**ETL Project: Sugar Analysis**

A picture containing indoor

Description automatically generated

By: Caitlan Beachey, Amaris Hassan, Cecilia Leung, Hillary Mandich, Kapir Pundhir

Date: December 19th, 2020

Contents

[**1. EXECUTIVE SUMMARY** 4](#_Toc59139590)

[**2. EXTRACTION** 5](#_Toc59139591)

[2.1 FIND THE OPTIMAL DATASETS 5](#_Toc59139592)

[2.2 DATASET SOURCES 6](#_Toc59139593)

[2.3 RELATIONSHIP OF DATASETS 6](#_Toc59139594)

[**3. TRANSFORMATON** 8](#_Toc59139595)

[**DATA CLEAN-UP FOR CSV FILES** 8](#_Toc59139596)

[3.1 REFORMATING COLUMN LABELS AND ROWS WITH ALL N/A 8](#_Toc59139597)

[3.2 DELETING UNWANTED COLUMNS 9](#_Toc59139598)

[3.3 COMBINING USEFUL INFORMATION FOR ANALYSIS 9](#_Toc59139599)

[3.4 ADD COUNTRY ID 10](#_Toc59139600)

[3.5 CONNECT DATA BY INSERTING COUNTRY CODE USING LOOKUP 10](#_Toc59139601)

[3.6 NORMALIZING 11](#_Toc59139602)

[**B. DATA TRANSFORM USING PANDAS** 12](#_Toc59139603)

[3.7 MELTING 12](#_Toc59139604)

[3.8 FORMATTING TABLES 13](#_Toc59139605)

[**4. LOAD** 14](#_Toc59139606)

[4.1 CREATE DB IN ELEPHANTSQL 14](#_Toc59139607)

[4.2 USE ERD DIAGRAM TO CREATE TABLES WITH ELEPHANTSQL DATABASE 14](#_Toc59139608)

[4.3 USING PANDAS TO CONNECT TO ELEPHANT SQL 15](#_Toc59139609)

[4.4 INSERT TABLE VALUES VIA PANDAS 16](#_Toc59139610)

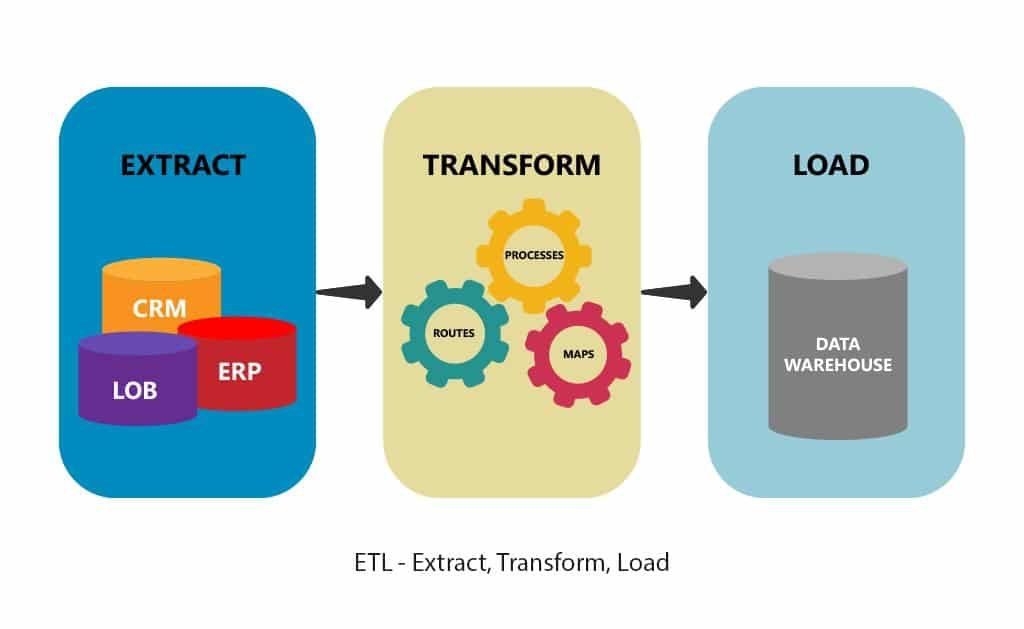
[4.5 CHECK IF TABLE IS LOADED ON ELEPHANTSQL 16](#_Toc59139611)

[**5. SUMMARY** 19](#_Toc59139612)

[5.1 LIMITATIONS 19](#_Toc59139613)

[5.2 NEXT STEPS 19](#_Toc59139614)

# **1. EXECUTIVE SUMMARY**



ETL stands for Extract, Transform and Load. Extracting involves collecting, reading, and migrating large volumes of raw data from various sources into one easily accessible database. Transformation makes data meaningful by reformatting, filtering, transposing, merging, and joining. Finally, loading uses various applications/software to load the data into the faster and produce faster results.

ETL provides numerous benefits as it breaks down data silos by gathering all relevant data into one database. Another advantage is it greatly assist the Data Analyst to analyze the data and turn it into business intelligence. Finally, with data readily available, it helps you to make better decisions in a timely manner.

In this proposal, we have conducted the ETL process through our Sugar Analysis Datasets. We will explain how we completed the extraction, transformation and loading the data in the SQL Web Server using various techniques and applications. Finally, we will provide a summary of the process that will explain why we did what we did, limitations, and next steps.

# **2. EXTRACTION**

**This process involves finding and collecting datasets from various types of sources.**

## 2.1 FIND THE OPTIMAL DATASETS

Extracting the right data sources is the key to success in the ETL process. Without, the right data, we can make the wrong analysis about our topic.

We looked at what datasets we had from Project 1 and then decided to add new datasets to enhance our sugar data analysis.

Note that all the existing datasets have been extracted, validated, and transformed from Project 1.

We used a total of 5 CSV Existing and New Datasets from Kaggle that came from various sources like WHO, FAO, World Bank and Data World. We were able to collect the data for at least 190 countries in each dataset.

The sources of our datasets are as follows:

## 2.2 DATASET SOURCES

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Source** | **Website** |
| Sugar Intake | FAO | https://www.kaggle.com/angelmm/healthteethsugar?select=sugar\_consumption.csv |
| Health Expenditure | WHO | https://www.kaggle.com/angelmm/healthteethsugar?select=healthexpend.csv |
| Income | Worldbank | https://www.kaggle.com/frankmollard/income-by-country |
| Obesity | WHO | https://www.kaggle.com/amanarora/obesity-among-adults-by-country-19752016?select=obesity-cleaned.csv |
| Country Codes | Data World | https://data.world/laurel/country-code-correspondence/workspace/file?filename=country\_codes.xlsx |
| Country Continents | Kaggle | https://www.kaggle.com/statchaitya/country-to-continent |

FAO = Food and Agriculture Organization

WHO = World Health Organization

## 2.3 RELATIONSHIP OF DATASETS

Before we proceed with any of the data clean up, we thought it would be a good idea to map out the relationship of the Databases. The perfect way is to map it out in an Entity Relationship Diagram (ERD) in http://www.quickdatabasediagrams.com.

Illustrating an ERD diagram is vital in database design because it:

* Increases understanding of how relationships and entities of the data.
* Decreases ambiguities and unnecessary processes.
* Outlines what applications and methods are needed to transform and load data.

See the below image to see how ERD of the Employee Databases.

**ERD Diagram:**

Diagram

Description automatically generated

# **3. TRANSFORMATON**

**Transformation is the process of converting the extracted data from its previous form into the form it needs to be in so that it can be placed into another database.**

## **DATA CLEAN-UP FOR CSV FILES**

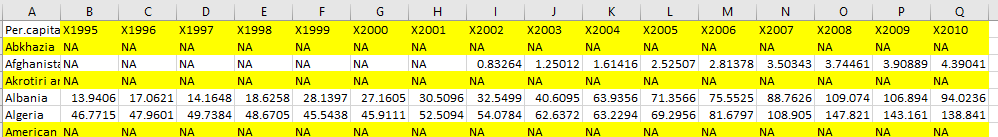
Data Cleaning involves reformatting the datasets to make sense of it. This could be removing irrelevant data, reformatting the data type, merging data, and looking up values to improve the data quality and usage before loading it up on the database. Having clean data is critical for decision making and data analysis.

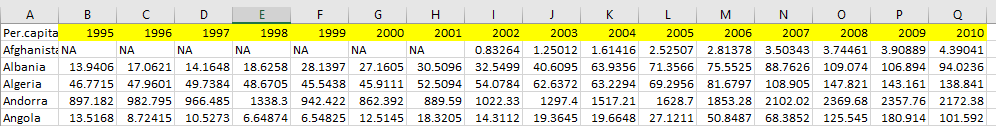
The below steps show what we have done to clean the data in the CSV files.

## 3.1 REFORMATING COLUMN LABELS AND ROWS WITH ALL N/A

**HEALTH EXPENDITURE TABLE**

* Remove the “X” infront of the year
* Remove the rows with #N/A

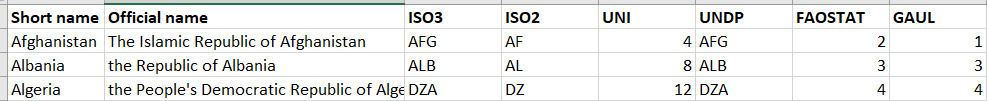




## 3.2 DELETING UNWANTED COLUMNS

**COUNTRY TABLE**

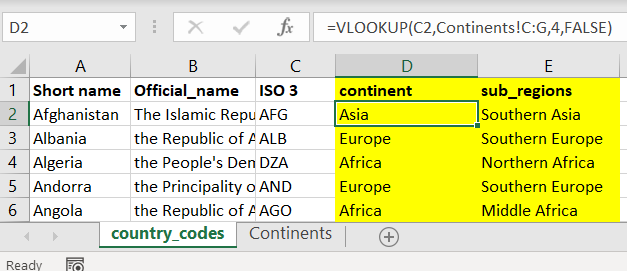
* Only one set of country code is necessary for performing data analysis.
* We only kept the ISO3 column and deleted the rest.



## 3.3 COMBINING USEFUL INFORMATION FOR ANALYSIS

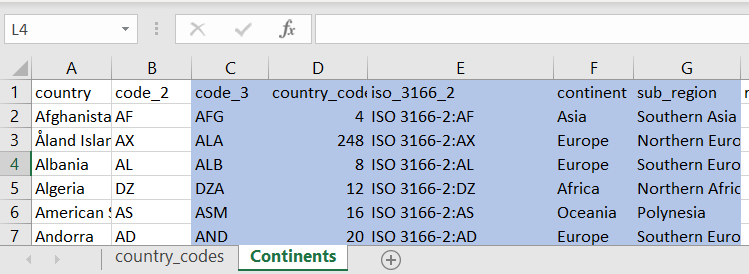
**COUNTRY TABLE AND CONTINENTS TABLE**

* Since the information have similar content, it will be better to combine the countries, continent and sub-regions into one table.
* Open the Continents table, copy and move tab to country table.
* Add a continents and sub\_region column beside the ISO3 column on the leftmost side of each of the 4 tables.
* Conduct a LOOKUP to obtain the country ID for each country.



**Continents Table (Lookup Table)**

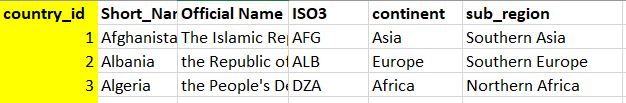
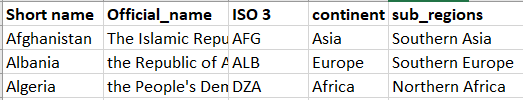
**Country Table (Insert Lookup Formula)**



## 3.4 ADD COUNTRY ID

**COUNTRY TABLE**

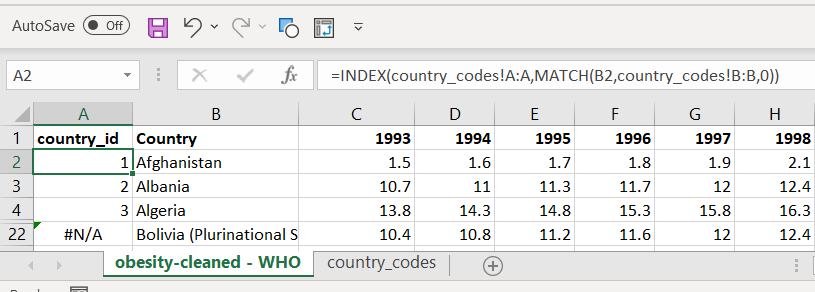
* Country code will be inserted in the first column of the Country table to identify each country as a unique value.
* First row will be 1, second row will be 2, and so forth.



## 3.5 CONNECT DATA BY INSERTING COUNTRY CODE USING LOOKUP

**INCOME TABLE, OBESITY TABLE, SUGAR CONSUMPTION TABLE, HEALTH EXPENDITURE TABLE, COUNTRY TABLE**

* Open the country table, copy and move tab to each of the 4 tables.
* Add a country\_id column on the leftmost side of each of the 4 tables.
* Conduct a LOOKUP to obtain the country ID for each country.
* Cannot have #N/A for country ID
  + If there is #N/A, determine if the country names on the respective tables match the country table..
* Upon completion, copy & paste value of country\_id and remove country\_codes tabs in the 4 tables
* Repeat this process for the Health Expenditures, Sugar Intake and Income tables



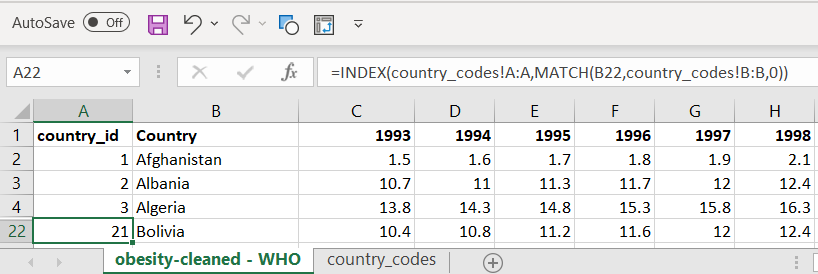
**Obesity Table (Insert INDEX MATCH Formula on first column)**

**Country Table (Lookup Value)**



**Obesity Table (Get rid of #N/A by matching country name)**

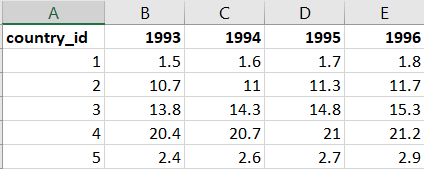
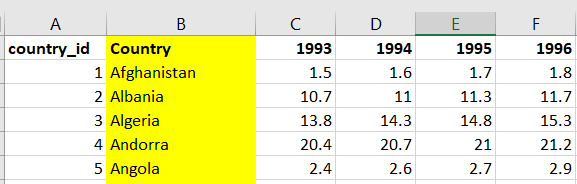
**Country Table (Lookup Value)**



## 3.6 NORMALIZING

**INCOME TABLE, OBESITY TABLE, SUGAR CONSUMPTION TABLE, HEALTH EXPENDITURE TABLE**

* Since we have a country code as a unique identifier for each table, we do not need to have all the country names that may take up storage space.
* The country code will be the primary linkage to all datasets.
* Repeat this process for the Health Expenditures, Sugar Intake and Income tables



## **B. DATA TRANSFORM USING PANDAS**

After cleaning the data for the CSV files, we are ready to upload them on Pandas to do further data transformation.

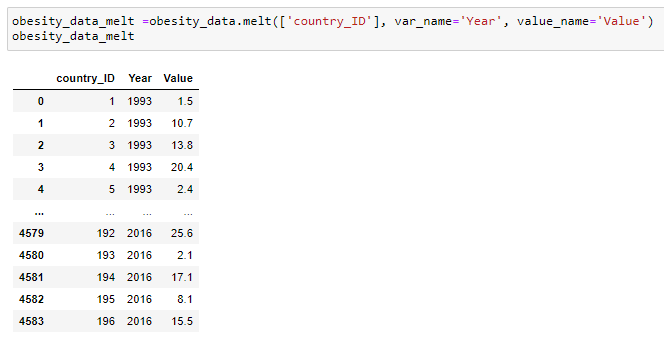
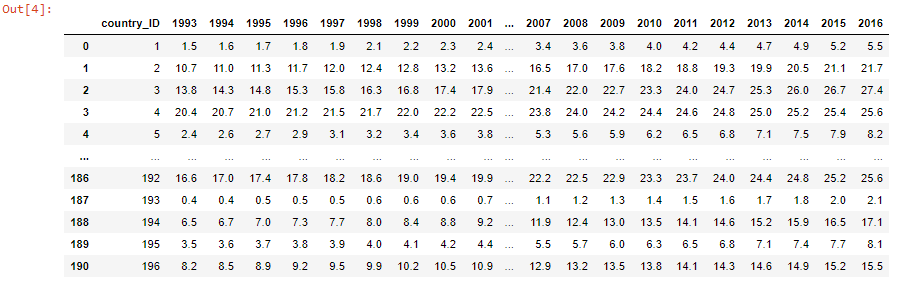
There are total of 5 Jupyter Notebooks to read and transform the tables for the 5 tables. Which is Country, Income, Obesity, Sugar Consumption and Health Expenditures.

These include using the melt function to set the year and values as 1 column, formatting the N/A and .. fields and taking them out, and finally putting the country codes as an index value

## 3.7 MELTING

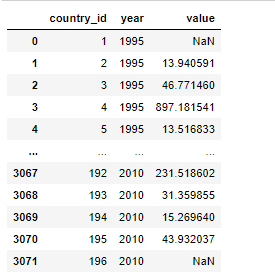
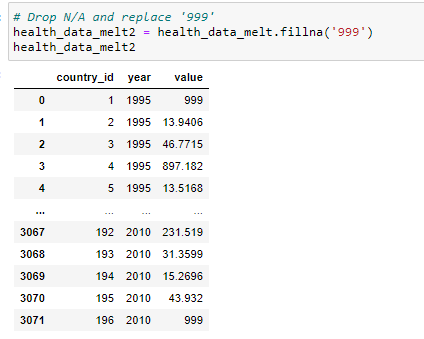
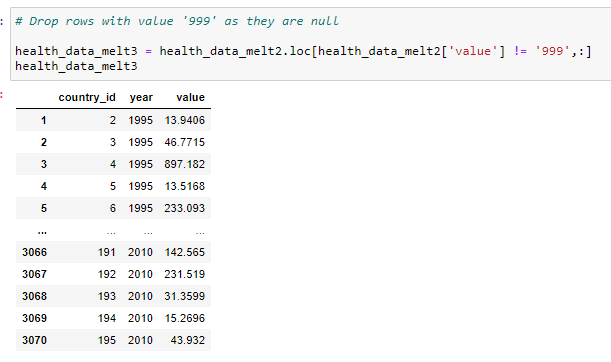
**INCOME TABLE, OBESITY TABLE, SUGAR CONSUMPTION TABLE, HEALTH EXPENDITURE TABLE**

* Ultimately, we want to have all the values in one column to compare the factors that contribute to sugar intake.
* To do this, we need to transpose the data using melt method.

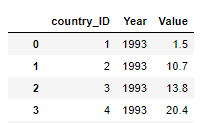
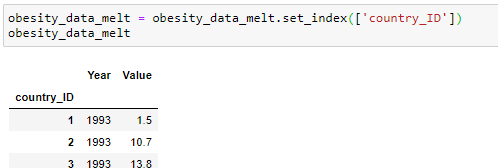


## 3.8 FORMATTING TABLES

* DROP N/A Values to conserve memory space for Health and Income Table.



* Index country\_codes for all tables as indexing plays a role in reshaping the DataFrame.



# **4. LOAD**

**Load is the process of writing the data into the target database.**

The below how we established a connection between Pandas and ElephantSQL to load the transformed tables.

## 4.1 CREATE DB IN ELEPHANTSQL

* Log into Elephant SQL (www.elephantsql.com)
* Create New Instance
* Input name for your database.
* Click Region
* Click Review

## 4.2 USE ERD DIAGRAM TO CREATE TABLES WITH ELEPHANTSQL DATABASE

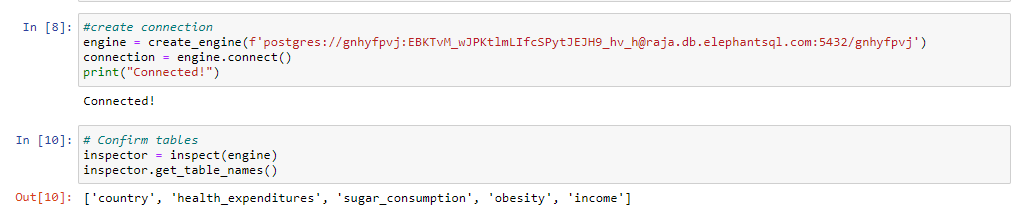
* Go to quick database to export ERD diagram in \*.sql format
* Save the file
* Click into the instance created
* Click Browser window
* Copy and paste the sql file into the box where you enter SQL query.
* Hit Enter

## 4.3 USING PANDAS TO CONNECT TO ELEPHANT SQL

* Open a new Jupyter notebook and run all the other 5 notebooks and connect to the database.

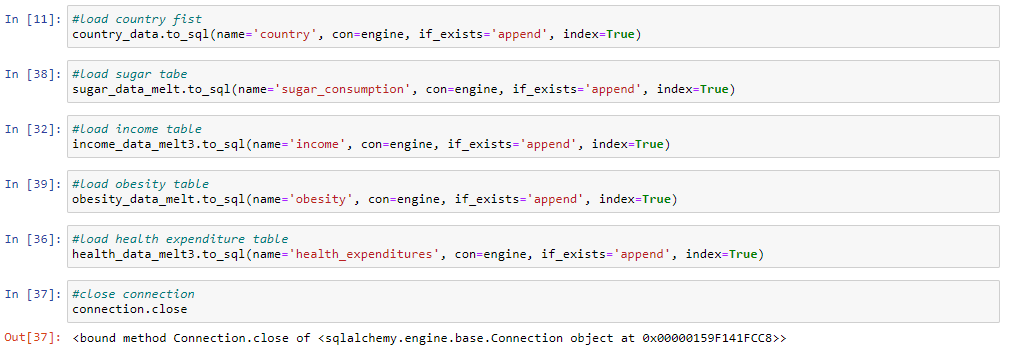


* Establish a connection to connect to our database.



## 4.4 INSERT TABLE VALUES VIA PANDAS

* Load panda Dataframes into ElephantSQL after connection have been established.

****

## 4.5 CHECK IF TABLE IS LOADED ON ELEPHANTSQL

* The below shows that the Country, Health Expenditures, Obesity, Income and Sugar Consumption Tables are successfully loaded into ElephantSQL.

Country Table:

Graphical user interface, application

Description automatically generated

Health Expenditure Table:

Table

Description automatically generated

Obesity Table:

Graphical user interface, application, table

Description automatically generated

Income Table:

Table

Description automatically generated

Sugar Consumption Table:

Graphical user interface, application, table

Description automatically generated

# **5. SUMMARY**

## 5.1 LIMITATIONS

* Since all datasets have different formats, creating uniform transformation can be difficult.
* As we use more complex datasets, it can be potentially more difficult to normalize data.
* If a minor spelling mistake is made in the code then the program will fail.
* ElephantSQL can only handle 5 connections at a time. In a group of 5, if multiple connections are running through different notebooks, then we will pass the free version limit.
* Only magic notebook can have an active connection.

## 5.2 NEXT STEPS

* Continue to build quantity and quality of datasets for further analysis.
* Analyze the correlation between health expenditures and sugar intake.
* Analyze the sugar intake by continents and regions.
* Create an HTML project page with Sugar Analysis.
* Web Scrape Diabetes Information