June 15, 2022

Chia yi quan

Student Number: 210220223

CM3010 Databases and Advanced Data Techniques

MID TERM REPORT

Table of Contents

[Stage 1. Find and critique a dataset 2](#_Toc107065118)

[Choosing A Source Of Data 2](#_Toc107065119)

[Assessing The Dataset 2](#_Toc107065120)

[**Reuse our work freely** 4](#_Toc107065121)

[Explanation Of Interest 4](#_Toc107065122)

[Stage 2. Model your data 5](#_Toc107065123)

[Stage 3. Create the database 11](#_Toc107065124)

[Building Of Database Structure In MySQL 11](#_Toc107065125)

[Populating Data From CSV To MySQL 14](#_Toc107065126)

[List SQL commands that answer questions identified in Stage 1/Step 3. 15](#_Toc107065127)

# Stage 1. Find and critique a dataset

## Choosing A Source Of Data

For this coursework, I have chosen an open data from [Our World In Data](https://ourworldindata.org/) and the dataset I used is the [coronavirus](https://ourworldindata.org/coronavirus-source-data) dataset. This is the [link](https://github.com/owid/covid-19-data/tree/master/public/data/) where the csv file is downloaded from. The reason why I picked this dataset is because coronavirus have been with us for the past 2+ years and there have been many variants of the virus and what made this dataset interesting is that we can examine the number of cases, death rate, vaccination and other data by country. Because the data is in csv, we could use the database to help us organize the data better which will allow us to visualize the data easily. The main objective of this database is to allow us to better visualize the coronavirus cases, the relationship of the death rate and vaccination around the world by country, continent, and as a world.

## Assessing The Dataset

**Quality**

Graphical user interface, text, application, email

Description automatically generated

The source can be found [here](https://github.com/owid/covid-19-data/blob/master/public/data/README.md) and the full list [here](https://github.com/owid/covid-19-data/blob/master/public/data/owid-covid-codebook.csv). The quality of the data is quite reliable as the data collected came from different places CCSE from Johns Hopkins University, United Nations, World Bank, Global Burden of Disease, Blavatnik School of Government, etc. The organization([Our World In Data](https://ourworldindata.org/about)) researches poverty, disease, hunger, climate change, war, existential risks, and inequality and data that are released by them have been used by journals as a source like The Washington Post, The New York Times and The Economist(cited from [wiki](https://en.wikipedia.org/wiki/Our_World_in_Data)).

**Detail**

The data source has a lot of details like country population, name, the covid cases and deaths, a country excess mortality, hospital and icu, reproduction rate, policy responses for each country, covid tests, vaccination data and other miscellaneous data of the country. Because there are a lot of data provided, most of them are helpful but some aren’t that helpful. However, because this data consisted of countries around the world, there are some country data that are not up to date, so it is a bit difficult to do 100% analysis of all country.

**Documentation**

The clarity of the data is quite straightforward. The documentation can be found [here](https://github.com/owid/covid-19-data/blob/master/public/data/README.md) and it have stated on every field what it description. The documentation is relatively easy to find because the csv file can only be downloaded from their github repository and on that page, they have displayed the README.md with all fields and it’s description.

**Interrelation**

If we just look at covid cases and deaths, the data will not mean anything but because this dataset provided a lot of other data, linking with other columns will create a more useful analysis. Stringency index can be linked with the covid cases and deaths which will let us analyse whether those policy in place decreases daily new case. Vaccination data and covid data would show us insight whether this vaccine have any effect of slowing/improve the covid situation. Using hospitalize data(covid patient hospitalized and covid icu patient) and covid data will show us is all the covid cases is very severe that people who infected with coronavirus need to be hospitalized or not.

**Use**

This data can be used for analysis in whether certain policy decision or certain factor can slow down the growth rate of the covid cases as new vaccine is being developed to counter this virus. Question like if the country has high chronic diseases(like diabetes) will it contribute to a higher death rate. However, the data does not record all the chronic disease for each days/month instead it records only the latest value. Other than that there are a lot of data that are missing due to some country not reporting or is not up to date.

**Discoverability**

The data of my chosen domain is relatively easy to find. There will other website they provide the csv data like [gov.sg](https://data.gov.sg/dataset/covid-19-case-numbers) provide covid cases in Singapore, [kaggle](https://www.kaggle.com/datasets/sudalairajkumar/novel-corona-virus-2019-dataset) provide the covid cases csv however, the author stopped updating the csv. There are other alternative that I have not explored but the current source I used have all the information and additional dataset which can be made a good use with it.

**Licensing**

So as mentioned in the [first section](#_Choosing_A_Source) that the data was found in [Our World In Data](https://ourworldindata.org/coronavirus-source-data), under the [licensing section](https://ourworldindata.org/coronavirus-source-data#licence), the data is open access license.

### **Reuse our work freely**

All visualizations, data, and code produced by Our World in Data are completely open access under the [Creative Commons BY license](https://creativecommons.org/licenses/by/4.0/). You have the permission to use, distribute, and reproduce these in any medium, provided the source and authors are credited.

The data produced by third parties and made available by Our World in Data is subject to the license terms from the original third-party authors. We will always indicate the original source of the data in our documentation, so you should always check the license of any such third-party data before use and redistribution.

All of [our charts can be embedded](https://ourworldindata.org/how-to-use-our-world-in-data#how-to-embed-interactive-charts-in-your-article) in any site.

Although this dataset is open access license, there are data that are from third-party authors who might have different licensing. However, even though the data is open access, we should always include the citations of the source when we use the data.

## Explanation Of Interest

As mentioned from the [first section](#_Choosing_A_Source), what made this dataset interesting is because we are still experiencing the pandemic when I was doing this coursework. So, the few questions I am interested is

1. Does more people get vaccinated reduces the covid cases or deaths?
2. Which top 10 country have the lowest total covid cases against the total population of the country?
3. As the cases increase, people who are vaccinated increases, do more people get hospitalized?

# Stage 2. Model your data

Diagram

Description automatically generated

**Fig 1.1 ER Model**

A screenshot of a computer

Description automatically generated with medium confidence

**Fig 1.2 Database Table**

Fig 1.1 is the er model, Fig 1.2 is the database table and all the table below are the [description](https://github.com/owid/covid-19-data/tree/master/public/data/) of all the columns that are in the csv. For this coursework I did not use all the columns in the csv provided. Most of the table uses iso\_code as the foreign key which is the primary key in Country table.

**Text

Description automatically generated**

From the above table, total\_cases, new\_cases\_smoothed, total\_cases\_per\_million, new\_cases\_per\_million, new\_cases\_smoothed\_per\_million I did not use is because storing new\_cases in the database would allow us to get the total number of cases so all the total\_cases column can be left out. new\_cases\_smoothed is around the same as new\_cases which makes all the other column irrelevant.

A picture containing table

Description automatically generated

From the above table, total\_deaths, new\_deaths\_smoothed, total\_deaths\_per\_million, new\_deaths\_per\_million, new\_deaths\_smoothed\_per\_million I did not use because storing new\_deaths would allow us to get the total number of deaths so all the total columns can be left out. new\_deaths\_smoothed is around the same as new\_deaths which make other columns irrelevant.

**Graphical user interface, text, application, email

Description automatically generated**

From the above table, all the column I did not use as I didn’t find it relevant with the analysis I trying to do.

**Graphical user interface, text

Description automatically generated**

From the above table, only icu\_patients and hosp\_patients are used because weekly\_icu\_admissions and weekly\_hosp\_admissions can be calculated with icu\_patients and hosp\_patients. For the other columns that are irrelevant as they mean the same thing as the used columns.

**A picture containing text

Description automatically generated**

From the above table, stringency\_index data is used.

**Graphical user interface, text, application

Description automatically generated**

For the above table, I did not use the reproduction\_rate because it is irrelevant to my analysis which I was doing.

**Graphical user interface, text, application, email

Description automatically generated**

From the above table, total\_tests data are not used as new\_tests are able to calculate it. total\_tests\_per\_thousand, new\_tests\_per\_thousand, new\_tests\_smoothed, new\_tests\_smoothed\_per thousand all became relevant when new\_tests is used. tests\_unit are irrelevant for this case and since positive\_rate are used tests\_per\_case has become irrelevant.

**Graphical user interface, text, application, email

Description automatically generated**

From the above table, total\_vaccinations, total\_vaccinations\_per\_hundred is irrelevant as new\_vaccinations column is used which can calculate the total. new\_vaccinations\_smoothed and new\_vaccinations\_smoothed\_per\_million become irrelevant as new\_vaccinations column are used. people\_fully\_vaccinated and people\_fully\_vaccinated\_per\_hundred become irrelevant as it stored the total number of people only. new\_people\_vaccinated\_smoothed\_per\_hundred is irrelevant as new\_people\_vaccinated\_smoothed is used instead. total\_boosters and total\_boosters\_per\_hundred are not used in this case as only the total number are stored.

**Graphical user interface, text

Description automatically generated with medium confidence**

From the table above, median\_age, aged\_70\_older, gdp\_per\_capita, extreme\_poverty, female\_smokers, male\_smokers, human\_development\_index, handwashing\_facilities, hospital\_beds\_per\_thousand are not used as it is not relevant to the analysis that I am doing. cardiovasc\_death\_rate and diabetes\_prevalence are not used as the data is from 2017 which are irrelevant to the analysis that I am doing.

# Stage 3. Create the database

## 3.1 Building Of Database Structure In MySQL

**Creating Database Command**

CREATE DATABASE Covid;

**Creating Of User And Granting Of The Appropriate Permission**

CREATE USER 'covidUser'@'127.0.0.1' IDENTIFIED WITH mysql\_native\_password BY 'password';

GRANT ALTER, SELECT, INSERT, CREATE, DROP, REFERENCES on Covid.\* TO 'covidUser'@'127.0.0.1' WITH GRANT OPTION;

FLUSH PRIVILEGES;

\*\* Reasoning for granting ALTER, SELECT, INSERT, CREATE, DROP, REFERENCES to covidUser and not just the SELECT option is because I wrote script to migrate the data into the database therefore this option are needed.\*\*

**Creating Of Tables In Database**

|  |  |
| --- | --- |
| **Tables** | **Query** |
| Continent | CREATE TABLE `Continent` (  `id` int AUTO\_INCREMENT PRIMARY KEY,  `continent` varchar(255)  ); |
| Country | CREATE TABLE `Country` (  `iso\_code` varchar(255) PRIMARY KEY,  `location` varchar(255),  `continent\_id` int  ); |
| NewCase | CREATE TABLE `NewCase` (  `id` int AUTO\_INCREMENT PRIMARY KEY,  `date` datetime,  `new\_case` int,  `new\_death` int,  `iso\_code` varchar(255)  ); |
| Hospitalize | CREATE TABLE `Hospitalize` (  `id` int AUTO\_INCREMENT PRIMARY KEY,  `date` datetime,  `icu\_patient` int,  `hosp\_patients` int,  `iso\_code` varchar(255)  ); |
| Test | CREATE TABLE `Test` (  `id` int AUTO\_INCREMENT PRIMARY KEY,  `date` datetime,  `new\_tests` int,  `positive\_rate` double,  `iso\_code` varchar(255)  ); |
| Vaccination | CREATE TABLE `Vaccination` (  `id` int AUTO\_INCREMENT PRIMARY KEY,  `date` datetime,  `new\_vaccinations\_smoothed` int,  `new\_people\_vaccinated\_smoothed` int,  `iso\_code` varchar(255)  ); |
| Population | CREATE TABLE `Population` (  `id` int AUTO\_INCREMENT PRIMARY KEY,  `population` int,  `population\_density` int,  `aged\_65\_older` double,  `life\_expectancy` double,  `iso\_code` varchar(255)  ); |
| Stringency | CREATE TABLE `Stringency` (  `id` int AUTO\_INCREMENT PRIMARY KEY,  `date` datetime,  `stringency\_index` double,  `iso\_code` varchar(255)  ); |

**Adding Foreign Key Constraint To Tables**

ALTER TABLE `Country` ADD FOREIGN KEY (`continent\_id`) REFERENCES `Continent` (`id`);

ALTER TABLE `NewCase` ADD FOREIGN KEY (`iso\_code`) REFERENCES `Country` (`iso\_code`);

ALTER TABLE `Hospitalize` ADD FOREIGN KEY (`iso\_code`) REFERENCES `Country` (`iso\_code`);

ALTER TABLE `Test` ADD FOREIGN KEY (`iso\_code`) REFERENCES `Country` (`iso\_code`);

ALTER TABLE `Vaccination` ADD FOREIGN KEY (`iso\_code`) REFERENCES `Country` (`iso\_code`);

ALTER TABLE `Population` ADD FOREIGN KEY (`iso\_code`) REFERENCES `Country` (`iso\_code`);

ALTER TABLE `Stringency` ADD FOREIGN KEY (`iso\_code`) REFERENCES `Country` (`iso\_code`);

## 3.2 Populating Data From CSV To MySQL

To populate data from csv to MySQL I chose to write a nodejs script. The script can be found mid-term/scripts/migration.js. I will go through briefly on the code that is in the migration.js. The code that are before the createTable function are setup code to use mysql library, reading the csv file data, and setting up the connection to connect to the database.

createTable function is to create all the tables and it’s references, the only different is that I added **IF NOT EXISTS** keyword so that if the table did not delete successfully it won’t crash the script when it tries to create the table again.

deleteTable function is to delete all the tables that are in Covid database. It will select all the tables from information\_schema.tables first then loop through the list of tables to delete except the Country and Continent table. Reasoning is that the primary key are used as foreign key in other table so deleting other tables first then Country table then Continent table would not throw error because of foreign key.

The run function is the main function of this script so when the script run, it will call deleteTable function and createTable function.

Text

Description automatically generated

The above screenshot is to create all the variable named according to their table name. The code after declaration of the variables is a for loop to access each row of data in the csv file. The first few if is to check for if the current row is the header in the csv which is the first row, if the first column is empty and if the iso\_code begin with OWID\_ (those are continent data which we don’t want to include). If those condition are met it will skip those rows. The continents variable type is a set because we only 1 unique record of the continent. For countryData and populationData it is type object, the reason for this is that we only want 1 record for each country. For countryData, every country will only have 1 entry for iso\_code, location and continent\_id. For populationData, all the data that are provided in the csv are duplicated data(latest number only) and they are all the same value throughout the csv for each country so there isn’t a point to store multiple entry of it and just store 1 record for each country. For the remaining table data, it will store all the entry as most of the data is updated daily. One thing to note is that because the csv have a lot of missing data which are blank for those number field so 0 will be inserted in.

After processing all the data, continents will be converted from set to array and is being inserted into the Continent table first. This is because country need to use the Continent table primary key as foreign key reference. After continents are being inserted, it will be retrieved out and transformed to an object with the continent name as the key and the id as value. countryData will then be iterated and changing the continent name in countryData to the id of the continent that is in the database. After that the countryData will be inserted into the database first because other tables uses iso\_code as the foreign key so it have to exist in the database first. populationData will have it’s key removed because we stored in object but because mysql library only accept array as the value so the key have to be removed. After removing the key from populationData, we will then insert all the data for the remaining tables.

To run the script, just open the terminal and **cd mid-term** directory and run the following command in the terminal **npm run migrate** and the script will run and do its job. Benefit of using a script is that if there is a need to migrate the newer csv data simply just put in the latest csv into the scripts folder and run the script which will save a lot of time. Another benefit of using a script is that it is more maintainable when there is a need to create a new table or update the newer data as there is no need to create a table to store all the data in the csv before storing it to different tables.

## 3.3 List SQL commands that answer questions identified in Stage 1/Step 3.

1. Does more people get vaccinated reduces the covid cases or deaths?

**Query**:

WITH newcase\_query AS (SELECT DATE\_FORMAT(`date`,'%Y-%m') AS date, FORMAT(SUM(new\_case),0) AS 'Monthly Total Case', FORMAT(SUM(new\_death),0) AS 'Monthly Total Death' FROM NewCase GROUP BY DATE\_FORMAT(`date`,'%Y-%m')), vaccination\_query AS (SELECT DATE\_FORMAT(`date`,'%Y-%m') AS date, FORMAT(SUM(new\_vaccinations\_smoothed),0) AS 'Monthly Total Vaccination', FORMAT(SUM(new\_people\_vaccinated\_smoothed),0) AS 'Monthly Total Unique Vaccinated People' FROM Vaccination GROUP BY DATE\_FORMAT(`date`,'%Y-%m')) SELECT \* FROM newcase\_query LEFT JOIN vaccination\_query USING (date) ORDER BY date;

**Result:**

Graphical user interface, text

Description automatically generated

This is the result for monthly world total case, total death, total vaccination and total unique vaccinated people. From the result, we can see that before the vaccination is release, the covid case and death has been increasing monthly. When there are vaccines, there was a shortage in the beginning and not a lot of people are able to get vaccinated. In Singapore, the government prioritized elderly who are above 65-year-old to get vaccinated first as they are in the high-risk category. From the data we can see that starting from June 2020 there is more vaccine created and more people are vaccinated. The unique vaccinated people means that new people getting their first vaccine dose. We can see that there are a lot of people who are vaccinated but the monthly total case has been increasing and still a lot of cases but then the death rate is not that high. In conclusion, more people get vaccinated does not mean that cases will decrease but death rate will decrease for sure.

1. Which top 10 country have the lowest total covid cases against the total population of the country?

**Query:**

WITH newCase\_query AS (SELECT SUM(new\_case) AS total\_case, iso\_code FROM NewCase GROUP BY iso\_code), country\_query AS (SELECT iso\_code, location FROM Country), population\_query AS (SELECT iso\_code, population, population\_density FROM Population) SELECT country\_query.location AS Country, FORMAT((newCase\_query.total\_case/population\_query.population),10) AS 'Total Covid Case Against Population', FORMAT(population,0) AS Population, population\_query.population\_density AS 'Population Density' FROM newCase\_query LEFT JOIN country\_query USING (iso\_code) LEFT JOIN population\_query USING (iso\_code) WHERE (newCase\_query.total\_case/population\_query.population) > 0 ORDER BY (newCase\_query.total\_case/population\_query.population) LIMIT 10;

**Result:**

Graphical user interface, text

Description automatically generated

From the result, this are the top 10 country that have and is sorted according to the total covid cases the country has divided by its population. The smaller the ratio would mean the country are better at controlling this pandemic. However, this can be a flaw in the dataset too because there are country that did not report their covid cases and death which make it inaccurate to say that this are the top 10 country that have the lowest total covid cases against the population.

1. As the cases increase, people who are vaccinated increases, do more people get hospitalized?

**Query:**

WITH newcase\_query AS (SELECT DATE\_FORMAT(`date`,'%Y-%m') AS date, FORMAT(SUM(new\_case),0) AS 'Monthly Total Case' FROM NewCase GROUP BY DATE\_FORMAT(`date`,'%Y-%m')), hospitalize\_query AS (SELECT DATE\_FORMAT(`date`,'%Y-%m') AS date, FORMAT(SUM(icu\_patient),0) AS 'Monthly Total ICU Patient', FORMAT(SUM(hosp\_patients),0) AS 'Monthly Total Hospital Patients' FROM Hospitalize GROUP BY DATE\_FORMAT(`date`,'%Y-%m')), vaccination\_query AS (SELECT DATE\_FORMAT(`date`,'%Y-%m') AS date, FORMAT(SUM(new\_vaccinations\_smoothed),0) AS 'Monthly Total Vaccination', FORMAT(SUM(new\_people\_vaccinated\_smoothed),0) AS 'Monthly Total Unique Vaccinated People' FROM Vaccination GROUP BY DATE\_FORMAT(`date`,'%Y-%m')) SELECT \* FROM newcase\_query LEFT JOIN hospitalize\_query USING (date) LEFT JOIN vaccination\_query USING (date) ORDER BY date;

**Result:**

Graphical user interface

Description automatically generated with medium confidence

Based on the result, we can see that the monthly total admitted icu patient because of covid for the whole world increases when the vaccine was not ready, and the monthly total hospital patient admitted because of covid increases. As more vaccine is available to the world and more people vaccinated, we can see that the monthly total icu patient who admitted because of covid, and monthly total hospital patient admitted because of covid have decreased significantly as the time goes.

So, the key takeaway from the 3 question I have gone through is that when there is vaccine, we should take it so that we will have lesser risk to death, personal hygiene and following country protocol is a must because if there is still a covid case in the world, it will still spread so taking care of ourselves also play a part of taking care of others.

# Stage 4. Create a simple web application

**Instruction To Run The Web Application**

My Cousera Lab URL is: <https://hub.labs.coursera.org:443/connect/sharedsgrfbvwc?forceRefresh=false&path=%2F%3Ffolder%3D%2Fhome%2Fcoder%2Fproject>

* **cd mid-term**
* Open the terminal and run **npm start**
* Visit <localhost:8088/>

**\*\*Note: Coursera Browser Preview doesn’t allow select option to be displayed, use the down and up arrow key to navigate to different option and use the enter button to confirm the select option\*\***

**Covid Cases And Vaccination Table**

Table

Description automatically generated

**World Page**

Table

Description automatically generated

**Continent Page**

**Table

Description automatically generated**

**Country Page**

The above table is about covid cases and vaccinations which address the question of Does more people get vaccinated reduces the covid cases or deaths?

For the table in world page, the table can filter by month which will show the daily data. This was achieved by changing DATE\_FORMAT(`date`,'%Y-%m') AS date to DATE\_FORMAT(`date`,'%Y-%m-%d') AS date and adding a WHERE DATE\_FORMAT(`date`,'%Y-%m')=? and changed GROUP BY DATE\_FORMAT(`date`,'%Y-%m') to GROUP BY DATE\_FORMAT(`date`,'%Y-%m-%d') for the query in [3.3](#_3.3_List_SQL) first question query.

For continent page, the table can be filter by month to show the daily data and by continent. To achieve that, I have added the WHERE iso\_code IN (SELECT iso\_code FROM Country WHERE continent\_id=? To the query in [3.3](#_3.3_List_SQL) first question query. And to display daily data I changed DATE\_FORMAT(`date`,'%Y-%m') AS date to DATE\_FORMAT(`date`,'%Y-%m-%d') AS date and added WHERE DATE\_FORMAT(`date`,'%Y-%m')=? AND iso\_code IN (SELECT iso\_code FROM Country WHERE continent\_id=?) GROUP BY DATE\_FORMAT(`date`,'%Y-%m-%d') for the query in [3.3](#_3.3_List_SQL) first question query.

For country page, the table can be filter by month which will show the daily data and by country. To achieve that, I have added the WHERE iso\_code=? To the query in [3.3](#_3.3_List_SQL) first question query. And to display daily data I changed DATE\_FORMAT(`date`,'%Y-%m') AS date to DATE\_FORMAT(`date`,'%Y-%m-%d') AS date and added WHERE DATE\_FORMAT(`date`,'%Y-%m')=? AND iso\_code=? GROUP BY DATE\_FORMAT(`date`,'%Y-%m-%d') for the query in [3.3](#_3.3_List_SQL) first question query.

**Covid Cases And Hospitalization**

Table

Description automatically generated

**World Page**

Table

Description automatically generated

**Continent Page**

**Table

Description automatically generated**

**Country Page**

The above table is the Covid Cases And Hospitalization table which address the question of As the cases increase, people who are vaccinated increases, do more people get hospitalized?

For the table in world page, the table can filter by month which will show the daily data. This was achieved by changing DATE\_FORMAT(`date`,'%Y-%m') AS date to DATE\_FORMAT(`date`,'%Y-%m-%d') AS date and adding WHERE DATE\_FORMAT(`date`,'%Y-%m')=? and changed GROUP BY DATE\_FORMAT(`date`,'%Y-%m') to GROUP BY DATE\_FORMAT(`date`,'%Y-%m-%d') for the query in [3.3](#_3.3_List_SQL) third question query.

For the table in continent page, the table can be filter by month which will show the daily data and by continent. To achieve that, I have added the WHERE iso\_code IN (SELECT iso\_code FROM Country WHERE continent\_id=? To the query in [3.3](#_3.3_List_SQL) third question query. And to display daily data I changed DATE\_FORMAT(`date`,'%Y-%m') AS date to DATE\_FORMAT(`date`,'%Y-%m-%d') AS date and added WHERE DATE\_FORMAT(`date`,'%Y-%m')=? AND iso\_code IN (SELECT iso\_code FROM Country WHERE continent\_id=?) GROUP BY DATE\_FORMAT(`date`,'%Y-%m-%d') for the query in [3.3](#_3.3_List_SQL) third question query.

For the table in country page, the table can be filter by month which will show the daily data and by continent. To achieve that, I have added a WHERE iso\_code=? To the query in [3.3](#_3.3_List_SQL) third question query. And to display daily data I changed DATE\_FORMAT(`date`,'%Y-%m') AS date to DATE\_FORMAT(`date`,'%Y-%m-%d') AS date and added WHERE DATE\_FORMAT(`date`,'%Y-%m')=? AND iso\_code=? GROUP BY DATE\_FORMAT(`date`,'%Y-%m-%d') for the query in [3.3](#_3.3_List_SQL) third question query.

Table

Description automatically generated

This is the Covid Cases Against Population which is addressing the question of Which top 10 country have the lowest total covid cases against the total population of the country? However, for this query I removed the LIMIT 10 from [3.3](#_3.3_List_SQL) second question query which will display for all the country.

Table

Description automatically generated

The last table which is Test table that can be found in Country page. This table shows the daily covid cases and death along side with the daily test and daily positive rate from the test conducted. This table will provide us how many were tested daily and its positive rate.