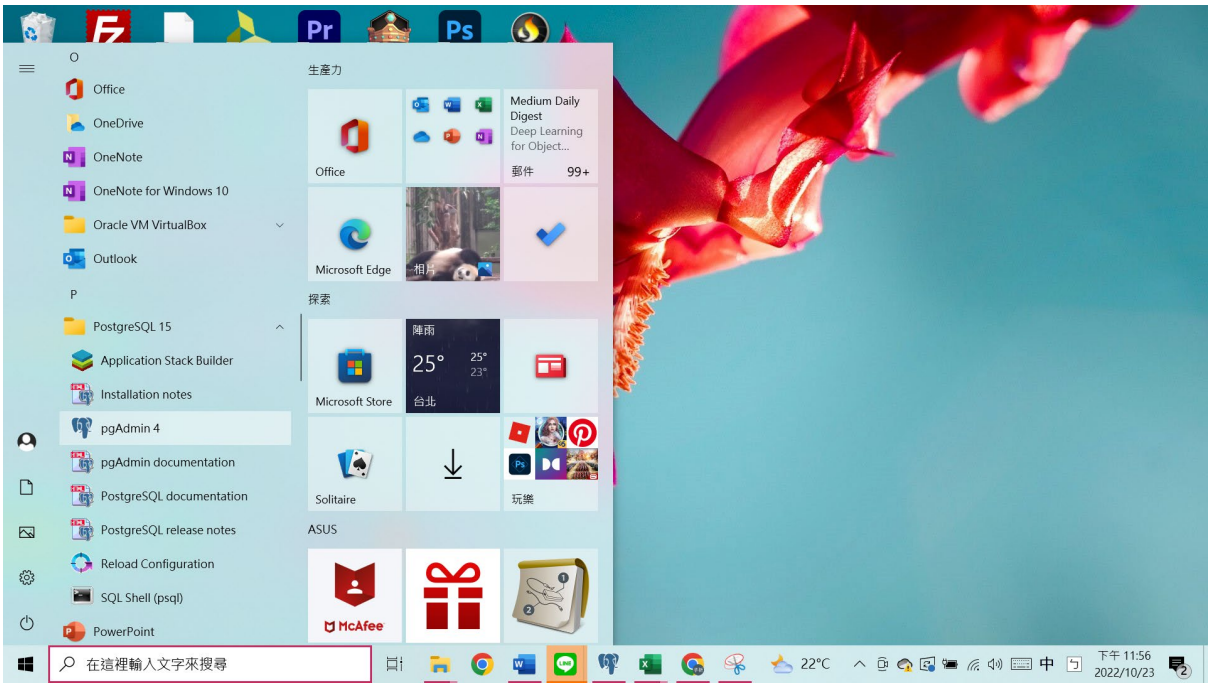
111-1 Database - HW1 0819823陳子祈

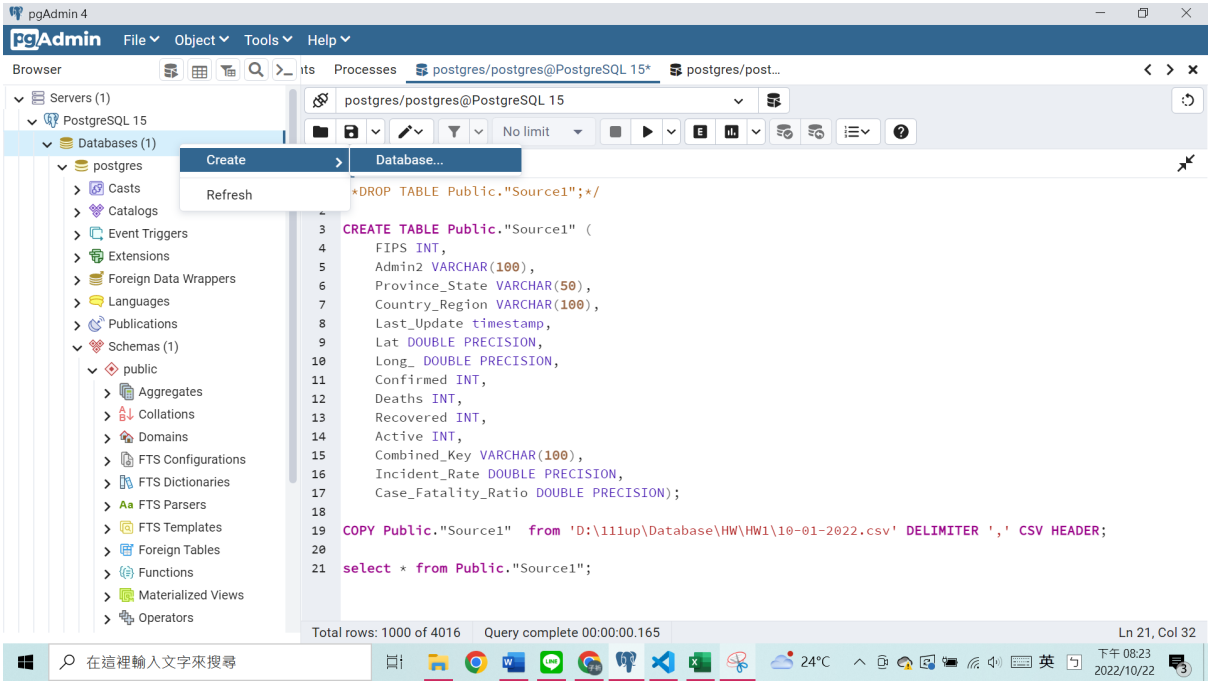
HW1 format:

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

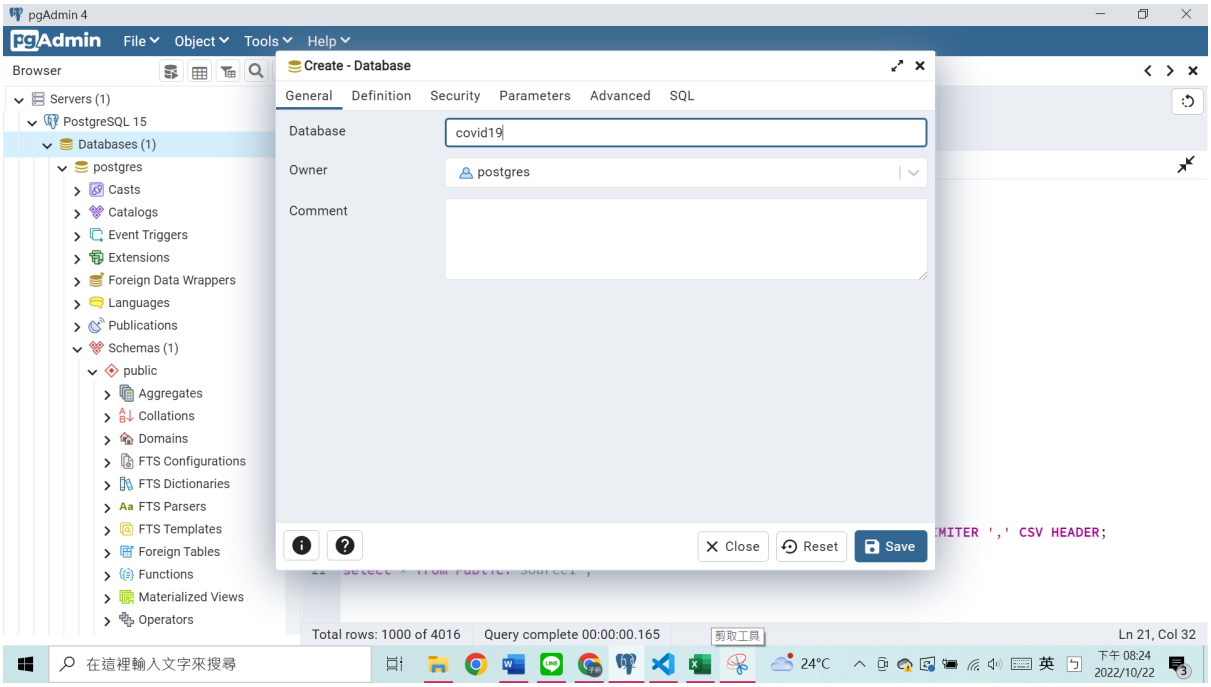
   Ans:



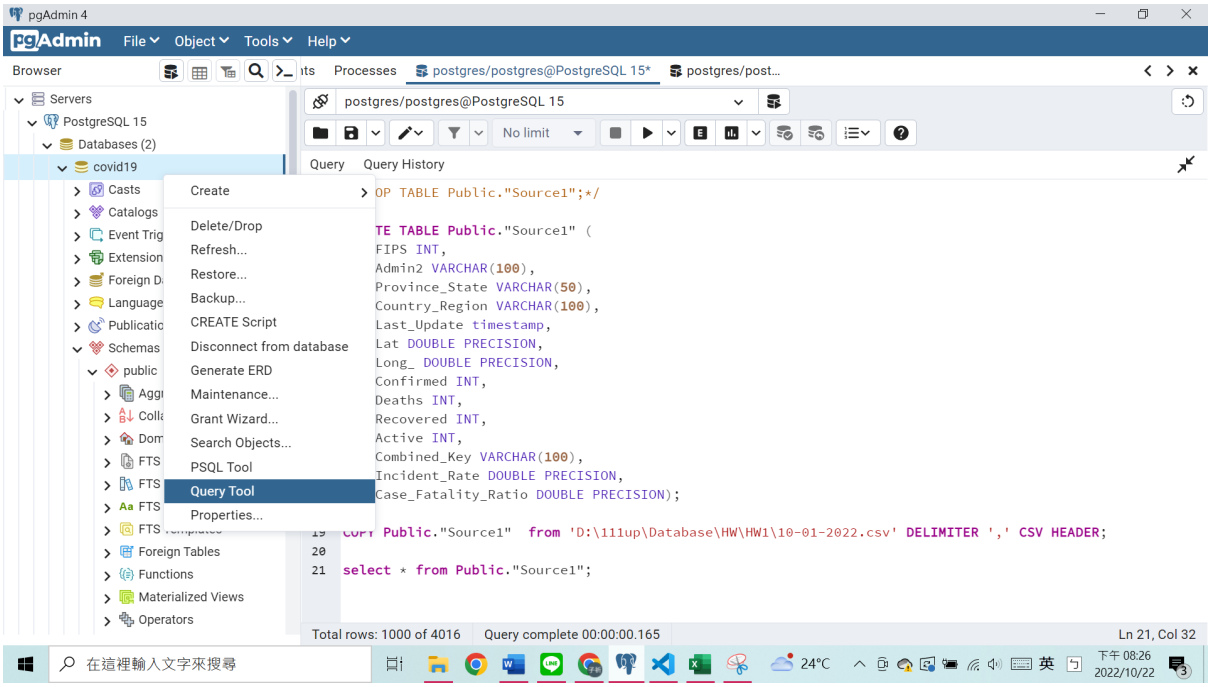
再桌面左下角點擊win符號，滑到P開頭的應用程式，展開PostgreSQL並點選pgAdmin 4啟動sql編輯環境。



在Database上點右鍵，然後按Create->Database。



進入create database的頁面，將此database取名為covid19。



左邊在covid19上點右鍵，進入query tool即可編輯程式碼。以上是如何launch database service及create database的操作。

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:

SQL statements:

Query Query History

```
2
3  CREATE TABLE Source1 (
4      FIPS INT,
5      Admin2 VARCHAR(100),
6      Province_State VARCHAR(100),
7      Country_Region VARCHAR(100),
8      Last_Update timestamp,
9      Lat DOUBLE PRECISION,
10     Long_ DOUBLE PRECISION,
11     Confirmed INT,
12     Deaths INT,
13     Recovered INT,
14     Active INT,
15     Combined_Key VARCHAR(100),
16     Incident_Rate DOUBLE PRECISION,
17     Case_Fatality_Ratio DOUBLE PRECISION);
18
```

Query Query History

```
18
19 CREATE TABLE Source2 (
20     FIPS INT,
21     Admin2 VARCHAR(100),
22     Province_State VARCHAR(100),
23     Country_Region VARCHAR(100),
24     Last_Update timestamp,
25     Lat DOUBLE PRECISION,
26     Long_ DOUBLE PRECISION,
27     Confirmed INT,
28     Deaths INT,
29     Recovered INT,
30     Active INT,
31     Combined_Key VARCHAR(100),
32     Incident_Rate DOUBLE PRECISION,
33     Case_Fatality_Ratio DOUBLE PRECISION);
34
```

```
35 CREATE TABLE Source3 (
36     Continent_Name VARCHAR(100),
37     Continent_Code VARCHAR(100),
38     Country_Name VARCHAR(100),
39     Two_Letter_Country_Code VARCHAR(100),
40     Three_Letter_Country_Code VARCHAR(100),
41     Country_Number INT
42     );
43
```

explanations:

我觀察csv檔案時發現所有的columns的data type有字串、整數、浮點數及時間這幾種data type，所以我在建立Source1、Source2、Source3的table時就把字串統一設定VARCHAR(100)、整數設定為INT、浮點數設定為DOUBLE PRECISION、時間設定為timestamp。

SQL statements:

```
44  COPY Source1 from 'D:\111up\Database\HW\HW1\10-01-2022.csv' DELIMITER ',' CSV HEADER;
45
46  COPY Source2  from 'D:\111up\Database\HW\HW1\10-11-2022.csv' DELIMITER ',' CSV HEADER;
47
48  COPY Source3  from 'D:\111up\Database\HW\HW1\country-and-continent-codes-list-csv.csv'
49  DELIMITER ',' CSV HEADER;
50
51  alter table Source2
52  rename column Country_Region to Country_Name;
53
54  alter table Source1
55  rename column Country_Region to Country_Name;
56
```

explanations:

這些是import 三個csv檔案的指令。另外，因為problem 4e要求Source2要根據Country_name來natural join Source3，所以我就把Source1及Source2的Country_Region改名為Country_name。

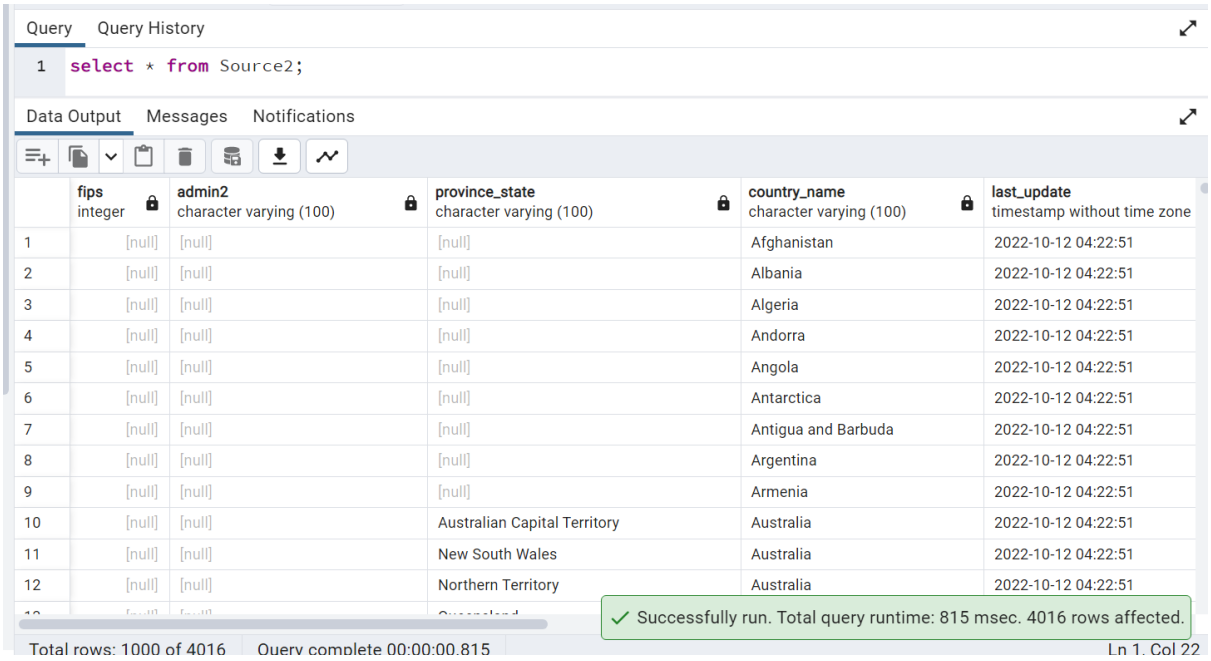
SQL statements:

```
Query  Query History
1  select * from Source1;
```

Data Output | Messages | Notifications

	fips integer	admin2 character varying (100)	province_state character varying (100)	country_name character varying (100)	last_update timestamp without time zone
1	[null]	[null]	[null]	Afghanistan	2022-10-02 04:21:23
2	[null]	[null]	[null]	Albania	2022-10-02 04:21:23
3	[null]	[null]	[null]	Algeria	2022-10-02 04:21:23
4	[null]	[null]	[null]	Andorra	2022-10-02 04:21:23
5	[null]	[null]	[null]	Angola	2022-10-02 04:21:23
6	[null]	[null]	[null]	Antarctica	2022-10-02 04:21:23
7	[null]	[null]	[null]	Antigua and Barbuda	2022-10-02 04:21:23
8	[null]	[null]	[null]	Argentina	2022-10-02 04:21:23
9	[null]	[null]	[null]	Armenia	2022-10-02 04:21:23
10	[null]	[null]	Australian Capital Territory	Australia	2022-10-02 04:21:23
11	[null]	[null]	New South Wales	Australia	2022-10-02 04:21:23
12	[null]	[null]	Northern Territory	Australia	2022-10-02 04:21:23

Successfully run. Total query runtime: 149 msec. 4016 rows affected.



Query Query History

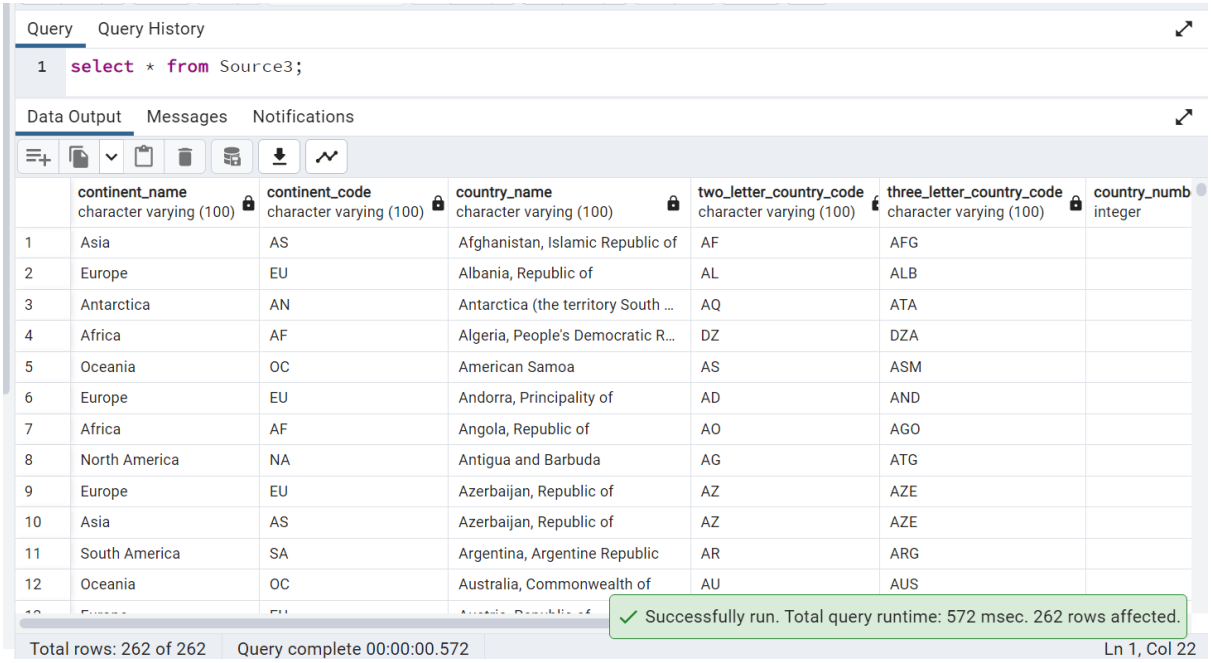
```
1  select * from Source2;
```

Data Output Messages Notifications

	fips integer	admin2 character varying (100)	province_state character varying (100)	country_name character varying (100)	last_update timestamp without time zone
1	[null]	[null]	[null]	Afghanistan	2022-10-12 04:22:51
2	[null]	[null]	[null]	Albania	2022-10-12 04:22:51
3	[null]	[null]	[null]	Algeria	2022-10-12 04:22:51
4	[null]	[null]	[null]	Andorra	2022-10-12 04:22:51
5	[null]	[null]	[null]	Angola	2022-10-12 04:22:51
6	[null]	[null]	[null]	Antarctica	2022-10-12 04:22:51
7	[null]	[null]	[null]	Antigua and Barbuda	2022-10-12 04:22:51
8	[null]	[null]	[null]	Argentina	2022-10-12 04:22:51
9	[null]	[null]	[null]	Armenia	2022-10-12 04:22:51
10	[null]	[null]	Australian Capital Territory	Australia	2022-10-12 04:22:51
11	[null]	[null]	New South Wales	Australia	2022-10-12 04:22:51
12	[null]	[null]	Northern Territory	Australia	2022-10-12 04:22:51

Successfully run. Total query runtime: 815 msec. 4016 rows affected.

Total rows: 1000 of 4016 Query complete 00:00:00.815 Ln 1, Col 22



Query Query History

```
1  select * from Source3;
```

Data Output Messages Notifications

	continent_name character varying (100)	continent_code character varying (100)	country_name character varying (100)	two_letter_country_code character varying (100)	three_letter_country_code character varying (100)	country_numb integer
1	Asia	AS	Afghanistan, Islamic Republic of	AF	AFG	
2	Europe	EU	Albania, Republic of	AL	ALB	
3	Antarctica	AN	Antarctica (the territory South ...	AQ	ATA	
4	Africa	AF	Algeria, People's Democratic R...	DZ	DZA	
5	Oceania	OC	American Samoa	AS	ASM	
6	Europe	EU	Andorra, Principality of	AD	AND	
7	Africa	AF	Angola, Republic of	AO	AGO	
8	North America	NA	Antigua and Barbuda	AG	ATG	
9	Europe	EU	Azerbaijan, Republic of	AZ	AZE	
10	Asia	AS	Azerbaijan, Republic of	AZ	AZE	
11	South America	SA	Argentina, Argentine Republic	AR	ARG	
12	Oceania	OC	Australia, Commonwealth of	AU	AUS	

Successfully run. Total query runtime: 572 msec. 262 rows affected.

Total rows: 262 of 262 Query complete 00:00:00.572 Ln 1, Col 22

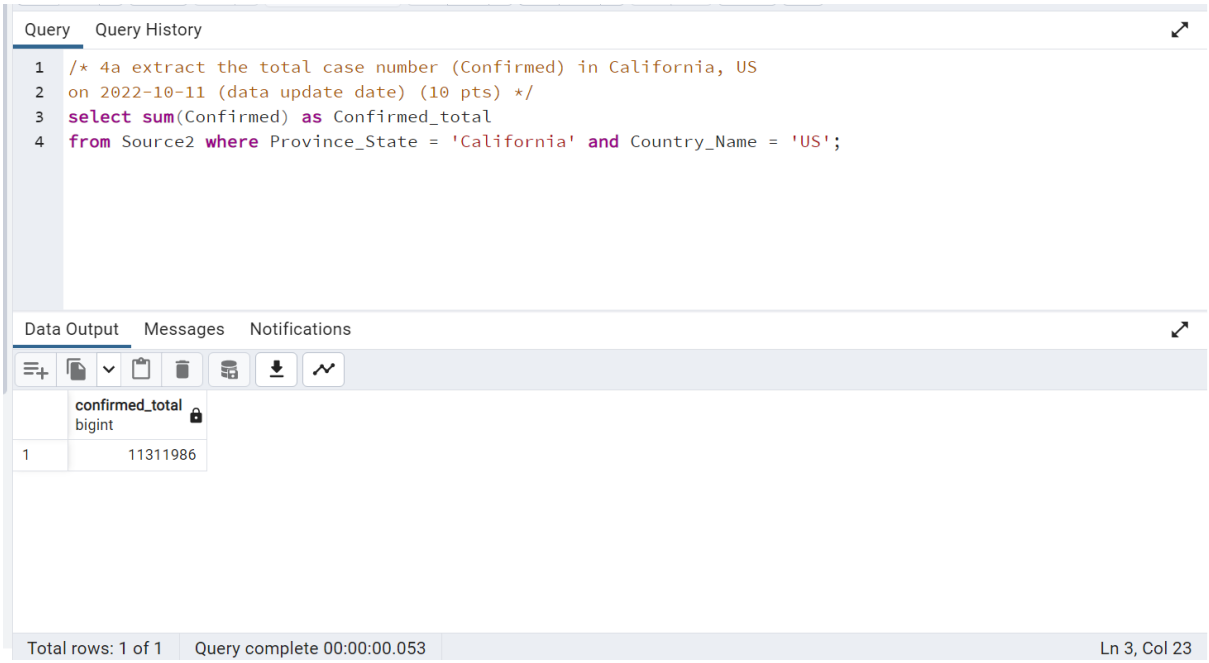
explanations:

這些是輸出Source1、Source2、Source3 table的指令及結果的截圖。

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:

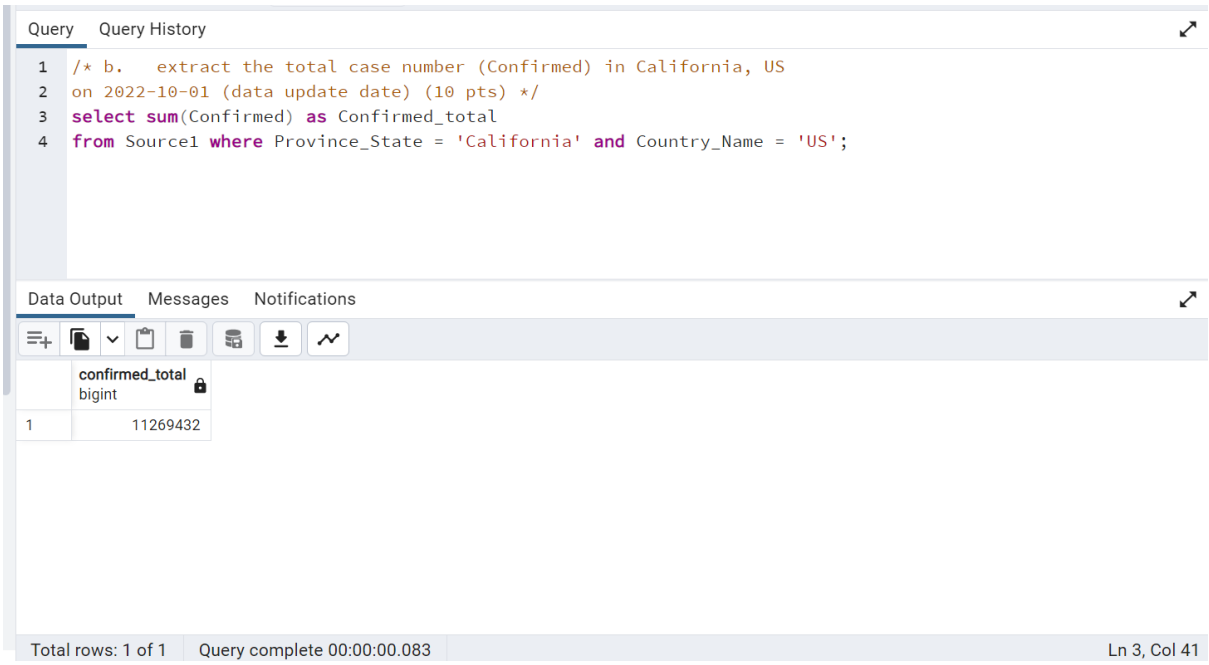
   SQL statements and output results:



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

   Ans:

   SQL statements and output results:



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

   Ans:

   SQL statements and output results:

```
Query  Query History
8  select sum (coalesce(Source2.Confirmed, 0) - coalesce(Source1.Confirmed, 0) ) as Confirmed_new_total
9         from (select Province_State, Country_Name,
10                    sum(Confirmed) Confirmed
11                    from Source2
12                    group by Province_State, Country_Name) Source2
13              full join (select Province_State, Country_Name,
14                                sum(Confirmed) Confirmed
15                                from Source1
16                                group by Province_State, Country_Name) Source1
17                       on Source1.Province_State = Source2.Province_State
18                       and Source1.Country_Name = Source2.Country_Name
19        where Source1.Province_State = 'California'
20        and Source2.Province_State = 'California'
21        and Source1.Country_Name = 'US'
22        and Source2.Country_Name = 'US';
23
```

Data Output Messages Notifications

	confirmed_new_total numeric
1	42554

Total rows: 1 of 1 Query complete 00:00:00.085 Ln 22, Col 37

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

Ans:

SQL statements and output results:

```
Query  Query History
2  with more than 20,000,000 total COVID-19 cases on 2022-10-11 (data update date) (10 pts) */
3  select Country_Name, sum(Confirmed) as Confirmed_total
4  from Source2
5  group by Country_Name
6  having sum(Confirmed) > 20000000;
```

Data Output Messages Notifications

	country_name character varying (100)	confirmed_total bigint
1	France	36187658
2	Korea, South	25025749
3	Italy	22896742
4	US	96775983
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 00:00:00.136 Ln 2, Col 90

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

Ans:

SQL statements and output results:

Query Query History

```sql
/* 4e
extract the country names (return Country_Region column)
and total confirmed COVID cases (return Confirmed column)
with more than 20,000,000 total COVID-19 cases
on 2022-10-11 (data update date).
Try to join the Country code and continents mapping table,
and return only the data from countries in Asia. (10 pts)
*/

update Source3
set Country_Name = Country_Name || ',';

update Source3
set Country_Name = LEFT(Country_Name, strpos(Country_Name, ',')-1 );

update Source3
set Country_Name = 'Russia' where Country_Name = 'Russian Federation';
update Source3
set Country_Name = 'Turkey'  where Three_Letter_Country_Code = 'TUR';
update Source3
set Country_Name = 'United Kingdom'  where Three_Letter_Country_Code = 'GBR';
update Source3
set Country_Name = 'US'  where Three_Letter_Country_Code = 'USA';
update Source3
set Country_Name = 'Korea, South' where Three_Letter_Country_Code = 'PRK';
update Source3
set Country_Name = 'Korea, North' where Three_Letter_Country_Code = 'KOR';

select Source2.Country_Name, sum(Source2.Confirmed) as Confirmed_total
from Source2 natural join Source3 where Source3.Continent_Name = 'Asia'
group by Source2.Country_Name
having sum(Source2.Confirmed) > 20000000;
```

Data Output Messages Notifications

	country_name character varying (100)	confirmed_total bigint
1	Korea, South	25025749
2	India	44616235
3	Japan	21593704
4	Russia	20929929

Total rows: 4 of 4 Query complete 00:00:00.089 Ln 27, Col 75

觀察Source2及Source3這兩個table可以發現，Source2大多不會有’,’ ，且Source2及Source3同一個Country_name有些名稱不一樣，比如在Source3的北韓叫做Korea, Democratic People's Republic of，但是在Source2的北韓叫做Korea, North。所以我的方法是先將Source3.Country_name後面全加上’,’ ，再消除Source3出現第一個’,’ 右邊的string，再透過Three_Letter_Country_Code對Source3.Country_name的個案進行改名。

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

Ans:

SQL statements:

Query | Query History

```
on 2022-10-11 (compare with 2022-10-01).
In descending order of newly diagnosed case numbers. (10 pts)
*/
select coalesce(Source1.Country_Name, Source2.Country_Name) Country_Name,
       coalesce(Source2.Confirmed, 0) - coalesce(Source1.Confirmed, 0) Confirmed_new_total
       from (select Country_Name,
                    sum(Confirmed) Confirmed
                    from Source2
                    group by Country_Name) Source2
            full join (select Country_Name,
                              sum(Confirmed) Confirmed
                              from Source1
                              group by Country_Name) Source1
                      on Source1.Country_Name = Source2.Country_Name
       where coalesce(Source2.Confirmed, 0) - coalesce(Source1.Confirmed, 0) > 100000
       order by coalesce(Source2.Confirmed, 0) - coalesce(Source1.Confirmed, 0) desc;
```

Data Output | Messages | Notifications

	country_name character varying (100)	confirmed_new_total bigint
1	Germany	871687
2	France	579373

Total rows: 10 of 10 | Query complete 00:00:00.119 | Ln 16, Col 35

output results:

Query | Query History

```
on 2022-10-11 (compare with 2022-10-01).
In descending order of newly diagnosed case numbers. (10 pts)
*/
select coalesce(Source1.Country_Name, Source2.Country_Name) Country_Name,
       coalesce(Source2.Confirmed, 0) - coalesce(Source1.Confirmed, 0) Confirmed_new_total
```

Data Output | Messages | Notifications

	country_name character varying (100)	confirmed_new_total bigint
1	Germany	871687
2	France	579373
3	Taiwan*	440596
4	Italy	396396
5	US	389029
6	Japan	264185
7	Russia	212106
8	Korea, South	206138
9	Austria	129544
10	Greece	106302

Total rows: 10 of 10 | Query complete 00:00:00.119 | Ln 16, Col 35