CURRENT VINTAGE TABLE: DATABASE METADATA OVERVIEW

Mahika Calyanakoti
Civic Data Fellow

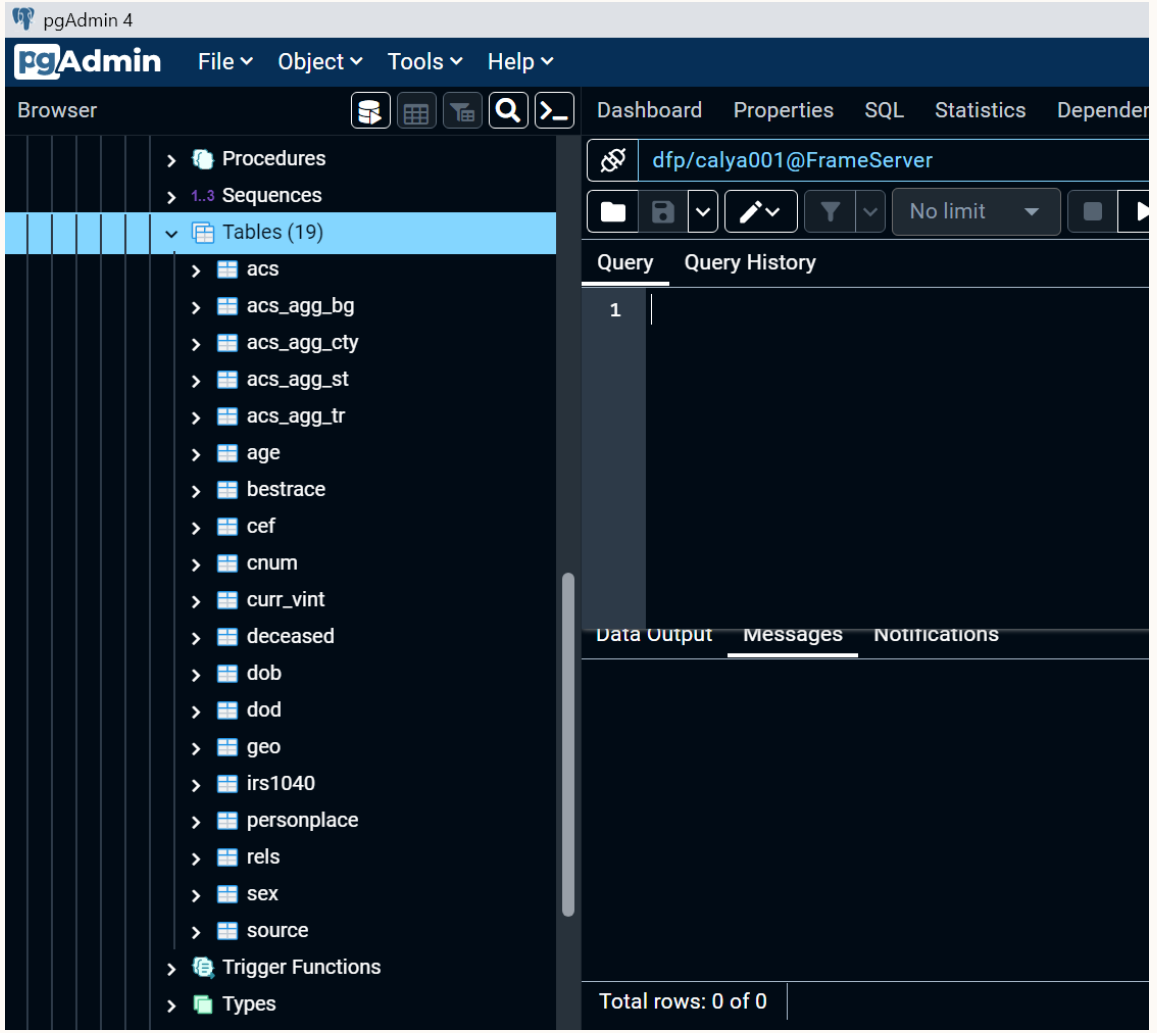
FRAMES PROGRAM

- Creating infrastructure to modernize the Census Bureau's statistical foundation
- Integrates various data input sources in a relational database
- Demographic: person-level data with identifiers
- Linked with Geospatial, Business, and Job frames

OBJECTIVES

- Snapshot of metadata with PostgreSQL - vintages in the database
- Query unique source IDs of all tables in a schema – quick and efficient!
- Save the code as a function to be used by others

PG ADMIN 4 VIEW



PROJECT TASKS

TOOLS

Use and get familiarized with PG Admin 4, PostgreSQL, Git & VS Code

QUERY

Wrote PostgreSQL to query the unique source IDs of all tables in a desired schema

VINTAGE TABLE

Saving the results in two columns: sourceid and table_name

MATERIALIZED VIEW

Using the source and curr_vint tables to compile metadata about the tables & vintages

DOCUMENT

Tracking script running time and writing documentation & in-line comments

CODE SNIPPET

create_curr_vint.sql

H: > create_curr_vint.sql

```
    -- set ownership to fed_intern
    v_sql_query := format('ALTER TABLE %I.curr_vint OWNER TO fed_intern', v_schema_name);
    EXECUTE v_sql_query;

    -- loop through each table in the desired schema (except 'source')
    FOR v_table_name IN
        SELECT table_name
        FROM information_schema.tables
        -- want to not include 'source' table since that includes datasets we don't want
        WHERE table_type = 'BASE TABLE' AND table_schema = v_schema_name AND table_name != 'source' AND table_name != 'curr_vint' AND
            -- only query tables that have a sourceid column
            EXISTS (
                SELECT 1
                FROM information_schema.columns
                WHERE table_schema = v_schema_name AND table_name = information_schema.tables.table_name AND column_name = 'sourceid'
            )
        -- TEST MODE: uncomment the following line and comment out the entire above "WHERE" statement
        -- WHERE table_type = 'BASE TABLE' AND table_schema = v_schema_name AND table_name = 'acs_agg_st'
    LOOP
        -- generate the sql insert statement based on the table name, finding distinct sourceid
        v_sql_query := format('INSERT INTO %I.%I (sourceid, table_name) SELECT DISTINCT sourceid, %L FROM %I.%I',
         v_schema_name, 'curr_vint', v_table_name, v_schema_name, v_table_name);
        -- v_sql_query := format('SELECT DISTINCT sourceid FROM %I.%I', v_schema_name, v_table_name);

        EXECUTE v_sql_query;
    END LOOP;

    -- end timer and print elapsed time
```

DIFFICULTIES & SOLUTIONS

RUNNING LARGE QUERIES

- Nohup in Linux to work around
- Running from terminal
- Using nohup.out
- Test on smaller data

EXPANDING TO DIFFERENT APPLICATIONS

- Clear user manual about adjustments
- Flexible variables

FAMILIARITY WITH NEW TOOLS

- Communication with team members
- Using past intern's online resources
- Google!

THANK YOU !

August 2023