

# Postgres Statistic Injection

Ethan and Roy

# Overview

- Be able to dump, modify, and load Postgres statistic information from one machine to another
- Run EXPLAIN on a query without having the actual data stored on that instance
- User-friendly interface for statistic modification

# Goals

75%

- CLI dump, modification, load (DONE)

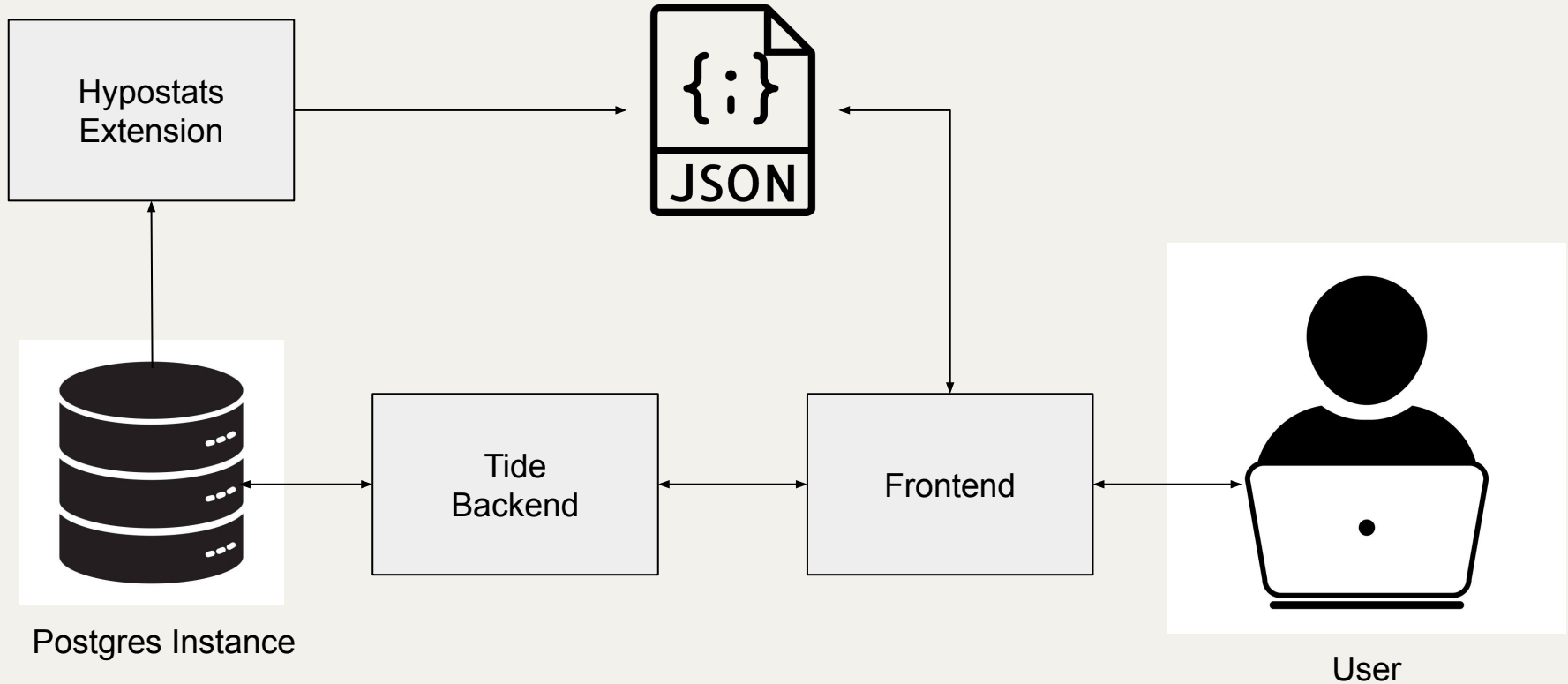
100%

- Working UI that supports those functions (DONE)
- Allow types of data that are not strictly numeric (MOSTLY DONE)
  - Can load in dates, strings, numeric types
- Verify modifications (NOT STARTED)

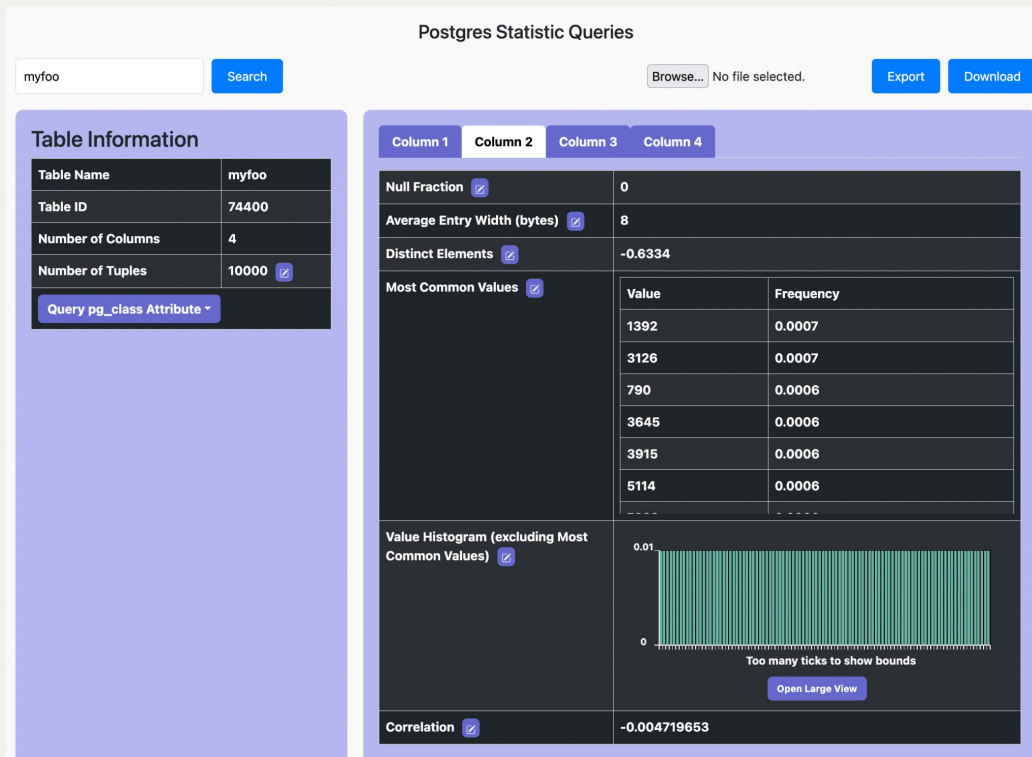
125%

- Delta changes (NOT STARTED)

# Architecture



# DEMO



# Testing

- Small, custom tables
- TPC-H queries
- Queries seem to change! (kinda...)
- Ex. TPC-H q1 (modifying reltuple count)

## Explain Plan

```
Finalize GroupAggregate (cost=240099.79..240101.86 rows=6 width=236)
  Group Key: _l_returnflag, _l_linestatus
  -> Gather Merge (cost=240099.79..240101.19 rows=12 width=236)
    Workers Planned: 2
    -> Sort (cost=239099.76..239099.78 rows=6 width=236)
      Sort Key: _l_returnflag, _l_linestatus
      -> Partial HashAggregate (cost=239099.52..239099.69 rows=6 width=236)
        Group Key: _l_returnflag, _l_linestatus
        -> Parallel Seq Scan on lineitem (cost=0.00..141624.64 rows=2436872 width=25)
          Filter: (_l_shipdate <= '1998-08-08 00:00:00'::timestamp without time zone)
```

## Explain Plan

```
GroupAggregate (cost=110573.83..110698.27 rows=6 width=236)
  Group Key: _l_returnflag, _l_linestatus
  -> Sort (cost=110573.83..110581.14 rows=2924 width=25)
    Sort Key: _l_returnflag, _l_linestatus
    -> Seq Scan on lineitem (cost=0.00..110405.50 rows=2924 width=25)
      Filter: (_l_shipdate <= '1998-08-08 00:00:00'::timestamp without time zone)
```

# Code Quality

- Automation scripts, dataset loading, modification API endpoints (good!)
- Code repetition, modularization and type verification (could improve...)
- For production: edge cases, error checking, and feature completeness

# Future Directions

- Porting to other instances of Postgres
  - Needs a “fake” table to fully run explain
- Type support and verification on the front-end (or run through the backend)



# Sources

- ChatGPT
- Wan's starter code
- <https://docs.rs/pgrx/latest/pgrx/>
- [https://github.com/postgres/postgres/blob/master/src/include/catalog/pg\\_statistic.h](https://github.com/postgres/postgres/blob/master/src/include/catalog/pg_statistic.h)
- <https://www.react-graph-gallery.com/histogram>  
(Histograms)