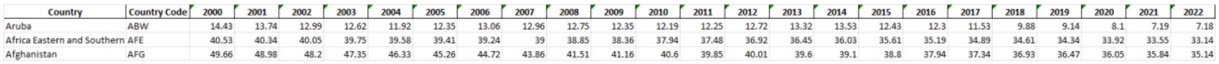
3. Process

In this phase, we focused on preparing the data for analysis by ensuring consistency and relevance. This involved standardizing all datasets to a vertical format for easier handling and applying SQL queries to filter information specific to European countries. These steps ensure the data is properly structured and aligned with the project's objectives, setting the stage for meaningful analysis in the next phase.

1 STANDARD LONG FORMAT

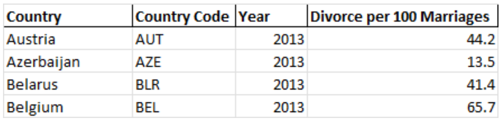
Since some dataset are on horizontal format let homogenize all the data set to vertical format.

From:



Country	Country Code	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
Aruba	ABW	14.43	13.74	12.99	12.62	11.92	12.35	13.06	12.96	12.75	12.35	12.19	12.25	12.72	13.32	13.53	12.43	12.3	11.53	9.88	9.14	8.1	7.19	7.18
Africa Eastern and Southern	AFE	40.53	40.34	40.05	39.75	39.58	39.41	39.24	39	38.85	38.36	37.94	37.48	36.92	36.45	36.03	35.61	35.19	34.89	34.61	34.34	33.92	33.55	33.14
Afghanistan	AFG	49.66	48.98	48.2	47.35	46.33	45.26	44.72	43.86	41.51	41.16	40.6	39.85	40.01	39.6	39.1	38.8	37.94	37.34	36.93	36.47	36.05	35.84	35.14

To:



Country	Country Code	Year	Divorce per 100 Marriages
Austria	AUT	2013	44.2
Azerbaijan	AZE	2013	13.5
Belarus	BLR	2013	41.4
Belgium	BEL	2013	65.7

The objective of this script is to transpose a dataset that contains GDP per capita data for various countries. The process begins with the loading of the primary dataset along with a reference file that includes country codes. Subsequently, the code merges these datasets to ensure that all countries possess corresponding country codes. Following this, the columns are reordered to position the "Country Code" column as desired. The key functionality of the code is derived from the pd.melt() function, which adeptly transforms the data from a wide format—where each year is represented as a distinct column—to a long format. In the long format, each row corresponds to a specific country-year combination, with the pertinent columns including "Country," "Country Code," "Year," and "GDP Per Capita." Ultimately, the transposed dataset is exported to an Excel file for further analysis.

We utilize this script for all the datasets that are required.

```python
# %% [markdown]
# Lets transpose the dataset to keep a clean and standard database

# %%
#Import Library
import pandas as pd

# %%
#Import files
file_path=(r'C:\Users\Admin\Desktop\GOOGLE DATA CERTIFICATE\CAPSTONE_GOOGLE_CERT\database_europe birthrate\CLEAN_DATASETS\GDP_Per_Capita_CLEAN.xlsx')
reference_file_path=(r'C:\Users\Admin\Desktop\GOOGLE DATA CERTIFICATE\CAPSTONE_GOOGLE_CERT\database_europe birthrate\DATABASE\Regions_weather_Geo.xlsx')

#Load file to df
df = pd.read_excel(file_path)
Country_Codes = pd.read_excel(reference_file_path)

# %%
#Merge the Country code to those that dont have it
df = df.merge(Country_Codes[["Country", "Country Code"]], on="Country", how="left", suffixes=("", "_from_reference"))
df.drop(columns=["Country Code_from_reference"], inplace=True)
df.head(2)

# %%
#Reorder the columns
cols = df.columns.tolist()

# Move "Country Code" to second position
cols.insert(1, cols.pop(cols.index("Country Code")))
df = df[cols]
df.head(4)

# %%
# Proceed with the transposing step
df= pd.melt(df, id_vars=["Country", "Country Code"],
                  value_vars=[str(year) for year in range(2000, 2023)],
                  var_name="Year", value_name="GDP Per Capita")
df.reset_index(drop=True, inplace=True)
df.head(4)

# %%
Output_path= (r'C:\Users\Admin\Desktop\GOOGLE DATA CERTIFICATE\CAPSTONE_GOOGLE_CERT\database_europe birthrate\CLEAN_DATASETS\GDP_Per_Capita_CLEAN1.xlsx')
df.to_excel(Output_path,index=False)

```

2 FILTER BY EUROPEAN COUNTRIES

Considering that a considerable portion of our dataset encompasses global information, we will utilize various SQL joins to filter and extract data specifically related to European countries. This same query methodology will be applicable across all datasets.

```
---FILTERING Birth Rate DATASET

SELECT b.Country, b.CountryCode,b.Year, b.BirthRate
FROM Birth_Rate_CLEAN AS b
INNER JOIN Regions_weather_Geo AS r
ON b.CountryCode = r.CountryCode;

```