|  |
| --- |
| Global Data on Happiness |
|  |

|  |  |
| --- | --- |
| Sept 16, 2019 | ETL - PROJECT |

**George Bigham & Dawn Brenner**

Global Data on Happiness

ETL - PROJECT

## INTRODUCTION

The purpose of this project was to build a project from the ground up using a typical flow of data in the workplace: 1) pull real data from various sources (extract), 2) clean and restructure it as needed (transform), and 3) write it into a database for storage (load). Extract, transform, and load - ETL.

## E - EXTRACT

We used the following data sources to gather interesting information about counties and happiness:

* World Happiness Report, Sustainable Development Solutions Network: This dataset includes happiness rankings and scores by country for 2015-2017. It also includes factors contributing to the happiness score. The data came in three separate CSV files, one per year.
* Suicide Rates Overview 1985 to 2016, compiled dataset by Rusty from various sources: This dataset includes the number of suicides per year per country by gender and age group. We were planning to include the Human Development Index (HDI) with our data, but found HDI already included in this dataset. This data came in a single CSV file. We thought this data interesting to include to see if happiness and suicide are as correlated as one would expect.
* Population, World Bank Population Data: This dataset includes the population per country per year. Population data allows users of the database to get the number of suicides per capital which is more meaningful than the number of suicides when comparing countries of various sizes and densities. This dataset was provided as a CSV file.

All CSV data was downloaded then converted to Pandas dataframes using pd.read\_csv.

*Lessons learned:*

* *There is a lot of interesting datasets available.*
* *Many dataset are a compilation of other datasets.*

## T - TRANSFORM

Python was used to clean and transform our data to achieve the data we wanted to upload to our database. The following was performed:

* World Happiness Report: This dataset required uploading multiple CSV files and appending them to the dataframe. **……..MORE??**
* Suicide Rates Overview 1985 to 2016: This dataset required aggregation of the data to determine a total number of suicides per country rather than separated by age and gender**……MORE??**
* Population: This dataset required dropping several rows, melting the columns into rows, and renaming a column. **……MORE??**
* Country: We created this dataframe to store the country\_id, country\_name, and region. Region was added to the table using a function that looked up countries in a dictionary and returned the associated region. **IS THIS RIGHT???**
* All Datasets: Many of the countries of the world were written differently in the different datasets. Because we were joining all of the tables by country, it was important all countries were represented consistently in the datasets. We created a lookup dataframe to perform this operation. The lookup table contained a set of names used for each country and a single country code for each set. For each row of the data tables, we grabbed the given country name, looked for it in the lookup table, and returned the country\_id to that country’s row of the data table. We appended the lookup dataframe with any alternative country names until all of the rows had a country\_id**. Is this right??**
* Anything else?

*Lessons learned:*

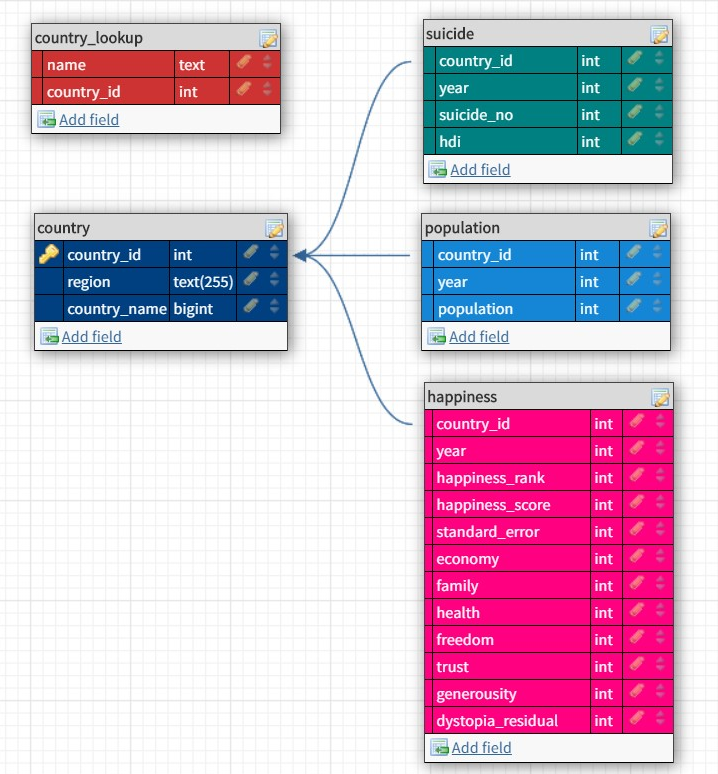
* *????*
* *There are more concerns when you know you have to get data into a relational database.*

## L - LOAD

Load: Load: the final database, tables/collections, and why this was chosen.

A relational database made sense for our data because we have several tables, the datasets were not overly large, and the database structure would enforce the use of consistent country names. **Anything else??**

We created the Entity Relationship Diagram (ERD) shown in the figure below and created the SQL code (schema.sql) to create the tables and relationships in our etlproject\_db. The end of our python script included a postgres connection to the etlproject\_db database and loaded each table to the database using the .to\_sql function.



*Lessons learned:*

* *????*
* *????*