

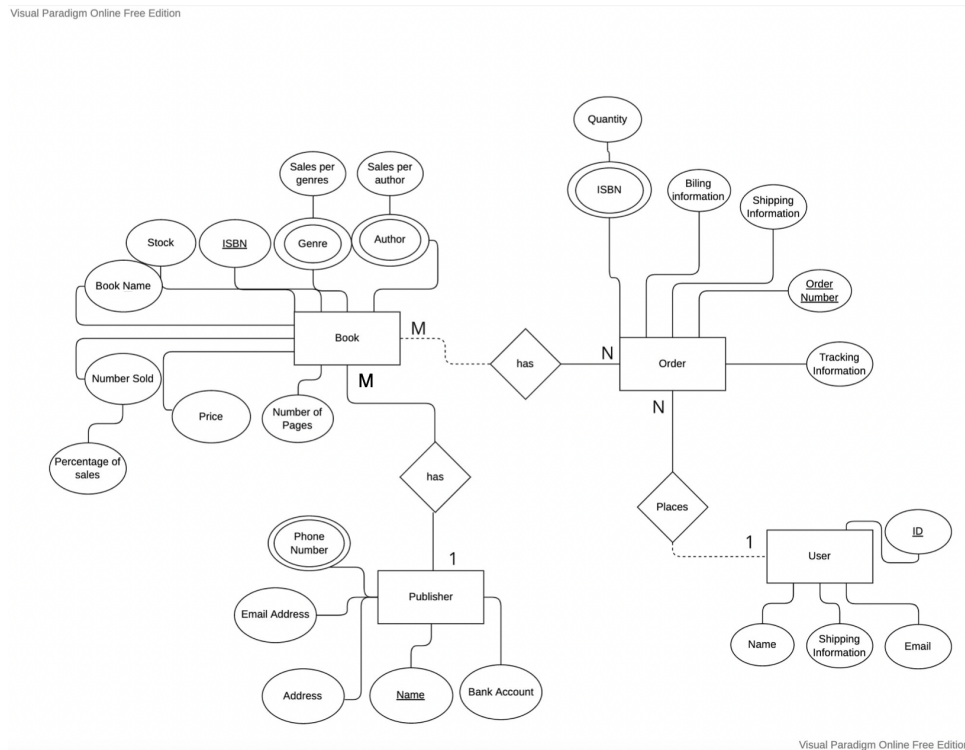
COMP 3005 - F2022

Term Project - Bookstore Database

Earle Estrella - 100907269

Rachel Wong - 101184274

Part 1 Conceptual Design



A book can only have a publisher but a publisher can publish more than one book.

An user can place an order when they want to and they can also place multiple orders.

An order can contain more than one book and a book can be ordered by more than one user(different orders).

Part 2 Reduction to Relation Schemas

The relation schemas are created from the four main relations shown as rectangles in the diagram in part 1. The Book relation is the most important one containing all the information related to all the books. Each Book has a publisher.

Book {ISBN, BookName, Author, Genre, Author Sales, Genre Sales, Price, Number of Pages, Stock, Publisher, Number Sold}

The publisher contains contact information and banking info to send them money when resupplying books.

Publisher{Name, Email, Address, Phone, Bank Account}

The user ID is all the information regarding the customer and ways to contact or deliver the products to them.

User{ID, Name, Email, Billing Info, Shipping Info}

The order relation deals with keeping track of all the transaction between user customers and the book store.

Order{Order Number, Billing Info, Shipping Info, Tracking Info, User ID, ISBN, Quantity}

Part 3 Normalization of Relation Schemas

R = Book

Dependencies:

ISBN \rightarrow BookName, Publisher

ISBN \rightarrow Author

ISBN \rightarrow Genre

Author \rightarrow Author Sales

Genre \rightarrow Genre Sales

Publisher \rightarrow Number of Pages

Number of Pages \rightarrow Price

Price \rightarrow Number Sold

Number Sold \rightarrow Stock

Since ISBN \rightarrow Publisher, Publisher \rightarrow Number of Pages and so on until Number Sold \rightarrow Stock.

By rule of chain of transitivity, this would imply:

ISBN \rightarrow Number of Pages, Price, Number Sold, Stock, Publisher

R₁ = {ISBN, BookName, Number of Pages, Price, Number Sold, Stock, Publisher}

While the same can be said for ISBN \rightarrow Author and ISBN \rightarrow Genre, Author and Genre are keys as well thus leaving us with the follow relations:

R₂ = {ISBN, Author}

R₃ = {ISBN, Genre}

R₄ = {Author, Author Sales}

R₅ = {Genre, Genre Sales}

R = User

Dependencies:

ID \rightarrow Name, Email, Billing Info, Shipping Info

The User relation is pretty straightforward as all the information and data are directly related to the user and would come from the user itself hence all the attributes can be determined by the User ID, thus:

R = {ID, Name, Email, Billing Info, Shipping Info}

R = Order

Dependencies:

OrderNumber → BillingInfo, ShippingInfo, TrackingInfo, Customer

The Order relation is also similar to the User relation. OrderNumber would include all the info (BillingInfo, ShippingInfo, TrackingInfo, Customer) and therefore all the attributes are determined by OrderNumber.

$R = \{\text{OrderNumber}, \text{BillingInfo}, \text{ShippingInfo}, \text{TrackingInfo}, \text{Customer}\}$

R = publisher

Dependencies:

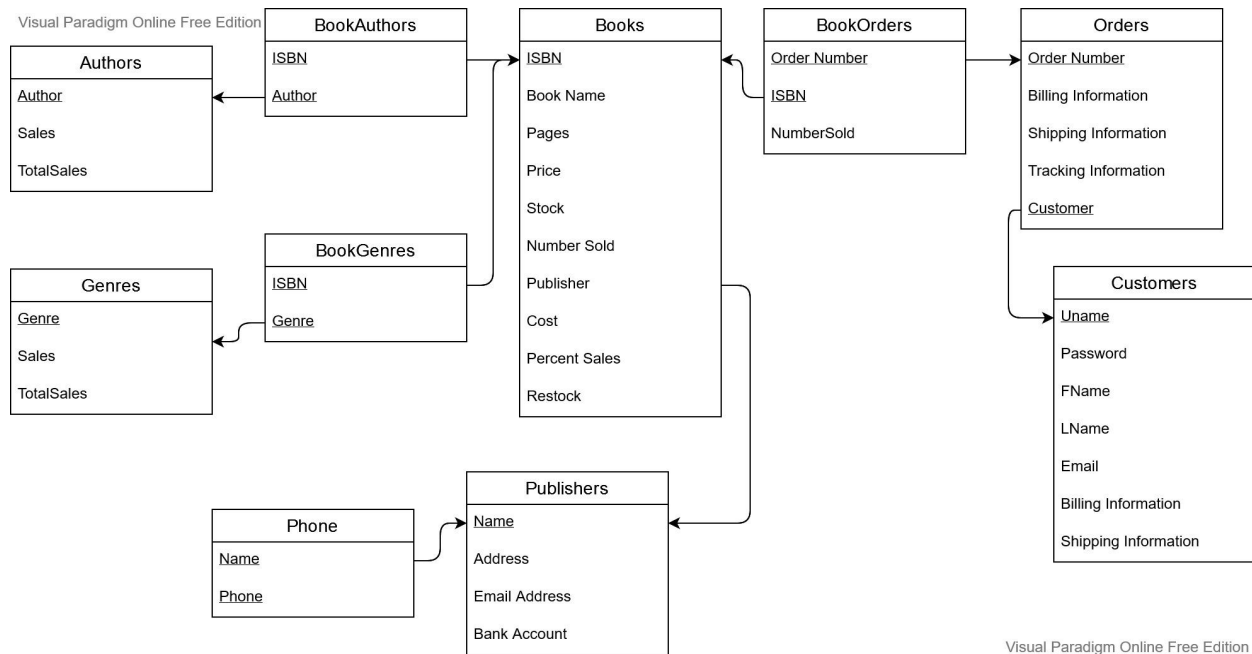
Name → Address, Email, BankAccount

The publisher relation is also similar to the above relation. The name of a publisher would include all the publisher personal information (address, email, bankAccount). All the attributes are determined by name.

$R = \{\text{Name}, \text{Address}, \text{Email}, \text{BankAccount}\}$

Part 4 Database Schema Diagram

Ideally after the normalization of the relational schemas, that should give an idea on what the final database schema diagram should look like. To take into account redundancy for multiple entries (Authors and Genres for Books, Phones for Publishers and Books for Orders), a new relation was made such that extraneous data was removed. Some attributes were added, split up or modified as the implementation of the program was underway. This includes username (uname) and password in customers relation and cost and restock in books relation.



Part 5 Implementation

Customer-side Client:

Registration, Login and Logout:

When we register a new user, we insert the information (Username, password, first name, last name, billing info, shipping info) to the database. They should be able to login after registration.

List of Books:

The search function in this page can be used to look for a book that has information containing the input string. By using regex, the string can be broken down into tokens separated by a white space and these tokens are then compared to strings contained in the book information. This includes not only the name of the book but also the genre, author and publisher.

The add to cart functionality was added to the list of books to allow the user to select multiple books to be added to the cart. To reduce clutter on the page, only 1 of each book selected is added to the cart. If the user has several books they want to buy then they can edit the amount before submission in the /order page in their cart.

Specific Books:

The specific book page displays all the relevant information the customer can use to decide on whether the book is interesting enough to them or not so they can choose to purchase it. The add to cart functionality was added to each book specific page. This allows the user to add several books of 1 type in the cart in one go.

Order/Finalize:

The order page contains text boxes to allow the user to manually edit the quantity of each item in the cart in case they change their mind. They can also delete items in the cart by making use of the checkboxes and the delete from cart button.

The customer can change the billing and shipping information for the order but the information is pre-populated with the information tied to the user who is currently logged in. This allows for the user to send gifts to friends or families.

Once the customer has finalized the orders, a record of the transaction will be added to the database. A record of the order will contain all the shipping information and the user tied to the order as well as a randomly generated tracking information given to the customer if the transaction is successful. Also, the list of books will be added along with the ordernumber tied to this record on a different relation to prevent redundancy.

Once these records are added into the relations, information on the books, authors and genres need to be changed. By updating the fields in them, the program keeps track of how many of each book is bought, how many are still in stock, and also updates the relevant or similar

information in the authors and genres relation. This helps in providing useful information such as who is the most popular author or which genre are books being bought more often from.

Tracking Information Search:

The user would type a tracking number and it will be sent into the server(orders). If it is found, then it should return the corresponding order information and its status back to the server.

Owner-side Client:

Login/Logout:

We only have one owner at this point. The account name and password are "owner".

Add Books:

Similar to adding new orders, adding a book requires it to be added to the database. The owner is prompted on this page to input any relevant information regarding the book. Once the owner inputs data and presses the add book button, this information is then validated before being sent to the server to make sure the right information is sent. Once the server receives the book information, it will add it to the books relation in the database.

Since each book may contain multiple authors and multiple genres, their respective relations (BookAuthors and BookGenres) are also updated to keep track of which authors or genres are tied to the book.

Remove Books:

When the user delete a book(by clicking a delete button), the isbn will be sent to the database. If a book is found, then it will be deleted in the database(bookauthors,book, bookgenres). Once the database is modified, we will send a response on the server side and updated the interface.

Display Reports and Finances:

The finances are broken down into three different pages. The first two pages are the authors and genres pages. These two pages are simple as its main purpose is to display the number of books sold and the total amount of sales from specific authors and genres. This will help identify which author or genre sells the most and sells the least. It helps the owner in deciding how often they should restock or buy new books under each author or genre.

The main finances page shows a list of all the books and the financial analysis to each book. It shows the cost to buy the books from the publisher, the total sales of each book, the cut the publisher receives from the sales and the profit the store gets from this transaction. This gives a good idea on how to manage the business side of owning a bookstore with profits in mind. This also shows whether a book is in need of restocking by showing a restock button if the book is less than the threshold of stocks remaining.

Part 6 Bonus Features

Search feature was improved so instead of just searching for names of books, it can search for any information within the book's list of attributes. This allows the user to search for a name, a publisher, a genre and an author at the same time. This was done by using RegEx.

Another feature that was added was allowing the user to modify the cart just before submission. This way, when a user changes their mind or accidentally added a lot of books in the cart in the books list page, they can remove a lot of books at the same time or modify the number of a specific book before buying.

Some additional mini features are login/logout on both customers and owner. They can also register if they are not existing user. The new customer's info will be sent to the database after they register so they can login to it right after registration.

Part 7 GitHub Repository

GitHub Link:

<https://github.com/Estearle/COMP3005-Project-F22>