**COSC 580 Project 2**

Group Member: Yuming Wang, Ziang He, Jiancheng Sun

Group Number: 6

1. **Source Identification**

For this project, we used three datasets to build our Database System:

* Dataset\_0: [United States COVID-19 Cases, Deaths, and Laboratory Testing (NAATs) by State, Territory, and Jurisdiction](https://covid.cdc.gov/covid-data-tracker/#cases_casesper100klast7days)
* Dataset\_1: [Data Table for Vaccinations Equity (SVI)](https://covid.cdc.gov/covid-data-tracker/#vaccination-equity)
* Dataset\_2: [Data Table for Vaccinations Equity (Metro/Non-Metro)](https://covid.cdc.gov/covid-data-tracker/#vaccination-equity)
  1. **Dataset\_0**

This dataset shows the number of COVID-19 ***cases, deaths and laboratory testing*** for every 100,000 people over the last 7 days, allowing us to compare areas with different population sizes.

* 1. **Dataset\_1 and Dataset\_2**

These two datasets provide a county-level view of COVID-19 vaccination coverage, social vulnerability and Metropolitan vs. Non-Metropolitan:

***Social vulnerability*** is measured by CDC Social Vulnerability Index (SVI), which uses U.S. Census data on categories like poverty, housing, and vehicle access to estimate a community’s ability to respond to and recover from disasters or disease outbreaks.

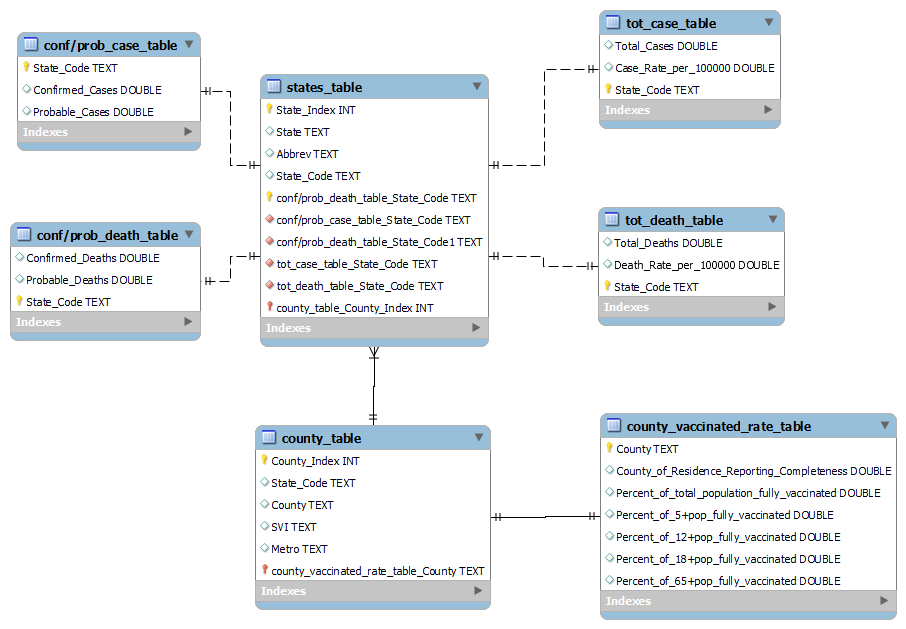
***Metropolitan vs. Non-Metropolitan*** classification is based off an aggregation of the six 2013 National Center for Health Statistics (NCHS) Urban-Rural classifications, where “Metro” counties include Large Central Metro, Large Fringe Metro, Medium Metro, and Small Metro and “Non-Metro” counties include Micropolitan and Non-Core (Rural).

* 1. **Data Preprocessing and Table Building**

For these datasets, we do some preprocessing to build our own database.

* For these datasets, we do some preprocessing to build our database. Although the datasets we collected contain ***regional information***, such as state names, some are abbreviated, and some are full. For this problem, we matched the abbreviation of the state name with the full name of the state name and constructed the “***states\_table***”.
* In addition, for the data of “Total\_Cases”, “Total\_Deaths”, “Confirmed\_Deaths”, “Probable\_Deaths”, “Confirmed\_Cases”, “Probable\_Cases” of each state in Dataset\_0, we have established different tables to store them to meet the requirements of ***3NF***. They all use "State\_Code" as the ***primary key***.
* The data in Dataset\_1 and Dataset\_2 ***overlap with multiple attributes***, which provide a county-level view of COVID-19 vaccination coverage from social vulnerability and Metropolitan vs. Non-Metropolitan, respectively. We extracted "SVI", "Metro", "County" and "State\_Code" from these two datasets and defined the “***County\_Index***” as the primary key to build the“***county\_table***”.
* In addition, for the ***overlapping parts*** of the two data: the proportion of fully vaccinated people in different age groups and the residence reporting completeness of each county, we extracted and constructed the“***county\_vaccinated\_rate\_table***”, and used “***county***” as the primary key.

1. **Schema Definition**



We used ***Entity Relationship Diagram (ERD)*** to present the relations and relationships. Every relation is in ***3NF***, which means every non-key attribute is dependent on the key

and nothing but the key.

1. **Functional Dependencies**

**3.1. county\_table**

|  |
| --- |
|  |
|  |

* 1. **states\_table**

|  |
| --- |
|  |
|  |

* 1. **county\_vaccinated\_rate\_table**

|  |
| --- |
|  |
|  |

* 1. **tot\_case\_table**

|  |
| --- |
|  |
|  |

* 1. **tot\_death\_table**

|  |
| --- |
|  |
|  |

* 1. **conf/prob\_case\_table**

|  |
| --- |
|  |
|  |

* 1. **conf/prob\_death\_table**

|  |
| --- |
|  |
|  |

1. **Canned Queries**

In this database, we can support some query functions.

Some examples of retrieving some ***extreme values***are given below:

Eg1:

|  |  |
| --- | --- |
| Query | select \* from tot\_case\_table where Total\_Cases = (select max(Total\_Cases) from tot\_case\_table); |
| Result |  |

Eg2:

|  |  |
| --- | --- |
| Query | select \* from tot\_case\_table where Total\_Cases = (select min(Total\_Cases) from tot\_case\_table); |
| Result |  |

Eg3:

|  |  |
| --- | --- |
| Query | select\*from tot\_death\_table where Total\_Deaths = (select max(Total\_Deaths) from tot\_death\_table); |
| Result |  |

Eg4:

|  |  |
| --- | --- |
| Query | select\*from tot\_death\_table where Total\_Deaths = (select min(Total\_Deaths) from tot\_death\_table); |
| Result |  |

Some examples of ***range searches*** are given below:

Eg5:

|  |  |
| --- | --- |
| Query | SELECT State\_Code, Case\_Rate\_per\_100000 FROM tot\_case\_table  order by Case\_Rate\_per\_100000 desc  limit 10; |
| Result |  |

Eg6:

|  |  |
| --- | --- |
| Query | SELECT State\_Code, Case\_Rate\_per\_100000 FROM tot\_case\_table  where Case\_Rate\_per\_100000 > 27000; |
| Result |  |

Eg7:

|  |  |
| --- | --- |
| Query | SELECT State\_Code, Death\_Rate\_per\_100000 FROM tot\_death\_table  where Death\_Rate\_per\_100000 > 350; |
| Result |  |

Here are some examples of ***sorting by value***:

Eg8:

|  |  |
| --- | --- |
| Query | select State\_Code, (count(case when SVI='High' then State\_Code end)/count(State\_Code)) as rate from county\_table  group by State\_Code  order by rate desc  limit 10; |
| Result |  |

Eg9:

|  |  |
| --- | --- |
| Query | select State\_Code,rate  from(select State\_Code, (count(case when SVI='High' then State\_Code end)/count(State\_Code)) as rate from county\_table  group by State\_Code  order by rate desc) as a  where rate > 0.5; |
| Result |  |

Eg10:

|  |  |
| --- | --- |
| Query | select State\_Code,rate  from(select State\_Code, (count(case when SVI='High' then State\_Code end)/count(State\_Code)) as rate from county\_table  group by State\_Code  order by rate desc) as a  where rate > 0.3; |
| Result |  |

Eg11:

|  |  |
| --- | --- |
| Query | select State\_Code,Total\_Deaths/Total\_Cases as death\_rate  from tot\_case\_table join tot\_death\_table using(State\_Code)  order by death\_rate desc  limit 10; |
| Result |  |

Eg12:

|  |  |
| --- | --- |
| Query | select Metro,SVI,count(\*) as group\_number from county\_table  group by Metro,SVI; |
| Result |  |