Analyze COVID-19 Data with SQL

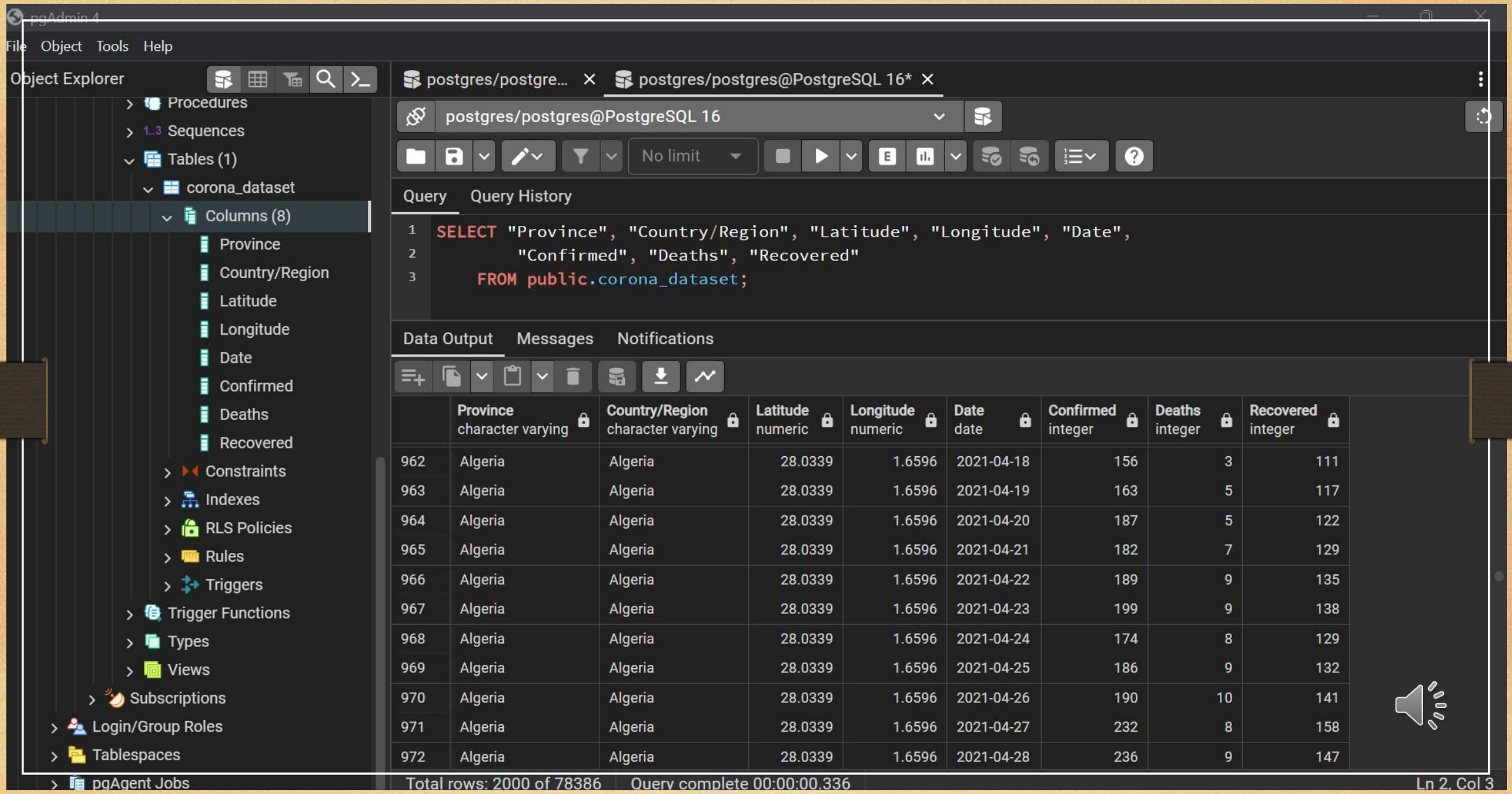
Exploring Confirmed Cases, Deaths, Recoveries

Presented By:
Majdi AlKotamy

Table Schema

- **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.

Table View


pgAdmin 4
File Object Tools Help
Object Explorer
Procedures
Sequences
Tables (1)
corona_dataset
Columns (8)
Province
Country/Region
Latitude
Longitude
Date
Confirmed
Deaths
Recovered
Constraints
Indexes
RLS Policies
Rules
Triggers
Trigger Functions
Types
Views
Subscriptions
Login/Group Roles
Tablespaces
pgAgent Jobs
postgres/postgre...
postgres/postgres@PostgreSQL 16*
postgres/postgres@PostgreSQL 16
No limit
Query
Query History
1 SELECT "Province", "Country/Region", "Latitude", "Longitude", "Date",
2 "Confirmed", "Deaths", "Recovered"
3 FROM public.corona_dataset;
Data Output
Messages
Notifications
Province character varying
Country/Region character varying
Latitude numeric
Longitude numeric
Date date
Confirmed integer
Deaths integer
Recovered integer
962 Algeria Algeria 28.0339 1.6596 2021-04-18 156 3 111
963 Algeria Algeria 28.0339 1.6596 2021-04-19 163 5 117
964 Algeria Algeria 28.0339 1.6596 2021-04-20 187 5 122
965 Algeria Algeria 28.0339 1.6596 2021-04-21 182 7 129
966 Algeria Algeria 28.0339 1.6596 2021-04-22 189 9 135
967 Algeria Algeria 28.0339 1.6596 2021-04-23 199 9 138
968 Algeria Algeria 28.0339 1.6596 2021-04-24 174 8 129
969 Algeria Algeria 28.0339 1.6596 2021-04-25 186 9 132
970 Algeria Algeria 28.0339 1.6596 2021-04-26 190 10 141
971 Algeria Algeria 28.0339 1.6596 2021-04-27 232 8 158
972 Algeria Algeria 28.0339 1.6596 2021-04-28 236 9 147
Total rows: 2000 of 78386
Query complete 00:00:00.336
Ln 2, Col 3

To avoid any errors, will check if there missing or null value

postgres/postgres@PostgreSQL 16*

postgres/postgres@PostgreSQL 16

No limit

Query | Query History

```
5
6  SELECT *
7  FROM corona_dataset
8  WHERE
9  "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;
13
14
```

Data Output | Messages | Notifications

Province character varying	Country/Region character varying	Latitude numeric	Longitude numeric	Date date	Confirmed integer	Deaths integer	Recovered integer

Query Query History

```
29  -- Q3. check total number of rows
30
31  SELECT COUNT(*) AS total_rows
32  FROM corona_dataset;
33
34
```

Data Output Messages Notifications

	total_rows bigint
1	78386

Query Query History

```
39
40  -- Q4. Check what is start_date and end_date
41
42  SELECT MIN("Date") AS start_date,
43         MAX("Date") AS end_date
44  FROM corona_dataset;
45
46
```

Data Output Messages Notifications

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

Query Query History

```
47
48  -- Q5. Number of month present in dataset
49
50  SELECT COUNT(DISTINCT DATE_TRUNC('month', "Date"))
51         AS num_months
52  FROM corona_dataset;
53
54
```

Data Output Messages Notifications

	num_months bigint
1	18

```sql
-- Q6. Find monthly average for confirmed, deaths, recovered

SELECT EXTRACT(MONTH FROM t."Date") as month , EXTRACT(YEAR FROM  "Date") AS year,
AVG("Confirmed") AS AVG_CONFIRMED,
AVG("Deaths") AS AVG_DEATHS,
AVG("Recovered") AS AVG_RECOVERED
FROM corona_dataset t
GROUP BY month, year
order by year, month;
```

Output | Messages | Notifications

month numeric	year numeric	avg_confirmed numeric	avg_deaths numeric	avg_recovered numeric
1	2020	4.1454545454545455	0.12337662337662337662	0.09285714285714285714
2	2020	15.2960143304970891	0.59359605911330049261	7.0320197044334975
3	2020	161.1302890657729367	8.6606619187264349	27.8739002932551320
4	2020	505.8004329004329004	41.5222943722943723	171.6422077922077922
5	2020	574.8498114788437369	30.2808965228320067	318.2963971512358609
6	2020	859.2281385281385281	29.8175324675324675	548.7915584415584416

```sql
-- Q7. Find most frequent value for confirmed, deaths, recovered each month

WITH MonthlyStats AS (
    SELECT
        EXTRACT(YEAR FROM "Date") AS "Year",
        EXTRACT(MONTH FROM "Date") AS "Month",
        "Confirmed",  "Deaths", "Recovered",
        ROW_NUMBER() OVER (PARTITION BY EXTRACT(YEAR FROM "Date"), EXTRACT(MONTH FROM "Date")
        ORDER BY COUNT(*) DESC) AS rn
    FROM  corona_dataset
    GROUP BY
        EXTRACT(YEAR FROM "Date"), EXTRACT(MONTH FROM "Date"),
        "Confirmed", "Deaths", "Recovered" )
SELECT
    TO_CHAR(TO_DATE("Month"::text, 'MM'), 'Month') AS "Month",   "Year",
    "Confirmed" AS "MostFrequentConfirmed",
    "Deaths" AS "MostFrequentDeaths",
    "Recovered" AS "MostFrequentRecovered"
FROM    MonthlyStats
WHERE    rn = 1;
```

Data Output | Messages | Graph Visualiser | Notifications

	Month text	Year numeric	MostFrequentConfirmed integer	MostFrequentDeaths integer	MostFrequentRecovered integer
1	January	2020	0	0	0
2	February	2020	0	0	0
3	March	2020	0	0	0
4	April	2020	0	0	0
5	May	2020	0	0	0
6	June	2020	0	0	0
7	July	2020	0	0	0
8	August	2020	0	0	0
9	September	2020	0	0	0
10	October	2020	0	0	0
11	November	2020	0	0	0
12	December	2020	0	0	0
13	January	2021	0	0	0
14	February	2021	0	0	0
15	March	2021	0	0	0
16	April	2021	0	0	0

Total rows: 18 of 18 | Query complete 00:00:00.442

```sql
-- Q8. Find minimum values for confirmed, deaths, recovered per year

SELECT
    EXTRACT(YEAR FROM "Date") AS "Year",
    MIN("Confirmed") AS "MinConfirmed",
    MIN("Deaths") AS "MinDeaths",
    MIN("Recovered") AS "MinRecovered"
FROM
    corona_dataset
GROUP BY "Year";
```

ta Output | Messages | Graph Visualiser | Notifications

Year numeric	MinConfirmed integer	MinDeaths integer	MinRecovered integer
2021	0	0	0
2020	0	0	0

```sql
-- Q9. Find maximum values of confirmed, deaths, recovered per year

SELECT
    EXTRACT(YEAR FROM "Date") AS "Year",
    MAX("Confirmed") AS "MaxConfirmed",
    MAX("Deaths") AS "MaxDeaths",
    MAX("Recovered") AS "MaxRecovered"
FROM
    corona_dataset
GROUP BY "Year";
```

Output | Messages | Graph Visualiser | Notifications

Year numeric	MaxConfirmed integer	MaxDeaths integer	MaxRecovered integer
2021	414188	7374	422436
2020	823225	3752	1123456

Query | Query History

```
115 -- Q10. The total number of case of confirmed, deaths, recovered each month
116
117 SELECT
118     EXTRACT(YEAR FROM "Date") AS "Year",
119     EXTRACT(MONTH FROM "Date") AS "Month",
120     SUM("Confirmed") AS "TotalConfirmed",
121     SUM("Deaths") AS "TotalDeaths",
122     SUM("Recovered") AS "TotalRecovered"
123 FROM corona_dataset
124 GROUP BY     "Year" , "Month"
125 ORDER BY     "Year", "Month";
126
```

Data Output | Messages | Graph Visualiser | Notifications

	Year numeric	Month numeric	TotalConfirmed bigint	TotalDeaths bigint	TotalRecovered 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

Total rows: 18 of 18 | Query complete 00:00:00.199

Query | Query History

```sql
-- Q11. Check how corona virus spread out with respect to confirmed case
--      (Eg.: total confirmed cases, their average, variance & STDEV )
WITH Stats AS (
    SELECT
        SUM("Confirmed") AS TotalConfirmedCases,
        AVG("Confirmed") AS AverageConfirmedCases,
        VARIANCE("Confirmed") AS VarianceConfirmedCases,
        STDDEV("Confirmed") AS StandardDeviationConfirmedCases
    FROM   corona_dataset)
SELECT
    TotalConfirmedCases,
    AverageConfirmedCases,
    VarianceConfirmedCases,
    StandardDeviationConfirmedCases
FROM     Stats;
```

Data Output | Messages | Graph Visualiser | Notifications

	totalconfirmedcases bigint	averageconfirmedcases numeric	varianceconfirmedcases numeric	standarddeviationconfirmedcases numeric
1	169065144	2156.8283111780164825	157290931.69817455	12541.56815148

Total rows: 1 of 1 | Query complete 00:00:00.173

```sql
-- Q12. Check how corona virus spread out with respect to death case per month
--      (Eg.: total confirmed cases, their average, variance & STDEV )
WITH MonthlyDeathStats AS (
    SELECT
        EXTRACT(YEAR FROM "Date") AS "Year",
        EXTRACT(MONTH FROM "Date") AS "Month",
        SUM("Deaths") AS "TotalDeaths",
        AVG("Deaths")  AS "AverageDeaths",
        VARIANCE("Deaths")  AS "VarianceDeaths",
        STDDEV("Deaths") AS "StandardDeviationDeaths"
    FROM        corona_dataset
    GROUP BY    "Year", "Month"
    ORDER BY    "Year", "Month" )
SELECT
    TO_CHAR(TO_DATE("Month"::text, 'MM'), 'Month') AS "Month",
    "Year",
    "TotalDeaths",
    "AverageDeaths",
    "VarianceDeaths",
    "StandardDeviationDeaths"
FROM     MonthlyDeathStats;

```

Data Output | Messages | Graph Visualiser | Notifications

Total rows: 18 of 18 | Query complete 00:00:00.226

Query | Query History

```sql
-- Q13. Check how corona virus spread out with respect to recovered case
--      (Eg.: total confirmed cases, their average, variance & STDEV )

WITH Stats AS (
    SELECT
        SUM("Recovered") AS TotalRecoveredCases,
        AVG("Recovered") AS AverageRecoveredCases,
        VARIANCE("Recovered") AS VarianceRecoveredCases,
        STDDEV("Recovered") AS StandardDeviationRecoveredCases
    FROM  corona_dataset )
SELECT
    TotalRecoveredCases,
    AverageRecoveredCases,
    VarianceRecoveredCases,
    StandardDeviationRecoveredCases
FROM   Stats;
```

Data Output | Messages | Graph Visualiser | Notifications

	totalrecoveredcases bigint	averagerecoveredcases numeric	variancerecoveredcases numeric	standarddeviationrecoveredcases numeric
1	113089548	1442.7263541959023295	107030888.69602982	10345.57338653

Total rows: 1 of 1 | Query complete 00:00:00.134

```sql
-- Q14. Find Country having highest number of the Confirmed case


SELECT
    "Country/Region" AS "Country",
    MAX("Confirmed") AS "HighestConfirmedCases"
FROM    corona_dataset
GROUP BY
    "Country/Region"
ORDER BY
    "HighestConfirmedCases" DESC
LIMIT 1;

```

Data Output | Messages | Graph Visualiser | Notifications

	Country character varying	HighestConfirmedCases integer
1	Turkey	823225

```sql
-- Q15. Find Country having lowest number of the death case

SELECT
    "Country/Region" AS "Country",
    MIN("Deaths") AS "LowestDeathCases"
FROM
    corona_dataset
GROUP BY
    "Country/Region"
ORDER BY
    "LowestDeathCases" ASC
LIMIT 1;

```

Data Output | Messages | Graph Visualiser | Notifications

	Country character varying	LowestDeathCases integer
1	Indonesia	0

Query | Query History

```sql
-- Q16. Find top 5 countries having highest recovered case
SELECT
    "Country/Region" AS "Country",
    SUM("Recovered") AS "TotalRecoveredCases"
FROM
    corona_dataset
GROUP BY
    "Country/Region"
ORDER BY
    "TotalRecoveredCases" DESC
LIMIT 5;
```

Data Output | Messages | Graph Visualiser | Notifications

	Country character varying	TotalRecoveredCases bigint
1	India	28089649
2	Brazil	15400169
3	US	6303715
4	Turkey	5202251
5	Russia	4745756

Total rows: 5 of 5 | Query complete 00:00:00.133