Data and Warehouse Visualization

Aaron Schneider

Project Report

Phase 1: Design

1.1 Chose a data source

I have chosen the COVID-19 dataset provided by Our World in Data (OWID), a non-profit organization that compiles and standardizes public health data from official sources such as the World Health Organization (WHO), the European Centre for Disease Prevention and Control (ECDC), and national governments. This dataset is available as a daily updated CSV file from their website: https://ourworldindata.org/covid-cases.

The data source provides a compelling narrative potential, allowing for multi-level analysis of the global pandemic. The dataset's granularity is at the country and date level, making it well-suited for a data warehouse with two distinct hierarchies:

- **Geographical Hierarchy:** Country → Continent → World
- **Temporal Hierarchy:** Date → Week → Month → Quarter → Year

This structure will enable various analytical queries, from tracking new cases and deaths on a daily basis to analyzing cumulative trends across continents or over different quarters. Furthermore, the dataset's historical nature and inclusion of metrics like total deaths and new deaths make it ideal for exploring how major events, such as the rollout of vaccines, may have influenced public health outcomes in different regions.

1.2 Entity-Relationship Model

To represent the logical structure of the COVID-19 dataset, I created an ER model consisting of two main entities: Countries and Daily_Covid_Stats.

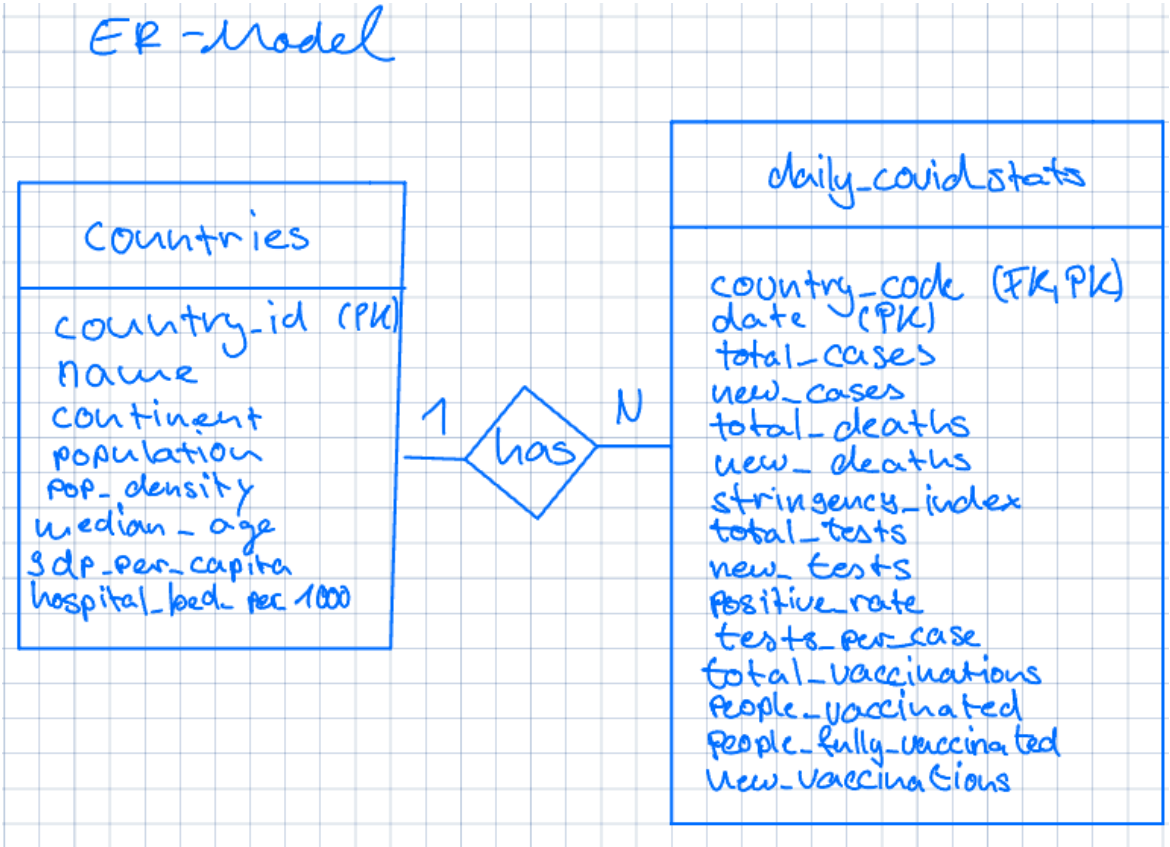
- The **Countries** table stores demographic and geographic data, such as population, GDP per capita, life expectancy, and continent. Each country is uniquely identified by a country_id (primary key).

- The **Daily_Covid_Stats** table contains the daily evolving pandemic metrics like total and new cases, deaths, and vaccination data. This table uses a composite primary key consisting of country_code and date and has a foreign key relationship with the Countries table.

This structure enables efficient time-series analysis at the country level while linking each country to broader regional and demographic attributes.

Cardinality:

One country has many daily records → 1:N relationship between Countries and Daily_Covid_Stats.



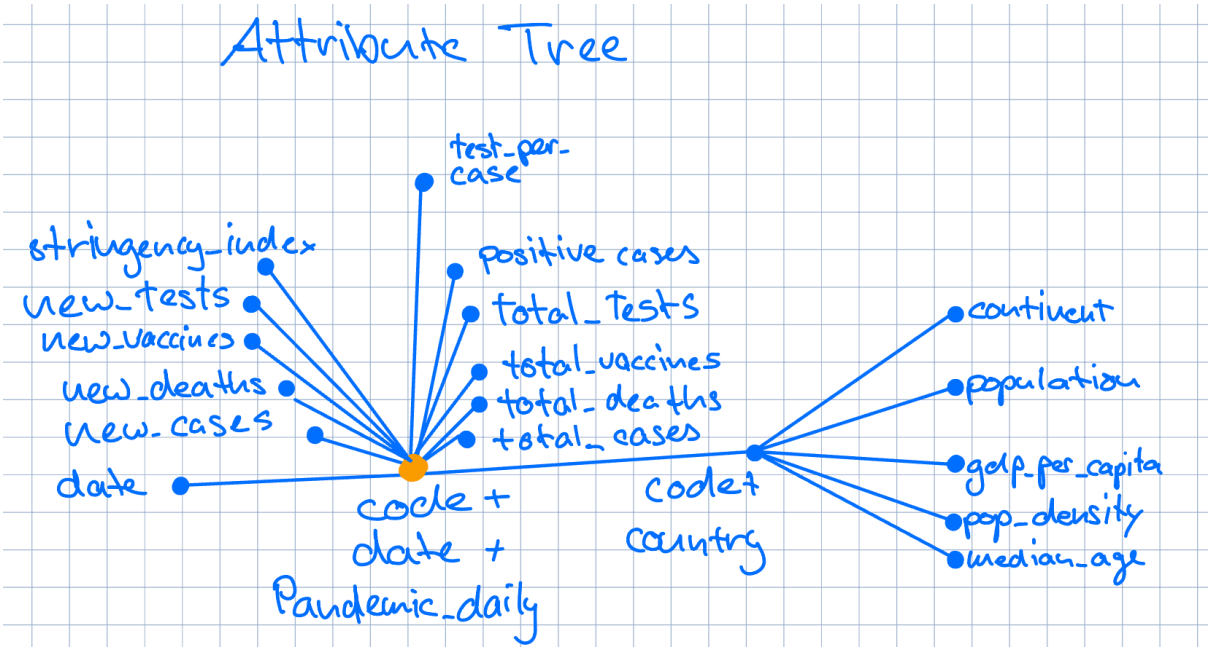
1.3 Attribute Tree

The attribute tree helps visualize the structure of the dataset by distinguishing between **dimensions** and **measures**.

- **Dimensions** include identifiers such as country_id and date, which form the basis for grouping and analyzing data. Temporal hierarchies (e.g., week → month → year) and geographical hierarchies (e.g., country → continent) are also represented.

- **Measures** are quantitative metrics such as new_cases, new_deaths, and stringency_index. These values are recorded per country and per day, and can be aggregated or filtered using the dimension attributes.

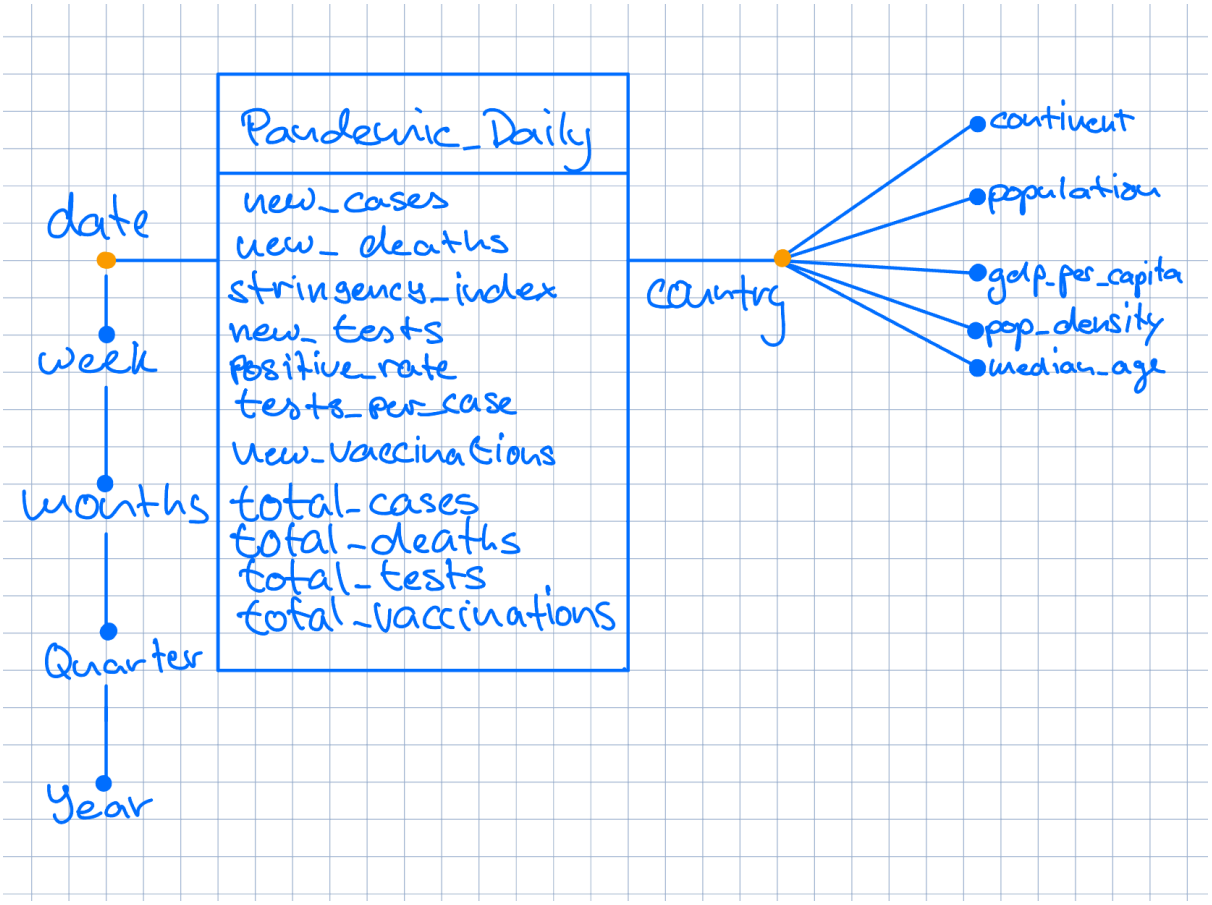
This clear separation of dimensions and measures supports efficient OLAP operations and enables meaningful visualizations and comparisons across different time periods and regions.



1.4 Fact Schema

I implemented a star schema to support OLAP-style analysis, consisting of the following:

- A **fact table**, Pandemic_Daily, which includes all the measurable attributes such as:
    - `new_cases, new_deaths, new_tests, positive_rate, stringency_index, and new_vaccinations`
- Two **dimension tables**:
    - **Date Dimension**, which supports hierarchies: Date → Week → Month → Year. This allows for temporal aggregation at various levels.
    - **Country Dimension**, which holds country-specific attributes (e.g. population, median age, GDP, etc.)



Phase 2: Data Management

2.1 Data Cleaning

To ensure the dataset was suitable for analysis and integration into a data warehouse, several cleaning steps were applied using Python and pandas:

1. **Column Removal**

   Columns with a high percentage of missing values or low analytical relevance (e.g., `human_development_index`, *`icu_patients`*, *`handwashing_facilities`*) were removed to reduce noise and improve performance.

2. **Handling Invalid Countries**

   Rows without a valid ISO country code were removed to exclude aggregates, regions, and invalid entries. Additionally, known aggregates and continents (e.g., "World", "Europe", "Asia") were explicitly filtered out to prevent double-counting and ensure only individual countries and territories remained.

3. **Missing Value Imputation**

   - For all cumulative columns, specifically `total_vaccinations`, `total_cases`, `total_deaths`, `people_vaccinated`, `people_fully_vaccinated`, and *`total_tests`* missing values were handled in a two-step process. First, values were forward-filled within each country using .groupby().ffill() to maintain the continuity of cumulative time series data. Any remaining missing values at the start of a country's time series were then set to zero, reflecting the assumption that no cases, deaths, or vaccinations had been reported up to that point.

- For all other columns, including daily `new_cases`, `new_deaths`, `new_tests`, `stringency_index`, `tests_per_case`, and `positive_rate`, missing values were intentionally left as NaN. This approach preserves the distinction between “no reported event” and “no data available,” ensuring that analyses do not inadvertently misinterpret missing data as zero. Retaining NaN values in these columns also encourages careful handling and transparent reporting of data gaps in subsequent analyses. This strategy was chosen to align with best practices in epidemiological data processing, where cumulative measures can safely be assumed to start at zero, but missing daily or rate data should not be imputed without additional information.

4. **Duplicates & Date Handling**

   Duplicate rows were removed, and the date column was parsed as a proper datetime object. The dataset was then sorted by country and date.

5. **Daily vs. Cumulative Metrics Selection**

   The dataset includes both daily incremental metrics (`new_cases`, `new_deaths`, `new_tests`, `new_vaccinations`) and cumulative totals (`total_cases`, `total_deaths`, `total_tests`, `total_vaccinations`). While cumulative totals could theoretically be calculated by summing daily values, I chose to retain both types for practical and analytical reasons:

   Performance considerations: Pre-calculated cumulative totals eliminate the need for expensive running sum operations across big amount of rows when creating visualizations or performing country comparisons. This is particularly important for Tableau dashboards where real-time aggregation would significantly impact performance.

   Data completeness: The source data (Our World in Data) often has gaps in daily reporting, especially in early pandemic periods or for countries with limited reporting infrastructure. Cumulative totals are

more reliably reported and maintained by health authorities, making them essential for consistent cross-country analysis.

Different analytical purposes: Daily metrics reveal trends and patterns (pandemic waves, policy impacts), while cumulative metrics enable direct comparisons of overall pandemic burden between countries. Each serves distinct visualization needs - daily for time-series analysis, cumulative for ranking and comparative dashboards.

Data integrity: Having both metrics allows for validation and quality checks. Discrepancies between calculated cumulative sums and reported totals can indicate data quality issues or reporting anomalies that require investigation.

NaN Reduction Overview

Before cleaning, the dataset contained a high proportion of missing values across many columns. After these targeted cleaning steps, the percentage of missing data was significantly reduced, improving reliability while still maintaining the original structure and intent of the dataset.

```
NaN percentage before cleaning:
human_development_index       100.000000
life_expectancy               100.000000
icu_patients                   92.205204
hosp_patients                  91.898323
total_boosters                 88.285886
new_vaccinations               86.274815
new_tests                      84.974154
people_fully_vaccinated        84.653722
people_vaccinated              84.255174
total_tests                    84.180247
total_vaccinations             83.416032
tests_per_case                 79.944283
positive_rate                  79.761947
reproduction_rate              62.998234
stringency_index               59.595235
handwashing_facilities         54.817645
hospital_beds_per_thousand     39.536330
extreme_poverty                37.092435
gdp_per_capita                 21.934072
continent                       7.574000
population_density              4.849130
median_age                      4.457357
code                            4.443010
new_cases                       3.555444
new_deaths                      3.403797
population                      3.288616
total_cases                     2.994488
total_deaths                    2.994488
date                            0.000000
country                         0.000000
```

```
NaN percentage after cleaning:
new_vaccinations           88.594788
new_tests                  83.674022
tests_per_case             78.208934
positive_rate              78.010821
reproduction_rate          60.217079
stringency_index           56.099156
gdp_per_capita             17.719044
new_cases                   2.406362
new_deaths                  2.241593
population_density          1.695539
total_tests                 0.000000
total_deaths                0.000000
date                        0.000000
total_vaccinations          0.000000
people_vaccinated           0.000000
people_fully_vaccinated     0.000000
total_cases                 0.000000
code                        0.000000
continent                   0.000000
population                  0.000000
median_age                  0.000000
country                     0.000000
```

2.2 ETL Process

The ETL (Extract, Transform, Load) process was implemented using Python and the pandas library to prepare the cleaned dataset for analysis in Tableau. The goal was to structure the data according to the star schema model, separating dimension tables from fact tables to optimize querying and visualization performance.

2.2.1 Extraction

The file covid_cleaned.csv served as the single source of truth for all subsequent transformations.

```
df = pd.read_csv('covid_cleaned.csv', sep=';')

df['date'] = pd.to_datetime(df['date'], errors='coerce')
```

2.2.2 Transformation

According to our Fact-Schema, the dataset was normalized into three main tables to enable efficient analysis:

1. **dim_country**
   This dimension table contains information about each country, including geographic and demographic attributes such as population, population density, GDP per capita, and median age. The country is uniquely identified using the ISO country code (code), avoiding the need for an artificial surrogate key.

2. **dim_date**
   The date dimension was generated by extracting unique dates from the dataset and enriching them with attributes like year, month, quarter, week, and day. This enables flexible time-based aggregations and filtering in Phase 3: Data Visualization. The original date column was renamed to date_id to serve as a primary key.

3. **fact_covid**
   The fact table contains all relevant numerical metrics such as case numbers, deaths, testing rates, and vaccination data. It was created by joining the cleaned dataset with both dim_country and dim_date to replace textual values with their corresponding dimension keys (code and date_id). This reduces redundancy and aligns with the star schema.

2.2.3 Load

The final step involved exporting the normalized tables into separate CSV files for integration into Tableau or further analysis. These files are:

- dim_country.csv
- dim_date.csv
- fact_covid.csv

Each CSV represents a clean, structured component of the data warehouse. Tableau can load these tables separately and use relationships (via code and date_id) to perform joins and enable powerful visual analytics based on country and time.

The export was done using pandas.DataFrame.to_csv() without index columns, making the files easy to use in other tools.

This ETL pipeline ensures that the data is clean, structured, and optimized for analytical processing and visualization.

Phase 3: Data Visualization

The final phase of this project involved creating interactive dashboards in Tableau to visualize key aspects of the COVID-19 pandemic using the prepared data warehouse. The visualizations were designed to be both informative and easy to interpret, allowing users to explore trends and make comparisons across different regions and time periods.

The dashboards are ordered to tell a cohesive story of the pandemic's progression and impact: starting with the overall scale, moving to global and regional mortality, then examining interventions like vaccinations, and finally allowing for a deep dive into specific country data.

3.1 Dashboard 1: Total Corona Cases

This dashboard provides an initial overview of the pandemic's scale by showing the cumulative total of reported COVID-19 cases over time.

- **"Total Corona Cases" bar chart:** This chart displays the maximum total cases accumulated over quarters from 2020 to 2025. It visually represents the rapid and continuous growth of the pandemic, setting the context for subsequent, more detailed analyses.
    - **OLAP Principles:** This chart supports **Drill-down** on the time hierarchy (Year, Quarter, Month, Day) and allows for **Slice and Dice** operations by selecting one or multiple countries.

3.2 Dashboard 2: Deaths per 100,000

This dashboard focuses on the mortality impact of the pandemic on a country-by-country basis.

- **"Deaths per 100 000" map:** This choropleth map visualizes the total deaths per 100,000 people by country as of April 2025. The gradient of color from light to dark orange indicates the severity of the death rate, enabling easy identification of regions with higher mortality.
    - **OLAP Principles:** This map supports **Drill-down** on the time dimension, allowing the user to observe changes in mortality rates across different periods.
- **"Most corona deaths per 100 000" bar chart:** This bar chart ranks countries by their total deaths per 100,000, providing a detailed

breakdown of the most affected nations. The bars are sorted in descending order, making it simple to identify the countries with the highest per capita death rates.

3.3 Dashboard 3: Regional Comparison

This dashboard allows for a detailed analysis of pandemic metrics by continent, focusing on how mortality evolved over time across different regions.

- **"Heatmap deaths per 100 000 by date and location":** This heatmap visualizes deaths per 100,000 over time, broken down by continent and quarter. The intensity of the blue-to-orange color scale represents the number of deaths, allowing users to quickly spot peak periods and compare the impact of the pandemic across continents in different quarters of the year. For example, Europe and North America show very high death rates in early 2021 and 2022.
    - **OLAP Principles:** This heatmap supports **Drill-down** on both the geographical hierarchy (Continent to Country) and the time hierarchy (Year, Quarter, Month, Day).
- **"Regional Comparison" chart:** This line chart plots `Avg. New Deaths` over time for each continent separately. This allows for a more granular comparison of new death trends across continents, highlighting when and where different regions experienced their peaks.
    - **OLAP Principles:** This chart allows for **Slice and Dice** operations by enabling the user to select specific continents for comparison.

3.4 Dashboard 4: Vaccination Trends

This dashboard provides a global overview of vaccination efforts and their potential correlation with cases and death rates.

- **"Vaccination around the world" map:** A choropleth map shows the average vaccination rate per country as of January 2023. The color intensity represents the vaccination rate, providing a quick visual comparison of vaccination progress globally.
    - **OLAP Principles:** This map supports **Drill-down** on the time dimension to observe vaccination rate changes over different periods.

- **"Vaccination vs Death Timeline" chart:** This multi-line chart plots `Avg. Vaccination Rate` (green line) and `New Deaths` (red line) over time. An additional line for `New Cases` (orange line) is included to provide further context. This chart visually suggests a correlation between the increase in vaccination rates and the subsequent decline in new cases and deaths, allowing for a high-level analysis of the pandemic's trajectory.

3.5 Dashboard 5: Country Deepdive

This dashboard offers a granular view of key pandemic metrics for a single selected country, allowing for detailed analysis of its specific situation over time.

- **"Deepdive on Country" multi-chart:** This dashboard presents four individual charts for a chosen country:
    - **"New Cases" line chart:** Shows the trend of new COVID-19 cases over time, quarter by quarter.
    - **"New Deaths" line chart:** Displays the trend of new deaths related to COVID-19 over time, quarter by quarter.
    - **"New Vaccinations" bar chart:** Illustrates the number of new vaccinations administered per quarter.
    - **"Total Tests" bar chart:** Shows the cumulative total of tests conducted over time, quarter by quarter. This deepdive allows users to examine the interplay of these critical metrics within a specific national context.
    - **OLAP Principles:** This dashboard primarily utilizes **Slice** functionality, allowing the user to select a single country for detailed analysis.