2.A

1. **Books Table**:
   * Fields: BookID (Primary Key), Title, Author, Genre, ISBN, PublicationYear, AvailableCopies
   * Constraints:
     + NOT NULL: All fields except AvailableCopies.
     + UNIQUE: ISBN (to ensure each book has a unique identifier).
2. **Authors Table**:
   * Fields: AuthorID (Primary Key), AuthorName, BirthYear, Nationality
   * Constraints:
     + NOT NULL: All fields.
3. **Customers Table**:
   * Fields: CustomerID (Primary Key), FirstName, LastName, Email, Phone, Address
   * Constraints:
     + NOT NULL: All fields.
4. **Borrowed Books Table**:
   * Fields: BorrowID (Primary Key), CustomerID (Foreign Key), BookID (Foreign Key), BorrowDate, ReturnDate
   * Constraints:
     + NOT NULL: BorrowID, CustomerID, BookID, BorrowDate.
     + CHECK: ReturnDate must be after BorrowDate.
5. **Genres Table**:
   * Fields: GenreID (Primary Key), GenreName
   * Constraints:
     + NOT NULL: All fields.
6. **Book Authors Table** (Many-to-Many Relationship):
   * Fields: BookID (Foreign Key), AuthorID (Foreign Key)
   * Constraints:
     + NOT NULL: Both fields.

**Relationships:**

* Books have a many-to-many relationship with Authors (via BookAuthors table).
* Customers borrow books (via BorrowedBooks table).

**Notes:**

* Primary keys are denoted by (Primary Key).
* Foreign keys are denoted by (Foreign Key).

3.A

1. Atomicity: This property ensures that a transaction is treated as a single unit of work. It means that either all operations within the transaction are successfully completed and committed to the database, or none of them are. If any part of the transaction fails, the entire transaction is rolled back, and the database remains unchanged.

2. Consistency: Consistency ensures that the database remains in a valid state before and after the transaction. It means that transactions should preserve the integrity constraints, data validations, and invariants defined in the database schema. Even though individual transactions might change the database state, the overall consistency of the data is maintained.

3. Isolation: Isolation ensures that the execution of concurrent transactions does not interfere with each other. Each transaction should operate independently of other transactions, as if it were the only transaction running on the system. Isolation prevents issues such as dirty reads, non-repeatable reads, and phantom reads, which can occur when multiple transactions access and modify the same data simultaneously.

4. Durability: Durability guarantees that once a transaction is committed, its changes are permanently saved in the database and are not lost, even in the event of a system failure. The changes made by a committed transaction should persist and be recoverable, even if the system crashes or loses power.

4.A

CREATE DATABASE my\_library\_db;

CREATE TABLE Books (

book\_id INT PRIMARY KEY,

title VARCHAR(255),

author\_id INT,

publication\_year INT,

genre VARCHAR(50)

);

ALTER TABLE Books

ADD COLUMN isbn VARCHAR(20);

DROP TABLE RedundantTable;

5.A