INST733 - Database Design
Section - IM101

Team Project Final Report(BookPantry Database)

05/09/2023

Team 4 (Primary Keys)

Rajeevan Sai Narasimha Madabushi

Sravya Lenka

Ushasri Bhogaraju

I. Introduction

Goodreads is a social cataloging website that allows users to search, rate, and review books. The website has over 100 million users and contains millions of book listings, author profiles, user reviews, and ratings. All three of us on the team like good books to read and sometimes find our next book from Goodreads. Taking inspiration from this, we started with the goal of designing a similar database that can store and manage vast amounts of data effectively and efficiently. The main objective of our project, BookPantry Database, is to identify highly-rated books for readers. It can also be useful for publishers and researchers to analyze reader ratings and reviews to identify trends and preferences, make informed decisions, and gain insights into the reading culture. As part of this project, we created a small-scale, non-trivial relational database by developing a logical design, implementing the physical design, populating the databases, and running CRUD operations.

II. Database Design and Implementation

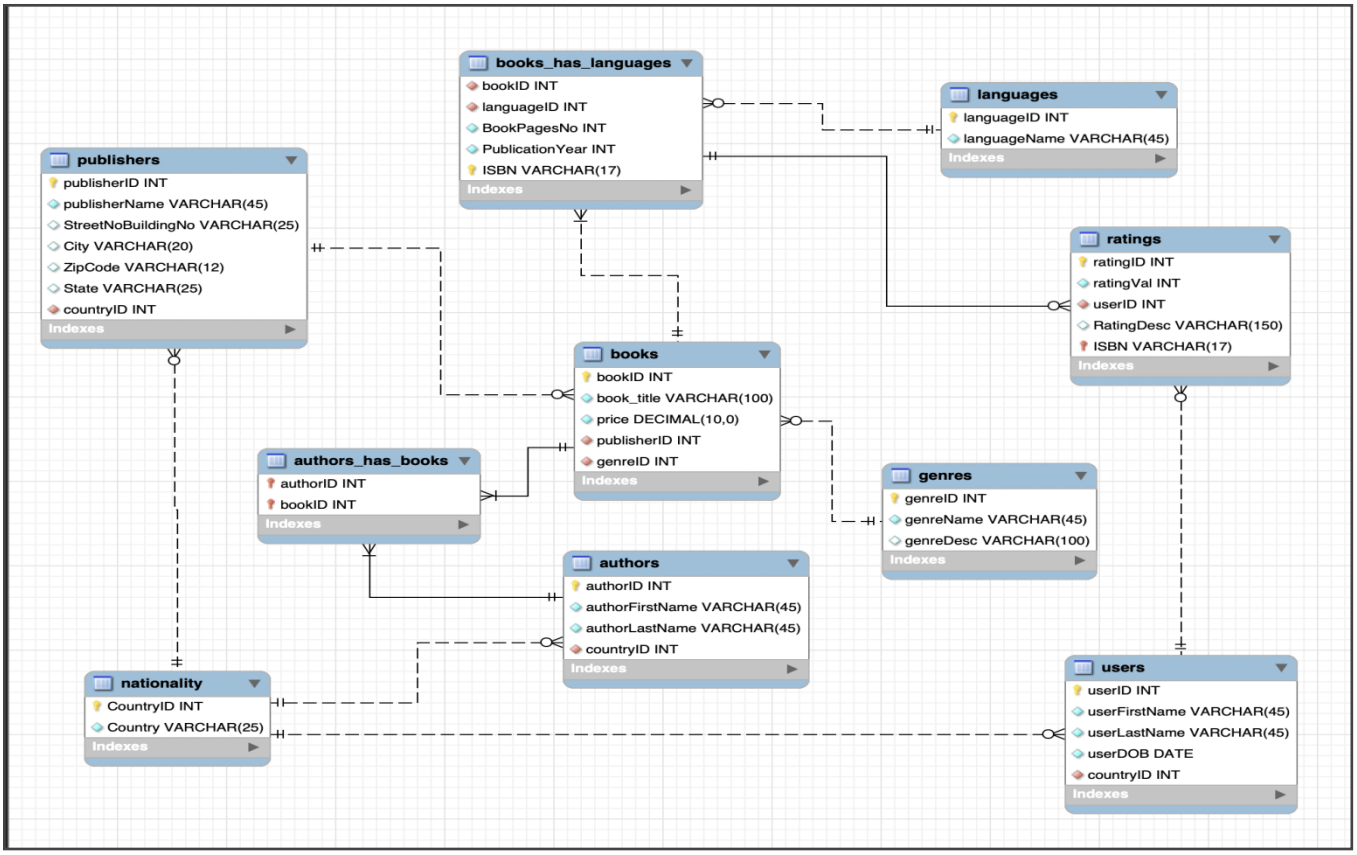
Logical Design

We designed our Enhanced Entity Relationship diagram using the bottom-up approach beginning with piecing together fields and dependencies, to reduce data redundancy.

Steps followed in creating the ERD

1. Creating a list of entities and their attributes in Excel.
2. Establishing relationships between the entities in Excel and identifying the Primary and Foreign keys.
3. Normalizing the tables up to 3NF and adding the relationships in Excel.

4. Using this as a reference, we created the tables by choosing the appropriate data types, their length, and primary and foreign key constraints.
5. We added identifying and non-identifying relationships between the tables and constantly refined them based on feedback and our assumptions about the database. The final image of our EER Diagram is added below.



Physical Design

Creation of Tables

Once we created the ERD model, we utilized the Forward Engineering option on MySQL WorkBench to create our database from our model. We synchronized the model to generate the script. After verifying the generated SQL script, we also created a few other constraints such as UnSigned(UN) for a few columns such as any id, year, etc. which can only take a positive value. We then verified if all our data types are correct and continued with the rest of the project.

Data Insertion

The first thing we did before entering the data was identify the independent tables, dependent tables, and joining tables. We started by inserting data into independent tables as data in these tables would not be dependent on other tables. Then, we inserted data into dependent and joining tables. We initially inserted around 20-30 records for each table as per the requirements and 50 records in the joining tables. Later, we added more records in respective tables to have a good representation of expected results such as multiple books by the same author, multiple books by the same publisher, and the same book in multiple languages.

Sample Data

Based on our requirements we determined the need to incorporate additional realistic information from external sources. For example, we wanted to query ratings for a particular age group for a set genre of books. The dataset only provided cumulative ratings count(average ratings) and reviews count. To perform the queries, we needed to add columns such as Ratingvalue, Rating description in Ratings table, CountryID, name in 'Nationality' table, 'Genres' in the Genre table, and 'Users' to the Kaggle dataset and obtain realistic data. We found the supplementary data online. Thus our BookPantry dataset is an enhanced version of the Goodreads dataset we found on Kaggle.

Views

Once we had all the data, we started looking at different views we could create as a part of our final project. We came up with multiple queries and implemented and saved 10 views on our DB backup. We also created a requirement fulfillment table that shows which view satisfies which of the 5 mentioned requirements. We have also created a table with the view name and view description. Both the tables are listed below.

View Description Table

View No	View Name	View Description
1	vw_avgRatingforDiffGenresthroughYears	Ratings for every book genre across different years.
2	vw_publishersAboveAvgPublisherRating	Publishers with ratings more than the average of all ratings of all publishers

3	vw_booksRatinginDiffLang	Rating of the same book in different languages, when the same book is published in multiple languages
4	vw_ageGroupGenreRatings	Ratings for different Genres from different age groups.
5	vw_top10RatedBooksAndAuthors	Top 10 Authors with the highest average ratings for the books they wrote.
6	vw_noOfBooksinEachGenre	Number of Books in each Genre
7	vw_publishersAndGenresWithAboveAvgRatings	Which Genre of books by which publishers have ratings more than the avg rating of all book ratings?
8	vw_top10AuthorswithMostBooks	Top 10 Authors with the most number of books.
9	vw_bookswithPagesMorethanAvgPages	Books with the number of pages more than the average number of pages of all the books.
10	vw_booksinEachLanguage	Number of books in each Language

Requirement Fulfillment Table

View No	View Name	Req. A	Req. B	Req. C	Req. D	Req. E
1	vw_avgRatingforDiffGenresthroughYears	×		×	×	
2	vw_publishersAboveAvgPublisherRat	×	×	×	×	×
3	vw_booksRatinginDiffLang	×	×	×	×	×
4	vw_ageGroupGenreRatings	×		×	×	
5	vw_top10RatedBooksAndAuthors	×		×	×	
6	vw_noOfBooksinEachGenre			×		
7	vw_publishersAndGenresWithAboveAvgRatings	×	×	×	×	×
8	vw_top10AuthorswithMostBooks	×		×	×	
9	vw_bookswithPagesMorethanAvgPages		×	×	×	×
10	vw_booksinEachLanguage			×	×	

III. Changes from the Original Design:

We started with 8 tables in our proposal namely (1) Books (2) Authors (3) Users (4) Publishers (5)Reviews (6)Ratings (7)Genre (8) Language. We made the following changes for our final project.

Removal of Reviews entity: In the process of Normalization, we found that the text-based 'Reviews' table can be removed and an attribute with the name 'RatingsDesc' (Rating description) can be added to the Ratings table itself.

Addition of Nationality table: We found that Publishers, Authors, and Users all had Nationality attributes and decided to add the Nationality table in our final design.

Two new tables by the names 'Books_has_Languages' and 'Authors_has_Books' were added to maintain the many-to-many relationships between Books and Languages and Authors and Books.

IV. Issues Experienced During Development:

We have faced a few issues in different stages of our project. However, we addressed these issues by engaging in discussions, generating ideas, and implementing solutions.

A. **Issues faced while creating the ERD:**

   The "books_has_languages" table has a Composite Key derived from the primary keys of "books" and "languages" tables. As the "ratings" table is related to "books_has_languages", a composite Foreign Key was created using "bookId" and "languageId" from the "books_has_languages" table. However, we discovered that "ISBN" would be a unique primary key for all combinations of "bookId" and "languageId" in the "books_has_languages" table. We modified the primary key of "books_has_languages" to "ISBN," which is also used as a foreign key in the "ratings" table.

B. **Issues faced while inserting data into tables:**

   While populating the Books-has-Languages table, we found that the Primary key did not auto-generate correctly. We corrected it and repopulated the table.

V. Critical Evaluation of DB:

Purpose: Our database serves a lot of purposes as mentioned in our introduction. It can be used by readers to identify highly-rated books, and by publishers and researchers to analyze reader ratings and reviews to identify trends and preferences and make informed decisions.

Design: The database is designed into well-organized tables, with correct data types and realistic values and the process we followed in designing the database was mentioned in the Logical and Physical design sections in this report. We designed our database in a way to accommodate the user requirements as mentioned in our purpose.

Performance: Since the bookpantry database is a small-scale database with few records, the queries were processed with speed and accuracy. We have not tested the Database with large volumes of data as it is beyond the scope of this project.

Cost: The database is built using MySQL Free edition. It is affordable and suits the purpose of this Project.

Strengths:

- The logical structure of our database considers the relationship between entities, keeping in mind the principles of data redundancy and integrity.
- Records added to the database accurately represent real-world data.
- It is accessible in terms of easily storing and retrieving information about books, authors, and ratings from the database.
- The queries can be efficiently used to generate answers to a few of the marketing research questions such as best-rated genres for each age group, and famous authors.
- While the data is currently limited, it can be spanned to include more information.

Limitations:

- Only a limited number of records for now.
- There are a few obvious limitations such as limited flexibility, our DB would not be the best option to handle unstructured data, and limited scalability since it might get hard to handle increasing amounts of columns for each table very often.
- In case we have many more tables and more data, it may become necessary to write complex queries to find answers for a few of the questions.

VI. Lessons Learned

- The importance of ERD in developing a database is crucial to have the structure pinned down before working on CRUD operations.
- Easy synchronization of ERD to generate a schema.
- Data derived using Excel Formulas is not infallible. Triangulation before populating tables is important.
- Auto-generation of primary keys using the 'Default' keyword, saves a lot of time. Obviates the need to spend time manually checking for duplicates.
- When using a joining table, the Composite Key generated does not necessarily need to be the Primary Key of the new table.

VII. Feasibility Analysis

Right now, we have 10 tables, and all the data we have is properly structured into different tables. Since the scope of the project is small, MySQL would be the better fit since it is more reliable, easy to use, easily scalable, and ACID-compliant. However, we did a feasibility analysis for our project with other databases and considered NoSql. NoSql such as MongoDB could be the best choice when you are dealing with a very large dataset with a lot of structured and unstructured data. Switching to NoSql for BookPantry would make sense if the project scope expands to include a wider range of data associated with users, and authors. It would allow for greater flexibility in storing and analyzing complex data structures that may arise from these expanded needs.

VIII. Potential Future Improvements

We chose a small scope for the purpose of this project. However, there is room for improvement that can be made in the future.

- We can add a new column edition to the books_has_languages table to include the edition of the books as well.
- A few new tables could be added. Such as stores - which would give us details about the availability of books in different stores, and sales table - which could give us the details about the sales of books.
- We could also add the type of books(ebook, audiobook, physical book) and all these could also have different ratings.

IX. References

1. Goodreads. (n.d.). Goodreads. Retrieved May 7, 2023, from https://www.goodreads.com/
2. Soumik. (2020, March 9). Goodreads-books. Kaggle. Retrieved May 7, 2023, from https://www.kaggle.com/datasets/jealousleopard/goodreadsbooks
3. Kleppmann, M. (2021). Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems. O'Reilly.

X. Appendix

Project Diary

Date	Task completed
02/01/2023	Project topic discussion - brainstorming ideas for the project.
02/04/2023	Searching for and analyzing various datasets, finalizing Goodreads dataset Kaggle. Choosing Project Team name as Primary Keys, other options considered Query Wizards, SQL Serpents
02/20/2023	Draft Project Proposal, discussion of tables and queries, construction of ERD
03/07/2023	Project Proposal finalization and submission
03/25/2023	Listing entities and attributes in an Excel sheet, normalization
04/07/2023 - 04/10/2023	In MySQL creating the tables, finalizing data types, and constraints, drawing ERD and submitting Project Progress Report
04/13/2023	ERD discussion with Dr. Duffy. Removal of Reviews table, addition- Nationality
04/14/2023	Rearranging attributes, fine-tuning data types and constraints, forward engineered the model and synchronized the ERD
04/15/2023 to 04/17/2023	Inserting data into tables
04/25/2023	Testing the database with different queries, discovering errors in the ratings table, and rectifying the same. Adding more data into all the tables.
04/30/2023	Writing required Views as per Project requirements. Starting on writing the Project Report
05/07/2023	Finalizing the Project Report