

**Task1:** Analyze a given business scenario and create an ER diagram that includes entities, relationships, attributes, and cardinality. Ensure that the diagram reflects proper normalization up to the third normal form.

Creating an ER diagram for an RDBMS (Relational Database Management System) involves defining the entities, attributes, and relationships as database tables with primary and foreign keys. Here's how to structure the tables based on the hypothetical online bookstore scenario:

### Entity-Relationship Diagram (ERD)

Here is the ERD reflecting the entities, attributes, and relationships in the online bookstore scenario.

#### Tables and Relationships

##### 1. Customer

CustomerID(Primary Key)

- Name
- Email
- Address

##### 2. Order

OrderID(Primary Key)

OrderDate

CustomerID(Foreign Key)

##### 3. Book

BookID(Primary Key)

- Title
- Genre
- Price
- Publisher

##### 4. Author

AuthorID (Primary Key)

- Name
- Bio

##### 5. OrderDetail

OrderDetailID (Primary Key)

OrderID(Foreign Key)

BookID (Foreign Key)

- Quantity

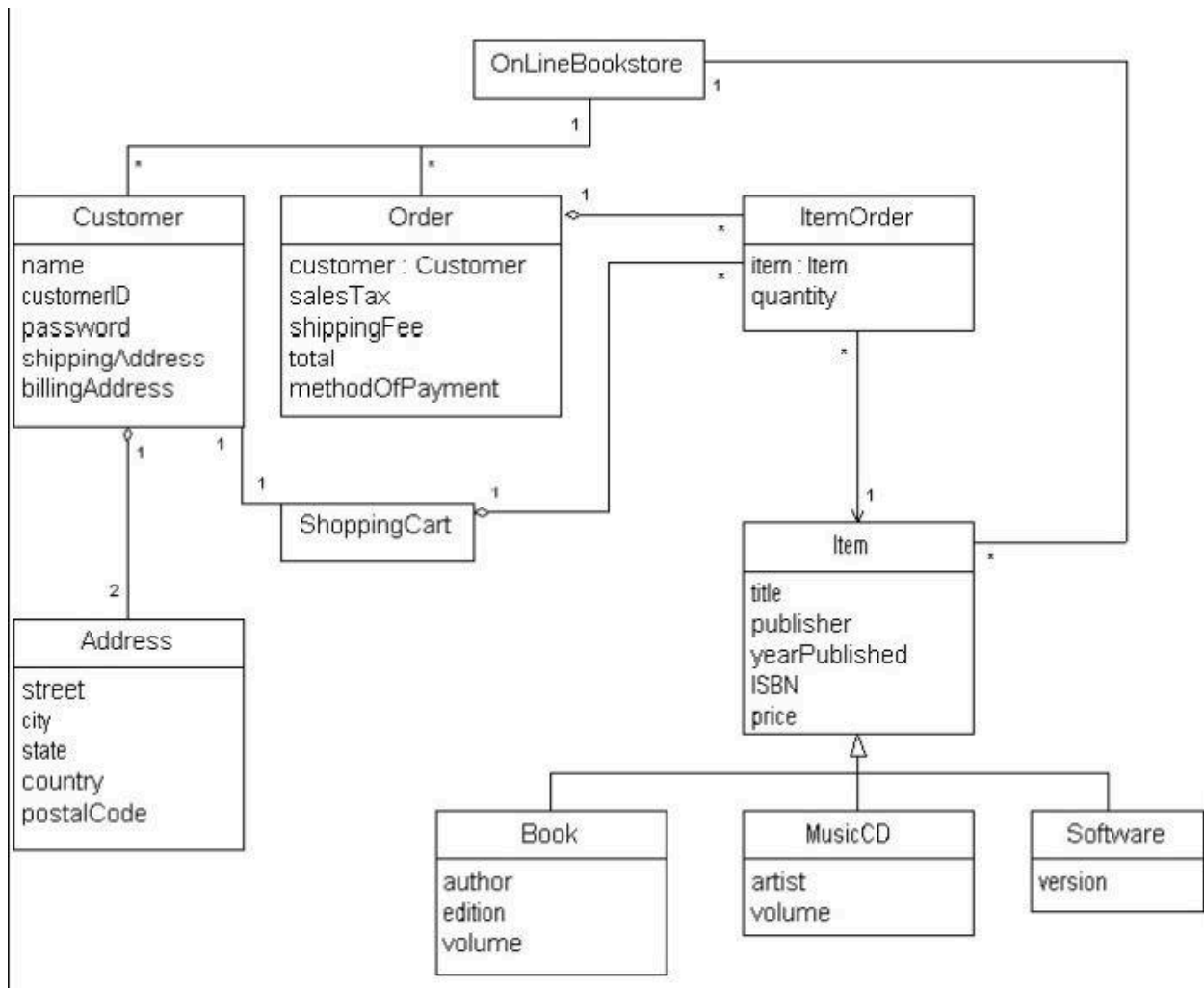
#### 6. BookAuthor

BookAuthorID (Primary Key)

BookID (Foreign Key)

AuthorID (Foreign Key)

#### ER Diagram Representation



#### Relationships and Cardinality

Customer - Order: One-to-Many (One customer can have many orders)

Order - Book: Many-to-Many (via OrderDetail)

Book - Author: Many-to-Many (via BookAuthor)

## SQL DDL Statements

To create these tables and relationships in an RDBMS, you would use SQL Data Definition Language (DDL) statements as follows:

-- Create Customer table

```
CREATE TABLE Customer (  
    CustomerID INT PRIMARY KEY,  
    Name VARCHAR(100),  
    Email VARCHAR(100),  
    Address VARCHAR(255)  
);
```

-- Create Order table

```
CREATE TABLE `Order` (  
    OrderID INT PRIMARY KEY,  
    OrderDate DATE,  
    CustomerID INT,  
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)  
);
```

-- Create Book table

```
CREATE TABLE Book (  
    BookID INT PRIMARY KEY,  
    Title VARCHAR(255),  
    Genre VARCHAR(100),  
    Price DECIMAL(10, 2),  
    Publisher VARCHAR(100)  
);
```

-- Create Author table

```
CREATE TABLE Author (  
    AuthorID INT PRIMARY KEY,  
    Name VARCHAR(100),  
    Bio TEXT  
);
```

-- Create OrderDetail table

```
CREATE TABLE OrderDetail (  
    OrderDetailID INT PRIMARY KEY,  
    OrderID INT,  
    BookID INT,  
    Quantity INT,
```

```
FOREIGN KEY (OrderID) REFERENCES `Order`(OrderID),  
FOREIGN KEY (BookID) REFERENCES Book(BookID)  
);
```

```
-- Create BookAuthor table  
CREATE TABLE BookAuthor (  
    BookAuthorID INT PRIMARY KEY,  
    BookID INT,  
    AuthorID INT,  
    FOREIGN KEY (BookID) REFERENCES Book(BookID),  
    FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID)  
);
```

Explanation of Normalization:

First Normal Form (1NF): All columns contain atomic values; no repeating groups or arrays.

Second Normal Form (2NF): All non-primary key attributes are fully functionally dependent on the primary key. Each entity is its own table with a primary key.

Third Normal Form (3NF): No transitive dependencies; all non-primary key attributes are only dependent on the primary key.

By following these steps, we've ensured the database is normalized up to the third normal form and accurately reflects the given business scenario.