Assignment: Advanced Online Bookstore Database Design

Introduction

You are tasked with creating a comprehensive database design for an online bookstore. The objective is to develop a schema that not only tracks books, customers, orders, and payments but also incorporates advanced elements like reviews, publisher information, and analytics capabilities. The assignment will challenge your skills in schema design, normalization, query writing, and understanding of database management and security principles.

Database Design

Books Table

- `BookID` (Primary Key): Unique identifier for each book.
- `ISBN`: International Standard Book Number.
- `Title`: Book title.
- `AuthorID` (Foreign Key): Identifier for the author.
- `GenreID` (Foreign Key): Identifier for the book genre.
- `PublisherID` (Foreign Key): Identifier for the publisher.
- `PublicationDate`: Date of publication.
- `Price`: Book price.
- `StockQuantity`: Available stock.

Authors Table

- `AuthorID` (Primary Key): Unique identifier for each author.
- `FirstName`: Author's first name.
- `LastName`: Author's last name.

Genres Table

- `GenreID` (Primary Key): Unique identifier for each genre.
- `GenreName`: Name of the genre.

Publishers Table

- `PublisherID` (Primary Key): Unique identifier for each publisher.
- `PublisherName`: Name of the publisher.
- `Address`: Publisher's address.

Customers Table

- `CustomerID` (Primary Key): Unique identifier for each customer.
- `FirstName`: First name.
- `LastName`: Last name.

- `Email`: Email address.
- `Address`: Physical address.

Orders Table

- `OrderID` (Primary Key): Unique identifier for each order.
- `CustomerID` (Foreign Key): Customer who placed the order.
- `OrderDate`: Date when the order was placed.
- `TotalAmount`: Total order amount.

OrderDetails Table

- `OrderDetailID` (Primary Key): Unique identifier for each order detail.
- `OrderID` (Foreign Key): Associated order.
- `BookID` (Foreign Key): Ordered book.
- `Quantity`: Number of copies ordered.
- `Price`: Price at the time of order.

Payments Table

- `PaymentID` (Primary Key): Unique identifier for each payment.
- `OrderID` (Foreign Key): Associated order.
- `PaymentDate`: Payment date.
- `Amount`: Amount paid.
- `PaymentMethod`: Method of payment (e.g., credit card, PayPal).

Reviews Table

- `ReviewID` (Primary Key): Unique identifier for each review.
- `BookID` (Foreign Key): Reviewed book.
- `CustomerID` (Foreign Key): Reviewing customers.
- `Rating`: Rating given.
- `Comment`: Review text.
- `ReviewDate`: Date of review.

Assignment Questions

Schema Design and ER Diagram:

- Create an ER diagram reflecting the above schema.
- Write SQL statements for creating the tables with appropriate data types and constraints, including foreign key relationships.

Data Manipulation and Querying:

- Insert sample data into each table using SQL insert statements.
- Write a query to select all books by a particular author and sort them by price in descending order.
- List all orders made by a particular customer, showing the order and total amount.

Advanced Querying and Analytics:

- Develop a query to identify the top 3 selling books in each genre monthly, using window functions for ranking.
- Construct a query to find customers with consistent monthly purchases over the last year.
- Create a query to detect "hidden gems" in the inventory: books with high ratings but low sales.Assuming average ratings above a certain threshold (e.g. 4.5) are considered high and sales below a certain number (e.g.100 units) are considered low.
- Write a query to identify books often bought together, indicating potential cross-selling opportunities.
- Create a query to analyze the total sales revenue generated by each author's books.
- Develop a query to analyze the trends in publishing over the years by counting the number of books published by each publisher per year. This can reveal how active different publishers are and how publishing trends change over time.
- Create a query to calculate the market share of each publisher based on the total sales of their books as a percentage of the total sales in the bookstore. This will give insight into which publishers dominate the market.
- Develop a query to list books that have a stock quantity below the average stock level of all books but have been sold more than 50 times. This query helps in identifying books that are high in demand but low in stock.

Normalization and Data Integrity:

- Discuss the normalization of the database design, identifying the normal form achieved for each table.
- Propose improvements or adjustments to the schema for higher normalization and better data integrity.

Security and Transaction Management:

- Suggest roles and permissions that should be defined for managing the bookstore database.
- Describe how transaction management can be implemented to ensure consistency and reliability in order processing and payment handling.

Performance and Scalability:

- Discuss indexing strategies for the tables to improve query performance.
- Suggest methods for scaling the database as the number of books, orders, and customers grows.

Deliverables

- A comprehensive document containing the ER diagram.
- SQL creation statements for the database schema.
- SQL queries for data manipulation and advanced analytics.