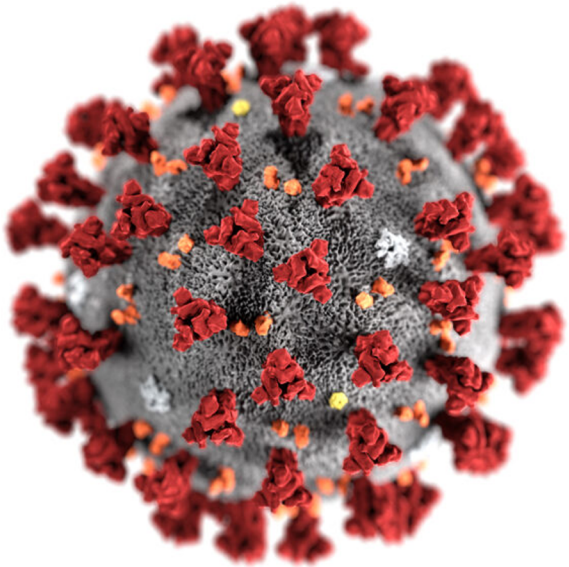
State-Based COVID-19 Vaccination Allocation Database Report

Justin Nichols
Andrew Kimball
Michael Gee
Cameron Gamble



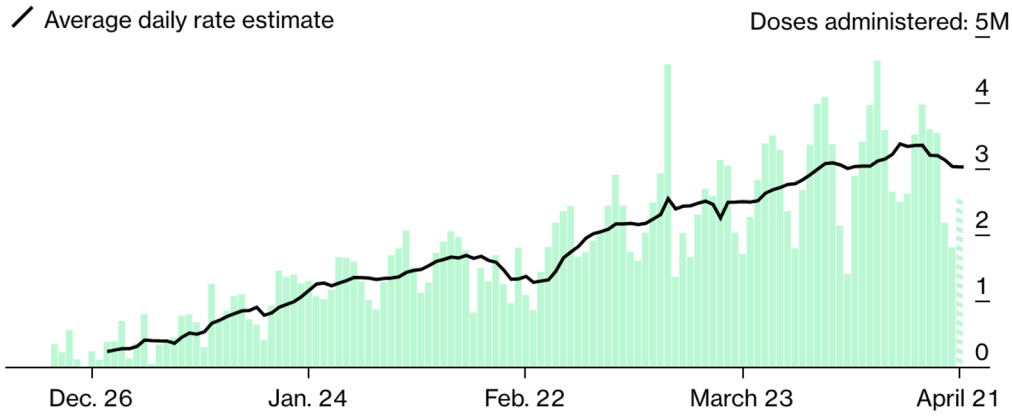
https://www.statnews.com/2020/02/11/disease-caused-by-the-novel-coronavirus-has-name-covid-19/

Justin

Goal and Motivation

The ultimate goal of this database application is to gather the current vaccination allocation, case count, and population of each U.S. state/territory to develop observations and make conclusions about the COVID-19 vaccine distribution around the country. The database gathers the number of allocated vaccines from the approved companies (Pfizer, Moderna, and Janssen) given to each state and compares it to the state's population and total COVID-19 case count. Our group wanted to pick a topic that was not only relevant but also had a multitude of data to gather and obtain results from. COVID-19 was an obvious choice for us, considering how much it has impacted the country in the last year.

Note: This project was started before the Janssen (Johnson & Johnson) vaccine was indefinitely suspended and does not reflect last week's development.

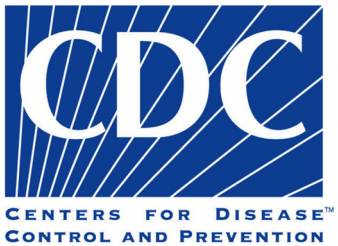


https://www.bloomberg.com/graphics/covid-vaccine-tracker-global-distribution/

Justin

Raw Data

All data was gathered in a csv format during the first week of April 2021. The case count data for each state was obtained from https://www.cdc.gov/. The population data for each state was obtained from https://www.census.gov/. The vaccine data corresponding to Moderna, Pfizer, and Janssen was obtained from https://www.data.gov/. The final raw data set included, Moderna.csv (1008 total entries), Pfizer.csv (1071 total entries), Janssen.csv (315 total entries), Population.csv (59 total entries), Cases.csv (59 total entries).



Justin

Tables

```
MariaDB [cs440238]> describe total_cases_per_state;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| state       | varchar(20) | NO   | PRI | NULL    |       |
| total_cases | int(11)     | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
2 rows in set (0.002 sec)
```

Primary Key: state

Functional Depencies: state → total_cases

Normal Form: The table schema is in BCNF.

```
MariaDB [cs440238]> describe total_population_per_state;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| state           | varchar(20) | NO   | PRI | NULL    |       |
| total_population | int(11)     | NO   |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
2 rows in set (0.004 sec)
```

Primary Key: state

Functional Depencies: state → total_population

Normal Form: The table schema is in BCNF.

Andrew

Tables (cont.)

```
MariaDB [cs440238]> describe total_vaccines_per_state;
+---------------+------------+------+-----+---------+-------+
| Field         | Type       | Null | Key | Default | Extra |
+---------------+------------+------+-----+---------+-------+
| state         | varchar(20) | NO   | PRI | NULL    |       |
| total_vaccines | int(11)    | NO   |     | NULL    |       |
+---------------+------------+------+-----+---------+-------+
2 rows in set (0.011 sec)
```

Primary Key: state

Functional Depencies: state → total_vaccines

Normal Form: The table schema is in BCNF.

```
MariaDB [cs440238]> describe moderna;
+-------------+------------+------+-----+---------+-------+
| Field       | Type       | Null | Key | Default | Extra |
+-------------+------------+------+-----+---------+-------+
| state       | varchar(20) | NO   | PRI | NULL    |       |
| week        | varchar(10) | NO   | PRI | NULL    |       |
| first_dose  | int(11)    | YES  |     | NULL    |       |
| second_dose | int(11)    | YES  |     | NULL    |       |
+-------------+------------+------+-----+---------+-------+
4 rows in set (0.002 sec)
```

Primary Key: state, week

Functional Depencies: state, week → first_dose;
state, week → second_dose

Normal Form: The table schema is in BCNF.

Andrew

Tables (cont.)

```
MariaDB [cs440238]> describe pfizer;
+-------------+------------+------+-----+---------+-------+
| Field       | Type       | Null | Key | Default | Extra |
+-------------+------------+------+-----+---------+-------+
| state       | varchar(20) | NO   | PRI | NULL    |       |
| week        | varchar(10) | NO   | PRI | NULL    |       |
| first_dose  | int(11)     | YES  |     | NULL    |       |
| second_dose | int(11)     | YES  |     | NULL    |       |
+-------------+------------+------+-----+---------+-------+
4 rows in set (0.003 sec)
```

Primary Key: state, week

Functional Depencies: state, week → first_dose;
state, week → second_dose

Normal Form: The table schema is in BCNF.

```
MariaDB [cs440238]> describe janssen;
+-----------+------------+------+-----+---------+-------+
| Field     | Type       | Null | Key | Default | Extra |
+-----------+------------+------+-----+---------+-------+
| state     | varchar(20) | NO   | PRI | NULL    |       |
| week      | varchar(10) | NO   | PRI | NULL    |       |
| only_dose | int(11)     | YES  |     | NULL    |       |
+-----------+------------+------+-----+---------+-------+
3 rows in set (0.002 sec)
```

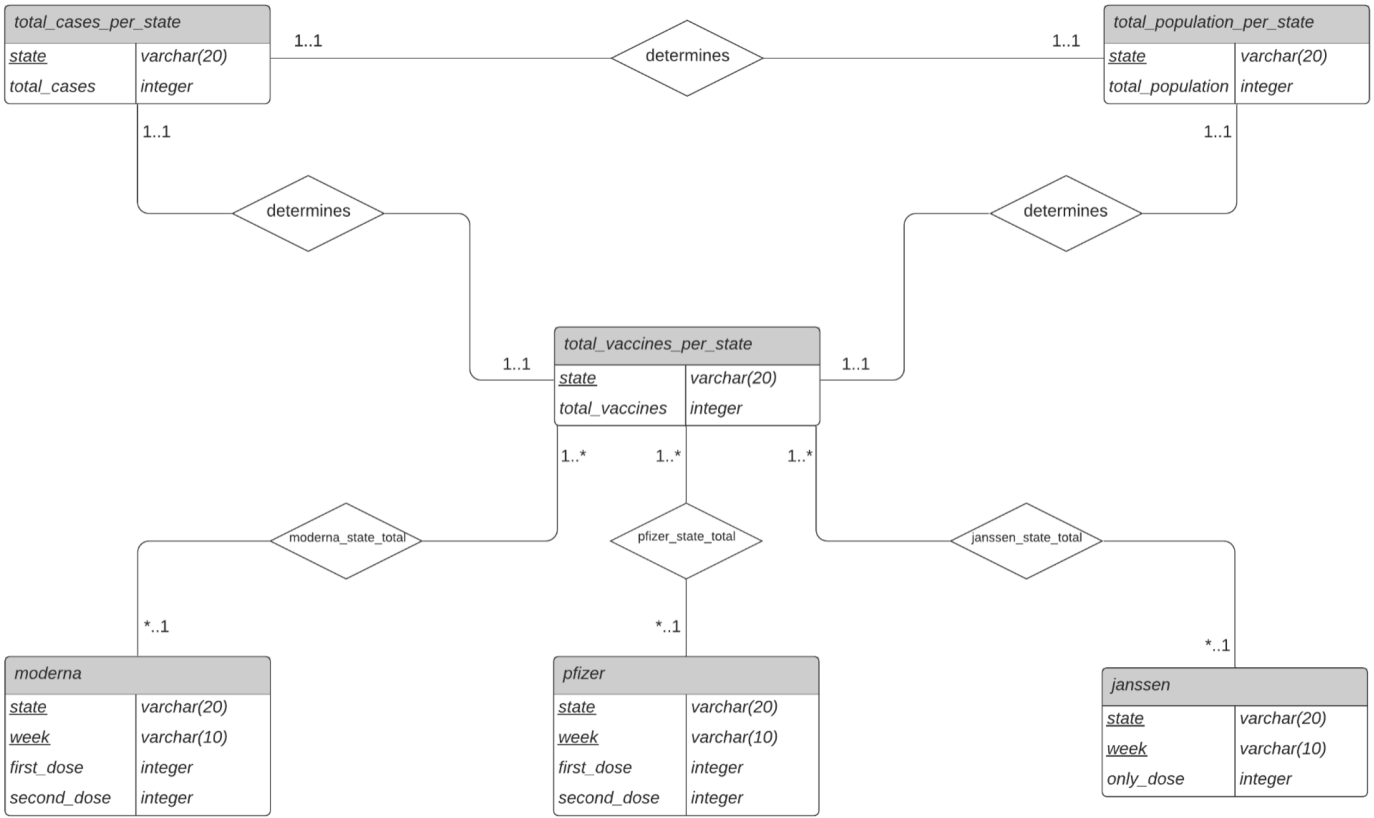
Primary Key: state, week

Functional Depencies: state, week → only_dose

Normal Form: The table schema is in BCNF.

Andrew

ER Diagram



Andrew

Query 1

This query finds the percentage of vaccines given to a state compared to its population and outputs the top ten states with the highest percentage.

```
select v.state, (total_vaccines / total_population) as vaccine_to_population_ratio
from total_vaccines_per_state v inner join total_population_per_state p
on v.state = p.state order by vaccine_to_population_ratio desc limit 10;
```

Result:

state	vaccine_to_population_ratio
Alaska	0.6058
U.S. Virgin Islands	0.4188
Vermont	0.3668
Puerto Rico	0.3646
District of Columbia	0.3564
West Virginia	0.3506
Maine	0.3484
Rhode Island	0.3483
New Hampshire	0.3445
Hawaii	0.3443

Michael

Query 2

This query finds the percent of each state's cases compared to its population and outputs the top ten states with the highest percentage.

```
select c.state, (total_cases / total_population) as case_to_population_ratio
from total_cases_per_state c inner join total_population_per_state p
on c.state = p.state order by case_to_population_ratio desc limit 10;
```

Result:

state	case_to_population_ratio
North Dakota	0.1350
South Dakota	0.1328
Rhode Island	0.1291
Utah	0.1201
Tennessee	0.1187
Arizona	0.1156
Iowa	0.1112
Oklahoma	0.1107
Arkansas	0.1094
Wisconsin	0.1091

Michael

Query 3

This query finds the percentage of vaccines given to a state compared to its cases and outputs the top ten states with the highest ratio.

```
select c.state, (total_vaccines / total_cases) as vaccine_to_cases_ratio
from total_cases_per_state c inner join total_vaccines_per_state v
on c.state = v.state order by vaccine_to_cases_ratio desc limit 10;
```

Result:

state	vaccine_to_cases_ratio
Marshall Islands	2700.0000
Hawaii	15.9033
U.S. Virgin Islands	15.3629
Vermont	11.9789
Puerto Rico	10.8707
Maine	9.3182
Oregon	8.4289
Alaska	7.0573
Washington	6.7883
District of Columbia	5.6638

Michael

Query 4

This query finds the total number of Moderna, Pfizer, and Janssen vaccines distributed to fully vaccinate a person.

```
select * from (
        (select sum(second_dose) as moderna from moderna) as t1,
        (select sum(second_dose) as pfizer from pfizer) as t2,
        (select sum(only_dose) as janssen from janssen) as t3
);
```

Result:

moderna	pfizer	janssen
49711600	53733030	10575700

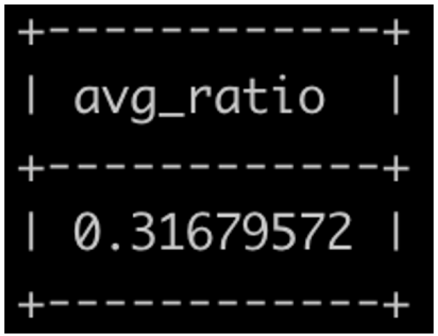
Michael

Query 5

This query finds the percent of vaccines given to each state compared to its population and takes the average. The output represents how much of the U.S. is covered by the number of current vaccines available.

```
select avg(vaccine_to_population_ratio) as avg_ratio from
       (select (total_vaccines / total_population) as vaccine_to_population_ratio from
       total_vaccines_per_state v, total_population_per_state p where v.state = p.state)
as state_vaccinations;
```

Result:



Michael

Query 6

This query finds the population, cases, and vaccines of states that start with ‘M’ or ‘N’ that have a population, case count, and vaccine number over 500,000.

```
select p.state, total_population, total_cases, total_vaccines
from total_population_per_state p, total_cases_per_state c, total_vaccines_per_state v
where p.state = c.state and c.state = v.state and p.state regexp '^M|^N' and
total_population >= 5000000 and total_cases >= 500000 and total_vaccines >= 500000
group by p.state order by p.state;
```

Result:

state	total_population	total_cases	total_vaccines
Massachusetts	6892503	633081	2334840
Michigan	9986857	739244	3317920
Minnesota	5639632	517881	1820165
Missouri	6137428	580980	2017930
New Jersey	8882190	905144	2955860
New York	19453561	1865349	3759985
North Carolina	10488084	912203	3365580

Cam

Query 7

This query finds the total numbers of vaccines distributed. This includes both first and second doses.

```
select (m1.total + m2.total + p1.total + p2.total + j1.total) as total_vaccines from (
        (select sum(m.first_dose) as total from moderna m) as m1,
        (select sum(m.second_dose) as total from moderna m) as m2,
        (select sum(p.first_dose) as total from pfizer p) as p1,
        (select sum(p.second_dose) as total from pfizer p) as p2,
        (select sum(j.only_dose) as total from janssen j) as j1
);
```

Result:

```
+----------------+
| total_vaccines |
+----------------+
|      217975090 |
+----------------+
```

Cam

Query 8

This query uses each vaccine's effectiveness to approximately find the number of people that still contracted COVID-19 after getting the vaccine.

```
select
        ((select sum(m.second_dose) from moderna m) * (1 - 0.941)) as moderna_infected,
        ((select sum(p.second_dose) from pfizer p) * (1 - 0.95)) as pfizer_infected,
        ((select sum(j.only_dose) from janssen j) * (1 - 0.72)) as janssen_infected;
```

Result:

```
+------------------+-----------------+------------------+
| moderna_infected | pfizer_infected | janssen_infected |
+------------------+-----------------+------------------+
|      2932984.400 |      2686651.50 |       2961196.00 |
+------------------+-----------------+------------------+
```

Cam

Query 9

This query finds the total number of vaccines distributed each week across the country.

```
select p_total.week, (ifnull(p_total.weekly_doses,0) + ifnull(m_total.weekly_doses,0)
+ ifnull(j_total.weekly_doses,0)) as weekly_vaccine_total from (
        (select week, sum(first_dose + second_dose) as weekly_doses from pfizer p group by week) as p_total
        left join
        (select week, sum(first_dose + second_dose) as weekly_doses from moderna m group by week) as m_total
        on p_total.week = m_total.week
        left join
        (select week, sum(only_dose) as weekly_doses from janssen j group by week) as j_total
        on m_total.week = j_total.week
) order by weekly_vaccine_total desc;
```

Result:

week	weekly_vaccine_total
4/5/21	21223620
3/29/21	20207120
3/1/21	17673560
3/22/21	16284880
3/15/21	16217860
12/21/20	16082350
3/8/21	15298380
2/22/21	13500040
2/8/21	11159750
2/15/21	10999750
2/1/21	10166530
12/28/20	9381050
1/25/21	8600350
1/18/21	8599750
1/4/21	8392500
1/11/21	8263500
12/14/20	5924100

Cam

Query 10

This query finds the state that obtained the highest percentage of vaccines compared to its population for each of the three vaccines.

```
select * from (
        (select vac1.state as moderna_state, max(vac1.moderna_percentage) as max_moderna_percentage from
                (select m.state, (sum(m.second_dose) / p.total_population) as moderna_percentage from
                moderna m join total_population_per_state p on m.state = p.state group by m.state
                ) as vac1
        group by vac1.moderna_percentage order by max_moderna_percentage desc limit 1) as max1,
        (select vac2.state as pfizer_state, max(vac2.pfizer_percentage) as max_pfizer_percentage from
                (select pf.state, (sum(pf.second_dose) / p.total_population) as pfizer_percentage from
                pfizer pf join total_population_per_state p on pf.state = p.state group by pf.state
                ) as vac2
        group by vac2.pfizer_percentage order by max_pfizer_percentage desc limit 1) as max2,
        (select vac3.state as janssen_state, max(vac3.janssen_percentage) as max_janssen_percentage from
                (select j.state, (sum(j.only_dose) / p.total_population) as janssen_percentage from
                janssen j join total_population_per_state p on j.state = p.state group by j.state
                ) as vac3
        group by vac3.janssen_percentage order by max_janssen_percentage desc limit 1) as max3
);
```

Result:

moderna_state	max_moderna_percentage	pfizer_state	max_pfizer_percentage	janssen_state	max_janssen_percentage
Alaska	0.2549	Alaska	0.3049	Palau	0.2110

Cam

What Did We Learn

- Throughout the development and implementation of this project, our team has learned the benefits of utilizing the database model and the theory of logical design to construct a complex database application.
- Finding data on our subject matter led to some difficulties due to it being so recently available. We find that it would probably be beneficial to work with more complete data sets made after events have taken place. This could allow us to draw better insights from more fleshed-out data.
- We also realize that coming up with good queries takes time and takes a good data set.

Justin