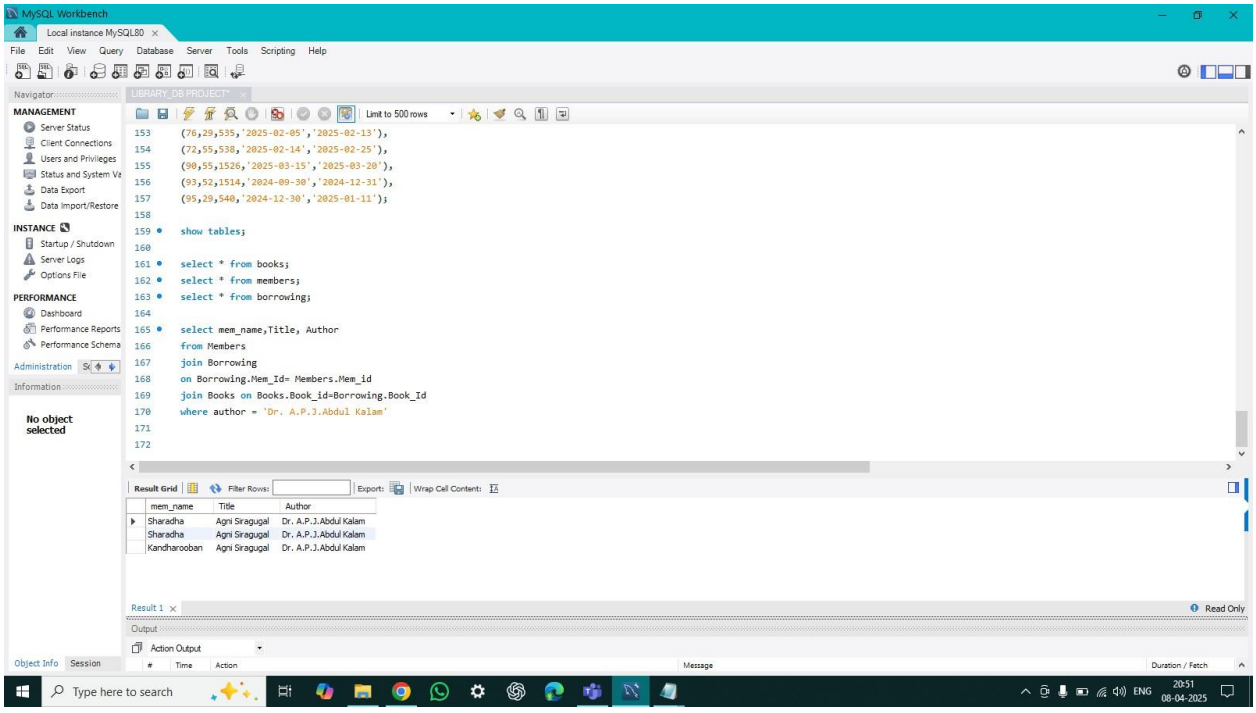
## 1. Data retrieval:

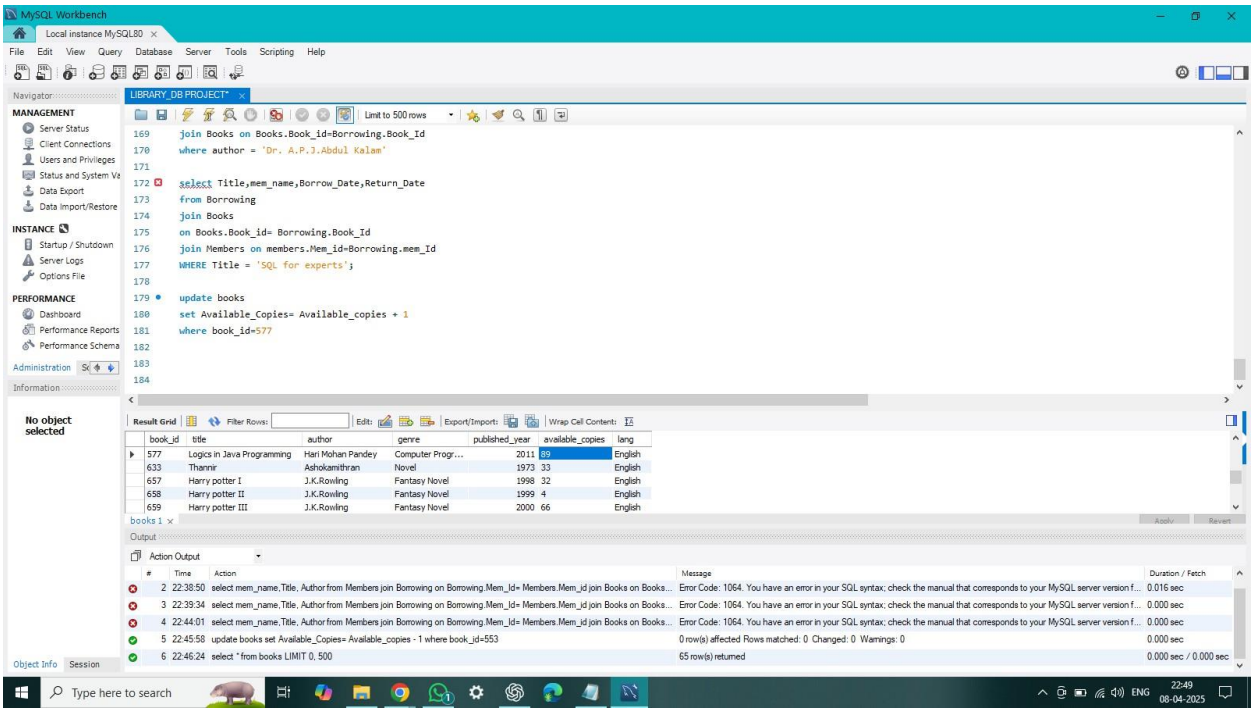
### Get a list of members borrowed a specific authors book.

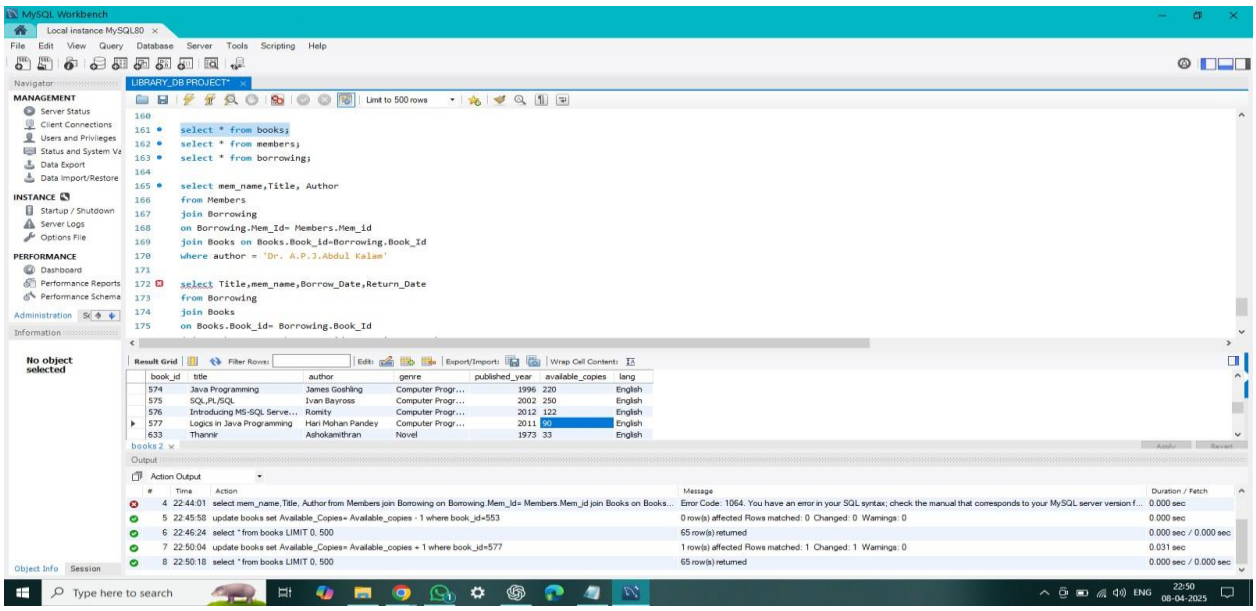
```
Select mem_name, Title, Author
from Members  join Borrowing
On Borrowing.Mem_Id=Members.Mem_id
join Books on Books.Book_id=Borrowing.Book_Id
where author = 'Dr. A.P.J.Abdul Kalam';
```



## 2. Query to update the available copies in specified id by1.

Update books set Available_Copies=Available_copies+1,where book_id=577;



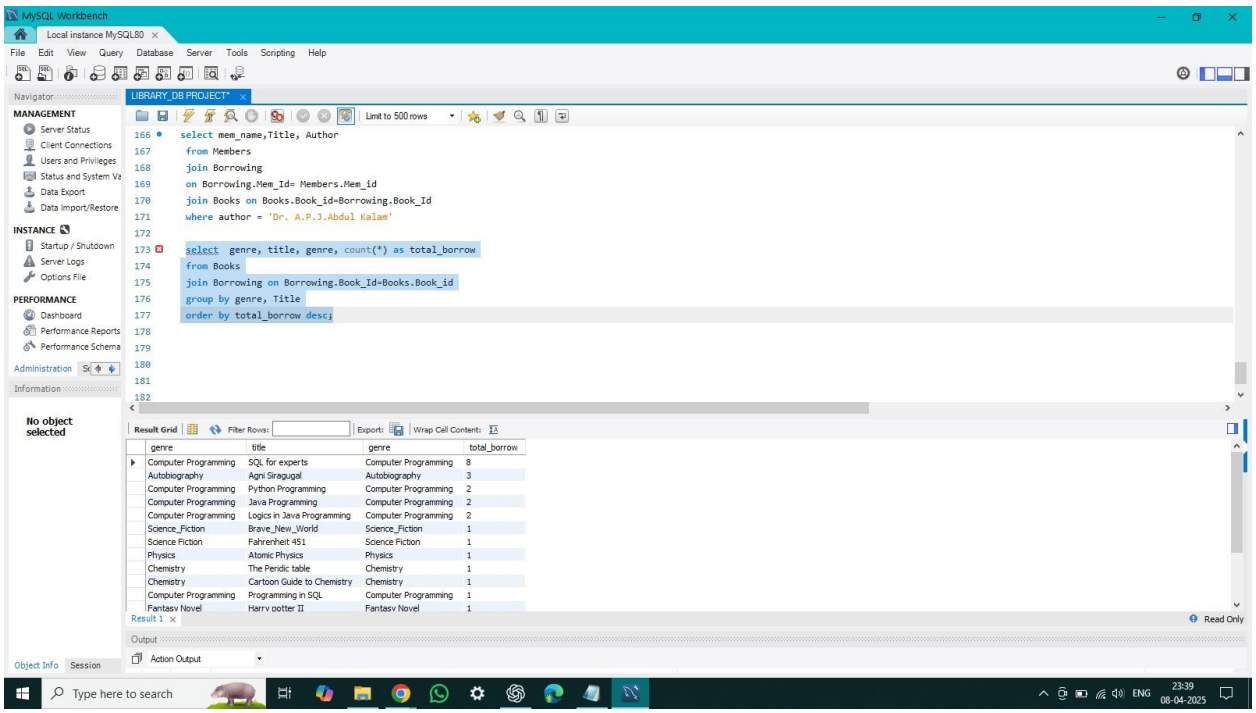



3. **Find the most popular genre in the library. Display the genre with the highest total number of books borrowed.**

Select genre, title, genre, count(*)as total_borrow

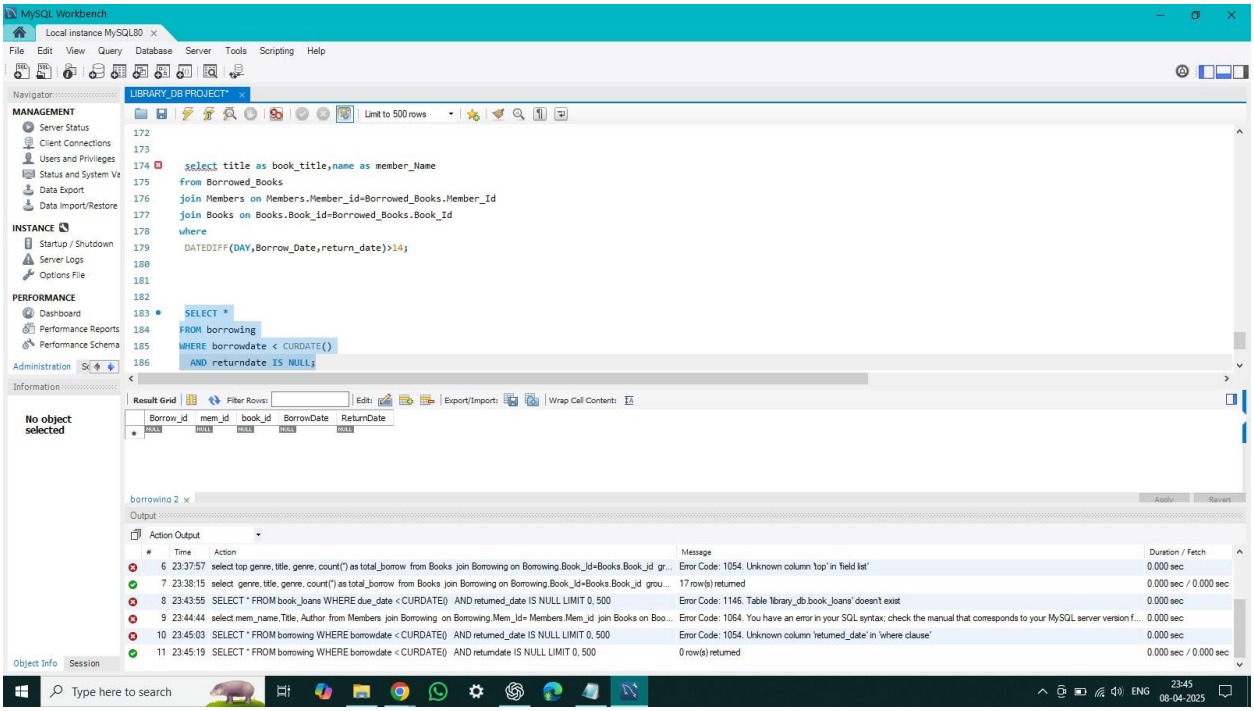
From Books join Borrowing on Borrowing.Book_Id=Books.Book_id

group by genre, Title order by total_borrow desc;



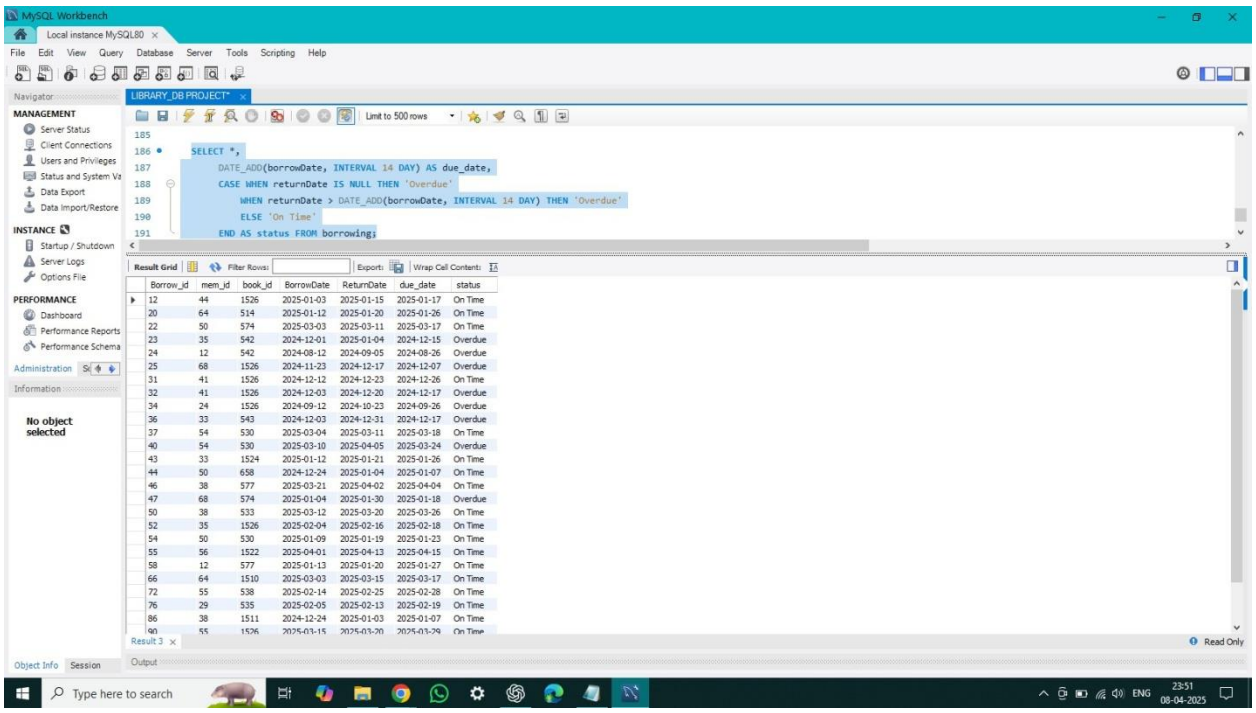
## 4. Query to calculate the overdue books at current date.

SELECT* FROM borrowing WHERE borrowdate <CURDATE();



## 5. Query to find Overdue Books Based on Entered Dates:

SELECT DATE_ADD(borrowDate,INTERVAL14DAY)ASdue_date, CASE WHEN returnDate IS NULL THEN 'Overdue' WHEN returnDate > DATE_ADD (borrowDate,INTERVAL14DAY) THEN 'Overdue' ELSE 'OnTime' END AS status FROM borrowing AND returndate IS NULL;

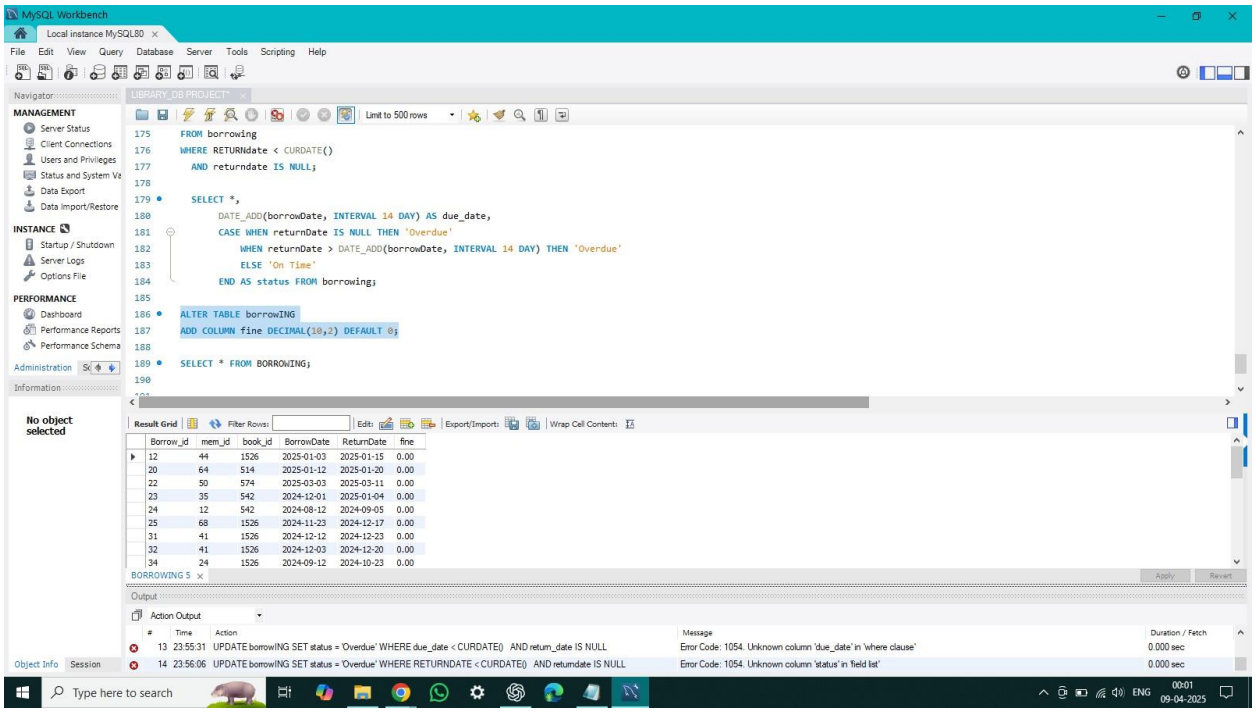


# 6. Use of DDL COMMAND:

## Alter table borrowing with fine.

ALTER TABLE borrowing

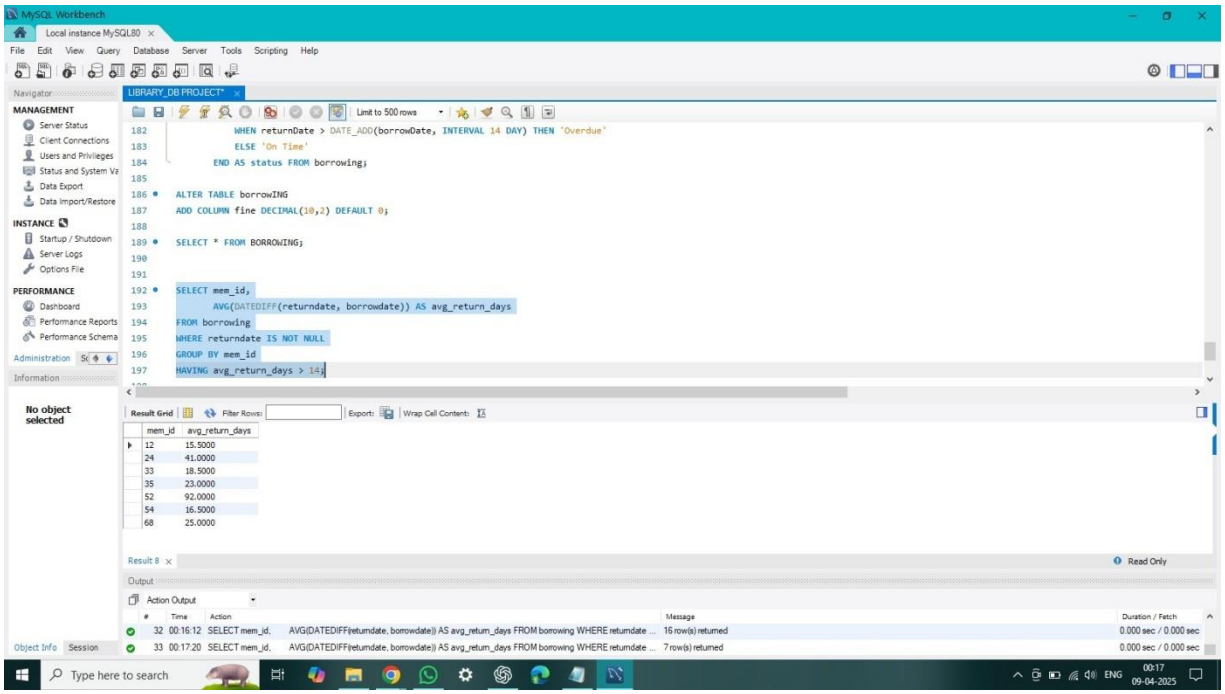
ADD COLUMN fine DECIMAL(10,2) DEFAULT 0;



**7. Find members who return books quickly. Calculate the average duration it takes for each member to return borrowed books and then identify members whose average return time is less than 14 days.**

SELECT borrow_id, book_id, mem_id, DATE_ADD(borrowDate, INTERVAL 14 DAY) AS due_date, CASE WHEN return Date ISNULLTHEN 'Overdue' WHEN returnDate>DATE_ADD(borrowDate,INTERVAL14DAY) THEN 'Overdue' ELSE 'OnTime' END AS status FROM borrowing

WHERE returnDate IS NULL OR returnDate>DATE_ADD(borrowDate, INTERVAL 14 DAY);



**8. Calculate the average duration in months for which books are borrowed. Display the book title and the average duration in months.**

SELECTCOUNT(MEM_ID)ASNO_OF_BOOKS_BORROWED,

CASE WHEN COUNT(MEM_ID)

BETWEEN 0 AND 3

THEN'REGULAR BORROWERS'

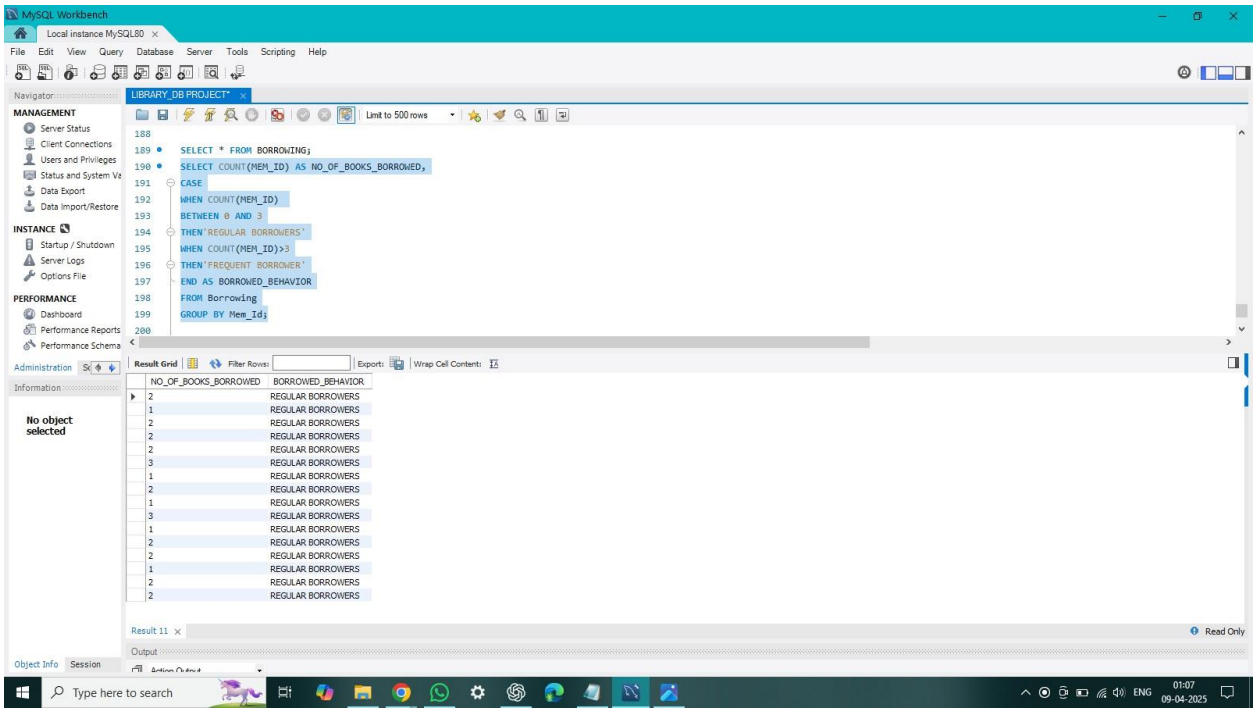
WHEN COUNT(MEM_ID)>3

THEN'FREQUENT BORROWER'

ENDASBORROWED_BEHAVIOR

FROM Borrowing

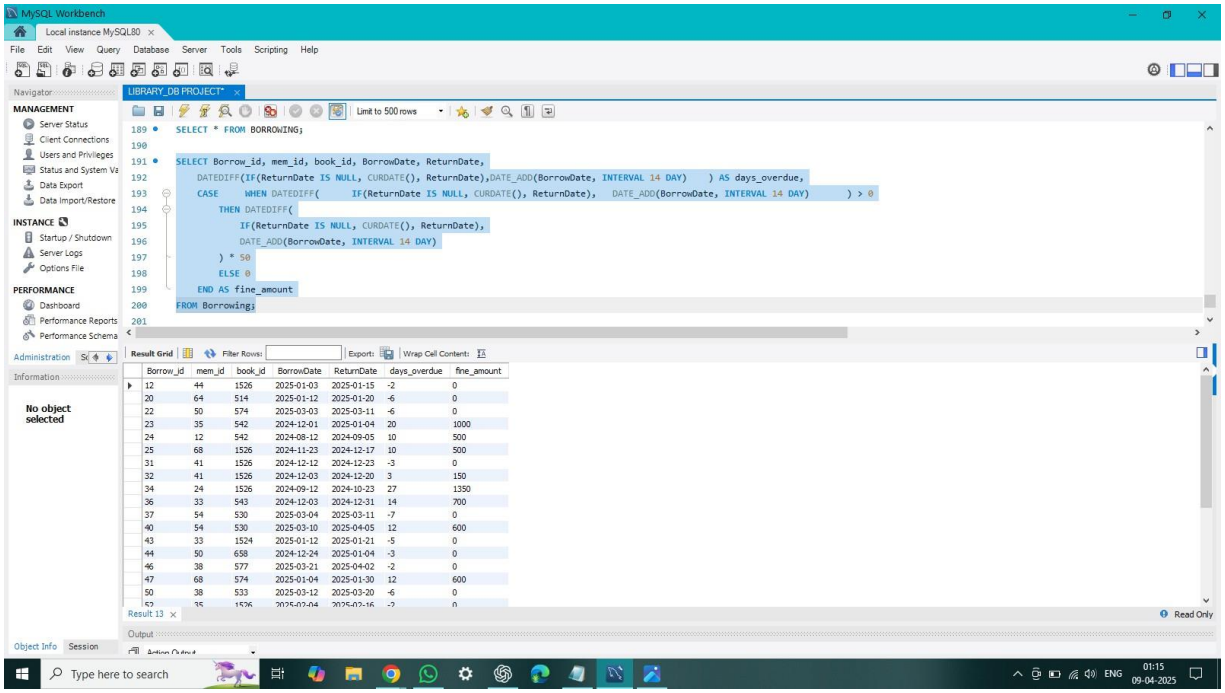
GROUPBY Mem_Id;



## 9. Calculate the late fees for each overdue book by multiples of 50.

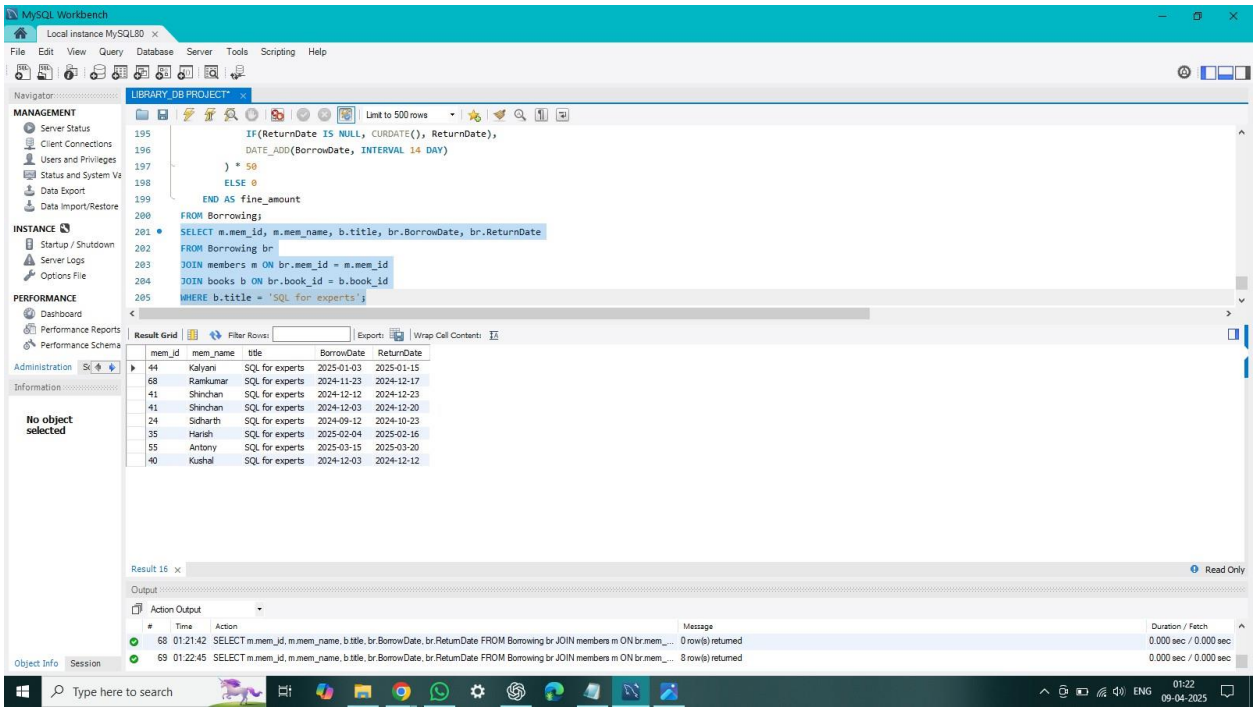
SELECT Borrow_id, mem_id, book_id, BorrowDate,ReturnDate,

DATEDIFF (IF(ReturnDate IS NULL, CURDATE(), ReturnDate),DATE_ADD (BorrowDate, INTERVAL 14 DAY) )AS days_overdue, CASE WHEN DATEDIFF (IF(ReturnDate ISNULL, CURDATE(), ReturnDate), DATE_ADD (BorrowDate, INTERVAL 14 DAY))>0 THEN DATEDIFF(IF(ReturnDate IS NULL, CURDATE(), ReturnDate), DATE_ADD(BorrowDate, INTERVAL 14 DAY)) * 50 ELSE 0 END AS fine_amount FROM Borrowing;



## 10. Find members who borrowed a specific book by book title

SELECT m.mem_id, m.mem_name, b.title, br.BorrowDate, br.ReturnDate FROM Borrowing br JOIN members m ONbr.mem_id=m.mem_id JOIN books b ON br.book_id = b.book_id WHERE b.title = 'SQL for experts';

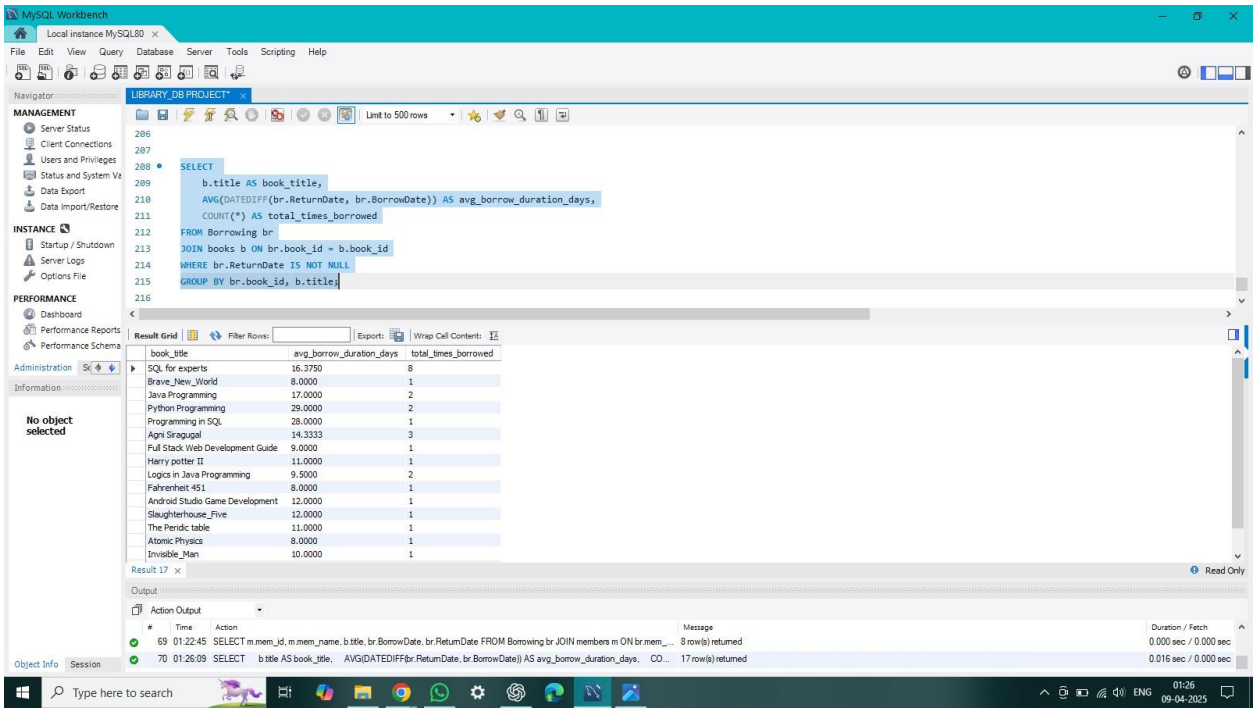


**11. Calculate the average duration a book is borrowed by members. Display the book title, the average duration in days and them number of times it has been borrowed.**

SELECT b.title AS book_title,AVG (DATEDIFF (br.ReturnDate, br.BorrowDate)) AS avg_borrow_duration_days, COUNT(*)AS total_times_borrowed

FROM Borrowing br JOIN books b ON br.book_id=b.book_id

WHERE br.ReturnDate IS NOT NULL GROUP BY br.book_id, b.title;

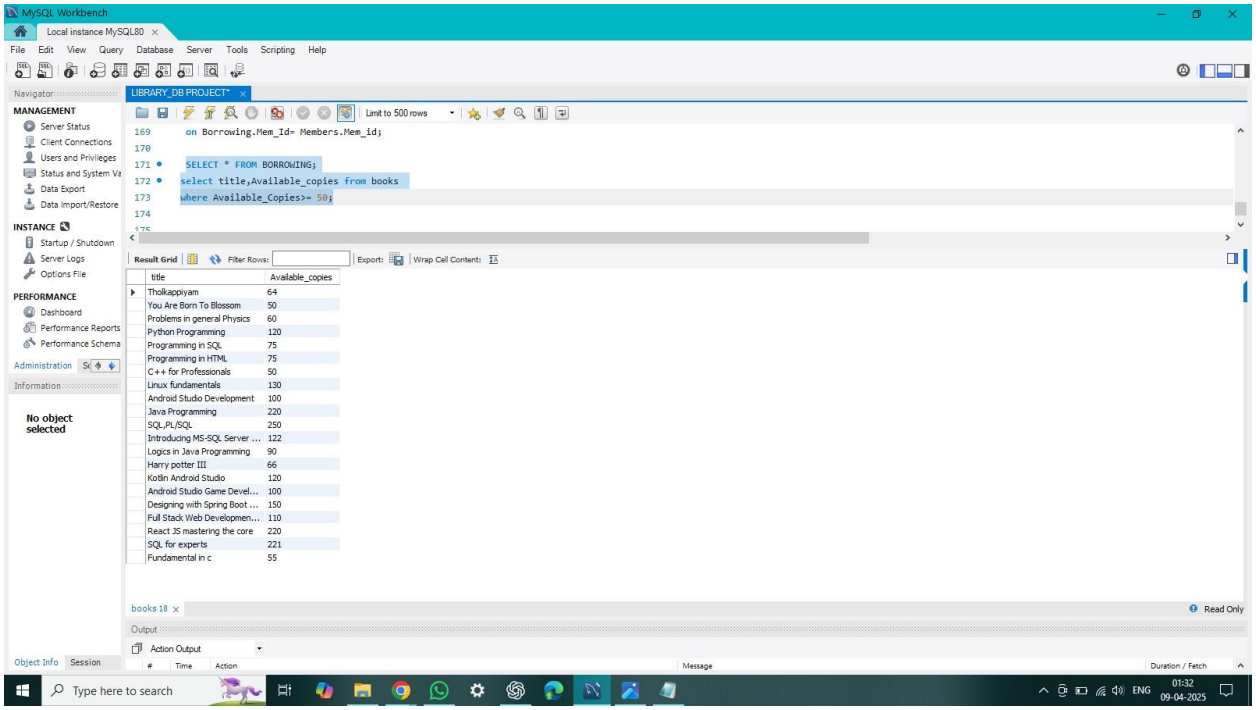


**12. Calculate the average duration a book is borrowed by members. Display the book title, the average duration in days, and the number of times it has been borrowed.**

SELECT*FROMBORROWING;

selecttitle,Available_copiesfrombooks

where Available_Copies>= 50;

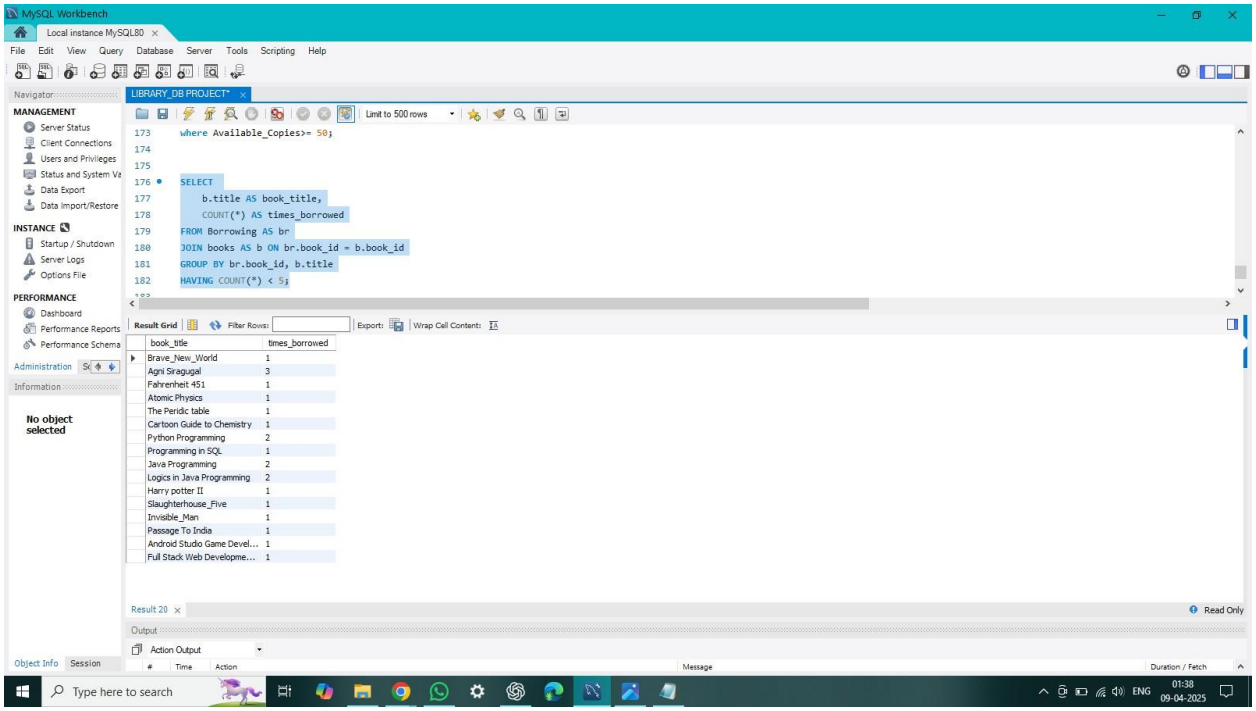


## 13. Filter to show only books borrowed less than 5 times.

SELECT b.title AS book_title, COUNT(*) AS times_borrowed

FROM Borrowing AS br JOIN books AS b ON br.book_id=b.book_id GROUP BY

br.book_id, b.title HAVING COUNT(*)<5;



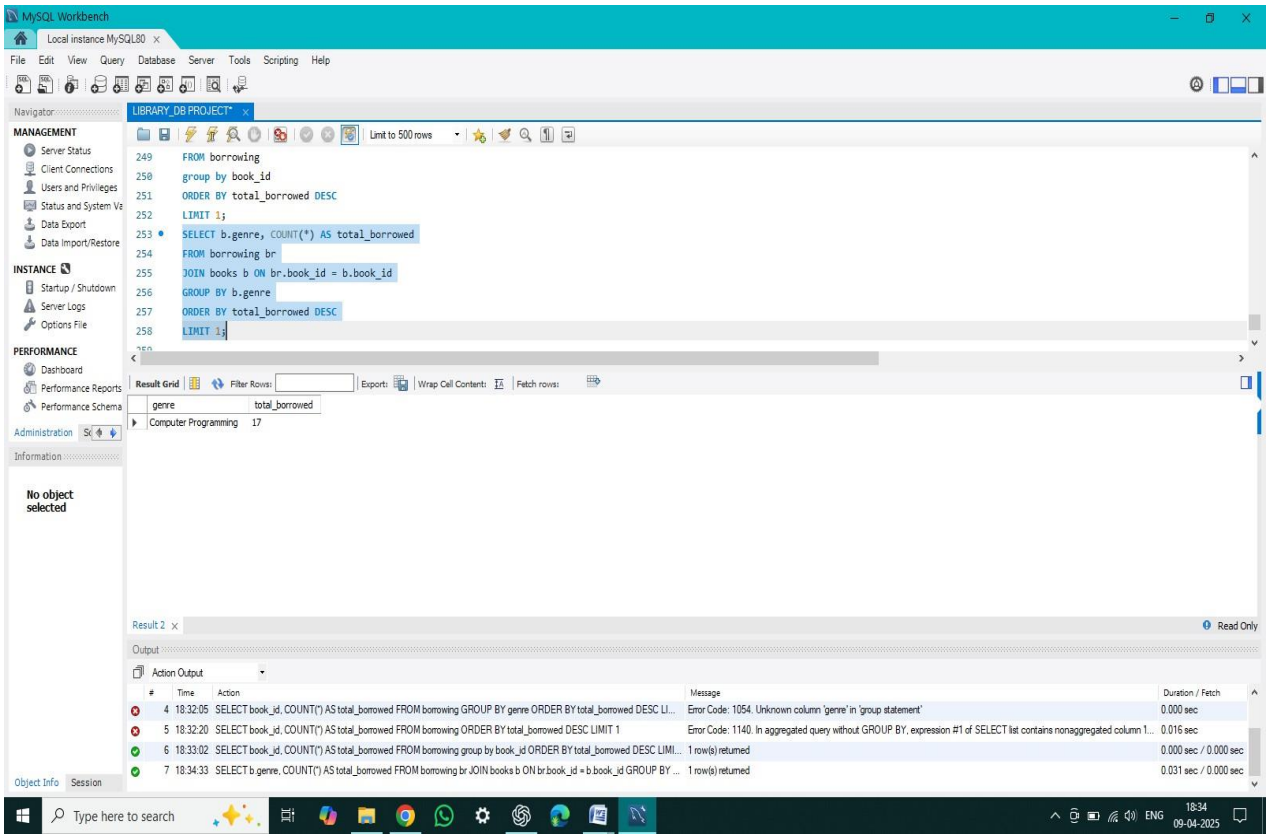
**14. Find the most popular genre in the library. Display the genre with the highest total number of books borrowed.**

SELECT b.genre, COUNT(*) AS total_borrowed FROM

borrowing br JOIN books b ON br.book_id = b.book_id

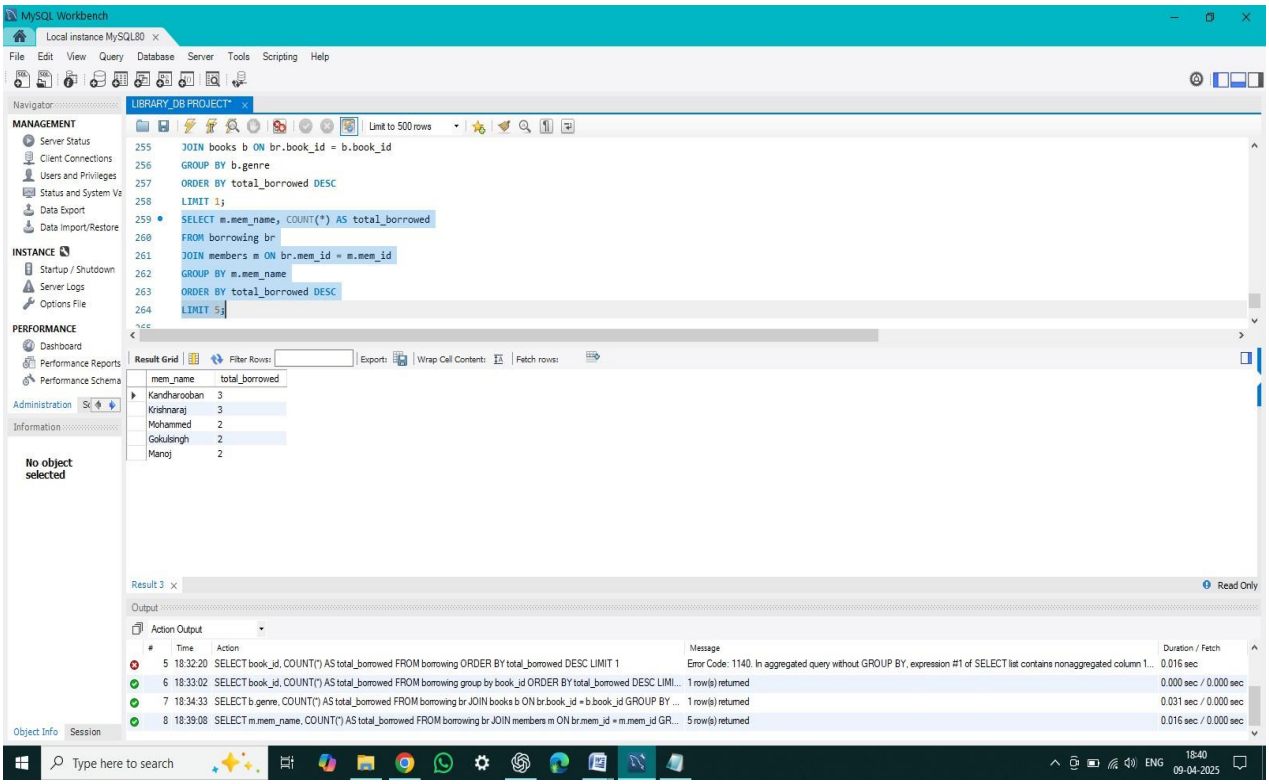
GROUP BY b.genre ORDER BY total_borrowed

DESC LIMIT1;



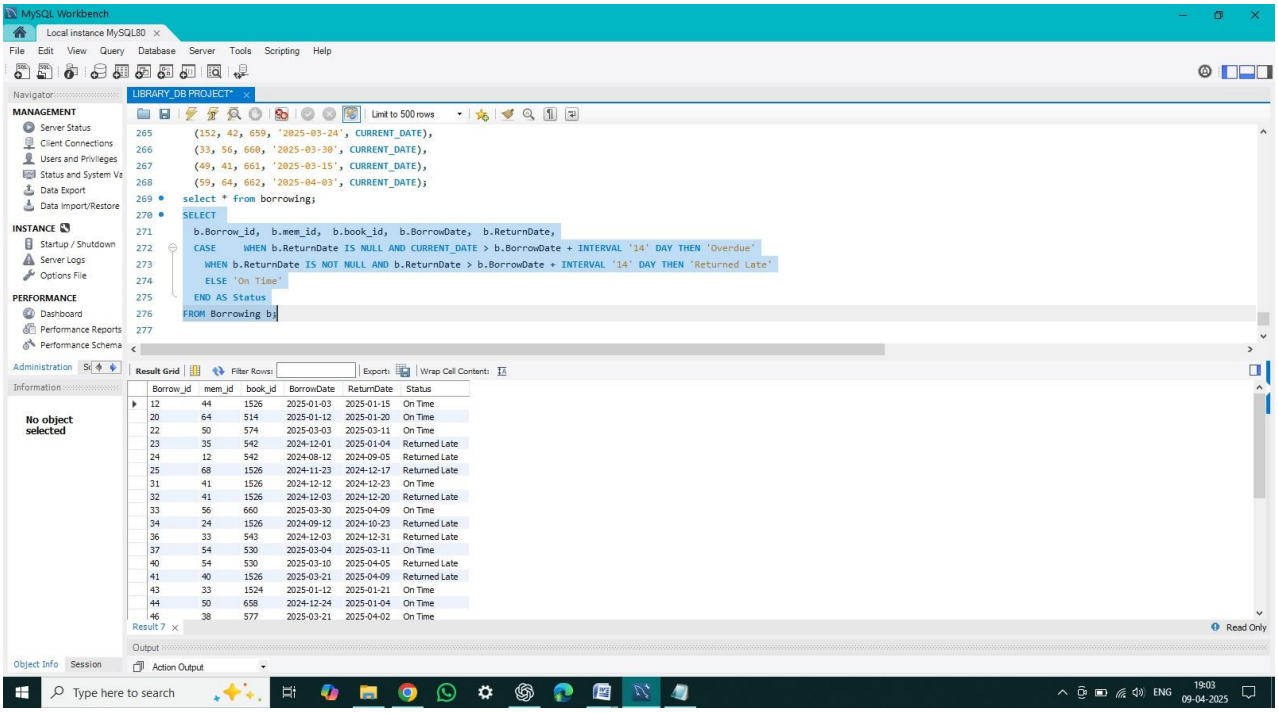
**15. Find the top 5 members who have borrowed the most books. Display their names and the number of books they have borrowed.**

```
SELECT m.mem_name, COUNT(*) AS total_borrowed
FROM  borrowing br
JOIN members m ON br.mem_id = m.mem_id GROUP
BY m.mem_name
ORDER BY total_borrowed DESC
LIMIT 5;
```



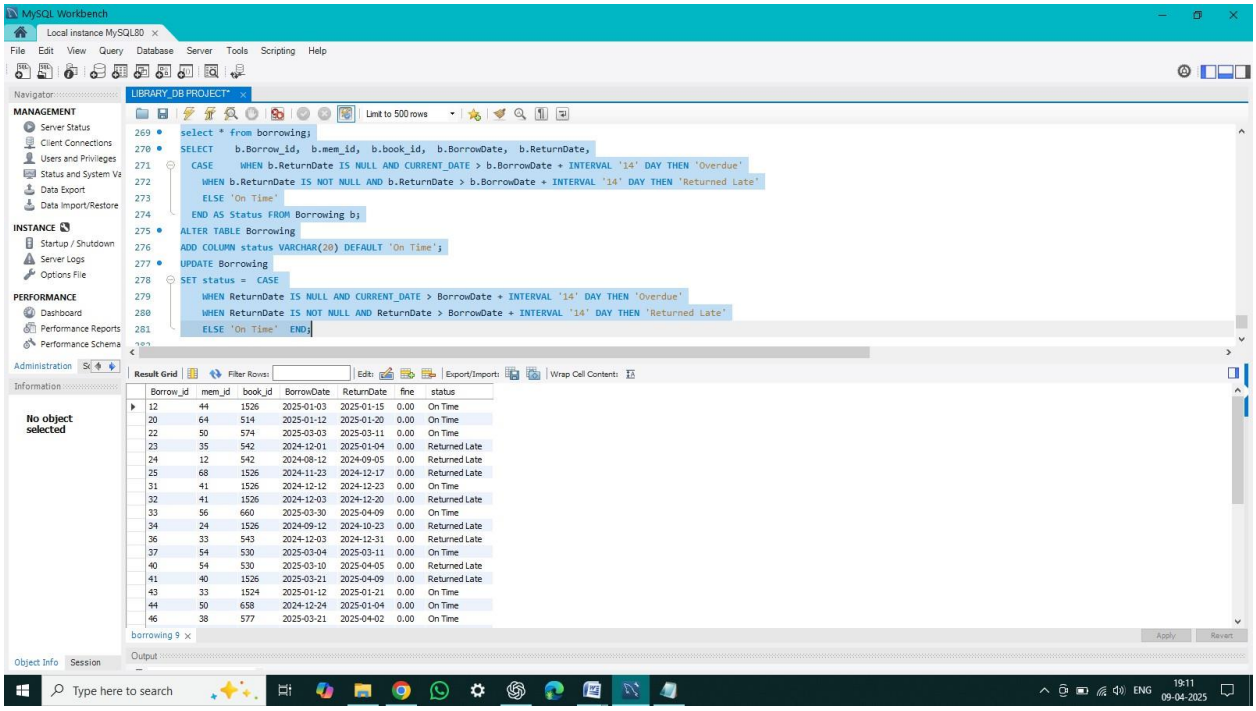
**16. To calculate over due borrowings. Write a query that compares the Return Date with the due date.**

SELECT b.Borrow_id, b.mem_id, b.book_id, b.BorrowDate, b.ReturnDate, CASE WHEN b.ReturnDate IS NULL AND CURRENT_DATE > b.BorrowDate + INTERVAL'14'DAYTHEN'Overdue'
WHEN b.ReturnDate IS NOTNULL AND b.ReturnDate>b.BorrowDate +INTERVAL'14'DAYTHEN'ReturnedLate'
ELSE 'OnTime' END
AS Status FROM
borrowing b;



## 17. Query to add status in borrowing table as default ALTER TABLE Borrowing.

ADD COLUMN status VARCHAR(20)DEFAULT 'OnTime';
UPDATE Borrowing
SET status =CASE
WHEN ReturnDate IS NULLANDCURRENT_DATE>BorrowDate+INTERVAL'14' DAY THEN 'Overdue'
WHEN ReturnDate ISNOT NULL AND ReturnDate > BorrowDate + INTERVAL'14' DAY THEN 'Returned Late'
ELSE'OnTime' END;



## 18. Calculate and add fine in borrowing table.

```
   UPDATE Borrowing
SET fine=          CASE
      WHEN Return DateIS NOT NULL AND ReturnDate>BorrowDate+INTERVAL14 DAY
THEN
          DATEDIFF(ReturnDate,BorrowDate+INTERVAL14DAY)*50
      WHEN  ReturnDate ISNULL AND CURRENT_DATE>BorrowDate+ INTERVAL 14
DAY THEN
          DATEDIFF(CURRENT_DATE,BorrowDate+INTERVAL14DAY)*50              ELSE0
END;
```