**Online Bookstore Management System**

*Project Report with ER Model Reference*

## **1. Introduction : The Digital Bookshelf**

## Think about your favourite local bookstore. Now, imagine the owner trying to keep track of every single book, every customer's order, and every sale using just spreadsheets or, even worse, a paper ledger. It would be a nightmare. A single typo could mean a book is listed as in-stock when it's not, leading to a disappointed customer. That's the problem we're solving. We're moving away from those clunky, old-school methods that are slow and full of potential mistakes. This project is about building a smart, centralized system that handles all of that automatically. It's a single place to manage everything—customer details, the book inventory, and all the transactions—smoothly and securely. Essentially, we're creating a reliable digital backbone for the bookstore, so the owners can spend less time fighting with paperwork and more time connecting people with books they'll love.

## **2. Problem Overview**

* **Manual Processing Delays** -> Slow retrieval and updating of information.
* **Data Redundancy & Inconsistency** -> Storing the same information (e.g., customer details) in multiple locations, leading to inconsistencies.
* **Lack of Data Integrity** -> High risk of human error during manual entry, resulting in inaccurate inventory counts, incorrect order details, and unreliable reporting.
* **Inefficient operations** -> Unprotected or poorly managed records pose significant threats.

A well-structured system eliminates these problems by providing a centralized, relational database with properly defined entities, relationships, and constraints.

## **3. Scope of the Project**

Designing a Relational Database

* **Comprehensive Data Storage** - Store all details of customers ,books, orders, employees, and suppliers.
* **Optimized Relationships** - Represent complex relationships between entities in a logical and optimized structure
* **Fast Retrieval & Updates** - Enable quick access and modification of records via structured queries.
* **Scalability & Security** - Support future growth and ensure data security with access control.

## **4. Objectives**

**1. Centralized Data Management** - Unified system, eliminating duplication.

**2. Data Retrieval** - Fast and accurate information access.

**3. Relationship Mapping -** Handle one-to-many and many-to-many relationships.

**4. Security & Access Control** - Protect sensitive information.

**5. Scalability & Maintainability** - Handle future growth and easy updates.

**6. Reporting & Analytics** - Generate comprehensive reports.

**7. Minimizing Redundancy** - Normalization for data consistency.

## 

## **5. Significance of the Project**

* **Reduced Operational Delays :** Streamlined processes for faster service.
* **Improved Accuracy :** Reliable and consistent records.

* **Enhanced Customer Satisfaction :** Quicker and more efficient service.
* **Strengthened Security :** Protection of sensitive financial data.

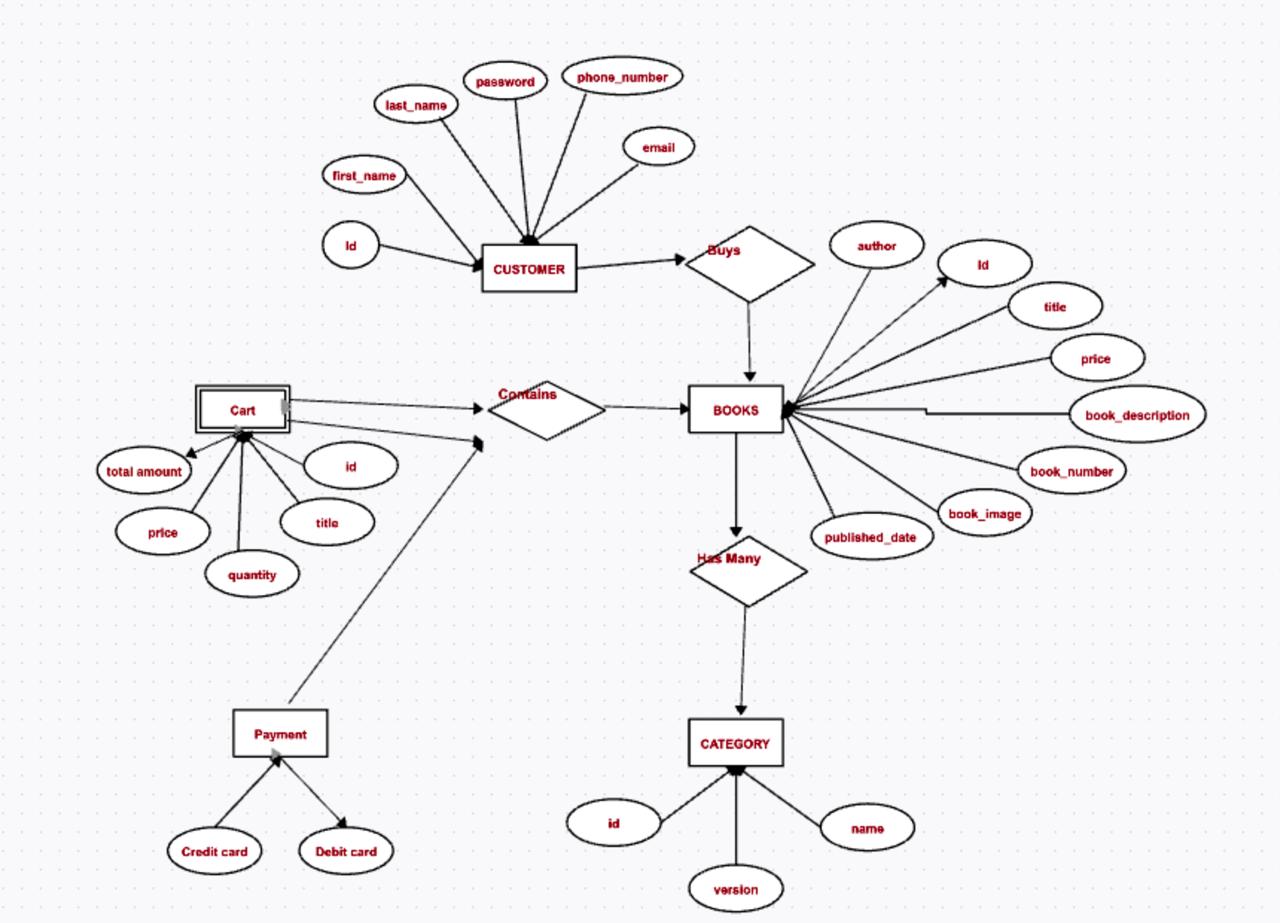
**6. Main Entities and Attributes**

|  |  |
| --- | --- |
| **Entity** | **Attributes** |
| Admin | \*AdminID (Primary Key)  \* Name  \* Email  \* Password |
| Customer | \* CustomerID (Primary Key)  \* Name  \* Email  \* Phone  \* Address |
| Order | \* OrderID (Primary Key)  \* CustomerID (Foreign Key)  \* OrderDate  \* TotalAmount  \* Status |
| Payment | \* PaymentID (Primary Key)  \* OrderID (Foreign Key)  \* PaymentDate  \* Amount  \* PaymentMethod |
| OrderItem | \* OrderItemID (Primary Key)  \* OrderID (Foreign Key)  \* BookID (Foreign Key)  \* Quantity  \* Subtotal |
| Book | \* BookID (Primary Key)  \* Title  \* Author  \* ISBN  \* Price  \* StockQuantity  \* PublisherID (Foreign Key)  \* CategoryID (Foreign Key) |
| Category | \* CategoryID (Primary Key)  \* CategoryName |
| Publisher | \* PublisherID (Primary Key)  \* PublisherName  \* Contact |
|  |  |
|  |  |
|  |  |

**Relationships**:

* Customer3Order: One-to-Many
* Book3Order\_Item: One-to-Many
* Order3Order\_Item: One-to-Many
* Book3Supplier: Many-to-Many (via a separate linking table if needed)

7. **ER MODEL :**



## 

**ER to Relational Model (Schema):**

Customer(Customer\_ID [PK],Name, Address, Phone, Email,Date\_of\_Birth)

Book(Book\_ID [PK], Title, Author, Genre, Price, Stock\_Quantity)

Order(Order\_ID [PK], Customer\_ID [FK], Order\_Date, Total\_Amount, Status)

Order\_Item(Order\_ID [FK], Book\_ID [FK], Quantity, Subtotal) 3 Composite PK

Employee(Employee\_ID [PK], Name, Designation, Salary)

Supplier(Supplier\_ID [PK], Name, Contact\_Person, Phone, Email)

**8. Relational Algebra (RA) Queries.**

**1. How to insert a new admin into the Admin table?**

INSERT INTO Admin (AdminID, Name, Email, Password)

VALUES (1, 'Aryan', 'aryan@gmail.com', 'pass123');

**2. How to insert a new customer into the Customer table?**

INSERT INTO Customer (CustomerID, Name, Email, Phone, Address)

VALUES (1, 'Rahul', 'rahul@gmail.com', '9876543210', 'Delhi');

**3. How to insert a new book into the Book table?**

INSERT INTO Book (BookID, Title, Author, ISBN, Price, StockQuantity, PublisherID, CategoryID)

VALUES (1, 'Book1', 'Author1', '1234567890', 500, 10, 1, 1);

**4. How to insert a new category into the Category table?**

INSERT INTO Category (CategoryID, CategoryName)

VALUES (1, 'Fiction');

**5. How to insert a new publisher into the Publisher table?**

INSERT INTO Publisher (PublisherID, PublisherName, Contact)

VALUES (1, 'Publisher1', '1234567890');

**6. How to create a new order in the Order table?**

INSERT INTO `Order` (OrderID, CustomerID, OrderDate, TotalAmount, Status)

VALUES (1, 1, CURDATE(), 1000, 'Pending');

**7. How to record a payment in the Payment table?**

INSERT INTO Payment (PaymentID, OrderID, PaymentDate, Amount, PaymentMethod)

VALUES (1, 1, CURDATE(), 1000, 'Credit Card');

**8. How to add items to an order in the OrderItem table?**

INSERT INTO OrderItem (OrderItemID, OrderID, BookID, Quantity, Subtotal)

VALUES (1, 1, 1, 2, 1000);

**9. How to update the price of a book?**

UPDATE Book SET Price = 600 WHERE BookID = 1;

**10. How to reduce the stock quantity after a purchase?**

UPDATE Book SET StockQuantity = StockQuantity - 2 WHERE BookID = 1;

**11. How to change the status of an order?**

UPDATE `Order` SET Status = 'Shipped' WHERE OrderID = 1;

**12. How to update the phone number of a customer?**

UPDATE Customer SET Phone = '9876501234' WHERE CustomerID = 1;

**13. How to delete a book from the Book table?**

DELETE FROM Book WHERE BookID = 1;

**14. How to delete a customer from the Customer table?**

DELETE FROM Customer WHERE CustomerID = 1;

**15. How to delete an order from the Order table?**

DELETE FROM `Order` WHERE OrderID = 1;

**16.How to delete all items of an order from OrderItem table?**

DELETE FROM OrderItem WHERE OrderID = 1;

**17. How to list all books along with their category and publisher?**

SELECT B.Title, B.Author, C.CategoryName, P.PublisherName

FROM Book B

JOIN Category C ON B.CategoryID = C.CategoryID

JOIN Publisher P ON B.PublisherID = P.PublisherID;

**18. How to view orders with customer details?**

SELECT O.OrderID, C.Name, O.OrderDate, O.Status

FROM `Order` O

JOIN Customer C ON O.CustomerID = C.CustomerID;

**19. How to list order items with corresponding book titles?**

SELECT OI.OrderItemID, B.Title, OI.Quantity, OI.Subtotal

FROM OrderItem OI

JOIN Book B ON OI.BookID = B.BookID

WHERE OI.OrderID = 1;

**20. How to find books written by a particular author?**

SELECT \* FROM Book WHERE Author LIKE '%Author1%';

**21. How to see payments for a specific order?**

SELECT PaymentID, PaymentDate, Amount, PaymentMethod

FROM Payment

WHERE OrderID = 1;

**22. How to count total orders for each customer?**

SELECT C.CustomerID, C.Name, COUNT(O.OrderID) AS TotalOrders

FROM Customer C

LEFT JOIN `Order` O ON C.CustomerID = O.CustomerID

GROUP BY C.CustomerID, C.Name;

**23. How to find top-selling books by quantity?**

SELECT B.Title, SUM(OI.Quantity) AS TotalSold

FROM Book B

JOIN OrderItem OI ON B.BookID = OI.BookID

GROUP BY B.Title

ORDER BY TotalSold DESC;

**24. How to find books which are out of stock?**

SELECT Title, StockQuantity FROM Book WHERE StockQuantity = 0;

**25. How to calculate total revenue of each order?**

SELECT O.OrderID, SUM(OI.Subtotal) AS TotalAmount

FROM `Order` O

JOIN OrderItem OI ON O.OrderID = OI.OrderID

GROUP BY O.OrderID;