**STUDENT NAME: HOÀNG VĂN MẠNH**

**STUDENT ID: ITDSIU21099**

**Principles of Database Management**

**HOMEWORK: ERD**

**Case Study: Online Bookstore**

An online bookstore wants to develop a new system to manage their inventory and customer orders. The system should allow customers to browse and search for books, add books to their shopping cart, and checkout using a credit card. The system should also allow the bookstore staff to manage their inventory, process orders, and generate reports.

Here are the requirements for the system:

1. Customers can browse and search for books by title, author, or category.

2. Customers can add books to their shopping cart, remove books from their shopping cart, and view the contents of their shopping cart.

3. Customers can checkout using a credit card. The system should validate the credit card information and generate an order confirmation.

4. The system should track the inventory of books. Each book has a unique ISBN number, title, author, category, price, and quantity in stock.

5. The system should allow bookstore staff to add new books to the inventory, update the quantity of books in stock, and remove books from the inventory.

6. The system should allow bookstore staff to process customer orders. Each order has a unique order number, date and time of order, customer information (name, email, phone number, shipping address), and a list of books ordered with their quantities and prices.

7. The system should generate reports on book sales, inventory levels, and customer orders.

Using these requirements, you can draw an ERD model for the online bookstore system. Your ERD should include entities, attributes, relationships, and cardinalities.

***List of entities and its attributes:***

1. **Customers**: This entity stores information about the customers such as name, email, and phone number….

* **Attributes of Customers**:

Customers\_ID (Primary Key),

Name,

Phone Number,

Email,

Shipping address,

Credit Card Number (Foreign Key).

1. **Credit Card:** This entity stores information about the credit card used to pay for the order, such as the credit card number, expiry date, and security code.

* **Attributes of Credit Card:**

CreditCard Number (Primary Key),

Expiry Date,

Security Code.

1. **Books**: Representing books have in inventory. Each book has a unique ISBN number, title, author, category, price, and quantity in stock.

* **Attributes of Books**:

ISBN (Primary Key),

Title,

Author,

Category (Foreign Key),

Price,

Quantity in stock.

1. **Category**: This entity stores information about the category of book

* **Attributes of Category**:

Category Name (Primary Key),

Book Title.

1. **Orders:** This entity store information about orders

* Attributes of Orders:

**OrderNumber** (Primary key),

Date and Time of Order,

CustomerID. (Foreign Key)

1. **Shopping Cart:** Representing all the critical information about orders like price, list of books, quantities.

* **Attributes of Shopping Cart:**

ShoppingCart\_ID (Primary Key)

List\_of\_Books,

Quantities,

Price.

1. **Staff:** This entity stores information about the bookstore staff

* **Attributes of Staff**:

Staff\_ID (Primary Key),

Staff\_Name,

Password.

***Relationship: an association among entities***

1. A customer can place zero or many orders.
2. An order must be placed by one or only one customer.
3. A shopping cart can have one or more books, and one book can be stored in many shopping carts.
4. An order must have one and only one credit card used for payment.
5. A credit card can be used for zero or many orders.
6. A customer can have one or many credit cards.
7. One category of book can have many books, but one book has only one category.
8. One staff can Update or Delete one or many book in inventory.

***Some cardinalities in this system:***

1. The Category entity has a one-to-many relationship with the Books entity: This means that one category of book can have many books, but one book has only one category.
2. The Books entity has a many-to-many relationship with Shopping Cart entity: multiple books can be added or removed in multiple shopping carts, and vice versa.
3. The Books entity has a many-to-many relationship with Orders entity: this means that multiple books can be associated with multiple orders, and vice versa.
4. The Orders entity has One-to-One relationship with Credit Card entity: this means that one Credit Card can only pay for one order and vice versa
5. The Credit Card entity has a One-to Many relationship with Orders entity: this means that One credit card can be paid for many orders.
6. The Customers entity has a one-to-many relationship with Orders entity: this means that One customer can have a lot of orders but one orders only has one customers.
7. The Customers entity has a one-to-many relationship with Credit Card entity: this means that One customer can have many credit cards, but one credit card can only be used by one customer.
8. The Staff entity has a one-to-many relationship with the Books entity: this means that one staff can update or delete many books, but each update is only updated by one staff.

***ERD model***

***Diagram

Description automatically generated***