

SARS-CoV-2, analysis using sqlite3 Saravana.M





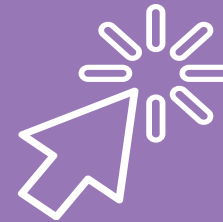
1

Brief overview of the presentation agenda.



2

Brief description of the COVID-19 dataset used.



3

Source of the dataset



4

Explanation of the variables included in the dataset (e.g., date, cases, deaths, location)



5

Clarifying with Question & Answer

PURPOSE OF THE ANALYSIS: UNDERSTANDING COVID-19 TRENDS USING A DATASET.

INTENTION

Analyzing Trends and Understanding Implications

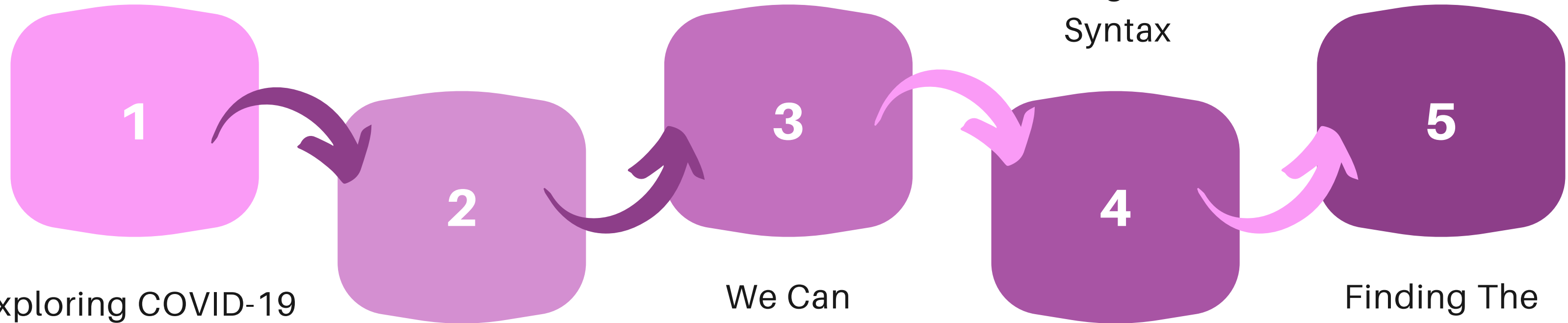
Steps to import the
COVID-19 dataset
into SQLite.

Using SQL
Syntax

Exploring COVID-19
Data Insights with
.CSV

We Can
Slove 16
Query

Finding The
Solution



File

Edit

View

Tools

Help

New Database

Open Database

Write Changes

Revert Changes

Open Project

Save Project

Attach Database

Database Structure

Browse Data

Edit Pragmas

Execute SQL

Table: Corona Virus Dataset

Filter in any column

| | Province | Country/Region | Latitude | Longitude | Date | Confirmed | Deaths | Recovered |
|----|-------------|----------------|----------|-----------|------------|-----------|--------|-----------|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 22-01-2020 | 0 | 0 | 0 |
| 2 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 23-01-2020 | 0 | 0 | 0 |
| 3 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 24-01-2020 | 0 | 0 | 0 |
| 4 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 25-01-2020 | 0 | 0 | 0 |
| 5 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 26-01-2020 | 0 | 0 | 0 |
| 6 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 27-01-2020 | 0 | 0 | 0 |
| 7 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 28-01-2020 | 0 | 0 | 0 |
| 8 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 29-01-2020 | 0 | 0 | 0 |
| 9 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 30-01-2020 | 0 | 0 | 0 |
| 10 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 31-01-2020 | 0 | 0 | 0 |
| 11 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 01-02-2020 | 0 | 0 | 0 |
| 12 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 02-02-2020 | 0 | 0 | 0 |
| 13 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 03-02-2020 | 0 | 0 | 0 |
| 14 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 04-02-2020 | 0 | 0 | 0 |
| 15 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 05-02-2020 | 0 | 0 | 0 |

1 - 15 of 78386

SQL Log

Show SQL submitted by Application

```

1 PRAGMA database_list;
2 SELECT type,name,sql,tbl_name FROM "main".sqlite_master;
3 SELECT COUNT(*) FROM "main"."Corona Virus Dataset"
4 SELECT "_rowid_",* FROM "main"."Corona Virus Dataset" LIMIT 0, 49999;
5

```

Opened 'C:/Users/DELL/Downloads/Corona 19 Virus .csv.sqbpro' from recent file list

1

2

Command-Line Interface (CLI): Open your command-line interface or terminal.

3

Use the **cd** command to navigate to the directory where your SQLite database is located.

4

Use the SQLite **.import** command to import the CSV data into a new or existing table.

5

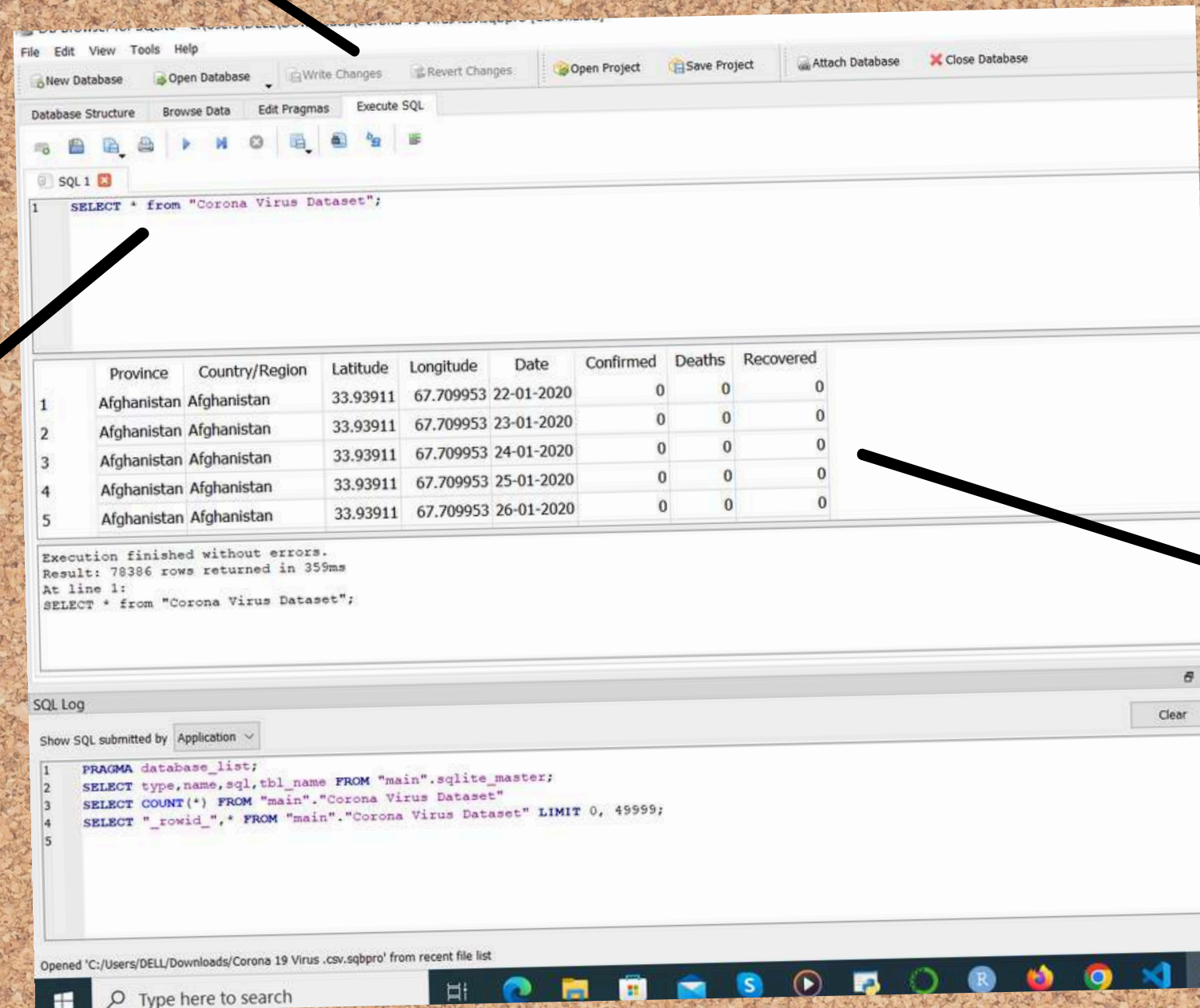
Check if the data was imported successfully by running a simple **SELECT** query.

Data Exploration

Use SQL queries to explore the imported data. For example:

Display import csv file

```
SELECT * FROM Corona  
Virus Dataset;
```



The screenshot shows a SQL browser window with a menu bar (File, Edit, View, Tools, Help) and a toolbar. The main area displays a query result for 'Corona Virus Dataset'. The query is 'SELECT * from "Corona Virus Dataset";'. The result is a table with 9 columns: Province, Country/Region, Latitude, Longitude, Date, Confirmed, Deaths, and Recovered. The first 5 rows are shown, all with values 0 for Confirmed, Deaths, and Recovered. Below the table, a message states 'Execution finished without errors. Result: 78386 rows returned in 359ms'. At the bottom, an 'SQL Log' panel shows the executed queries.

| | Province | Country/Region | Latitude | Longitude | Date | Confirmed | Deaths | Recovered |
|---|-------------|----------------|----------|-----------|------------|-----------|--------|-----------|
| 1 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 22-01-2020 | 0 | 0 | 0 |
| 2 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 23-01-2020 | 0 | 0 | 0 |
| 3 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 24-01-2020 | 0 | 0 | 0 |
| 4 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 25-01-2020 | 0 | 0 | 0 |
| 5 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 26-01-2020 | 0 | 0 | 0 |

Execution finished without errors.
Result: 78386 rows returned in 359ms
At line 1:
SELECT * from "Corona Virus Dataset";

SQL Log

```
1 PRAGMA database_list;  
2 SELECT type,name,sql,tbl_name FROM "main".sqlite_master;  
3 SELECT COUNT(*) FROM "main"."Corona Virus Dataset"  
4 SELECT "_rowid_",* FROM "main"."Corona Virus Dataset" LIMIT 0, 49999;  
5
```

Opened 'C:/Users/DELL/Downloads/Corona 19 Virus .csv.sqbp' from recent file list

Display 5 row in Dataset using

```
SELECT *FROM `Corona Virus  
Dataset` LIMIT 5;
```


Checking Null Value

```
SELECT *FROM `Corona Virus  
Dataset`WHERE Confirmed IS  
NULL;
```

Executing Successful without
Errors

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbpro [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project

Database Structure Browse Data Edit Pragmas Execute SQL


SQL 1

```
1 SELECT *FROM `Corona Virus Dataset`WHERE Confirmed IS NULL;
```

Execution finished without errors.
Result: 0 rows returned in 108ms
At line 1:
SELECT *FROM `Corona Virus Dataset`WHERE Confirmed IS NULL;

If NULL values are present, update them with zeros for all columns.

```
UPDATE `Corona Virus Dataset` SET  
Confirmed = COALESCE(Confirmed, 0),  
Deaths = COALESCE(Deaths, 0),  
Recovered = COALESCE(Recovered, 0);  
SELECT *FROM `Corona Virus Dataset`;
```



Update them with zeros for
all Null Values columns.

Execution Finished
without Errors

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbp [Corona.db]

File
Edit
View
Tools
Help

New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database

Database Structure
Browse Data
Edit Pragmas
Execute SQL

SQL 1

1 UPDATE `Corona Virus Dataset` SET Confirmed = COALESCE(Confirmed, 0), Deaths = COALESCE(Deaths, 0), Recovered = COALESCE(Recovered, 0);
2 SELECT *FROM `Corona Virus Dataset`;
3

| | Province | Country/Region | Latitude | Longitude | Date | Confirmed | Deaths | Recovered |
|----|-------------|----------------|----------|-----------|------------|-----------|--------|-----------|
| 64 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 25-03-2020 | 33 | 1 | 1 |
| 65 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 26-03-2020 | 4 | 1 | 0 |
| 66 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 27-03-2020 | 11 | 0 | 0 |
| 67 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 28-03-2020 | 16 | 1 | 0 |
| 68 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 29-03-2020 | 11 | 0 | 0 |
| 69 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 30-03-2020 | 28 | 0 | 0 |
| 70 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 31-03-2020 | 29 | 0 | 3 |

Execution finished without errors.
Result: 78386 rows returned in 345ms
At line 2:
SELECT *FROM `Corona Virus Dataset`;

SQL Log

Show SQL submitted by Application

2328 SELECT *FROM `Corona Virus Dataset` LIMIT 0, 49999;

2329 SELECT COUNT(*) FROM (SELECT *FROM `Corona Virus Dataset`);

2330 SELECT *FROM `Corona Virus Dataset` LIMIT 0, 49999;

2331 SELECT COUNT(*) FROM (SELECT *FROM `Corona Virus Dataset`);

2332 SELECT *FROM `Corona Virus Dataset` LIMIT 0, 49999;

2333 SELECT COUNT(*) FROM (SELECT *FROM `Corona Virus Dataset`);

2334 SELECT *FROM `Corona Virus Dataset` LIMIT 0, 49999;

2335

check total number of rows

```
SELECT COUNT(*) FROM "Corona virus  
Dataset";
```

Display Total Number Rows
Presented in The Table
Total Number Rows
78386;

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbpro [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT COUNT(*) FROM "Corona virus Dataset" ;
```

| | COUNT(*) |
|---|----------|
| 1 | 78386 |

Execution finished without errors.
Result: 1 rows returned in 80ms
At line 1:
SELECT COUNT(*) FROM "Corona virus Dataset" ;

SQL Log

Show SQL submitted by Application

Results of the last executed statements.
You may want to collapse this panel and use the SQL Log dock with User

Q4. Check what is start_date and end_date

```
SELECT MIN('2020-01-22') AS  
start_date, MAX('2021-06-13') AS  
end_date FROM "Corona Virus  
Dataset";
```

Display start_date
and end_date

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbpro [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT MIN('2020-01-22') AS start_date, MAX('2021-06-13') AS end_date FROM "Corona Virus Dataset";
```

| | start_date | end_date |
|---|------------|------------|
| 1 | 2020-01-22 | 2021-06-13 |

Execution finished without errors.
Result: 1 rows returned in 92ms
At line 1:
SELECT MIN('2020-01-22') AS start_date, MAX('2021-06-13') AS end_date FROM "Corona Virus Dataset";

Number of month present in dataset

```
SELECT COUNT(DISTINCT  
SUBSTR(Date, 5)) AS Number  
_Month FROM `Corona Virus  
Dataset`;
```

Number Of Month
Presented in Dataset

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbpro [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT COUNT(DISTINCT SUBSTR(Date, 5)) AS Number_Month FROM `Corona Virus Dataset`;
```

| | Number_Month |
|---|--------------|
| 1 | 16 |

Execution finished without errors.
Result: 1 rows returned in 166ms
At line 1:
SELECT COUNT(DISTINCT SUBSTR(Date, 5)) AS Number_Month FROM `Corona Virus Dataset`;

Q6. Find monthly average
for confirmed, deaths, recovered

```
SELECT
  SUBSTR(Date,5) AS Month,
  AVG(Confirmed) AS Avg_Confirmed,
  AVG(Deaths) AS Avg_Deaths,
  AVG(Recovered) AS Avg_Recovered
FROM
  `Corona Virus Dataset`
GROUP BY
  SUBSTR(Date, 5);
```

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbpro [Coron

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT
2     SUBSTR(Date, 5) AS Month,
3     AVG(Confirmed) AS Avg_Confirmed,
4     AVG(Deaths) AS Avg_Deaths,
5     AVG(Recovered) AS Avg_Recovered
6 FROM
7     `Corona Virus Dataset`
8 GROUP BY
9     SUBSTR(Date, 5);
10
```

| | Month | Avg_Confirmed | Avg_Deaths | Avg_Recovered |
|---|--------|------------------|------------------|------------------|
| 1 | 0-2020 | 2412.19962295769 | 36.7582739840804 | 1420.64306661081 |
| 2 | 1-2020 | 2695.18214285714 | 42.6034090909091 | 1489.03165584416 |
| 3 | 1-2021 | 3911.22852953498 | 84.1837033933808 | 1919.63699204022 |
| 4 | 2-2020 | 2100.12023809524 | 37.0830086580087 | 1293.97272727273 |
| 5 | 2-2021 | 2433.36363636364 | 69.1648886827458 | 1558.39169758813 |
| 6 | 3-2020 | 161.130289065773 | 8.66066191872643 | 27.8739002932551 |
| 7 | 3-2021 | 2916.79723502304 | 59.1998324256389 | 1652.28592375367 |

Execution finished without errors.
Result: 16 rows returned in 365ms
At line 1:
SELECT
 SUBSTR(Date, 5) AS Month,

Find most frequent value for confirmed, deaths, recovered each month

SELECT SUBSTR(Date, 5) AS Month,
MAX[Confirmed] AS
Most_Frequent_Confirmed, MAX[Deaths] AS
Most_Frequent_Deaths, MAX[Recovered] AS
Most_Frequent_Recovered
FROM
`Corona Virus Dataset`
GROUP BY
SUBSTR(Date,5);

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbp [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT SUBSTR(Date, 5) AS Month, MAX(Confirmed) AS Most_Frequent_Confirmed, MAX(Deaths) AS Most_Frequent_Deaths, MAX(Recovered) AS Most_Frequent_Recovered
2 FROM
3 `Corona Virus Dataset`
4 GROUP BY
5 SUBSTR(Date,5);
6
7
8
```

| | Month | Most_Frequent_Confirmed | Most_Frequent_Deaths | Most_Frequent_Recovered |
|---|--------|-------------------------|----------------------|-------------------------|
| 1 | 0-2020 | 99264 | 3351 | 388340 |
| 2 | 1-2020 | 207933 | 2259 | 139292 |
| 3 | 1-2021 | 300462 | 4475 | 87090 |
| 4 | 2-2020 | 823225 | 3752 | 1123456 |
| 5 | 2-2021 | 134975 | 3907 | 98389 |
| 6 | 3-2020 | 26314 | 1085 | 4289 |
| 7 | 3-2021 | 100158 | 3869 | 102138 |

Execution finished without errors.
Result: 16 rows returned in 360ms
At line 1:
SELECT SUBSTR(Date, 5) AS Month, MAX(Confirmed) AS Most_Frequent_Confirmed, MAX(Deaths) AS Most_Frequent_Deaths, MAX(Recovered) AS Most_Frequent_Recovered
FROM

SQL Log

Show SQL submitted by Application

```
961 GROUP BY
962 SUBSTR(Date, 5) LIMIT 0, 49999;
963 ROLLBACK TO SAVEPOINT "RESTOREPOINT";
964 PRAGMA database_list;
965 SELECT type,name,sql,tbl_name FROM "main".sqlite_master;
966 RELEASE "RESTOREPOINT";
967
```

Clear

Mode: Text

NULL

Type of data currently in cell: NULL
0 byte(s)

Find minimum values for confirmed, deaths, recovered per year

```
SELECT SUBSTR(Date, 7, 5) AS Year,
Min[Confirmed] AS Min_Confirmed,
Min[Deaths] AS Min_Deaths,
Min[Recovered] AS Min_Recovered
FROM `Corona Virus Dataset` GROUP BY
SUBSTR(Date, 7, 5);
```

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbpro [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save P

Database Structure Browse Data Edit Pragma Execute SQL

SQL 1

```
1 SELECT
2     SUBSTR(Date, 7, 5) AS Year,
3     Min(Confirmed) AS Min_Confirmed,
4     Min(Deaths) AS Min_Deaths,
5     Min(Recovered) AS Min_Recovered
6 FROM
7     `Corona Virus Dataset`
8 GROUP BY
9     SUBSTR(Date, 7, 5);
10
```

| | Year | Min_Confirmed | Min_Deaths | Min_Recovered |
|---|------|---------------|------------|---------------|
| 1 | 2020 | 0 | 0 | 0 |
| 2 | 2021 | 0 | 0 | 0 |

Execution finished without errors.
Result: 2 rows returned in 462ms
At line 1:
SELECT
 SUBSTR(Date, 7, 5) AS Year,

SQL Log

Find maximum values of confirmed, deaths, recovered per year

SELECT SUBSTR(Date, 7, 5) AS Year,
Max[Confirmed] AS Max_Confirmed,
Max[Deaths] AS Max_Deaths,
Max[Recovered] AMax_Recovered
FROM `Corona Virus Dataset`
GROUP BY SUBSTR(Date, 7, 5);

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbpro [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT
2     SUBSTR(Date, 7, 5) AS Year,
3     Max(Confirmed) AS Min_Confirmed,
4     Max(Deaths) AS Min_Deaths,
5     Max(Recovered) AS Min_Recovered
6 FROM
7     `Corona Virus Dataset`
8 GROUP BY
9     SUBSTR(Date, 7, 5);
10
```

| | Year | Min_Confirmed | Min_Deaths | Min_Recovered |
|---|------|---------------|------------|---------------|
| 1 | 2020 | 823225 | 3752 | 1123456 |
| 2 | 2021 | 414188 | 7374 | 422436 |

Execution finished without errors.

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

```
SELECT SUBSTR(Date ,4 ) AS Month,  
SUM(Confirmed) AS Total_Confirmed,  
SUM(Deaths) AS Total_Deaths,  
SUM(Recovered) AS  
Total_RecoveredFROM  
`CoronaVirusDataset`  
GROUP BY SUBSTR(Date , 4);
```

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbpro [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save F

Database Structure Browse Data Edit Pragma Execute SQL

SQL 1

```
1 SELECT  
2     SUBSTR(Date ,4 ) AS Month,  
3     SUM(Confirmed) AS Total_Confirmed,  
4     SUM(Deaths) AS Total_Deaths,  
5     SUM(Recovered) AS Total_Recovered  
6 FROM  
7     `Corona Virus Dataset`  
8 GROUP BY  
9     SUBSTR(Date , 4);  
10
```

| | Month | Total_Confirmed | Total_Deaths | Total_Recovered |
|---|---------|-----------------|--------------|-----------------|
| 1 | 01-2020 | 6384 | 190 | 143 |
| 2 | 01-2021 | 18672205 | 401893 | 9164347 |
| 3 | 02-2020 | 68312 | 2651 | 31405 |
| 4 | 02-2021 | 10492664 | 298239 | 6719785 |
| 5 | 03-2020 | 769236 | 41346 | 133070 |
| 6 | 03-2021 | 13924790 | 282620 | 7888013 |
| 7 | 04-2020 | 2336798 | 191833 | 792987 |

Execution finished without errors.
Result: 18 rows returned in 2220ms
At line 1:
SELECT
 SUBSTR(Date ,4) AS Month,

Check how corona virus spread
out with respect to confirmed
case

--a- Calculate total confirmed cases
SELECT
SUM(Confirmed) AS
Total_Confirmed_Cases
FROM
`Corona Virus Dataset`;

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbp [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project

Database Structure Browse Data Edit Pragas Execute SQL

SQL 1

```
--Check how corona virus spread out with respect to confirmed case
-- Calculate total confirmed cases
SELECT
    SUM(Confirmed) AS Total_Confirmed_Cases
FROM
    `Corona Virus Dataset`;
```

| | Total_Confirmed_Cases |
|---|-----------------------|
| 1 | 169065144 |

Execution finished without errors.
Result: 1 rows returned in 131ms
At line 1:
--Check how corona virus spread out with respect to confirmed case
-- Calculate total confirmed cases

SQL Log

Show SQL submitted by Application

```
464 SUM(Confirmed) AS Total_Confirmed_Cases
465 FROM
466 `Corona Virus Dataset` LIMIT 0, 49999;
467 ROLLBACK TO SAVEPOINT "RESTOREPOINT";
468 PRAGMA database_list;
469 SELECT type,name,sql,tbl_name FROM "main".sqlite_master;
470 RELEASE "RESTOREPOINT";
471
```

Showing Total Comfirmed Cases in
The DataSet

Check how corona virus spread
out with respect to confirmed
case

SELECT
avg(Confirmed) AS
Total_Confirmed_Cases
FROM
`Corona Virus Dataset`;

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona.sqbp [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT
2   avg(Confirmed) AS Average_Confirmed_Cases
3 FROM
4   `Corona Virus Dataset`;
5
6
```

| | Average_Confirmed_Cases |
|---|-------------------------|
| 1 | 2156.82831117802 |

Execution finished without errors.
Result: 1 rows returned in 113ms
At line 1:
SELECT
avg(Confirmed) AS Average Confirmed Cases

SQL Log

Show SQL submitted by Application

```
133   avg(Confirmed) AS Average_Confirmed_Cases
134 FROM
135   `Corona Virus Dataset` LIMIT 0, 49999;
136 ROLLBACK TO SAVEPOINT "RESTOREPOINT";
137 PRAGMA database_list;
138 SELECT type,name,sql,tbl_name FROM "main".sqlite_master;
139 RELEASE "RESTOREPOINT";
140
```

Showing Avg Comfirmed
Cases in The DataSet By using
Avg Function

Check how corona virus spread
out with respect to confirmed
case

SELECT
SUM[(Confirmed -
Mean_Confirmed) * (Confirmed -
Mean_Confirmed)] / COUNT(*) AS
Variance_Confirmed_Cases
FROM
'Corona Virus Dataset';

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona.sqbpro [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragma Execute SQL

SQL 1

```
1 SELECT
2     AVG(Confirmed * Confirmed) - AVG(Confirmed) * AVG(Confirmed) AS Variance_Confirmed_Cases
3 FROM
4     `Corona Virus Dataset`;
5
6
7
8
```

| | Variance_Confirmed_Cases |
|---|--------------------------|
| 1 | 157288925.077966 |

Execution finished without errors.
Result: 1 rows returned in 132ms
At line 1:
SELECT
 AVG(Confirmed * Confirmed) - AVG(Confirmed) * AVG(Confirmed) AS Variance_Confirmed_Cases

SQL Log

Show SQL submitted by Application

```
181     AVG(Confirmed * Confirmed) - AVG(Confirmed) * AVG(Confirmed) AS Variance_Confirmed_Cases
182 FROM
183     `Corona Virus Dataset` LIMIT 0, 49999;
184 ROLLBACK TO SAVEPOINT "RESTOREPOINT";
185 PRAGMA database_list;
186 SELECT type,name,sql,tbl_name FROM "main".sqlite_master;
187 RELEASE "RESTOREPOINT";
188
```

Showing Variance Comfirmed
Cases in The DataSet By using
Stddv Function

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

SELECT
SUBSTR(Date,2) AS Month,
SUM(Deaths) AS Total_Death_Cases
FROM
`Corona Virus Dataset`
GROUP BY
SUBSTR(Date,4);

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbpro [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
-- Calculate total death cases per month
SELECT
  SUBSTR(Date,2) AS Month,
  SUM(Deaths) AS Total_Death_Cases
FROM
  `Corona Virus Dataset`
GROUP BY
  SUBSTR(Date,4);
```

| | Month | Total_Death_Cases |
|----|-----------|-------------------|
| 7 | 1-04-2020 | 191833 |
| 8 | 1-04-2021 | 362387 |
| 9 | 1-05-2020 | 144561 |
| 10 | 1-05-2021 | 366549 |
| 11 | 1-06-2020 | 137757 |
| 12 | 1-06-2021 | 132657 |
| 13 | 1-07-2020 | 167613 |

Execution finished without errors.
Result: 18 rows returned in 308ms
At line 1:
-- Calculate total death cases per month
SELECT

SQL Log

Show SQL submitted by Application

```
1239 `Corona Virus Dataset`
1240 GROUP BY
1241 SUBSTR(Date,4) LIMIT 0, 49999;
1242 ROLLBACK TO SAVEPOINT "RESTOREPOINT";
1243 PRAGMA database_list;
1244 SELECT type,name,sql,tbl_name FROM "main".sqlite_master;
1245 RELEASE "RESTOREPOINT";
1246
```

Showing Total Death
Cases in The DataSet By using
Function

Check how corona virus spread
out with respect to confirmed
case

SELECT
SUBSTR(Date,2) AS Month,
avg(Deaths) AS Total_Death_Cases
FROM
`Corona Virus Dataset`
GROUP BY
SUBSTR(Date,4);

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbpro [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Data

Database Structure Browse Data Edit Pragma Execute SQL

SQL 1

```
1 -- Calculate total death cases per month
2 SELECT
3     SUBSTR(Date,2) AS Month,
4     avg(Deaths) AS Total_Death_Cases
5 FROM
6     `Corona Virus Dataset`
7 GROUP BY
8     SUBSTR(Date,4);
9
10
```

| | Month | Total_Death_Cases |
|---|-----------|-------------------|
| 1 | 2-01-2020 | 0.123376623376623 |
| 2 | 1-01-2021 | 84.1837033933808 |
| 3 | 1-02-2020 | 0.593596059113301 |
| 4 | 1-02-2021 | 69.1648886827458 |
| 5 | 1-03-2020 | 8.66066191872643 |
| 6 | 1-03-2021 | 59.1998324256389 |
| 7 | 1-04-2020 | 41.5222943722944 |

Execution finished without errors.
Result: 18 rows returned in 305ms
At line 1:
-- Calculate total death cases per month
SELECT

SQL Log

Show SQL submitted by Application

```
1258 `Corona Virus Dataset`
1259 GROUP BY
1260     SUBSTR(Date,4) LIMIT 0, 49999;
1261 ROLLBACK TO SAVEPOINT "RESTOREPOINT";
1262 PRAGMA database_list;
1263 SELECT type,name,sql,tbl_name FROM "main".sqlite_master;
1264 RELEASE "RESTOREPOINT";
1265
```

Showing Total Death
Cases in The DataSet By using
Avg Function

Check how corona virus spread
out with respect to confirmed
case

SELECT
SUBSTR(Date, 2) AS Month,
[sum[Confirmed * Confirmed] / COUNT(*)]
- [SUM[Confirmed] * SUM[Confirmed] /
COUNT(*) / COUNT(*)] AS
Variance_Recovered_Cases
FROM
`Corona Virus Dataset`
GROUP BY
SUBSTR(Date, 4);

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbp [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close

Database Structure Browse Data Edit Pragas Execute SQL

SQL 1

```
-- Calculate variance of death cases per month
SELECT
    SUBSTR(Date, 2) AS Month,
    (SUM(Deaths * Deaths) / COUNT(*) - (SUM(Deaths) * SUM(Deaths) / COUNT(*) / COUNT(*) AS Variance_Death_
FROM
    `Corona Virus Dataset`
GROUP BY
    SUBSTR(Date, 4);
```

| | Month | Variance_Death_Cases |
|----|-----------|----------------------|
| 7 | 1-04-2020 | 40504 |
| 8 | 1-04-2021 | 94612 |
| 9 | 1-05-2020 | 20685 |
| 10 | 1-05-2021 | 131769 |
| 11 | 1-06-2020 | 16929 |
| 12 | 1-06-2021 | 112964 |
| 13 | 1-07-2020 | 21140 |

Execution finished without errors.
Result: 18 rows returned in 397ms
At line 1:
-- Calculate variance of death cases per month
SELECT

SQL Log

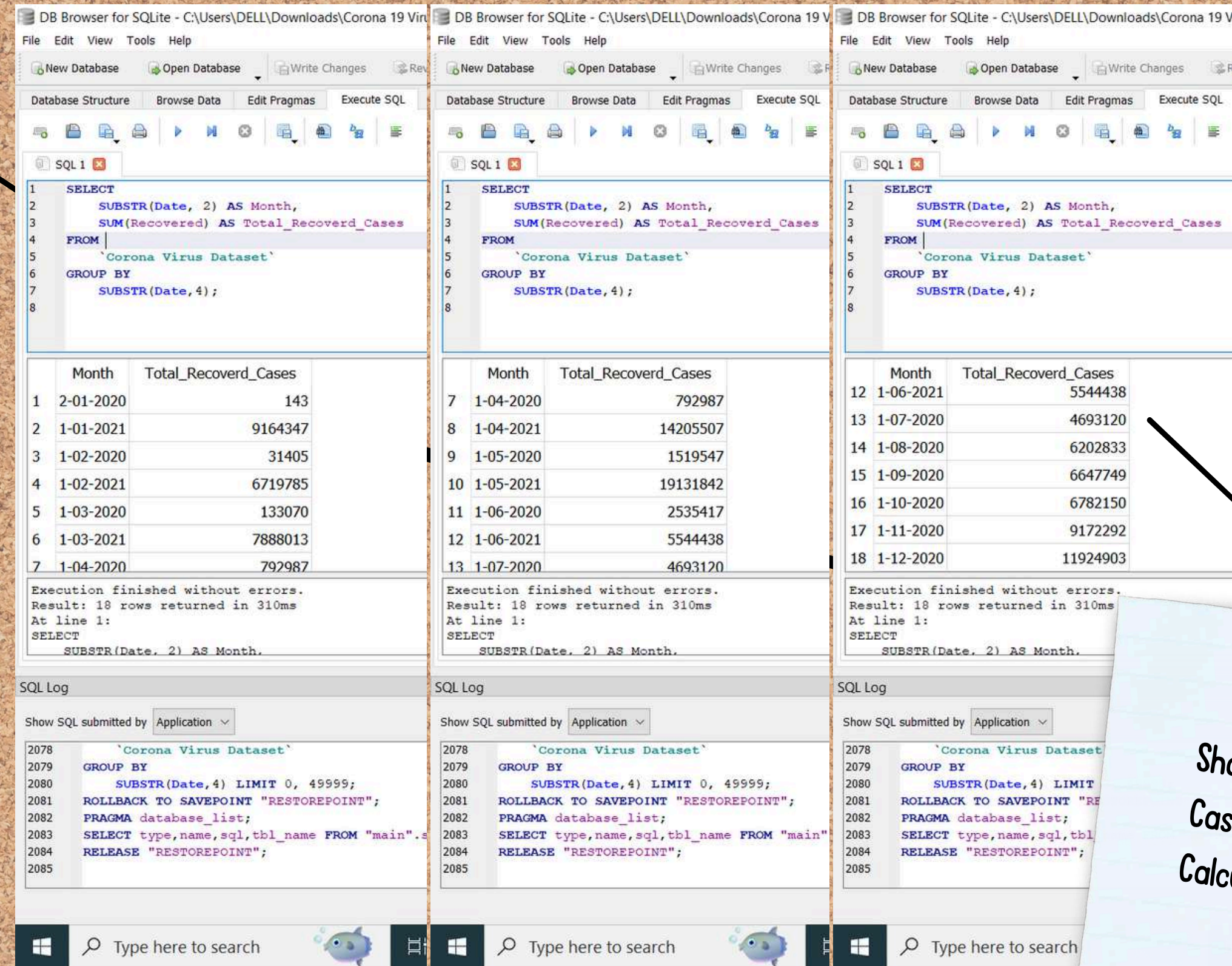
Show SQL submitted by Application

```
1463 `Corona Virus Dataset`
1464 GROUP BY
1465 SUBSTR(Date, 4) LIMIT 0, 49999;
1466 ROLLBACK TO SAVEPOINT "RESTOREPOINT";
1467 PRAGMA database_list;
1468 SELECT type,name,sql,tbl_name FROM "main".sqlite_master;
1469 RELEASE "RESTOREPOINT";
1470
```

Showing Total Death
Cases in The DataSet By
Calculating Variance

13. Check how corona virus spread out with respect to recovered case

Syntax:-
SELECT SUBSTR(Date, 2) AS Month,
[sum[Confirmed * Confirmed] / COUNT(*)] -
[SUM[Confirmed] * SUM[Confirmed] /
COUNT(*) / COUNT(*)] AS
Variance_Recovered_Cases
FROM
'Corona Virus Dataset'
GROUP BY
SUBSTR(Date, 4);



DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Vir

File Edit View Tools Help

New Database Open Database Write Changes

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT
2   SUBSTR(Date, 2) AS Month,
3   SUM(Recovered) AS Total_Recoverd_Cases
4 FROM
5   'Corona Virus Dataset'
6 GROUP BY
7   SUBSTR(Date, 4);
8
```

| | Month | Total_Recoverd_Cases |
|---|-----------|----------------------|
| 1 | 2-01-2020 | 143 |
| 2 | 1-01-2021 | 9164347 |
| 3 | 1-02-2020 | 31405 |
| 4 | 1-02-2021 | 6719785 |
| 5 | 1-03-2020 | 133070 |
| 6 | 1-03-2021 | 7888013 |
| 7 | 1-04-2020 | 792987 |

Execution finished without errors.
Result: 18 rows returned in 310ms
At line 1:
SELECT
SUBSTR(Date, 2) AS Month.

SQL Log

Show SQL submitted by Application

2078 'Corona Virus Dataset'
2079 GROUP BY
2080 SUBSTR(Date, 4) LIMIT 0, 49999;
2081 ROLLBACK TO SAVEPOINT "RESTOREPOINT";
2082 PRAGMA database_list;
2083 SELECT type, name, sql, tbl_name FROM "main".s
2084 RELEASE "RESTOREPOINT";
2085

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 V

File Edit View Tools Help

New Database Open Database Write Changes

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT
2   SUBSTR(Date, 2) AS Month,
3   SUM(Recovered) AS Total_Recoverd_Cases
4 FROM
5   'Corona Virus Dataset'
6 GROUP BY
7   SUBSTR(Date, 4);
8
```

| | Month | Total_Recoverd_Cases |
|----|-----------|----------------------|
| 7 | 1-04-2020 | 792987 |
| 8 | 1-04-2021 | 14205507 |
| 9 | 1-05-2020 | 1519547 |
| 10 | 1-05-2021 | 19131842 |
| 11 | 1-06-2020 | 2535417 |
| 12 | 1-06-2021 | 5544438 |
| 13 | 1-07-2020 | 4693120 |

Execution finished without errors.
Result: 18 rows returned in 310ms
At line 1:
SELECT
SUBSTR(Date, 2) AS Month.

SQL Log

Show SQL submitted by Application

2078 'Corona Virus Dataset'
2079 GROUP BY
2080 SUBSTR(Date, 4) LIMIT 0, 49999;
2081 ROLLBACK TO SAVEPOINT "RESTOREPOINT";
2082 PRAGMA database_list;
2083 SELECT type, name, sql, tbl_name FROM "main".s
2084 RELEASE "RESTOREPOINT";
2085

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 V

File Edit View Tools Help

New Database Open Database Write Changes

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT
2   SUBSTR(Date, 2) AS Month,
3   SUM(Recovered) AS Total_Recoverd_Cases
4 FROM
5   'Corona Virus Dataset'
6 GROUP BY
7   SUBSTR(Date, 4);
8
```

| | Month | Total_Recoverd_Cases |
|----|-----------|----------------------|
| 12 | 1-06-2021 | 5544438 |
| 13 | 1-07-2020 | 4693120 |
| 14 | 1-08-2020 | 6202833 |
| 15 | 1-09-2020 | 6647749 |
| 16 | 1-10-2020 | 6782150 |
| 17 | 1-11-2020 | 9172292 |
| 18 | 1-12-2020 | 11924903 |

Execution finished without errors.
Result: 18 rows returned in 310ms
At line 1:
SELECT
SUBSTR(Date, 2) AS Month.

SQL Log

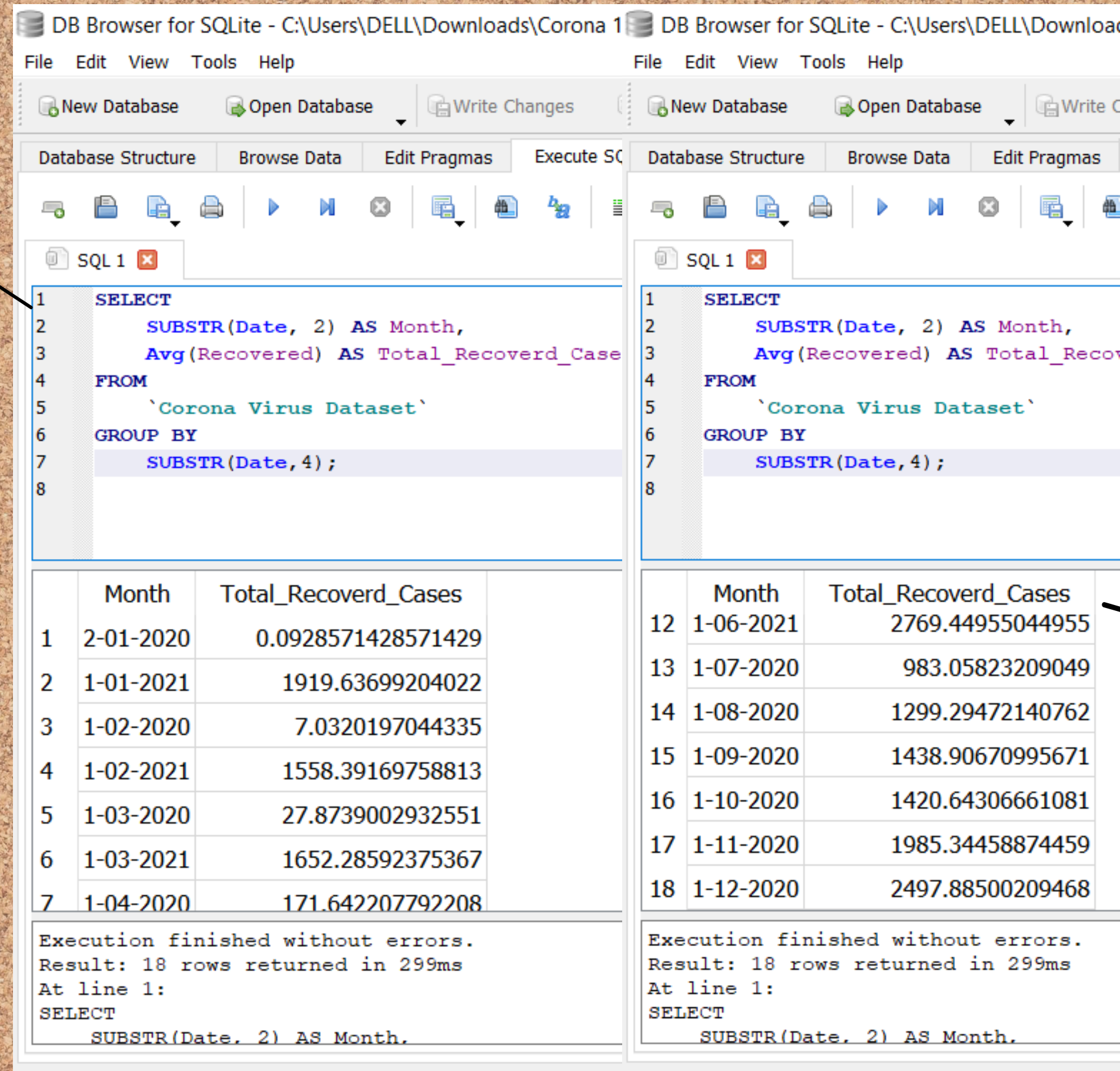
Show SQL submitted by Application

2078 'Corona Virus Dataset'
2079 GROUP BY
2080 SUBSTR(Date, 4) LIMIT 0, 49999;
2081 ROLLBACK TO SAVEPOINT "RESTOREPOINT";
2082 PRAGMA database_list;
2083 SELECT type, name, sql, tbl_name FROM "main".s
2084 RELEASE "RESTOREPOINT";
2085

Showing Total Recovery Cases in The DataSet By Calculating Sum Function

13. Check how corona virus spread out with respect to recovered case

Syntax:-
SELECT
SUBSTR(Date, 2) AS Month,
SUM(Recovered) AS
Total_Recoverd_Cases
FROM
'Corona Virus Dataset'
GROUP BY
SUBSTR(Date,4);



DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 1

File Edit View Tools Help

New Database Open Database Write Changes

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT
2     SUBSTR(Date, 2) AS Month,
3     Avg(Recovered) AS Total_Recoverd_Case
4 FROM
5     `Corona Virus Dataset`
6 GROUP BY
7     SUBSTR(Date, 4);
8
```

| | Month | Total_Recoverd_Cases |
|---|-----------|----------------------|
| 1 | 2-01-2020 | 0.0928571428571429 |
| 2 | 1-01-2021 | 1919.63699204022 |
| 3 | 1-02-2020 | 7.0320197044335 |
| 4 | 1-02-2021 | 1558.39169758813 |
| 5 | 1-03-2020 | 27.8739002932551 |
| 6 | 1-03-2021 | 1652.28592375367 |
| 7 | 1-04-2020 | 171.642207792208 |

Execution finished without errors.
Result: 18 rows returned in 299ms
At line 1:
SELECT
SUBSTR(Date, 2) AS Month,

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 1

File Edit View Tools Help

New Database Open Database Write Changes

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT
2     SUBSTR(Date, 2) AS Month,
3     Avg(Recovered) AS Total_Recoverd_Case
4 FROM
5     `Corona Virus Dataset`
6 GROUP BY
7     SUBSTR(Date, 4);
8
```

| | | |
|----|-----------|------------------|
| 12 | 1-06-2021 | 2769.44955044955 |
| 13 | 1-07-2020 | 983.05823209049 |
| 14 | 1-08-2020 | 1299.29472140762 |
| 15 | 1-09-2020 | 1438.90670995671 |
| 16 | 1-10-2020 | 1420.64306661081 |
| 17 | 1-11-2020 | 1985.34458874459 |
| 18 | 1-12-2020 | 2497.88500209468 |

Execution finished without errors.
Result: 18 rows returned in 299ms
At line 1:
SELECT
SUBSTR(Date, 2) AS Month,

Showing Total Recovery Cases & Their Average in The DataSet By Calculating Avg Function

Find Country having highest
number of the Confirmed case

SELECT "Country/Region" AS
Country, MAX(Confirmed) AS
Highest_Confirmed_Cases FROM
'Corona Virus Dataset';

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbp [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Data

Database Structure Browse Data Edit Pragma Execute SQL

SQL 1

```
--Find top 5 countries having highest recovered case  
SELECT "Country/Region" AS Country, MAX(Confirmed) AS Highest_Confirmed_Cases FROM `Corona Virus Dataset`
```

| | Country | Highest_Confirmed_Cases |
|---|---------|-------------------------|
| 1 | Turkey | 823225 |

Execution finished without errors.
Result: 1 rows returned in 96ms
At line 1:
--Find top 5 countries having highest recovered case
SELECT "Country/Region" AS Country, MAX(Confirmed) AS Highest_Confirmed_Cases FROM `Corona Virus Dataset`

SQL Log

Show SQL submitted by Application

```
2175 SAVEPOINT "RESTOREPOINT";  
2176 SELECT COUNT(*) FROM (SELECT "Country/Region" AS Country, MAX(Confirmed) AS Highest_Confirmed_Cases FROM `Corona Virus Dataset`);  
2177 SELECT "Country/Region" AS Country, MAX(Confirmed) AS Highest_Confirmed_Cases FROM `Corona Virus Dataset`;  
2178 ROLLBACK TO SAVEPOINT "RESTOREPOINT";  
2179 PRAGMA database_list;  
2180 SELECT type,name,sql,tbl_name FROM "main".sqlite_master;  
2181 RELEASE "RESTOREPOINT";  
2182
```

Showing Highest Confirmed
Cases in The DataSet By Max
Function

Find Country having lowest
number of the death case

SELECT "Country/Region" AS
Country, Min(Deaths) AS
lowest_Deaths_Cases FROM
`Corona Virus Dataset`;

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbbpro [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
-- Find Country having lowest number of the death case  
SELECT "Country/Region" AS Country, Min(Deaths) AS lowest_Deaths_Cases FROM `Corona Virus Dataset`;
```

| | Country | lowest_Deaths_Cases |
|---|-------------|---------------------|
| 1 | Afghanistan | 0 |

Execution finished without errors.
Result: 1 rows returned in 107ms
At line 1:
-- Find Country having lowest number of the death case
SELECT "Country/Region" AS Country, Min(Deaths) AS lowest_Deaths_Cases FROM `Corona Virus Dataset`;

SQL Log

Show SQL submitted by Application

```
2221 SAVEPOINT "RESTOREPOINT";  
2222 SELECT COUNT(*) FROM (SELECT "Country/Region" AS Country, Min(Deaths) AS lowest_Deaths_Cases FROM  
2223 SELECT "Country/Region" AS Country, Min(Deaths) AS lowest_Deaths_Cases FROM `Corona Virus Dataset`  
2224 ROLLBACK TO SAVEPOINT "RESTOREPOINT";  
2225 PRAGMA database_list;  
2226 SELECT type,name,sql,tbl_name FROM "main".sqlite_master;  
2227 RELEASE "RESTOREPOINT";  
2228
```

Showing Lowest Number Death
Cases in The DataSet By Min
Function

Q 16. Find top 5 countries having highest recovered case

```
SELECT "Country/Region" AS
Country, SUM(Recovered) AS
Total_Recovered_Cases FROM
`Corona Virus Dataset`
GROUP BY "Country/Region" ORDER
BY Total_Recovered_Cases DESC
LIMIT 5;
```

DB Browser for SQLite - C:\Users\DELL\Downloads\Corona 19 Virus .csv.sqbp [Corona.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragma Execute SQL

SQL 1

```
--Find top 5 countries having highest recovered case
SELECT "Country/Region" AS Country, SUM(Recovered) AS Total_Recovered_Cases FROM `Corona Virus Dataset`
GROUP BY "Country/Region" ORDER BY Total_Recovered_Cases DESC LIMIT 5;
```

| | Country | Total_Recovered_Cases |
|---|---------|-----------------------|
| 1 | India | 28089649 |
| 2 | Brazil | 15400169 |
| 3 | US | 6303715 |
| 4 | Turkey | 5202251 |
| 5 | Russia | 4745756 |

Execution finished without errors.
Result: 5 rows returned in 300ms
At line 1:
--Find top 5 countries having highest recovered case
SELECT "Country/Region" AS Country, SUM(Recovered) AS Total_Recovered_Cases FROM `Corona Virus Dataset`

SQL Log

Show SQL submitted by Application

```
2255 LGROUP BY "Country/Region" ORDER BY Total_Recovered_Cases DESC LIMIT 5);
2256 SELECT "Country/Region" AS Country, SUM(Recovered) AS Total_Recovered_Cases FROM `Corona Virus Data
2257 GROUP BY "Country/Region" ORDER BY Total_Recovered_Cases DESC LIMIT 5
2258 ROLLBACK TO SAVEPOINT "RESTOREPOINT";
2259 PRAGMA database_list;
2260 SELECT type, name, sql, tbl_name FROM "main".sqlite_master;
2261 RELEASE "RESTOREPOINT";
2262
```

Showing Top 5 Highest Recovery Country in The DataSet

Conclusion:-

Reflection on the significance of using SQLite for COVID-19 data analysis.

Potential areas for future research or analysis.

Summary of key findings from the analysis.

References:-

Acknowledgment of any libraries or tools utilized (e.g., SQLite).

List of data sources, references, and resources used in the presentation.

Connect with us.



Github

<https://github.com/saravanaaayyapa28>



Social Media

www.linkedin.com/in/saravana28