SOEN 363 - Final Report

SOEN 363
Project Phase I

Team Members

Baraa Chrit 40225403

Jad Hanna 40132590

Mahmoud Mohamed 40163777

Mostafa Mohamed 40201893

Project Title: COVID-19 Vaccine and Disease Analytics

Instructor: Ali Jannatpour

Winter 2025
Due Date: April 4, 2025

1. System Overview

This project's objective is to create and deploy a relational database system. This database will combine public health information from two different sources.

Sources:

- Vaccination metadata from the OpenFDA API.
- Real-time COVID-19 statistics from the Disease.sh API.

Our system makes meaningful analysis of the connections between vaccination distribution and disease indicators across many countries.

In addition, the database includes brand name, type, route of administration, manufacturer, ingredients, dosage information, and adverse reactions. It also collects data on total cases, deaths, recoveries, and testing rates for each country. So, by linking both data sources by country identifier (country name or ISO code), our system makes Insightful analysis and reporting possible by cross-reference vaccination usage with disease impact.

Our system is built using PostgreSQL, and populated using Python script that handles API data fetching, cleaning, transformation, and insertion into the relational schema. It also supports important relational features like IS-A relationships, weak entities, aggregation, views with access control, and is designed to follow normalization principles.

And this data will be migrated and examined using a NoSQL paradigm in Phase II of the project, which is built upon this database. By now, the relational database can already handle a large range of complex SQL queries, for example joins, subqueries, groups, and set operations. This allows for a better analysis of the relationships between pharmaceutical data and public health data.

2. Data Sources

The project integrates data from the OpenFDA Drug NDC API and the Disease.sh API, which are publicly accessible APIs. Using country identifiers as a common connection between both of them, each of the API provides free data that is used to complete different relational database components.

2.1 Disease.sh API

The Disease.sh API provides up-to-date COVID-19 statistics for countries around the world. For each country, the API returns information such as:

- Total confirmed cases
- Daily new cases
- Deaths and daily deaths
- Recoveries and daily recoveries
- Active and critical cases
- Testing counts and per-million statistics
- Country population
- ISO country code

This data is used to populate two tables:

- `Country` — stores general information like name, ISO code, population, and last update timestamp.
- `DiseaseStats` — stores detailed disease-related metrics for each country.

Each entry in `DiseaseStats` references a corresponding country in the `Country` table via a foreign key relationship.

2.2 OpenFDA Vaccine API

The OpenFDA Drug NDC API provides structured metadata about vaccines available in the U.S. pharmaceutical registry. For each vaccine product, the API returns fields such as:

- National Drug Code (NDC)
- Brand and generic names
- Route of administration (e.g., injection, oral)
- Product type (e.g., "VACCINE")
- Manufacturer details

- Active and inactive ingredients
- Dosage form
- UNIIs and UPCs
- Optional warning text

This data is used to populate multiple tables:

- `Vaccine` — core vaccine metadata.
- `Ingredient` — active and inactive components of the vaccine.
- `DosageInfo` — dosage description (used to model an IS-A relationship).
- `Warning` — text-based safety notices tied to each vaccine (weak entity).

Furthermore, the CountryVaccineUsage database connects vaccinations to countries using gathered country information, that is gotten from company names or addresses. And when this data is missing, then vaccines are randomly assigned to 1-5 countries and labeled with usage feedback to replicate real-world distribution patterns.

2.3 Data Integration

The integration of the two data sources is done using country names or ISO 2-letter codes. During processing, the system creates a mapping of country_name → country_id from Disease.sh data. And this mapping connects vaccination entries (from OpenFDA) to the associated countries in the Country table, and this will allow for meaningful relationships and queries across the two datasets.

3. Schema Design

Our database schema is designed to represent and normalize data from both the Disease.sh and OpenFDA APIs. It supports complex relationships such as IS-A hierarchies, weak entities, and many-to-many associations. In addition, our design ensures consistency, scalability, and extensibility for future data searches and migrations.

3.1 Overview of Tables

Table Name	Purpose
`Country`	Stores basic country information including name, code, population, and last update time.
`DiseaseStats`	Stores COVID-19-related statistics for each country, including case numbers, recoveries, and testing rates.
`Vaccine`	Stores core vaccine metadata such as brand name, generic name, route of administration, and manufacturer.
`Ingredient`	Stores both active and inactive components of a vaccine.
`DosageInfo`	Stores textual dosage information for vaccines. Demonstrates an IS-A relationship with the `Vaccine` table.
`Warning`	Stores safety warnings related to vaccines. Designed as a weak entity that depends on the vaccine's existence.
`CountryVaccineUsage`	Aggregates which vaccines are used in which countries, supporting a many-to-many relationship. Includes optional usage notes.

3.2 Relationships

- **One-to-Many**:
    - `Country` → `DiseaseStats`: A country can have multiple disease stat entries (though currently 1:1 is used).
    - `Vaccine` → `Ingredient`, `Warning`: A vaccine can have many ingredients and warnings.
- **IS-A Relationship**:
    - `DosageInfo` shares the same primary key (`vaccine_ndc`) as `Vaccine`. It models a subtype of vaccines that have specific dosage details.
- **Weak Entity**:
    - `Warning` uses a composite primary key of `id` and `vaccine_ndc`, relying on the `Vaccine` table for full identification and referential integrity.
- **Aggregation**:
    - `CountryVaccineUsage` aggregates `Country` and `Vaccine`, modeling their many-to-many usage relationship. This table includes descriptive notes on vaccine usage per region.

3.3 Domains and Custom Types

- `dosage_text`: A domain that enforces a maximum length of 1000 characters for dosage entries.
- `warning_text`: A domain for storing warning messages.
- `route_enum`: A custom enum type that limits the route of administration to predefined values such as `'INJECTION'`, `'ORAL'`, `'TOPICAL'`, etc.

3.4 Views

Two views are implemented to simulate role-based access control:

- `VaccinePublicView`: Shows limited vaccine fields (e.g., brand, generic name, and route).

- `VaccineFullView`: Displays all columns from the `Vaccine` table.

4. Challenges Faced

1) Mismatch Between Data Sources.
    - Disease.sh and OpenFDA, they use different formats for country names, which is making it hard to connect the two. We had to match them manually or assign random countries when needed.

2) Missing or Incomplete Data.
    - Some of the vaccines were missing important information like route or ingredients. So, we had to clean the data and use default values when possible.

3) API Rate Limits.
    - The OpenFDA API limited how many requests we could make at once. We added delays between requests to avoid getting blocked.