

A duck for your dashboard

Performant data apps in the browser with DuckDB

Robert Kosara
Observable



Agenda

DuckDB in Observable notebooks

DuckDB in Observable Framework

Customer POC

Wish list

Conclusions

Q+A

DuckDB in notebooks





Pizza Paradise Playground

Pizza Paradise is a fake dataset on a pizzeria franchise.

Pizza Paradise is trying to figure out which of our pizza products perform best in the market. Customers can pick from our selection of top pizzas, such as your classic *pepperoni pizza*, *hawaiian pizza*, *meat lovers*, and many more.

Pizza Paradise has many pizzerias throughout California and Nevada, so it's helpful for us to know which stores do the best. And of course, inventory is a big part of our pizza making process. Do we have enough inventory to keep up with the demand?

They're moving into new pizza territory with their newest *oxtail pies*. It's taken up a large share of our sales.

```
import {db} from "@observablehq/pizza-paradise"
```



How was Pizza Paradise DB made? Check out [this notebook](#) for how we did it.

```
orders = ▶ Array(2046713) [Row, Row, Row, Row, Row, Row, Row, Row, Row, Row, Row, Row, Row, Row, Row, Row, Row, Row, Row, Row, ...]
```

```
① orders = db.query("SELECT * FROM orders") ▶
```

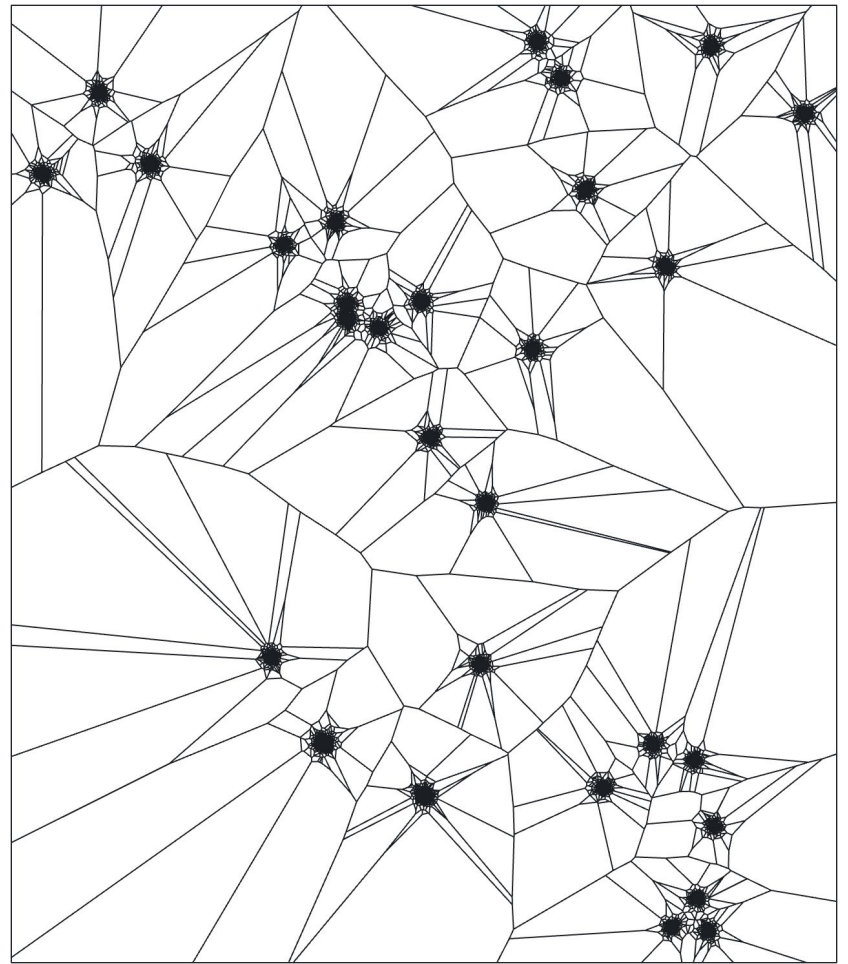
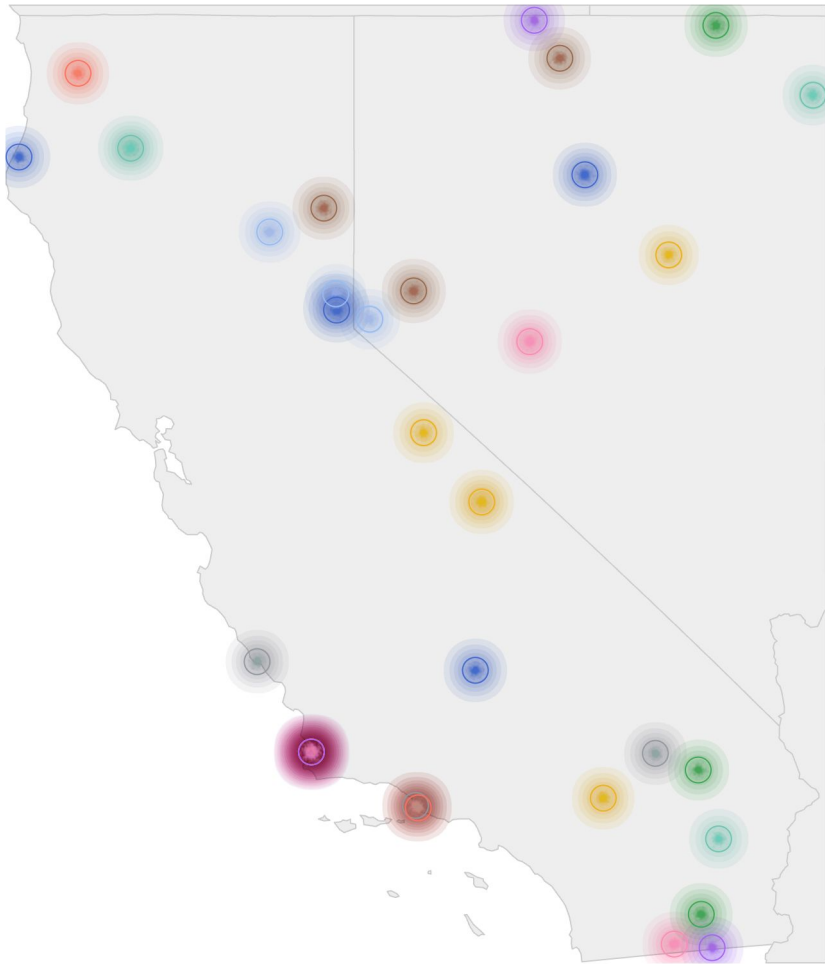
```
distanceFromStore = ▶ Float64Array(2046713) [1.5245699544001918, 5.408841357241775, 2.560939402538244, 0.943015769609154, 0.5104252400929]
```

```
① distanceFromStore = {  
  const c = d3.index(customers, (d) => d.id);  
  const s = d3.index(stores, (d) => d.id);  
  return Plot.valueof(orders, ({ customerId, storeId }) =>  
    d3.geoDistance(  

```

customers3 db ▾

```
SELECT c.longitude as longitude
       , c.latitude as latitude
       , c.id as customerId
       , o.storeId as storeId
       , COUNT(*) orders
FROM orders as o
JOIN customers as c
  ON o.customerId = c.id
GROUP BY 1, 2, 3, 4
```

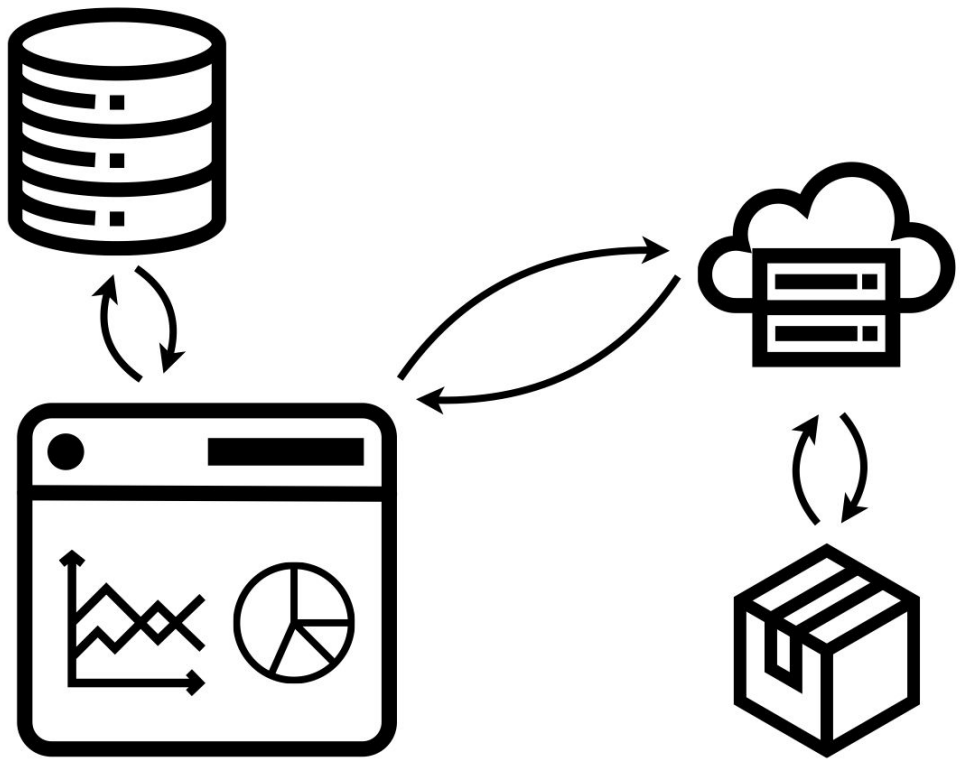


DuckDB in Observable notebooks

- **DuckDBClient** in standard library
- **Data wrangling and querying**
 - Load CSV, etc.
 - Cleanup
 - Query to aggregate, join, etc.
- **Use from data table cell, SQL cell, Observable Plot**
- **Ideal for running in the browser**
- **Easily handles millions of rows**

DuckDB in Observable Framework







Demo

DuckDB in Observable Framework

- **SQL frontmatter to load CSV, parquet, DuckDB, etc.**
 - Creates tables in DuckDB instance
 - Triggers data loaders
 - Easy querying from fenced SQL code blocks
- **Use in data loaders to wrangle multi-GB files**
 - Full SQL incredibly useful for data wrangling
 - Fast!
- **Tied into Observable Plot**
- **Used extensively in large customer POC**

Customer POC



Customer POC

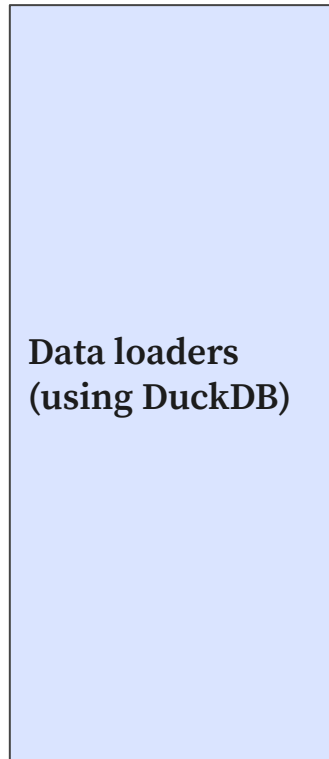
- **Customer provides analytics for large number of clients**
- **Wants faster and more modern data app**
 - Fast loading
 - Richer visualization and interaction
- **Needs to create separate apps for 100s or 1000s of customers**
- **Large amount of data for each customer (millions of rows)**

- **Initially: DuckDB to query all data, then wrangle in JavaScript**
- **But then...**

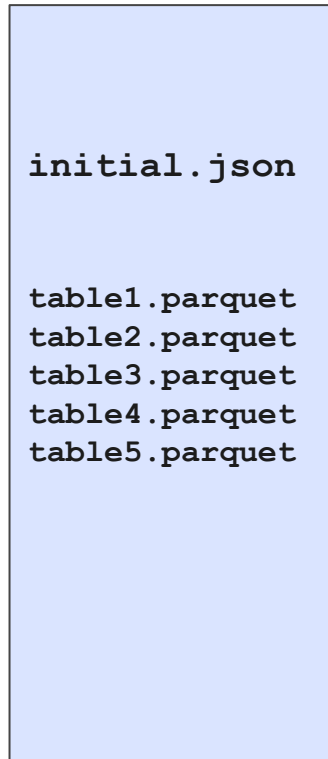
*I went for a walk,
and came to the realization that
I needed to **trust DuckDB**.*

Fil Rivière
Observable

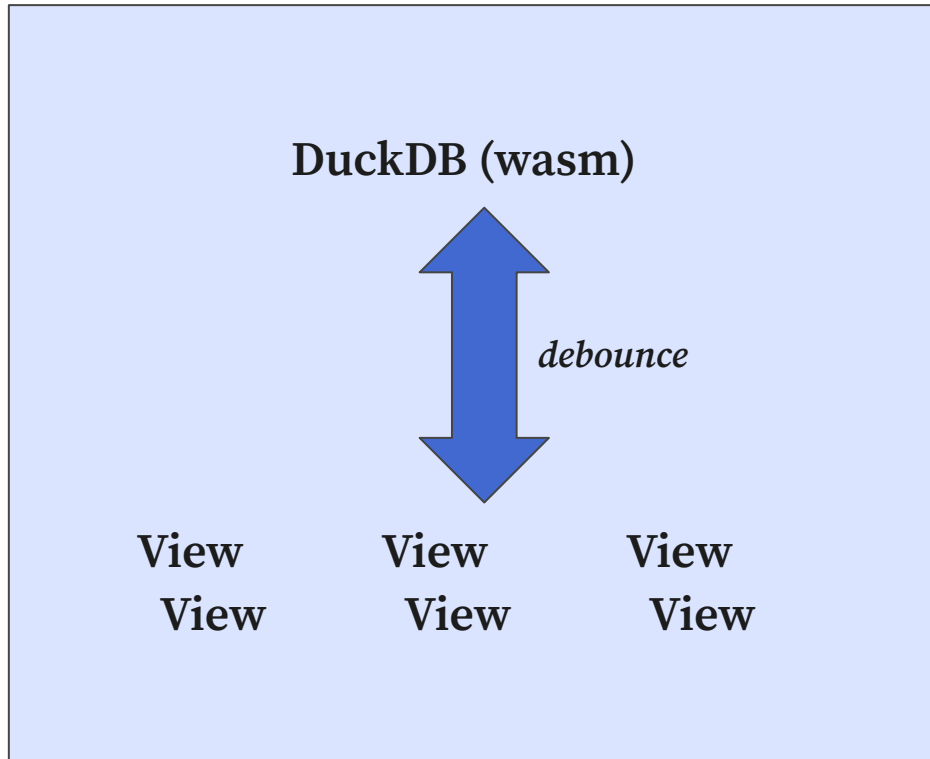
Build



Server



Client



Tricks for faster loading

- Load a small file with initial values to show something quickly
 - `initial.json`
- Bootstrap DuckDB once initial dashboard is loaded
- “Rehydrate” dashboard views with the actual data
 - Parquet files for large tables (millions of rows each)
- Interaction drives database queries
 - Requires debouncing of queries
 - DB can't keep up with interaction at 60fps
- DuckDB also used in build step to process 100s of multi-GB files

Wish list



Our wish list...

- **Support for rehydration**
 - Cache queried results inside DuckDB file format?
- **Debouncing**
 - Prepared query with parameters passed in
 - New call to same query cancels previous one
- **DuckDB WASM optimizations**
 - Smaller client (perhaps space optimizations?)
 - Modularization to load just what is needed, bootstrap faster
- **DuckDB file format stability**
 - Problem during POC, older format not always readable

A duck for your dashboard

- DuckDB in Observable notebooks
- DuckDB in Observable data apps
- DuckDB in data loaders
- DuckDB on the backend
- DuckDB on the frontend
- DuckDB for early exploration
- DuckDB for development
- DuckDB for production use
- **DuckDB everywhere!**

A duck for your dashboard

Performant data apps in the browser with DuckDB

Robert Kosara
Observable