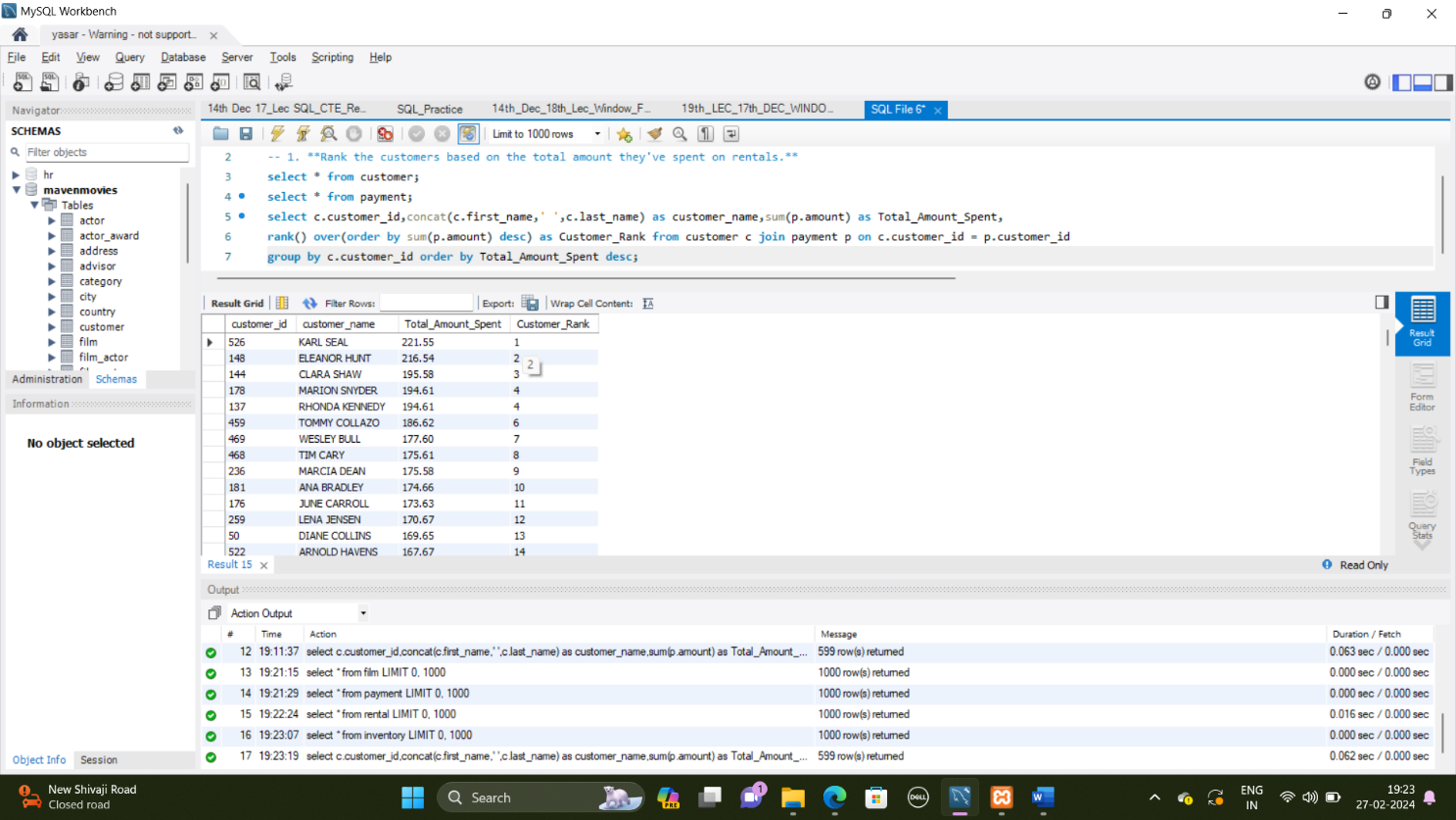
**Assignment: Windows Functions**

1. Rank the customers based on the total amount they've spent on rentals.

select c.customer\_id,concat(c.first\_name,' ',c.last\_name) as customer\_name,sum(p.amount) as Total\_Amount\_Spent,

rank() over(order by sum(p.amount) desc) as Customer\_Rank from customer c join payment p on c.customer\_id = p.customer\_id

group by c.customer\_id order by Total\_Amount\_Spent desc;

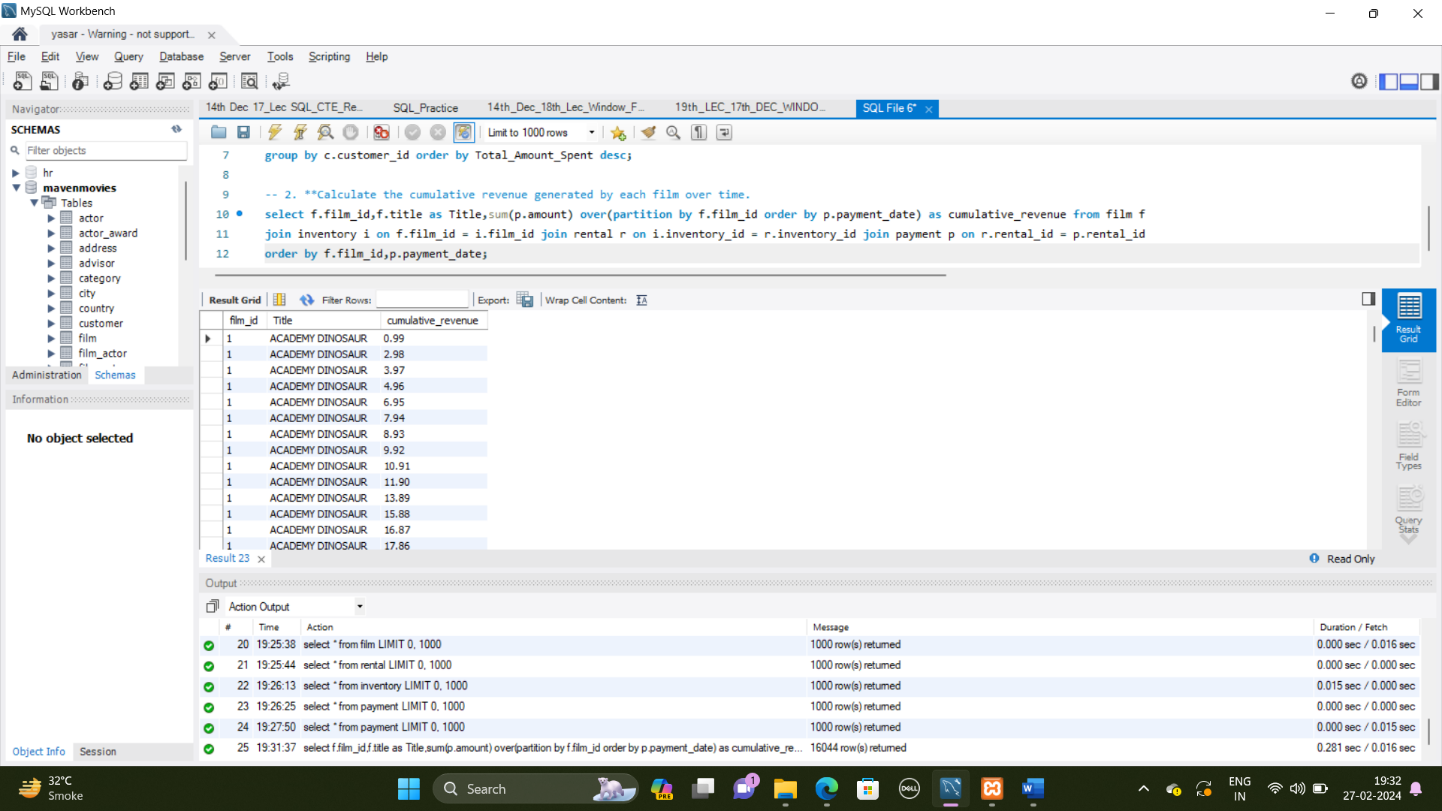


1. Calculate the cumulative revenue generated by each film over time.

select f.film\_id,f.title as Title,sum(p.amount) over(partition by f.film\_id order by p.payment\_date) as cumulative\_revenue from film f

join inventory i on f.film\_id = i.film\_id join rental r on i.inventory\_id = r.inventory\_id join payment p on r.rental\_id = p.rental\_id

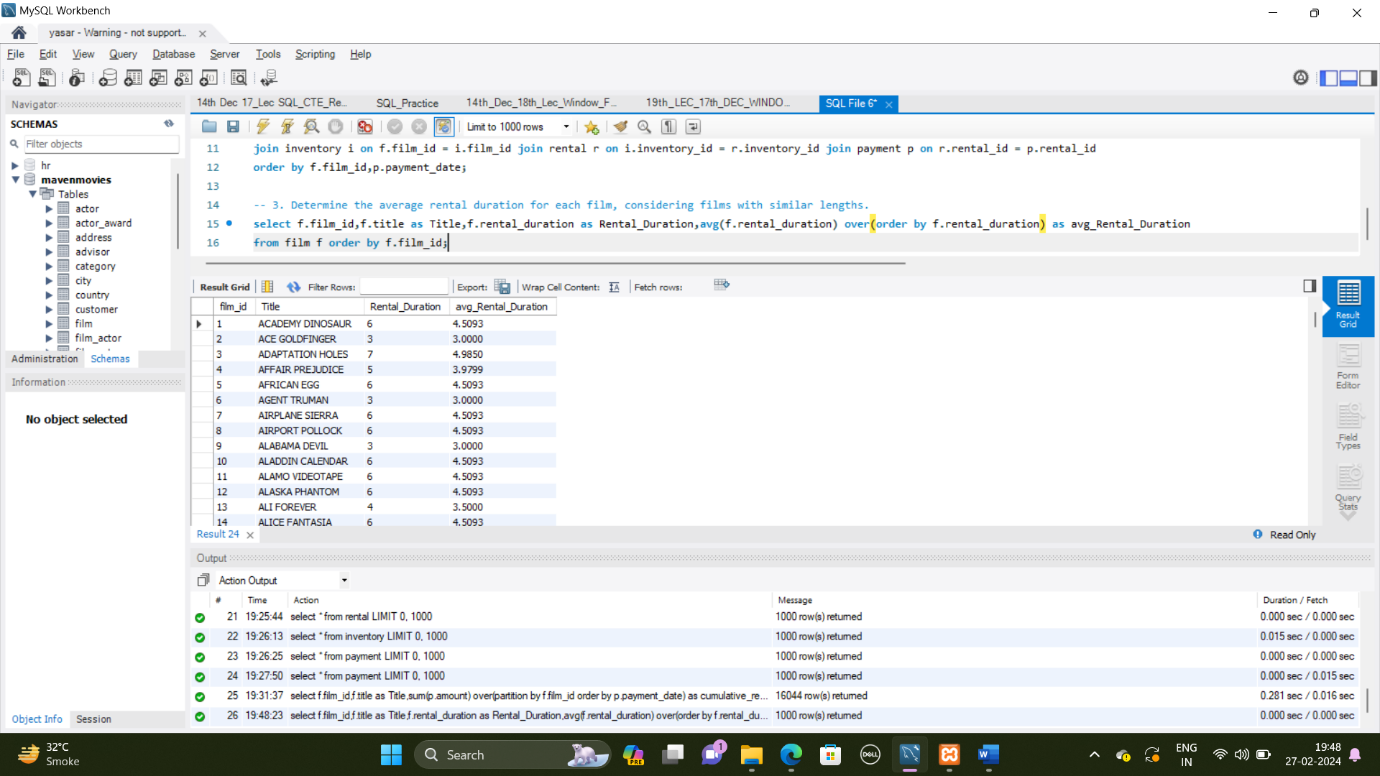
order by f.film\_id,p.payment\_date;



1. Determine the average rental duration for each film, considering films with similar lengths.

select f.film\_id,f.title as Title,f.rental\_duration as Rental\_Duration,avg(f.rental\_duration) over(order by f.rental\_duration) as avg\_Rental\_Duration

from film f order by f.film\_id;



1. Identify the top 3 films in each category based on their rental counts

with ranked\_films as (

select f.film\_id, f.title, c.name as Category, count(rental\_id) as rental\_count,

row\_number() over (partition by c.name order by count(rental\_id) desc) as Category\_Rank

from film f join film\_category fc on f.film\_id = fc.film\_id

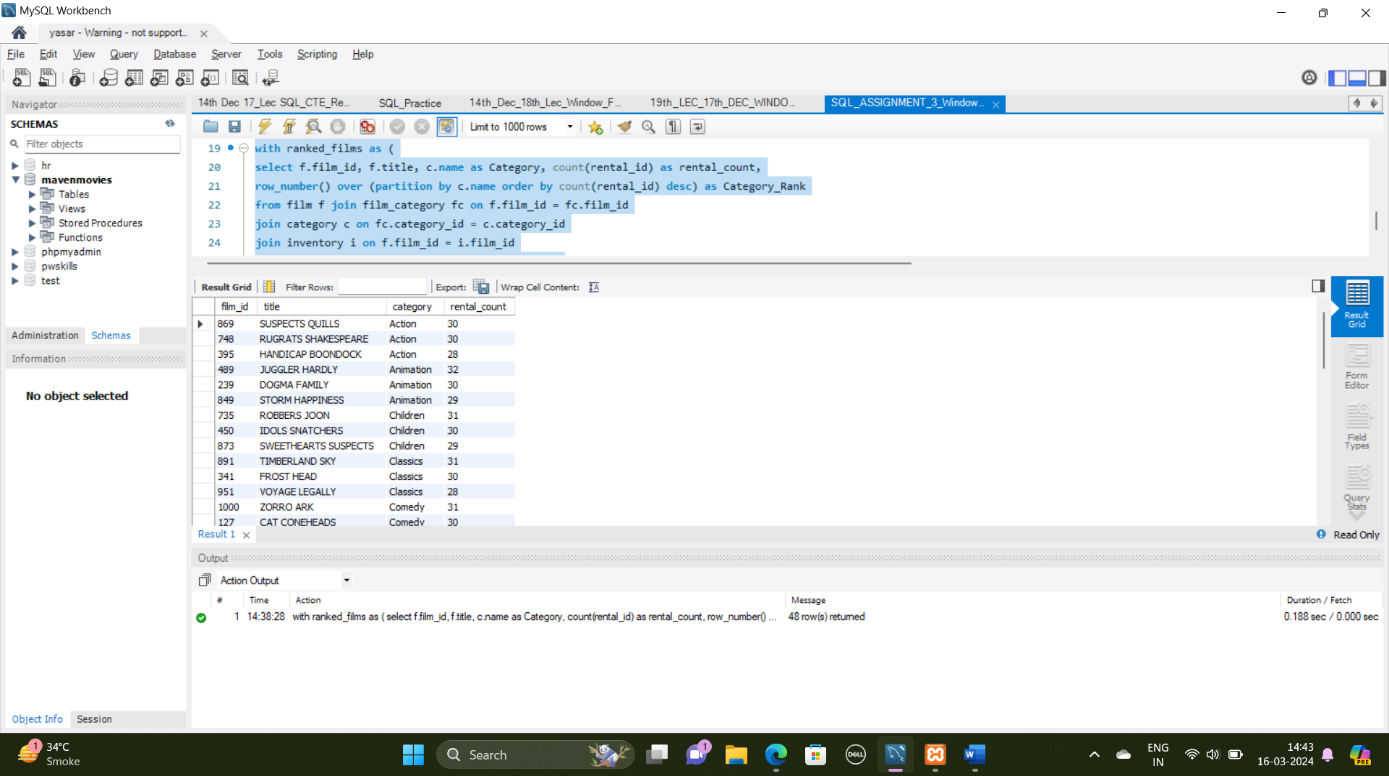
join category c on fc.category\_id = c.category\_id

join inventory i on f.film\_id = i.film\_id

join rental r on i.inventory\_id = r.inventory\_id

group by f.film\_id,f.title,c.name)

select film\_id,title,category,rental\_count from ranked\_films where Category\_Rank <= 3 order by category,Category\_Rank;



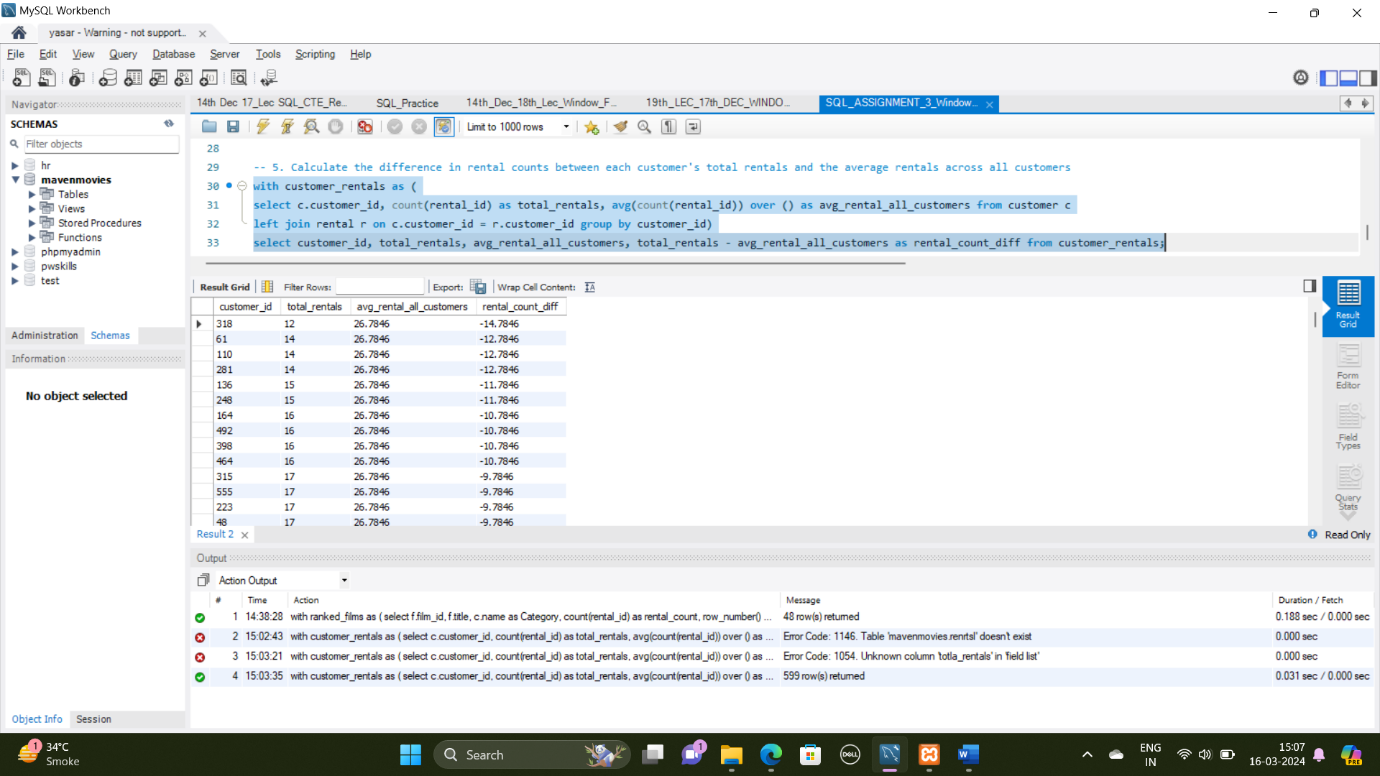
1. Calculate the difference in rental counts between each customer's total rentals and the average rentals across all customers.

with customer\_rentals as (

select c.customer\_id, count(rental\_id) as total\_rentals, avg(count(rental\_id)) over () as avg\_rental\_all\_customers from customer c

left join rental r on c.customer\_id = r.customer\_id group by customer\_id)

select customer\_id, total\_rentals, avg\_rental\_all\_customers, total\_rentals - avg\_rental\_all\_customers as rental\_count\_diff from customer\_rentals;



1. Find the monthly revenue trend for the entire rental store over time.

with monthly\_revenue as (

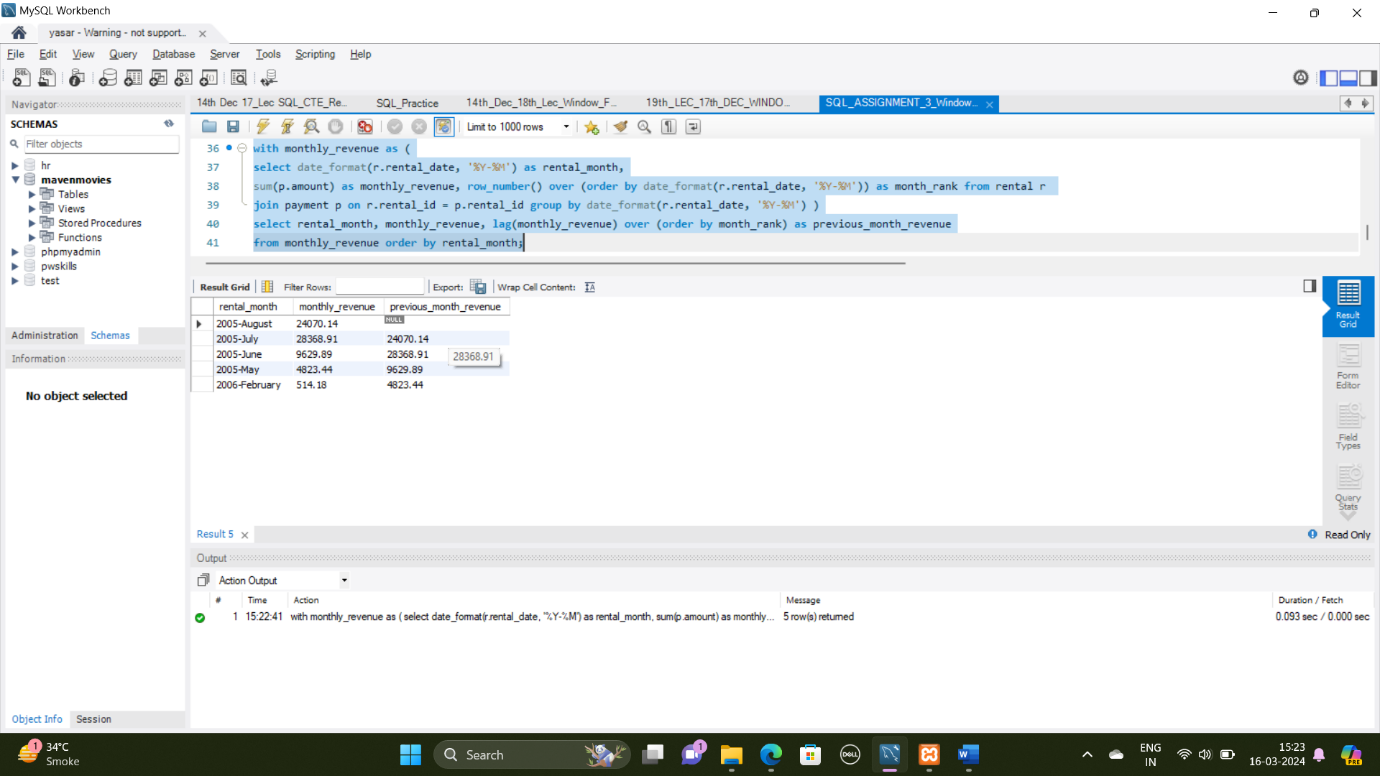
select date\_format(r.rental\_date, '%Y-%M') as rental\_month,

sum(p.amount) as monthly\_revenue, row\_number() over (order by date\_format(r.rental\_date, '%Y-%M')) as month\_rank from rental r

join payment p on r.rental\_id = p.rental\_id group by date\_format(r.rental\_date, '%Y-%M') )

select rental\_month, monthly\_revenue, lag(monthly\_revenue) over (order by month\_rank) as previous\_month\_revenue

from monthly\_revenue order by rental\_month;



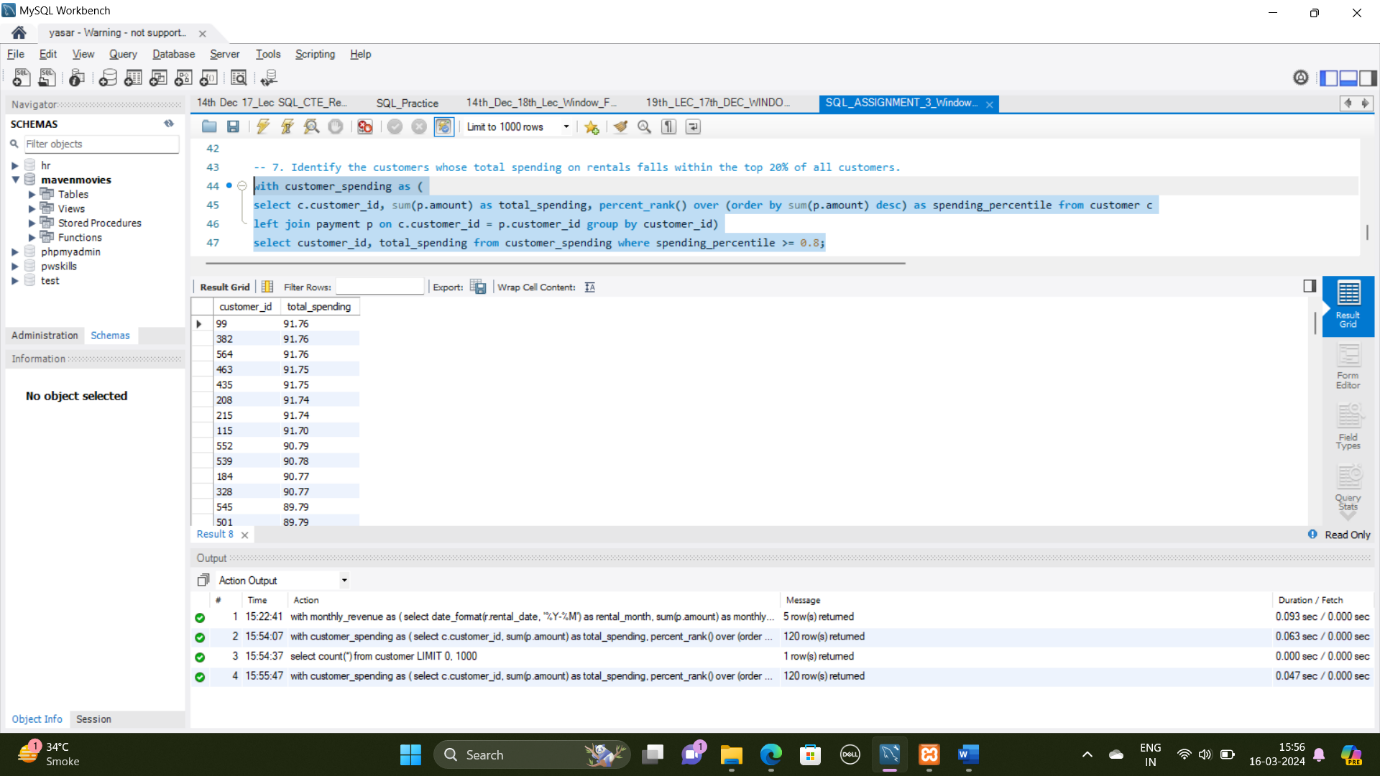
1. Identify the customers whose total spending on rentals falls within the top 20% of all customers.

with customer\_spending as (

select c.customer\_id, sum(p.amount) as total\_spending, percent\_rank() over (order by sum(p.amount) desc) as spending\_percentile from customer c

left join payment p on c.customer\_id = p.customer\_id group by customer\_id)

select customer\_id, total\_spending from customer\_spending where spending\_percentile >= 0.8;



1. Calculate the running total of rentals per category, ordered by rental count.

WITH category\_rentals AS (

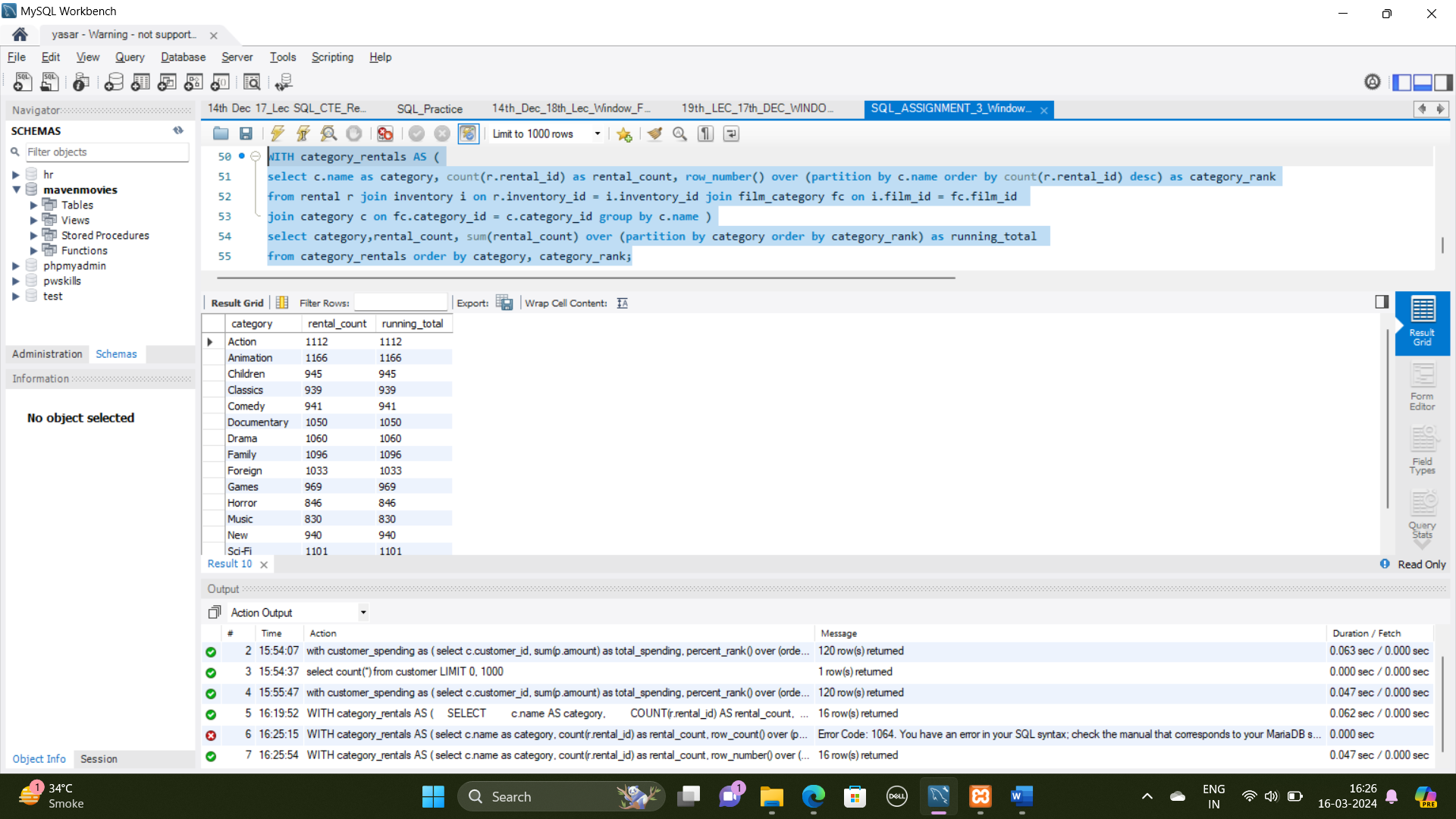
select c.name as category, count(r.rental\_id) as rental\_count, row\_number() over (partition by c.name order by count(r.rental\_id) desc) as category\_rank

from rental r join inventory i on r.inventory\_id = i.inventory\_id join film\_category fc on i.film\_id = fc.film\_id

join category c on fc.category\_id = c.category\_id group by c.name )

select category,rental\_count, sum(rental\_count) over (partition by category order by category\_rank) as running\_total

from category\_rentals order by category, category\_rank;



1. Find the films that have been rented less than the average rental count for their respective categories.

with category\_avg\_rental as (

select fc.film\_id, c.name AS category, count(r.rental\_id) as rental\_count, avg(count(r.rental\_id)) over (partition by c.name) as avg\_rental\_count

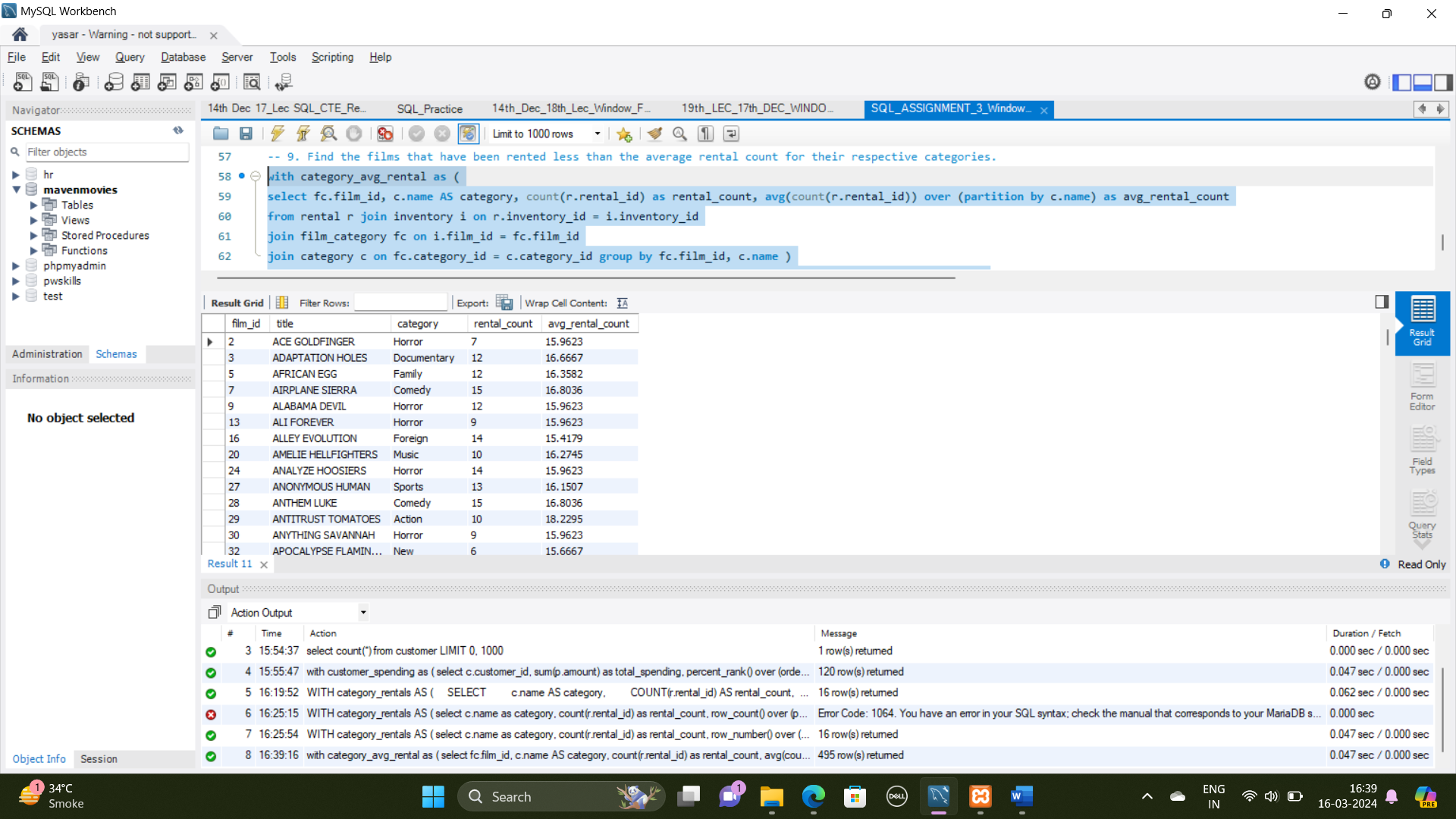
from rental r join inventory i on r.inventory\_id = i.inventory\_id

join film\_category fc on i.film\_id = fc.film\_id

join category c on fc.category\_id = c.category\_id group by fc.film\_id, c.name )

select f.film\_id, f.title, car.category, car.rental\_count, car.avg\_rental\_count from category\_avg\_rental car

join film f on car.film\_id = f.film\_id where car.rental\_count < car.avg\_rental\_count;



1. Identify the top 5 months with the highest revenue and display the revenue generated in each month.

WITH monthly\_revenue AS (

SELECT

DATE\_FORMAT(r.rental\_date, '%Y-%m') AS rental\_month,

SUM(p.amount) AS monthly\_revenue,

ROW\_NUMBER() OVER (ORDER BY SUM(p.amount) DESC) AS month\_rank

FROM

rental r

JOIN

payment p ON r.rental\_id = p.rental\_id

GROUP BY

DATE\_FORMAT(r.rental\_date, '%Y-%m')

)

SELECT

rental\_month,

monthly\_revenue

FROM

monthly\_revenue

WHERE

month\_rank <= 5

ORDER BY

month\_rank;

