**ETL Project**

*By Akuete Kpodar, Blake Skinner, Olesea Armasari, Sherwin Macalintal.*

**Extraction**

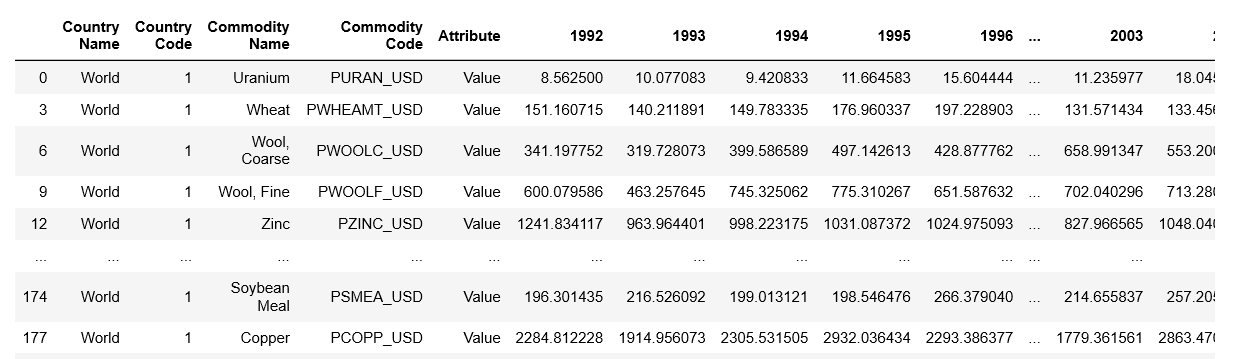
We used 3 datasets from the public platform Kaggle and Data World. All of our data was based on countries of the world ranging over various years from 1992 to 2012. These were the most recent ones we could find. The sources for our dataset are as follows:

* FAO (The Food Balance) from Kaggle.
* Global land temperature by Country from Data World.
* IMF Primary commodity prices from Data World (UN Comtrade Database).

**Transformation**

Our first steps in cleaning up the datasets involved figuring out which variables were not relevant.

For the Price Point Dataset (Figure 1), we dropped the columns related to countries that did not have any values in the last 20 years ( 1992-2012), we dropped the years prior to 1992 because they were irrelevant to our data tables.



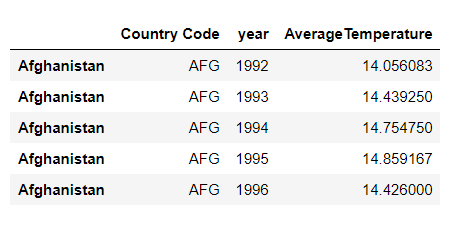
**Figure 1:** Dropped unnecessary columns.

For the agriculture dataset and the temperature dataset, we dropped the unnecessary columns related to the countries and products that would not be of any value for the purposes of analysis. We limited our datasets to the years of 1992 – 2012 to illustrate the most complete datasets from different data sources.

 **Figure 2:** Illustrates dropping unnecessary columns and indexing the agriculture dataset to country code.

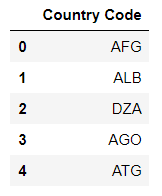
Using pandas and data frame manipulation we mapped key value pairs through the .to\_dict function to create new csv’s for the purpose of our SQL queries.

Through mapping we were able to create clean data sets indexing country codes to country names. We proceed to drop any nulls in our data sets.



**Figure 3:** Illustrates a clean temperature data sets where all the nulls were dropped.

To further clean the data, we renamed column headers to universally match from our different data sources.



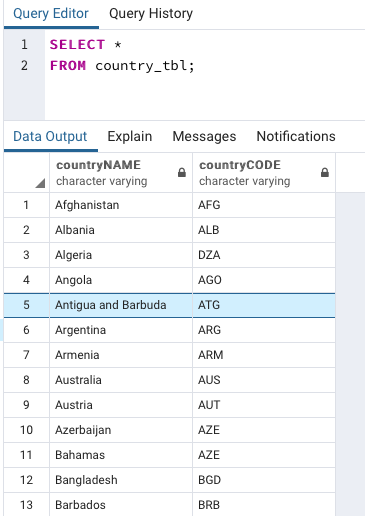
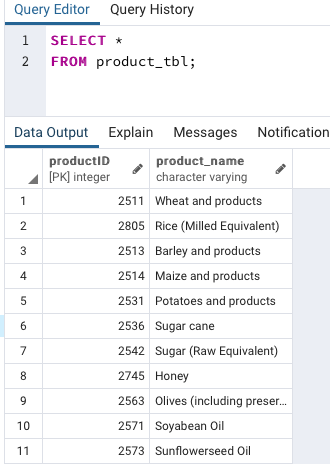
**Load**

We created the database named Agri\_and\_Temp\_db in Postgres. We utilized the GUI to create the five tables with their corresponding columns as shown on our Entity Relation Diagram. After creating the tables, we imported the csv’s of the clean datasets into each table and verified successful loading through a SELECT \* query shown on the screenshots provided.

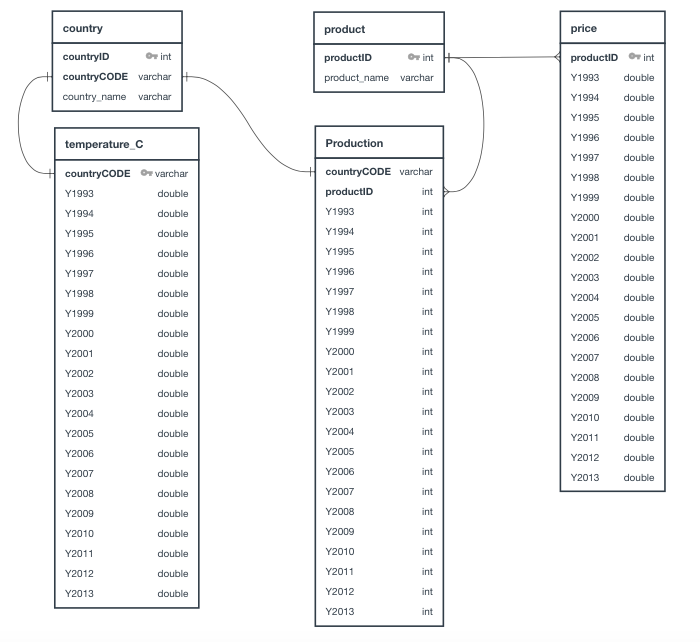
Our tables are:

1. country\_tbl, serves as the reference for countries and their corresponding codes.
2. product\_tbl, serves as the reference for the products and their respective
3. IDs.production\_tbl,  has the quantities of each product categories
4. temperature\_tbl, has the average yearly temperatures in Celcius
5. price\_tbl, the yearly average world price (in USD) of each product category. Unfortunately not all products on product\_tbl exist in price\_tbl.

The last three tables have values from 1992 to 2012.

**Figure 4:** country\_tbl **Figure 5: product\_tbl**

****

**Figure 6:** Entity Relationship Diagram

A screenshot of a cell phone

Description automatically generated

**Figure 7:** production\_tbl

**A screenshot of a cell phone

Description automatically generated**

**Figure 8:** temperature\_tbl

A screenshot of a cell phone

Description automatically generated

**Figure 9:** price\_tbl

**Summary:**

The final output will provide the user ways to analyze trends and relationships (if any exists) between:

1. average temperature with the agricultural products production by country or globally,
2. average temperature with the agricultural products price, globally and
3. individual agricultural products production with their prices or vice-versa.

Then use those findings to create a mathematical model to forecast values of agricultural production, temperature or price in the future.