1. **Fine Calculation:** Calculate the total fines owed by each member, considering overdue books and a daily fine rate (e.g., $0.25 per day):

SELECT

c.unique\_id AS member\_id,

c.name AS member\_name,

ROUND(SUM(

CASE

WHEN bb.due\_date < COALESCE(bb.return\_date, CURRENT\_DATE)

THEN DATEDIFF(COALESCE(bb.return\_date, CURRENT\_DATE), bb.due\_date) \* @daily\_fine\_rate

ELSE 0

END

), 2) AS total\_fine

FROM

Client c

LEFT JOIN

BookBorrowing bb ON c.unique\_id = bb.client\_id

LEFT JOIN

DigitalMediaBorrowing dmb ON c.unique\_id = dmb.client\_id

LEFT JOIN

MagazineBorrowing mb ON c.unique\_id = mb.client\_id

WHERE

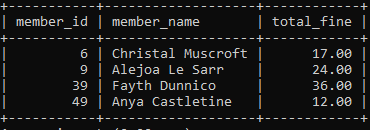
c.account\_status = 'Active'

GROUP BY

c.unique\_id, c.name

HAVING

total\_fine > 0;



1. **Book Availability:** Display a list of all available books (not currently borrowed) within a specific genre.

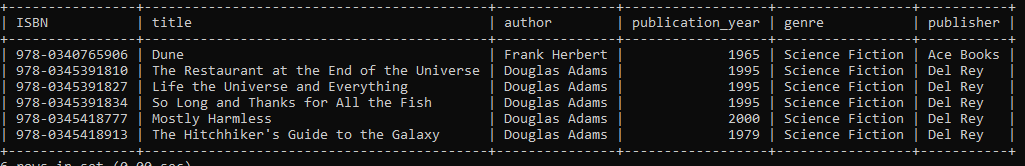
SELECT b.ISBN, b.title, b.author, b.publication\_year, b.genre, b.publisher

FROM Book b

JOIN Book\_Copy bc ON b.ISBN = bc.ISBN

WHERE bc.status = 'Available'

AND b.genre = 'Science Fiction';



1. **Frequent Borrowers of a Specific Genre:** Identify the members who have borrowed the most books in a particular genre (e.g., "Mystery") in the last year.

SELECT

c.unique\_id AS member\_id,

c.name AS member\_name,

COUNT(bb.item\_id) AS books\_borrowed,

b.genre

FROM

Client c

JOIN

BookBorrowing bb ON c.unique\_id = bb.client\_id

JOIN

Book\_Copy bc ON bb.item\_id = bc.copy\_id

JOIN

Book b ON bc.ISBN = b.ISBN

WHERE

b.genre = 'Fantasy’'

AND bb.borrow\_date >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)

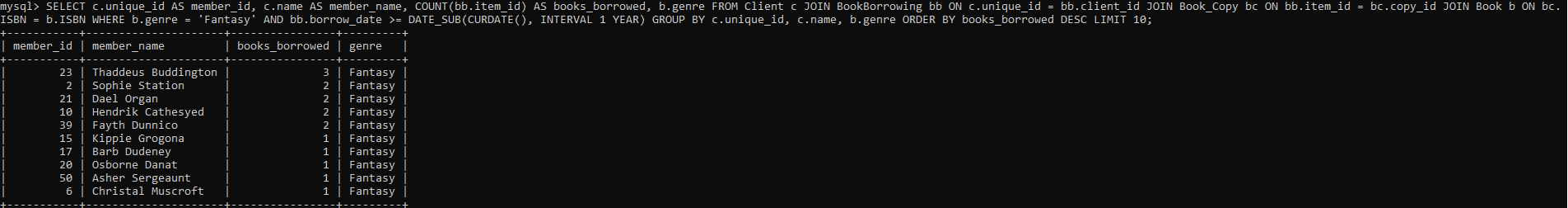
GROUP BY

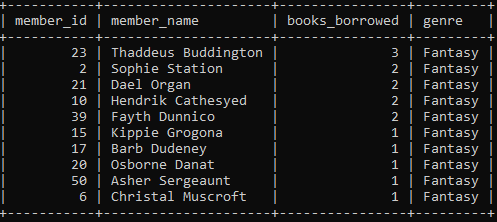
c.unique\_id, c.name, b.genre

ORDER BY

books\_borrowed DESC

LIMIT 10;





1. **Books Due Soon:** Generate a report of all books due within the next week, sorted by due date.

SELECT

bb.item\_id AS book\_copy\_id,

b.title AS book\_title,

b.ISBN,

c.name AS borrower\_name,

bb.due\_date

FROM

BookBorrowing bb

JOIN

Book\_Copy bc ON bb.item\_id = bc.copy\_id

JOIN

Book b ON bc.ISBN = b.ISBN

JOIN

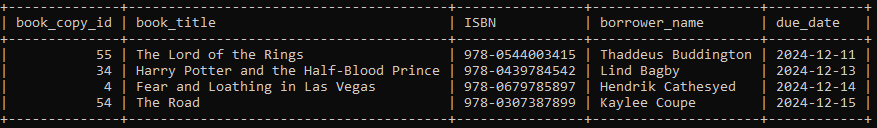
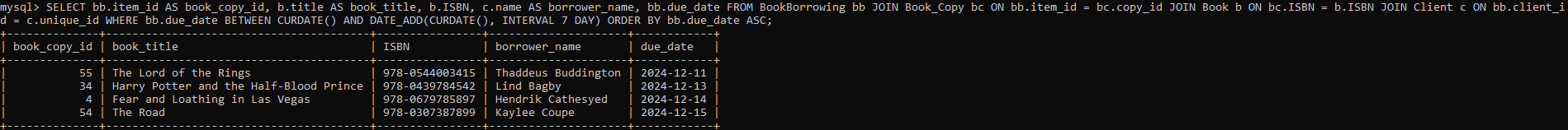
Client c ON bb.client\_id = c.unique\_id

WHERE

bb.due\_date BETWEEN CURDATE() AND DATE\_ADD(CURDATE(), INTERVAL 7 DAY)

ORDER BY

bb.due\_date ASC;



1. **Members with Overdue Books:** List all members who currently have at least one overdue book, along with the titles of the overdue books.

SELECT Client.name, Book.title

FROM Client

JOIN BookBorrowing ON Client.unique\_id = BookBorrowing.client\_id

JOIN Book\_Copy ON BookBorrowing.item\_id = Book\_Copy.copy\_id

JOIN Book ON Book\_Copy.ISBN = Book.ISBN

WHERE

EXISTS (

SELECT 1

FROM BookBorrowing

WHERE client\_id = Client.unique\_id

)

ORDER BY

Client.name, Book.title;



**Average Borrowing Time:** Calculate the average number of days members borrow books for a specific genre.

SELECT b.genre, AVG(DATEDIFF(COALESCE(bb.return\_date, CURDATE()), bb.borrow\_date)) AS average\_borrow\_days

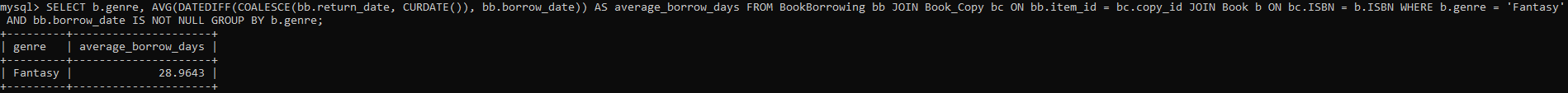
FROM BookBorrowing bb

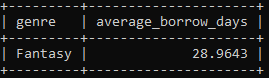
JOIN Book\_Copy bc ON bb.item\_id = bc.copy\_id

JOIN Book b ON bc.ISBN = b.ISBN

WHERE b.genre = 'Fantasy' AND bb.borrow\_date IS NOT NULL

GROUP BY b.genre;



****

1. **Most Popular Author in the Last Month:** Determine the author whose books have been borrowed the most in the last month.

SELECT

b.author,

COUNT(bb.item\_id) AS borrow\_count

FROM

BookBorrowing bb

JOIN

Book\_Copy bc ON bb.item\_id = bc.copy\_id

JOIN

Book b ON bc.ISBN = b.ISBN

WHERE

bb.borrow\_date >= DATE\_SUB(CURDATE(), INTERVAL 1 MONTH)

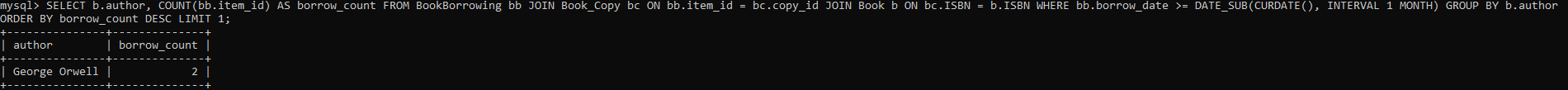
GROUP BY

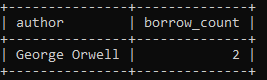
b.author

ORDER BY

borrow\_count DESC

LIMIT 1;





1. **Report 1: Generate a Collection Analysis Report:** This report should provide a comprehensive analysis of the library's book collection, examining the distribution of books by genre, identifying trends in acquisition over the past 5 years, and assessing the age of the collection to identify outdated materials. Highlight books with low circulation and analyze borrowing patterns to identify under-represented genres or authors. Your report should provide insights for collection development and management decisions

-- Books by genre

SELECT genre, COUNT(\*) AS total\_books

FROM Book

GROUP BY genre

ORDER BY total\_books DESC;

-- Acquisition Trends

SELECT

IFNULL(YEAR(earliest\_borrow\_date), 'Not Borrowed Yet') AS acquisition\_year,

COUNT(DISTINCT bc.copy\_id) AS total\_books

FROM

Book\_Copy bc

LEFT JOIN (

SELECT

item\_id,

MIN(borrow\_date) AS earliest\_borrow\_date

FROM

BookBorrowing

GROUP BY

item\_id

) bb ON bc.copy\_id = bb.item\_id

GROUP BY

acquisition\_year

ORDER BY

acquisition\_year;

-- Low circulation Books, shows a list of books having 5 or less borrows total and orders by lowest first

SELECT b.title, COUNT(bb.item\_id) AS borrow\_count

FROM Book b

JOIN Book\_Copy bc ON b.ISBN = bc.ISBN

LEFT JOIN BookBorrowing bb ON bc.copy\_id = bb.item\_id

GROUP BY b.title

HAVING borrow\_count < 5

ORDER BY borrow\_count;

-- Borrowing Pattern by Genre

SELECT b.genre, COUNT(bb.item\_id) AS borrow\_count

FROM Book b

JOIN Book\_Copy bc ON b.ISBN = bc.ISBN

JOIN BookBorrowing bb ON bc.copy\_id = bb.item\_id

GROUP BY b.genre

ORDER BY borrow\_count DESC;

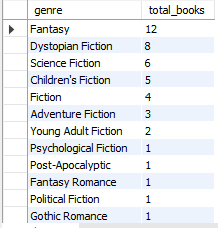
– Average age of books since publication

SELECT AVG(YEAR(CURDATE()) - publication\_year)

FROM Book\_Copy bc

JOIN Book ON bc.ISBN = Book.ISBN;

**Books by Genre:**



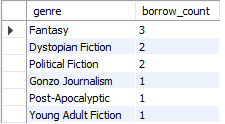
**Acquisition Trends:**



**Low Circulation Books:**



**Borrowing Pattern by Genre:**



**Average time since book was published:**

SELECT AVG(YEAR(CURDATE()) - publication\_year) as average\_years\_since\_published

FROM Book\_Copy bc

JOIN Book ON bc.ISBN = Book.ISBN;



**Outdated Books (Books published before average publishing time):**

SELECT Distinct(b.title), b.ISBN, author, publication\_year, genre, publisher

FROM Book b

JOIN Book\_Copy bc ON b.ISBN = bc.ISBN

WHERE (YEAR(CURDATE()) - b.publication\_year) > (

SELECT AVG(YEAR(CURDATE()) - publication\_year)

FROM Book b2

JOIN Book\_Copy bc2 ON b2.ISBN = bc2.ISBN

);

