LIBRARY MANAGEMENT SYSTEM DATABASE

OVERVIEW:

The Library Management System Database empowers library administrators to effectively manage their library resources, make data-driven decisions, and provide a seamless experience to library patrons. By incorporating advanced SQL Server concepts, this system optimizes data processing, retrieval, and reporting, ultimately enhancing the overall functionality and efficiency of the library management process.

The concept behind this project is to create a library management system that is capable to issue books and let consumers check different books and their titles categorically. It keeps track of all the details about the books in the library, borrowing date return and the total number of books available in the Library. The user will find this automated system easy instead of using the manual writing system.

DATABASE SCHEMA

The database schema typically includes tables for Books, Authors, Patron, Borrowed Books Details with well-defined relationships and constraints to maintain data integrity.

1. Authors Table: This table stores information about the different authors, including their Author ID which has Primary Key (unique identifier) and will automatically increments as new records added, Name, Birthdate, and Nationality.

2. Books Table: This table contains details about books, such as a Book ID which has Primary Key (unique identifier) and will automatically increments as new records added, Title, ISBN(International Standard Book Number which is unique identifier for books(to avoid duplicate number, available copies in library and publication year of book and Author ID to

link each book to its author referred as a foreign key from Author Table to establish the relationship between authors and their books.

3. Patron Table: It records details about the about library patrons, including Patron ID which has Primary Key (unique identifier) and will automatically increments as new records added, Name, and Contact information (e.g., email, phone number, address).

4. Borrowed Books Table: This table is designed to track the borrowing history of books by Patron. It includes Borrow ID which has Primary Key (unique identifier), Book ID refers to the Book ID from books table, Patron ID refers Patron ID from Patron Table and borrowing and return dates.

SQL QUERIES

- Created Database and used it.

```
Create Database Library;
Use Library;
```

- Creation of table.

1) Author Table

```
   Create table Authors(
     Author_ID INT Primary Key Identity(1,1),
     Author_Name Varchar(100),
     BirthDate Date,
     Nationality Varchar(30)
   );
```

2) Books Table

```
   Create Table Books(
      Books_Id INT Primary Key Identity(101,1),
      Title Varchar(100),
      ISBN Varchar(40) Unique,
      Publication_Year INT,
      Books_Availability INT,
      Author_ID INT
      FOREIGN KEY (Author_ID) REFERENCES Authors(Author_ID),
);
```

3) Patron Table

```
   CREATE TABLE Patron (
       Patron_ID INT PRIMARY KEY Identity(1001,1),
       First_Name VARCHAR(20) NOT NULL,
       Last_Name VARCHAR(20) NOT NULL,
       Email VARCHAR(50),
       P_Address VARCHAR(100),
       Phone_Number Varchar(100)
   );
```

4) Borrowed Books Table

```
CREATE TABLE Borrowed_Books (
    Borrow_ID Varchar(50) PRIMARY KEY,
    Books_Id INT,
    Patron_ID INT,
    Borrow_Date DATE NOT NULL,
    Return_Date DATE,
    FOREIGN KEY (Books_Id) REFERENCES Books(Books_Id),
    FOREIGN KEY (Patron_ID) REFERENCES Patron(Patron_ID)
);
```

- Insertion of data

1) Author table

```
INSERT INTO Authors(Author_Name, BirthDate, Nationality)
VALUES
 ('Harper Lee', '1926-04-28', 'United States'),
 ('George Orwell', '1903-06-25', 'United Kingdom'),
 ('F. Scott Fitzgerald', '1896-09-24', 'United States'),
 ('Jane Austen', '1775-12-16', 'United Kingdom'),
 ('J.D. Salinger', '1919-01-01', 'United States'),
 ('Aldous Huxley', '1894-07-26', 'United Kingdom'),
 ('J.R.R. Tolkien', '1892-01-03', 'United Kingdom'),
 ('Homer','1900-02-11', 'Greece'),
 ('Virginia Woolf','2022-05-05','United States'),
 ('Agatha Christie', '1890-09-15', 'British');
```

2) Books table

```
   INSERT INTO books (Title, Author_ID, Publication_Year,ISBN
Books_Availability)
VALUES
    ('To Kill a Mockingbird',1, 1960, '978-0-06-112008-4', 5),
    ('1984', 2, 1949, '978-0-452-28423-4', 3),
    ('The Great Gatsby', 3, 1925, '978-0-7432-7356-5', 7),
    ('Pride and Prejudice',4, 1813, '978-0-19-953556-9', 2),
    ('The Catcher in the Rye',5, 1951, '978-0-316-76948-0', 4),
    ('Brave New World',6, 1932, '978-0-06-085052-4', 1),
    ('The Hobbit',7, 1937, '978-0-618-34641-9', 6),
    ('The Lord of the Rings', 8, 1954, '978-0-345-33968-3', 2),
    ('To the Lighthouse', 9, 1927, '978-0-15-690739-2', 5),
    ('The Odyssey', 10, 1999 , 'N/A', 3);
```

3)Patron table

```
    INSERT INTO Patron(First_Name,Last_Name,Email,P_Address,
Phone_Number)
    VALUES
     ('John','Doe’,‘john.doe@example.com','123 Main St,US','212-354-
5500'),
     ('Alice', 'Smith', 'alice.smith@example.com','456 Elm St,
US','212-420-5800'),
     ('Bob', 'Johnson', 'bob.johnson@example.com', '789 Oak St,
UK','212-366-2000'),
     ('Emily', 'Davis', 'emily.davis@example.com','321 Maple
St,US','646-307-5745'),
     ('Michael', 'Wilson', 'michael.wilson@example.com', '567 Pine St,
UK','212-207-7000'),
     ('Sara', 'Anderson', 'sara.anderson@example.com', '890 Birch St,
US','212-419-5300'),
     ('David', 'Brown', 'david.brown@example.com','234 Cedar St,
UK','212-940-7390'),
     ('Lisa', 'Lee', 'lisa.lee@example.com', '432 Redwood
St,Canada','212-366-2000'),
     ('Chris', 'Miller', 'chris.miller@example.com', '765 Willow St,
Noida','212-366-2000'),
     ('Grace', 'Moore', 'grace.moore@example.com', '876 Oak
St,Pune','646-307-5745');
```

4) Borrowed books table

```
INSERT INTO Borrowed_Books (Borrow_ID,Books_Id,Patron_ID,Borrow_Date,
Return_Date)
VALUES
    ('B1',101, 1001, '2023-09-01', '2023-09-15'),
    ('B2',102, 1002, '2023-09-02', '2023-09-16'),
    ('B3',103, 1003, '2023-09-03', '2023-09-17'),
    ('B4',104, 1004, '2023-09-04', '2023-09-18'),
    ('B5',105, 1005, '2023-09-05', '2023-09-19'),
    ('B6',106, 1006, '2023-09-06', '2023-09-20'),
    ('B7',107, 1007, '2023-09-07', '2023-09-21'),
    ('B8',108, 1008, '2023-09-08', '2023-09-22'),
    ('B9',109, 1009, '2023-09-09', '2023-09-23'),
  ('B10',110,1010, '2023-09-10', '2023-09-24');
```

- Retrieving and displaying information

To see how many table exist in database

```
    Select * from sys.tables;
```

To see information in specific table

```
   Select * from Authors;
    Select * from Patron;
    Select * from Books;
    Select * from Borrowed_Books;
```

To search for a books by title, author name and ISBN

```
--Search for a books by title-
Select * from Books
where Title = '1984';

Select * from Books
where Title = 'The Lord of the Rings';

--Search for a book by ISBN--
Select * from Books
where ISBN = '978-0-06-112008-4';

Select * from Books
where ISBN = '978-0-06-085052-4';

--Search for a book by Author name--
Select b.Title, b.Author_ID, b.Publication_Year, b.ISBN,
b.Books_Availability, a.Author_Name from Books as b
join Authors as a on a.Author_ID = b.Author_ID
where Author_Name = 'George Orwell';

Select b.Title, b.Author_ID, b.Publication_Year, b.ISBN,
b.Books_Availability, a.Author_Name from Books as b
inner join Authors as a on a.Author_ID = b.Author_ID
where Author_Name = 'F. Scott Fitzgerald';
```

- Borrowing and returning books

```
--Borrowing a book--INSERT INTO Borrowed_Books
(Borrow_ID,Books_Id,Patron_ID,Borrow_Date)
VALUES
    ('B11',101, 1002, '2023-09-25');
```

```
--Returning a book--
Update Borrowed_Books
Set Return_Date = '2023-09-28'
where Borrow_ID = 'B11';
```

- To track book borrowing and return history

```
SELECT b.Title, p.Patron_ID, bb.Borrow_Date, bb.Return_Date
FROM Borrowed_Books as bb
JOIN Books as b
ON bb.Books_ID = b.Books_ID
JOIN Patron as p
ON bb.Patron_ID = p.Patron_ID;
```

- Select Into statement-it is used to insert data from on table to another without creating new table.

```
Select * into Patron_New
From Patron
Where Patron_ID >= 1005;
```

- Merge Statement to track Patron Information.

```
Merge Patron as p
Using Patron_New as Pt
On p.Patron_ID = pt.Patron_ID
when not matched by target then
insert (First_Name,Last_Name,Email,P_Address,Phone_Number)
values(pt.First_Name,pt.Last_Name,pt.Email,pt.P_Address,pt.Phon
e_Number)
when matched then update set
p.First_Name = pt.First_Name,
p.Last_Name = pt.Last_Name,
p.Email = pt.Email,
p.P_Address = pt.P_Address,
p.Phone_Number = pt.Phone_Number;
```

- Complex SQL queries using window functions and CTEs to identifying top-rated books by each author.

Step1-Created Window Function

```
SELECT a.Author_ID,Author_Name,Title,
ROW_NUMBER() OVER (ORDER by a.Author_ID) AS Rank
```

```sql
From Authors as a join Books as b
on a.Author_ID=b.Author_ID;
```

Step 2- Created CTE named "Top_Rated_Books" to assign a rank to each book within each author based on their ratings, with the highest rating receiving rank 1.Finally, we select the top-rated book for each author by filtering on Rank = 1.

```sql
with Top_Rated_Books as(
SELECT a.Author_ID,Author_Name,Title,
ROW_NUMBER() OVER (ORDER by a.Author_ID) AS Rank
From Authors as a join Books as b
on a.Author_ID=b.Author_ID)
SELECT Author_ID,Author_Name,Title as TopRatedBook
FROM Top_Rated_Books
Where Rank=1;
```

Testing

1. Database and Tables Creation: The SQL statements begin by creating database for your Library Management System and necessary tables .Each table is designed to store specific information related to authors, books, library patrons, and the borrowing history of books.

2. Data Insertion: After creating the tables, the statements insert sample data into each table to populate them for testing and demonstration purposes. Sample data includes author details, book information, patron information, and borrowing history.

3. Updating Book Availability: Two SQL statements demonstrate how to updating the "Availability" field in the "Books" table. One statement update borrowing date and the other return date.

4. Searching for Books by Title, ISBN or Author: These SQL queries show how to search for books by title, ISBN or author name in the "Books" table. The queries also involve joining the "Books" table with the "Authors" table to retrieve author information.

5. Tracking Book Borrowing History: This SQL query retrieves information about book borrowing history from the "Borrowed_Books" table. It joins

with the "Books" and "Patron" tables to display details such as book titles, patron id, borrowing dates, and return dates.

6.Complex Queries- This SQL query is used to identifying top-rated books by each author using window function and CTE. Different advanced SQL queries are also used such as CTEs, window function, sub-quries, Select into, merge, etc. to retrieve meaningful information.

Challenges Faced

- Designing an effective database schema that meets the project's requirements.
- Dealing with complex SQL queries, especially when trying to perform advanced queries, it is very challenging to write, understand and maintain.
- Identifying and resolving errors in SQL code.
- Maintaining accurate and up-to date documentation for the database schema and SQL queries.

Conclusion

The Library Management System (LMS) Database project represents a significant advancement in library management technology by embracing advanced SQL Server concepts and implementing a well-structured database schema, libraries can improve their services, make informed decisions, and enhance the overall patron experience. This project serves as a valuable asset for libraries seeking to modernize and streamline their operations in an increasingly digital age.

***