

Corona Virus Analysis With SQL

MENTORNESS INTERNSHIP PRO
SUBMITTED BY:- ANKIT KUMAR
BATCH: MIP-DA-09

CONTENT

PROJECT OVERVIEW

DATASET DESCRIPTION

DATA EXPLORATION AND ANALYSIS

(*QUERIES*)

PROJECT OVERVIEW

The corona virus pandemic has had a major effect on public health, and there is a pressing need for data-driven insights to comprehend how the virus is spreading. We will be examining a corona virus dataset as data analysts in order to provide useful discoveries that we will share.

ATASET

Description of each column in dataset:

Province: Geographic subdivision within a country/region.

Country/Region: Geographic entity where data is recorded.

Latitude: North-south position on Earth's surface.

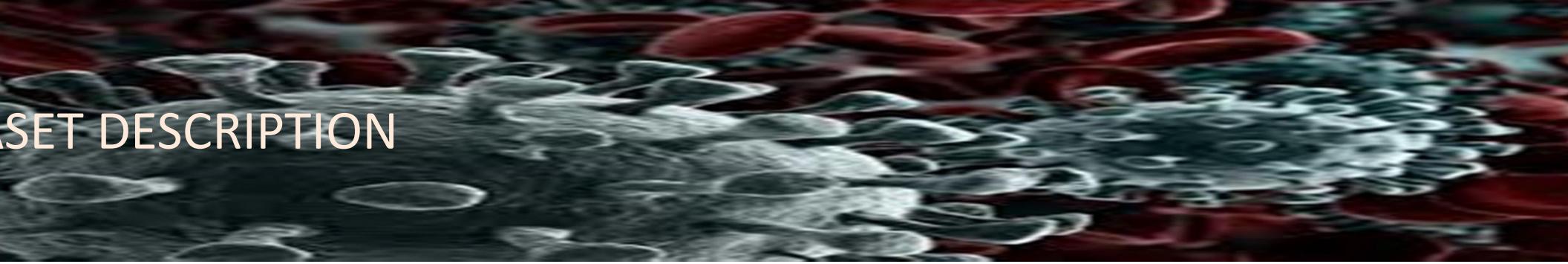
Longitude: East-west position on Earth's surface.

Date: Recorded date of CORONA VIRUS data.

Confirmed: Number of diagnosed CORONA VIRUS cases.

Deaths: Number of CORONA VIRUS related deaths.

Recovered: Number of recovered CORONA VIRUS cases.



Dataset Description

Query History

```
select* from corona_virus|
```

Output Messages Notifications

province character varying	country character varying	latitude numeric	longitude numeric	date timestamp without time zone	confirmed integer	deaths integer	recovered integer
Afghanistan	Afghanistan	33.93911	67.709953	2020-01-22 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-01-23 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-01-24 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-01-25 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-01-26 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-01-27 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-01-28 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-01-29 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-01-30 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-01-31 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-02-01 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-02-02 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-02-03 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-02-04 00:00:00	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	2020-02-05 00:00:00	0	0	0

DATA EXPLORATION AND ANALYSIS (QUERIES)

--Q1: Write a code to check the missing values

```
select * from corona_virus  
where province is null or country is null  
or latitude is null or longitude is null  
or date is null or confirmed is null  
or deaths is null or recovered is null;
```

Data Output Messages Notifications

province	country	latitude	longitude	date	confirmed	deaths	recovered
character varying	character varying	numeric	numeric	timestamp without time zone	integer	integer	integer

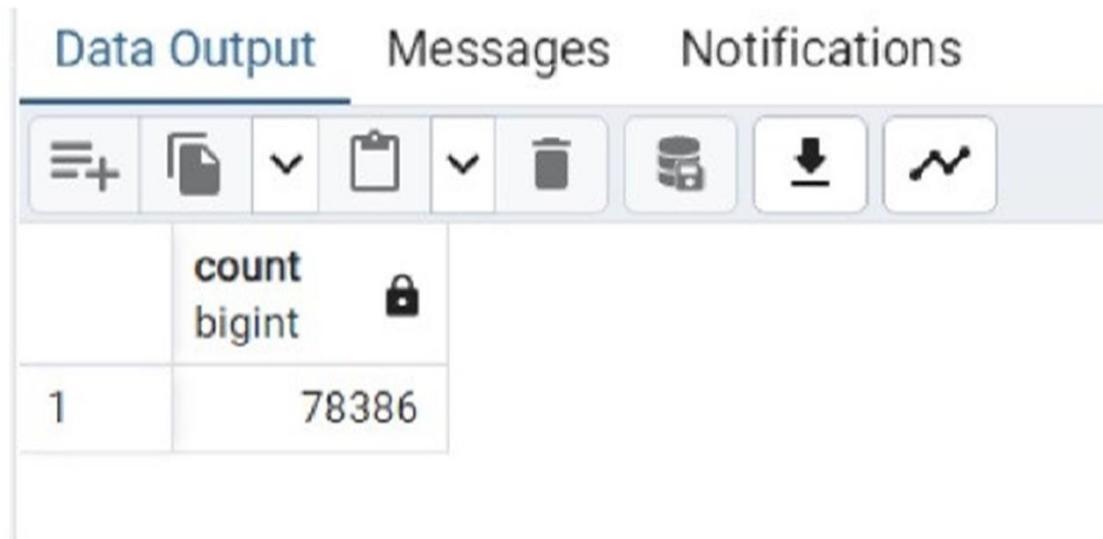
--Q2: If null values are present, update them with zeros for all columns

```
update corona_virus set province= coalesce(province,''),  
country= coalesce(country,''),  
latitude = coalesce(latitude,0),  
longitude= coalesce(longitude,0),  
confirmed= coalesce(confirmed,0),  
deaths= coalesce(deaths,0),  
recovered= coalesce(recovered,0);
```

--Q3: Check total number of rows

```
select count(*) from corona_virus;
```

Data Output Messages Notifications



	count	bigint	lock
1	78386		

--Q4: Check what is start date

```
select date from corona_virus  
order by date  
limit 1;
```

Data Output		Messages	Notifications
	date		🔒
1	2020-01-22 00:00:00		

--Check what is end date

```
Select date from corona_virus  
order by date desc  
limit 1;
```

Data Output		Messages	Notifications
	date		🔒
1	2021-06-13 00:00:00		

-Q5: Number of months present in Dataset

```
select Extract(month from date) as Number_of_month  
from corona_virus  
group by number_of_month  
order by number_of_month;
```

Data Output Messages Notifications

The screenshot shows a data output interface with a toolbar at the top and a table below. The toolbar includes icons for file operations like new, open, save, and delete, as well as other functions. The table has a single column labeled 'number_of_month' with a type of 'numeric'. The data consists of 12 rows, each containing a number from 1 to 12.

	number_of_month	numeric
1		1
2		2
3		3
4		4
5		5
6		6
7		7
8		8
9		9
10		10
11		11
12		12

nd monthly average for confirmed,deaths,recovered

extract(month from date) as month,

irmed) as avg_confirmed,

hs) as avg_deaths,

vered) as avg_recovered

cna_virus

month

month

Data Output Messages Notifications

The screenshot shows a software interface for managing data outputs. At the top, there are tabs for "Data Output", "Messages", and "Notifications". Below the tabs is a toolbar with several icons: a plus sign, a file icon, a dropdown arrow, a clipboard icon, another dropdown arrow, a trash can icon, a database icon, a download arrow, and a refresh/clock icon. The main area is a table with four columns: "month" (numeric), "avg_confirmed" (numeric), "avg_deaths" (numeric), and "avg_recovered" (numeric). The table has 12 rows, one for each month from 1 to 12, displaying the calculated averages for each category.

	month numeric	avg_confirmed numeric	avg_deaths numeric	avg_recovered numeric
1	1	2958.2814380741210010	63.6811846689895470	1451.45549
2	2	1203.1187058555479608	34.2777398040555935	769.10344
3	3	1538.9637620444072057	33.9302471721826561	840.07991
4	4	2602.5778138528138528	59.9805194805194805	1623.21363
5	5	2290.0519480519480519	53.5305823209049016	2162.90207
6	6	1357.8852310480217457	40.8356991845363938	1220.15327
7	7	1432.3611227482195224	35.1095517385839966	983.05823
8	8	1611.8428990364474235	37.5366568914956012	1299.29472
9	9	1784.5874458874458874	34.7772727272727273	1438.90670
10	10	2412.1996229576874738	36.7582739840804357	1420.64306
11	11	3592.1943722943722944	56.7634199134199134	1985.34458
12	12	4050.4396732299958106	71.2182656053623796	2497.88500

most frequent value for confirmed, deaths, recovered each month

act(month from date) as month,

ed) as most_frequent_confirmed,

as most_frequent_deaths,

ed) as most_frequent_recovered

virus

nth

nth;

Data Output Messages Notifications

≡+ ↻ ⌂ ⌂ ↴ ↵

	month numeric	most_frequent_confirmed integer	most_frequent_deaths integer	most_frequent integer
1	1	300462	4475	
2	2	134975	3907	
3	3	100158	3869	
4	4	401993	4249	
5	5	414188	4529	
6	6	134154	7374	
7	7	75866	1595	
8	8	85687	1505	
9	9	97894	1703	
10	10	99264	3351	
11	11	207933	2259	
12	12	823225	3752	

--Q8: Find minimum values for confirmed,deaths,recovered per year

```
select extract(year from date) as year,  
min(confirmed) as min_confirmed,  
min(deaths) as min_deaths,  
min(recovered) as min_recovered  
from corona_virus  
group by year  
order by year;
```

Data Output Messages Notifications

The screenshot shows a database interface with a toolbar at the top containing various icons for file operations like new, save, and delete. Below the toolbar is a table with four columns: 'year' (numeric), 'min_confirmed' (integer), 'min_deaths' (integer), and 'min_recovered' (integer). There are two rows of data: one for the year 2020 with all minimum values set to 0, and another for the year 2021 with all minimum values set to 0.

	year numeric	min_confirmed integer	min_deaths integer	min_recovered integer
1	2020	0	0	0
2	2021	0	0	0

--Q9: Find maximum values for confirmed,deaths,recovered per year

```
select extract(year from date) as year,  
max(confirmed) as max_confirmed,  
max(deaths) as max_deaths,  
max(recovered) as max_recovered  
from corona_virus  
group by year  
order by year;
```

Data Output Messages Notifications

The screenshot shows a database query results interface. At the top, there are tabs for 'Data Output', 'Messages', and 'Notifications'. Below the tabs is a toolbar with various icons for file operations like new, open, save, and download. The main area displays a table with four columns: 'year', 'max_confirmed', 'max_deaths', and 'max_recovered'. The first row corresponds to the year 2020 with values 823225, 3752, and 1123456 respectively. The second row corresponds to the year 2021 with values 414188, 7374, and 422436.

	year numeric	max_confirmed integer	max_deaths integer	max_recovered integer
1	2020	823225	3752	1123456
2	2021	414188	7374	422436

the total number of case of confirmed,deaths,recovered each month

```
extract(month from date) as month,
```

```
confirmed) as total_confirmed,
```

```
aths) as total_deaths,
```

```
covered) as total_recovered
```

```
cna_virus
```

```
month
```

```
month;
```

Data Output Messages Notifications



	month numeric	totalConfirmed bigint	totalDeaths bigint	totalRecovered bigint
1	1	6314	6314	
2	2	8778	8778	
3	3	9548	9548	
4	4	9240	9240	
5	5	9548	9548	
6	6	6622	6622	
7	7	4774	4774	
8	8	4774	4774	
9	9	4620	4620	
10	10	4774	4774	
11	11	4620	4620	
12	12	4774	4774	

--Q11: Check how the corona virus spread out with respect to confirmed case

```
select count(confirmed) as total_confirmed_cases,  
avg(confirmed) as avg_confirmed,  
variance(confirmed) as variance_confirmed,  
STDDEV(confirmed) as std_confirmed  
from corona_virus;
```

Data Output Messages Notifications



	total_confirmed_cases bigint	avg_confirmed numeric	variance_confirmed numeric	std_confirmed numeric
1	78386	2156.8283111780164825	157290931.69817455	12541.56815148

how corona virus spread out with respect to death case per month

```
ct(month from date) as month,  
 ) as total_deaths_per_month,  
 as avg_deaths_per_months,  
ths) as var_deaths_per_months,  
s) as std_deaths_per_months  
virus  
th  
th;
```

Data Output Messages Notifications

≡+ ↻ ↽ ↾ ↿ ↽ ↴ ↵

month	total_deaths_per_month	avg_deaths_per_months	var_deaths_per_months	std_d
numeric	bigint	numeric	numeric	numeric
1	6314	63.6811846689895470	79012.044546925182	
2	8778	34.2777398040555935	34852.618305840004	
3	9548	33.9302471721826561	29785.052429518872	
4	9240	59.9805194805194805	67905.924720587346	
5	9548	53.5305823209049016	76775.779414471802	
6	6622	40.8356991845363938	46250.187470278323	
7	4774	35.1095517385839966	21144.584057079556	
8	4774	37.5366568914956012	23277.872425108734	
9	4620	34.7772727272727273	20107.121414513177	
10	4774	36.7582739840804357	17583.754252708491	
11	4620	56.7634199134199134	27779.806542101184	
12	4774	71.2182656053623796	65359.059829716994	

--Q13: Check how corona virus spread out with respect to recovered case

```
Select count(recovered) as total_recovered,  
avg(recovered) as avg_recovered,  
variance(recovered) as var_recovered,  
stddev(recovered) as std_recovered  
from corona_virus;
```

Data Output Messages Notifications



	total_recovered bigint	avg_recovered numeric	var_recovered numeric	std_recovered numeric
1	78386	1442.7263541959023295	107030888.69602982	10345.57338653

--Q14: Find country having highest number of the confirmed case

```
Select country,max(confirmed) as highest_confirmed_case from corona_virus  
group by country  
order by highest_confirmed_case desc  
limit 1;
```

Data Output Messages Notifications

	country	highest_confirmed_case
	character varying	integer
1	Turkey	823225

6: Find Country having lowest number of the death case

select country, min(deaths) as lowest_deaths_case from corona_virus

group by country

order by lowest_deaths_case

	country character varying	lowest_deaths_case integer
104	Panama	0
105	Yemen	0
106	South Sudan	0
107	Lithuania	0
108	Bulgaria	0
109	Croatia	0
110	Tunisia	0
111	North Macedonia	0
112	Morocco	0
113	Mexico	0
114	Nepal	0
115	Tanzania	0
116	Poland	0
117	Lebanon	0
118	Costa Rica	0
119	Haiti	0
120	Samoa	0
121	Somalia	0

--Q16: Find top 5 countries having highest recovered case

```
select country,max(recovered) as highest_recovered_case from corona_virus  
group by country  
order by highest_recovered_case desc  
limit 5;
```

Data Output Messages Notifications

The screenshot shows a database interface with a toolbar at the top and a table below it. The table has two columns: 'country' and 'highest_recovered_case'. The data is as follows:

	country	highest_recovered_case
1	Turkey	1123456
2	India	422436
3	Brazil	388340
4	US	150267
5	Colombia	89557

**THANK
YOU!**