

Library Management System

By

AYUSHI ROY (RA2211026010439)

ANIRUDH ARCOT (RA2211026010449)

21CSC205P – Database Management Systems



**FACULTY OF ENGINEERING AND
TECHNOLOGYSCHOOL OF COMPUTING**

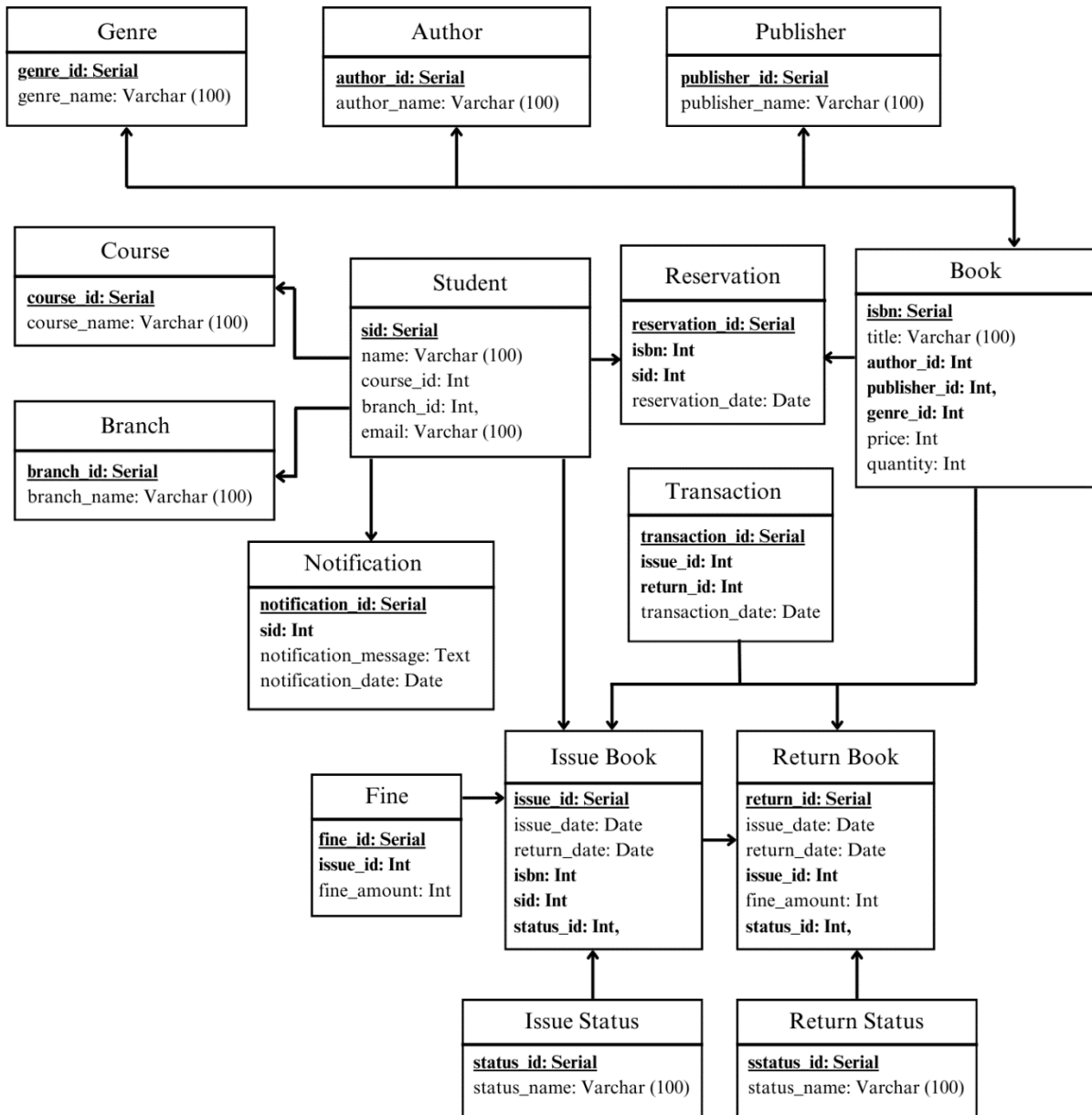
**SRM INSTITUTE OF SCIENCE AND
TECHNOLOGYKATTANKULATHUR**

March 2024

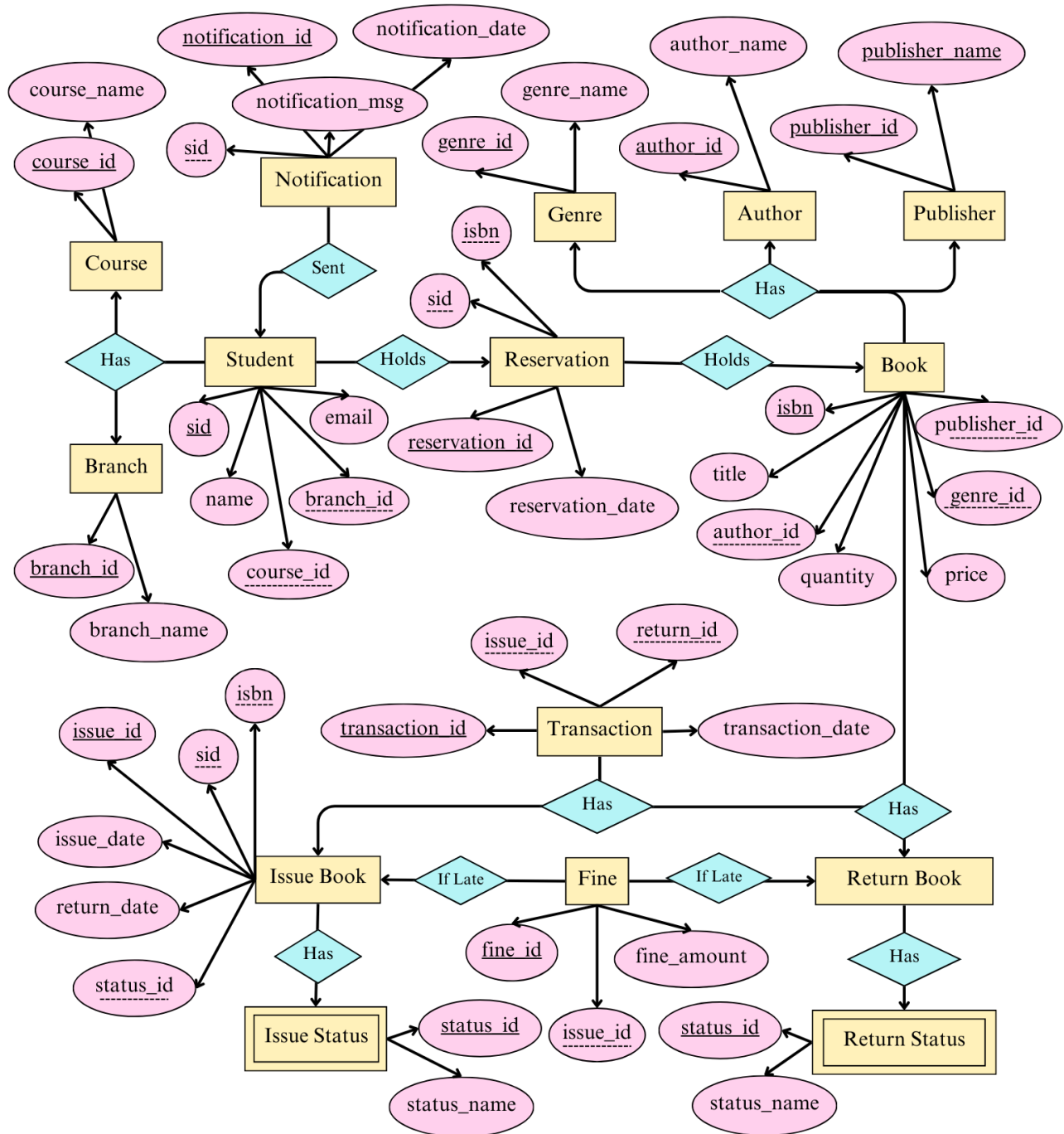
TABLE OF CONTENTS

| CHAPTER NO | CONTENTS | PAGE NO |
|-------------------|------------------------------------|----------------|
| 1 | Diagrams | 3 |
| | 1.1 UML Class Diagram | 3 |
| | 1.2 Entity Relationship Diagram | 4 |
| 2 | All Tables Used | 5-9 |
| 3 | 100 Queries | 10-50 |

I. UML Class Diagram



II. Entity – Relationship Diagram



III. Tables Used

i. Author

```
library_management_system=# SELECT * FROM author;
author_id | author_name
-----+-----
          1 | Jane Austen
          2 | George Orwell
          3 | Agatha Christie
          4 | Stephen King
          5 | J.K. Rowling
(5 rows)
```

ii. Publisher

```
library_management_system=# SELECT * FROM publisher;
publisher_id | publisher_name
-----+-----
             1 | Penguin Random House
             2 | HarperCollins
             3 | Simon & Schuster
             4 | Macmillan Publishers
             5 | Hachette Livre
(5 rows)
```

iii. Genre

```
library_management_system=# SELECT * FROM genre;
genre_id | genre_name
-----+-----
         1 | Fiction
         2 | Non-Fiction
         3 | Science Fiction
         4 | Mystery
         5 | Thriller
(5 rows)
```

iv. Book

```
library_management_system=# SELECT * FROM book;
```

| isbn | title | author_id | publisher_id | genre_id | price | quantity |
|------|--|-----------|--------------|----------|-------|----------|
| 1 | Pride and Prejudice | 1 | 1 | 1 | 10 | 5 |
| 2 | 1984 | 2 | 2 | 3 | 12 | 7 |
| 3 | Murder on the Orient Express | 3 | 3 | 4 | 8 | 3 |
| 4 | The Shining | 4 | 4 | 5 | 15 | 4 |
| 5 | Harry Potter and the Philosopher's Stone | 5 | 5 | 1 | 13 | 6 |

(5 rows)

v. Reservation

```
library_management_system=# SELECT * FROM reservation;
```

| reservation_id | isbn | sid | reservation_date |
|----------------|------|-----|------------------|
| 1 | 1 | 3 | 2024-03-10 |
| 2 | 2 | 4 | 2024-03-12 |
| 3 | 3 | 5 | 2024-03-14 |
| 4 | 4 | 1 | 2024-03-16 |
| 5 | 5 | 2 | 2024-03-18 |

(5 rows)

vi. Student

```
library_management_system=# SELECT * FROM student;
```

| sid | name | course_id | branch_id | email |
|-----|-------------|-----------|-----------|-------------------------|
| 1 | John Doe | 1 | 3 | john.doe@example.com |
| 2 | Alice Smith | 2 | 1 | alice.smith@example.com |
| 3 | Bob Johnson | 3 | 5 | bob.johnson@example.com |
| 4 | Emma Brown | 4 | 2 | emma.brown@example.com |
| 5 | Michael Lee | 5 | 4 | michael.lee@example.com |

(5 rows)

vii. Course

```
library_management_system=# SELECT * FROM course;
course_id |      course_name
-----+-----
          1 | Computer Science
          2 | English Literature
          3 | Biology
          4 | History
          5 | Physics
(5 rows)
```

viii. Branch

```
library_management_system=# SELECT * FROM branch;
branch_id |      branch_name
-----+-----
          1 | Computer Science
          2 | Mechanical
          3 | ECE
          4 | Civil Engineering
          5 | Electrical Engineering
(5 rows)
```

ix. Transaction

```
library_management_system=# SELECT * FROM transaction;
transaction_id | issue_id | return_id | transaction_date
-----+-----+-----+-----
              1 |         1 |          1 | 2024-03-15 10:30:00
              2 |         2 |          2 | 2024-03-18 11:45:00
              3 |         3 |          3 | 2024-03-20 09:20:00
              4 |         4 |          4 | 2024-03-22 13:10:00
              5 |         5 |          5 | 2024-03-24 14:55:00
(5 rows)
```

x. Issue Book

```
library_management_system=# SELECT * FROM issue_book;
```

| issue_id | isbn | sid | issue_date | return_date | status_id |
|----------|------|-----|------------|-------------|-----------|
| 1 | 1 | 1 | 2024-03-01 | 2024-03-15 | 1 |
| 2 | 2 | 2 | 2024-03-03 | 2024-03-18 | 1 |
| 3 | 3 | 3 | 2024-03-05 | 2024-03-20 | 1 |
| 4 | 4 | 4 | 2024-03-07 | 2024-03-22 | 3 |
| 5 | 5 | 5 | 2024-03-09 | 2024-03-24 | 1 |

(5 rows)

xi. Return Book

```
library_management_system=# SELECT * FROM return_book;
```

| return_id | issue_id | return_date | fine_amount | status_id |
|-----------|----------|-------------|-------------|-----------|
| 1 | 1 | 2024-03-15 | 0 | 1 |
| 2 | 2 | 2024-03-18 | 0 | 1 |
| 3 | 3 | 2024-03-20 | 0 | 1 |
| 4 | 4 | 2024-03-22 | 5 | 2 |
| 5 | 5 | 2024-03-24 | 0 | 1 |

(5 rows)

xii. Issue Status

```
library_management_system=# SELECT * FROM issue_status;
```

| status_id | status_name |
|-----------|-------------|
| 1 | Issued |
| 2 | Returned |
| 3 | Overdue |
| 4 | Lost |
| 5 | Damaged |

(5 rows)

xiii. Return Status

```
library_management_system=# SELECT * FROM return_status;
```

| status_id | status_name |
|-----------|------------------|
| 1 | Returned on Time |
| 2 | Returned Late |
| 3 | Lost |
| 4 | Damaged |
| 5 | Renewed |

(5 rows)

xiv. Fine (in dollars)

```
library_management_system=# SELECT * FROM fine;
```

| fine_id | issue_id | fine_amount |
|---------|----------|-------------|
| 1 | 4 | 5 |
| 2 | 5 | 0 |
| 3 | 3 | 0 |
| 4 | 2 | 0 |
| 5 | 1 | 0 |

(5 rows)

xv. Notification

```
library_management_system=# SELECT * FROM notification;
```

| notification_id | sid | notification_message | notification_date |
|-----------------|-----|--|---------------------|
| 1 | 1 | Your book is overdue. | 2024-03-23 08:00:00 |
| 2 | 2 | Your book reservation is ready for pickup. | 2024-03-25 09:30:00 |
| 3 | 3 | Your book is overdue. | 2024-03-26 10:15:00 |
| 4 | 4 | Your book reservation is ready for pickup. | 2024-03-28 11:20:00 |
| 5 | 5 | Your book is overdue. | 2024-03-29 12:45:00 |

(5 rows)

IV. 100 Queries

SELECT STATEMENTS:

1. Retrieve all data from the student table:

SELECT * FROM student;

```
library_management_system=# SELECT * FROM student;
 sid |      name      | course_id | branch_id |      email
-----+-----+-----+-----+-----
   1 | John Doe       |         1 |         3 | john.doe@example.com
   2 | Alice Smith    |         2 |         1 | alice.smith@example.com
   3 | Bob Johnson    |         3 |         5 | bob.johnson@example.com
   4 | Emma Brown     |         4 |         2 | emma.brown@example.com
   5 | Michael Lee    |         5 |         4 | michael.lee@example.com
(5 rows)
```

2. Retrieve all data from the book table:

SELECT * FROM book;

```
library_management_system=# SELECT * FROM book;
 isbn |      title      | author_id | publisher_id | genre_id | price | quantity
-----+-----+-----+-----+-----+-----+-----
   1 | Pride and Prejudice |         1 |         1 |         1 |    10 |         5
   2 | 1984              |         2 |         2 |         3 |    12 |         7
   3 | Murder on the Orient Express |         3 |         3 |         4 |     8 |         3
   4 | The Shining        |         4 |         4 |         5 |    15 |         4
   5 | Harry Potter and the Philosopher's Stone |         5 |         5 |         1 |    13 |         6
(5 rows)
```

3. Retrieve all data from the issue_book table:

SELECT * FROM issue_book;

```
library_management_system=# SELECT * FROM issue_book;
 issue_id | isbn | sid | issue_date | return_date | status_id
-----+-----+-----+-----+-----+-----
       1 |    1 |   1 | 2024-03-01 | 2024-03-15 |         1
       2 |    2 |   2 | 2024-03-03 | 2024-03-18 |         1
       3 |    3 |   3 | 2024-03-05 | 2024-03-20 |         1
       4 |    4 |   4 | 2024-03-07 | 2024-03-22 |         3
       5 |    5 |   5 | 2024-03-09 | 2024-03-24 |         1
(5 rows)
```

4. Retrieve all data from the reservation table:

```
SELECT * FROM reservation;
```

```
library_management_system=# SELECT * FROM reservation;
```

| reservation_id | isbn | sid | reservation_date |
|----------------|------|-----|------------------|
| 1 | 1 | 3 | 2024-03-10 |
| 2 | 2 | 4 | 2024-03-12 |
| 3 | 3 | 5 | 2024-03-14 |
| 4 | 4 | 1 | 2024-03-16 |
| 5 | 5 | 2 | 2024-03-18 |

(5 rows)

5. Retrieve all data from the notification table:

```
SELECT * FROM notification;
```

```
library_management_system=# SELECT * FROM reservation;
```

| reservation_id | isbn | sid | reservation_date |
|----------------|------|-----|------------------|
| 1 | 1 | 3 | 2024-03-10 |
| 2 | 2 | 4 | 2024-03-12 |
| 3 | 3 | 5 | 2024-03-14 |
| 4 | 4 | 1 | 2024-03-16 |
| 5 | 5 | 2 | 2024-03-18 |

(5 rows)

WHERE:

6. Retrieve information about John Doe

```
SELECT * FROM student WHERE name = 'John Doe';
```

| sid | name | course_id | branch_id | email |
|-----|----------|-----------|-----------|----------------------|
| 1 | John Doe | 1 | 3 | john.doe@example.com |

(1 row)

7. Retrieve information about books with a price greater than \$10:

```
SELECT * FROM book WHERE price > 10;
```

```
library_management_system=# SELECT * FROM book WHERE price > 10;
 isbn |          title          | author_id | publisher_id | genre_id | price | quantity
-----+-----+-----+-----+-----+-----+-----
    2 | 1984                    |         2 |          2   |        3 |    12 |         7
    4 | The Shining              |         4 |          4   |        5 |    15 |         4
    5 | Harry Potter and the Philosopher's Stone |         5 |          5   |        1 |    13 |         6
(3 rows)
```

8. Retrieve information about issued books:

```
SELECT * FROM issue_book WHERE return_date IS NULL;
```

```
issue_id | isbn | sid | issue_date | return_date | status_id
-----+-----+-----+-----+-----+-----
(0 rows)
```

9. Retrieve information about students who have overdue books:

```
SELECT * FROM notification WHERE notification_message = 'Your book is overdue.';
```

```
notification_id | sid | notification_message | notification_date
-----+-----+-----+-----
            1 |   1 | Your book is overdue. | 2024-03-23 08:00:00
            3 |   3 | Your book is overdue. | 2024-03-26 10:15:00
            5 |   5 | Your book is overdue. | 2024-03-29 12:45:00
(3 rows)
```

10. Show information of students from branch id 1

```
SELECT * FROM student WHERE branch_id = 1;
```

```
library_management_system=# SELECT *
FROM student
WHERE branch_id = 1;
 sid |   name   | course_id | branch_id | email
-----+-----+-----+-----+-----
    2 | Alice Smith |         2 |          1 | alice.smith@example.com
(1 row)
```

GROUP BY:

11. Count of Books per Genre:

```
library_management_system=# SELECT genre_id, COUNT(*) AS num_books
FROM book
GROUP BY genre_id;
```

| genre_id | num_books |
|----------|-----------|
| 3 | 2 |
| 5 | 2 |
| 4 | 2 |
| 2 | 2 |
| 7 | 1 |
| 1 | 6 |

(6 rows)

12. Average Price of Books per Publisher:

```
library_management_system=# SELECT publisher_id, AVG(price) AS avg_price
FROM book
GROUP BY publisher_id;
```

| publisher_id | avg_price |
|--------------|--------------------|
| 9 | 15.000000000000000 |
| 3 | 11.000000000000000 |
| 5 | 16.500000000000000 |
| 4 | 12.500000000000000 |
| 10 | 13.000000000000000 |
| 6 | 10.000000000000000 |
| 2 | 11.500000000000000 |
| 7 | 12.000000000000000 |
| 1 | 11.000000000000000 |
| 8 | 11.000000000000000 |

(10 rows)

13. Maximum Price of Books per Genre:

```
library_management_system=# SELECT genre_id, MAX(price) AS max_price
FROM book
GROUP BY genre_id;
```

| genre_id | max_price |
|----------|-----------|
| 3 | 12 |
| 5 | 15 |
| 4 | 15 |
| 2 | 12 |
| 7 | 20 |
| 1 | 14 |

(6 rows)

14. Total Quantity of Books per Author:

```
library_management_system=# SELECT author_id, SUM(quantity) AS total_quantity
FROM book
GROUP BY author_id;
author_id | total_quantity
-----+-----
          9 |              7
          3 |             10
          5 |             16
          4 |              9
         10 |             10
          6 |              5
          2 |             13
          7 |              8
          1 |             13
          8 |              6
(10 rows)
```

15. Number of Students per Course:

```
library_management_system=# SELECT course_id, COUNT(*) AS num_students
FROM student
GROUP BY course_id;
course_id | num_students
-----+-----
          9 |            1
          3 |            1
          5 |            1
          4 |            1
         10 |            1
          6 |            1
          2 |            1
          7 |            1
          1 |            1
          8 |            1
(10 rows)
```


INNER JOIN:

16. Retrieve Books with Their Authors:

```
library_management_system=# SELECT book.title, author.author_name
FROM book
INNER JOIN author ON book.author_id = author.author_id;
```

| title | author_name |
|--|-----------------|
| Pride and Prejudice | Jane Austen |
| 1984 | George Orwell |
| Murder on the Orient Express | Agatha Christie |
| The Shining | Stephen King |
| Harry Potter and the Philosopher's Stone | J.K. Rowling |
| Emma | Maya Angelou |
| The Adventures of Huckleberry Finn | J.R.R. Tolkien |
| The Picture of Dorian Gray | Dale Carnegie |
| A Tale of Two Cities | Stephen King |
| The Lord of the Rings | Rupi Kaur |
| Wuthering Heights | Jane Austen |
| Moby Dick | George Orwell |
| Alice's Adventures in Wonderland | Agatha Christie |
| Frankenstein | Stephen King |
| Dracula | J.K. Rowling |

(15 rows)

17. Retrieve Students with Their Courses:

```
library_management_system=# SELECT student.name, course.course_name
FROM student
INNER JOIN course ON student.course_id = course.course_id;
```

| name | course_name |
|---------------------|------------------------|
| John Doe | Computer Science |
| Alice Smith | English Literature |
| Bob Johnson | Biology |
| Emma Brown | History |
| Michael Lee | Physics |
| Ethan Parker | Chemistry |
| Mia Thompson | Mechanical Engineering |
| Alexander Hernandez | Psychology |
| Madison Lopez | Political Science |
| James Garcia | Art History |

(10 rows)

18. Retrieve Issued Books with Their Return Status:

```
library_management_system=# SELECT issue_book.issue_date, return_status.status_name
FROM issue_book
INNER JOIN return_status ON issue_book.status_id = return_status.status_id;
```

| issue_date | status_name |
|------------|------------------|
| 2024-03-01 | Returned on Time |
| 2024-03-03 | Returned on Time |
| 2024-03-05 | Returned on Time |
| 2024-03-07 | Lost |
| 2024-03-09 | Returned on Time |

(5 rows)

19. Retrieve Books with Their Genres and Publishers:

```
library_management_system=# SELECT book.title, genre.genre_name, publisher.publisher_name
FROM book
INNER JOIN genre ON book.genre_id = genre.genre_id
INNER JOIN publisher ON book.publisher_id = publisher.publisher_id;
```

| title | genre_name | publisher_name |
|--|-----------------|----------------------|
| Pride and Prejudice | Fiction | Penguin Random House |
| 1984 | Science Fiction | HarperCollins |
| Murder on the Orient Express | Mystery | Simon & Schuster |
| The Shining | Thriller | Macmillan Publishers |
| Harry Potter and the Philosopher's Stone | Fiction | Hachette Livre |
| Emma | Fiction | Penguin Random House |
| The Adventures of Huckleberry Finn | Non-Fiction | HarperCollins |
| The Picture of Dorian Gray | Fiction | Simon & Schuster |
| A Tale of Two Cities | Fiction | Macmillan Publishers |
| The Lord of the Rings | Fantasy | Hachette Livre |
| Wuthering Heights | Fiction | Vintage Books |
| Moby Dick | Non-Fiction | Penguin Books |
| Alice's Adventures in Wonderland | Science Fiction | HarperCollins |
| Frankenstein | Mystery | Simon & Schuster |
| Dracula | Thriller | Random House |

(15 rows)

20. Retrieve Students with Their Branches:

```
library_management_system=# SELECT student.name, branch.branch_name
FROM student
INNER JOIN branch ON student.branch_id = branch.branch_id;
```

| name | branch_name |
|---------------------|------------------------------|
| John Doe | ECE |
| Alice Smith | Computer Science |
| Bob Johnson | Electrical Engineering |
| Emma Brown | Mechanical |
| Michael Lee | Civil Engineering |
| Ethan Parker | Chemical Engineering |
| Mia Thompson | Mechanical Engineering |
| Alexander Hernandez | Psychology Department |
| Madison Lopez | Political Science Department |
| James Garcia | Art Department |

(10 rows)

FULL OUTER JOIN:

21. Retrieve Authors with Their Books (including Authors with No Books and Books with No Authors):

```
library_management_system=# SELECT book.title, author.author_name
FROM author
FULL OUTER JOIN book ON author.author_id = book.author_id;
```

| title | author_name |
|--|-----------------|
| Pride and Prejudice | Jane Austen |
| 1984 | George Orwell |
| Murder on the Orient Express | Agatha Christie |
| The Shining | Stephen King |
| Harry Potter and the Philosopher's Stone | J.K. Rowling |
| Emma | Maya Angelou |
| The Adventures of Huckleberry Finn | J.R.R. Tolkien |
| The Picture of Dorian Gray | Dale Carnegie |
| A Tale of Two Cities | Stephen King |
| The Lord of the Rings | Rupi Kaur |
| Wuthering Heights | Jane Austen |
| Moby Dick | George Orwell |
| Alice's Adventures in Wonderland | Agatha Christie |
| Frankenstein | Stephen King |
| Dracula | J.K. Rowling |
| | H.C.Verma |

(16 rows)

22. Retrieve Courses with Their Students (including Courses with No Students and Students with No Courses):

```
library_management_system=# SELECT student.name, course.course_name
FROM course
FULL OUTER JOIN student ON course.course_id = student.course_id;
```

| name | course_name |
|---------------------|------------------------|
| John Doe | Computer Science |
| Alice Smith | English Literature |
| Bob Johnson | Biology |
| Emma Brown | History |
| Michael Lee | Physics |
| Ethan Parker | Chemistry |
| Mia Thompson | Mechanical Engineering |
| Alexander Hernandez | Psychology |
| Madison Lopez | Political Science |
| James Garcia | Art History |
| | Bio-Technology |

(11 rows)

23. Retrieve Return Statuses with Their Issued Books (including Return Statuses with No Issued Books and Issued Books with No Return Statuses):

```
library_management_system=# SELECT issue_book.issue_date, return_status.status_name
FROM return_status
FULL OUTER JOIN issue_book ON return_status.status_id = issue_book.status_id;
 issue_date |  status_name
-----+-----
2024-03-01 | Returned on Time
2024-03-03 | Returned on Time
2024-03-05 | Returned on Time
2024-03-07 | Lost
2024-03-09 | Returned on Time
            | Returned Late
            | Renewed
            | Damaged
(8 rows)
```

24. Retrieve Genres with Their Books (including Genres with No Books and Books with No Genres):

```
library_management_system=# SELECT book.title, genre.genre_name
FROM genre
FULL OUTER JOIN book ON genre.genre_id = book.genre_id;
          title          |  genre_name
-----+-----
Pride and Prejudice      | Fiction
1984                     | Science Fiction
Murder on the Orient Express | Mystery
The Shining              | Thriller
Harry Potter and the Philosopher's Stone | Fiction
Emma                     | Fiction
The Adventures of Huckleberry Finn | Non-Fiction
The Picture of Dorian Gray | Fiction
A Tale of Two Cities     | Fiction
The Lord of the Rings    | Fantasy
Wuthering Heights       | Fiction
Moby Dick                | Non-Fiction
Alice's Adventures in Wonderland | Science Fiction
Frankenstein             | Mystery
Dracula                  | Thriller
                        | Poetry
                        | Self-Help
                        | Biography
                        | Horror
(19 rows)
```

25. Retrieve Branches with Their Students (including Branches with No Students and Students with No Branches):

```
library_management_system=# SELECT student.name, branch.branch_name
FROM branch
FULL OUTER JOIN student ON branch.branch_id = student.branch_id;
```

| name | branch_name |
|---------------------|------------------------------|
| John Doe | ECE |
| Alice Smith | Computer Science |
| Bob Johnson | Electrical Engineering |
| Emma Brown | Mechanical |
| Michael Lee | Civil Engineering |
| Ethan Parker | Chemical Engineering |
| Mia Thompson | Mechanical Engineering |
| Alexander Hernandez | Psychology Department |
| Madison Lopez | Political Science Department |
| James Garcia | Art Department |

(10 rows)

LEFT JOIN:

26. Retrieve Books with Their Authors (including Orphan Books):

```
library_management_system=# SELECT book.title, author.author_name
FROM book
LEFT JOIN author ON book.author_id = author.author_id;
```

| title | author_name |
|--|-----------------|
| Pride and Prejudice | Jane Austen |
| 1984 | George Orwell |
| Murder on the Orient Express | Agatha Christie |
| The Shining | Stephen King |
| Harry Potter and the Philosopher's Stone | J.K. Rowling |
| Emma | Maya Angelou |
| The Adventures of Huckleberry Finn | J.R.R. Tolkien |
| The Picture of Dorian Gray | Dale Carnegie |
| A Tale of Two Cities | Stephen King |
| The Lord of the Rings | Rupi Kaur |
| Wuthering Heights | Jane Austen |
| Moby Dick | George Orwell |
| Alice's Adventures in Wonderland | Agatha Christie |
| Frankenstein | Stephen King |
| Dracula | J.K. Rowling |

(15 rows)

27. Retrieve Students with Their Courses (including Students without Courses):

```
library_management_system=# SELECT student.name, course.course_name
FROM student
LEFT JOIN course ON student.course_id = course.course_id;
```

| name | course_name |
|---------------------|------------------------|
| John Doe | Computer Science |
| Alice Smith | English Literature |
| Bob Johnson | Biology |
| Emma Brown | History |
| Michael Lee | Physics |
| Ethan Parker | Chemistry |
| Mia Thompson | Mechanical Engineering |
| Alexander Hernandez | Psychology |
| Madison Lopez | Political Science |
| James Garcia | Art History |

(10 rows)

28. Retrieve Issued Books with Their Return Status (including Unreturned Books):

```
library_management_system=# SELECT issue_book.issue_date, return_status.status_name
FROM issue_book
LEFT JOIN return_status ON issue_book.status_id = return_status.status_id;
```

| issue_date | status_name |
|------------|------------------|
| 2024-03-01 | Returned on Time |
| 2024-03-03 | Returned on Time |
| 2024-03-05 | Returned on Time |
| 2024-03-07 | Lost |
| 2024-03-09 | Returned on Time |

(5 rows)

29. Retrieve Books with Their Genres and Publishers (including Books with Missing Genre/Publisher):

```
library_management_system=# SELECT book.title, genre.genre_name, publisher.publisher_name
FROM book
LEFT JOIN genre ON book.genre_id = genre.genre_id
LEFT JOIN publisher ON book.publisher_id = publisher.publisher_id;
```

| title | genre_name | publisher_name |
|--|-----------------|----------------------|
| Pride and Prejudice | Fiction | Penguin Random House |
| 1984 | Science Fiction | HarperCollins |
| Murder on the Orient Express | Mystery | Simon & Schuster |
| The Shining | Thriller | Macmillan Publishers |
| Harry Potter and the Philosopher's Stone | Fiction | Hachette Livre |
| Emma | Fiction | Penguin Random House |
| The Adventures of Huckleberry Finn | Non-Fiction | HarperCollins |
| The Picture of Dorian Gray | Fiction | Simon & Schuster |
| A Tale of Two Cities | Fiction | Macmillan Publishers |
| The Lord of the Rings | Fantasy | Hachette Livre |
| Wuthering Heights | Fiction | Vintage Books |
| Moby Dick | Non-Fiction | Penguin Books |
| Alice's Adventures in Wonderland | Science Fiction | HarperCollins |
| Frankenstein | Mystery | Simon & Schuster |
| Dracula | Thriller | Random House |

(15 rows)

30. Retrieve Students with Their Branches (including Students without Branches):

```
library_management_system=# SELECT student.name, branch.branch_name
FROM student
LEFT JOIN branch ON student.branch_id = branch.branch_id;
```

| name | branch_name |
|---------------------|------------------------------|
| John Doe | ECE |
| Alice Smith | Computer Science |
| Bob Johnson | Electrical Engineering |
| Emma Brown | Mechanical |
| Michael Lee | Civil Engineering |
| Ethan Parker | Chemical Engineering |
| Mia Thompson | Mechanical Engineering |
| Alexander Hernandez | Psychology Department |
| Madison Lopez | Political Science Department |
| James Garcia | Art Department |

(10 rows)

RIGHT JOIN:

31. Retrieve Authors with Their Books (including Authors with No Books):

```
library_management_system=# SELECT book.title, author.author_name
FROM author
RIGHT JOIN book ON author.author_id = book.author_id;
```

| title | author_name |
|--|-----------------|
| Pride and Prejudice | Jane Austen |
| 1984 | George Orwell |
| Murder on the Orient Express | Agatha Christie |
| The Shining | Stephen King |
| Harry Potter and the Philosopher's Stone | J.K. Rowling |
| Emma | Maya Angelou |
| The Adventures of Huckleberry Finn | J.R.R. Tolkien |
| The Picture of Dorian Gray | Dale Carnegie |
| A Tale of Two Cities | Stephen King |
| The Lord of the Rings | Rupi Kaur |
| Wuthering Heights | Jane Austen |
| Moby Dick | George Orwell |
| Alice's Adventures in Wonderland | Agatha Christie |
| Frankenstein | Stephen King |
| Dracula | J.K. Rowling |

(15 rows)

32. Retrieve Courses with Their Students (including Courses with No Students):

```
library_management_system=# SELECT student.name, course.course_name
FROM course
RIGHT JOIN student ON course.course_id = student.course_id;
```

| name | course_name |
|---------------------|------------------------|
| John Doe | Computer Science |
| Alice Smith | English Literature |
| Bob Johnson | Biology |
| Emma Brown | History |
| Michael Lee | Physics |
| Ethan Parker | Chemistry |
| Mia Thompson | Mechanical Engineering |
| Alexander Hernandez | Psychology |
| Madison Lopez | Political Science |
| James Garcia | Art History |

(10 rows)

33. Retrieve Return Statuses with Their Issued Books (including Return Statuses with No Issued Books):

```
library_management_system=# SELECT issue_book.issue_date, return_status.status_name
FROM return_status
RIGHT JOIN issue_book ON return_status.status_id = issue_book.status_id;
 issue_date |    status_name
-----+-----
2024-03-01 | Returned on Time
2024-03-03 | Returned on Time
2024-03-05 | Returned on Time
2024-03-07 | Lost
2024-03-09 | Returned on Time
(5 rows)
```

34. Retrieve Genres with Their Books (including Genres with No Books):

```
library_management_system=# SELECT book.title, genre.genre_name
FROM genre
RIGHT JOIN book ON genre.genre_id = book.genre_id;
          title          | genre_name
-----+-----
Pride and Prejudice      | Fiction
1984                     | Science Fiction
Murder on the Orient Express | Mystery
The Shining              | Thriller
Harry Potter and the Philosopher's Stone | Fiction
Emma                     | Fiction
The Adventures of Huckleberry Finn | Non-Fiction
The Picture of Dorian Gray | Fiction
A Tale of Two Cities     | Fiction
The Lord of the Rings    | Fantasy
Wuthering Heights       | Fiction
Moby Dick                | Non-Fiction
Alice's Adventures in Wonderland | Science Fiction
Frankenstein             | Mystery
Dracula                  | Thriller
(15 rows)
```

35. Retrieve Branches with Their Students (including Branches with No Students):

```
library_management_system=# SELECT student.name, branch.branch_name
FROM branch
RIGHT JOIN student ON branch.branch_id = student.branch_id;
```

| name | branch_name |
|---------------------|------------------------------|
| John Doe | ECE |
| Alice Smith | Computer Science |
| Bob Johnson | Electrical Engineering |
| Emma Brown | Mechanical |
| Michael Lee | Civil Engineering |
| Ethan Parker | Chemical Engineering |
| Mia Thompson | Mechanical Engineering |
| Alexander Hernandez | Psychology Department |
| Madison Lopez | Political Science Department |
| James Garcia | Art Department |

(10 rows)

ORDER BY:

36. Order Students by Name:

```
library_management_system=# SELECT * FROM student ORDER BY name;
```

| sid | name | course_id | branch_id | email |
|-----|---------------------|-----------|-----------|---------------------------------|
| 8 | Alexander Hernandez | 8 | 8 | alexander.hernandez@example.com |
| 2 | Alice Smith | 2 | 1 | alice.smith@example.com |
| 3 | Bob Johnson | 3 | 5 | bob.johnson@example.com |
| 4 | Emma Brown | 4 | 2 | emma.brown@example.com |
| 6 | Ethan Parker | 6 | 6 | ethan.parker@example.com |
| 10 | James Garcia | 10 | 10 | james.garcia@example.com |
| 1 | John Doe | 1 | 3 | john.doe@example.com |
| 9 | Madison Lopez | 9 | 9 | madison.lopez@example.com |
| 7 | Mia Thompson | 7 | 7 | mia.thompson@example.com |
| 5 | Michael Lee | 5 | 4 | michael.lee@example.com |

(10 rows)

37. Order Books by Price in Descending Order:

```
library_management_system=# SELECT * FROM book ORDER BY price DESC;
```

| isbn | title | author_id | publisher_id | genre_id | price | quantity |
|------|--|-----------|--------------|----------|-------|----------|
| 20 | The Lord of the Rings | 10 | 5 | 7 | 20 | 10 |
| 24 | Frankenstein | 4 | 9 | 4 | 15 | 5 |
| 4 | The Shining | 4 | 4 | 5 | 15 | 4 |
| 18 | The Picture of Dorian Gray | 8 | 3 | 1 | 14 | 6 |
| 25 | Dracula | 5 | 10 | 5 | 13 | 10 |
| 5 | Harry Potter and the Philosopher's Stone | 5 | 5 | 1 | 13 | 6 |
| 22 | Moby Dick | 2 | 7 | 2 | 12 | 6 |
| 2 | 1984 | 2 | 2 | 3 | 12 | 7 |
| 16 | Emma | 6 | 1 | 1 | 12 | 5 |
| 23 | Alice's Adventures in Wonderland | 3 | 8 | 3 | 11 | 7 |
| 17 | The Adventures of Huckleberry Finn | 7 | 2 | 2 | 11 | 8 |
| 1 | Pride and Prejudice | 1 | 1 | 1 | 10 | 5 |
| 19 | A Tale of Two Cities | 9 | 4 | 1 | 10 | 7 |
| 21 | Wuthering Heights | 1 | 6 | 1 | 10 | 8 |
| 3 | Murder on the Orient Express | 3 | 3 | 4 | 8 | 3 |

(15 rows)

38. Order Authors by Number of Books Written:

| author_id | author_name | num_books |
|-----------|-----------------|-----------|
| 5 | J.K. Rowling | 2 |
| 4 | Stephen King | 2 |
| 2 | George Orwell | 2 |
| 1 | Jane Austen | 2 |
| 3 | Agatha Christie | 2 |
| 7 | J.R.R. Tolkien | 1 |
| 9 | Stephen King | 1 |
| 10 | Rupi Kaur | 1 |
| 6 | Maya Angelou | 1 |
| 8 | Dale Carnegie | 1 |

(10 rows)

39. Order Branches by Number of Students in Descending Order:

```
library_management_system=# SELECT branch.branch_id, branch.branch_name, COUNT(*) AS num_students
FROM student
INNER JOIN branch ON student.branch_id = branch.branch_id
GROUP BY branch.branch_id, branch.branch_name
ORDER BY num_students DESC;
```

| branch_id | branch_name | num_students |
|-----------|------------------------------|--------------|
| 9 | Political Science Department | 1 |
| 3 | ECE | 1 |
| 5 | Electrical Engineering | 1 |
| 4 | Civil Engineering | 1 |
| 10 | Art Department | 1 |
| 6 | Chemical Engineering | 1 |
| 2 | Mechanical | 1 |
| 7 | Mechanical Engineering | 1 |
| 1 | Computer Science | 1 |
| 8 | Psychology Department | 1 |

(10 rows)

40. Order Issued Books by Issue Date in Ascending Order:

```
library_management_system=# SELECT * FROM issue_book ORDER BY issue_date ASC;
```

| issue_id | isbn | sid | issue_date | return_date | status_id |
|----------|------|-----|------------|-------------|-----------|
| 1 | 1 | 1 | 2024-03-01 | 2024-03-15 | 1 |
| 2 | 2 | 2 | 2024-03-03 | 2024-03-18 | 1 |
| 3 | 3 | 3 | 2024-03-05 | 2024-03-20 | 1 |
| 4 | 4 | 4 | 2024-03-07 | 2024-03-22 | 3 |
| 5 | 5 | 5 | 2024-03-09 | 2024-03-24 | 1 |

(5 rows)

COUNT:

41. Count the Number of Books:

```
library_management_system=# SELECT COUNT(*) AS num_books
FROM book;
```

| num_books |
|-----------|
| 15 |

(1 row)

42. Count the Number of Students:

```
library_management_system=# SELECT COUNT(*) AS num_students
FROM student;
 num_students
-----
              10
(1 row)
```

43. Count the Number of Distinct Genres:

```
library_management_system=# SELECT COUNT(DISTINCT genre_id) AS num_genres
FROM book;
 num_genres
-----
              6
(1 row)
```

44. Count the Number of Issued Books:

```
library_management_system=# SELECT COUNT(*) AS num_issued_books
FROM issue_book;
 num_issued_books
-----
                  5
(1 row)
```

45. Count the Number of Authors with Multiple Books:

```
 num_authors_with_multiple_books
-----
                                  5
(1 row)
```

SUM:

46. Calculate the Total Price of All Books:

```
library_management_system=# SELECT SUM(price) AS total_price
FROM book;
total_price
-----
          186
(1 row)
```

47. Calculate the Total Quantity of Books:

```
library_management_system=# SELECT SUM(quantity) AS total_quantity
FROM book;
total_quantity
-----
           97
(1 row)
```

48. Calculate the Total Fine Amount Collected:

```
library_management_system=# SELECT SUM(fine_amount) AS total_fine_amount
FROM fine;
total_fine_amount
-----
                5
(1 row)
```

49. Calculate the Total Number of Pages of All Books:

```
library_management_system=# SELECT SUM(pages) AS total_pages
FROM book;
total_pages
-----
       6087
(1 row)
```

50. Calculate the Total Price of Books per Genre:

```
library_management_system=# SELECT genre_id, SUM(price) AS total_price
FROM book
GROUP BY genre_id;
 genre_id | total_price
-----+-----
         3 |          23
         5 |          28
         4 |          23
         2 |          23
         7 |          20
         1 |          69
(6 rows)
```

AVG:

51. Calculate the Average Price of Books:

```
library_management_system=# SELECT AVG(price) AS avg_price
FROM book;
      avg_price
-----
12.4000000000000000
(1 row)
```

52. Calculate the Average Quantity of Books:

```
library_management_system=# SELECT AVG(quantity) AS avg_quantity
FROM book;
      avg_quantity
-----
6.4666666666666667
(1 row)
```

53. Calculate the Average Fine Amount:

```
library_management_system=# SELECT AVG(fine_amount) AS avg_fine_amount
FROM fine;
      avg_fine_amount
-----
1.000000000000000000
(1 row)
```

54. Calculate the Average Rating of Books:

```
library_management_system=# SELECT AVG(rating) AS avg_rating
FROM book;
      avg_rating
-----
4.133333333333333
(1 row)
```

55. Calculate the Average Price of Books per Genre:

```
library_management_system=# SELECT genre_id, AVG(price) AS avg_price
FROM book
GROUP BY genre_id;
 genre_id |      avg_price
-----+-----
        3 | 11.500000000000000
        5 | 14.000000000000000
        4 | 11.500000000000000
        2 | 11.500000000000000
        7 | 20.000000000000000
        1 | 11.500000000000000
(6 rows)
```

MIN:

56. Find the Minimum Price of Books:

```
library_management_system=# SELECT MIN(price) AS min_price
FROM book;
      min_price
-----
            8
(1 row)
```

57. Find the Earliest Issue Date of Books:

```
library_management_system=# SELECT MIN(issue_date) AS earliest_issue_date
FROM issue_book;
earliest_issue_date
-----
2024-03-01
(1 row)
```

58. Find the Minimum Fine Amount:

```
library_management_system=# SELECT MIN(fine_amount) AS min_fine_amount
FROM fine;
min_fine_amount
-----
0
(1 row)
```

59. Find the Lowest Rating of Books:

```
library_management_system=# SELECT MIN(rating) AS min_rating
FROM book;
min_rating
-----
3
(1 row)
```

60. Find the Minimum Price of Books per Genre:

```
library_management_system=# SELECT genre_id, MIN(price) AS min_price
FROM book
GROUP BY genre_id;
genre_id | min_price
-----+-----
3 | 11
5 | 13
4 | 8
2 | 11
7 | 20
1 | 10
(6 rows)
```


MAX:

61. Find the Maximum Price of Books:

```
library_management_system=# SELECT MAX(price) AS max_price
FROM book;
max_price
-----
          20
(1 row)
```

62. Find the Latest Return Date of Books:

```
library_management_system=# SELECT MAX(return_date) AS latest_return_date
FROM return_book;
latest_return_date
-----
2024-03-24
(1 row)
```

63. Find the Maximum Fine Amount:

```
library_management_system=# SELECT MAX(fine_amount) AS max_fine_amount
FROM fine;
max_fine_amount
-----
          5
(1 row)
```

64. Find the Highest Rating of Books:

```
library_management_system=# SELECT MAX(rating) AS max_rating
FROM book;
max_rating
-----
          5
(1 row)
```


65. Find the Maximum Price of Books per Genre:

```
library_management_system=# SELECT genre_id, MAX(price) AS max_price
FROM book
GROUP BY genre_id;
 genre_id | max_price
-----+-----
         3 |         12
         5 |         15
         4 |         15
         2 |         12
         7 |         20
         1 |         14
(6 rows)
```

Sub-Queries

66. Single-Row Subquery

```
library_management_system=# SELECT title
FROM book
WHERE price = (SELECT MAX(price) FROM book);
      title
-----
The Lord of the Rings
(1 row)
```

67. Find the Name of the Author of the Book with the Most Pages:

```
library_management_system=# SELECT author_name
FROM author
WHERE author_id = (SELECT author_id FROM book WHERE pages = (SELECT MAX(pages) FROM book));
 author_name
-----
Rupi Kaur
(1 row)
```

68. Get the Name of the Publisher of the Book with the Lowest Rating:

```
library_management_system=# SELECT publisher_name
FROM publisher
WHERE publisher_id = (
    SELECT DISTINCT publisher_id
    FROM book
    WHERE rating = (SELECT MIN(rating) FROM book)
    LIMIT 1
);
publisher_name
-----
HarperCollins
(1 row)
```

69. Find the Course Name of the Student with the Highest ID:

```
library_management_system=# SELECT course_name
FROM course
WHERE course_id = (SELECT course_id FROM student WHERE sid = (SELECT MAX(sid) FROM student));
course_name
-----
Art History
(1 row)
```

70. Retrieve the Branch Name of the Student with the Most Issued Books:

```
library_management_system=# SELECT branch_name
FROM branch
WHERE branch_id = (SELECT branch_id FROM student WHERE sid = (SELECT sid FROM (
    SELECT sid, COUNT(*) AS num_issued_books
    FROM issue_book
    GROUP BY sid
    ORDER BY num_issued_books DESC
    LIMIT 1) AS subquery
));
branch_name
-----
Electrical Engineering
(1 row)
```

Correlated Subquery

71. Find Authors with Books in Multiple Genres:

```
library_management_system=# SELECT DISTINCT author_name
FROM author a
WHERE EXISTS (
    SELECT 1
    FROM book b
    WHERE b.author_id = a.author_id
    GROUP BY b.author_id
    HAVING COUNT(DISTINCT b.genre_id) > 1
);
 author_name
-----
Agatha Christie
George Orwell
J.K. Rowling
Stephen King
(4 rows)
```

72. Retrieve Books with Price Higher Than Average for Their Genre:

```
library_management_system=# SELECT title
FROM book b1
WHERE price > (
    SELECT AVG(price)
    FROM book b2
    WHERE b1.genre_id = b2.genre_id
);
          title
-----
1984
The Shining
Harry Potter and the Philosopher's Stone
Emma
The Picture of Dorian Gray
Moby Dick
Frankenstein
(7 rows)
```

73. Find Publishers with Books Written by Multiple Authors:

```
library_management_system=# SELECT DISTINCT publisher_name
FROM publisher p
WHERE EXISTS (
    SELECT 1
    FROM book b
    WHERE b.publisher_id = p.publisher_id
    GROUP BY b.publisher_id
    HAVING COUNT(DISTINCT b.author_id) > 1
);
 publisher_name
-----
Hachette Livre
HarperCollins
Macmillan Publishers
Penguin Random House
Simon & Schuster
(5 rows)
```

74. Retrieve Students Enrolled in the Same Course as John Doe:

```
library_management_system=# SELECT name
FROM student
WHERE course_id = (
    SELECT course_id
    FROM student
    WHERE name = 'John Doe'
);
 name
-----
John Doe
(1 row)
```

75. Get Books Published by the Same Publisher as 'Pride and Prejudice':

```
library_management_system=# SELECT title
FROM book
WHERE publisher_id = (
    SELECT publisher_id
    FROM book
    WHERE title = 'Pride and Prejudice'
);
      title
-----
Pride and Prejudice
Emma
(2 rows)
```

Nested Subquery

76. Find Students Enrolled in Courses with the Highest Number of Students:

```
library_management_system=# SELECT name
FROM student
WHERE course_id IN (
    SELECT course_id
    FROM (
        SELECT course_id, COUNT(*) AS num_students
        FROM student
        GROUP BY course_id
        ORDER BY num_students DESC
        LIMIT 1
    ) AS max_students_course
);
      name
-----
Madison Lopez
(1 row)
```

77. Retrieve Authors with Books in the Same Genre as '1984':

```
library_management_system=# SELECT DISTINCT author_name
FROM author
WHERE EXISTS (
    SELECT 1
    FROM book
    WHERE genre_id = (
        SELECT genre_id
        FROM book
        WHERE title = '1984'
    )
    AND book.author_id = author.author_id
);
author_name
-----
Agatha Christie
George Orwell
(2 rows)
```

78. Find Books Published by Publishers with Books in the Most Popular Genre:

```
library_management_system=# SELECT title
FROM book
WHERE publisher_id IN (
  SELECT publisher_id
  FROM book
  WHERE genre_id = (
    SELECT genre_id
    FROM (
      SELECT genre_id, COUNT(*) AS num_books
      FROM book
      GROUP BY genre_id
      ORDER BY num_books DESC
      LIMIT 1
    ) AS popular_genre
  )
);
```

| title |
|-------|
|-------|

| |
|--|
| Pride and Prejudice |
| Murder on the Orient Express |
| The Shining |
| Harry Potter and the Philosopher's Stone |
| Emma |
| The Picture of Dorian Gray |
| A Tale of Two Cities |
| The Lord of the Rings |
| Wuthering Heights |

(9 rows)

79. Retrieve Courses with the Lowest Average Rating:

```
library_management_system=# SELECT course_name
FROM course
WHERE course_id IN (
    SELECT course_id
    FROM (
        SELECT course_id, AVG(rating) AS avg_rating
        FROM (
            SELECT student.course_id, book.rating
            FROM student
            INNER JOIN issue_book ON student.sid = issue_book.sid
            INNER JOIN book ON issue_book.isbn = book.isbn
        ) AS student_book_ratings
        GROUP BY course_id
        ORDER BY avg_rating ASC
        LIMIT 1
    ) AS min_avg_rating_course
);
 course_name
-----
History
(1 row)
```

80. Find Students Enrolled in Courses with Books Written by Their Favorite Authors:

```
library_management_system=# SELECT name
FROM student s1
WHERE course_id IN (
    SELECT course_id
    FROM student s2
    WHERE s2.sid = (
        SELECT sid
        FROM (
            SELECT sid, MAX(author_id) AS favorite_author_id
            FROM (
                SELECT student.sid, book.author_id
                FROM student
                INNER JOIN issue_book ON student.sid = issue_book.sid
                INNER JOIN book ON issue_book.isbn = book.isbn
            ) AS student_book_authors
            GROUP BY sid
            ORDER BY MAX(author_id) DESC -- Add an ORDER BY clause to ensure a single row is selected
            LIMIT 1 -- Limit the result to one row
        ) AS favorite_authors
    )
);
 name
-----
Michael Lee
(1 row)
```


BETWEEN

81. Retrieve Books with Prices Between \$10 and \$15:

```
library_management_system=# SELECT title, price
FROM book
WHERE price BETWEEN 10 AND 15;
```

| title | price |
|--|-------|
| Pride and Prejudice | 10 |
| 1984 | 12 |
| The Shining | 15 |
| Harry Potter and the Philosopher's Stone | 13 |
| Emma | 12 |
| The Adventures of Huckleberry Finn | 11 |
| The Picture of Dorian Gray | 14 |
| A Tale of Two Cities | 10 |
| Wuthering Heights | 10 |
| Moby Dick | 12 |
| Alice's Adventures in Wonderland | 11 |
| Frankenstein | 15 |
| Dracula | 13 |

(13 rows)

82. Find Students Enrolled in Courses with IDs Between 3 and 7:

```
library_management_system=# SELECT name, course_id
FROM student
WHERE course_id BETWEEN 3 AND 7;
```

| name | course_id |
|--------------|-----------|
| Bob Johnson | 3 |
| Emma Brown | 4 |
| Michael Lee | 5 |
| Ethan Parker | 6 |
| Mia Thompson | 7 |

(5 rows)

83. Retrieve Books with ISBNs Between 10 and 20:

```
library_management_system=# SELECT title, isbn
FROM book
WHERE isbn BETWEEN 10 AND 20;
```

| title | isbn |
|------------------------------------|------|
| Emma | 16 |
| The Adventures of Huckleberry Finn | 17 |
| The Picture of Dorian Gray | 18 |
| A Tale of Two Cities | 19 |
| The Lord of the Rings | 20 |

(5 rows)

84. Find Publishers with IDs Between 3 and 5:

```
library_management_system=# SELECT publisher_name
FROM publisher
WHERE publisher_id BETWEEN 3 AND 5;
```

| publisher_name |
|----------------------|
| Simon & Schuster |
| Macmillan Publishers |
| Hachette Livre |

(3 rows)

85. Retrieve Books Published by Publishers with IDs Between 1 and 3:

```
library_management_system=# SELECT title, publisher_id
FROM book
WHERE publisher_id BETWEEN 1 AND 3;
```

| title | publisher_id |
|------------------------------------|--------------|
| Pride and Prejudice | 1 |
| 1984 | 2 |
| Murder on the Orient Express | 3 |
| Emma | 1 |
| The Adventures of Huckleberry Finn | 2 |
| The Picture of Dorian Gray | 3 |

(6 rows)

UNION

86. Retrieve Names of Students and Titles of Books in a Single Result Set:

```
library_management_system=# SELECT name AS result FROM student
UNION
SELECT title FROM book;
      result
```

```
Wuthering Heights
Alice's Adventures in Wonderland
Harry Potter and the Philosopher's Stone
A Tale of Two Cities
Alexander Hernandez
James Garcia
Mia Thompson
Frankenstein
1984
Emma Brown
Pride and Prejudice
The Shining
Murder on the Orient Express
Bob Johnson
Dracula
The Adventures of Huckleberry Finn
Ethan Parker
Moby Dick
Madison Lopez
Alice Smith
The Picture of Dorian Gray
Michael Lee
John Doe
The Lord of the Rings
Emma
(25 rows)
```

87. Find Courses and Genres Present in the Database:

```
library_management_system=# SELECT course_name AS result FROM course
UNION
SELECT genre_name FROM genre;
      result
-----
Non-Fiction
Poetry
Physics
Art History
Science Fiction
Biography
Computer Science
Chemistry
English Literature
History
Political Science
Fantasy
Horror
Psychology
Biology
Thriller
Self-Help
Mechanical Engineering
Bio-Technology
Mystery
Fiction
(21 rows)
```

88. Retrieve Authors and Publishers in a Single List:

```
library_management_system=# SELECT author_name AS result FROM author
UNION
SELECT publisher_name FROM publisher;
      result
-----
Agatha Christie
Penguin Random House
H.C.Verma
Macmillan Publishers
Stephen King
Maya Angelou
Simon & Schuster
J.K. Rowling
Random House
Vintage Books
Rupi Kaur
Dale Carnegie
George Orwell
Jane Austen
HarperCollins
Hachette Livre
J.R.R. Tolkien
Penguin Books
(18 rows)
```

89. List Student Emails and Titles of Books as a Combined Set:

```
library_management_system=# SELECT email AS result FROM student
UNION
SELECT title FROM book;
      result
-----
emma.brown@example.com
Wuthering Heights
bob.johnson@example.com
Alice's Adventures in Wonderland
ethan.parker@example.com
alice.smith@example.com
Harry Potter and the Philosopher's Stone
A Tale of Two Cities
mia.thompson@example.com
james.garcia@example.com
Frankenstein
1984
michael.lee@example.com
Pride and Prejudice
The Shining
Murder on the Orient Express
Dracula
alexander.hernandez@example.com
The Adventures of Huckleberry Finn
madison.lopez@example.com
Moby Dick
The Picture of Dorian Gray
The Lord of the Rings
Emma
john.doe@example.com
(25 rows)
```

90. Find Branches and Courses Available in the Database:

```
library_management_system=# SELECT branch_name AS result FROM branch
UNION
SELECT course_name FROM course;
      result
-----
Mechanical
ECE
Chemical Engineering
Electrical Engineering
Physics
Civil Engineering
Art History
Art Department
Computer Science
Chemistry
English Literature
History
Psychology Department
Political Science
Psychology
Biology
Mechanical Engineering
Political Science Department
Bio-Technology
(19 rows)
```

IN

91. Retrieve Titles of Books Published by Publishers with IDs 1, 3, and 5:

```
library_management_system=# SELECT title FROM book WHERE publisher_id IN (1, 3, 5);
      title
-----
Pride and Prejudice
Murder on the Orient Express
Harry Potter and the Philosopher's Stone
Emma
The Picture of Dorian Gray
The Lord of the Rings
(6 rows)
```


92. Find Names of Students Enrolled in Courses with IDs 1, 3, and 5:

```
library_management_system=# SELECT name FROM student WHERE course_id IN (1, 3, 5);
name
-----
John Doe
Bob Johnson
Michael Lee
(3 rows)
```

93. Retrieve Titles of Books with Ratings 8, 9, and 10:

```
library_management_system=# SELECT title FROM book WHERE rating IN (8, 9, 10);
title
-----
(0 rows)
```

94. Find Authors Who Have Published Books with ISBNs 3, 7, and 10:

```
library_management_system=# SELECT author_name FROM author WHERE author_id IN (
    SELECT author_id FROM book WHERE isbn IN (3, 7, 10)
);
author_name
-----
Agatha Christie
(1 row)
```

95. Retrieve Titles of Books with Genres 'Fiction' and 'Fantasy':

```
library_management_system=# SELECT title FROM book WHERE genre_id IN (
    SELECT genre_id FROM genre WHERE genre_name IN ('Fiction', 'Fantasy')
);
title
-----
Pride and Prejudice
Harry Potter and the Philosopher's Stone
Emma
The Picture of Dorian Gray
A Tale of Two Cities
The Lord of the Rings
Wuthering Heights
(7 rows)
```

LIMIT:

96. Retrieve the First 5 Books from the Database:

```
library_management_system=# SELECT * FROM book LIMIT 5;
```

| isbn | title | author_id | publisher_id | genre_id | price | quantity | rating | pages |
|------|--|-----------|--------------|----------|-------|----------|--------|-------|
| 1 | Pride and Prejudice | 1 | 1 | 1 | 10 | 5 | 4 | 279 |
| 2 | 1984 | 2 | 2 | 3 | 12 | 7 | 5 | 328 |
| 3 | Murder on the Orient Express | 3 | 3 | 4 | 8 | 3 | 4 | 274 |
| 4 | The Shining | 4 | 4 | 5 | 15 | 4 | 3 | 447 |
| 5 | Harry Potter and the Philosopher's Stone | 5 | 5 | 1 | 13 | 6 | 5 | 332 |

(5 rows)

97. Find the Top 5 Authors with the Most Published Books

```
library_management_system=# SELECT author.author_id, author.author_name, COUNT(*) AS num_books
FROM book
INNER JOIN author ON book.author_id = author.author_id
GROUP BY author.author_id, author.author_name
ORDER BY num_books DESC
LIMIT 5;
```

| author_id | author_name | num_books |
|-----------|-----------------|-----------|
| 5 | J.K. Rowling | 2 |
| 4 | Stephen King | 2 |
| 2 | George Orwell | 2 |
| 1 | Jane Austen | 2 |
| 3 | Agatha Christie | 2 |

(5 rows)

98. Retrieve the First 5 Students Enrolled in Computer Science:

```
library_management_system=# SELECT * FROM student WHERE course_id = 1 LIMIT 5;
```

| sid | name | course_id | branch_id | email |
|-----|----------|-----------|-----------|----------------------|
| 1 | John Doe | 1 | 3 | john.doe@example.com |

(1 row)

99. Find the 5 Most Expensive Books:

```
library_management_system=# SELECT * FROM book ORDER BY price DESC LIMIT 5;
```

| isbn | title | author_id | publisher_id | genre_id | price | quantity | rating | pages |
|------|--|-----------|--------------|----------|-------|----------|--------|-------|
| 20 | The Lord of the Rings | 10 | 5 | 7 | 20 | 10 | 5 | 1178 |
| 24 | Frankenstein | 4 | 9 | 4 | 15 | 5 | 4 | 280 |
| 4 | The Shining | 4 | 4 | 5 | 15 | 4 | 3 | 447 |
| 18 | The Picture of Dorian Gray | 8 | 3 | 1 | 14 | 6 | 5 | 254 |
| 5 | Harry Potter and the Philosopher's Stone | 5 | 5 | 1 | 13 | 6 | 5 | 332 |

(5 rows)

100. Retrieve the Top 5 Publishers with the Most Published Books:

```
library_management_system=# SELECT publisher.publisher_id, publisher.publisher_name, COUNT(*) AS num_books
FROM book
INNER JOIN publisher ON book.publisher_id = publisher.publisher_id
GROUP BY publisher.publisher_id, publisher.publisher_name
ORDER BY num_books DESC
LIMIT 5;
```

| publisher_id | publisher_name | num_books |
|--------------|----------------------|-----------|
| 5 | Hachette Livre | 2 |
| 4 | Macmillan Publishers | 2 |
| 2 | HarperCollins | 2 |
| 1 | Penguin Random House | 2 |
| 3 | Simon & Schuster | 2 |

(5 rows)