
# Big Data - Ex 1 Summary

Github Repo: https://github.com/voidbar/runi-bigdata-2024
Submitted By: Tomer Lev (207906058), Amir Sharir (315857722), Elad Solomon(204313076)

# Database Selection and Reasoning

We have chosen a database that deals with air pollution measurement information in Seoul, South Korea. We chose this dataset as it comprised multiple tables which we could then use to demonstrate Cassandra capabilities when joining all of those into a single table.

# Database Design and Challenges

In order to allow for insightful analysis in an efficient manner, we decided to create a big dataset containing all the tables from the 3 CSV datasets joined together. This will allow us later on to utilize cassandra's CQL strengths, without the need to JOIN the data. We do this because Cassandra does not support joins or complex subqueries, which are common in SQL and other rational databases, that we were used to working with. We addressed this by normalizing our data, essentially pre-joining data in all of our tables and then doing additional processing using Python. This required a good understanding of our application's query patterns and data structure in Cassandra DB.

A joined schema for the air pollution measurements in Seoul dataset will look as follows:
**Measurement Data Table**
This table stores the air pollution measurements. Queries on this table are likely to be based on date, station code, and item code.

- **Partition Key:** (station_code, item_code)
- **Clustering Key:** measurement_date
- **Columns:**

| | |
|---|---|
| ○ measurement_date (*timestamp*)<br>○ station_code (*int*)<br>○ item_code (*int*)<br>○ average_value (*float*)<br>○ instrument_status (*int*)<br>○ item_name (*text*)<br>○ unit_of_measurement (*text*) | ○ good_blue (*float*)<br>○ normal_green (*float*)<br>○ bad_yellow (*float*)<br>○ very_bad_red (*float*)<br>○ station_name (*text*)<br>○ address (*text*)<br>○ latitude (*float*)<br>○ longitude (*float*) |

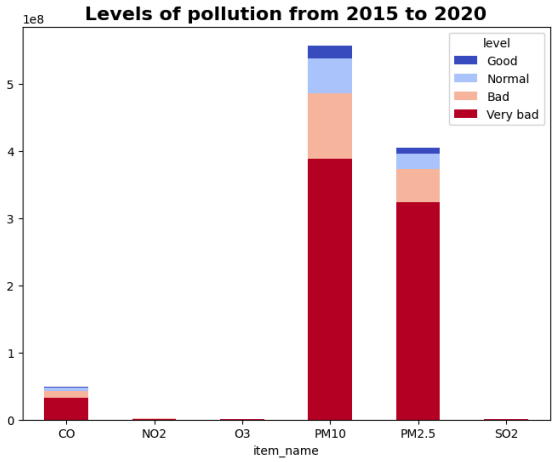
# Data Analysis and Selected Results

| |
|---|
| *Showcase a map with all the locations where a pollutant item was viewed* |

```
SELECT latitude, longitude
FROM measurement_joined
WHERE item_code = 6
ALLOW FILTERING;
```


*Showcase the level of pollution for each item, between 2015 to 2020 relative on the different pollution levels*

```
SELECT item_name, good_blue,
normal_green, bad_yellow,
very_bad_red
FROM measurement_joined
WHERE measurement_date >=
'2015-01-01 00:00:00'
  AND measurement_date <=
'2020-12-31 23:59:59'
  ALLOW FILTERING;
```




*calculate the average average_value pollution Level for a specific pollutant for each station_code in a date limit*

```
SELECT station_code, item_code,
      AVG(average_value)
    FROM measurement_joined
    WHERE item_code = 3 AND
 measurement_date >= '2019-01-01
 00:00:00' AND measurement_date <=
      '2019-12-01 00:00:00'
 GROUP BY station_code, item_code
```