

# DIGITAL LIBRARY

## SQL QUERIES USED TO RETRIVE AND MANAGE BOOKS, STUDENTS

### 1. List of all books currently issued

```
SELECT b.title, s.name AS student_name, i.issue_date, i.return_date
FROM IssueLogs i
JOIN Books b ON i.book_id = b.book_id
JOIN Students s ON i.student_id = s.student_id
WHERE i.return_date IS NULL;
```

### 2. Books issued today

```
SELECT b.title, s.name, i.issue_date
FROM IssueLogs i
JOIN Books b ON i.book_id = b.book_id
JOIN Students s ON i.student_id = s.student_id
WHERE i.issue_date = CURDATE();
```

### 3. Books overdue (not returned past 14 days

```
SELECT b.title, s.name, i.issue_date, i.return_date
FROM IssueLogs i
JOIN Books b ON i.book_id = b.book_id
JOIN Students s ON i.student_id = s.student_id
WHERE i.return_date IS NULL AND i.issue_date < CURDATE() - INTERVAL 14
DAY;
```

### 4. All books issued by a specific student

```
SELECT b.title, i.issue_date, i.return_date
FROM IssueLogs i
JOIN Books b ON i.book_id = b.book_id
WHERE i.student_id = 1; -- change to student ID
```

### 5. Count of available books in each category

```
SELECT category, SUM(available_copies) AS available
FROM Books
GROUP BY category;
```

**6. Monthly issue report**

```
SELECT MONTH(i.issue_date) AS month, COUNT(*) AS total_issued
FROM IssueLogs i
GROUP BY MONTH(i.issue_date)
ORDER BY month;
```

**7. List all books**

```
SELECT * FROM Books;
```

**8. Find a student by name**

```
SELECT * FROM Students
WHERE name LIKE '%Ravi%';
```

**9. Get total number of available books**

```
SELECT SUM(available_copies) AS total_available FROM Books;
```

**10. Check which students haven't returned a specific book**

```
SELECT s.name, i.issue_date
FROM IssueLogs i
JOIN Students s ON i.student_id = s.student_id
WHERE i.book_id = 105 AND i.return_date IS NULL;
```

**11. List unique book categories**

```
SELECT DISTINCT category FROM Books;
```