**Case Study: Library Management System**

In this case study, you will work with a database schema designed for managing a library. The schema consists of six tables: Books, Members, Loans, Authors, BookAuthors, and Fines. Your task is to create these tables, insert records, and perform various SQL queries to manage and analyze library operations, member activities, book loans, and fines.

**Database Schema**

1. **Books**
   * BookID (INT, Primary Key)
   * Title (VARCHAR)
   * Author (VARCHAR)
   * PublicationYear (YEAR)
   * Genre (VARCHAR)
2. **Members**
   * MemberID (INT, Primary Key)
   * FirstName (VARCHAR)
   * LastName (VARCHAR)
   * Email (VARCHAR)
   * MembershipDate (DATE)
3. **Loans**
   * LoanID (INT, Primary Key)
   * **BookID** (INT, Foreign Key referencing Books.BookID)
   * **MemberID** (INT, Foreign Key referencing Members.MemberID)
   * LoanDate (DATE)
   * ReturnDate (DATE)
4. **Authors**
   * AuthorID (INT, Primary Key)
   * AuthorName (VARCHAR)
   * BirthYear (YEAR)
5. **BookAuthors**
   * **BookID** (INT, Foreign Key referencing Books.BookID)
   * **AuthorID** (INT, Foreign Key referencing Authors.AuthorID)
6. **Fines**
   * FineID (INT, Primary Key)
   * **LoanID** (INT, Foreign Key referencing Loans.LoanID)
   * FineAmount (DECIMAL)
   * PaidDate (DATE)

**Questions:**

1. **Create Table:**
   * Write an SQL statement to create all tables with the specified columns.
2. **Insert Records:**
   * Insert at least 10 records in all the tables.
3. **Select Records:**
   * Write a query to select all books published before 2000 from the Books table.
4. **Where Clause (AND/OR):**
   * Write a query to select all Loans where the LoanDate is in 2024 and the ReturnDate is NULL.
5. **LIKE Operator:**
   * Write a query to select all Books where the Title contains 'Science'.
6. **CASE Statement:**
   * Write a query to select Title and a new column Availability from the Books table. If a book has been loaned out (i.e., exists in Loans table with a NULL ReturnDate), set Availability to 'Checked Out', otherwise 'Available'.
7. **Subquery:**
   * Write a query to find all Members who have borrowed more than 5 books. Use a subquery to find these MemberIDs.
8. **Group By:**
   * Write a query to get the total number of books borrowed by each Member. Group the results by MemberID.
9. **Having Clause:**
   * Write a query to get the total FineAmount collected for each LoanID, but only include loans where the total fine amount is greater than $10. Use the HAVING clause.
10. **Limit:**
    * Write a query to select the top 5 most frequently borrowed books.
11. **Inner Join:**
    * Write a query to join Loans with Books to get a list of all loans with Title, LoanDate, and ReturnDate.
12. **Outer Join:**
    * Write a query to get a list of all Books and any associated loans. Include books that might not be currently borrowed.
13. **Join with Aggregation:**
    * Write a query to get the total number of books each Author has written. Use an INNER JOIN between Books and BookAuthors, and group by AuthorID.
14. **Subquery with Join:**
    * Write a query to find all Books that were written by authors born after 1970. Use a subquery in the WHERE clause to find these AuthorIDs.
15. **Advanced Join:**
    * Write a query to list Title, AuthorName, and FineAmount for all books where a fine has been recorded. Use INNER JOIN and LEFT JOIN as necessary to get all required details.