**Library Data Analysis**

**/\* CRUD Operations**

**Create: Inserted sample records into the books table.**

**Read: Retrieved and displayed data from various tables.**

**Update: Updated records in the employees table.**

**Delete: Removed records from the members table as needed. \*/**

**-- 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.');

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

UPDATE members

SET member\_address = '125 Oak St'

WHERE member\_id = 'C103';

**-- 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';

**-- Task 4: Retrieve All Books Issued by a Specific Employee -- Objective: Select all books issued by the employee with emp\_id = 'E101'.**

select \* from employees

where emp\_id = 'E101';

**-- 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 \* from members;

select \* from issued\_status;

select m.member\_name,m.member\_id, count( i.issued\_book\_name) as No\_of\_book\_issued from members as m

join issued\_status as i

on m.member\_id = i.issued\_member\_id

group by 1,2

having count( i.issued\_book\_name)>1;

**/\* 3. CTAS (Create Table As Select)**

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

create table Total\_issued\_book\_count as

select b.book\_title , b.isbn, count(i.issued\_book\_name)as No\_book\_issued

from books as b join issued\_status as i

on b.isbn= i.issued\_book\_isbn

group by 1,2

order by 2 desc;

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

select \* from books;

select \* from issued\_status;

select \* from books

where category = 'Fiction';

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

select \* from books;

select \* from issued\_status;

select category, COUNT(\*),sum(rental\_price) as Total\_rental\_income

from books

group by 1;

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

SELECT \* FROM members

WHERE reg\_date >= CURRENT\_DATE - INTERVAL '180 days';

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

select (e.emp\_name)as branch\_manager, e.branch\_id,b.manager\_id,b.branch\_address

from employees as e

join branch as b

on e.branch\_id = b. branch\_id

where e.position ='Manager';

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

CREATE TABLE expensive\_books AS

SELECT \* FROM books

WHERE rental\_price > 7.00;

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

select \* from return\_status;

select \* from issued\_status;

select i.issued\_book\_isbn from issued\_status as i

left join return\_status as r

on i.issued\_id = r.issued\_id

where r.issued\_id is null;

select i.issued\_book\_name,i.issued\_member\_id from issued\_status as i

left join return\_status as r

on i.issued\_id = r.issued\_id

where r.return\_id is null;