# 📚 Library Management System – SQL + Power BI Project

---

## 📌 Project Overview

\*\*Project Title:\*\* Library Management System

\*\*Level:\*\* Intermediate

\*\*Tools Used:\*\* SQL Server, Power BI

\*\*Database:\*\* LibraryDB

This project simulates a simple Library Management System using SQL for backend data handling and Power BI for data visualization. It helps in understanding how books are issued, returned, and how library data can be analyzed for insights.



---

## 🎯 Objectives

- Design and build a structured SQL database for a library system using normalization principles.

- Create and run SQL queries to analyze book availability, member activity, overdue returns, and monthly trends.

- Understand data relationships using ER diagrams and JOIN operations.

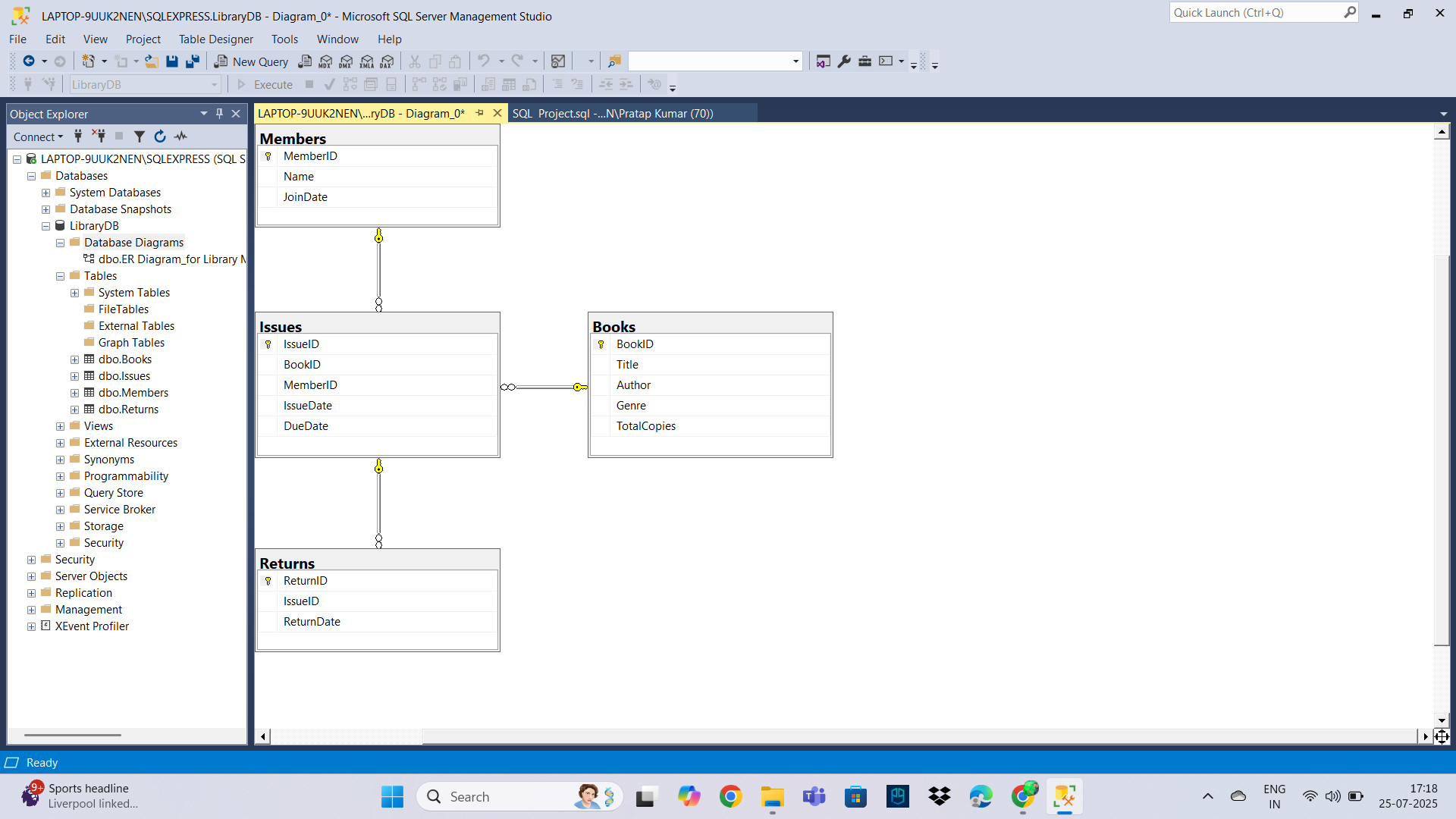
- Visualize key performance metrics in Power BI to reveal borrowing patterns, inventory status, and late returns.

- Showcase end-to-end data analytics workflow from data modeling to reporting findings in Power BI dashboards.

---

## 🧱 Project Structure

### 1. 📁 Database Design (ER Diagram)



The database consists of 4 tables:

- \*\*Books\*\* (BookID, Title, Author, Genre, TotalCopies)

- \*\*Members\*\* (MemberID, Name, JoinDate)

- \*\*Issues\*\* (IssueID, BookID, MemberID, IssueDate, DueDate)

- \*\*Returns\*\* (ReturnID, IssueID, ReturnDate)

Each table includes relevant columns and relationships.

--Creating Database

CREATE DATABASE LibraryDB

-- Creating "Books" Table

DROP TABLE IF EXISTS Books

CREATE TABLE Books

(

BookID INT PRIMARY KEY,

Title VARCHAR(100),

Author VARCHAR(100),

Genre VARCHAR(50),

TotalCopies INT

)

-- Creating "Members" Table

DROP TABLE IF EXISTS Members

CREATE TABLE Members

(

MemberID INT PRIMARY KEY,

Name VARCHAR(100),

JoinDate DATE

)

-- Creating "Issues" Table

DROP TABLE IF EXISTS Issues

CREATE TABLE Issues

(

IssueID INT PRIMARY KEY,

BookID INT,

MemberID INT,

IssueDate DATE,

DueDate DATE,

FOREIGN KEY (BookID) REFERENCES Books(BookID),

FOREIGN KEY (MemberID) REFERENCES Members(MemberID)

)

-- Creating "Returns" Table

DROP TABLE IF EXISTS Returns

CREATE TABLE Returns

(

ReturnID INT PRIMARY KEY,

IssueID INT,

ReturnDate DATE,

FOREIGN KEY (IssueID) REFERENCES Issues(IssueID)

)

---

### 2. 💻 SQL Tasks & Queries

Task 1: Get all members who joined after June 2023

SELECT \* FROM Members

WHERE JoinDate > '2023-06-01'

Task 2: Find all books in the 'Programming' genre

SELECT \* FROM Books

WHERE Genre = 'Programming'

Task 3: Find books with total copies between 4 and 6

SELECT \* FROM Books

WHERE TotalCopies BETWEEN 4 AND 6

Task 4: Find members whose name contains 'a'

SELECT \* FROM Members

WHERE Name LIKE '%a%'

Task 5: Count how many issues each book has

SELECT BookID, COUNT(\*) AS IssueCount

FROM Issues

GROUP BY BookID

Task 6: List books by number of total copies (highest first)

SELECT \* FROM Books

ORDER BY TotalCopies DESC

### 4. Data Analysis & Findings

Task 7: Show all books with available copies

SELECT

b.Title,

b.TotalCopies,

(b.TotalCopies - COUNT(i.IssueID)) AS AvailableCopies

FROM Books b

LEFT JOIN Issues i ON b.BookID = i.BookID

GROUP BY b.BookID, b.Title, b.TotalCopies

Task 8: Most borrowed books

SELECT

b.Title, COUNT(i.BookID) AS TimesBorrowed

FROM Issues i

JOIN Books b ON i.BookID = b.BookID

GROUP BY b.Title, b.BookID

ORDER BY TimesBorrowed

Task 9: Active members (who borrowed more than 1 book)

SELECT

m.Name,

COUNT(i.IssueID) AS TotalBooksBorrowed

FROM Members m

JOIN Issues i ON m.MemberID = i.MemberID

GROUP BY m.Name, i.MemberID

HAVING COUNT(i.IssueID) > 1

Task 10: Overdue books (not returned by due date)

SELECT

i.IssueID,

b.Title,

m.Name,

i.DueDate,

COALESCE(CONVERT(VARCHAR, r.ReturnDate, 23), 'Not Returned') AS ReturnDate

FROM Issues i

JOIN Books b ON i.BookID = b.BookID

JOIN Members m ON i.MemberID = m.MemberID

LEFT JOIN Returns r ON i.IssueID = r.IssueID

WHERE (r.ReturnDate IS NULL AND i.DueDate < CAST(GETDATE() AS DATE))

OR (r.ReturnDate > i.DueDate)

Task 11: Monthly book issue trend

SELECT

FORMAT(IssueDate, 'yyyy-MM') AS Month,

COUNT(\*) AS TotalIssues

FROM Issues

GROUP BY FORMAT(IssueDate, 'yyyy-MM')

ORDER BY Month

---

###5. 📊 Power BI Dashboard

Report built in Power BI Desktop (July 2025 version).

\*\*Key Visualizations:\*\*

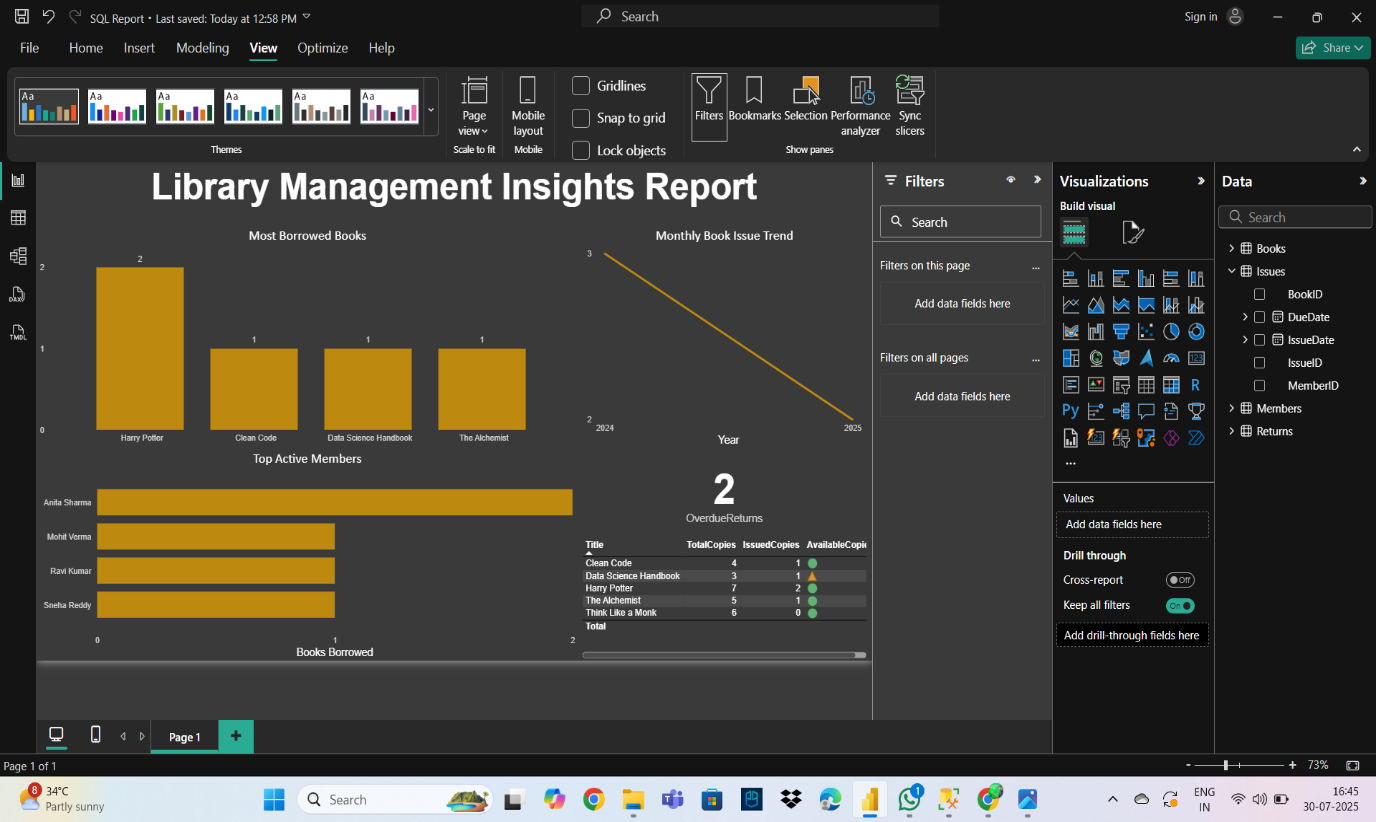
Most Borrowed Books (Column Chart) to show which books are most in demand.

Top Active Members (Bar Chart) identifies the most engaged members.

Available Copies (Conditional Formatting Table) quickly identifies which books are running low.

Monthly Book Issue Trend (Line Chart) reveals books peak and low season.

Overdue Returns (KPI Card) highlights books overdue cases.



---

###6. 📈 Insights & Findings

📖 Harry Potter was the most borrowed book.

🙋 Anita Sharma borrowed the most books.

❗ 2 books were returned late or not at all.

🟠 Conditional formatting shows low stock books in yellow icon.

📅 Book issues peaked in late 2024.

---

###7. 🔍 How to Use

Download SQL\_Project.sql and execute in SSMS.

Use the ER Diagram to understand schema relationships.

Open Library Insights Report.pbix in Power BI Desktop.

Refresh the SQL connection if needed.

Interact with filters to explore trends.

---

###8. 📌 Conclusion

This project demonstrates how a simple library dataset can be turned into actionable business intelligence through clean data modeling, SQL querying, and effective visualization using Power BI.

---

###9. ✍️ Author

Satya Kameswari

Aspiring Data Analyst | Passionate about SQL & Power BI

GitHub: [https://github.com/SatyaKameswari-analyst]

- 📄 [SQL Project Script]( [SQL%20 Project.sql](https://1drv.ms/u/c/8aa5a3f9e1b2e1cb/EZvAnO_mzU9Av4CCwpom__sBnjZZgiC2wnbriC4u3YC3Kg?e=zDBDJC))

- 📊 [Power BI Report]( [LIbrary %20Insights %20Report.pbix](https://1drv.ms/u/c/8aa5a3f9e1b2e1cb/Ef8EXncNUzFCogifl30JBWoB2JL21sxvJEgHgqOs7ASVBg?e=OeHP77))

- 🖼️ [See Screenshots]( [Library%20 Management%20 System%20 Analysis %20SQL](https://1drv.ms/f/c/8aa5a3f9e1b2e1cb/Ely6Jlrq64VErzUOSOsUtdsB2UGEX7zTrbO-c5X4yoBh4g?e=Pmhp5u))