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

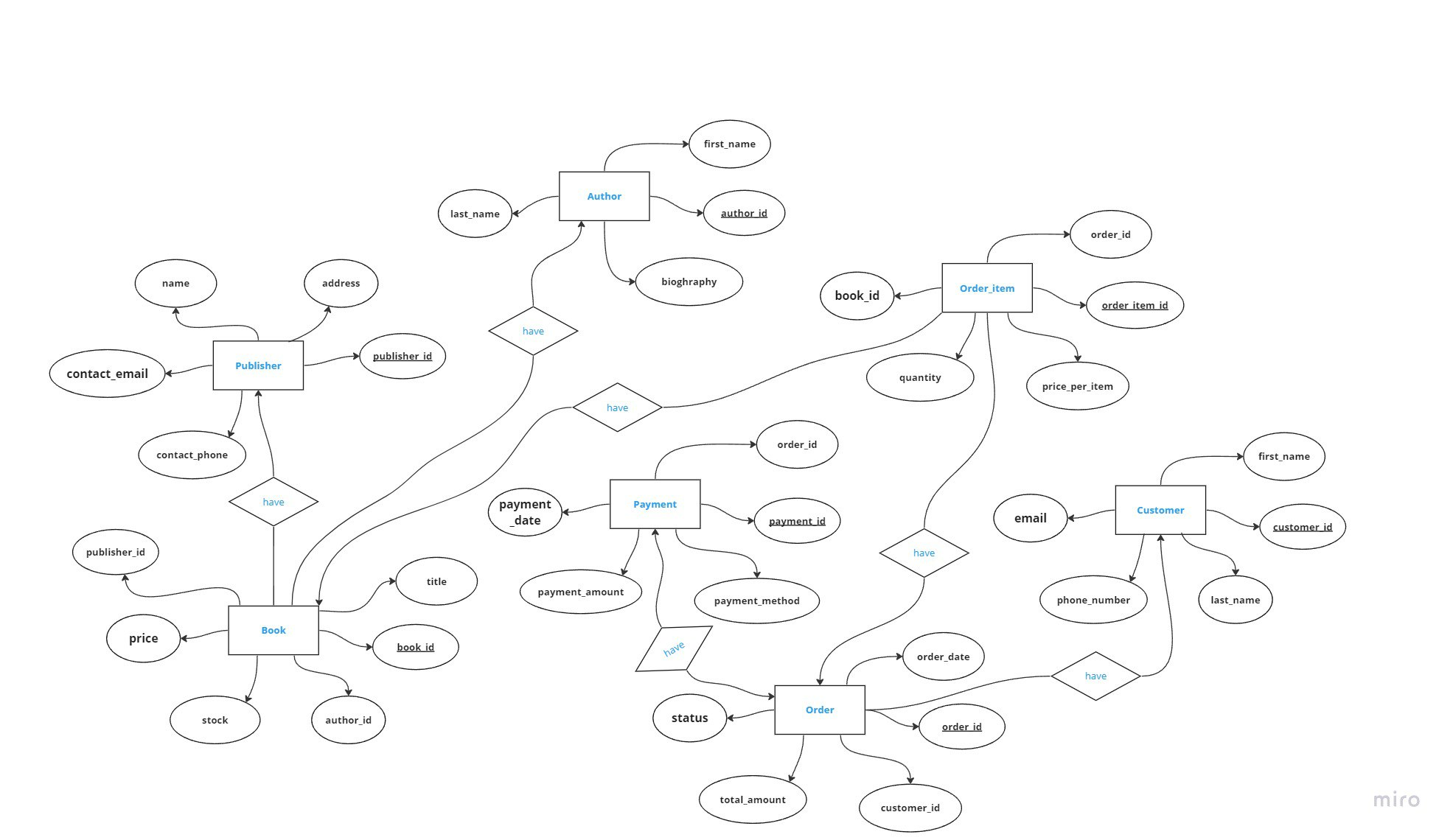
Book: book\_id, title, author\_id, publisher\_id, ISBN, price, publication\_date, stock

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.