# Design Decisions

For the most part, our data was relatively easy to clean and get organized. Most of the interesting decision decisions that we made revolved around foreign key constraints, specifically the foreign keys that referenced the country attribute from the CountryInfo table.

Since country was used as a foreign key, that meant that every country from the tables that referenced the country attribute in CountryInfo must be found in the CountryInfo table. If a table row (from any table other than CountryInfo) attempted to be inserted while referencing a country that didn’t exist in CountryInfo, an error would occur upon insertion since the foreign key constraint did not hold. One solution we had was to add all countries that were found in the other tables that were not originally in CountryInfo, to the CountryInfo table. This way, every country would be able to have a proper reference for its foreign key and stop having errors.

However, this outcome led to another decision we had to make. Since these newly added countries were not originally in the CountryInfo table, we only had the country name for the new rows, and none of the other attributes. All attributes in the CountryInfo table were originally necessary upon input (all columns had not null constraints). We then had to decide whether to change our original decision, and remove all occurrences of countries that were not originally found in CountryInfo from all the other tables, or if instead we should remove the not null constraints from the CountryInfo columns that would be empty for those new rows.

After looking at the questions that we had decided to explore with our database, we realized that the other information in CountryInfo was only required for one of the three of our questions, whereas the different countries in the various other tables would be used for all of the questions. This lead us to decide to remove the not null constraints on the CountryInfo attributes, (besides the primary key, aka the country attribute, which always needs to / will always be not null anyways) since it allowed us to include more countries for all of our questions of interest, instead of cutting down our countries to only benefit one of our questions of interest.

Another point of interest that we came across again had to do with the country foreign key constraint. We quickly noticed that even though a country name was in the CountryInfo table, when inserting a new row into another table who referenced the country in CountryInfo, it would say that the foreign key could not be found. This occurred in cases where the country attribute for USA was “United States” in CountryInfo, meanwhile in another table like CovidEffects, the country attribute for USA was “United States of America”. Every human is able to recognize that both of these names have the same meaning, but to the database, it obviously was unable to identify both attributes as being the same and was unable to find “United States of America” in CountryInfo. This led us to editing all instances of USA in other tables to convert the name into “United States”, so that the foreign key would hold properly.

Overall, the data cleaning itself wasn’t too difficult and more just time consuming. We already had all of our data in csv files, so it was just a matter of choosing the columns that we wanted for each table, putting them in their own csv files, making the small changes to the CountryInfo table (to add countries that didn’t originally exist) and fixing the country names in all tables to match properly to be able to reference the existing / proper foreign key.