# DATABASE DESIGN DOCUMENT
# BOOKSTORE MANAGEMENT SYSTEM
# GROUP 18

## 1. Entity Definitions

- **Table: Book –** Keeps track of all the books present in the bookstore.
    - **ISBN**: Unique identifier for each book.
    - **Title**: The title of the book.
    - **Publication Year**: The year when the book was published.
    - **Price**: The selling price of the book.
- **Table: Wrote –** An associative entity that provides relation between an author and the book written by him/her.
    - **Wrote ID:** Composite identifier key made up of Author ID and ISBN.
    - **Author ID:** The identifier of the author who wrote the book.
    - **ISBN:** The ISBN of the book the author wrote.
    - **Role**: The role of the author while writing the book.
- **Table: Author –** Keeps records of all the authors whose books are present in the bookstore.
    - **Author ID**: Unique identifier for each author.
    - **Author Name (Composite)**: The name of the author, having first name and last name of the author.
    - **Nationality (Multi-valued)**: The nationality of the author.
    - **Initials**: The initials of the author.
- **Table: Publisher –** Stores information about each book's publisher.
    - **Publisher ID**: Unique identifier for each publisher.
    - **Publisher Name**: The name of the publisher.
    - **Publisher City**: The city where the publisher is located.
- **Table: Merchandise –** Keeps track of all the merchandise present in the store.
    - **Merchandise ID**: Unique identifier for each merchandise item.
    - **Description**: A brief description of the merchandise item.
    - **Price**: The selling price of the merchandise item.

- **Table: Inventory –** Keeps track of the quantity of the book/merchandise present in the store that is available for purchasing.
    - **Inventory ID**: Unique identifier for each inventory.
    - **Quantity**: The number of copies of the same books/merchandise kept in that inventory.
- **Table: Review –** Store reviews given for a book/merchandise.
    - **Review ID**: Unique identifier for each review.
    - **Content**: The content of the review.
    - **Reviewer**: The author of the review.
    - **Creation Date**: The date when the review was given.
    - **Rating**: The rating of the book/merchandise out of 5.
- **Table: Customer –** Stores information about the customer who visits the store to rent books or purchase books/merchandise.
    - **Customer ID**: Unique identifier for each customer.
    - **Customer Name**: The name of the customer.
    - **Customer Type**: The type (student, parent, regular etc.) of the customer.
    - **Phone Number (Multi-valued)**: The contact number of the customer.
    - **Customer Address (Multi-valued)**: The address of the customer.
- **Table: Employee –** Stores information about the employee who is working in the store and has worked there in the past.
    - **Employee ID**: Unique identifier for each employee.
    - **Name (Composite)**: The name of the employee, having first name and last name of the employee.
    - **Date of birth**: The date of birth of the employee.
    - **Age (Derived)**: The age of the employee which will be derived from the date of birth.
    - **Schedule Time**: The time when the employee works in the store.
    - **Salary**: The current salary of the employee.
    - **Hire Date:** The date when the employee was hired.
    - **Termination Date:** The date when the employee left the job.
- **Table: Order –** Stores the information of an order placed by a customer.
    - **Order ID**: Unique identifier for each order.
    - **Order Amount**: The total amount of the order.
    - **Item Count**: The totals number of items present in the order.

    - **Order Date**: The date when the order was placed.
- **Table: OrderItem –** An associative entity that stores the information of each item present in the order.
    - **OrderItem ID**: Unique identifier for each order item.
    - **Quantity:** The quantity of the specific item ordered.
    - **Price:** The price of the individual item.
- **Table: Promotion –** Keeps track of all the promotions that the store offers.
    - **Promotion ID**: Unique identifier for each promotion.
    - **Code**: The code to avail the promotion.
    - **Description**: The description of the promotion.
    - **Discount Percent**: The discount provided through the promotion.
- **Table: BookRent –** Keeps track of all the books rented by the customers.
    - **BookRent ID**: Unique identifier for each time a book is rented.
    - **Rent Date**: The date when a book is rented by a customer.
    - **Return Date**: The date when the book is returned to the store.
- **Table: Penalty –** Keeps record of all the penalties paid by the customers due to late returns.
    - **Penalty ID**: Unique identifier for each penalty.
    - **Penalty Amount**: The amount of the penalty if the book is returned after its return date.
- **Table: Category –** Stores information about the category to which a book/merchandise belong.
    - **Category ID**: Unique identifier for each category.
    - **Description**: A brief description of the category.

# 2. Relationship Definitions

Below are the relationships along with their cardinalities based on the ERD:

1. **Customer** places an **Order**.
    - Cardinality: One to Many (1:N)
    - A single customer can place multiple orders, but each order is placed by one and only one customer.
2. Each **Order** contains multiple **OrderItem**.
    - Cardinality: One to Many (1:N)

    - A single order can contain multiple order items, but each order item belongs to one and only one order.
3. Each **OrderItem** is associated with a **Book** or a **Merchandise**.
    - Cardinality: One to One (1:1)
    - Each order item corresponds to one book/merchandise, and each book/ merchandise can be in one order item.
4. Each **Book** may be written by one or multiple **Author(s)**.
    - Cardinality: Many to One (N:N)
    - Each book may be written by multiple authors, and each author may write multiple books.
5. Each **Book/Merchandise** may receive a **Review**.
    - Cardinality: Many to One (N:1)
    - Each books/merchandise may receive multiple reviews, but each review is only for one book/merchandise.
6. Each **Book/Merchandise** belongs to a **Category**.
    - Cardinality: Many to One (N:1)
    - Each book/merchandise belongs to a single category, but each category can belong to multiple books/merchandise.
7. Each **Book** is published by a **Publisher**.
    - Cardinality: Many to One (N:1)
    - Multiple books can be published by a single publisher, but each book is published by one and only one publisher.
8. Each **Customer** may issue a **BookRent**.
    - Cardinality: One to Many (1:N)
    - Each customer may rent many books, and each book rent corresponds to one customer.
9. Each **BookRent** may incur a **Penalty**.
    - Cardinality: One to One (1:1)
    - Each late book rent return incurs one penalty, and each penalty corresponds to one late book rent return.
10. Each **PROMOTION** may apply to an **ORDER**.
    - Cardinality: One to Many (1:N)
    - Each promotion may apply to multiple orders, but one order can only apply one promotion.

11. Each **INVENTORY** contains one book/merchandise.
    - Cardinality: One to One (1:1)
    - Each inventory can store one book/merchandise and each book/merchandise must be contained in one inventory.
12. Each **EMPLOYEE** can manage one or multiple **INVENTORY**.
    - Cardinality: One to Many (1:N)
    - Each employee may manage zero or multiple inventories, but each inventory is managed by only one employee.

Each objective relates to the entities and relationships in the following way:

**1. Efficient Resource Management:**

The centralised database system, encompassing tables such as Book, Author, Publisher, Merchandise, Inventory, and Employee, serves as the foundation for efficient resource management. These entities store comprehensive details about books, authors, merchandise, and inventory levels, enabling streamlined operations for store owners and customers. By providing easy access to information, the system facilitates efficient retrieval and management of resources, enhancing overall operational efficiency.

**2. Smooth Borrowing and Returning:**

Through the BookRent and Penalty tables, the system ensures a seamless borrowing and returning process for customers. By managing book borrowing, return dates, and borrower information, the system minimises administrative burden and ensures accurate tracking of member activities. Business logic embedded in the system calculates return dates, manages fines for late returns, and maintains accountability, thereby promoting a smooth and hassle-free borrowing experience for customers.

**3. Categorization of Books and Merchandise:**

Leveraging the Category table and associated relationships, the system enables effective categorisation of books and merchandise. By organising items based on genres, and other relevant criteria, the system enhances accessibility for both store owners and customers. Through intuitive categorisation, users can easily navigate the store's inventory, locate specific items of interest, and make informed purchasing decisions, thereby enhancing the overall shopping experience.

**4. Adherence to Industry Standards:**

The database design adheres to industry best practices, ensuring data integrity, reliability, and scalability. By following standards for data normalisation, the system minimises redundancy and

maintains consistency across the database schema. Implementation of data integrity constraints, validation rules, and proper indexing further enhances data quality and reliability. These measures not only ensure compliance with industry standards but also lay the foundation for robust and future-proof database operations.

**5. Employee Management System:**

For bookstores requiring employee management features, the Employee table provides a comprehensive platform for managing human resources. Attributes such as employee details, schedules, and leave management facilitate efficient workforce management. By integrating employee-related functionalities into the system, bookstores can optimise staffing levels, improve employee productivity, and enhance overall operational efficiency.