

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
VIỆN TRÍ TUỆ NHÂN TẠO

BÁO CÁO MÔN HỌC CƠ SỞ DỮ LIỆU
ĐỀ TÀI
QUẢN LÝ CỬA HÀNG BÁN SÁCH ONLINE

Nhóm sinh viên thực hiện: Group 100:

1. Nguyễn Gia Huy
2. Hoàng Văn Dương
3. Phạm Tiến Dũng
4. Nguyễn Thế Cường

Mục lục

I.	Chọn chủ đề	3
II.	Mô tả Thực Thể cho Cơ Sở Dữ Liệu	3
III.	Chuyển mô hình ER sang mô hình quan hệ với các bảng tương ứng và phụ thuộc hàm, sau đó chuẩn hóa thành dạng 3NF:	7
IV.	Tạo trong DBMS một cơ sở dữ liệu tương ứng với mô hình quan hệ.	14
V.	Tạo ràng buộc (constraints) bằng cách sử dụng câu lệnh « alter table »	18
VI.	Chèn dữ liệu vào cơ sở dữ liệu	19
a)	Chèn dữ liệu vào bảng Country	19
b)	Chèn dữ liệu vào bảng City	19
c)	Chèn dữ liệu vào bảng User	20
d)	Chèn dữ liệu vào bảng Language.....	20
e)	Chèn dữ liệu vào bảng Author	21
f)	Chèn dữ liệu vào bảng Book	22
g)	Chèn dữ liệu vào bảng Review	22
h)	Chèn dữ liệu vào bảng Cart	23
i)	Chèn dữ liệu vào bảng Orders.....	23
j)	Chèn dữ liệu vào bảng Wishlist.....	24
k)	Chèn dữ liệu vào bảng Payment.....	24
VII.	Viết truy vấn cho từng yêu cầu:	25
VIII.	Viết giao dịch (transaction) sử dụng lệnh ROLLBACK	29
IX.	Viết trigger.	30
X.	Viết thủ tục (procedure):	34
XI.	Kết luận	36

I. Chọn chủ đề

Trong một thế giới ngày càng số hóa, **cửa hàng sách trực tuyến** trở thành một phần không thể thiếu đối với những người yêu sách. Hệ thống này không chỉ giúp người dùng dễ dàng tìm kiếm và mua sách mà còn yêu cầu một cơ sở dữ liệu mạnh mẽ để quản lý sách, đơn hàng và người dùng một cách hiệu quả. Trong bài tập lớn này, em sẽ xây dựng **cơ sở dữ liệu cho cửa hàng sách trực tuyến**, từ việc thiết kế bảng cho đến quản lý thông tin khách hàng, sách, đơn hàng và thanh toán. Đây là cơ hội để em áp dụng kiến thức về cơ sở dữ liệu vào một hệ thống thực tế, giúp em hiểu rõ hơn về cách vận hành các dịch vụ thương mại điện tử.

II. Mô tả Thực Thể cho Cơ Sở Dữ Liệu

1. Country

- **Mô tả:** Lưu trữ thông tin về các quốc gia nơi các thành phố và người dùng cư trú. Bảng này được tham chiếu bởi bảng City để xác định thành phố thuộc quốc gia nào.

2. City

- **Mô tả:** Lưu trữ thông tin về các thành phố nơi người dùng sinh sống. Mỗi thành phố được liên kết với một quốc gia thông qua khóa ngoại tham chiếu đến bảng Country.

3. User

- **Mô tả:** Lưu trữ thông tin về người dùng của hệ thống, bao gồm cả quản trị viên và khách hàng. Thông tin này bao gồm chi tiết cá nhân, vai trò và liên kết với bảng City để xác định nơi cư trú.

4. Language

- **Mô tả:** Lưu trữ thông tin về các ngôn ngữ mà sách trong hệ thống sử dụng. Bảng Book tham chiếu đến bảng này thông qua khóa ngoại.

5. Book

- **Mô tả:** Lưu trữ thông tin về các cuốn sách có trong hệ thống. Mỗi cuốn sách được liên kết với một tác giả và ngôn ngữ. Bảng này còn chứa thông tin về giá, thể loại và số lượng sách tồn kho.

6. Author

- **Mô tả:** Lưu trữ thông tin về các tác giả của sách trong hệ thống. Bảng này lưu giữ tiểu sử và được tham chiếu bởi bảng Book.

7. Review

- **Mô tả:** Lưu trữ thông tin đánh giá của khách hàng về sách. Mỗi đánh giá được liên kết với một cuốn sách cụ thể và người dùng đã thực hiện đánh giá đó.

8. Cart

- **Mô tả:** Lưu trữ thông tin về giỏ hàng của người dùng. Mỗi giỏ hàng thuộc về một người dùng cụ thể và chứa chi tiết như tổng số sách, tổng giá trị, và các chiết khấu áp dụng.

9. Orders

- **Mô tả:** Lưu trữ thông tin về các đơn hàng của người dùng. Mỗi đơn hàng liên kết với một giỏ hàng và chứa thông tin về địa chỉ giao hàng, trạng thái, và thanh toán.

10.Payment

- **Mô tả:** Lưu trữ thông tin về thanh toán cho các đơn hàng. Bảng này theo dõi trạng thái thanh toán, phương thức thanh toán và tham chiếu đến bảng User và Orders.

11.Wishlist

- **Mô tả:** Lưu trữ thông tin về các sách mà người dùng muốn lưu để mua sau. Mỗi mục trong bảng này liên kết với một giỏ hàng và một cuốn sách cụ thể.

Mô hình Quan hệ

1. Country → City

- **Quan hệ:** Một-nhiều (One-to-Many)
- **Khóa ngoại:** City.Country → Country.Country_Id
- **Mô tả:** Một quốc gia có thể chứa nhiều thành phố.

2. City → User

- **Quan hệ:** Một-nhiều (One-to-Many)
- **Khóa ngoại:** User.City → City.Pincode
- **Mô tả:** Một thành phố có thể có nhiều người dùng cư trú.

3. Language → Book

- **Quan hệ:** Một-nhiều (One-to-Many)

- **Khóa ngoại:** Book.Language → Language.Language_Id
- **Mô tả:** Một ngôn ngữ có thể áp dụng cho nhiều cuốn sách.

4. Author → Book

- **Quan hệ:** Một-nhiều (One-to-Many)
- **Khóa ngoại:** Book.Author_Id → Author.Author_Id
- **Mô tả:** Một tác giả có thể viết nhiều cuốn sách.

5. User → Cart

- **Quan hệ:** Một-một (One-to-One)
- **Khóa ngoại:** Cart.User_Id → User.User_Id
- **Mô tả:** Mỗi người dùng chỉ có một giỏ hàng.

6. Cart → Orders

- **Quan hệ:** Một-một (One-to-One)
- **Khóa ngoại:** Orders.Cart_Id → Cart.Cart_Id
- **Mô tả:** Mỗi giỏ hàng chỉ tạo ra một đơn hàng.

7. User → Orders

- **Quan hệ:** Một-nhiều (One-to-Many)
- **Khóa ngoại:** Orders.Customer_Id → User.User_Id
- **Mô tả:** Một người dùng có thể đặt nhiều đơn hàng.

8. Orders → Payment

- **Quan hệ:** Một-một (One-to-One)
- **Khóa ngoại:** Payment.Order_Id → Orders.Order_Id
- **Mô tả:** Mỗi đơn hàng chỉ có một thông tin thanh toán.

9. User → Payment

- **Quan hệ:** Một-nhiều (One-to-Many)
- **Khóa ngoại:** Payment.Customer_Id → User.User_Id

- **Mô tả:** Một người dùng có thể thực hiện nhiều thanh toán.

10. **User** → **Review**

- **Quan hệ:** Một-nhiều (One-to-Many)
- **Khóa ngoại:** Review.Customer_Id → User.User_Id
- **Mô tả:** Một người dùng có thể viết nhiều đánh giá.

11. **Book** → **Review**

- **Quan hệ:** Một-nhiều (One-to-Many)
- **Khóa ngoại:** Review.Book_Id → Book.Book_Id
- **Mô tả:** Một cuốn sách có thể nhận được nhiều đánh giá.

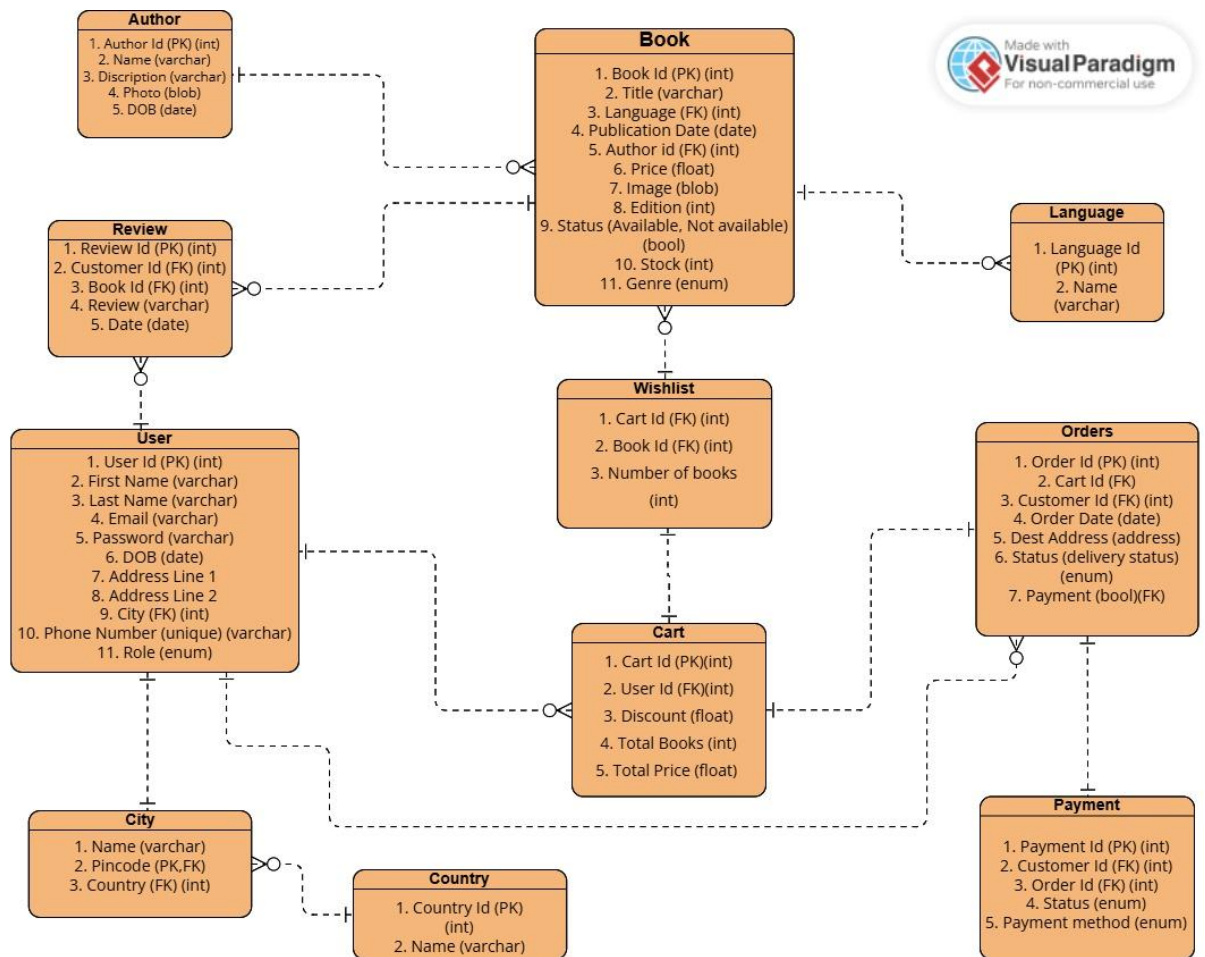
12. **Cart** → **Wishlist**

- **Quan hệ:** Một-nhiều (One-to-Many)
- **Khóa ngoại:** Wishlist.Cart_Id → Cart.Cart_Id
- **Mô tả:** Một giỏ hàng có thể chứa nhiều mục trong danh sách yêu thích.

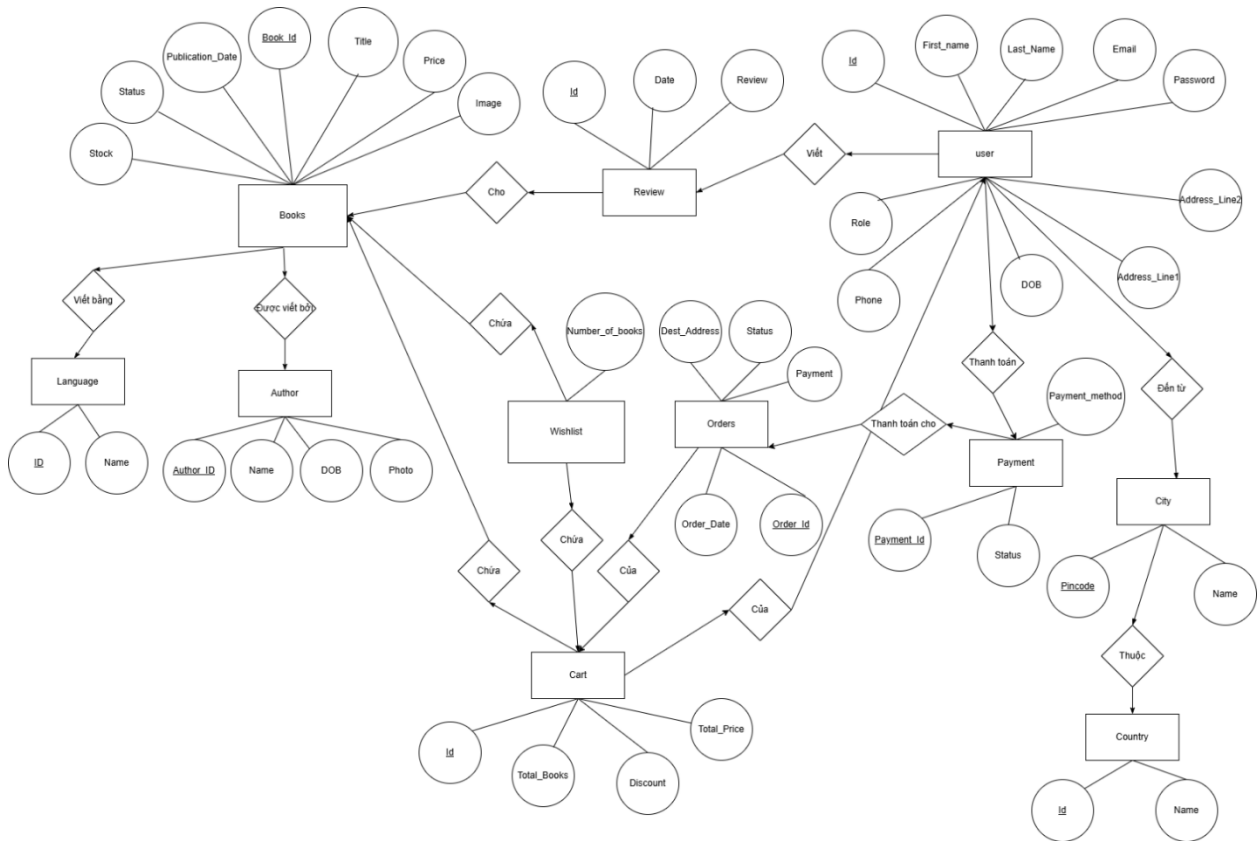
13. **Book** → **Wishlist**

- **Quan hệ:** Một-nhiều (One-to-Many)
- **Khóa ngoại:** Wishlist.Book_Id → Book.Book_Id
- **Mô tả:** Một cuốn sách có thể xuất hiện trong nhiều danh sách yêu thích.

III. Chuyển mô hình ER sang mô hình quan hệ với các bảng tương ứng và phụ thuộc hàm, sau đó chuẩn hóa thành dạng 3NF:



Mô hình thực thể/mối quan hệ đề xuất cho hệ thống :



Thực thể và thuộc tính:

1. Country:

- **Country_Id (Primary Key)**: Mã quốc gia
- **Name**: Tên quốc gia

2. City:

- **Pincode (Primary Key)**: Mã bưu điện thành phố
- **Name**: Tên thành phố
- **Country (Foreign Key)**: Mã quốc gia tham chiếu đến bảng Country

3. User:

- **User_Id (Primary Key)**: Mã người dùng
- **First_Name**: Tên người dùng
- **Last_Name**: Họ người dùng
- **Email**: Email người dùng (Unique)
- **Password**: Mật khẩu người dùng
- **DOB**: Ngày sinh

- **Address_Line1:** Địa chỉ 1
- **Address_Line2:** Địa chỉ 2
- **City (Foreign Key):** Thành phố của người dùng (tham chiếu đến bảng City)
- **Phone_Number:** Số điện thoại (Unique)
- **Role:** Vai trò người dùng (Admin hoặc Customer)

4. Language:

- **Language_Id (Primary Key):** Mã ngôn ngữ
- **Name:** Tên ngôn ngữ

5. Book:

- **Book_Id (Primary Key):** Mã sách
- **Title:** Tiêu đề sách
- **Language (Foreign Key):** Mã ngôn ngữ của sách (tham chiếu đến bảng Language)
- **Publication_Date:** Ngày xuất bản
- **Author_Id (Foreign Key):** Mã tác giả của sách (tham chiếu đến bảng Author)
- **Price:** Giá sách (Kiểm tra điều kiện ≥ 0)
- **Image:** Hình ảnh sách
- **Edition:** Phiên bản sách
- **Status:** Trạng thái sách (còn hay hết hàng)
- **Stock:** Số lượng sách trong kho (Kiểm tra điều kiện ≥ 0)
- **Genre:** Thể loại sách (Fiction, Non-Fiction, Science, Biography, Other)

6. Author:

- **Author_Id (Primary Key):** Mã tác giả
- **Name:** Tên tác giả
- **Description:** Mô tả về tác giả
- **Photo:** Ảnh tác giả
- **DOB:** Ngày sinh tác giả

7. Review:

- **Review_Id (Primary Key):** Mã đánh giá
- **Customer_Id (Foreign Key):** Mã khách hàng (tham chiếu đến bảng User)
- **Book_Id (Foreign Key):** Mã sách (tham chiếu đến bảng Book)
- **Review:** Nội dung đánh giá

- **Date:** Ngày đánh giá

8. Cart:

- **Cart_Id (Primary Key):** Mã giỏ hàng
- **User_Id (Foreign Key):** Mã người dùng (tham chiếu đến bảng User)
- **Total_Books:** Tổng số sách trong giỏ hàng (Kiểm tra điều kiện ≥ 0)
- **Total_Price:** Tổng giá trị đơn hàng (Kiểm tra điều kiện ≥ 0)
- **Discount:** Chiết khấu

9. Orders:

- **Order_Id (Primary Key):** Mã đơn hàng
- **Cart_Id (Foreign Key):** Mã giỏ hàng (tham chiếu đến bảng Cart)
- **Customer_Id (Foreign Key):** Mã khách hàng (tham chiếu đến bảng User)
- **Order_Date:** Ngày đặt hàng
- **Dest_Address:** Địa chỉ giao hàng
- **Status:** Trạng thái đơn hàng (Pending, Shipped, Delivered, Cancelled)
- **Payment:** Thanh toán (True hoặc False)

10. Payment:

- **Payment_Id (Primary Key):** Mã thanh toán
- **Customer_Id (Foreign Key):** Mã khách hàng (tham chiếu đến bảng User)
- **Order_Id (Foreign Key):** Mã đơn hàng (tham chiếu đến bảng Orders)
- **Status:** Trạng thái thanh toán (Success, Pending, Failed)
- **Payment_Method:** Phương thức thanh toán (Credit Card, Debit Card, Net Banking, UPI, Cash on Delivery)

11. Wishlist:

- **Cart_Id (Foreign Key):** Mã giỏ hàng (tham chiếu đến bảng Cart)
- **Book_Id (Foreign Key):** Mã sách (tham chiếu đến bảng Book)
- **Number_of_Books:** Số lượng sách yêu thích (Kiểm tra điều kiện ≥ 1)

Các phụ thuộc hàm:

1. Country:

- {Country_Id} -> {Name}

2. City:

- {Pincode} -> {Name, Country}

3. User:

- {User_Id} -> {First_Name, Last_Name, Email, Password, DOB, Address_Line1, Address_Line2, City, Phone_Number, Role}

4. Language:

- {Language_Id} -> {Name}

5. Book:

- {Book_Id} -> {Title, Language, Publication_Date, Author_Id, Price, Image, Edition, Status, Stock, Genre}

6. Author:

- {Author_Id} -> {Name, Description, Photo, DOB}

7. Review:

- {Review_Id} -> {Customer_Id, Book_Id, Review, Date}

8. Cart:

- {Cart_Id} -> {User_Id, Total_Books, Total_Price, Discount}

9. Orders:

- {Order_Id} -> {Cart_Id, Customer_Id, Order_Date, Dest_Address, Status, Payment}

10. Payment:

- {Payment_Id} -> {Customer_Id, Order_Id, Status, Payment_Method}

11. Wishlist:

- {Cart_Id, Book_Id} -> {Number_of_Books}

Mối quan hệ trong 3NF

1. Trong Country:

R1(Country) = (Country_Id, Name)

Functional Dependencies:

Country_Id -> Name

2. Trong City:

R2(City) = (Pincode, Name, Country)

Functional Dependencies:

Pincode -> Name, Country

3. Trong User:

R3(User) = (User_Id, First_Name, Last_Name, Email, Password, DOB, Address_Line1, Address_Line2, City, Phone_Number, Role)

Functional Dependencies:

User_Id -> First_Name, Last_Name, Email, Password, DOB, Address_Line1, Address_Line2, City, Phone_Number, Role

R4(Language) = (Language_Id, Name)

Functional Dependencies:

Language_Id -> Name

5. Trong Book:

R5(Book) = (Book_Id, Title, Language, Publication_Date, Author_Id, Price, Image, Edition, Status, Stock, Genre)

Functional Dependencies:

Book_Id -> Title, Language, Publication_Date, Author_Id, Price, Image, Edition, Status, Stock, Genre

6. Trong Author:

R6(Author) = (Author_Id, Name, Description, Photo, DOB)

Functional Dependencies:

Author_Id -> Name, Description, Photo, DOB

7. Trong Review:

R7(Review) = (Review_Id, Customer_Id, Book_Id, Review, Date)

Functional Dependencies:

Review_Id -> Customer_Id, Book_Id, Review, Date

8. Trong Cart:

R8(Cart) = (Cart_Id, User_Id, Total_Books, Total_Price, Discount)

Functional Dependencies:

Cart_Id -> User_Id, Total_Books, Total_Price, Discount

9. Trong Orders:

R9(Orders) = (Order_Id, Cart_Id, Customer_Id, Order_Date, Dest_Address, Status, Payment)

Functional Dependencies:

Order_Id -> Cart_Id, Customer_Id, Order_Date, Dest_Address, Status, Payment

10. Trong Payment:

R10(Payment) = (Payment_Id, Customer_Id, Order_Id, Status, Payment_Method)

Functional Dependencies:

Payment_Id -> Customer_Id, Order_Id, Status, Payment_Method

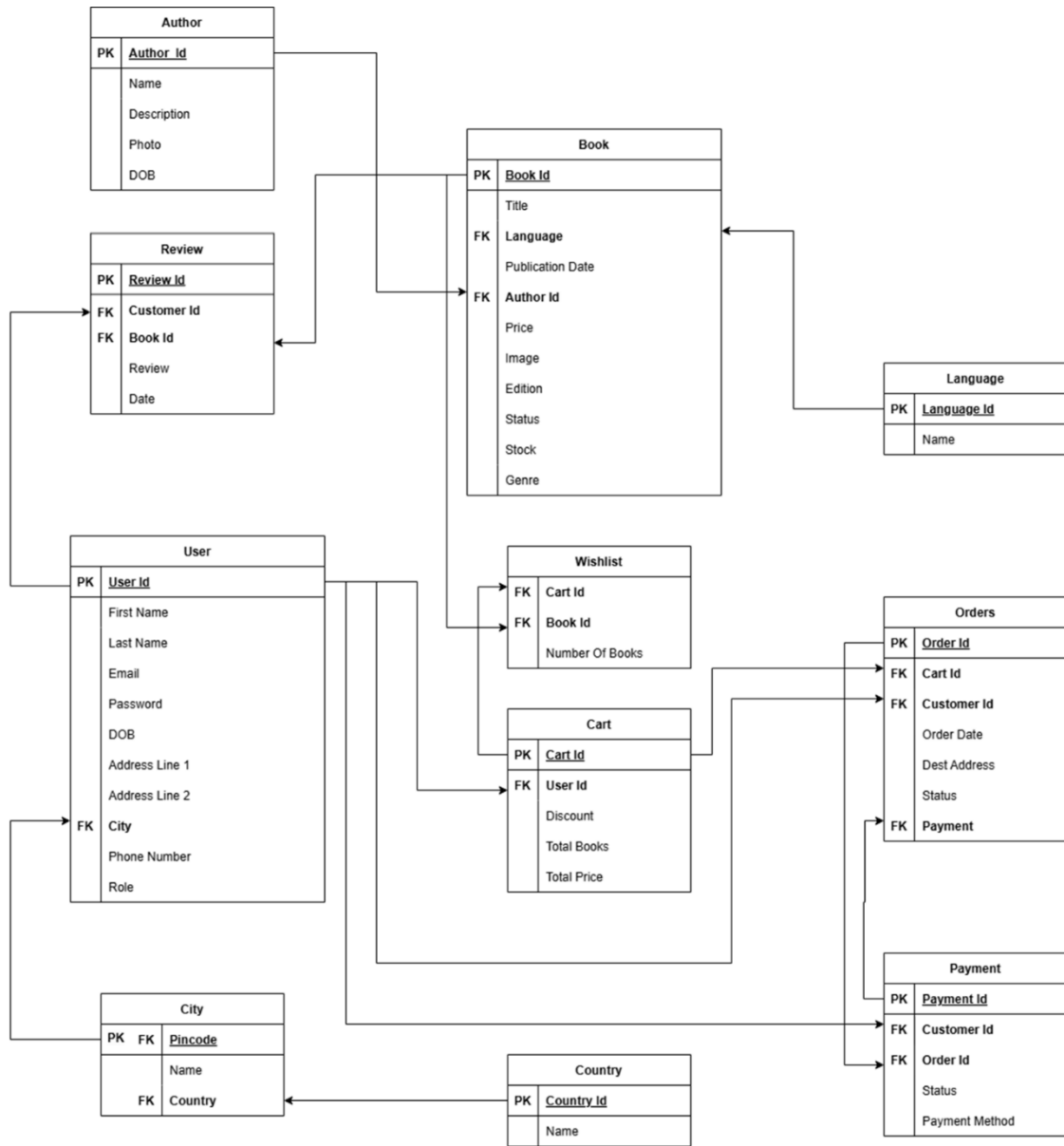
11. Trong Wishlist:

R11(Wishlist) = (Cart_Id, Book_Id, Number_of_Books)

Functional Dependencies:

Cart_Id, Book_Id -> Number_of_Books

Mô hình quan hệ cho hệ thống :



IV. Tạo trong DBMS một cơ sở dữ liệu tương ứng với mô hình quan hệ.

Báo cáo từng bảng trong mã

1. Country

- **Mô tả:** Lưu trữ thông tin về các quốc gia.
- **Các trường:**

- Country_Id: Khóa chính, tự động tăng, là định danh duy nhất cho mỗi quốc gia.
 - Name: Tên của quốc gia (không được để trống).
-

2. City

- **Mô tả:** Lưu chi tiết thông tin về các thành phố.
 - **Các trường:**
 - Pincode: Khóa chính, tự động tăng, là định danh duy nhất cho mỗi thành phố.
 - Name: Tên của thành phố (không được để trống).
 - Country: Khóa ngoại tham chiếu đến bảng Country, trường Country_Id.
-

3. User

- **Mô tả:** Lưu thông tin chi tiết của người dùng, bao gồm khách hàng và quản trị viên.
 - **Các trường:**
 - User_Id: Khóa chính, tự động tăng, là định danh duy nhất cho mỗi người dùng.
 - First_Name: Tên đầu của người dùng (không được để trống).
 - Last_Name: Họ của người dùng (tùy chọn).
 - Email: Địa chỉ email duy nhất của người dùng (không được để trống).
 - Password: Mật khẩu của tài khoản người dùng (không được để trống).
 - DOB: Ngày sinh của người dùng (tùy chọn).
 - Address_Line1: Dòng địa chỉ thứ nhất của người dùng.
 - Address_Line2: Dòng địa chỉ thứ hai của người dùng (tùy chọn).
 - City: Khóa ngoại tham chiếu đến bảng City, trường Pincode.
 - Phone_Number: Số điện thoại duy nhất (không được để trống).
 - Role: Xác định vai trò của người dùng, có thể là "Admin" hoặc "Customer."
-

4. Language

- **Mô tả:** Lưu thông tin về các ngôn ngữ có sẵn cho sách.
 - **Các trường:**
 - Language_Id: Khóa chính, tự động tăng.
 - Name: Tên của ngôn ngữ (không được để trống).
-

5. Book

- **Mô tả:** Chứa thông tin về các cuốn sách có sẵn trong hệ thống.
 - **Các trường:**
 - Book_Id: Khóa chính, tự động tăng.
 - Title: Tựa đề của cuốn sách (không được để trống).
 - Language: Khóa ngoại tham chiếu đến bảng Language, trường Language_Id.
 - Publication_Date: Ngày xuất bản của cuốn sách.
 - Author_Id: Khóa ngoại tham chiếu đến bảng Author, trường Author_Id.
 - Price: Giá của cuốn sách (phải ≥ 0).
 - Image: Hình ảnh của cuốn sách (dữ liệu nhị phân).
 - Edition: Phiên bản của cuốn sách.
 - Status: Trạng thái sẵn có (true hoặc false).
 - Stock: Số lượng sách có sẵn (phải ≥ 0).
 - Genre: Thể loại của sách (Ví dụ: Fiction, Non-Fiction).
-

6. Author

- **Mô tả:** Chứa thông tin về tác giả của sách.
 - **Các trường:**
 - Author_Id: Khóa chính, tự động tăng.
 - Name: Tên của tác giả (không được để trống).
 - Description: Thông tin chi tiết hoặc tiểu sử ngắn gọn của tác giả.
 - Photo: Ảnh của tác giả (dữ liệu nhị phân).
 - DOB: Ngày sinh của tác giả.
-

7. Review

- **Mô tả:** Lưu trữ đánh giá của khách hàng về sách.
 - **Các trường:**
 - Review_Id: Khóa chính, tự động tăng.
 - Customer_Id: Khóa ngoại tham chiếu đến bảng User, trường User_Id.
 - Book_Id: Khóa ngoại tham chiếu đến bảng Book, trường Book_Id.
 - Review: Nội dung đánh giá (tối đa 1000 ký tự).
 - Date: Ngày viết đánh giá.
-

8. Cart

- **Mô tả:** Lưu thông tin về giỏ hàng của người dùng.

- **Các trường:**
 - Cart_Id: Khóa chính, tự động tăng.
 - User_Id: Khóa ngoại tham chiếu đến bảng User, trường User_Id.
 - Total_Books: Tổng số sách trong giỏ (phải ≥ 0).
 - Total_Price: Tổng giá trị của giỏ hàng (phải ≥ 0).
 - Discount: Giảm giá áp dụng cho giỏ hàng.
-

9. Orders

- **Mô tả:** Lưu trữ thông tin đơn đặt hàng.
 - **Các trường:**
 - Order_Id: Khóa chính, tự động tăng.
 - Cart_Id: Khóa ngoại tham chiếu đến bảng Cart, trường Cart_Id.
 - Customer_Id: Khóa ngoại tham chiếu đến bảng User, trường User_Id.
 - Order_Date: Ngày đặt hàng (không được để trống).
 - Dest_Address: Địa chỉ giao hàng (không được để trống).
 - Status: Trạng thái hiện tại của đơn hàng (Ví dụ: Pending, Shipped).
 - Payment: Trạng thái thanh toán (true hoặc false).
-

10. Payment

- **Mô tả:** Quản lý thông tin thanh toán cho các đơn đặt hàng.
 - **Các trường:**
 - Payment_Id: Khóa chính, tự động tăng.
 - Customer_Id: Khóa ngoại tham chiếu đến bảng User, trường User_Id.
 - Order_Id: Khóa ngoại tham chiếu đến bảng Orders, trường Order_Id.
 - Status: Trạng thái thanh toán (Ví dụ: Success, Pending).
 - Payment_Method: Phương thức thanh toán (Ví dụ: Credit Card, UPI).
-

11. Wishlist

- **Mô tả:** Lưu các sách mà người dùng muốn lưu lại để mua sau.
 - **Các trường:**
 - Cart_Id: Khóa ngoại tham chiếu đến bảng Cart, trường Cart_Id.
 - Book_Id: Khóa ngoại tham chiếu đến bảng Book, trường Book_Id.
- Number_of_Books: Số lượng sách (phải ≥ 1).

V. Tạo ràng buộc (constraints) bằng cách sử dụng câu lệnh « alter table »

```
ALTER TABLE City
  ADD CONSTRAINT FK_City_Country FOREIGN KEY (Country) REFERENCES
  Country (Country_Id);
ALTER TABLE User
  ADD CONSTRAINT FK_User_City FOREIGN KEY (City) REFERENCES City
  (Pincode);

ALTER TABLE Book
  ADD CONSTRAINT FK_Book_Language FOREIGN KEY (Language)
  REFERENCES Language (Language_Id),
  ADD CONSTRAINT FK_Book_Author FOREIGN KEY (Author_Id) REFERENCES
  Author (Author_Id);

ALTER TABLE Cart
  ADD CONSTRAINT FK_Cart_User FOREIGN KEY (User_Id) REFERENCES User
  (User_Id);

ALTER TABLE Orders
  ADD CONSTRAINT FK_Orders_Cart FOREIGN KEY (Cart_Id) REFERENCES
  Cart (Cart_Id),
  ADD CONSTRAINT FK_Orders_Customer FOREIGN KEY (Customer_Id)
  REFERENCES User (User_Id);

ALTER TABLE Payment
  ADD CONSTRAINT FK_Payment_Customer FOREIGN KEY (Customer_Id)
  REFERENCES User (User_Id),
  ADD CONSTRAINT FK_Payment_Order FOREIGN KEY (Order_Id)
  REFERENCES Orders (Order_Id);

ALTER TABLE Review
  ADD CONSTRAINT FK_Review_Customer FOREIGN KEY (Customer_Id)
  REFERENCES User (User_Id),
  ADD CONSTRAINT FK_Review_Book FOREIGN KEY (Book_Id) REFERENCES
  Book (Book_Id);

ALTER TABLE Wishlist
  ADD CONSTRAINT FK_Wishlist_Cart FOREIGN KEY (Cart_Id) REFERENCES
  Cart (Cart_Id),
  ADD CONSTRAINT FK_Wishlist_Book FOREIGN KEY (Book_Id) REFERENCES
  Book (Book_Id)
```

VI. Chèn dữ liệu vào cơ sở dữ liệu

a) Chèn dữ liệu vào bảng Country

```
INSERT INTO Country (Country_Id, Name) VALUES  
(1, 'United States'),  
(2, 'United Kingdom'),  
(3, 'France'),  
(4, 'Germany'),  
(5, 'China');
```

Country_Id	Name
1	United States
2	United Kingdom
3	France
4	Germany
5	China

b) Chèn dữ liệu vào bảng City

```
INSERT INTO City (Pincode, Name, Country) VALUES  
(1, 'New York', 1),  
(2, 'London', 2),  
(3, 'Paris', 3),  
(4, 'Berlin', 4),  
(5, 'Shanghai', 5);
```

Pincode ↑	Name	Country
Sort Pincode	Filter...	Filter...
1	New York	1
2	London	2
3	Paris	3
4	Berlin	4
5	Shanghai	5


c) Chèn dữ liệu vào bảng User

```
INSERT INTO User (User_Id, First_Name, Last_Name, Email, Password, DOB, Address_Line1, City, Phone_Number, Role) VALUES
(1, 'Alice', 'Smith', 'alicesmith01@gmail.com', 'Alice@123', '1990-01-01', '123 Elm St', 1, '1234567890', 'Customer'),
(2, 'Bob', 'Johnson', 'bobjohnson22@gmail.com', 'Bob@456!', '1985-02-15', '456 Oak St', 2, '9876543211', 'Customer'),
(3, 'Charlie', 'Brown', 'charliebrow36@gmail.com', 'Charlie#789$', '1975-03-30', '789 Pine St', 3, '1928374652', 'Customer'),
(4, 'Daisy', 'Miller', 'daisymiller49@gmail.com', 'Daisy@321!', '2000-05-20', '654 Maple St', 4, '5647382913', 'Customer'),
(5, 'Eve', 'Davis', 'evedavis58@gmail.com', 'Eve#654$', '1995-07-15', '321 Cedar St', 1, '8273649584', 'Customer'),
(6, 'Fan', 'Xiaoping', 'fxy1506@gmail.com', 'Fxy#jqkA', '2005-06-15', '17 Kuomintang St', 5, '8271239584', 'Customer');
```

User_Id	First_Name	Last_Name	Email	Password	DOB
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
1	Alice	Smith	alicesmith01@gmail.com	Alice@123	1990-01-01
2	Bob	Johnson	bobjohnson22@gmail.com	Bob@456!	1985-02-15
3	Charlie	Brown	charliebrow36@gmail.com	Charlie#789\$	1975-03-30
4	Daisy	Miller	daisymiller49@gmail.com	Daisy@321!	2000-05-20
5	Eve	Davis	evedavis58@gmail.com	Eve#654\$	1995-07-15
6	Fan	Xiaoping	fxy1506@gmail.com	Fxy#jqkA	2005-06-15






d) Chèn dữ liệu vào bảng Language

```
INSERT INTO Language (Language_Id, Name) VALUES
(1, 'English'),
(2, 'Spanish'),
(3, 'French'),
(4, 'German'),
(5, 'Chinese'),
(6, 'Japanese');
```

Language_Id	Name
 Filter...	 Filter...
1	English
2	Spanish
3	French
4	German
5	Chinese
6	Japanese







e) Chèn dữ liệu vào bảng Author

```
INSERT INTO Author (Author_Id, Name, Description, Photo, DOB) VALUES
(1, 'J.K. Rowling', 'Author of Harry Potter series', 'photo1.jpg', '1965-07-31'),
(2, 'Gabriel García Márquez', 'Author of Cien años de soledad and Love in the Time of Cholera', 'photo2.jpg', '1927-03-06'),
(3, 'George Orwell', 'Author of dystopian novels', 'photo3.jpg', '1903-06-25'),
(4, 'Jane Austen', 'Author of Pride and Prejudice', 'photo4.jpg', '1775-12-16'),
(5, 'Victor Hugo', 'Author of Les Misérables', 'photo5.jpg', '1802-02-26'),
(6, 'Johann Wolfgang von Goethe', 'Author of Faust and pioneer of German literature', 'photo6.jpg', '1749-08-28'),
(7, 'Cao Xueqin', 'Author of Dream of the Red Chamber', 'photo7.jpg', '1715-01-01');
```

Author_Id	Name	Description	Photo	DOB
 Filter...	 Filter...	 Filter...	 Filter...	 Filter...
1	J.K. Rowling	Author of Harry Potter series	70686f746f312e6a7067	1965-07-31
2	Gabriel García Márquez	Author of Cien años de soledad and Love in the Time of Cholera	70686f746f322e6a7067	1927-03-06
3	George Orwell	Author of dystopian novels	70686f746f332e6a7067	1903-06-25
4	Jane Austen	Author of Pride and Prejudice	70686f746f342e6a7067	1775-12-16
5	Victor Hugo	Author of Les Misérables	70686f746f352e6a7067	1802-02-26
6	Johann Wolfgang von G...	Author of Faust and pioneer of German literature	70686f746f362e6a7067	1749-08-28
7	Cao Xueqin	Author of Dream of the Red Chamber	70686f746f372e6a7067	1715-01-01






f) Chèn dữ liệu vào bảng Book

```
INSERT INTO Book (Book_Id, Title, Language, Publication_Date, Author_Id, Price, Edition, Stock, Status, Genre) VALUES
(1, 'Harry Potter and the Philosopher''s Stone', 1, '1997-06-26', 1, 29.99, 1, 100, TRUE, 'Fiction'),
(2, 'Harry Potter and the Chamber of Secrets', 1, '1998-07-02', 1, 31.99, 1, 90, TRUE, 'Fiction'),
(3, 'Cien anos de soledad', 2, '1967-06-05', 2, 25.99, 1, 60, TRUE, 'Non-Fiction'),
(4, 'Love in the Time of Cholera', 1, '1985-03-06', 2, 28.99, 1, 55, TRUE, 'Science'),
(5, '1984', 1, '1949-06-08', 3, 20.99, 1, 70, TRUE, 'Non-Fiction'),
(6, 'Animal Farm', 1, '1945-08-17', 3, 15.99, 1, 85, TRUE, 'Other'),
(7, 'Pride and Prejudice', 1, '1813-01-28', 4, 18.99, 1, 100, TRUE, 'Science'),
(8, 'Les Misérables', 3, '1862-04-03', 5, 25.99, 1, 50, TRUE, 'Non-Fiction'),
(9, 'Faust', 4, '1808-01-01', 6, 30.00, 1, 20, TRUE, 'Biography'),
(10, 'Dream of the Red Chamber', 5, '1791-01-01', 7, 35.99, 1, 40, TRUE, 'Biography');
```

Book_Id	Title	Language	Publication_Date	Author_Id	Price
 Filter...	 Filter...	 Filter...	 Filter...	 Filter...	 Filter...
1	Harry Potter and the Philosopher's Stone	1	1997-06-26	1	29.99
2	Harry Potter and the Chamber of Secrets	1	1998-07-02	1	31.99
3	Cien anos de soledad	2	1967-06-05	2	25.99
4	Love in the Time of Cholera	1	1985-03-06	2	28.99
5	1984	1	1949-06-08	3	20.99
6	Animal Farm	1	1945-08-17	3	15.99
7	Pride and Prejudice	1	1813-01-28	4	18.99
8	Les Misérables	3	1862-04-03	5	25.99
9	Faust	4	1808-01-01	6	30
10	Dream of the Red Chamber	5	1791-01-01	7	35.99






g) Chèn dữ liệu vào bảng Review

```
INSERT INTO Review (Review_Id, Customer_Id, Book_Id, Review, Date) VALUES
(1, 1, 1, 'Great book! A magical journey that introduces readers to the enchanting world of Hogwarts.', '2024-12-05'),
(2, 1, 5, 'A chilling look into a dystopian world, must read for 1984 fans.', '2024-12-07'),
(3, 3, 6, 'A powerful political allegory that critiques the corruption of power through a group of farm animals.', '2024-12-11'),
(4, 3, 7, 'A timeless romance with witty dialogue and sharp social commentary, highly recommended it!', '2024-12-18'),
(5, 5, 9, 'A deeply philosophical work that delves into human desires, morality, and the quest for meaning.', '2024-12-19');
```

Review_Id	Customer_Id	Book_Id	Review	Date
 Filter...	 Filter...	 Filter...	 Filter...	 Fil
1	1	1	Great book! A magical journey that introduces readers to the enchanting world of Hogwarts.	2024-12-05
2	1	5	A chilling look into a dystopian world, must read for 1984 fans.	2024-12-07
3	3	6	A powerful political allegory that critiques the corruption of power through a group of farm animals.	2024-12-11
4	3	7	A timeless romance with witty dialogue and sharp social commentary, highly recommended it!	2024-12-18
5	5	9	A deeply philosophical work that delves into human desires, morality, and the quest for meaning.	2024-12-19








h) Chèn dữ liệu vào bảng Cart

```
INSERT INTO Cart (Cart_Id, User_Id, Total_Books, Total_Price, Discount) VALUES
(1, 4, 1, 25.99, 0.00),
(2, 2, 3, 77.97, 5.00),
(3, 1, 3, 71.97, 2.00),
(4, 5, 1, 30.00, 0.00),
(5, 6, 3, 107.97, 3.00),
(6, 3, 3, 53.97, 1.00);
```

Cart_Id	User_Id	Total_Books	Total_Price	Discount
 Filter...	 Filter...	 Filter...	 Filter...	 Filter...
1	4	1	25.99	0
2	2	3	77.97	5
3	1	3	71.97	2
4	5	1	30	0
5	6	3	107.97	3
6	3	3	53.97	1

i) Chèn dữ liệu vào bảng Orders

```
INSERT INTO Orders (Order_Id, Cart_Id, Customer_Id, Order_Date, Dest_Address, Status, Payment) VALUES
(1, 3, 1, '2024-11-25', '123 Elm St, New York', 'Shipped', TRUE),
(2, 2, 2, '2024-12-03', '456 Oak St, London', 'Pending', FALSE),
(3, 6, 3, '2024-12-07', '789 Pine St, Paris', 'Shipped', TRUE),
(4, 1, 4, '2024-12-12', '654 Maple St, Berlin', 'Delivered', TRUE),
(5, 4, 5, '2024-12-13', '321 Cedar St, New York', 'Shipped', TRUE),
(6, 5, 6, '2024-12-15', '17 Kuomintang St, Shanghai', 'Delivered', TRUE);
```

Order_Id	Cart_Id	Customer_Id	Order_Date	Dest_Address	Status	Payment
 Filter...	 Filter...	 Filter...	 Filter...	 Filter...	 Filter...	 Filter...
1	3	1	2024-11-25	123 Elm St, New York	Shipped	1
2	2	2	2024-12-03	456 Oak St, London	Pending	0
3	6	3	2024-12-07	789 Pine St, Paris	Shipped	1
4	1	4	2024-12-12	654 Maple St, Berlin	Delivered	1
5	4	5	2024-12-13	321 Cedar St, New York	Shipped	1
6	5	6	2024-12-15	17 Kuomintang St, Shan...	Delivered	1

j) Chèn dữ liệu vào bảng Wishlist

```
INSERT INTO Wishlist (Cart_Id, Book_Id, Number_Of_Books) VALUES
(1, 8, 1),
(2, 3, 3),
(3, 1, 1),
(3, 5, 2),
(4, 9, 1),
(5, 10, 3),
(6, 6, 1),
(6, 7, 2);
```

Cart_Id	Book_Id	Number_of_Books
<input type="text" value="a b c Filter..."/>	<input type="text" value="a b c Filter..."/>	<input type="text" value="a b c Filter..."/>
1	8	1
2	3	3
3	1	1
3	5	2
4	9	1
5	10	3
6	6	1
6	7	2

k) Chèn dữ liệu vào bảng Payment

```
INSERT INTO Payment (Payment_Id, Customer_Id, Order_Id, Status, Payment_Method) VALUES
(1, 1, 1, 'Success', 'Debit Card'),
(2, 2, 2, 'Pending', 'Cash on Delivery'),
(3, 3, 3, 'Success', 'Credit Card'),
(4, 3, 4, 'Success', 'Debit Card'),
(5, 4, 5, 'Success', 'Net Banking'),
(6, 5, 6, 'Success', 'Cash on Delivery');
```

Payment_Id	Customer_Id	Order_Id	Status	Payment_Method
<input type="text" value="a b c Filter..."/>	<input type="text" value="a b c Filter..."/>	<input type="text" value="a b c Filter..."/>	<input type="text" value="a b c Filter..."/>	<input type="text" value="a b c Filter..."/>
1	1	1	Success	Debit Card
2	2	2	Pending	Cash on Delivery
3	3	3	Success	Credit Card
4	3	4	Success	Debit Card
5	4	5	Success	Net Banking
6	5	6	Success	Cash on Delivery

VII. Viết truy vấn cho từng yêu cầu:

- a. Truy vấn sử dụng inner join.

```
SELECT user.User_Id as User_Id, First_Name, Last_Name, sum(Total_Books)
as total_books
FROM cart
    INNER JOIN user on cart.User_Id = user.User_Id
where Role = 'Customer'
GROUP BY User_Id;
```

User_Id	First_Name	Last_Name	total_books
abc Filter...	abc Filter...	abc Filter...	abc Filter...
4	Daisy	Miller	1
2	Bob	Johnson	3
1	Alice	Smith	3
5	Eve	Davis	1
6	Fan	Xiaoping	3
3	Charlie	Brown	3

- b. Truy vấn sử dụng outer join.

```
SELECT book.Title AS book_title, language.Name AS Language
FROM book LEFT JOIN language
ON book.Language = language.Language_Id
UNION
SELECT book.Title AS book_title, language.Name AS Language
FROM book RIGHT JOIN language
ON book.Language = language.Language_Id;
```

book_title	Language
abc Filter...	abc Filter...
Harry Potter and the Philosopher's Stone	English
Harry Potter and the Chamber of Secrets	English
Cien anos de soledad	Spanish
Love in the Time of Cholera	English
1984	English
Animal Farm	English
Pride and Prejudice	English
Les Misérables	French
Faust	German
Dream of the Red Chamber	Chinese
NULL	Japanese

c. Sử dụng subquery trong where.

```

SELECT Title
FROM book as b
WHERE b.Book_Id IN (SELECT Book_Id FROM Orders WHERE Status =
'Shipped');
-- Tìm sách có giá lớn hơn giá trung bình của tất cả các sách
SELECT Title, Price
FROM Book
WHERE Price > (SELECT AVG(Price) FROM Book);
-- Tìm người dùng có đánh giá sách có id là 1
SELECT First_Name, Last_Name, (SELECT Review FROM Review WHERE
Customer_Id = User.User_Id AND Book_Id = 1) AS Review
FROM User
WHERE User_Id IN (SELECT Customer_Id FROM Review WHERE Book_Id = 1)
AND EXISTS (SELECT 1 FROM Review WHERE Customer_Id = User.User_Id
AND Book_Id = 1 AND Review IS NOT NULL);

```

Title
abc Filter...
Harry Potter and the Philosopher's Stone
Harry Potter and the Chamber of Secrets
Cien anos de soledad
Love in the Time of Cholera
1984
Animal Farm
Pride and Prejudice
Les Misérables
Faust
Dream of the Red Chamber

Title	Price
abc Filter...	abc Filter...
Harry Potter and the Philosopher's Stone	29.99
Harry Potter and the Chamber of Secrets	31.99
Love in the Time of Cholera	28.99
Faust	30
Dream of the Red Chamber	35.99

First_Name	Last_Name	Review
abc Filter...	abc Filter...	abc Filter...
Alice	Smith	Great book! A magical journey that introduces readers to the enchanting world of Hogwarts.

d. Sử dụng subquery trong from.

```
-- Mỗi một quyển sách có bao nhiêu lượt review
SELECT b.*, (SELECT COUNT(r.Review_Id) FROM review r WHERE r.Book_Id =
b.Book_Id) AS Total_Review
FROM book b;
-- Mỗi User đã mua bao nhiêu quyển sách?
SELECT u.User_Id,
       u.First_Name AS NAME,
       (SELECT SUM(c.Total_Books) FROM cart c WHERE c.User_Id = u.User_Id) AS
Total_Books_Buy
FROM user u;
```

Book_Id	Title	Language	Publication_Date	Author_Id
abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...
1	Harry Potter and the Philosopher's Stone	1	1997-06-26	1
2	Harry Potter and the Chamber of Secrets	1	1998-07-02	1
3	Cien anos de soledad	2	1967-06-05	2
4	Love in the Time of Cholera	1	1985-03-06	2
5	1984	1	1949-06-08	3
6	Animal Farm	1	1945-08-17	3
7	Pride and Prejudice	1	1813-01-28	4
8	Les Misérables	3	1862-04-03	5
9	Faust	4	1808-01-01	6
10	Dream of the Red Chamber	5	1791-01-01	7

User_Id	NAME	Total_Books_Buy
abc Filter...	abc Filter...	abc Filter...
1	Alice	3
2	Bob	3
3	Charlie	3
4	Daisy	1
5	Eve	1
6	Fan	3

e. Truy vấn sử dụng group by và aggregate functions.

```
-- Mỗi một tác giả đã viết ra bao nhiêu quyển sách?
SELECT a.Author_Id, a.Name, COUNT(b.Book_Id) AS Total_Books
FROM author a
      JOIN book b ON a.Author_Id = b.Author_Id
GROUP BY a.Author_Id;

-- Mỗi một user đã review bao nhiêu bài
SELECT u.User_Id, u.Last_Name AS Name, COUNT(r.Review_Id) AS Total_Reviews
FROM user u
      JOIN review r ON u.User_Id = r.Customer_Id
GROUP BY u.User_Id;
```

Author_Id	Name	Total_Books
abc Filter...	abc Filter...	abc Filter...
1	J.K. Rowling	2
2	Gabriel García Márquez	2
3	George Orwell	2
4	Jane Austen	1
5	Victor Hugo	1
6	Johann Wolfgang von G...	1
7	Cao Xueqin	1

User_Id	Name	Total_Reviews
abc Filter...	abc Filter...	abc Filter...
1	Smith	2
3	Brown	2
5	Davis	1

VIII. Viết giao dịch (transaction) sử dụng lệnh **ROLLBACK**.

```
-- Thêm 1 khách hàng mới vào bảng, sau đó quay lại trước khi thêm
START TRANSACTION;
INSERT INTO User (User_Id, First_Name, Last_Name, Email, Password, DOB,
Address_Line1, City, Phone_Number, Role)
VALUES (7, 'Ivan', 'Brown', 'ivan.brown@example.com', 'Ivan123!', '1985-12-
25', '456 Brown St', 1, '1112223333', 'Customer');
ROLLBACK;

-- Cập nhật trạng thái của 1 đơn hàng, sau đó quay lại trước khi thêm
START TRANSACTION;
UPDATE Orders
SET Status = 'Delivered', Payment = TRUE
WHERE Order_Id = 3;
ROLLBACK;

-- Xóa sách và review về sách, sau đó quay lại trước khi thêm
DELETE FROM Review
```

```

WHERE Book_Id = 6;
DELETE FROM Book
WHERE Book_Id = 6;
ROLLBACK;

-- Thêm sản phẩm cho 1 người dùng vào giỏ hàng, cập nhật số lượng sách, sau
đó quay lại ban đầu
START TRANSACTION;
-- 1. Thêm sản phẩm vào wishlist của người dùng
INSERT INTO Wishlist (Cart_Id, Book_Id, Number_Of_Books)
VALUES (10, 2, 3);
-- 2. Cập nhật giỏ hàng
UPDATE Cart
SET cart.Total_Books = cart.Total_Books + (SELECT Number_of_Books FROM
Wishlist WHERE Book_Id = 2),
   cart.Total_Price = cart.Total_Price + ((SELECT Number_of_Books FROM
Wishlist WHERE Book_Id = 2) * (SELECT Price FROM Book WHERE Book_Id = 2))
WHERE Cart_Id = 10;
-- 3. Cập nhật số lượng sách trong bảng Book (Giảm số lượng sách tồn kho)
UPDATE Book
SET Stock = Stock - (SELECT Number_of_Books FROM Wishlist WHERE Book_Id
= 2)
WHERE Book_Id = 2;
ROLLBACK;

```

IX. Viết trigger.

Trong quá trình sử dụng hệ thống, trigger đóng vai trò quan trọng trong quy trình hoạt động đặt hàng, khi đặt hàng trigger sẽ tự động kiểm tra xem kho hàng có còn hay không, tăng giảm số lượng tồn kho khi đơn hàng được đặt, hay hủy. Đồng thời kiểm tra trạng thái thanh toán của đơn hàng tương ứng.

```

-- 1. Giảm stock của sách khi đặt hàng
DROP TRIGGER IF EXISTS tr_update_stock_after_order;
DELIMITER $$
CREATE TRIGGER tr_update_stock_after_order
AFTER INSERT ON Orders
FOR EACH ROW
BEGIN
  DECLARE book_id_tmp INT;

```

```

DECLARE book_quantity INT;
DECLARE new_stock INT;
DECLARE done INT DEFAULT 0;
DECLARE cur CURSOR FOR
    SELECT w.Book_Id, w.Number_of_books
    FROM Wishlist w
    WHERE w.Cart_Id = NEW.Cart_Id;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
IF (NEW.Status != 'Cancelled') THEN
    OPEN cur;

    read_loop: LOOP
        FETCH cur INTO book_id_tmp, book_quantity;
        IF done THEN
            LEAVE read_loop;
        END IF;
        SELECT Stock INTO new_stock FROM Book WHERE Book_Id =
book_id_tmp;
        IF new_stock < book_quantity THEN
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Insufficient stock for
order. Please reduce quantity.';
        END IF;
        UPDATE Book
        SET Stock = Stock - book_quantity
        WHERE Book_Id = book_id_tmp;
    END LOOP;
END IF;
CLOSE cur;
END$$
DELIMITER ;

-- 2. Thực hiện việc tăng stock lên khi order bị cancel
DROP TRIGGER IF EXISTS tr_update_stock_after_cancel_order;
DELIMITER $$
CREATE TRIGGER tr_update_stock_after_cancel_order
AFTER UPDATE ON Orders
FOR EACH ROW
BEGIN
    DECLARE book_id_tmp INT;
    DECLARE book_quantity INT;
    DECLARE done INT DEFAULT 0;

```

```

SELECT w.Book_Id, w.Number_of_books
FROM Wishlist w
WHERE w.Cart_Id = NEW.Cart_Id;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
IF (NEW.Status = 'Cancelled') AND (OLD.Status != 'Cancelled') THEN
OPEN cur;
read_loop: LOOP
    FETCH cur INTO book_id_tmp, book_quantity;
    IF done THEN
        LEAVE read_loop;
    END IF;
    UPDATE Book
    SET Stock = Stock + book_quantity
    WHERE Book_Id = book_id_tmp;
END LOOP;
CLOSE cur;
END IF;
END$$
DELIMITER ;

-- 3. Cập nhật total_book và total_price trong cart khi thêm hoặc sửa wishlist
DROP TRIGGER IF EXISTS tr_update_cart_on_wishlist_insert;
DELIMITER $$
CREATE TRIGGER tr_update_cart_on_wishlist_insert
AFTER INSERT ON Wishlist
FOR EACH ROW
BEGIN
    DECLARE total_books INT DEFAULT 0;
    DECLARE total_price FLOAT DEFAULT 0.0;
    SELECT SUM(w.Number_of_books)
    INTO total_books
    FROM Wishlist w
    WHERE w.Cart_Id = NEW.Cart_Id;
    SELECT SUM(w.Number_of_books * b.Price)
    INTO total_price
    FROM Wishlist w
    JOIN Book b ON w.Book_Id = b.Book_Id
    WHERE w.Cart_Id = NEW.Cart_Id;
    UPDATE Cart
    SET Total_Books = total_books,
        Total_Price = total_price

```



```

    WHERE Cart_Id = NEW.Cart_Id;
END$$
DELIMITER ;

DROP TRIGGER IF EXISTS tr_update_cart_on_wishlist_update;
DELIMITER $$
CREATE TRIGGER tr_update_cart_on_wishlist_update
AFTER UPDATE ON Wishlist
FOR EACH ROW
BEGIN
    DECLARE total_books INT DEFAULT 0;
    DECLARE total_price FLOAT DEFAULT 0.0;
    SELECT SUM(w.Number_of_books)
    INTO total_books
    FROM Wishlist w
    WHERE w.Cart_Id = NEW.Cart_Id;
    SELECT SUM(w.Number_of_books * b.Price)
    INTO total_price
    FROM Wishlist w
    JOIN Book b ON w.Book_Id = b.Book_Id
    WHERE w.Cart_Id = NEW.Cart_Id;
    UPDATE Cart
    SET Total_Books = total_books,
        Total_Price = total_price
    WHERE Cart_Id = NEW.Cart_Id;
END$$
DELIMITER ;

-- 4. Khi payment chuyển thành Success thì đổi payment trong order thành 1
DROP TRIGGER IF EXISTS tr_update_order_payment_status;
DELIMITER $$
CREATE TRIGGER tr_update_order_payment_status
AFTER UPDATE ON Payment
FOR EACH ROW
BEGIN
    IF NEW.Status = 'Success' AND OLD.Status != 'Success' THEN
        UPDATE Orders
        SET Payment = 1
        WHERE Order_Id = NEW.Order_Id;
    ELSE
        UPDATE Orders
        SET Payment = 0

```

```

WHERE Order_Id = NEW.Order_Id;
END IF;
END$$
DELIMITER ;

```

X. Viết thủ tục (procedure):

```

-- 1. Insert orders mới đồng thời tạo một payment tương ứng
DELIMITER $$
CREATE PROCEDURE IF NOT EXISTS insert_order_and_payment(
    IN p_cartID INT,
    IN p_customerId INT,
    IN p_order_date DATE,
    IN p_Dest_Address VARCHAR(255),
    IN p_Status ENUM ('Pending', 'Shipped', 'Delivered', 'Cancelled'),
    IN p_Payment_Method ENUM ('Credit Card', 'Debit Card', 'Net Banking', 'UPI',
    'Cash on Delivery')
)
BEGIN
    DECLARE last_order_id INT;
    INSERT INTO orders (Cart_Id, Customer_Id, Order_Date, Dest_Address, Status,
    Payment)
    VALUES (p_cartID, p_customerId, p_order_date, p_Dest_Address, p_Status, 0);
    SET last_order_id = LAST_INSERT_ID();
    INSERT INTO payment (Customer_Id, Order_Id, Status, Payment_Method)
    VALUES (p_customerId, last_order_id, 'Pending', p_Payment_Method);
END $$
DELIMITER ;

-- 2. Hiển thị thông tin sách bằng ID
CREATE PROCEDURE getBookById(IN p_Id INT)
BEGIN
    SELECT * FROM book WHERE Book_Id = p_Id;
END;

-- 3. Hiển thị các bài review của user bằng User_Id
CREATE PROCEDURE getReviewsByUserId(IN p_userId INT)
BEGIN
    SELECT u.User_Id, CONCAT(u.First_Name, ' ', u.Last_Name) AS Full_Name,
    b.Book_Id, b.Title, r.Review
    FROM user u JOIN review r ON u.User_Id = r.Customer_Id
    JOIN Book b ON r.Book_Id = b.Book_Id WHERE u.User_Id = p_userId;
END;

```

-- 4. Hiển thị số tiền mua sách của user bằng User_Id

```
CREATE PROCEDURE getTotalPriceByUserId(
```

```
  IN p_userId INT,
```

```
  OUT p_totalPrice FLOAT
```

```
)
```

```
BEGIN
```

```
  SELECT
```

```
    IFNULL(SUM(Total_Price), 0)
```

```
  INTO p_totalPrice
```

```
  FROM
```

```
    cart
```

```
  GROUP BY User_Id
```

```
  HAVING User_Id = p_userId;
```

```
END;
```

-- 5. Hiển thị những khách hàng chưa nhận được hàng

```
CREATE PROCEDURE displayCustomerNotReceived()
```

```
BEGIN
```

```
  SELECT u.User_Id, CONCAT(u.First_Name, ' ', u.Last_Name) AS
```

```
Customer_Name, o.Dest_Address, o.Status AS Status FROM user u
```

```
  JOIN cart c ON u.User_Id = c.User_Id
```

```
  JOIN orders o ON o.Customer_Id = c.User_Id
```

```
  WHERE NOT o.Status = 'Shipped';
```

```
END;
```

-- 6. Hiển thị số sách được bán ra của mỗi tác giả

```
CREATE PROCEDURE displaySellBooksOfAuthors()
```

```
BEGIN
```

```
  SELECT
```

```
    a.Author_Id,
```

```
    a.Name,
```

```
    IFNULL(
```

```
      (SELECT SUM(w.Number_Of_Books)
```

```
      FROM book b
```

```
      JOIN Wishlist w ON b.Book_Id = w.Book_Id
```

```
      WHERE b.Author_Id = a.Author_Id), 0
```

```
    ) AS Total_Books_Sell_For_Each_Author
```

```
  FROM
```

```
    Author a;
```

```
END;
```

XI. Kết luận

Nhóm đã thực hiện đủ quy trình xây dựng một hệ thống quản lý bán sách trực tuyến, cung cấp đầy đủ các thiết kế, chức năng cần thiết để vận hành hệ thống thông qua các yêu cầu cụ thể.