

OSA CON 24



PG_DUCKDB:

Adding analytics to your application database

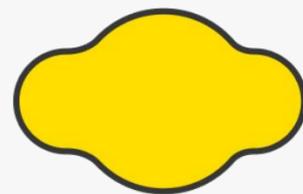
Boaz Leskes

Founding Engineer @ MotherDuck
@bleskes everywhere

November 19-21, 2024

WHO AM I?

- Leading the Database group at MotherDuck
- Cloud Tech Lead @ Elastic
- Distinguished Eng. @ Elastic(search)
- CTO @ Buzzcapture
- ...



Connection to OSS:

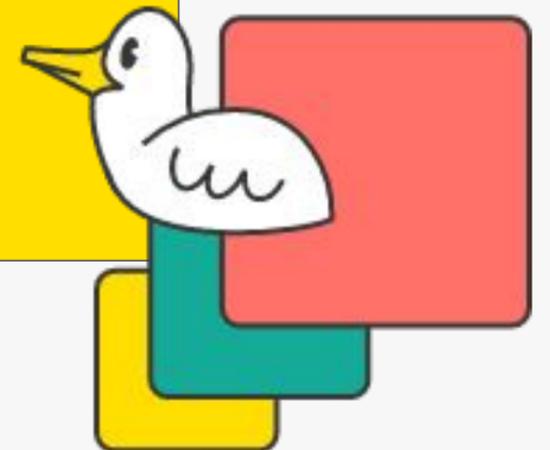
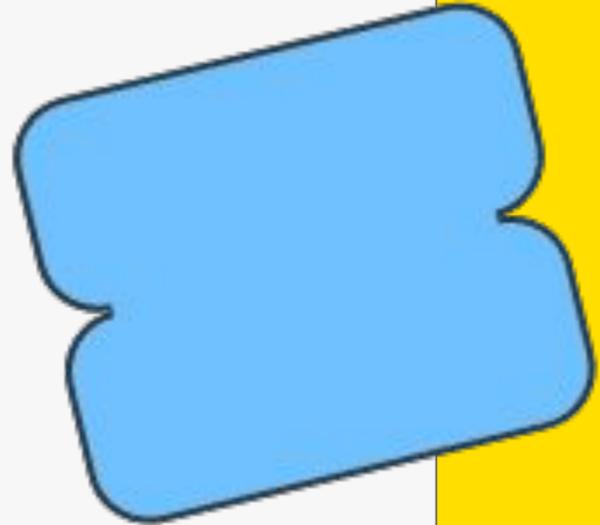


So you need an
application database?

Postgres



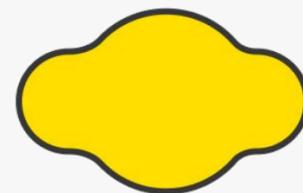
PostgreSQL



What makes Postgres a great database?



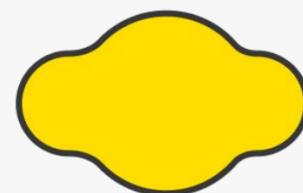
- Open source
- > 25% Fortune 500 companies use it for mission-critical applications
- > 10% annual growth rate in adoption over recent years
- > 300 contributors from all over the world working on each release



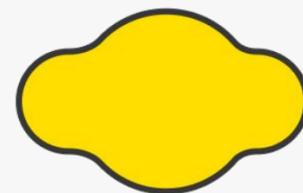
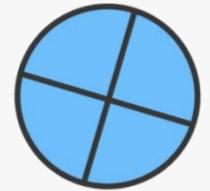
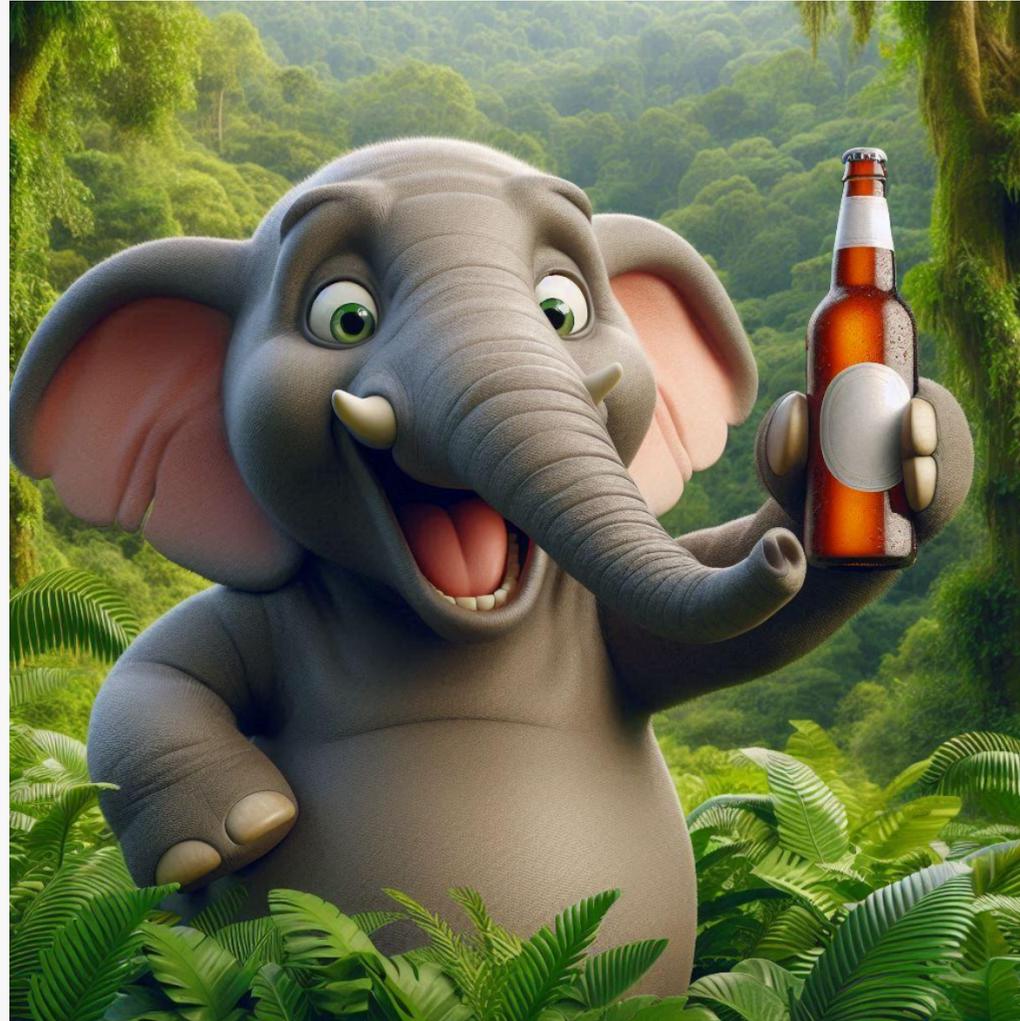
But . . .

It is not a great analytical database:

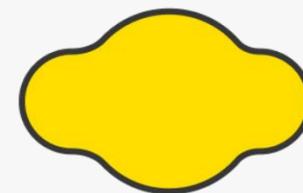
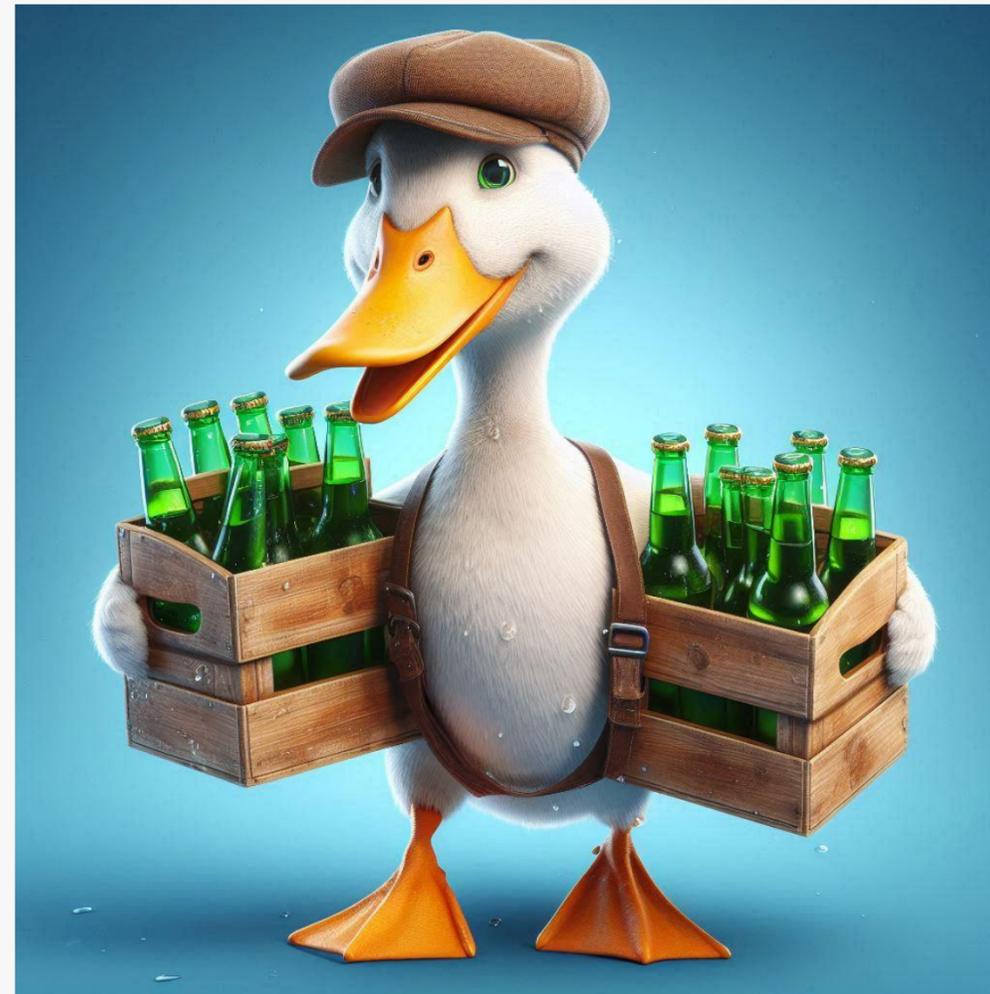
- No columnar storage
- Limited parallel processing
- Million single row queries but not a single query of a millions rows

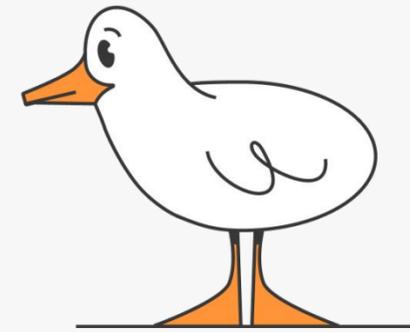


OLTP vs OLAP



OLTP vs OLAP





What is **pg_duckdb**?

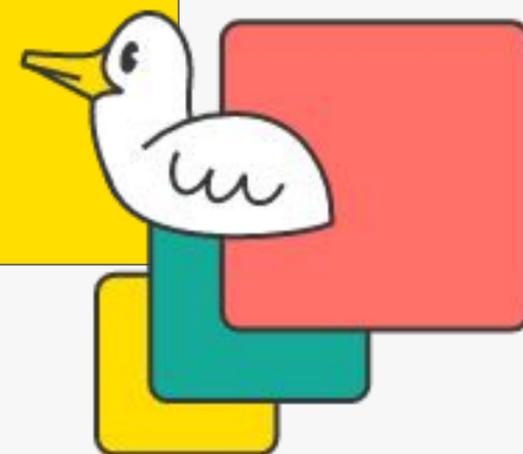
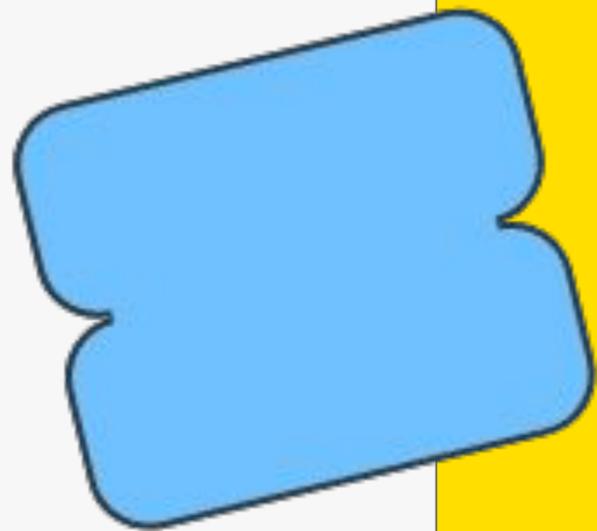
pg_duckdb is a PostgreSQL extension that integrates DuckDB's analytics engine directly into PostgreSQL, allowing for rapid analytical queries alongside traditional transactional workloads.

pg_duckdb integrates well with MotherDuck, for easy Cloud access, BI tools, and scaling of your analytical queries.

Ehhhmm what???



DuckDB



What is DuckDB?



created at:



created by:



maintained by:



What is DuckDB?



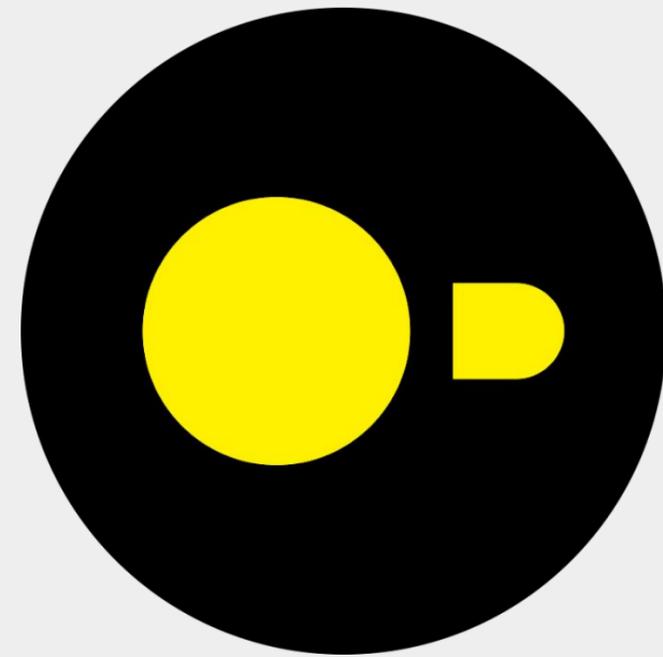
An analytical SQL database

Runs in-process

MIT licensed

Clients in 15+ languages

7M+ downloads / month on PyPI
alone



DuckDB is a new category of database

In-Process



Client-Server



Transactional

Analytical

DuckDB is a new category of database

In-Process



Client-Server



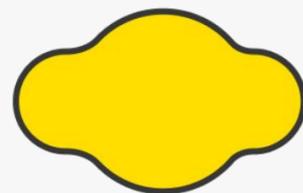
Transactional

Analytical

A small recap

Now, we have two great databases

- Postgres for transactional workloads
- DuckDB for analytical workloads
- How do we combined them together?

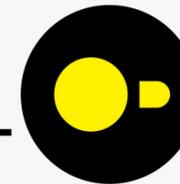


PG Analytics with DuckDB

In-Process



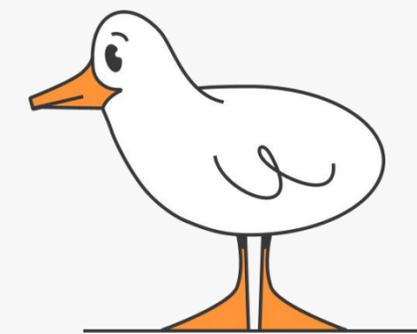
Client-Server



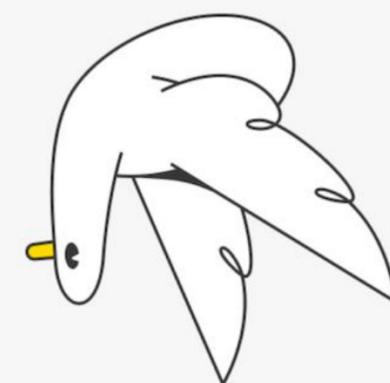
Transactional

Analytical

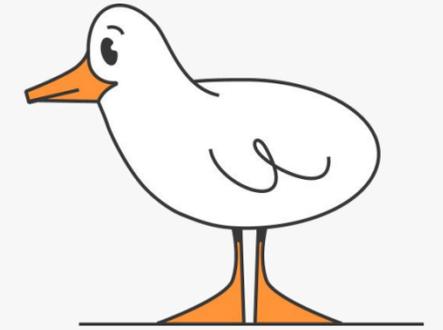
What does that look like?



What does that look like?



PostgreSQL native analytics



```
SELECT Title, max("Days In Top 10")::int as MaxDaysInTop10
FROM
read_parquet('s3://duckdb-md-dataset-121/netflix_daily_top_10.parquet')
  AS ("Days In Top 10" varchar, Title varchar, Type varchar)
WHERE Type = 'TV Show'
GROUP BY Title
ORDER BY MaxDaysInTop10 DESC
LIMIT 5;
```

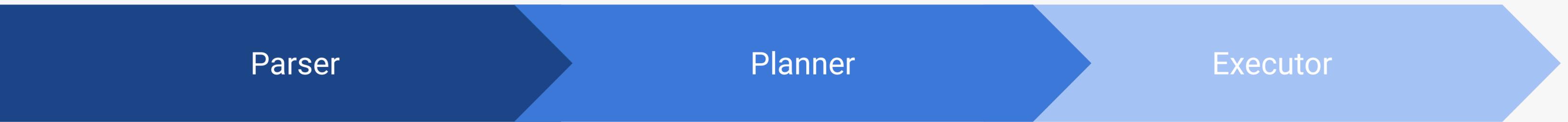
Ducks & Elephants
are different species

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

In other words



How does it work?



Parser

Planner

Executor

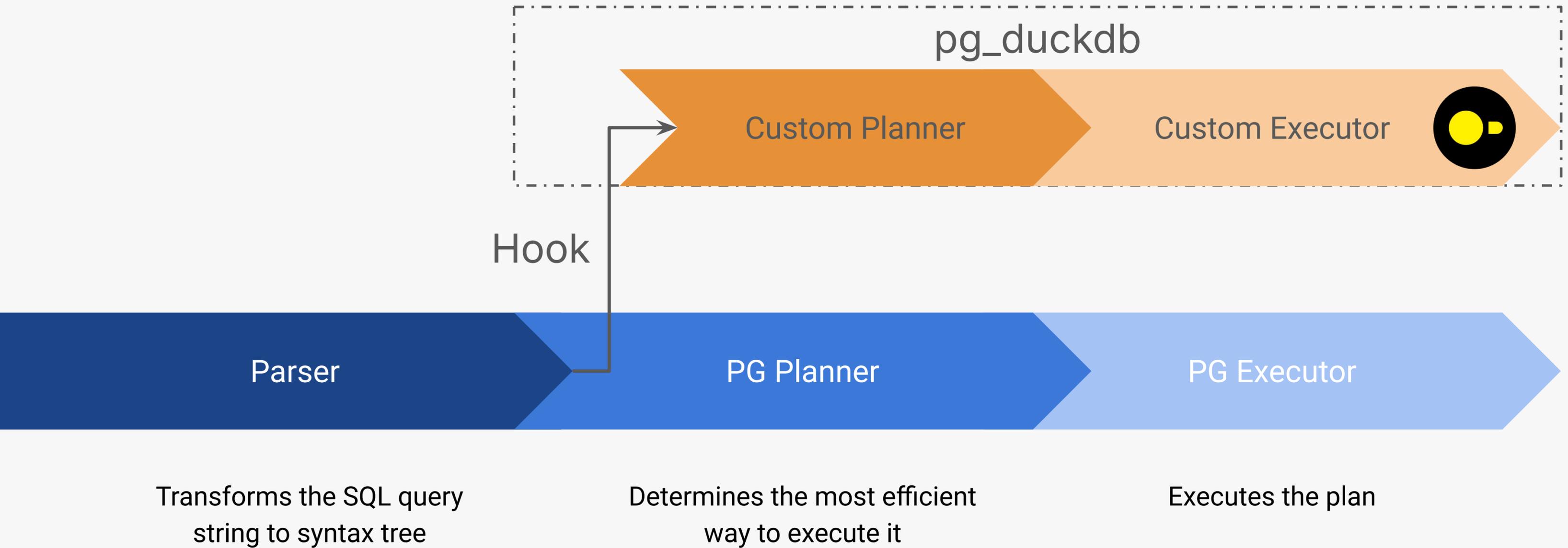
Transforms the SQL query
string to syntax tree

Determines the most efficient
way to execute it

Executes the plan

Simplified query processing in Postgres

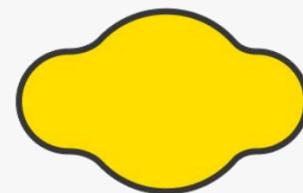
How does it work?



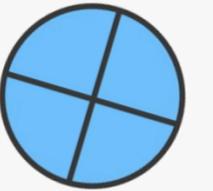
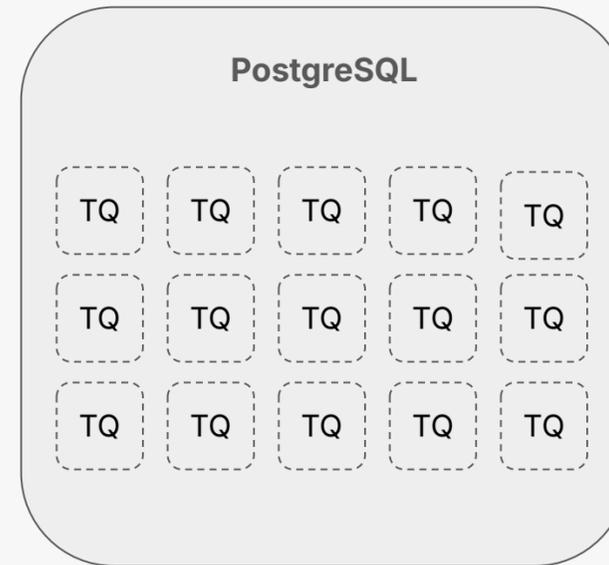
How does it work?



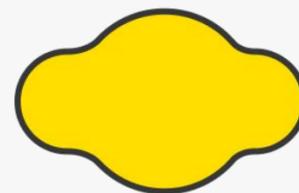
- Under some conditions, `pg_duckdb` “steals” the query
 - if it involves a MotherDuck table
 - if it involves `parquet/csv/json` scanning
 - if `duckdb.force_execution` is set
- Then DuckDB fully executes the query
- But DuckDB is also able to read Postgres’ tables



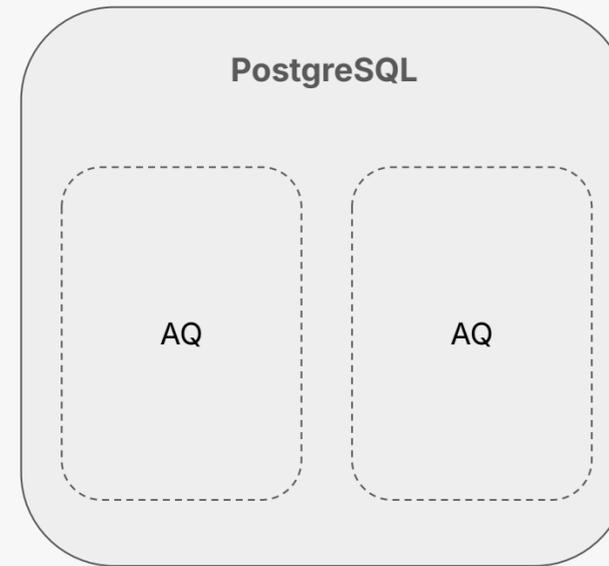
A few words about resources



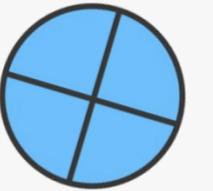
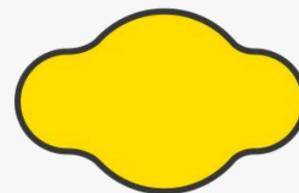
Lite transactional queries



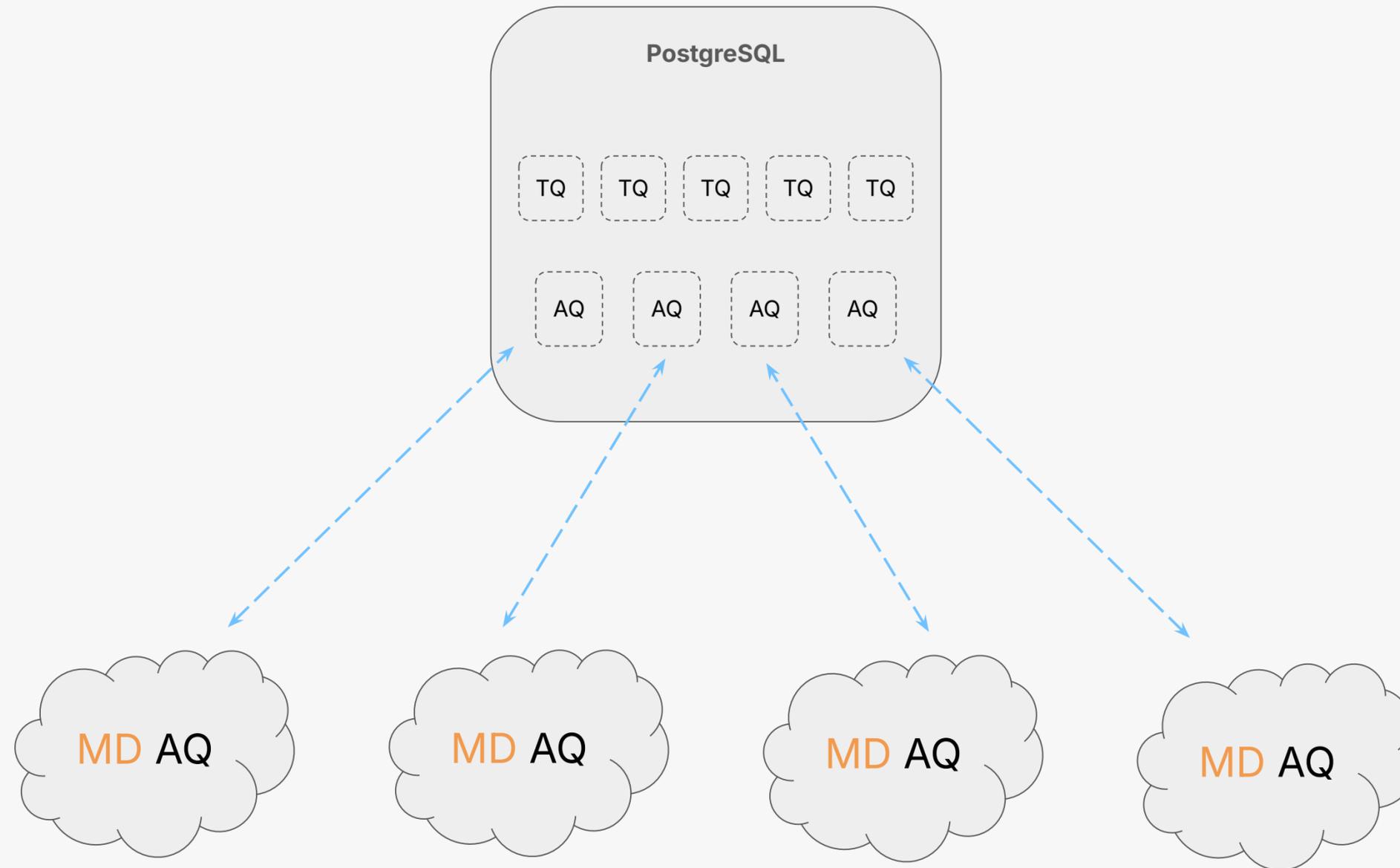
A few words about resources



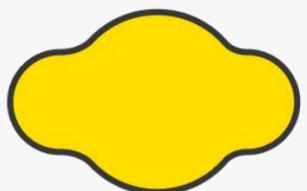
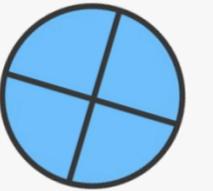
Analytics



A few words about resources



MotherDuck on-demand resources

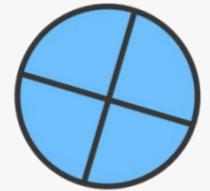




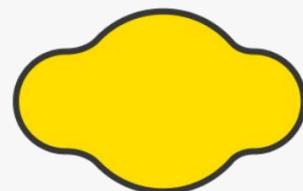
What can it do?

- Read/write parquet & csv from/to blob storage
- Use DuckDB engine on Postgres tables
- TEMP tables in DuckDB columnar format
- Offload analytics to MotherDuck

Analytics on PG data



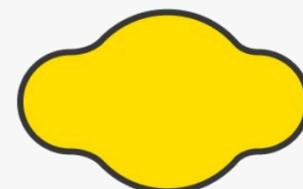
```
SET duckdb.force_execution TO true;
SELECT
  EXTRACT(YEAR FROM timestamp) AS year,
  EXTRACT(MONTH FROM timestamp) AS month,
  COUNT(*) AS keyword_mentions
FROM hacker_news
WHERE
  (title LIKE '%duckdb%' OR text LIKE '%duckdb%')
GROUP BY year, month
ORDER BY year ASC, month ASC;
```



Analytics on PG data



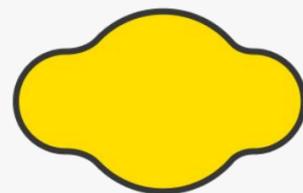
year	month	keyword_mentions
2022	1	6
2022	2	4
2022	3	10
2022	4	9
2022	5	43
2022	6	8
2022	7	15
.....		



Summarize PG data for your BI Tool



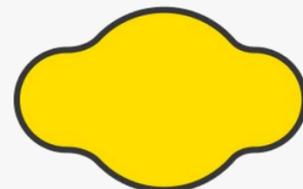
```
CREATE TABLE hacker_news_duckdb_motherduck USING duckdb AS
SELECT
  EXTRACT(YEAR FROM timestamp) AS year,
  EXTRACT(MONTH FROM timestamp) AS month,
  COUNT(*) AS keyword_mentions
FROM hacker_news
WHERE
  (title LIKE '%duckdb%' OR text LIKE '%duckdb%')
GROUP BY year, month
ORDER BY year ASC, month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



Archive in MotherDuck for enrich with PG Data



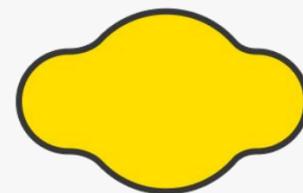
```
SELECT
  EXTRACT(YEAR FROM timestamp) AS year,
  EXTRACT(MONTH FROM timestamp) AS month,
  COUNT(*) AS keyword_mentions
FROM (
  SELECT * FROM hacker_news_last_month UNION ALL
  SELECT * FROM hacker_news_motherduck_archive)
WHERE
  (title LIKE '%duckdb%' OR text LIKE '%duckdb%')
GROUP BY year, month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



More Cloud options



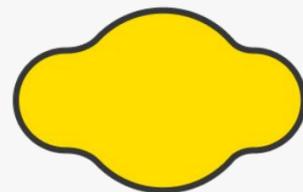
```
SELECT Title, max("Days In Top 10")::int as MaxDaysInTop10
FROM
read_parquet('s3://duckdb-md-dataset-121/netflix_daily_top_10.parquet')
  AS ("Days In Top 10" varchar, Title varchar, Type varchar)
WHERE Type = 'TV Show'
GROUP BY Title
ORDER BY MaxDaysInTop10 DESC
LIMIT 5;
```



Reading an Iceberg table



```
-- Install the iceberg extension
SELECT duckdb.install_extension('iceberg');
-- Total quantity of items ordered for each `l_shipmode`
SELECT l_shipmode, SUM(l_quantity) AS total_quantity
FROM
iceberg_scan('s3://us-prd-motherduck-open-datasets/iceberg/lineitem_ice
berg', allow_moved_paths := true)
  AS I(l_shipmode varchar, l_quantity int)
GROUP BY l_shipmode
ORDER BY total_quantity DESC;
```



But is it fast???



It depends...

But sometimes yes!



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 

Q&A



```
docker exec -it pg_duckdb sh -c 'echo "duckdb.motherduck_token = "<YOUR_MOTHERDUCK_TOKEN>" >> /var/lib/postgresql/data/postgresql.conf'
```

Links

- https://github.com/duckdb/pg_duckdb
- <https://motherduck.com>
- <https://motherduck.com/blog/pgduckdb-beta-release-duckdb-postgres/>