Decisions

Changing our Schema

* changed the percentage attribute in the CountryEdExpendPerGDP and CountryEdExpendPerOthers Relations, and publicExpenditure in CountryDatafrom SMALLINT to REAL
* changed teachers from INT to REAL (it’s in thousands)
* changed the country and region attributes in all the Relations from VARCHAR(30) to VARCHAR(60)

Cleaning the Data

* We first cleaned the csv files that we obtained online using Python, then imported the resulting csv files into SQL and further cleaned the data. We also made manual modifications in some places in the countries data where it made more sense to adjust manually.
* Cleaning data from the csv file containing mapping from countries to subregions (found here: [ISO-3166-Countries-with-Regional-Codes/all.csv at master · lukes/ISO-3166-Countries-with-Regional-Codes](https://github.com/lukes/ISO-3166-Countries-with-Regional-Codes/blob/master/all/all.csv))
  1. With Python pandas, read the csv file into a pandas dataframe, selecting only the name (of the country), region and sub-region columns
  2. Converted all country names to lowercase and removed any parentheses (e.g. for the country “Bolivia (Plurinational State of)”, we only wanted “Bolivia”)
  3. Wrote the resulting dataframes to countries\_and\_regions\_cleaned.csv
  4. In SQL, created a table, CountriesAndRegions, for the cleaned countries\_and\_regions.csv, and imported the csv file into it with \copy
  5. Deleted the country Antarctica from the table (since we decided our relation will not include it)
  6. The original csv file split the regions differently from the other csv files we found: it split Europe and Oceania into more subregions, while the other data did not. So for these regions, we set the subregion to be the region. (e.g. a Country whose subregion was originally “Southern Europe” is updated to be the region “Europe”)
  7. Inserted into the Subregions and Countries Relations (which we had created in phase 2), the distinct countries and subregions columns from the table CountriesAndRegions
  8. Removed the table CountriesAndRegions with DROP TABLE
* Cleaning csv files on Education, found here: (1) [8 Teaching staff at the primary, secondary and tertiary levels](https://data.un.org/_Docs/SYB/PDFs/SYB62_323_201906_Teaching%2520Staff%2520in%2520Education.pdf) and (2) [Enrollment in primary, secondary and tertiary education levels](https://data.un.org/_Docs/SYB/PDFs/SYB62_309_201906_Education.pdf) and (3) [9 Public expenditure on education](https://data.un.org/_Docs/SYB/PDFs/SYB62_245_201905_Public%2520Expenditure%2520on%2520Education.pdf)
  1. With python pandas, read each csv file into a pandas dataframe named (1) teaching\_staff and (2) gross\_enrollment\_ratio, selecting only the country or region name, year, series (what the row value is represents, e.g. “Public expenditure on education (% of GDP)”), and the value
  2. Created two dataframes from public\_expenditure, expend\_per\_gdp and expend\_per\_others, selecting the rows with series “Public expenditure on education (% of GDP)” and “Public expenditure on education (% of government expenditure)” respectively.
  3. Converted all country names to lowercase and removed any parentheses (e.g. for the country “Bolivia (Plurinational State of)”, we only wanted “Bolivia”)
  4. Created two dataframes from each of the above, one for each year range (2005-2010 and 2010-2015) by selecting the rows that had the appropriate year values
  5. For each of the values that we will have in the RegionData and CountryData relations (publicExpenditure, teachers, pupilTeacherRatio, grossEnrollmentRatioMale, grossEnrollmentRatioFemale) we created 6 dataframes, one for each year range (2005-2010 and 2010-2015), and for each of the teaching levels (primary, secondary, and tertiary), and grouped by the country, and found the median value for each
  6. Added relevant columns and column values to be able to differentiate these parameters and then concatenated the 6 dataframes for each value into a single dataframe (i.e. stacked the rows of all the dataframes into a single dataframe) and wrote each one to a csv file (teachers.csv, pupil\_teacher\_ratio.csv, expenditure.csv, ger\_female.csv, and ger\_male.csv)
  7. In SQL, created tables for each of the cleaned csv files from the previous step (TeacherData, PupilTeacherRatioData, GrossEnrollementRatioMaleData, GrossEnrollementRatioMaleData, and ExpenditureData), and imported each csv file into the appropriate table with \copy
  8. Created a view, all\_but\_expenditure, that selects all the rows from a natural join between all the tables except for ExpenditureData from the previous step.
  9. Inserted into the RegionData table (which we had created in phase 2), the natural join between all\_but\_expenditure and subregions tables, selecting the appropriate column names
  10. Similarly, inserted into the CountriesData tables, the natural join between all\_but\_expenditure and and expenditureData, selecting the appropriate column names
  11. Inserted into the CountryEdExpendPerGDP and CountryEdExpendPerOthers tables (which we created in phase 2) the csv files from step 2 using \copy.
  12. Removed the tables TeacherData, PupilTeacherRatioData, GrossEnrollementRatioMaleData, GrossEnrollementRatioMaleData, and ExpenditureData