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

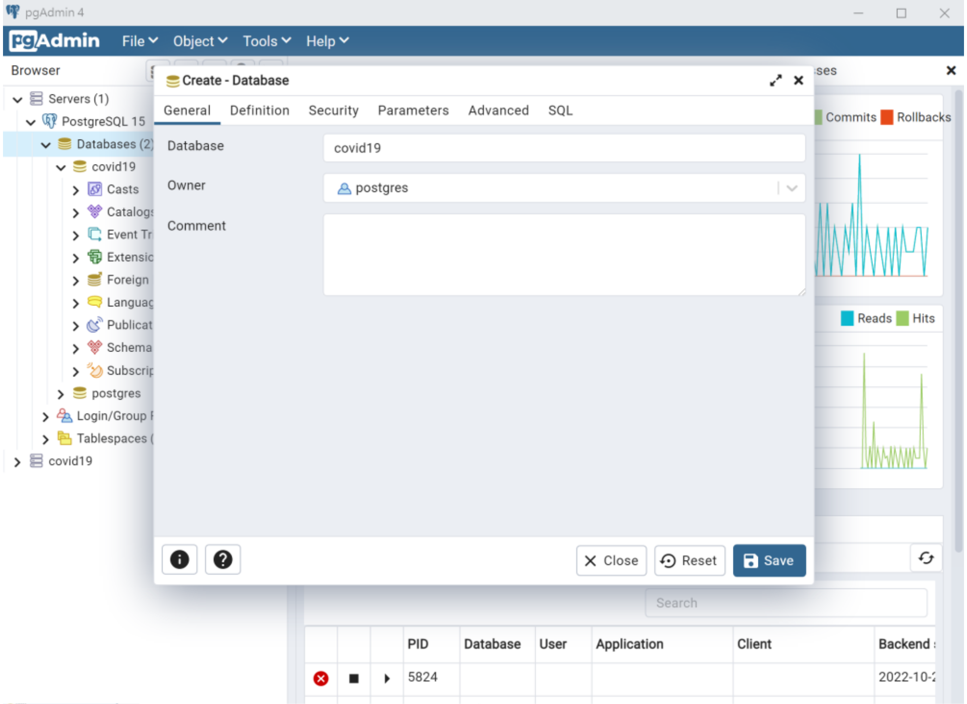
   Ans:

   Step 1: Download PostgreSQL.

   Step 2: Open pgAdmin 4.

   Step 3: Open “Source", “PostgreSQL 15”, then right click “Databases”.

   Step 4: Inout the name “covid19”, then click “Save”.



Create - Database

General

Database: covid19

Owner: postgres

Comment:

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:

   Step 1: Create tables for the three .csv files.

      Here I name the file for 10/2 to be octsec, which means October Second.

      Then, octtwe for 10/12, which means October Twelfth.

      Then, source3 for the third file, because I am bad at naming.

      Below are the screenshots for the three tables.

      The data types are selected by observing the format in the csv files.

```sql
CREATE TABLE octsec(
    FIPS int,
    Admin2 varchar(50),
    Province_State varchar(50),
    Country_Region varchar(50),
    Last_Update timestamp,
    Lat float(53),
    Long_ float(53),
    Confirmed int,
    Deaths int,
    Recovered int,
    Active int,
    Combined_Key varchar(100),
    Incident_Rate float(53),
    Case_Fatality_Ratio float(53),
    Primary Key (Combined_Key)
);
```

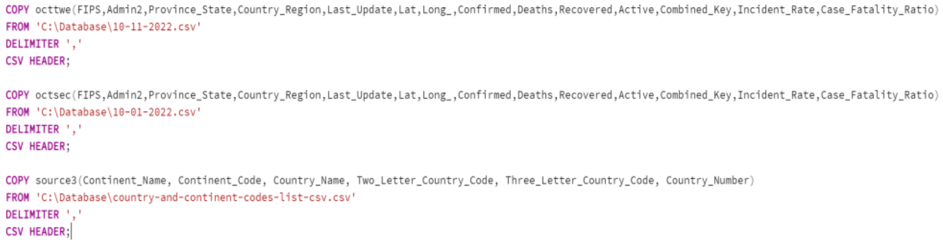
```sql
CREATE TABLE octtwe(
    FIPS int,
    Admin2 varchar(50),
    Province_State varchar(50),
    Country_Region varchar(50),
    Last_Update timestamp,
    Lat float(53),
    Long_ float(53),
    Confirmed int,
    Deaths int,
    Recovered int,
    Active int,
    Combined_Key varchar(100),
    Incident_Rate float(53),
    Case_Fatality_Ratio float(53),
    Primary Key(Combined_Key)
);
```

```sql
CREATE TABLE source3(
    Continent_Name varchar(50),
    Continent_Code varchar(2),
    Country_Name varchar(100),
    Two_Letter_Country_Code varchar(2),
    Three_Letter_Country_Code varchar(3),
    Country_Number int,
    Primary Key (Continent_Code, Country_Name)
);
```

Step 2: Import the .csv files to the database with COPY statements.

Here is the screenshots of COPY statements for the three tables.

This is the way given by the professor in the “Hint” link.



```sql
COPY octtwe(FIPS,Admin2,Province_State,Country_Region,Last_Update,Lat,Long_,Confirmed,Deaths,Recovered,Active,Combined_Key,Incident_Rate,Case_Fatality_Ratio)
FROM 'C:\Database\10-11-2022.csv'
DELIMITER ','
CSV HEADER;

COPY octsec(FIPS,Admin2,Province_State,Country_Region,Last_Update,Lat,Long_,Confirmed,Deaths,Recovered,Active,Combined_Key,Incident_Rate,Case_Fatality_Ratio)
FROM 'C:\Database\10-01-2022.csv'
DELIMITER ','
CSV HEADER;

COPY source3(Continent_Name, Continent_Code, Country_Name, Two_Letter_Country_Code, Three_Letter_Country_Code, Country_Number)
FROM 'C:\Database\country-and-continent-codes-list-csv.csv'
DELIMITER ','
CSV HEADER;
```

Step 3: Click the “Run” button above, then it’s done!

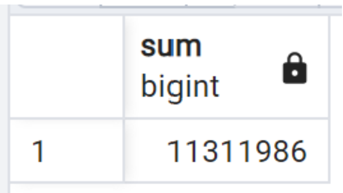
3. The **SQL statements** and **output results** of 4a (10pt). If the SQL statements or output results are not provided, you will not get the points.

   Ans:

   SQL Statement:

   ```
           SELECT SUM(Confirmed)
           FROM octtwe
           where Province_State = 'California' and Country_Region= 'US';
   ```

   Result:



	sum bigint 🔒
1	11311986

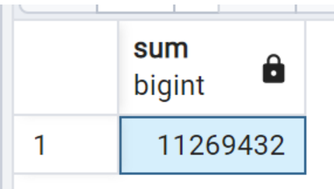
4. The SQL statements and output results of 4b (10pt)

   Ans:

   SQL Statement:

   ```
           SELECT SUM(Confirmed)
           FROM octsec
           where Province_State = 'California' and Country_Region= 'US';
   ```

   Result:



	sum bigint 🔒
1	11269432

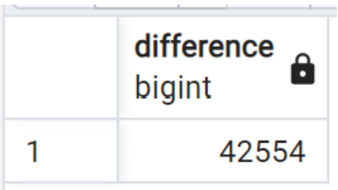
5. The SQL statements and output results of 4c (10pt)

   Ans:

   SQL Statement:

   ```
   SELECT sum(octtwe.confirmed-octsec.confirmed) as difference
           FROM octtwe,octsec
           WHERE octtwe.Combined_Key=octsec.Combined_Key and
     octtwe.Province_State='California' and octsec.Country_Region='US';
   ```

   Result:



	difference bigint 🔒
1	42554

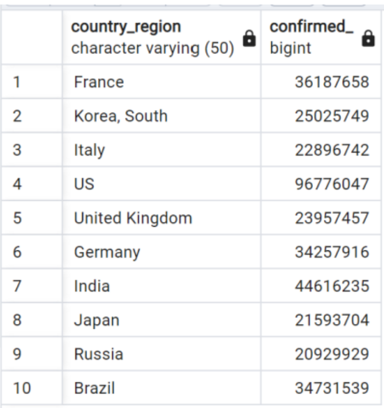
6. The SQL statements and output results of 4d (10pt)

   Ans:

   SQL Statement:

```
SELECT Country_Region, SUM(Confirmed) AS Confirmed_
        FROM octtwe
        GROUP BY Country_Region
        HAVING SUM(Confirmed)>20000000
```

   Result:



	country_region character varying (50) 🔒	confirmed_ bigint 🔒
1	France	36187658
2	Korea, South	25025749
3	Italy	22896742
4	US	96776047
5	United Kingdom	23957457
6	Germany	34257916
7	India	44616235
8	Japan	21593704
9	Russia	20929929
10	Brazil	34731539

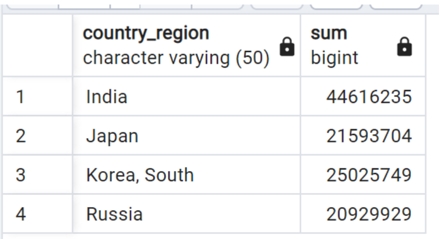
7. The SQL statements and output results of 4e (10pt)

   Ans:

   SQL Statement:

```
SELECT Country_Region, SUM(Confirmed)
        FROM octtwe join source3 on
   octtwe.Country_Region=source3.Country_Name and
   Source3.Continent_Code='AS'
        GROUP BY Country_Region
        HAVING SUM(Confirmed)>20000000
```

   Result:



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

   (In this part, I changed the names of the ten countries shown in the last part in the third .csv file in order to keep the names the same and get correct answers.)

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

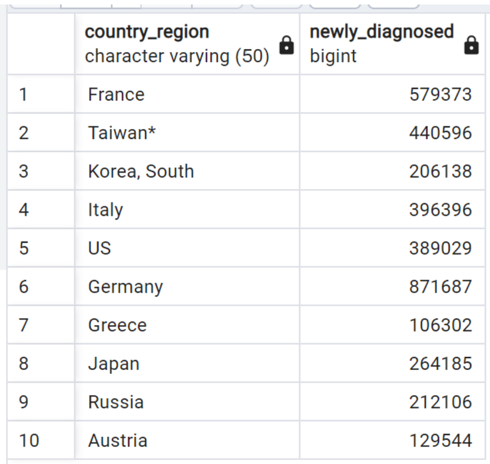
   Ans:

   SQL Statement:

```
SELECT Country_Region, SUM(Confirmed) as Newly_Diagnosed
```

```
        FROM (SELECT octtwe.Country_Region as Country_Region,
 octtwe.Confirmed-octsec.Confirmed as Confirmed
FROM octtwe, octsec
WHERE octtwe.Combined_Key=octsec.Combined_Key) as EE
        GROUP BY Country_Region
        HAVING SUM(Confirmed)>100000
```

Result:



	country_region character varying (50)	newly_diagnosed bigint
1	France	579373
2	Taiwan*	440596
3	Korea, South	206138
4	Italy	396396
5	US	389029
6	Germany	871687
7	Greece	106302
8	Japan	264185
9	Russia	212106
10	Austria	129544