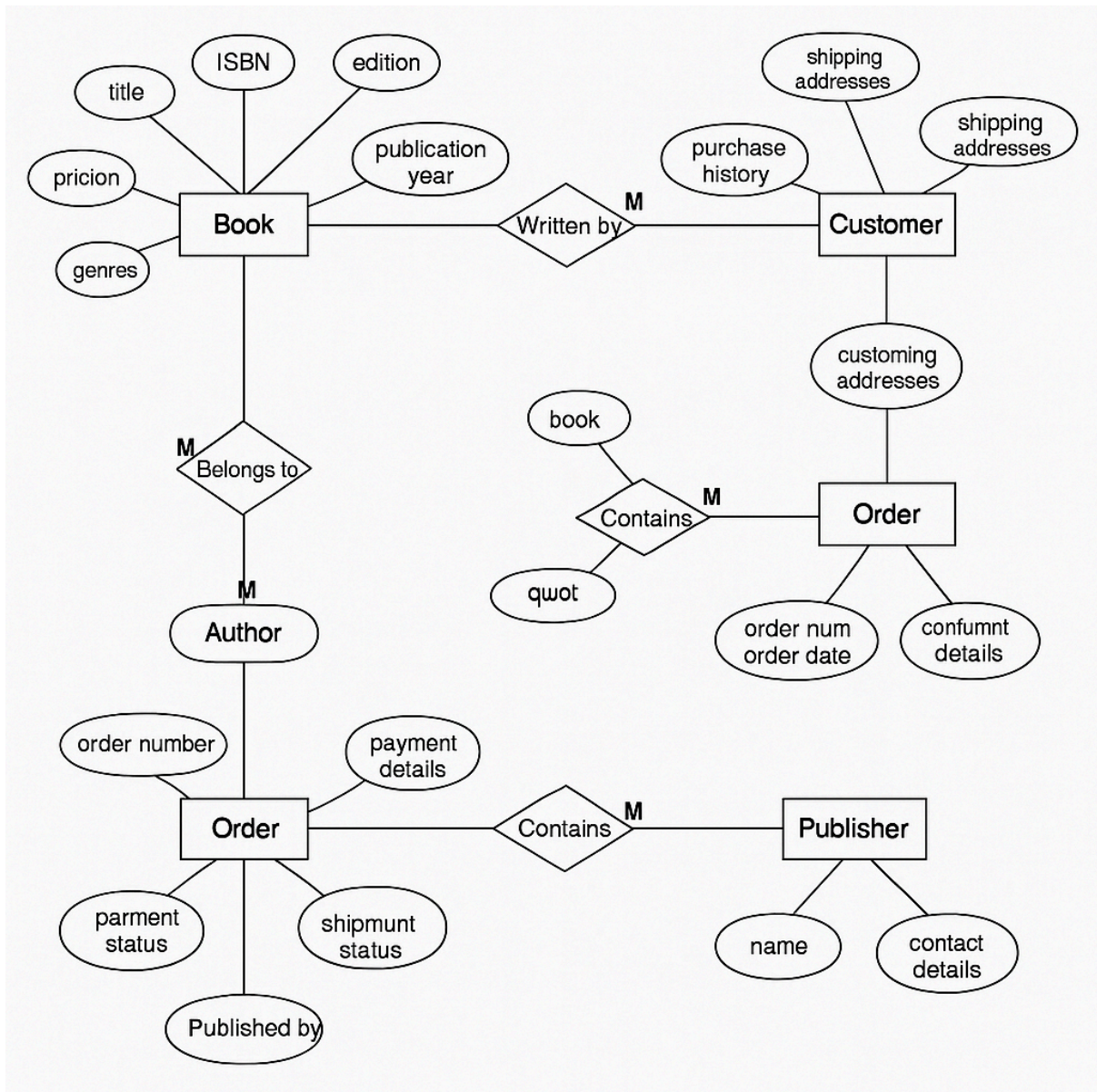




# Online Book Publishing and Sales Platform DataBase

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. similar to this





## SQL Table Creation Statements:

```
CREATE TABLE customer (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(100),  
    email VARCHAR(100),  
    phone VARCHAR(20)  
);
```

```
CREATE TABLE address (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    address_line VARCHAR(255),  
    city VARCHAR(100),  
    state VARCHAR(100),  
    postal_code VARCHAR(20),  
    country VARCHAR(100)  
);
```

```
CREATE TABLE customer_address (  
    customer_id INT,  
    address_id INT,  
    PRIMARY KEY (customer_id, address_id),  
    FOREIGN KEY (customer_id) REFERENCES customer(id) ON DELETE CASCADE,  
    FOREIGN KEY (address_id) REFERENCES address(id) ON DELETE CASCADE  
);
```

```
CREATE TABLE book (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    title VARCHAR(255),  
    isbn VARCHAR(20) UNIQUE  
);
```

```
CREATE TABLE book_edition (  
    id INT PRIMARY KEY,  
    book_id INT,  
    edition_number INT,  
    publication_year YEAR,  
    price DECIMAL(10, 2),  
    publisher_id INT,  
    is_special_edition BOOLEAN DEFAULT FALSE,
```

```
is_preorder_available BOOLEAN DEFAULT FALSE,  
FOREIGN KEY (book_id) REFERENCES book(id) ON DELETE CASCADE,  
FOREIGN KEY (publisher_id) REFERENCES publisher(id) ON DELETE SET NULL  
);
```

```
CREATE TABLE author (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  name VARCHAR(150),  
  biography TEXT  
);
```

```
CREATE TABLE author_book (  
  author_id INT,  
  book_id INT,  
  PRIMARY KEY (author_id, book_id),  
  FOREIGN KEY (author_id) REFERENCES author(id) ON DELETE CASCADE,  
  FOREIGN KEY (book_id) REFERENCES book(id) ON DELETE CASCADE  
);
```

```
CREATE TABLE publisher (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  name VARCHAR(150),  
  contact_details TEXT  
);
```

```
CREATE TABLE orders (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  customer_id INT,  
  order_date DATE,  
  payment_details TEXT,  
  shipment_status VARCHAR(50),  
  FOREIGN KEY (customer_id) REFERENCES customer(id) ON DELETE CASCADE  
);
```

```
CREATE TABLE order_item (  
  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(id) ON DELETE CASCADE,  
  FOREIGN KEY (edition_id) REFERENCES book_edition(id) ON DELETE CASCADE  
);
```

```
CREATE TABLE wishlist (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  customer_id INT,  
  FOREIGN KEY (customer_id) REFERENCES customer(id) ON DELETE CASCADE  
);
```

```
CREATE TABLE wishlist_item (  
  wishlist_id INT,
```

```
book_id INT,  
PRIMARY KEY (wishlist_id, book_id),  
FOREIGN KEY (wishlist_id) REFERENCES wishlist(id) ON DELETE CASCADE,  
FOREIGN KEY (book_id) REFERENCES book(id) ON DELETE CASCADE  
);  
CREATE TABLE book_genre (  
    book_id INT,  
    genre VARCHAR(100),  
    PRIMARY KEY (book_id, genre),  
    FOREIGN KEY (book_id) REFERENCES book(id) ON DELETE CASCADE  
);
```

