ETL Project

Group members: Zach, Shanna, Stephanie, Lucas

**Extraction**

The data used includes 3 data sets, each with their own Jupyter notebook:

* [The Economic Freedom Index](https://www.kaggle.com/lewisduncan93/the-economic-freedom-index) (Kaggle) – economic\_freedom\_index.ipynb
* [World Happiness Report](https://www.kaggle.com/unsdsn/world-happiness?select=2019.csv) (Kaggle) - happiness\_2019. ipynb
* [REST Countries](https://rapidapi.com/ajayakv/api/rest-countries) (API) – coutry\_info. ipynb

The two csv files from Kaggle contain data that was collected in 2019. The data that was pulled from the API is current data. However, we only selected columns that wouldn’t vary by year such as: region, subregion, area, and currency.

**Transformation**

Unnecessary columns were dropped from the data sets as in the example for the REST countries API found in “**country\_info.ipynb”**

Table, calendar

Description automatically generatedTable

Description automatically generated

Similarly, “**happiness\_2019.ipynb”** dropped “Overall Rank” from the columns using Pandas.

**Table

Description automatically generated**

**Table

Description automatically generated**

Columns were also cleaned to remove spaces for easier coding in SQL.

Text

Description automatically generated

The same methodology was repeated for each data source.

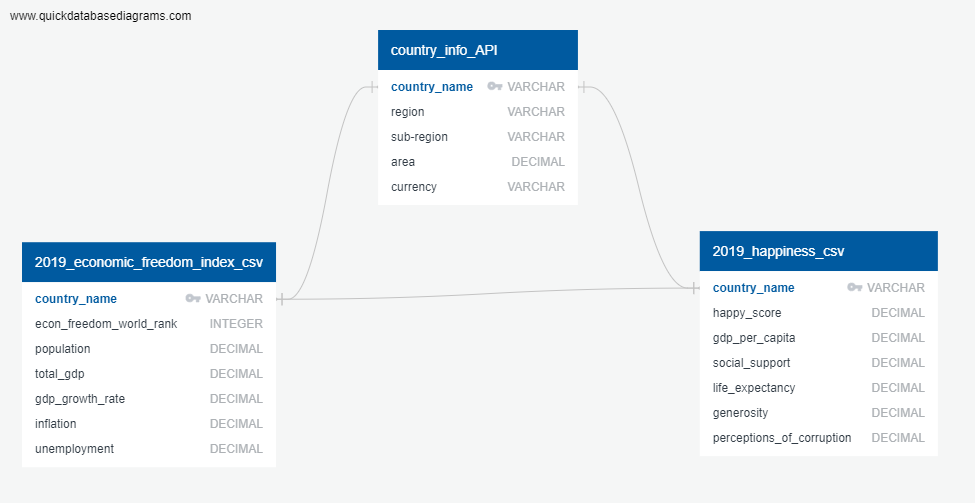
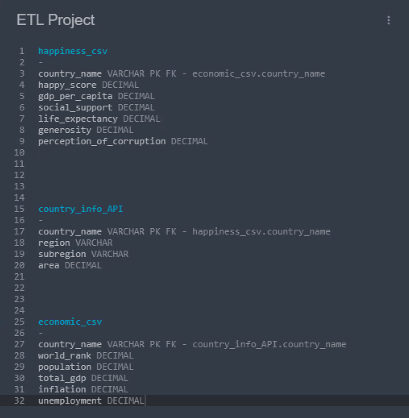
In the economic\_freedom\_index.ipynb, special characters were also removed as each data set was transformed to a csv and then loaded in Postgres.

Text

Description automatically generated

**Load**

Using Quick Database Diagram tool, we created our Schema. We used Juptyer notebook to clean and organize our data. Finally, we imported our cleaned data in to Postgress tables; Hapiness\_csv, country\_info\_API and economic\_csv. Assigning Primary and Foreign keys to country name to connect the tables.



**Challenges**

1. The original data source was on weather, however the weather API only allows to query 5 days of current temperature
2. Selected a different weather source, Meteostat, with the goal to use the capital city json file as a list to run a loop. However, the API does not allow for fetching lists effectively
3. Normalizing the data on country. We had created a 3rd table to normalize across happiness and population, as some key countries were being dropped (eg. Russia) due to naming differences. With the addition of a 3rd data source, this was more challenging. We chose to leverage list comprehension in Jupyter Notebook and drop the number of entries from 250 to 140 where all names were harmonized.

**Conclusion**

By merging these datasets, we could study:

* How unemployment affects happiness score
* How social support and access to healthcare affects life expectancy
* Does the economic freedom ranking have any correlation with happiness score?

Here’s an example of a query using Postgres SQL to run the operation:

1. Pull happy score, unemployment by country. This example pulls Albania.

Graphical user interface, text, application, Teams

Description automatically generated