Project Report

2.1. Conceptual Design

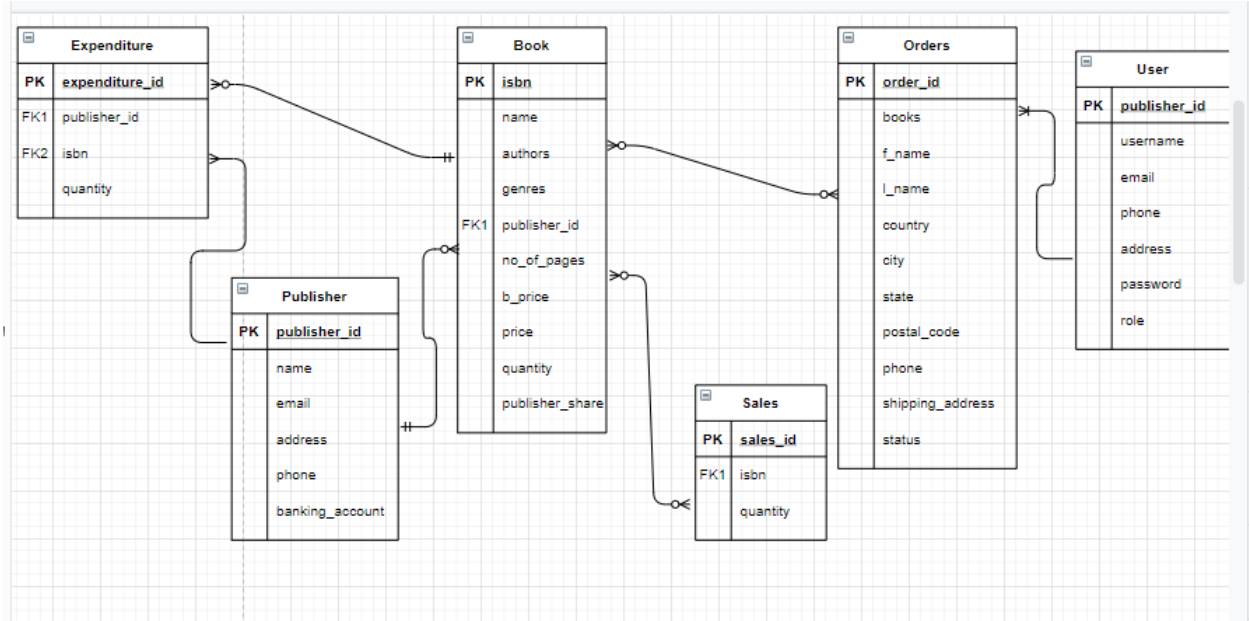

Figure 1 ER-Diagram of the database

Assumptions Made in the Diagram Regarding Cardinalities and Participation Type

Book and Expenditure Tables

These tables have a one-mandatory-to-many-optional relationship. This means that an expenditure must have only one type of book but a book can be found in many expenditures.

Book and Orders Tables

Book and order tables have a many-optional-to-many-optional relationship. An order placed by a user can have multiple books and a single book can be found in many orders.

Publisher and Book Tables

These tables have a one-mandatory-to-many-optional relationship. According to the ER diagram, a book can only have one publisher whereas a publisher can publish many books.

Publisher and Expenditure Tables

Publisher and Expenditure tables have a one-to-many relationship. This means that an expenditure can only have one publisher whereas a publisher can be in many expenditures.

Book and Sales Tables

These tables have a many-to-many relationship. A single sale can have many books and a book can be found in multiple sales.

User and Orders Tables

These tables have a one-to-many relationship. According to the design of the database, a user can place many orders but an order can only be placed by one user.

2.2. Reduction to Relation Schemas

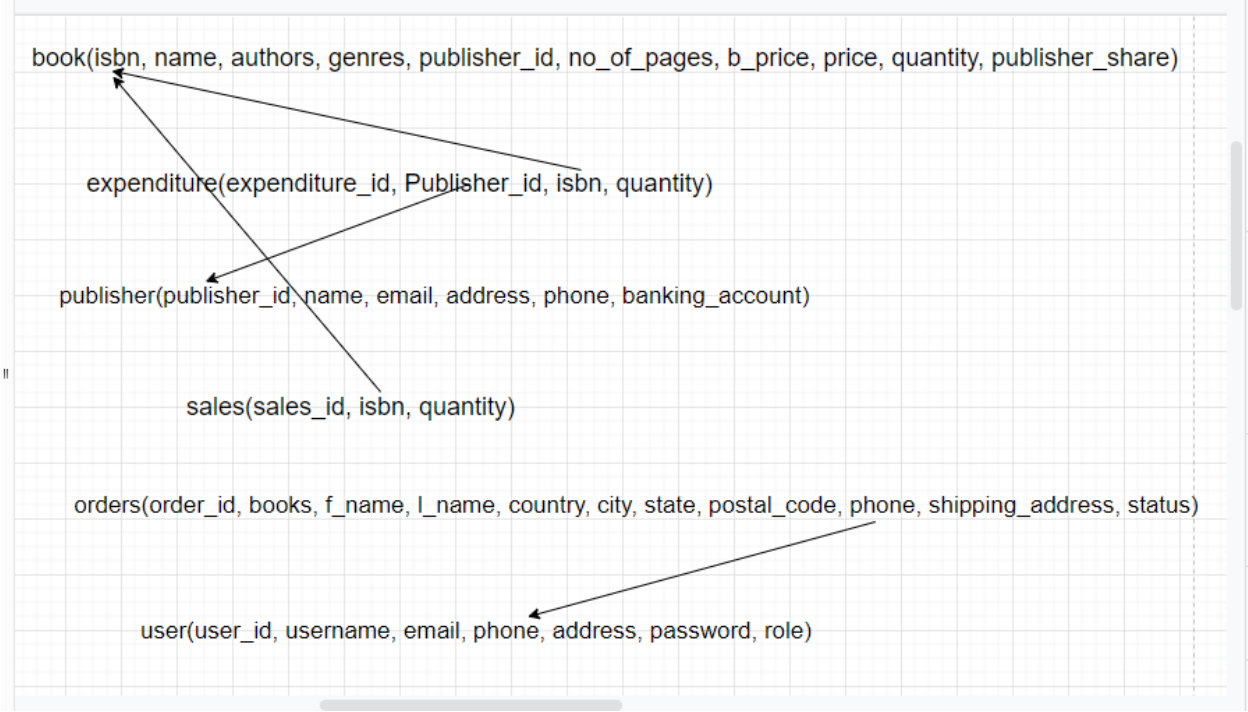

Figure 2 Relation Schema

2.3. Normalization of Relation Schemas

Functional Dependencies

Multivalued Dependency

There is a multivalued dependency in the book table where the name, authors, genres, publisher_id, no_of_pages, b_price, price, quantity, and publisher_share are functionally dependent on ISBN.

An example of this dependency can be represented as below.

isbn -> name

isbn -> authors

Trivial Functional Dependency

This type of functional dependency is found in the user table. The user_id is a subset of {user_id, username} as follows.

{user_id, username} -> user_id

Non-Trivial Functional Dependency

There is a non-trivial dependency in the book table whereby if we know the book name, we know the author of the book but the author is not a subset of the book name. This dependency can be represented as follows.

{name} -> {author}

Transitive Dependency

There is a transitive dependency in the publisher table. If we know the publisher's name, we know their email and if we know the email, we know the phone. Thus, if we know the publisher's name, we know the phone as followed.

{name} -> {email}

{email} -> {phone}

Thus, {name} -> {phone} makes sense.

The relation schema is not in the good normal form. This is because violates the rule of the third normal form since some attributes are transitively dependent. For instance, the publisher's name,

email, and phone in the publisher table are transitively dependent. To make the relation schema be in a good normal form, we can decompose the table as follows.

Publisher (publisher_id, name, email, address, banking_account)

Contact (email, phone)

2.4. Database Schema Diagram

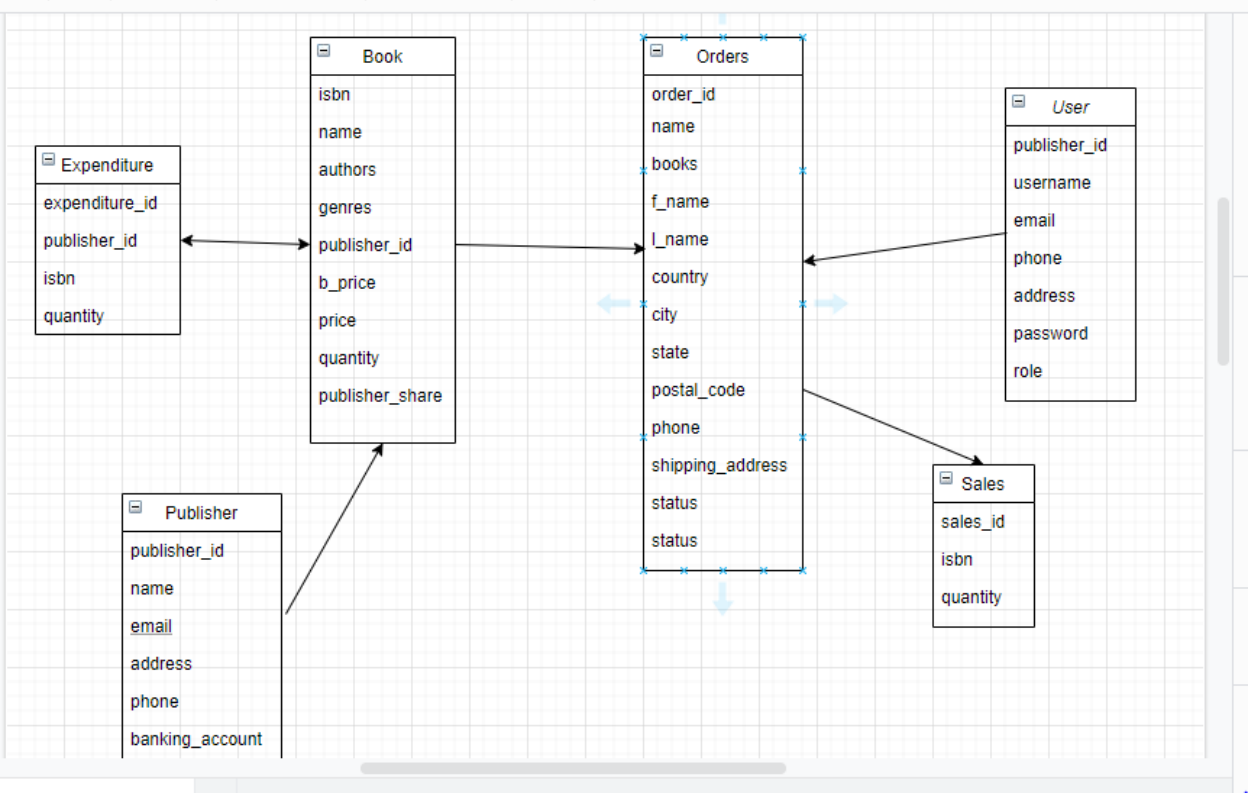

Figure 3 Database Schema Diagram

2.5. Implementation

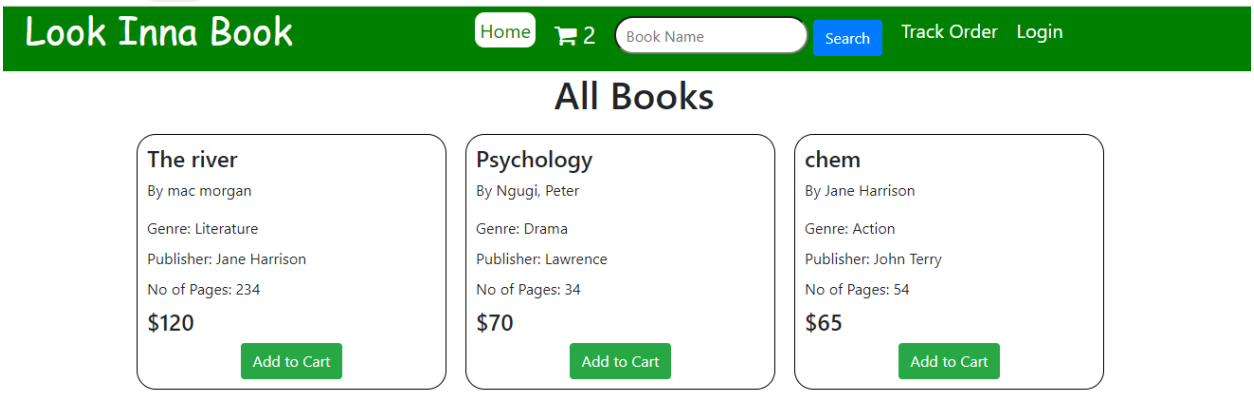

Figure 4 User Interface

Figure 5 Admin Interface