Design Decisions:

First, our TA suggested we expand our question a bit more, and come up with better investigative questions. Here’s our new set of questions:

1. What percentage of a given country’s population has been reported as a confirmed case of Covid-19 since 1/3/2020 for each quarter? What percentage of a given country’s population has been fully vaccinated for each quarter? What’s the quarterly change of these two measurements?
2. How are the vaccination process and daily cases related? Does higher vaccination rate imply less daily cases? How’s the vaccination rate different across different WHO regions?
3. How are public health measures and daily cases related? Does stricter public health measurement imply less daily cases? Which measurement might contribute the most in preventing the spread of Covid-19?

Then, we adjusted our schema according to TA’s comments and our new set of questions:

Country(country_name, Alpha2_code, Alpha3_code)
Population (country_code3, population_2020)
Daily_cases (date, country_code2, new_cases, new_death)
Vaccination (country_code3, WHO_region, total_vacc, people_fully_vaccinated, first_vaccine_date)
PHSM (date, country_code3, masks, travel, gatherings, schools, businesses, movements, total_measurement)

As our TA suggested, we changed the country name in each table into the country code (since we found out that different tables might have different names for the same country). Since the datasets we found used two different codes for the countries, we added another table, Country, as reference. (usl for country code dataset: https://gist.github.com/tadast/8827699#file-countries_codes_and_coordinates-csv). We also changed the awkward attribute names as our TA pointed out so that it’s clearer what information does each column contain. Comments regarding our schema are added in our schema.ddl file.

We added constraints here, since we forgot to put them in our proposal.

Population (country_code3) ⊆ Country(Alpha3_code)
Daily_cases (country_code2) ⊆ Country(Alpha2_code)
Vaccination (country_code3) ⊆ Country(Alpha3_code)
PHSM (country_code3) ⊆ Country(Alpha3_code)

Vaccination (WHO_region) ⊆ {'AFRO', 'AMRO', 'SEARO', 'EURO', 'EMRO', 'WPRO'}
# since these are the six WHO regions

Cleaning Process:

First, we download all the datasets from the resource website using the urls in our phase_1 proposal. Since there are only a few unused column, the null value row and opening description rows, we decide to delete them using Excel:

For Country, we deleted columns Numeric code, Latitude(average), Longitude(average), and removed null value rows and attribute name rows. We trimmed the columns and removed unnecessary quotation marks by using find and replace in Excel. Then, we saved the file as cleaned_country.csv.

For Population, we deleted the columns Country Name, Indicator Name, Indicator Code, and populations from 1960 to 2019. We removed the null value row and any opening description rows, along with the attribute names row. Then, we saved the file as cleaned_pop.csv.

For Daily_cases, we deleted columns Country, Cumulative_cases, and Cumulative_deaths. We removed the null value row and the attribute names row. Then, we saved the file as cleaned_daily.csv.

For Vaccination, we deleted columns Country, Data_source, Date_updated, Persons_vaccinated_1plus_dose, Total_vaccinations_per100, Persons_vaccinated_1plus_dose_per100, Persons_fully_vaccinated_per100, Vaccines_used and Number_vaccines_types_used. We removed the null value row and the attribute names row. Then, we saved the file as cleaned_vacc.csv.

For PHSM (Public Health Social Measurement), we deleted columns Country, WHO_region and Measures_in_place. We removed the null value row and the attribute names row. Then, we saved the file as cleaned_phsm.csv.

After these, we are left with the columns we need, as indicated in our schema.

Then, we also cleaned the data using sql and inserted them at the same time (see also data.sql). We need to check each table so that they satisfy the foreign key constraints. Therefore, we first copy the data from cleaned_country.csv file to Country since other tables need to reference codes that are stored in Country. Next, for each other table, we first create a temporary table with no foreign key constraints and copy the data from csv to these temporary tables. Then, for each one of these temporary tables, check if the country code within the table is also in table Country, if yes then insert the row into the formal table. In this way, we cleaned out all the data that violates the foreign key constraints and inserted the cleaned data into our database. Last, drop all the temporary tables and we are finished with data cleaning and inserting.