**ETL Project – Child mortality and poverty**

Arnaz Alavi

Jennifer Puszczewicz

Max Thachet

**Project goal:**

To construct a database containing worldwide data on child mortality from ages 1-4, and child poverty rates by country, extracting the data from a reliable source, cleaning the data, and loading the transformed data into our database.

**Data source:**

Our data was sourced from data.unicef.org

We pulled a csv file from our source on worldwide child mortality narrowing the dataset request to ages 1-4, and we pulled a second csv file from our source providing worldwide poverty data based on average deprivations per child.

**Data cleaning:**

In excel we removed all regions listed under “country code:name”, leaving only countries in this column, and removed all entries in the “sex” column that were not “Female” or “Male”. We realize that using Excel is only a solution if our database will not be updated with new data in the future, which would require the cleaning to be done after the files are uploaded into our Jupyter Notebook. This was a quicker fix for a one-time download.

In our Jupyter Notebook we removed the columns in each dataset that did not provide helpful information, including columns that contained predominantly NaN data, or did not contain unique data. We also removed upper and lower bound data, as we did not find a clear definition of how that data was calculated, leaving its meaning unclear.

We then renamed the columns for clarity and uniformity as shown:

Table

Description automatically generated

We split the string data in the columns, from both datasets, to make it more readable and searchable by users:

Table

Description automatically generated

We removed the leading spaces on the column names using lstrip:

**for** col **in** columns:

new\_UNICEF\_child\_mortality\_data\_df[col] = new\_UNICEF\_child\_mortality\_data\_df[col].str.lstrip()

**Create Database**:

We created our local database (Unicef\_Data\_Warehouse) and created tables in pgAdmin/Postgres, using country\_code\_name, sex, and timeperiod as Primary Keys:

|  |  |
| --- | --- |
|  |  |
|  |  |

We connected our Jupyter Notebook dataframes, by importing create\_engine and inspect from sqlalchemy, and connecting the Notebook to the SQL database:

**import** **pandas** **as** **pd**

**from** **sqlalchemy** **import** create\_engine, inspect

conn = f"postgresql://postgres:postgres@localhost:5432/UNICEF\_Data\_Warehouse"

engine = create\_engine(conn)

We used pandas to load our dataframes into the database:

new\_UNICEF\_child\_mortality\_data\_df.to\_sql(name='child\_mortality', con=engine, if\_exists='replace', index=**False**)

new\_UNICEF\_child\_poverty\_data\_df.to\_sql(name='child\_poverty', con=engine, if\_exists='replace', index=**False**)

We used Inner join on country name and time period to create a database that only provided data on countries where both mortality and poverty data were available. We confirmed that it loaded successfully by querying the database from SQL:

Graphical user interface, text, application, email

Description automatically generated

We also confirmed that it loaded correctly by querying from Jupyter Notebook:

Text

Description automatically generated