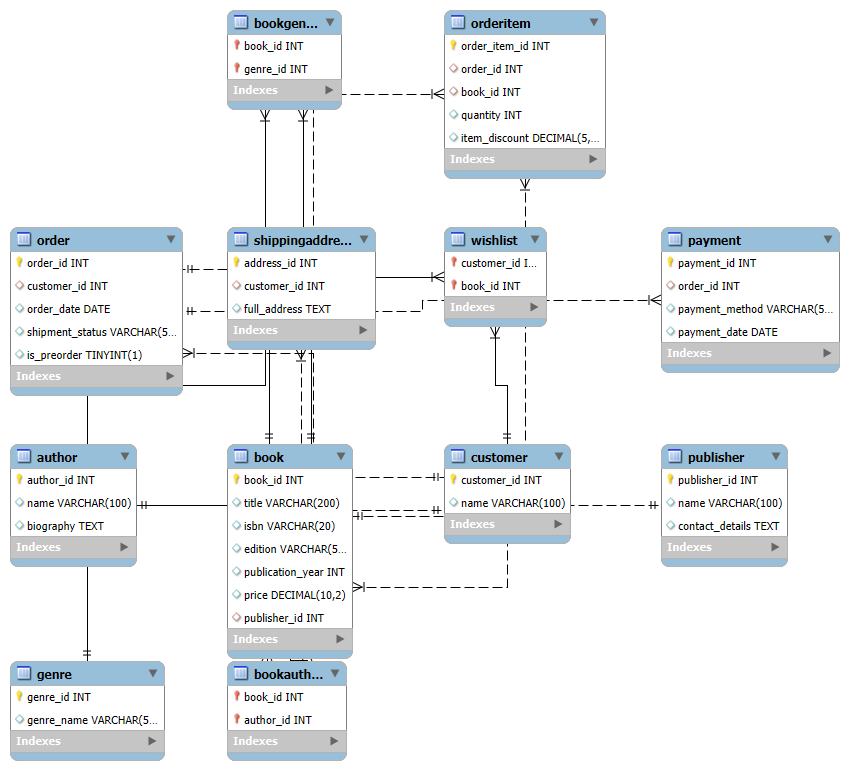
**Problem 3: Online Book Publishing and Sales Platform**

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 wishlist 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.



**-- Author table**

CREATE TABLE Author (

author\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100),

biography TEXT

);

**-- Publisher table**

CREATE TABLE Publisher (

publisher\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100),

contact\_details TEXT

);

**-- Genre table**

CREATE TABLE Genre (

genre\_id INT PRIMARY KEY AUTO\_INCREMENT,

genre\_name VARCHAR(50) UNIQUE

);

**-- Book Edition table (includes ISBN + Edition + Year combo as unique)**

CREATE TABLE Book (

book\_id INT PRIMARY KEY AUTO\_INCREMENT,

title VARCHAR(200),

isbn VARCHAR(20),

edition VARCHAR(50),

publication\_year INT,

price DECIMAL(10, 2),

publisher\_id INT,

FOREIGN KEY (publisher\_id) REFERENCES Publisher(publisher\_id)

);

**-- Book-Author (Many-to-Many)**

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)

);

**-- Book-Genre (Many-to-Many)**

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)

);

**-- Customer table**

CREATE TABLE Customer (

customer\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100)

);

**-- Customer Shipping Addresses (Multiple Addresses)**

CREATE TABLE ShippingAddress (

address\_id INT PRIMARY KEY AUTO\_INCREMENT,

customer\_id INT,

full\_address TEXT,

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id)

);

**-- Wishlist (Many-to-Many between Customer and Book)**

CREATE TABLE Wishlist (

customer\_id INT,

book\_id INT,

PRIMARY KEY (customer\_id, book\_id),

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id),

FOREIGN KEY (book\_id) REFERENCES Book(book\_id)

);

**-- Order table**

CREATE TABLE `Order` (

order\_id INT PRIMARY KEY AUTO\_INCREMENT,

customer\_id INT,

order\_date DATE,

shipment\_status VARCHAR(50),

is\_preorder BOOLEAN DEFAULT FALSE,

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id)

);

**-- Payment details**

CREATE TABLE Payment (

payment\_id INT PRIMARY KEY AUTO\_INCREMENT,

order\_id INT,

payment\_method VARCHAR(50),

payment\_date DATE,

FOREIGN KEY (order\_id) REFERENCES `Order`(order\_id)

);

**-- Order details (books in each order, quantity, per item discount)**

CREATE TABLE OrderItem (

order\_item\_id INT PRIMARY KEY AUTO\_INCREMENT,

order\_id INT,

book\_id INT,

quantity INT,

item\_discount DECIMAL(5,2), -- e.g., 10.00 means 10% discount

FOREIGN KEY (order\_id) REFERENCES `Order`(order\_id),

FOREIGN KEY (book\_id) REFERENCES Book(book\_id)

);

**Inserting Values:**

**-- Author**

INSERT INTO Author (name, biography) VALUES

('J.K. Rowling', 'British author best known for the Harry Potter series.'),

('George R.R. Martin', 'American novelist and short-story writer.'),

('Yuval Noah Harari', 'Historian and author of "Sapiens".');

**-- Publisher**

INSERT INTO Publisher (name, contact\_details) VALUES

('Bloomsbury Publishing', 'London, UK - contact@bloomsbury.com'),

('Penguin Random House', 'New York, USA - info@penguinrandomhouse.com'),

('HarperCollins', 'USA - hello@harpercollins.com');

**-- Genre**

INSERT INTO Genre (genre\_name) VALUES

('Fantasy'),

('History'),

('Science');

**-- Book**

INSERT INTO Book (title, isbn, edition, publication\_year, price, publisher\_id) VALUES

('Harry Potter and the Philosopher\'s Stone', '9780747532743', '1st', 1997, 19.99, 1),

('A Game of Thrones', '9780553103540', '1st', 1996, 24.99, 2),

('Sapiens: A Brief History of Humankind', '9780062316097', '2nd', 2014, 18.50, 3);

**-- BookAuthor**

INSERT INTO BookAuthor VALUES

(1, 1),

(2, 2),

(3, 3);

**-- BookGenre**

INSERT INTO BookGenre VALUES

(1, 1),

(2, 1),

(3, 2),

(3, 3);

**-- Customer**

INSERT INTO Customer (name) VALUES

('Alice Smith'),

('Bob Johnson'),

('Charlie Lee');

**-- ShippingAddress**

INSERT INTO ShippingAddress (customer\_id, full\_address) VALUES

(1, '123 Elm Street, Springfield'),

(1, '456 Oak Avenue, Metropolis'),

(2, '789 Pine Lane, Gotham');

**-- Wishlist**

INSERT INTO Wishlist VALUES

(1, 2),

(1, 3),

(2, 1);

**-- Order**

INSERT INTO `Order` (customer\_id, order\_date, shipment\_status, is\_preorder) VALUES

(1, '2025-06-01', 'Shipped', FALSE),

(2, '2025-06-05', 'Processing', TRUE),

(3, '2025-06-10', 'Delivered', FALSE);

**-- Payment**

INSERT INTO Payment (order\_id, payment\_method, payment\_date) VALUES

(1, 'Credit Card', '2025-06-01'),

(2, 'PayPal', '2025-06-05'),

(3, 'Debit Card', '2025-06-10');

**-- OrderItem**

INSERT INTO OrderItem (order\_id, book\_id, quantity, item\_discount) VALUES

(1, 1, 2, 5.00),

(1, 3, 1, 0.00),

(2, 2, 1, 10.00);

