SQL Project

Library Management Analysis

by Gitanjali Pekamwar

❑ Table of Content:

➢ Introduction

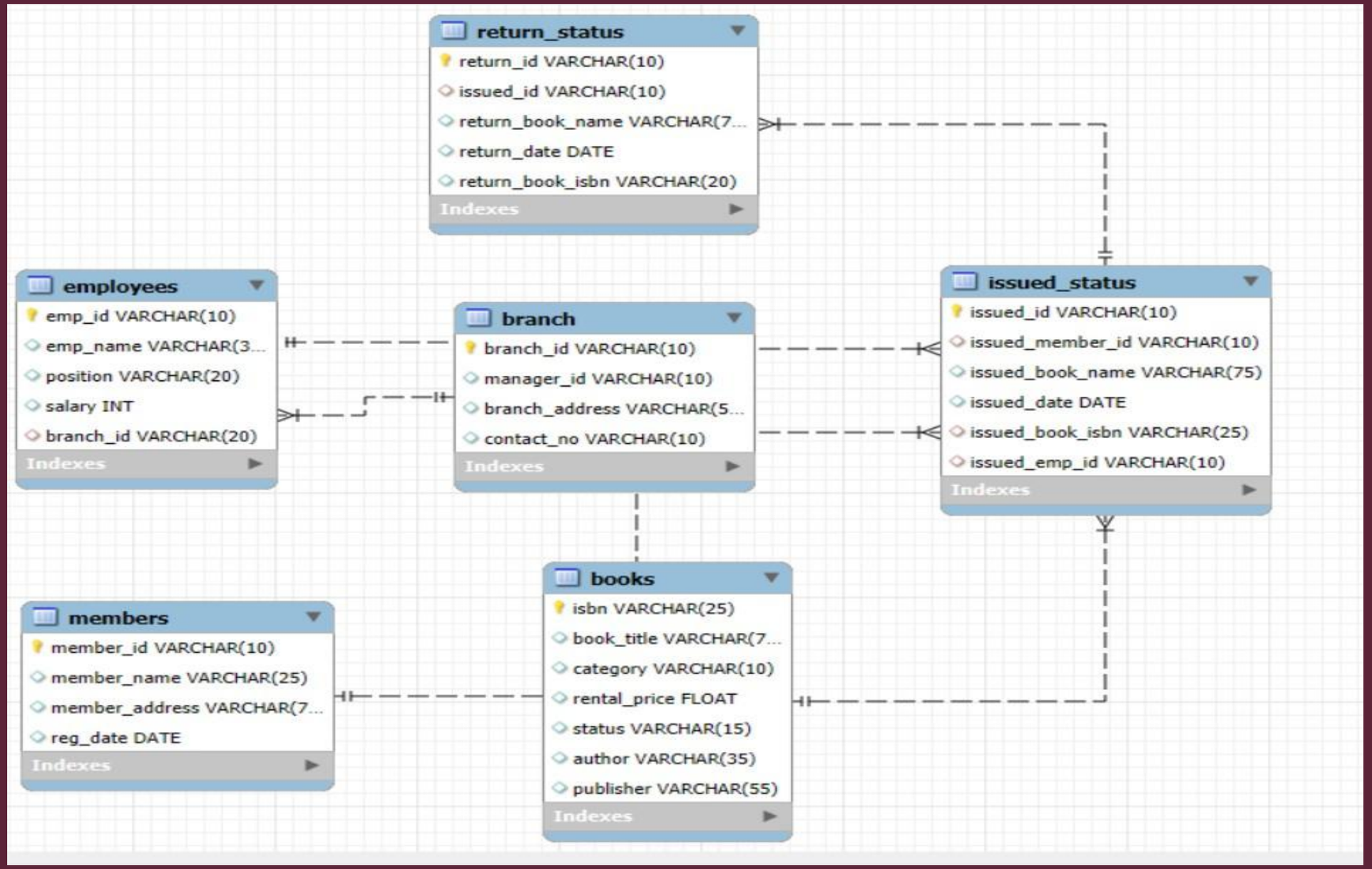
❑ Project Goals

- This project demonstrates the implementation of a Library Management System using SQL. It includes creating and managing tables, performing CRUD operations, and executing advanced SQL queries. The goal is to showcase skills in database design, manipulation, and querying, analyze library data to improve operations and enhance the overall library experience.

❑ Data Source

- The project utilized a variety of data sources, including books, employees, branch , members, returned status, issued status.

❑ Database Structure



➢ Tech-Stacked Used

- MYSQL Workbench 8.0 CE : I used MySQL workbench for querying and analysis the dataset .

- Gamma : I used Gamma for generating the presentation .

- Microsoft Power-Point: I used Microsoft Power-Point for modifying presentation manually.

Creating Tables

```sql
-- Library Management System project2
-- creating branch table
CREATE TABLE branch (
branch_id   varchar(10) primary key,
manager_id  varchar(10),
branch_address  varchar(55),
contact_no varchar(10)
);
```

```sql
-- creating employees table
CREATE TABLE employees
 (
emp_id  VARCHAR(10) PRIMARY KEY,
emp_name VARCHAR(30),
position VARCHAR(20),
salary  INT,
branch_id VARCHAR(20)
);
```

```sql
-- CREATING BOOKS TABLE
CREATE TABLE books
  (
    isbn VARCHAR(25) PRIMARY KEY,
    book_title VARCHAR(75),
    category VARCHAR(10),
    rental_price FLOAT,
    status VARCHAR(15),
    author   VARCHAR(35),
    publisher VARCHAR(55)
    );
```

```sql
-- CREATING TABLE MEMBERS
CREATE TABLE members(
member_id   VARCHAR(10) PRIMARY KEY,
member_name VARCHAR(25),
member_address  VARCHAR(75),
reg_date DATE
);
```

```sql
-- CREATING TABLE ISSUED_STATUS
CREATE TABLE issued_status (
issued_id   VARCHAR(10) PRIMARY KEY,
issued_member_id VARCHAR(10),    -- fk
issued_book_name VARCHAR(75),
issued_date DATE,
issued_book_isbn VARCHAR(25),     -- fk
issued_emp_id VARCHAR(10)              -- fk
);
```

```sql
-- CREATING TABLE RETURN_STATUS
CREATE TABLE return_status (
return_id   VARCHAR(10) PRIMARY KEY,
issued_id   VARCHAR(10),
return_book_name    VARCHAR(75),
return_date DATE,
return_book_isbn VARCHAR(20)
);
```

CRUD Operations

```sql
-- Task1) 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.');
```

```sql
-- task2: Update an Existing Member's Address
update members
set member_address = ' 125 Main st'
where member_id='C101';
```

```sql
-- Task 3: Delete a Record from the Issued Status Table
-- Objective: Delete the record with issued_id = 'IS121'
delete from issued_status
where issued_id = 'IS121'
```

```sql
-- Task 4: Retrieve All Books Issued by a Specific Employee
-- Objective: Select all books issued by the employee with emp_id = 'E101';
select * from issued_status
where issued_emp_id = 'E101';
```

```sql
-- Task 5: List Members Who Have Issued More Than One Book**
-- Objective: Use GROUP BY to find members who have issued more than one book.
select issued_member_id ,
    count(issued_id) as total_book_issued
from issued_status
group by 1
having count(issued_id > 1);
```

CTAS (Create Table As Select)

```
43    -- Task 6: Create Summary Tables**: Used CTAS to generate new tables based on
44    -- query results - each book and total book_issued_cnt
45
46 •  CREATE TABLE book_issued_cnt AS SELECT b.isbn, b.book_title,
47                 COUNT(ist.issued_id) AS no_issued FROM
48        books AS b
49            LEFT JOIN
50        issued_status AS ist ON ist.issued_id = b.isbn
51    GROUP BY 1;
```

Data Analysis Tasks

```
54    -- Task 7. Retrieve All Books in a Specific Category
55
56 •  SELECT
57        *
58    FROM books
59    WHERE category = 'classic';
```

isbn	book_title	category	rental_price	status	author	publisher
978-0-06-112008-4	To Kill a Mockingbird	Classic	5	yes	Harper Lee	J.B. Lippincott & Co.
978-0-14-027526-3	A Tale of Two Cities	Classic	4.5	yes	Charles Dickens	Penguin Books
978-0-14-143951-8	Pride and Prejudice	Classic	5	yes	Jane Austen	Penguin Classics
978-0-141-44171-6	Jane Eyre	Classic	4	yes	Charlotte Bronte	Penguin Classics
978-0-330-25864-8	Animal Farm	Classic	5.5	yes	George Orwell	Penguin Books
978-0-451-52994-2	Moby Dick	Classic	6.5	yes	Herman Melville	Penguin Books
978-0-525-47535-5	The Great Gatsby	Classic	8	yes	F. Scott Fitzgerald	Scribner
978-0-553-29698-2	The Catcher in the Rye	Classic	7	yes	J.D. Salinger	Little, Brown and Company
978-1-60129-456-2	To Kill a Mockingbird	Classic	6	yes	Harper Lee	J.B. Lippincott & Co.

Task 8: Find Total Rental Income by Category

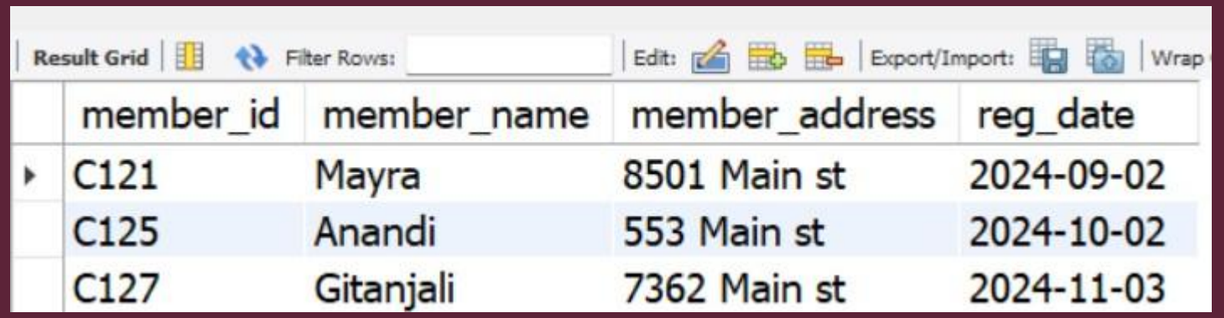
```sql
--  Task 8: Find Total Rental Income by Category:
SELECT
    b.category,
    SUM(b.rental_price) as Total_rental_price,
    COUNT(*)
FROM
issued_status as ist
JOIN
books as b
ON b.isbn = ist.issued_book_isbn
GROUP BY 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conte

category	Total_rental_price	COUNT(*)
Classic	59	10
Fiction	14.5	3
Fantasy	28.5	4
History	49.5	7
Dystopian	25.5	4
Children	7.5	2
Horror	7	1
Mystery	7.5	1

Task9. List Members Who Registered in the Last 180 Days:

```
SELECT
    *
FROM
    members
WHERE
    reg_date > DATE_SUB(CURRENT_DATE(),
        INTERVAL 180 DAY);
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

member_id	member_name	member_address	reg_date
C121	Mayra	8501 Main st	2024-09-02
C125	Anandi	553 Main st	2024-10-02
C127	Gitanjali	7362 Main st	2024-11-03

Task10.List Employees with Their Branch Manager's Name and their branch details

```sql
SELECT
    e1.emp_id,
    e1.emp_name,
    e1.position,
    e1.salary,
    b.*,
    e2.emp_name as manager
FROM employees as e1
JOIN
branch as b
ON e1.branch_id = b.branch_id
JOIN
employees as e2
ON e2.emp_id = b.manager_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

emp_id	emp_name	position	salary	branch_id	manager_id	branch_address	contact_no	manager
E109	Daniel Anderson	Manager	57000	B003	E109	789 Oak St	+919099988678	Daniel Anderson
E106	Michelle Ramirez	Assistant	43000	B001	E109	123 Main St	+919099988676	Daniel Anderson
E105	Sarah Brown	Assistant	42000	B001	E109	123 Main St	+919099988676	Daniel Anderson
E104	Emily Davis	Assistant	40000	B001	E109	123 Main St	+919099988676	Daniel Anderson
E103	Mike Johnson	Librarian	55000	B001	E109	123 Main St	+919099988676	Daniel Anderson
E102	Jane Smith	Clerk	45000	B002	E109	456 Elm St	+919099988677	Daniel Anderson
E101	John Doe	Clerk	60000	B001	E109	123 Main St	+919099988676	Daniel Anderson
E111	Christopher Lee	Assistant	65000	B005	E110	890 Maple St	+919099988680	Laura Martinez
E110	Laura Martinez	Manager	41000	B005	E110	890 Maple St	+919099988680	Laura Martinez
E108	Jessica Taylor	Clerk	46000	B004	E110	567 Pine St	+919099988679	Laura Martinez
E107	Michael Thomp...	Clerk	62000	B005	E110	890 Maple St	+919099988680	Laura Martinez

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

```
107 •   CREATE TABLE expensive_books AS
108     SELECT * FROM books
109     WHERE rental_price > 7.00;
110
111 •   SELECT * FROM expensive_books;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

isbn	book_title	category	rental_price	status	author	publisher
978-0-09-957807-9	A Game of Thrones	Fantasy	7.5	yes	George R.R. Martin	Bantam
978-0-307-58837-1	Sapiens: A Brief History of...	History	8	no	Yuval Noah Harari	Harper Perennial
978-0-393-05081-8	A Peoples History of the U...	History	9	yes	Howard Zinn	Harper Perennial
978-0-525-47535-5	The Great Gatsby	Classic	8	yes	F. Scott Fitzgerald	Scribner
978-0-7432-4722-4	The Da Vinci Code	Mystery	8	yes	Dan Brown	Doubleday
978-0-7432-4722-5	Angels & Demons	Mystery	7.5	yes	Dan Brown	Doubleday

Task 12: Retrieve the List of Books Not Yet Returned

```
115 •  SELECT * FROM issued_status as ist
116    LEFT JOIN
117    return_status as rs
118    ON rs.issued_id = ist.issued_id
119    WHERE rs.return_id IS NULL;
120    ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

issued_id	issued_member_id	issued_book_name	issued_date	issued_book_isbn	issued_emp_id	return_id	issued_id	return
IS122	C102	Fahrenheit 451	2024-03-26	978-0-451-52993-5	E109	NULL	NULL	NULL
IS123	C103	Dune	2024-03-27	978-0-345-39180-3	E109	NULL	NULL	NULL
IS124	C104	Where the Wild Things Are	2024-03-28	978-0-06-025492-6	E110	NULL	NULL	NULL
IS125	C105	The Kite Runner	2024-03-29	978-0-06-112241-5	E110	NULL	NULL	NULL
IS126	C105	Charlotte's Web	2024-03-30	978-0-06-440055-8	E110	NULL	NULL	NULL
IS127	C105	Beloved	2024-03-31	978-0-679-77644-3	E110	NULL	NULL	NULL
IS128	C105	A Tale of Two Cities	2024-04-01	978-0-14-027526-3	E110	NULL	NULL	NULL
IS129	C105	The Stand	2024-04-02	978-0-7434-7679-3	E110	NULL	NULL	NULL

Advanced Operations

Task 13: Identify Members with Overdue Books(overdue>30)

```sql
SELECT
    ist.issued_member_id,
    m.member_name,
    bk.book_title,
    ist.issued_date,
    IF(DATEDIFF(CURRENT_DATE, ist.issued_date) > 30,
    DATEDIFF(CURRENT_DATE, ist.issued_date), 0) AS over_dues_days
FROM issued_status AS ist
JOIN members AS m ON m.member_id = ist.issued_member_id
JOIN books AS bk ON bk.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 DATEDIFF(CURRENT_DATE, ist.issued_date) > 30
    order by 1 desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

issued_member_id	member_name	book_title	issued_date	over_dues_days
C110	Jack Wilson	Animal Farm	2024-04-13	290
C109	Ivy Martinez	Harry Potter and the Sorc...	2024-04-12	291
C108	Henry Anderson	The Great Gatsby	2024-04-11	292
C107	Grace Taylor	Angels & Demons	2024-04-06	297
C107	Grace Taylor	The Diary of a Young Girl	2024-04-07	296
C107	Grace Taylor	Sapiens: A Brief History of...	2024-04-08	295
C107	Grace Taylor	1491: New Revelations of ...	2024-04-09	294
C107	Grace Taylor	The Catcher in the Rye	2024-04-10	293

Task14.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.

```sql
144 •   create table branch_report as
145     SELECT
146         b.branch_id,
147         b.manager_id,
148         COUNT(ist.issued_id) as number_book_issued,
149         COUNT(rs.return_id) as number_of_book_return,
150         SUM(bk.rental_price) as total_revenue
151     FROM issued_status as ist
152     JOIN
153     employees as e
154     ON e.emp_id = ist.issued_emp_id
155     JOIN branch as b
156     ON e.branch_id = b.branch_id
157     LEFT JOIN return_status as rs
158     ON rs.issued_id = ist.issued_id
159     JOIN
160     books as bk
161     ON ist.issued_book_isbn = bk.isbn
162     GROUP BY 1, 2;
163 •   SELECT * FROM branch_report;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

branch_id	manager_id	number_book_issued	number_of_book_return	total_revenue
B001	E109	16	8	105
B005	E110	9	3	50
B004	E110	4	3	26.5
B003	E109	1	0	5.5
B002	E109	2	0	12

Task 15: 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.

```sql
SELECT
    e.emp_name,
    COUNT(ist.issued_id) as no_book_issued,
    b.branch_id
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, b.branch_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

emp_name	no_book_issued	branch_id
Emily Davis	4	B001
Sarah Brown	4	B001
Michelle Ramirez	6	B001
Michael Thompson	3	B005
Jessica Taylor	4	B004
Daniel Anderson	2	B003
Laura Martinez	6	B005
John Doe	2	B001

Result 1

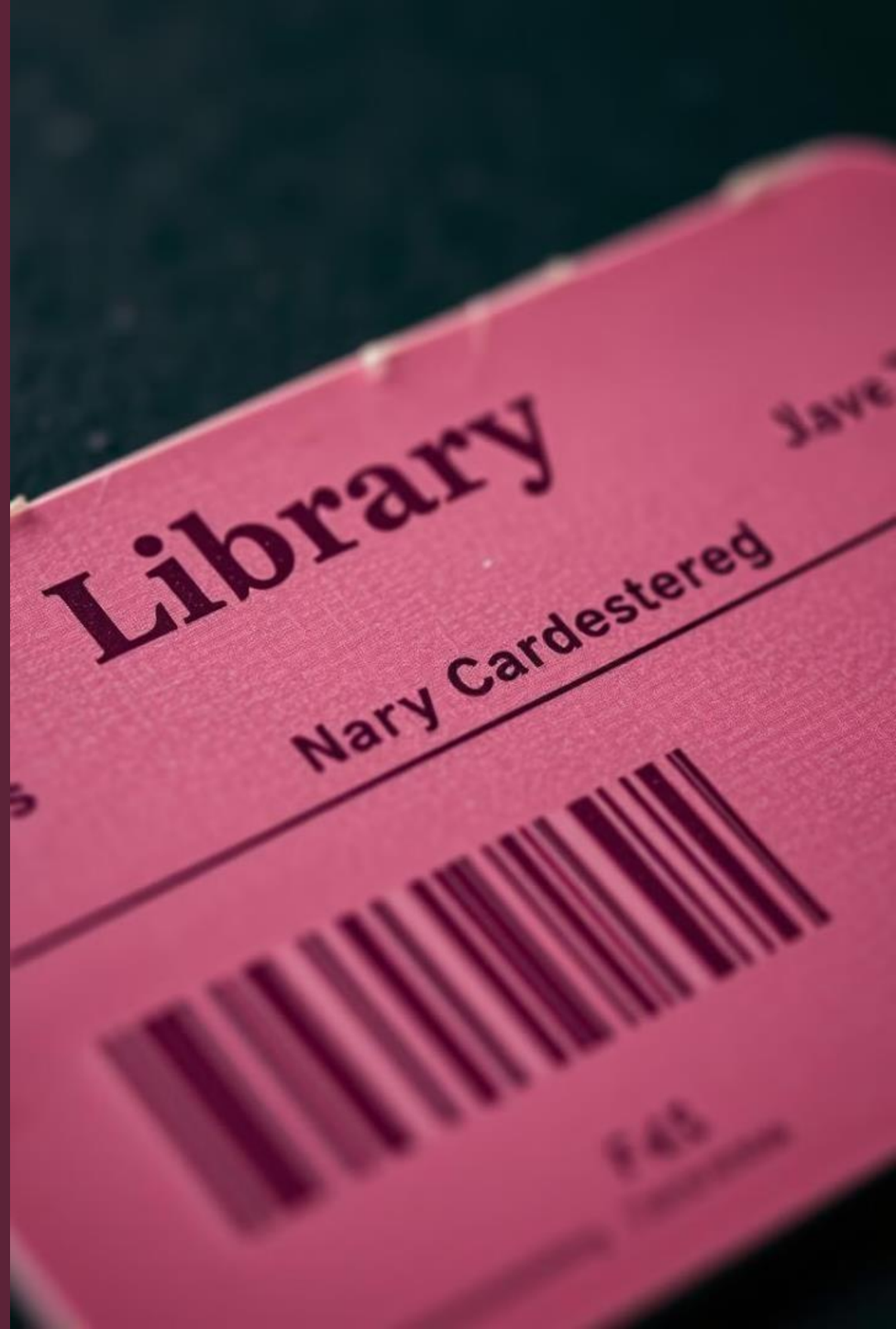
Insights

- ✓ Firstly, I Understood the dataset , explored all the tables ,their columns and relationship between all the tables and structured ER Diagram.
- ✓ Performed queries to create all the tables, inserted values in those tables using Insert Command.
- ✓ Performed CRUD operation to explore and understand Dataset and practiced CRUD operation.
- ✓ Performed CTAS (Create Table AS Select)
- ✓ In task7. found all the books details according to specific category.
- ✓ In task8. found Total rental income of the books according do different books category by using Joins.
- ✓ In tassk9. I found the list of 3 members who have registered in last 180 days.
- ✓ In task10. I used Joins between 3 tables to find out List Employees with Their Branch Manager's Name and their branch details.
- ✓ In task 11. I have Created a Table of Books with Rental Price Above a Certain Threshold using Create Statement and where clause.
- ✓ In task 12. I found the list of members id and details who have yet not returned thebooks

- ✓ In task13. I found the members details who has the overdue days of books using the advanced SQL commands .
- ✓ In task14 & task15. I used Advanced SQL queries to get the insights.

Conclusion

This project served as a model for how data analysis can improve library operations and given me hands-on practice by querying and analyzing the real-world dataset , It highlights the value of data-driven decision making in modern libraries.



Thank You!

---- Gitanjali Pekamwar