Database Design Justification for BookHub Platform

1. Introduction

The database design for the BookHub platform ensures efficient management of users, book listings, transactions, and reviews. The design follows best practices in database normalization, entity relationships, and scalability to handle future growth. The database model was designed using the Crow's Foot design model.

2. Normalization Considerations

Normalization is applied to reduce data redundancy and improve consistency:

- **First Normal Form (1NF):** Each table has a primary key, and all attributes contain atomic values.
- **Second Normal Form (2NF):** All non-key attributes depend on the whole primary key (e.g., a transaction table links buyers and sellers to avoid data duplication).
- **Third Normal Form (3NF):** All attributes are independent of non-key attributes (e.g., user information is separate from book listings to prevent anomalies when a user updates their profile).

3. Entity Relationships and Cardinalities

The relationships between entities are designed to reflect real-world interactions:

User Management

- Each **User** (`user_id`) can list multiple books (**1:M** relationship).
- Users have profiles with attributes such as `email`, `password`, and `registration_date`.

Book Listings

- Each **Book** (`book_id`) has a single seller (**1:M** relationship with Users).
- Attributes include `title`, `genre`, author, `price`, and `condition`.

Transactions (Buying/Selling)

- Each **Transaction** (`transaction_id`) is linked to one book and involves one buyer and one seller (**1:1** relationship between Transactions and Books, and **1:M** between Users and Transactions).
- Status tracking (`Pending`, `Completed`, `Canceled`) ensures process monitoring.

Reviews and Ratings

- Users can write multiple reviews, but each review belongs to one book (**1:M** relationship).
- Reviews include `rating`, `comment`, and `review_date` to track user feedback.

4. Scalability and Performance Optimization

- **Indexes:** Used on `user_id`, `book_id`, `transaction_id`, and `exchange_id` for faster lookups.
- **Foreign Keys:** Enforce referential integrity across related tables.
- **Partitioning:** Possible future optimization for large datasets, such as archiving completed transactions and exchanges.

Conclusion

The database design optimally supports the BookHub platform's core functionalities while ensuring data integrity, normalization, and scalability. By structuring relationships efficiently, the system minimizes redundancy, improves query performance, and supports long-term growth.

Group_32_Assignment