# SARS-CoV-2, analysis using sqlite3

Saravana.M

Power By Mentorness

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)

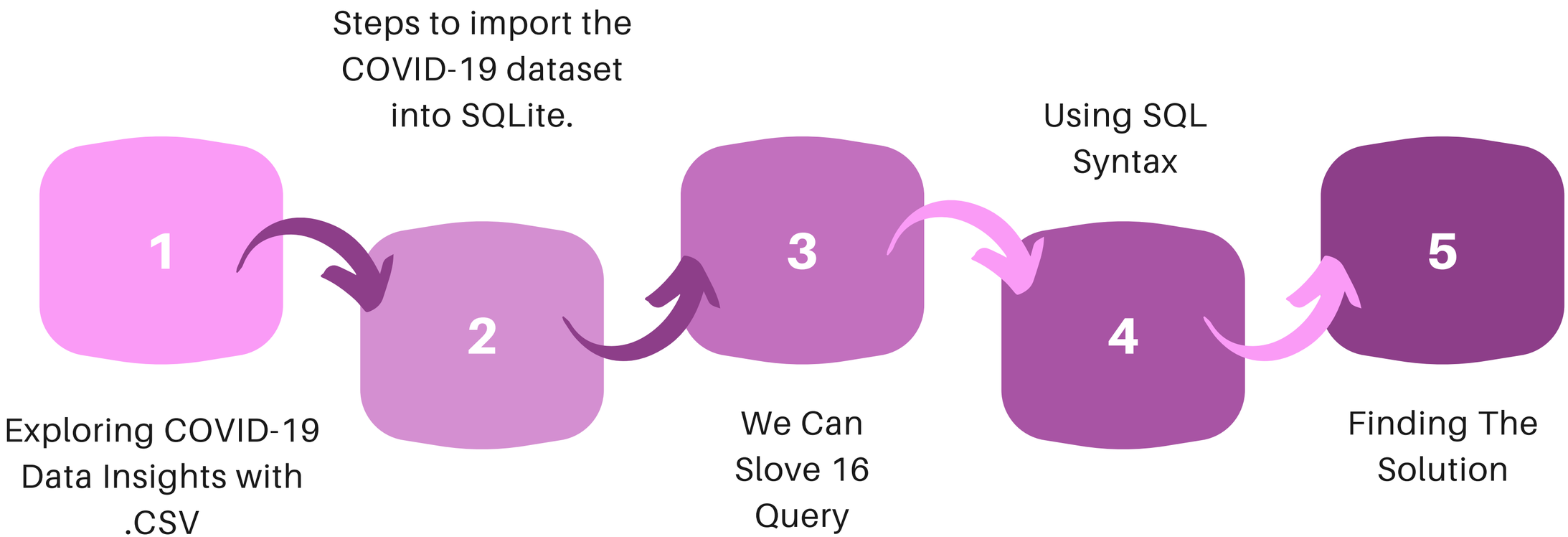
5

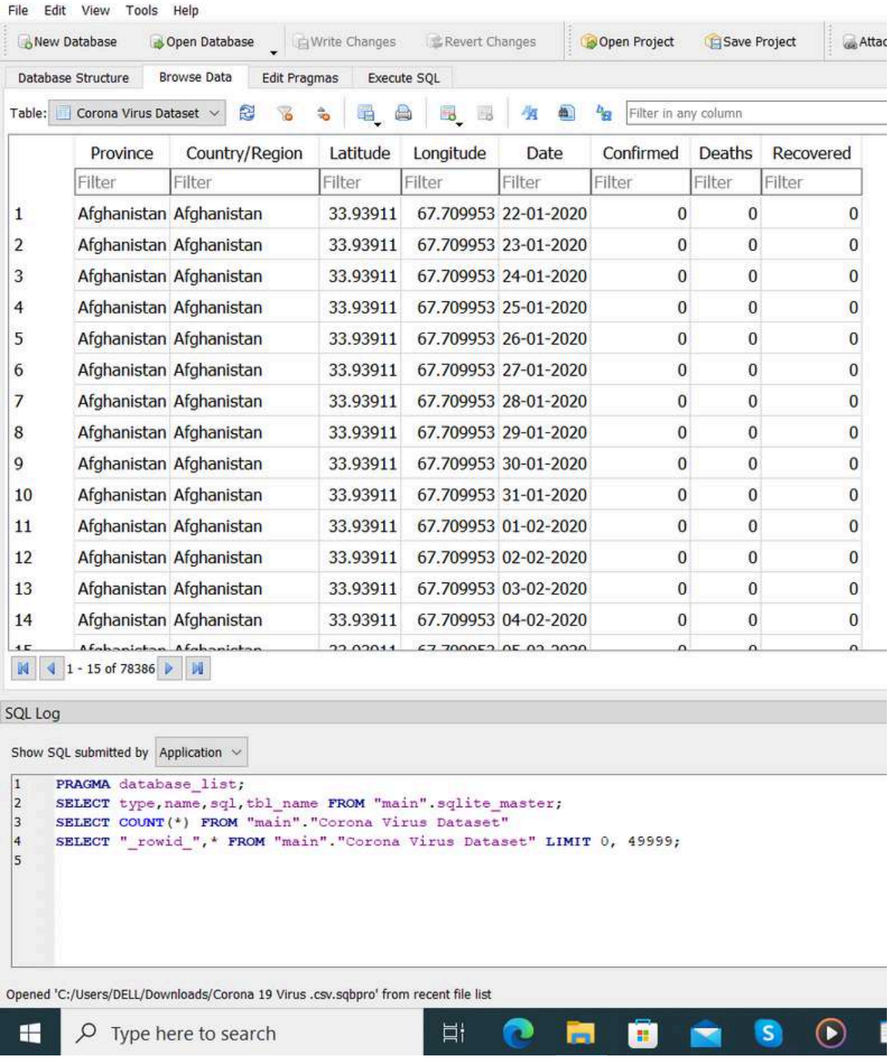
Clarifying with Question & Answer

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

# INTENTION

Analyzing Trends and Understanding Implications






## 1 INSTALL SQLITE:

If you haven't already, install SQLite on your system. You can download it from the official website or use a package manager for your operating system.

## 2 OPEN SQLITE

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

## 3 NAVIGATE TO YOUR SQLITE DATABASE LOCATION

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

## 4 IMPORT CSV DATASET

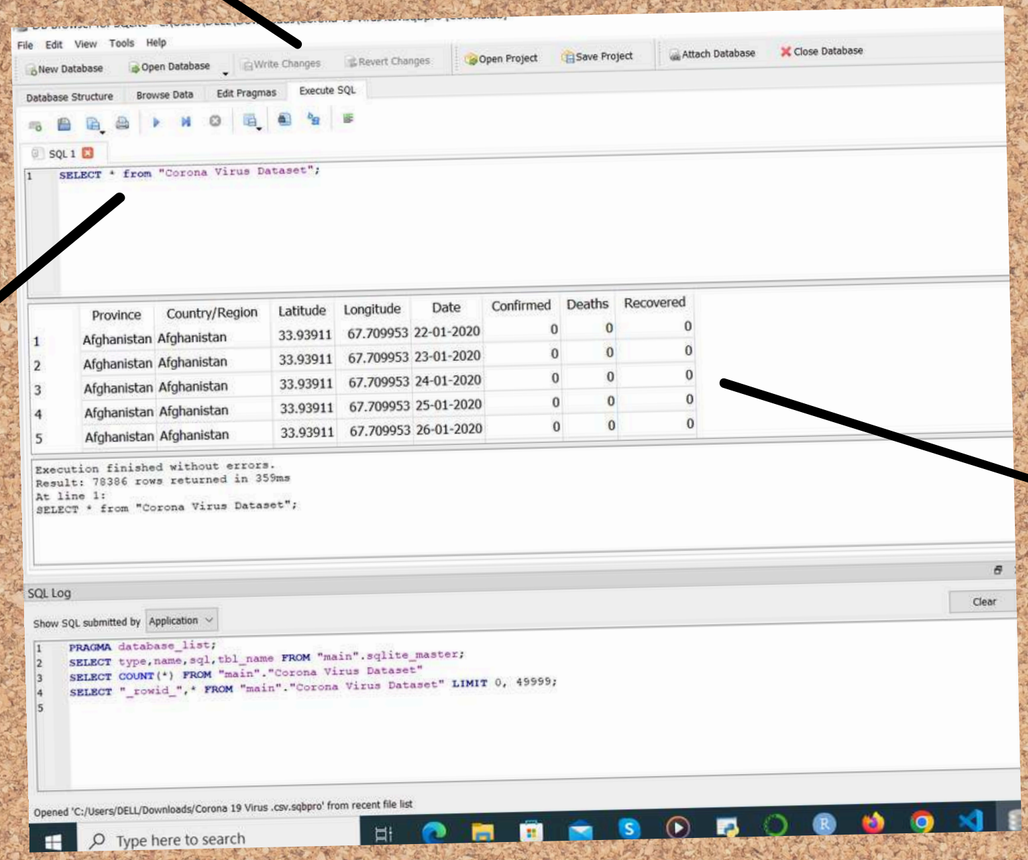
Use the SQLite **.import** command to import the CSV data into a new or existing table.

## 5 VERIFY DATA IMPORT

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

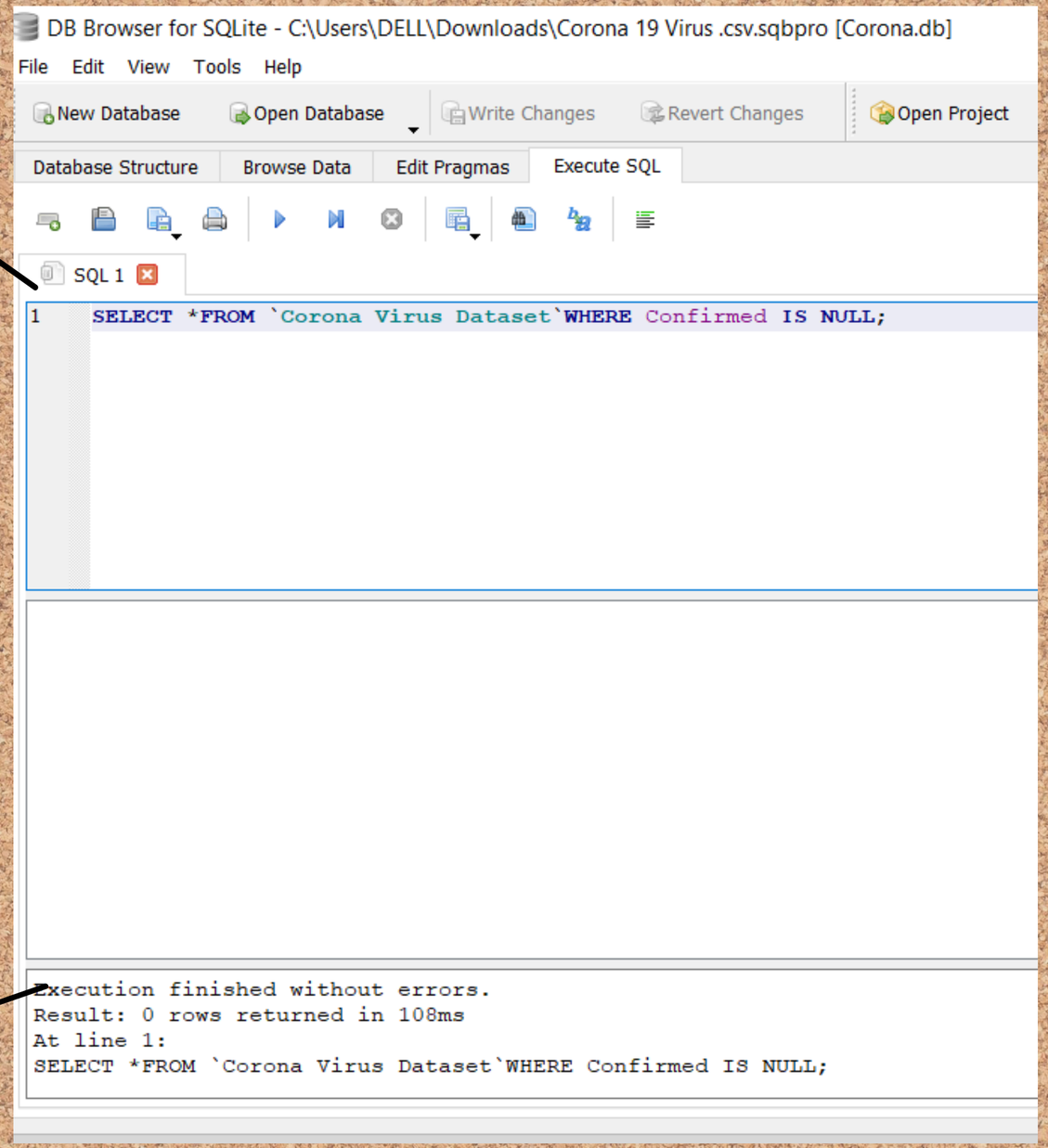
SELECT * FROM Corona Virus Dataset;

# 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;



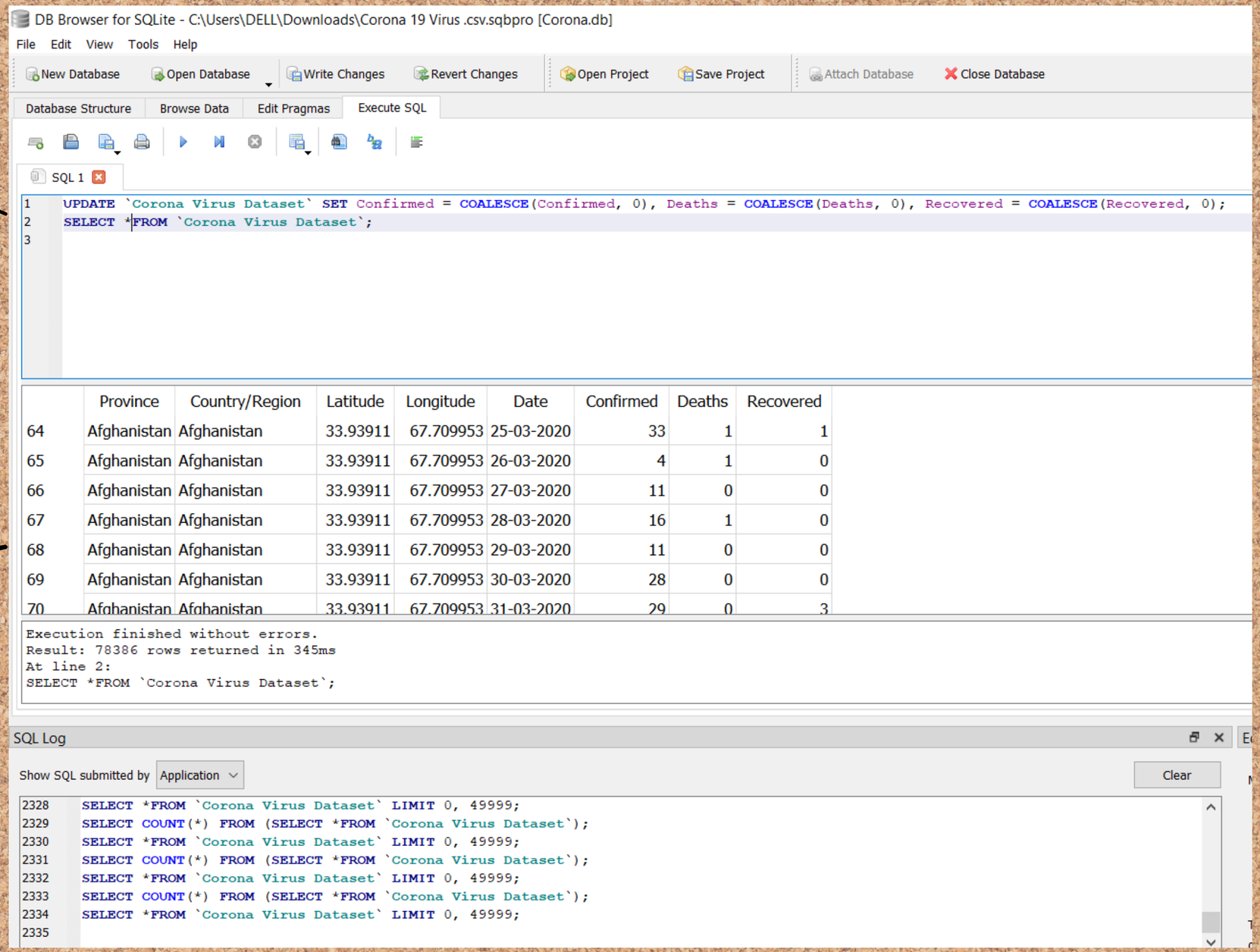

Executing Successful without Errors

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



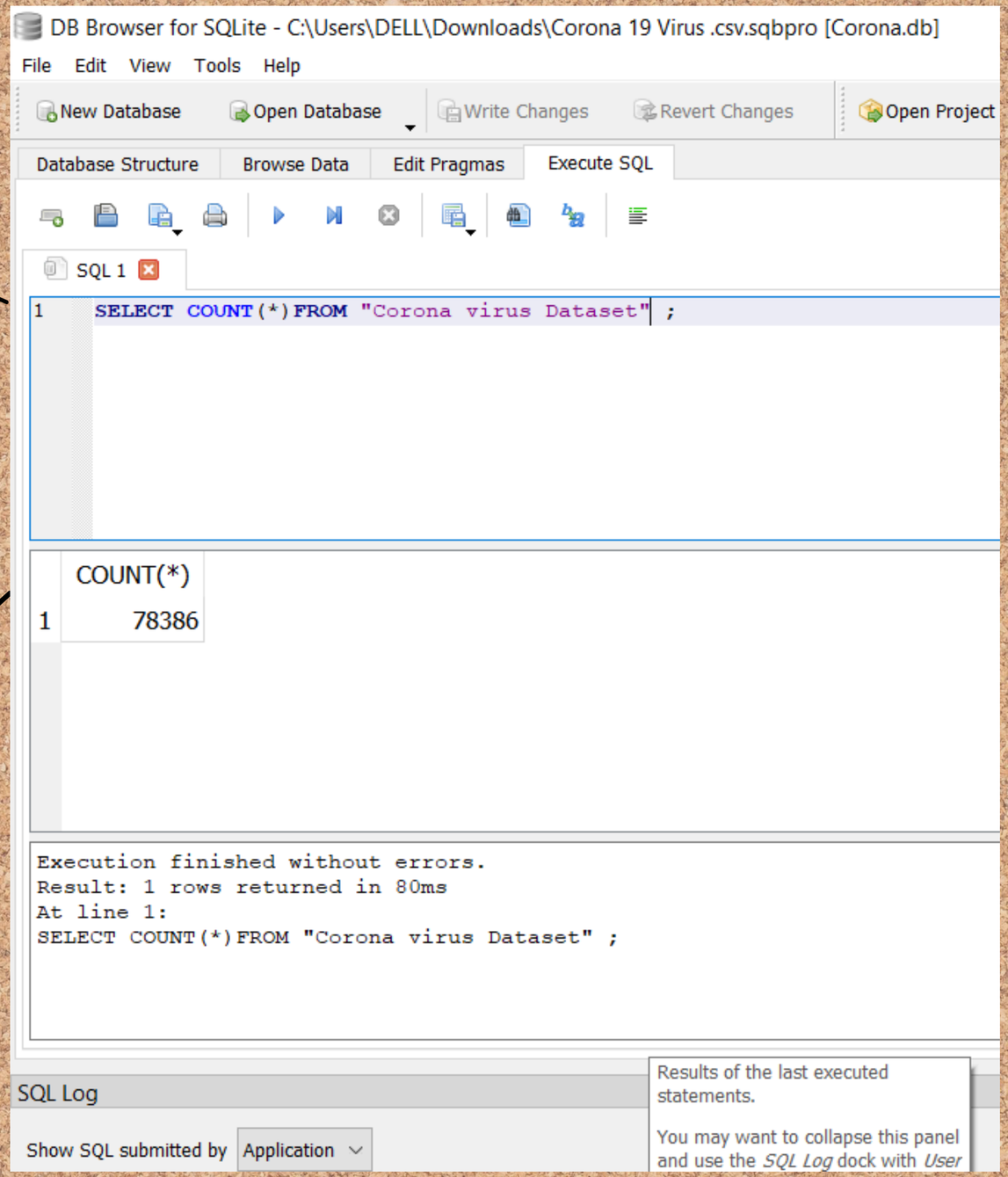

| | Province | Country/Region | Latitude | Longitude | Date | Confirmed | Deaths | Recovered |
|---|---|---|---|---|---|---|---|---|
| 64 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 25-03-2020 | 33 | 1 | 1 |
| 65 | Afghanistan | Afghanistan | 33.93911 | 67.709953 | 26-03-2020 | 4 | 1 | 0 |
| 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 |

check total number of rows

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

Display Total Number Rows Presented in The Table

Total Number Rows 78386;



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";
```

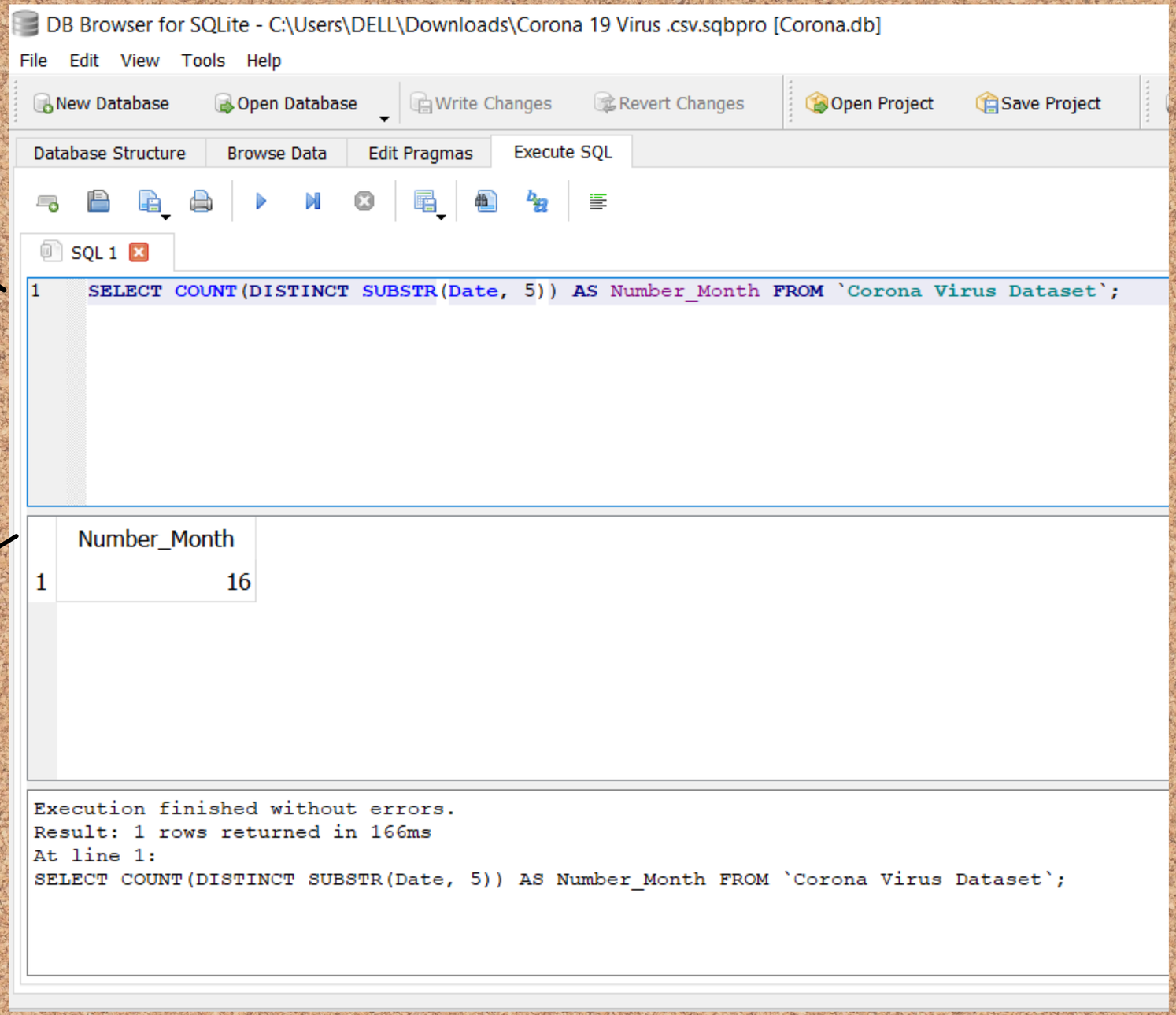
| | start_date | end_date |
|---|---|---|
| 1 | 2020-01-22 | 2021-06-13 |

```
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



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,
```

SQL Log

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.sqbpro [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 | Clear

```
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
```

Edit Database Cell

Mode: Text

NULL

Type of data currently in cell: NUL

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 Pragmas | 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
```

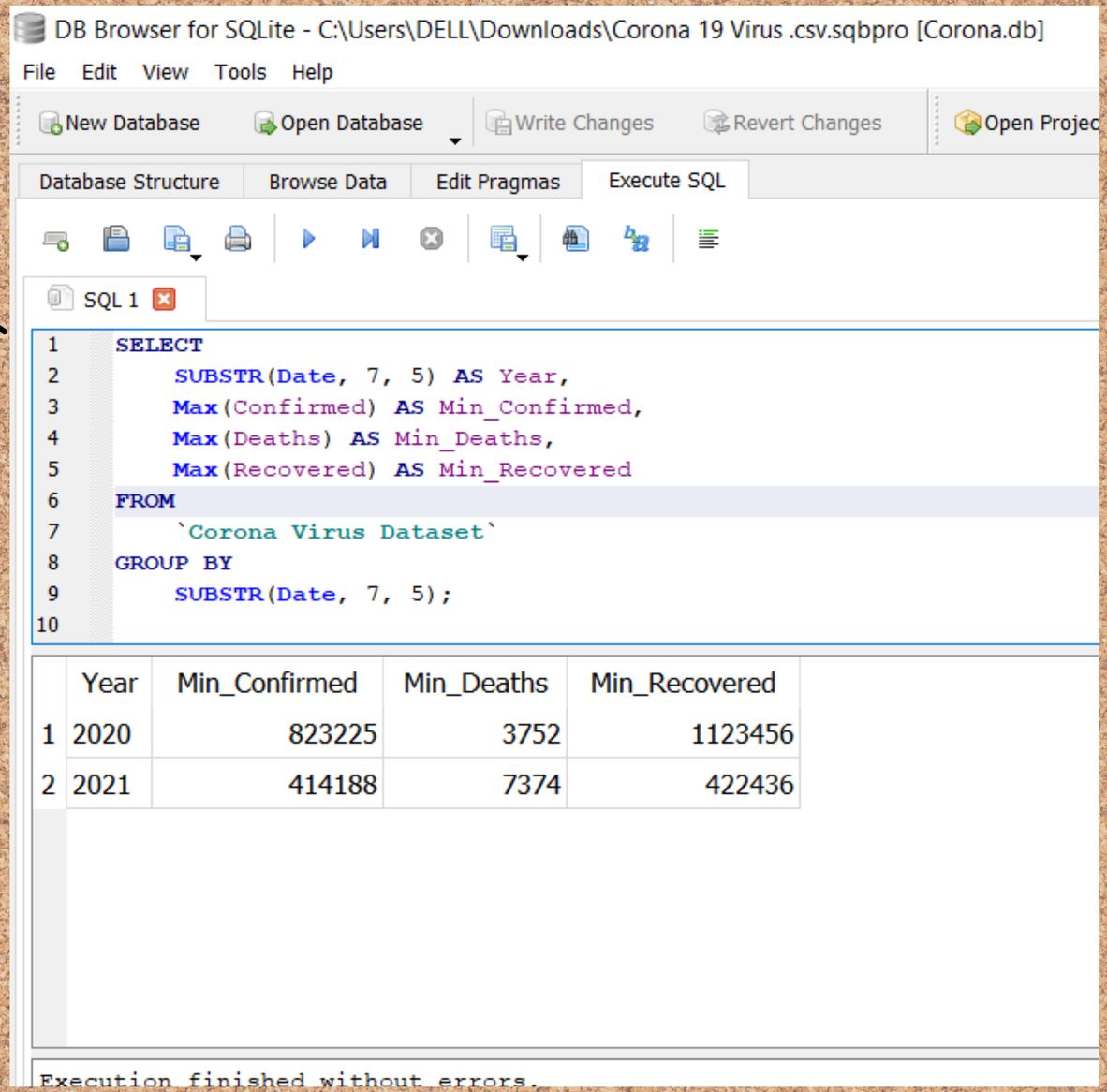
| | Year | Min_Confirmed | Min_Deaths | Min_Recovered |
|---|---|---|---|---|
| 1 | 2020 | 0 | 0 | 0 |
| 2 | 2021 | 0 | 0 | 0 |

```
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);



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 Pragmas | 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
```

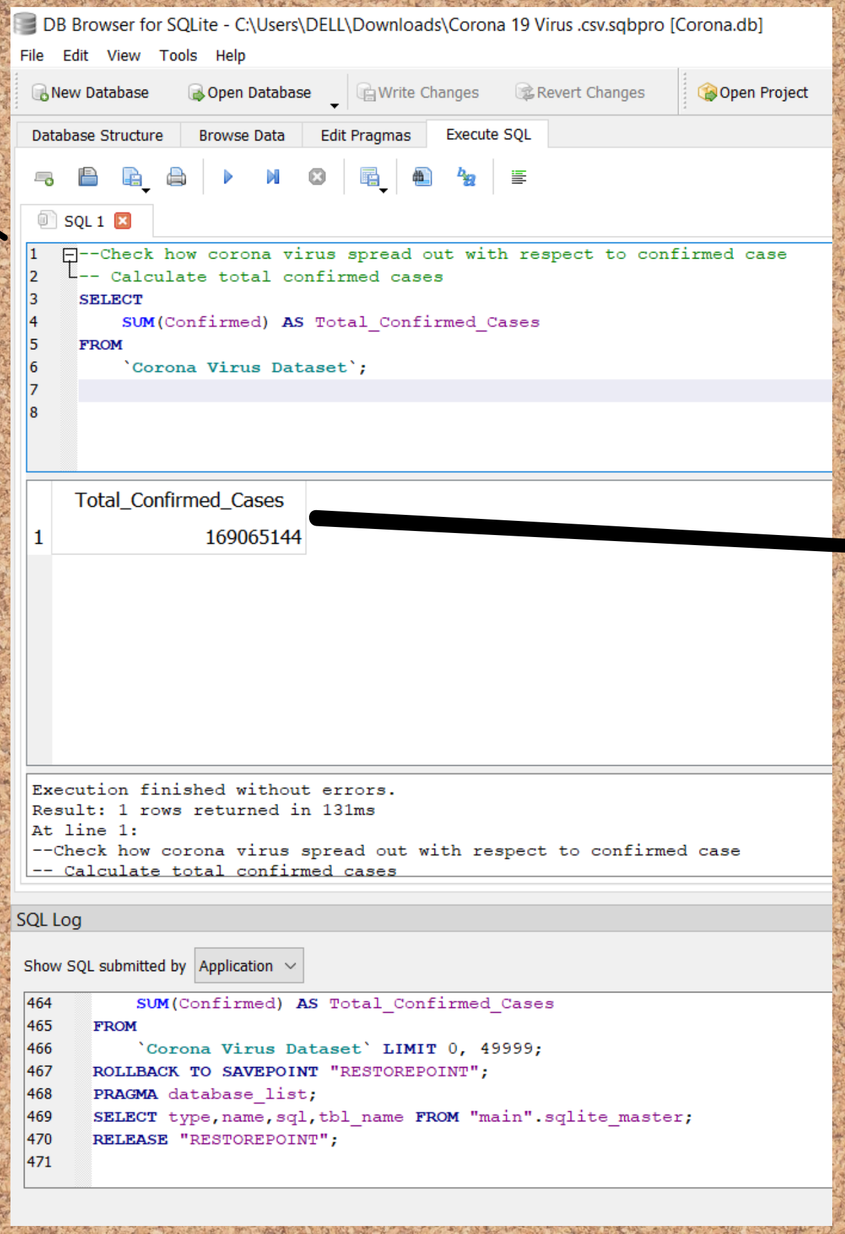
| | Month | Total_Confirmed | Total_Deaths | Total_Recovered |
|---|---|---|---|---|
| 1 | 01-2020 | 6384 | 190 | 143 |
| 2 | 01-2021 | 18672205 | 401893 | 9164347 |
| 3 | 02-2020 | 68312 | 2651 | 31405 |
| 4 | 02-2021 | 10492664 | 298239 | 6719785 |
| 5 | 03-2020 | 769236 | 41346 | 133070 |
| 6 | 03-2021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

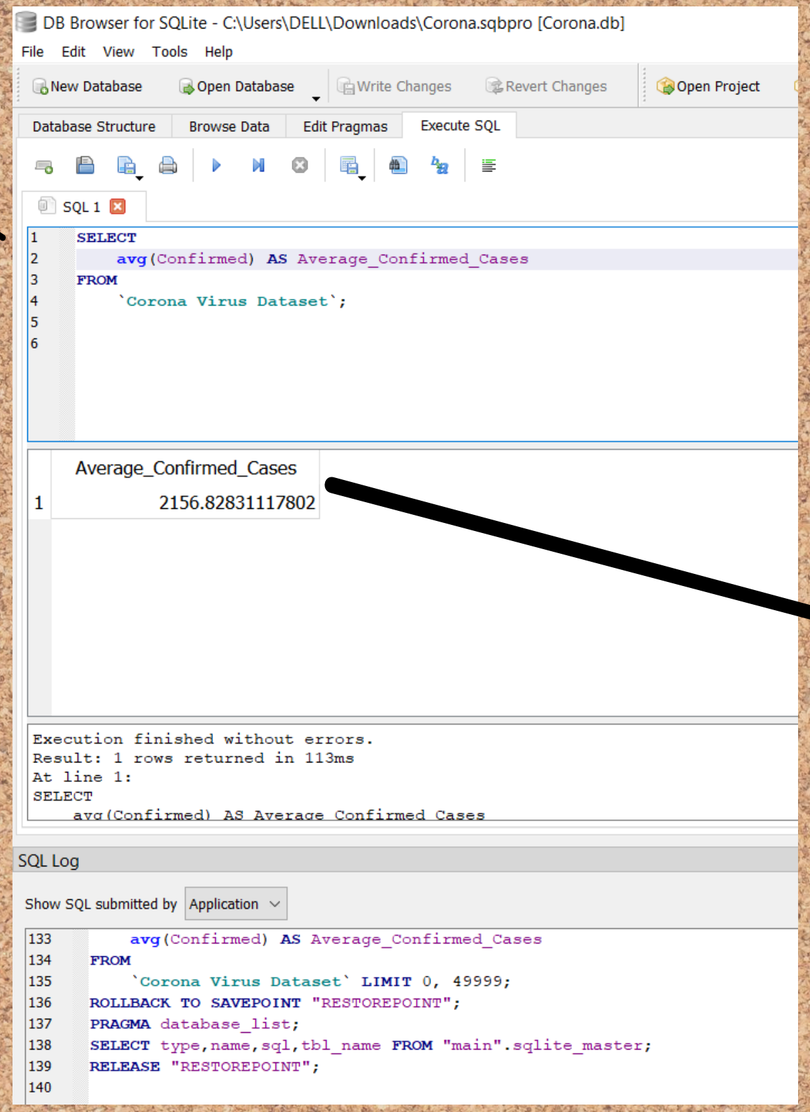
```
SELECT
SUM(Confirmed) AS
Total_Confirmed_Cases
FROM
`Corona Virus Dataset`;
```




Showing Total Comfirmed Cases in The DataSet

Check how corona virus spread out with respect to confirmed case

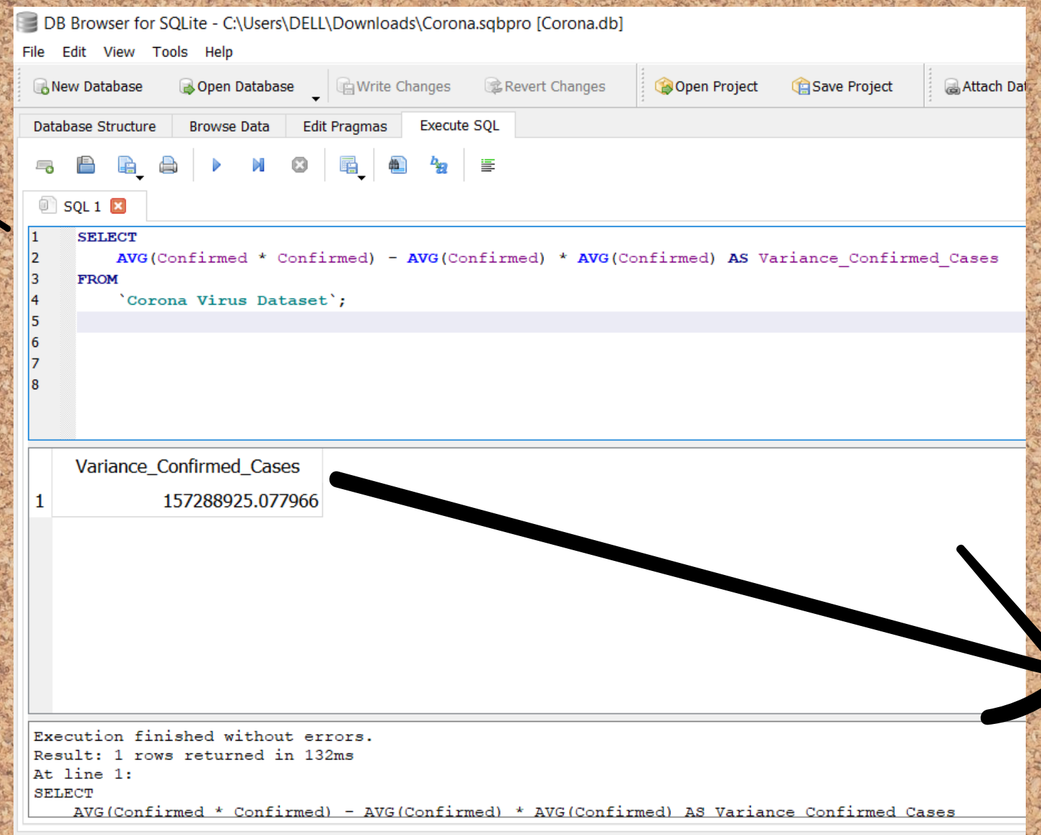
SELECT
avg(Confirmed) AS
Total_Confirmed_Cases
FROM
`Corona Virus Dataset`;




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`;
```




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

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

| | 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 Pragmas | 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

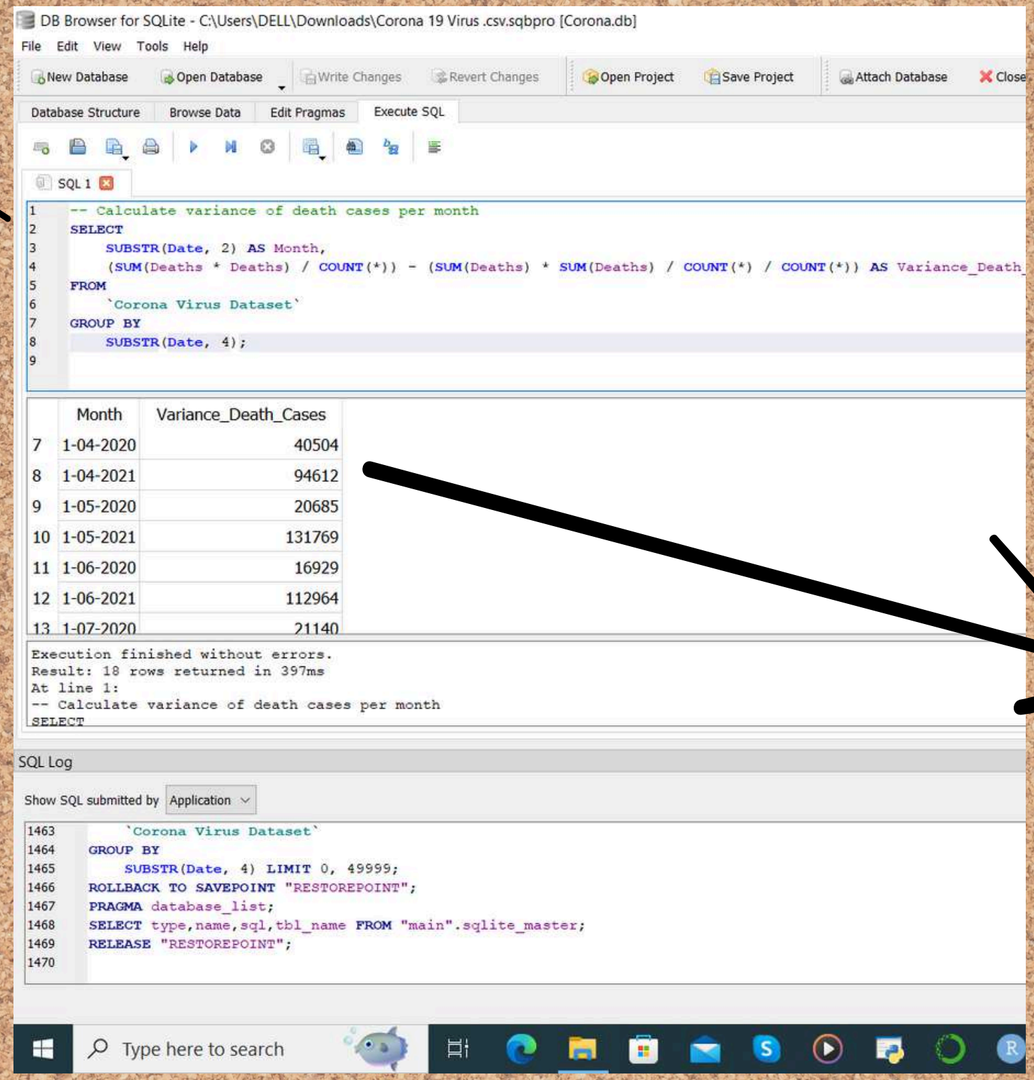
Show SQL submitted by Application

```
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);
```

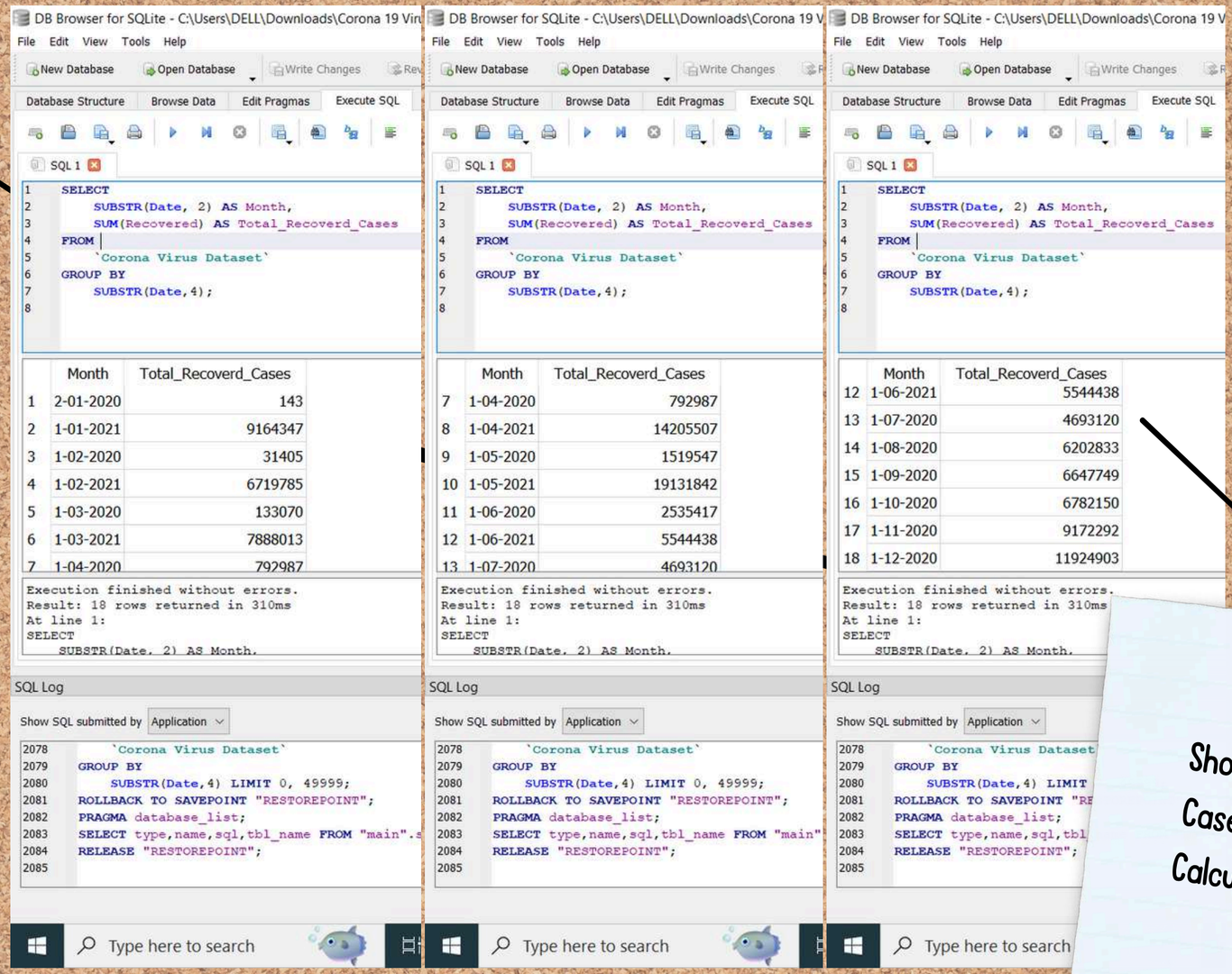



Showing Total Death Cases in The DataSet By Calculating Variance

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);
```

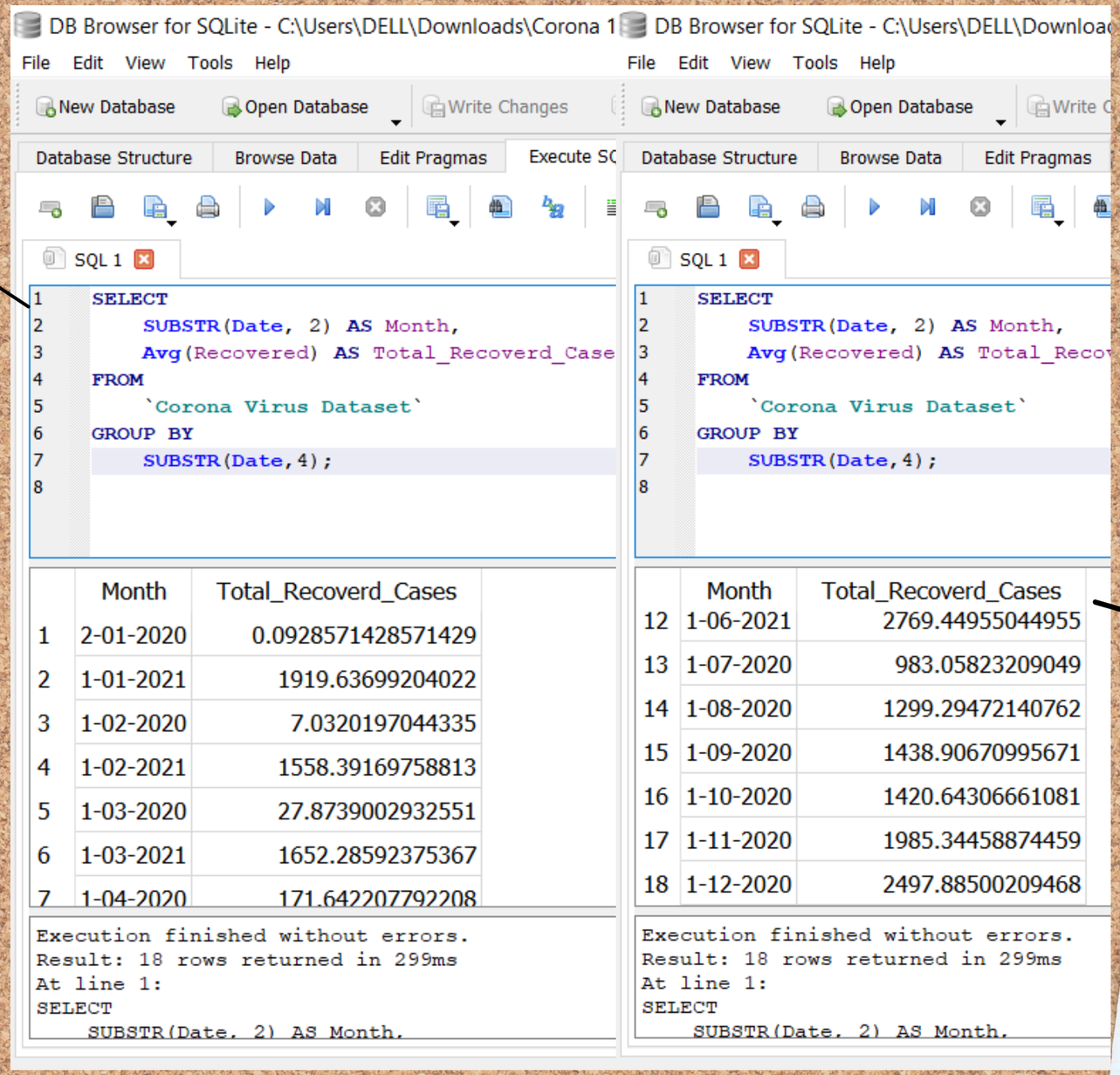



Showing Total Recovery Cases in The DataSet By Calculating Sum Function

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

Syntax:-

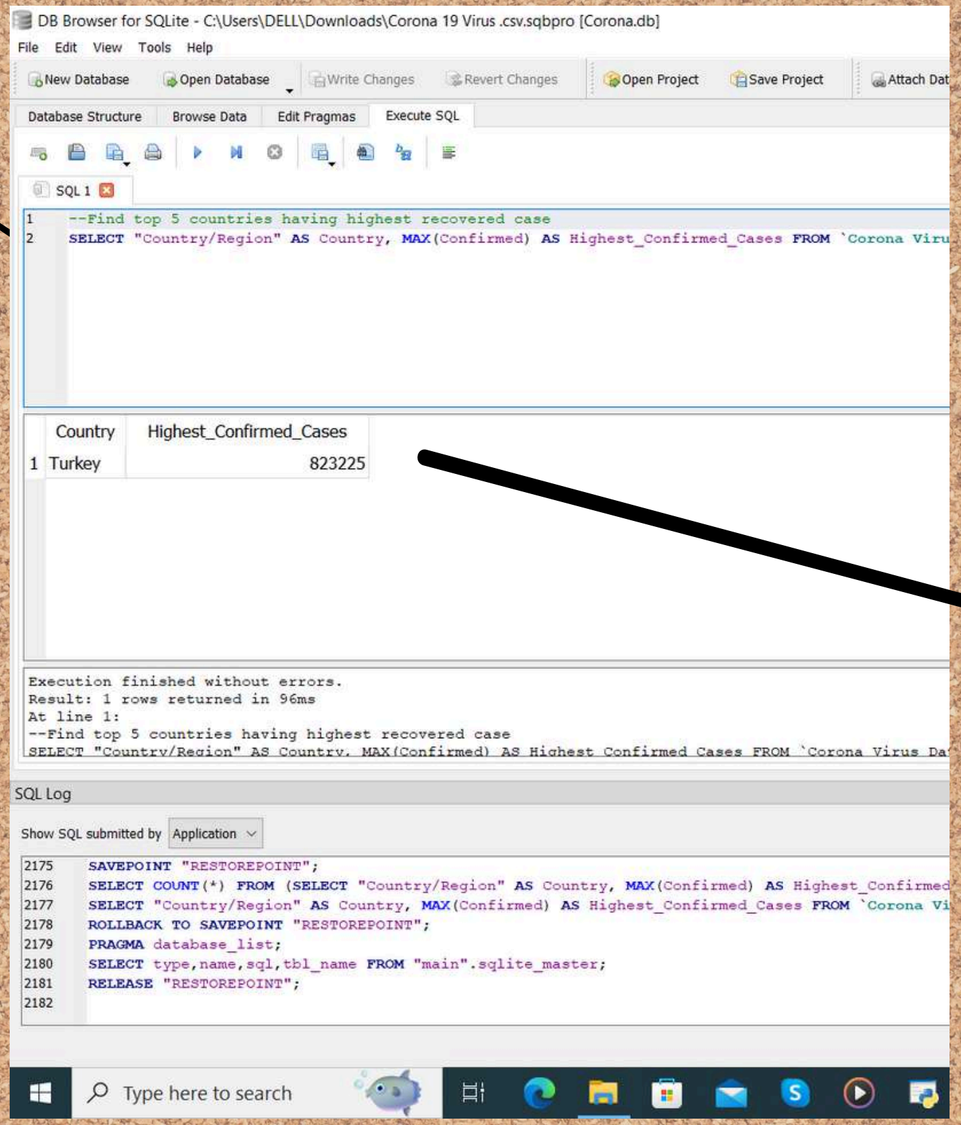
```
SELECT
SUBSTR(Date, 2) AS Month,
SUM(Recovered) AS
Total_Recoverd_Cases
FROM
`Corona Virus Dataset`
GROUP BY
SUBSTR(Date,4);
```




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

Find Country having highest number of the Confirmed case

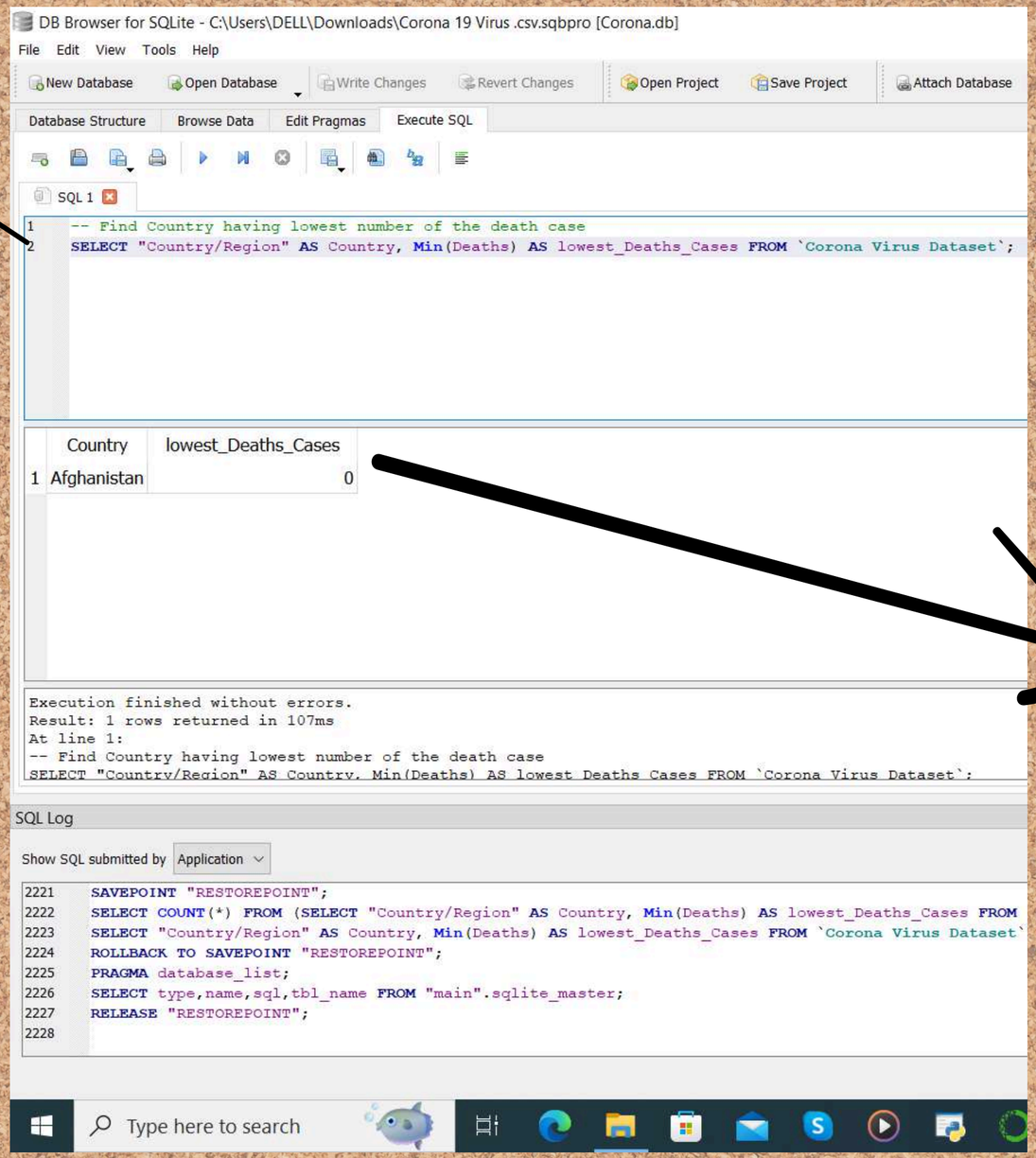
SELECT "Country/Region" AS Country, MAX(Confirmed) AS Highest_Confirmed_Cases FROM `Corona Virus Dataset`;




Showing Highest Confirmed Cases in The DataSet By Max Function

Find Country having lowest number of the death case

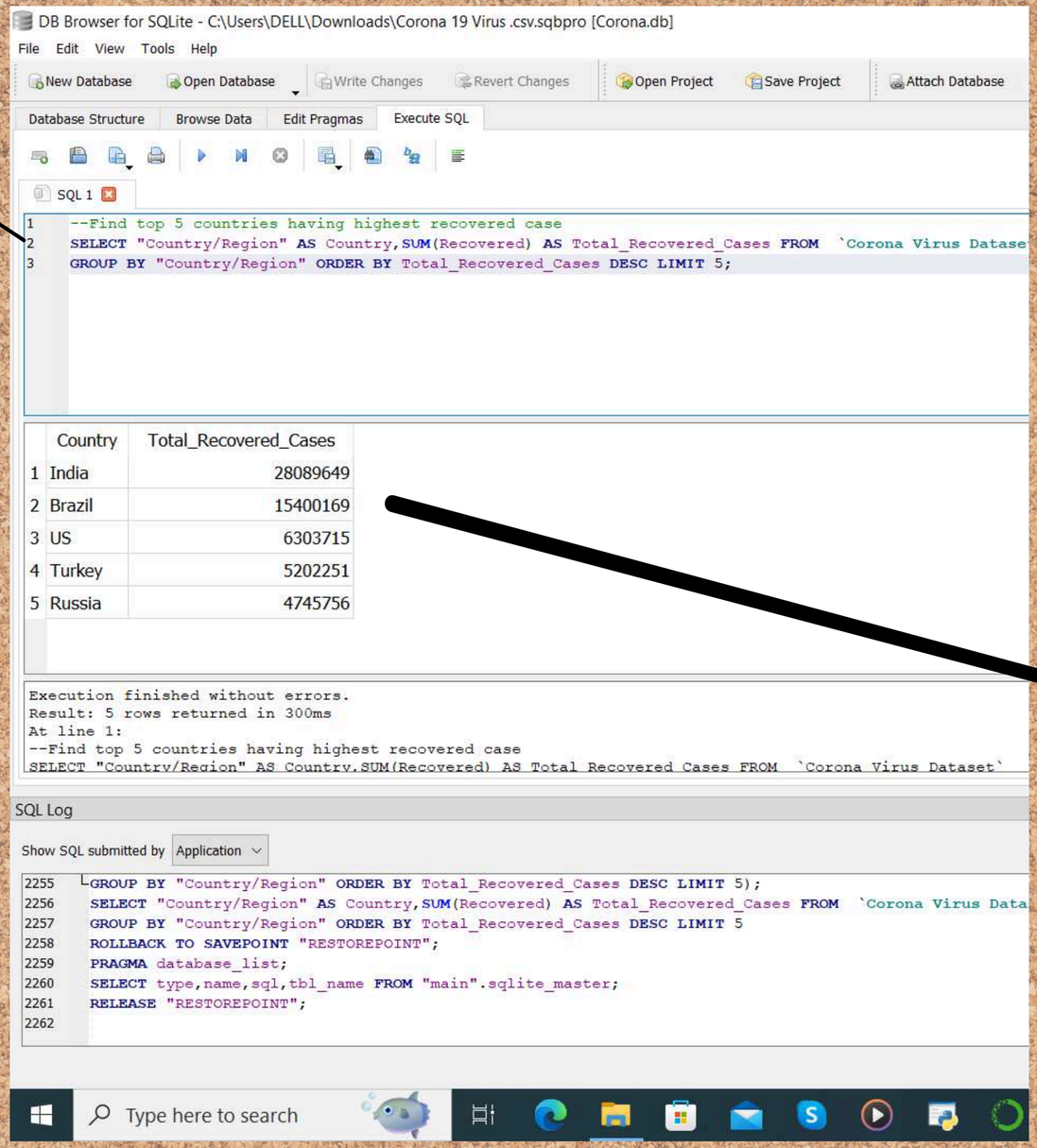
SELECT "Country/Region" AS Country, Min(Deaths) AS lowest_Deaths_Cases FROM `Corona Virus Dataset`;




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;




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.

Power By Mentorness

# References:-

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

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

Power By Mentorness

# Connect with us.

Github

**https//github.com/saravanaayyapa28**

Social Media

**www.linkedin.com/in/saravana28**