# Project Title: Online Bookstore Database

ماهي المتطلبات؟

* إدارة المستخدمين
  + العمليات
    - انشاء حساب
    - الحذف
    - التعديل
    - البحث
    - العرض
    - تسجيل الدخول
  + التقارير
    - المبالغ التي تم استلامها
    - الطلبات التي قام بانشائها
* إدارة الأصناف
  + العمليات
    - الإضافة
    - الحذف
    - التعديل
    - البحث
    - العرض
  + التقارير
    - الكتب التابعة لكل صنف
    - عدد الكتب لكل صنف
* إدارة العملاء
  + العمليات
    - الإضافة
    - الحذف
    - التعديل
    - البحث
    - العرض
  + التقارير
    - جلب الطلبات التي قدمها العميل
    - الطلبات وتفاصيل المنتجات لكل عميل
    - الطلبات المدفوعة والتي لم تدفع
* إدارة الكتب
  + العمليات
    - الإضافة
    - الحذف
    - التعديل
    - البحث
    - العرض
  + التقارير
    - معلومات الكتب والمؤلفين والتصنيف لكل كتاب
* إدارة المؤلفين
  + العمليات
    - الإضافة
    - التعديل
    - االحذف
    - العرض
    - البحث
  + التقارير
    - الكتب التي ألفها
* إدارة الطلبات
* إدارة المدفوعات

Description:

The Online Bookstore Database is designed to support the operations and management of an online bookstore. It stores information about **books**, **authors**, **genres**, **customers**, **orders**, and **order details**. The database enables the efficient management of book inventory, customer orders, and facilitates seamless online transactions.

Requirements:

1. **Books**:

- Each book should have a unique BookID, along with attributes such as Title, ISBN, Publication Date, Price, and the corresponding AuthorID.

- Books should be associated with one or more genres to facilitate genre-based browsing and searching.

1. **Authors**:

- Each author should have a unique AuthorID, along with attributes like AuthorName, BirthDate, Nationality, and Biography.

- Authors can be associated with multiple books.

1. **Genres**:

- Each genre should have a unique GenreID, along with attributes such as GenreName and Description.

- Genres allow for categorizing and classifying books based on their themes or subject matter.

1. **Customers**:

- Each customer should have a unique CustomerID, along with attributes like FirstName, LastName, Email, Address, and Phone.

- Customers should be able to register, update their information, and place orders.

1. **Orders**:

- Each order should have a unique OrderID, along with attributes like CustomerID, OrderDate, and TotalAmount.

- Orders represent individual transactions made by customers and should be associated with the respective customer who placed the order.

1. **Order Details**:

- Each order can have multiple order details, representing individual items within the order.

- Order details should include attributes such as OrderDetailID, BookID, Quantity, and UnitPrice.

- The BookID in the order details table should reference the BookID in the Books table.

Tasks:

Create a Database Called **OnlineBookstore**

Create the tables needed and the relationships with respect to naming conventions, indexes, names of constraints, stored procedures, views, functions, and so on.

1. Querying and Reporting:

1. Create the necessary tables with appropriate columns and define the primary and foreign keys.

1. Populate the tables with sample data.

1. Write SQL queries to retrieve information such as:

- Get all books written by a specific author.

- Get the total number of orders placed by each customer.

- Get the top-selling genres.

- Get the average price of books in each genre.

- Get the list of customers who have purchased a particular book.

1. Implement additional features such as inserting new records, updating existing records, or deleting records from the tables.

- The database should support efficient querying to retrieve information such as book details, customer orders, sales reports, and popular genres.

- Queries should allow for searching books by title, author, genre, and other relevant criteria.

1. Security and Access Control:

- Implement appropriate security measures to protect sensitive customer information and prevent unauthorized access to the database.

- Establish user roles and access control mechanisms to restrict access to certain functionalities and data based on user privileges.

1. Scalability and Performance:

- The database should be designed and optimized to handle a large number of books, authors, customers, and orders.

- Indexing and performance tuning techniques should be employed to ensure efficient query execution and response times.

These requirements provide a foundation for designing and implementing the Online Bookstore Database. However, it's important to note that specific requirements may vary based on the unique needs and goals of the online bookstore.

Design and implement a database for an online bookstore. The database should include the following tables:

1. Authors : Stores information about authors.

- Columns: AuthorID (Primary Key), AuthorName, BirthDate, Nationality, Biography, etc.

1. Books : Stores information about books.

- Columns: BookID (Primary Key), Title, ISBN, PublicationDate, Price, AuthorID (Foreign Key), etc.

1. Genres : Stores information about book genres.

- Columns: GenreID (Primary Key), GenreName, Description, etc.

1. BookGenres : A junction table that establishes a many-to-many relationship between books and genres.

- Columns: BookID (Foreign Key), GenreID (Foreign Key)

1. Customers : Stores information about customers.

- Columns: CustomerID (Primary Key), FirstName, LastName, Email, Address, Phone, etc.

1. Orders : Stores information about customer orders.

- Columns: OrderID (Primary Key), CustomerID (Foreign Key), OrderDate, TotalAmount, etc.

1. OrderDetails : Stores information about individual items within an order.

- Columns: OrderDetailID (Primary Key), OrderID (Foreign Key), BookID (Foreign Key), Quantity, UnitPrice, etc.

This project allows you to practice creating tables, establishing relationships (including many-to-many relationships), and querying the database using SQL statements. You can also implement additional features such as tracking customer reviews, managing inventory, or generating sales reports.

Here are a few tasks you can work on:

Remember to plan and design the database structure before starting the implementation. It's a good practice to normalize the tables and ensure data integrity through proper relationships and constraints.

Feel free to tailor the project according to your interests or requirements. Happy practicing!



