



# Capital University of Science & Technology (CUST)

## SEMESTER PROJECT

---

**DEPARTMENT OF:** SOFTWARE ENGINEERING

**NAME:** MUHAMMAD KHAN

**REG NO:** BSE233019

**GROUP MEMBERS:**

1. RANA HAMMAD MUSHTAQ-(BSE233041)
2. MUHAMMAD ABDULLAH KHALID-(BSE233051)
3. MUHAMMAD ASIF-(BSE233078)

**DATE:** 24-10-2024

**SECTION:** 2

**COURSE:** INTRO TO DATABASE SYSTEM

**TOPIC:** ONLINE BOOKSTORE DATABASE

**SUMBITTIED TO:** MAAM HINA RASHID

# ONLINE BOOKSTORE DATABASE

## 1. Database Planning

### Explanation:

The planning phase involves identifying the key entities and understanding their relationships. The primary entities in the online bookstore include:

- **Understanding Relationships Between Entities:** It's important to recognize how these entities are interconnected:
  - **One-to-Many Relationships:** One author can write many books (e.g., J.K. Rowling wrote several Harry Potter books). Conversely, each book is written by a single author.
  - **Many-to-One Relationships:** A single book can have multiple reviews from different customers, but each review pertains to just one specific book.
  - **One-to-Many Orders:** A customer can place multiple orders over time, but each order is associated with just one customer.
- **Database Constraints:** To ensure data integrity and proper organization, we will use:
  - **Primary Keys:** These are unique identifiers for each entity. For example, every book will have a unique Book\_ID, and each customer will have a unique Customer\_ID.
  - **Foreign Keys:** These keys create relationships between entities. For instance, the Books table will include an Author\_ID to indicate which author wrote each book.
- **Normalization:** This is a process used to organize the database to minimize redundancy. By normalizing the data, we ensure that we do not store the same piece of information in multiple places, which helps maintain accuracy and efficiency.

## 2. System Definition

**System Scope:** This system is designed to manage all operations related to the online bookstore. Key functionalities include:

- **Inventory Management:** Keeping track of all books available for sale, including their stock levels.
- **Customer Management:** Storing and organizing customer data, ensuring quick access to their information for personalized service.
- **Order Management:** Overseeing the entire order process from selection to payment and delivery.
- **Review Management:** Collecting and displaying customer reviews to provide insights for other shoppers.
- **Recommendation Features:** Utilizing algorithms to suggest books to customers based on their behavior and preferences.

**Purpose:** The system's purpose is to centralize and automate the operations of an online bookstore, allowing us to:

- Enhance efficiency in order processing, ensuring timely fulfillment of customer orders.
- Streamline inventory management, reducing the likelihood of stockouts or overstock situations.
- Provide customers with a seamless and enjoyable shopping experience by making it easy for them to find, purchase, and review books.

### 3. Requirements Collection and Analysis

#### Fact-Finding Techniques:

- **Interviews:** Discussions with bookstore owners to identify essential features like inventory updates and customer service tools.
- **Questionnaires:** Gathered customer preferences for searching, purchasing, and payment methods.
- **Document Analysis:** Studied features of successful systems like Amazon for inspiration, such as personalized recommendations and effective review systems.

#### Key Insights:

- The system must handle large data volumes.
- It needs to have a user-friendly interface for both customers and administrators.
- It must support advanced search, ordering, and recommendation features.

### 4. Database Design

#### Entities and Attributes:

1. **Customer:** Customer\_ID, name, email, phone number, preferences.
2. **Order:** Order\_ID, order date, total amount, order status, payment method.
3. **Review:** Review\_ID, rating, comment, customer ID, book ID.
4. **Book:** Book\_ID, title, genre, price, stock quantity, ISBN.
5. **Author:** Author\_ID, name, biography, birthdate.
6. **Publisher:** Publisher\_ID, name, contact info.
7. **Order\_Item:** Order\_Item\_ID, quantity, price, order ID, book ID.
8. **Shipment:** Shipment\_ID, tracking ID, shipment date, status, order ID.
9. **Warehouse:** Warehouse\_ID, location, manager, capacity.

#### Relationships:

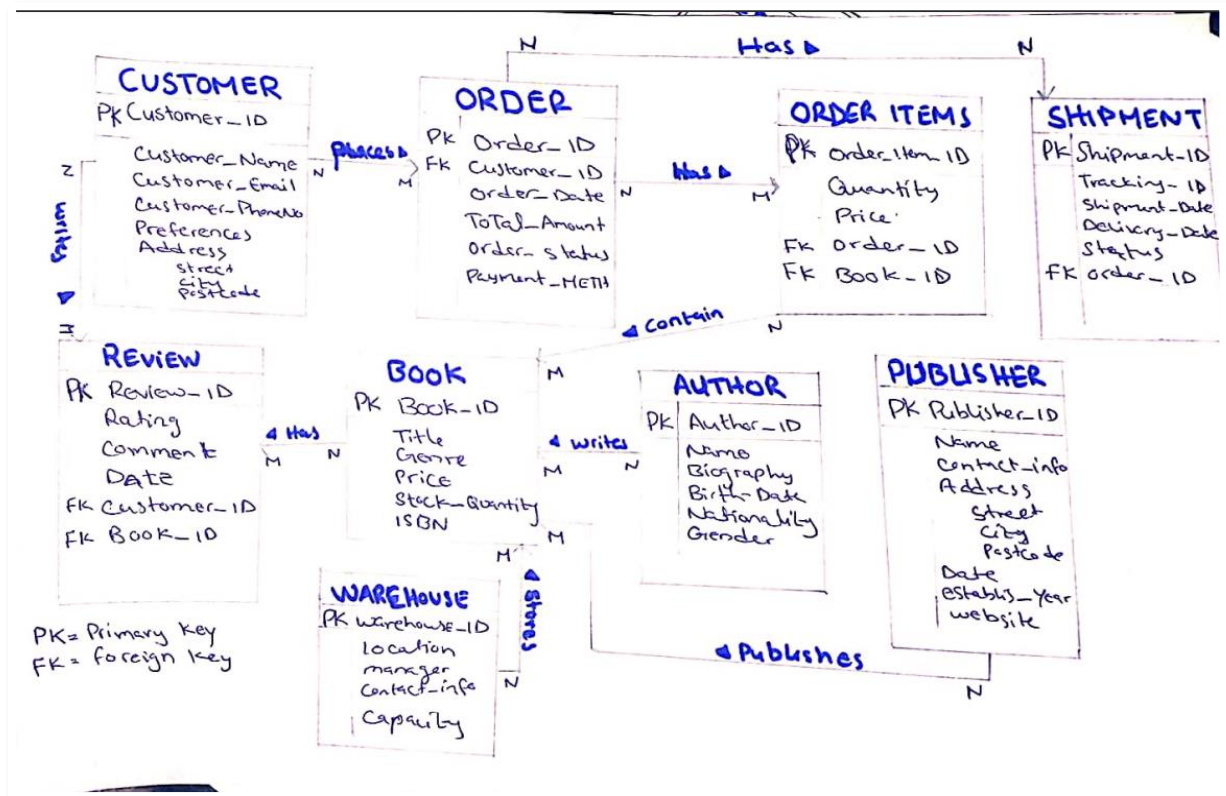
1. **Customer - Order**
  - Relationship Name: *Places\_Order*
  - Degree: 2 (binary relationship)
  - Cardinality: *One-to-Many* (One customer can place multiple orders, but each order is placed by one customer)
2. **Customer - Review**
  - Relationship Name: *Writes\_Review*
  - Degree: 2

- Cardinality: *One-to-Many* (One customer can write multiple reviews, but each review is written by one customer)
- 3. **Order - Order\_Item**
  - Relationship Name: *Has\_Items*
  - Degree: 2
  - Cardinality: *One-to-Many* (One order can have multiple items, but each item belongs to one order)
- 4. **Order\_Item - Book**
  - Relationship Name: *Contains\_Book*
  - Degree: 2
  - Cardinality: *Many-to-One* (One order item represents one book, but each book can appear in multiple order items)
- 5. **Author - Book**
  - Relationship Name: *Writes\_Book*
  - Degree: 2
  - Cardinality: *One-to-Many* (One author can write multiple books, but each book is written by one author)
- 6. **Publisher - Book**
  - Relationship Name: *Publishes\_Book*
  - Degree: 2
  - Cardinality: *One-to-Many* (One publisher can publish multiple books, but each book is published by one publisher)
- 7. **Book - Review**
  - Relationship Name: *Has\_Review*
  - Degree: 2
  - Cardinality: *One-to-Many* (One book can have multiple reviews, but each review is for one specific book)
- 8. **Order - Shipment**
  - Relationship Name: *Has\_Shipment*
  - Degree: 2
  - Cardinality: *One-to-One* (Each order has one shipment associated with it, and each shipment is linked to one order)
- 9. **Warehouse - Book**
  - Relationship Name: *Stores\_Book*
  - Degree: 2
  - Cardinality: *One-to-Many* (One warehouse can store multiple books, but each book is associated with a specific warehouse)

### **ER Diagram Explanation:**

This ER diagram for the Online Bookstore Management System now includes nine main entities: *Customer*, *Order*, *Review*, *Book*, *Author*, *Publisher*, *Order\_Item*, *Shipment*, and *Warehouse*. Each entity has specific attributes with primary keys. Relationships represent interactions such as orders linked to shipments and books stored in warehouses. Cardinality constraints define these relationships, like one-to-many between *Customer* and *Order*, and one-to-one between *Order* and *Shipment*.

## ER Diagram:



## 5. DBMS Selection (MySQL)

### Chosen

MySQL has been selected for the database management system. Its ease of use, reliability, and support for relational databases make it ideal for handling the data complexity of the online bookstore system.

### DBMS:

## 6. Application Design

### User Interface Design:

The application will have distinct interfaces for:

- **Customers:** A shopping cart, personalized book recommendations, and order history.
- **Administrators:** A dashboard for inventory management, order tracking, and customer service.
- **Authors/Publishers:** A portal to view sales data and manage book details.

### Functional Components:

- **Book Catalog:** Browse and filter books.
- **Customer Management:** Manage user profiles and preferences.
- **Order Management:** Order placement, status tracking, and payment processing.

## **7. Prototyping (Optional)**

A prototype will be developed to validate the user interface and functionality before full-scale implementation. This will focus on the book browsing, search features, and order process to gather feedback from a sample of users.

This structure aligns with your project's database lifecycle while addressing all key components like entities, relationships, and DBMS selection.