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, and area.

**Transformation**

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

Table

Description automatically generatedTable, calendar

Description automatically generated

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

**Table

Description automatically generatedTable

Description automatically generated**

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

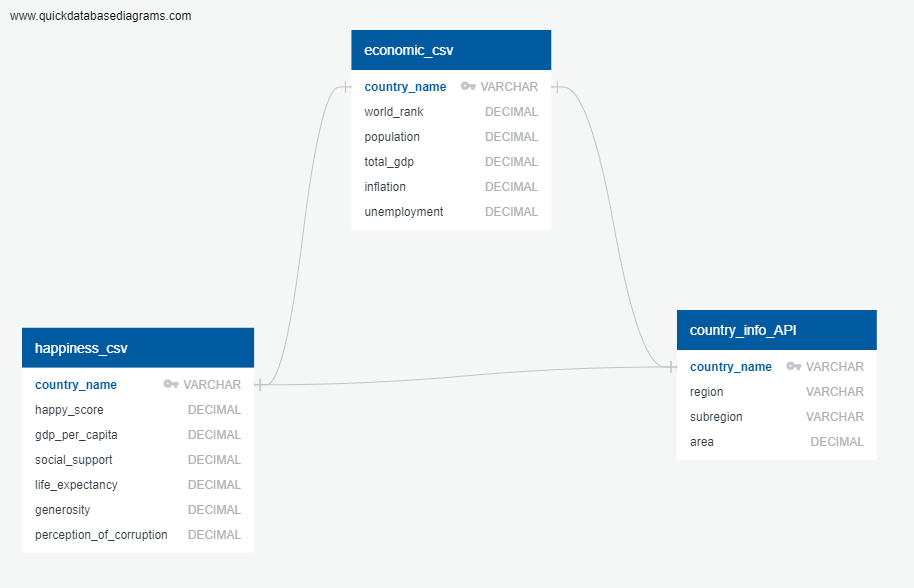
Text

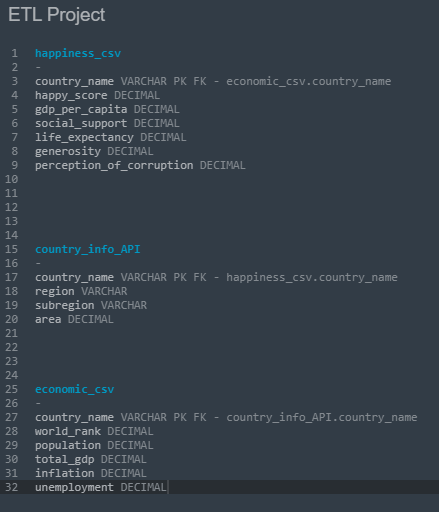
Description automatically generated

The same methodology was repeated for each data source.

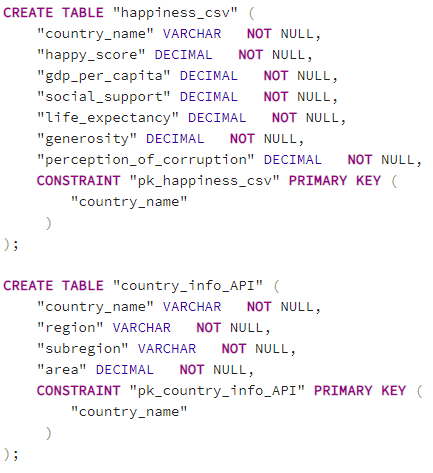
In the economic\_freedom index.ipynb, special characters were also removed as each data set was transformed to a csv and then loaded in Postgres.

**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.

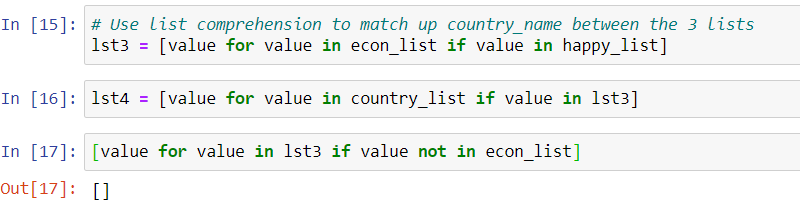
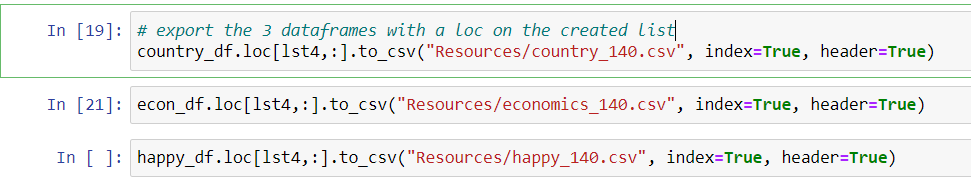


This is the table schema that was used to import the 3 csv files into postgres.

****

****

**Challenges**

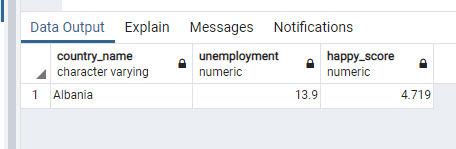
1. Our original idea was to compare a country’s happiness to its weather. First, we tried to use the openweather API. However, openweather only allows to query 5 days of current temperature.
2. Then, we selected a different weather source, Meteostat, with the goal to use the capital city json file as a list to run a loop. However, the API does not allow for fetching lists effectively. Also, it would have been difficult to compare weather on the country level. Therefore, we decided to compare country’s happiness with certain economic factors.
3. Normalizing the data on country. One dataset contained 186 countries, one contained 156 countries, and another contained 250 entries. We used list comprehension to match country\_name across the three datasets. We came up with a list of 140 countries. Then, we exported the data using the loc method. This is the final data that was put into our postgres server.

**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. Graphical user interface, text, application, Teams

   Description automatically generatedPull happy score, unemployment by country. This example pulls Albania.