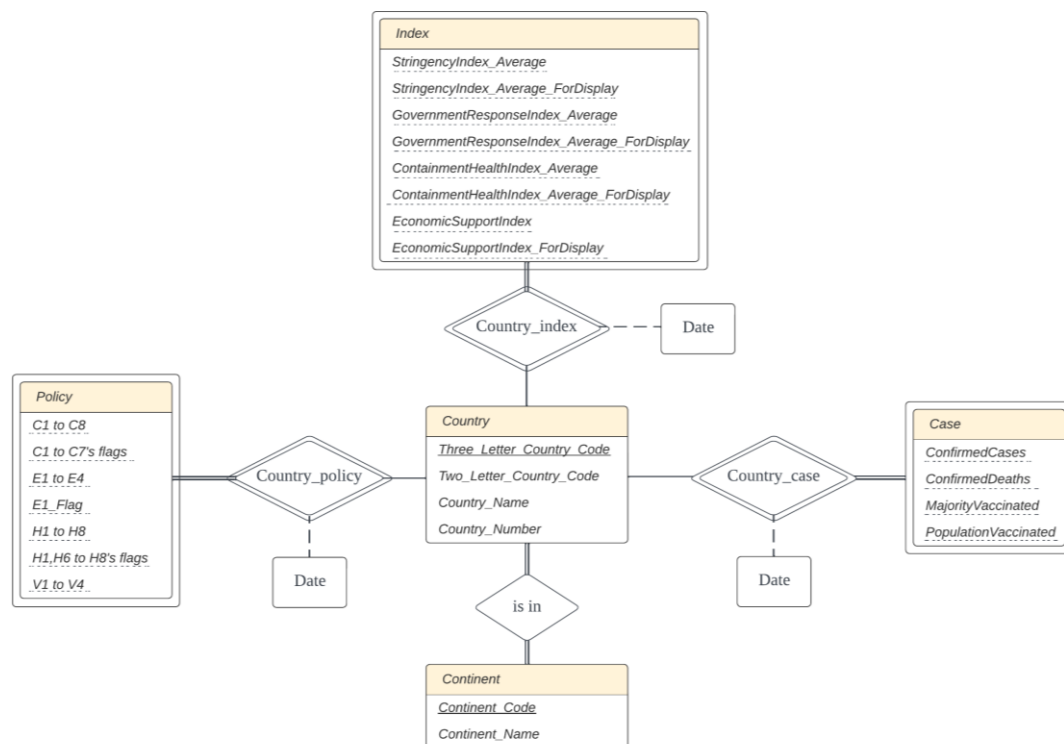


Database_HW2

1. ER diagram with entity sets and relationship sets, with or without attributes. Add constraints if needed. (30pts)

Ans:

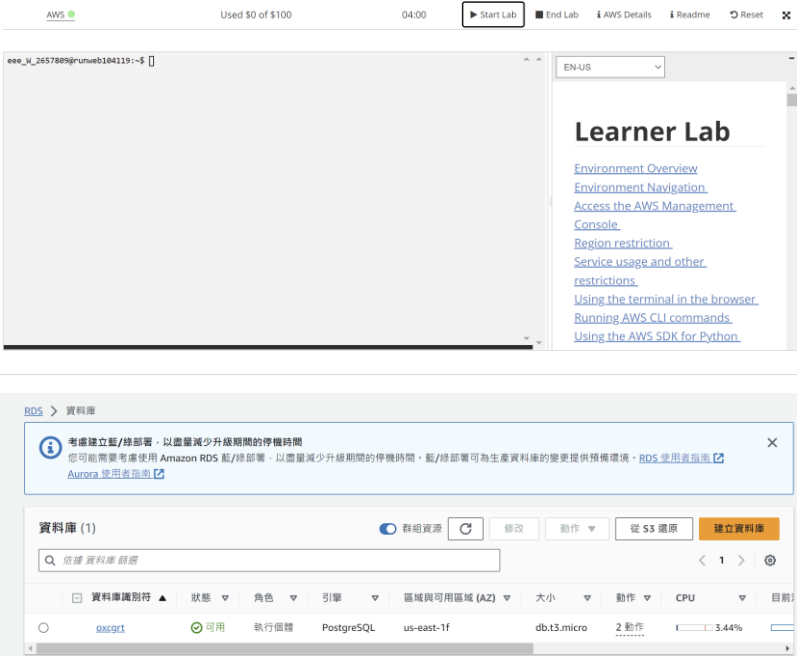
- (1) Because “RegionName”, “RegionCode” and “MI_Wildcard” are all NULL, and “Jurisdiction” are all NAT_TOTAL in all rows, I remove column “RegionName”, “RegionCode”, “MI_Wildcard” and “Jurisdiction” in “OxCGRT_nat_latest.csv”.
- (2) Because “Kosovo” isn’t in “country-and-continent-codes-list-csv”, I add it in the csv file.
- (3) Some countries in “country-and-continent-codes-list-csv” don’t have “Two_Letter_Country_Code”, and these countries are also not in “OxCGRT_nat_latest.csv”. Therefore, I remove those countries, including “Disputed Territory, Iraq-Saudi Arabia Neutral Zone, United Nations Neutral Zone, Spratly Islands”.



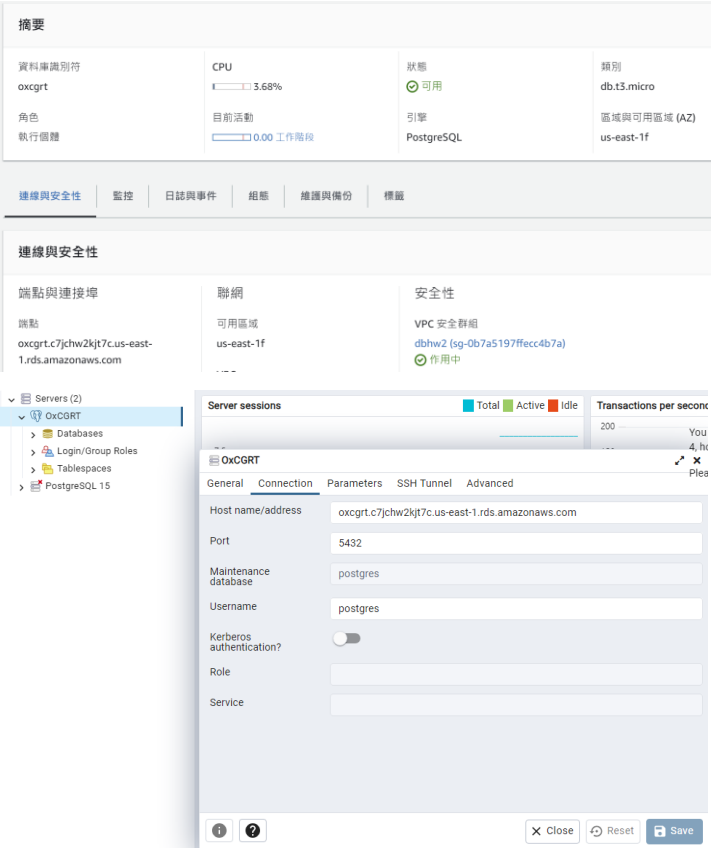
2. Provide print screens of the 1) AWS RDS launch page, and 2) the way you connect to the AWS RDS (PostgreSQL console tool, pgAdmin, or other IDE’s connection page, with the same IP or URL with your AWS RDS) (10pts)

Ans:

(1)



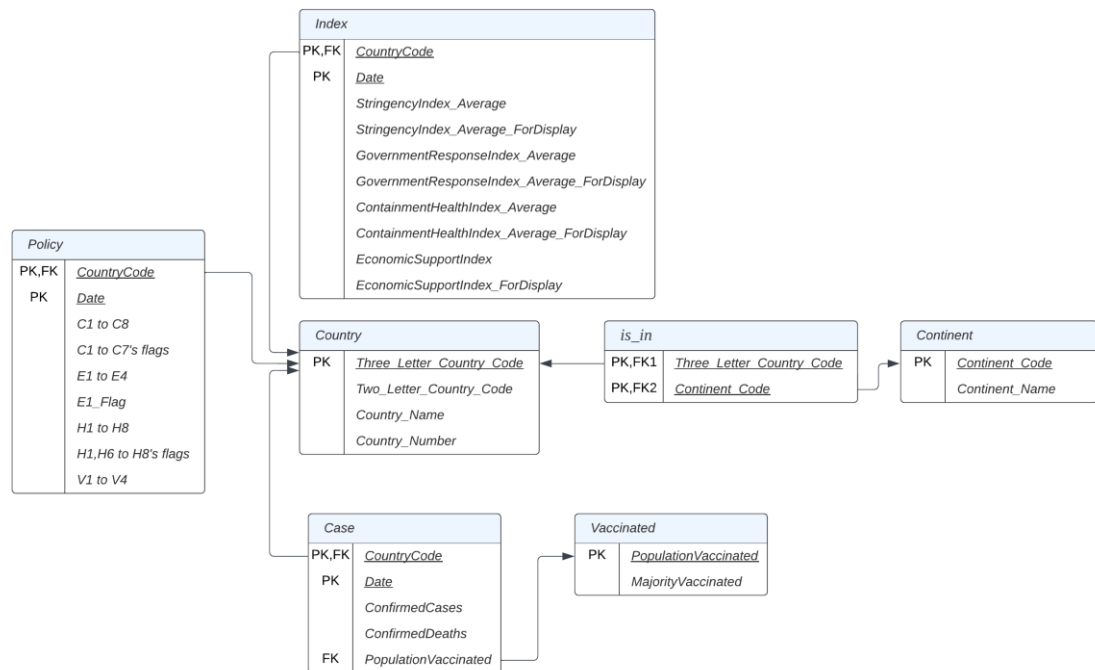
(2) Using “pgAdmin”



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)

Ans:

- Schema of the tables:



- Print screen:

- Continent:

```

1 create table continent(
2     Continent_Name varchar(100),
3     Continent_Code char(2),
4     primary key (Continent_Code)
5 );
  
```

- Country:

```

6 create table country(
7     Country_Name varchar(100) not NULL,
8     Two_Letter_Country_Code char(2),
9     Three_Letter_Country_Code char(3),
10    Country_Number int,
11    primary key(Three_Letter_Country_Code)
12 );
  
```

- is_in:

```

13 create table is_in(
14     Continent_Code char(2),
15     Three_Letter_Country_Code char(3),
16     primary key(Continent_Code,Three_Letter_Country_Code),
17     foreign key(Continent_Code) references continent,
18     foreign key(Three_Letter_Country_Code) references country
19 );
  
```

- Policies:

```

20 create table policies(
21     CountryCode varchar(3) not null,
22     Date numeric(8,0) not null,
23     C1M int,C1M_flag bool,C2M int,C2M_flag bool,
24     C3M int,C3M_flag bool,C4M int,C4M_flag bool,
25     C5M int,C5M_flag bool,C6M int,C6M_flag bool,
26     C7M int,C7M_flag bool,C8EV int,
27     E1 int,E1_flag bool,E2 int,E3 double precision,E4 numeric,
28     H1 int,H1_flag bool,
29     H2 int,H3 int,H4 double precision,H5 double precision,
30     H6 int,H6_flag bool,H7 int,H7_flag bool,H8 int,H8_flag bool,
31     V1 int,V2A int,V2B varchar(20),V2C varchar(20),
32     V2D int,V2E int,V2F int,V2G int,V3 int,V4 int,
33     primary key(CountryCode,Date),
34     foreign key(CountryCode)references country(Three_Letter_Country_Code)
35 );

```

- Vaccinated:

```

36 create table vaccinated(
37     PopulationVaccinated double precision,
38     MajorityVaccinated varchar(2),
39     primary key(PopulationVaccinated)
40 );

```

- Cases:

```

41 create table cases(
42     CountryCode varchar(3) not null,
43     Date numeric(8,0) not null,
44     ConfirmedCases int,
45     ConfirmedDeaths int,
46     PopulationVaccinated double precision,
47     primary key(CountryCode,Date),
48     foreign key(CountryCode)references country(Three_Letter_Country_Code),
49     foreign key(PopulationVaccinated)references vaccinated(PopulationVaccinated)
50 );

```

- Indices:

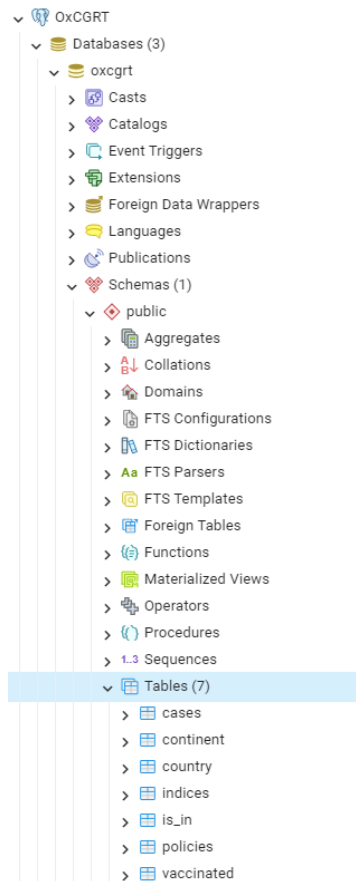
Because question requirement of 4a and 4b query need to output country name using the name in “OxCGRT_nat_latest.csv” not “country-and-continent-codes-list.csv”. Only for convenience, I add the column “Country_Name” in indices, so I don’t need to modify the name in “country-and-continent-codes-list.csv”.

However, I do not add the column “Country_Name” in ER diagram or schema.

```

51 create table indices(
52     Country_Name varchar(100),
53     CountryCode varchar(3) not null,
54     Date numeric(8,0) not null,
55     StringencyIndex_Average double precision,
56     StringencyIndex_Average_ForDisplay double precision,
57     GovernmentResponseIndex_Average double precision,
58     GovernmentResponseIndex_Average_ForDisplay double precision,
59     ContainmentHealthIndex_Average double precision,
60     ContainmentHealthIndex_Average_ForDisplay double precision,
61     EconomicSupportIndex double precision,
62     EconomicSupportIndex_ForDisplay double precision,
63     primary key(CountryCode,Date),
64     foreign key(CountryCode)references country(Three_Letter_Country_Code)
65 );

```



4. Clearly indicate the level of normal form, test the level of normal form for each table (10pts)

Ans:

- *Continent (Continent_Name, Continent_Code) :*
 - Normal form: BCNF
 - Test:

The primary key is *Continent_Code*. Both *Continent_Name* and *Continent_Code* are superkeys.
- *Country (Country_Name, Two_Letter_Country_Code, Three_Letter_Country_Code, Country_Number) :*
 - Normal form: BCNF
 - Test:

The primary key is *Three_Letter_Country_Code*. All the attributes in this table are superkeys, so it satisfies BCNF.
- *Is_in (Continent_Code, Three_Letter_Country_Code) :*
 - Normal form: BCNF
 - Test:

All the attributes of the table are primary keys, so it satisfies BCNF.

- Policies (CountryCode, Date, *C1~C8*, *C1~C7's flag*, ...) :
 - Normal form: BCNF
 - Test:

{ *CountryCode*, *Date* } is superkey and { *C1~C8*, *C1~C7's flag*, ... } is fully functionally dependent on { *CountryCode*, *Date* }, which means *CountryCode* or *Date* doesn't have functional dependency with other attributes on their own.
- Vaccinated (PopulationVaccinated, *MajorityVaccinated*) :
 - Normal form: BCNF
 - Test:

PopulationVaccinated is the primary key, so it's a superkey for this table.
- Cases (CountryCode, Date, *ConfirmedCases*, ...) :
 - Normal form: BCNF
 - Test:

{ *CountryCode*, *Date* } is superkey and { *ConfirmedCases*, ... } is fully functionally dependent on { *CountryCode*, *Date* }, which means *CountryCode* or *Date* doesn't have functional dependency with other attributes on their own.
- Indices (CountryCode, Date, *StringencyIndex_Average*, ...) :
 - Normal form: BCNF
 - Test:

{ *CountryCode*, *Date* } is superkey and { *StringencyIndex_Average*, *GovernmentResponseIndex_Average*, ... } is fully functionally dependent on { *CountryCode*, *Date* }, which means *CountryCode* or *Date* doesn't have functional dependency with other attributes on their own.

5. List the functional dependency of each table. (10pts)

Ans:

- Continent:
 - *Continent_Name* → *Continent_Code*
 - *Continent_Code* → *Continent_Name*
- Country:
 - *Three_Letter_Country_Code* → { *Country_Name*, *Two_Letter_Country_Code*, *Country_Number* }
 - *Country_Name* → { *Three_Letter_Country_Code*, *Two_Letter_Country_Code*, *Country_Number* }
 - *Two_Letter_Country_Code* → { *Three_Letter_Country_Code*, *Country_Name*, *Country_Number* }
 - *Country_Number* → { *Three_Letter_Country_Code*, *Country_Name*, *Two_Letter_Country_Code* }

- Is_in:
Because all the attributes are primary keys so there's no functional dependency.
- Policies:
 - $\{ \text{CountryCode}, \text{Date} \} \rightarrow \{ C1\sim C8, C1\sim C7\text{'s flag}, E1\sim E4\&\text{their flags}, H1\sim H8\&\text{their flags}, V1\sim V4 \}$
- Vaccinated:
 - $\text{PopulationVaccinated} \rightarrow \text{MajorityVaccinated}$
- Cases:
 - $\{ \text{CountryCode}, \text{Date} \} \rightarrow \{ \text{ConfirmedCases}, \text{ConfirmedDeaths}, \text{PopulationVaccinated} \}$
- Indices:
 - $\{ \text{CountryCode}, \text{Date} \} \rightarrow \{ \text{StringencyIndex_Average}, \text{GovernmentResponseIndex_Average}, \text{ContainmentHealthIndex_Average}, \text{EconomicSupportIndex and their ForDisplay indices} \}$

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

Ans:

- SQL statements:

```

1 WITH stringency_data(Date,Continent_Name,Country_Name,StringencyIndex) AS (
2     SELECT
3         indices.Date,
4         continent.Continent_Name,
5         indices.Country_Name,
6         indices.StringencyIndex_Average_ForDisplay as StringencyIndex,
7         RANK() OVER (PARTITION BY Continent_Name,Date ORDER BY StringencyIndex_Average_ForDisplay DESC) AS Rank_Max,
8         RANK() OVER (PARTITION BY Continent_Name,Date ORDER BY StringencyIndex_Average_ForDisplay ASC) AS Rank_Min
9     FROM indices
10    RIGHT JOIN country ON country.Three_Letter_Country_Code=indices.CountryCode
11    RIGHT JOIN is_in ON is_in.Three_Letter_Country_Code=country.Three_Letter_Country_Code
12    RIGHT JOIN continent ON continent.Continent_Code=is_in.Continent_Code
13    WHERE indices.StringencyIndex_Average_ForDisplay IS NOT NULL
14    AND (indices.Date='20221201' OR indices.Date='20220401' OR indices.Date='20210401' OR indices.Date='20200401')
15 )
16 SELECT Date,Continent_Name,Country_Name,StringencyIndex
17 FROM stringency_data
18 WHERE Rank_Max = 1 OR Rank_Min = 1
19 ORDER BY Date,Continent_Name,StringencyIndex;

```

- Output results:
The data on 2022/12/01 are all empty, so I didn't output the results.

- 2020/04/01:

	date numeric (8) 🔒	continent_name character varying (100) 🔒	country_name character varying (100) 🔒	stringencyindex double precision 🔒
1	20200401	Africa	Burundi	13.89
2	20200401	Africa	Congo	97.22
3	20200401	Asia	Tajikistan	19.44
4	20200401	Asia	Jordan	100
5	20200401	Asia	Philippines	100
6	20200401	Asia	India	100
7	20200401	Asia	Sri Lanka	100
8	20200401	Asia	Georgia	100
9	20200401	Europe	Belarus	12.04
10	20200401	Europe	Serbia	100
11	20200401	Europe	Georgia	100
12	20200401	North America	Nicaragua	15.74
13	20200401	North America	Honduras	100
14	20200401	Oceania	Kiribati	40.74
15	20200401	Oceania	New Zealand	96.3
16	20200401	South America	Guyana	57.41
17	20200401	South America	Argentina	100

- 2021/04/01:

	date numeric (8) 🔒	continent_name character varying (100) 🔒	country_name character varying (100) 🔒	stringencyindex double precision 🔒
18	20210401	Africa	Tanzania	8.33
19	20210401	Africa	Mauritius	96.3
20	20210401	Asia	Laos	16.67
21	20210401	Asia	Timor-Leste	85.19
22	20210401	Europe	Russia	36.57
23	20210401	Europe	Greece	87.96
24	20210401	North America	Nicaragua	13.89
25	20210401	North America	Honduras	82.41
26	20210401	Oceania	New Zealand	22.22
27	20210401	Oceania	Kiribati	22.22
28	20210401	Oceania	Vanuatu	22.22
29	20210401	Oceania	Papua New Guinea	62.04
30	20210401	South America	Bolivia	25
31	20210401	South America	Venezuela	87.96

- 2022/04/01:

	date numeric (8) 🔒	continent_name character varying (100) 🔒	country_name character varying (100) 🔒	stringencyindex double precision 🔒
32	20220401	Africa	Gabon	11.11
33	20220401	Africa	Seychelles	56.48
34	20220401	Asia	Mongolia	0
35	20220401	Asia	Myanmar	78.7
36	20220401	Europe	Andorra	8.33
37	20220401	Europe	Ukraine	60.16
38	20220401	North America	Nicaragua	8.33
39	20220401	North America	Dominican Republic	8.33
40	20220401	North America	Dominica	59.41
41	20220401	Oceania	Fiji	32.42
42	20220401	Oceania	Vanuatu	85.19
43	20220401	South America	Uruguay	14.82
44	20220401	South America	Suriname	50.65

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

Ans:

- SQL statements (of 2020/04/01) :

The SQL statements of 2021/04/01 and 2022/04/01 are similar to the below code.

```

1  WITH cases_200401(Date,Continent_Name,Country_Code,ConfirmedCases) AS(
2      SELECT
3          cases.Date,
4          continent.Continent_Name,
5          country.Three_Letter_Country_Code,
6          cases.ConfirmedCases
7      FROM cases
8          RIGHT JOIN country ON country.Three_Letter_Country_Code=cases.CountryCode
9          RIGHT JOIN is_in ON is_in.Three_Letter_Country_Code=country.Three_Letter_Country_Code
10         RIGHT JOIN continent ON continent.Continent_Code=is_in.Continent_Code
11         WHERE Date='20200325' OR Date='20200401'
12 ),
13 average_7day_200401(Continent_Name,Country_Code,Case_average) AS(
14     SELECT t1.Continent_Name,t1.Country_Code,
15     CASE
16     WHEN ((CAST(t2.ConfirmedCases AS FLOAT(2))-CAST(t1.ConfirmedCases AS FLOAT(2)))/7)=0 THEN 0.10
17     ELSE ((CAST(t2.ConfirmedCases AS FLOAT(2))-CAST(t1.ConfirmedCases AS FLOAT(2)))/7)
18     END
19     FROM cases_200401 AS t1, cases_200401 AS t2
20     WHERE t1.Country_Code=t2.Country_Code
21     AND t2.Date='20200401' AND t1.Date='20200325'
22 ),
23 overindex_200401(Date,Continent_Name,Country_Name,Over_Stringency_Index) AS(
24     SELECT '20200401' AS Date, a1.Continent_Name, indices.Country_Name,
25     (indices.StringencyIndex_Average_ForDisplay/a1.Case_average) AS Over_Stringency_Index,
26     RANK() OVER (PARTITION BY Continent_Name ORDER BY
27     (indices.StringencyIndex_Average_ForDisplay/a1.Case_average) DESC) AS Rank_Max,
28     RANK() OVER (PARTITION BY Continent_Name ORDER BY
29     (indices.StringencyIndex_Average_ForDisplay/a1.Case_average) ASC) AS Rank_Min
30     FROM average_7day_200401 AS a1, indices
31     WHERE a1.Country_Code=indices.CountryCode
32     AND indices.StringencyIndex_Average_ForDisplay IS NOT NULL
33     AND indices.Date='20200401'
34 )
35 SELECT
36     Date,Continent_Name,Country_Name,Over_Stringency_Index
37 FROM overindex_200401
38 WHERE Rank_Max = 1 OR Rank_Min = 1
39 ORDER BY Continent_Name;
```

- Output results:

The data on 2022/12/01 are all empty, so I didn't output the results.

- 20200401:

	date text	continent_name character varying (100)	country_name character varying (100)	over_stringency_index double precision
1	20200401	Africa	South Africa	0.9176154992548434
2	20200401	Africa	Lesotho	907.3999999999999
3	20200401	Asia	Iran	0.01953100699844479
4	20200401	Asia	Timor-Leste	750
5	20200401	Europe	Spain	0.010921194806146181
6	20200401	Europe	Monaco	25.655
7	20200401	North America	United States	0.003199142418831577
8	20200401	North America	Belize	525
9	20200401	Oceania	Australia	0.19979983987189753
10	20200401	Oceania	Tonga	935.1999999999999
11	20200401	South America	Brazil	0.12185427370387672
12	20200401	South America	Suriname	249.55

- 20210401:

	date text	continent_name character varying (100)	country_name character varying (100)	over_stringency_index double precision
1	20210401	Africa	Cameroon	0.014737062518695783
2	20210401	Africa	Congo	472.2
3	20210401	Asia	India	0.0008874230797035569
4	20210401	Asia	Tajikistan	287
5	20210401	Europe	France	0.001785796728049861
6	20210401	Europe	Faeroe Islands	481.49999999999994
7	20210401	North America	United States	0.0008497995449127751
8	20210401	North America	Greenland	370.4
9	20210401	Oceania	Papua New Guinea	0.2168147778332501
10	20210401	Oceania	Fiji	490.7
11	20210401	South America	Brazil	0.0009635592281575945
12	20210401	South America	Suriname	13.313243243243244

- 20220401:

	date text	continent_name character varying (100)	country_name character varying (100)	over_stringency_index double precision
1	20220401	Africa	Botswana	0.002338608812776602
2	20220401	Africa	Guinea	462.99999999999994
3	20220401	Asia	Mongolia	0
4	20220401	Asia	Macao	324.09999999999997
5	20220401	Europe	France	0.00013628144279986265
6	20220401	Europe	Faeroe Islands	111.1
7	20220401	North America	United States	0.0010651089391162697
8	20220401	North America	El Salvador	332.2
9	20220401	Oceania	Australia	0.000745880681107425
10	20220401	Oceania	Kiribati	106.1375
11	20220401	South America	Brazil	0.0013755748352534015
12	20220401	South America	Guyana	5.18