Marc Palomo
Maralee Capella
ETL Project
5-13-19

EXTRACT

For our ETL project, we were interested in investigating healthcare trends. We went to the World Health Organization (WHO) website. Once on the WHO site, we navigated to the Global Health Observatory (GHO) data repository. We searched through a lot of different data tables for various healthcare information sets. Eventually, we determined most tables had overlapping countries and years; however, we tried to find 2 specific tables that would have large overlaps. We downloaded two CSV files titled “Reported deaths data by country (malaria deaths)” and “Medical doctors.” The links are included at the bottom of this section. We viewed both CSVs to determine that both datasets contained both overlapping variables in large quantities. We found that both datasets contained data for country and year despite not completely overlapping on every country or every country and year.

Reported deaths data by country (malaria deaths)
http://apps.who.int/gho/data/node.main.A1367?lang=en

Medical doctors
http://apps.who.int/gho/data/node.main.HWFGRP_0020?lang=en

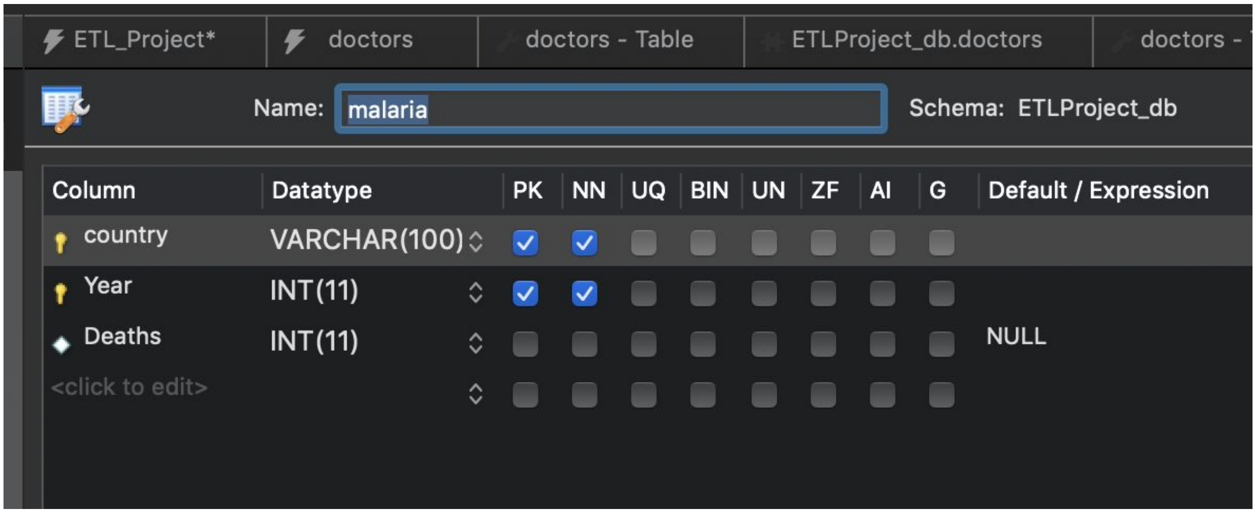
TRANSFORM

We used Pandas and Jupyter notebook to read and transform our two CSVs into dataframes. Then, we determined the years for which we had overlapping data. The “malaria deaths” CSV only contained data from 2000 forward, so we selected only the years equal to or greater than 2000 in the “Medical doctors” CSV. Out of curiosity we ran len() functions to determine the number of unique countries in both files. After, we selected the columns of interest and saved as new dataframes. We also renamed the columns to desired names so that they would match on both dataframes. Additionally, we changed the index to “country.” We wanted to transform the datasets into the same shape. We originally attempted to accomplish this task using Pivot Tables in Pandas. However, we ran into trouble when we wanted to turn the years from headers to a column. We found a simpler way to accomplish the same task by utilizing the melt function. This solution allowed us to use the original shape of the “Medical Doctors” table. We transformed the year headers from the “Malaria Deaths” dataframe into a column in order to match the “Medical Doctors” dataframe. This set us up for success when uploading the data and anticipating joins based on country and year.

LOAD

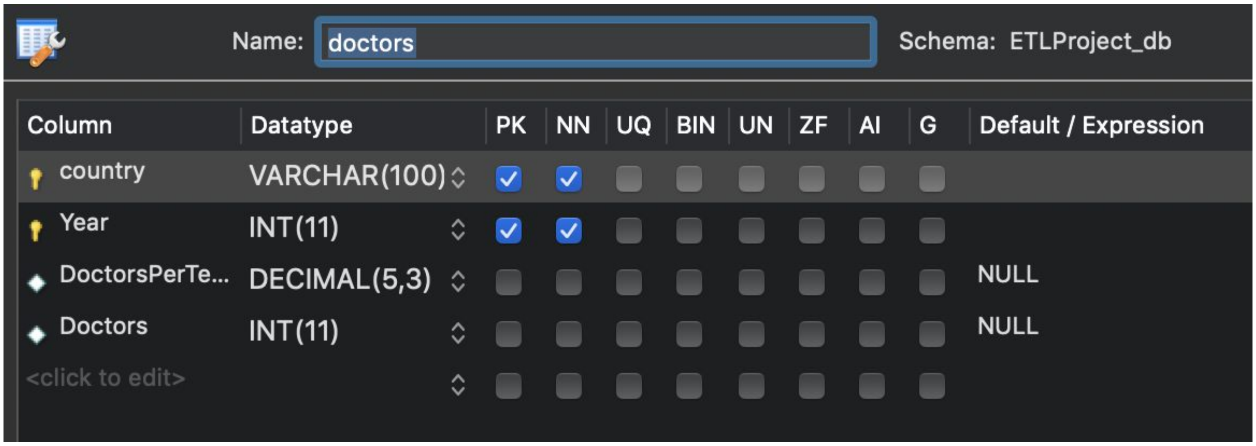
We used *create_engine* from sqlalchemy to create a database connection in order to have a relational database to contain our data. First we had to create the database in MySQL and add both of the tables with the appropriate columns. Once created, we checked that the table names (doctors, malaria) were correct and showing up in jupyter notebook. Once we were confident, we added the dataframes from Pandas into the database in MySQL. In the event of future data we wanted to make sure any future dataframes using our code would append the tables and not duplicate the database in MySQL.

In order to complete this non-duplication in MySQL, we wanted to improve on what we added by altering the tables with a composite primary key. A composite primary key, we learned, takes two columns and allows them to both function as the primary key. While we don't have a separate primary key column, we've successfully made the country and year in both tables the composite primary key. In the future, new data should only upload with new country and year combinations and should update any existing data.



Name: malaria Schema: ETLProject_db

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
country	VARCHAR(100)	☑	☑	☐	☐	☐	☐	☐	☐	
Year	INT(11)	☑	☑	☐	☐	☐	☐	☐	☐	
Deaths	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
<click to edit>		☐	☐	☐	☐	☐	☐	☐	☐	



Name: doctors Schema: ETLProject_db

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
country	VARCHAR(100)	☑	☑	☐	☐	☐	☐	☐	☐	
Year	INT(11)	☑	☑	☐	☐	☐	☐	☐	☐	
DoctorsPerTe...	DECIMAL(5,3)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Doctors	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
<click to edit>		☐	☐	☐	☐	☐	☐	☐	☐	

Possible future uses of the datasets

We were interested in joining these two sets of data to determine if there may be a correlation between the number of doctors in each country and the number of deaths attributed to malaria in each country. In addition to the cleaning and transforming above, we also performed queries to determine for which countries there were NULLs and countries that were not present in both datasets.

Using the same datasets and joining them with other health data (such as deaths from tuberculosis or HIV), we could further explore a possible correlation between number of doctors and deaths attributed to various diseases.