Bookshop Management System - Project Report

# Introduction:

The Bookshop Management System is a software application designed to manage the operations of a bookshop efficiently. This system will help bookshop owners and employees to manage customers, authors, publishers, books, orders, and payments effectively.

# Objectives:

To improve inventory management and reduce overstocking.

To facilitate easy search and retrieval of books.

To streamline order management and payment processing.

To provide a user-friendly interface for bookshop staff and customers.

To generate various reports for better decision-making.

# Entities and Attributes:

Customer: customer_id, first_name, last_name, email, phone_number

Author: author_id, first_name, last_name, biography

Publisher: publisher_id, name, contact_name, phone_number, email

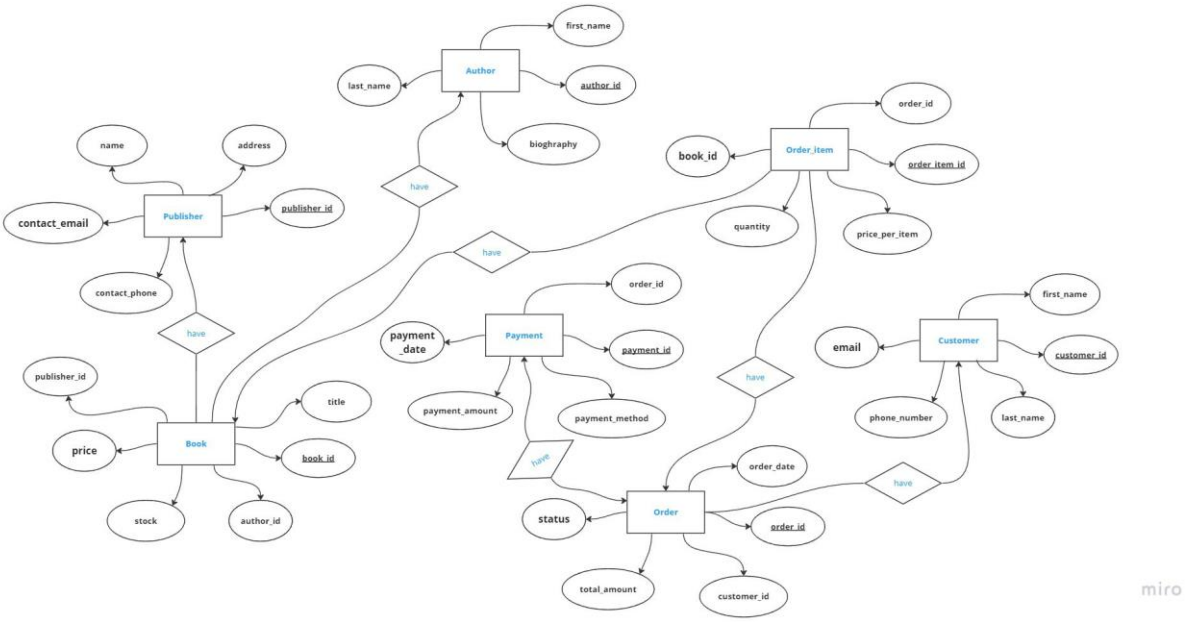
Book: book_id, title, author_id, publisher_id, ISBN, price, publication_date, stock

Order: order_id, customer_id, order_date, total_amount, status

Order_Item: order_item_id, order_id, book_id, quantity, price

Payment: payment_id, order_id, payment_date, payment_method, amount

# ER Diagram




## Normal Forms

The database structure follows 1NF, 2NF, and 3NF:

### 1NF (First Normal Form):

The first normal form requires that all attributes within a table are atomic, meaning that each attribute contains only one value and no nested data structures or repeating groups. In addition, each table must have a primary key that uniquely identifies each row.

In our Bookshop Management System, each table has a primary key, and all attributes are atomic. For example, the Customer table has a primary key "customer_id," and all other attributes (first_name, last_name, email, phone_number) contain single values.

### 2NF (Second Normal Form):

The second normal form builds upon the first normal form by adding the condition that all non-prime attributes (attributes that are not part of any candidate key) must

be fully functionally dependent on the primary key. This means that there should be no partial dependencies, where a non-prime attribute depends on only a part of the primary key.

In the Bookshop Management System, all non-prime attributes are fully functionally dependent on the primary key. For example, in the Order_Item table, the "price" and "quantity" attributes depend only on the primary key "order_item_id" and not on any part of a composite key (if one existed).

**3NF (Third Normal Form):**

The third normal form builds upon the second normal form by adding the condition that all non-prime attributes must be non-transitively dependent on the primary key. This means that there should be no transitive dependencies, where a non-prime attribute depends on another non-prime attribute that in turn depends on the primary key.

In the Bookshop Management System, all non-prime attributes are non-transitively dependent on the primary key. For example, in the Book table, the "price" attribute depends directly on the primary key "book_id" and not on any other non-prime attributes, such as "author_id" or "publisher_id".

# PL/SQL Components

The project contains several PL/SQL procedures, functions, and triggers to manage the book shop data:

## 4.1 Procedures

4.1.1 book_sales_summary

This procedure retrieves the total sales for each book, grouped by the book_id, and displays the results using the DBMS_OUTPUT package. The procedure performs a GROUP BY operation on the Order_Item table to calculate the total sales for each book.

4.1.2 delete_book

This procedure deletes a book from the Book table based on the provided book_id. It uses the SQL%ROWCOUNT attribute to determine the number of rows affected by the DELETE operation and displays the result using the DBMS_OUTPUT package.

#### 4.1.3 insert_book

This procedure inserts a new book into the Book table. It includes a user-defined exception that disallows the insertion of a book with a title shorter than 5 characters. If the title is too short, the procedure raises the title_too_short exception and displays an error message using the DBMS_OUTPUT package.

### 4.2 Functions

#### 4.2.1 count_records

This function counts the number of records in a given table and returns the result as an INTEGER. It accepts a table name as an input parameter and dynamically generates a SELECT COUNT(*) statement using the provided table name.

### 4.3 Triggers

#### 4.3.1 book_insert_trig

This trigger runs before an INSERT operation on the Book table. It selects the current number of rows in the table and displays the result using the DBMS_OUTPUT package.

## Conclusion

In conclusion, the Bookshop Management System is designed to help bookshop owners and employees manage their operations efficiently. The system provides a user-friendly interface and covers all aspects of bookshop management, including inventory, order management, and payment processing. By implementing this system, bookshops can improve their overall efficiency and customer satisfaction.