is

Update: In response to a trademark notice from the PostgreSQL Community Association of Canada, domain has been changed from “Postgres.Is”...

Feb 17 11 1

See more recommendations