

Project Report

Conceptual Design

As description about the bookstore, the ERD for the online bookstore as following:

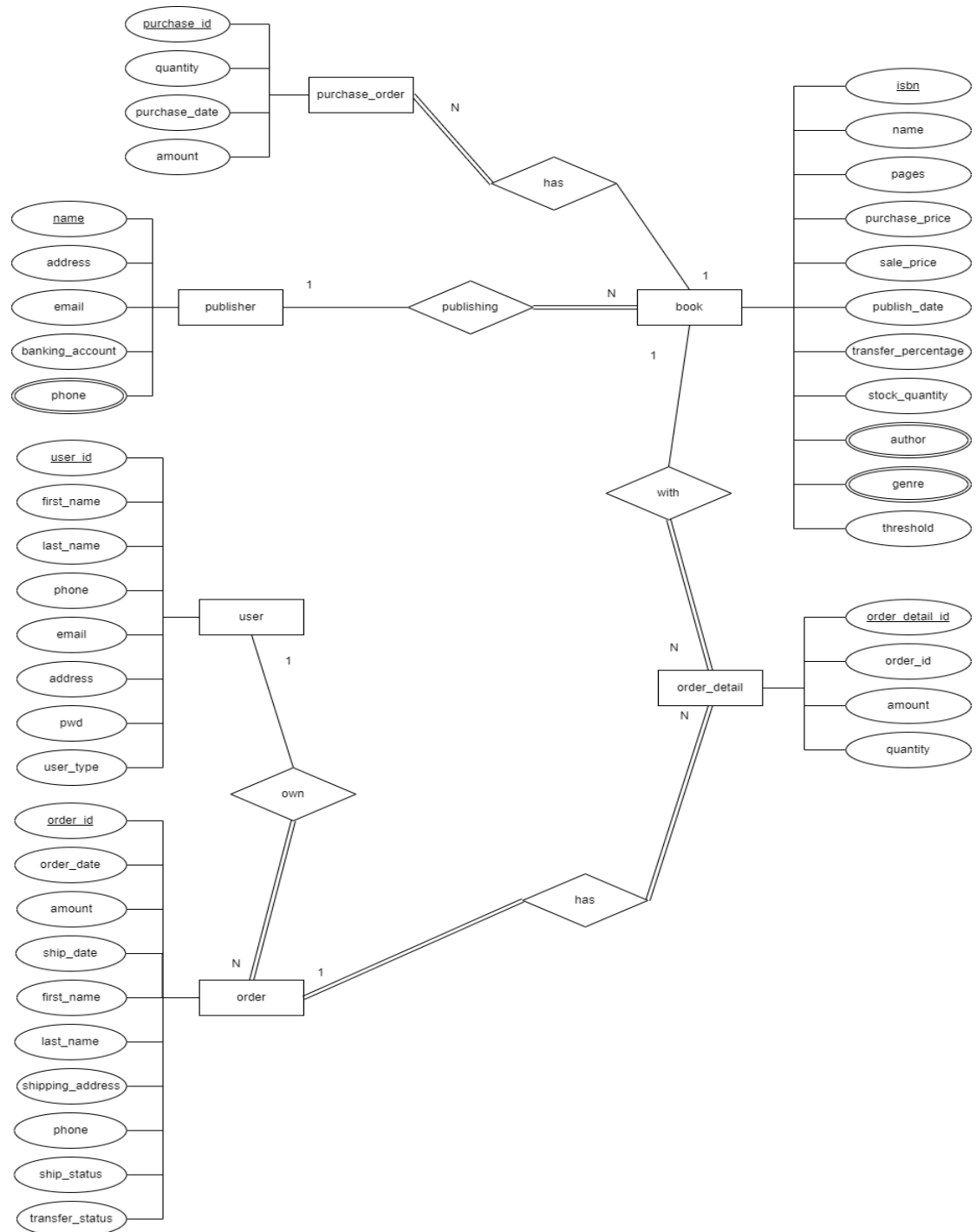


Figure 1-ERD

As showing in the diagram.

1. All books must have one publisher; One publisher may have many books.

In the description, the owner of the bookstore should transfer a percentage of the sales of books to the publishers and the percentage of different books is different, so in the book entity, I added transfer percentage attribute to record the transfer percentage, when the user needs to transfer sale to publishers, he/she can use these attributes to calculate the sales finally.

The owner should have access to report that show sales vs. expenditures, sales per genres, sales per author, so the book entity has the genres, authors, sale price and purchase price. With these attributes, the program can compute the statistic information for different factor by using these attributes with GROUP function in SQL .

2. One user may have many orders and one order must be belong to one user.

When one user wants to buy books online, he/she must register and login the system firstly, when the user registering, he/she should enter his/her name, phone, address email and user id and password. Because of the owner of the bookstore also needs to login the system to manage the information, so I add user type in entity user to distinguish different users.

When one user buys some books, he/she may use new name, address or phone as the shipping information, so I added these attributes in the entity. In my design, when the owner of bookstore transfer sales to publisher, he will transfer according to the sale order and when he has transfer one order, the database should record the transfer status, so the order entity has transfer status attribute.

In the order entity, it has one shipping status attribute and this attribute only record the final shipping status. If the user wants to find all shipping status, it should add new shipping status entity to stratify the requirements. In my design, I didn't put the shipping information as one entity because of the relationship between order and shipping information is 1:1.

3. One order at least has one detail item and order detail items must be belong to one order. As the order entity only records the basic information of the sale, in the order detail entity, it will record the sale details. Every order detail record will be associate with book ISBN and includes the quantity the user buying.
4. One order detail must have one book and one book may belong to one order detail.
5. When the books' quantity in stock is less than the threshold, the program will purchase book automatically. One purchase order must be with one book and one book may be belong to purchase order.

In the order entity, it has recorded the sale date, so when one book's quantity in stock is less than the threshold, the program will use the sale date to find the count of the book in the previous month and order this book automatically from the publishers. The purchase entity will record the purchase information from the publishers.

Reduction to Relation Schemas

When completing designing the ER diagram, I convert ER diagram to relationship schema as following:



Figure 2-Relation Schema

Normalization of Relation Schemas

In the ER diagram, attribute phone in entity publisher and author, genre in entity book are multi values, so the schema is not in 1NF. In order to make the schema in 3NF, I add new entity into the schema and split phone from publisher and author and genre from book. After splitting these attributes, the final schema is in 3NF. The relation schema as following:

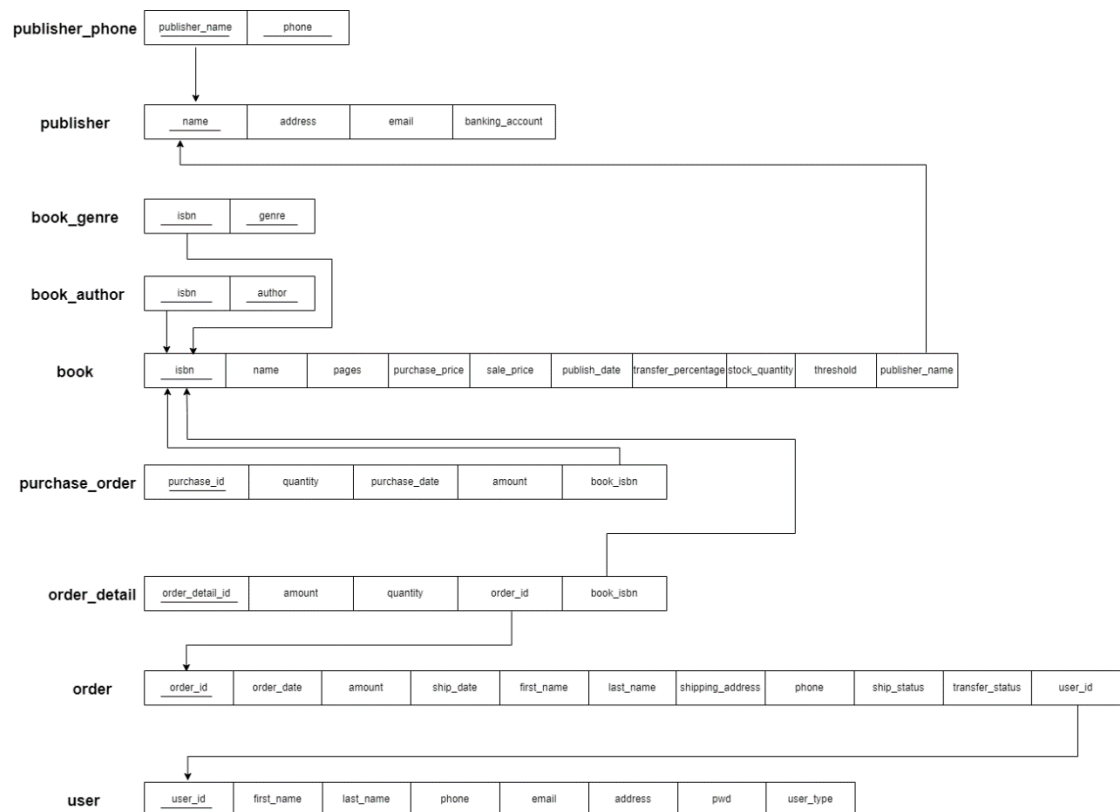


Figure 3-Normation of Relationship Schema

When completing normalizing, the schema is in 3NF. When user inserting, deleting or

updating record in the schema, it will not make the operation anomaly. For example, when the program display book with one same author, it can use table book_author join book to get the exact result.

Database Schema Diagram

From the relation schema in figure 3, it can get the database schema easily. One entity in the relation schema will be converted into one table and the reference relationship will be converted to foreign key references. After converting every entity in the relationship schema, the database schema is showing as following:

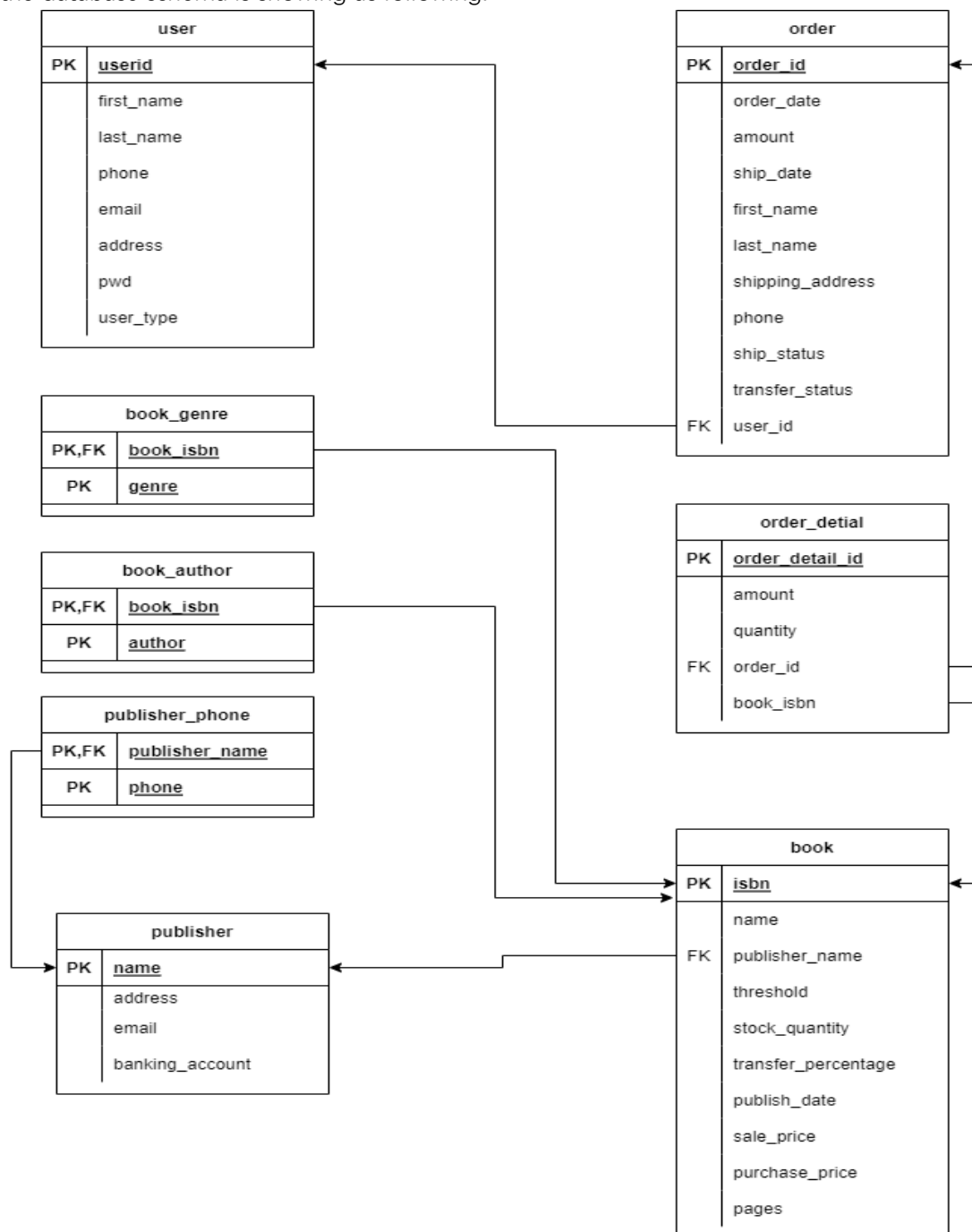


Figure 4-Database Schema

COMP3005 project

Team member: ChangpengHou 101105315

Github link: <https://github.com/timothyhou8927/Timothy1>