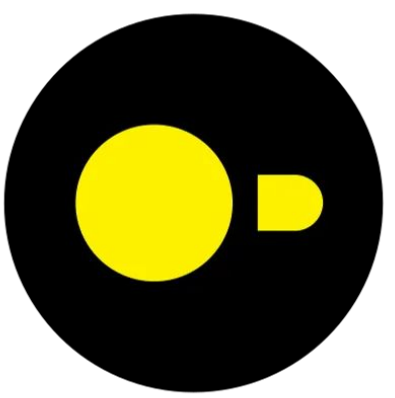
Why DuckDB?
Real Benchmark Results



DuckDB

By: Nael Aqel

Overview

- **Tools:** Pandas vs Polars vs DuckDB
- **Datasets (Synthetic E-Commerce):**
  - customers: 2,000,000 rows
  - orders: 8,000,000 rows
  - order_items: 20,000,000 rows
  - products: 20,000 rows
  - product_reviews: 4,000,000 rows
- **Tests:** File Loading + Multi-table Joins
- **Formats:** CSV vs Parquet
- **Key Metric:** Speedup Ratio

Loading Tables Performance

DuckDB is the Winner (specially for big datasets >1M rows):

- **CSV format:** ~2-200x faster than Pandas and Polars
- **Parquet format:** ~150-3,800x faster than Pandas and Polars
- For smaller datasets, DuckDB still leads but Polars may compete

Key insight: Performance gaps widen dramatically with dataset size

Joining Tables Performance

DuckDB is the Winner:

- **CSV format:** ~25-150x faster than Pandas and Polars
- **Parquet format:** ~1,500-12,000x faster than Pandas and Polars
- **Big Advantage:** No memory loading required

Takeaway: DuckDB's direct file joins are game-changing

Recommended Workflow

Optimal Data Processing Strategy:

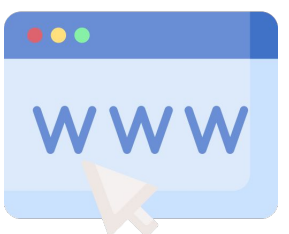
- **Use Parquet format** (massive performance boost)
- **Load with DuckDB**
- **Perform joins and transformations in DuckDB SQL**
- **Then convert to DataFrame when needed:**
  - **Pandas:** use .df()
  - **Polars:** use .pl()

Bottom line: Start with DuckDB, convert only when necessary

Links

- DuckDB Documentation
- Notebook Link
- Datasets Link
- Datasets Generator Notebook
- Datasets Generator Repo

FOLLOW ME :)



naelaqel.com

naelaqel1