Online Book Store

Design an Entity-Relationship schema for an online book publishing and sales platform. The database should contain information about books with title, ISBN, edition, publication year, publisher, genres, and price. Authors have ID, name, biography, and are associated with multiple books.

Customers have customer ID, name, purchase history, shipping addresses, and wish list items. Orders have order number, order date, customer placing the order, list of books ordered with quantity and per item discounts, payment details, and shipment status.

Publishers have names, contact details, and the books they publish. Books can be written by multiple authors and can belong to multiple genres. Customers can place multiple orders, have multiple shipping addresses, and maintain a wishlist of books.

Each edition of a book is published by exactly one publisher, and books can have multiple editions sold in different years. Orders can contain multiple books with different quantities and item-specific discounts. Assume scenarios such as co-authored books, special editions, and pre-order capabilities.

🗒️ SQL Table Creation Statements:

1) Author Table

```
CREATE TABLE Author (
    author_id INT PRIMARY KEY,
    name VARCHAR(100),
    biography TEXT
);
```

2) Publisher Table

```
CREATE TABLE Publisher (
    publisher_id INT PRIMARY KEY,
    name VARCHAR(100),
    contact_details TEXT
);
```

3) Genre Table

```
CREATE TABLE Genre (
    genre_id INT PRIMARY KEY,
    genre_name VARCHAR(100)
);
```

4) Book Table (general book info, not edition-specific)

```
CREATE TABLE Book (
    book_id INT PRIMARY KEY,
    title VARCHAR(200)
);
```

5) BookEdition Table (edition-specific info)

```
CREATE TABLE BookEdition (
    edition_id INT PRIMARY KEY,
    book_id INT,
    ISBN VARCHAR(20) UNIQUE,
    edition_number INT,
    publication_year YEAR,
    publisher_id INT,
    price DECIMAL(10,2),
    FOREIGN KEY (book_id) REFERENCES Book(book_id),
    FOREIGN KEY (publisher_id) REFERENCES Publisher(publisher_id)
);
```

6) BookAuthor (many-to-many between books and authors)

```
CREATE TABLE BookAuthor (
    book_id INT,
    author_id INT,
    PRIMARY KEY (book_id, author_id),
    FOREIGN KEY (book_id) REFERENCES Book(book_id),
    FOREIGN KEY (author_id) REFERENCES Author(author_id)
);
```

7) BookGenre (many-to-many between books and genres)

```
CREATE TABLE BookGenre (
    book_id INT,
    genre_id INT,
    PRIMARY KEY (book_id, genre_id),
    FOREIGN KEY (book_id) REFERENCES Book(book_id),
    FOREIGN KEY (genre_id) REFERENCES Genre(genre_id)
);
```

8) Customer Table

```
CREATE TABLE Customer (
    customer_id INT PRIMARY KEY,
    name VARCHAR(100)
);
```

9) Shipping Address Table (multiple per customer)

```
CREATE TABLE ShippingAddress (
    address_id INT PRIMARY KEY,
    customer_id INT,
    address TEXT,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);
```

10) Wishlist Table

```
CREATE TABLE Wishlist (
    customer_id INT,
    edition_id INT,
    PRIMARY KEY (customer_id, edition_id),
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
    FOREIGN KEY (edition_id) REFERENCES BookEdition(edition_id)
);
```

11) Orders Table

```
CREATE TABLE Orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    order_date DATE,
    payment_details TEXT,
    shipment_status VARCHAR(50),
    address_id INT,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
    FOREIGN KEY (address_id) REFERENCES ShippingAddress(address_id));
```

12) OrderItems Table (books in an order)

```
CREATE TABLE OrderItems (
    order_id INT,
    edition_id INT,
    quantity INT,
    item_discount DECIMAL(5,2),
    PRIMARY KEY (order_id, edition_id),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
    FOREIGN KEY (edition_id) REFERENCES BookEdition(edition_id)
);
```

🧾Values of Each Table :-

1) Author Table

```
INSERT INTO Author (author_id, name, biography) VALUES
(1, 'J.K. Rowling', 'British author, known for the Harry Potter series.'),
(2, 'George R.R. Martin', 'Author of the A Song of Ice and Fire series.');
```

2) Publisher Table

```
INSERT INTO Publisher (publisher_id, name, contact_details) VALUES
(1, 'Bloomsbury Publishing', 'contact@bloomsbury.com, UK'),
(2, 'Bantam Books', 'info@bantambooks.com, USA');
```

3) Genre Table

```
INSERT INTO Genre (genre_id, genre_name) VALUES
(1, 'Fantasy'),
(2, 'Adventure'),
(3, 'Drama');
```

4) Book Table (general book info, not edition-specific)

```
INSERT INTO Book (book_id, title) VALUES
(1, 'Harry Potter and the Philosopher\'s Stone'),
(2, 'A Game of Thrones');
```

5) BookEdition Table (edition-specific info)

```
INSERT INTO BookEdition (edition_id, book_id, ISBN, edition_number,
publication_year, publisher_id, price) VALUES
(1, 1, '9780747532699', 1, 1997, 1, 499.99),
(2, 2, '9780553103540', 1, 1996, 2, 599.99);
```

6) BookAuthor (many-to-many between books and authors)

```
INSERT INTO BookAuthor (book_id, author_id) VALUES
(1, 1),
(2, 2);
```

7) BookGenre (many-to-many between books and genres)

```
INSERT INTO BookGenre (book_id, genre_id) VALUES
(1, 1),
(1, 2),
(2, 1),
(2, 3);
```

8) Customer Table

```
INSERT INTO Customer (customer_id, name) VALUES
(1, 'Alice'),
(2, 'Bob');
```

9) Shipping Address Table (multiple per customer)

```
INSERT INTO ShippingAddress (address_id, customer_id, address)
VALUES
(1, 1, '123 Magic St, London, UK'),
(2, 2, '456 Castle Rd, New York, USA');
```

10) Wishlist Table

```
INSERT INTO Wishlist (customer_id, edition_id) VALUES
(1, 1),
(2, 2);
```

11) Orders Table

```
INSERT INTO Orders (order_id, customer_id, order_date, payment_details,
shipment_status, address_id) VALUES
(101, 1, '2025-06-10', 'Credit Card', 'Shipped', 1),
(102, 2, '2025-06-12', 'PayPal', 'Pending', 2);
```

12) OrderItems Table (books in an order)

```
INSERT INTO OrderItems (order_id, edition_id, quantity, item_discount)
VALUES
(101, 1, 1, 0.00),
(102, 2, 2, 10.00);
```

SYSTEM GENERATED ER DIAGRAM