

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**VIỆN TRÍ TUỆ NHÂN TẠO**

-----\*\*\*-----



**BÁO CÁO MÔN HỌC CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI**

**HỆ THỐNG QUẢN LÝ BÁN SÁCH**

**Nhóm sinh viên thực hiện:**

1. Trần Quốc Khánh MSV 23020387
2. Hoàng Ngọc Nam MSV 23020403
3. Nguyễn Văn Huy MSV 23020379
4. Nguyễn Anh Kiệt MSV 23020383

**Giảng viên hướng dẫn: TS. Trần Hồng Việt**

**HÀ NỘI, 12/2024**

## MỞ ĐẦU

Trong thời đại số hóa, các cửa hàng và nền tảng bán sách đang đối mặt với nhu cầu quản lý số lượng lớn sách và đơn hàng một cách hiệu quả và chính xác. Việc quản lý thủ công không chỉ tốn kém thời gian mà còn dễ xảy ra sai sót. Để giải quyết vấn đề này, việc xây dựng một hệ thống quản lý bán sách tự động hóa là vô cùng cần thiết.

Hệ thống quản lý bán sách sẽ giúp tự động hóa các quy trình như:

- Nhập liệu thông tin sách, bao gồm các thông tin về tên sách, tác giả, thể loại, giá bán và số lượng tồn kho.
- Quản lý đơn hàng từ khách hàng, bao gồm việc theo dõi trạng thái đơn hàng, thanh toán và giao hàng.
- Thống kê doanh thu và phân tích số liệu bán hàng theo các tiêu chí như thời gian, thể loại sách và nhà xuất bản.
- Tìm kiếm thông tin sách nhanh chóng theo tên sách, tác giả, thể loại, ISBN hoặc các tiêu chí khác.
- Quản lý các chương trình khuyến mãi và giảm giá cho khách hàng.

Với mục tiêu xây dựng một hệ thống quản lý sách hiệu quả, chúng tôi đã thực hiện nghiên cứu và xây dựng báo cáo này. Báo cáo bao gồm các phần sau:

**Phần I: Phát biểu bài toán:**

**Phần II: Mô tả nghiệp vụ hệ thống**

**Phần III: Mô hình ER của hệ thống**

**Phần IV: Mô hình quan hệ**

**Phần V: Đặc tả yêu cầu dữ liệu**

**Phần VI: Danh sách các ràng buộc dữ liệu**

**Phần VII: Cài đặt vật lý**

**Phần VIII: Kết luận:**

## Mục Lục

MỞ ĐẦU.....	2
I: PHÁT BIỂU BÀI TOÁN.....	4
II. MÔ TẢ NGHIỆP VỤ HỆ THỐNG.....	4
III: Mô hình ER của hệ thống.....	5
IV: Mô hình quan hệ (chuyển từ mô hình ER).....	6
V: ĐẶC TẢ KIỂU DỮ LIỆU.....	7
VI: DANH SÁCH CÁC RÀNG BUỘC DỮ LIỆU.....	12
VII: CÀI ĐẶT VẬT LÝ.....	14
VIII: KẾT LUẬN.....	14

### I: PHÁT BIỂU BÀI TOÁN

Xây dựng cơ sở dữ liệu quản lý bán sách trực tuyến với mục tiêu hỗ trợ quản lý hiệu quả hoạt động kinh doanh và mang đến trải nghiệm tốt nhất cho khách hàng. Khách hàng có thể xem, tìm kiếm và đặt mua sách một cách dễ dàng, đồng thời có thể để lại những đánh giá, nhận xét về sách đã mua nhằm cung cấp thông tin hữu ích cho các khách hàng khác, đồng thời giúp cải thiện chất lượng sản phẩm và dịch vụ của hệ thống.

## II. MÔ TẢ NGHIỆP VỤ HỆ THỐNG

**Quản lý danh mục sách:** Lưu trữ thông tin về sách và một số thông tin liên quan như thể loại, tác giả, nhà xuất bản. Cho phép quản trị viên thêm mới, cập nhật hoặc xóa thông tin sách khi cần thiết..

**Quản lý thông tin khách hàng:** Hệ thống quản lý sẽ lưu trữ các thông tin về khách hàng và các thông tin khác như đơn hàng, danh sách yêu thích và đánh giá của khách hàng. Ngoài ra còn hỗ trợ chỉnh sửa thông tin khách hàng khi có sự thay đổi.

**Quản lý đơn hàng:** Khi có đơn hàng, quản trị viên có thể theo dõi và chuẩn bị đơn hàng để có thể giao đến khách hàng sớm nhất.

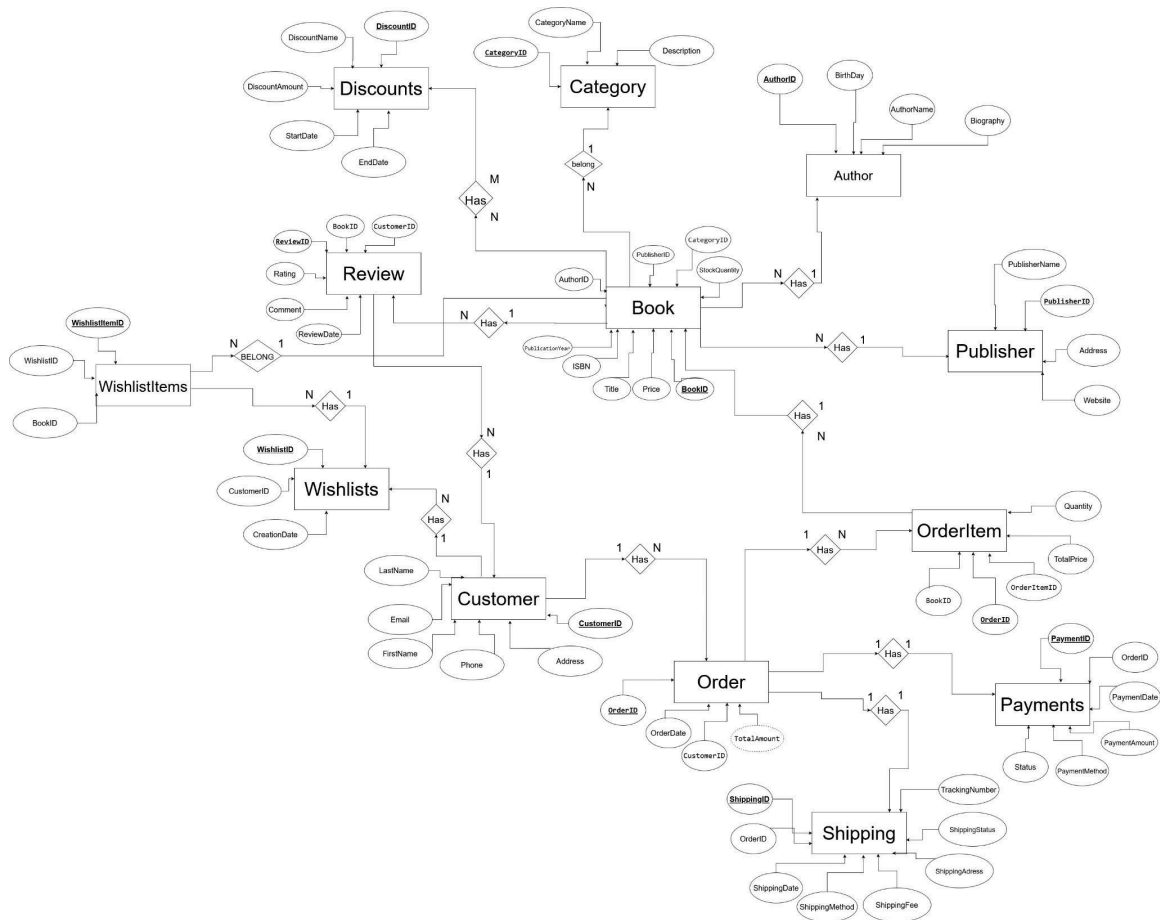
**Quản lý vận chuyển:** Khi có đơn hàng, thông tin về vận chuyển sẽ được cập nhật, bao gồm ngày giao hàng, trạng thái vận chuyển và mã vận chuyển để khách hàng có thể nắm bắt được thời gian nhận hàng.

**Quản lý đánh giá:** Hệ thống cho phép khách hàng đánh giá sách đã mua bằng cách ghi nhận các thông tin như điểm đánh giá (số sao), bình luận và ngày đánh giá. Quản trị viên có thể theo dõi, quản lý và xử lý các đánh giá không phù hợp để đảm bảo tính minh bạch và chất lượng của hệ thống.

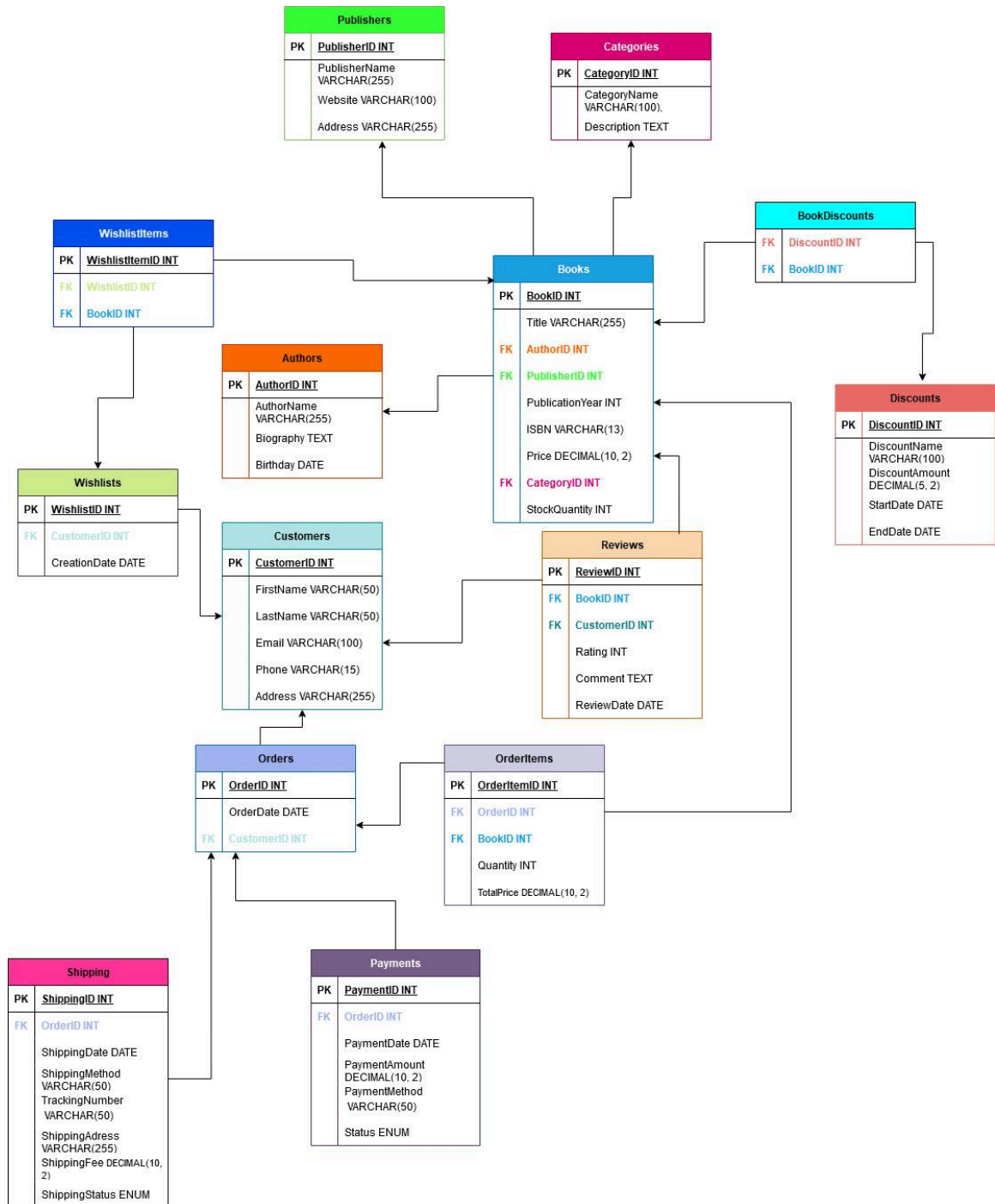
**Quản lý Thanh toán:** Theo dõi các khoản thanh toán cho các đơn hàng, bao gồm phương thức thanh toán và số tiền đã thanh toán, giúp thống kê doanh thu chính xác, giảm thiểu rủi ro gian lận và tạo báo cáo tài chính hàng tháng, từ đó đưa ra kế hoạch phù hợp.

**Quản lý Giảm giá:** Áp dụng các chương trình giảm giá cho các sách để giảm số lượng hàng tồn kho cũng như nâng cao trải nghiệm của khách hàng.

### III: Mô hình ER của hệ thống



## IV: Mô hình quan hệ (chuyển từ mô hình ER)



- **AUTHORS** ( AuthorID , AuthorName, Biography, Birthday)
- **BOOKS** ( BookID , Title, AuthorID, PublisherID, CategoryID, ISBN, PublicationYear, Price, StockQuantity)
- **PUBLISHERS** ( PublisherID , PublisherName, Website,Address)
- **CATEGORIES** ( CategoryID , CategoryName, Description)
- **CUSTOMERS** ( CustomerID , FirstName, LastName, Email, Phone,Address)
- **WISHLISTS** ( WishlistID , CustomerID, CreationDate)
- **WISHLISTITEMS** ( WishlistItemID , WishlistID, BookID)
- **ORDERS** ( OrderID , OrderDate, CustomerID)
- **ORDERITEMS** ( OrderItemID , OrderID, BookID, Quantity, TotalPrice)
- **PAYMENTS** ( PaymentID , OrderID, PaymentDate, PaymentAmount, PaymentMethod, Status)
- **REVIEWS** ( ReviewID , BookID, CustomerID, Rating, Comment, ReviewDate)
- **SHIPPING** ( ShippingID , OrderID, ShippingDate, ShippingAddress, ShippingMethod, TrackingNumber, ShippingFee, ShippingStatus)
- **DISCOUNTS** ( DiscountID , DiscountName, DiscountAmount, StartDate, EndDate)
- **BOOKDISCOUNTS** (DiscountID, BookID)

## V: ĐẶC TẢ KIỂU DỮ LIỆU

Data Element	Description	Data Type	Length	Example
AuthorID	Unique identifier for the author	INT	N/A	1, 2, 3
AuthorName	Name of the author	VARCHAR	255	Author A
Biography	Biography of the author	TEXT	N/A	Biography of Author A
Birthday	Author's date of birth	DATE	YYYY-MM-DD	1970-01-01
PublisherID	Unique identifier for the publisher	INT	N/A	1, 2, 3
PublisherName	Name of the publisher	VARCHAR	255	Publisher A
Website	Publisher's website	VARCHAR	100	<a href="http://publishera.com">http://publishera.com</a>
Address	Customer address	VARCHAR	255	123 Main St
CategoryID	Unique identifier for the category	INT	N/A	1, 2, 3
CategoryName	Name of the book category	VARCHAR	100	Fiction, Science
Description	Description of the category	TEXT	N/A	Books that contain fictional stories
BookID	Unique identifier for the book	INT	N/A	1, 2, 3
Title	Title of the book	VARCHAR	255	Book A
PublicationYear	Year the book was published	INT	N/A	2020, 2021, 2019
ISBN	International Standard Book Number	VARCHAR	13	1234567890123



Data Element	Description	Data Type	Length	Example
Price	Price of the book	DECIMAL	10,2	19.99, 29.99
StockQuantity	Number of books available in stock	INT	N/A	100, 50
CustomerID	Unique identifier for the customer	INT	N/A	1, 2, 3
FirstName	Customer's first name	VARCHAR	50	John
LastName	Customer's last name	VARCHAR	50	Doe
Email	Customer's email address	VARCHAR	100	john.doe@example.com
Phone	Customer's phone number	VARCHAR	15	123-456-7890
OrderID	Unique identifier for the order	INT	N/A	1, 2, 3
OrderDate	Date the order was placed	DATE	YYYY-MM-DD	2024-06-06
OrderItemID	Unique identifier for the order item	INT	N/A	1, 2, 3
Quantity	Quantity of books ordered	INT	N/A	1, 3, 5
TotalPrice	Total price for the order item	DECIMAL	10,2	39.98, 119.97
DiscountID	Unique identifier for the discount	INT	N/A	1, 2, 3
DiscountName	Name of the discount	VARCHAR	100	Holiday Special
DiscountAmount	Discount amount	DECIMAL	5,2	15.00, 10.00
StartDate	Start date of the discount	DATE	YYYY-MM-DD	2024-12-01

Data Element	Description	Data Type	Length	Example
EndDate	End date of the discount	DATE	YYYY-M M-DD	2024-12-31
ReviewID	Unique identifier for the review	INT	N/A	1, 2, 3
Rating	Rating given by the customer	INT	1–5	3, 4
Comment	Review comment	TEXT	N/A	A masterpiece of sci-fi!
ReviewDate	Date the review was submitted	DATE	YYYY-M M-DD	2024-03-12
PaymentID	Unique identifier for the payment	INT	N/A	1, 2, 5
PaymentDate	Date the payment was made	DATE	YYYY-M M-DD	2024-02-02
PaymentAmount	Total payment amount	DECIMAL	10,2	39.99
PaymentMethod	Method used for payment	VARCHAR	50	Credit Card, PayPal
Status	Payment status (Pending, Completed)	ENUM có DEFAULT 'Pending'	N/A	Pending
WishlistID	Unique identifier for the wishlist	INT	N/A	1, 2
CreationDate	Date when the wishlist was created	DATE	YYYY-M M-DD	2024-06-11
WishlistItemID	Unique identifier for the wishlist item	INT	N/A	1, 2, 3
ShippingID	Unique identifier for the shipping record	INT	N/A	1, 2, 3

Data Element	Description	Data Type	Length	Example
ShippingDate	Date when the order was shipped	DATE	YYYY-MM-DD	2024-10-11
ShippingMethod	Method of shipping (e.g., Standard, Express)	VARCHAR	50	Standard, Express
TrackingNumber	Tracking number for the shipment	VARCHAR	50	TRK010
ShippingAddress	Delivery address for the shipment	VARCHAR	255	707 Fir St
ShippingFee	Cost of shipping for the order	DECIMAL	(10,2)	10.00, 5.00
ShippingStatus	Shipping Status(Pending, Processing, Shipped, Delivered, Cancelled)	ENUM có DEFAULT 'Pending'	N/A	Pending, Shipped

## VI: DANH SÁCH CÁC RÀNG BUỘC DỮ LIỆU

- **PUBLISHERS và BOOKS:** Có mối quan hệ thông qua trường "PublisherID". Mỗi nhà xuất bản sẽ có một hoặc nhiều cuốn sách tương ứng.
- **AUTHORS và BOOKS:** Có mối quan hệ thông qua trường "AuthorID". Mỗi tác giả có thể viết một hoặc nhiều cuốn sách.
- **BOOKS và CATEGORIES:** Có mối quan hệ thông qua trường "CategoryID". Mỗi cuốn sách sẽ thuộc một thể loại duy nhất.
- **BOOKS và BOOKDISCOUNTS:** Có mối quan hệ thông qua trường "BookID". Mỗi cuốn sách có thể có một hoặc nhiều chương trình giảm giá.
- **DISCOUNTS và BOOKDISCOUNTS:** Có mối quan hệ thông qua trường "DiscountID". Mỗi chương trình giảm giá có thể áp dụng cho một hoặc nhiều cuốn sách.

- **CUSTOMERS và WISHLISTS**: Có mối quan hệ thông qua trường "CustomerID". Mỗi khách hàng có thể có một hoặc nhiều danh sách yêu thích sách.
- **WISHLISTS và WISHLISTITEMS**: Có mối quan hệ thông qua trường "WishlistID". Mỗi danh sách yêu thích có thể chứa nhiều mục (sách).
- **BOOKS và REVIEWS**: Có mối quan hệ thông qua trường "BookID". Mỗi cuốn sách có thể có một hoặc nhiều đánh giá từ khách hàng.
- **CUSTOMERS và REVIEWS**: Có mối quan hệ thông qua trường "CustomerID". Mỗi khách hàng có thể viết nhiều đánh giá cho các cuốn sách khác nhau.
- **CUSTOMERS và ORDERS**: Có mối quan hệ thông qua trường "CustomerID". Mỗi khách hàng có thể thực hiện nhiều đơn hàng.
- **ORDERS và ORDERITEMS**: Có mối quan hệ thông qua trường "OrderID". Mỗi đơn hàng có thể chứa nhiều mặt hàng (sách).
- **ORDERITEMS và BOOKS**: Có mối quan hệ thông qua trường "BookID". Mỗi mặt hàng trong đơn hàng sẽ là một cuốn sách.
- **ORDERS và PAYMENTS**: Có mối quan hệ thông qua trường "OrderID". Mỗi đơn hàng sẽ có ít nhất một khoản thanh toán tương ứng.
- **CUSTOMERS và PAYMENTS**: Có mối quan hệ thông qua trường "CustomerID". Mỗi khách hàng có thể thực hiện nhiều khoản thanh toán cho các đơn hàng của mình.
- **ORDERS và SHIPPING**: Có mối quan hệ thông qua trường "OrderID". Mỗi đơn hàng sẽ có một khoản vận chuyển tương ứng.

## VII: CÀI ĐẶT VẬT LÝ

1. Tạo table cùng các ràng buộc, insert dữ liệu. Cài đặt Index nếu có.
2. Các câu truy vấn cùng câu trả lời SQL tương ứng
3. Các trigger để cài đặt các ràng buộc phức tạp
4. Các thủ tục và hàm để giải quyết nghiệp vụ
- 5.

## 2. Một số câu truy vấn cùng câu trả lời SQL tương ứng:

### PHẦN A SỬ DỤNG INNER JOIN.

-- Lấy danh sách tất cả sách trong đơn hàng, bao gồm tên sách và số lượng mua.

```
SELECT Books.Title, OrderItems.Quantity
```

```
FROM OrderItems
```

```
JOIN Books ON OrderItems.BookID = Books.BookID;
```

	Title	Quantity
►	Book A	2
	Book B	1
	Book C	3
	Book D	1
	Book E	4
	Book F	2
	Book G	1
	Book H	2
	Book I	3
	Book J	5

-- Lấy thông tin về khách hàng và đơn hàng mà họ đã đặt.

```
SELECT Customers.FirstName, Customers.LastName, Orders.OrderDate,  
Orders.OrderID
```

```
FROM Orders
```

```
JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

	FirstName	LastName	OrderDate	OrderID
►	John	Doe	2024-01-01	1
	Jane	Smith	2024-02-02	2
	Alice	Johnson	2024-03-03	3
	Bob	Williams	2024-04-04	4
	Charlie	Brown	2024-05-05	5
	David	Jones	2024-06-06	6
	Eve	Miller	2024-07-07	7
	Frank	Davis	2024-08-08	8
	Grace	Garcia	2024-09-09	9
	Hannah	Martinez	2024-10-10	10

**PHẦN B SỬ DỤNG OUTER JOIN.**

```
-- Lấy danh sách tất cả các sách và thông tin giảm giá nếu có.  
SELECT Books.Title, Discounts.DiscountName, Discounts.DiscountAmount  
FROM Books  
LEFT JOIN BookDiscounts ON Books.BookID = BookDiscounts.BookID  
LEFT JOIN Discounts ON BookDiscounts.DiscountID = Discounts.DiscountID;
```

	Title	DiscountName	DiscountAmount
►	Book A	Winter Sale	20.00
	Book B	Summer Offer	15.00
	Book C	Black Friday	50.00
	Book D	Holiday Special	30.00
	Book E	Back to School	10.00
	Book F	Flash Sale	25.00
	Book G	Christmas Discount	35.00
	Book H	New Year Sale	40.00
	Book I	Easter Deal	18.00
	Book J	Spring Discount	12.00

```
-- Lấy danh sách khách hàng và thông tin wishlist của họ, kể cả khi không có wishlist.  
SELECT Customers.FirstName, Customers.LastName, Wishlists.WishlistID,  
Wishlists.CreationDate  
FROM Customers  
LEFT JOIN Wishlists ON Customers.CustomerID = Wishlists.CustomerID;
```

	FirstName	LastName	WishlistID	CreationDate
►	John	Doe	1	2024-01-06
	Jane	Smith	2	2024-02-07
	Alice	Johnson	3	2024-03-08
	Bob	Williams	4	2024-04-09
	Charlie	Brown	5	2024-05-10
	David	Jones	6	2024-06-11
	Eve	Miller	7	2024-07-12
	Frank	Davis	8	2024-08-13
	Grace	Garcia	9	2024-09-14
	Hannah	Martinez	10	2024-10-15

## PHẦN C SỬ DỤNG SUBQUERY TRONG WHERE.

-- Lấy danh sách tên các khách hàng đã đặt đơn hàng trong năm 2024.

```
SELECT concat(LastName,',',FirstName) AS name
FROM Customers
WHERE CustomerID IN (
    SELECT CustomerID
    FROM Orders
    WHERE YEAR(OrderDate) = 2024
);
```

	name
▶	Doe John
	Smith Jane
	Johnson Alice
	Williams Bob
	Brown Charlie
	Jones David
	Miller Eve
	Davis Frank
	Garcia Grace
	Martinez Hannah

-- Tìm các sách có giảm giá cao hơn mức giảm giá trung bình.

```
SELECT Books.Title, Discounts.DiscountAmount
FROM Books
JOIN BookDiscounts ON Books.BookID = BookDiscounts.BookID
JOIN Discounts ON BookDiscounts.DiscountID = Discounts.DiscountID
WHERE Discounts.DiscountAmount > (
    SELECT AVG(DiscountAmount) FROM Discounts
);
```

	Title	DiscountAmount
▶	Book C	50.00
	Book D	30.00
	Book G	35.00
	Book H	40.00

## PHẦN D SỬ DỤNG SUBQUERY TRONG FROM.

-- Tìm tổng số lượng sách được bán ra theo từng danh mục sách.

```
SELECT Categories.CategoryName, SUM(Subquery.Quantity) AS TotalQuantity
FROM (
    SELECT Books.CategoryID, OrderItems.Quantity
    FROM OrderItems
    JOIN Books ON OrderItems.BookID = Books.BookID
) AS Subquery
JOIN Categories ON Subquery.CategoryID = Categories.CategoryID
GROUP BY Categories.CategoryName;
```

	CategoryName	TotalQuantity
▶	Fiction	2
	Non-Fiction	1
	Science	3
	Technology	1
	Fantasy	4
	Romance	2
	History	1
	Biography	2
	Self-Help	3
	Mystery	5

## PHẦN E SỬ DỤNG GROUP BY VÀ CÁC HÀM TỔNG HỢP.

-- Tìm tổng doanh thu và số lượng sách bán được theo từng tháng.

```
SELECT MONTH(Orders.OrderDate) AS Month, SUM(OrderItems.TotalPrice)
AS TotalRevenue, SUM(OrderItems.Quantity) AS TotalBooksSold
FROM Orders
JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID
GROUP BY MONTH(Orders.OrderDate);
```



	Month	TotalRevenue	TotalBooksSold
▶	1	39.98	2
	2	29.99	1
	3	119.97	3
	4	24.99	1
	5	59.96	4
	6	69.98	2
	7	44.99	1
	8	109.98	2
	9	194.97	3
	10	374.95	5

-- Tính số lượng đánh giá và điểm đánh giá trung bình cho mỗi cuốn sách.

```
SELECT Books.Title, COUNT(Reviews.ReviewID) AS TotalReviews,
AVG(Reviews.Rating) AS AverageRating
FROM Books
LEFT JOIN Reviews ON Books.BookID = Reviews.BookID
GROUP BY Books.Title;
```

	Title	TotalReviews	AverageRating
▶	Book A	1	5.0000
	Book B	1	4.0000
	Book C	1	5.0000
	Book D	1	3.0000
	Book E	1	4.0000
	Book F	1	5.0000
	Book G	1	4.0000
	Book H	1	5.0000
	Book I	1	4.0000
	Book J	1	5.0000

### 3. Sử dụng trigger để cập nhật số lượng hàng trong kho khi có đơn hàng mới:

DELIMITER \$\$

CREATE TRIGGER ReduceStockAfterPurchase

AFTER INSERT ON OrderItems

FOR EACH ROW

BEGIN

UPDATE Books

SET StockQuantity = StockQuantity - NEW.Quantity

WHERE BookID = NEW.BookID;

-- Kiểm tra nếu số lượng tồn kho âm

IF (SELECT StockQuantity FROM Books WHERE BookID = NEW.BookID) < 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Stock quantity cannot be negative. Check book availability.';

END IF;

END;

\$\$

DELIMITER ;

Đầu tiên sẽ thử insert một OrderItems có số lượng lớn hơn so với số lượng trong kho hàng:

```
INSERT INTO OrderItems (OrderItemID, OrderID, BookID, Quantity, TotalPrice)
VALUES
(1, 1, 1, 200, 39.98);
```

Nó sẽ in ra lỗi:

Error Code: 1644. The order quantity exceeds available stock. Please check the stock.

Error Code: 1644. The order quantity exceeds available stock. Please check the stock.

Khi insert một OrderItems có số lượng nhỏ hơn hoặc bằng so với số lượng trong kho hàng:

```
INSERT INTO OrderItems (OrderItemID, OrderID, BookID, Quantity, TotalPrice)
VALUES
(1, 1, 1, 2, 39.98),
```

	BookID	Title	AuthorID	PublisherID	PublicationYear	ISBN	Price	CategoryID	StockQuantity
►	1	Book A	1	1	2020	1234567890123	19.99	1	98

Kết quả trả lại là 98 do ban đầu có 100 sản phẩm trong kho bớt đi 2 sản phẩm để giao hàng.

## 4. ProcEDURE:

### 4.1 Tính tổng doanh thu trong một tháng theo năm.

DELIMITER \$\$

```
CREATE PROCEDURE CalculateMonthlyRevenue(IN p_Year INT, IN p_Month
INT)
```

```
BEGIN
```

```
    DECLARE totalRevenue DECIMAL(10, 2);
```

```
-- Tính tổng doanh thu của tháng từ Payments
```

```

SELECT IFNULL(SUM(Payments.PaymentAmount),0)
INTO totalRevenue
FROM Payments
JOIN Orders ON Payments.OrderID = Orders.OrderID
WHERE YEAR(Payments.PaymentDate) = p_Year
AND MONTH(Payments.PaymentDate) = p_Month;

```

-- Hiển thị doanh thu

```

SELECT totalRevenue AS MonthlyRevenue;
IF totalRevenue = 0 THEN
    SELECT 'Không có doanh thu trong tháng này.' AS Message;
ELSE
    SELECT CONCAT('Doanh thu tháng ', p_Month, ' năm ', p_Year, ' là: ',
totalRevenue) AS Message;
END IF;
END$$

```

DELIMITER ;

Gọi hàm để tính tổng doanh thu tháng 9 năm 2024.

```
CALL CalculateMonthlyRevenue(2024, 9);
```

	Message
▶	Doanh thu tháng 9 năm 2024 là: 199.97

```
CALL CalculateMonthlyRevenue(2024, 11);
```

	Message
▶	Không có doanh thu trong tháng này.

## 4.2 Xử lý thanh toán cho đơn hàng và cập nhật trạng thái( Có sử dụng ROLLBACK)

DELIMITER \$\$

```
CREATE PROCEDURE ProcessPayments(IN p_OrderID INT, IN  
p_PaymentAmount DECIMAL(10,2))
```

```
BEGIN
```

```
    DECLARE totalAmount DECIMAL(10,2);
```

```
    DECLARE currentStatus ENUM('Pending', 'Completed');
```

```
    DECLARE exit handler for sqlexception
```

```
BEGIN
```

```
    -- Xử lý lỗi: Rollback giao dịch khi có lỗi
```

```
    ROLLBACK;
```

```
    SELECT 'Có lỗi trong quá trình thanh toán. Thay đổi đã bị hoàn tác.' AS  
Message;
```

```
END;
```

```
    -- Bắt đầu giao dịch
```

```
    START TRANSACTION;
```

```
    -- Kiểm tra trạng thái của thanh toán
```

```
    SELECT Status INTO currentStatus
```

```
    FROM Payments
```

```
    WHERE OrderID = p_OrderID;
```

```
IF currentStatus = 'Completed' THEN
```

```
    SIGNAL SQLSTATE '45000';
```

```
END IF;
```

```
-- Kiểm tra số tiền thanh toán có hợp lệ không
```

```
IF p_PaymentAmount <= 0 THEN
```

```
    -- Nếu số tiền thanh toán không hợp lệ, kích hoạt lỗi và thực hiện
```

```
ROLLBACK
```

```
    SIGNAL SQLSTATE '45000';
```

```
END IF;
```

```
-- Lấy tổng số tiền của đơn hàng từ bảng Payments
```

```
SELECT PaymentAmount INTO totalAmount
```

```
FROM Payments
```

```
WHERE OrderID = p_OrderID;
```

```
-- Kiểm tra nếu số tiền thanh toán không khớp với tổng số tiền của đơn hàng
```

```
IF ROUND(totalAmount, 2) != ROUND(p_PaymentAmount, 2) THEN
```

```
    -- Nếu số tiền thanh toán không khớp, rollback và hiển thị lỗi
```

```
    SIGNAL SQLSTATE '45000';
```

```
END IF;
```

```
-- Nếu không có lỗi, cập nhật trạng thái thanh toán thành 'Completed'
```

```
UPDATE Payments
```

```
SET status = 'Completed', PaymentAmount = p_PaymentAmount, PaymentDate  
= NOW()
```

```
WHERE OrderID = p_OrderID;
```

```
-- Nếu không có lỗi, commit giao dịch
```

```
COMMIT;
```

```
SELECT 'Thanh toán thành công.' AS Message;
```

```
END$$
```

```
DELIMITER ;
```

```
-- Gọi Procedure với tham số
```

```
CALL ProcessPayments(1, 44.98);
```

	Message
▶	Thanh toán thành công.

	PaymentID	OrderID	PaymentDate	PaymentAmount	PaymentMethod	Status
▶	1	1	2024-12-15	44.98	Credit Card	Completed

Hiện ra thanh toán thành công và cập nhật trạng thái của hoá đơn.

**Tiếp tục gọi**

```
CALL ProcessPayments(1, 44.98);
```

Message:	Có lỗi trong quá trình thanh toán. Thay đổi đã bị hoàn tác.
----------	---

Lỗi do hoá đơn đã được thanh toán rồi.

## VIII: KẾT LUẬN

Sau quá trình nghiên cứu và triển khai, hệ thống quản lý bán sách được xây dựng đã đạt được các mục tiêu đề ra, thể hiện qua các phần cụ thể như sau:

### 1. Phân tích bài toán và mô hình hóa dữ liệu

Chúng tôi đã xác định rõ các nghiệp vụ cốt lõi của hệ thống bán sách, bao gồm quản lý danh mục sách, khách hàng, đơn hàng, vận chuyển, và thanh toán. Mô hình ER và mô hình quan hệ được thiết kế đảm bảo sự chặt chẽ về logic, đầy đủ để phục vụ các yêu cầu thực tiễn.

Kết quả: Hệ thống cơ sở dữ liệu không chỉ hỗ trợ lưu trữ dữ liệu một cách hiệu quả mà còn cung cấp khả năng truy xuất thông tin nhanh chóng và chính xác.

### 2. Xây dựng cơ sở dữ liệu vật lý và triển khai các tính năng nghiệp vụ

Các bảng dữ liệu, ràng buộc, trigger, và stored procedures được triển khai một cách hợp lý để đảm bảo tính toàn vẹn dữ liệu và hỗ trợ các quy trình nghiệp vụ phức tạp như quản lý giảm giá, cập nhật số lượng sách tồn kho, và xử lý thanh toán.

Kết quả: Các trigger đã tự động hóa được việc kiểm tra số lượng sách trong kho khi có đơn hàng mới, giảm thiểu sai sót trong quản lý tồn kho. Stored procedures hỗ trợ xử lý các báo cáo doanh thu, đồng thời đảm bảo giao dịch thanh toán được thực hiện an toàn, chính xác.

### 3. Thực hiện các truy vấn SQL

Hệ thống đã triển khai các truy vấn SQL với nhiều kỹ thuật như INNER JOIN, OUTER JOIN, SUBQUERY, GROUP BY để giải quyết các yêu cầu phức tạp như thống kê doanh thu, số lượng sách bán theo tháng, hoặc đánh giá chất lượng sách qua phản hồi khách hàng.



Kết quả: Các truy vấn cung cấp thông tin chi tiết và chính xác

#### 4. **Đánh giá tổng thể**

- **Ưu điểm:** Hệ thống đáp ứng đầy đủ các yêu cầu cơ bản của nghiệp vụ bán sách, giúp tự động hóa quy trình, giảm thiểu rủi ro sai sót, đồng thời tăng hiệu quả trong quản lý thông tin. Các chức năng như quản lý giảm giá và đánh giá sách góp phần nâng cao trải nghiệm của khách hàng.
- **Hạn chế:** Hệ thống hiện tại tập trung vào quản lý dữ liệu và tự động hóa nội bộ, và quản lý chi tiết như lịch sử mua hàng hay báo cáo doanh thu theo khu vực, hạn chế khả năng đáp ứng nhu cầu quản lý toàn diện và nâng cao trải nghiệm khách hàng.

#### **Giải thích kết quả thu được**

Hệ thống đã chứng minh khả năng xử lý và quản lý thông tin hiệu quả qua các thử nghiệm:

- Khi nhập các đơn hàng với số lượng sách vượt quá tồn kho, hệ thống tự động phát hiện và ngăn chặn, đảm bảo tính toàn vẹn dữ liệu.
- Việc tính toán doanh thu theo tháng và năm được thực hiện chính xác, tạo điều kiện thuận lợi cho việc lập kế hoạch kinh doanh.
- Chức năng quản lý giảm giá đã thành công trong việc kết nối các bảng liên quan, cho phép áp dụng nhiều chương trình giảm giá linh hoạt.