**Project Proposal:** We will create a database that reflects the air quality of the fastest-growing cities in the US which is a valuable use case for real estate builders for analysis.

**Strategy:**

14 US cities with the highest population growth → take the latitude and longitude from the metropolitan areas of those cities → query open weather api for historical data based on those zip codes → pull other sources as CSV files where available.

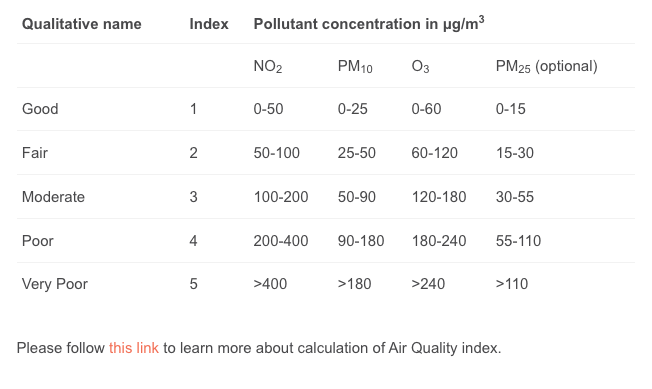
**Extract**

**Datasets used and their sources:**

**Air Quality Dataset Source (Used 3 sources)**

-Open weather API for historical air pollution data: <https://openweathermap.org/api/air-pollution>

Air quality is measured by:

****

-Air Quality Historical Data:[**https://aqicn.org/data-platform/register/**](https://aqicn.org/data-platform/register/)

-Daily temperature of major cities: https://www.kaggle.com/datasets/sudalairajkumar/daily-temperature-of-major-cities

**US Cities Dataset (Used 2 sources)**

-Dataset for US cities

<https://simplemaps.com/data/us-cities>

We determined we should look at the data for the top x growing cities in the United States and found good data for the top 14 from world population review. This in combination with the data from simple maps is to be used to acquire latitude/longitude data required by the openweather api.

-"Fastest Growing Cities in the Us 2022", World Population Review, <https://worldpopulationreview.com.> Accessed 17 March 2022.

**Lessons Learned:**

* everything should always be lowercase
* The blob is your friend
* The government does not build effective APIs

**Transform**

Cleaning

* We had to change data types for date, which was stored as a string, and City, which was stored as an object.
* We removed irrelevant data from our US Cities CSV, like zip codes and time zone.
* We had to match all columns in the CSVs to ensure consistency and easy merging
* The original city\_temperature file was too big to be uploaded to GitHub so it was narrowed down some before we got to v2.
* Our city\_temperature\_v2 CSV and US Cities contained cities we were not interested in, so we had to remove any row not from the US and not from the top growing city list
* We only used data from after 2019, so we had to remove some rows in our city\_temperature\_v2 CSV based on the year the temperature was recorded
* We took daily air quality data from [**aqicn.org**](https://aqicn.org/data-platform/register/), which publishes quarterly CSVs of their data. We concatenated 13 CSVs from that site which provide us with the data from 2019-2022
* Open weather data needed to be placed in more organized lists to calculate averages from the data like we desired

Merge

* We merged our two city dfs (one which had extensive information about US cities, and one that listed the fastest growing cities in the US) to get all information we want on from the big database for our target cities
* We used the same process to narrow down the information in our temperature CSVto only show us the data for the cities we targeted
* We merged our US air quality data with the city dataframe on City to narrow down our air quality data to the target cities

Filtering

* We filtered the air quality data by country = US to only show the US cities.
* We used .loc to create tables for specific Specie

Formatting

* We had to change our case sensitive columns to lowercase

**Load**

We decided to use PGAdmin which is a relational database for storage for the following reasons:

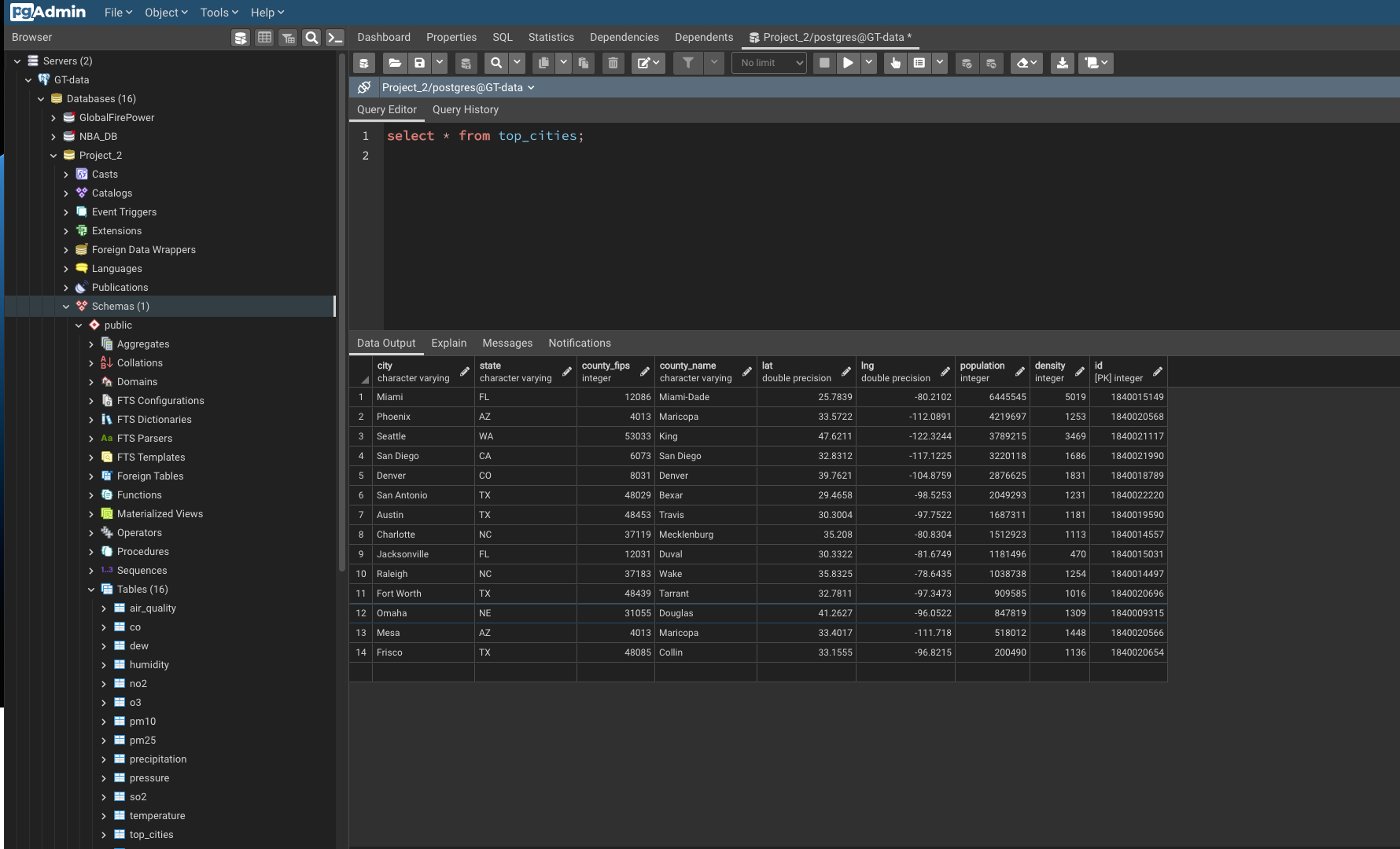
1. Our data was cleaned and structured well.
2. Since we need ease of access to our users we thought a relational database is the way to go as it doesn’t include a hierarchical model.
3. Since our data was not complex and doesn’t require complex querying it was best to use a relational database.
4. PGAdmin supports our JSON input which we used for our Open Weather Map Air quality data source.

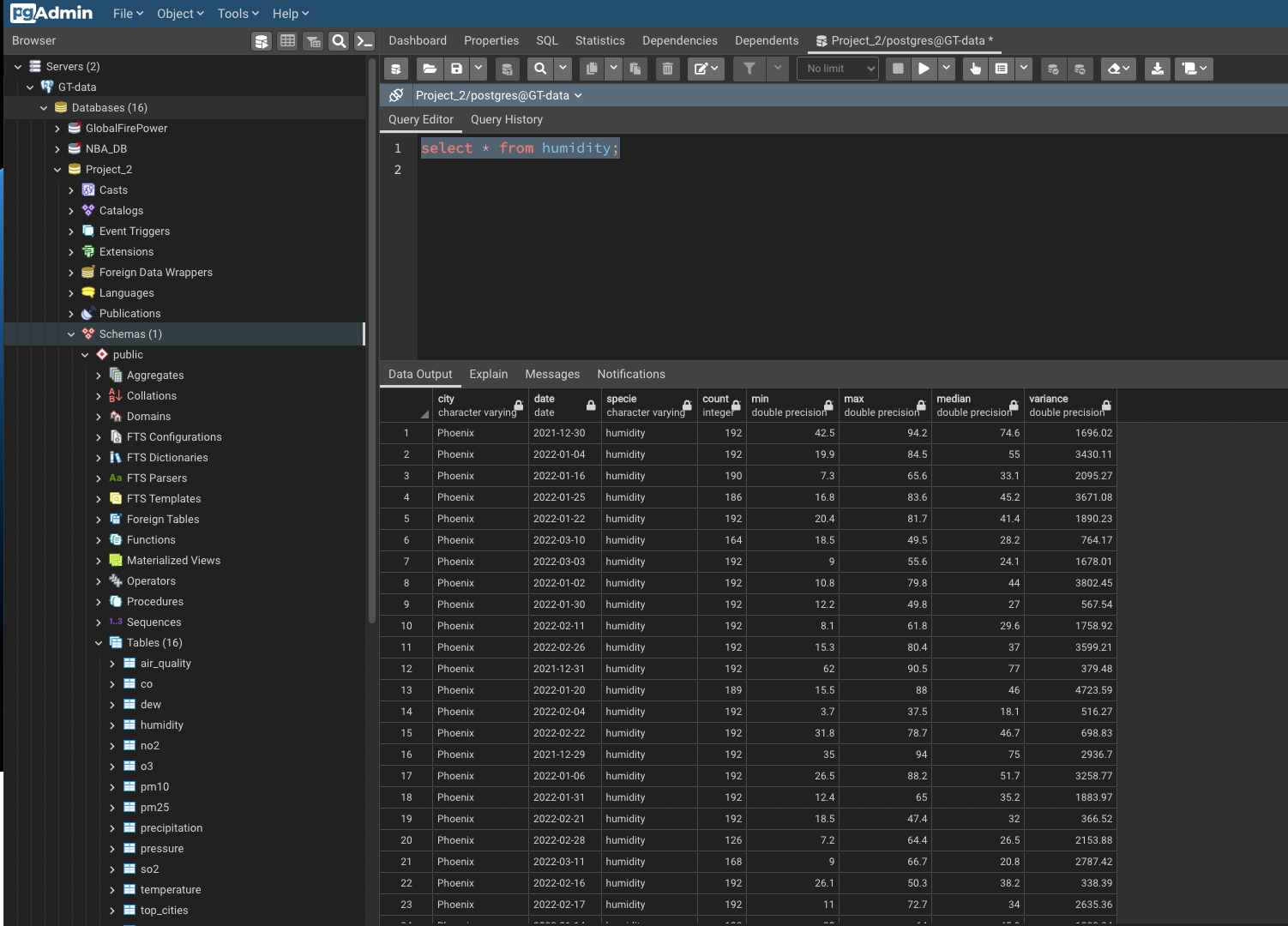
For creating the tables we used <https://app.quickdatabasediagrams.com> to create our schemas before importing our datasets from pandas.

The final tables created in our PGAdmin database for user view:

* top\_cities
* Co (Carbon Monoxide)
* dew
* humidity
* no2 (Nitrogen Dioxide)
* o3 (Ozone)
* pm10 (Particulate Matter 10)
* pm25 (Particulate Matter 25)
* precipitation
* pressure
* so2 (Sulfur Dioxide)
* temperature
* wind\_direction
* wind\_gusts
* Wind\_speed
* Air\_Quality

Some examples of our created tables in PGAdmin database.





Text

Description automatically generated

Dataframe representation of the Air\_Quality table