UNIVERSITY INSTITUTE OF COMPUTING

CASE STUDY REPORT
ON
ONLINE BOOKSTORE MANAGEMENT SYSTEM

Program Name: BCA

Subject Name/Code: Database Management System (23CAT-251)

Submitted By:

Name:Yuvraj Srivastava

UID: 21BCA1210

Section: 4-'A'

Submitted To:

Name: Mr. Arvinder Singh

Designation: Assistant Professor

INTRODUCTION

The Online Bookstore Management System is a database project developed using MySQL, aimed at managing book data, customers, suppliers, orders, payments, and inventory stock. The main objective is to digitize bookstore operations, reduce manual workload, and improve efficiency and data accessibility.

The objective is to build a relational database that ensures data integrity, easy access, and efficient handling of daily bookstore operations. Using Entity-Relationship modelling and SQL queries, the system allows for smooth inventory tracking, order processing, billing management, and customer handling.

This system allows:

- Easy addition and retrieval of book and customer data
- Efficient order processing and inventory updates
- Maintenance of customer order history and invoices
- Payment tracking and report generation
- Supplier stock management

TECHNIQUES

The primary technology used in this project is MySQL, an open-source relational database management system. The following techniques have been implemented:

Entity-Relationship Modeling for data structure visualisation

Normalisation to organise data efficiently and remove redundancy

SQL Queries for data manipulation and retrieval

Use of Constraints like PRIMARY KEY, FOREIGN KEY to enforce relationships

Join operations to combine data from multiple tables

Aggregate Functions to summarize and analyze data

Filtering and Sorting to extract meaningful insights from the dataset

Stored Procedures and Views (optional enhancements) for automation

SYSTEM CONFIGURATION

Software Requirements:

- MySQL Server / Workbench
- ER Diagram Design Tool (e.g., dbdiagram.io, Lucid chart)
- Text Editor or IDE (e.g., VS Code, Sublime Text)

Hardware Requirements:

- RAM: Minimum 4 GB (8 GB recommended for better performance)
- Processor: Intel i3 or above
- Storage: Minimum 500 MB for database and MySQL setup

Operating System: Windows/Linux/macOS

The system has been developed and tested on Windows 10 with MySQL Workbench 8.0.34. The choice of open-source tools makes the project accessible and deployable on a wide range of platforms.

INPUT

The system accepts a variety of inputs to populate the database and simulate bookstore functionality. These inputs include:

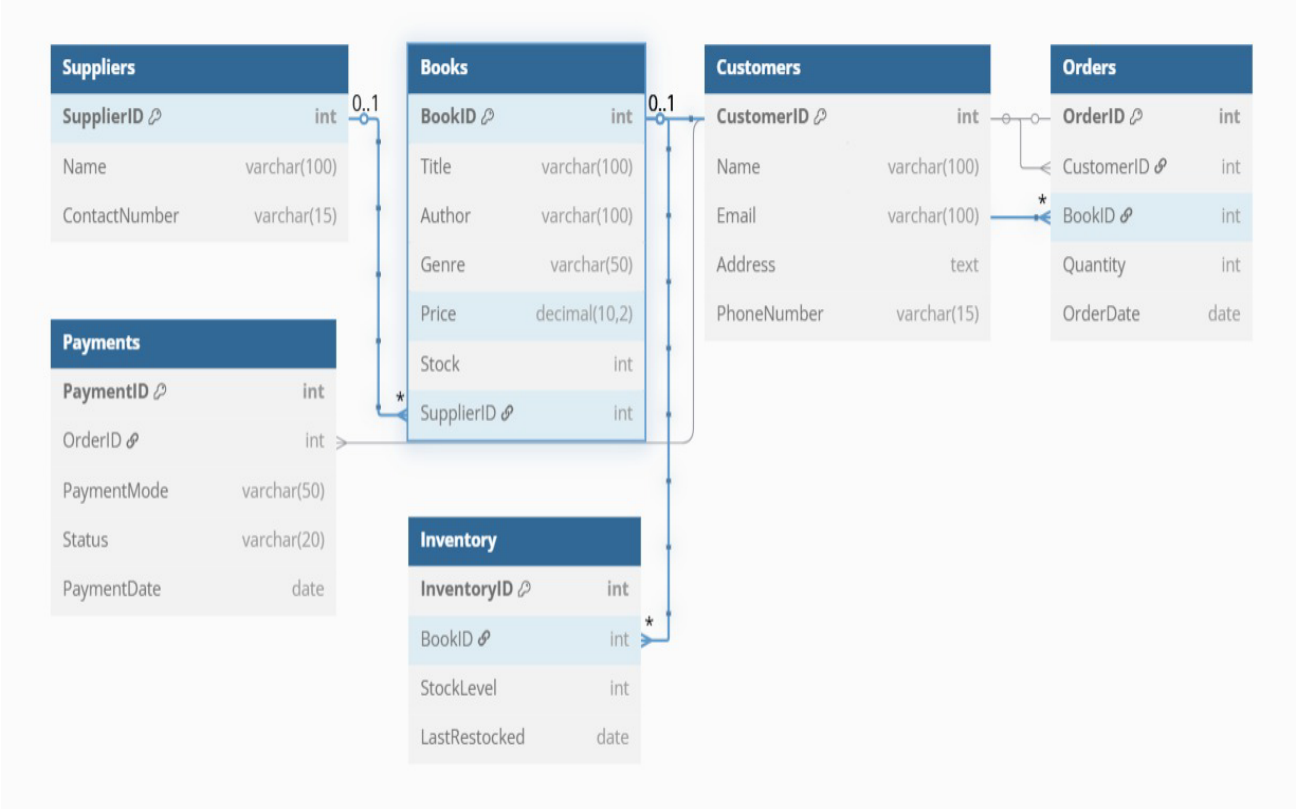
Book Information: Title, Author, Genre, Price, Stock
Customer Information: Name, Email, Address, Contact Number
Order Details: Order ID, Book ID, Customer ID, Quantity, Order Date
Payment Details: Mode, Status, Payment Date
Supplier Information: Name, Contact Number
Inventory: Stock Level, Last Restocked

Each input plays a vital role in maintaining an up-to-date and accurate database.

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

ENTITY-RELATIONSHIP DIAGRAM



The Entity-Relationship (ER) diagram outlines the structure and relationships among different entities of the bookstore. It forms the blueprint for the actual database schema.

Each entity has clearly defined attributes and is connected using appropriate relationships like one-to-many and one-to-one, ensuring normalization and avoiding data redundancy.

RELATIONSHIP BETWEEN TABLES

These relationships ensure that the relational database mirrors real-world interactions within a Bookstore

No.	Relationship	Parent Table	Child Table	Foreign Key	Description
1	One-to-many	Books	Orders	BookID	A book can appear in many orders
2	One-to-many	Customers	Orders	CustomerID	A customer can place many orders
3	One-to-one	Orders	Payments	OrderID	Each order has one payment
4	One-to-many	Suppliers	Books	SupplierID	A supplier can supply many books
5	One-to-many	Books	Inventory	BookID	Inventory tracks each book's stock

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

TABULAR FORMAT (SCHEMA)

Table Name	Primary Key	Foreign Key	Description
Books	BookID	SupplierID	Stores book data
Customers	CustomerID	—	Stores customer details
Orders	OrderID	CustomerID, BookID	Stores order details
Payments	PaymentID	OrderID	Tracks payments
Suppliers	SupplierID	—	Stores supplier info
Inventory	InventoryID	BookID	Tracks stock for each book

TABLE CREATION

```sql
-- 1. Suppliers Table
CREATE TABLE Suppliers (
    SupplierID INT PRIMARY KEY,
    Name VARCHAR(100),
    ContactNumber VARCHAR(15)
);
```

```sql
INSERT INTO Suppliers VALUES
(1, 'Pearson Publishers', '9876543210'),
(2, 'McGraw-Hill Education', '9123456780'),
(3, 'Global Book Distributors', '9988776655');
```

```sql
-- 2. Books Table
CREATE TABLE Books (
    BookID INT PRIMARY KEY,
    Title VARCHAR(100),
    Author VARCHAR(100),
    Genre VARCHAR(50),
    Price DECIMAL(10,2),
    Stock INT,
    SupplierID INT,
    FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID)
);
```

```sql
INSERT INTO Books VALUES
(101, 'The Alchemist', 'Paulo Coelho', 'Fiction', 399.00, 50, 1),
(102, 'Clean Code', 'Robert Martin', 'Programming', 699.00, 20, 2),
(103, 'Introduction to Algorithms', 'CLRS', 'Education', 1200.00, 15, 2),
(104, '1984', 'George Orwell', 'Fiction', 299.00, 0, 1),
(105, 'Data Structures in C', 'Reema Thareja', 'Education', 550.00, 9, 3);

-- 3. Customers Table
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Address TEXT,
    PhoneNumber VARCHAR(15)
);
```

```sql
INSERT INTO Customers VALUES
(201, 'Aman Sharma', 'aman@example.com', 'Delhi', '8881234567'),
(202, 'Sneha Mehta', 'sneha@example.com', 'Mumbai', '9998765432'),
(203, 'Ravi Yadav', 'ravi@example.com', 'Lucknow', '9876541230');

-- 4. Orders Table
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    BookID INT,
    Quantity INT,
    OrderDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (BookID) REFERENCES Books(BookID)
);
```

```sql
INSERT INTO Orders VALUES
(301, 201, 101, 2, '2025-04-10'),
(302, 202, 103, 1, '2025-04-11'),
(303, 203, 104, 1, '2025-04-12'),
(304, 201, 105, 3, '2025-04-13');

-- 5. Payments Table
CREATE TABLE Payments (
    PaymentID INT PRIMARY KEY,
    OrderID INT UNIQUE,
    PaymentMode VARCHAR(50),
    Status VARCHAR(20),
    PaymentDate DATE,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
);
```

```sql
INSERT INTO Payments VALUES
(401, 301, 'Credit Card', 'Paid', '2025-04-10'),
(402, 302, 'UPI', 'Paid', '2025-04-11'),
(403, 303, 'Cash on Delivery', 'Unpaid', NULL),
(404, 304, 'Net Banking', 'Paid', '2025-04-13');

-- 6. Inventory Table
CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY,
    BookID INT,
    StockLevel INT,
    LastRestocked DATE,
    FOREIGN KEY (BookID) REFERENCES Books(BookID)
);

INSERT INTO Inventory VALUES
(501, 101, 50, '2025-03-20'),
(502, 102, 20, '2025-03-25'),
(503, 103, 15, '2025-04-01'),
(504, 104, 0, '2025-02-18'),
```

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

SQL QUERIES (13 Queries)

```
SELECT b.Title, b.Author, s.Name AS SupplierName
FROM Books b
JOIN Suppliers s ON b.SupplierID = s.SupplierID;
```

Title	Author	SupplierName
The Alchemist	Paulo Coelho	Pearson Publishers
1984	George Orwell	Pearson Publishers
Clean Code	Robert Martin	McGraw-Hill Education
Introduction to Algorithms	CLRS	McGraw-Hill Education
Data Structures in C	Reema Thareja	Global Book Distributors

```
SELECT DISTINCT c.CustomerID, c.Name, c.Email
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID;
```

CustomerID	Name	Email
201	Aman Sharma	aman@example.com
202	Sneha Mehta	sneha@example.com
203	Ravi Yadav	ravi@example.com

```
SELECT o.OrderID, c.Name AS Customer, b.Title AS Book, p.Status AS PaymentStatus
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Books b ON o.BookID = b.BookID
LEFT JOIN Payments p ON o.OrderID = p.OrderID;
```

OrderID	Customer	Book	PaymentStatus
301	Aman Sharma	The Alchemist	Paid
304	Aman Sharma	Data Structures in C	Paid
302	Sneha Mehta	Introduction to Algorithms	Paid
303	Ravi Yadav	1984	Unpaid

```
SELECT Title, Author
FROM Books
WHERE Stock = 0;
```

Title	Author
1984	George Orwell

```
SELECT SUM(b.Price * o.Quantity) AS TotalRevenue
FROM Orders o
JOIN Payments p ON o.OrderID = p.OrderID
JOIN Books b ON o.BookID = b.BookID
WHERE p.Status = 'Paid';
```

TotalRevenue
3648.00

```
SELECT b.Title, i.LastRestocked
FROM Books b
JOIN Inventory i ON b.BookID = i.BookID
ORDER BY i.LastRestocked DESC;
```

Title	LastRestocked
Data Structures in C	2025-04-05
Introduction to Algorithms	2025-04-01
Clean Code	2025-03-25
The Alchemist	2025-03-20
1984	2025-02-18

```
SELECT b.Title, i.StockLevel
FROM Books b
JOIN Inventory i ON b.BookID = i.BookID
WHERE i.StockLevel < 10;
```

Title	StockLevel
1984	0
Data Structures in C	9

```
SELECT c.Name, COUNT(o.OrderID) AS OrderCount
FROM Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.Name;
```

Name	OrderCount
Aman Sharma	2
Sneha Mehta	1
Ravi Yadav	1

```
SELECT s.Name, COUNT(b.BookID) AS BookCount
FROM Suppliers s
LEFT JOIN Books b ON s.SupplierID = b.SupplierID
GROUP BY s.Name;
```

Name	BookCount
Pearson Publishers	2
McGraw-Hill Education	2
Global Book Distributors	1

```sql
SELECT o.OrderID, c.Name, b.Title, o.OrderDate
FROM Orders o
JOIN Payments p ON o.OrderID = p.OrderID
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Books b ON o.BookID = b.BookID
WHERE p.Status = 'Unpaid';
```

OrderID	Name	Title	OrderDate
303	Ravi Yadav	1984	2025-04-12

```sql
SELECT Genre, AVG(Price) AS AvgPrice
FROM Books
GROUP BY Genre;
```

Genre	AvgPrice
Fiction	349.000000
Programming	699.000000
Education	875.000000

```sql
SELECT DISTINCT c.Name, b.Title, b.Price
FROM Orders o
JOIN Books b ON o.BookID = b.BookID
JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE b.Price > 1000;
```

Name	Title	Price
Sneha Mehta	Introduction to Algorithms	1200.00

```sql
SELECT b.Title
FROM Books b
LEFT JOIN Orders o ON b.BookID = o.BookID
WHERE o.BookID IS NULL;
```

Title
Clean Code

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

SUMMARY

The Online Bookstore Management System (OBMS) designed in this project serves as a comprehensive solution for managing online bookstore operations through a relational MySQL database. It incorporates crucial modules such as book listings, customer details, order tracking, inventory control, payments, and supplier coordination.

Each entity is normalized and linked using appropriate relationships to ensure data consistency, integrity, and reduced redundancy. The implementation of SQL queries demonstrates the ability to retrieve complex information using techniques like joins, aggregate functions, filtering, and grouping.

This system lays the foundation for building a larger, more interactive platform to support bookstore management.

CONCLUSION

Observations:
The database successfully simulates core functionalities of an online bookstore.
Relationships between entities are clearly defined and enforced.
SQL queries enhance usability and data insight.
Scalable for larger e-commerce implementations.

Limitations:
The system lacks a user-friendly interface.
Payment module assumes manual updates.
No login/authentication features included.
Static data environment.

Future Scope:
Integration with a web or mobile frontend
Implementing role-based access (admin/customer)
Linking inventory to restocking automation
Real-time alerts and dashboard analytics
Connecting with payment and shipping A