SQL Server

DDL Statements

1. Create a database name “BookStoreDB” and create following tables in the database with the below mentioned columns using SQL Server.

    I. Author – AuthorId, AuthorName, DateOfBirth, State, City, Phone.
    II. Publisher – PublisherId, PublisherName, DateOfBirth, State, City, Phone.
    III. Category – CategoryId, CategoryName, Description.
    IV. Book – BookId, Title, Description, Price, ISBN, PublicationDate, Image.
    V. Order – OrderId, Date, Quantity, UnitPrice, ShippingAddress.

2. Modify the BookStoreDB and add all the required keys. Establish relationships among tables and apply following business rules.

    I. A Book can have multiple authors.
    II. An Author can write more than one book.
    III. A Book belongs to only one category.
    IV. A Book can be published by only one publishing house.
    V. An order can be placed for a single book but multiple quantities.

SELECT Statements

3. Write the appropriate SQL queries against BookStoreDB database to support following operations:

    a. Get All the books written by specific author
    b. Get all the books written by specific author and published by specific publisher belonging to “Technical” book Category
    c. Get total books published by each publisher.
    d. Get all the books for which the orders are placed.

Stored Procedures

4. Write the following stored procedure using SQL Server in BookStoreDB database to support following operations:

    a. Get All the books written by specific author
    b. Get all the books written by specific author and published by specific publisher belonging to “Technical” book Category
    c. Get total books published by each publisher.

d. Insert a particular book
e. Update a particular book by id
f. Delete a particular book by id

Triggers

5. Let's assume that we have a table name "Book_History" table. If a particular book is deleted from the "Book" table, an entry with same book records to "Book_History" table must take place. Automate this process using trigger.

6. The "Book" table got an attribute "Price". Let's assume that we have a business requirement where we must ensure that the "Price" should not be less than 1. If any insert or update statement tries to make the "Price" less than 1, the SQL Server must terminate such insert or update statements. Write an appropriate trigger to implement the business requirement.

7. Create a trigger on the table "Order" and add the following functionalities. When a new order is placed, it should check whether the required quantity is available in the "Book" table. If not, it should show appropriate message and the insert statement to "Order" table should be terminated. If the quantity in book table is sufficient, it should deduct the quantity ordered from the quantity in hand in the book table and update the quantity.