



Database - HW2

Part 1

ER Diagram

Avoid Redundancy

Add Constraints

Part 2

Part 3

Schema Diagram

SQL queries to create table

ScreenShot

Part 4

Countries

Continent

Country_Continent

Statistic

Indices

Policy_Indicators

Part 5

Countries

Continent

Country_Continent

Statistic

Indices

Policy_Indicators

Part 6

SQL Query

Output Results

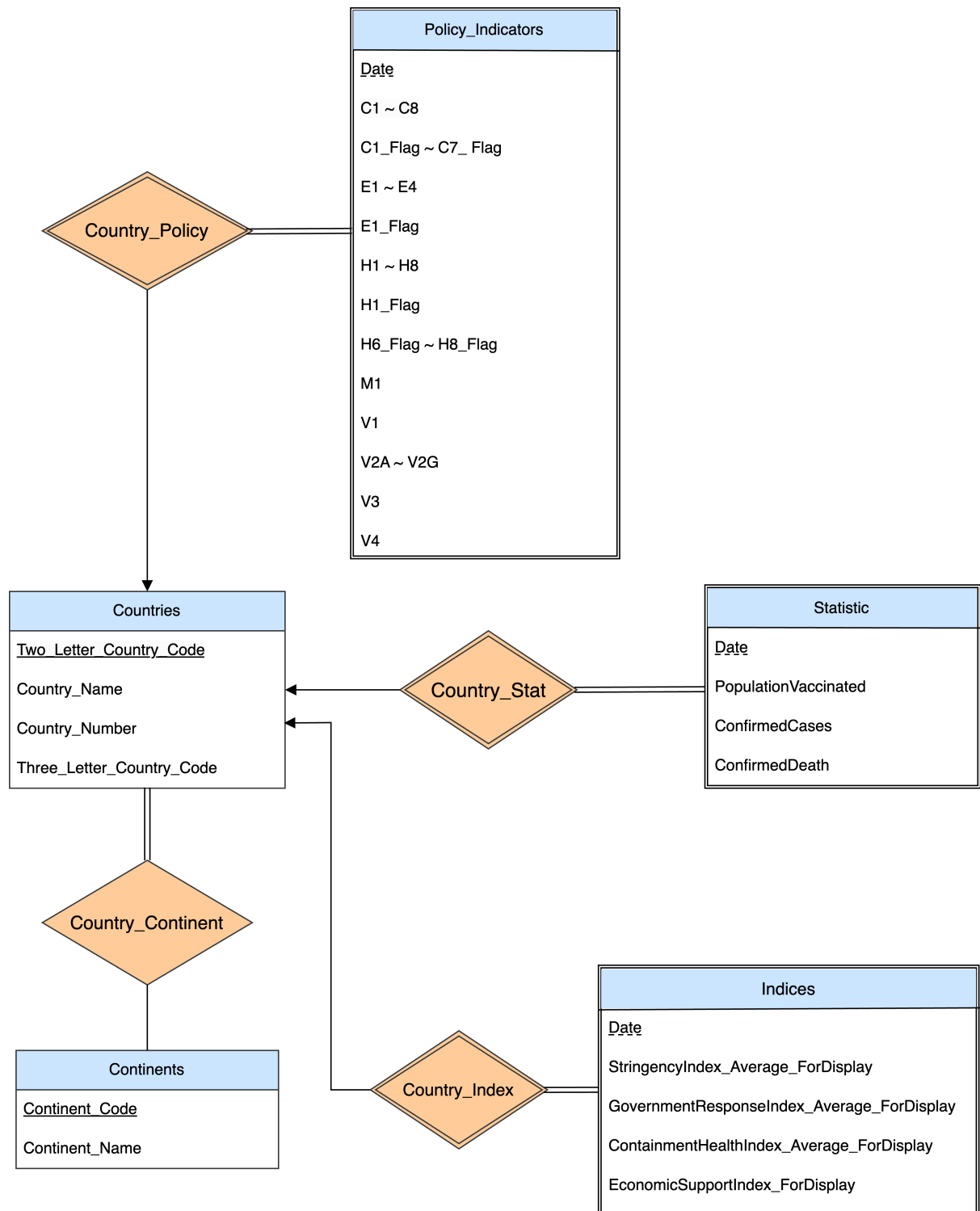
Part 7

SQL Query

Output Results

Part 1

ER Diagram



Avoid Redundancy

- Keep every relation in BCNF to avoid redundancy. This will explain later.
- Remove columns “*RegionName*”, “*RegionCode*” and “*Jurisdiction*”. Because all datas from the columns of “*RegionName*” and “*RegionCode*” are all **NULL** data. And all datas from the column of “*Jurisdiction*” are all “**NAT_TOTAL**”.

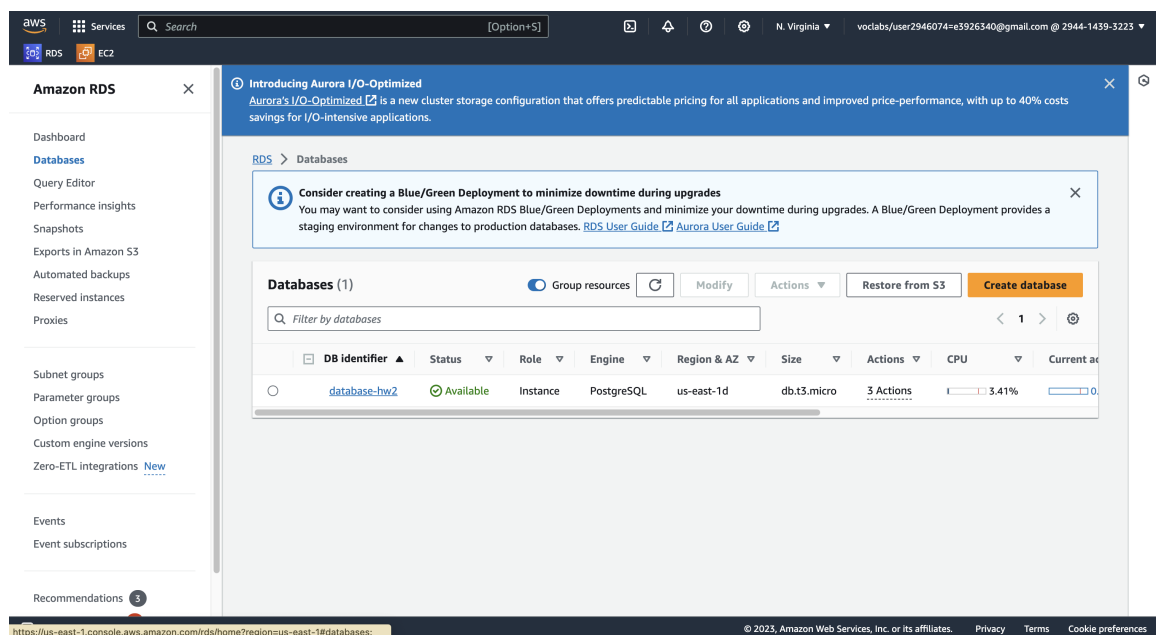
- Remove column “*MajorityVaccinated*” because it can be simply derived by “*PopulationVaccinated*”.
- Remove columns “*StringencyIndex_Average*”, “*GovernmentResponseIndex_Average*”, “*ContainmentHealthIndex_Average*”, “*EconomicSupportIndex*”. Because these datas are highly redundant with “*ForDisplay*” version. The only difference is that “*ForDisplay*” version have more datas than origin, so we can still have original version by deleting some rows if we want.

Add Constraints

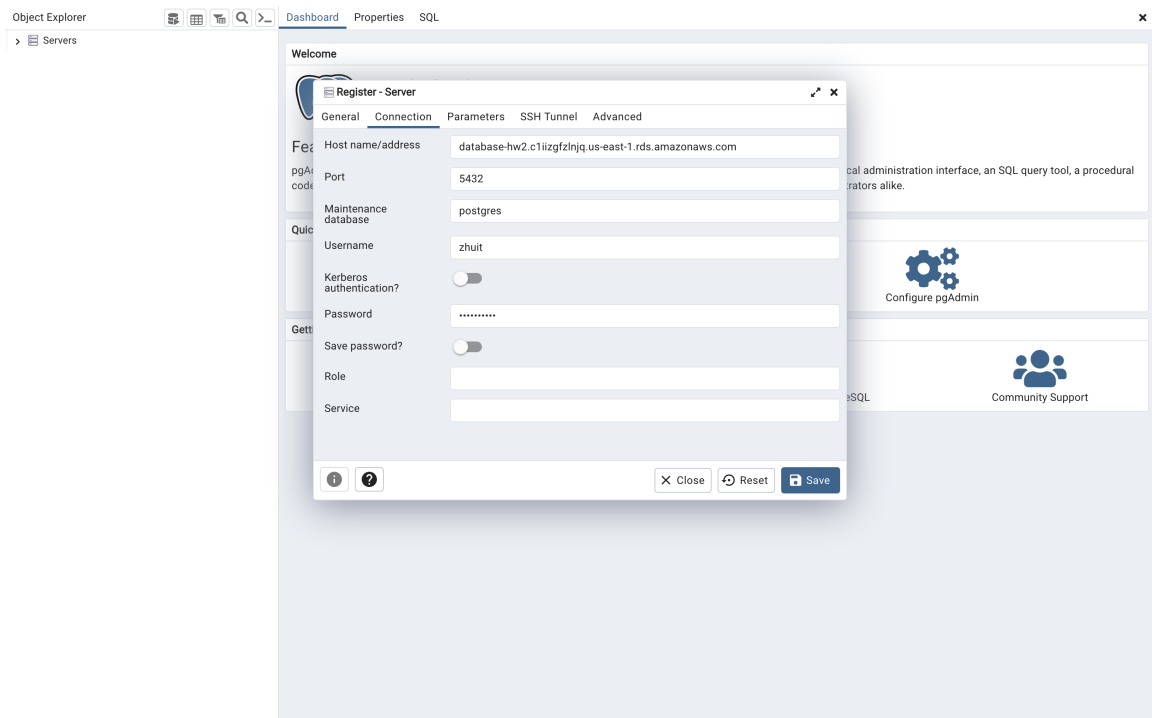
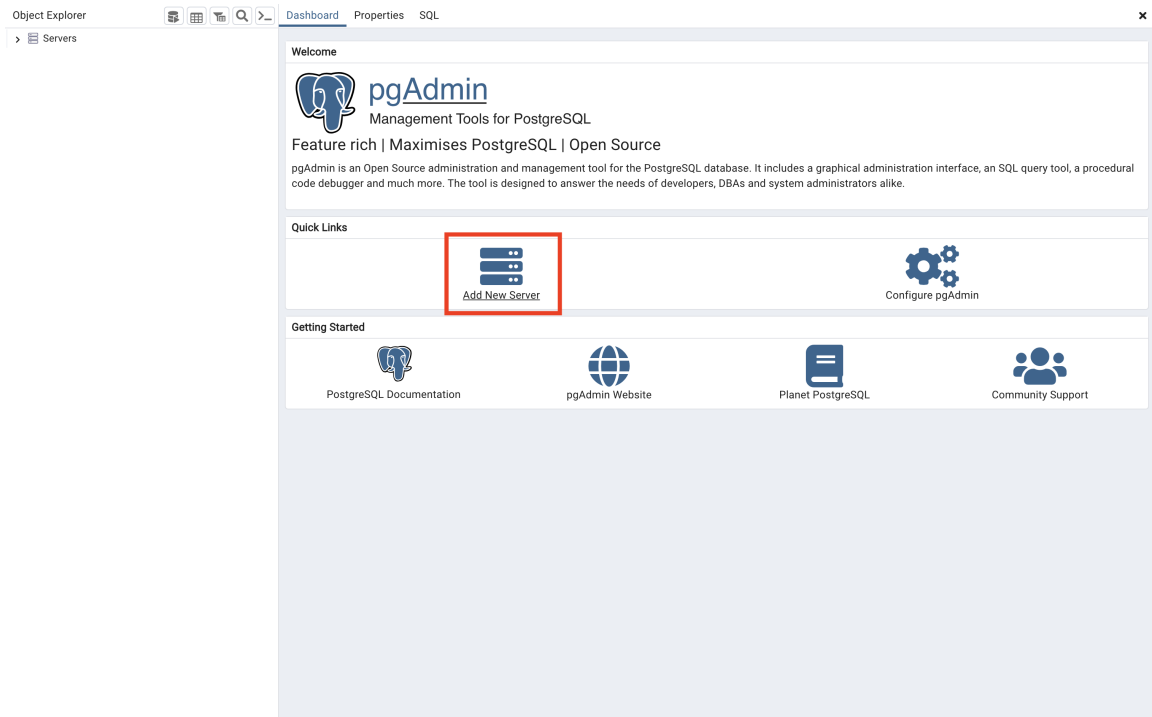
- Here we choose Many-to-Many cardinality constraint for “*Country_Continent*” relationship set. Because there are some countries that cross two continents, such as Turkey.
- “*Countries*” entity set has total participation in the relationship of “*Country_Continent*”. Because there exists a continent correspond to every country in “*Countries*” entity set. However, “*Continents*” entity set has partial participation. Because there doesn’t exist a country in the “*Countries*” entity set that is in Antarctica.

Part 2

- AWS RDS launch page

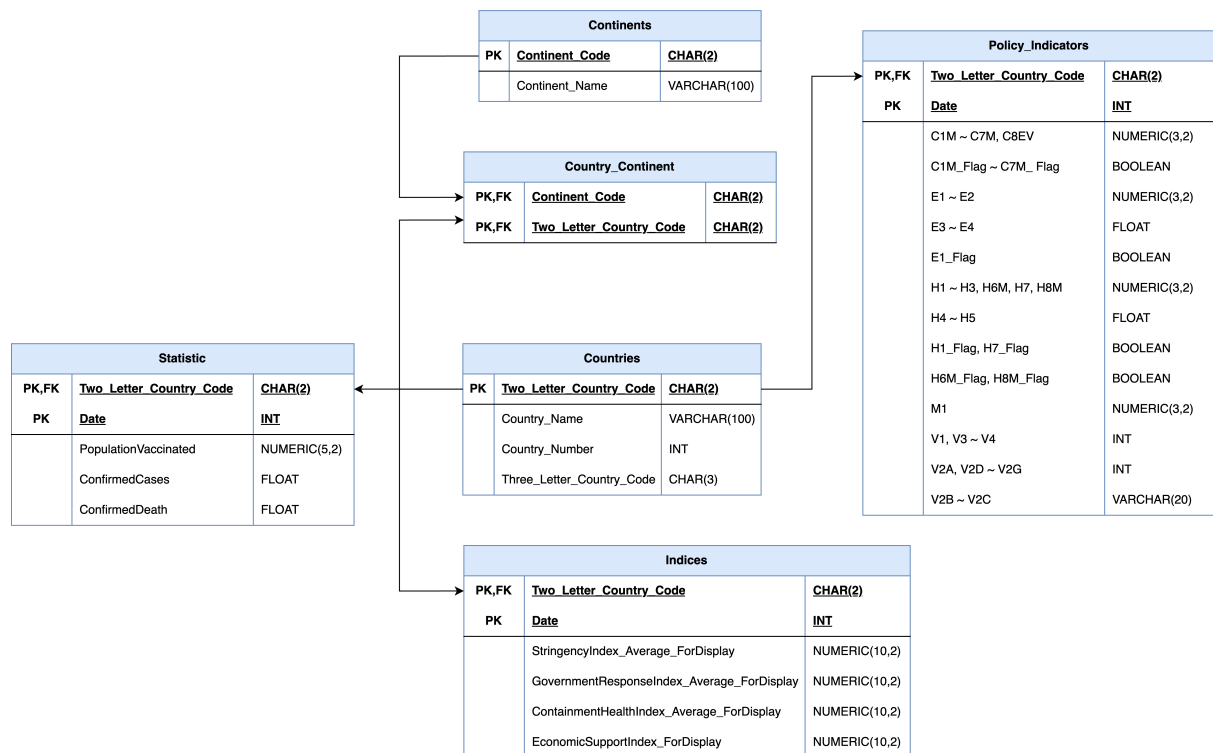


- Connect to the AWS RDS



Part 3

Schema Diagram



SQL queries to create table

▼ Create “Countries” Relation

```
CREATE TABLE public.Countries
(
    Two_Letter_Country_Code    CHAR(2),
    Three_Letter_Country_Code  CHAR(3),
    Country_Name                VARCHAR(100),
    Country_Number              INT,
    primary key (Two_Letter_Country_Code)
)
```

▼ Create “Continents” Relation

```
CREATE TABLE public.Continents
(
    Continent_Code              CHAR(2),
    Continent_Name               VARCHAR(100),
    primary key (Continent_Code)
)
```

▼ Create “Country_Continent” Relation

```

CREATE TABLE public.Country_Continent
(
    Two_Letter_Country_Code    CHAR(2),
    Continent_Code              CHAR(2),
    primary key (Two_Letter_Country_Code, Continent_Code),
    foreign key (Two_Letter_Country_Code) references Countries(Two_Letter_Country_Code),
    foreign key (Continent_Code) references Continents(Continent_Code)
)

```

▼ Create “*Policy_Indicators*” Relation

```

CREATE TABLE public.Policy_Indicators
(
    Two_Letter_Country_Code    CHAR(2),
    Date                       INT,
    C1M                        NUMERIC(3,2),
    C2M                        NUMERIC(3,2),
    C3M                        NUMERIC(3,2),
    C4M                        NUMERIC(3,2),
    C5M                        NUMERIC(3,2),
    C6M                        NUMERIC(3,2),
    C7M                        NUMERIC(3,2),
    C8EV                       NUMERIC(3,2),
    C1M_Flag                   BOOLEAN,
    C2M_Flag                   BOOLEAN,
    C3M_Flag                   BOOLEAN,
    C4M_Flag                   BOOLEAN,
    C5M_Flag                   BOOLEAN,
    C6M_Flag                   BOOLEAN,
    C7M_Flag                   BOOLEAN,
    E1                         NUMERIC(3,2),
    E2                         NUMERIC(3,2),
    E3                         FLOAT,
    E4                         FLOAT,
    E1_Flag                    BOOLEAN,
    H1                         NUMERIC(3,2),
    H2                         NUMERIC(3,2),
    H3                         NUMERIC(3,2),
    H4                         FLOAT,
    H5                         FLOAT,
    H6M                        NUMERIC(3,2),
    H7                         NUMERIC(3,2),
    H8M                        NUMERIC(3,2),
    H1_Flag                    BOOLEAN,
    H6M_Flag                   BOOLEAN,
    H7_Flag                    BOOLEAN,
    H8M_Flag                   BOOLEAN,
    M1                         NUMERIC(3,2),
    V1                         INT,
    V2A                        V,
    V2B                        VARCHAR(20),
    V2C                        VARCHAR(20),
    V2D                        INT,
    V2E                        INT,
    V2F                        INT,

```

```

V2G          INT,
V3           INT,
V4           INT,
primary key (Two_Letter_Country_Code, Date),
foreign key (Two_Letter_Country_Code) references Countries(Two_Letter_Country_Code)
)

```

▼ Create “Statistic” Relation

```

CREATE TABLE public.Statistic
(
    Two_Letter_Country_Code    CHAR(2),
    Date                      INT,
    PopulationVaccinated       NUMERIC(5,2),
    ConfirmedCases             FLOAT,
    ConfirmedDeath             FLOAT,
    primary key (Two_Letter_Country_Code, Date),
    foreign key (Two_Letter_Country_Code) references Countries(Two_Letter_Country_Code)
)

```

▼ Create “Indices” Relation

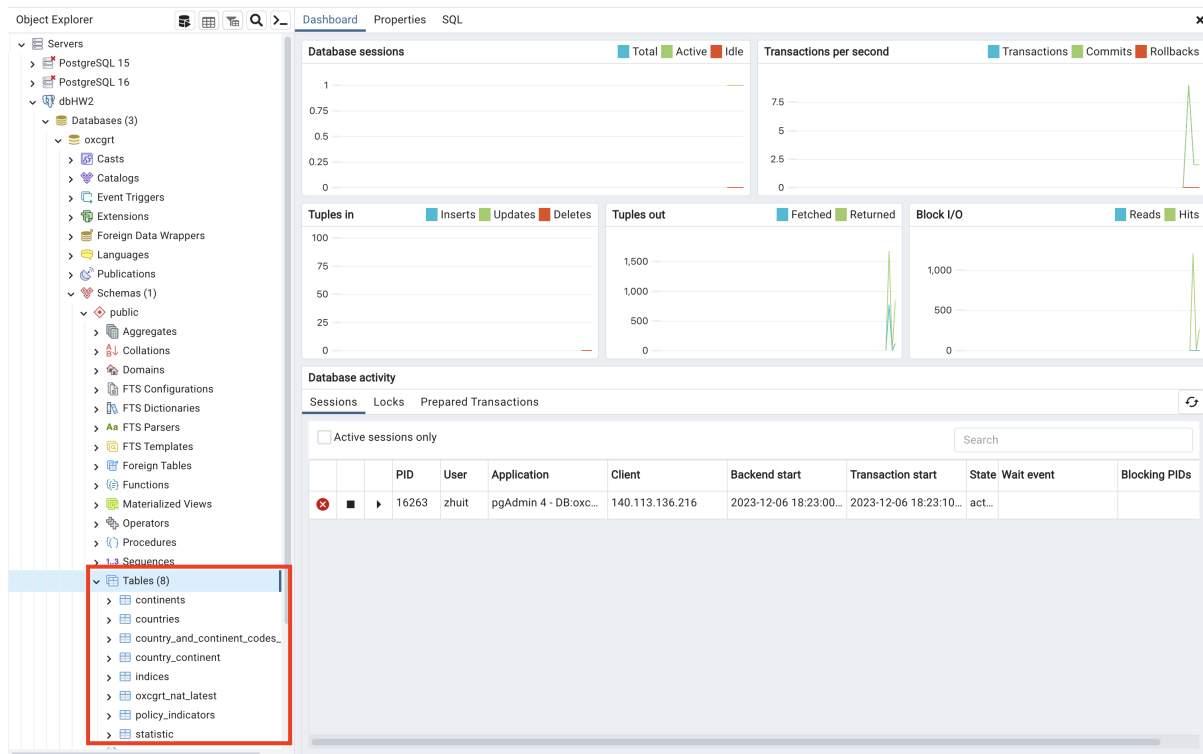
```

CREATE TABLE public.Indices
(
    Two_Letter_Country_Code    CHAR(2),
    Date                      INT,
    StringencyIndex_Average_ForDisplay NUMERIC(10,2),
    GovernmentResponseIndex_Average_ForDisplay NUMERIC(10,2),
    ContainmentHealthIndex_Average_ForDisplay NUMERIC(10,2),
    EconomicSupportIndex_ForDisplay NUMERIC(10,2),
    primary key (Two_Letter_Country_Code, Date),
    foreign key (Two_Letter_Country_Code) references Countries(Two_Letter_Country_Code)
)

```

ScreenShot

- “oxcgrt_nat_latest” and “country_and_continent_codes_list” are tables of original csv files.



Part 4

Test 1NF

- All datas from original csv file have no repeat group and all attributes are single value and atomic. So the original table are already 1NF.

In this part we need to test normal form of each tables, so we'll need to check functional dependency sets shown on the next part.

Countries

- Normal Form: **BCNF**
- Test Normal Form
 - Two_Letter_Country_Code is a candidate key for R.
 - Three_Letter_Country_Code is a candidate key for R.
 - Country_Name is a candidate key for R.
 - Country_Number is a candidate key for R.
 - So for all non-trivial functional dependencies $\alpha \rightarrow \beta$ in F^+ , α are super key.
 - Thus it's BCNF.

Continent

- Normal Form: **BCNF**
- Test Normal Form
 - Continent_Code is a candidate key for R.
 - Continent_Name is a candidate key for R.
 - So for all non-trivial functional dependencies $\alpha \rightarrow \beta$ in F^+ , α are super key.
 - **Thus it's BCNF.**

Country_Continent

- Normal Form: **BCNF**
- Test Normal Form
 - {Continent_Code, Two_Letter_Country_Code} is a candidate key for R.
 - So for all non-trivial functional dependencies $\alpha \rightarrow \beta$ in F^+ , α are super key.
 - **Thus it's BCNF.**

Statistic

- Normal Form: **BCNF**
- Test Normal Form
 - {Continent_Code, Two_Letter_Country_Code} is a candidate key for R.
 - So for all non-trivial functional dependencies $\alpha \rightarrow \beta$ in F^+ , α are super key.
 - **Thus it's BCNF.**

Indices

- Normal Form: **BCNF**
- Test Normal Form
 - {Continent_Code, Two_Letter_Country_Code} is a candidate key for R.
 - So for all non-trivial functional dependencies $\alpha \rightarrow \beta$ in F^+ , α are super key.
 - **Thus it's BCNF.**

Policy_Indicators

- Normal Form: **BCNF**
- Test Normal Form

- $\{\text{Continent_Code}, \text{Two_Letter_Country_Code}\}$ is a candidate key for R.
- So for all non-trivial functional dependencies $\alpha \rightarrow \beta$ in F^+ , α are super key.
- **Thus it's BCNF.**

Part 5

Countries

```
Two_Letter_Country_Code -> R
Three_Letter_Country_Code -> R
Country_Name -> R
Country_Number -> R
```

Continent

```
Continent_Code -> Continent_Name
Continent_Name -> Continent_Code
```

Country_Continent

```
{Continent_Code, Two_Letter_Country_Code} -> R
```

Statistic

```
{Two_Letter_Country_Code, Date} -> R
```

Indices

```
{Two_Letter_Country_Code, Date} -> R
```

Policy_Indicators

```
{Two_Letter_Country_Code, Date} -> R
```

Part 6

▼ SQL Query

```
WITH StringencyIndex(Date, Continent_Code, Maximum, Minimum) AS(
    SELECT
        Date,
        Continent_Code,
        MAX(StringencyIndex_Average_ForDisplay),
        Min(StringencyIndex_Average_ForDisplay)
    FROM
        Indices NATURAL JOIN Country_Continent
    WHERE
        Date = 20200401 OR Date = 20210401 OR Date = 20220401 OR Date = 20221201
    GROUP BY
        Date, Continent_Code
),
CountryCode_Index AS(
    SELECT
        S.Date,
        S.Continent_Code,
        S.Maximum,
        MaxCountry_Continent.Two_Letter_Country_Code AS Max_CountryCode,
        S.Minimum,
        MinCountry_Continent.Two_Letter_Country_Code AS Min_CountryCode
    FROM
        StringencyIndex AS S
        LEFT JOIN (Indices NATURAL JOIN Country_Continent) AS MaxCountry_Continent
            ON (S.Maximum = MaxCountry_Continent.StringencyIndex_Average_ForDisplay
                AND S.Date = MaxCountry_Continent.Date
                AND S.Continent_Code = MaxCountry_Continent.Continent_Code)
        LEFT JOIN (Indices NATURAL JOIN Country_Continent) AS MinCountry_Continent
            ON (S.Minimum = MinCountry_Continent.StringencyIndex_Average_ForDisplay
                AND S.Date = MinCountry_Continent.Date
                AND S.Continent_Code = MinCountry_Continent.Continent_Code)
    ORDER BY
        S.Date, S.Continent_Code
),
Country_N_Continent AS(
    SELECT
        Countries.Two_Letter_Country_Code,
        Countries.Country_Name,
        Continents.Continent_Code,
        Continents.Continent_Name
    FROM
        Countries NATURAL JOIN Country_Continent NATURAL JOIN Continents
)
SELECT
    S.Date,
    MaxCountryName.Continent_Name,
    S.Maximum AS Max_Stringency_Index,
    MaxCountryName.Country_Name AS Max_Country_Name,
    S.Minimum AS Min_Stringency_Index,
    MinCountryName.Country_Name AS Min_Country_Name
FROM
    CountryCode_Index AS S
    LEFT JOIN Country_N_Continent AS MaxCountryName
        ON (S.Max_CountryCode = MaxCountryName.Two_Letter_Country_Code
            AND S.Continent_Code = MaxCountryName.Continent_Code)
```

```

LEFT JOIN Country_N_Continent AS MinCountryName
ON (S.Min_CountryCode = MinCountryName.Two_Letter_Country_Code
AND S.Continent_Code = MinCountryName.Continent_Code)
ORDER BY
S.Date, MaxCountryName.Continent_Name

```

Output Results

▼ 2020/04/01

	date integer	continent_name character varying (100)	max_stringency_index numeric	max_country_name character varying (100)	min_stringency_index numeric	min_country_name character varying (100)
1	20200401	Africa	97.22	Congo	13.89	Burundi
2	20200401	Asia	100.00	Philippines	19.44	Tajikistan
3	20200401	Asia	100.00	Sri Lanka	19.44	Tajikistan
4	20200401	Asia	100.00	Jordan	19.44	Tajikistan
5	20200401	Asia	100.00	India	19.44	Tajikistan
6	20200401	Asia	100.00	Georgia	19.44	Tajikistan
7	20200401	Europe	100.00	Serbia	12.04	Belarus
8	20200401	Europe	100.00	Georgia	12.04	Belarus
9	20200401	North America	100.00	Honduras	15.74	Nicaragua
10	20200401	Oceania	96.30	New Zealand	40.74	Kiribati
11	20200401	South America	100.00	Argentina	57.41	Guyana

▼ 2021/04/01

	date integer	continent_name character varying (100)	max_stringency_index numeric	max_country_name character varying (100)	min_stringency_index numeric	min_country_name character varying (100)
1	20210401	Africa	96.30	Mauritius	8.33	Tanzania
2	20210401	Asia	85.19	Timor-Leste	16.67	Laos
3	20210401	Europe	87.96	Greece	36.57	Russia
4	20210401	North America	82.41	Honduras	13.89	Nicaragua
5	20210401	Oceania	62.04	Papua New Guinea	22.22	Kiribati
6	20210401	Oceania	62.04	Papua New Guinea	22.22	New Zealand
7	20210401	Oceania	62.04	Papua New Guinea	22.22	Vanuatu
8	20210401	South America	87.96	Venezuela	25.00	Bolivia

▼ 2022/04/01

	date integer	continent_name character varying (100)	max_stringency_index numeric	max_country_name character varying (100)	min_stringency_index numeric	min_country_name character varying (100)
1	20220401	Africa	56.48	Seychelles	11.11	Gabon
2	20220401	Asia	78.70	Myanmar	0.00	Mongolia
3	20220401	Europe	60.16	Ukraine	8.33	Andorra
4	20220401	North America	59.41	Dominica	8.33	Nicaragua
5	20220401	North America	59.41	Dominica	8.33	Dominican Republic
6	20220401	Oceania	85.19	Vanuatu	32.42	Fiji
7	20220401	South America	50.65	Suriname	14.82	Uruguay

▼ 2022/12/01

No data.

Part 7

- Update all NULL value in ConfirmedCases to 0.
- Replace the 0 moving average with 0.1.

▼ SQL Query

```
WITH Old_MovingAverage_Country AS(
    SELECT
        Date,
        Two_Letter_Country_Code,
        ConfirmedCases,
        (ConfirmedCases - LAG(ConfirmedCases, 7) OVER (
            PARTITION BY
                Two_Letter_Country_Code
            ORDER BY
                Date
        ) ) / 7 AS Moving_Average
    FROM
        Statistic
),
MovingAverage_Country AS(
    SELECT
        Date,
        Two_Letter_Country_Code,
        COALESCE(NULLIF(Moving_Average, 0), 0.1) AS Moving_Average
    FROM
        Old_MovingAverage_Country
),
OverStringencyIndices AS(
    SELECT
        I.Date,
        I.Two_Letter_Country_Code,
        C.Continent_Code,
        I.StringencyIndex_Average_ForDisplay / M.Moving_Average AS OverStringencyIndex
    FROM
        Indices AS I
        NATURAL JOIN MovingAverage_Country AS M
        NATURAL JOIN Country_Continent AS C
),
MaxMin_OverStringencyIndices(Date, Continent_Code, Maximum, Minimum) AS(
    SELECT
        Date,
        Continent_Code,
        MAX(OverStringencyIndex),
        Min(OverStringencyIndex)
    FROM
        OverStringencyIndices
    WHERE
        Date = 20220401 OR Date = 20210401 OR Date = 20200401 OR Date = 20221201
    GROUP BY
        Date, Continent_Code
),
CountryCode_Index AS(
    SELECT
```

```

O.Date,
    O.Continent_Code,
O.Maximum,
O.Minimum,
MaxCountryCode.Two_Letter_Country_Code AS Max_CountryCode,
MinCountryCode.Two_Letter_Country_Code As Min_CountryCode
FROM
    MaxMin_OverStringencyIndices AS O
LEFT JOIN OverStringencyIndices AS MaxCountryCode
    ON (O.Maximum = MaxCountryCode.OverStringencyIndex
        AND O.Continent_Code = MaxCountryCode.Continent_Code
        AND O.Date = MaxCountryCode.Date)
LEFT JOIN OverStringencyIndices AS MinCountryCode
    ON (O.Minimum = MinCountryCode.OverStringencyIndex
        AND O.Continent_Code = MinCountryCode.Continent_Code
        AND O.Date = MinCountryCode.Date)

),
Country_N_Continent AS(
SELECT
    Countries.Two_Letter_Country_Code,
    Countries.Country_Name,
    Continents.Continent_Name
FROM
    Countries NATURAL JOIN Country_Continent NATURAL JOIN Continents
)
SELECT
    S.Date,
    MaxCountryName.Continent_Name AS Continent,
    MaxCountryName.Country_Name AS Max_Country,
    S.Maximum AS Max_Over_Stringency_Index,
    MinCountryName.Country_Name AS Min_Country,
    S.Minimum AS Min_Over_Stringency_Index
FROM
    CountryCode_Index AS S
LEFT JOIN Country_N_Continent AS MaxCountryName
    ON S.Max_CountryCode = MaxCountryName.Two_Letter_Country_Code
LEFT JOIN Country_N_Continent AS MinCountryName
    ON S.Min_CountryCode = MinCountryName.Two_Letter_Country_Code
ORDER BY
    S.Continent_Code

```

Output Results

▼ 2020/04/01

	date integer	continent character varying (100)	max_country character varying (100)	max_over_stringency_index double precision	min_country character varying (100)	min_over_stringency_index double precision
1	20200401	Africa	Lesotho	907.3999999999999	South Africa	0.9176154992548434
2	20200401	Asia	Timor-Leste	750	Iran	0.01953100699844479
3	20200401	Europe	Monaco	25.655	Spain	0.010921194806146181
4	20200401	North America	Belize	525	United States	0.003199142418831577
5	20200401	Oceania	Tonga	935.1999999999999	Australia	0.19979983987189753
6	20200401	South America	Suriname	249.55	Brazil	0.12185427370387672

▼ 2021/04/01

	date integer	continent character varying (100)	max_country character varying (100)	max_over_stringency_index double precision	min_country character varying (100)	min_over_stringency_index double precision
1	20210401	Africa	Congo	472.2	Cameroon	0.014737062518695783
2	20210401	Asia	Tajikistan	287	India	0.0008874230797035569
3	20210401	Europe	Faeroe Islands	481.49999999999994	France	0.001785796728049861
4	20210401	North America	Greenland	370.4	United States	0.000849797703331448
5	20210401	Oceania	Fiji	490.7	Papua New Guinea	0.2168147778332501
6	20210401	South America	Suriname	13.313243243243244	Brazil	0.0009635592281575945

▼ 2022/04/01

	date integer	continent character varying (100)	max_country character varying (100)	max_over_stringency_index double precision	min_country character varying (100)	min_over_stringency_index double precision
1	20220401	Africa	Guinea	462.99999999999994	Botswana	0.002338608812776602
2	20220401	Asia	Macao	324.09999999999997	Mongolia	0
3	20220401	Europe	Faeroe Islands	111.1	France	0.00013628130182056675
4	20220401	North America	El Salvador	332.2	United States	0.0010650926731927328
5	20220401	Oceania	Kiribati	106.1375	Australia	0.000745880681107425
6	20220401	South America	Guyana	5.18	Brazil	0.0013755748352534015

▼ 2022/12/01

No data.