**Interviewee Role**



**Interview Questions and Answers**

Q1: Can you describe the types of items the online bookstore holds?

A1: The bookstore holds a variety of items, including physical books, eBooks, and audiobooks. Each item type has unique attributes such as ISBN for physical books, eBook formats.

Q2: How are new customers registered in the bookstore?

A2: New customers can register by providing their name, email, password, shipping address, and payment details. A unique customer ID is assigned to each new customer, and their registration date is also recorded.

Q3: What information do you track about each book in the store?

A3: For each book, we track the title, author(s), ISBN, price, publisher, publication year, genre, format (hardcover, paperback, eBook, audiobook), availability (in stock or out of stock), and ratings/reviews from customers.

Q4: How do you handle the ordering process?

A4: When a customer places an order, we record the books purchased, the customer’s details, the shipping address, the payment method, the order date, and the expected delivery date. Orders are marked as "pending" until payment is confirmed, after which they are processed for shipping.

Q5: Do you allow customers to pre-order books? How is that managed?

A5: Yes, customers can pre-order books that are not yet released. We track the pre-order details by recording the customer’s ID, book ISBN, and expected release date. When the book is released, the customer is notified, and the order is processed.

Q6: How do you track if a book is out of stock or no longer available?

A6: If a book is out of stock, it is marked as unavailable on the website. When the stock is replenished, the book is made available for purchase again. Books that are no longer available or discontinued are removed from the catalog.

Q7: Is there a limit to the number of books a customer can purchase at one time?

A7: There is no specific limit to the number of books a customer can purchase, but there are guidelines in place for bulk purchases to prevent misuse. However, for pre-orders, there may be limits on the number of units a customer can pre-order for specific high-demand books.

Q8: Can you generate reports from your system? What kinds of reports are needed?

A8: Yes, we generate various reports, including:

* Sales Reports: Monthly, quarterly, and yearly sales performance.
* Inventory Reports: Stock levels, restocks needed, and out-of-stock items.
* Customer Activity Reports: Most active customers, purchase history, and reviews.
* Top-Selling Books: To track the most popular books and make informed inventory decisions.

Q15: Can customers track the status of their orders?

A15: Yes, customers can track the status of their orders through their account on the website. They can view the order's progress, from "processing" to "shipped" and "delivered," and receive notifications when the order is out for delivery. Shipping details and tracking numbers are also provided once the order is dispatched.

Analysis

Entities and Attributes:

Entities:

* Book
* Customer
* Order
* Payment

Attributes for Each Entity:

1. **Book Entity**:
   1. BookID (Primary Key, Integer)
   2. Title (String)
   3. Author (String)
   4. Price (Decimal)
2. **Customer Entity**:
   1. CustomerID (Primary Key, Integer)
   2. Name (String)
   3. Email (String)
   4. ShippingAddress (Text)
3. **Order Entity**:
   1. OrderID (Primary Key, Integer)
   2. OrderDate (Date)
   3. TotalPrice (Decimal)
   4. CustomerID (Foreign Key, Integer)
4. **Payment Entity**:
   1. PaymentID (Primary Key, Integer)
   2. OrderID (Foreign Key, Integer)
   3. PaymentDate (Date)
   4. PaymentAmount (Decimal)

Relationships:

* Book ↔ Order: A Book can be part of many Orders, and an Order can contain many Books (Many-to-Many).
* Customer ↔ Order: A Customer can place many Orders, but each Order is placed by exactly one Customer (One-to-Many).
* Order ↔ Payment: Each Order can have one or more Payments, and each Payment is for exactly one Order (One-to-Many).

**Requirements for the Online Bookstore:**

* **Book**: The bookstore sells books. Each book will have details like **title**, **author(s)**, **ISBN number**, and **price**.
* **Customer**: Customers purchase books. Each customer has details like **name**, **email**, **shipping address**, and **membership type** (e.g., regular, premium).
* **Order**: An order contains information about the **book(s)** a customer’s purchases, including the **order date**, **status** (e.g., shipped, pending), and **total price**.
* **Payment**: Payments are made for orders. A payment has details such as **payment ID**, **order ID**, **payment method** (e.g., credit card, PayPal), and **payment date**.

**How to determine the requirements**:

* I interview a worker from the online store and selected entities based on the core operations of an Online Bookstore: managing books, customers, orders, and payments.
* Identifying the core attributes that describe each entity will help us organize the data and relationships.

**Cardinality and Participation Constraint:**

**Book ↔ Order**: Many-to-many (M:N), with the **Order\_Book** junction table.

* **Partial participation** for **Books** (a book may or may not be part of an order).
* **Total participation** for **Orders** (every order must have at least one book).

**Customer ↔ Order**: One-to-many (1:N).

* **Total participation** for **Orders** (every order must be linked to a customer).
* **Partial participation** for **Customers** (not all customers may place orders).

**Order ↔ Payment**: One-to-many (1:N).

* **Total participation** for **Orders** (every order must have a payment).
* **Partial participation** for **Payments** (a payment might not be made for an order immediately)

**Relationships and Cardinality:**

The relationships define how the entities interact with each other, and cardinality helps to specify the nature of the relationship.

**Book ↔ Order (Many-to-Many Relationship):**

* **Explanation**: A book can appear in many orders, and each order can contain many books.
  + For example, a customer can buy multiple books in a single order, and a specific book can be ordered by multiple customers in different orders.
* **Junction Table**: You need a **junction table** to manage this many-to-many relationship. This table is often called **Order\_Book** and will include **BookID** and **OrderID** as foreign keys.
  + **Cardinality**: Each book can be part of multiple orders, and each order can contain multiple books. The cardinality is **M:N** (many-to-many).