ETL Project

Team 4



Raphael Serrano, Swobabika Jena, Thomas Maina and Diana Madonko

Table of  
Contents

Overview 1

Data Extraction 1

Data Transformation 2

Loading Data 2

Overview

Due to the pandemic, we saw the printed word bounce back in 2020 with sales of eBooks and audiobooks showing double digit growth in just a year. There was also an increase in print books too as digital may be convenient, some people still prefer paperback. Therefore, we have chosen to explore published books around the world and related information. What we hope to achieve with this data is to be able to assist people with book choices as it would provide more information on ratings, affordability and other areas that influence their choices.

Data Extraction

Using Kaggle’s API, we have obtained a book dataset from Goodreads which is updated every 2 days, with over 10,000,000 books available on the site’s archives. The Goodreads data has several csv files which include various columns such as title, author, pages number, ratings, ISBN etc.

As the Goodreads data did not contain all the categories, we required we also extracted data from the Google Books API which included information such as retail price, currency, country etc.

To assist with merging our data, we extracted a CSV file of ISBN numbers from Data World.

**Sources**

Goodreads Book Datasets with User Rating from Kaggle

Google Books API

ISBN data from Data World

Data Transformation

To clean up our data we used a Jupyter Notebook. Firstly we imported the csv files into the notebook and read our data file with the pandas library. The next step was to create a ‘for’ loop, which was used to load multiple CSV files into a single DataFrame and cleaned the data frame to only include the columns required for our analysis. As an additional clean-up process to ensure consistency with all loaded data, we removed non-English characters, filtered languages, dropped duplicates and removed rows with missing values.

Once the clean CSV data was appended to the main Dataframe we used a python ‘for’ loop to loop through ISBNs which were the key for our SQL tables. In the same loop we did a Googlebooks API call to extract the author information, print type, categories and price list as these were not available in our CSVs from Goodreads. Finally we converted the directory of lists to DataFrames to be able to store in our database.

To identify any possible issues with the data, we did some extra analysis. We used pandas describe to show a statistical overview of the data so we could easily distinguish outliers. An example of such outliers would be the minimum page number equating to zero, after further investigation we concluded that these were Audiobooks. From our analysis we were also able to identify the top 10 publishers in the Goodreads dataset. We were able to ascertain that the period with the most published books was between 2001 and 2010. We were also able to establish that there is no correlation between the number of pages in a book and it’s rating.

Loading Data

SQL was used to load the extracted and transformed data. As we extracted the data from different platforms, we needed database that will accommodate different types of data and a relational database seemed to be the best fit for our data. We created the connection using Jupyter Notebook and pushed and joined the tables using the ISBN numbers as our key for joining the data.

**Extraction** Graphical user interface, text, application

Description automatically generated

Calendar

Description automatically generated

**Transformation & Analysis – Filtering data**

Text

Description automatically generated

Table

Description automatically generated with medium confidence

Chart

Description automatically generated

Chart, histogram

Description automatically generated

**Chart, scatter chart

Description automatically generated**

**Loading to SQL: Database Connection and Data Frame Loading**

Graphical user interface, text, application

Description automatically generated

**Loading to SQL – Table Creation**

A picture containing table

Description automatically generated