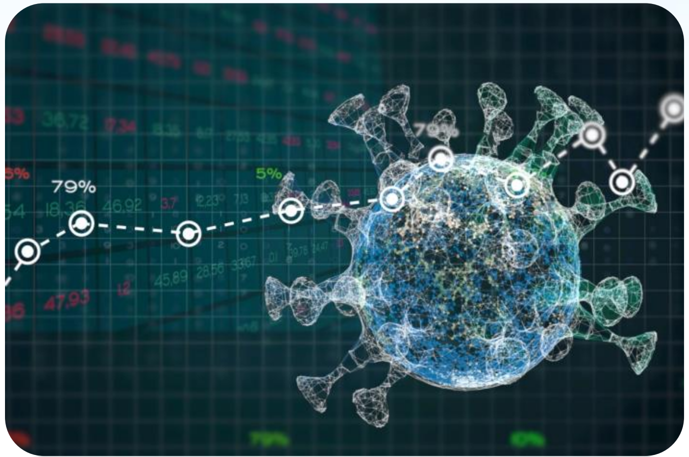
MENTOR NESS

Task 2: Corona Virus Analysis with SQL

- **Name:** Meet Mukesh Vaghasiya
- **Profile:** Data Analyst Intern
- **Batch:** MIP-DA-03

Project Overview



- COVID-19's impact on public health underscores the need for data-driven insights to understand its spread.
- Tasked as a data analyst, the objective is to analyze a COVID-19 dataset for valuable insights.
- Through rigorous analysis, we aim to uncover patterns and trends to understand virus transmission better.
- Data-driven insights will aid in combatting the pandemic and protecting public health.

Dataset Description

Description of each column in the dataset (Corona Virus Dataset)

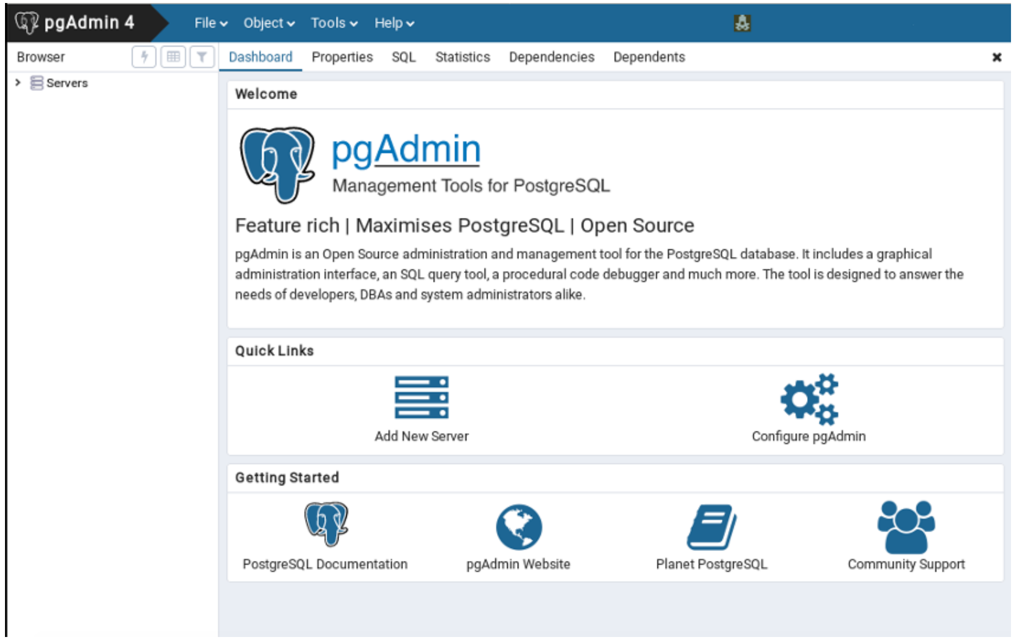
- **Province:** Geographic subdivision within a country/region.
- **Country/Region:** Geographic entity where data is recorded.
- **Latitude:** North-south position on Earth's surface.
- **Longitude:** East-west position on Earth's surface.
- **Date:** Recorded date of CORONA VIRUS data.
- **Confirmed:** Number of diagnosed CORONA VIRUS cases.
- **Deaths:** Number of CORONA VIRUS-related deaths.
- **Recovered:** Number of recovered CORONA VIRUS cases

DBMS and Tool Used

- **Database Management System Used**

PostgreSQL

- **Management Tool: pgAdmin 4**

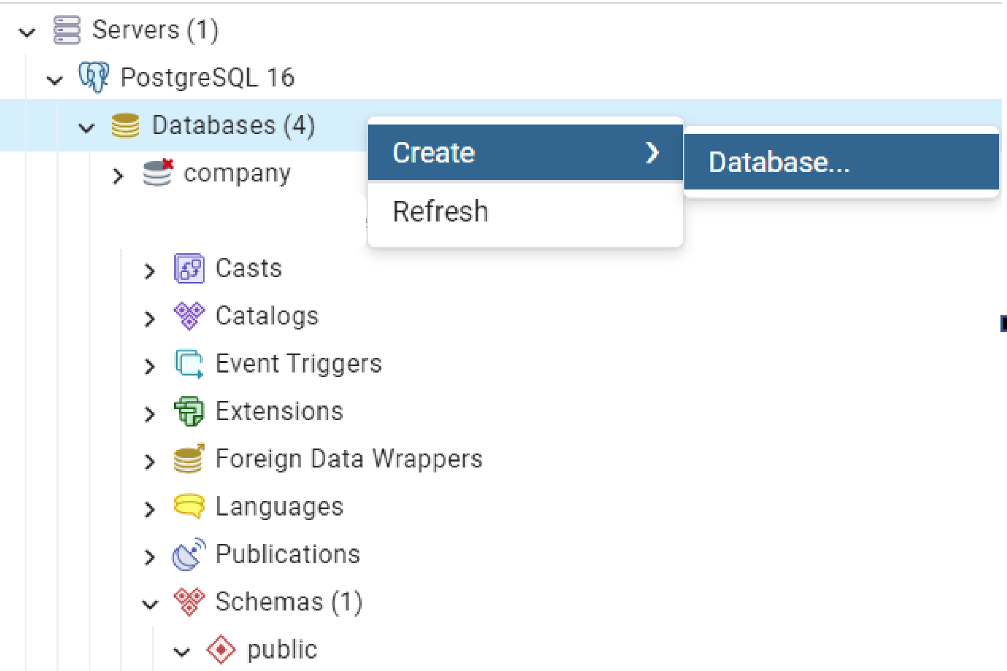


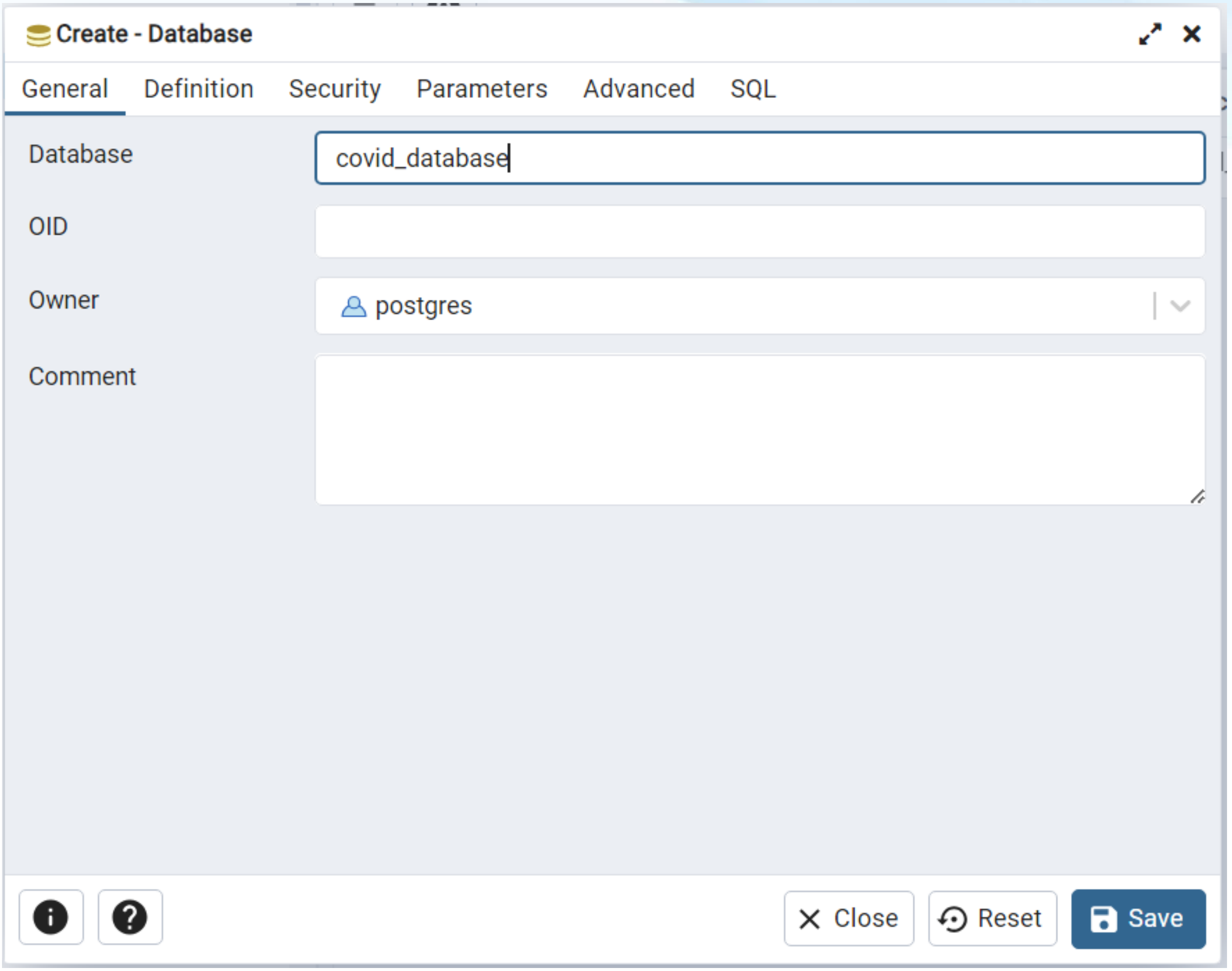
Feature rich | Maximises PostgreSQL | Open Source

pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

Creating Database

- "covid_database"





Database: covid_database

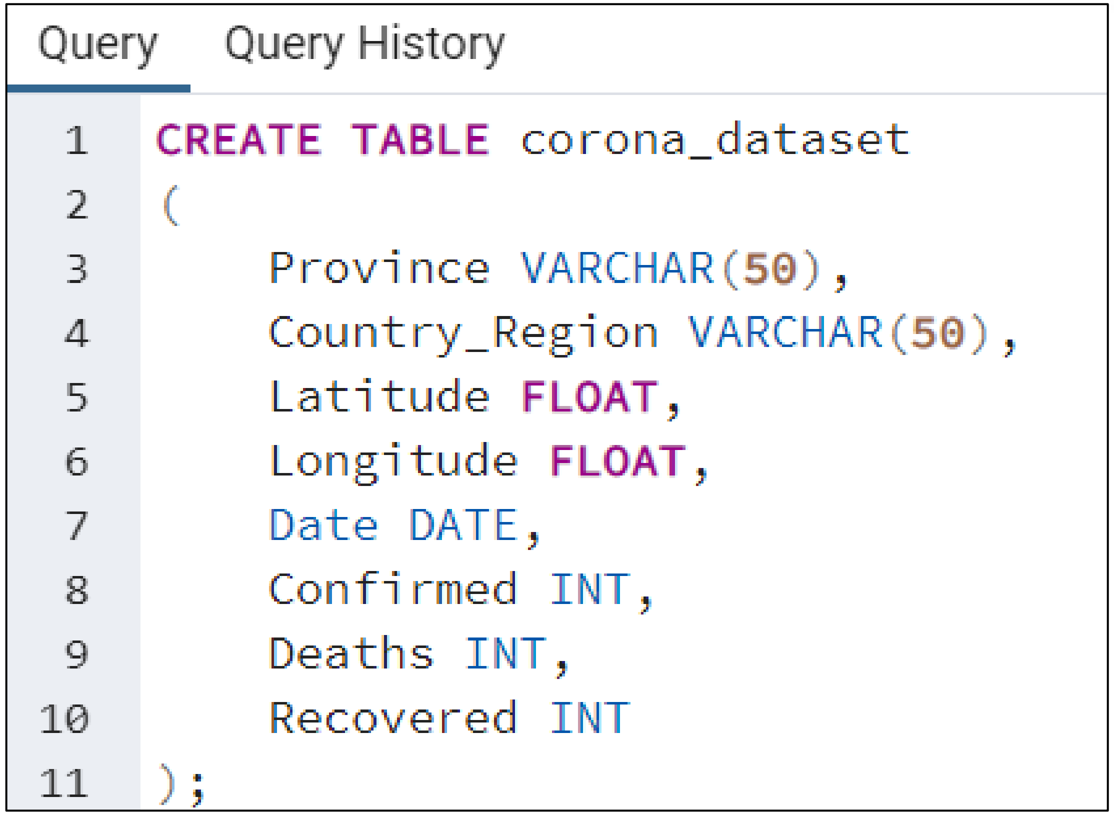
OID:

Owner: postgres

Comment:

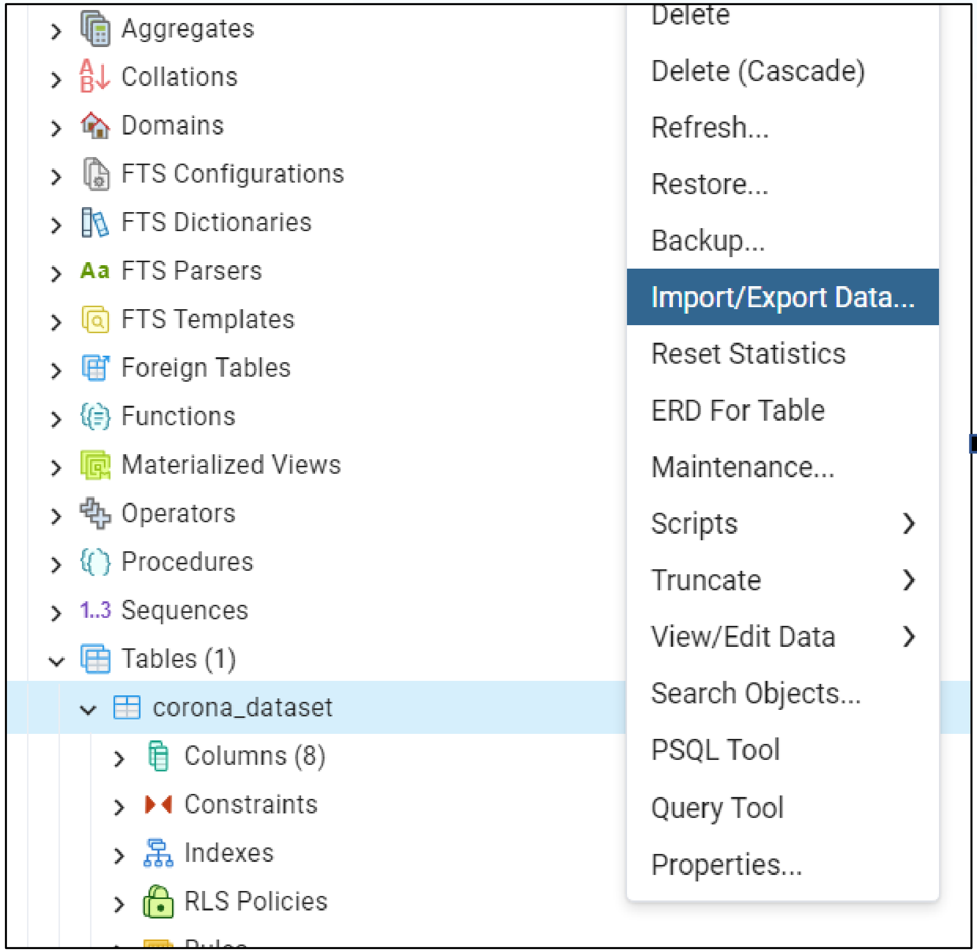
Creating Table

- **Query to Create the table**

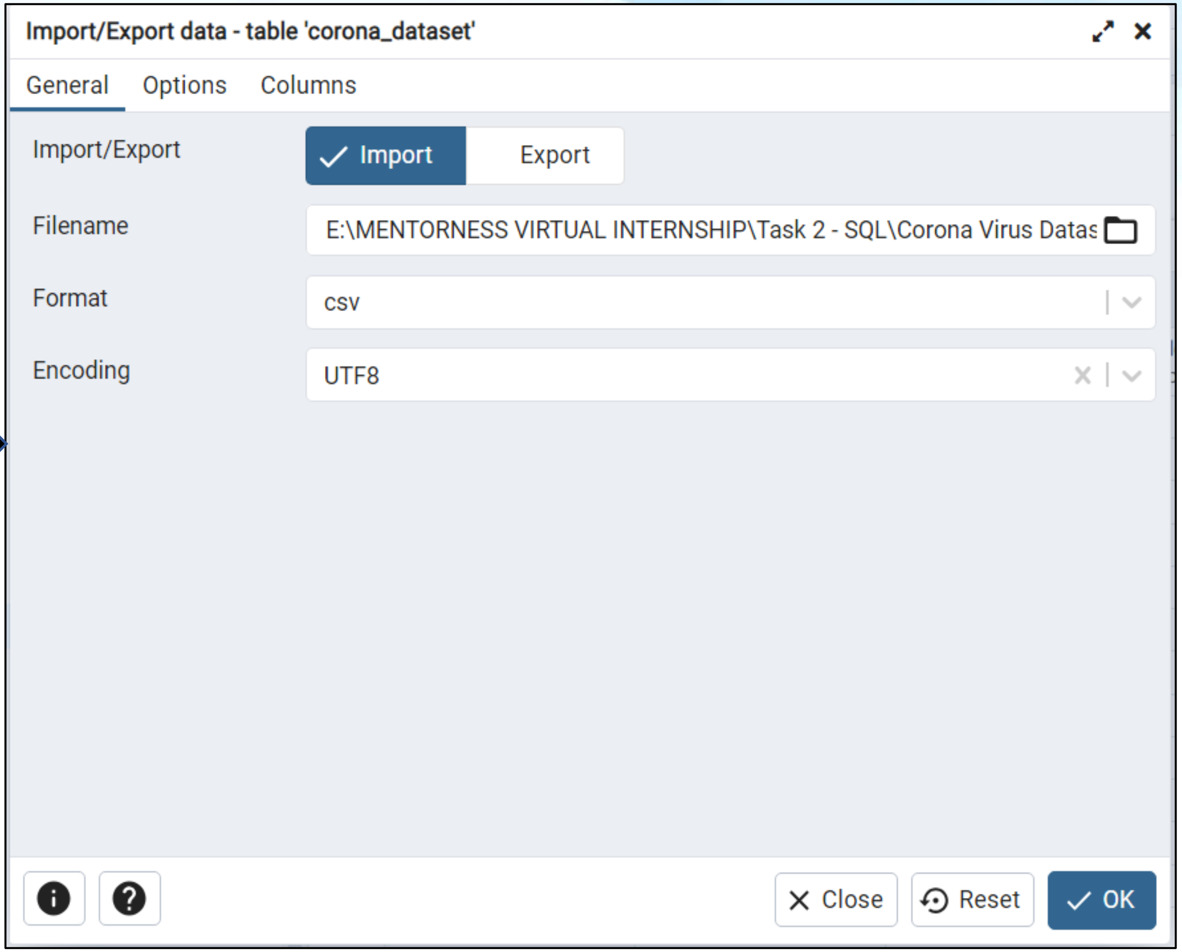


```sql
CREATE TABLE corona_dataset
(
    Province VARCHAR(50),
    Country_Region VARCHAR(50),
    Latitude FLOAT,
    Longitude FLOAT,
    Date DATE,
    Confirmed INT,
    Deaths INT,
    Recovered INT
);
```

How to Import Data into Table?



Left Click on corona_dataset & Select "Import/Export Data"



Import/Export: ✓ Import | Export

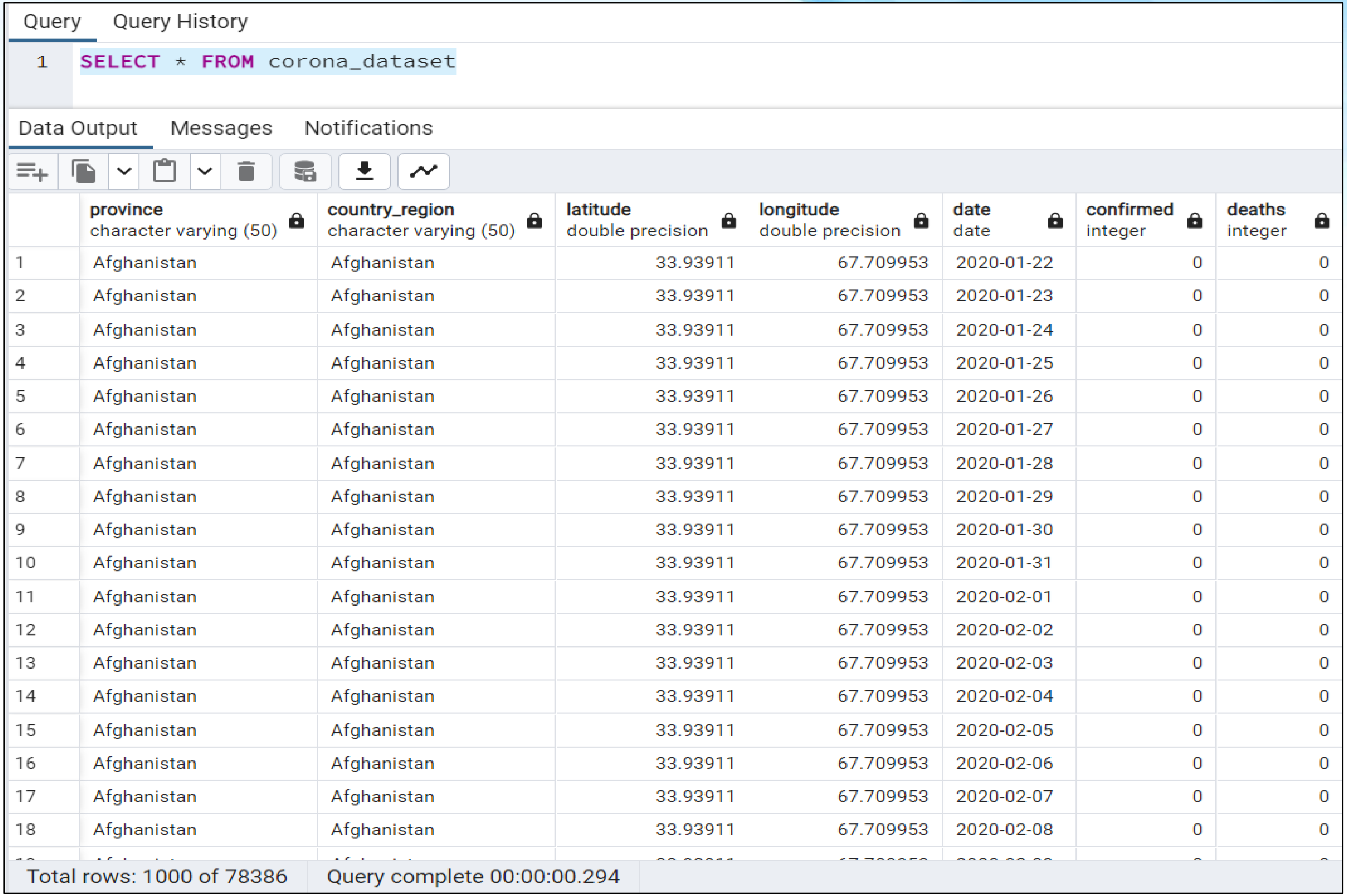
Filename: E:\MENTORNESS VIRTUAL INTERNSHIP\Task 2 - SQL\Corona Virus Datas

Format: csv

Encoding: UTF8

Select the Path which leads to "Corona Dataset.csv" file

Imported Data into Table



```
SELECT * FROM corona_dataset
```

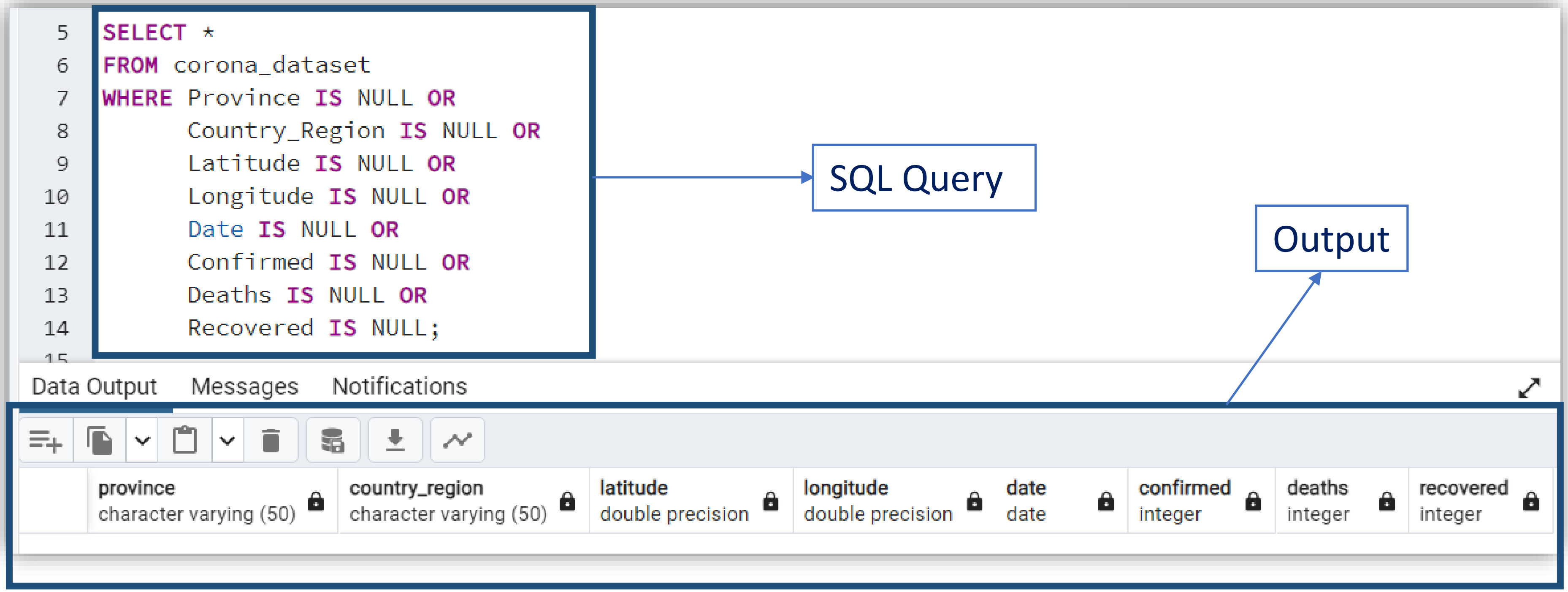
	province character varying (50)	country_region character varying (50)	latitude double precision	longitude double precision	date date	confirmed integer	deaths integer
1	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-22	0	0
2	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-23	0	0
3	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-24	0	0
4	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-25	0	0
5	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-26	0	0
6	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-27	0	0
7	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-28	0	0
8	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-29	0	0
9	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-30	0	0
10	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-31	0	0
11	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-01	0	0
12	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-02	0	0
13	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-03	0	0
14	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-04	0	0
15	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-05	0	0
16	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-06	0	0
17	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-07	0	0
18	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-08	0	0

Total rows: 1000 of 78386 Query complete 00:00:00.294

Data Cleaning

To avoid any errors, we check for missing value / null value

- **1. Write a code to check NULL values**

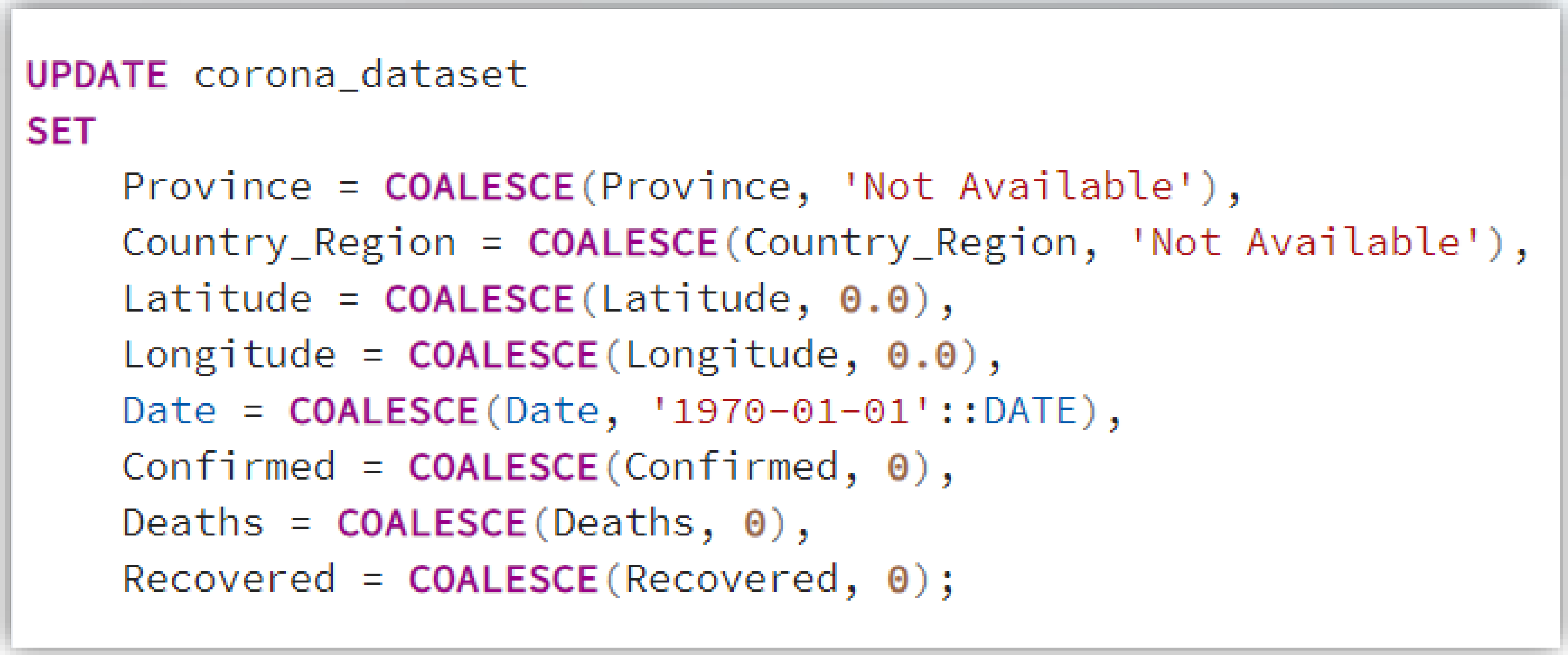


```
SELECT *
FROM corona_dataset
WHERE Province IS NULL OR
      Country_Region IS NULL OR
      Latitude IS NULL OR
      Longitude IS NULL OR
      Date IS NULL OR
      Confirmed IS NULL OR
      Deaths IS NULL OR
      Recovered IS NULL;
```

	province character varying (50)	country_region character varying (50)	latitude double precision	longitude double precision	date date	confirmed integer	deaths integer	recovered integer

- **Inference:** Based on the analysis conducted, it is evident that there are **no null values** present in any of the columns within the dataset.

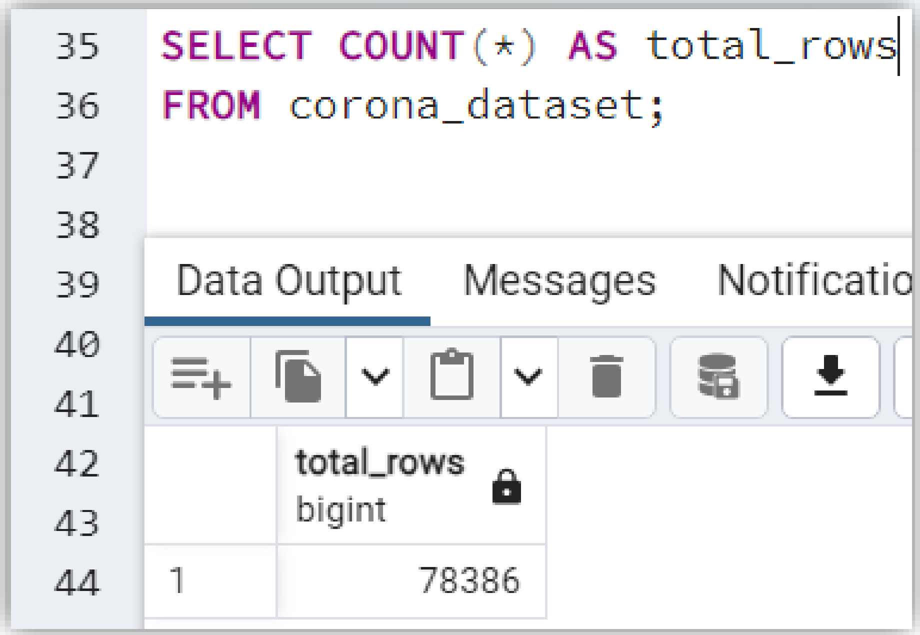
- 2. **<u>If NULL values are present, update them with zeros for all columns</u>**



```sql
UPDATE corona_dataset
SET
    Province = COALESCE(Province, 'Not Available'),
    Country_Region = COALESCE(Country_Region, 'Not Available'),
    Latitude = COALESCE(Latitude, 0.0),
    Longitude = COALESCE(Longitude, 0.0),
    Date = COALESCE(Date, '1970-01-01'::DATE),
    Confirmed = COALESCE(Confirmed, 0),
    Deaths = COALESCE(Deaths, 0),
    Recovered = COALESCE(Recovered, 0);
```

- We have observed that the dataset does not contain any null values. However, in the event of null values being present, we would have addressed them using the aforementioned query.

- 3. **Check the total number of rows**

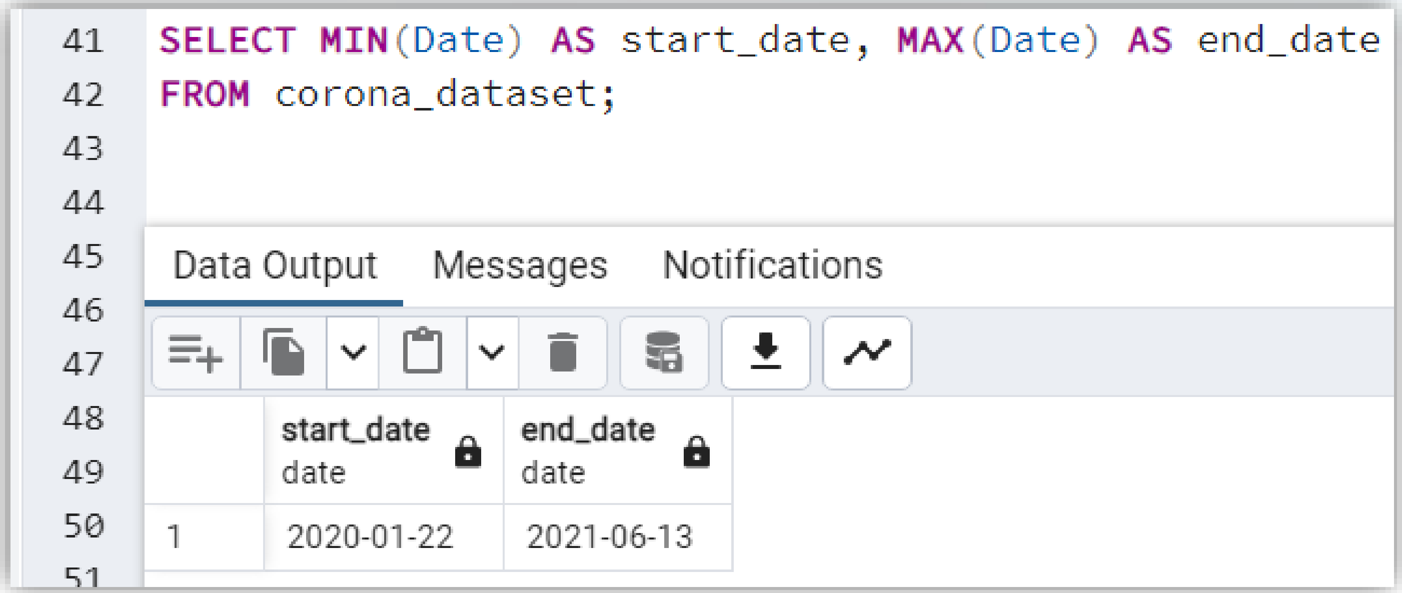


```
SELECT COUNT(*) AS total_rows
FROM corona_dataset;
```

	total_rows bigint
1	78386

- **Inference:** The total number of records stored in the table is **78386**

4. Check what is the start date and end date

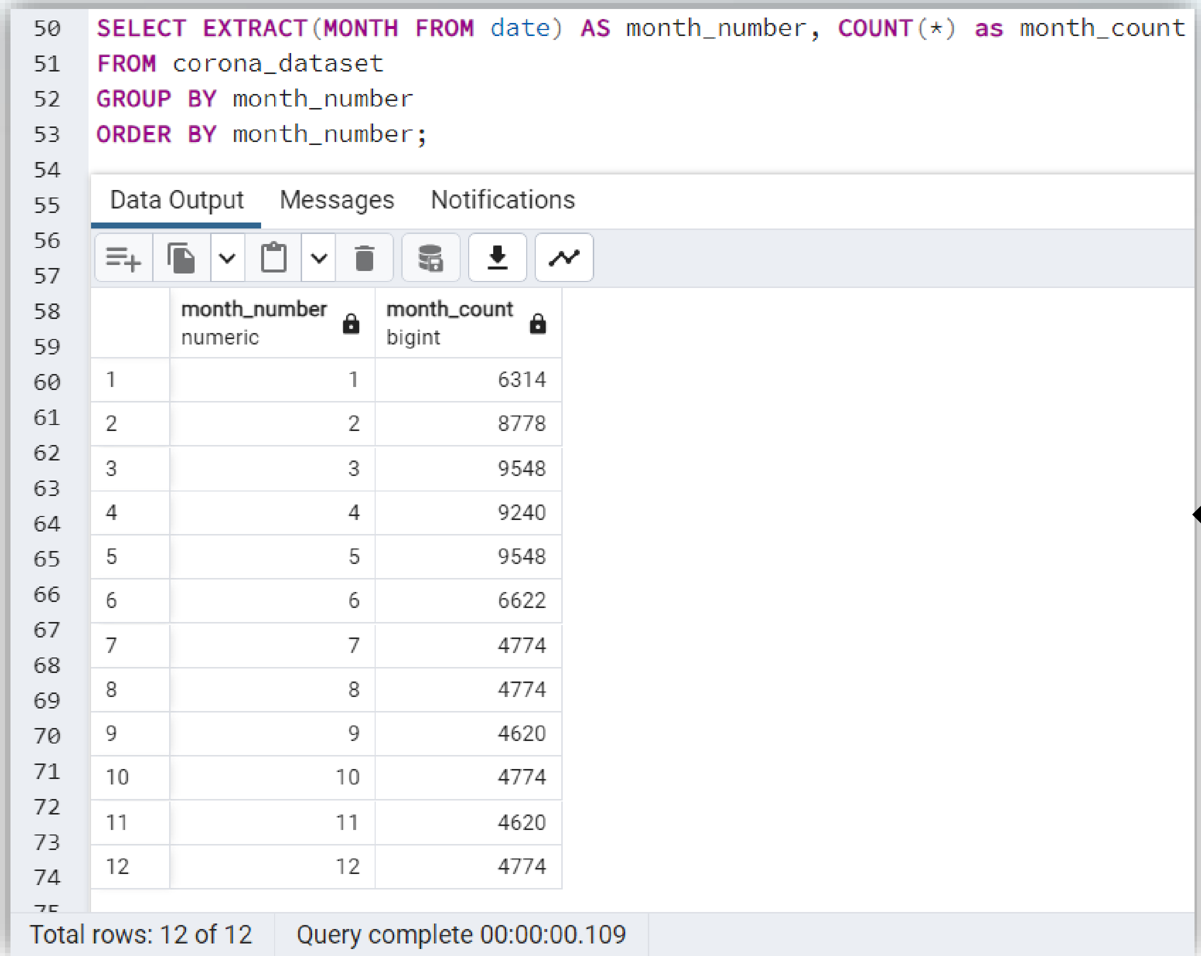


```sql
SELECT MIN(Date) AS start_date, MAX(Date) AS end_date
FROM corona_dataset;
```

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

- **Inference:** According to the dataset, the **start date** of the COVID-19 pandemic is recorded as January 22, 2020 **(22-01-2020)** with the **end date** noted as June 13, 2021 **(13-06-2021)**

5. Number of months present in the dataset



```sql
SELECT EXTRACT(MONTH FROM date) AS month_number, COUNT(*) as month_count
FROM corona_dataset
GROUP BY month_number
ORDER BY month_number;
```

	month_number numeric	month_count bigint
1	1	6314
2	2	8778
3	3	9548
4	4	9240
5	5	9548
6	6	6622
7	7	4774
8	8	4774
9	9	4620
10	10	4774
11	11	4620
12	12	4774

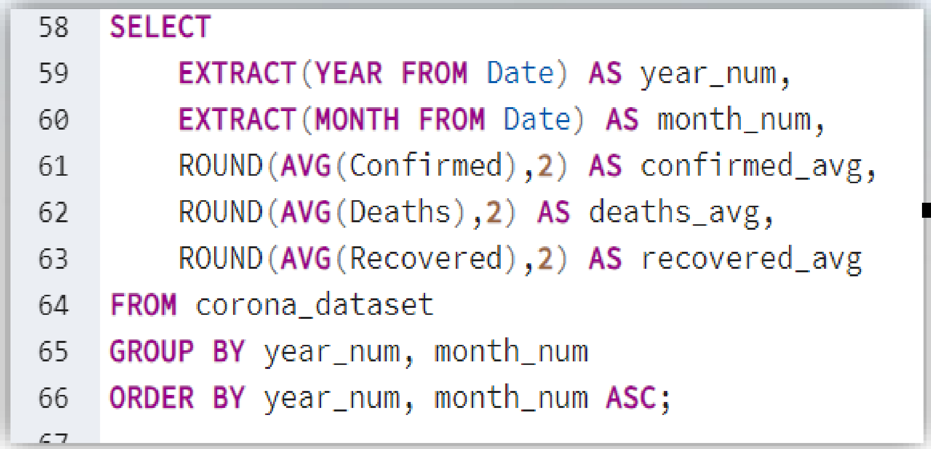
Total rows: 12 of 12   Query complete 00:00:00.109

- In the output, **"month_number"** represents distinct months, while **"month_count"** denotes the frequency of COVID cases associated with each respective month_number.

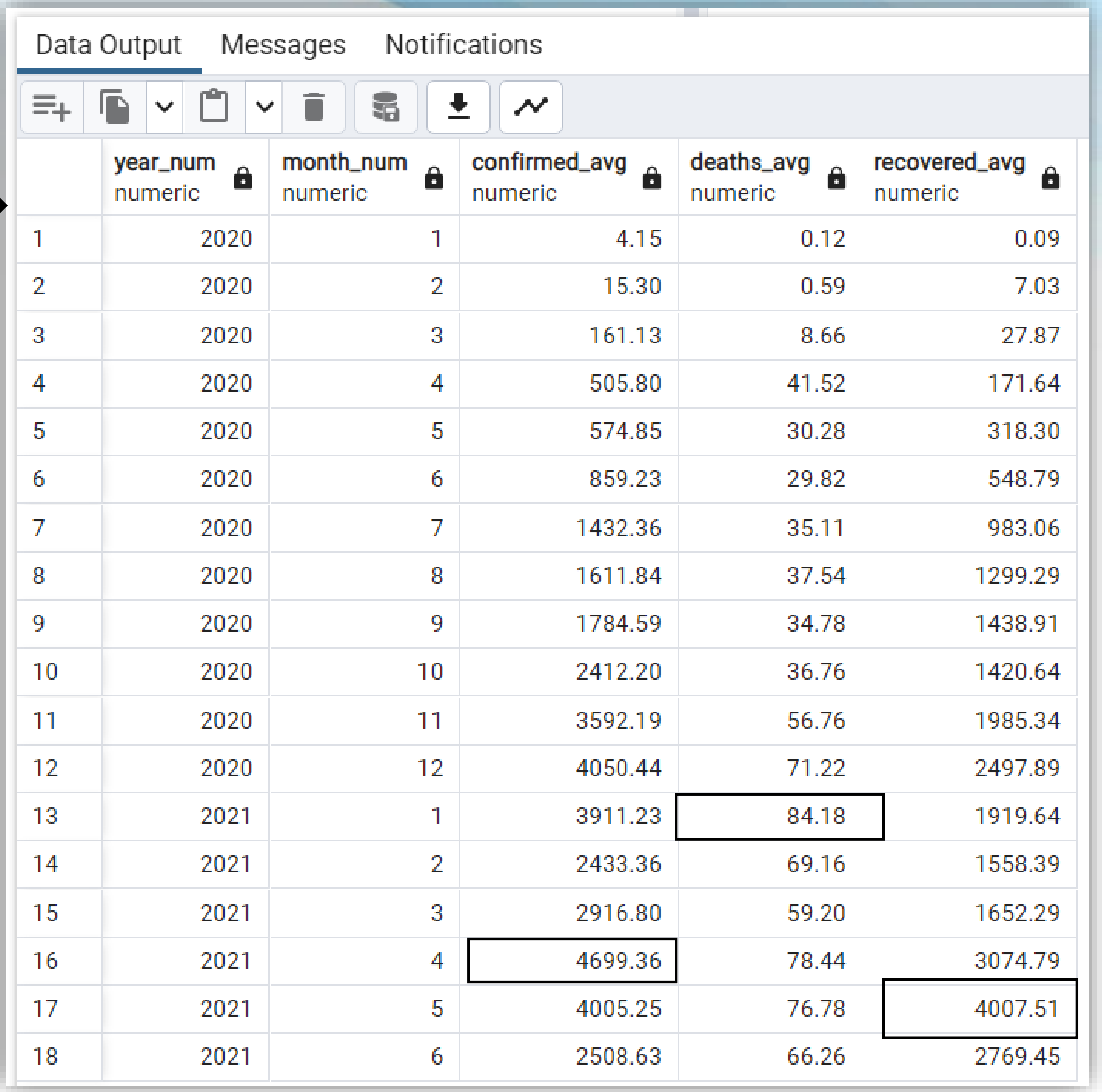
- For instance, if we consider **January (month_number = 1),** with a month_count of 6314, it indicates that there were 6314 occurrences of COVID-19 reported across various countries/regions during the month of January in both 2020 and 2021, as per the dataset.

- **Inference:** The dataset covers a total of **12 unique months**

▪ 6. Find the monthly average for confirmed, deaths, recovered



```sql
SELECT
    EXTRACT(YEAR FROM Date) AS year_num,
    EXTRACT(MONTH FROM Date) AS month_num,
    ROUND(AVG(Confirmed),2) AS confirmed_avg,
    ROUND(AVG(Deaths),2) AS deaths_avg,
    ROUND(AVG(Recovered),2) AS recovered_avg
FROM corona_dataset
GROUP BY year_num, month_num
ORDER BY year_num, month_num ASC;
```

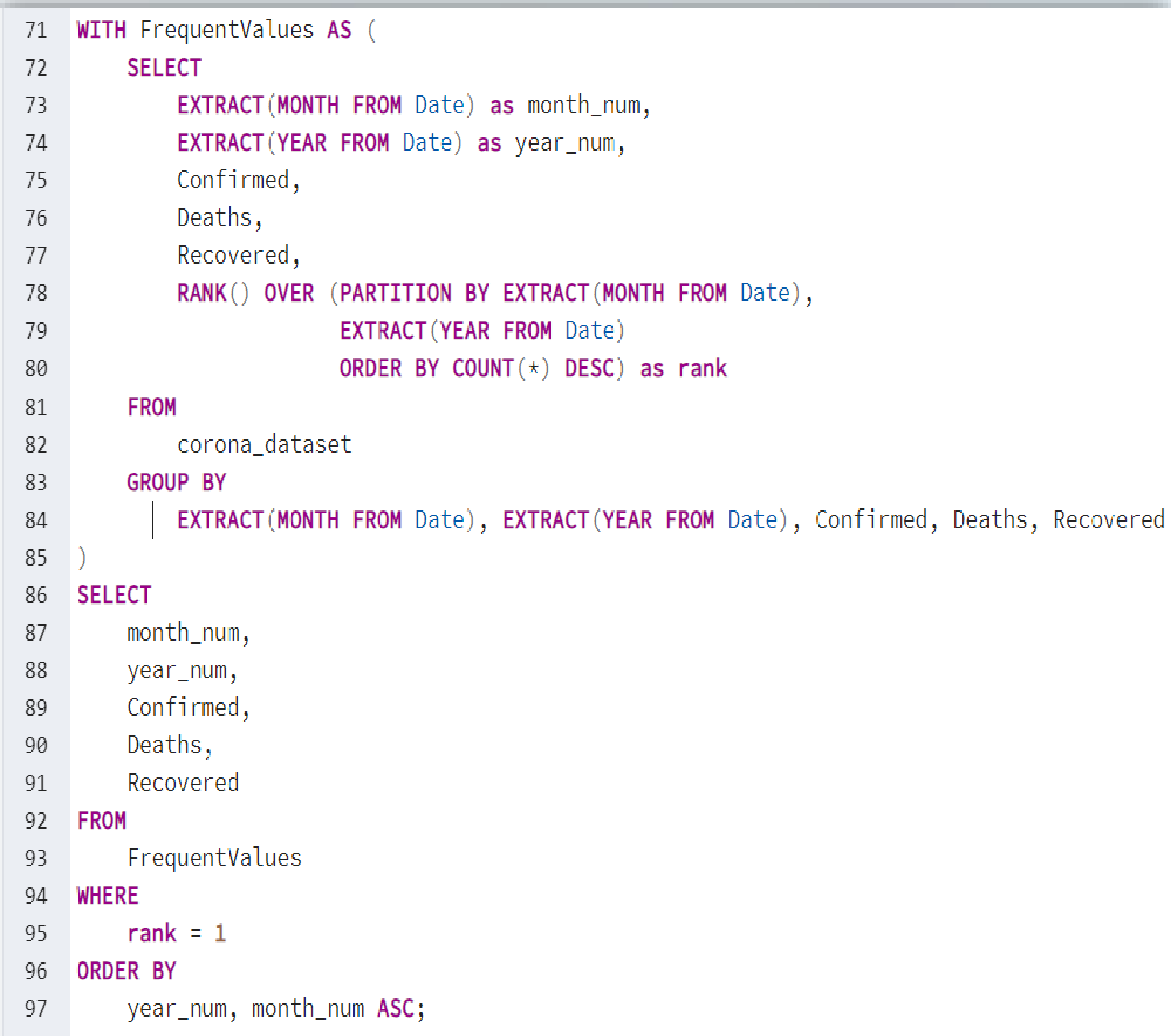


	year_num numeric	month_num numeric	confirmed_avg numeric	deaths_avg numeric	recovered_avg numeric
1	2020	1	4.15	0.12	0.09
2	2020	2	15.30	0.59	7.03
3	2020	3	161.13	8.66	27.87
4	2020	4	505.80	41.52	171.64
5	2020	5	574.85	30.28	318.30
6	2020	6	859.23	29.82	548.79
7	2020	7	1432.36	35.11	983.06
8	2020	8	1611.84	37.54	1299.29
9	2020	9	1784.59	34.78	1438.91
10	2020	10	2412.20	36.76	1420.64
11	2020	11	3592.19	56.76	1985.34
12	2020	12	4050.44	71.22	2497.89
13	2021	1	3911.23	84.18	1919.64
14	2021	2	2433.36	69.16	1558.39
15	2021	3	2916.80	59.20	1652.29
16	2021	4	4699.36	78.44	3074.79
17	2021	5	4005.25	76.78	4007.51
18	2021	6	2508.63	66.26	2769.45

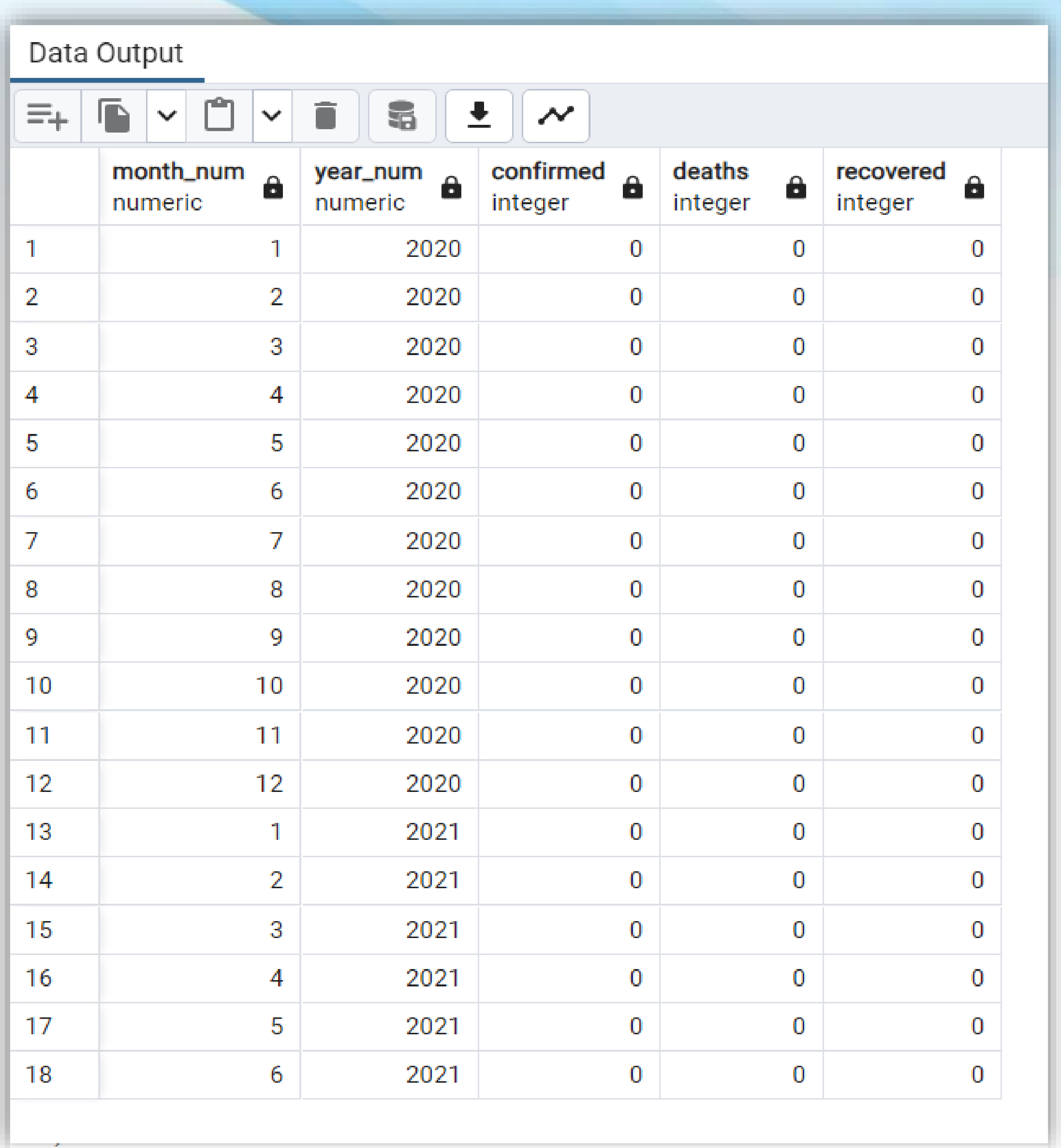
- ❑ Based on the output provided, it is apparent that the **highest average values** for confirmed cases, deaths, and recovered cases are as follows:

- ❑ Confirmed cases: 4699.36 in April 2021
- ❑ Deaths: 84.18 in January 2021
- ❑ Recovered cases: 4007.51 in May 2021

▪ 7. Find the most frequent value for confirmed, deaths, recovered each month

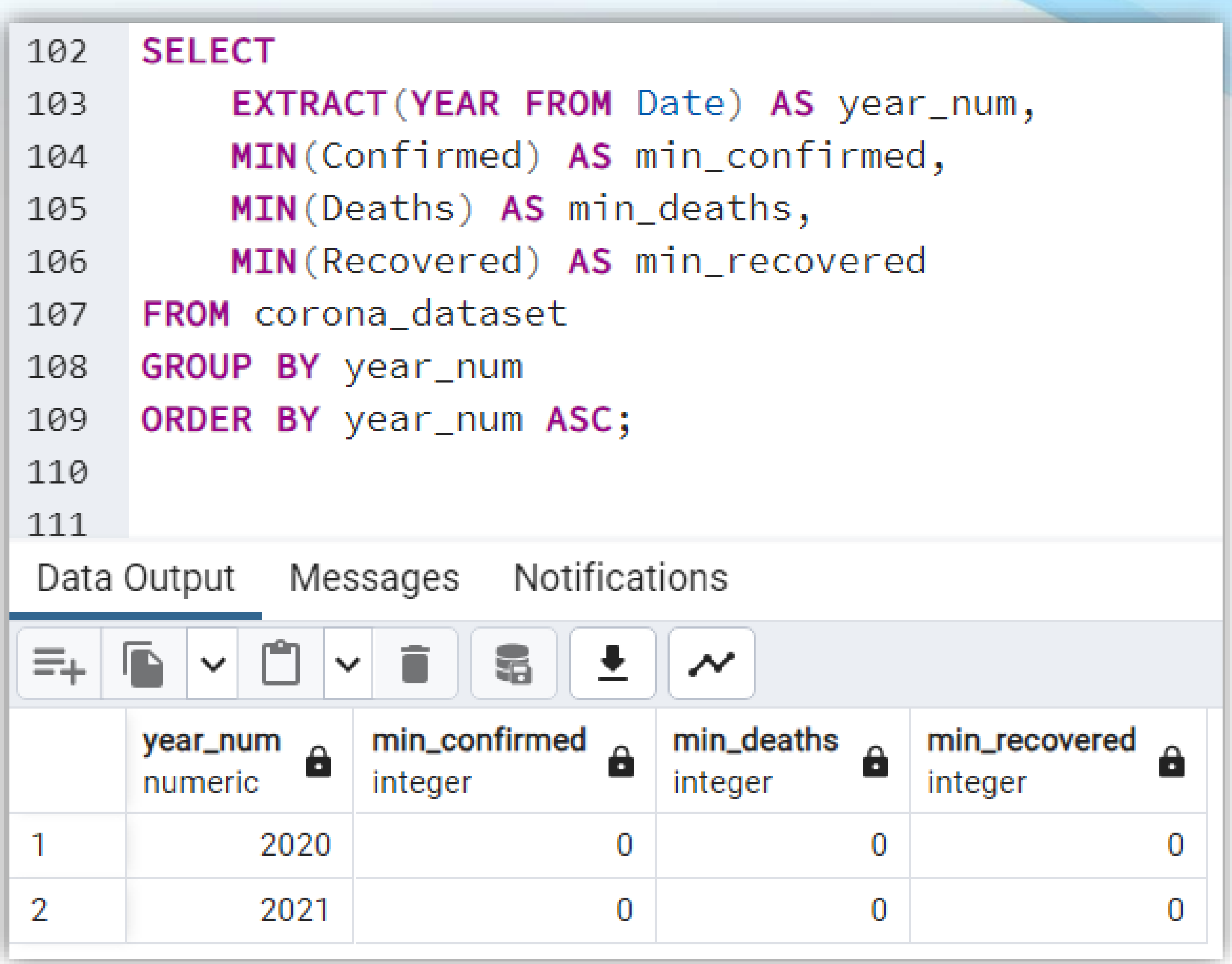


```sql
WITH FrequentValues AS (
    SELECT
        EXTRACT(MONTH FROM Date) as month_num,
        EXTRACT(YEAR FROM Date) as year_num,
        Confirmed,
        Deaths,
        Recovered,
        RANK() OVER (PARTITION BY EXTRACT(MONTH FROM Date),
                     EXTRACT(YEAR FROM Date)
                     ORDER BY COUNT(*) DESC) as rank
    FROM
        corona_dataset
    GROUP BY
        EXTRACT(MONTH FROM Date), EXTRACT(YEAR FROM Date), Confirmed, Deaths, Recovered
)
SELECT
    month_num,
    year_num,
    Confirmed,
    Deaths,
    Recovered
FROM
    FrequentValues
WHERE
    rank = 1
ORDER BY
    year_num, month_num ASC;
```



	month_num numeric	year_num numeric	confirmed integer	deaths integer	recovered integer
1	1	2020	0	0	0
2	2	2020	0	0	0
3	3	2020	0	0	0
4	4	2020	0	0	0
5	5	2020	0	0	0
6	6	2020	0	0	0
7	7	2020	0	0	0
8	8	2020	0	0	0
9	9	2020	0	0	0
10	10	2020	0	0	0
11	11	2020	0	0	0
12	12	2020	0	0	0
13	1	2021	0	0	0
14	2	2021	0	0	0
15	3	2021	0	0	0
16	4	2021	0	0	0
17	5	2021	0	0	0
18	6	2021	0	0	0

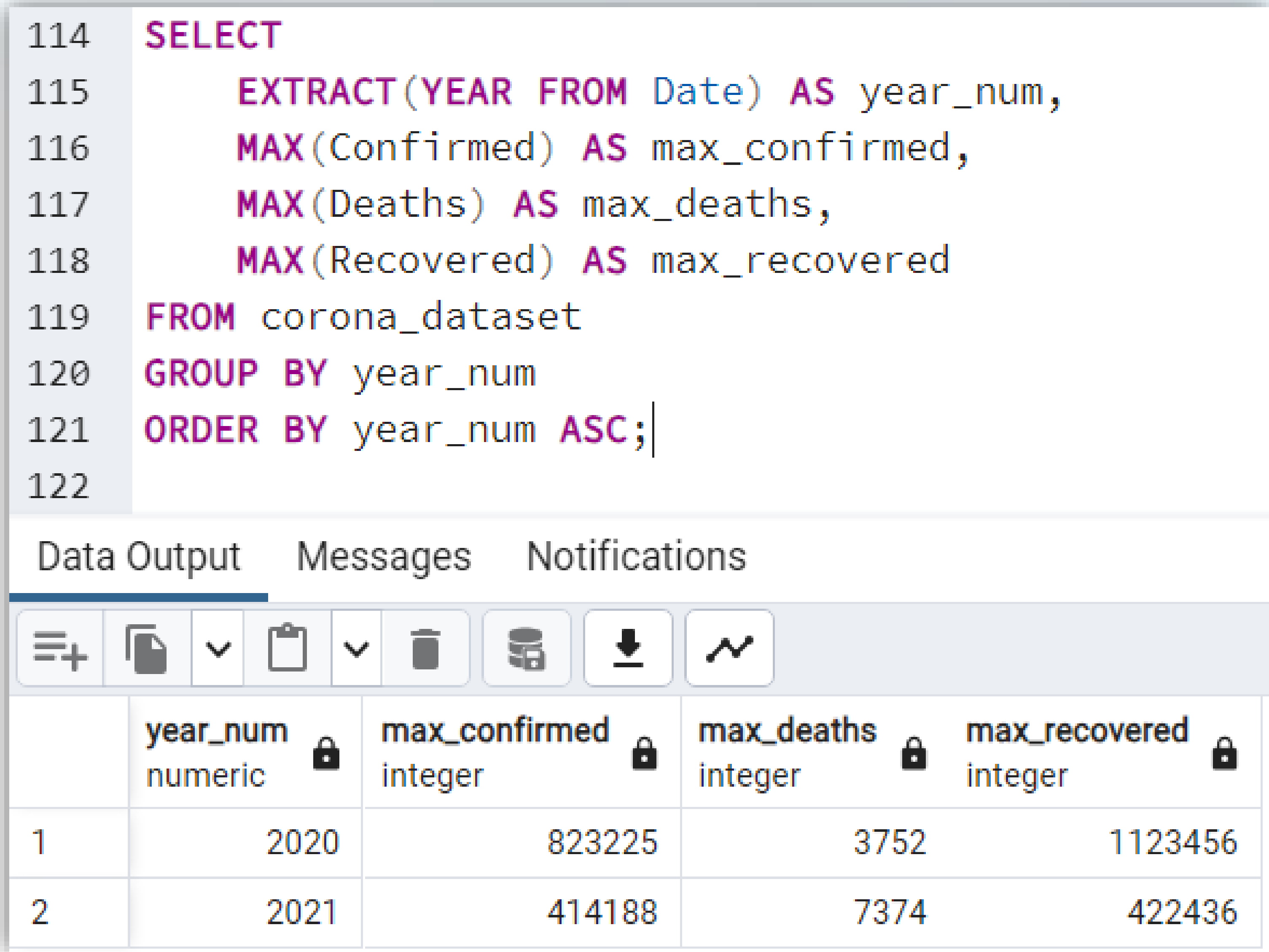
- **8. Find minimum values for confirmed, deaths, recovered per year**



```
SELECT
    EXTRACT(YEAR FROM Date) AS year_num,
    MIN(Confirmed) AS min_confirmed,
    MIN(Deaths) AS min_deaths,
    MIN(Recovered) AS min_recovered
FROM corona_dataset
GROUP BY year_num
ORDER BY year_num ASC;
```

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

- **9. Find maximum values for confirmed, deaths, recovered per year**

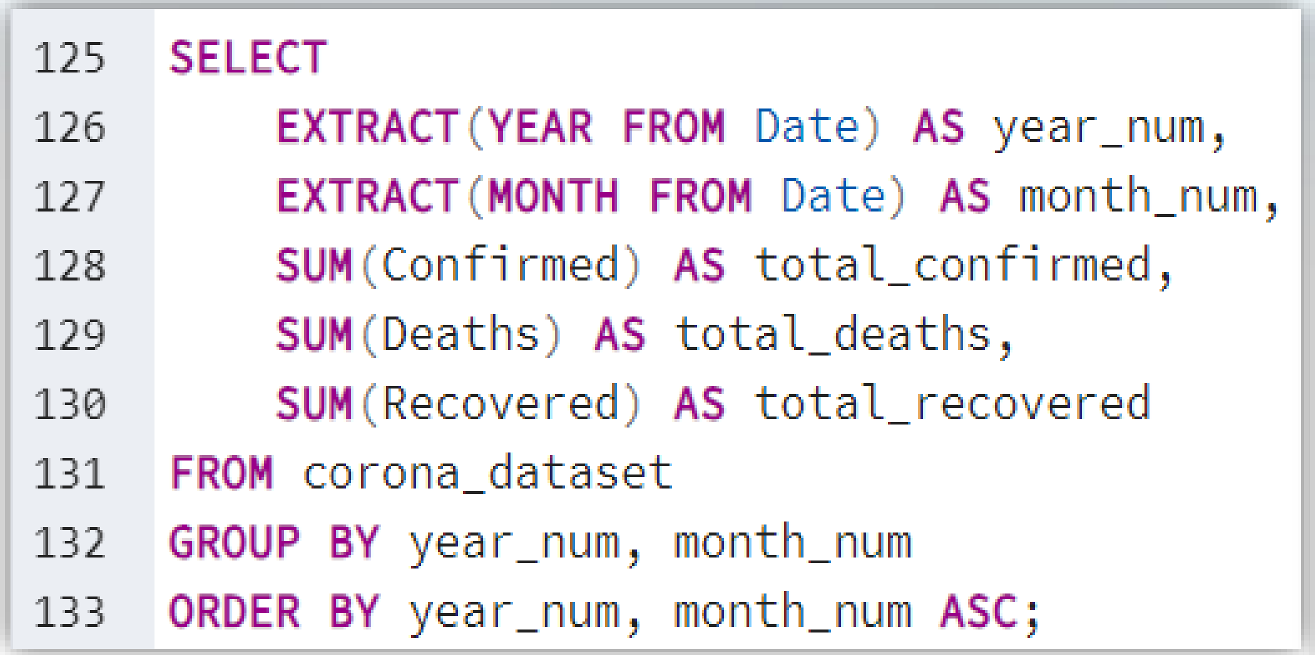


```sql
SELECT
    EXTRACT(YEAR FROM Date) AS year_num,
    MAX(Confirmed) AS max_confirmed,
    MAX(Deaths) AS max_deaths,
    MAX(Recovered) AS max_recovered
FROM corona_dataset
GROUP BY year_num
ORDER BY year_num ASC;
```

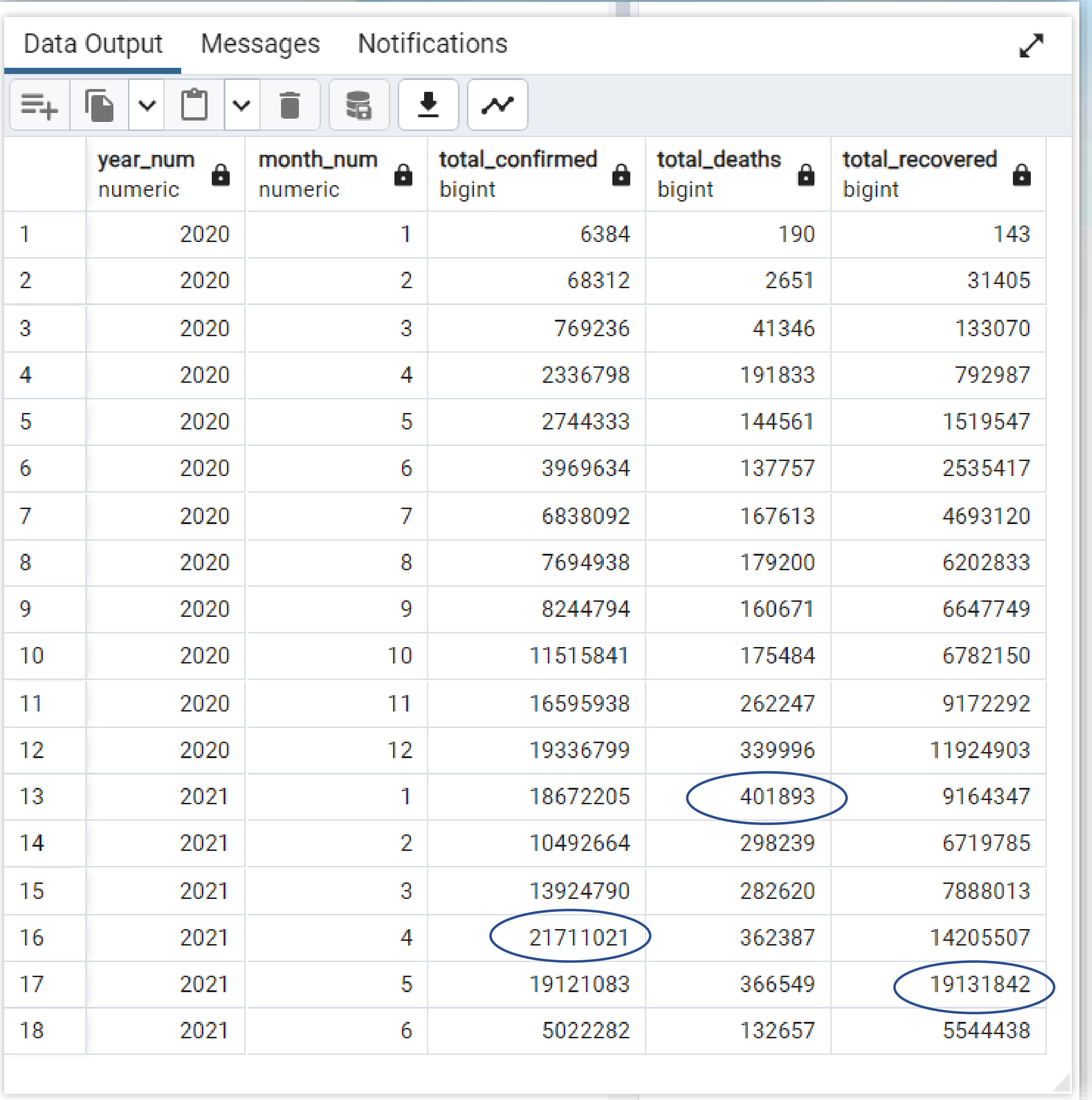
	year_num numeric	max_confirmed integer	max_deaths integer	max_recovered integer
1	2020	823225	3752	1123456
2	2021	414188	7374	422436

- ❑ The year 2020 records the highest number of confirmed cases, with a total of 823,225 cases.
- ❑ In contrast, the year 2021 reports the highest number of deaths, totaling 7,374.
- ❑ However, the maximum number of recovered cases, amounting to 1,123,456, is reported in the year 2020.

- **10. The total number of case of confirmed, deaths, recovered each month**



```sql
SELECT
    EXTRACT(YEAR FROM Date) AS year_num,
    EXTRACT(MONTH FROM Date) AS month_num,
    SUM(Confirmed) AS total_confirmed,
    SUM(Deaths) AS total_deaths,
    SUM(Recovered) AS total_recovered
FROM corona_dataset
GROUP BY year_num, month_num
ORDER BY year_num, month_num ASC;
```



	year_num numeric	month_num numeric	total_confirmed bigint	total_deaths bigint	total_recovered bigint
1	2020	1	6384	190	143
2	2020	2	68312	2651	31405
3	2020	3	769236	41346	133070
4	2020	4	2336798	191833	792987
5	2020	5	2744333	144561	1519547
6	2020	6	3969634	137757	2535417
7	2020	7	6838092	167613	4693120
8	2020	8	7694938	179200	6202833
9	2020	9	8244794	160671	6647749
10	2020	10	11515841	175484	6782150
11	2020	11	16595938	262247	9172292
12	2020	12	19336799	339996	11924903
13	2021	1	18672205	401893	9164347
14	2021	2	10492664	298239	6719785
15	2021	3	13924790	282620	7888013
16	2021	4	21711021	362387	14205507
17	2021	5	19121083	366549	19131842
18	2021	6	5022282	132657	5544438

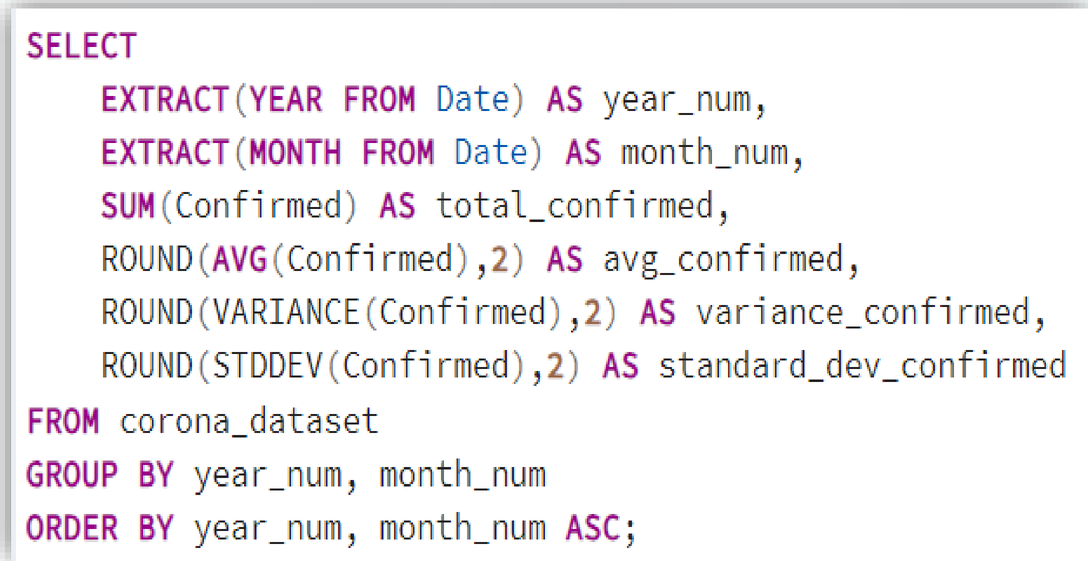
❑ The total number of confirmed cases reached its peak in April 2021, with a total count of 21,711,021.

❑ In contrast, the highest number of deaths was recorded in January 2021, totaling 401,893.

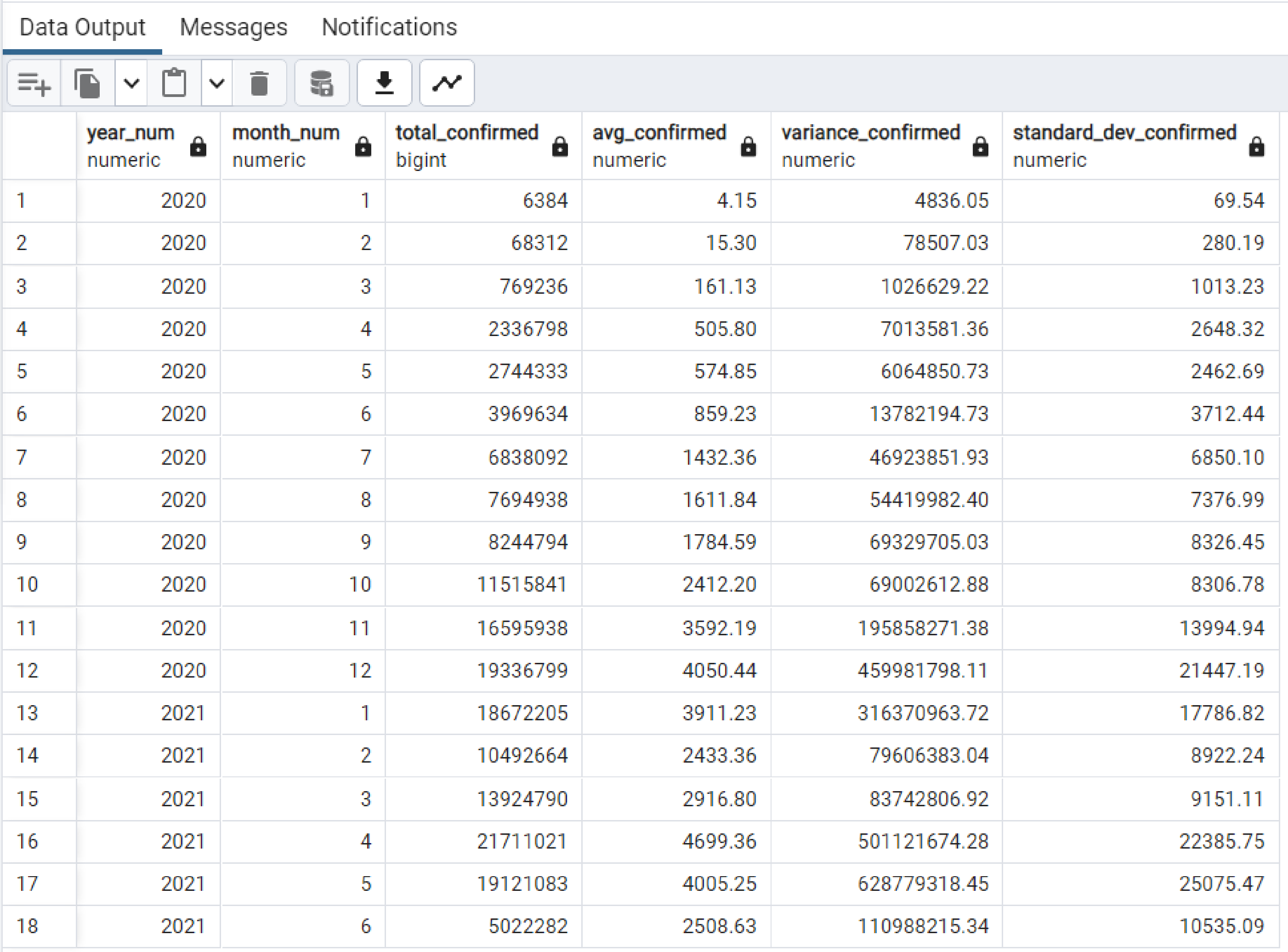
❑ Furthermore, the maximum number of recovered cases was reported in May 2021, amounting to 19,131,842.

- 11. **Check how coronavirus spread out with respect to confirmed cases per month**

(Eg: total confirmed cases, their average, variance & STDEV)



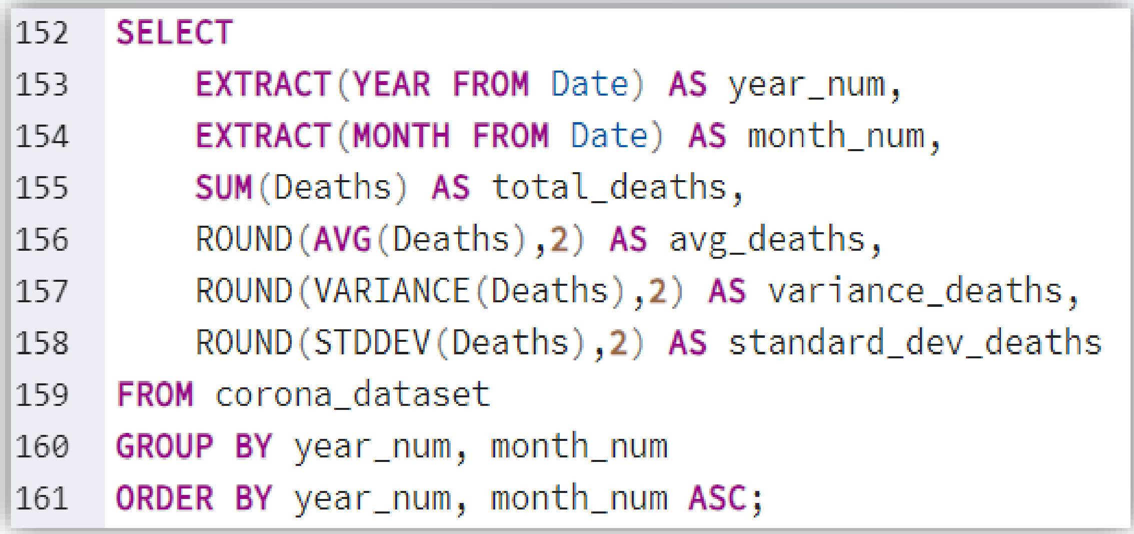
```
SELECT
    EXTRACT(YEAR FROM Date) AS year_num,
    EXTRACT(MONTH FROM Date) AS month_num,
    SUM(Confirmed) AS total_confirmed,
    ROUND(AVG(Confirmed),2) AS avg_confirmed,
    ROUND(VARIANCE(Confirmed),2) AS variance_confirmed,
    ROUND(STDDEV(Confirmed),2) AS standard_dev_confirmed
FROM corona_dataset
GROUP BY year_num, month_num
ORDER BY year_num, month_num ASC;
```



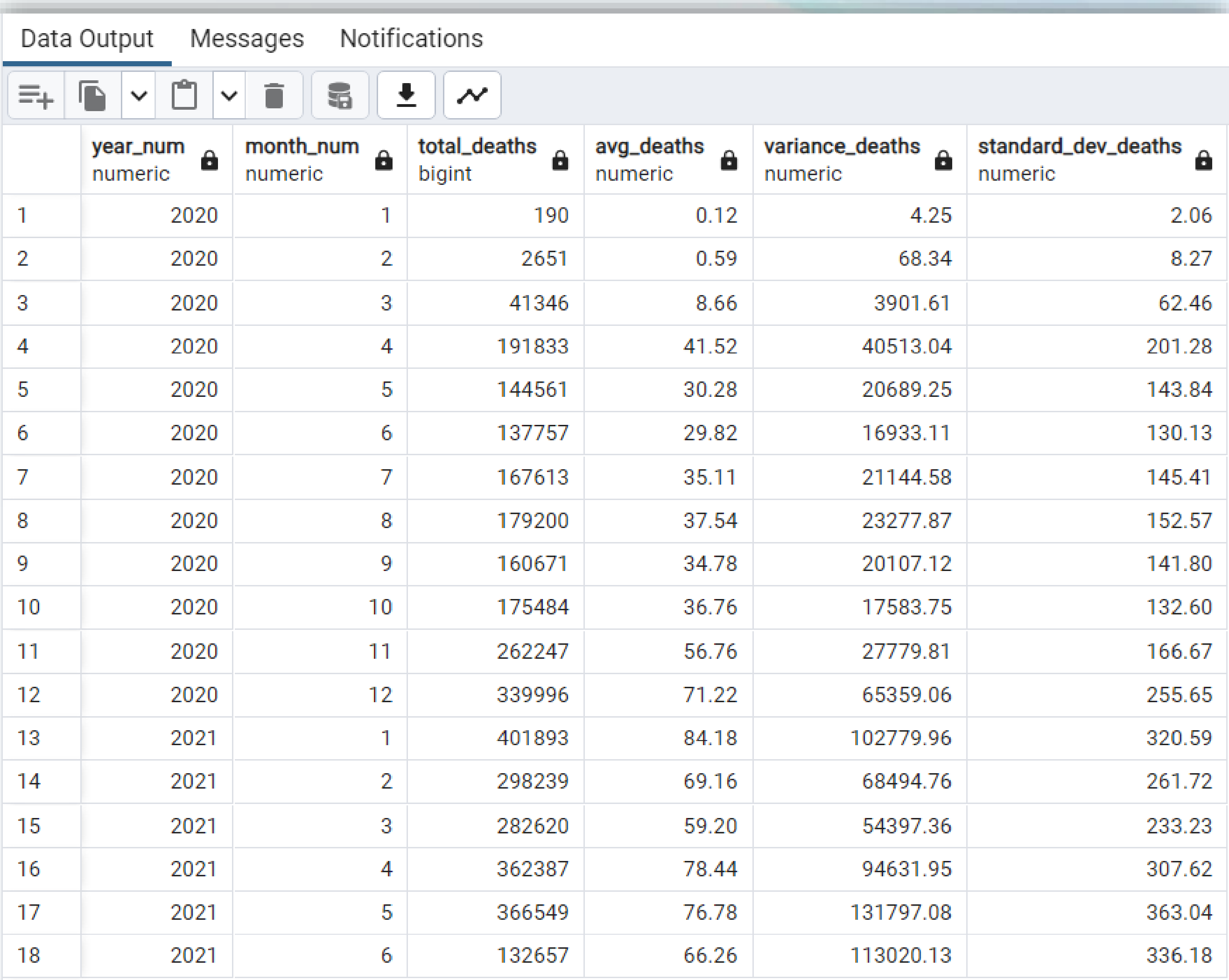
	year_num numeric	month_num numeric	total_confirmed bigint	avg_confirmed numeric	variance_confirmed numeric	standard_dev_confirmed numeric
1	2020	1	6384	4.15	4836.05	69.54
2	2020	2	68312	15.30	78507.03	280.19
3	2020	3	769236	161.13	1026629.22	1013.23
4	2020	4	2336798	505.80	7013581.36	2648.32
5	2020	5	2744333	574.85	6064850.73	2462.69
6	2020	6	3969634	859.23	13782194.73	3712.44
7	2020	7	6838092	1432.36	46923851.93	6850.10
8	2020	8	7694938	1611.84	54419982.40	7376.99
9	2020	9	8244794	1784.59	69329705.03	8326.45
10	2020	10	11515841	2412.20	69002612.88	8306.78
11	2020	11	16595938	3592.19	195858271.38	13994.94
12	2020	12	19336799	4050.44	459981798.11	21447.19
13	2021	1	18672205	3911.23	316370963.72	17786.82
14	2021	2	10492664	2433.36	79606383.04	8922.24
15	2021	3	13924790	2916.80	83742806.92	9151.11
16	2021	4	21711021	4699.36	501121674.28	22385.75
17	2021	5	19121083	4005.25	628779318.45	25075.47
18	2021	6	5022282	2508.63	110988215.34	10535.09

- 12. Check how coronavirus spread out with respect to death cases per month

  (Eg: total death cases, their average, variance & STDEV)



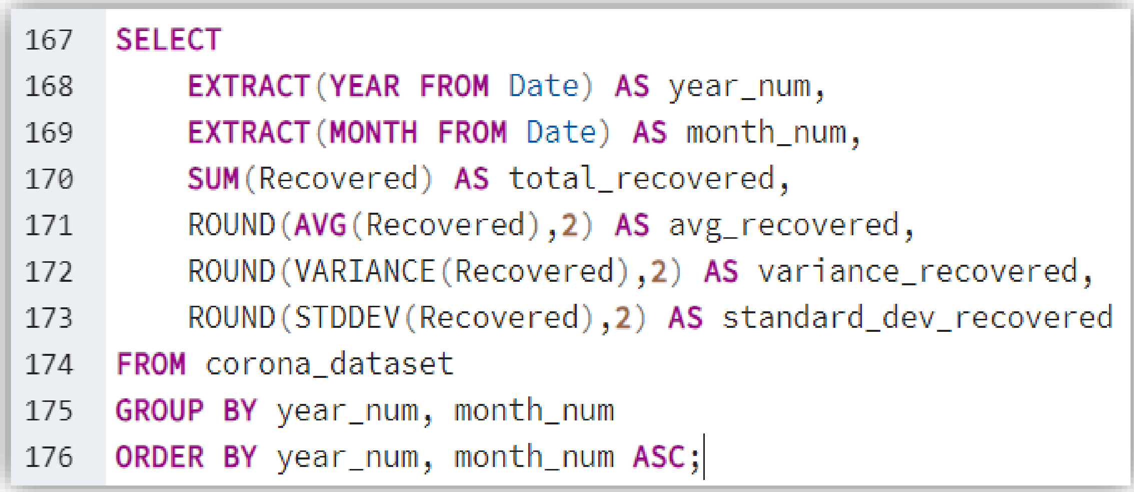
```sql
SELECT
    EXTRACT(YEAR FROM Date) AS year_num,
    EXTRACT(MONTH FROM Date) AS month_num,
    SUM(Deaths) AS total_deaths,
    ROUND(AVG(Deaths),2) AS avg_deaths,
    ROUND(VARIANCE(Deaths),2) AS variance_deaths,
    ROUND(STDDEV(Deaths),2) AS standard_dev_deaths
FROM corona_dataset
GROUP BY year_num, month_num
ORDER BY year_num, month_num ASC;
```



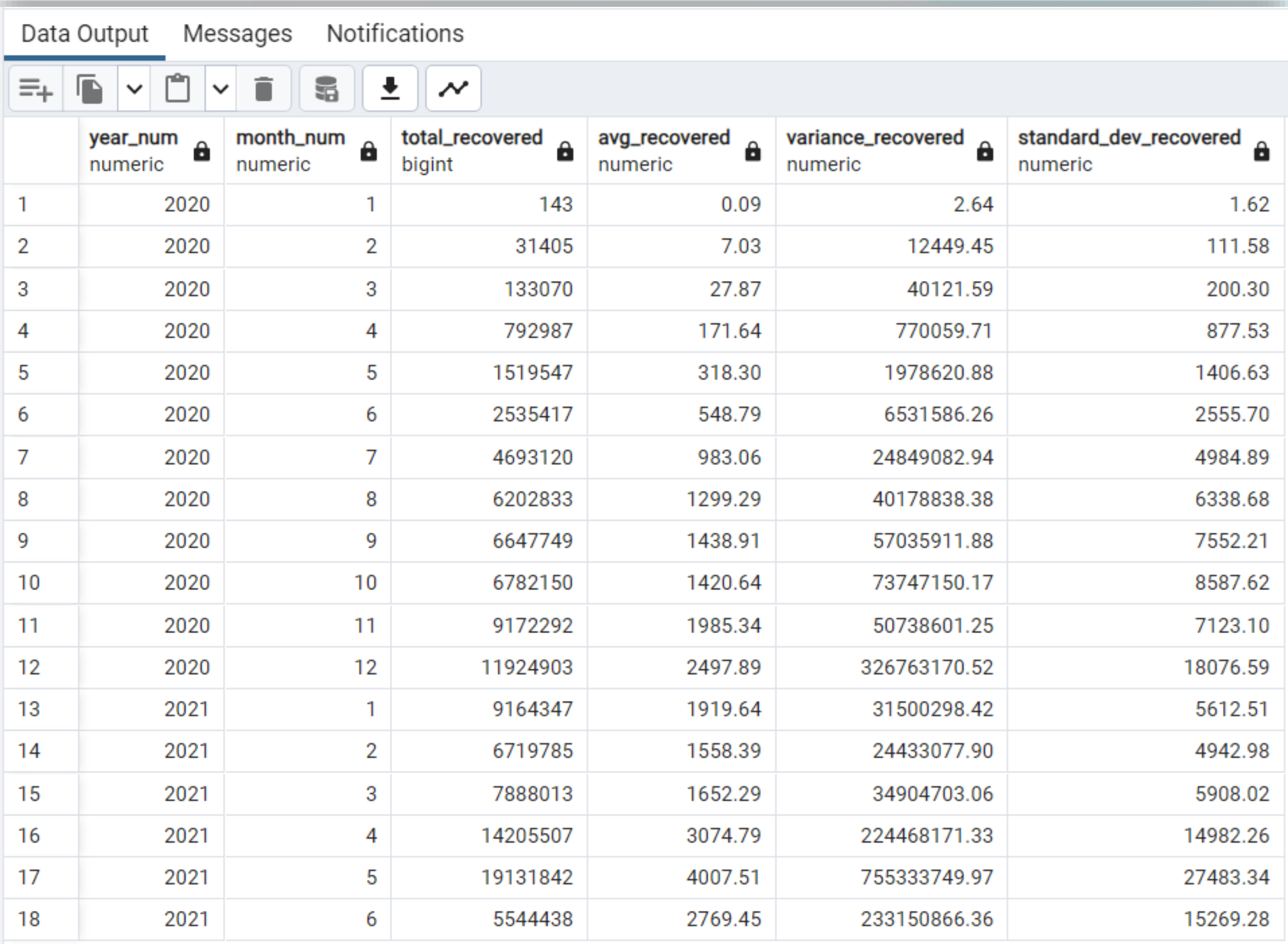
	year_num numeric	month_num numeric	total_deaths bigint	avg_deaths numeric	variance_deaths numeric	standard_dev_deaths numeric
1	2020	1	190	0.12	4.25	2.06
2	2020	2	2651	0.59	68.34	8.27
3	2020	3	41346	8.66	3901.61	62.46
4	2020	4	191833	41.52	40513.04	201.28
5	2020	5	144561	30.28	20689.25	143.84
6	2020	6	137757	29.82	16933.11	130.13
7	2020	7	167613	35.11	21144.58	145.41
8	2020	8	179200	37.54	23277.87	152.57
9	2020	9	160671	34.78	20107.12	141.80
10	2020	10	175484	36.76	17583.75	132.60
11	2020	11	262247	56.76	27779.81	166.67
12	2020	12	339996	71.22	65359.06	255.65
13	2021	1	401893	84.18	102779.96	320.59
14	2021	2	298239	69.16	68494.76	261.72
15	2021	3	282620	59.20	54397.36	233.23
16	2021	4	362387	78.44	94631.95	307.62
17	2021	5	366549	76.78	131797.08	363.04
18	2021	6	132657	66.26	113020.13	336.18

- 13. **Check how coronavirus spread out with respect to recovered cases per month**

(Eg: total recovered cases, their average, variance & STDEV)

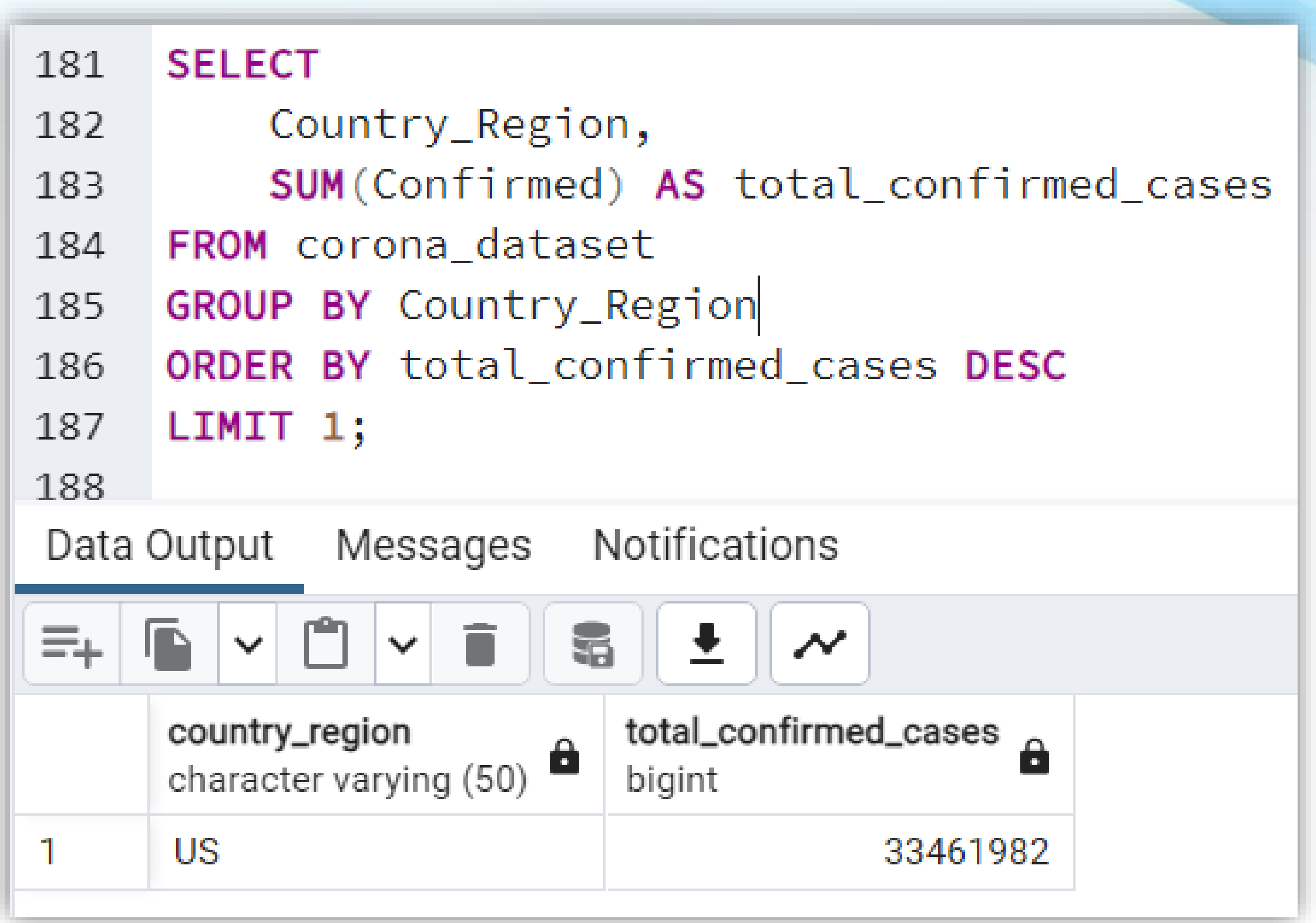


```sql
SELECT
    EXTRACT(YEAR FROM Date) AS year_num,
    EXTRACT(MONTH FROM Date) AS month_num,
    SUM(Recovered) AS total_recovered,
    ROUND(AVG(Recovered),2) AS avg_recovered,
    ROUND(VARIANCE(Recovered),2) AS variance_recovered,
    ROUND(STDDEV(Recovered),2) AS standard_dev_recovered
FROM corona_dataset
GROUP BY year_num, month_num
ORDER BY year_num, month_num ASC;
```



	year_num numeric	month_num numeric	total_recovered bigint	avg_recovered numeric	variance_recovered numeric	standard_dev_recovered numeric
1	2020	1	143	0.09	2.64	1.62
2	2020	2	31405	7.03	12449.45	111.58
3	2020	3	133070	27.87	40121.59	200.30
4	2020	4	792987	171.64	770059.71	877.53
5	2020	5	1519547	318.30	1978620.88	1406.63
6	2020	6	2535417	548.79	6531586.26	2555.70
7	2020	7	4693120	983.06	24849082.94	4984.89
8	2020	8	6202833	1299.29	40178838.38	6338.68
9	2020	9	6647749	1438.91	57035911.88	7552.21
10	2020	10	6782150	1420.64	73747150.17	8587.62
11	2020	11	9172292	1985.34	50738601.25	7123.10
12	2020	12	11924903	2497.89	326763170.52	18076.59
13	2021	1	9164347	1919.64	31500298.42	5612.51
14	2021	2	6719785	1558.39	24433077.90	4942.98
15	2021	3	7888013	1652.29	34904703.06	5908.02
16	2021	4	14205507	3074.79	224468171.33	14982.26
17	2021	5	19131842	4007.51	755333749.97	27483.34
18	2021	6	5544438	2769.45	233150866.36	15269.28

- **14. Find the Country having the highest number of Confirmed cases**

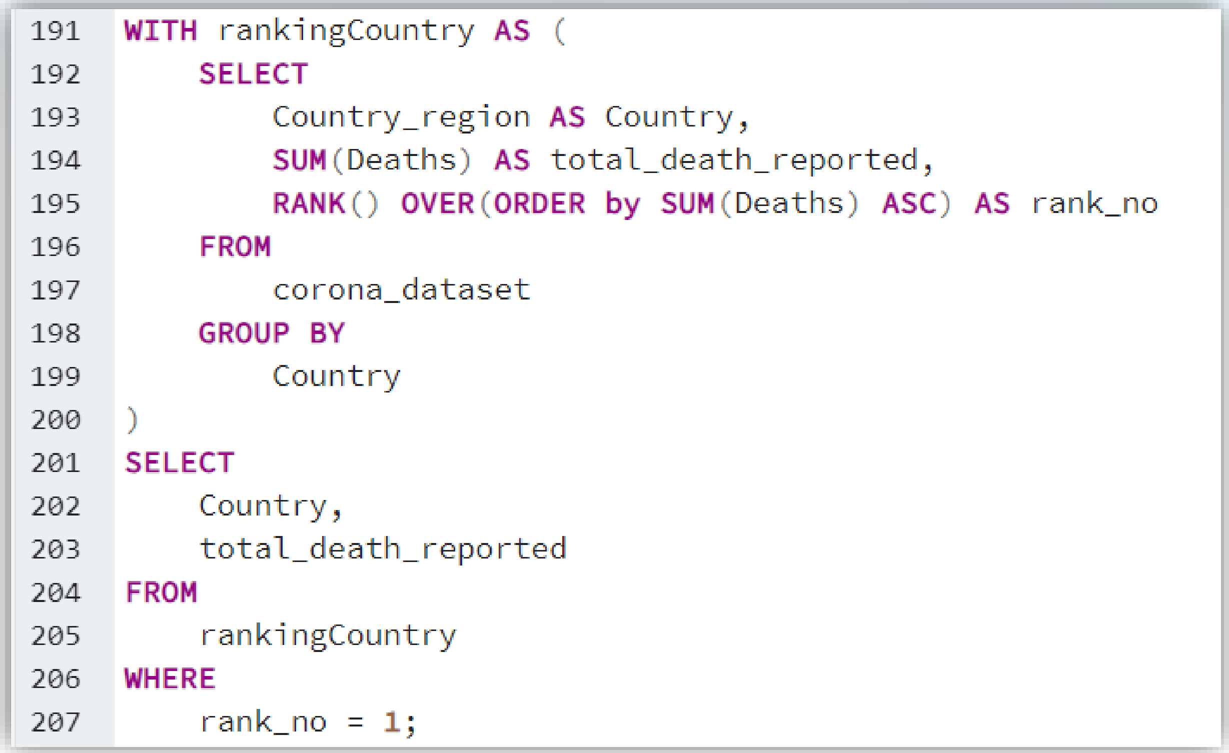


```sql
SELECT
    Country_Region,
    SUM(Confirmed) AS total_confirmed_cases
FROM corona_dataset
GROUP BY Country_Region
ORDER BY total_confirmed_cases DESC
LIMIT 1;
```

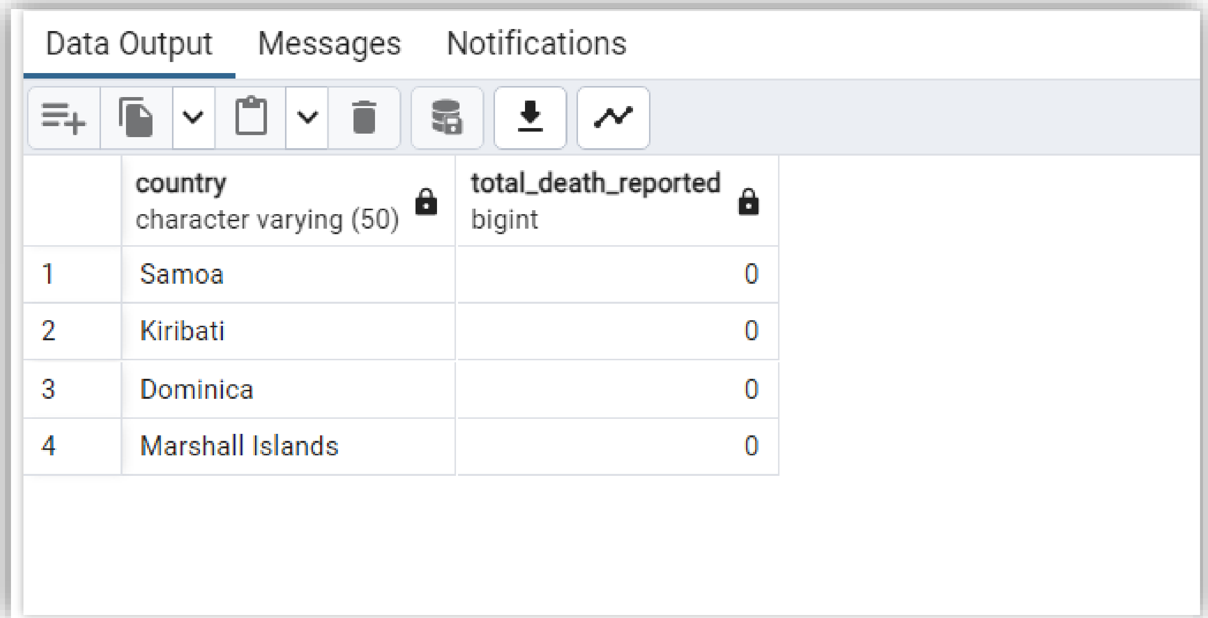
	country_region character varying (50)	total_confirmed_cases bigint
1	US	33461982

- **Inference: US** has the highest number of confirmed COVID-19 cases, totaling 33,461,982 according to the dataset

▪ 15. Find the Country having the lowest number of death cases



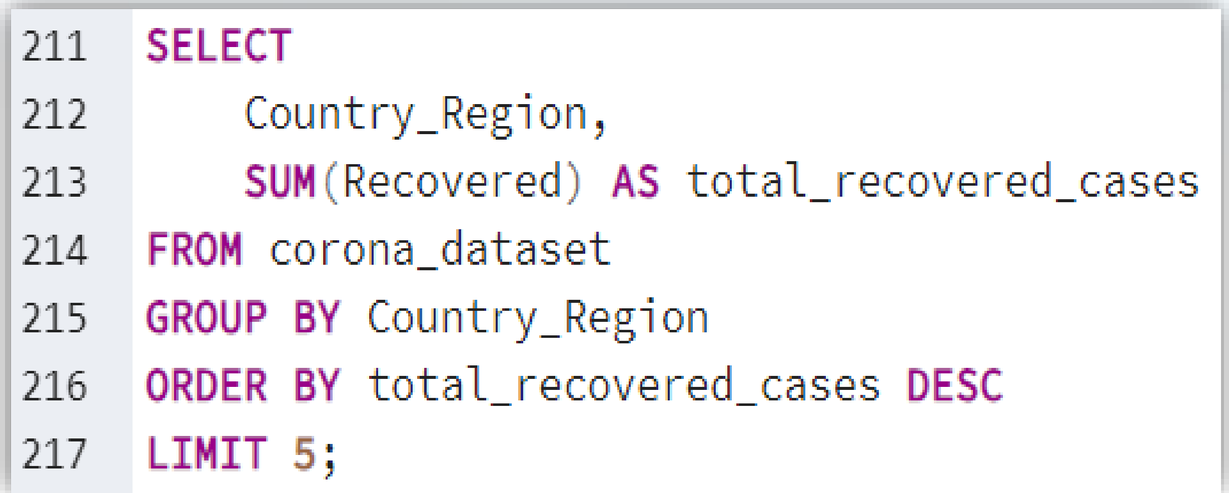
```
WITH rankingCountry AS (
    SELECT
        Country_region AS Country,
        SUM(Deaths) AS total_death_reported,
        RANK() OVER(ORDER by SUM(Deaths) ASC) AS rank_no
    FROM
        corona_dataset
    GROUP BY
        Country
)
SELECT
    Country,
    total_death_reported
FROM
    rankingCountry
WHERE
    rank_no = 1;
```



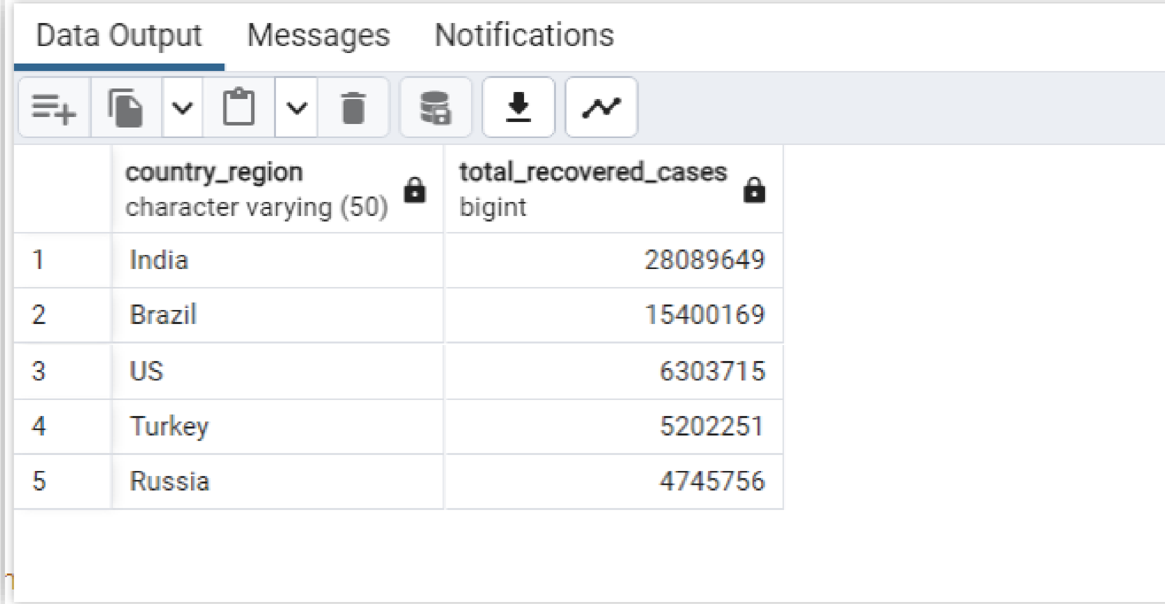
	country character varying (50)	total_death_reported bigint
1	Samoa	0
2	Kiribati	0
3	Dominica	0
4	Marshall Islands	0

❑ **Samoa, Kiribati, Dominica**, and the **Marshall Islands** have reported the lowest number of death cases, with each country recording 0 fatalities

- **16. Find top 5 countries having highest recovered cases**



```sql
SELECT
    Country_Region,
    SUM(Recovered) AS total_recovered_cases
FROM corona_dataset
GROUP BY Country_Region
ORDER BY total_recovered_cases DESC
LIMIT 5;
```



	country_region character varying (50)	total_recovered_cases bigint
1	India	28089649
2	Brazil	15400169
3	US	6303715
4	Turkey	5202251
5	Russia	4745756

❑ **India, Brazil, US, Turkey,** and **Russia** are the top five countries with the highest number of recovered COVID-19 cases.

Insights

After analyzing the COVID dataset using SQL, several insights have been uncovered:

1. **COVID-19 Pandemic duration: January 22, 2020, to June 13, 2021.**
2. **India has the highest number of recovered cases.**
3. **Samoa, Kiribati, Dominica, and the Marshall Islands have the lowest death counts.**
4. **The US leads in confirmed COVID-19 cases.**
5. **Peak confirmed cases occurred in April 2021.**
6. **Peak death rate in January 2021.**

These insights provide valuable information for understanding the progression and impact of the COVID-19 pandemic based on the provided dataset.

Thank You!