

1. Database Creation:

The project begins by creating a database called "LMS" with appropriate file settings and initial table allocation.

2. Table Creation:

Tables are designed to store data for books, authors, books-authors relationships, members, loans, and feedback. Primary keys, foreign keys, and constraints are defined for data integrity.

3. Alter, Drop, and Modify Tables & Columns:

The project demonstrates altering, dropping, and modifying table columns to adapt to changing requirements.

4. Create Clustered and Non-clustered Index:

Clustered and non-clustered indexes are implemented to improve query performance.

5. Create Sequence & Alter Sequence:

Sequences are utilized to generate unique identifiers for table records, and sequence properties are altered for specific use cases.

6. Create a View & Alter View:

Views are created to simplify data access and present a subset of data from the Books table

7. Create Stored Procedures:

Stored procedures are implemented for inserting, updating, and deleting data from the Books table using parameters.

8. Create Functions:

Scalar-valued and table-valued functions are created to perform computations and return specific results.

9. Create Triggers:

Triggers are defined for the Loans table to restrict updates and deletions based on specified conditions.

10. Data Insertion:

Data is inserted into the Books, Authors, BooksAuthors, Members, Loans, and feedback tables to populate the LMS database.

11. Data Retrieval:

Various SQL queries are designed to retrieve data from different tables using SELECT, INNER JOIN, LEFT OUTER JOIN, CROSS JOIN, SUBQUERIES, EXISTS, CTE, FETCH-OFFSET, ROLLUP, CUBE, IIF, CASE, CHOOSE, COALESCE, ISNULL, ANY, SOME, and ALL clauses.

12. Miscellaneous Operations:

The case study includes additional functionalities such as the use of the WAITFOR clause, examining object details with SP\_HELPTEXT, and implementing RANK functions.

This project demonstrates the design and implementation of a robust Library Management System using SQL Server. The database schema is designed for efficient data management and retrieval, and various SQL queries are optimized for faster performance. The system allows for easy management of books, authors, members, loans, and feedback, making it a valuable tool for any library organization