Instruction of the Data Wrangling Task

Data Sources

1. The numbers in these data sources are made-up instead of real-world statistics.
2. There are null values in the data. Please ignore any row in any data source that has a null value.
3. The range of years are 2015-2021

GDP Data

`gdp.db` is a sqlite3 database file. In table `gdp`, there are four columns:

- `Country`, text
- `State_id`, integer
- `Year`, integer
- `GDP`, real

Each row represents the GDP of that state in the given year. Example:

Country	State_id	Year	GDP
Switzerland	2	2018	110

This means that state #2 in Switzerland has 110 billion GDP in 2018.

Score Data

`score.csv` is a csv file with the following columns:

- `Country`
- `State_id`
- `Year`
- `Score`

Each row represents the "Happiness Score" of that state in the given year. Example:

Country	State_id	Year	Score
Switzerland	2	2018	1.5

This means that state #2 in Switzerland has a "Happiness Score" of 1.5 in 2018.

Region Data

`region.csv` is a csv file with the following columns:

- `Country`
- `Region`

Each row is a mapping between the country and region. Example:

Country	Region
Norway	Western Europe

This means that Norway is in Western Europe.

Some `Region` is "`-`". Please just keep them as-is. Many countries in the GDP and score data are actually missing in this table. Please left a null `Region` in the final compiled data table.

Instructions

1. Compute the GDP of each year of each country. It's the summation of the GDP in each state. If you're comfortable using SQL, please do this using SQL.
2. Compute the "Happiness Score" of each year of each country. It's the average of the score in each state.
3. Join the country-year level GDP, Score, and region data together into one data table with the following columns:

- `Country`
- `Region`

- `Year`
- `GDP`
- `Score`

For example, this row shows that Norway is in Western Europe, and in 2018, Norway has a GDP of 500 billion and a happiness score of 1.4

Country	Region	Year	GDP	Score
Norway	Western Europe	2018	500	1.4

4. Sort the rows in the following way:
    1. Firstly by ascending order of year. So the first chunk of rows should be 2015 data, next chunk should be 2016 data...
    2. Within each chunk, sort by descending order of happiness score.

Here is the illustration of the order:

Order
No.1 highest score in 2015
No.2 highest score in 2015
...
Lowest score in 2015
No.1 highest score in 2016
...

5. Save the result data table as a csv file. Submit the csv file and your code.