Big Data Assignment 1

KFWJOR001 MRCGAB004 WHLJOS001 CRGMAT002

March 1, 2024

uct.png

Contents

1 Find or Create a Suitable Data Set

1.1 Explanation of data set

Link to the dataset: `https://github.com/zygmuntz/goodbooks-10k`

The dataset initially contained multiple csv files representing information on books, and user data on book ratings. This dataset was chosen as its ideal for a MongoDB database due to its semi-structured nature and nested data, which is particularly useful for storing ratings and book tags.

Dataset Content:

- **books.csv**: Each entry represents a book with a unique `book_id`. There are multiple data fields for a book:
  - `book_id`, `goodreads_book_id`, `best_book_id`, `work_id`: Unique id's representing a book, each with a different purpose. We only used `book_id` and `goodreads_book_id` as they're used to link books to user `ratings` and user `to_read` lists.
  - `ratings_1`, `ratings_2`, `...`: Number of user ratings by rating value. eg. `ratings_1` represents the number of 1 star ratings given to that book.
  - The rest of the fields are self explanatory but include info relating to authors, title, release date, and isbn number.
- **ratings.csv**: Each entry is a `user_id` to `book_id` mapping with a rating.
  - `book_tags.csv`: Each entry is a `book_id` to `tag_id` mapping.
  - `tags.csv`: Each entry is a tag_{id} to tag_{name} mapping.
  - `to_read.csv` : Each entry is a `user_id` to `goodreads_book_id` mapping which represents a user adding a book to their `to_read` list.

1.2 Data Pre-Processing:

The data was processed such that the data was represented in JSON format with evidence of nested objects so that we could demonstrate the capabilities of MongoDB

Here is a quick outline on how we processed the data to create JSON files:
Libraries used: `Pandas`, `PyArrow`, `Faker`

`Pandas` was used to load the csv files into dataframes where we merged data and applied `group by` aggregate functions to obtain lists of data objects per a unique entry id. This was useful, for example, when we obtained a list of tags per `book_id`.

`Faker` was used to generate random usernames for each id that were then written to `user_data.csv`. The `dataframes` were then converted into JSON files.

All data pre-processing code is in the data-processing directory but the output JSON files are included in the final submission. **