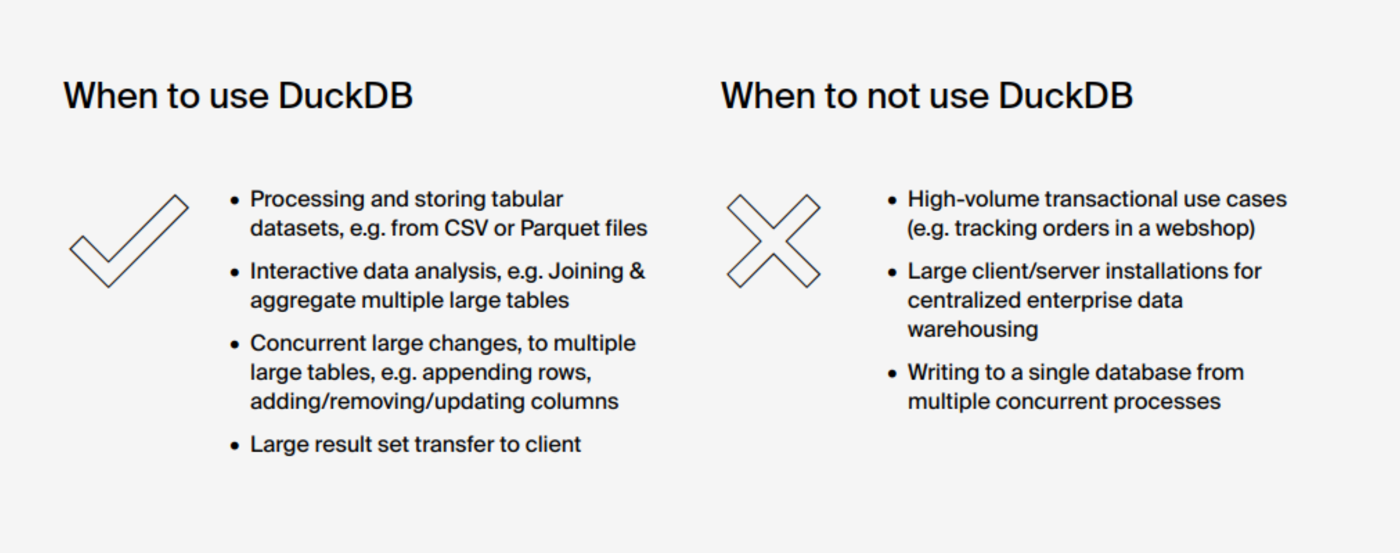
Data Wrangling

DuckDB

DuckDB is a lightweight OLAP type database system written in C++ and designed to be used for EDA style activities:



When to use DuckDB

✓

- Processing and storing tabular datasets, e.g. from CSV or Parquet files
- Interactive data analysis, e.g. Joining & aggregate multiple large tables
- Concurrent large changes, to multiple large tables, e.g. appending rows, adding/removing/updating columns
- Large result set transfer to client

When to not use DuckDB

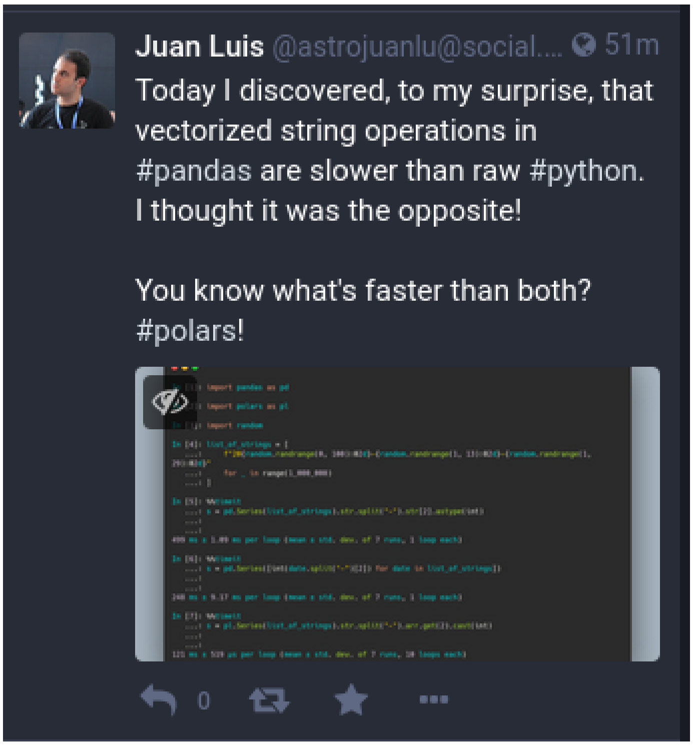
✗

- High-volume transactional use cases (e.g. tracking orders in a webshop)
- Large client/server installations for centralized enterprise data warehousing
- Writing to a single database from multiple concurrent processes

From their website: advice on when to use and not to use DuckDB

Polars

Polars is a rust-based data frames library with Python bindings



Juan Luis @astrojuanlu@social.... 51m

Today I discovered, to my surprise, that vectorized string operations in #pandas are slower than raw #python. I thought it was the opposite!

You know what's faster than both? #polars!

```
In [1]: import pandas as pd

In [2]: import polars as pl

In [3]: import random

In [4]: list_of_strings = [
   ...:     f"20{random.randrange(0, 100):02d}-{random.randrange(1, 13):02d}-{random.randrange(1,
29):02d}"
   ...:     for _ in range(1_000_000)
   ...: ]

In [5]: %%timeit
   ...: s = pd.Series(list_of_strings).str.split("-").str[2].astype(int)
   ...:
   ...:
499 ms ± 1.09 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)

In [6]: %%timeit
   ...: s = pd.Series([int(date.split("-")[2]) for date in list_of_strings])
   ...:
   ...:
240 ms ± 9.17 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)

In [7]: %%timeit
   ...: s = pl.Series(list_of_strings).str.split("-").arr.get(2).cast(int)
   ...:
   ...:
121 ms ± 519 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

Here is a talk that Juan Luis gave about the library

Revision #6
Created 4 November 2022 11:58:18 by James
Updated 8 February 2024 21:20:15 by James