Implementing a Database System with DuckDB for Local Processing and MotherDuck for Scalable Cloud Storage

Jesus L. Monroy · Follow

Published in T3CH · 4 min read · 2 days ago

61 ◯
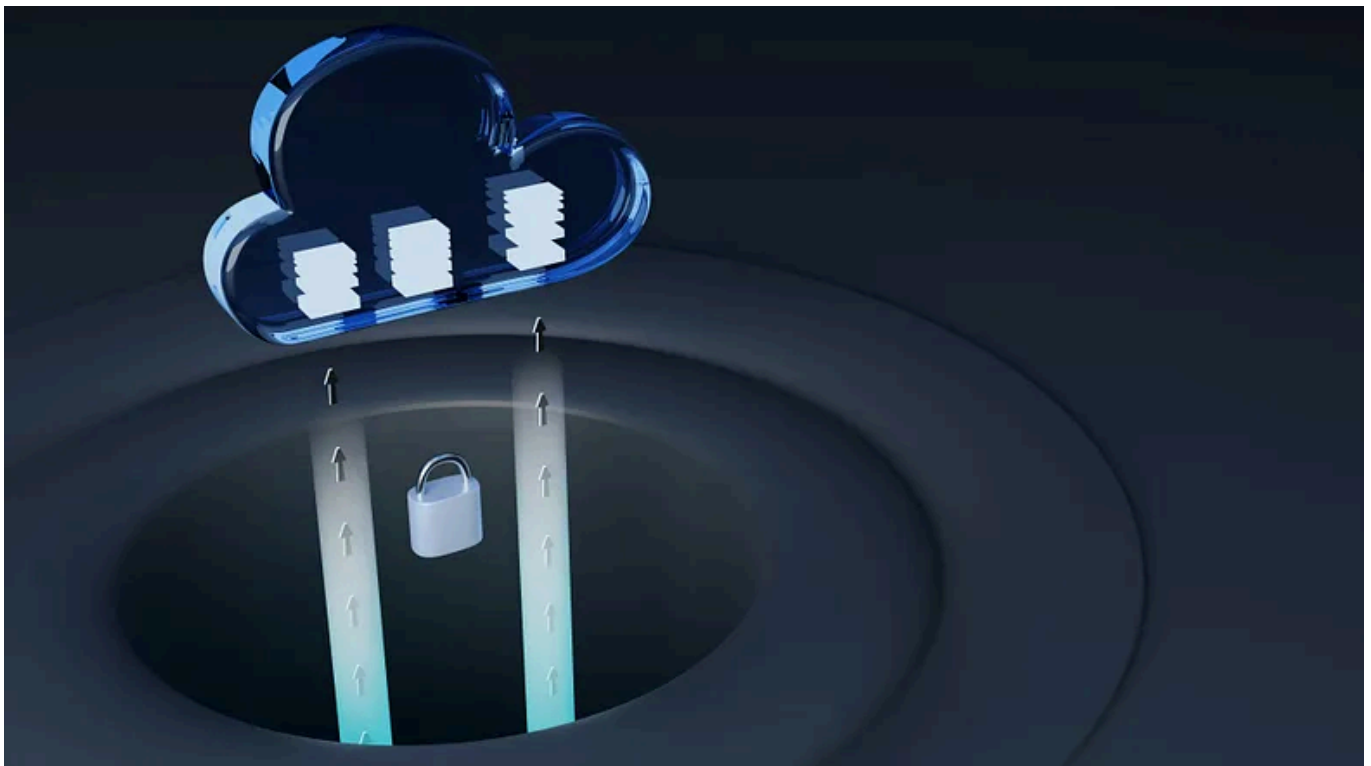
Overview

This project explores how to leverage the strengths of DuckDB and MotherDuck to build a robust data processing and storage solution. DuckDB excels at fast in-memory analytics, while MotherDuck provides a scalable and cost-effective cloud data warehouse. By combining these technologies, you can achieve optimal performance for both local and cloud-based data operations.

Environment settings

```python
import polars as pl
import duckdb as db
import glob
```

Extraction from data sources

- csv files

```python
csv_files = glob.glob('./datasets/*.csv')
```

```python
list(enumerate(csv_files))
```

```
[(0, './datasets/watercollection.csv'),
 (1, './datasets/ContainsNull.csv'),
 (2, './datasets/sales_info.csv'),
 (3, './datasets/cdmx-subway.csv'),
 (4, './datasets/airports.csv'),
 (5, './datasets/colors.csv'),
 (6, './datasets/sets.csv'),
 (7, './datasets/appl_stock.csv'),
 (8, './datasets/sales.csv')]
```

- json files

```python
json_files = glob.glob('./datasets/*.json')
```

```python
list(enumerate(json_files))
```

```
[(0, './datasets/prevalencia.json'), (1, './datasets/people.json')]
```

- database tables

```python
db_files = glob.glob('datasets/*.db')
```

```python
list(enumerate(db_files))
```

```
[(0, 'datasets/retail_db.db'), (1, 'datasets/restaurants.db')]
```

Data warehouse creation

```python
conn = db.connect('my_database.db')
```

Data warehouse load

```python
conn.sql(f"create or replace table water_collection as
        select * from '{csv_files[0]}' ")
```

```python
conn.sql(f"create or replace table contains_null as
        select * from '{csv_files[1]}' ")
```

```python
conn.sql(f"create or replace table sales_info as
        select * from '{csv_files[2]}' ")
```

```python
conn.sql(f"create or replace table cdmx_subway as
        select * from '{csv_files[3]}' ")
```

```python
conn.sql(f"create or replace table airports as
        select * from '{csv_files[4]}' ")
```

```python
conn.sql(f"create or replace table colors as
        select * from '{csv_files[5]}' ")
```

```python
conn.sql(f"create or replace table sets as
        select * from '{csv_files[6]}' ")
```

```python
conn.sql(f"create or replace table appl_stock as
        select * from '{csv_files[7]}' ")
```

```python
conn.sql(f"create or replace table sales as
        select * from '{csv_files[8]}' ")
```

```python
conn.sql(f"create or replace table prevalencia as
        select * from '{json_files[0]}' ")
```

```python
conn.sql(f"create or replace table people as
        select * from '{json_files[1]}' ")
```

```python
retail = db.connect('./datasets/retail_db.db')
retail.sql('show tables')
```

```
┌─────────────────────┐
│        name         │
│       varchar       │
├─────────────────────┤
│     retail_sales    │
└─────────────────────┘
```

```python
retail_sales_pl = retail.sql('select * from retail_sales').pl()
```

```python
conn.execute("create or replace table retail_sales as from retail_sales_pl");
```

```python
restaurants = db.connect('./datasets/restaurants.db')
restaurants.sql('show tables')
```

```
┌─────────────────────┐
│        name         │
│       varchar       │
├─────────────────────┤
│     restaurants     │
└─────────────────────┘
```

```python
restaurants_pl = restaurants.sql('select * from restaurants').pl()
```

```python
conn.execute("create or replace table restaurants as from restaurants_pl");
```

Data retrieval

```
conn.sql('show databases')
```

```
| database_name |
|    varchar    |
|---------------|
| my_database   |
```

```
conn.sql('show tables')
```

```
┌─────────────────────────┐
│          name           │
│         varchar         │
├─────────────────────────┤
│ airports                │
│ appl_stock              │
│ cdmx_subway             │
│ colors                  │
│ contains_null           │
│ people                  │
│ prevalencia             │
│ restaurants             │
│ retail_sales            │
│ sales                   │
│ sales_info              │
│ sets                    │
│ water_collection        │
├─────────────────────────┤
│         13 rows         │
└─────────────────────────┘
```

```
conn.sql('select * from restaurants limit 5').pl()
```

name	rating_count	cost	city	cuisine	rating
str	i64	f64	str	str	i64
"The Golden Wok"	1477	33.620488	"Berlin"	"American"	5
"Greek Gyros"	770	68.388874	"New York"	"French"	1
"Taste of Italy"	4420	88.23168	"Amsterdam"	"Chinese"	0
"Midnight Diner"	2155	12.965985	"Lisbon"	"Mexican"	1
"Taste of Italy"	3375	52.785226	"Sydney"	"Chinese"	1

Cloud Data Warehouse with MotherDuck

```python
dw = db.connect('md')
```

```python
dw.sql('select current_database()').show()
```

```
┌───────────────────┐
│ current_database() │
│      varchar       │
├───────────────────┤
│ my_portfolio       │
└───────────────────┘
```

Convert queries from local database to polars

```python
airports_pl = conn.sql('select * from airports').pl()
appl_stock_pl = conn.sql('select * from appl_stock').pl()
cdmx_subway_pl = conn.sql('select * from cdmx_subway').pl()
colors_pl = conn.sql('select * from colors').pl()
contains_null_pl = conn.sql('select * from contains_null').pl()
people_pl = conn.sql('select * from people').pl()
prevalencia_pl = conn.sql('select * from prevalencia').pl()
restaurants_pl = conn.sql('select * from restaurants').pl()
retail_sales_pl = conn.sql('select * from retail_sales').pl()
sales_pl = conn.sql('select * from sales').pl()
sales_info_pl = conn.sql('select * from sales_info').pl()
sets_pl = conn.sql('select * from sets').pl()
water_collection_pl = conn.sql('select * from water_collection').pl()
```

Upload dataframes to MotherDuck

```python
dw.sql(f"create or replace table airports as select * from airports_pl");
```

```python
dw.sql(f"create or replace table appl_stock as
    select * from appl_stock_pl");
```

```python
dw.sql(f"create or replace table cdmx_subway as
    select * from cdmx_subway_pl");
```

```python
dw.sql(f"create or replace table colors as select * from colors_pl");
```

```python
dw.sql(f"create or replace table contains_null as
    select * from contains_null_pl");
```

```python
dw.sql(f"create or replace table people as select * from people_pl");
```

```python
dw.sql(f"create or replace table prevalencia as
    select * from prevalencia_pl");
```

```python
dw.sql(f"create or replace table restaurants as
    select * from restaurants_pl");
```

```python
dw.sql(f"create or replace table retail_sales as
    select * from retail_sales_pl");
```

```python
dw.sql(f"create or replace table sales as
    select * from sales_pl");
```

```python
dw.sql(f"create or replace table sales_info as
    select * from sales_info_pl");
```

```python
dw.sql(f"create or replace table sets as
    select * from sets_pl");
```

```python
dw.sql(f"create or replace table water_collection as
    select * from water_collection_pl");
```

Check uploaded tables

```python
dw.sql('show tables')
```

```
┌─────────────────────────┐
│          name           │
│         varchar         │
├─────────────────────────┤
│ airports                │
│ appl_stock              │
│ cdmx_subway             │
│ colors                  │
│ contains_null           │
│ people                  │
│ prevalencia             │
│ restaurants             │
│ retail_sales            │
│ sales                   │
│ sales_info              │
│ sets                    │
│ water_collection        │
├─────────────────────────┤
│        13 rows          │
└─────────────────────────┘
```

Close all database connections

```
# close db connections
conn.close()
retail.close()
restaurants.close()
dw.close()
```

Conclusions

By combining `DuckDB`'s in-memory processing capabilities with `MotherDuck`'s cloud-based data warehousing, you can create a powerful and flexible data

processing and storage solution. This approach allows you to efficiently handle both local and cloud-based data operations, optimize performance,

Contact

Portfolio | Linkedin | Twitter

Data Science   Python   Duckdb   Motherduck   Etl

Published in T3CH

Follow

605 Followers · Last published 22 hours ago

Snoop & Learn about Technology, AI, Hacking, Coding, Software, News, Tools, Leaks, Bug Bounty, OSINT & Cybersecurity !i! But, not limited 2, anything that is Tech Linked…You'll probably find here ! ;)—Stay ahead with Latest Tech News! -> You write about? Just ping to join !

Written by Jesus L. Monroy

Follow

56 Followers · 23 Following

Economist & Data Scientist

No responses yet

More from Jesus L. Monroy and T3CH

In T3CH by Jesus L. Monroy

Designing Elegant Tables with Great Tables and Python

Great Tables Python is a powerful and versatile library that simplifies the process o...

Nov 1, 2024 👏 216
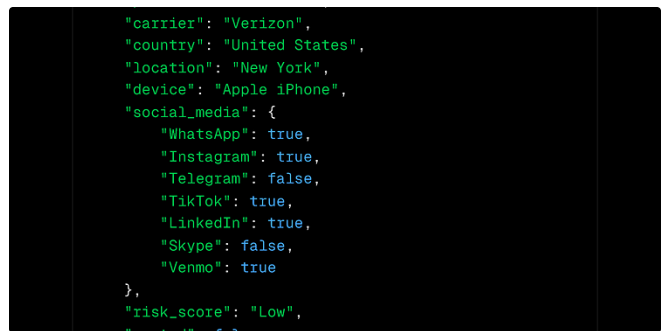
In T3CH by Khaleel Khan

Unlock Hidden Secrets: How This Tool Reveals Everything About An...

Deep-HLR: An Essential Tool for Fraud Prevention and OSINT Investigations

Aug 23, 2024 👏 453 💬 14
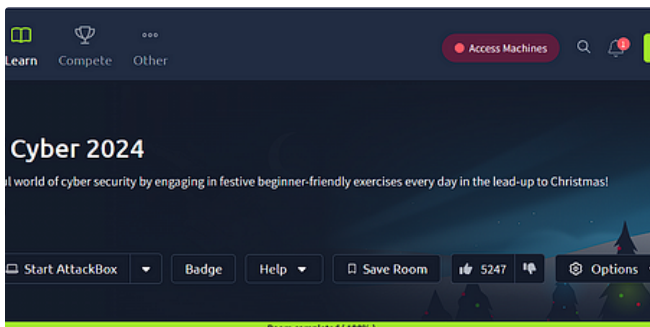
In T3CH by TRedEye

Advent of Cyber 2024 {All Tasks Update daily}—Tryhackme...

Advent of Cyber 2024 BY ::-> TRedEye

Dec 3, 2024 · 356 · 2

In T3CH by Jesus L. Monroy

Building ETL Pipelines in BigQuery with Python

In the realm of data analytics, Extract, Transform, Load (ETL) processes play a...

Oct 31, 2024 · 56

See all from Jesus L. Monroy   See all from T3CH

Recommended from Medium

In T3CH by TRedEye

In T3CH by Jesus L. Monroy

PY In Python in Plain English by Raphael Schols

How to Turn PDF Documents into Data Tables with Python

Learn how to extract data from PDFs and structure it into tables

✦ Jan 13   👏 180   💬 2   🔖   •••

Tomer Gabay

How to Setup Your Macbook for Data Science in 2025

Easy Steps to Get the Best Experience From Your MacBook as a Data Scientist

✦ Dec 28, 2024   👏 174   💬 1   🔖   •••

Lists
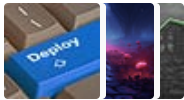
Predictive Modeling w/ Python
20 stories · 1789 saves

Coding & Development
11 stories · 981 saves

Practical Guides to Machine Learning
10 stories · 2165 saves

ChatGPT prompts
51 stories · 2487 saves