

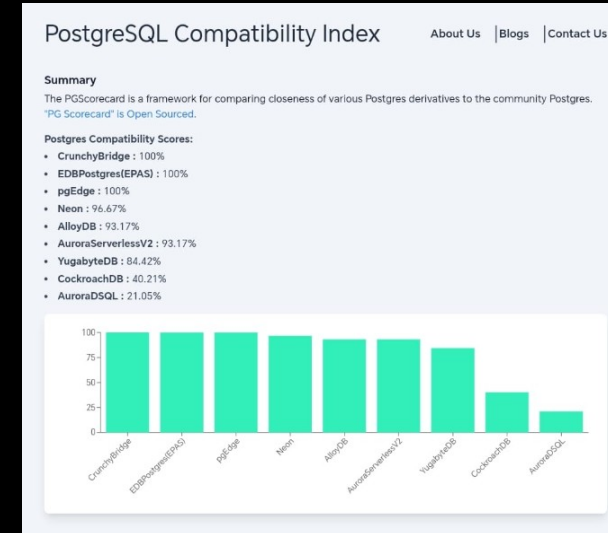
Mayur@Veeam

# Why Developers Using AI Is Great For DBA Job Security?

ME@POSTGRES SLACK



PGSCORECARD.COM



DATABASE COMEDY BLOG



PG EXTENSIONS DEVELOPMENT

# Agenda



Brief introduction to AI Hallucinations and LLM lies.



Prevalence of AI generated fake data/articles in PostgreSQL community.



Concept of Data Cannibalization and limits of AI training.



What AI can do for your Postgres database (with human DBA oversight)?

# AI Hallucinations





# Air Canada ordered to pay customer who was misled by airline's chatbot

Company claimed its chatbot 'was responsible for its own actions' when giving wrong information about bereavement fare



AI hallucination arises from LLMs' reliance on statistical patterns, prioritizing coherence over factual accuracy.

Hallucination is intrinsic because LLMs optimize for token prediction, not truthfulness.

No absolute truth: It can't understand context or verify reality like humans.

AI can't understand context or verify reality like humans





Optimizing for next token, not reality.

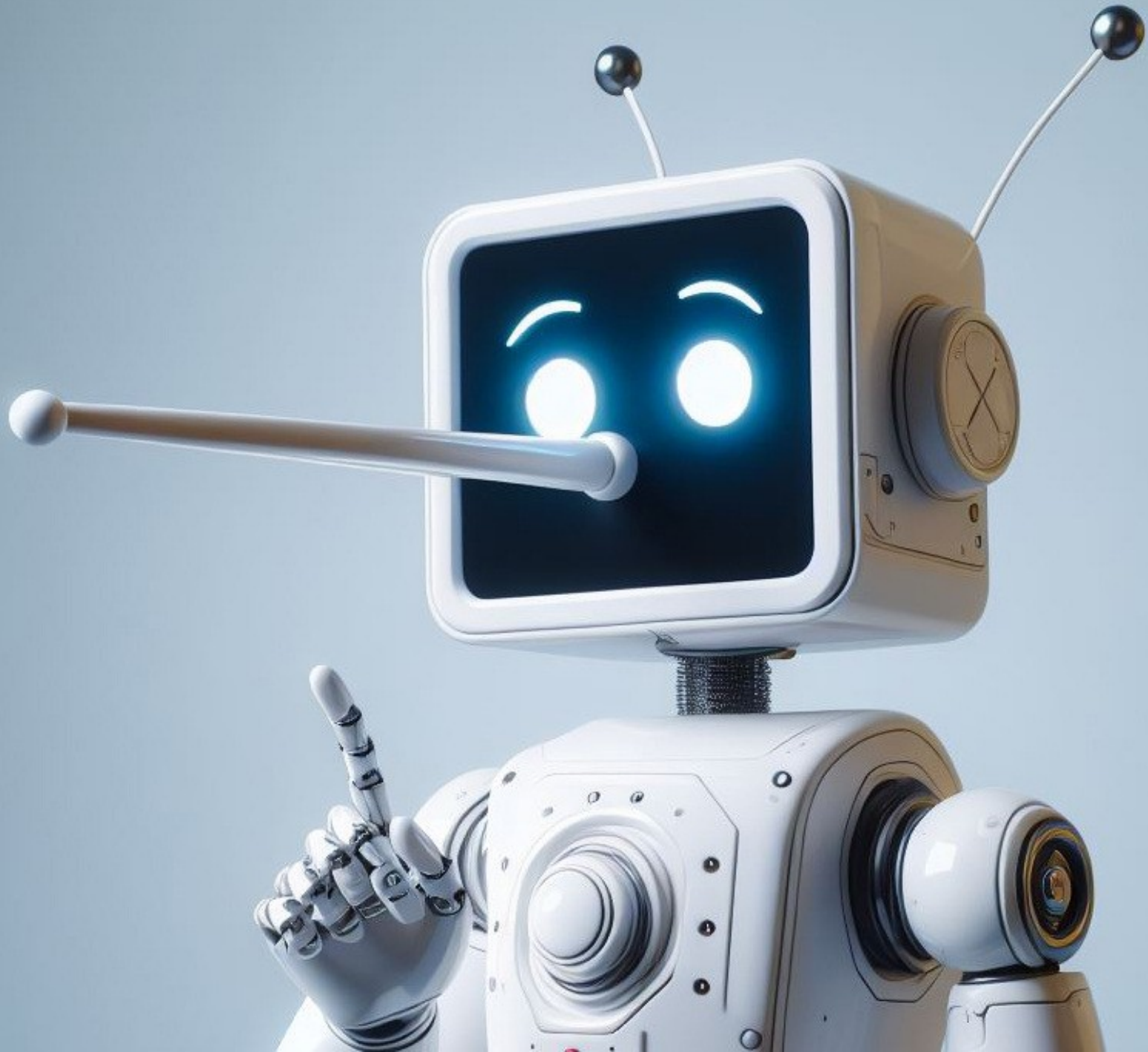




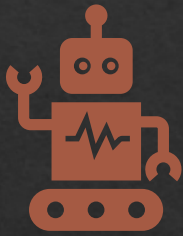
### **It's all hallucination**

The takeaway here? It's all hallucination, but we only call it that when we notice it's wrong. The problem is, large language models are so good at what they do that what they make up looks right most of the time. And that makes trusting them hard.

# ChatGPT Lies



# Minimize Hallucinations



Retrieval-  
augmented  
generation.



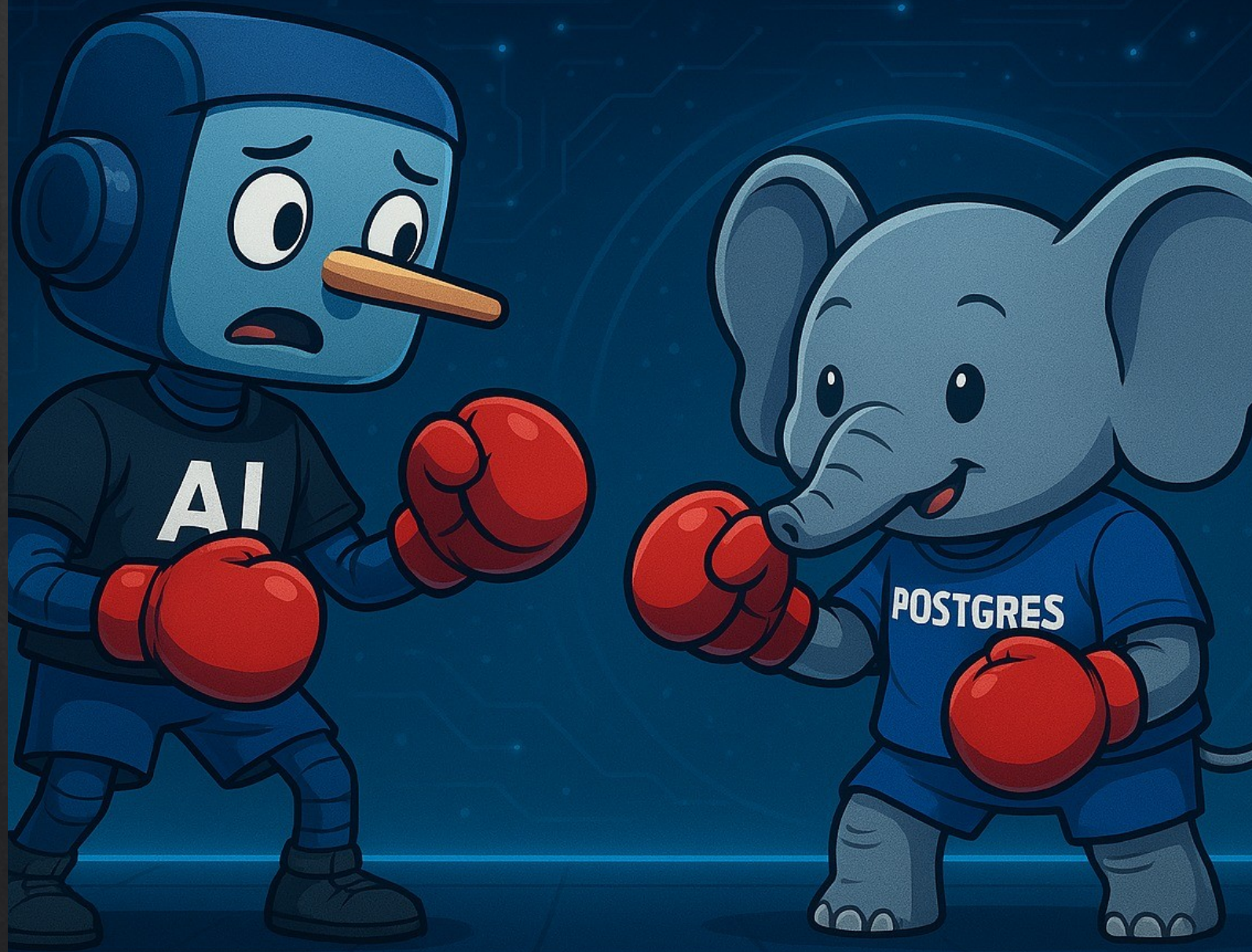
Auto chain-of-  
thought prompting.



Larger models.



# LLMs vs Postgresql Community







623 followers

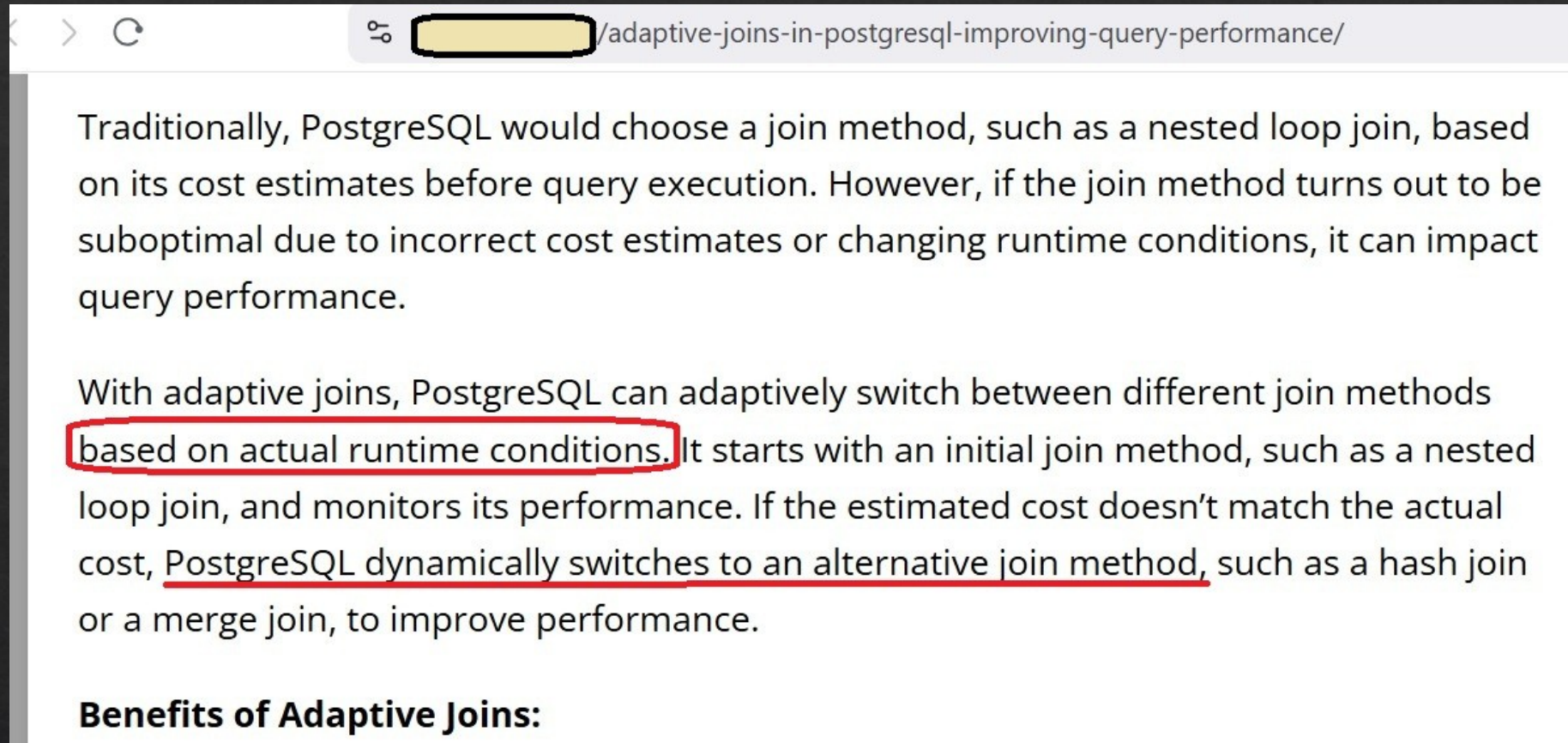
6d • 🌐

Adaptive joins in [#PostgreSQL](#) optimize performance by dynamically adjusting methods during execution. This article examines their benefits and impact on efficiency. [#PostgreSQLPerformance](#) [#dba](#)



Adaptive Joins in PostgreSQL: Improving Query Performance

AI invents adaptive optimizer in Postgres.



The screenshot shows a web browser window with a single tab. The address bar contains a URL starting with a domain name that has been redacted with a yellow box, followed by "/adaptive-joins-in-postgresql-improving-query-performance/". The page content is white with black text. The first paragraph explains that PostgreSQL traditionally chooses a join method based on cost estimates, but this can be suboptimal. The second paragraph describes adaptive joins, where PostgreSQL can switch between join methods based on actual runtime conditions. In this paragraph, the phrase "based on actual runtime conditions." is enclosed in a red rectangular box, and the phrase "PostgreSQL dynamically switches to an alternative join method," is underlined in red. The third paragraph is a section header "Benefits of Adaptive Joins:".

Traditionally, PostgreSQL would choose a join method, such as a nested loop join, based on its cost estimates before query execution. However, if the join method turns out to be suboptimal due to incorrect cost estimates or changing runtime conditions, it can impact query performance.

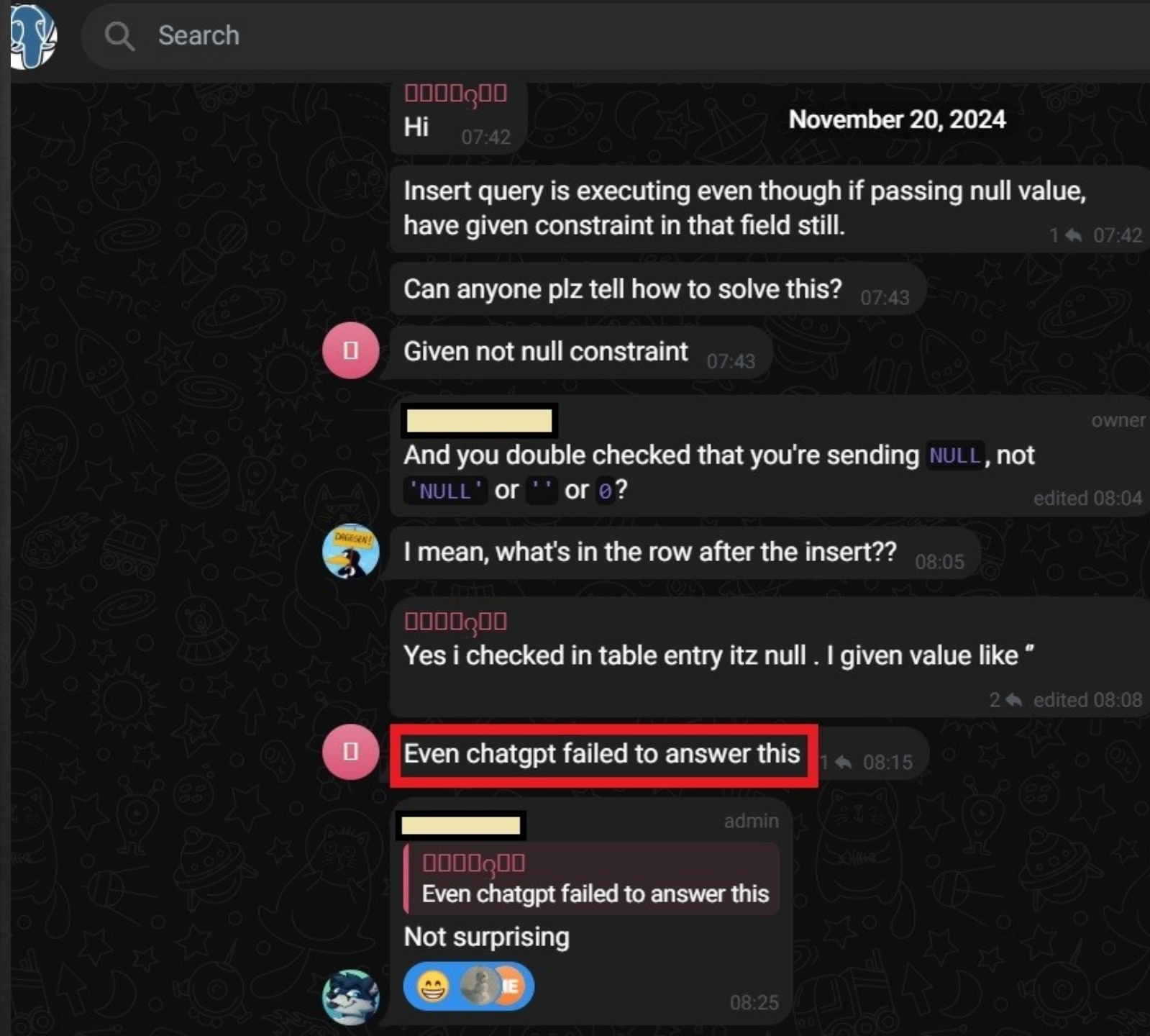
With adaptive joins, PostgreSQL can adaptively switch between different join methods **based on actual runtime conditions.** It starts with an initial join method, such as a nested loop join, and monitors its performance. If the estimated cost doesn't match the actual cost, PostgreSQL dynamically switches to an alternative join method, such as a hash join or a merge join, to improve performance.

**Benefits of Adaptive Joins:**

AI trained on Oracle data perhaps

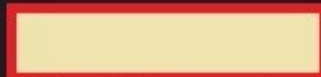


# Even ChatGPT Failed to Answer This!



# Non-existent TTL indexes

## Thread



Today at 2:41 PM

Hi - We are looking to use TTL (Time To Live) index to delete records based on

I am getting conflicting views on internet about TTL index feature being available in PostgreSQL.

Couldn't find any links in PostgreSQL documentation around this. Can someone help?

- If TTL index feature is available in PostgreSQL ?
- If yes, any related link would be really useful.

Thanks 🙏

3 replies



Today at 2:46 PM

You can create an index on a date column to make deletions based on the date.

However, PostgreSQL doesn't have any automation out-of-the-box to delete records older than X. You can automate your application to do that or use a cron job to run the deletions.



Today at 2:46 PM

It's not a thing in postgresql. But an indexed timestampz field will let you delete records older than X. Matthias says.



enhancing-postgresql-performance-with-background-workers-initialization-configuration-and-...

## Optimizing PostgreSQL with the Help of Background Workers

Aug 4, 2024 · 4 min read

### Leveraging Background Workers in PostgreSQL for Enhanced Performance

PostgreSQL's background workers provide a powerful mechanism for extending the database's capabilities by running custom processes alongside the core PostgreSQL server. These background workers can perform various tasks such as maintenance operations, data processing, and monitoring. Understanding how to initialize, register, and manage background workers is crucial for harnessing their full potential. In this detailed blog, we explore the intricacies of using background workers in PostgreSQL, including how to register them, configure their behavior, and manage their lifecycle.

# Not All Fun & Games



## Conclusion

Background workers in PostgreSQL provide a flexible and powerful way to extend database functionality. By understanding how to initialize, configure, and manage these processes, you can effectively leverage them to perform a variety of tasks, from maintenance operations to complex data processing. Properly using background workers can significantly enhance the performance and capabilities of your PostgreSQL deployment.

# Not All Fun & Games



Is "pg\_upgrade --link" method of postgres upgrade possible on windows for postgres 14 to 16?

No, `pg_upgrade --link` is **not** possible on Windows for **PostgreSQL 14 to 16** (or any version).

### Why `--link` Doesn't Work on Windows:

#### 1. Hard Linking is Not Supported for Directories in NTFS

- PostgreSQL's `--link` mode relies on **hard links** to reference files instead of copying them.
- While **NTFS supports hard links for individual files**, it **does not support hard links for directories**, which PostgreSQL requires.

#### 2. Explicit Restriction in PostgreSQL Source Code



- The `pg_upgrade` utility **disables `--link` mode on Windows** because of filesystem limitations.

Ask anything

# Expert Liar



Name	Full Path	Type	Target Path	Created Time
2613	D:\pgdata17\base\1\2613	Hard Link	D:\pgdata\base\1\2613	2/6/2025 2:58:20 PM
2613	D:\pgdata17\base\5\2613	Hard Link	D:\pgdata\base\5\2613	2/6/2025 2:58:20 PM
2613	D:\pgdata17\base\71251465\2613	Hard Link	D:\pgdata\base\71251465\2613	2/6/2025 2:59:35 PM
2683	D:\pgdata17\base\1\2683	Hard Link	D:\pgdata\base\1\2683	2/6/2025 2:58:20 PM
2683	D:\pgdata17\base\5\2683	Hard Link	D:\pgdata\base\5\2683	2/6/2025 2:58:20 PM
2683	D:\pgdata17\base\71251465\2683	Hard Link	D:\pgdata\base\71251465\2683	2/6/2025 2:59:35 PM
71251966	D:\pgdata17\base\71251465\71251966	Hard Link	D:\pgdata\base\71251465\71251966	2/6/2025 3:00:00 PM
71251966_fsm	D:\pgdata17\base\71251465\7125196...	Hard Link	D:\pgdata\base\71251465\71251966_fsm	2/6/2025 3:00:07 PM
71251966_vm	D:\pgdata17\base\71251465\7125196...	Hard Link	D:\pgdata\base\71251465\71251966_vm	2/6/2025 3:00:37 PM
71251974	D:\pgdata17\base\71251465\71251974	Hard Link	D:\pgdata\base\71251465\71251974	2/6/2025 3:00:00 PM
71251975	D:\pgdata17\base\71251465\71251975	Hard Link	D:\pgdata\base\71251465\71251975	2/6/2025 3:00:00 PM
71252328	D:\pgdata17\base\71251465\71252328	Hard Link	D:\pgdata\base\71251465\71252328	2/6/2025 3:00:00 PM
71252331	D:\pgdata17\base\71251465\71252331	Hard Link	D:\pgdata\base\71251465\71252331	2/6/2025 3:00:00 PM
71252332	D:\pgdata17\base\71251465\71252332	Hard Link	D:\pgdata\base\71251465\71252332	2/6/2025 3:00:00 PM
71252338	D:\pgdata17\base\71251465\71252338	Hard Link	D:\pgdata\base\71251465\71252338	2/6/2025 3:00:00 PM
71252344	D:\pgdata17\base\71251465\71252344	Hard Link	D:\pgdata\base\71251465\71252344	2/6/2025 3:00:00 PM
71252345	D:\pgdata17\base\71251465\71252345	Hard Link	D:\pgdata\base\71251465\71252345	2/6/2025 3:00:00 PM
71252349	D:\pgdata17\base\71251465\71252349	Hard Link	D:\pgdata\base\71251465\71252349	2/6/2025 3:00:00 PM
71252352	D:\pgdata17\base\71251465\71252352	Hard Link	D:\pgdata\base\71251465\71252352	2/6/2025 3:00:00 PM
71252353	D:\pgdata17\base\71251465\71252353	Hard Link	D:\pgdata\base\71251465\71252353	2/6/2025 3:00:00 PM
71252357	D:\pgdata17\base\71251465\71252357	Hard Link	D:\pgdata\base\71251465\71252357	2/6/2025 3:00:00 PM
71252371	D:\pgdata17\base\71251465\71252371	Hard Link	D:\pgdata\base\71251465\71252371	2/6/2025 3:00:00 PM
71252376	D:\pgdata17\base\71251465\71252376	Hard Link	D:\pgdata\base\71251465\71252376	2/6/2025 3:00:00 PM
71252377	D:\pgdata17\base\71251465\71252377	Hard Link	D:\pgdata\base\71251465\71252377	2/6/2025 3:00:00 PM
71252408	D:\pgdata17\base\71251465\71252408	Hard Link	D:\pgdata\base\71251465\71252408	2/6/2025 3:00:00 PM
71252414	D:\pgdata17\base\71251465\71252414	Hard Link	D:\pgdata\base\71251465\71252414	2/6/2025 3:00:00 PM

Test exposes the lie.





Jan 22nd at 6:22 PM

Is chatgpt mistaken here about reindexing:

image.png ▼

#### REINDEX

- Reason: Index structures can also change between major versions. If the upgrade process (e.g., `pg_upgrade`) does not fully rebuild indexes, they may remain suboptimal or incompatible with the new version.
- Scenarios for REINDEX:
  - If `pg_upgrade` with the `--link` option was used, the index structure may not have been rebuilt, making a `REINDEX` necessary.
  - If there were issues during the upgrade, corrupted or outdated index structures could affect performance.
- Recommendation: Rebuild indexes to ensure they align with the new PostgreSQL version's index implementation.

#### Conclusion:

While not absolutely mandatory, running

`VACUUM`, `ANALYZE`, and `REINDEX` is a best practice after a major PostgreSQL version upgrade. These steps help ensure database performance and stability by adapting the data structures and statistics to the upgraded version. (edited)

# Unnecessary workload increase

✓

I am restricted to using only two servers/hosts (hostA and hostB) and need to setup postgres high availability using patroni. Provide all the steps needed to do this.

Reasoned about setting up Postgres HA with Patroni for 7 seconds ✓

Below is a general example of how to set up a Patroni-based PostgreSQL high-availability cluster using **only two hosts** (HostA and HostB). *Bear in mind that a two-host cluster is not best practice*, because with only two nodes, you cannot achieve a proper quorum-based consensus. This can lead to split-brain scenarios if there's a network partition or one node fails. A typical recommendation is to have a minimum of three nodes (or use an external DCS that resides on a third node) to achieve reliable HA.

However, if you only have two servers and must proceed, the steps below outline one way to do it. The example uses **etcd** as the distributed configuration store (DCS), but you can substitute Consul or ZooKeeper similarly.

---

## 1. Prepare the Environment

# 2 Node Patroni HA cluster: LLMI VS Human

# Human response

## Two-nodes failover cluster #748

🔒 Closed



Does anyone do it?

I found only 3 decision how to create it but all they is not stable:

1. patroni **synchronous** mode + two different **etcd** on nodes(have a cluster ip (with **keepalived**) and used **rsync**) - if have a transactions after copyng and before fail - **second** node cannot start
2. patroni **asynchronous** mode, everything else is the same - it start, but data loses
3. Just idea: same as 1 way, but create etcd cluster, on fail delete failed node from cluster, after start recover cluster - too many risks, or not?



It is not possible. How do you tell the network failure between two nodes apart from the failure of the master?

If you run Patroni with 2 nodes Etcd (or whatever else) cluster running on top of 2 nodes Postgres cluster, the node puts Etcd to the read-only mode forever, eliminating the possibility of Patroni ever electing the new master.

If you are looking for a completely separate patroni cluster running in a standby node until switched manually. However, it will not work if you have only 2 nodes, for the same reasons as above.





< > ↺ nature.com/articles/s41586-024-07566-y#MOESM1

nature > articles > article

Article | [Open access](#) | Published: 24 July 2024

# AI models collapse when trained on recursively generated data

[Ilia Shumailov](#) ✉, [Zakhar Shumaylov](#) ✉, [Yiren Zhao](#), [Nicolas Papernot](#), [Ross Anderson](#) & [Yarin Gal](#) ✉

[Nature](#) **631**, 755–759 (2024) | [Cite this article](#)

**429k** Accesses | **3157** Altmetric | [Metrics](#)

**i** An [Author Correction](#) to this article was published on 21 March 2025

**Download PDF** ↓

## Associated content

Nature Outlook

### Robotics and artificial intelligence

### AI produces gibberish when trained on too much AI-generated data

Emily Wenger  
**Nature** | **News & Views** | 24 Jul 2024

# DATA Cannibalism

## Computer Science &gt; Machine Learning

[Submitted on 26 Oct 2022 (v1), last revised 4 Jun 2024 (this version, v2)]

## Will we run out of data? Limits of LLM scaling based on human-generated data

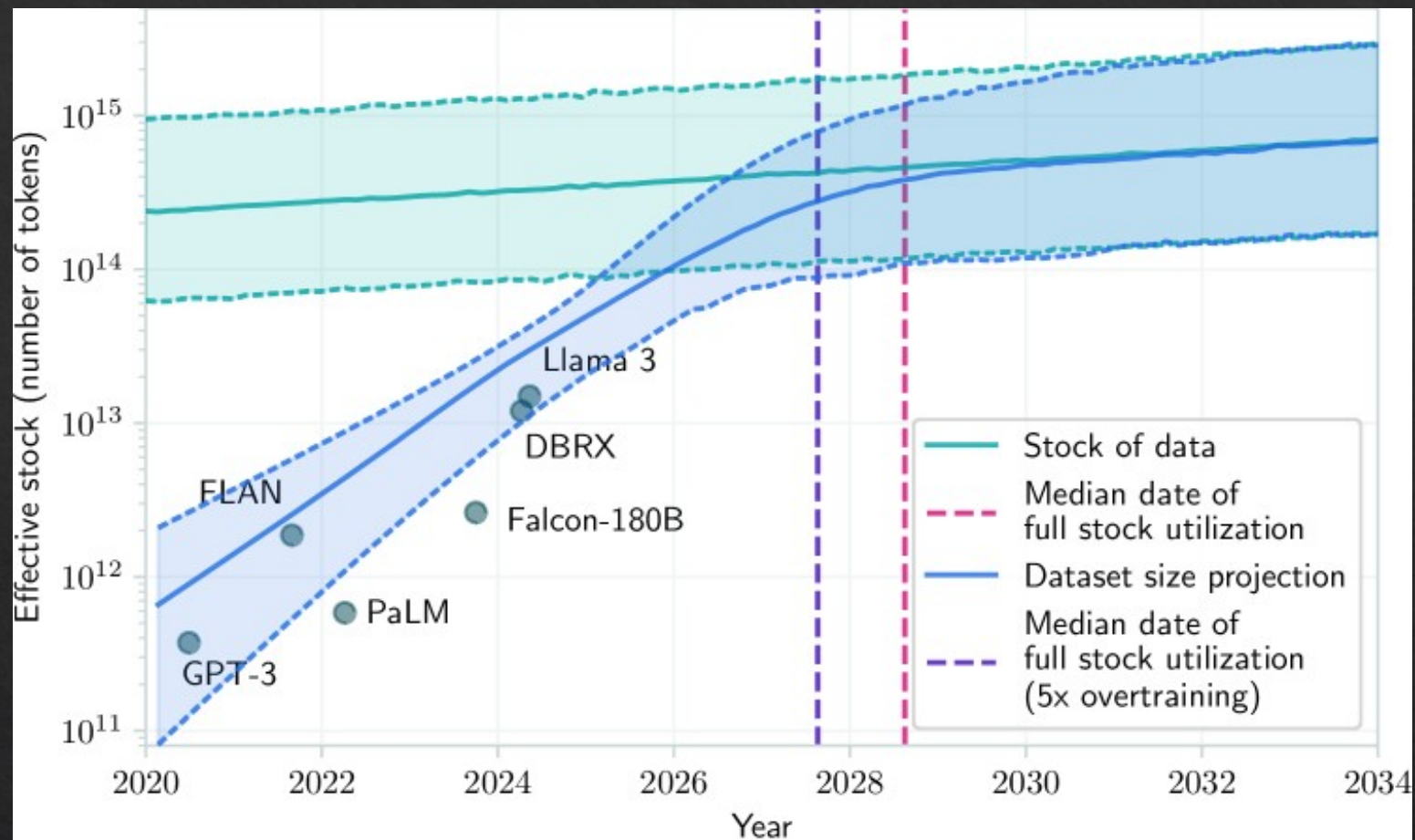
Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, Marius Hobbhahn

We investigate the potential constraints on LLM scaling posed by the availability of public human-generated text data. We forecast the growing demand for training data based on current trends and estimate the total stock of public human text data. Our findings indicate that if current LLM development trends continue, models will be trained on datasets roughly equal in size to the available stock of public human text data between 2026 and 2032, or slightly earlier if models are overtrained. We explore how progress in language modeling can continue when human-generated text datasets cannot be scaled any further. We argue that synthetic data generation, transfer learning from data-rich domains, and data efficiency improvements might support further progress.

Subjects: **Machine Learning (cs.LG)**; Artificial Intelligence (cs.AI); Computation and Language (cs.CL); Computer Vision and Pattern Recognition (cs.CV); Computers and Society (cs.CY)

Cite as: [arXiv:2211.04325](https://arxiv.org/abs/2211.04325) [cs.LG]

# DATA Cannibalism



# DATA Cannibalism



mdpi.com

## A Performance Benchmark for the PostgreSQL and MySQL Databases

by [redacted] and [redacted]

Department of Electrical and Computer Engineering, [redacted] Canada  
\* Author to whom correspondence should be addressed.

*Future Internet* **2024**, *16*(10), 382; <https://doi.org/10.3390/fi16100382>

Submission received: 30 September 2024 / Revised: 16 October 2024 / Accepted: 18 October 2024 / Published: 19 October 2024

(This article belongs to the Special Issue Distributed Storage of Large Knowledge Graphs with Mobility Data)

[Download](#) [Browse Figures](#) [Versions Notes](#)

mdpi.com 1999-5903/16/10/382

2 [Download PDF](#)

### 1. Introduction

In the digital world, latency is the new outage. Simply put, latency is the time required to perform any action or operation. For example, when a user takes time to display all related results. The time difference between the request and the response is called latency. It is essential to study the latency of a system, as it has a direct impact on the continuous user authentication, reducing latency is critical as the user latency can create an opportunity for hackers. As a rule of thumb,

mdpi.com/1999-5903

**MDPI**

PostgreSQL is derived from the POSTGRES package developed at the University of California at Berkeley [14]. It was led by Professor Stockbroker and sponsored by the Defense Advanced Research Projects Agency (DARPA). POSTGRES has undergone many releases since 1987. In 1994 [15], Andrew Yu and Jolly Chen added the SQL language to POSTGRES

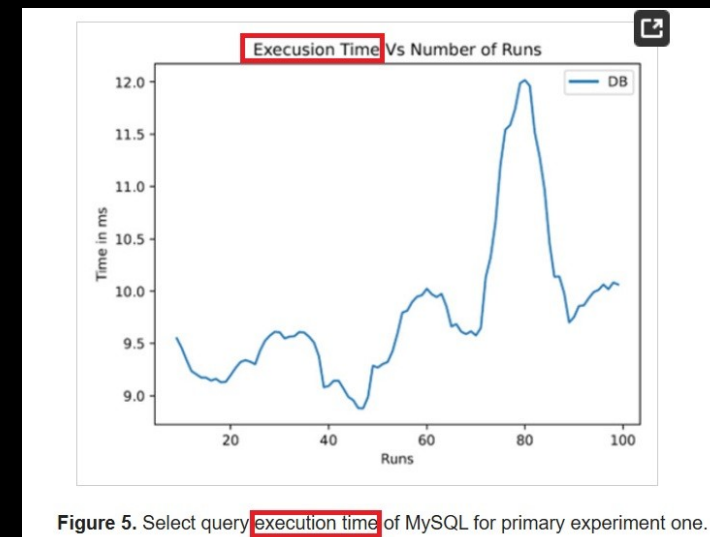


Figure 5. Select query execution time of MySQL for primary experiment one.

DATA CANNIBALISM

**NOM NOM NOM NOM**



**.....WAIT!**

imgflip.com

# What AI Can do for YOU?



DB Parameter tuning ( DBTune)

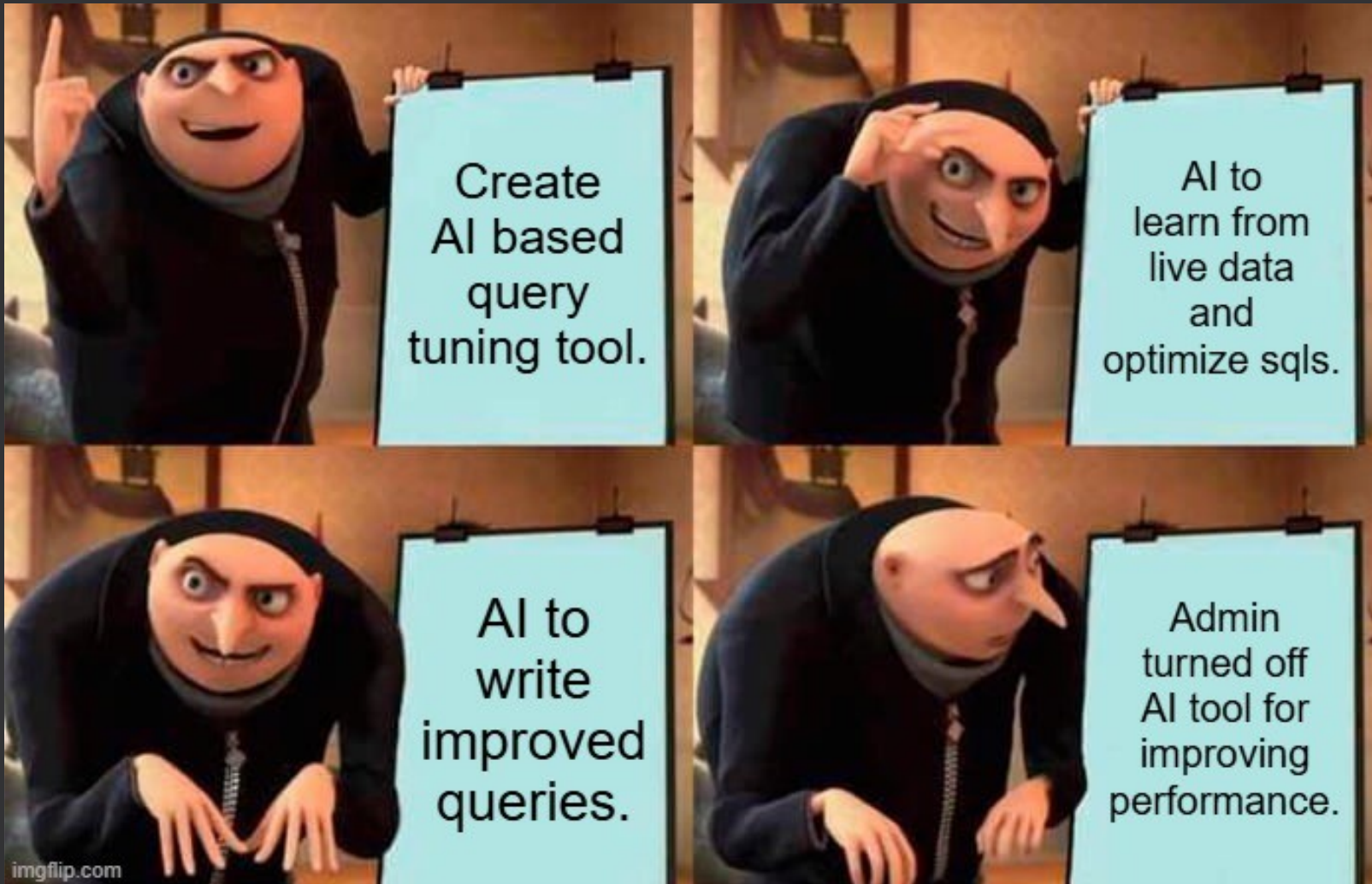


Auto-Indexing  
(PGAnalyze)




Ops Automation (AWS Devops Guru, dba.ai, Xata agent)





# Generative AI is not replacing jobs or hurting wages at all, economists claim

'When we look at the outcomes, it really has not moved the needle'

 [Thomas Claburn](#)

Tue 29 Apr 2025 // 07:18 UTC

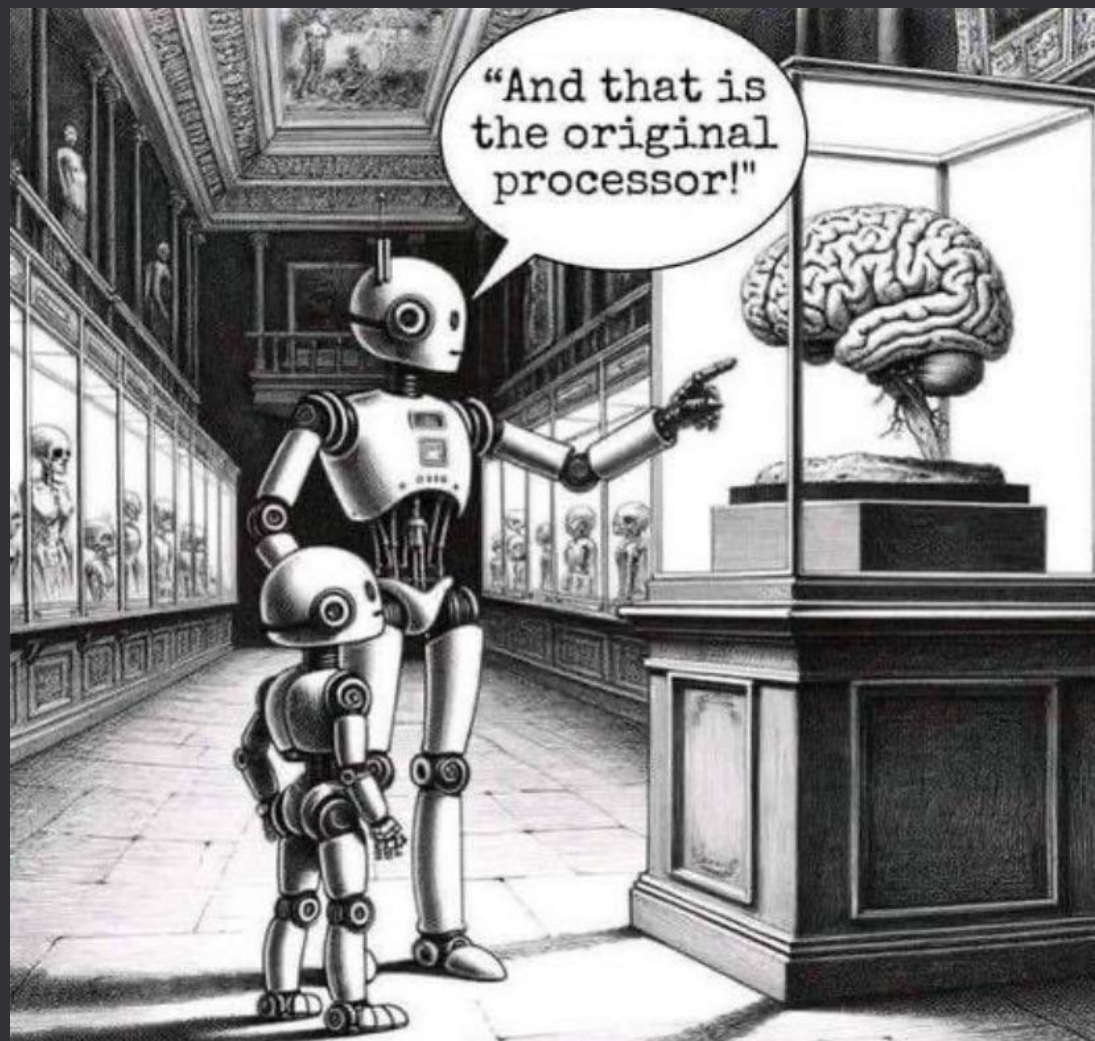
Instead of depressing wages or taking jobs, generative AI chatbots like ChatGPT, Claude, and Gemini have had almost no significant wage or labor impact so far – a finding that calls into question the huge capital expenditures required to create and run AI models.

In [a working paper](#) released earlier this month, economists Anders Humlum and Emilie Vestergaard looked at the labor market impact of AI chatbots on 11 occupations, covering 25,000 workers and 7,000 workplaces in Denmark in 2023 and 2024.

Many of these occupations have been described as being vulnerable to AI: accountants, customer support specialists, financial advisors, HR professionals, IT support specialists, journalists, legal professionals, marketing professionals, office clerks, software developers, and teachers.



## Future of AI : Expectation



## Future of AI : Reality



# References



Postgres Slack (Unofficial) <https://pgtreats.info/slack-invite>



Postgres Telegram (Unofficial) : <https://t.me/postgreschat> & [https://t.me/pg\\_sql](https://t.me/pg_sql)



[AI Hallucinations - Technology Review Article](#)



[AI Model Collapse Paper, Nature](#)



[AI running out of Human data](#)



Thank you

