# *Pollution in Poland*

## *ETL & Data App*

### *PROJECT OVERVIEW*

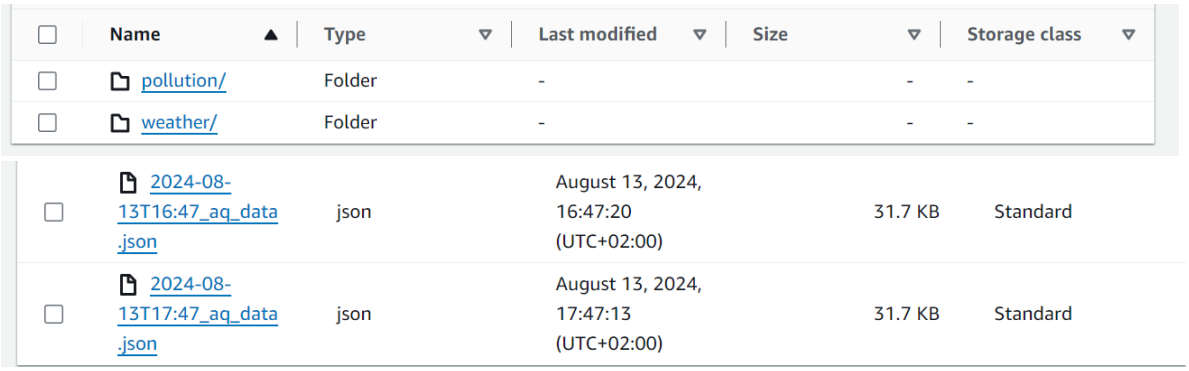
This is a data engineering project focused on air pollution data in biggest Polish cities (or city districts) with population over 100k. Pollution data is accompanied by weather info for better visualization and more possibilities.

## 1. Data Source

The list of locations (cities and/or districts comes from OpenDataSoft OpenDataSoft - Cities with population over 1000
The file provided above has been filtered according to my project specific needs – i filtered the data for polish cities with a population of at least 100 000. The file sits in GitHub repository, for which a Snowflake connection is created – making it possible to fetch latest version and run stored procedure that upserts the table.

Based on these locations and their coordinates, here comes the main data – air pollution & weather data from OpenWeather API.
Python script calls both APIs for each pair of lat&lon and provides detailed response.

The response data, after minor modifications is uploaded to S3 bucket, with a timestamp, into 2 separate directories:

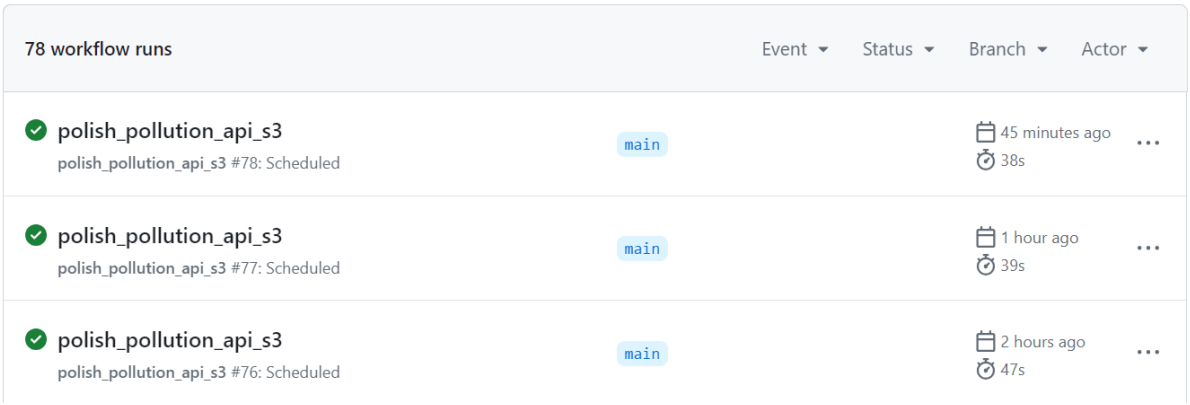


| Name | Type | Last modified | Size | Storage class |
|---|---|---|---|---|
| pollution/ | Folder | - | - | - |
| weather/ | Folder | - | - | - |
| 2024-08-13T16:47_aq_data.json | json | August 13, 2024, 16:47:20 (UTC+02:00) | 31.7 KB | Standard |
| 2024-08-13T17:47_aq_data.json | json | August 13, 2024, 17:47:13 (UTC+02:00) | 31.7 KB | Standard |

## 2. Data ingestion

The ingestion process is realized by the use of GitHub Actions. The python script, after configuring AWS credentials and API key as GitHub secrets, is scheduled to run every hour at 45th minute, to provide 24files per day, per location.

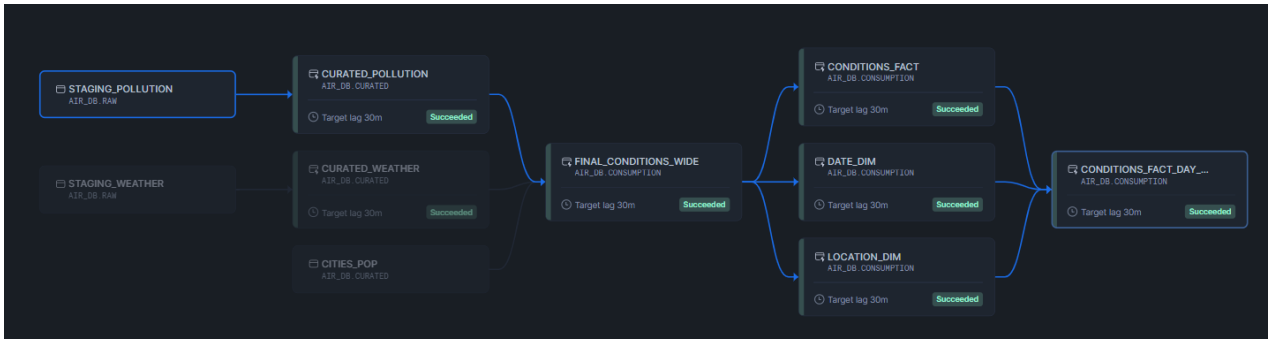
```yaml
on:
  # Optional manual trigger - debugging
  # workflow_dispatch:
  schedule:
    - cron: '45 * * * *' # run script every hour at 45th min
```



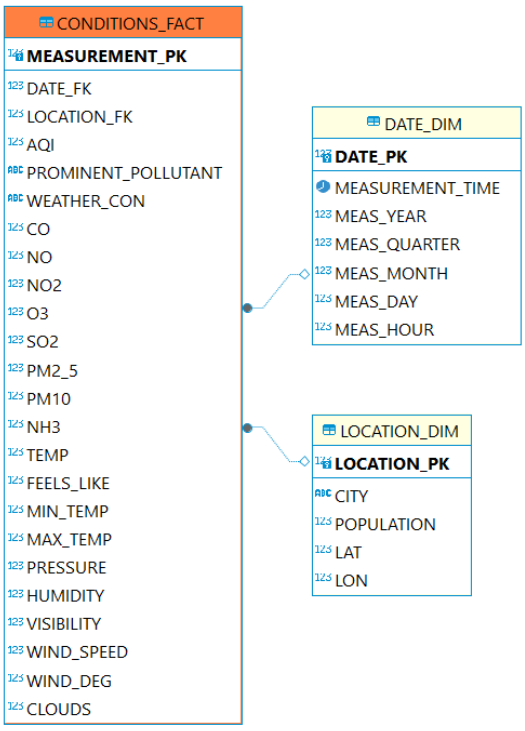

Besides that, a SnowPipe (actually 2 of them) is configured to automatically ingest fresh JSON files from S3 to Snowflake, once the SQS queue sends such notification. Raw data (with some metadata) is copied into staging tables, separately for pollution and weather.

## 3. Data transformation & storage

For the whole transformation and storage, only Snowflake is used, of which most stands for Dynamic Tables – as one of the main utils of this project. Starting from raw data, dynamic tables are connected one to another, creating a Wide Table on the way, which later gets split into star schema – fact and dimension tables. To top it out, an aggregated, day-level fact table is created for faster querying.

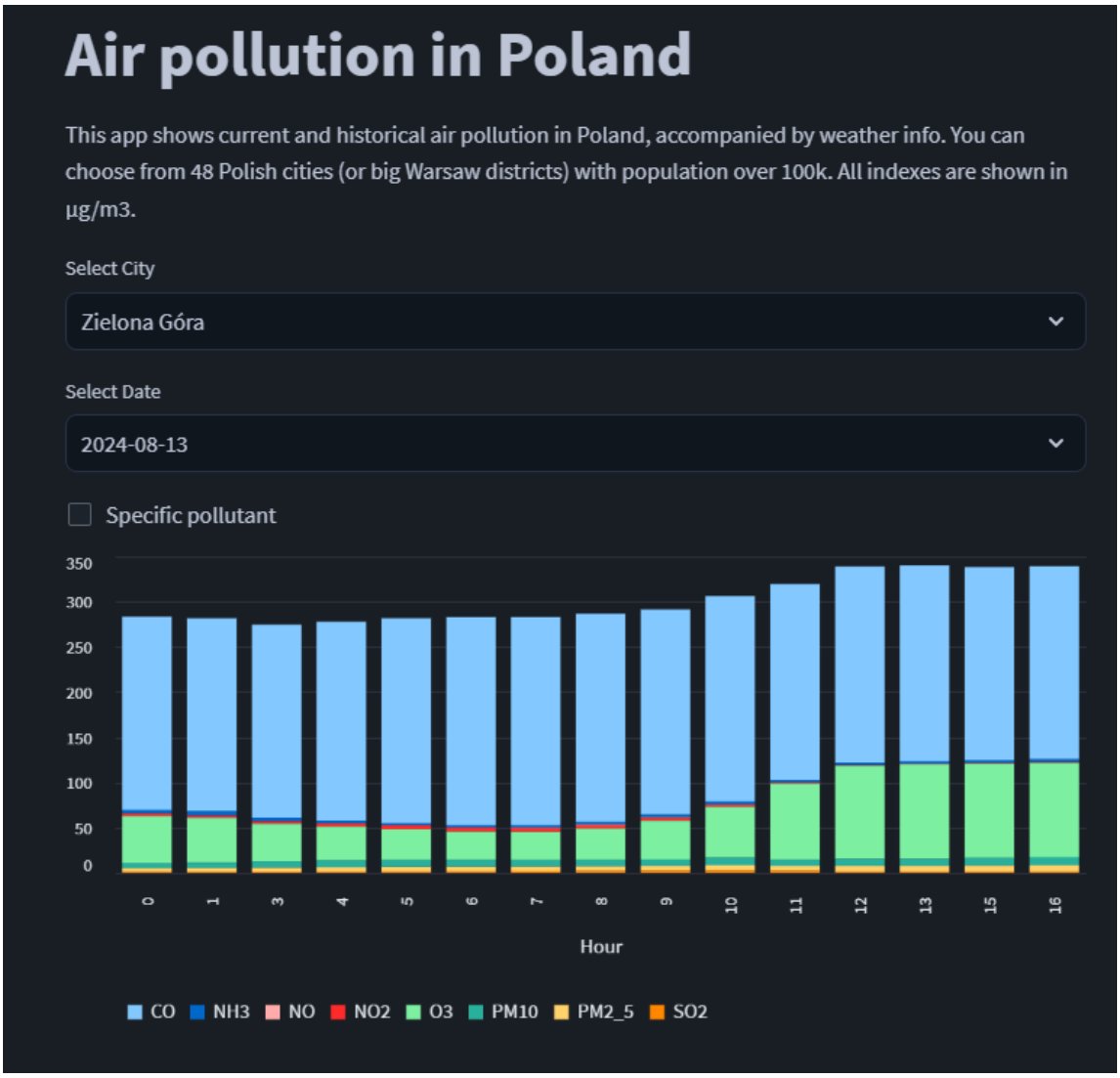




Although it is not possible to create foreign key constraints on Snowflake's dynamic tables, the following picture shows theoretical relationship in this star schema:

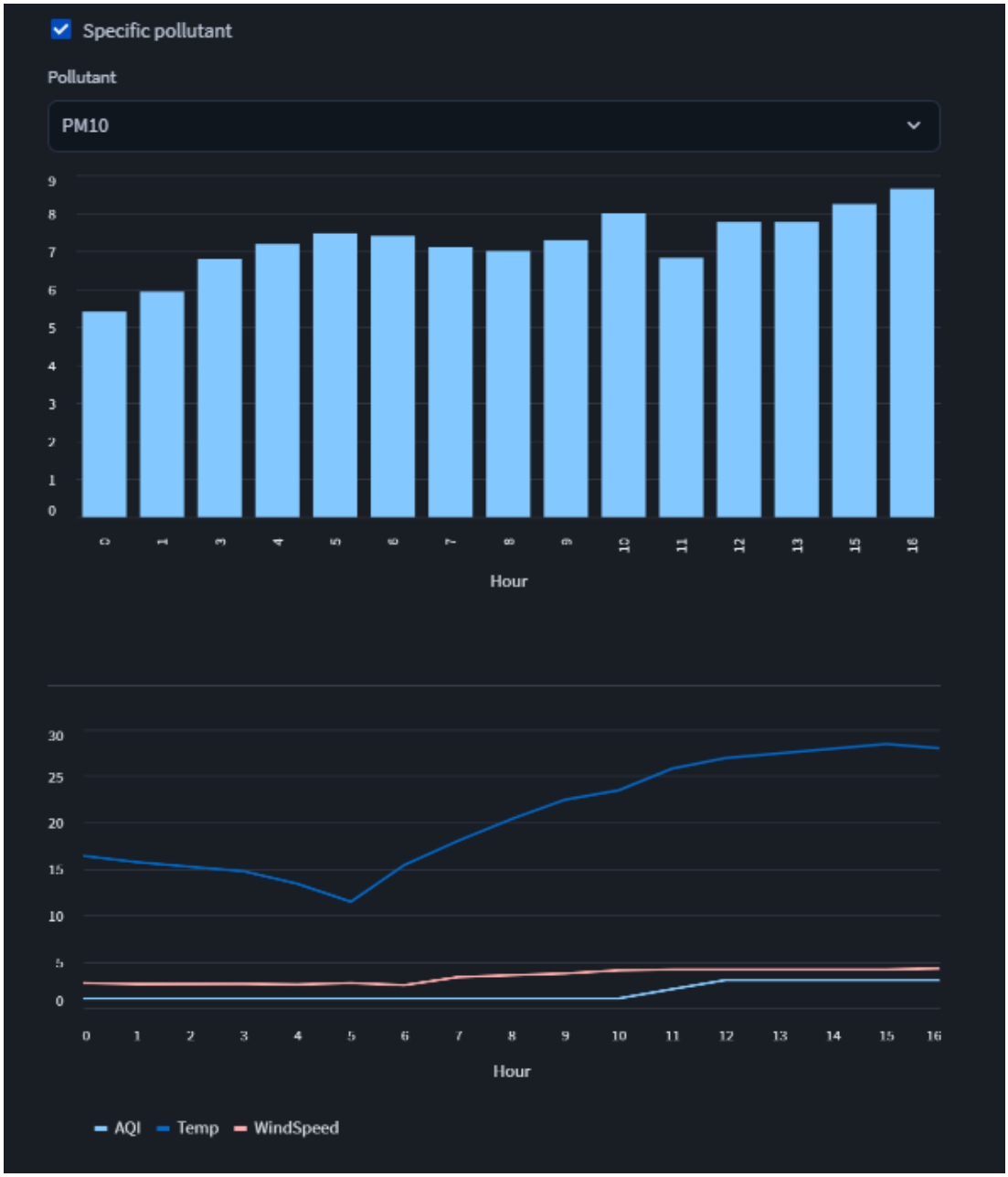



## 4. Visualization

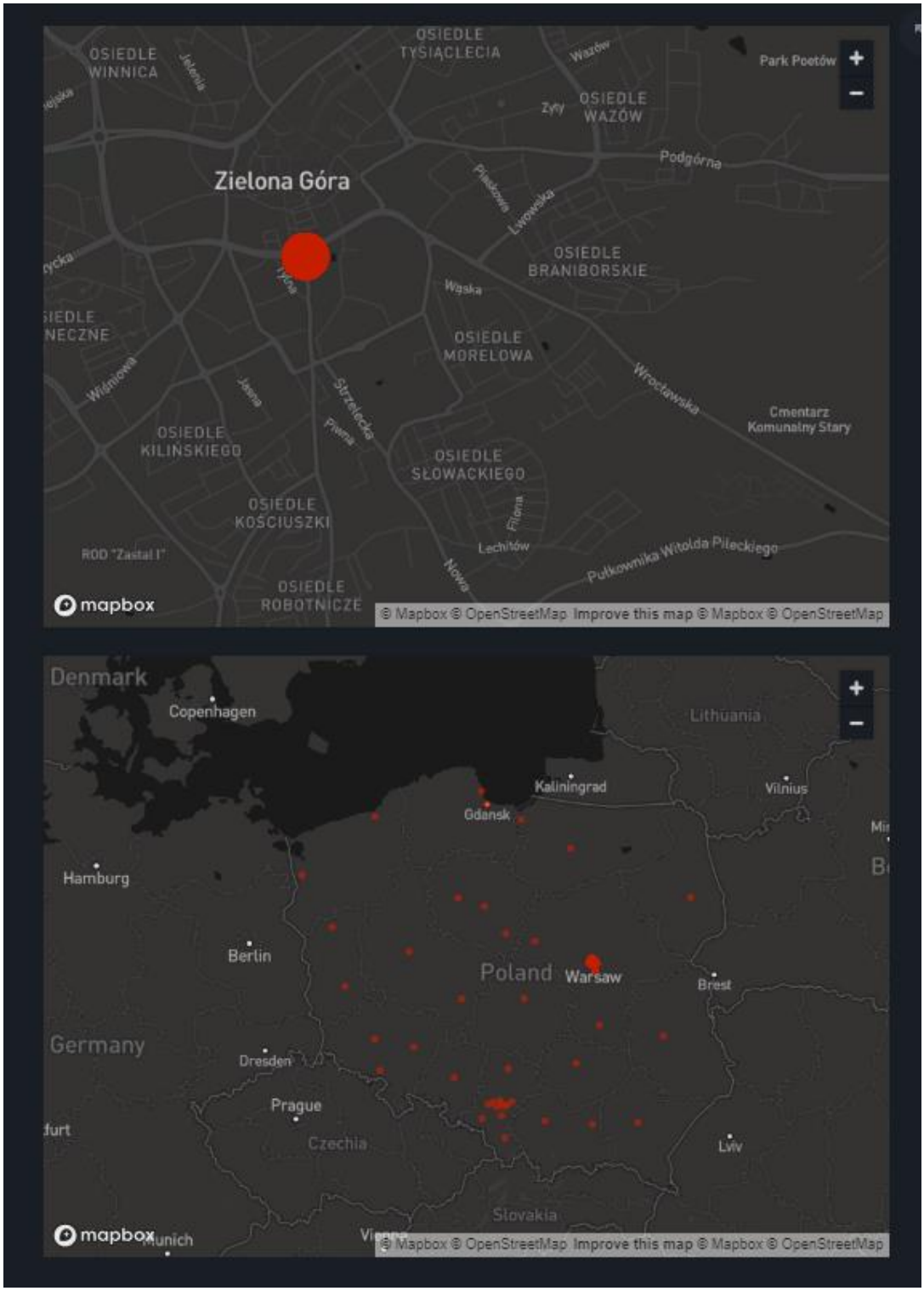
One of key elements in this project, is the data app created using built-in version of Streamlit in Snowflake.
The app starts with a  few dropdown lists, where you can choose specific location and date. Besides that, you can choose between all-in-1 pollutants approach to show on graphs and a single-pollutant mode where you search for specific index.
The app also shows current selected location on the map, as well as all available location on the zoomed out map.




*Streamlit Data App – Part #1/3*




*Streamlit Data App – Part #2/3*




*Streamlit Data App – Part #3/3*