

```
-- Total Number of Books in the Library
SELECT COUNT(*) AS total_books FROM books;
--Total books are: 35

-- Total Number of Members in the Library
SELECT COUNT(*) AS total_members FROM members;
-- There are total 12 Members

-- Total Number of Employees in the Library
SELECT COUNT(*) AS total_employee FROM employees;
-- There are total 11 employee

-- Total Number of Branches
SELECT COUNT(*) AS total_branches FROM branch;
-- So there are total 5 branches available

-- Books Currently available for rent

SELECT COUNT(*) AS available_books FROM books
WHERE status = 'Yes';
-- Currently 32 books are available

-- Find out the book with the highest rental price.
SELECT book_title, rental_price FROM books
ORDER BY rental_price DESC
LIMIT 1;

-- Note: As here it Tried to use LIMIT, I Got error because SQL server does not support LIMIT so instead of using is
-- I'll now use TOP (n) for desired result

SELECT TOP 1 book_title, rental_price
FROM books
ORDER BY rental_price DESC;
-- A People History of the united states is the highest rental price book.

-- Total Revenue from Book Rentals
SELECT SUM(rental_price)AS total_revenue FROM books
WHERE status = 'Yes';
-- Total Revenue is $200

-- Which book has been borrowed the most (based on issue records)?
SELECT TOP 1 issued_book_name, COUNT(*) AS borrow_count
FROM issued_status
GROUP BY issued_book_name
ORDER BY borrow_count DESC;
-- Most issued books is "Aminam Farm"
```

```
-- Find which member has issued the most books.
SELECT TOP 1 issued_member_id, COUNT(*) AS borrow_count
FROM issued_status
GROUP BY issued_member_id
ORDER BY borrow_count DESC;
-- Lvy Martinez | C109

-- Which branch has the most issued books? or Most Active Branch
SELECT TOP 1 br.branch_id, COUNT(i.issued_id) AS total_issues
FROM issued_status i
JOIN books b ON i.issued_book_isbn = b.isbn
JOIN employees e ON i.issued_emp_id = e.emp_id
JOIN branch br ON e.branch_id = br.branch_id
GROUP BY br.branch_id
ORDER BY total_issues DESC;
-- So Branch ID B001 is the branch with the most issued books: 17

-- Which employee has issued the most books?

SELECT TOP 1 e.emp_name, COUNT(s.issued_id) AS total_issues
FROM issued_status s
JOIN employees e ON s.issued_emp_id = e.emp_id
GROUP BY e.emp_name
ORDER BY total_issues DESC;
-- Laura Martinez is the employee to issue the most books

-- Find the number of books issued between two specific dates.
SELECT COUNT(*) AS total_books_issued FROM issued_status
WHERE issued_date BETWEEN '2024-03-18' AND '2024-04-12';
-- 26 books issued

-- Find the total number of books in each category.
SELECT category, COUNT(*) AS total_books FROM books
GROUP BY category;

-- List employees working in each branch along with their positions.
SELECT e.emp_name, b.branch_id, e.position FROM employees e
JOIN branch b
ON e.branch_id = b.branch_id;

-- Calculate the total rental income per book category.
SELECT b.category, SUM(b.rental_price) AS total_rental_income
```

```
FROM books b
JOIN issued_status i ON b.isbn = i.issued_book_isbn
GROUP BY b.category;
```

-- Identify the top 3 highest-paid employees.

```
SELECT TOP 3 emp_name, salary FROM employees
ORDER BY salary DESC;
```

-- Or We Can Use CTE and Window function

```
WITH RankedEmployees AS (
    SELECT emp_name, salary,
           ROW_NUMBER() OVER (ORDER BY salary DESC) AS row_num
    FROM employees
)
SELECT emp_name, salary
FROM RankedEmployees
WHERE row_num <= 3;
```

-- Find members who have not issued any books.

```
SELECT m.member_name AS members FROM members m
JOIN issued_status i
ON m.member_id = i.issued_member_id
WHERE i.issued_member_id IS NULL;
```

-- Means all the members have issued at least one book

-- List books that have never been issued.

```
SELECT book_title
FROM books
WHERE isbn NOT IN (SELECT issued_book_isbn FROM issued_status);
```

-- Calculate the average rental price of books per publisher.

```
SELECT publisher, AVG(rental_price) AS avg_rental_price
FROM books
GROUP BY publisher;
```

-- Find the number of books issued by each employee.

```
SELECT e.emp_name, COUNT(i.issued_id) AS total_books_issued
FROM employees e
JOIN issued_status i ON e.emp_id = i.issued_emp_id
GROUP BY e.emp_name;
```

-- Find the latest return date for each member.

```
SELECT r.issued_id, MAX(r.return_date) AS last_return_date
```

```
FROM return_status r
GROUP BY r.issued_id;

-- Retrieve all books along with their issued status (issued or available).
SELECT b.book_title,
       CASE
         WHEN i.issued_book_isbn IS NOT NULL THEN 'Issued'
         ELSE 'Available'
       END AS book_status
FROM books b
LEFT JOIN issued_status i ON b.isbn = i.issued_book_isbn;

-- Find the number of books issued per month.
SELECT MONTH(issued_date) AS month, COUNT(*) AS books_issued
FROM issued_status
GROUP BY MONTH(issued_date)
ORDER BY month;

-- Find the members who issued more than 5 books.

SELECT issued_member_id, COUNT(*) AS total_books_issued
FROM issued_status
GROUP BY issued_member_id
HAVING COUNT(*) > 5;

-- Identify overdue books (assuming a 30-day return policy).
SELECT i.issued_book_name, m.member_name, i.issued_date, r.return_date
FROM issued_status i
JOIN members m ON i.issued_member_id = m.member_id
LEFT JOIN return_status r ON i.issued_id = r.issued_id
WHERE DATEDIFF(DAY, i.issued_date, COALESCE(r.return_date, GETDATE())) > 30;

-- Identify books that have been issued but not returned
SELECT issued_book_isbn
FROM issued_status

EXCEPT

SELECT return_book_isbn
FROM return_status;

-- Use of SET OPERATORS

-- Find books that have been both issued and returned
SELECT issued_book_isbn
FROM issued_status
```

INTERSECT

```
SELECT return_book_isbn
FROM return_status;
```

-- List all distinct books that have been issued or returned

```
SELECT issued_book_isbn
FROM issued_status
```

UNION

```
SELECT return_book_isbn
FROM return_status;
```

-- Identify overdue books using EXCEPT with date conditions

```
SELECT issued_book_isbn
FROM issued_status
WHERE DATEDIFF(DAY, issued_date, GETDATE()) > 30
```

EXCEPT

```
SELECT return_book_isbn
FROM return_status;
```