

Database Management System

Semester-III (Batch-2024)

Online Bookstore Management System

G11-DB-PID-1



Supervised By:

Mr. Sachin Garg

Submitted By:

Anshika Sharma, 2410990847 (G-11)

Ashu, 2410990872 (G-11)

Sukhpreet Singh, 2410990885 (G-11)

**Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab**

Online Bookstore Management System

1. Introduction

The Online Book Management System is a digital platform that helps manage books, users, and transactions in an organized and reliable way. It brings together all book details, user information, and activities such as issuing, returning, and reserving books into one secure system that can be easily accessed anytime.

At the heart of this system is the Database Management System (DBMS), which plays a key role in storing, organizing, and protecting data. It ensures that every record whether it is about a book's availability or a user's borrowing history is accurate and up to date. The DBMS also allows quick searching, prevents duplication, and makes sure that sensitive information is handled securely.

Overall, the Online Book Management System provides a user-friendly and efficient way to handle book-related services, making the process convenient for readers while giving administrators powerful tools to manage resources effectively.

2. Problem Overview

An online bookstore involves multiple interconnected entities such as Books, Customers, Orders, Payments, and Administrators. Without an effective database management system, the following challenges arise:

- Difficulty in handling large volumes of book and customer data.
- Slow and inefficient book searches without proper indexing.
- Errors and inconsistencies in inventory and stock availability.
- Inaccurate recording of orders, payments, and transaction histories.
- Security vulnerabilities in managing customer details and payment information.
- Lack of reliable reports and analytics for sales and decision-making.

A Database Management System (DBMS) addresses these issues by providing a secure, consistent, and efficient framework to store, manage, and retrieve data, ensuring smooth operation of the online bookstore.

3. Scope

The scope of this project is to design and implement a relational database that will effectively manage the operations of an online bookstore. The system will be designed to:

- Store comprehensive details of all key entities, including books, authors, publishers, customers, orders, order items, and shippings.
- Represent the complex relationships between these entities in a logical and optimized structure to avoid data redundancy and ensure consistency.
- Enable efficient retrieval, insertion, and updating of records through structured queries, supporting all core business functions.
- Support future scalability by accommodating a growing catalog of books and an increasing volume of customer orders.
- Ensure data security and integrity to protect sensitive customer and sales information.

This project will serve as a foundational management tool, designed to optimize business processes and provide a reliable platform for a modern e-commerce bookstore.

4. Objectives

The primary objectives of the **Online Bookstore Management System** are:

1. **Centralized Data Management:** Develop a relational database that integrates all bookstore data into a unified system, eliminating duplication and redundancy.
2. **Efficient Data Retrieval:** Optimize the database design to allow fast and accurate retrieval of information for books, authors, customers, and orders.
3. **Relationship Mapping:** Implement proper handling of one-to-many and many-to-many relationships, such as:
 - One customer placing **multiple orders**.
 - One author writing **many books**.
 - One order containing **multiple books**.
4. **Data Integrity and Consistency:** Enforce **Primary Keys**, **Foreign Keys**, and constraints to maintain the accuracy and validity of all records, from book prices to customer addresses.
5. **Security and Access Control:** Protect sensitive information through proper database security measures to safeguard customer data and sales records.
6. **Scalability and Maintainability:** Ensure the database can handle future growth in book inventory and customer volume and can be easily updated without disrupting operations.

7. **Support for Reporting and Analytics:** Provide the ability to generate reports for:
 - Best-selling book summaries.
 - Customer purchasing trends.
 - Inventory levels and sales performance.
8. **Minimizing Redundancy:** Apply normalization techniques to reduce data duplication, optimize storage, and maintain consistency across all tables.

5. Significance of the Project

The proposed Online Bookstore Management System will significantly improve e-commerce operations by providing a structured and reliable platform for all business activities. Its key contributions include:

- **Improving Operational Efficiency:** The system automates manual processes, such as inventory tracking and order fulfillment, which reduces delays and operational costs.
- **Enhancing Data Reliability:** By eliminating redundancy and enforcing data integrity, the project ensures that information about books, customers, and sales is consistently accurate.
- **Boosting Customer Satisfaction:** Faster and more accurate order processing, along with a reliable system for managing customer data, leads to smoother and positive shopping experience.
- **Enabling Data-Driven Decisions:** The system provides a foundation for generating real-time reports and analytics on sales trends and inventory levels, which helps management make informed business decisions.
- **Ensuring Scalability:** The project's relational database design allows the bookstore to easily grow its catalog and customer base without requiring a complete overhaul of its data management system.

Main Entities and Attributes:

1. Books

- Book_ID (PK)
- Title
- ISBN
- Price
- Genre
- Book_Status
- Author_ID (FK)
- Publisher_ID (FK)

2. Author

- Author_ID (PK)
- Name
- Nationality
- DOB

3. Publisher

- Publisher_ID (PK)
- Name
- E-Mail
- Address

4. Customer

- Customer_ID (PK)
- Name
- Password
- E-Mail
- SignUpDate
- Address
- Phone_Number

5. Orders

- Order_ID (PK)
- Customer_ID (FK)
- OrderDate
- OrderStatus
- Total_Amount

6. Payments

- Payment_ID (PK)
- Order_ID (FK)
- Amount
- Payment_Date
- Payment_Status
- Payment_Method

7. Shippings

- Shipping_ID (PK)
- Order_ID (FK)
- Shipping_Date
- Shipping_Status
- Shipping_Address

8. Subscription

- Subscription_ID (PK)
- SubscriptionType
- Start_Date
- End_Date
- Subscription_Status

9. Review

- Review_ID (PK)
- Book_ID (FK)
- Customer_ID (FK)
- Ratings
- Comments
- Review_Date

10. Order_Details

- OrderDetails_ID (PK)
- Order_ID (FK)
- Book_ID (FK)
- Quantity
- ItemPrice

Relationships

- **Author-Book:** One-to-Many
- **Publisher-Book:** One-to-Many
- **Book-Order:** Many-to-Many
- **Book-Review:** One-to-Many
- **Book-Order_Details:** One-to-Many
- **Customer-Order:** One-to-Many
- **Order-Shipping:** One-to-One
- **Order-Payment:** One-to-One
- **Order-Order_Details:** One-to-One
- **Customer-Shipping:** One-to-Many
- **Customer-Payment:** One-to-Many
- **Customer-Review:** One-to-Many
- **Customer-Subscription:** One-to-Many
- **Subscription-Payment:** One-to-Many

ER Diagram

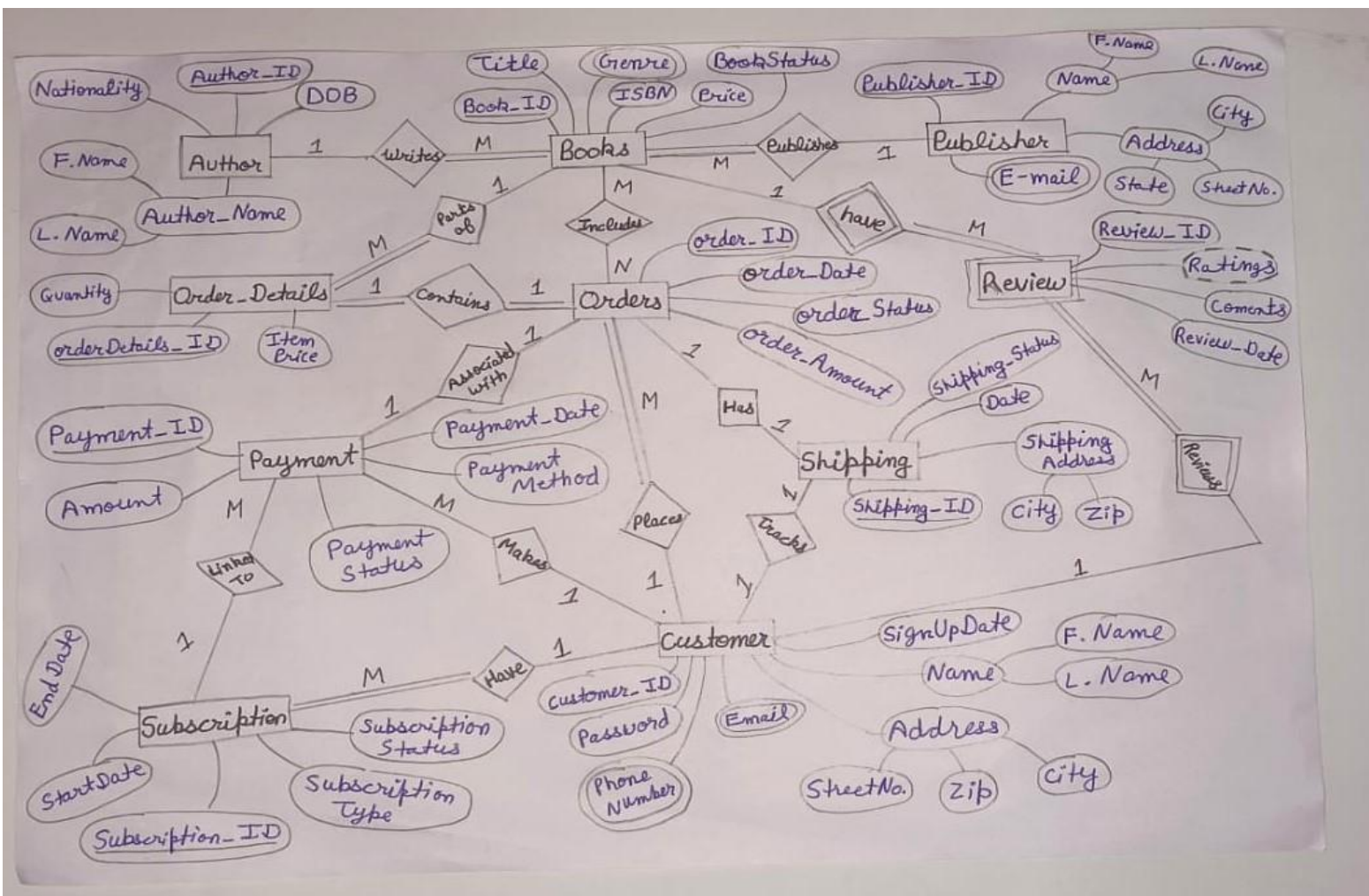


Fig.1: ER Diagram of Online Bookstore Management System

ER Model of Online Bookstore Management System

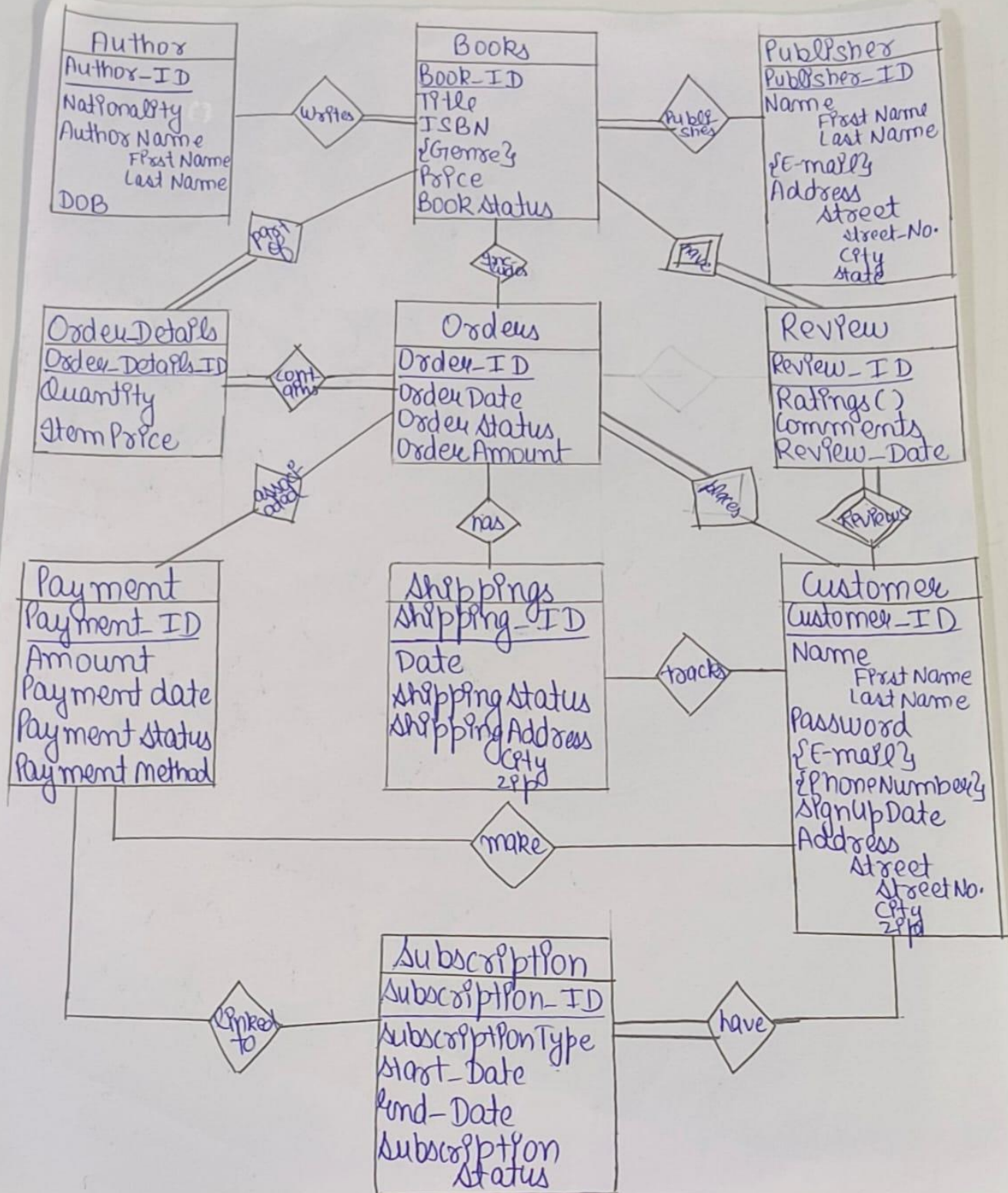


Fig.2: ER Model of Online Bookstore Management System

ER to Relational Model (Schema)

Books (Book_ID [PK], Title, ISBN, Price, Genre, Book_Status, Author_ID [FK], Publisher_ID[FK])

Author (Author_ID [PK], Name, DOB, Nationality)

Publisher (Publisher_ID [PK], Name, E-mail, Address)

Customer (Customer_ID [PK], Name, Password, E-mail, Address, SignUpDate, Phone_Number)

Orders (Order_ID [PK], Customer_ID [FK], OrderDate, OrderStatus, Total_Amount)

Payments (Payment_ID [PK], Order_ID [FK], Amount, Payment_Date, Payment_Status, Payment_Method, Subscription_ID [FK], Customer_ID [FK])

Shippings (Shipping_ID [PK], Order_ID [FK], Shipping_Date, Shipping_Status, Shipping_Address, Customer_ID[FK])

Subscription (Subscription_ID [PK], SubscriptionType, Start_Date, End_Date, Subscription_Status, Customer_ID [FK])

Review (Review_ID[PK], Book_ID [FK], Customer_ID [FK], Ratings, Comments, Review_Date)

Order_Details (OrderDetails_ID [PK], Order_ID [FK], Book_ID [FK], Quantity, ItemPrice)

Relational Algebra (RA) (Queries)

Q1. Find the titles of all books.

$\pi_{\text{Title}}(\text{Book})$

Q2. List all customers with the name 'Maria'.

$\sigma_{\text{Name}='Maria'}(\text{Customer})$

Q3. Find all reviews with a rating of 5.

$\pi_{\text{Comment}}(\sigma_{\text{Rating}=5}(\text{Review}))$

Q4. Get the details of all pending orders.

$\sigma_{\text{OrderStatus}='Pending'}(\text{Orders})$

Q5. Find the titles and prices of all books by author 'J.K. Rowling'.

$\pi_{\text{Title, Price}}(\sigma_{\text{Name}='J.K.Rowling'}(\text{Book} \bowtie \text{Author}))$

Q6. Get the names and email addresses of all customers who have ordered a book.

$\pi_{\text{Name, Email}}(\text{Customer} \bowtie \text{Orders} \bowtie \text{Order_Details})$

Q7. Find the titles of all books that have a review with a rating of 4 or higher.

$\pi_{\text{Title}}(\sigma_{\text{Rating} \geq 4}(\text{Book} \bowtie \text{Review}))$

Q8. Find the names of all customers who have placed an order.

$\pi_{\text{Name}}(\text{Customer} \bowtie \text{Orders})$

Q9. List all book titles and the names of their authors.

$\pi_{\text{Title}, \text{Name}}(\text{Book} \bowtie \text{Author})$

Q10. Get the names of customers who have a subscription.

$\pi_{\text{Name}}(\text{Customer} \bowtie \text{Subscription})$

Q11. Find the titles of all books published by 'Penguin'.

$\pi_{\text{Title}}(\sigma_{\text{Name}='Penguin'}(\text{Book} \bowtie \text{Publisher}))$

Q12. Get the total amount of all payments made by a customer.

$\pi_{\text{Amount}}(\text{Customer} \bowtie \text{Payments})$

Q13. Find the names of all authors whose books have a rating of 5.

$\pi_{\text{Name}}(\sigma_{\text{Rating}=5}(\text{Author} \bowtie \text{Book} \bowtie \text{Review}))$

Q14. Get the email of a customer who placed an order on '2024-05-15'.

$\pi_{\text{Email}}(\sigma_{\text{OrderDate}='2024-05-15'}(\text{Customer} \bowtie \text{Orders}))$

Q13. Find the names of customers who have submitted a review.

$\pi_{\text{Name}}(\text{Customer} \bowtie \text{Review})$

Q15. Find all payments for orders that are 'Shipped'.

$\pi_{\text{Payments.Amount}}(\sigma_{\text{ShippingStatus}='Shipped'}(\text{Payments} \bowtie \text{Orders} \bowtie \text{Shippings}))$

Q16. Get the titles of books that have been reviewed.

$\pi_{\text{Title}}(\text{Book} \bowtie \text{Review})$

Q17. Find the titles of all books that have been ordered at least once.

$\pi_{\text{Title}}(\text{Book} \bowtie \text{Order_Details})$

Q18. Find all orders that contain the book 'The Martian'.

$\pi_{\text{OrderID}}(\sigma_{\text{Title}='TheMartian'}(\text{Orders} \bowtie \text{Order_Details} \bowtie \text{Book}))$

Q19. Find the total quantity of the book 'The Martian' that has been ordered.

$\pi_{\text{Quantity}}(\sigma_{\text{Title}='TheMartian'}(\text{Book} \bowtie \text{Order_Details}))$

Q20. Find all customers who have a subscription and have ordered the book 'The Martian'.

$\pi_{\text{Name}}(\text{Customer} \bowtie \text{Subscription} \bowtie \text{Orders} \bowtie \text{Order_Items} \bowtie \sigma_{\text{Title}='TheMartian'}(\text{Book}))$

Q21. Find all customers who have an outstanding payment for their subscription (assuming payment has a status 'Pending').

$\pi_{\text{Name}}(\sigma_{\text{Status}='Pending'}(\text{Customer} \bowtie \text{Payments} \bowtie \text{Subscription}))$

Q22. Find all customers who have either made a payment or written a review.

$(\pi_{\text{CustomerID}}(\text{Customer} \bowtie \text{Payments})) \cup (\pi_{\text{CustomerID}}(\text{Customer} \bowtie \text{Review}))$

Q23. Get the names of customers who have a subscription but have not written a review.

$(\pi_{\text{CustomerID}}(\text{Customer} \bowtie \text{Subscription})) - (\pi_{\text{CustomerID}}(\text{Customer} \bowtie \text{Review}))$

Q24. Find all customers who have both a subscription and a review.

$(\pi_{\text{CustomerID}}(\text{Customer} \bowtie \text{Subscription})) \cap (\pi_{\text{CustomerID}}(\text{Customer} \bowtie \text{Review}))$

Q25. List the names of all authors and publishers that have the same name.

$\pi_{\text{Author.Name}}(\text{Author}) \cap \pi_{\text{Publisher.Name}}(\text{Publisher})$