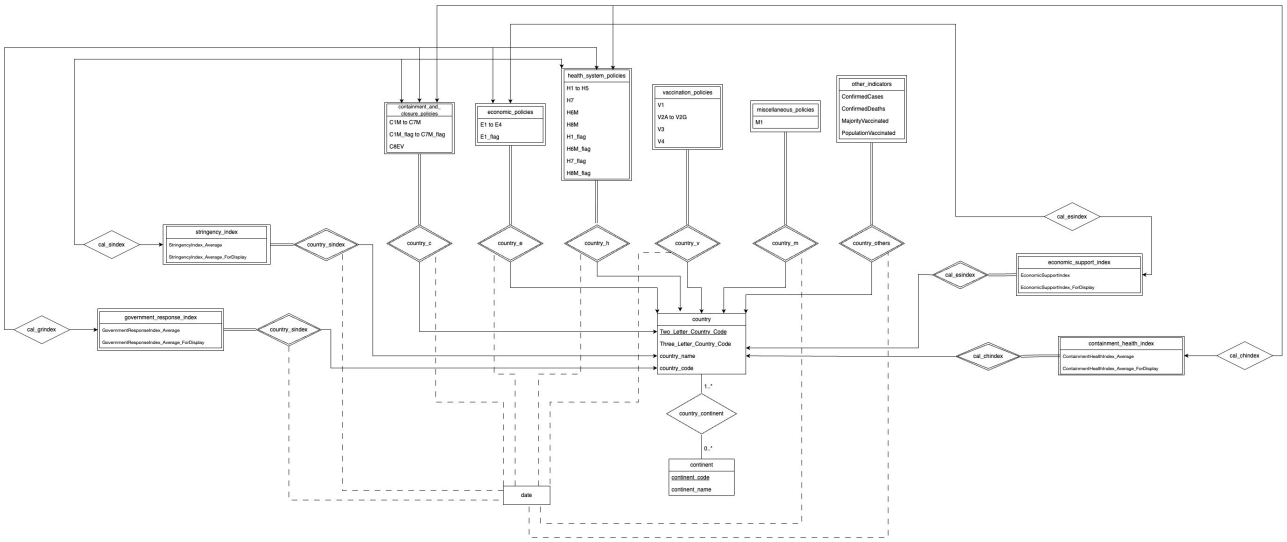
112-1 Introduction to Database HW2

111550057 莊婷馨

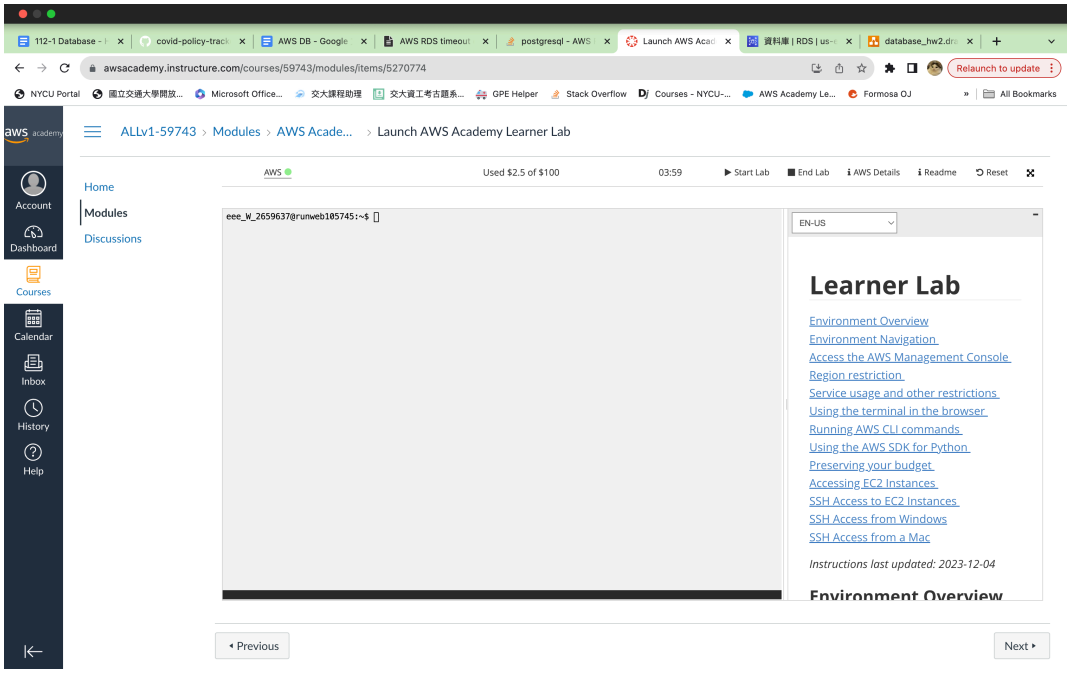
1. ER diagram with entity sets and relationship sets, with or without attributes. Add constraints if needed.

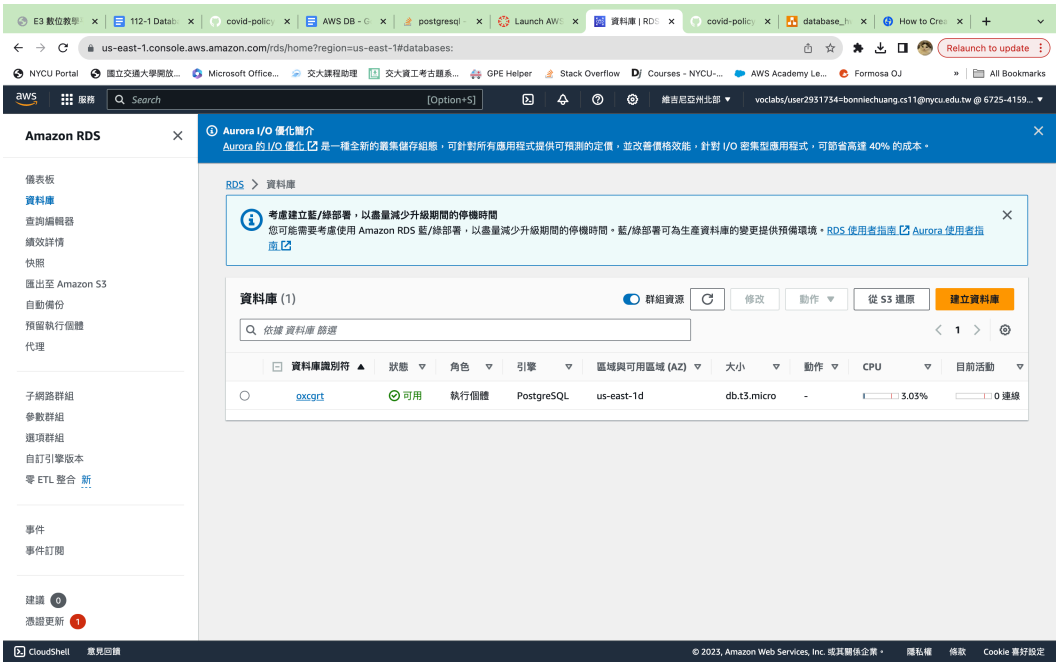


(submitted separately as .jpg file)

2. Provide print screens

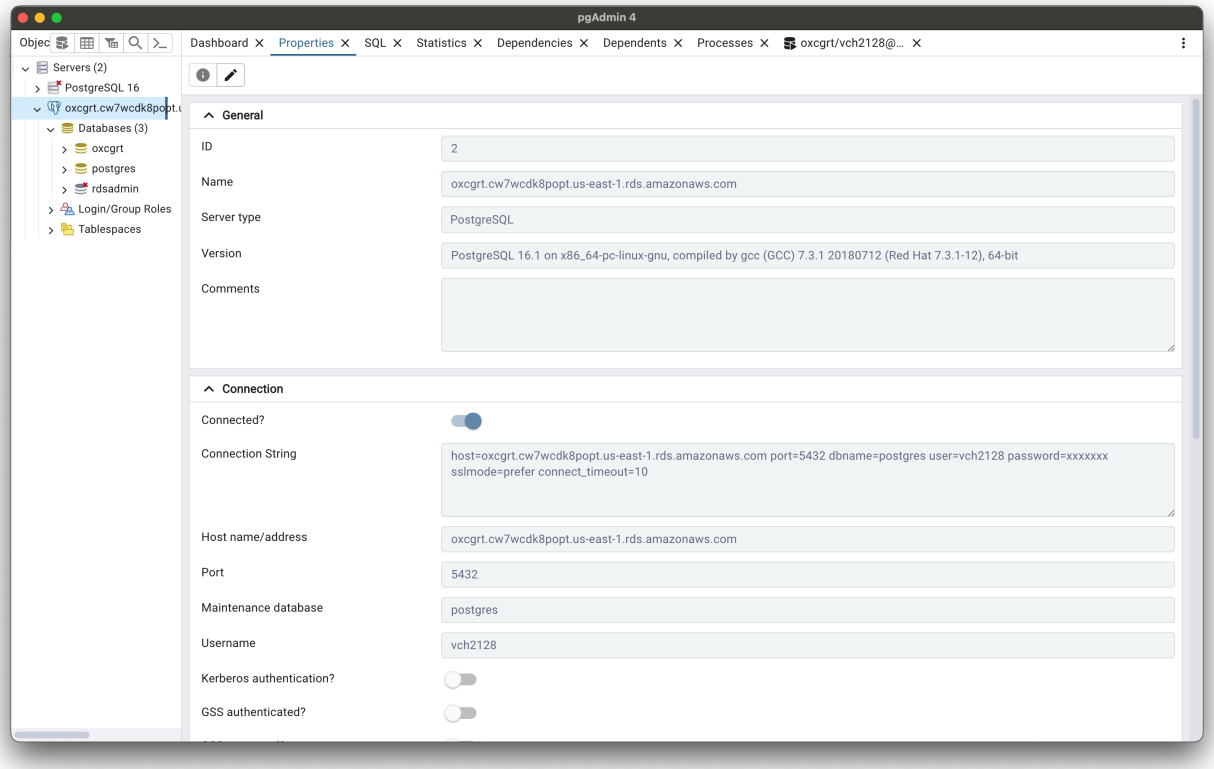
1. AWS RDS launch page





2. the way you connect to the AWS RDS (PostgreSQL console tool, pgAdmin, or other IDE's connection page, with the same IP or URL with your AWS RDS)

Ans : pgAdmin4



3. Please provide the schema after decomposition, of each table, and a print screen to show that the tables have been created in your database on AWS RDS. (10+10pts)

```
create table continent(
Continent_Code char(2),
Continent_Name char(50),
primary key (Continent_Code)
);
create table country(
Three_Letter_Country_Code char(3),
Two_Letter_Country_Code char(2),
Country_Name varchar(500),
Country_Number int,
primary key (Two_Letter_Country_Code)
);

create table country_continent(
Two_Letter_Country_Code char(2),
Continent_Code char(2),
primary key (Two_Letter_Country_Code,Continent_Code),
foreign key (Continent_Code) references continent(Continent_Code),
foreign key (Two_Letter_Country_Code) references country(Two_Letter_Country_Code)
);
```

```
  create table c_policies(
  country_code char(2),
  rdate numeric(8,0),
  C1M float(2),
  C1M_flag bool,
  C2M float(2),
  C2M_flag bool,
  C3M float(2),
  C3M_flag bool,
  C4M float(2),
  C4M_flag bool,
  C5M float(2),
  C5M_flag bool,
  C6M float(2),
  C6M_flag bool,
  C7M float(2),
  C7M_flag bool,
  C8EV float(2),
  primary key (country_code,rdate),
  foreign key (country_code) references country(two_letter_country_code)
  );
```

```
  create table v_policies(
  country_code char(2),
  rdate numeric(8,0),
  V1 int,
  V2A int,
  V2B varchar(15),
  V2C varchar(15),
  V2D int,
  V2E int,
  V2F int,
  V2G int,
  V3 int,
  V4 int,
  primary key (country_code,rdate),
  foreign key (country_code) references country(two_letter_country_code)
  );
```

```
create table e_policies(
country_code char(2),
rdate numeric(8,0),
E1 float(2),
E1_flag bool,
E2 float(2),
E3 float(2),
E4 float(2),
primary key (country_code,rdate),
foreign key (country_code) references country(two_letter_country_code)
);

create table h_policies(
country_code char(2),
rdate numeric(8,0),
H1 float(2),
H1_flag bool,
H2 float(2),
H3 float(2),
H4 float(2),
H5 float(2),
H6M float(2),
H6M_flag bool,
H7 float(2),
H7_flag bool,
H8M float(2),
H8M_flag bool,
primary key (country_code,rdate),
foreign key (country_code) references country(two_letter_country_code)
);

create table oindicators(
country_code char(2),
rdate numeric(8,0),
confirmed_cases int,
confirmed_deaths int,
population_vaccinated float(2),
primary key (country_code,rdate),
foreign key (country_code) references country(two_letter_country_code)
foreign key (population_vaccinated) references majority_v(population_vaccinated)
);

create table majority_v(
population_vaccinated float(2),
majority_vaccinated varchar(2),
primary key (population_vaccinated)
);


create table ch_index(
country_code char(2),
rdate numeric(8,0),
containmenthealthindex_average float(2),
containmenthealthindex_average_fordisplay float(2),
primary key (country_code,rdate),
foreign key (country_code) references country(two_letter_country_code)
);
```

```
create table s_index(
country_code char(2),
rdate numeric(8,0),
stringencyindex_average float(2),
stringencyindex_average_fordisplay float(2),
primary key (country_code,rdate),
foreign key (country_code) references country(two_letter_country_code)
);

create table gr_index(
country_code char(2),
rdate numeric(8,0),
governmentresponseindex_average float(2),
governmentresponseindex_average_fordisplay float(2),
primary key (country_code,rdate),
foreign key (country_code) references country(two_letter_country_code)
);

create table es_index(
country_code char(2),
rdate numeric(8,0),
economicsupportindex_average float(2),
economicsupportindex_average_fordisplay float(2),
primary key (country_code,rdate),
foreign key (country_code) references country(two_letter_country_code)
);
```

```
oxcgrt=> \d
              List of relations
 Schema |       Name         | Type  |  Owner
--------+--------------------+-------+---------
 public | c_policies         | table | vch2128
 public | ch_index           | table | vch2128
 public | continent          | table | vch2128
 public | country            | table | vch2128
 public | country_continent  | table | vch2128
 public | e_policies         | table | vch2128
 public | es_index           | table | vch2128
 public | gr_index           | table | vch2128
 public | h_policies         | table | vch2128
 public | majority_v         | table | vch2128
 public | oindicators        | table | vch2128
 public | s_index            | table | vch2128
 public | v_policies         | table | vch2128
(13 rows)
```

4. Clearly indicate the level of normal form, test the level of normal form for each table
    1. country
        - 3NF
            - candidate keys: {two_letter_country_code}, {country_name}
            - three_letter_country_code and country_number cannot be candidate key because there are null values
            - Since all non-prime attributes only depend on candidate key, it is in 3NF.

2. continent
   - 3NF
     - Only two attributes, both are candidate keys, and they depend on each other.
     - No transitive dependency

3. country_continent
   - BCNF
     - The primary key is composed of all attributes.
     - All functional dependencies are trivial.

4. c_policies, e_policies, h_policies, v_policies
   - BCNF
     - All non-prime attributes depend on the primary key.
     - No other functional dependencies.
     - For all a->b , a is a superkey for R.

5. oindicators
   - BCNF
     - All non-prime attributes depend on the primary key.
     - No other functional dependencies.
     - For all a->b , a is a superkey for R.

6. majority_v
   - BCNF
     - Only one functional dependency, and the one depended on is the primary key.
     - For all a->b , a is a superkey for R.

7. s_index, gr_index, es_index, ch_index
   - BCNF
     - All non-prime attributes depend on the primary key.
     - No other functional dependencies.
     - For all a->b , a is a superkey for R.

5. List the functional dependency of each table.

```
//continent
continent_code -> continent_name
continent_name -> continent_code

//country
two_letter_country_code -> three_letter_country_code
two_letter_country_code -> country_number
two_letter_country_code -> country_name
country_name -> two_letter_country_code
country_name -> three_letter_country_code
country_name -> country_number

//country_continent
{country_code,continent_code} -> {country_code,continent_code}

//c_policies
{country_code,rdate} -> C1M~C8EV and flags

//e_policies
{country_code,rdate} -> E1~E4 and flags

//h_policies
{country_code,rdate} -> H1~H8 and flags

//v_policies
{country_code,rdate} -> V1~V4

//oindicators
{country_code,rdate} -> confirmed_cases
{country_code,rdate} -> confirmed_deaths
{country_code,rdate} -> population_vaccinated

//majority_v
{population_vaccinated} -> majority_vaccinated

//s_index
{country_code,rdate} -> stringencyindex_average
{country_code,rdate} -> stringencyindex_average_fordisplay

//gr_index
{country_code,rdate} -> governmentresponseindex_average
{country_code,rdate} -> governmentresponseindex_fordisplay

//es_index
{country_code,rdate} -> economicsupportindex_average
{country_code,rdate} -> economicsupportindex_average_fordisplay

//ch_index
{country_code,rdate} -> containmenthealthindex_average
{country_code,rdate} -> containmenthealthindex_fordisplay
```

6. The SQL statements (in .sql file) and output results of 4a

- SQL statements

```
with cc as(
select country_code,continent_code,rdate,StringencyIndex_Average_ForDisplay as sindex
from s_index
join country_continent on s_index.country_code=country_continent.two_letter_country_code
where rdate=20221201 or rdate=20220401 or rdate=20210401 or rdate=20200401
),
gmax as(
select continent_code,rdate,max(sindex) as sindex
from cc
group by (continent_code,rdate)
),
gmin as(
select continent_code,rdate,min(sindex) as sindex
from cc
group by (continent_code,rdate)
),
gmaxc as(
select cc.country_code,cc.continent_code,cc.rdate,cc.sindex
from cc
join gmax
on cc.continent_code=gmax.continent_code and cc.rdate=gmax.rdate and gmax.sindex=cc.sindex
),
gminc as(
select cc.country_code,cc.continent_code,cc.rdate,cc.sindex
from cc
join gmin
on cc.continent_code=gmin.continent_code and cc.rdate=gmin.rdate and gmin.sindex=cc.sindex
)
(select continent_name,country_name,rdate as date,sindex as stringency_index
from gmaxc
join country on gmaxc.country_code=country.two_letter_country_code
join continent on gmaxc.continent_code=continent.continent_code
)
union
(select continent_name,country_name,rdate as date,sindex as stringency_index
from gminc
join country on gminc.country_code=country.two_letter_country_code
join continent on gminc.continent_code=continent.continent_code
)
order by date,continent_name
```

- output results
  - no results for date=20221201 because there is no data for the date

	continent_name character	country_name character varying (500)	date numeric (8)	stringency_index real
1	Africa	Burundi	20200401	13.89
2	Africa	Congo	20200401	97.22
3	Asia	Tajikistan	20200401	19.44
4	Asia	Georgia	20200401	100
5	Asia	India	20200401	100
6	Asia	Jordan	20200401	100
7	Asia	Philippines	20200401	100
8	Asia	Sri Lanka	20200401	100
9	Europe	Serbia	20200401	100
10	Europe	Georgia	20200401	100
11	Europe	Belarus	20200401	12.04
12	North America	Honduras	20200401	100
13	North America	Nicaragua	20200401	15.74
14	Oceania	Kiribati	20200401	40.74
15	Oceania	New Zealand	20200401	96.3
16	South America	Argentina	20200401	100
17	South America	Guyana	20200401	57.41
18	Africa	Tanzania	20210401	8.33
19	Africa	Mauritius	20210401	96.3
20	Asia	Laos	20210401	16.67
21	Asia	Timor-Leste	20210401	85.19
22	Europe	Greece	20210401	87.96
23	Europe	Russia	20210401	36.57
24	North America	Nicaragua	20210401	13.89
25	North America	Honduras	20210401	82.41
26	Oceania	New Zealand	20210401	22.22
27	Oceania	Vanuatu	20210401	22.22
28	Oceania	Kiribati	20210401	22.22
29	Oceania	Papua New Guinea	20210401	62.04
30	South America	Bolivia	20210401	25
31	South America	Venezuela	20210401	87.96
32	Africa	Gabon	20220401	11.11
33	Africa	Seychelles	20220401	56.48
34	Asia	Mongolia	20220401	0
35	Asia	Myanmar	20220401	78.7
36	Europe	Andorra	20220401	8.33
37	Europe	Ukraine	20220401	60.16
38	North America	Dominican Republic	20220401	8.33
39	North America	Nicaragua	20220401	8.33
40	North America	Dominica	20220401	59.41
41	Oceania	Vanuatu	20220401	85.19
42	Oceania	Fiji	20220401	32.42
43	South America	Suriname	20220401	50.65
44	South America	Uruguay	20220401	14.82

7. The SQL statements (in .sql file) and output results of 4b (10pts)

- SQL statements

```
with ccsindex as(
select country_code,continent_code,rdate,StringencyIndex_Average_ForDisplay as sindex
from s_index
join country_continent on s_index.country_code=country_continent.two_letter_country_code
where rdate=20221201 or rdate=20220401 or rdate=20210401 or rdate=20200401
),
earlier as(
select country_code,rdate,confirmed_cases as cases
from oindicators
where rdate=20221124 or rdate=20220325 or rdate=20210325 or rdate=20200325
),
now as(
select country_code,rdate,confirmed_cases as cases
from oindicators
where rdate=20221201 or rdate=20220401 or rdate=20210401 or rdate=20200401
),
dif as(
select now.country_code,now.rdate,earlier.cases as ecases,now.cases as ncases
from earlier
join now on earlier.country_code=now.country_code
where (earlier.rdate=20221124 and now.rdate=20221201)
  or (earlier.rdate=20220325 and now.rdate=20220401)
  or (earlier.rdate=20210325 and now.rdate=20210401)
  or (earlier.rdate=20200325 and now.rdate=20200401)
),
diff as (
select country_code,rdate,(cast(ncases as float(2))-cast(ecases as float(2)))/7 as casesgrowth
from dif
),
oversindex as (
select diff.country_code,continent_code,diff.rdate,case
                    when casesgrowth!=0 then sindex/casesgrowth
                    when casesgrowth=0 then sindex/0.1
                    end as osindex
from ccsindex
join diff
on diff.country_code=ccsindex.country_code and diff.rdate=ccsindex.rdate
),
omax as (
select continent_code,rdate,max(osindex) as osindex
from oversindex
group by (continent_code,rdate)
),
omin as (
select continent_code,rdate,min(osindex) as osindex
from oversindex
group by (continent_code,rdate)
),
omm as(
(select omax.continent_code,country_code,omax.rdate,omax.osindex
from omax
join oversindex
on omax.continent_code=oversindex.continent_code
  and omax.rdate=oversindex.rdate
  and omax.osindex=oversindex.osindex)
union
(select omin.continent_code,country_code,omin.rdate,omin.osindex
from omin
join oversindex
on omin.continent_code=oversindex.continent_code
  and omin.rdate=oversindex.rdate
  and omin.osindex=oversindex.osindex)
)
select continent_name,country_name,rdate as date,osindex as over_stringency_index
from omm
join country on omm.country_code=country.two_letter_country_code
join continent on omm.continent_code=continent.continent_code
order by date,continent_name;
```

- output results
  - no results for date=20221201 because there is no data for the date

	continent_name character	country_name character varying (500)	date numeric (8)	over_stringency_index double precision
1	Africa	South Africa	20200401	0.9176154897038876
2	Africa	Lesotho	20200401	907.3999786376953
3	Asia	Iran	20200401	0.019531006946534056
4	Asia	Timor-Leste	20200401	750
5	Europe	Spain	20200401	0.010921195119129787
6	Europe	Monaco	20200401	25.65499973297119
7	North America	Belize	20200401	525
8	North America	United States	20200401	0.0031991425262797306
9	Oceania	Tonga	20200401	935.1999664306641
10	Oceania	Australia	20200401	0.19979984842366083
11	South America	Brazil	20200401	0.12185427520053513
12	South America	Suriname	20200401	249.55001068115234
13	Africa	Congo	20210401	472.20001220703125
14	Africa	Cameroon	20210401	0.01473706194358139
15	Asia	India	20210401	0.0008874230633242567
16	Asia	Tajikistan	20210401	287.00000762939453
17	Europe	France	20210401	0.0017857966405600621
18	Europe	Faeroe Islands	20210401	481.50001525878906
19	North America	Greenland	20210401	370.40000915527344
20	North America	United States	20210401	0.000849799551856855
21	Oceania	Papua New Guinea	20210401	0.2168147810327965
22	Oceania	Fiji	20210401	490.6999969482422
23	South America	Brazil	20210401	0.0009635592530669499
24	South America	Suriname	20210401	13.313243762866872
25	Africa	Botswana	20220401	0.002338608870580606
26	Africa	Guinea	20220401	462.99999237060547
27	Asia	Mongolia	20220401	0
28	Asia	Macao	20220401	324.0999984741211
29	Europe	France	20220401	0.00013628144058999039
30	Europe	Faeroe Islands	20220401	111.09999656677246
31	North America	United States	20220401	0.0010651089173668248
32	North America	El Salvador	20220401	332.20001220703125
33	Oceania	Australia	20220401	0.000745880683779704
34	Oceania	Kiribati	20220401	106.13750267028809
35	South America	Guyana	20220401	5.179999923706054
36	South America	Brazil	20220401	0.0013755748289236047