

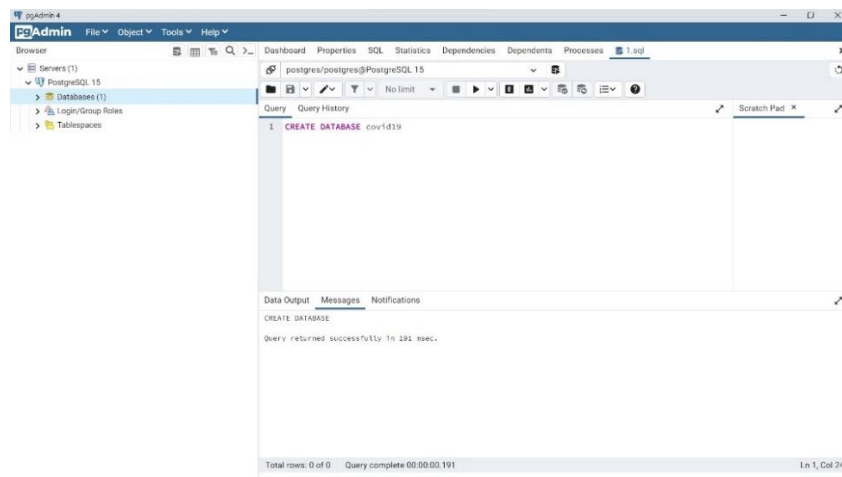
110550108 施柏江

Introduction to Database System_HW1

1. The process of creating the “covid19” databases (can be screenshot and/or SQL/non-SQL statements with text explanation) (10pts)

Ans:

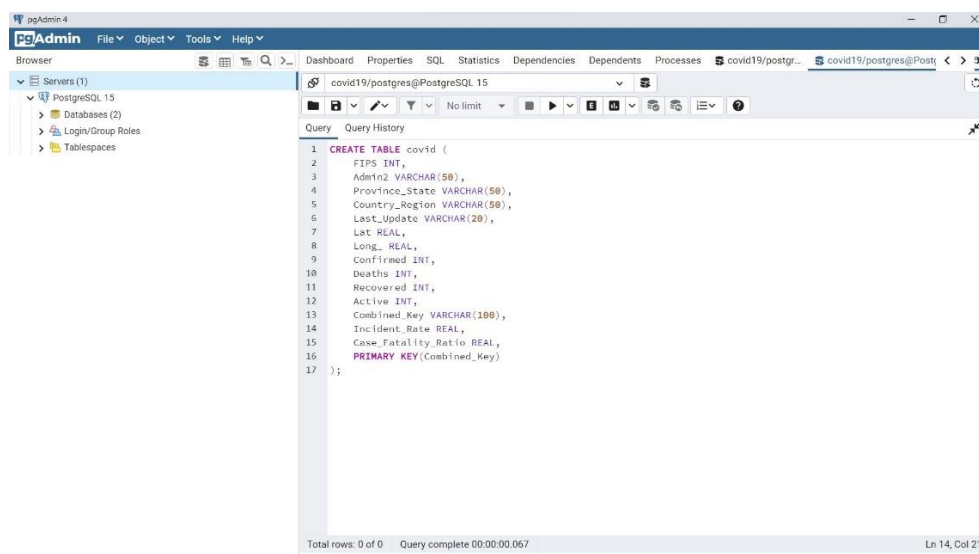
Create database covid19



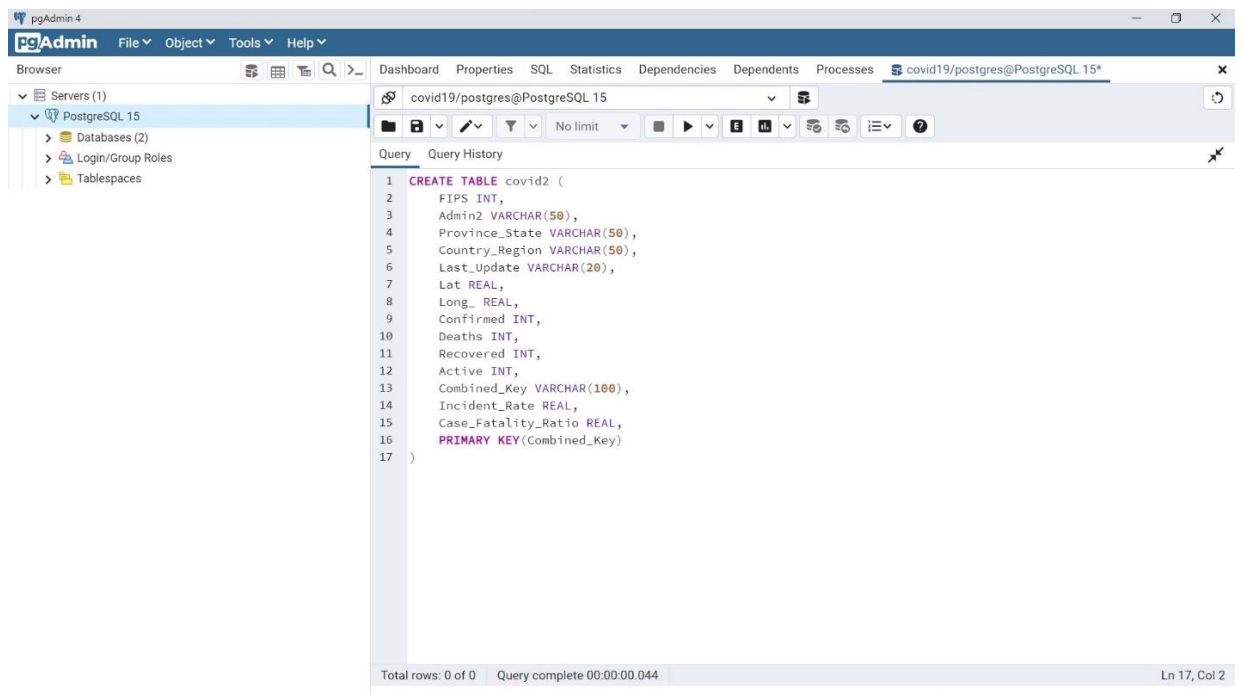
2. The process of importing three required .csv files into covid19 database (can be screenshot and/or SQL/non-SQL statements with text explanation). Please included/described the data type and keys of the imported table in your screenshot, SQL statements, and explanations (30pts)

Ans:

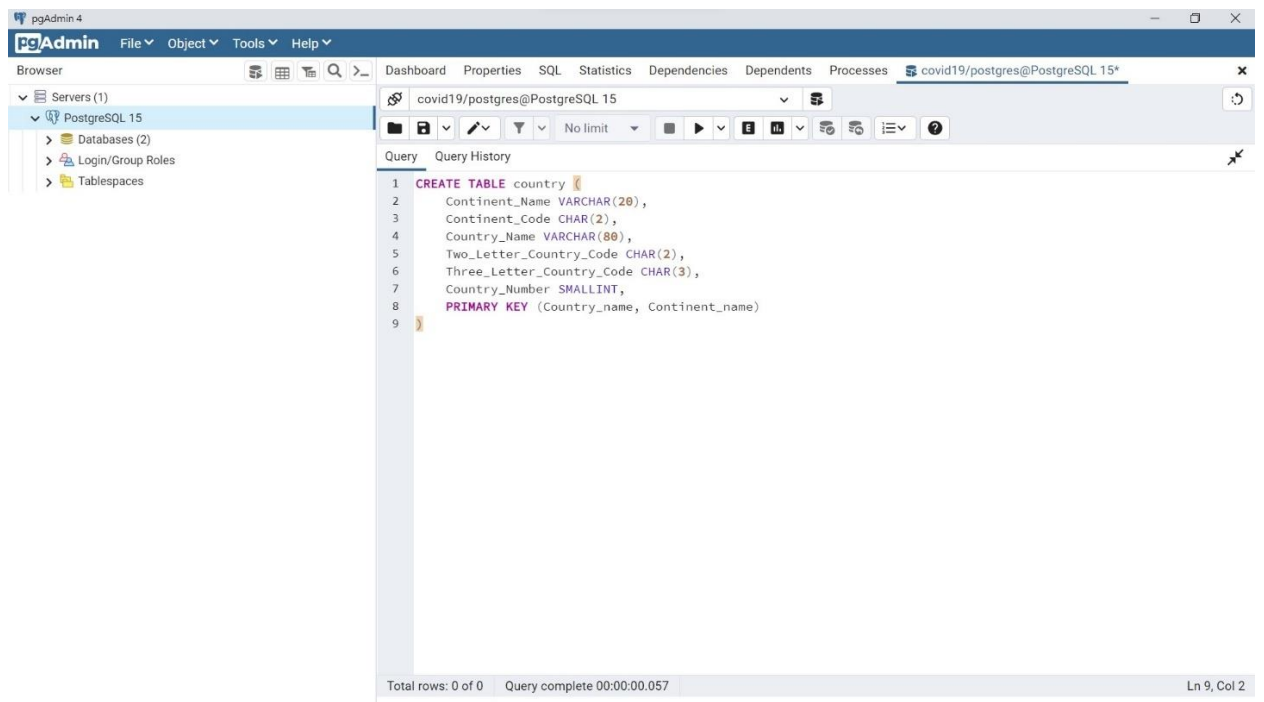
創建 table “covid” 存放 10-11-2022 的 data



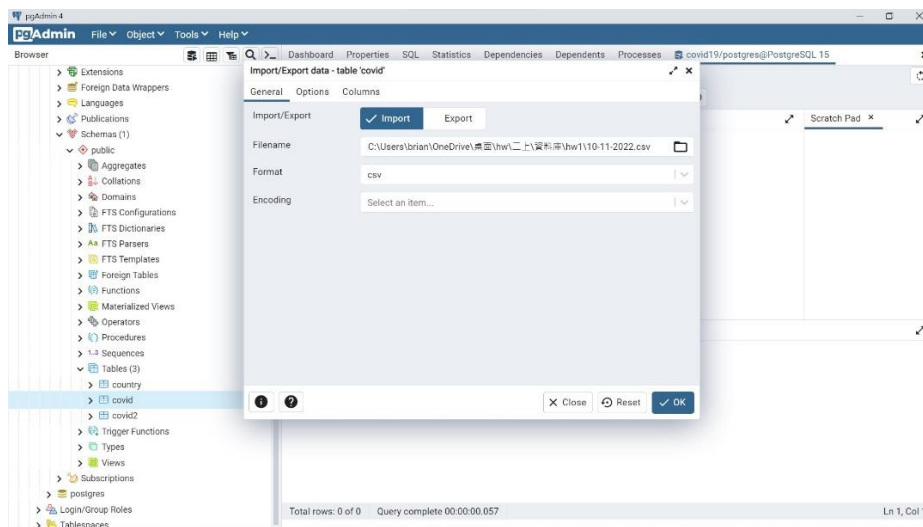
創建 table “covid2” 存放 10-01-2022 的 data



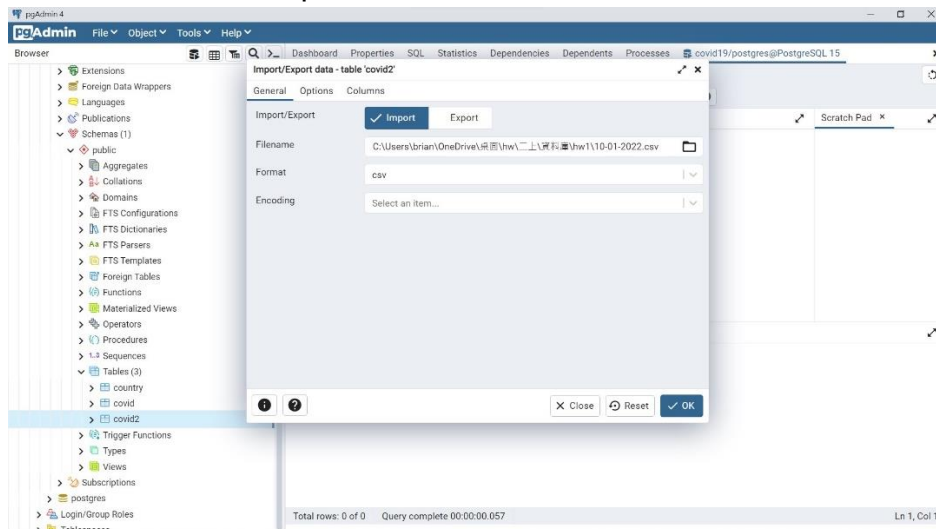
創建 table “country” 存放 Country code and continents mapping table 的 data
(因資料內 country_name 並非 unique，故選用 country_name 和
continent_name 作為 primary keys)



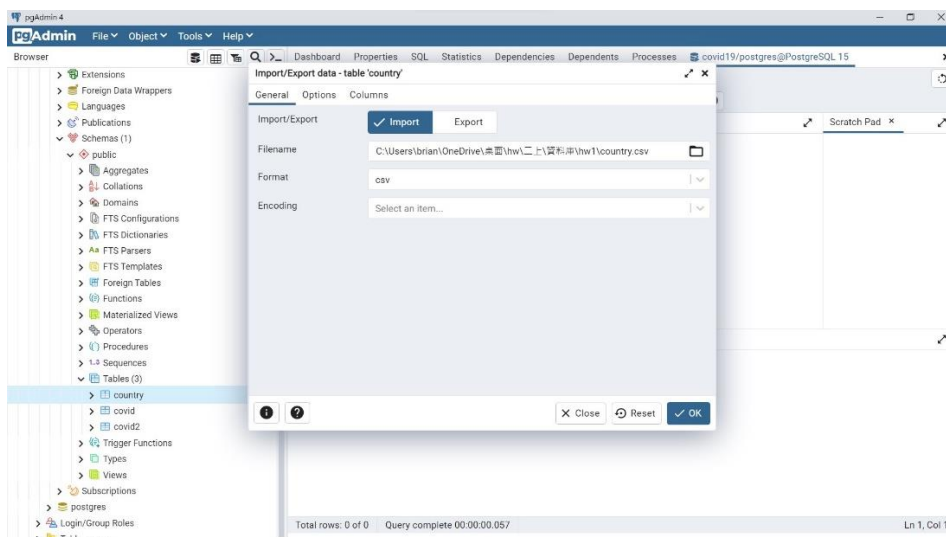
Import the data of 10-11-2022



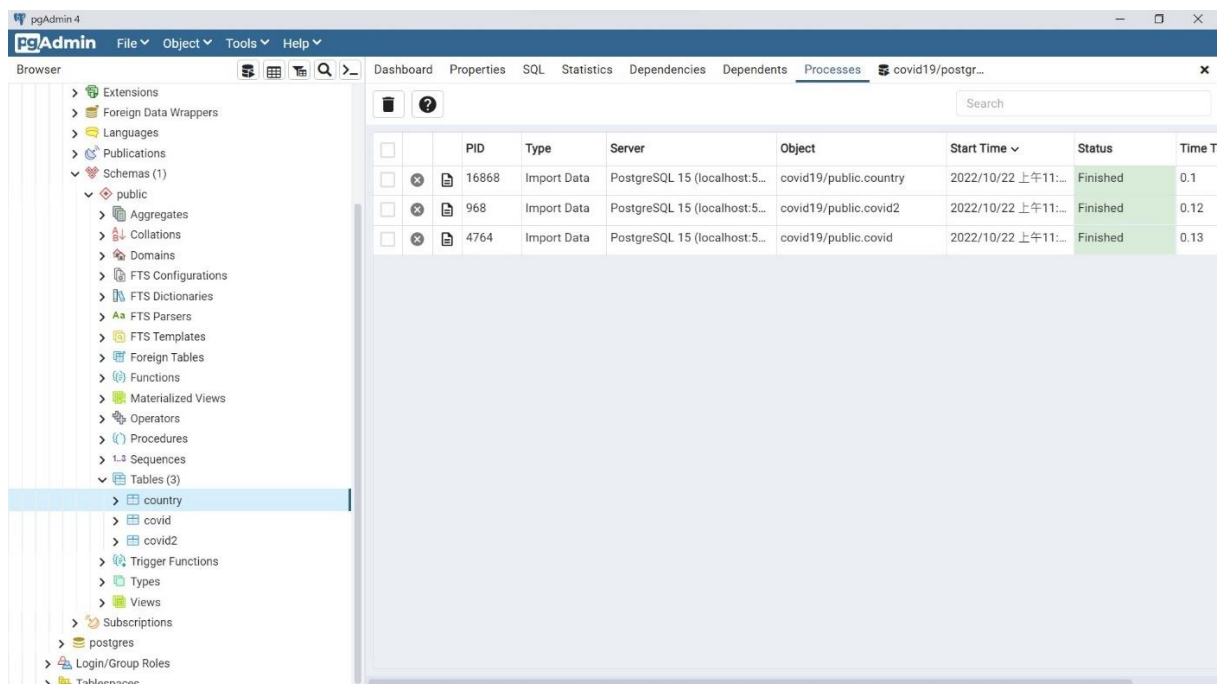
Import the data of 10-01-2022



Import the data of Country code and continents mapping table

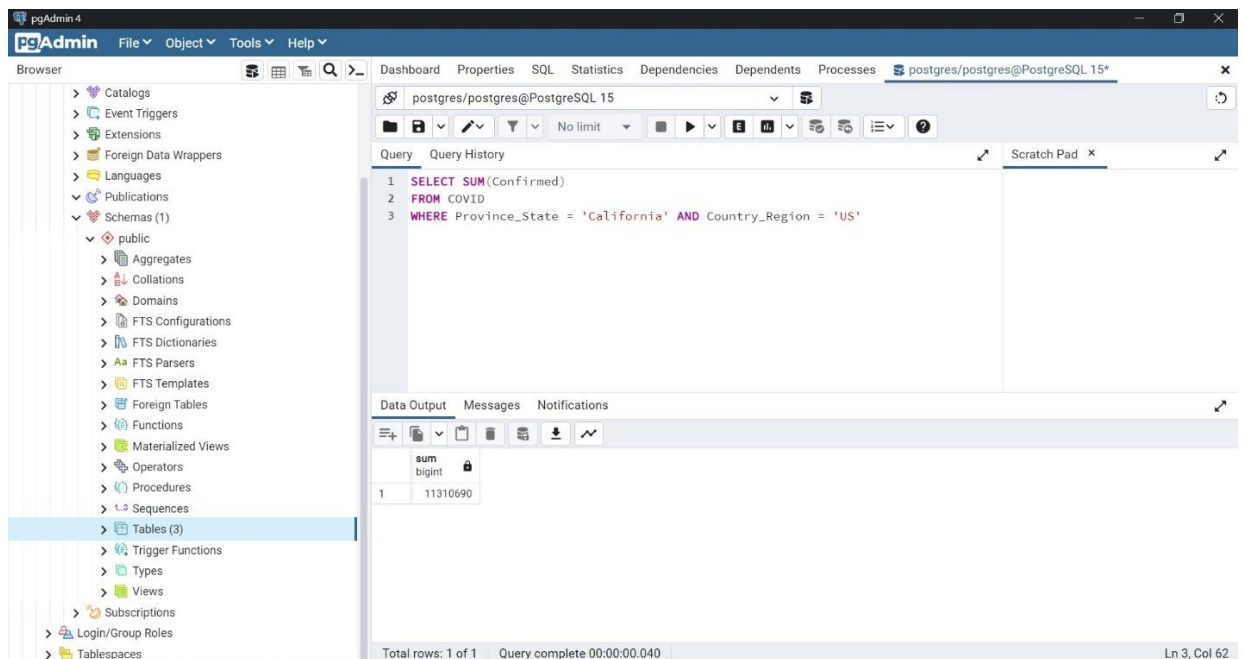


Finish importing three required .csv files into covid19 database



3. The **SQL statements** and **output results** of 4a (10pt). If the SQL statements or output results are not provided, you will not get the points.

Ans:



4. The SQL statements and output results of 4b (10pt)

Ans:

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'covid19' database selected. The main pane shows a SQL query:

```
1 SELECT SUM(Confirmed)
2 FROM COVID2
3 WHERE Province_State = 'California' AND Country_Region = 'US'
```

The 'Data Output' tab shows the result of the query:

sum
11268292

Total rows: 1 of 1 Query complete 00:00:00.042 Ln 3, Col 23

5. The SQL statements and output results of 4c (10pt)

Ans:

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'covid' database selected. The main pane shows a SQL query:

```
1 SELECT SUM(covid.Confirmed) - SUM(covid2.Confirmed) as difference
2 FROM covid join covid2
3 USING (combined_key)
4 WHERE covid.Province_State = 'California' AND covid.Country_Region = 'US'
5 AND covid2.Province_State = 'California' AND covid2.Country_Region = 'US'
```

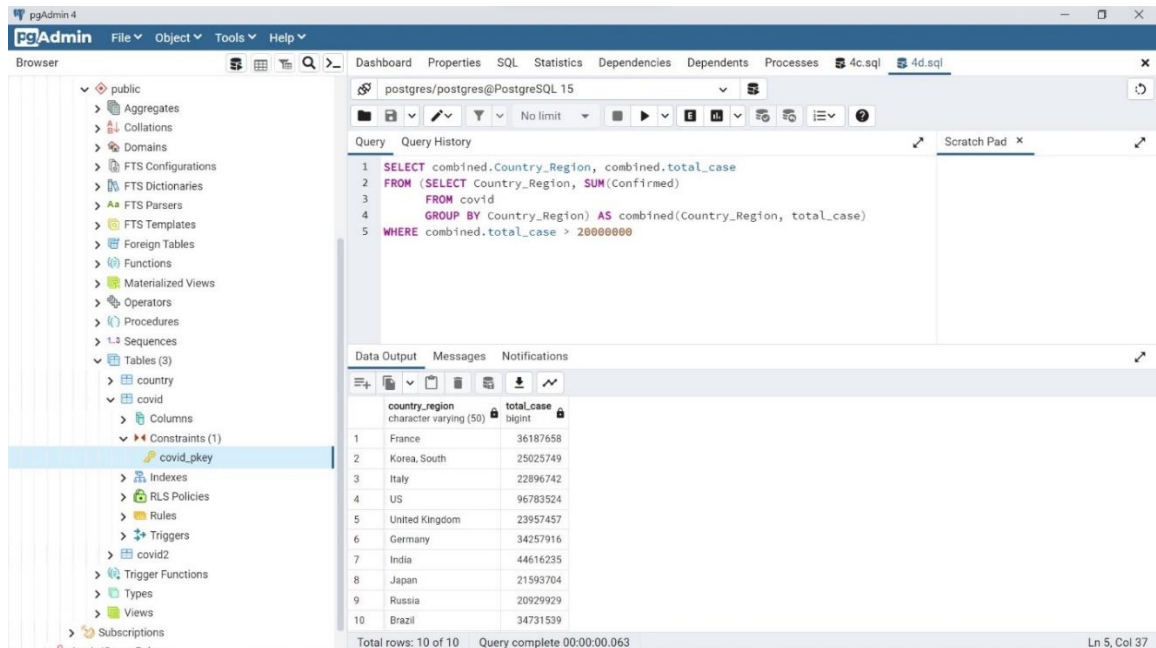
The 'Data Output' tab shows the result of the query:

difference
42398

Total rows: 1 of 1 Query complete 00:00:00.052 Ln 5, Col 75

6. The SQL statements and output results of 4d (10pt)

Ans:



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'covid' table selected under the 'public' schema. The main window shows a SQL query in the 'Query' tab:

```
1 SELECT combined.Country_Region, combined.total_case
2 FROM (SELECT Country_Region, SUM(Confirmed)
3       FROM covid
4       GROUP BY Country_Region) AS combined(Country_Region, total_case)
5 WHERE combined.total_case > 20000000
```

The 'Data Output' tab shows the results of the query:

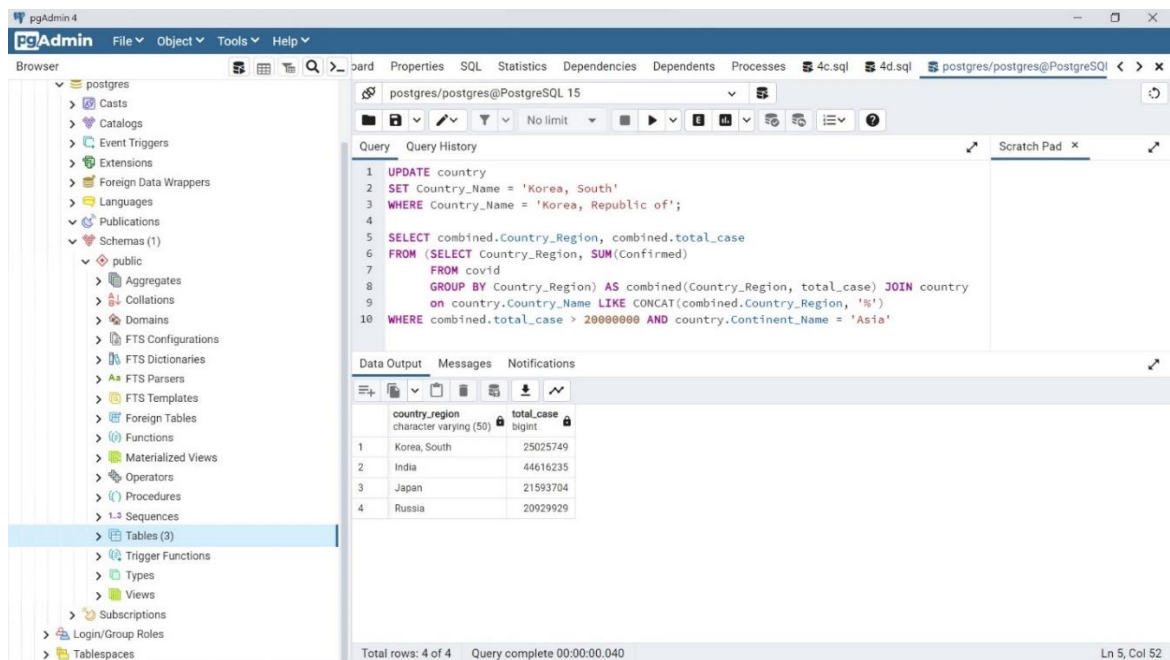
	country_region character varying (50)	total_case bigint
1	France	36187658
2	Korea, South	25025749
3	Italy	22896742
4	US	96783524
5	United Kingdom	23957457
6	Germany	34257916
7	India	44616235
8	Japan	21593704
9	Russia	20929929
10	Brazil	34731539

Total rows: 10 of 10 Query complete 00:00:00.063 Ln 5, Col 37

7. The SQL statements and output results of 4e (10pt)

Ans:

因南韓在 10-11-2022 裡和在 Country code and continents mapping table 裡的名稱不一致，所以我把名稱統一為 Korea, South。



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'country' table selected under the 'public' schema. The main window shows a SQL query in the 'Query' tab:

```
1 UPDATE country
2 SET Country_Name = 'Korea, South'
3 WHERE Country_Name = 'Korea, Republic of';
4
5 SELECT combined.Country_Region, combined.total_case
6 FROM (SELECT Country_Region, SUM(Confirmed)
7       FROM covid
8       GROUP BY Country_Region) AS combined(Country_Region, total_case) JOIN country
9 on country.Country_Name LIKE CONCAT(combined.Country_Region, '%')
10 WHERE combined.total_case > 20000000 AND country.Continent_Name = 'Asia'
```

The 'Data Output' tab shows the results of the query:

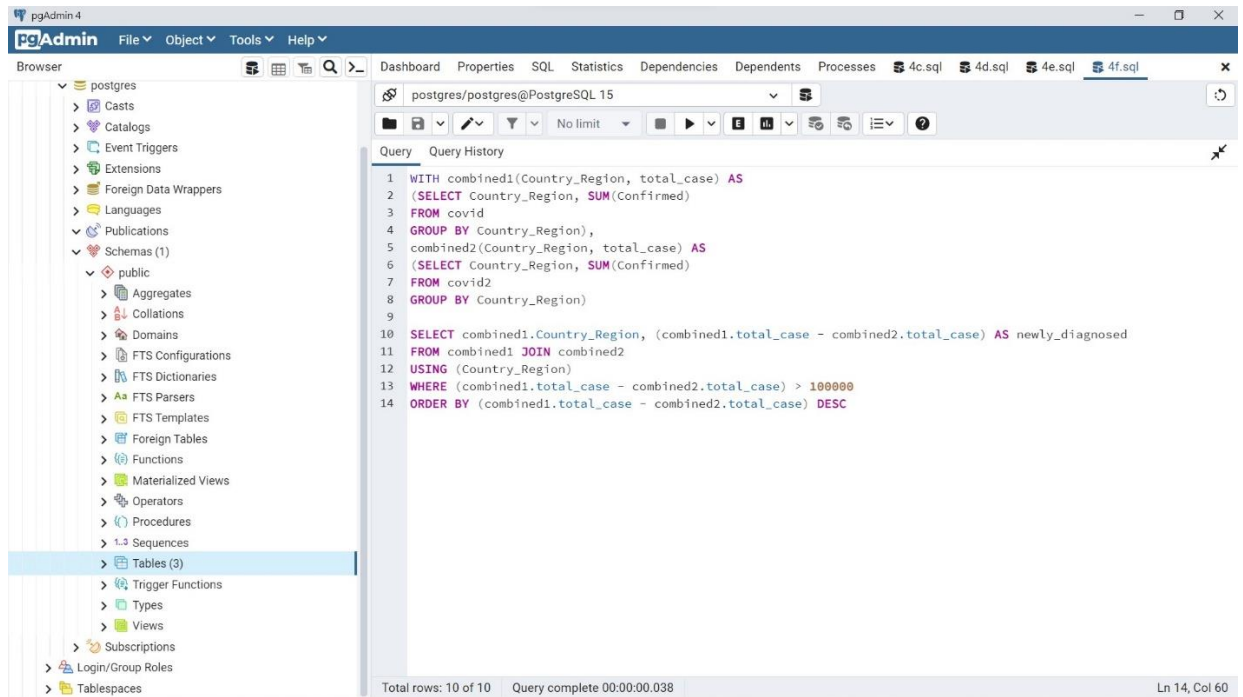
	country_region character varying (50)	total_case bigint
1	Korea, South	25025749
2	India	44616235
3	Japan	21593704
4	Russia	20929929

Total rows: 4 of 4 Query complete 00:00:00.040 Ln 5, Col 52

8. The SQL statements and output results of 4f (10pt)

Ans:

SQL statements

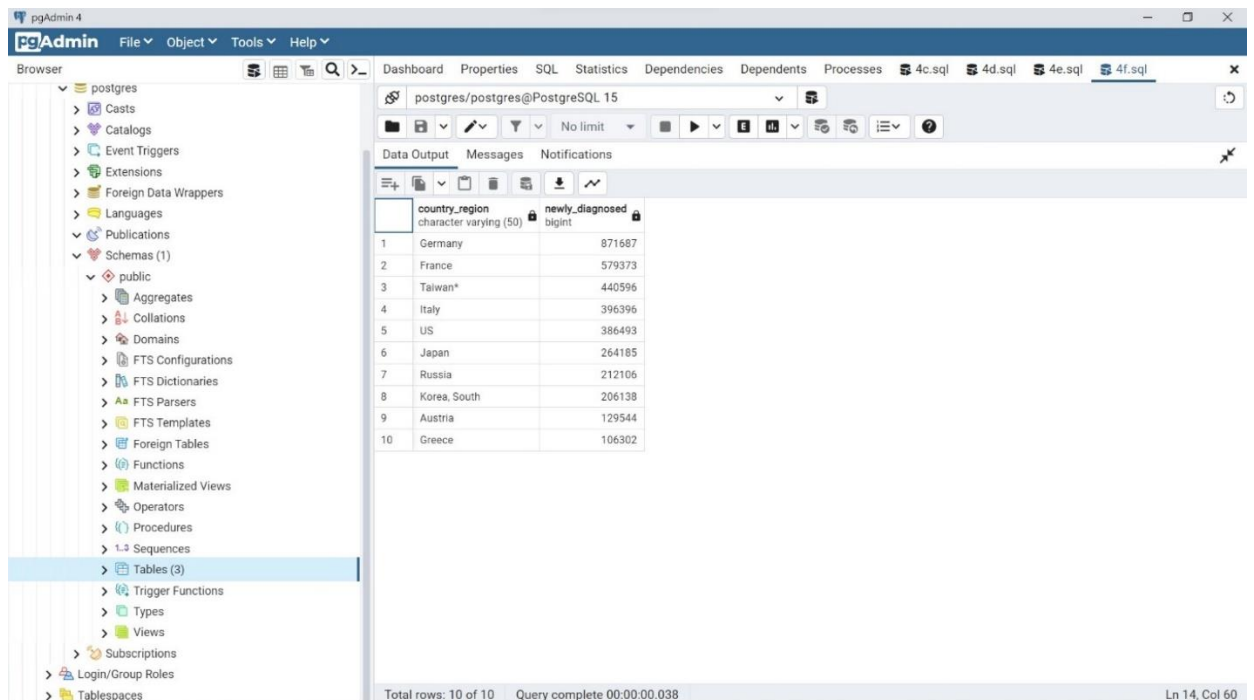


The screenshot shows the pgAdmin 4 interface with a SQL query editor. The query is as follows:

```
1 WITH combined1(Country_Region, total_case) AS
2 (SELECT Country_Region, SUM(Confirmed)
3  FROM covid
4  GROUP BY Country_Region),
5 combined2(Country_Region, total_case) AS
6 (SELECT Country_Region, SUM(Confirmed)
7  FROM covid2
8  GROUP BY Country_Region)
9
10 SELECT combined1.Country_Region, (combined1.total_case - combined2.total_case) AS newly_diagnosed
11 FROM combined1 JOIN combined2
12 USING (Country_Region)
13 WHERE (combined1.total_case - combined2.total_case) > 100000
14 ORDER BY (combined1.total_case - combined2.total_case) DESC
```

The status bar at the bottom indicates: Total rows: 10 of 10, Query complete 00:00:00.038, Ln 14, Col 60.

output results



The screenshot shows the pgAdmin 4 interface with the query results displayed in the Data Output tab. The results are as follows:

	country_region character varying (50)	newly_diagnosed bigint
1	Germany	871687
2	France	579373
3	Taiwan*	440596
4	Italy	396396
5	US	386493
6	Japan	264185
7	Russia	212106
8	Korea, South	206138
9	Austria	129544
10	Greece	106302

The status bar at the bottom indicates: Total rows: 10 of 10, Query complete 00:00:00.038, Ln 14, Col 60.