

## Root1SQL Assignment

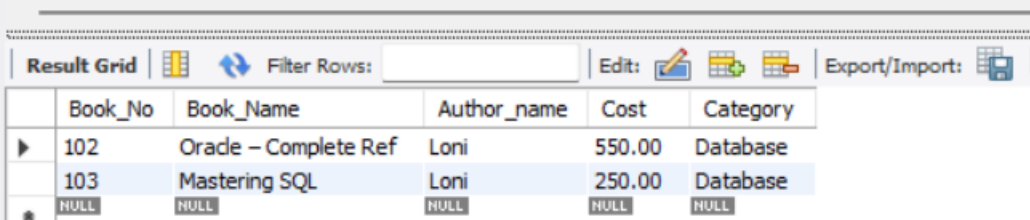
### Library System

Use the Schema from previous assignment to perform the given set of queries.

1) List all the books that are written by Author Loni and has price

less then 600.

```
3 • SELECT * FROM books WHERE author_name = 'Loni' AND cost < 600;
4
```



The screenshot shows a database query result in a 'Result Grid' window. The query is: `SELECT * FROM books WHERE author_name = 'Loni' AND cost < 600;`. The result table has columns: Book\_No, Book\_Name, Author\_name, Cost, and Category. It contains two rows: one for 'Orade - Complete Ref' with a cost of 550.00, and another for 'Mastering SQL' with a cost of 250.00. Both are in the 'Database' category. The bottom row shows NULL values for all columns.

Book_No	Book_Name	Author_name	Cost	Category
102	Orade - Complete Ref	Loni	550.00	Database
103	Mastering SQL	Loni	250.00	Database
NULL	NULL	NULL	NULL	NULL

2) List the Issue details for the books that are not returned yet.

`SELECT * FROM issue WHERE return_date IS NULL;`

3) Update all the blank return\_date with 31-Dec-06 excluding 7005

and 7006.

`update issue set return_date = '2006-12-31' where return_date is NULL and lib_issue_id not in (7005,7006);`

4) List all the Issue details that have books issued for more then 30

days.

`SELECT * FROM issue WHERE DATEDIFF(CURDATE(), IFNULL(return_date, CURDATE())) > 30;`

5) List all the books that have price in range of 500 to 750 and has

category as Database.

`SELECT * FROM books WHERE cost BETWEEN 500 AND 750 AND category = 'Database';`

6) List all the books that belong to any one of the following categories Science, Database, Fiction, Management.

```
SELECT * FROM books Where category IN('Science','DataBase','Friction');
```

7) List all the members in the descending order of Penalty due on

them.

```
SELECT * FROM Member order by Penalty_Amount desc;
```

8) Modify the price of book with id 103 to Rs 300 and category to

RDBMS.

```
UPDATE books Set cost = 300,Category = 'RDBMS' where book_no = 101;
```

9) List all the books in ascending order of category and descending

order of price.

```
SELECT * FROM books order by category asc , cost desc;
```

10) List all the books that contain word SQL in the name of the book.

```
SELECT * FROM books Where book_name like '%SQL%';
```

11) List the Lib\_Issue\_Id, Issue\_Date, Return\_Date and No of days

Book was issued.

```
SELECT lib_issue_id, issue_date, return_date, DATEDIFF(IFNULL(return_date, CURDATE()), issue_date) AS days_issued FROM issue;
```

12) Find the details of the member of the Library in the order of their joining the library.

```
SELECT * FROM member ORDER BY ACC_Open_date;
```

13) Display the count of total no of books issued to Member 101.

```
SELECT COUNT(*) FROM issue WHERE member_id = 101;
```

14) Display the total penalty due for all the members.

```
SELECT SUM(Penalty_AMOUNT) As TOTAl_Penalty from member;
```

15) Display the total no of members

```
Select count(*) from member;
```

**16) Display the total no of books issued**

Select count(\*) from Issue;

**17) Display the average membership fees paid by all the members**

Select AVG(fees\_paid) from member;

**18) List the various categories and count of books in each category.**

SELECT category, COUNT(\*) FROM books GROUP BY category;

**19) List the book\_No and the number of times the book is issued in the descending order of count.**

SELECT book\_no, count(\*) as ISSUE\_date from issue group by book\_no order by issue\_Date desc;

**20) Find the maximum, minimum, total and average penalty amount in the member table.**

SELECT MAX(penalty\_amount), MIN(penalty\_amount), SUM(penalty\_amount),  
AVG(penalty\_amount) FROM member;

**21) Display the member id and the no of books for each member that has issued more then 2 books.**

SELECT member\_id ,count(\*) as book\_issue from issue group by member\_id having count(\*) >= 2;

**22) Display the member id, book no and no of times the same book is issued by the member in the descending order of count.**

Select member\_id, book\_no, count(\*) As issue\_count from issue group by member\_id, book\_no  
order by issue\_count desc;

**23) Display the month and no of books issued each month in the descending order of count.**

Select Month(issue\_date) as issue\_month ,count(\*) as count from issue group by  
Month(issue\_date) order by count desc;

**24) List the book\_no of all the books that are not issued to any member so far.**

SELECT book\_no FROM books WHERE book\_no NOT IN (SELECT DISTINCT book\_no FROM issue);

**25) List all the member id that exist in member table and has also at least one book issued by them.**

SELECT DISTINCT member\_id FROM member WHERE member\_id IN (SELECT DISTINCT member\_id  
FROM issue);

**26) List the member ID with highest and lowest no of books issued.**

```
SELECT member_id, COUNT(*) AS issue_count FROM issue GROUP BY member_id ORDER BY issue_count DESC LIMIT 1;
```

```
SELECT member_id, COUNT(*) AS issue_count FROM issue GROUP BY member_id ORDER BY issue_count ASC LIMIT 1;
```

**27) List all the Issue\_details for books issued in December and July**

**without using any arithmetic, Logical or comparison operator.**

```
SELECT * FROM issue WHERE FORMAT(issue_date, 'MMMM') IN ('December', 'July');
```

**28) List the Book\_No, Book\_Name and Issue\_date for all the books**

**that are issued in month of December and belong to category Database.**

```
SELECT b.book_no, b.book_name, i.issue_date FROM books b JOIN issue i ON b.book_no = i.book_no WHERE MONTH(i.issue_date) = 12 AND b.category = 'Database';
```

**29) List the Member Id, Member Name and max books allowed in the**

**descending order of the max books allowed.**

```
SELECT Member_id, Member_name, Max_books_Allowed from member order by max_books_allowed desc;
```

**30) List the Book No, Book Name, Issue\_date and Return\_Date for all**

**the books issued by Richa Sharma.**

```
SELECT i.book_no, b.book_name, i.issue_date, i.return_date FROM issue i JOIN books b ON i.book_no = b.book_no JOIN member m ON i.member_id = m.member_id WHERE m.member_name = 'Richa Sharma';
```

**31) List the details of all the members that have issued books in**

**Database category.**

```
SELECT DISTINCT m.* FROM member m JOIN issue i ON m.member_id = i.member_id JOIN books b ON i.book_no = b.book_no WHERE b.category = 'Database';
```

**32) List all the books that have highest price in their own category.**

```
SELECT * FROM books b1 WHERE cost = (SELECT MAX(cost) FROM books b2 WHERE b2.category = b1.category);
```

**33) List all the Issue\_Details where Issue\_date is not within the**

**Acc\_open\_date and Return\_date for that member.**

```
SELECT * FROM issue i JOIN member m ON i.member_id = m.member_id WHERE i.issue_date NOT BETWEEN m.acc_open_date AND IFNULL(i.return_date, CURDATE());
```

**34) List all the members that have not issued a single book so far.**

```
SELECT * FROM member WHERE member_id NOT IN (SELECT DISTINCT member_id FROM issue);
```

**35) List all the members that have issued the same book as issued by**

**Garima.**

```
SELECT DISTINCT i2.member_id FROM issue i1 JOIN issue i2 ON i1.book_no = i2.book_no WHERE  
i1.member_id = (SELECT member_id FROM member WHERE member_name = 'Garima') AND  
i2.member_id != i1.member_id;
```

**36) List the Book\_Name, Price of all the books that are not returned**

**for more then 30 days.**

```
SELECT b.book_name, b.cost FROM books b JOIN issue i ON b.book_no = i.book_no WHERE  
i.return_date IS NULL AND DATEDIFF(CURDATE(), i.issue_date) > 30;
```

**37) List all the authors and book\_name that has more then 1 book**

**written by them.**

```
SELECT author_name, book_name FROM books WHERE author_name IN (SELECT author_name  
FROM books GROUP BY author_name HAVING COUNT(*) > 1);
```

**38) List the Member ID, Member Name of the people that have issued**

**the highest and the lowest no of books.**

```
SELECT m.member_id, m.member_name FROM members m JOIN (SELECT member_id FROM issue  
GROUP BY member_id ORDER BY COUNT(*) DESC LIMIT 1) AS high ON m.member_id =  
high.member_id
```

UNION

```
SELECT m.member_id, m.member_name FROM members m JOIN ( SELECT member_id FROM issue  
GROUP BY member_id ORDER BY COUNT(*) ASC LIMIT 1 ) AS low ON m.member_id =  
low.member_id;
```

**39) List the details of highest 3 priced books.**

```
SELECT * FROM books ORDER BY cost DESC LIMIT 3;
```

**40) List the total cost of all the books that are currently issued but not  
returned.**

```
SELECT SUM(cost) FROM books WHERE book_no IN (SELECT book_no FROM issue WHERE  
return_date IS NULL);
```

**41) List the details of the book that has been issued maximum no of times.**

```
SELECT * FROM books WHERE book_no = (SELECT book_no FROM issue GROUP BY book_no ORDER BY COUNT(*) DESC LIMIT 1);
```

**42) List how many books are issued to lifetime members.**

```
SELECT COUNT(*) FROM issue i JOIN member m ON i.member_id = m.member_id WHERE m.membership_type = 'Lifetime';
```

**43) List all member types and how many members are there in each type.**

```
SELECT membership_type, COUNT(*) FROM member GROUP BY membership_type;
```

**44) List first 5 members who had joined library.**

```
SELECT * FROM member ORDER BY ACC_OPEN_DATE LIMIT 5;
```

**45) List the members with their member type, who have issued books**

**during the period 1st December to 31st December.**

```
SELECT m.member_id, m.member_name, m.membership_type FROM member m JOIN issue i ON m.member_id = i.member_id WHERE i.issue_date BETWEEN '2006-12-01' AND '2006-12-31';
```

**46) List all the members who have not returned books yet.**

```
SELECT DISTINCT m.* FROM member m JOIN issue i ON m.member_id = i.member_id WHERE i.return_date IS NULL;
```

**47) List all the members who joined library on the same date Garima**

**joined.**

```
SELECT * FROM member WHERE ACC_OPEN_DATE = (SELECT ACC_OPEN_DATE FROM member WHERE member_name = 'Garima');
```

**48) List all the members who has issued books from author "Loni" in the month of December**

```
SELECT DISTINCT m.* FROM member m JOIN issue i ON m.member_id = i.member_id JOIN books b ON i.book_no = b.book_no WHERE MONTH(i.issue_date) = 12 AND b.author_name = 'Loni';
```

**49) List names of the authors whose books are least issued by lifetime members.**

```
SELECT author_name FROM books b JOIN issue i ON b.book_no = i.book_no JOIN member m ON i.member_id = m.member_id WHERE m.membership_type = 'Lifetime' GROUP BY author_name ORDER BY COUNT(*) ASC;
```

**50) List the names of members who has issued the books whose cost is more than 300 rupees and whose author is "Scott Urman"**

```
SELECT m.member_name FROM member m JOIN issue i ON m.member_id = i.member_id JOIN books b ON i.book_no = b.book_no WHERE b.author_name = 'Scott Urman' AND b.cost > 300;
```

**51) List all lifetime members who joined library during 1st January 2006 to 31st December 2006 but issued only one book.**

```
SELECT m.* FROM member m JOIN issue i ON m.member_id = i.member_id WHERE m.membership_type = 'Lifetime' AND m.ACC_Open_date BETWEEN '2006-01-01' AND '2006-12-31' GROUP BY m.member_id HAVING COUNT(i.book_no) = 1;
```

**52) Modify the Penalty\_Amount for Garima Sen to Rs 100.**

```
UPDATE member SET penalty_amount = 100 WHERE member_name = 'Garima Sen';
```