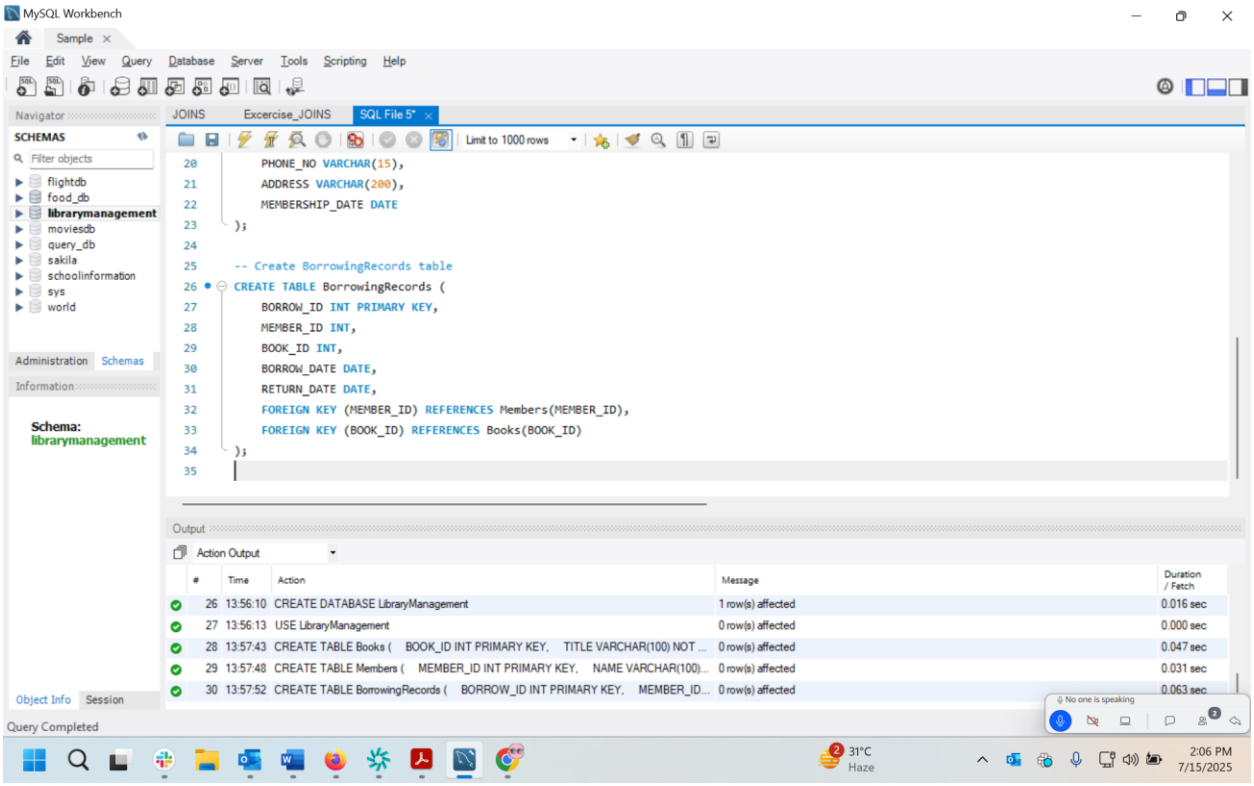
Prepared by- Shubham Mishra
Email – mshraea@amazon.com
GitHub Account - https://github.com/mshraea

I have used SQL workbench for all 4 tasks.

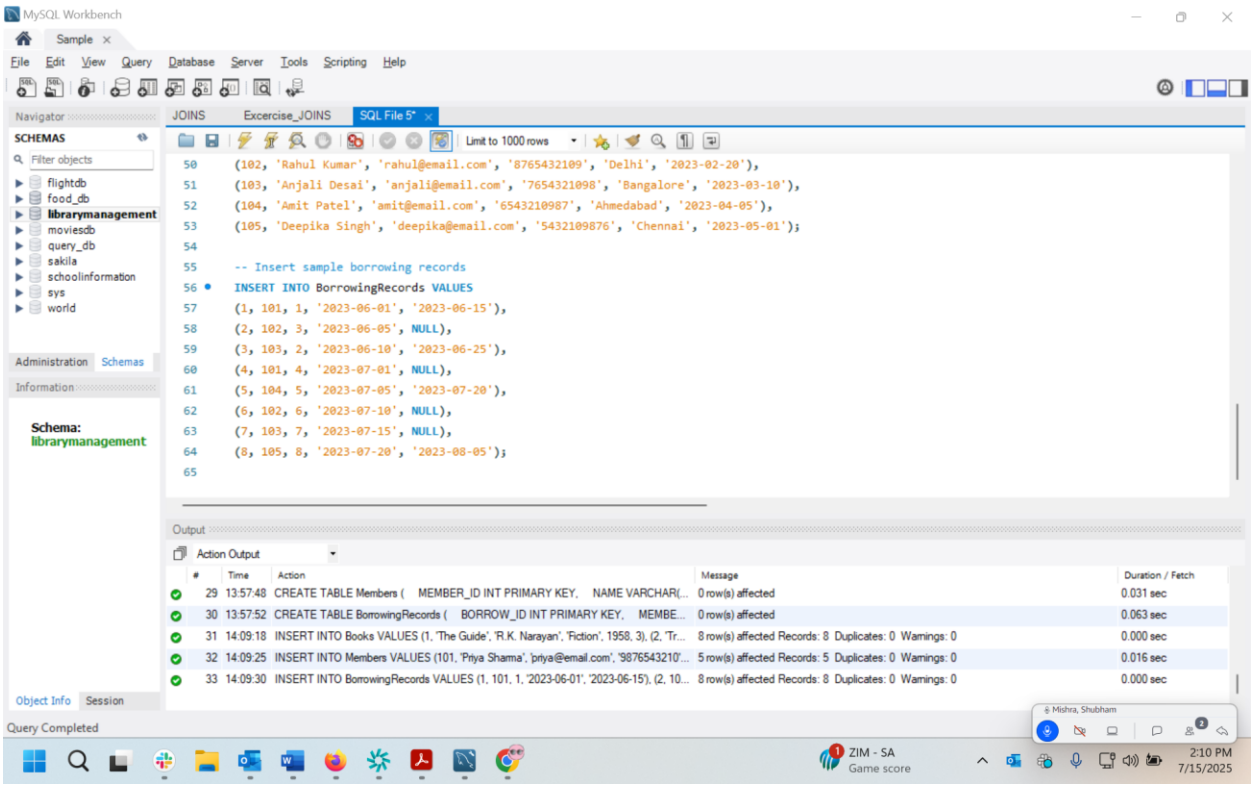
Task 1

Project Title: Library Management System (using SQL)

Database Creation:

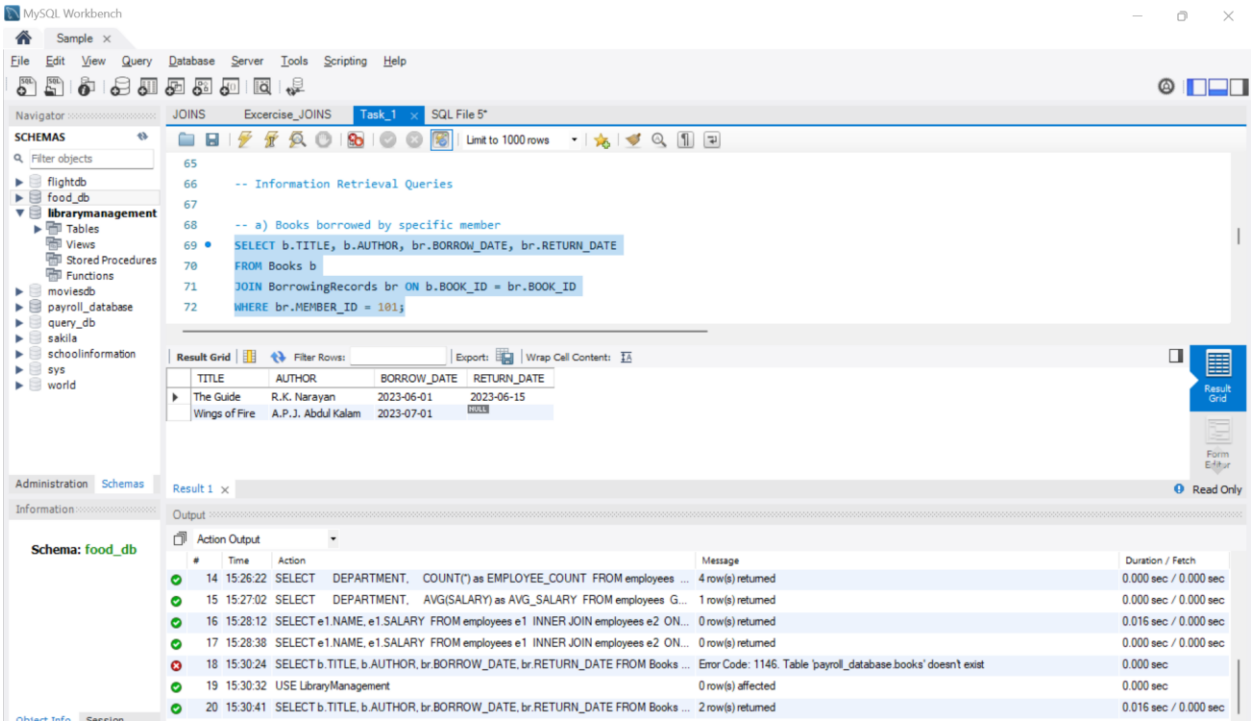


Data Creation:

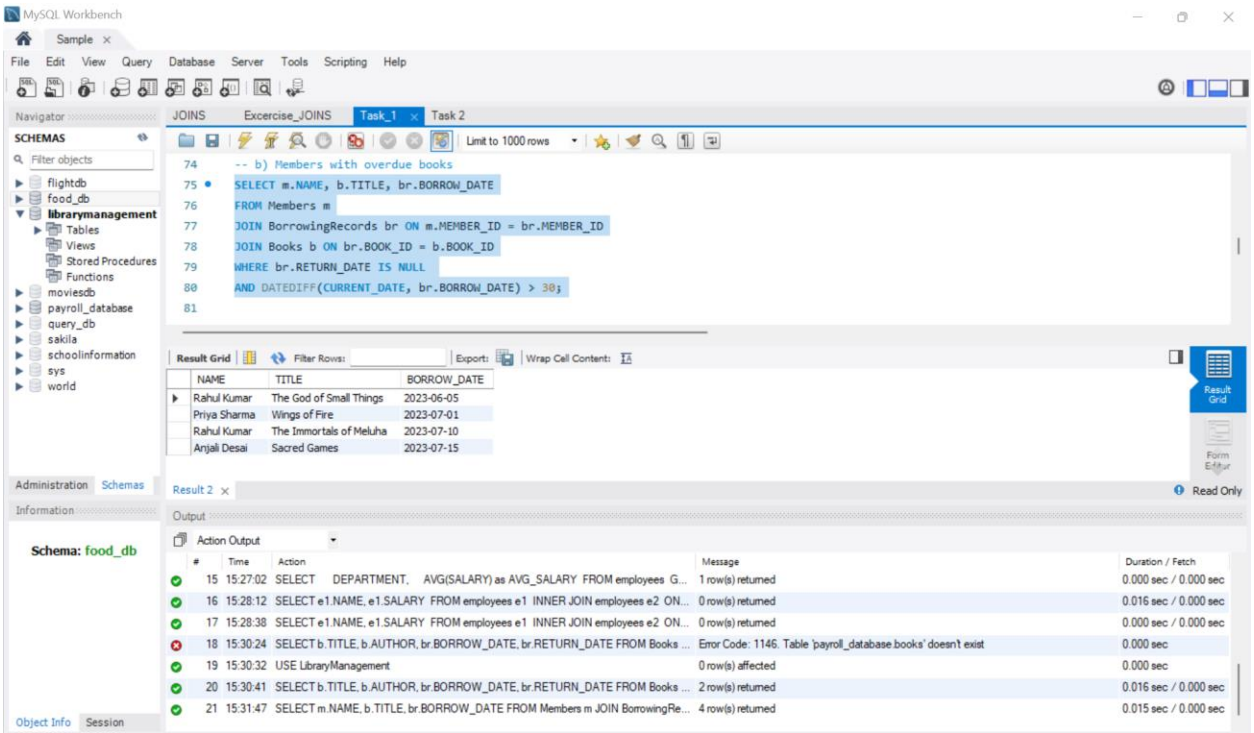


Information Retrieval:

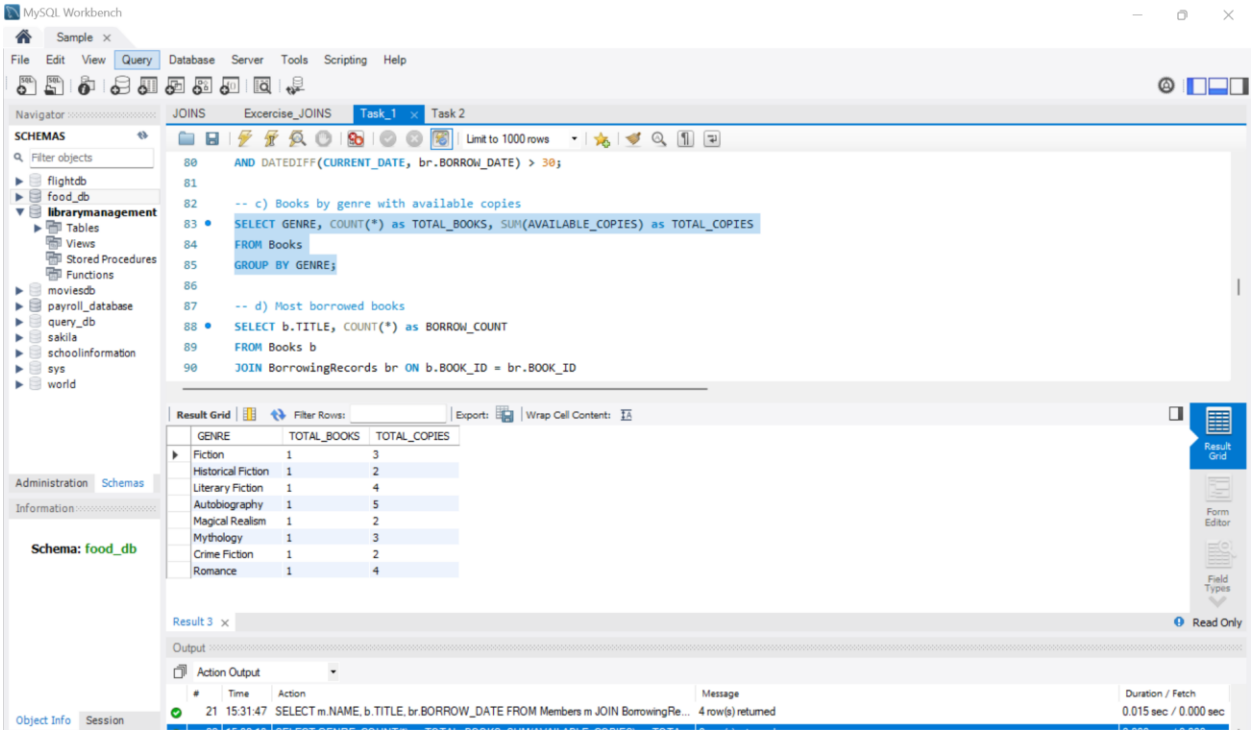
a) Retrieve a list of books currently borrowed by a specific member.



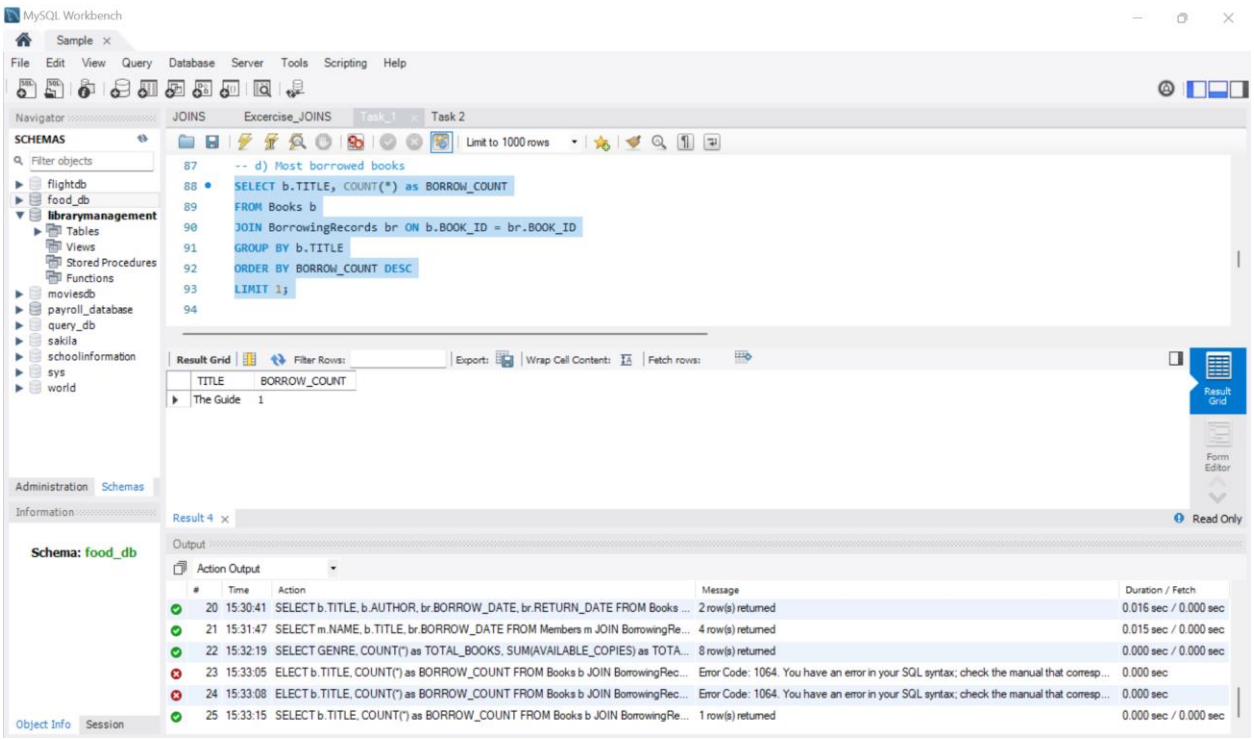
b) Find members who have overdue books (borrowed more than 30 days ago, not returned)



c) Retrieve books by genre along with the count of available copies.

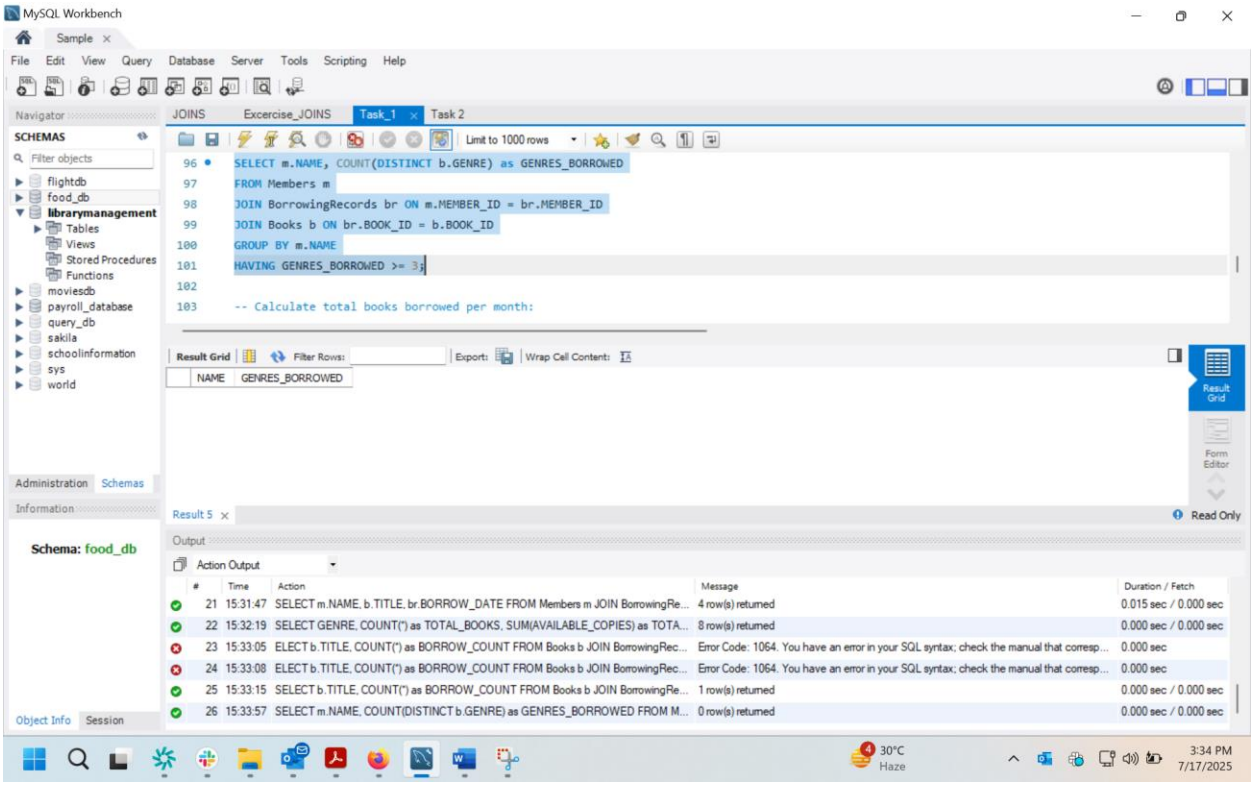


d) Find the most borrowed book(s) overall.



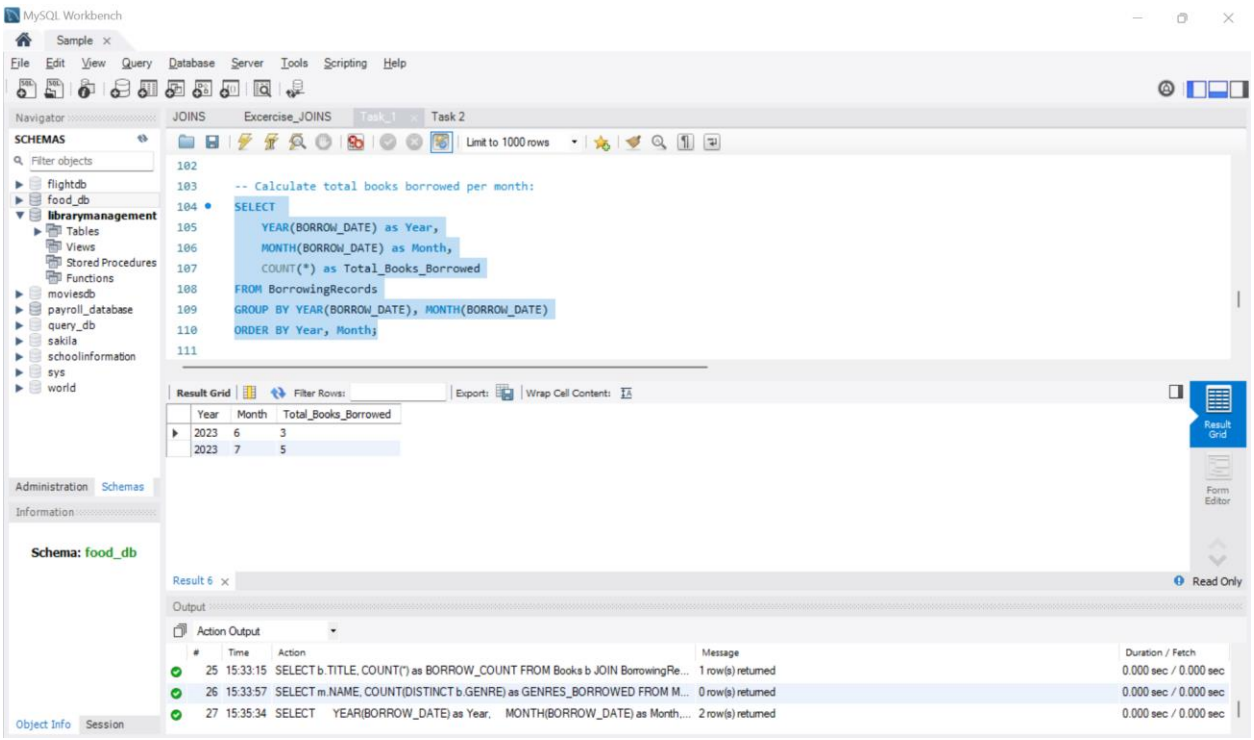
e) Retrieve members who have borrowed books from at least three different genres.

There are no members who have borrowed at least 3+ genres

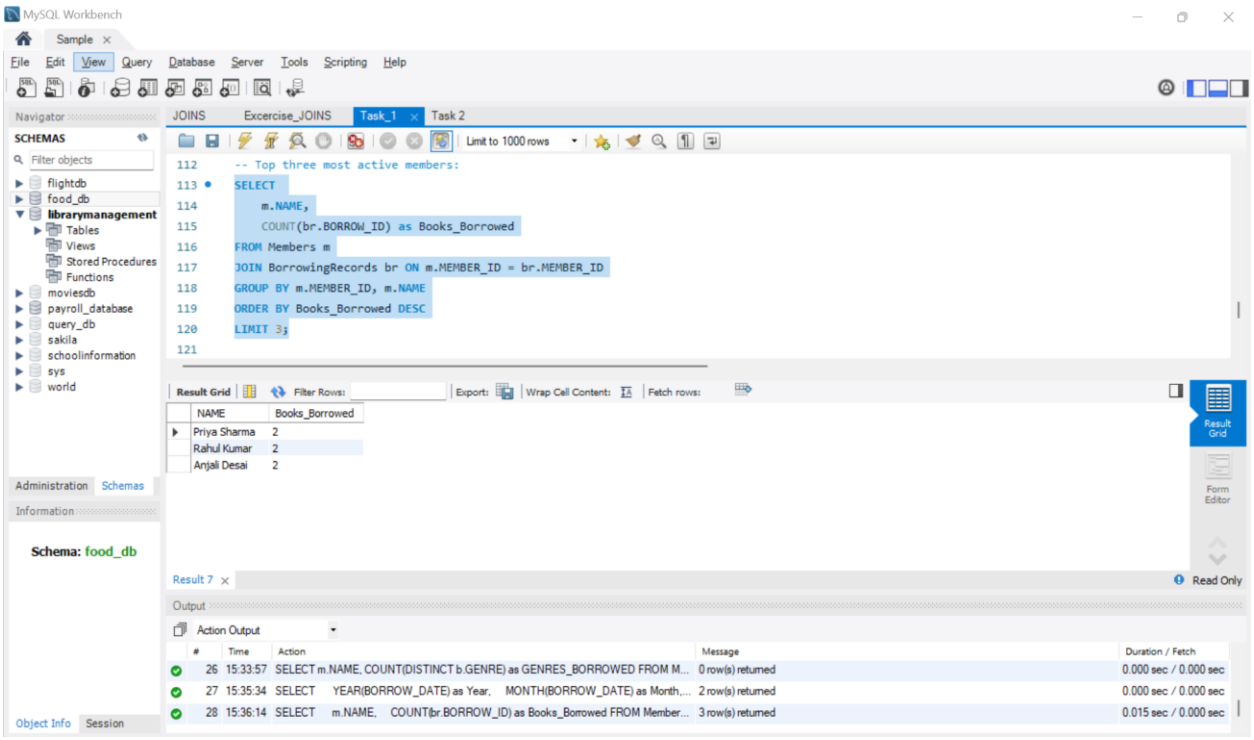


Reporting and Analytics:

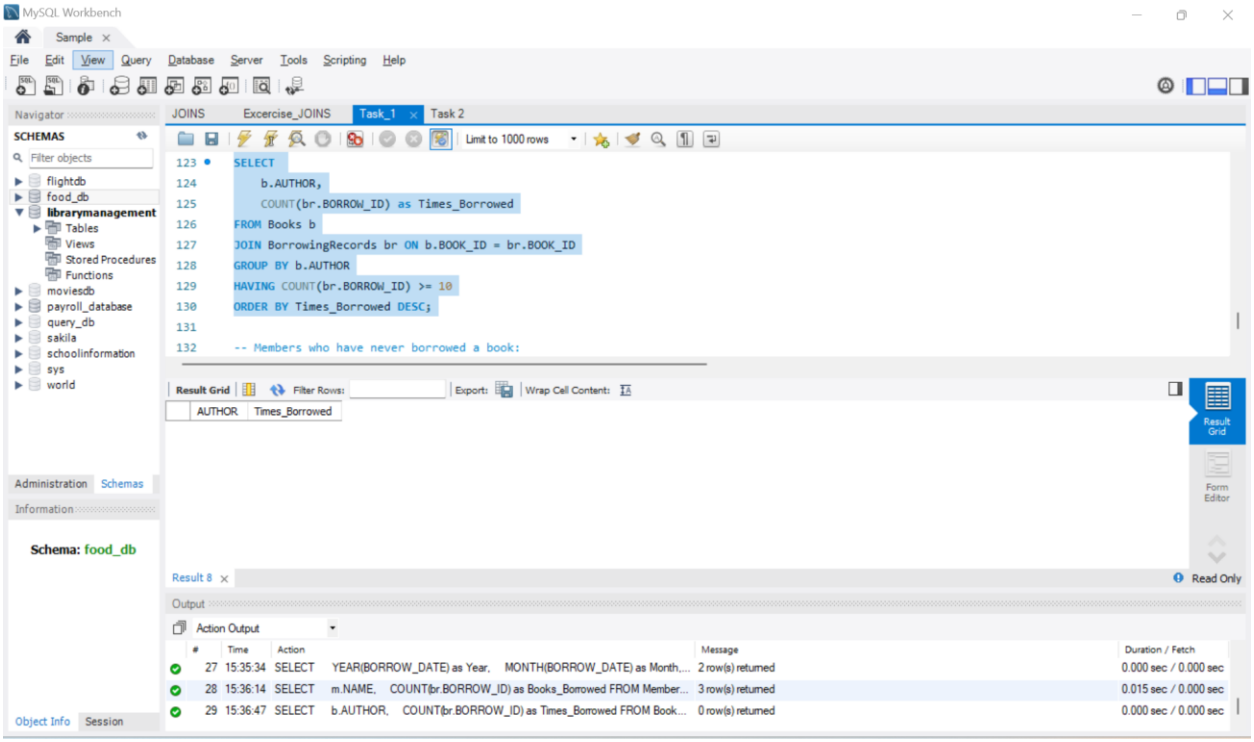
a) Calculate the total books borrowed per month:



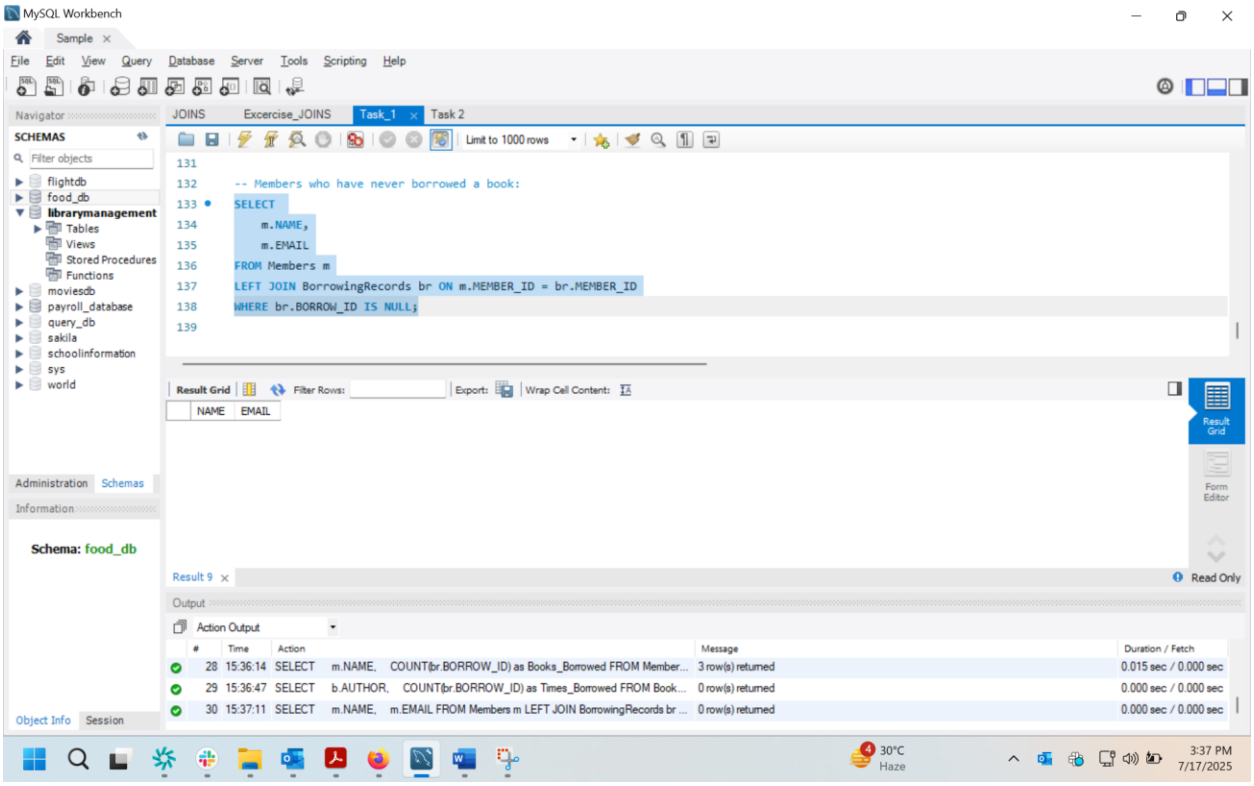
b) Top three most active members:



c) Authors whose books were borrowed at least 10 times: **No Records**



d) Members who have never borrowed a book:

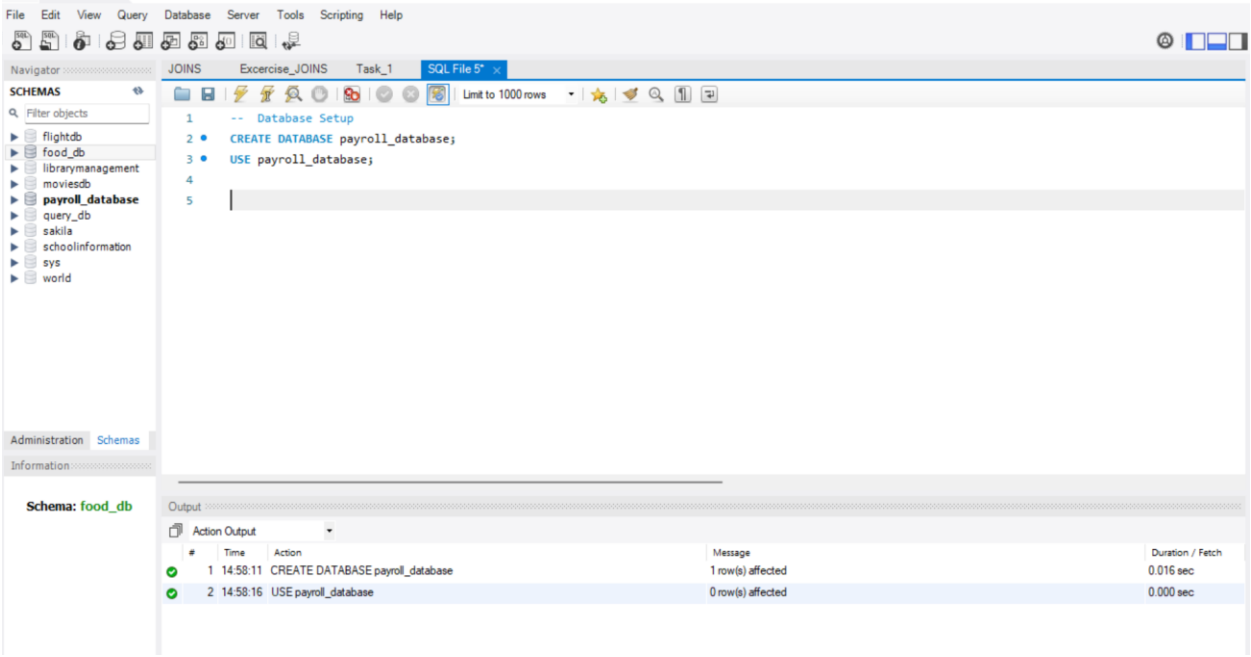


TASK 2

Project: Employee Payroll Management System

Database Setup:

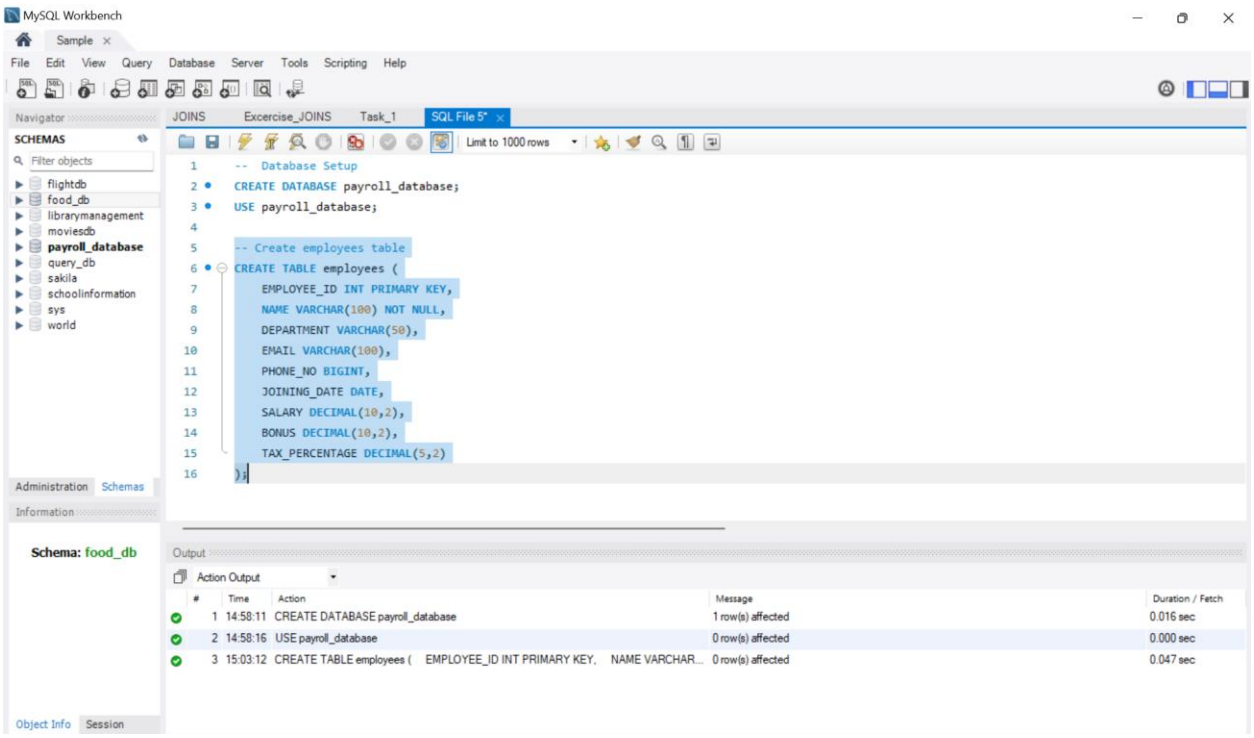
Create a database named payroll database.



Create a table employee with columns: EMPLOYEE_ID (integer), NAME (text),

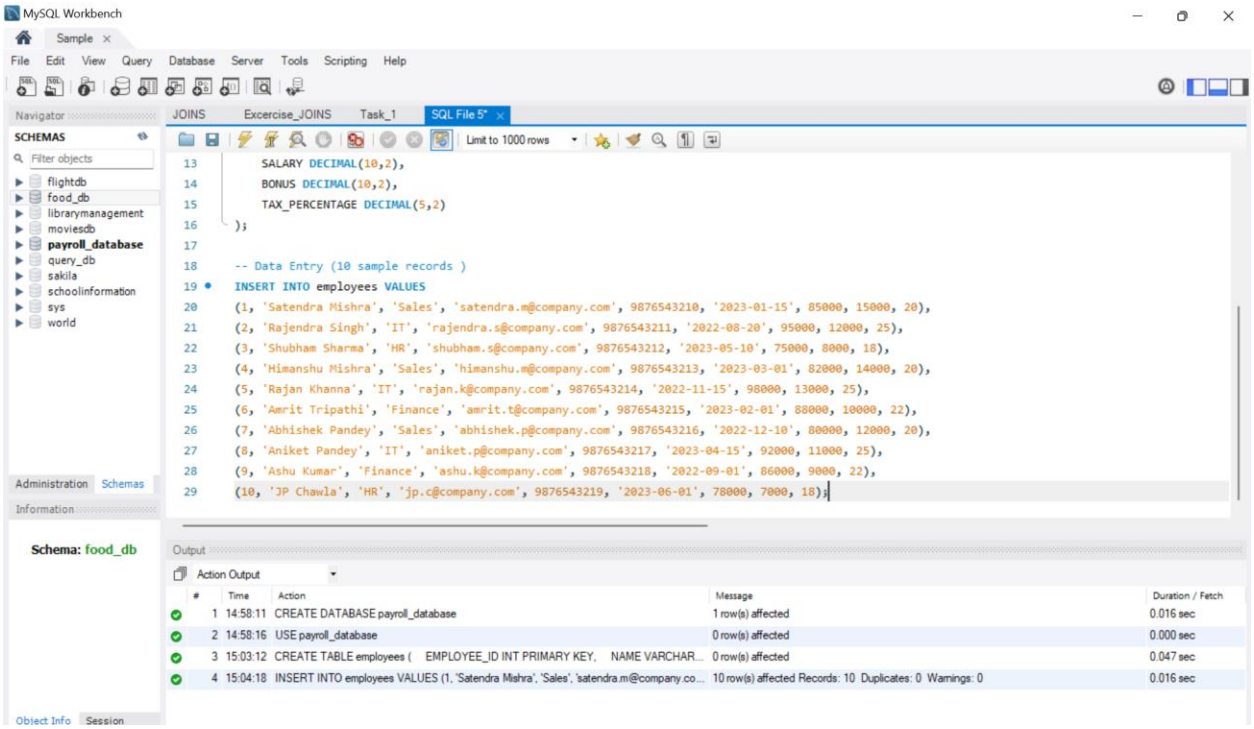
DEPARTMENT (text), EMAIL (text), PHONE_NO (numeric), JOINING_DATE (date), SALARY

(numeric), BONUS (numeric), TAX_PERCENTAGE (numeric).



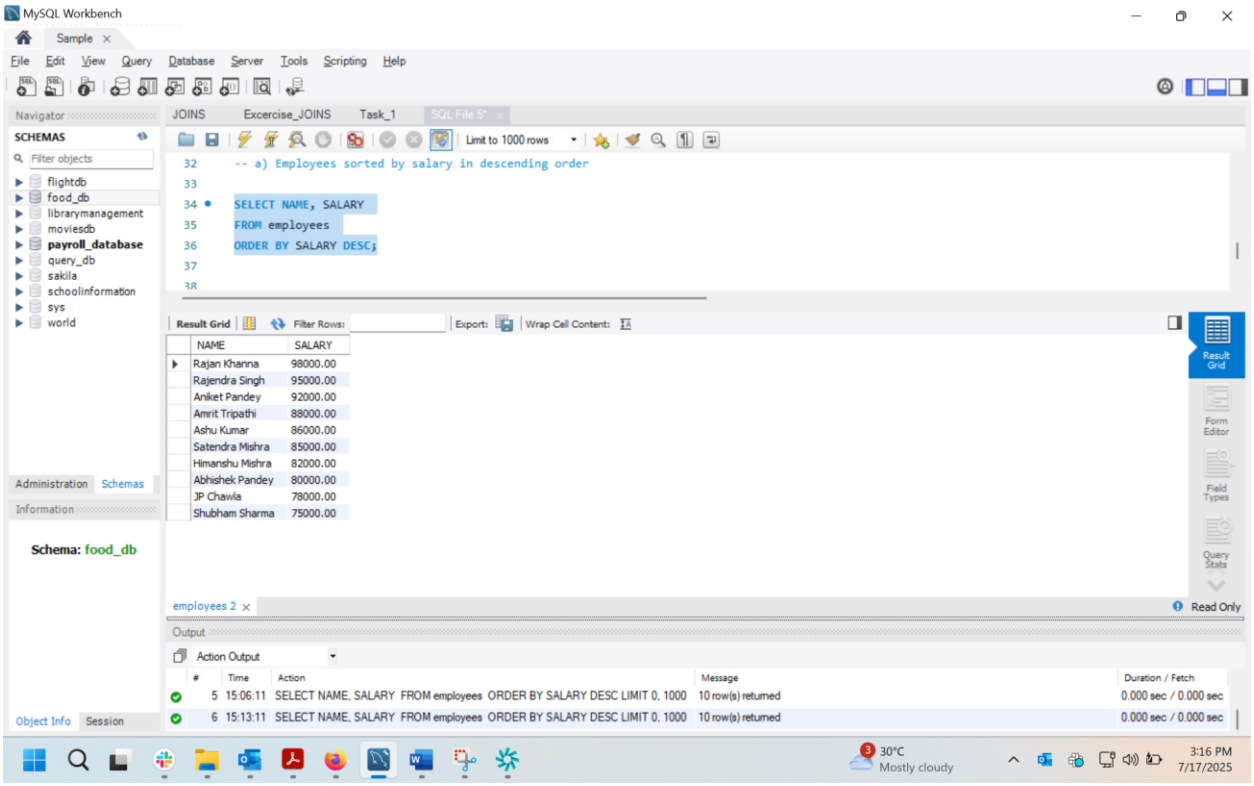
Data Entry:

Insert 10 sample employee records.

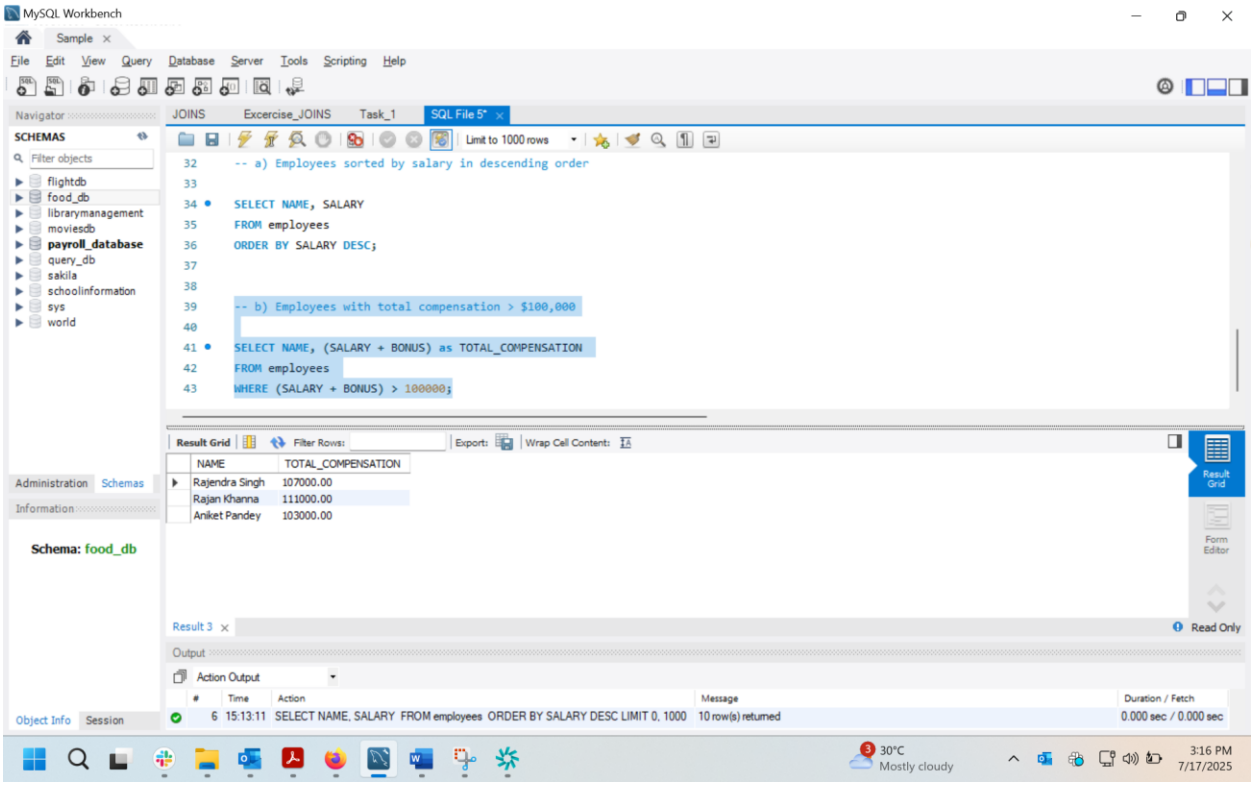


Payroll Queries:

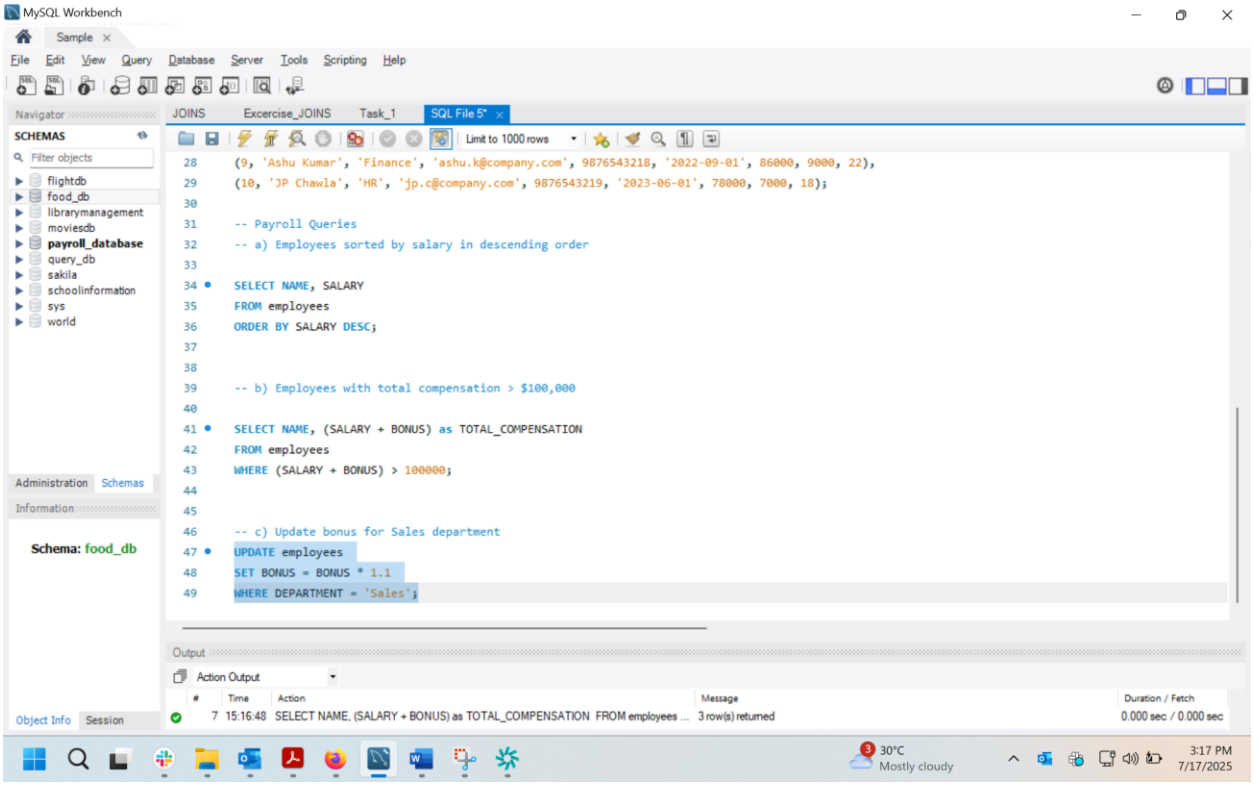
a) Retrieve the list of employees sorted by salary in descending order



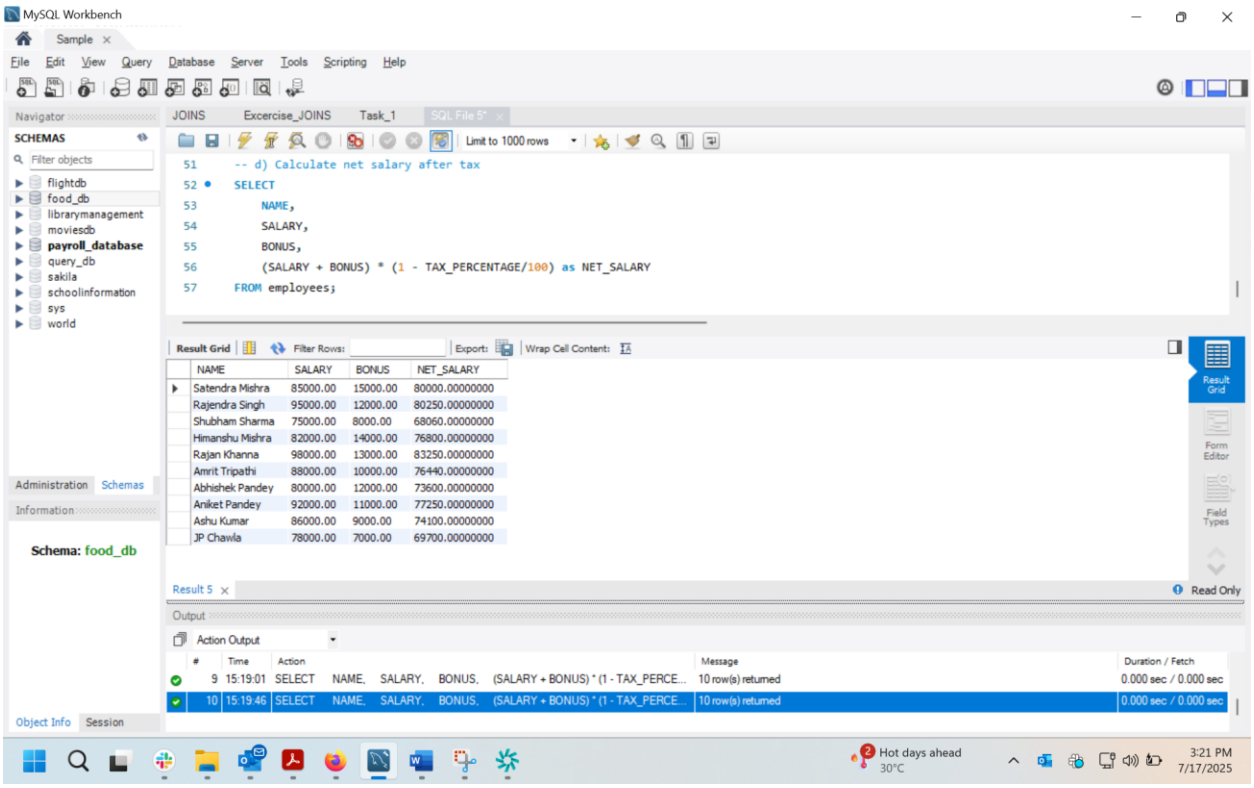
b) Find employees with total compensation (SALARY + BONUS) greater than $100,000.



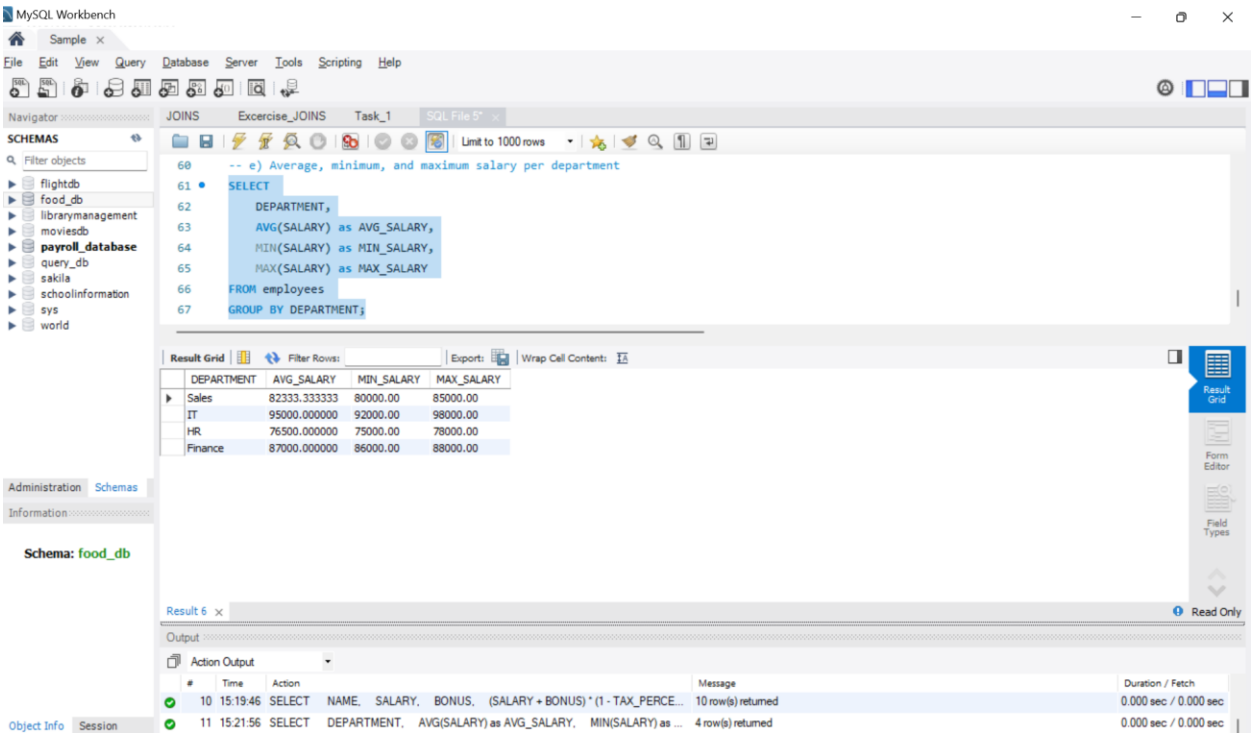
c) Update the bonus for employees in the ‘Sales’ department by 10%.



d) Calculate the net salary after deducting tax for all employees.

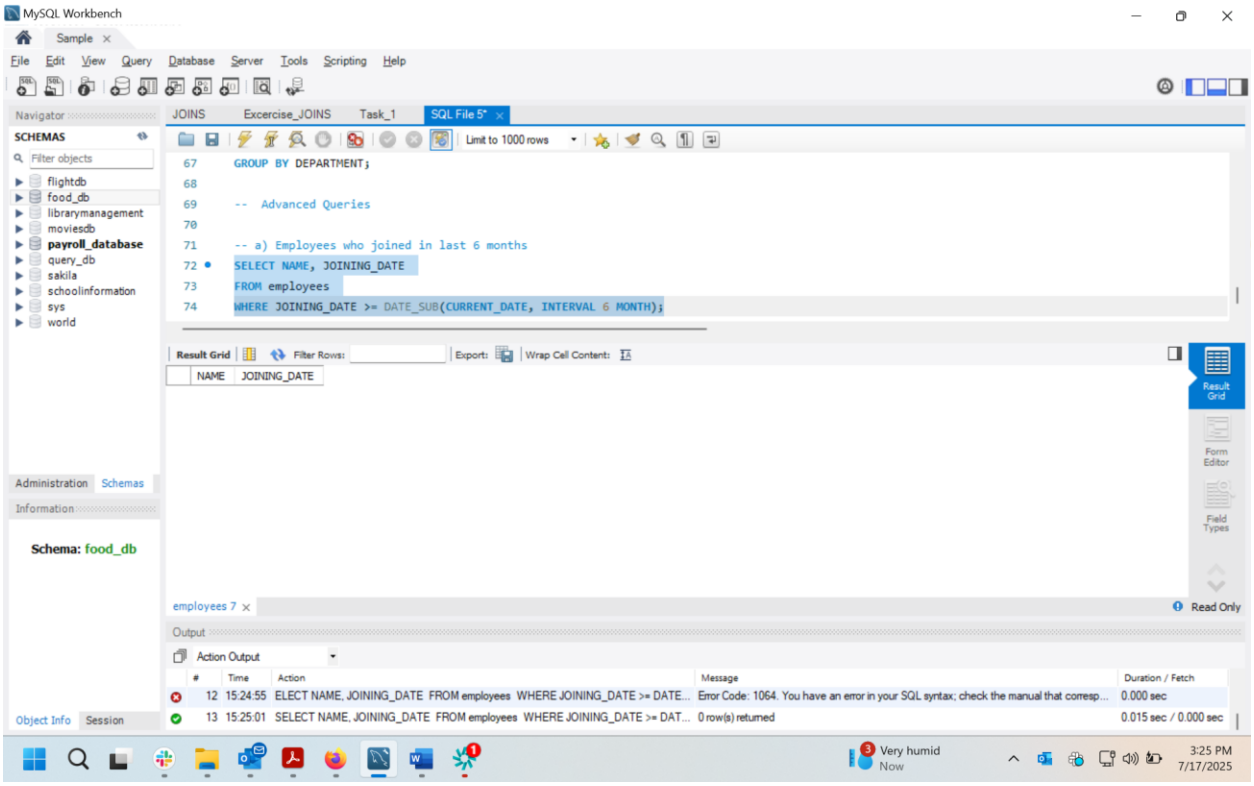


e) Retrieve the average, minimum, and maximum salary per department.

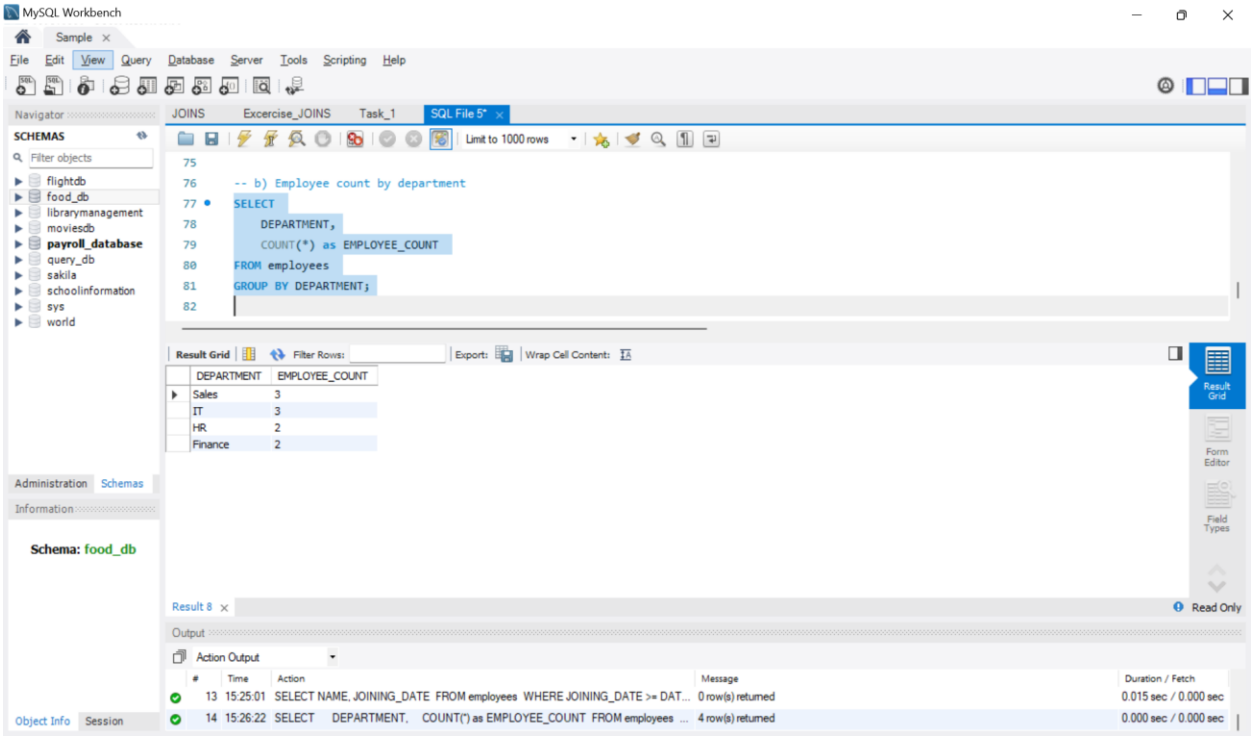


Advanced Queries:

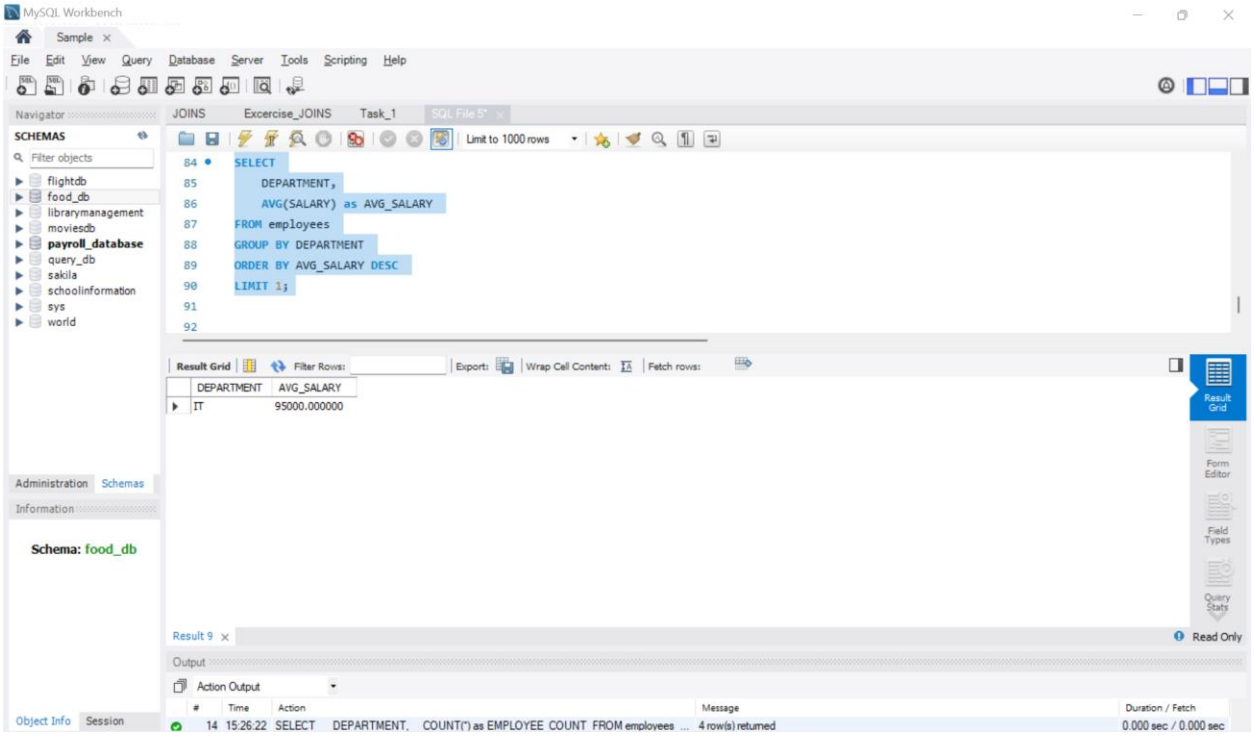
a) Retrieve employees who have joined in the last 6 months.



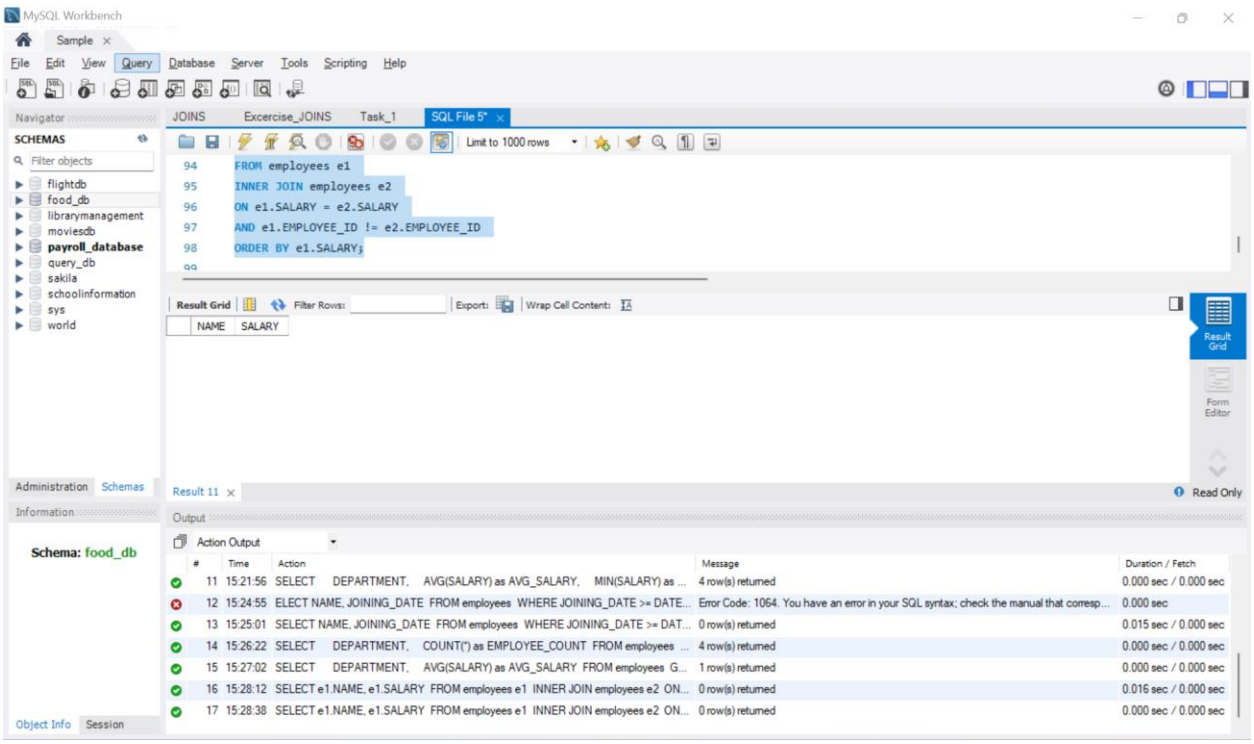
b) Group employees by department and count how many employees each has.



c) Find the department with the highest average salary.



d) Identify employees who have the same salary as at least one other employee.

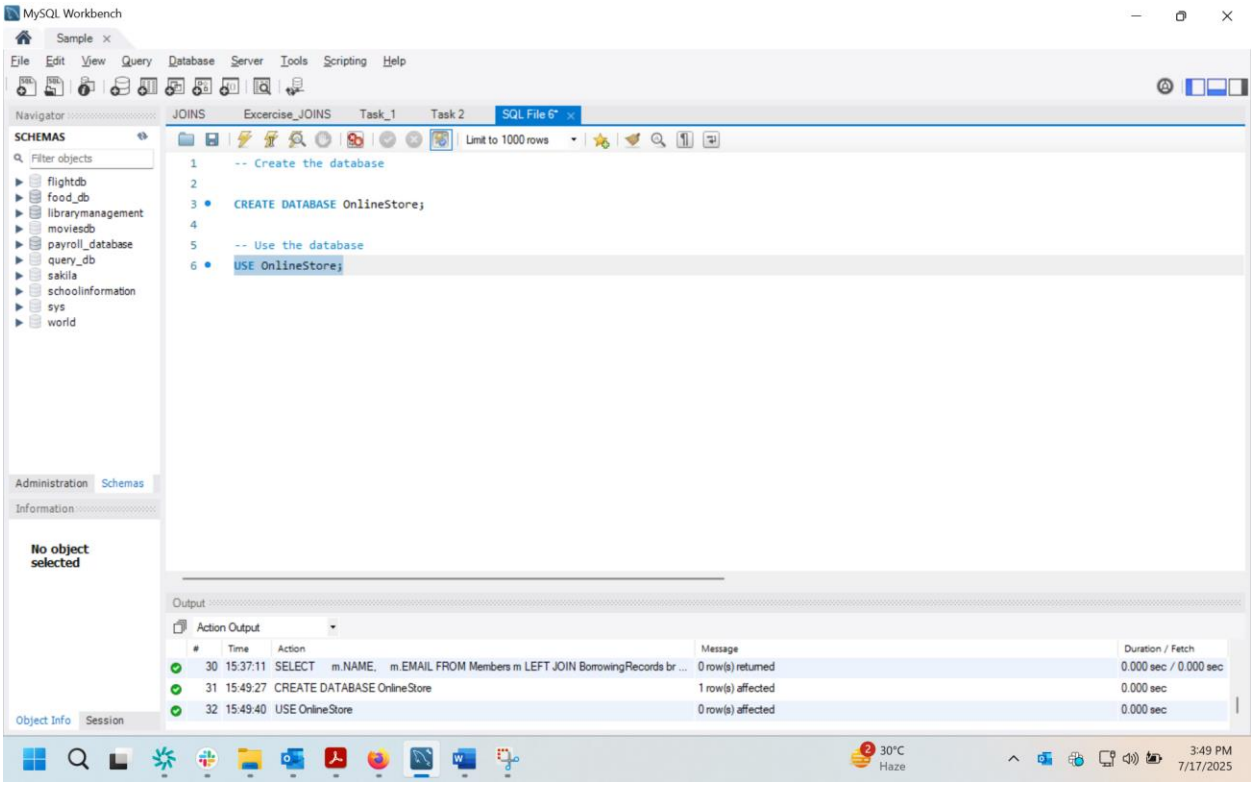


Task 3

Project: Online Store Order Management System

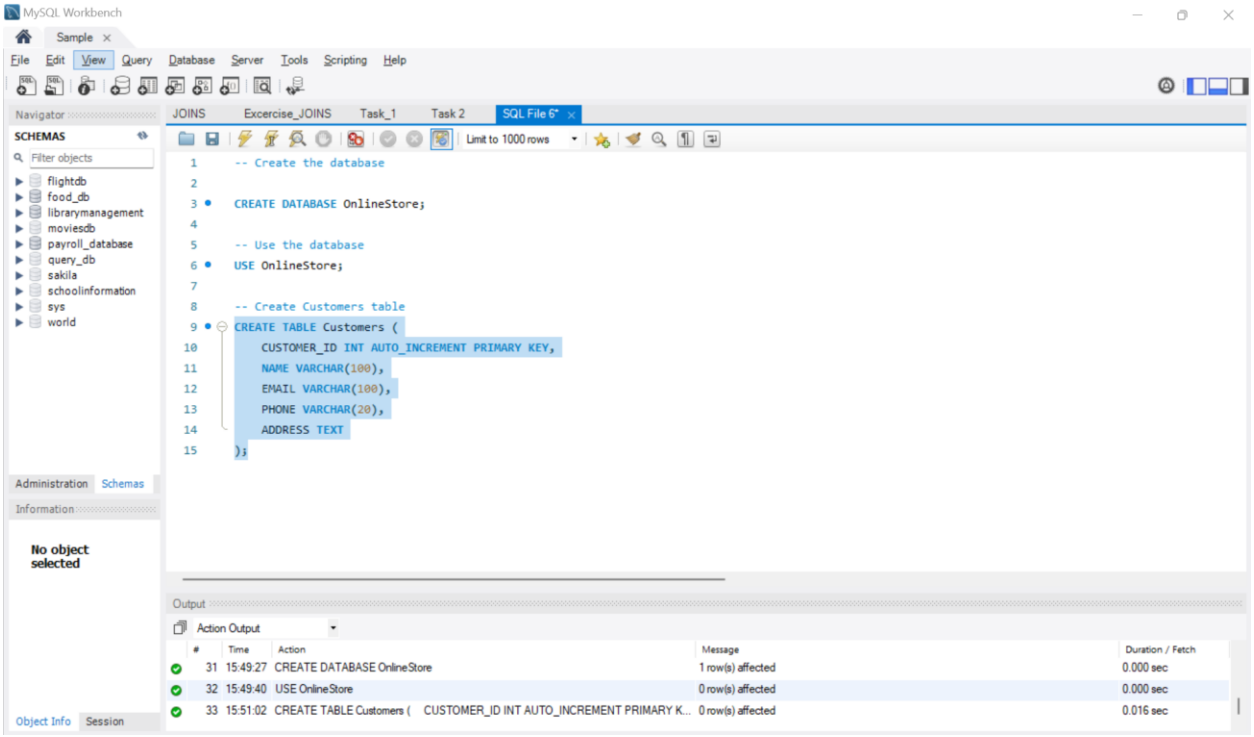
Database Creation:

Create a database named Online Store.

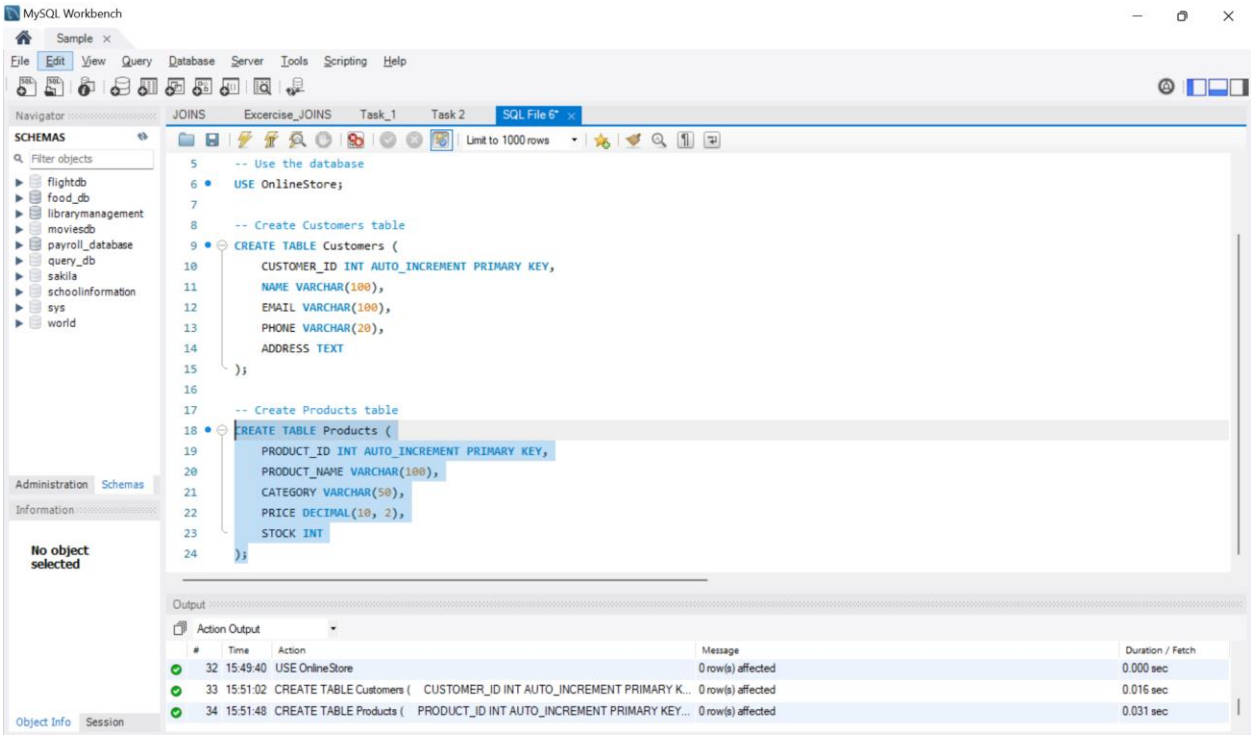


Create tables:

Customers (CUSTOMER_ID, NAME, EMAIL, PHONE, ADDRESS)

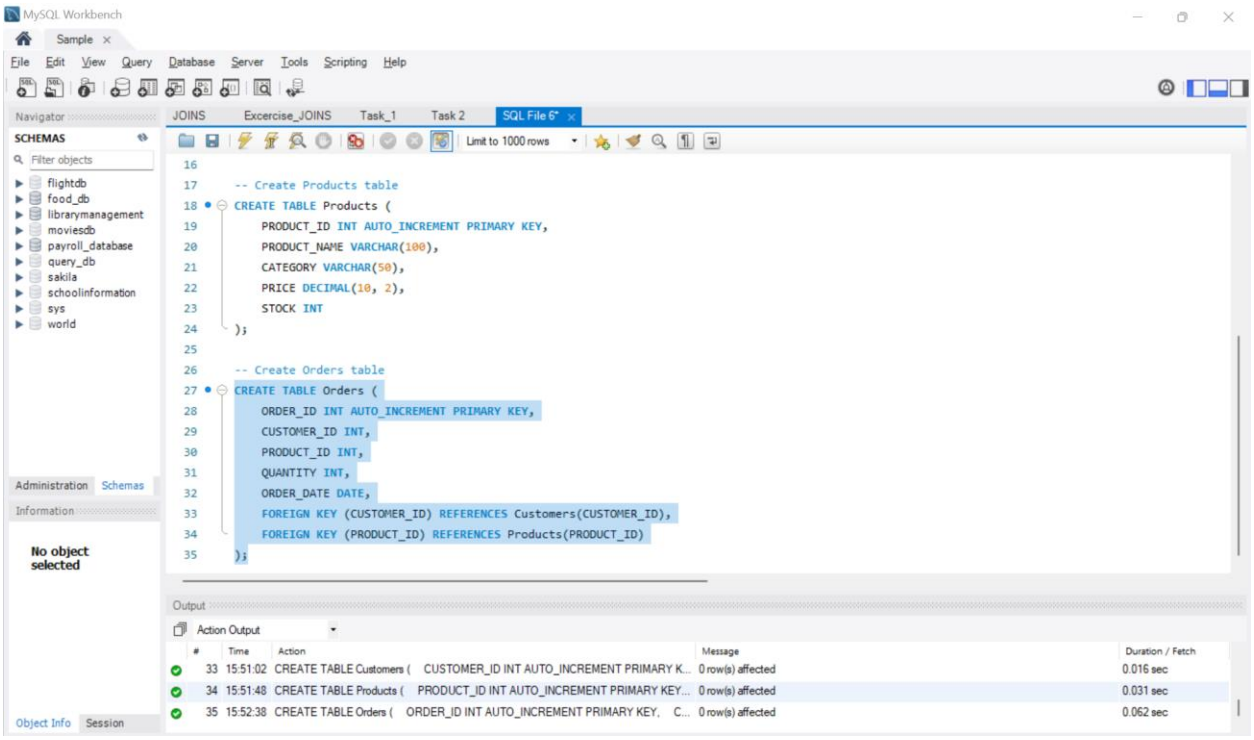


Products (PRODUCT_ID, PRODUCT_NAME, CATEGORY, PRICE, STOCK)



Orders (ORDER_ID, CUSTOMER_ID, PRODUCT_ID, QUANTITY, ORDER_DATE)

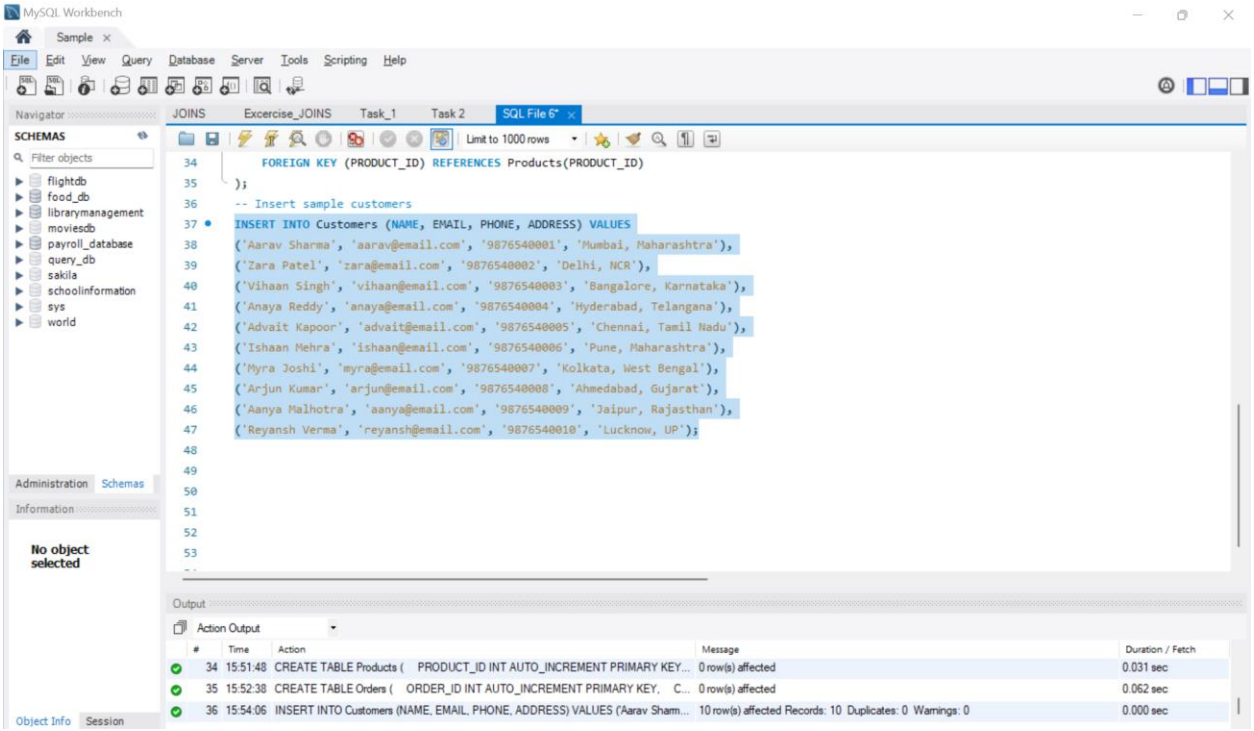
Set up foreign keys linking Orders to Customers and Products



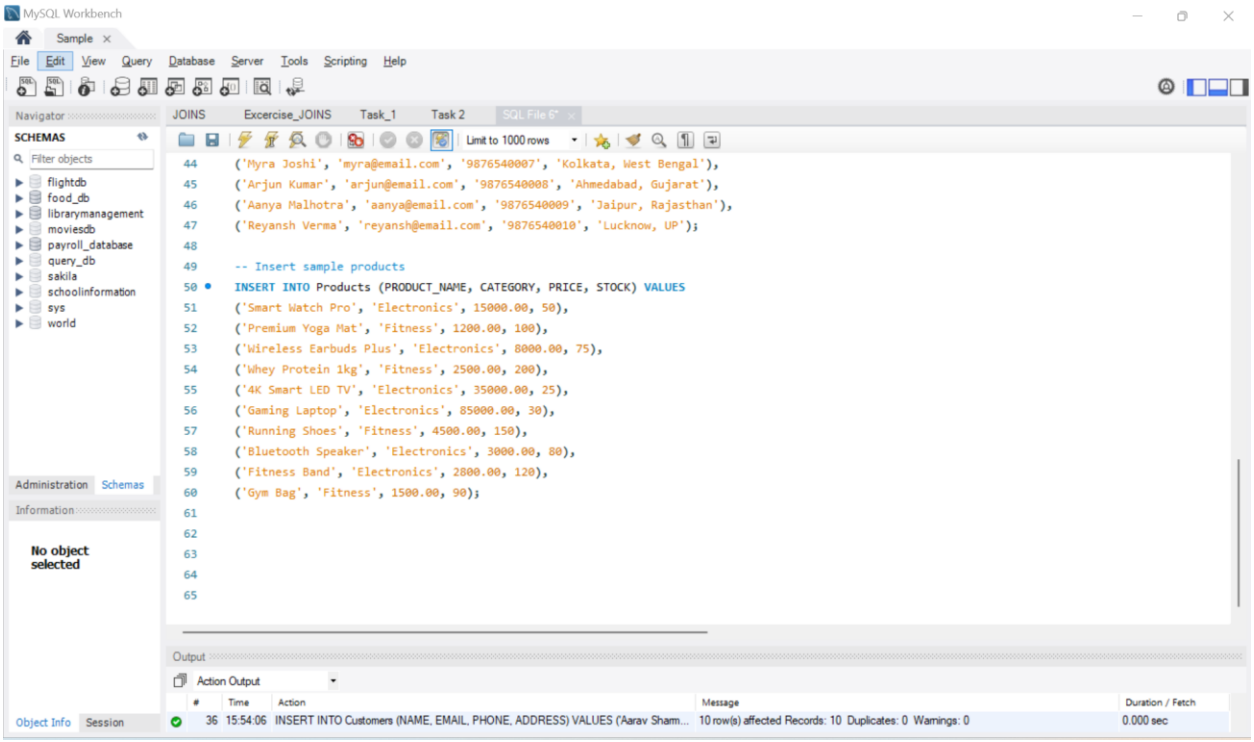
Data Creation:

Insert sample data for customers, products, and orders.

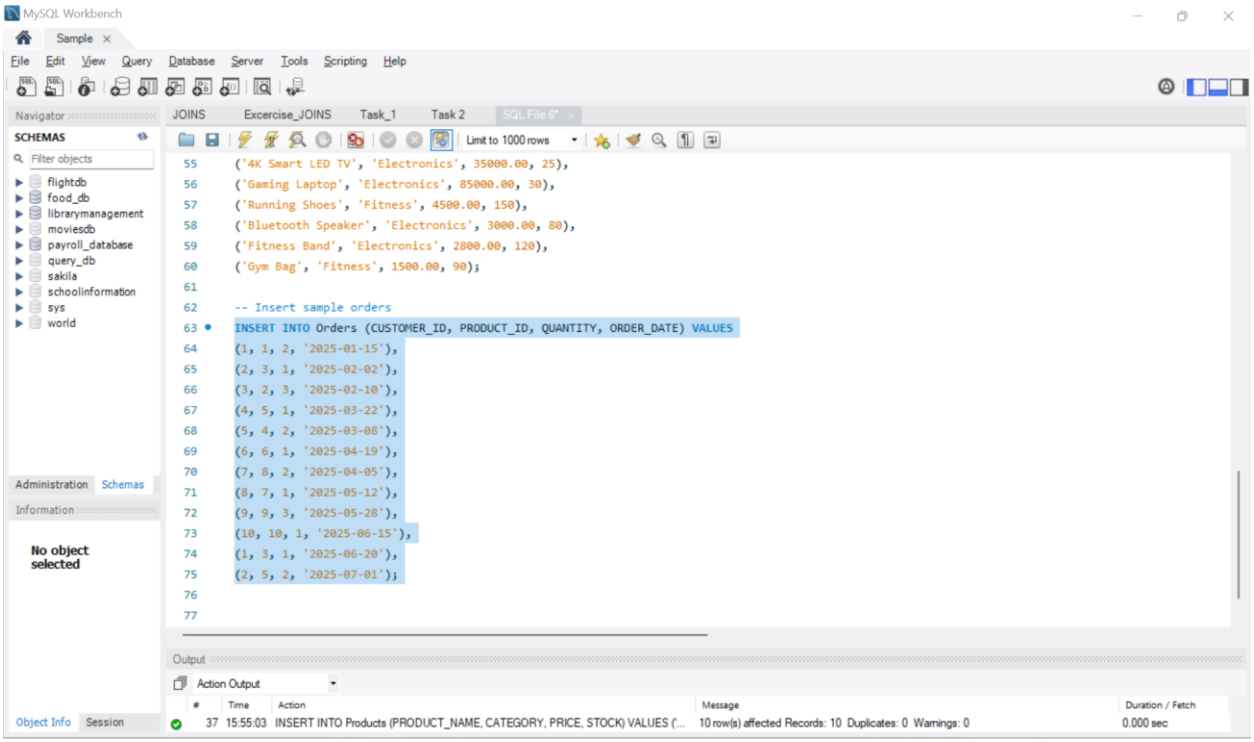
Customers-



Products-

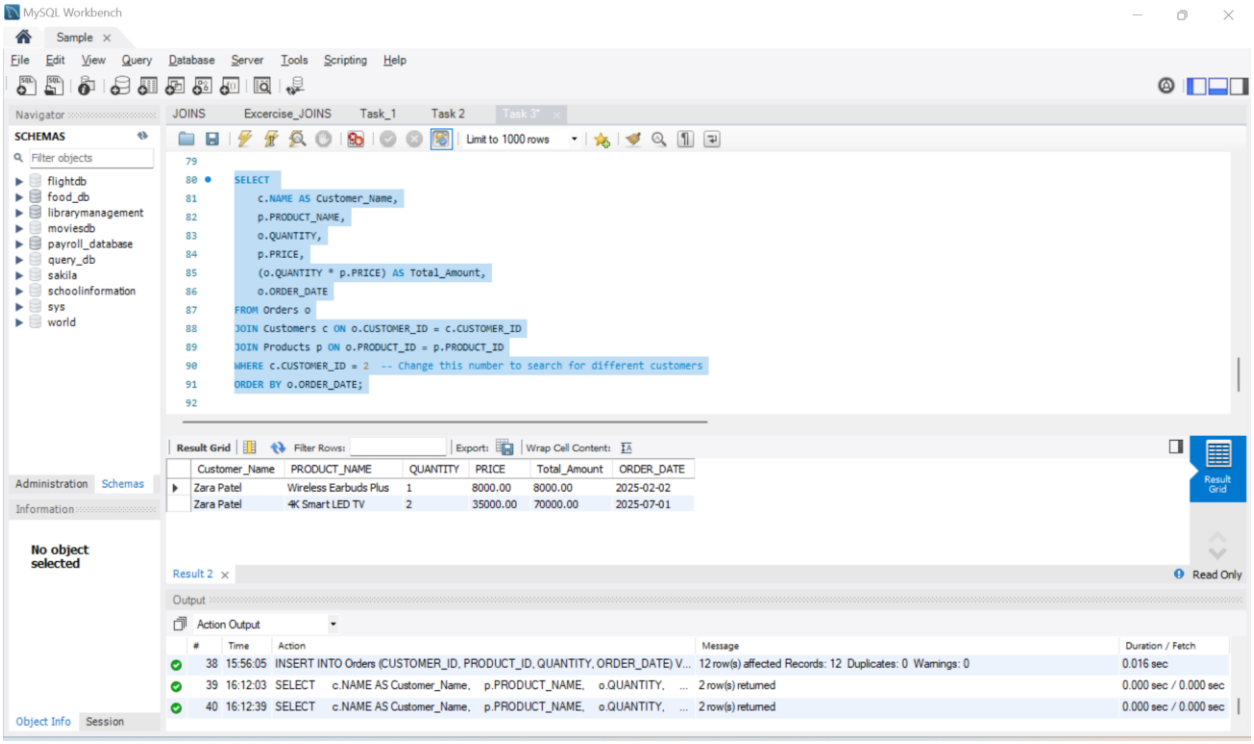


Orders



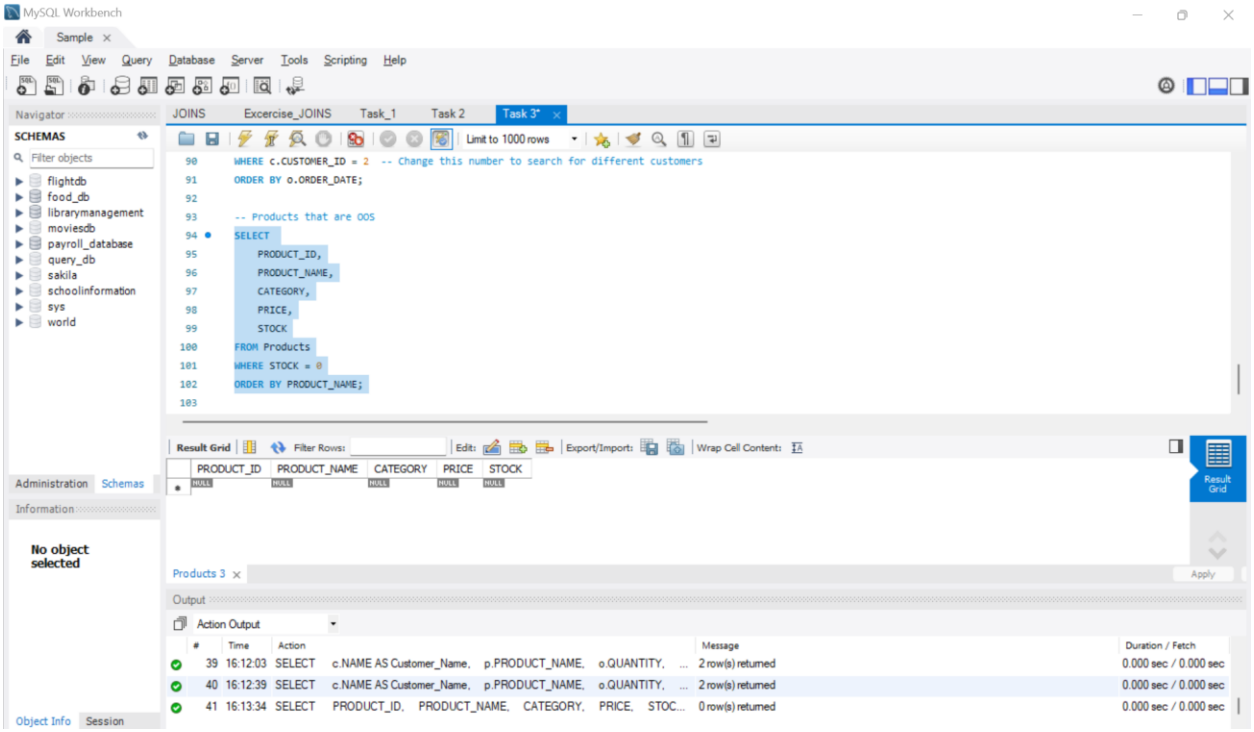
Order Management:

a) Retrieve all orders placed by a specific customer.

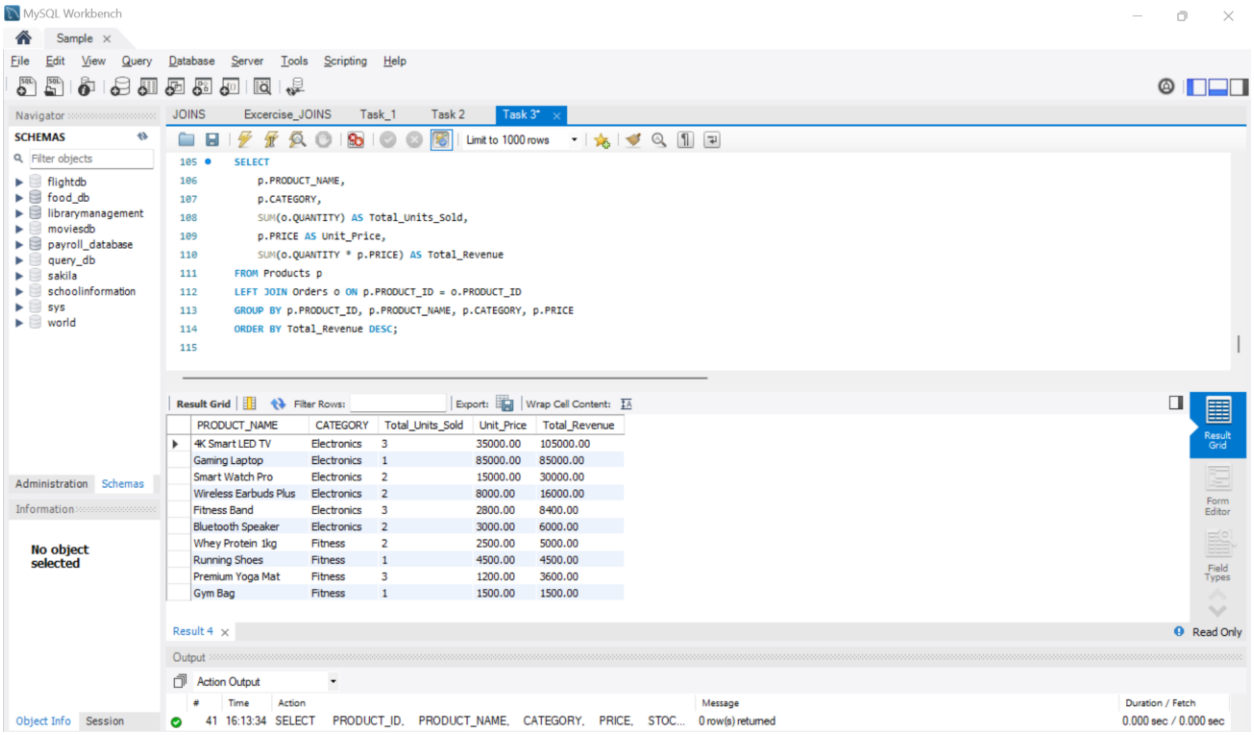


b) Find products that are out of stock.

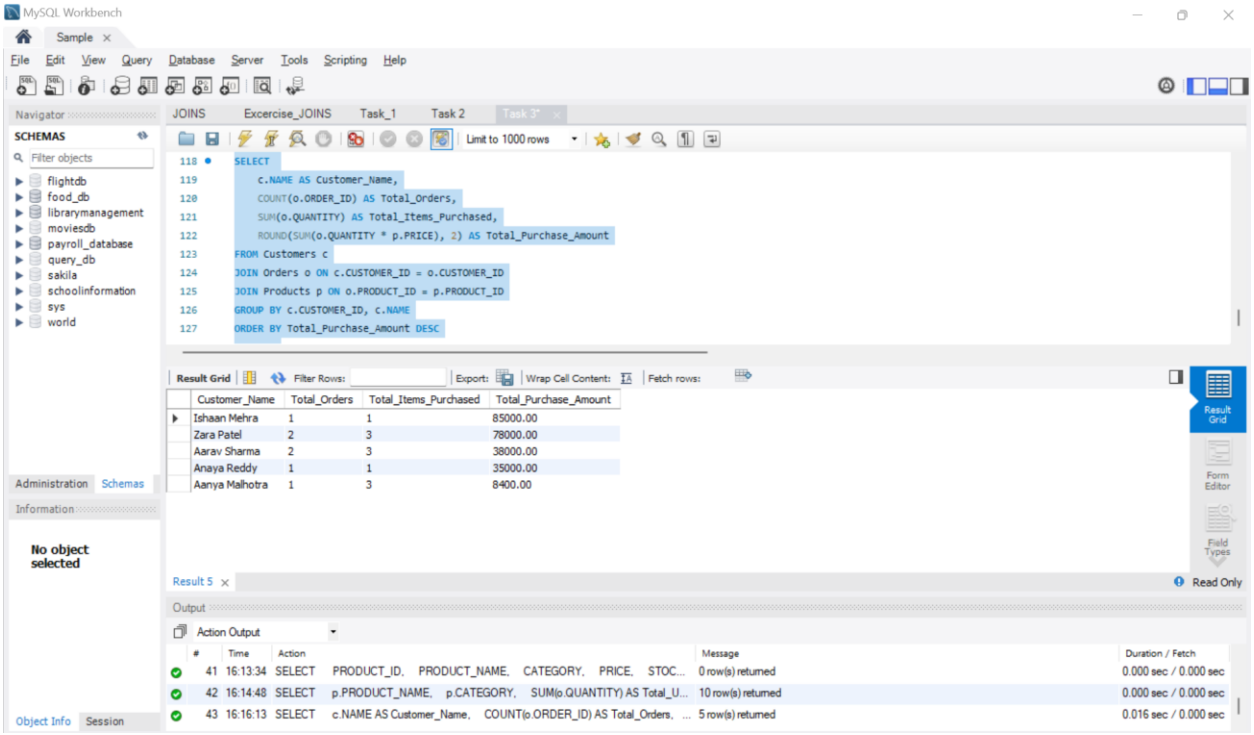
There are 0 products which are currently OOS.



c) Calculate the total revenue generated per product.

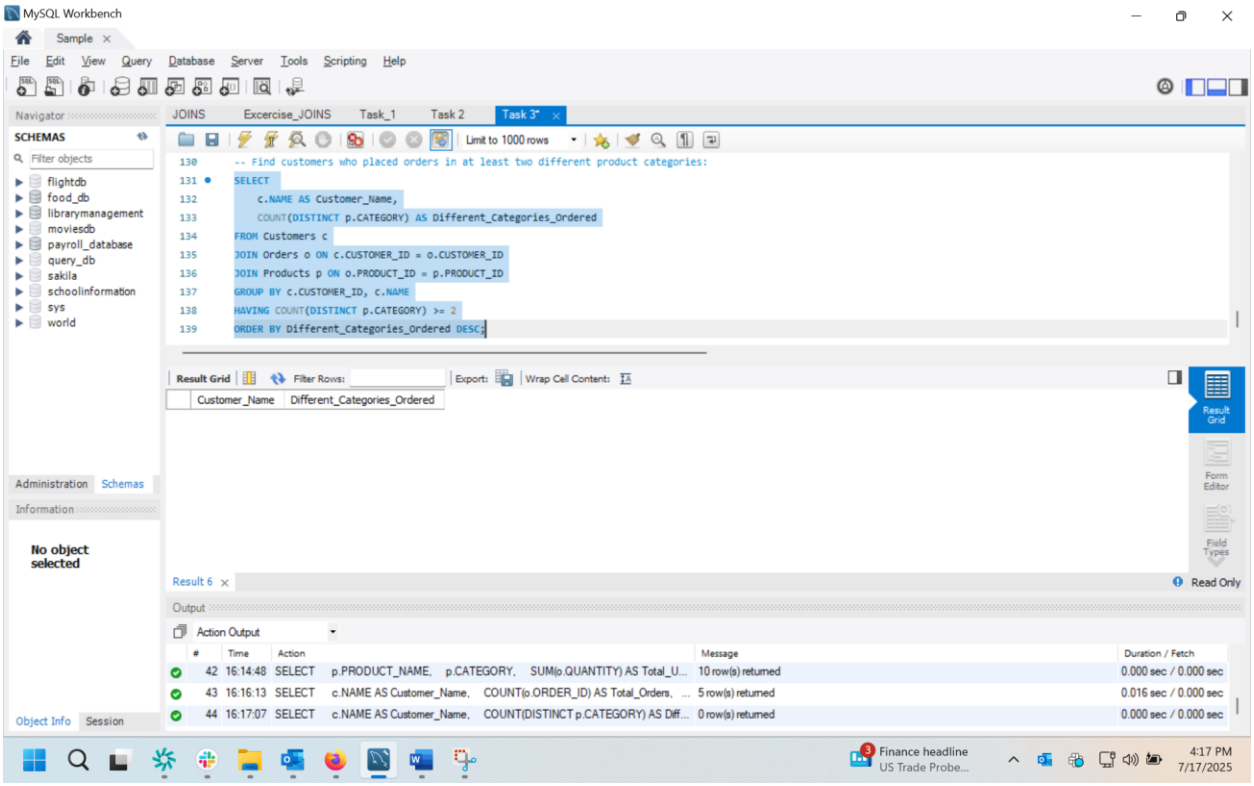


d) Retrieve the top 5 customers by total purchase amount.



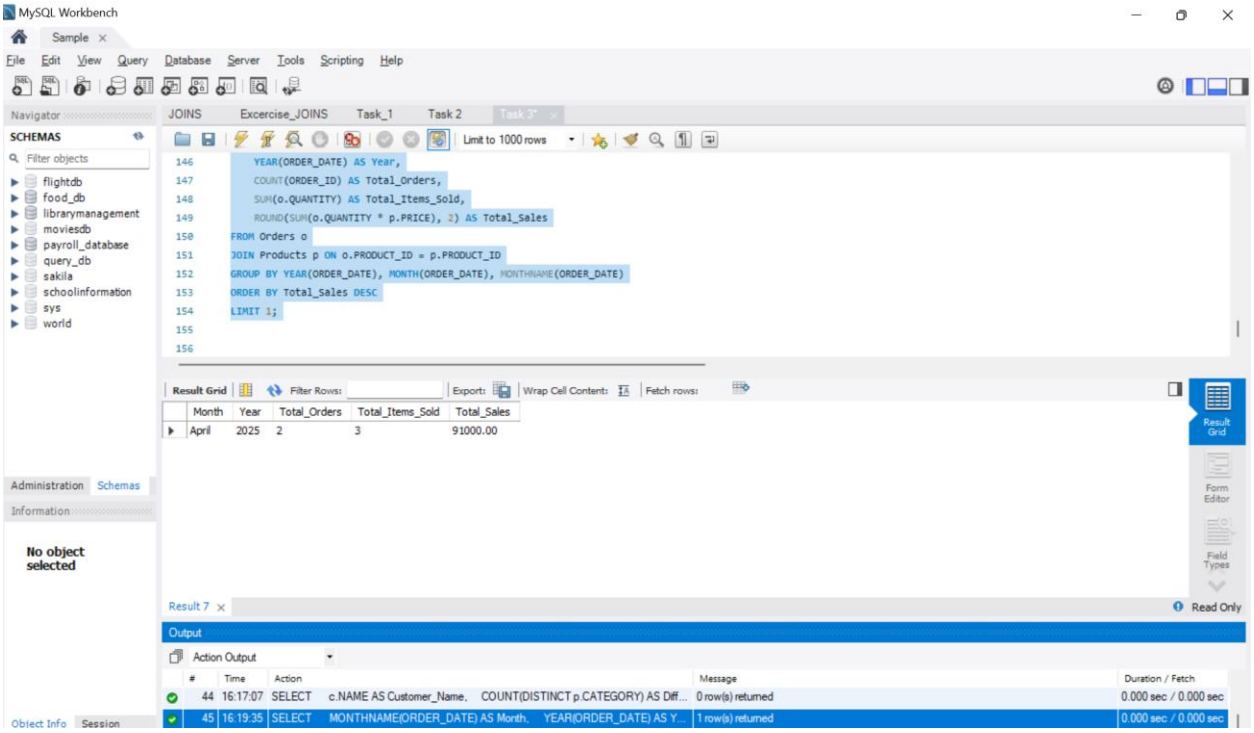
e) Find customers who place orders in at least two different product categories.

There are no customers with different categories ordered.

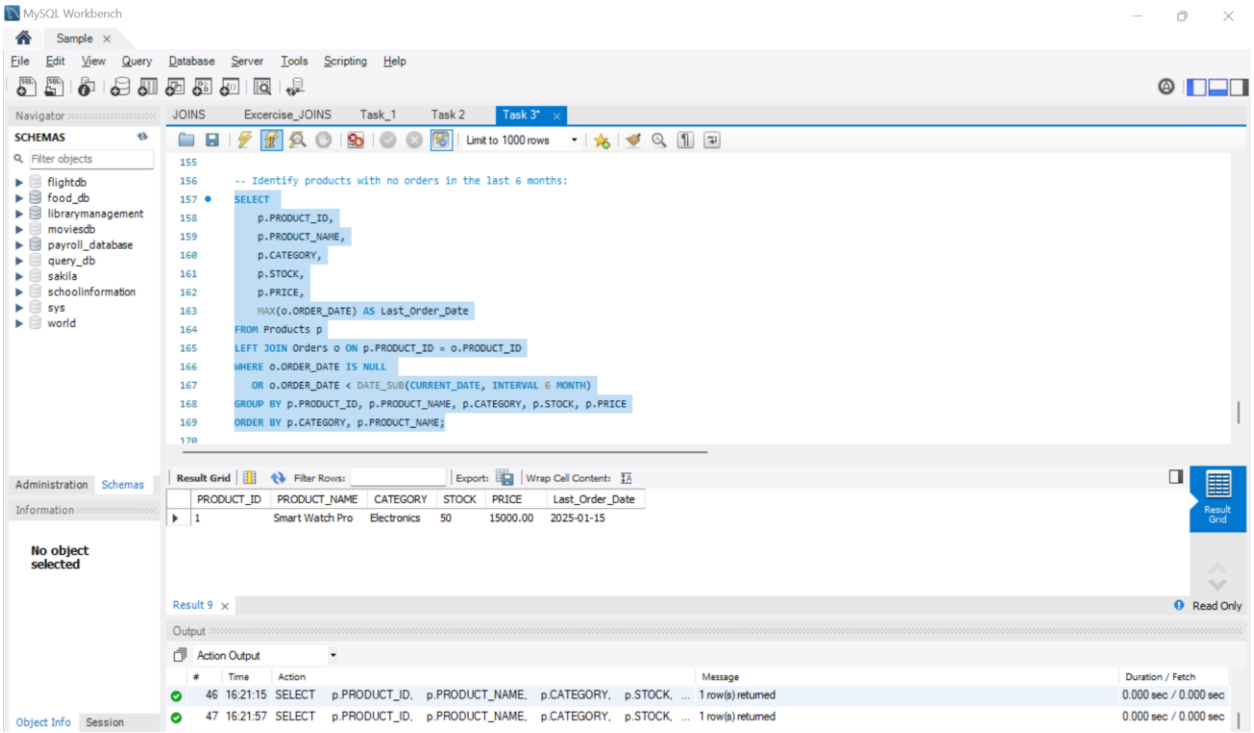


Analytics:

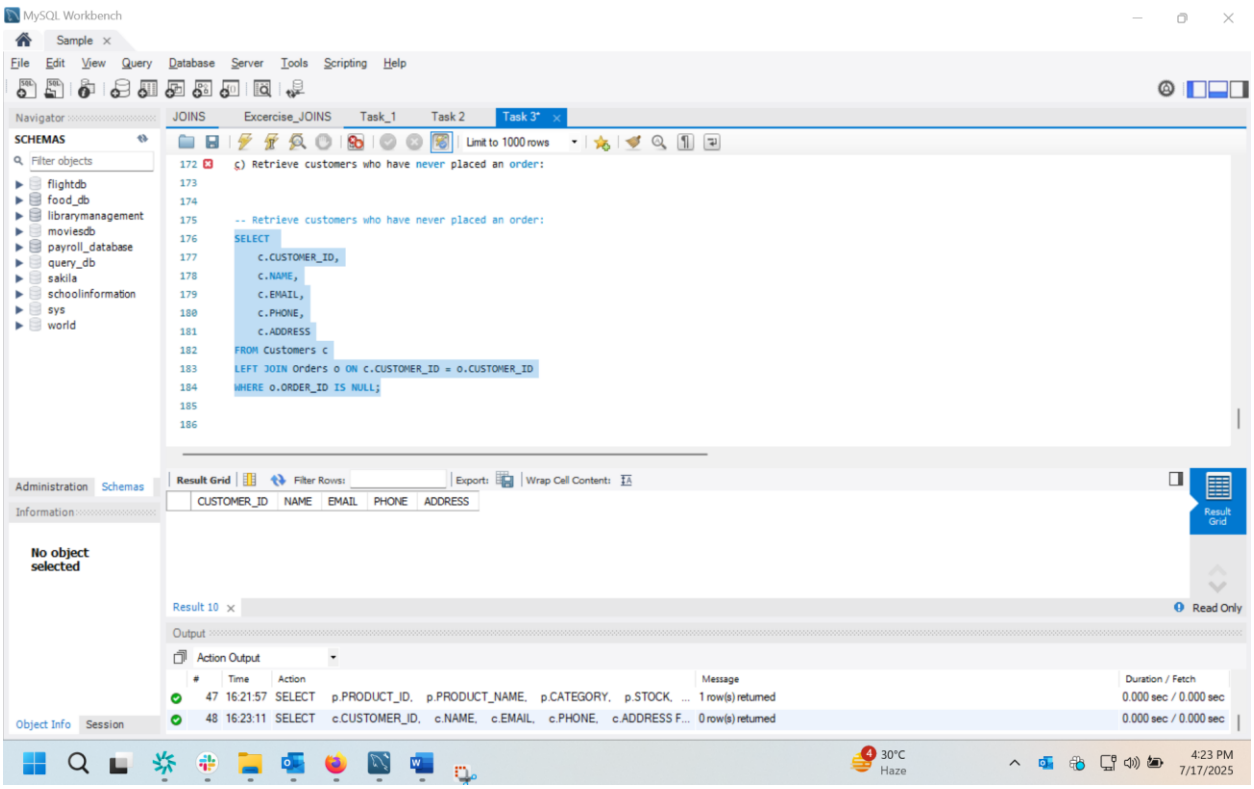
a) Find the month with the highest total sales.



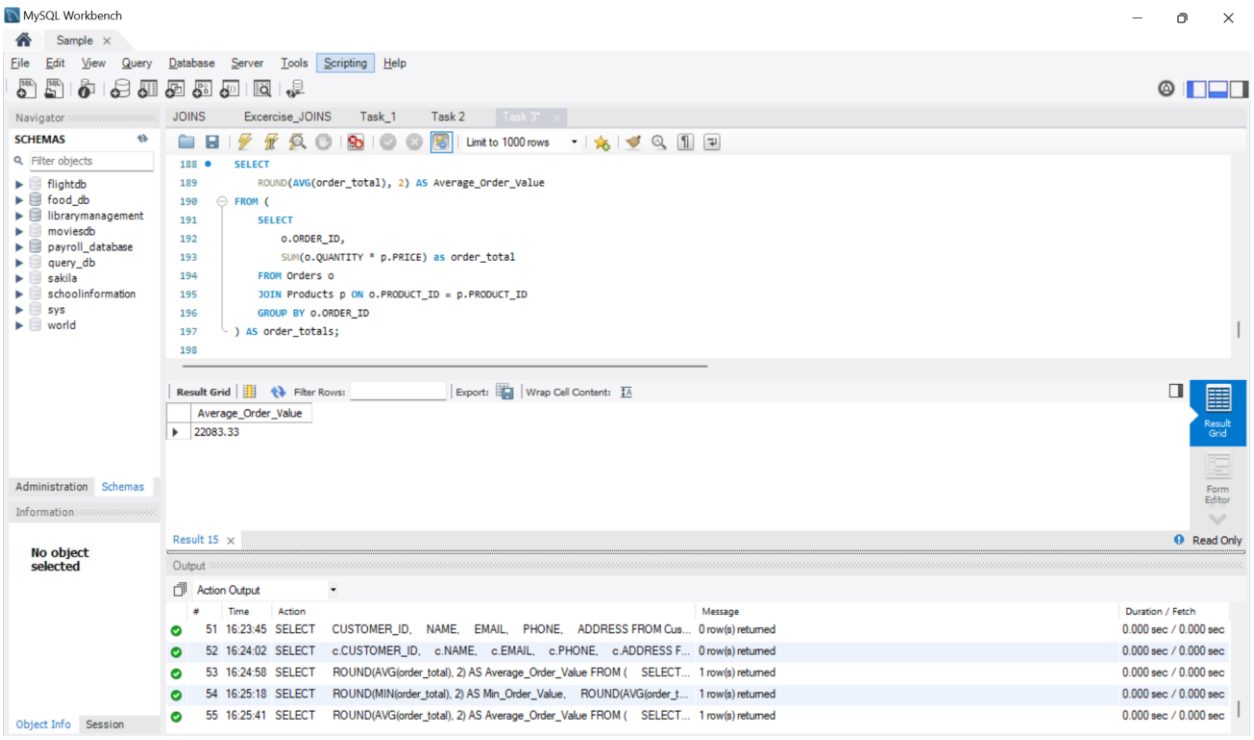
b) Identify products with no orders in the last 6 months.



c) Retrieve customers who have never placed an order.



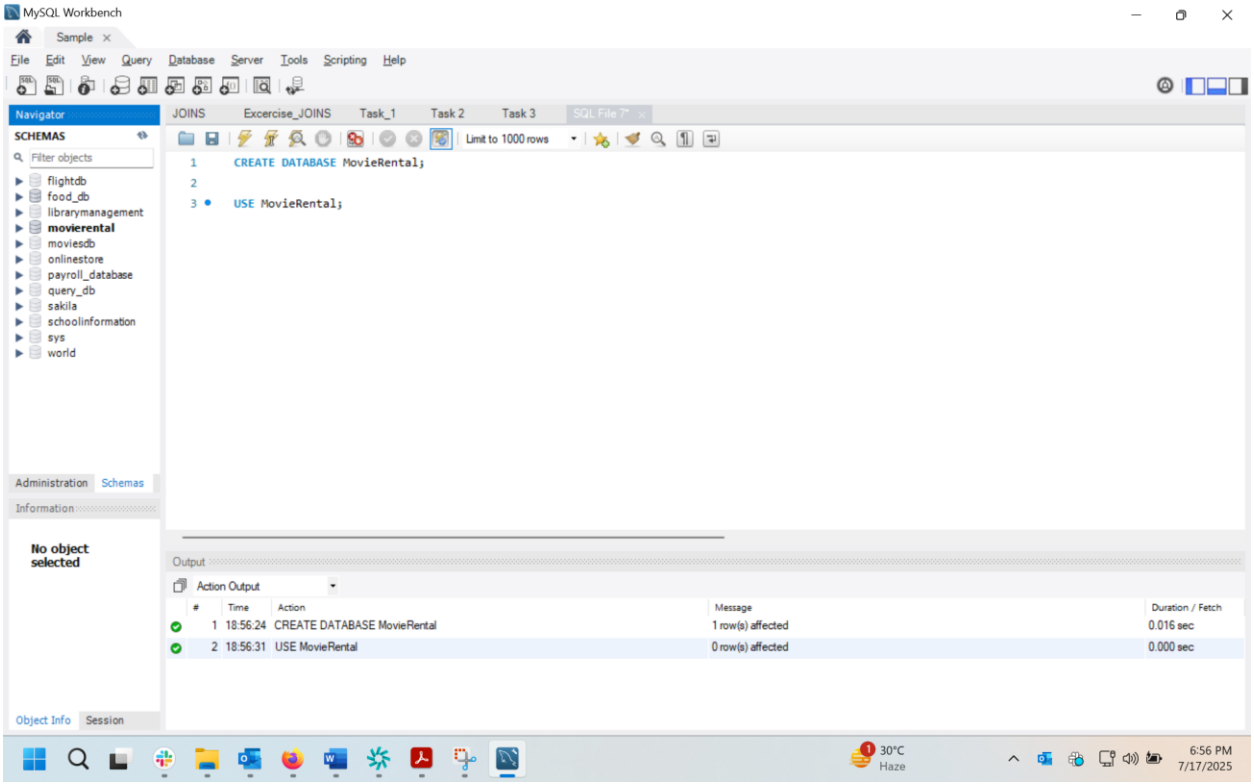
d) Calculate the average order value across all orders.



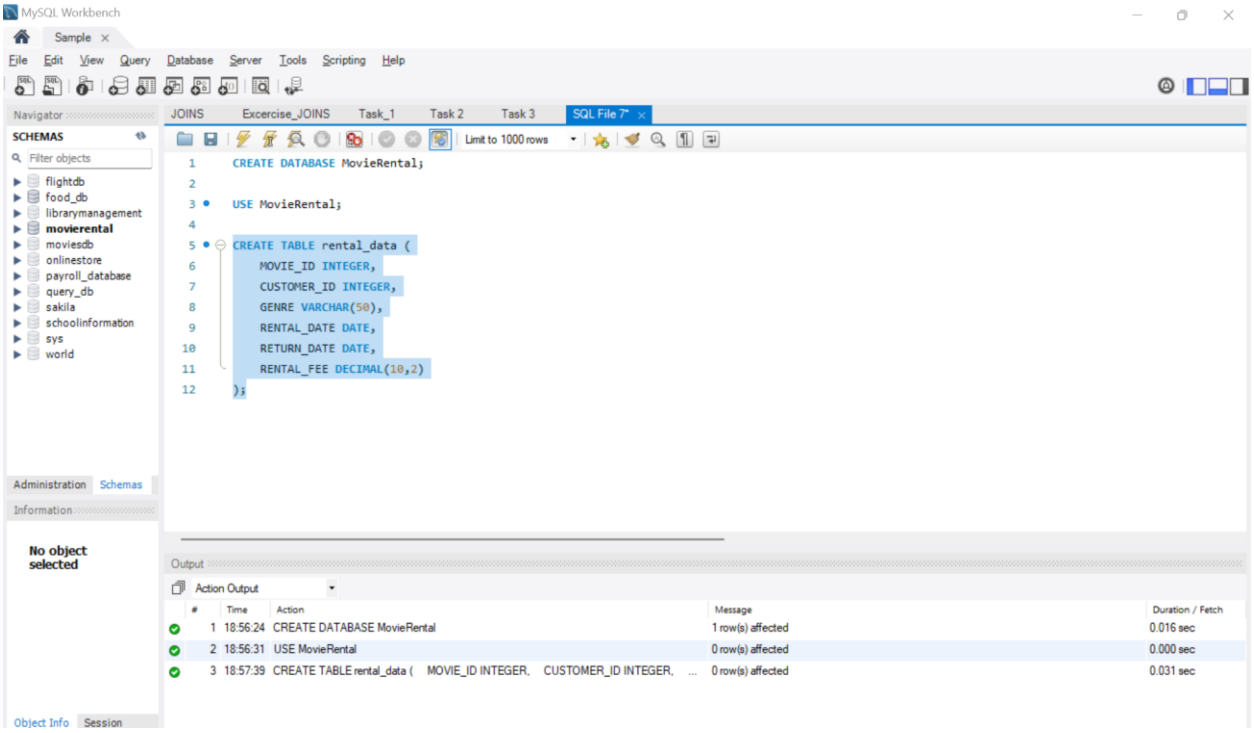
Task 4

Project: Movie Rental Analysis System

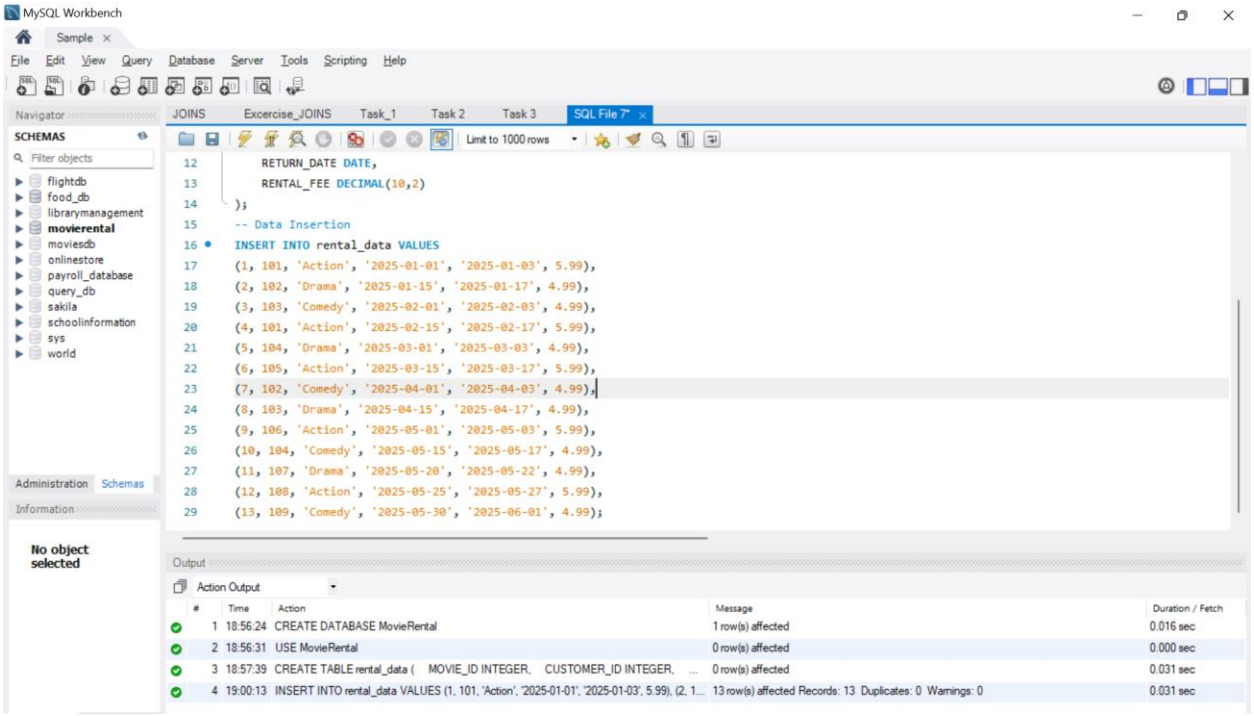
Database Creation:



Create table rental data with columns:

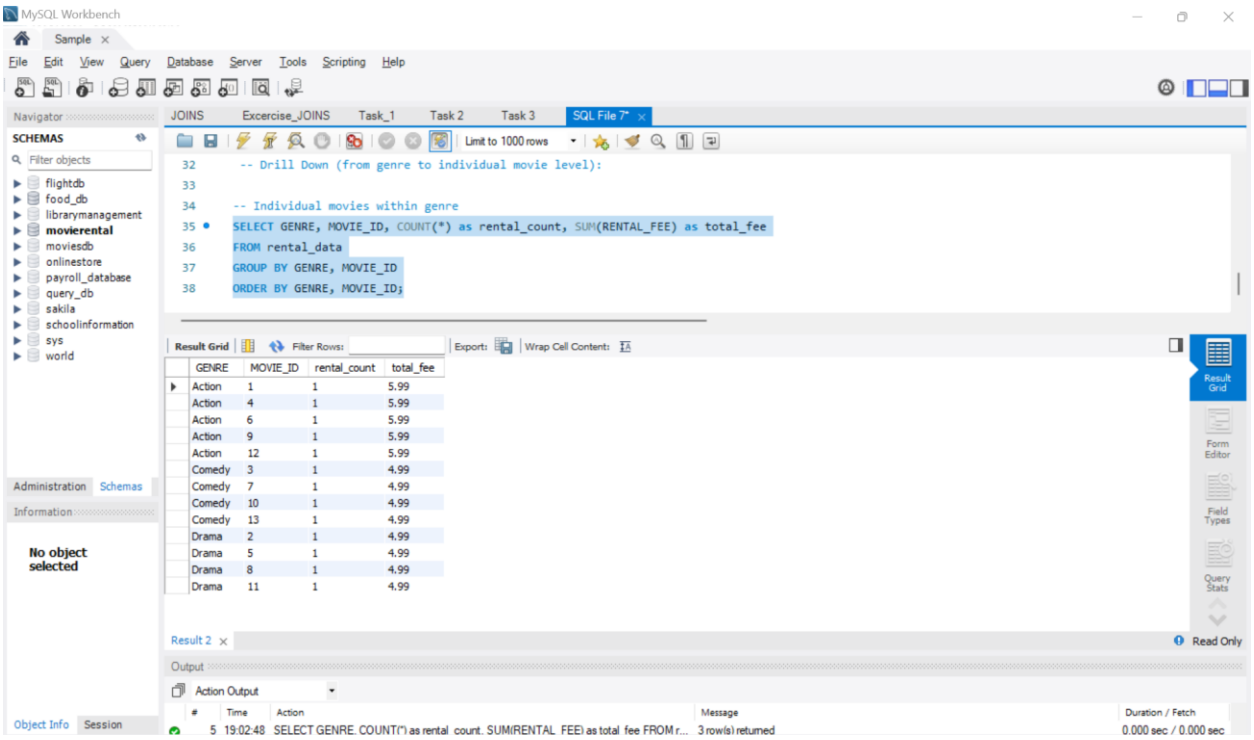


Data Creation: Added 13 sample records:

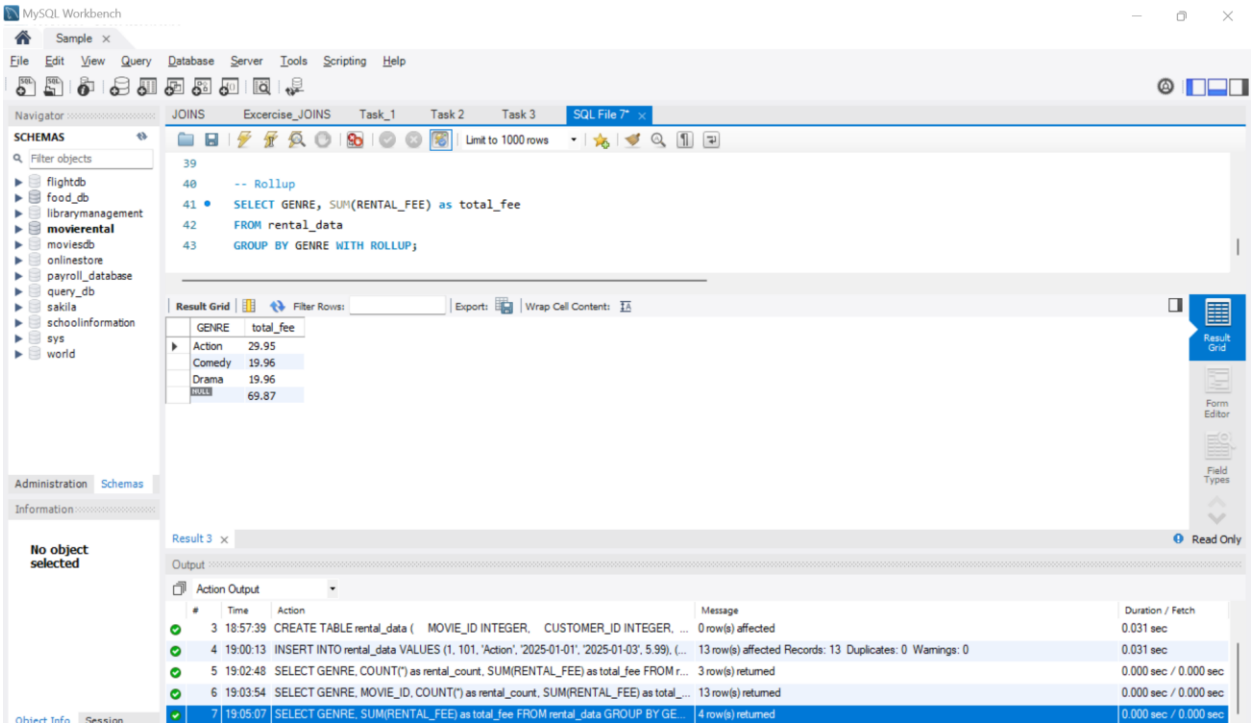


OLAP Operations:

a) Drill Down: Analyze rentals from genre to individual movie level.

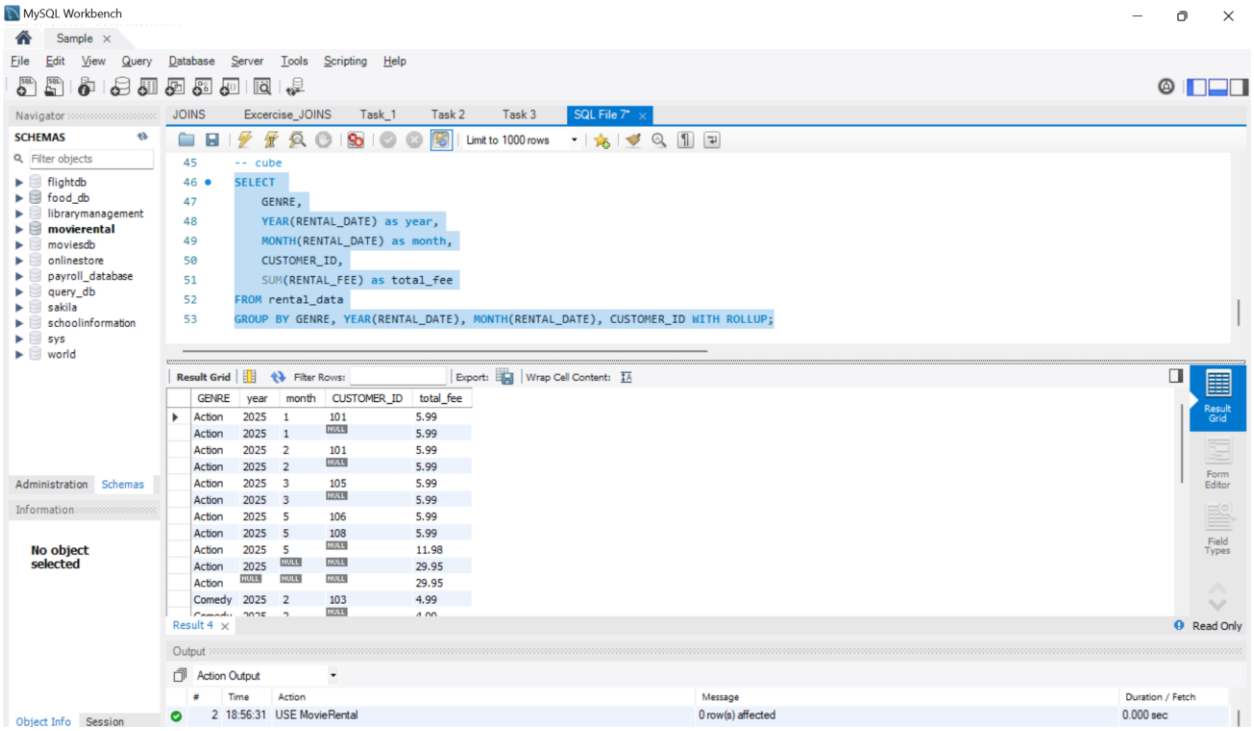


b) Rollup: Summarize total rental fees by genre and then overall.

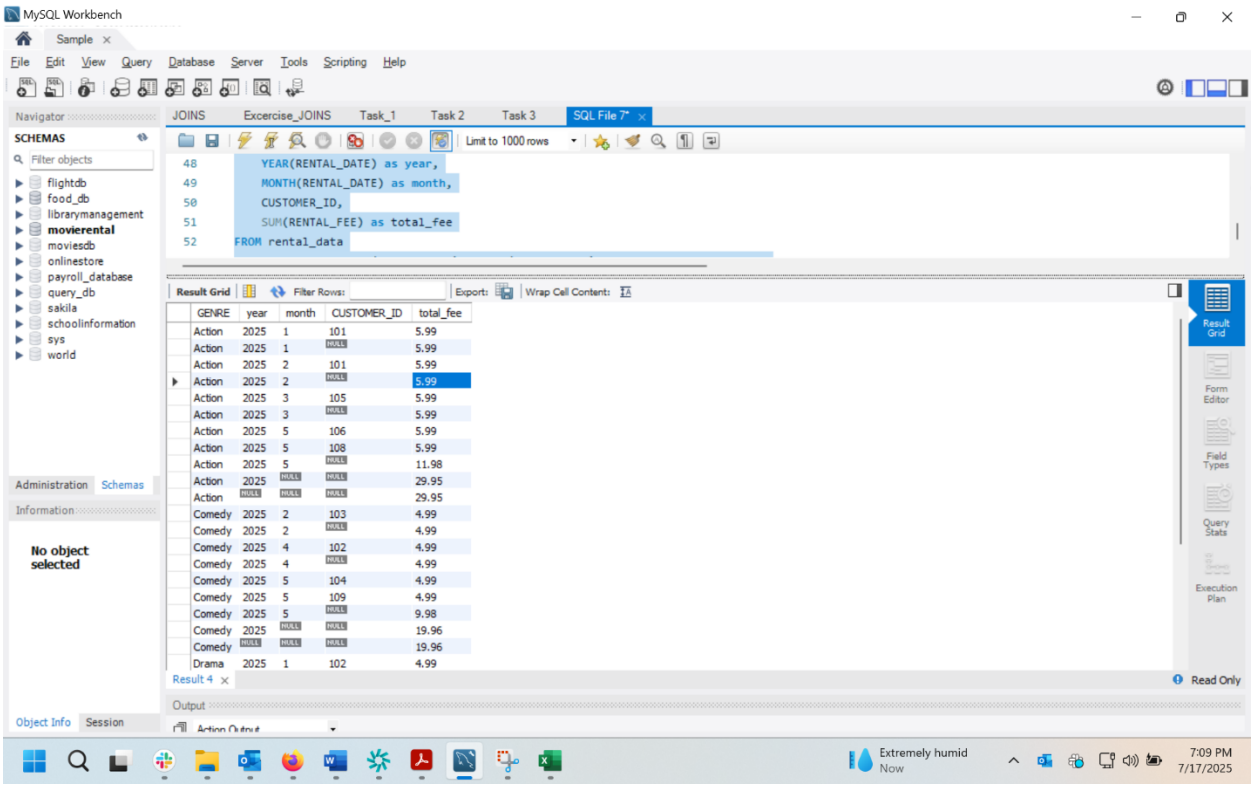


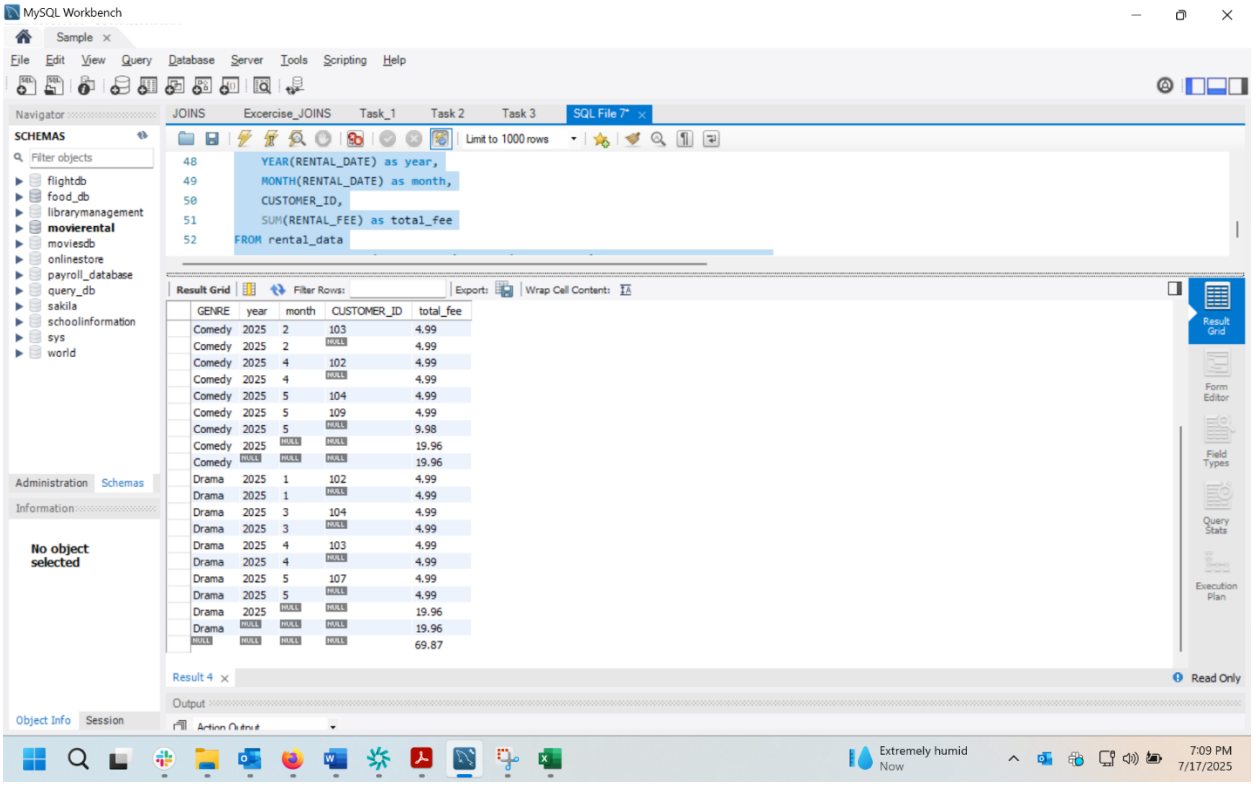
c) Cube: Analyze total rental fees across combinations of genre, rental date, and customer.

Query-

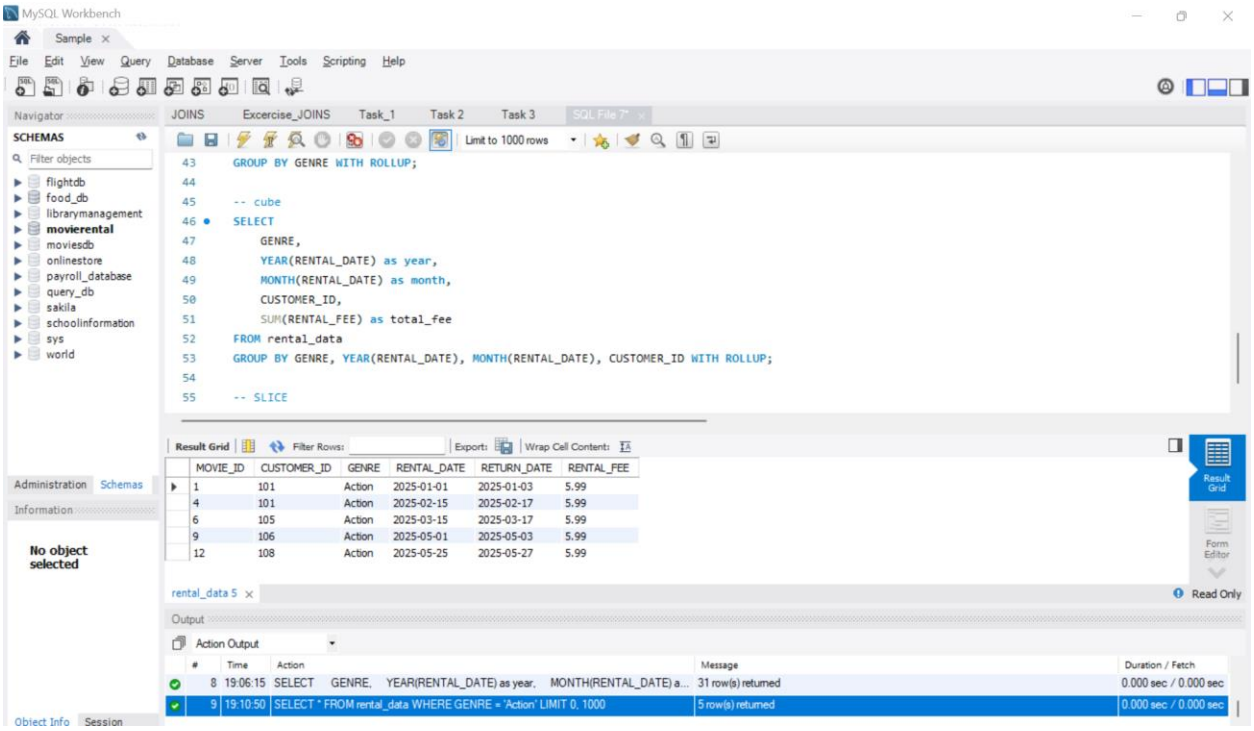


Output-

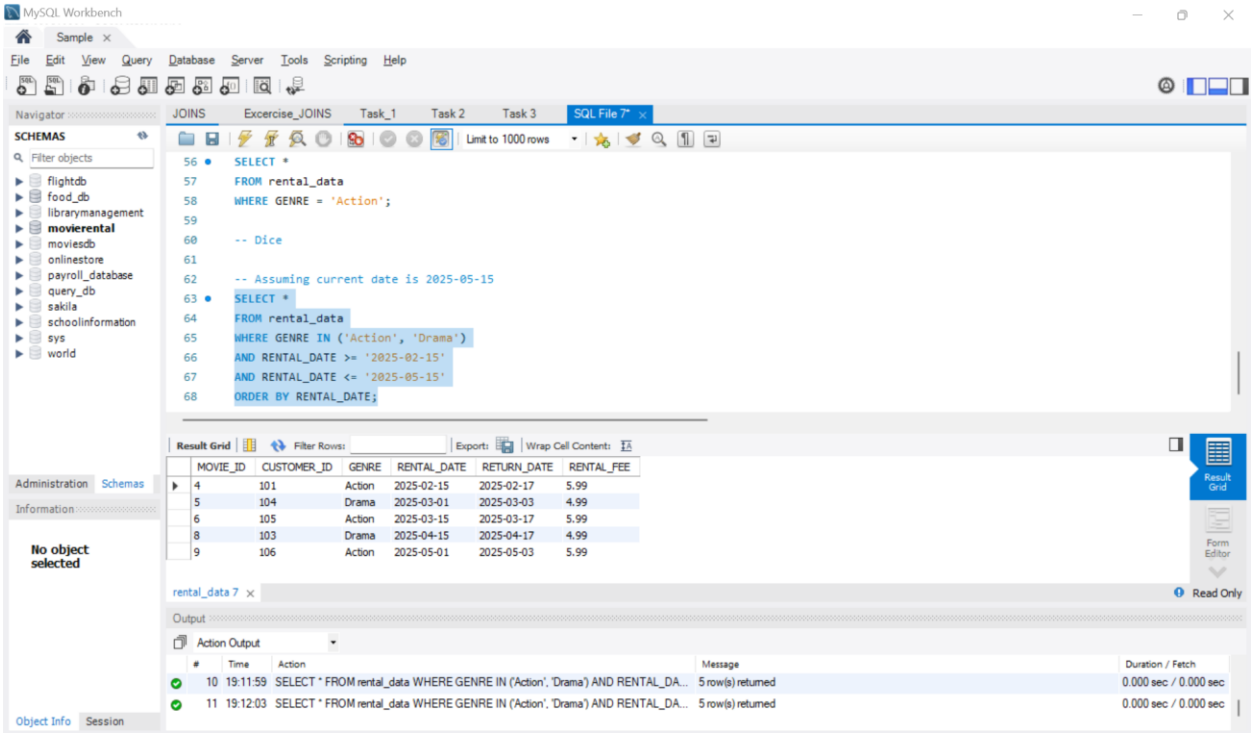

MySQL Workbench
Sample
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
flightdb
food_db
librarymanagement
movierental
moviesdb
onlinestore
payroll_database
query_db
sakila
schoolinformation
sys
world
Administration Schemas
Information
No object selected
Object Info Session
JOINS Excercise_JOINS Task_1 Task 2 Task 3 SQL File 7*
Limit to 1000 rows
48 YEAR(RENTAL_DATE) as year,
49 MONTH(RENTAL_DATE) as month,
50 CUSTOMER_ID,
51 SUM(RENTAL_FEE) as total_fee
52 FROM rental_data
Result Grid Filter Rows: Export: Wrap Cell Content:
GENRE year month CUSTOMER_ID total_fee
Action 2025 1 101 5.99
Action 2025 1 NULL 5.99
Action 2025 2 101 5.99
Action 2025 2 NULL 5.99
Action 2025 3 105 5.99
Action 2025 3 NULL 5.99
Action 2025 5 106 5.99
Action 2025 5 108 5.99
Action 2025 5 NULL 11.98
Action 2025 NULL NULL 29.95
Action NULL NULL NULL 29.95
Comedy 2025 2 103 4.99
Comedy 2025 2 NULL 4.99
Comedy 2025 4 102 4.99
Comedy 2025 4 NULL 4.99
Comedy 2025 5 104 4.99
Comedy 2025 5 109 4.99
Comedy 2025 5 NULL 9.98
Comedy 2025 NULL NULL 19.96
Comedy NULL NULL NULL 19.96
Drama 2025 1 102 4.99
Result 4
Read Only
Output
Extremely humid Now
7:09 PM 7/17/2025


MySQL Workbench
Sample
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
flightdb
food_db
librarymanagement
movierental
moviesdb
onlinestore
payroll_database
query_db
sakila
schoolinformation
sys
world
Administration Schemas
Information
No object selected
Object Info Session
JOINS Excercise_JOINS Task_1 Task 2 Task 3 SQL File 7*
Limit to 1000 rows
48 YEAR(RENTAL_DATE) as year,
49 MONTH(RENTAL_DATE) as month,
50 CUSTOMER_ID,
51 SUM(RENTAL_FEE) as total_fee
52 FROM rental_data
Result Grid Filter Rows: Export: Wrap Cell Content:
GENRE year month CUSTOMER_ID total_fee
Comedy 2025 2 103 4.99
Comedy 2025 2 NULL 4.99
Comedy 2025 4 102 4.99
Comedy 2025 4 NULL 4.99
Comedy 2025 5 104 4.99
Comedy 2025 5 109 4.99
Comedy 2025 5 NULL 9.98
Comedy 2025 NULL NULL 19.96
Comedy NULL NULL NULL 19.96
Drama 2025 1 102 4.99
Drama 2025 1 NULL 4.99
Drama 2025 3 104 4.99
Drama 2025 3 NULL 4.99
Drama 2025 4 103 4.99
Drama 2025 4 NULL 4.99
Drama 2025 5 107 4.99
Drama 2025 5 NULL 4.99
Drama 2025 NULL NULL 19.96
Drama NULL NULL NULL 19.96
NULL NULL NULL NULL 69.87
Result 4
Read Only
Output
Extremely humid Now
7:09 PM 7/17/2025

d) Slice: Extract rentals only from the ‘Action’ genre.



e) Dice: Extract rentals where GENRE = 'Action' or 'Drama' and RENTAL_DATE is in the last 3 months.


MySQL Workbench
Sample
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
flightdb
food_db
librarymanagement
movierental
moviesdb
onlinestore
payroll_database
query_db
sakila
schoolinformation
sys
world
Administration Schemas
Information
No object selected
Object Info Session
JOINS Excercise_JOINS Task_1 Task 2 Task 3 SQL File 7*
Limit to 1000 rows
56 SELECT *
57 FROM rental_data
58 WHERE GENRE = 'Action';
59
60 -- Dice
61
62 -- Assuming current date is 2025-05-15
63 SELECT *
64 FROM rental_data
65 WHERE GENRE IN ('Action', 'Drama')
66 AND RENTAL_DATE >= '2025-02-15'
67 AND RENTAL_DATE <= '2025-05-15'
68 ORDER BY RENTAL_DATE;
Result Grid Filter Rows: Export: Wrap Cell Content:
MOVIE_ID CUSTOMER_ID GENRE RENTAL_DATE RETURN_DATE RENTAL_FEE
4 101 Action 2025-02-15 2025-02-17 5.99
5 104 Drama 2025-03-01 2025-03-03 4.99
6 105 Action 2025-03-15 2025-03-17 5.99
8 103 Drama 2025-04-15 2025-04-17 4.99
9 106 Action 2025-05-01 2025-05-03 5.99
Result Grid
Form Editor
rental_data 7
Read Only
Output
Action Output
Time Action Message Duration / Fetch
10 19:11:59 SELECT * FROM rental_data WHERE GENRE IN ('Action', 'Drama') AND RENTAL_DA... 5 row(s) returned 0.000 sec / 0.000 sec
11 19:12:03 SELECT * FROM rental_data WHERE GENRE IN ('Action', 'Drama') AND RENTAL_DA... 5 row(s) returned 0.000 sec / 0.000 sec