We downloaded data from the WorldBank database, using only the variables used in the Schema for every country.

Because the data downloaded in the csv file was organized in such a way that would make it difficult to load into each table in the schema using "\COPY", we downloaded data into separate .csv files for each table.

While downloading the .csv files, we had to determine if there are attributes that were simply not available in the WorldBank database. We noticed that age group, race and gender as pertains to the variables we are examining are not available. Therefore, we decided to remove all agegroup, race, gender, agegroupgettingkilled and agegroupkilling variables from all the tables that contained them.

We noticed that for all the variables, the data was not available for every year. For some of the variables, the data was not available for most years. Furthermore, the way the data is organized in the .csv files would make it difficult to implement a year attribute in the tables. Therefore, we used the year 2010 as it had the most data available, and the year 2019 for populations, being the most recent year. The Schema was modified so that year is not an attribute in any of the tables.

To ensure the data loads faster given primary key and not null constraints, we cleaned the .csv files manually to ensure that there are no null value, but also because the absence of a value for an attribute in the .csv file was represented with '..', which is technically not a null value, yet could lead to a type error and nonsensical data values. We removed any nonsensical or null values from the table. We also removed data from the table that would violate the foreign key constraints – one particular value for country was present in multiple tables that was not in the Country table, and the corresponding row was manually removed.

Next, we had to ensure that the types for each attribute specified in each table align with the values present in the data we downloaded. For example, if a particular attributed is listed as an 'int' we had to ensure that it is indeed presented in integer format in the csv files for that attribute, and change accordingly if it isn't.

One of the first observations we made is that each country had a code. For example, Afghanistan's code is 'AFG'. The code is always a string of 3 uppercase letters for every country. Therefore, we changed cID in the Country table from an int to char(3).

The data we were looking for was often available in more than one form, or using more than one measurement. We changed income tax rate to tax revenue as income taxes relative to GDP were not available. Educational attainment data was available at different levels, so we used lower secondary. GDPPerCapita is measured in 2010 US $ and government expenditure and debt is expressed as percentage of GDP.

We also downloaded GDP per capita measured in various ways (ie., constant US, current US, etc) and noticed that GDP per capita in constant 2020 US $ was the most commonly available GDP per capita data, so we used that to construct the GDPPerCapita table.

Finally, we changed the type for cName (the name of the country) from varchar(100) to varchar(50) as none of the countries had name lengths approaching 100. All were below 50.