← GO BACK TO BLOG

SPLICING DUCK AND ELEPHANT DNA

2024/08/15

BY JORDAN TIGANI AND BRETT GRIFFIN

Share in

CONTENT

INTRODUCING THE DUCKDB + POSTGRES EXTENSION

You can have your analytics and transact them too

We're excited to announce **pg_duckdb**, an open-source Postgres extension that embeds DuckDB's analytics engine into Postgres for fast analytical queries in your favorite transactional database.

Postgres is generating a lot of excitement, having been named **2023 DBMS of the Year** by DB-Engines and recognized as the most popular database in the **2024 Stack Overflow** understand what is going on in a data set, it can be painfully slow. For example, you might want to know how revenue is growing in the Netherlands, or how many of your customers have names that rhyme with “Duck.” These are analytical queries and often require separate ways of storing and processing the data to operate efficiently.

People have tried to add Band-Aids to improve Postgres analytical performance. but they haven’t been particularly successful because being good at analytics requires different techniques for running your queries, like being able to operate over batches of rows at once, and avoiding decompressing data until it is absolutely needed. And typically, that takes a purpose-built analytical engine, which takes a ton of effort.

This is where DuckDB comes in. DuckDB is an in-process OLAP database and uses a vectorized query engine to process chunks (vectors) of data at a time. This makes it valuable for answering analytical questions about what is going on in the data. DuckDB’s popularity has been soaring due to its speed, ease of use, and versatility.

Postgres has a rich extension model that lets you do things like search over vector embeddings and handle geospatial data. DuckDB is an embedded database so you can build it into other software. What happens if you put those two together? Can you make a terrific transactional database that can also do awesome analytics?

Today, we’re announcing our collaboration on `pg_duckdb`, a Postgres extension that combines Postgres and DuckDB. It is fully open source, with a permissive MIT license. What’s more, the IP is owned by the DuckDB foundation, which will ensure that it stays open source. It is hosted in the official DuckDB GitHub **repository**.

THE CHALLENGES AHEAD

In order to really make a DuckDB Postgres extension that looks and feels just like Postgres, it is going to take a lot of work to get right. It is going to need significant DuckDB experience, since it will need improvements to DuckDB. In addition, it will also require a lot of Postgres knowledge to figure out how to weave DuckDB seamlessly into how Postgres executes queries.

In order to gather the right experts, we helped put together a consortium of companies, each of whom can provide unique skills to make the project successful:

extensions and storage. They are key drivers and contributors to the project.

- **Neon** has been building serverless managed Postgres and is lending experience about what will run well in production and how to make DuckDB work with Postgres Storage
- **Microsoft** has a ton of Postgres know-how including several Postgres committers and are also participating in the project.

> “A lot of developers use Postgres as a general purpose database and analytics is a major use case that Postgres didn't address well until now. This will be a big win for our users and generally for the Postgres ecosystem to support columnstore data and run analytics well. We are excited to add this extension to our platform and also contribute to this project." -- Nikita Shamgunov, CEO and founder of Neon DB

We recognize that we aren’t the first people with this idea; in fact, there have been several other folks who have built DuckDB as a Postgres extension. Crunchy Data has a commercial version. ParadeDB built `pg_analytics` which has similar functionality, but has a somewhat more restrictive license. But we realized that those projects, on their own, are going to struggle to be successful without commitment to do the internal engine work in DuckDB. By building in the open and making sure that DuckDB can operate seamlessly in a Postgres environment, we believe that we will be helping these projects as well.

WHY, YOU MIGHT ASK, DOES MOTHERDUCK CARE ABOUT POSTGRES?

After all, isn’t MotherDuck a cloud hosted DuckDB?

First, we are committed to a thriving DuckDB ecosystem. If DuckDB becomes ubiquitous, then that is good for everyone. We want to see DuckDB in as many different places and applications as possible. And Postgres has millions of users; if a healthy proportion of those people starts becoming familiar with DuckDB, that is a win for duck fans everywhere.

Second, our motto at MotherDuck is, “If you can Duck, you can MotherDuck.” Our aim is to ensure that anywhere you can run DuckDB, running MotherDuck is as simple as opening a database with the `md:` prefix. MotherDuck allows any DuckDB user to scale into the cloud, collaborate with colleagues, and reliably manage their data.

Moreover, it is common in analytics to want to offload your data from your transactional database into an analytical store. The pg_duckdb extension along with MotherDuck can help; you can just run a query in Postgres that pulls recent data from your Postgres database and write it to MotherDuck. You don't need to export and reimport data, or set up CDC.

Finally, there are some downsides to running analytics on the same database that runs your application. Analytics can be resource hungry in terms of the amount of memory and CPU needed to make it run well. Above a certain size, folks may not want to run this on their production transactional database. MotherDuck will help offload this to the cloud, in a way that people don't even have to change the queries that they're running; they just get faster.

BUILDING IN THE OPEN

We're announcing early, with the intention of building in the open with a public roadmap. The pg_duckdb extension is fully usable to query over data in a data lake, to run analytical queries over Postgres, and to store data in a local DuckDB database.

Today at **DuckCon 5**, Joe Sciarrino from **Hydra** showed off the extension and some of its capabilities, and Frances Perry from MotherDuck demonstrated pg_duckdb running queries combining Postgres and MotherDuck. If you didn't make it to that event, you'll be able to check out the videos once they're posted.

Key features in the roadmap include:

- Seamless MotherDuck support to be able to access your MotherDuck data in the cloud and your Postgres data at the same time.
- Postgres native storage that will write data into Postgres storage pages and write-ahead log, which will let pg_duckdb to integrate with existing backup and replication.
- Full type compatibility with Postgres. Postgres already supports a lot of data types, our goal is to support them all.
- Full function compatibility with Postgres; any Postgres function that you run should also work in DuckDB.
- Seamless semantic compatibility. There are subtle differences between how any two database engines compute results, even ones that support the same SQL operations. Things like how to handle rounding or decimals of certain precision, or how to deal with

Check out the **repository** today. We are excited to build this in the open and embrace contributions, feedback, and suggestions from everybody. As they say, “if you want to go far, go together.” We recognize that there are a lot of technical challenges ahead, and we welcome help and guidance on the project.

Also, please share your feedback with us on the MotherDuck **Slack**! If you’d like to discuss your use case in more detail, please connect with us - we’d love to learn more about what you’re building.

NEXT POSTS

VIEW ALL