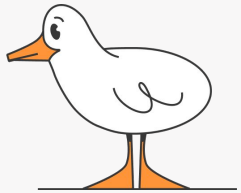
MotherDuck

PG_DUCKDB: DUCKING AWESOME ANALYTICS IN POSTGRES

Jelte Fennema-Nio (@JelteF)

2025-06-11



What is **pg_duckdb**?

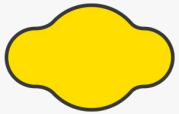
pg_duckdb is a Postgres extension that embeds DuckDB inside Postgres

Ehhhmm what???

MotherDuck

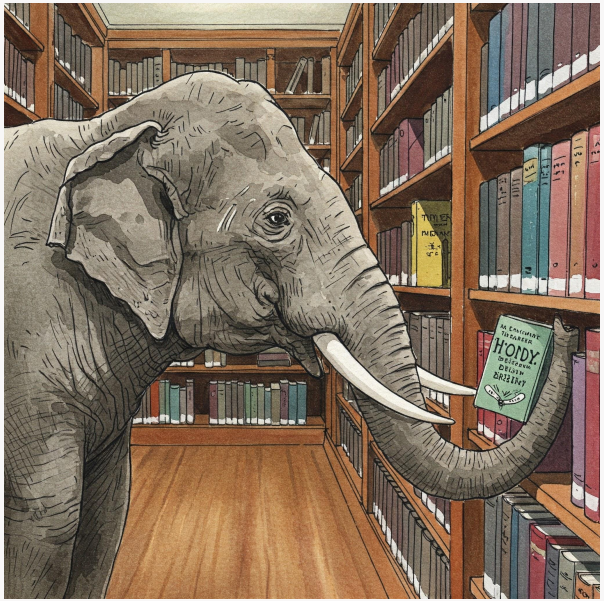
Postgres is an amazing database

- Open source
- Many contributors
- Very stable
- Lots of built in functionality and extensible



MotherDuck

Great at transactional workloads (OLTP)



MotherDuck

But not at analytics… (OLAP)

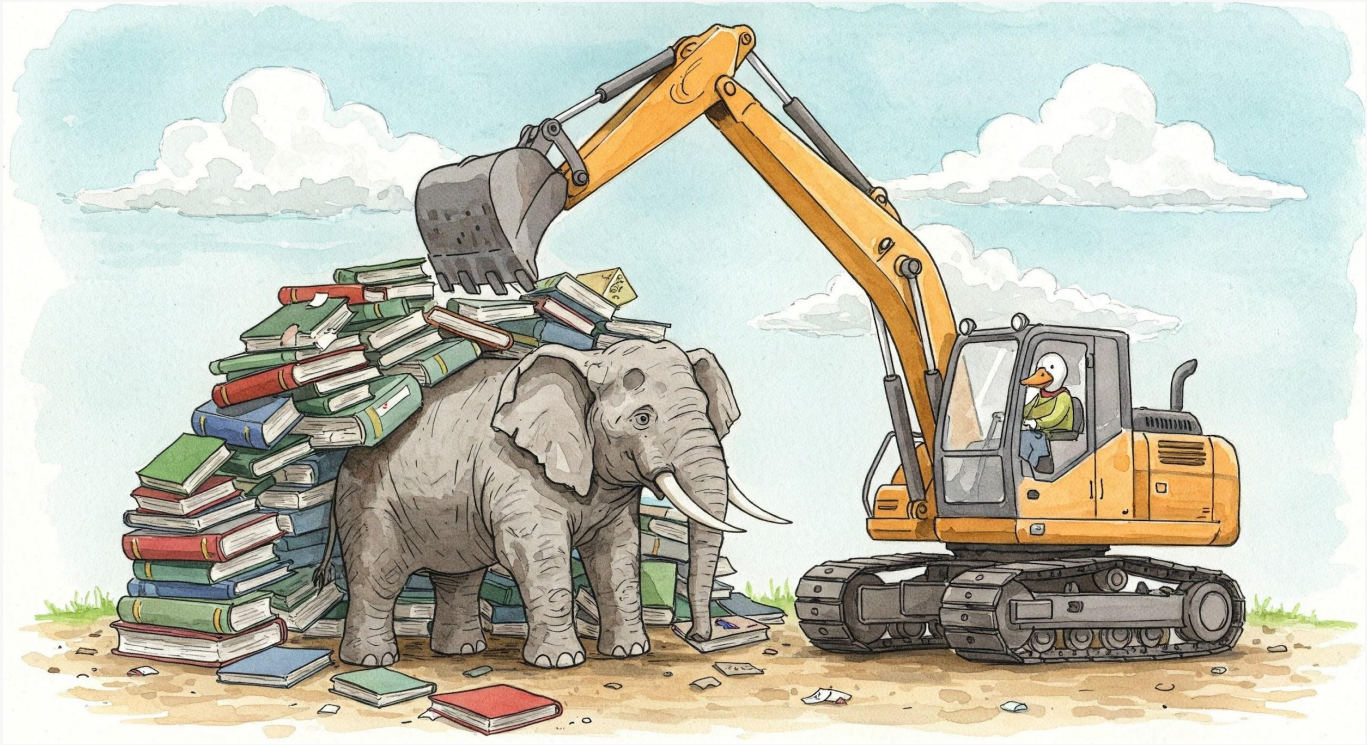
MotherDuck

Why not great at OLAP?

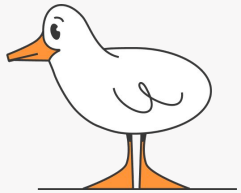
- No columnar storage
- Limited parallel processing

MotherDuck

DuckDB to the rescue

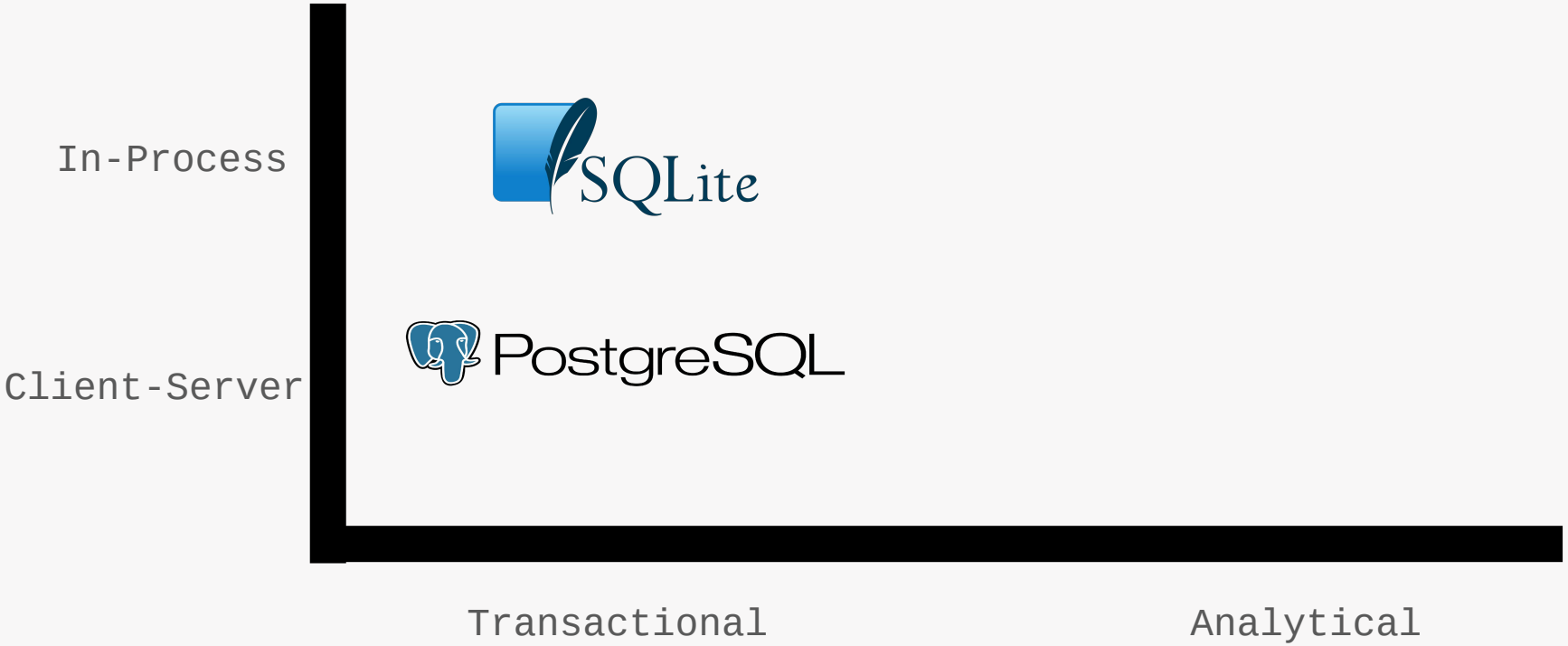


What is **DuckDB**?

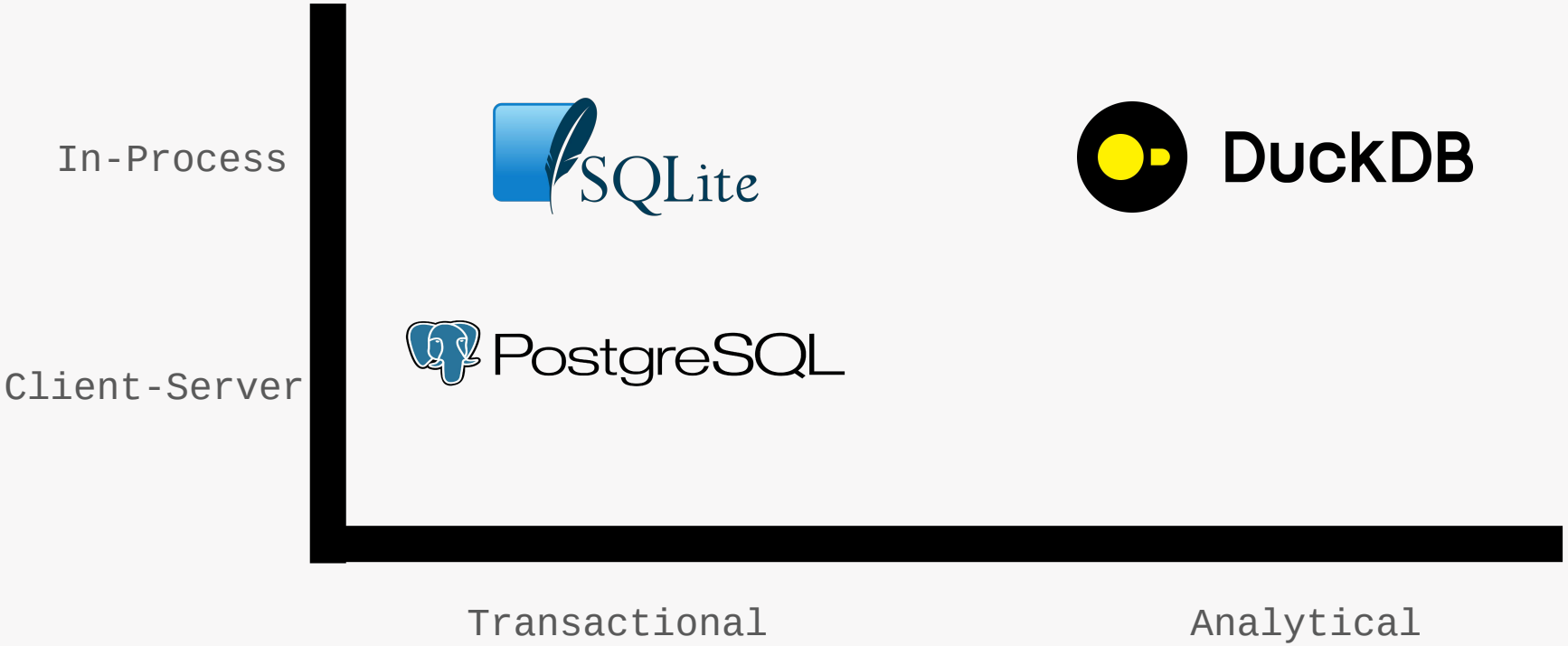


Lightweight in-process SQL Analytics Engine

DuckDB is a new category of database



DuckDB is a new category of database

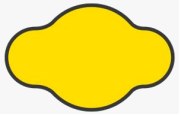


MotherDuck

A small recap

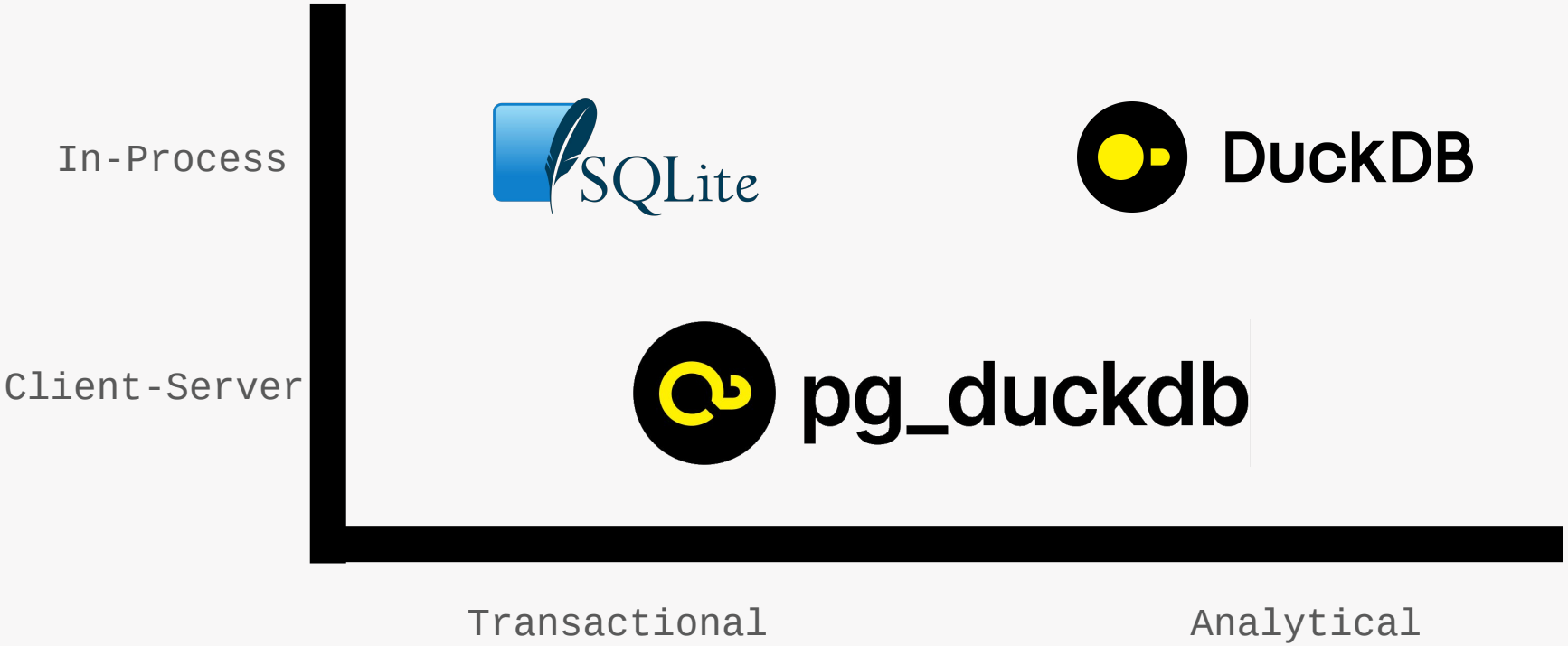
Now, we have two great databases

- Postgres for transactional workloads
- DuckDB for analytical workloads

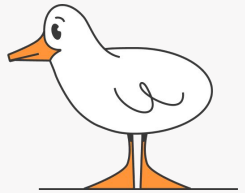


PG Analytics with DuckDB

PG Analytics with DuckDB

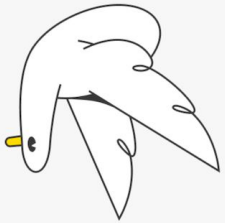


What does that look like?



What does that look like?

What does that look like?



Ducks & Elephants are different species

MotherDuck

C	vs	C++
elog(ERROR, …)	vs	exceptions
processes	vs	threads

MotherDuck

So we did lots of work

How does it work?

MotherDuck

Transforms the SQL query string to syntax tree

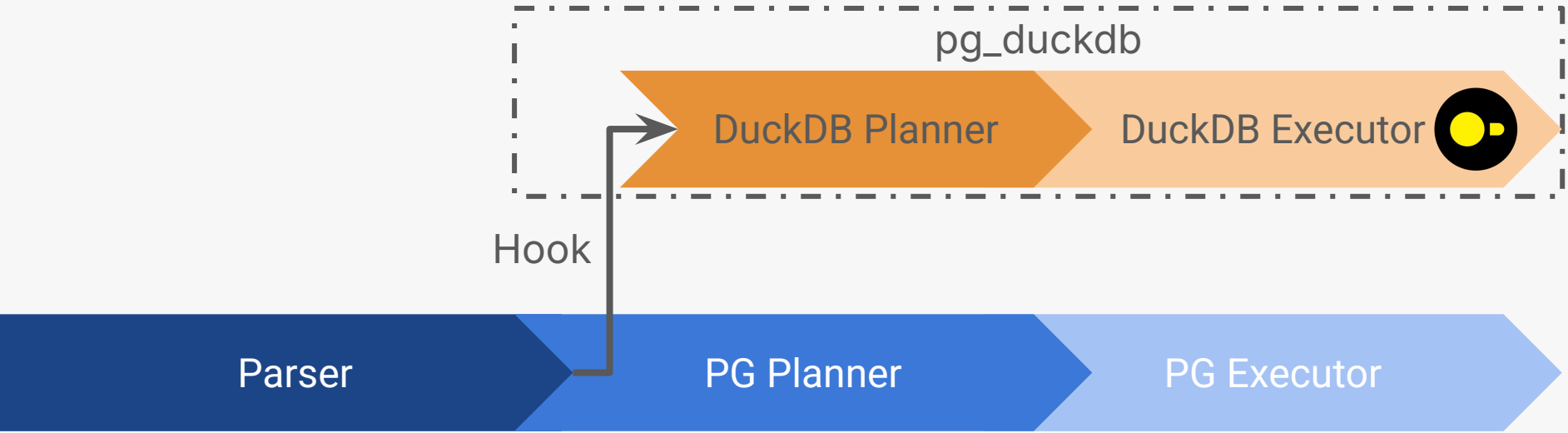
Determines the most efficient way to execute it

Executes the plan

Simplified query processing in Postgres

pg_duckdb “steals” the query

MotherDuck



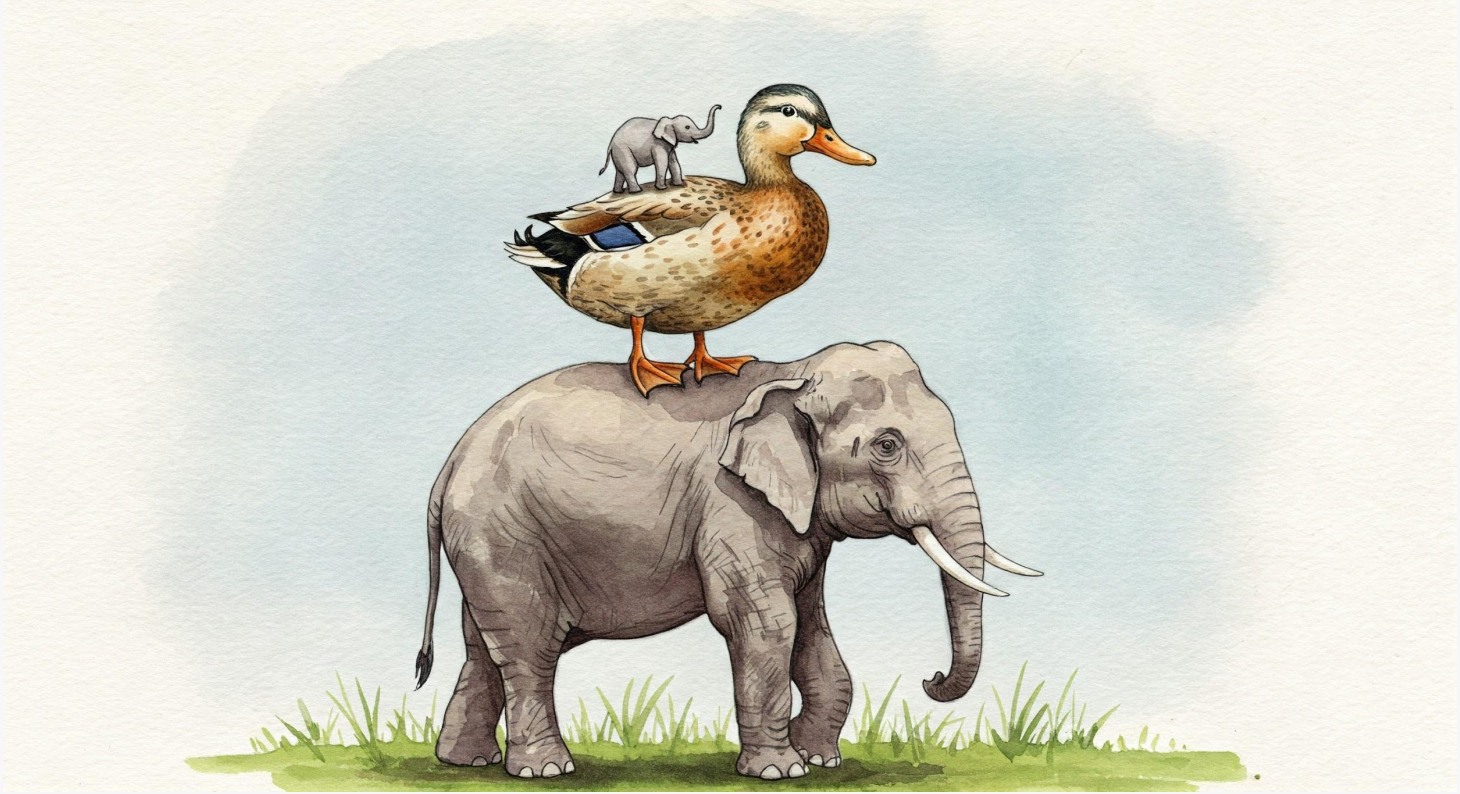
Transforms the SQL query string to syntax tree

Determines the most efficient way to execute it

Executes the plan

pg_duckdb can read PG data

MotherDuck



What can it do?

What can it do?

1. Use DuckDB engine on Postgres tables

What can it do?

1. Use DuckDB engine on Postgres tables
2. Read/write data in blob storage

What can it do?

1. Use DuckDB engine on Postgres tables
2. Read/write data in blob storage
3. Offload analytics to MotherDuck

DuckDB engine on Postgres tables

DuckDB engine on Postgres tables

Very simple:

```
SET duckdb.force_execution = true;
```

But is it fast???

It depends…

But sometimes yes!

ClickBench results

	PostgreSQL (with indexes) (c6a.4xlarge, 500gb gp2)	pg_duckdb (with indexes) (c6a.4xlarge, 500gb gp2)
Q10.	**2.053s (×1.00)**	2.382s (×1.16)
Q11.	735.659s (×2.08)	**353.140s (×1.00)**
Q12.	**2.207s (×1.00)**	3.453s (×1.56)
Q13.	9.281s (×2.03)	**4.557s (×1.00)**
Q14.	255.972s (×1.03)	248.691s (×1.00)

One extreme example

One extreme example

1. **Set up TPC-DS with 10GB and no indexes**

One extreme example

1. Set up TPC-DS with 10GB and no indexes
2. Run Q1 -> ⌛⌛⌛ wait 10 minutes and give up

One extreme example

1. Set up TPC-DS with 10GB and no indexes
2. Run Q1 -> ⌛⌛⌛ wait 10 minutes and give up
3. SET duckdb.force_execution = true;

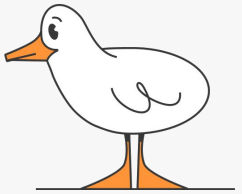
One extreme example

1. Set up TPC-DS with 10GB and no indexes
2. Run Q1 -> ⌛⌛⌛ wait 10 minutes and give up
3. SET duckdb.force_execution = true;
4. Run Q1 -> done in 450ms!

One extreme example

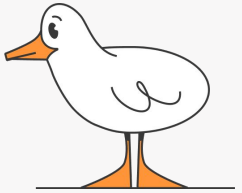
1. Set up TPC-DS with 10GB and no indexes
2. Run Q1 -> ⌛⌛⌛ wait 10 minutes and give up
3. SET duckdb.force_execution = true;
4. Run Q1 -> done in 450ms!
5. Easiest query optimization ever 🎉

Read from blob storage



```
SELECT *
FROM read_parquet('s3://<my-bucket>/netflix_daily_top_10.parquet')
LIMIT 5;
```

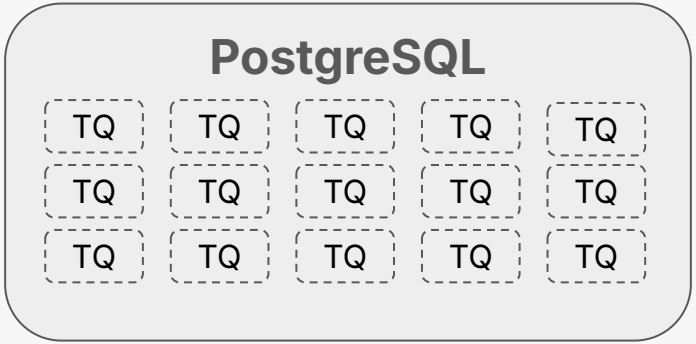
Read from blob storage



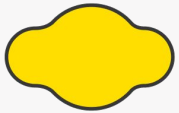
```
SELECT r['Title'], max(r['Days In Top 10'])::int as MaxDaysInTop10
FROM read_parquet('s3://<my-bucket>/netflix_daily_top_10.parquet') r
WHERE r['Type'] = 'TV Show'
GROUP BY r['Title']
ORDER BY MaxDaysInTop10 DESC
LIMIT 5;
```

MotherDuck

A few words about resources

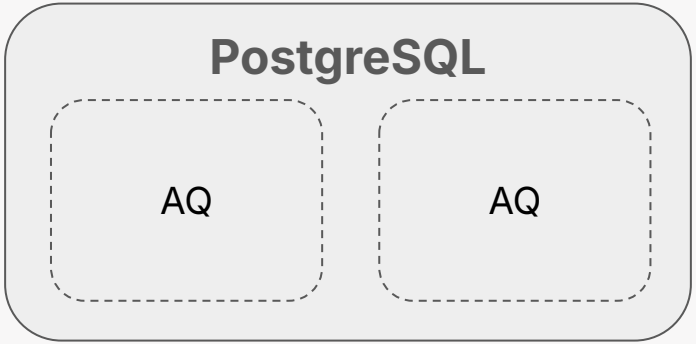


Lite transactional queries



MotherDuck

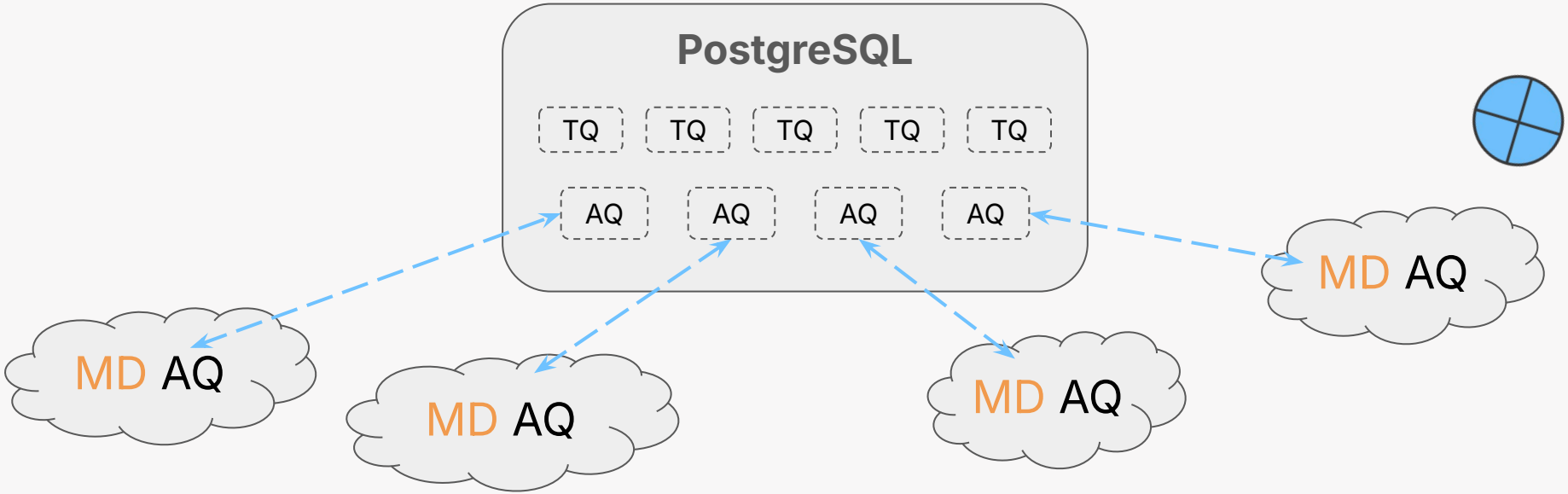
A few words about resources



Analytics

MotherDuck

A few words about resources



MotherDuck on-demand resources

Copy data to MotherDuck

```
CREATE TABLE hacker_news_motherduck_archive
USING duckdb AS
SELECT * FROM hacker_news;
```

MotherDuck

Query it like normal

```
SELECT
    EXTRACT(YEAR FROM timestamp) AS year,
    EXTRACT(MONTH FROM timestamp) AS month,
    COUNT(*) AS keyword_mentions
FROM hacker_news_motherduck_archive
WHERE
    (title LIKE '%duckdb%' OR text LIKE '%duckdb%')
GROUP BY year, month
ORDER BY year ASC, month ASC;
```

MotherDuck

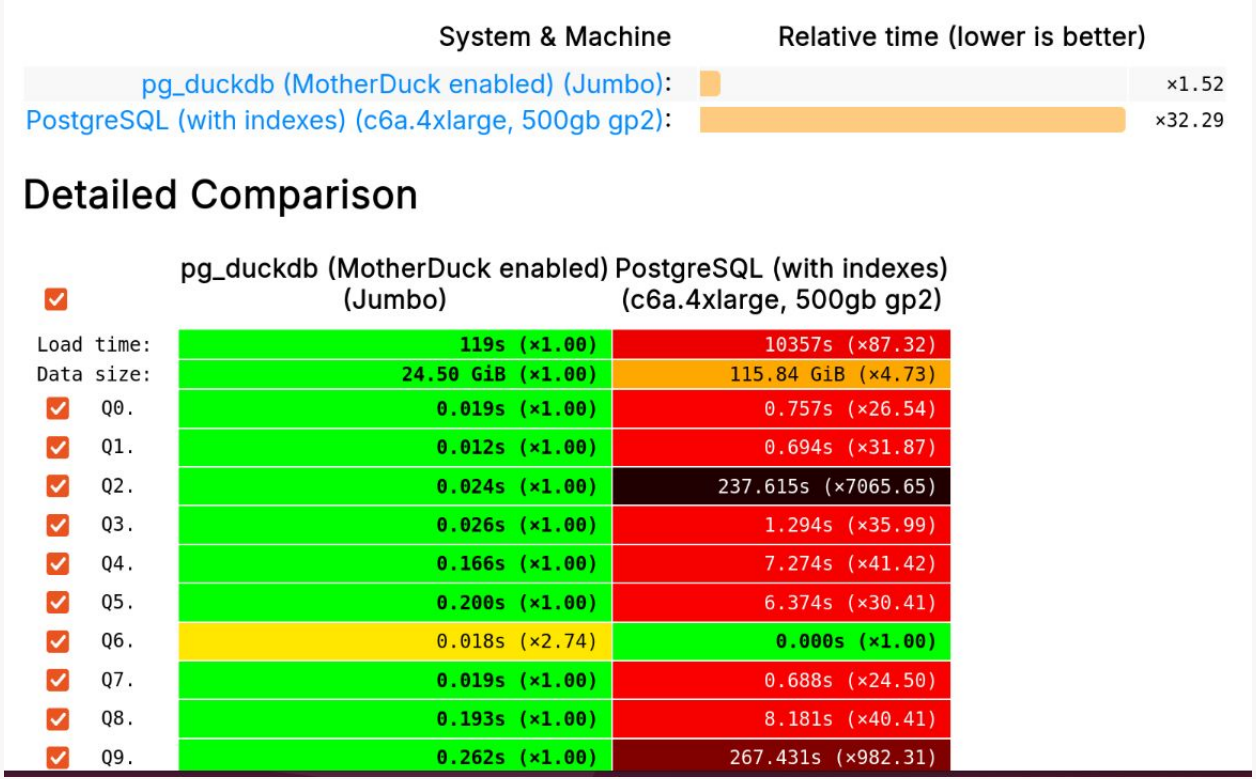
Combine with PG data

```
SELECT
    EXTRACT(YEAR FROM timestamp) AS year,
    EXTRACT(MONTH FROM timestamp) AS month,
    COUNT(*) AS keyword_mentions
FROM (
  SELECT * FROM hacker_news_last_month UNION ALL
  SELECT * FROM hacker_news_motherduck_archive)
WHERE
    (title LIKE '%duckdb%' OR text LIKE '%duckdb%')
GROUP BY year, month
ORDER BY year ASC, month ASC;
```

MotherDuck

But is it fast???

For analytics: YES!



System & Machine	Relative time (lower is better)
pg_duckdb (MotherDuck enabled) (Jumbo):	×1.52
PostgreSQL (with indexes) (c6a.4xlarge, 500gb gp2):	×32.29

Detailed Comparison

	pg_duckdb (MotherDuck enabled) (Jumbo)	PostgreSQL (with indexes) (c6a.4xlarge, 500gb gp2)
Load time:	119s (×1.00)	10357s (×87.32)
Data size:	24.50 GiB (×1.00)	115.84 GiB (×4.73)
Q0.	0.019s (×1.00)	0.757s (×26.54)
Q1.	0.012s (×1.00)	0.694s (×31.87)
Q2.	0.024s (×1.00)	237.615s (×7065.65)
Q3.	0.026s (×1.00)	1.294s (×35.99)
Q4.	0.166s (×1.00)	7.274s (×41.42)
Q5.	0.200s (×1.00)	6.374s (×30.41)
Q6.	0.018s (×2.74)	0.000s (×1.00)
Q7.	0.019s (×1.00)	0.688s (×24.50)
Q8.	0.193s (×1.00)	8.181s (×40.41)
Q9.	0.262s (×1.00)	267.431s (×982.31)

Please try it

- MIT licensed
- github.com/duckdb/pg_duckdb
- Feedback welcome