

# Library Management System - SQL Project Documentation

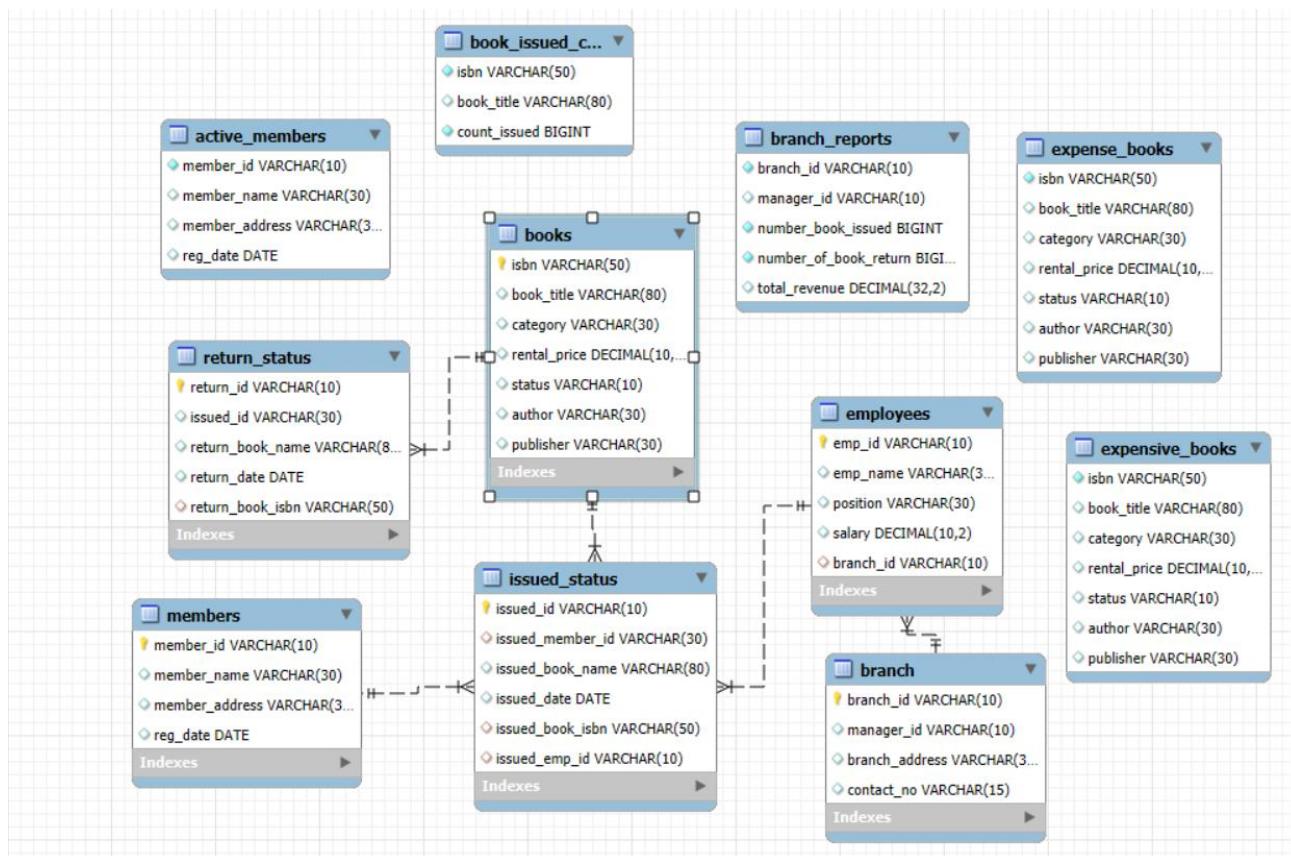
## Overview

The Library Management System (LMS) is a structured SQL project designed to manage core library operations such as maintaining book records, managing members, tracking book issues and returns, and generating analytical reports. The system includes multiple interconnected tables that simulate a real-world library workflow using SQL queries ranging from basic CRUD operations to advanced JOINs, aggregate functions, and CTAS (Create Table As Select).

## Objectives

- Design a normalized database structure for library operations.
- Insert realistic sample data for all tables.
- Perform CRUD operations to understand data manipulation.
- Use JOIN operations to link related tables and extract meaningful insights.
- Use aggregate functions to analyze performance metrics.
- Generate summary and analytical reports using CTAS.
- Track overdue books and identify member/employee patterns.
- Build a foundation for a scalable SQL-based library system.

## ER Diagram



## Task And Insights

-- Task 1. Create a New Book Record -- "978-1-60129-456-2', 'To Kill a Mockingbird', 'Classic', 6.00, 'yes', 'Harper Lee', 'J.B. Lippincott & Co.'"

INSERT INTO books(isbn, book\_title, category, rental\_price, status, author, publisher) VALUES ('978-1-60129-456-2', 'To Kill a Mockingbird', 'Classic', 6.00, 'yes', 'Harper Lee', 'J.B. Lippincott & Co.');

### Insights:

This query inserts a new book into the books table using an INSERT statement. It ensures all required fields such as ISBN, category, rental price, author, and publisher are provided.

978-1-60129-456-2	To Kill a Mockingbird	Classic	6.00	yes	Harper Lee	J.B. Lippincott & Co.
NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- Task 2: Update an Existing Member's Address.

update members

set member\_address = '999 New St, Suite 10'

where member\_id = 'c119';

### Insights:

Uses UPDATE with a WHERE clause to modify only the selected member's address, ensuring data integrity.

C119	John	999 New St, Suite 10	2024-05-01
NULL	NULL	NULL	NULL

-- Task 3: Delete a Record from the Issued Status Table. Objective: Delete the record with issued\_id = 'IS121' from the issued\_status table.

Delete from issued\_status

where issued\_id = 'IS121';

### Insights:

DELETE removes a specific issued book entry using issued\_id as the filter. This maintains accuracy in issued records.

IS119	C110	Brave New World	2024-03-23	978-0-452-28240-7	E108
IS120	C110	The Road	2024-03-24	978-0-670-81302-4	E108
IS122	C102	Fahrenheit 451	2024-03-26	978-0-451-52993-5	E109
IS123	C103	Dune	2024-03-27	978-0-345-39180-3	E109

-- Task 4: Retrieve All Books Issued by a Specific Employee.

select issued\_book\_name from issued\_status

where issued\_emp\_id = 'E104';

### Insights:

Uses a SELECT with a WHERE filter to list all books issued by employee 'E104'. Helps track employee performance.

issued_book_name
Animal Farm
One Hundred Years of Solitude
The Great Gatsby
The Great Gatsby

## -- Task 5: List Members Who Have Issued More Than One Book

```
select issued_emp_id, count(*) as issued_book  
from issued_status  
group by issued_emp_id  
having count(*) > 1;
```

### Insights:

Uses GROUP BY and HAVING to count books issued by each employee. HAVING ensures only those with more than one entry are shown.

issued_emp_id	issued_book
E101	2
E102	2
E104	4
E105	4
E106	6
E107	3
E108	4
E109	2
E110	6

## -- Task 6: Create Summary Tables: Used CTAS to generate new tables based on query results - each book and total book\_issued\_cnt.

```
create table book_issued_cnt as  
select b.isbn,b.book_title,count(ist.issued_id) as count_issued  
from issued_status as ist  
join books as b on ist.issued_book_isbn = b.isbn  
group by b.isbn,b.book_title;  
select * from book_issued_cnt;
```

### Insights:

Uses CTAS + JOIN to count how many times each book was issued, producing a new summary table.

isbn	book_title	count_issued
978-0-06-025492-6	Where the Wild Things Are	1
978-0-06-112008-4	To Kill a Mockingbird	1
978-0-06-112241-5	The Kite Runner	1
978-0-06-440055-8	Charlotte's Web	1
978-0-09-957807-9	A Game of Thrones	1
978-0-14-027526-3	A Tale of Two Cities	1
978-0-14-044930-3	The Histories	1
978-0-14-118776-1	One Hundred Years of Solitude	1
978-0-14-143951-8	Pride and Prejudice	1
978-0-141-44171-6	Jane Eyre	1
978-0-19-280551-1	The Guns of August	1
978-0-307-37840-1	The Alchemist	1
978-0-307-58837-1	Sapiens: A Brief History of Hu...	1
978-0-330-25864-8	Animal Farm	2
978-0-345-39180-3	Dune	1
978-0-375-41398-8	The Diary of a Young Girl	1
978-0-393-05081-8	A Peoples History of the Unite...	1
978-0-393-91257-8	Guns, Germs, and Steel: The ...	1
978-0-451-52993-5	Fahrenheit 451	1
978-0-451-52994-2	Moby Dick	1
978-0-452-28240-7	Brave New World	1
978-0-525-47535-5	The Great Gatsby	2
978-0-553-29698-2	The Catcher in the Rye	1
978-0-670-81302-4	The Road	1
978-0-679-64115-3	1984	1

## -- Task 7. Retrieve All Books in a Specific Category:

```
select * from books  
where category = 'History';
```

### Insights:

Simple SELECT with WHERE to filter books by category (e.g., History).

isbn	book_title	category	rental_price	status	author	publisher
978-0-14-044930-3	The Histories	History	5.50	Yes	Herodotus	Penguin Classics
978-0-19-280551-1	The Guns of August	History	7.00	Yes	Barbara W. Tuchman	Oxford University Press
978-0-307-58837-1	Sapiens: A Brief History of Humankind	History	8.00	no	Yuval Noah Harari	Harper Perennial
978-0-375-41398-8	The Diary of a Young Girl	History	6.50	no	Anne Frank	Bantam
978-0-393-05081-8	A Peoples History of the United States	History	9.00	Yes	Howard Zinn	Harper Perennial
978-0-393-91257-8	Guns, Germs, and Steel: The Fates of Human S...	History	7.00	Yes	Jared Diamond	W. W. Norton & Company
978-0-7432-7357-1	1491: New Revelations of the Americas Before ...	History	6.50	no	Charles C. Mann	Vintage Books
NULL	NULL	NULL	NULL	NULL	NULL	NULL

## -- Task 8: Find Total Rental Income by Category:

```
select category,sum(rental_price) as total_renatl,  
count(*) as book_count  
from books as b  
join issued_status as ist  
on ist.issued_book_isbn = b.isbn  
group by category;
```

### Insights:

Uses SUM and COUNT on rental\_price grouped by category. This provides revenue analytics per book category.

category	total_renatl	book_count
Children	7.50	2
Classic	59.00	10
Fiction	14.50	3
Fantasy	28.50	4
History	49.50	7
Literary Fiction	6.50	1
Science Fiction	8.50	1
Dystopian	25.50	4
Mystery	7.50	1
Horror	7.00	1

## -- Task 9. List Members Who Registered in the Last 180 Days:

```
select * from members  
where reg_date = date_sub(curdate(),interval 180 day);
```

### Insights:

Checks registration date using DATE\_SUB and CURDATE. Helps identify recent members.

member_id	member_name	member_address	reg_date
NULL	NULL	NULL	NULL

-- Task 10. List Employees with Their Branch Manager's Name and their branch details:

```
select e.emp_id as employee_id,  
e.emp_name as employee_name,  
b.branch_id as branch_id,  
m.emp_name as manager  
from employees e  
join branch b  
on e.branch_id = b.branch_id  
join employees m  
on m.emp_id = b.manager_id;
```

**Insights:**

A self-join on employees maps employees to branch managers. Useful for HR and branch hierarchy reports.

employee_id	employee_name	branch_id	manager
E106	Michelle Ramirez	B001	Daniel Anderson
E105	Sarah Brown	B001	Daniel Anderson
E104	Emily Davis	B001	Daniel Anderson
E103	Mike Johnson	B001	Daniel Anderson
E101	John Doe	B001	Daniel Anderson
E102	Jane Smith	B002	Daniel Anderson
E109	Daniel Anderson	B003	Daniel Anderson
E108	Jessica Taylor	B004	Laura Martinez
E111	Christopher Lee	B005	Laura Martinez
E110	Laura Martinez	B005	Laura Martinez
E107	Michael Thompson	B005	Laura Martinez

-- Task 11. Create a Table of Books with Rental Price Above a Certain Threshold:

```
create table expense_books as  
select * from books  
where rental_price > 7.00;  
select * from expense_books;
```

**Insights:**

Uses CTAS to create a new table expense\_books for high-rental books. Helps identify premium books.

isbn	book_title	category	rental_price	status	author	publisher
978-0-09-957807-9	A Game of Thrones	Fantasy	7.50	yes	George R.R. Martin	Bantam
978-0-307-58837-1	Sapiens: A Brief History of Humankind	History	8.00	no	Yuval Noah Harari	Harper Perennial
978-0-345-39180-3	Dune	Science Fiction	8.50	yes	Frank Herbert	Ace
978-0-393-05081-8	A Peoples History of the United States	History	9.00	yes	Howard Zinn	Harper Perennial
978-0-525-47535-5	The Great Gatsby	Classic	8.00	yes	F. Scott Fitzgerald	Scribner
978-0-7432-4722-4	The Da Vinci Code	Mystery	8.00	yes	Dan Brown	Doubleday
978-0-7432-4722-5	Angels & Demons	Mystery	7.50	yes	Dan Brown	Doubleday

-- Task 12: Retrieve the List of Books Not Yet Returned

```
select *from issued_status as ist  
left join return_status as rs  
on ist.issued_id = rs.issued_id  
where rs.return_id is null;
```

**Insights:**

LEFT JOIN issued\_status with return\_status and filters rows where return\_id is NULL. Identifies pending returns.

issued_id	issued_member_id	issued_book_name	issued_date	issued_book_isbn	issued_emp_id	return_id	issued_id	return_book_name	return_date	return_book_isbn
IS122	C102	Fahrenheit 451	2024-03-26	978-0-451-52993-5	E109	NULL	NULL	NULL	NULL	NULL
IS123	C103	Dune	2024-03-27	978-0-345-39180-3	E109	NULL	NULL	NULL	NULL	NULL
IS124	C104	Where the Wild Things Are	2024-03-28	978-0-06-025492-6	E110	NULL	NULL	NULL	NULL	NULL
IS125	C105	The Kite Runner	2024-03-29	978-0-06-112241-5	E110	NULL	NULL	NULL	NULL	NULL
IS126	C105	Charlotte's Web	2024-03-30	978-0-06-440055-8	E110	NULL	NULL	NULL	NULL	NULL
IS127	C105	Beloved	2024-03-31	978-0-679-77644-3	E110	NULL	NULL	NULL	NULL	NULL
IS128	C105	A Tale of Two Cities	2024-04-01	978-0-14-027526-3	E110	NULL	NULL	NULL	NULL	NULL
IS129	C105	The Stand	2024-04-02	978-0-7434-7679-3	E110	NULL	NULL	NULL	NULL	NULL
IS130	C106	Moby Dick	2024-04-03	978-0-451-52994-2	E101	NULL	NULL	NULL	NULL	NULL
IS131	C106	To Kill a Mockingbird	2024-04-04	978-0-06-112008-4	E101	NULL	NULL	NULL	NULL	NULL
IS132	C106	The Hobbit	2024-04-05	978-0-7432-7356-4	E106	NULL	NULL	NULL	NULL	NULL
IS133	C107	Angels & Demons	2024-04-06	978-0-7432-4722-5	E106	NULL	NULL	NULL	NULL	NULL
IS134	C107	The Diary of a Young Girl	2024-04-07	978-0-375-41398-8	E106	NULL	NULL	NULL	NULL	NULL
IS135	C107	Sapiens: A Brief History o...	2024-04-08	978-0-307-58837-1	E108	NULL	NULL	NULL	NULL	NULL
IS136	C107	1491: New Revelations of...	2024-04-09	978-0-7432-7357-1	E102	NULL	NULL	NULL	NULL	NULL
IS137	C107	The Catcher in the Rye	2024-04-10	978-0-553-29698-2	E103	NULL	NULL	NULL	NULL	NULL
IS138	C108	The Great Gatsby	2024-04-11	978-0-525-47535-5	E104	NULL	NULL	NULL	NULL	NULL
IS139	C109	Harry Potter and the Sorc...	2024-04-12	978-0-679-76489-8	E105	NULL	NULL	NULL	NULL	NULL

-- Task 13: Identify Members with Overdue Books Write a query to identify members who have overdue books (assume a 30-day return period). Display the member's\_id, member's name, book title, issue date, and days overdue

```
select ist.issued_member_id,m.member_name,b.book_title,ist.issued_date,
date_sub(current_date(),interval 30 day) as over_due_days
from issued_status as ist
join members as m on m.member_id = ist.issued_member_id
join books as b on b.isbn = ist.issued_book_isbn
left join return_status as rs on rs.issued_id = ist.issued_id
where rs.return_date is null
and date_sub(current_date(),interval 30 day) > 30
order by ist.issued_member_id,m.member_name,b.book_title,ist.issued_date;
```

#### Insights:

Compares issue date with CURRENT\_DATE minus 30 days. Detects overdue books and displays member & book details.

issued_member_id	member_name	book_title	issued_date	over_due_days
C102	Bob Smith	Fahrenheit 451	2024-03-26	2025-10-26
C103	Carol Davis	Dune	2024-03-27	2025-10-26
C104	Dave Wilson	Where the Wild Things Are	2024-03-28	2025-10-26
C105	Eve Brown	A Tale of Two Cities	2024-04-01	2025-10-26
C105	Eve Brown	Beloved	2024-03-31	2025-10-26
C105	Eve Brown	Charlotte's Web	2024-03-30	2025-10-26
C105	Eve Brown	The Kite Runner	2024-03-29	2025-10-26
C105	Eve Brown	The Stand	2024-04-02	2025-10-26
C106	Frank Thomas	Moby Dick	2024-04-03	2025-10-26
C106	Frank Thomas	The Hobbit	2024-04-05	2025-10-26
C106	Frank Thomas	To Kill a Mockingbird	2024-04-04	2025-10-26
C107	Grace Taylor	1491: New Revelations of...	2024-04-09	2025-10-26
C107	Grace Taylor	Angels & Demons	2024-04-06	2025-10-26
C107	Grace Taylor	Sapiens: A Brief History o...	2024-04-08	2025-10-26
C107	Grace Taylor	The Catcher in the Rye	2024-04-10	2025-10-26
C108	Henry Anderson	The Great Gatsby	2024-04-11	2025-10-26
C109	Ivy Martinez	Harry Potter and the Sorc...	2024-04-12	2025-10-26

-- Task 14: Branch Performance Report Create a query that generates a performance report for each branch, showing the number of books issued, the number of books returned, and the total revenue generated from book rentals

```
create table branch_reports as
select b.branch_id,b.manager_id,
count(ist.issued_id) as number_book_issued,
count(rs.return_id) as number_of_book_return,
sum(bk.rental_price) as total_revenue
from issued_status as ist
join employees as e on e.emp_id = ist.issued_emp_id
join branch as b on e.branch_id = b.branch_id
left join return_status as rs on rs.issued_id = ist.issued_id
join books as bk on ist.issued_book_isbn = bk.isbn
group by b.branch_id,b.manager_id;
select * from branch_reports;
```

**Insights:**

Combines multiple JOINs to calculate: issued books, returned books, and total rental revenue per branch. Creates a summary table branch\_reports.

branch_id	manager_id	number_book_issued	number_of_book_return	total_revenue
B001	E109	17	9	111.50
B002	E109	2	0	12.00
B003	E109	2	0	14.00
B004	E110	4	3	26.50
B005	E110	9	3	50.00

-- Task 15: CTAS: Create a Table of Active Members Use the CREATE TABLE AS (CTAS) statement to create a new table active\_members containing members who have issued at least one book in the last 2 months.

```
create table active_members as
select distinct m.member_id,m.member_name,m.member_address,m.reg_date
from members as m
join issued_status as ist on m.member_id = ist.issued_member_id
where ist.issued_date >= date_sub(current_date,interval 2 month);
select * from active_members;
```

**Insights:**

Identifies members who issued a book in last 2 months using DATE\_SUB. Stores them in a new table active\_members.

member_id	member_name	member_address	reg_date

-- Task 16: Find Employees with the Most Book Issues Processed Write a query to find the top 3 employees who have processed the most book issues. Display the employee name, number of books processed, and their branch.

```
select e.emp_name,b.branch_address as branch,
count(ist.issued_id) as no_book_issued
from issued_status as ist
join employees as e on e.emp_id = ist.issued_emp_id
join branch as b on e.branch_id = b.branch_id
group by e.emp_name,branch
```

```
order by count(ist.issued_id) desc limit 3;
```

**Insights:**

Ranks employees by number of processed issues using ORDER BY COUNT(\*) DESC LIMIT 3.

emp_name	branch	no_book_issued
Laura Martinez	890 Maple St	6
Michelle Ramirez	123 Main St	6
Emily Davis	123 Main St	4

**-- Task 17: Identify Members Issuing High-Risk Books** Write a query to identify members who have issued books more than twice with the status "damaged" in the books table. Display the member name, book title, and the number of times they've issued damaged books.

```
select m.member_name,b.book_title,  
count(*) as damaged_issue_count  
from issued_status as ist  
join members as m on ist.issued_member_id = m.member_id  
join books as b on ist.issued_book_isbn = b.isbn  
where b.status = 'damaged'  
group by m.member_name,b.book_title  
having count(*) > 2;
```

**Insights:**

Uses GROUP BY and HAVING COUNT(\*) > 2 to detect repeated issuers of damaged books. Helpful for risk management.

member_name	book_title	damaged_issue_count

---

## Conclusion

This Library Management System project highlights the practical application of SQL in solving real-world data management challenges. The system demonstrates how relational databases can be used to efficiently manage books, members, employees, and operational workflows such as issuing and returning books.

Through the execution of 17 structured SQL tasks, this project explores advanced concepts such as JOIN operations, aggregate functions, subqueries, CTAS (Create Table As Select), overdue identification, and performance reporting.

Overall, this documentation provides a complete understanding of SQL database design and query implementation, showcasing strong analytical, database development, and problem-solving skills.