

1.

```
postgres=# CREATE DATABASE covid19;  
CREATE DATABASE
```

2.

```
covid19=# create table CovidInfo1011(  
covid19(# fips int,  
covid19(# admin2 varchar(50),  
covid19(# province_state varchar(50),  
covid19(# country_region varchar(50),  
covid19(# last_update timestamp,  
covid19(# lat float(12),  
covid19(# long float(12),  
covid19(# confirmed int,  
covid19(# deaths int,  
covid19(# recovered varchar(20),  
covid19(# active varchar(20),  
covid19(# combined_key varchar(60),  
covid19(# incident_rate double precision,  
covid19(# case_fatality_ratio double precision,  
covid19(# primary key (combined_key)  
covid19(# );  
CREATE TABLE  
covid19=# create table CovidInfo1001(  
covid19(# fips int,  
covid19(# admin2 varchar(50),  
covid19(# province_state varchar(50),  
covid19(# country_region varchar(50),  
covid19(# last_update timestamp,  
covid19(# lat float(12),  
covid19(# long float(12),  
covid19(# confirmed int,  
covid19(# deaths int,  
covid19(# recovered varchar(20),  
covid19(# active varchar(20),  
covid19(# combined_key varchar(60),  
covid19(# incident_rate double precision,  
covid19(# case_fatality_ratio double precision,  
covid19(# primary key (combined_key)  
covid19(# );  
CREATE TABLE  
covid19=# create table country_code(  
covid19(# continent_name varchar(20),  
covid19(# continent_code char(2),  
covid19(# country_name varchar(55),  
covid19(# two_letter_country_code char(2),  
covid19(# three_letter_country_code char(3),  
covid19(# country_number numeric(3,0),  
covid19(# primary key(continent_name,country_name)  
covid19(# );  
CREATE TABLE
```

匯入 csv 檔：用 pgAdmin 設好 binary path 後，右鍵按 table，選 import file，然後選去掉第一排的載下來 csv 檔。

3.

Query

Query History

1

`select sum(confirmed)`

2

`from covidinfo1011`

3

`where combined_key like '%California, US';`

Data output

Messages

Notifications

sum
bigint

1

11310690

4.

Query

Query History

1

`select sum(confirmed)`

2

`from covidinfo1001`

3

`where combined_key like '%California, US';`

Data output

Messages

Notifications

sum
bigint

1

11268292

5.

Query
Query History

```

1 select sum(covidinfo1011.confirmed - covidinfo1001.confirmed)
2 from covidinfo1011,covidinfo1001
3 where covidinfo1001.combined_key like '%California, US'
4 and covidinfo1011.combined_key=covidinfo1001.combined_key;

```

Data output
Messages
Notifications

+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	sum bigint	🔒
1	42398	

6.

Query
Query History

```

1 select country_region,sum(confirmed)
2 from covidinfo1011
3 group by country_region
4 having sum(confirmed)>20000000;

```

Data output
Messages
Notifications

+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	country_region character varying (50)	sum bigint	🔒	🔒
1	France	36187658		
2	Korea, South	25025749		
3	Italy	22896742		
4	US	96783014		
5	United Kingdom	23957457		
6	Germany	34257916		
7	India	44616235		
8	Japan	21593704		
9	Russia	20929929		
10	Brazil	34731539		

7.

```
covid19=# update country_code
covid19=# set country_name='Korea, South'
covid19=# where country_name = 'Korea, Republic of';
```

//改名子

Query Query History

```
1 select newT.country_region, newT.total
2 from(select country_region,sum(confirmed) as total
3      from covidinfo1011
4      group by country_region
5      having sum(confirmed)>20000000) as newT, country_code
6 where country_code.continent_code='AS' and newT.country_region = country_code.country_name;
```

Data output Messages Notifications

	country_region character varying (50)	total bigint
1	Japan	21593704
2	Korea, South	25025749

8.

Query Query History

```
1 select newT1011.country_region,newT1011.total-newT1001.total as newcase
2 from(select country_region,sum(confirmed) as total
3      from covidinfo1011
4      group by country_region
5      having sum(confirmed)>20000000) as newT1011,
6      (select country_region,sum(confirmed) as total
7      from covidinfo1001
8      group by country_region
9      having sum(confirmed)>20000000) as newT1001
10 where newT1011.total-newT1001.total>100000 and newT1011.country_region=newT1001.country_region
11 order by newT1011.total-newT1001.total desc;
```

Data output Messages Notifications

	country_region character varying (50)	newcase bigint
1	Germany	871687
2	France	579373
3	Italy	396396
4	US	385970
5	Japan	264185
6	Russia	212106
7	Korea, South	206138