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
- Dataset_1: Data Table for Vaccinations Equity (SVI)
- Dataset_2: Data Table for Vaccinations Equity (Metro/Non-Metro)

1.1. 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.

1.2. 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).

1.3. Data Preprocessing and Table Building

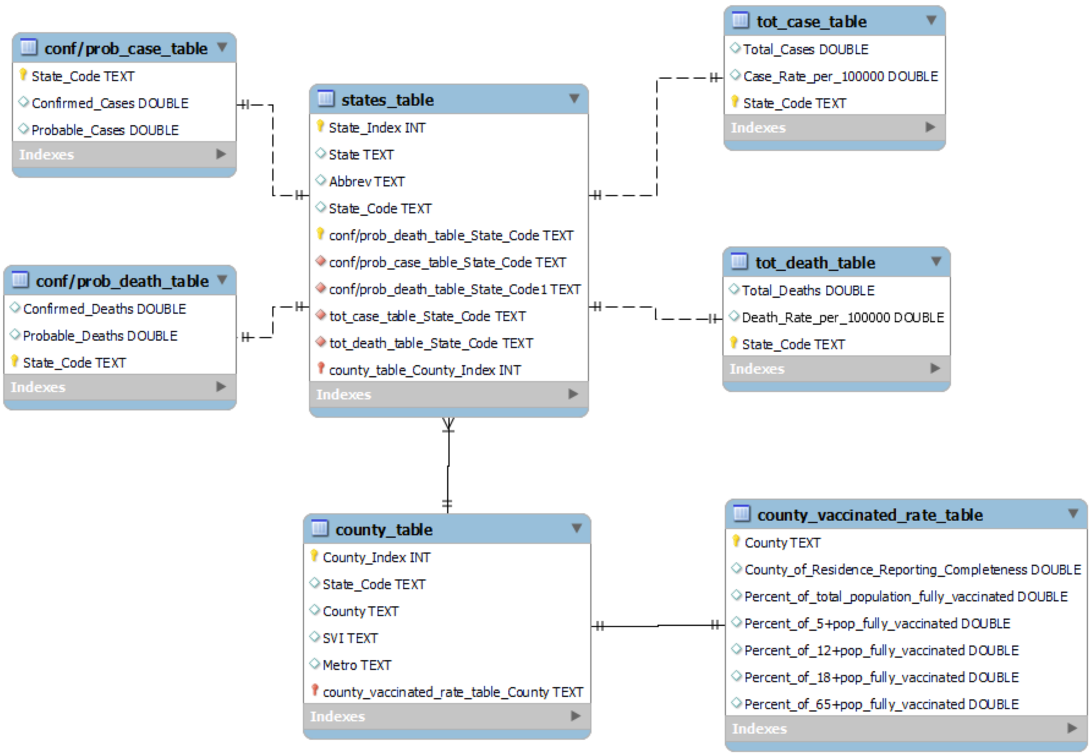
For these datasets, we do some preprocessing to build our own database.

- 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.

2. Schema Definition

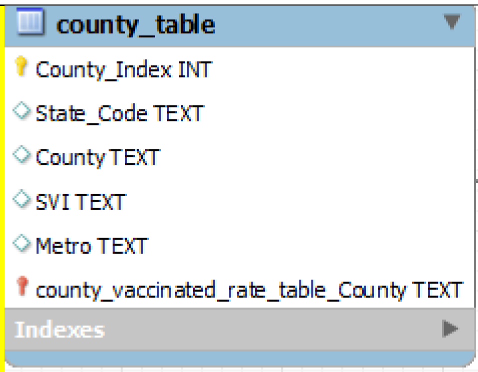


We used ***Entity Relationship Diagram (ERD)*** to present the relations and relationships. Every relation is in ***3NF***, which means every non-key attribute is dependent on the key and nothing but the key.

3. Functional Dependencies

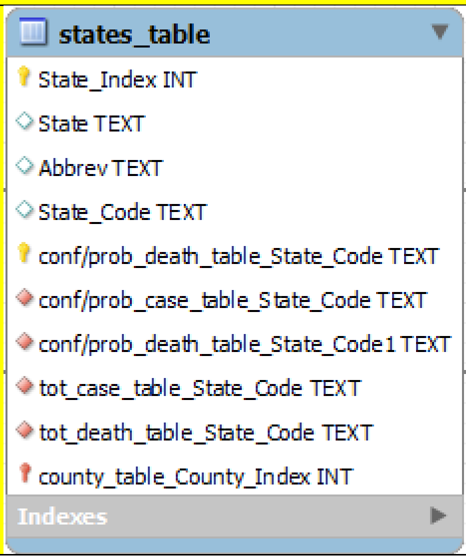
3.1. county_table

$$\boxed{FD:\{County\_Index\# \rightarrow State\_Code, County, SVI, Metro\}}$$



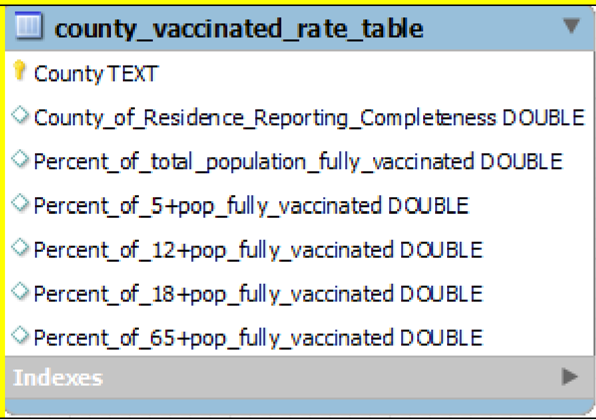
3.2. states_table

$FD:\{State\_Index\# \rightarrow State\_Code, Abbrev, State\}$



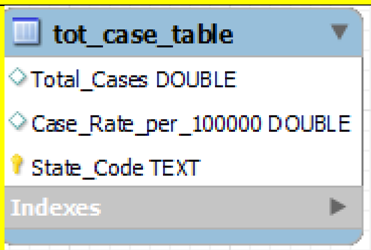
3.3. county_vaccinated_rate_table

$FD:\{County\# \rightarrow \text{County\_of\_Residence\_Reporting\_Completeness}, \text{Percent\_of\_total\_population\_fully\_vaccinated}, \text{Percent\_of\_5+pop\_fully\_vaccinated}, \text{Percent\_of\_12+pop\_fully\_vaccinated}, \text{Percent\_of\_18+pop\_fully\_vaccinated}, \text{Percent\_of\_65+pop\_fully\_vaccinated}\}$



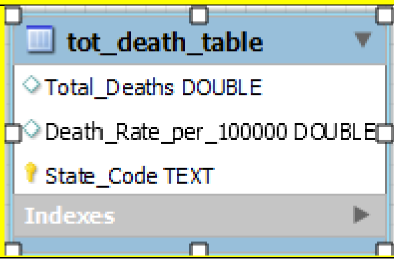
3.4. tot_case_table

$FD:\{State\_Code\# \rightarrow \text{Total\_Cases}, \text{Case\_Rate\_per\_100000}\}$



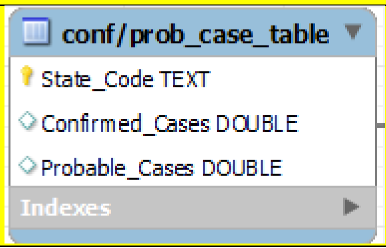
3.5. tot_death_table

$FD:\{State\_Code\# \rightarrow \text{Total\_Deaths}, \text{Death\_Rate\_per\_100000}\}$



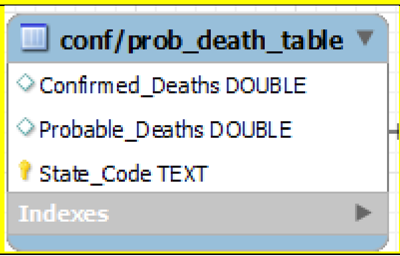
3.6. conf/prob_case_table

$FD:\{State\_Code\# \rightarrow \text{Confirmed\_Caes}, Probable\_Cases\}$



3.7. conf/prob_death_table

$FD:\{State\_Code\# \rightarrow \text{Confirmed\_Deaths}, Probable\_Deaths\}$



4. 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	<table><tr><th>State_Code</th><th>Total_Cases</th><th>Case_Rate_per_100000</th></tr><tr><td>CA</td><td>9015587</td><td>22817</td></tr></table>

Eg2:

Query	select * from tot_case_table where Total_Cases = (select min(Total_Cases) from tot_case_table);
Result	<table><tr><th>State_Code</th><th>Total_Cases</th><th>Case_Rate_per_100000</th></tr><tr><td>VT</td><td>105475</td><td>16903</td></tr></table>

Eg3:

Query	select*from tot_death_table where Total_Deaths = (select max(Total_Deaths) from tot_death_table);
Result	<table><tr><th>State_Code</th><th>Total_Deaths</th><th>Death_Rate_per_100000</th></tr><tr><td>CA</td><td>86185</td><td>218</td></tr></table>

Eg4:

Query	select*from tot_death_table where Total_Deaths = (select min(Total_Deaths) from tot_death_table);
Result	<table><tr><th>State_Code</th><th>Total_Deaths</th><th>Death_Rate_per_100000</th></tr><tr><td>VT</td><td>578</td><td>92</td></tr></table>

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;

<table>
<tr><td>Result</td><td><table><tr><th>State_Code</th><th>Case_Rate_per_100000</th></tr><tr><td>AK</td><td>32105</td></tr><tr><td>RI</td><td>32086</td></tr><tr><td>ND</td><td>31368</td></tr><tr><td>TN</td><td>29510</td></tr><tr><td>KY</td><td>28979</td></tr><tr><td>UT</td><td>28861</td></tr><tr><td>SC</td><td>28444</td></tr><tr><td>WV</td><td>27587</td></tr><tr><td>AR</td><td>27337</td></tr><tr><td>AZ</td><td>27303</td></tr></table></td></tr>
</table>

Eg6:

<table>
<tr><td>Query</td><td>SELECT State_Code, Case_Rate_per_100000 FROM tot_case_table
where Case_Rate_per_100000 > 27000;</td></tr>
<tr><td>Result</td><td><table><tr><th>State_Code</th><th>Case_Rate_per_100000</th></tr><tr><td>AK</td><td>32105</td></tr><tr><td>AZ</td><td>27303</td></tr><tr><td>AR</td><td>27337</td></tr><tr><td>FL</td><td>27120</td></tr><tr><td>KY</td><td>28979</td></tr><tr><td>ND</td><td>31368</td></tr><tr><td>RI</td><td>32086</td></tr><tr><td>SC</td><td>28444</td></tr><tr><td>TN</td><td>29510</td></tr><tr><td>UT</td><td>28861</td></tr><tr><td>WV</td><td>27587</td></tr><tr><td>WI</td><td>27088</td></tr></table></td></tr>
</table>

Eg7:

<table>
<tr><td>Query</td><td>SELECT State_Code, Death_Rate_per_100000 FROM tot_death_table
where Death_Rate_per_100000 > 350;</td></tr>
<tr><td>Result</td><td><table><tr><th>State_Code</th><th>Death_Rate_per_100000</th></tr><tr><td>AL</td><td>384</td></tr><tr><td>AZ</td><td>385</td></tr><tr><td>AR</td><td>359</td></tr><tr><td>LA</td><td>362</td></tr><tr><td>MI</td><td>352</td></tr><tr><td>MS</td><td>411</td></tr><tr><td>NJ</td><td>372</td></tr><tr><td>TN</td><td>363</td></tr><tr><td>WV</td><td>365</td></tr></table></td></tr>
</table>

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

Eg8:

<table>
<tr><td>Query</td><td>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;</td></tr>
</table>

Result	

State_Code	rate
AZ	0.7333
LA	0.7031
MS	0.6707
NM	0.6364
GA	0.5786
SC	0.5435
AR	0.5333
OK	0.4805
AL	0.4478
NC	0.4400

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	

State_Code	rate
AZ	0.7333
LA	0.7031
MS	0.6707
NM	0.6364
GA	0.5786
SC	0.5435
AR	0.5333

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

State_Code	rate
AZ	0.7333
LA	0.7031
MS	0.6707
NM	0.6364
GA	0.5786
SC	0.5435
AR	0.5333
OK	0.4805
AL	0.4478
NC	0.4400
FL	0.4328
TX	0.4291
CA	0.3966
KY	0.3583

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

State_Code	death_rate
PA	0.015815759965569366
MS	0.015457743323458990
NJ	0.015175476921269441
MI	0.014831186743799074
AL	0.014609718348648763
GA	0.014594692200207716
CT	0.014553975161581344
LA	0.014462994222350310
NV	0.014392601664567291
MD	0.014151215332653455

Eg12:

Query

```
select Metro,SVI,count(*) as group_number from county_table
group by Metro,SVI;
```

Result

Metro	SVI	group_number
Metro	Low-Mod	344
Metro	Low	341
Non-metro	High	584
Metro	Mod-High	329
Metro	High	221
Non-metro	Mod-High	475
Non-metro	Low-Mod	459
		2
Non-metro	Low	465
	Low-Mod	1
Non-metro		1