Analyzing Factors Influencing Book Popularity on Goodreads

Group Project - Concept

Abah Oyibi *Research Question, ER Diagram, Query 4 and Plot*

Ákos Lesznik *Choice of Database, Popularity Criteria, Queries 5, 6, Plot and Result Analysis*

Ana Margarida Pina Pereira *Research Question, Gender guesser code, Query 2, Plot and Result Analysis, Presentation*

Leon Josef Moik *Research Question, ER Diagram, Query 3 and Plot*

Santiago Velasco Torre *Choice of Database, Data Modeling and Ingestion, Query 1, Plot and Result Analysis, Reproducibility Aspects*

Research Problem and Motivation

What factors contribute to the popularity of books on Goodreads? Are there shortcuts to appealing to the largest audience? Identifying common traits among popular books on this widely used platform could provide valuable insights for authors, publishers, and readers.

Dataset: https://www.kaggle.com/datasets/jealousleopard/goodreadsbooks/data

With this dataset we can assess popularity through a popularity score, that includes “Number of Ratings”, ”Average Rating” and “Number of Reviews”, and check if there is any correlation with other factors such as publication year, author, or book length.

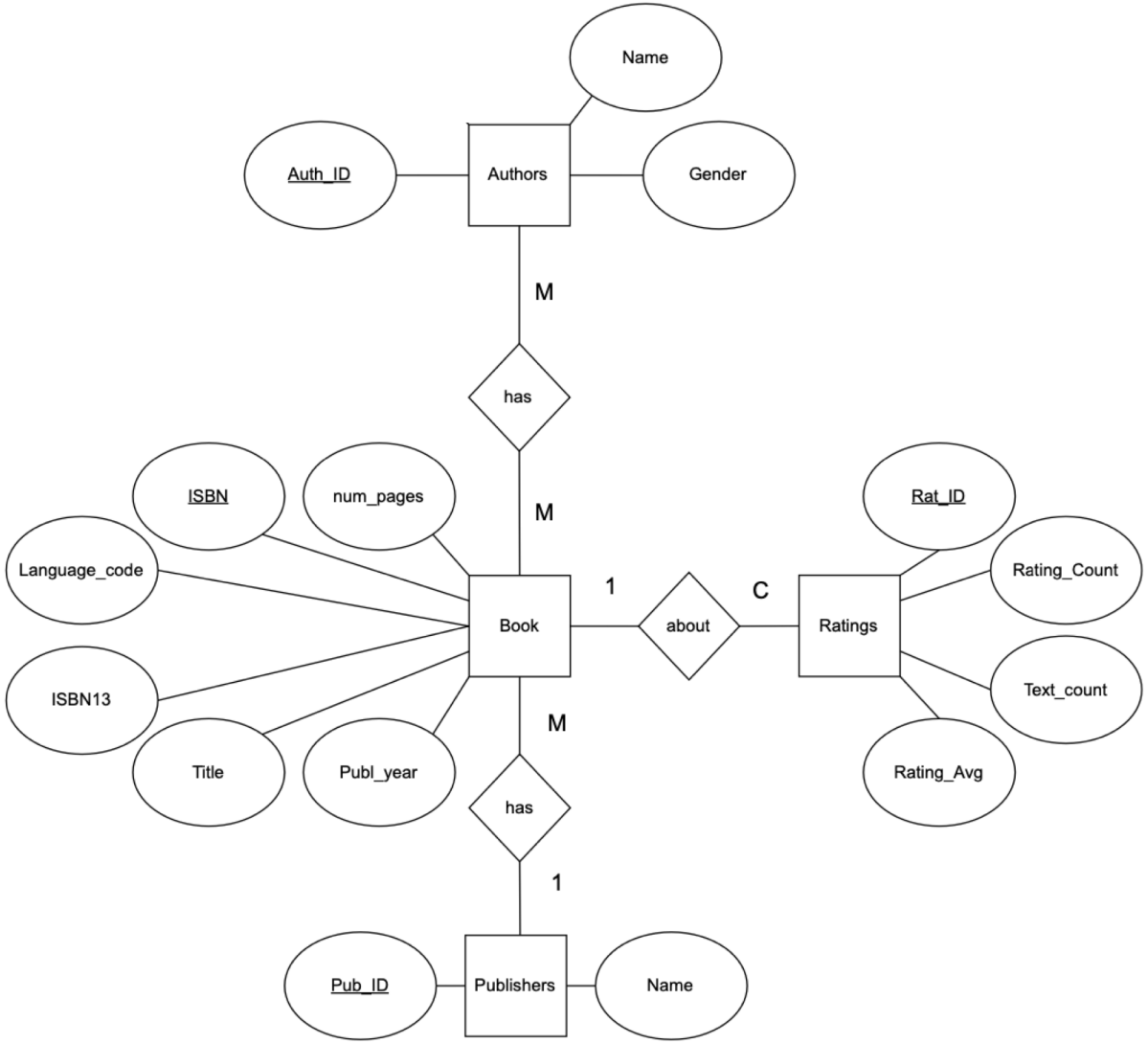
A database efficiently organizes and analyses large datasets, enabling us to calculate popularity scores and uncover patterns and correlations.

Data Modeling and Data Ingestion

With the ER diagram we where able to separate our database into the respective tables: *book*, *publisher*, *ratings* and *authors*.
(Code: *PreProcess.py*)

For the authors we had to create a *book_authors* table to handle the M:M relationship, which associates 1 book to each author.
(Code: *TableCreationSchema.sql* and *CopyStatements.sql*)

Using https://pypi.org/project/gender-detector/ we added the attribute *gender* to author.
(Code: *gender_guesser.py*)



Database Queries and Data Handling

To answer the research question we divided the work into 6 queries that answer the following questions:

1. Does the number of pages of a book influence the popularity? (Code: *Query1.py*)
2. Does the number of books an author published influence the popularity of its books? (Code: *Query2.py*)
3. Does the gender of the author influence the popularity of a book? (Code: *Query3.py*)
4. Does the number of authors influence the popularity of the book? (Code: *Query4.py*)
5. Does the language of the book influence the amount of reviews of a book? (Code: *Query5.py*)
6. Does the publisher have an effect on the popularity of a book? (Code: *Query6.py*)

Our popularity score was calculated using the equation: $\frac{ratings\_count}{AVG(ratings\_count)} * average\_rating$

Presentation and Discussion of Results

By analyzing each query result and their respective plot (in folder *Plot results*) we conclude:

1. While a few books with very few pages might be less popular, the number of pages and popularity do not seem to relate.
2. Most analyzed authors have written fewer than 10 books. Authors with a higher number of published books do not necessarily achieve higher average popularity scores. Highest popularity scores are concentrated among authors in the 1–10 book range;
3. Female authors have a slightly higher mean popularity score which is not statistically significant. Gender does not appear to affect popularity;
4. Books authored by a single individual have the highest average popularity scores. Popularity generally decreases as the number of authors increases, with a few exceptions.
5. Writing in English appears to greatly enhance a book's chances of becoming popular;
6. Certain publishers achieve significantly higher average scores than others, suggesting that being associated with a particular publisher may contribute to a book's popularity.

Factors we considered that influence book popularity: **Language**, **Publisher**, **Number of authors**

Reproducibility Aspects

1. Download the complete dataset (Books_database): Since the file may be corrupted or disorganized for data ingestion, you must run the `PreProces.py` script. This script will organize the database into the appropriate tables for data ingestion.

2. Set up in PostgreSQL: After preprocessing the dataset, execute the schema in PostgreSQL. Then, load the data into the database using the `COPY` statements with your own file paths.

3. Fill the "Gender" column: Once the data ingestion is complete, populate the "Gender" column in the author's table using the `gender_guesser.py` script. (pip install gender-detector).

4. Complete the process: After completing the data ingestion and verifying the database's functionality, execute the queries to answer the specified questions. Make sure to change the database configuration on the beginning of the python made queries. Path changes are also required to run *Query4.py*, its respective plot file *Analysis_query4.ipynb* and *Analysis_query5_6.py*.

Github Repository: https://github.com/SantiagoVelasco2003/GroupProject_Databases.git