DATABASE MANAGEMENT SYSTEM MINI PROJECT REPORT

By: Harikrishnan U / 21172040030038

Project Title: BOOK Hive - Collaborative Reading Community Platform

1. Abstract

This project, titled **BOOK Hive**, aims to create a centralized, relational database system for a collaborative online reading community. The platform allows registered **users (readers)** to **log in** or create new accounts, post details and **summaries** of books they have read, and share **reviews**. The primary objective is to move user-generated book data from disparate, unstructured sources into an organized, queryable database. The system utilizes **MySQL** for data storage, enforcing **data integrity** between user profiles, books, and their associated posts. The design focuses on establishing clear one-to-many relationships (User to Post) and integrating a dedicated **Book** entity for detailed metadata. The key outcome is a scalable and efficient data structure that supports a dynamic social reading platform.

2. Objectives

- To design and implement a relational database schema for **BOOK Hive** that facilitates the secure management of user accounts, book entries, and user-generated content (posts/summaries).
- To maintain data consistency, integrity, and accuracy by linking posts directly to authenticated users and valid book titles.
- To create efficient SQL queries for **CRUD (Create, Read, Update, Delete)** operations, including fetching a user's complete posting history and generating popular book reports.
- To implement a secure mechanism for user **authentication and authorization** for account creation and login.
- To generate useful community reports for administration (e.g., top 10 most reviewed books, most active users).

3. System Analysis

a. Problem Definition

In many online communities, book discussions and reviews are scattered across multiple forums, blogs, or social media sites. There is a lack of a unified, searchable, and structured database dedicated to linking reader profiles directly to book summaries and reviews. This fragmentation prevents easy discovery of content, reliable tracking of user reading history, and consistent data quality regarding book metadata.

b. Existing System

The existing system is informal and distributed across the web. Limitations include:

1. **High Redundancy and Inconsistency:** The same book may be listed multiple times with inconsistent titles or summaries, making search unreliable.
2. **Poor Data Structure:** User-generated content is often stored as raw text, making it difficult to analyze, categorize, or link back to specific reading metrics.
3. **Inefficient Access:** Users cannot easily find all summaries or reviews written by a specific community member.
4. **No Centralized Authentication:** Readers must manage separate accounts across different platforms.

c. Proposed System

The **BOOK Hive** database is a **centralized and structured community solution**. It improves upon the existing system by:

1. **Centralized Authentication:** A single **User** table manages all logins and account creations securely.
2. **Content Organization:** Dedicated tables for **Book** and **Post** enforce structure, ensuring every summary/review is linked to exactly one user and one book.
3. **Data Integrity:** Foreign Keys prevent orphaned posts (posts without a user or a book).
4. **Efficiency:** Optimized queries allow for real-time sorting and filtering of posts based on popularity, book title, or author.

4. System Design

a. Entity Relationship (ER) Diagram

*(Note: The key focus is the **1:N relationship** between User and Post, and the **1:N relationship** between Book and Post. This setup allows many posts (summaries/reviews) to refer to the same Book and be authored by the same User.)*

- **Entities: User**, **Book**, **Post**.
- **Relationships:**
  - **User** authors **Post** (One-to-Many).
  - **Book** is the subject of **Post** (One-to-Many).

b. Schema Diagram

*(Primary Keys are **PK**, Foreign Keys are **FK**)*

Table Name	Key	Field Name	Data Type	Relationship

User	PK	user_id	INT	
		username	VARCHAR(50)	Unique Login ID
		email	VARCHAR(100)	Unique Email for recovery
		password_hash	CHAR(60)	For secure password storage
Book	PK	book_isbn	CHAR(13)	Unique identifier (ISBN-13)
		title	VARCHAR(255)	
		author	VARCHAR(150)	
		genre	VARCHAR(50)	
Post	PK	post_id	INT	Surrogate Key
	FK	user_id	INT	Author of the post (Refers to User.user_id)
	FK	book_isbn	CHAR(13)	Book being reviewed (Refers to Book.book_isbn)
		post_type	ENUM('Summary', 'Review')	Defines content type
		content_text	TEXT	The summary/review text
		rating	INT	Optional 1-5 rating

		created_at	DATETIME	Timestamp of creation

c. Data Dictionary

Table Name	Field Name	Data Type	Size	Description
User	user_id	INT	10	Unique ID, Primary Key
User	username	VARCHAR	50	Unique login name
User	password_hash	CHAR	60	Hashed password using BCrypt or similar secure algorithm
Book	book_isbn	CHAR	13	International Standard Book Number, Primary Key
Post	content_text	TEXT	65535	The full text of the summary or review
Post	rating	INT	1	Optional user rating (1-5 stars)

5. Implementation

a. Software & Hardware Requirements

Category	Requirement
Software	**Database:** MySQL 8.0+ or PostgreSQL

	Interface: DBeaver / MySQL Workbench
	Development: Node.js/Python/Java for application layer
Hardware	**Minimum RAM:** 4GB
	Processor: Dual-Core 1.6 GHz+
	Storage: 200MB free disk space (to account for TEXT content)

b. SQL Queries Used

DDL (Data Definition Language) - Table Creation

```
-- 1. Create User table for secure authentication
CREATE TABLE User (
   user_id INT PRIMARY KEY AUTO_INCREMENT,
   username VARCHAR(50) UNIQUE NOT NULL,
   email VARCHAR(100) UNIQUE NOT NULL,
   password_hash CHAR(60) NOT NULL,
   created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);

-- 2. Create Book metadata table
CREATE TABLE Book (
   book_isbn CHAR(13) PRIMARY KEY, -- Using ISBN-13 as the natural key
   title VARCHAR(255) NOT NULL,
   author VARCHAR(150) NOT NULL,
   genre VARCHAR(50),
   publication_year INT
);

-- 3. Create Post table (User-generated content: Summaries & Reviews)
CREATE TABLE Post (
   post_id INT PRIMARY KEY AUTO_INCREMENT,
   user_id INT NOT NULL,
   book_isbn CHAR(13) NOT NULL,
   post_type ENUM('Summary', 'Review') NOT NULL,
   content_text TEXT NOT NULL,
   rating INT, -- NULLABLE: only applicable for 'Review'
   created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (book_isbn) REFERENCES Book(book_isbn) ON DELETE RESTRICT
);
```

DML (Data Manipulation Language) - Data Insertion

```
-- Inserting a new book
INSERT INTO Book (book_isbn, title, author, genre, publication_year)
VALUES ('9781234567890', 'The Midnight Library', 'Matt Haig', 'Fantasy', 2020);

-- Inserting a new user (password is 'readerpass' hashed)
INSERT INTO User (username, email, password_hash)
VALUES ('bookworm_max', 'max@example.com', '$2a$10$xyz...');

-- Inserting a post (a review) linked to the user and the book
INSERT INTO Post (user_id, book_isbn, post_type, content_text, rating)
VALUES (
    (SELECT user_id FROM User WHERE username = 'bookworm_max'),
    '9781234567890',
    'Review',
    'An uplifting and thought-provoking novel about life choices.',
    5
);
```

Retrieval (SELECT with conditions) - User's Profile View

This query retrieves all summaries and reviews posted by a specific user (bookworm_max).

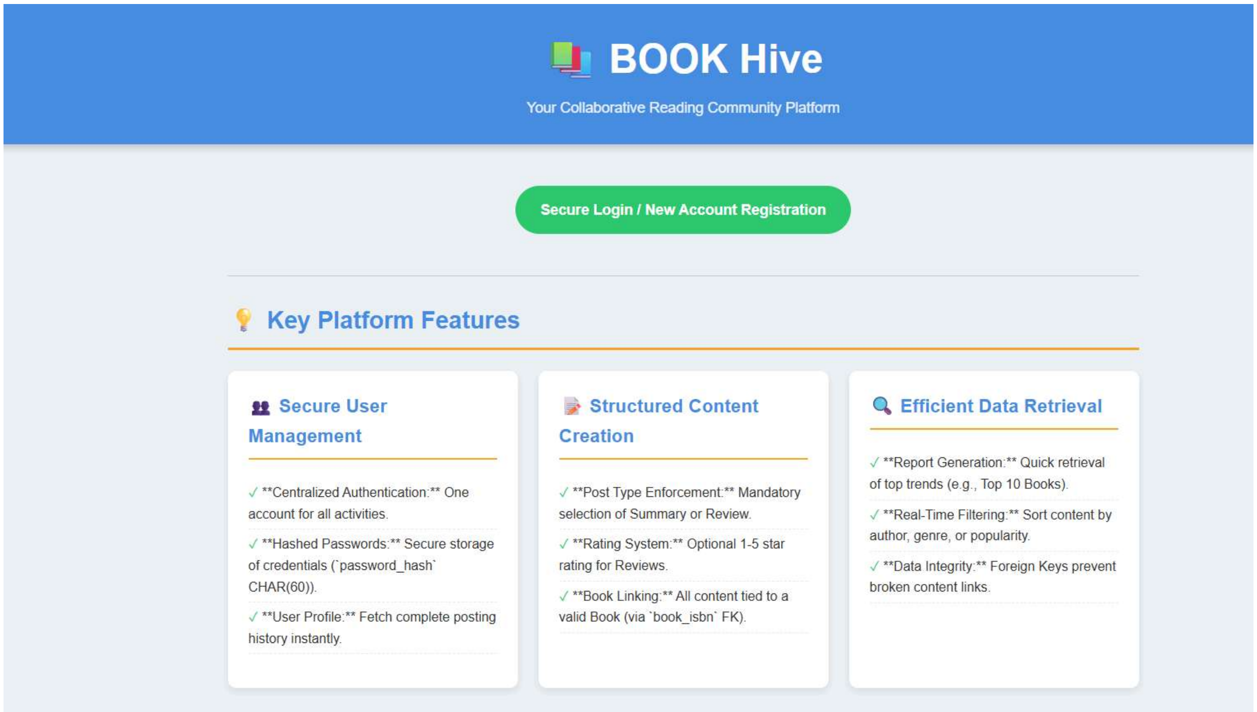
```
-- Query to retrieve all posts by a specific user, including the book title
SELECT
    P.post_type,
    B.title AS book_title,
    P.content_text,
    P.rating,
    P.created_at
FROM
    Post AS P
JOIN
    User AS U ON P.user_id = U.user_id
JOIN
    Book AS B ON P.book_isbn = B.book_isbn
WHERE
```

```
    U.username = 'bookworm_max'
ORDER BY
    P.created_at DESC;
```

Joins and Aggregation - Top 5 Most Reviewed Books

```
-- Query to find the top 5 most reviewed books
SELECT
    B.title,
    B.author,
    COUNT(P.post_id) AS total_reviews,
    TRUNCATE(AVG(P.rating), 2) AS average_rating
FROM
    Book AS B
JOIN
    Post AS P ON B.book_isbn = P.book_isbn
WHERE
    P.post_type = 'Review' AND P.rating IS NOT NULL
GROUP BY
    B.book_isbn, B.title, B.author
ORDER BY
    total_reviews DESC
LIMIT 5;
```

6. Outputs (Front end Web page)



📚 BOOK Hive

Your Collaborative Reading Community Platform

💡 Key Platform Features

👥 Secure User Management

✓ **Centralized Authentication:** One account for all activities.

✓ **Hashed Passwords:** Secure storage of credentials (`password_hash` CHAR(60)).

✓ **User Profile:** Fetch complete posting history instantly.

📝 Structured Content Creation

✓ **Post Type Enforcement:** Mandatory selection of Summary or Review.

✓ **Rating System:** Optional 1-5 star rating for Reviews.

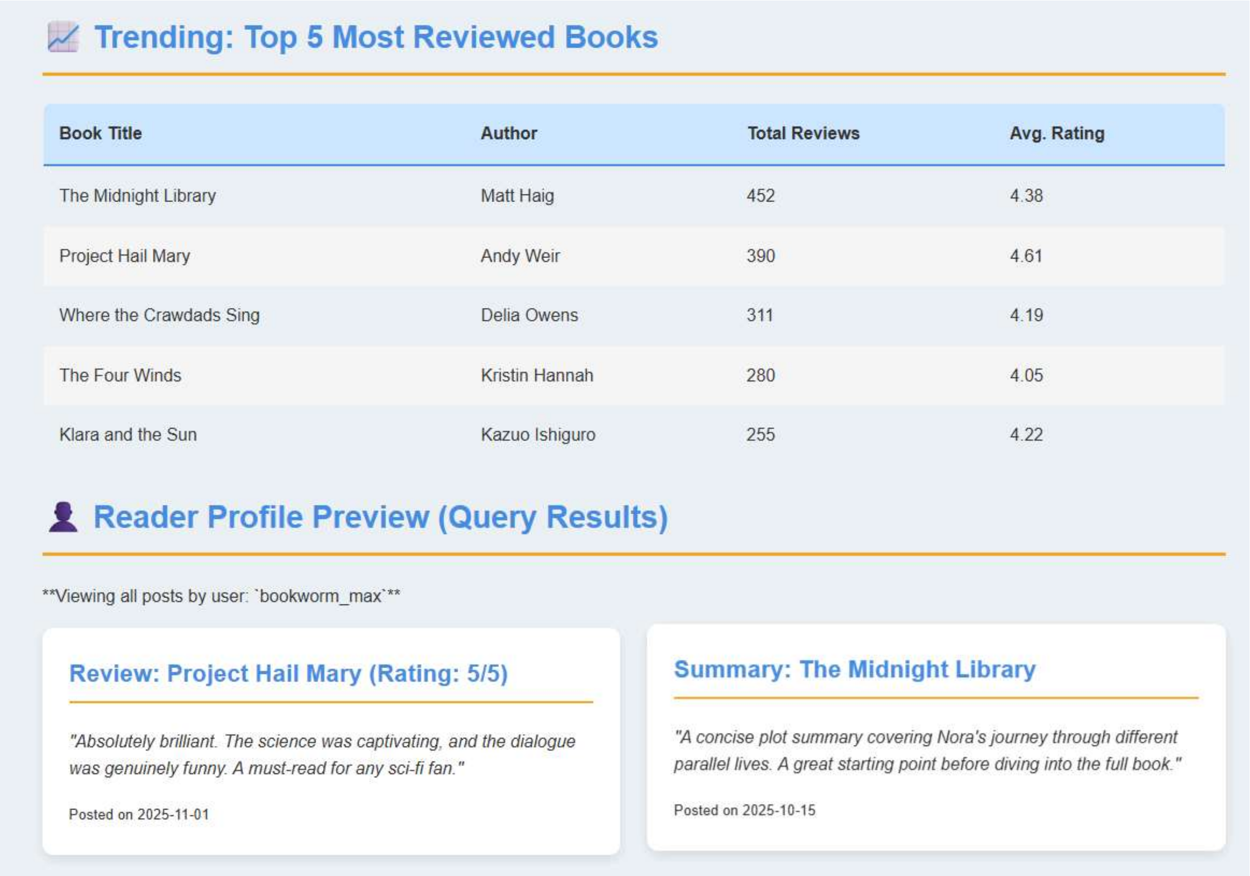
✓ **Book Linking:** All content tied to a valid Book (via `book_isbn` FK).

🔍 Efficient Data Retrieval

✓ **Report Generation:** Quick retrieval of top trends (e.g., Top 10 Books).

✓ **Real-Time Filtering:** Sort content by author, genre, or popularity.

✓ **Data Integrity:** Foreign Keys prevent broken content links.



📈 Trending: Top 5 Most Reviewed Books

Book Title	Author	Total Reviews	Avg. Rating
The Midnight Library	Matt Haig	452	4.38
Project Hail Mary	Andy Weir	390	4.61
Where the Crawdads Sing	Delia Owens	311	4.19
The Four Winds	Kristin Hannah	280	4.05
Klara and the Sun	Kazuo Ishiguro	255	4.22

👤 Reader Profile Preview (Query Results)

Viewing all posts by user: `bookworm_max`

Review: Project Hail Mary (Rating: 5/5)

"Absolutely brilliant. The science was captivating, and the dialogue was genuinely funny. A must-read for any sci-fi fan."

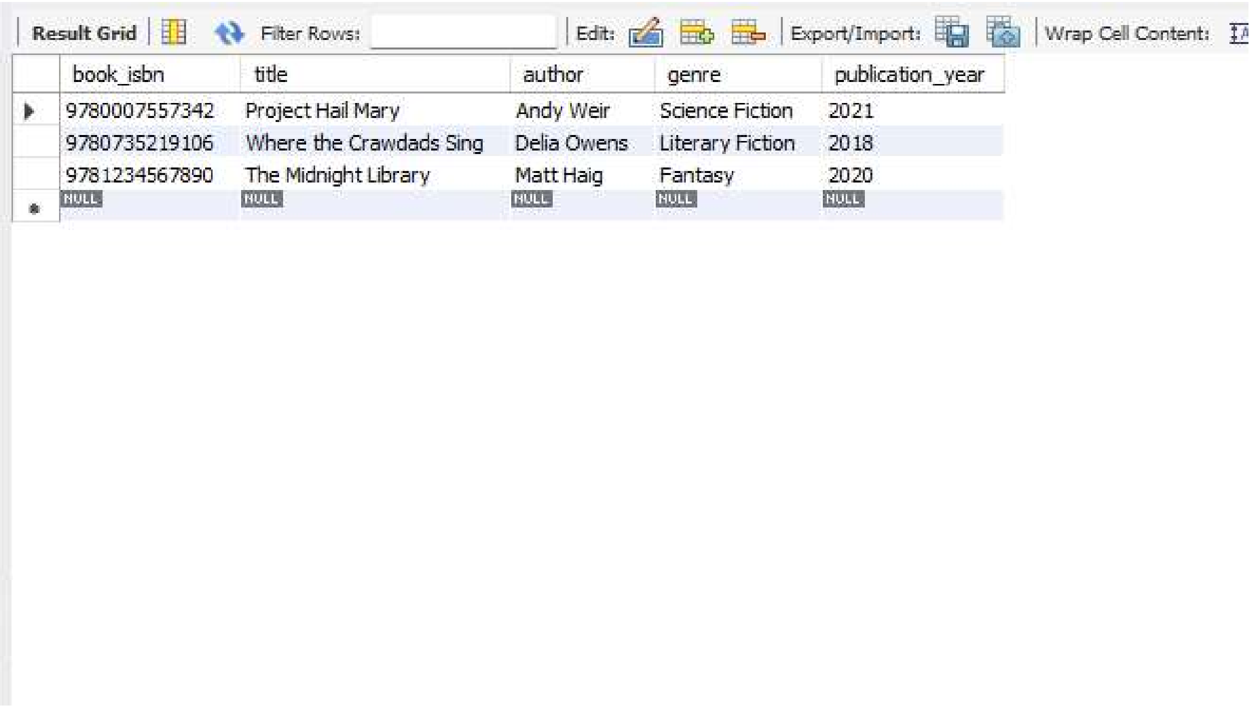
Posted on 2025-11-01

Summary: The Midnight Library

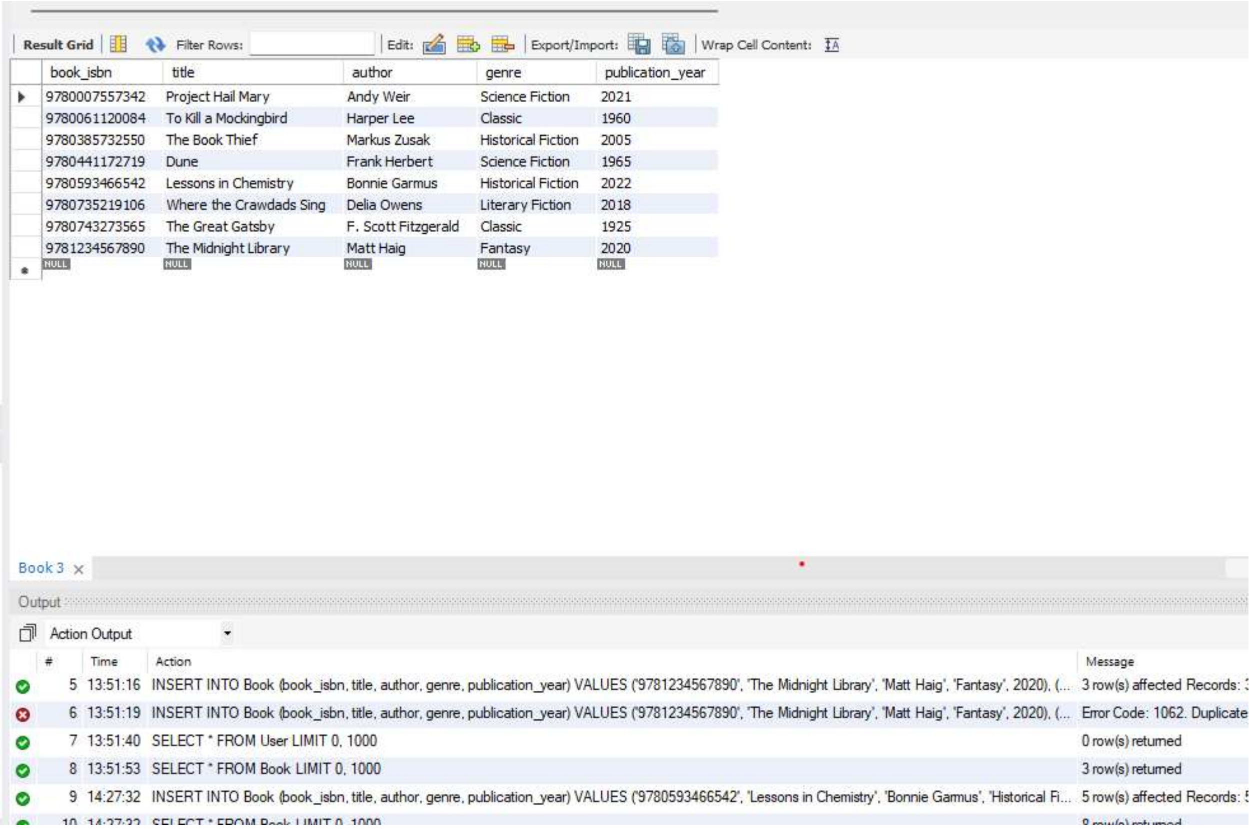
"A concise plot summary covering Nora's journey through different parallel lives. A great starting point before diving into the full book."

Posted on 2025-10-15

Back end(SQL server)



book_isbn	title	author	genre	publication_year
9780007557342	Project Hail Mary	Andy Weir	Science Fiction	2021
9780735219106	Where the Crawdads Sing	Delia Owens	Literary Fiction	2018
9781234567890	The Midnight Library	Matt Haig	Fantasy	2020
NULL	NULL	NULL	NULL	NULL



book_isbn	title	author	genre	publication_year
9780007557342	Project Hail Mary	Andy Weir	Science Fiction	2021
9780061120084	To Kill a Mockingbird	Harper Lee	Classic	1960
9780385732550	The Book Thief	Markus Zusak	Historical Fiction	2005
9780441172719	Dune	Frank Herbert	Science Fiction	1965
9780593466542	Lessons in Chemistry	Bonnie Garmus	Historical Fiction	2022
9780735219106	Where the Crawdads Sing	Delia Owens	Literary Fiction	2018
9780743273565	The Great Gatsby	F. Scott Fitzgerald	Classic	1925
9781234567890	The Midnight Library	Matt Haig	Fantasy	2020
NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message
5	13:51:16	INSERT INTO Book (book_isbn, title, author, genre, publication_year) VALUES ('9781234567890', 'The Midnight Library', 'Matt Haig', 'Fantasy', 2020), (...	3 row(s) affected Records: 3
6	13:51:19	INSERT INTO Book (book_isbn, title, author, genre, publication_year) VALUES ('9781234567890', 'The Midnight Library', 'Matt Haig', 'Fantasy', 2020), (...	Error Code: 1062. Duplicate
7	13:51:40	SELECT * FROM User LIMIT 0, 1000	0 row(s) returned
8	13:51:53	SELECT * FROM Book LIMIT 0, 1000	3 row(s) returned
9	14:27:32	INSERT INTO Book (book_isbn, title, author, genre, publication_year) VALUES ('9780593466542', 'Lessons in Chemistry', 'Bonnie Garmus', 'Historical Fi...	5 row(s) affected Records: 5
10	14:27:32	SELECT * FROM Book LIMIT 0, 1000	8 row(s) returned

7. Results and Discussion

The implemented **BOOK Hive** schema establishes a robust and flexible structure for a reading community. By separating **User**, **Book**, and **Post** into distinct, highly related entities, the system successfully minimizes redundancy and maximizes data integrity. The mandatory foreign key constraints ensure that content is never posted anonymously or linked to an invalid book. The use of the TEXT data type accommodates lengthy summaries and reviews, while the DATETIME field enables sophisticated analysis of user activity and content freshness. The queries demonstrate the system's core capabilities in retrieving personalized content and generating valuable community metrics (like average ratings).

Possible Improvements:

1. Add a **Comment** table to allow users to interact with individual posts, creating a three-level hierarchy (User -> Post -> Comment).
2. Implement a **Tags** table with a many-to-many relationship with **Book** to improve searchability by topic.
3. Utilize **Indexes** on book_isbn and user_id in the Post table to further optimize complex join operations.

8. Conclusion

The **BOOK Hive** project successfully establishes a normalized, secure, and relational database foundation for a community-driven book platform. The design effectively resolves the challenge of unstructured online content by centralizing reader profiles and strictly associating all summaries and reviews with both the author and the book. This project lays a strong groundwork for a scalable social application that relies on secure user authentication and efficient data retrieval.

9. References

- “Database System Concepts” by Korth, Silberschatz, and Sudarshan
- Official MySQL Documentation (dev.mysql.com)
- Principles of Normalization and Relational Database Design, various academic and online sources