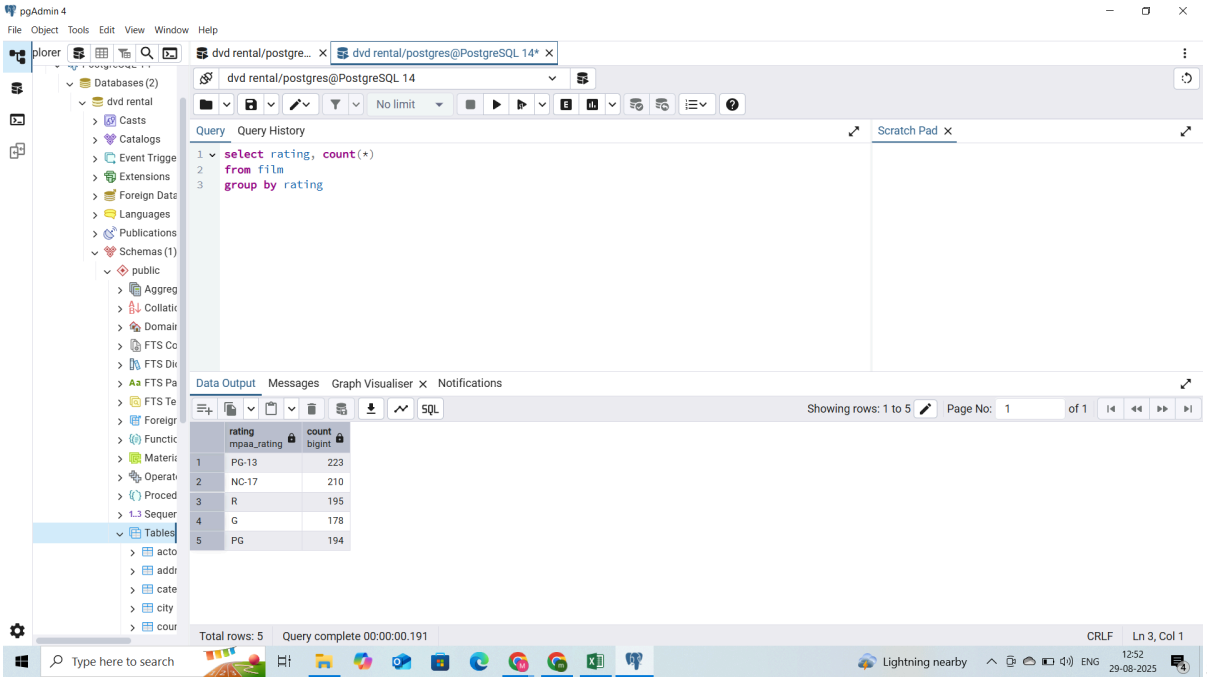
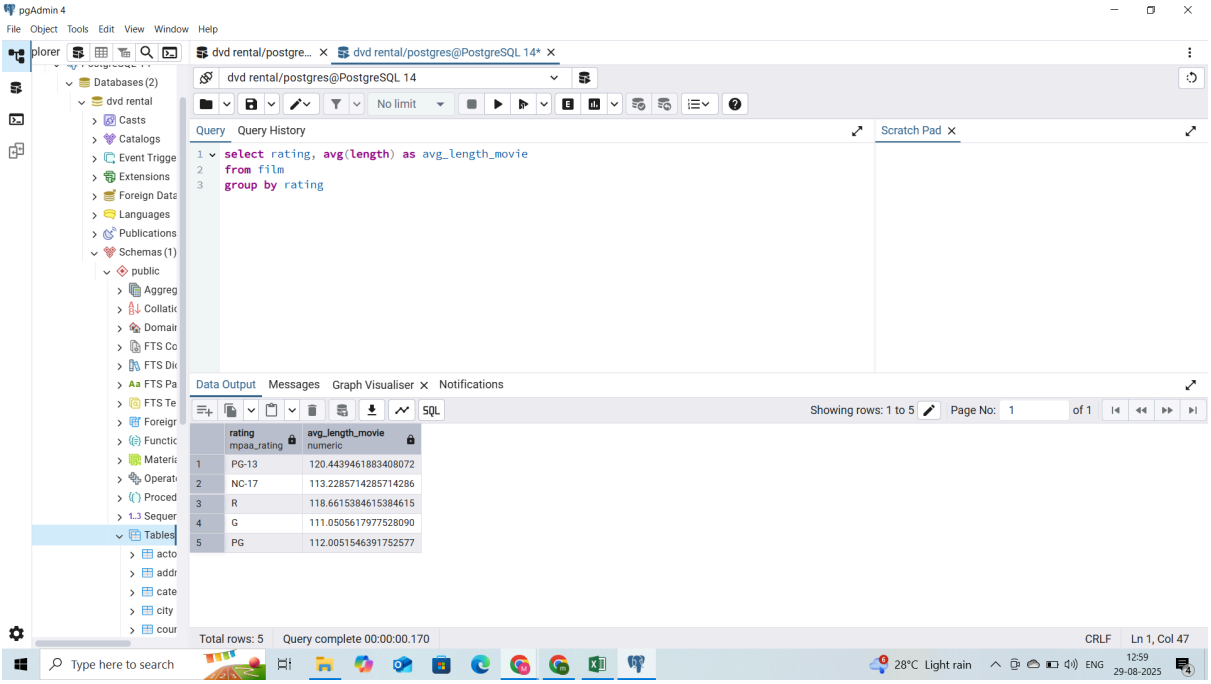
What is the count of movies across ratings?



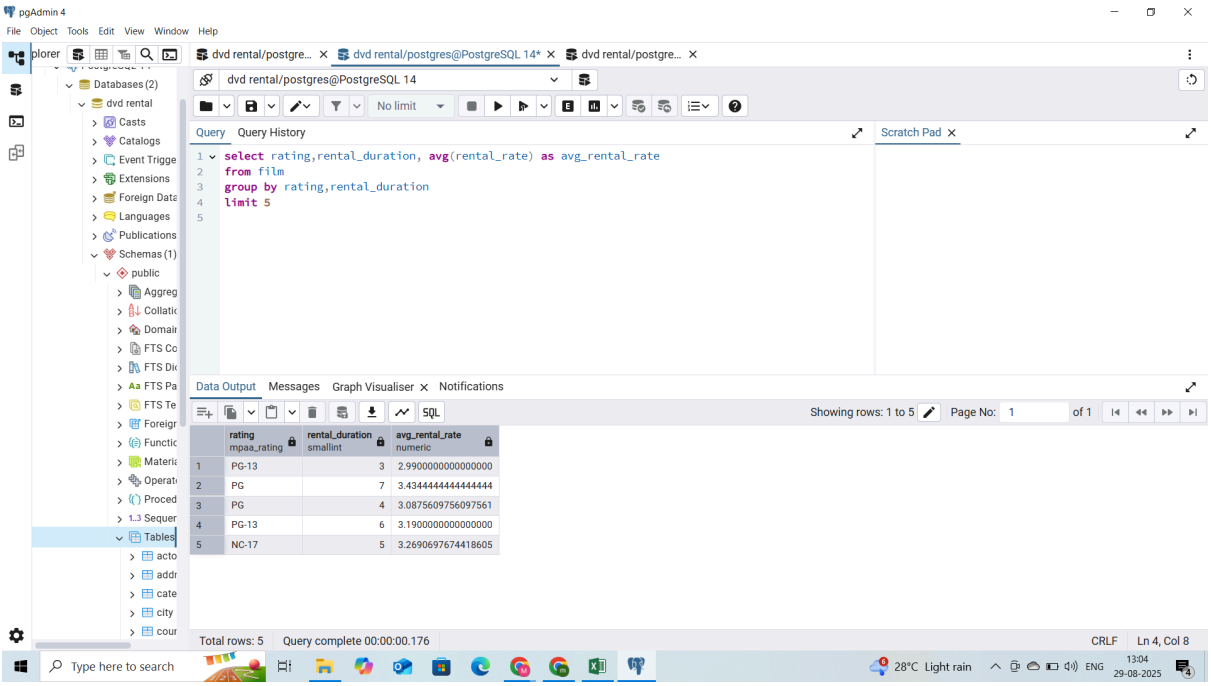
select
rating, count(*)
from film
group by rating

2)What is the average length of a movie across ratings?

select rating, avg(length) as avg_length_movie
from film
group by rating

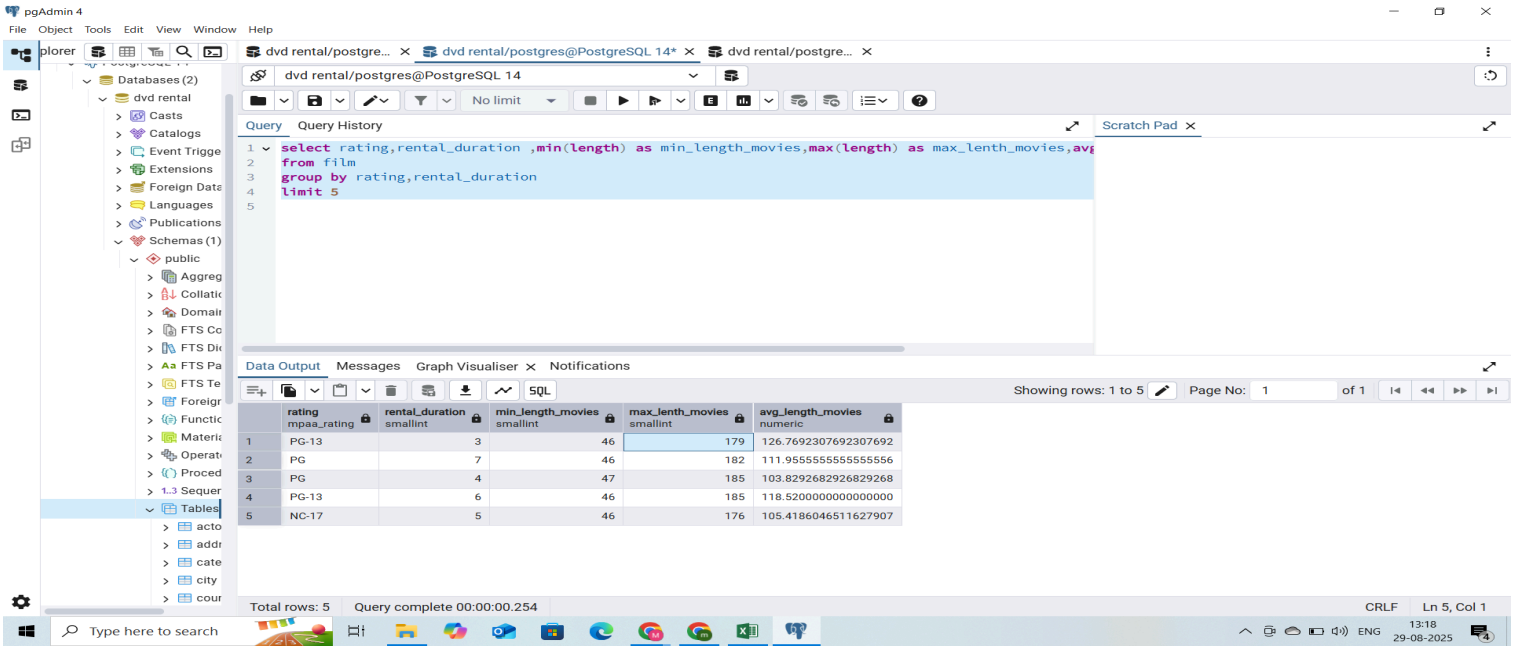


3) What is the average rental rate across rating & rental duration?
select rating,rental_duration, avg(rental_rate) as avg_rental_rate
from film
group by rating,rental_duration
limit 5



4)What is the max, min and avg of length across rating & rental duration?

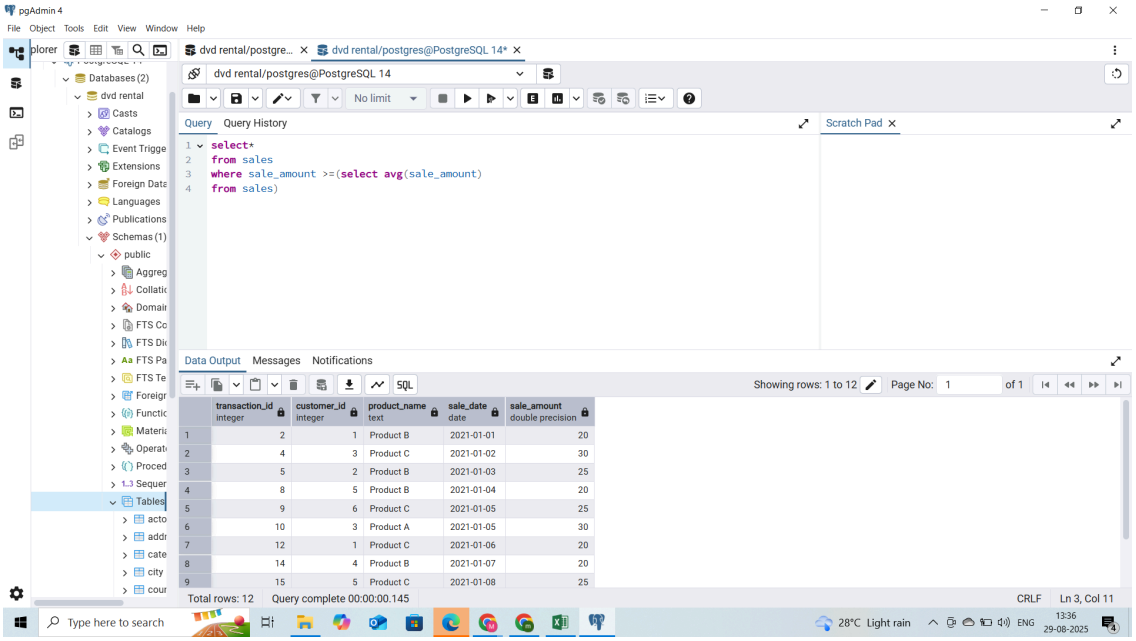
select rating,rental_duration ,min(length) as min_length_movies,max(length) as max_lenth_movies,avg(length)as avg_length_movies
from film
group by rating,rental_duration
limit 5



//Subquery to filter results//

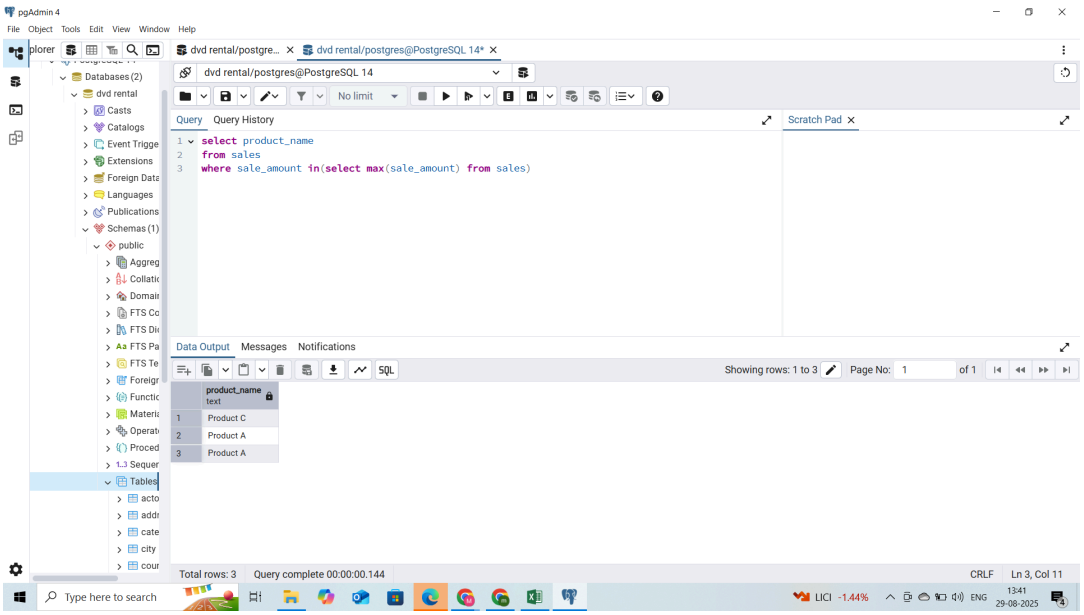
1)Get all the transactions greater than equal to avg of sale amount .

```
select*
from sales
where sale_amount >=(select avg(sale_amount)
from sales)
```



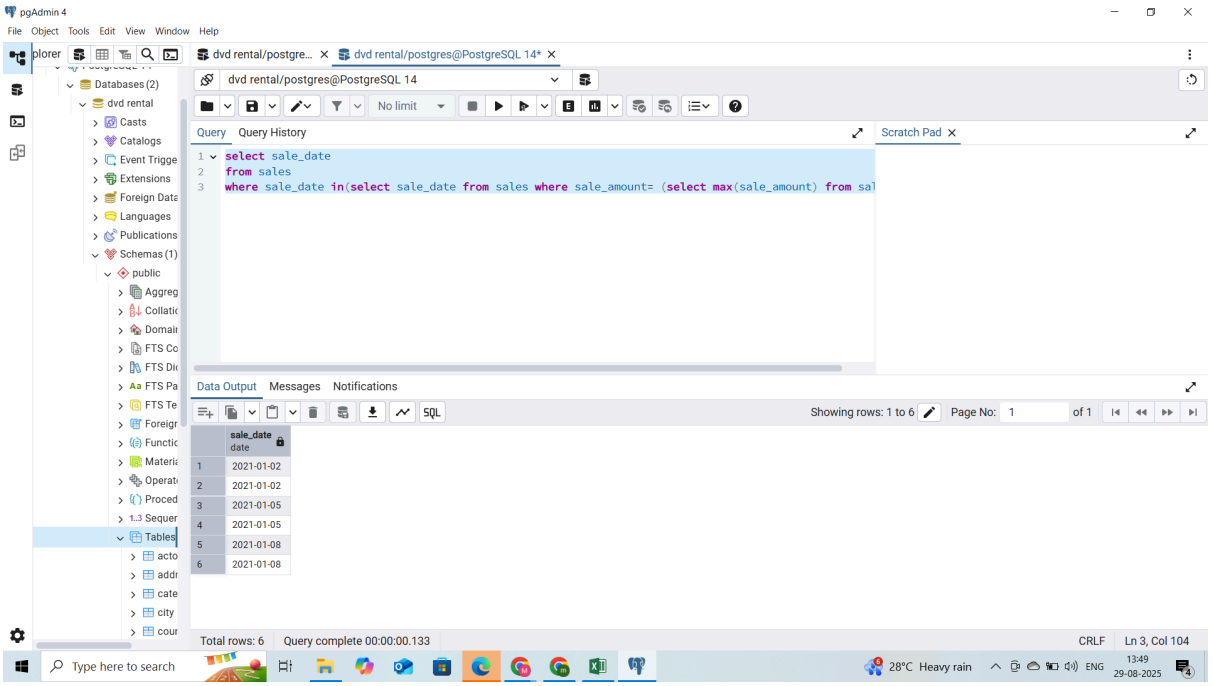
2)product_name which sale was the maximum sales

```
select product_name
from sales
where sale_amount in(select max(sale_amount) from sales)
```



3)count of transactions on dates on which sale was maximum sales amount

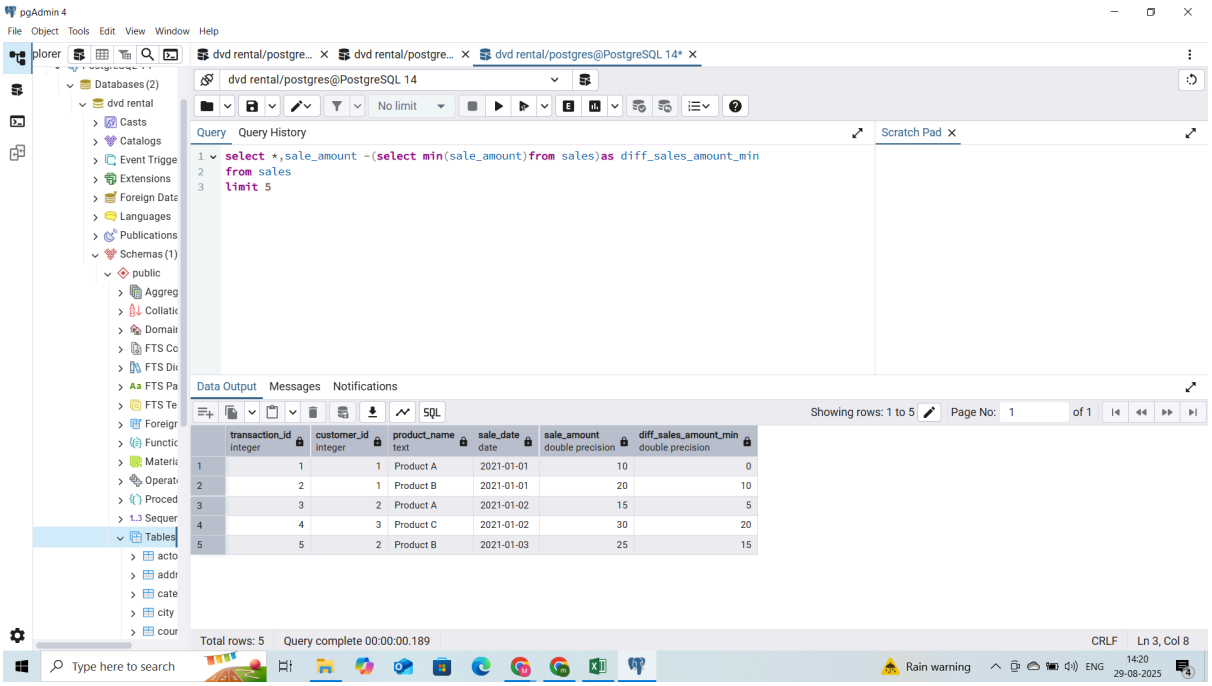
select sale_date
from sales
where sale_date in(select sale_date from sales where sale_amount= (select max(sale_amount) from sales))



//Using subquery to create column//

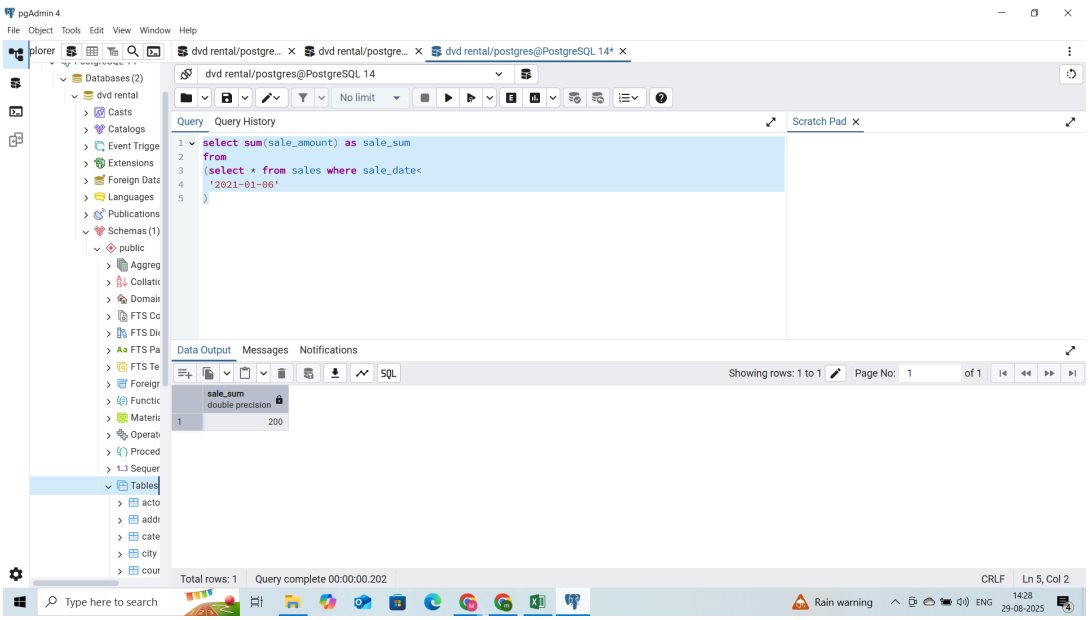
1) Create column which will be difference of sale amount and minimum of sale amount

select *,sale_amount -(select min(sale_amount)from sales)as diff_sales_amount_min from sales



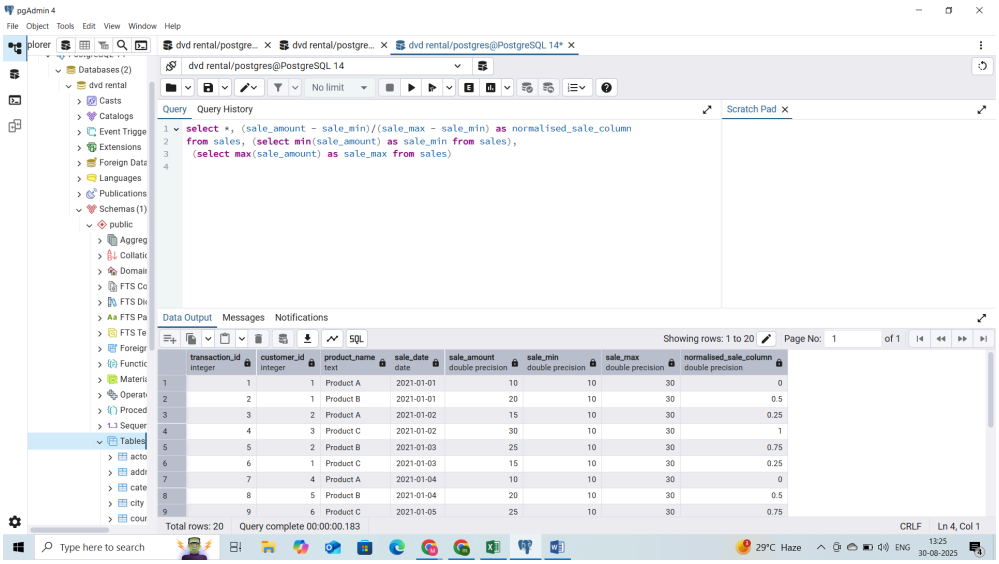
//Subquery to subset a table(filter a table)(for which values u want table in select)//
1)Sum of sale_amount for date before 6th Jan 2021

```
select sum(sale_amount) as sale_sum
from
(select * from sales where sale_date<
 '2021-01-06'
)
```

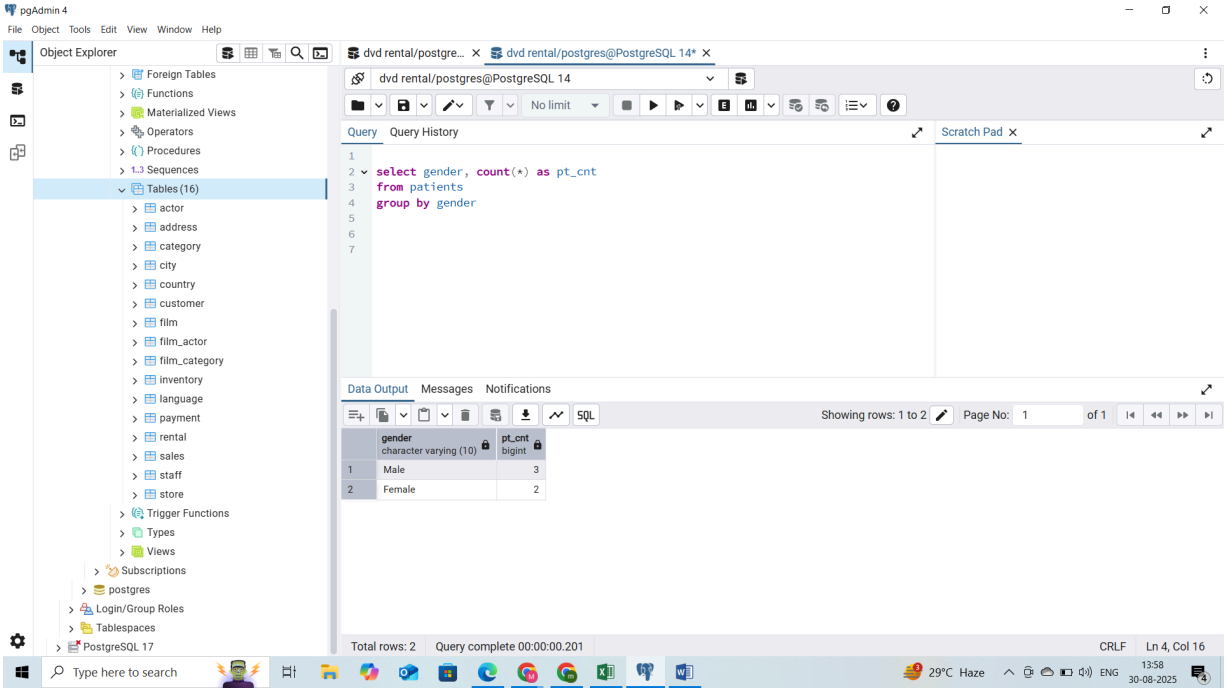


//Subquery to concat values//

select *, (sale_amount - sale_min)/(sale_max - sale_min) as normalised_sale_column from sales, (select min(sale_amount) as sale_min from sales), (select max(sale_amount) as sale_max from sales)

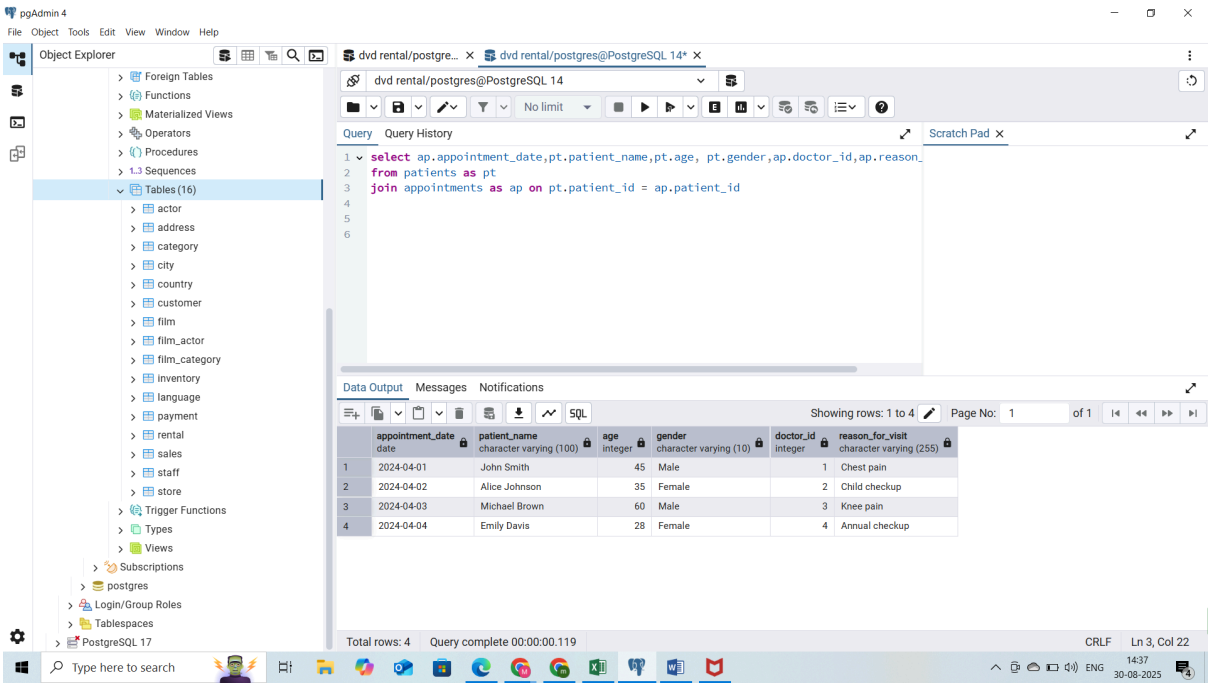


Q1. Find the number of male and female patients.
select gender, count(*) as pt_cnt
from patients
Group by gender



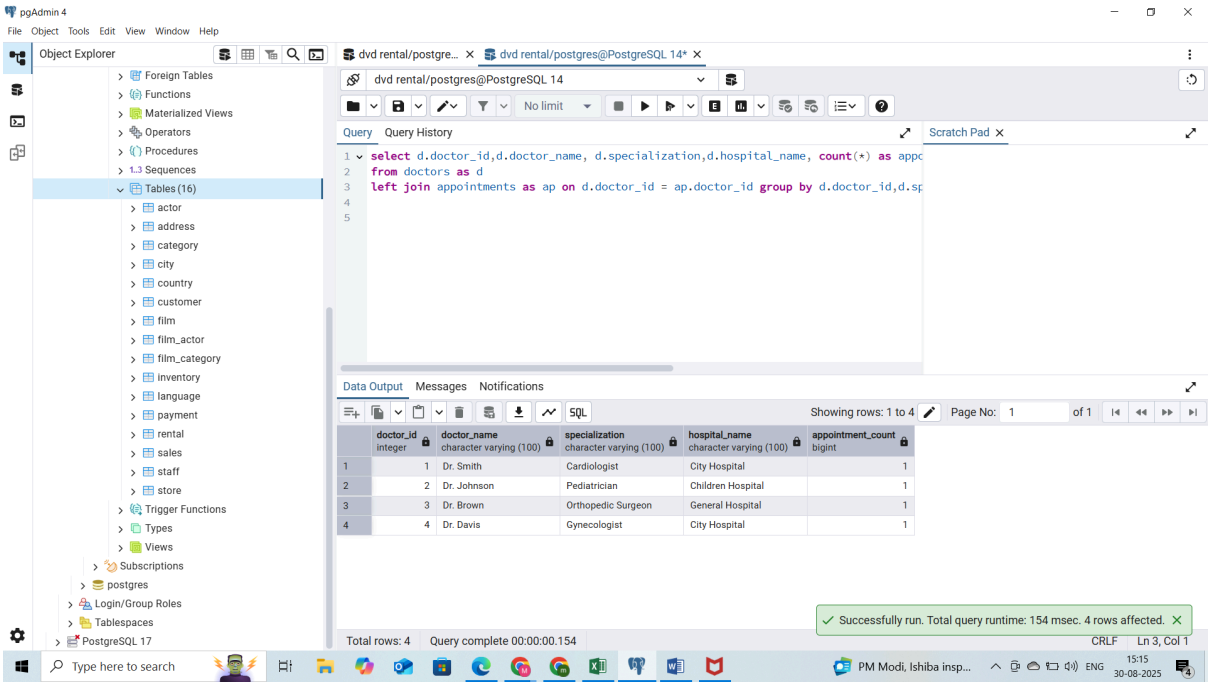
Q2. Find the details of patient as pratient_name, age, gender, doctor_id, appointment_date, reason_for_visit

select ap.appointment_date,pt.patient_name,pt.age, pt.gender,ap.doctor_id,ap.reason_for_visit from patients as pt join appointments as ap on pt.patient_id = ap.patient_id



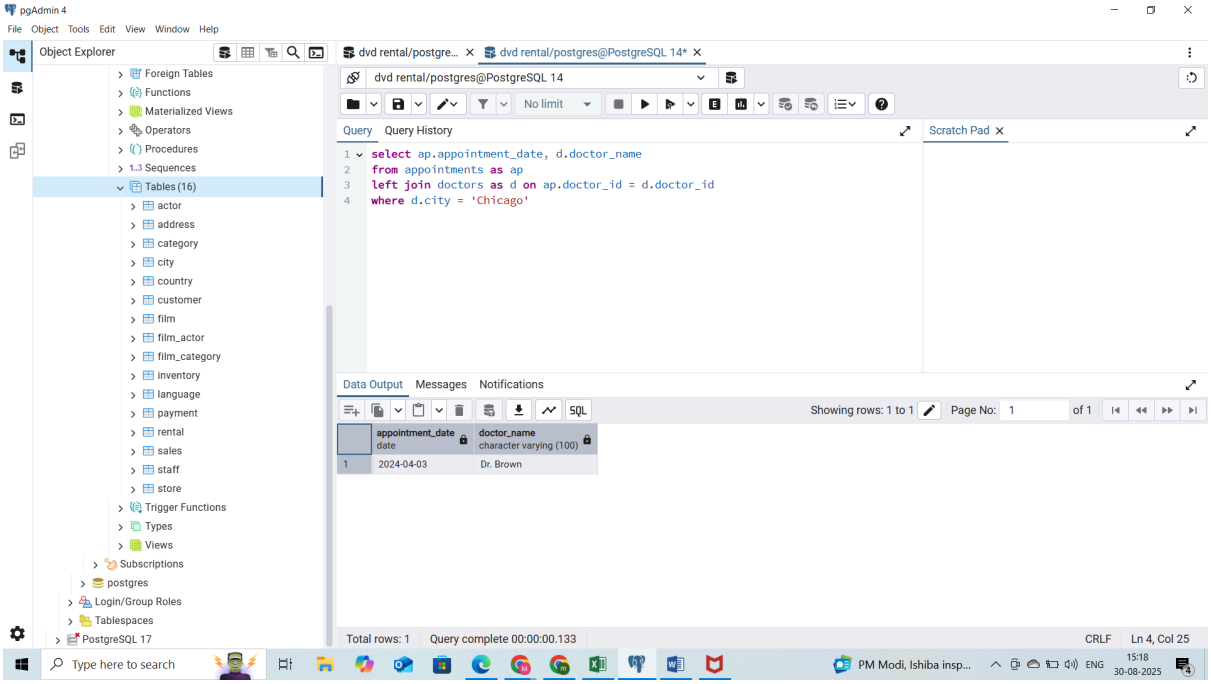
Q3. Display all doctors along with the number of appointments they have scheduled.

select d.doctor_id,d.doctor_name, d.specialization,d.hospital_name, count(*) as appointment_count from doctors as d left join appointments as ap on d.doctor_id = ap.doctor_id group by d.doctor_id,d.specialization,d.doctor_name, d.hospital_name order by doctor_id



Q4. Display the appointment dates for all appointments scheduled with doctors in Chicago.

select ap.appointment_date, d.doctor_name
 from appointments as ap
 left join doctors as d on ap.doctor_id = d.doctor_id where d.city = 'Chicago'



Q5. Generate all possible combinations of patients and doctors for referral purpose

```
select p.patient_name,d.doctor_name
from patients as p
cross join doctors as d
```