Online Bookstore Management System

A full-stack web application for managing an online bookstore with customer and admin functionalities.

Features

Customer Features

- **User Authentication**: Sign up, login, and profile management
- **Book Browsing**: Search books by title, author, ISBN, publisher, or category
- **Shopping Cart**: Add books to cart, update quantities, remove items
- **Checkout**: Place orders with credit card validation
- **Order History**: View past orders with details

Admin Features

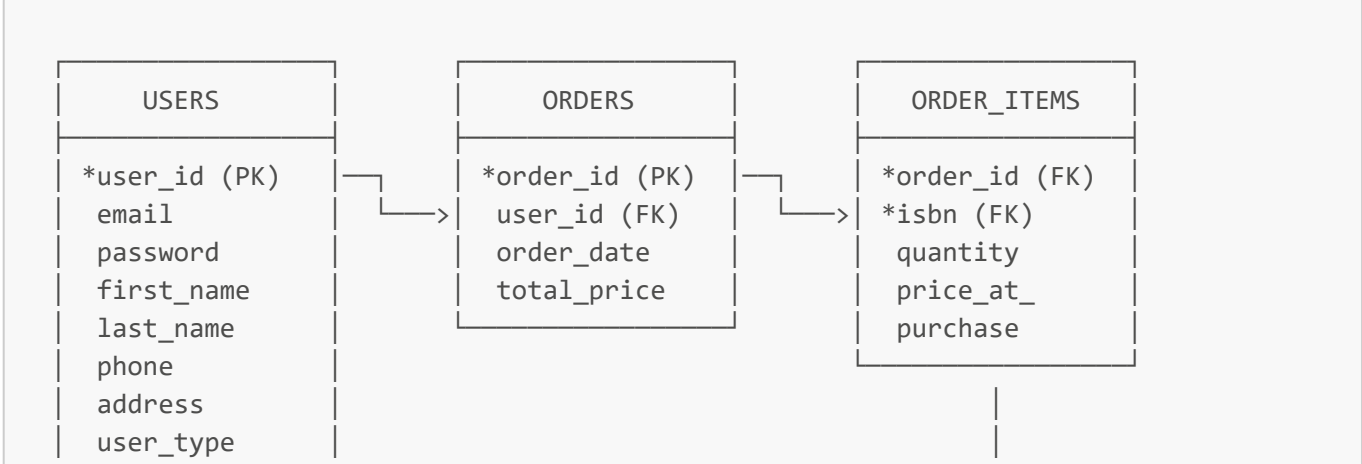
- **Book Management**: Add new books, edit existing books, manage inventory
- **Publisher Orders**: Create manual orders from publishers, confirm pending orders
- **System Reports**:
  - Total sales for the previous month
  - Total sales for a specific date
  - Top 5 customers (last 3 months)
  - Top 10 selling books (last 3 months)
  - Book replenishment order count

System Features

- **Automatic Stock Management**: Triggers handle stock updates
- **Low Stock Alerts**: Automatic publisher orders when stock falls below threshold
- **Real-time Search**: Instant filtering across all book attributes
- **Responsive Design**: Works on desktop and mobile devices

Database Schema

Entity-Relationship Diagram



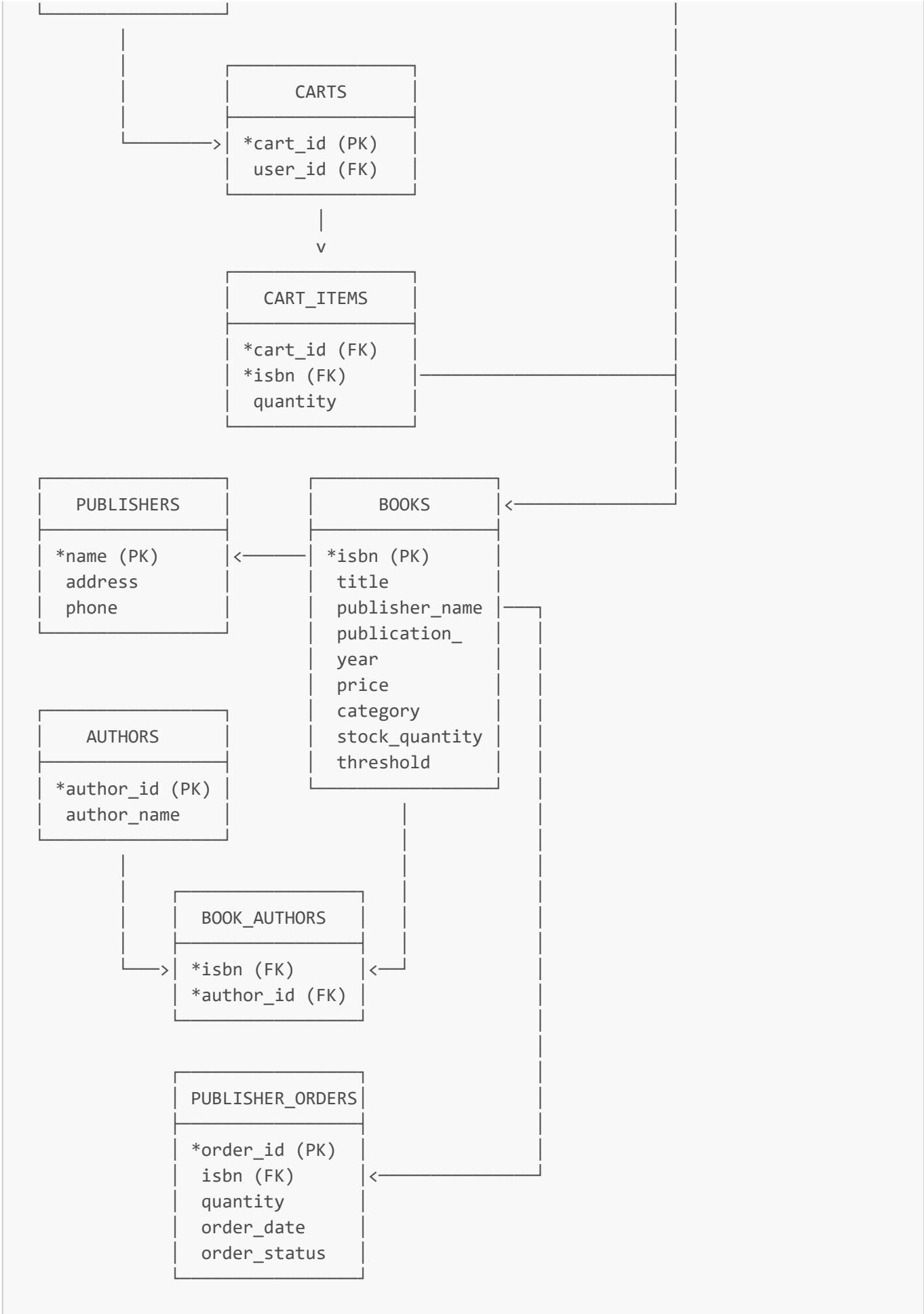


Table Definitions

1. USERS

Stores customer and admin account information.

Column	Type	Constraints	Description
user_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique user identifier
email	VARCHAR(255)	UNIQUE, NOT NULL	User's email address
password	VARCHAR(255)	NOT NULL	Hashed password (bcrypt)
first_name	VARCHAR(100)	NOT NULL	User's first name
last_name	VARCHAR(100)	NOT NULL	User's last name
phone	VARCHAR(20)	NOT NULL	Contact phone number
address	TEXT	NOT NULL	Shipping address
user_type	ENUM('CUSTOMER', 'ADMIN')	DEFAULT 'CUSTOMER'	User role

2. BOOKS

Stores book inventory information.

Column	Type	Constraints	Description
isbn	VARCHAR(20)	PRIMARY KEY	International Standard Book Number
title	VARCHAR(255)	NOT NULL	Book title
publisher_name	VARCHAR(255)	FOREIGN KEY	Reference to publisher
publication_year	INT	NOT NULL	Year of publication
price	DECIMAL(10,2)	NOT NULL	Book price
category	ENUM	NOT NULL	Science, Art, Religion, History, Geography
stock_quantity	INT	DEFAULT 0	Current stock level
threshold	INT	NOT NULL	Minimum stock before reorder

3. AUTHORS

Stores author information.

Column	Type	Constraints	Description
author_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique author identifier
author_name	VARCHAR(255)	NOT NULL	Author's full name

4. BOOK_AUTHORS (Junction Table)

Links books to their authors (many-to-many relationship).

Column	Type	Constraints	Description
isbn	VARCHAR(20)	PRIMARY KEY, FOREIGN KEY	Reference to book
author_id	INT	PRIMARY KEY, FOREIGN KEY	Reference to author

5. PUBLISHERS

Stores publisher information.

Column	Type	Constraints	Description
name	VARCHAR(255)	PRIMARY KEY	Publisher name
address	TEXT		Publisher address
phone	VARCHAR(20)		Publisher phone

6. CARTS

Stores shopping cart for each user.

Column	Type	Constraints	Description
cart_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique cart identifier
user_id	INT	FOREIGN KEY	Reference to user

7. CART_ITEMS

Stores items in shopping carts.

Column	Type	Constraints	Description
cart_id	INT	PRIMARY KEY, FOREIGN KEY	Reference to cart
isbn	VARCHAR(20)	PRIMARY KEY, FOREIGN KEY	Reference to book
quantity	INT	NOT NULL	Number of copies

8. ORDERS

Stores customer orders.

Column	Type	Constraints	Description
order_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique order identifier
user_id	INT	FOREIGN KEY	Reference to user

Column	Type	Constraints	Description
order_date	DATETIME	NOT NULL	When order was placed
total_price	DECIMAL(10,2)	NOT NULL	Total order amount

9. ORDER_ITEMS

Stores items in each order.

Column	Type	Constraints	Description
order_id	INT	PRIMARY KEY, FOREIGN KEY	Reference to order
isbn	VARCHAR(20)	PRIMARY KEY, FOREIGN KEY	Reference to book
quantity	INT	NOT NULL	Number of copies ordered
price_at_purchase	DECIMAL(10,2)	NOT NULL	Price when purchased

10. PUBLISHER_ORDERS

Stores orders placed to publishers for restocking.

Column	Type	Constraints	Description
order_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique order identifier
isbn	VARCHAR(20)	FOREIGN KEY	Reference to book
quantity	INT	NOT NULL	Number of copies to order
order_date	DATETIME	NOT NULL	When order was placed
order_status	ENUM('Pending', 'Confirmed')	DEFAULT 'Pending'	Order status

---

SQL Queries Reference

Authentication Queries

Check if Email Already Exists

```
SELECT user_id FROM users WHERE email = ?
```

Purpose: Validates during signup that the email is not already registered in the system.

Register New User

```
INSERT INTO users (email, password, first_name, last_name, phone, address,
user_type)
VALUES (?, ?, ?, ?, ?, ?, 'CUSTOMER')
```

Purpose: Creates a new customer account with hashed password. All new registrations default to 'CUSTOMER' role.

User Login Authentication

```
SELECT user_id, email, password, user_type FROM users WHERE email = ?
```

Purpose: Retrieves user credentials for authentication. The password hash is compared with bcrypt.

Get User Profile

```
SELECT user_id, first_name, last_name, email, phone, address, user_type
FROM users WHERE user_id = ?
```

Purpose: Fetches the complete profile information for display in the user's profile page.

Update User Profile

```
UPDATE users SET first_name = ?, last_name = ?, email = ?, phone = ?, address = ?,
password = ?
WHERE user_id = ?
```

Purpose: Updates user profile information. Only provided fields are updated (dynamic query building).

Book Management Queries

Check if ISBN Already Exists

```
SELECT isbn FROM books WHERE isbn = ?
```

Purpose: Prevents duplicate book entries by checking if ISBN is already in the database.

Verify Publisher Exists

```
SELECT name FROM publishers WHERE name = ?
```

Purpose: Ensures the publisher exists before adding a new book (referential integrity).

Add New Book

```
INSERT INTO books (isbn, title, publisher_name, publication_year, price, category,
stock_quantity, threshold)
VALUES (?, ?, ?, ?, ?, ?, ?, ?)
```

Purpose: Inserts a new book into the inventory with all required details including stock threshold for automatic reordering.

Find Author by Name

```
SELECT author_id FROM authors WHERE author_name = ?
```

Purpose: Checks if an author already exists to avoid duplicate entries.

Create New Author

```
INSERT INTO authors (author_name) VALUES (?)
```

Purpose: Adds a new author to the database if they don't already exist.

Link Book to Author

```
INSERT INTO book_authors (isbn, author_id) VALUES (?, ?)
```

Purpose: Creates the many-to-many relationship between books and authors in the junction table.

Get Book Details

```
SELECT * FROM books WHERE isbn = ?
```

Purpose: Retrieves a specific book's information for editing or validation.

Update Book Information

```
UPDATE books SET title = ?, publisher_name = ?, publication_year = ?, price = ?,
category = ?, stock_quantity = ?
WHERE isbn = ?
```

Purpose: Modifies book details. This is a dynamic query - only changed fields are updated.

Remove Book-Author Links

```
DELETE FROM book_authors WHERE isbn = ?
```

Purpose: Clears existing author associations before updating with new authors.

Search Books with Filters

```
SELECT DISTINCT b.isbn, b.title, b.publication_year, b.price, b.category,
       b.stock_quantity, b.publisher_name,
       GROUP_CONCAT(DISTINCT a.author_name SEPARATOR ', ') as authors
FROM books b
LEFT JOIN book_authors ba ON b.isbn = ba.isbn
LEFT JOIN authors a ON ba.author_id = a.author_id
WHERE 1=1
  AND b.isbn = ?
  AND b.title LIKE ?
  AND b.category = ?
  AND a.author_name LIKE ?
  AND b.publisher_name LIKE ?
GROUP BY b.isbn, b.title, b.publication_year, b.price, b.category,
b.stock_quantity, b.publisher_name
```

Purpose: Flexible search query that filters books by multiple criteria. Uses `WHERE 1=1` to allow dynamic filter addition. `GROUP_CONCAT` combines multiple authors into a comma-separated string.

Get Single Book with Full Details

```
SELECT b.*, p.address, p.phone,
       GROUP_CONCAT(DISTINCT a.author_name SEPARATOR ', ') as authors
FROM books b
LEFT JOIN publishers p ON b.publisher_name = p.name
LEFT JOIN book_authors ba ON b.isbn = ba.isbn
LEFT JOIN authors a ON ba.author_id = a.author_id
WHERE b.isbn = ?
GROUP BY b.isbn
```

Purpose: Retrieves complete book information including publisher details and all authors for the book details page.

Get All Books (Admin)

```
SELECT b.*, p.address, p.phone,
       GROUP_CONCAT(DISTINCT a.author_name SEPARATOR ', ') as authors
FROM books b
LEFT JOIN publishers p ON b.publisher_name = p.name
LEFT JOIN book_authors ba ON b.isbn = ba.isbn
LEFT JOIN authors a ON ba.author_id = a.author_id
GROUP BY b.isbn
ORDER BY b.title
```

Purpose: Lists all books with their details for the admin management interface, sorted alphabetically by title.

Shopping Cart Queries

Get User's Cart

```
SELECT cart_id FROM carts WHERE user_id = ?
```

Purpose: Finds the existing cart for a user or returns empty if no cart exists.

Create New Cart

```
INSERT INTO carts (user_id) VALUES (?)
```

Purpose: Creates a new shopping cart for a user when they first add an item.

Check Book Stock for Cart

```
SELECT isbn, title, price, stock_quantity FROM books WHERE isbn = ?
```

Purpose: Validates book availability and retrieves pricing before adding to cart.

Check if Item Already in Cart

```
SELECT quantity FROM cart_items WHERE cart_id = ? AND isbn = ?
```

Purpose: Determines if the book is already in the cart to update quantity instead of adding duplicate entry.

Update Cart Item Quantity

```
UPDATE cart_items SET quantity = ? WHERE cart_id = ? AND isbn = ?
```

Purpose: Increases or modifies the quantity of an item already in the cart.

Add New Item to Cart

```
INSERT INTO cart_items (cart_id, isbn, quantity) VALUES (?, ?, ?)
```

Purpose: Adds a new book to the user's shopping cart.

View Cart Contents

```
SELECT ci.isbn, b.title, b.price, ci.quantity,
       (b.price * ci.quantity) as item_total,
       GROUP_CONCAT(DISTINCT a.author_name SEPARATOR ', ') as authors
FROM cart_items ci
JOIN books b ON ci.isbn = b.isbn
LEFT JOIN book_authors ba ON b.isbn = ba.isbn
LEFT JOIN authors a ON ba.author_id = a.author_id
WHERE ci.cart_id = ?
GROUP BY ci.isbn, b.title, b.price, ci.quantity
```

Purpose: Retrieves all cart items with book details, calculates line item totals, and includes author names for display.

Remove Item from Cart

```
DELETE FROM cart_items WHERE cart_id = ? AND isbn = ?
```

Purpose: Removes a specific book from the user's cart.

Verify Stock for Cart Update

```
SELECT ci.isbn, b.stock_quantity
FROM cart_items ci
JOIN books b ON ci.isbn = b.isbn
WHERE ci.cart_id = ? AND ci.isbn = ?
```

Purpose: Validates sufficient stock exists before allowing quantity increase in cart.

Order Processing Queries

Get Cart Items for Checkout

```
SELECT ci.isbn, ci.quantity, b.price, b.stock_quantity, b.title
FROM cart_items ci
JOIN books b ON ci.isbn = b.isbn
WHERE ci.cart_id = ?
```

Purpose: Retrieves all cart items with current prices and stock levels for order processing.

Create New Order

```
INSERT INTO orders (user_id, order_date, total_price)
VALUES (?, NOW(), ?)
```

Purpose: Creates the order header with the calculated total. `NOW()` captures the exact timestamp.

Create Order Line Items

```
INSERT INTO order_items (order_id, isbn, quantity, price_at_purchase)
VALUES (?, ?, ?, ?)
```

Purpose: Records each book in the order with the price at time of purchase (price history preservation).

Clear Cart After Checkout

```
DELETE FROM cart_items WHERE cart_id = ?
```

Purpose: Removes all items from the cart after successful order placement.

Delete Empty Cart

```
DELETE FROM carts WHERE cart_id = ?
```

Purpose: Removes the cart record after checkout to maintain clean data.

Get Customer Order History

```
SELECT o.order_id, o.order_date, o.total_price
FROM orders o
WHERE o.user_id = ?
ORDER BY o.order_date DESC
```

Purpose: Lists all orders for a customer, most recent first.

Get Order Line Items

```
SELECT oi.isbn, b.title, oi.quantity, oi.price_at_purchase,
       (oi.quantity * oi.price_at_purchase) as item_total,
       GROUP_CONCAT(DISTINCT a.author_name SEPARATOR ', ') as authors
FROM order_items oi
JOIN books b ON oi.isbn = b.isbn
LEFT JOIN book_authors ba ON b.isbn = ba.isbn
LEFT JOIN authors a ON ba.author_id = a.author_id
WHERE oi.order_id = ?
GROUP BY oi.isbn, b.title, oi.quantity, oi.price_at_purchase
```

Purpose: Retrieves detailed information for each item in an order, showing the historical purchase price.

Publisher Order Queries

Get Publisher Order Details

```
SELECT * FROM publisher_orders WHERE order_id = ?
```

Purpose: Retrieves a specific publisher order for status checking before confirmation.

Confirm Publisher Order

```
UPDATE publisher_orders SET order_status = 'Confirmed' WHERE order_id = ?
```

Purpose: Updates order status to confirmed. A trigger adds the ordered quantity to book stock.

Get All Publisher Orders (Admin)

```
SELECT po.*, b.title
FROM publisher_orders po
JOIN books b ON po.isbn = b.isbn
ORDER BY po.order_date DESC
```

Purpose: Lists all publisher orders with book titles for the admin interface, most recent first.

Create Manual Publisher Order

```
INSERT INTO publisher_orders (isbn, quantity, order_date, order_status)
VALUES (?, ?, NOW(), 'Pending')
```

Purpose: Allows admin to manually create a restock order outside of the automatic threshold system.

Get Books for Order Dropdown

```
SELECT isbn, title, stock_quantity, threshold
FROM books
ORDER BY title
```

Purpose: Provides book list with current stock levels for the admin order creation form.

Report Queries

Total Sales Last Month

```
SELECT SUM(total_price) as total_sales
FROM orders
WHERE order_date >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH)
AND order_date < CURDATE()
```

Purpose: Calculates total revenue from the previous month for the monthly sales report.

Total Sales by Specific Date

```
SELECT SUM(total_price) as total_sales
FROM orders
WHERE DATE(order_date) = ?
```

Purpose: Returns total sales for a specific date selected by the admin.

Top 5 Customers (Last 3 Months)

```
SELECT u.user_id, u.email, u.first_name, u.last_name,
       SUM(o.total_price) as total_purchases
FROM users u
JOIN orders o ON u.user_id = o.user_id
WHERE o.order_date >= DATE_SUB(CURDATE(), INTERVAL 3 MONTH)
GROUP BY u.user_id, u.email, u.first_name, u.last_name
ORDER BY total_purchases DESC
LIMIT 5
```

Purpose: Identifies the highest-spending customers over the last 3 months for VIP recognition.

Top 10 Selling Books (Last 3 Months)

```
SELECT b.isbn, b.title,
       SUM(oi.quantity) as total_sold
FROM books b
JOIN order_items oi ON b.isbn = oi.isbn
JOIN orders o ON oi.order_id = o.order_id
WHERE o.order_date >= DATE_SUB(CURDATE(), INTERVAL 3 MONTH)
GROUP BY b.isbn, b.title
```

```
ORDER BY total_sold DESC
LIMIT 10
```

Purpose: Shows best-selling books by units sold for inventory and marketing decisions.

Book Replenishment Order Count

```
SELECT COUNT(*) as order_count
FROM publisher_orders
WHERE isbn = ?
```

Purpose: Shows how many times a specific book has been reordered from publishers.

Get Book Title for Report

```
SELECT title FROM books WHERE isbn = ?
```

Purpose: Retrieves book title for display in the order count report.