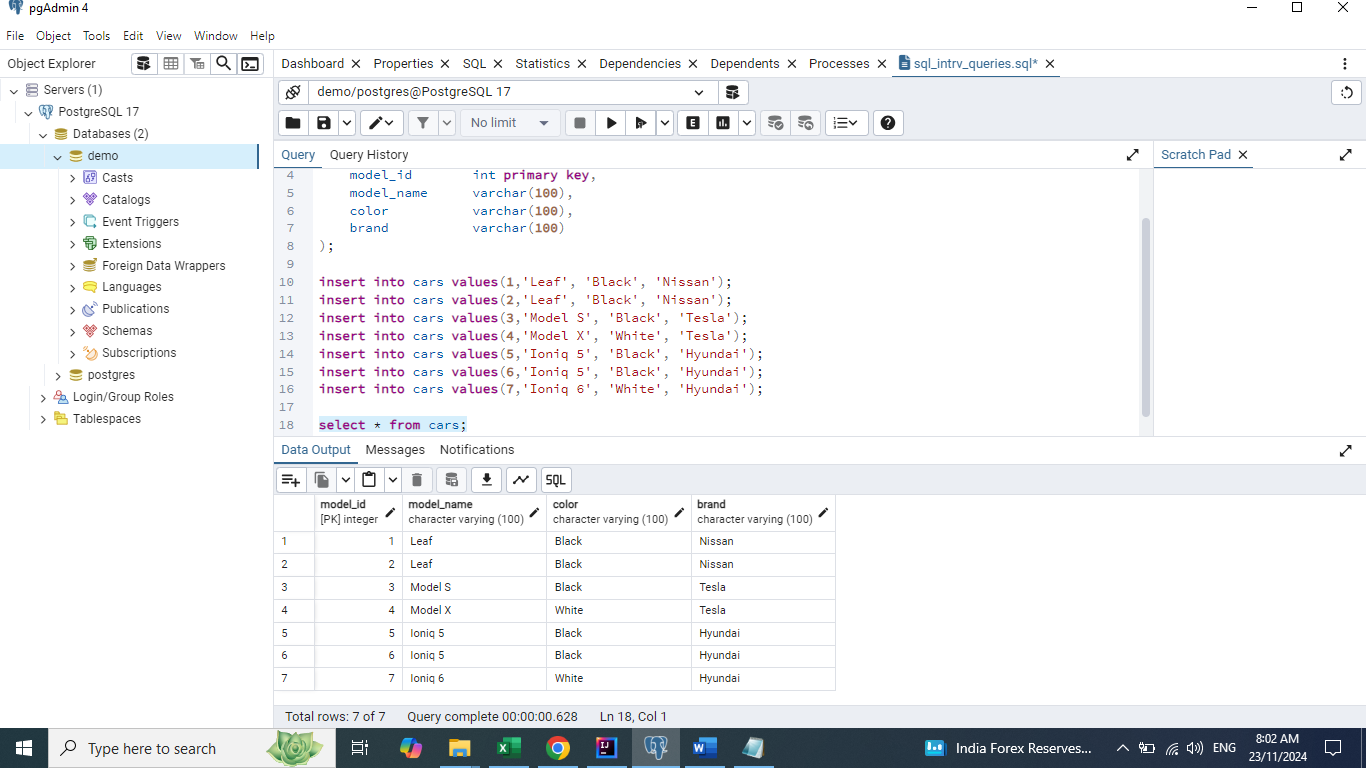
**SQL Interview Questions**

# Delete Duplicate Records



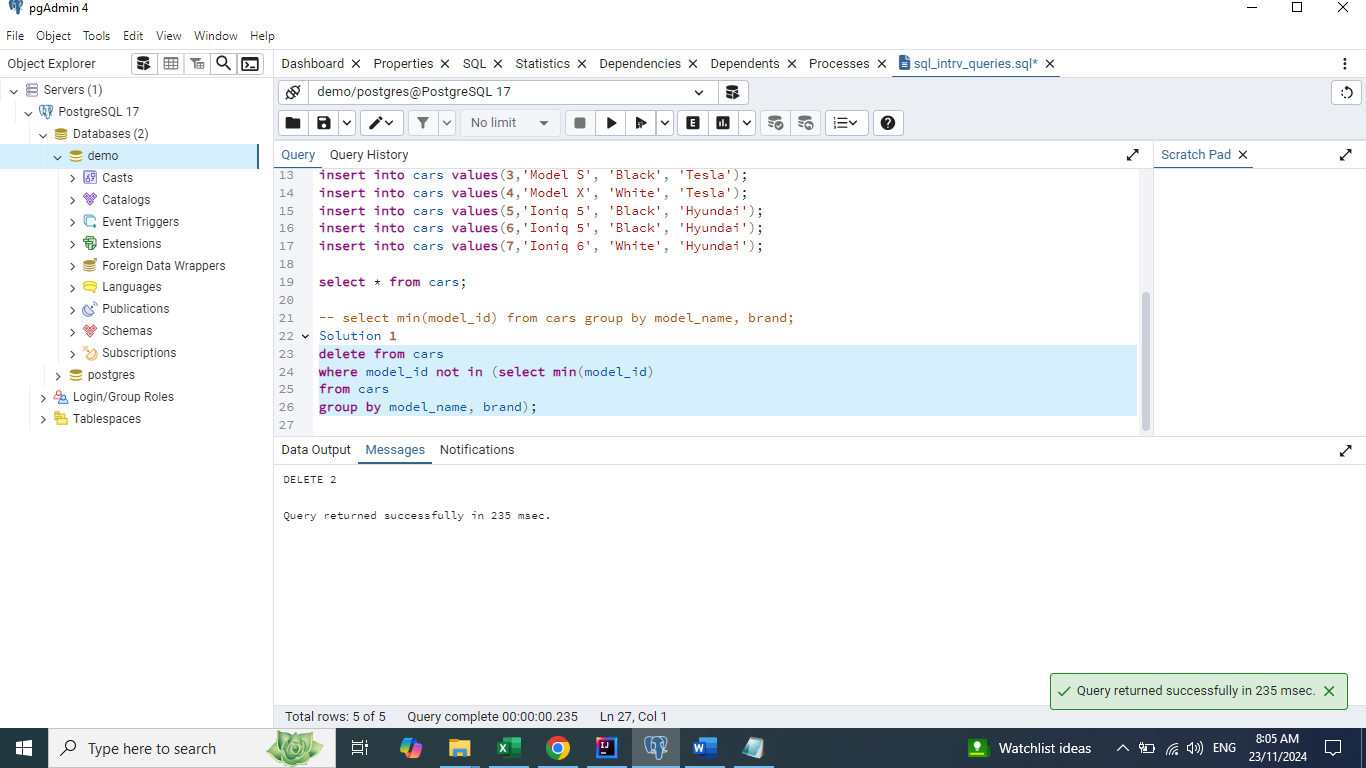
## Solution 1

**delete from cars**

**where model\_id not in (select min(model\_id)**

**from cars**

**group by model\_name, brand);**



## Solution 2

**Note:**

ctid - Kind of unique identifier which is present in PostgresSQL by default.

select max(ctid) from cars group by model\_name, brand having count(1) > 1;

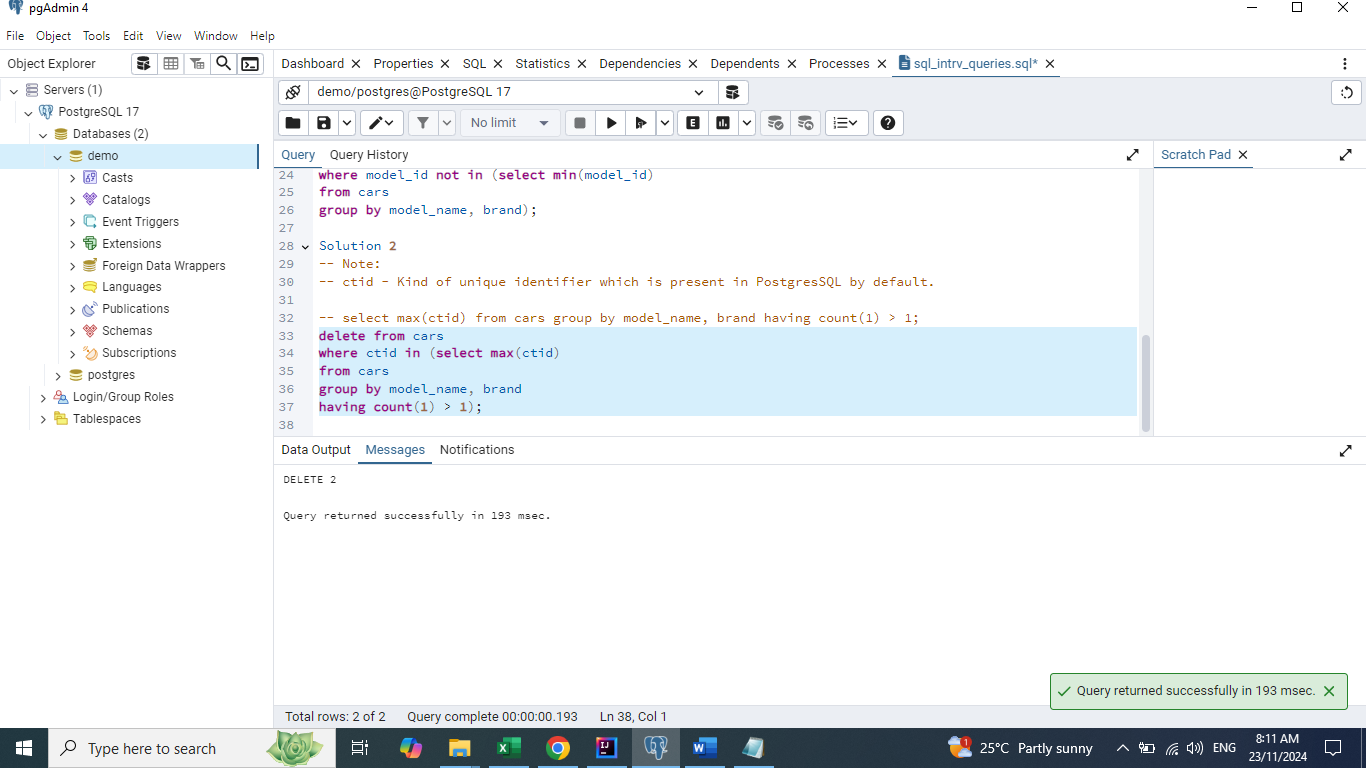
**delete from cars**

**where ctid in (select max(ctid)**

**from cars**

**group by model\_name, brand**

**having count(1) > 1);**



## Solution 3

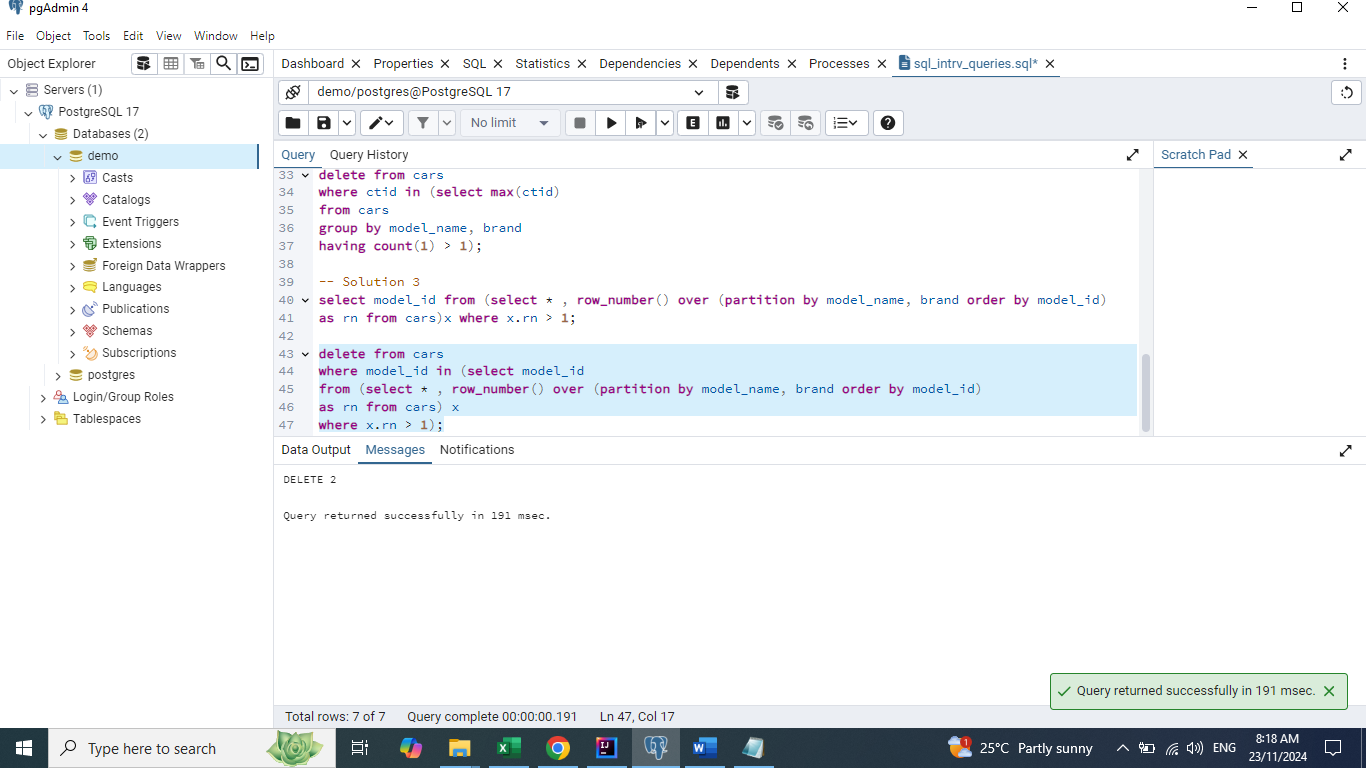
**delete from cars**

**where model\_id in (select model\_id**

**from (select \* , row\_number() over (partition by model\_name, brand order by model\_id)**

**as rn from cars) x**

**where x.rn > 1);**



# Display highest and lowest salary

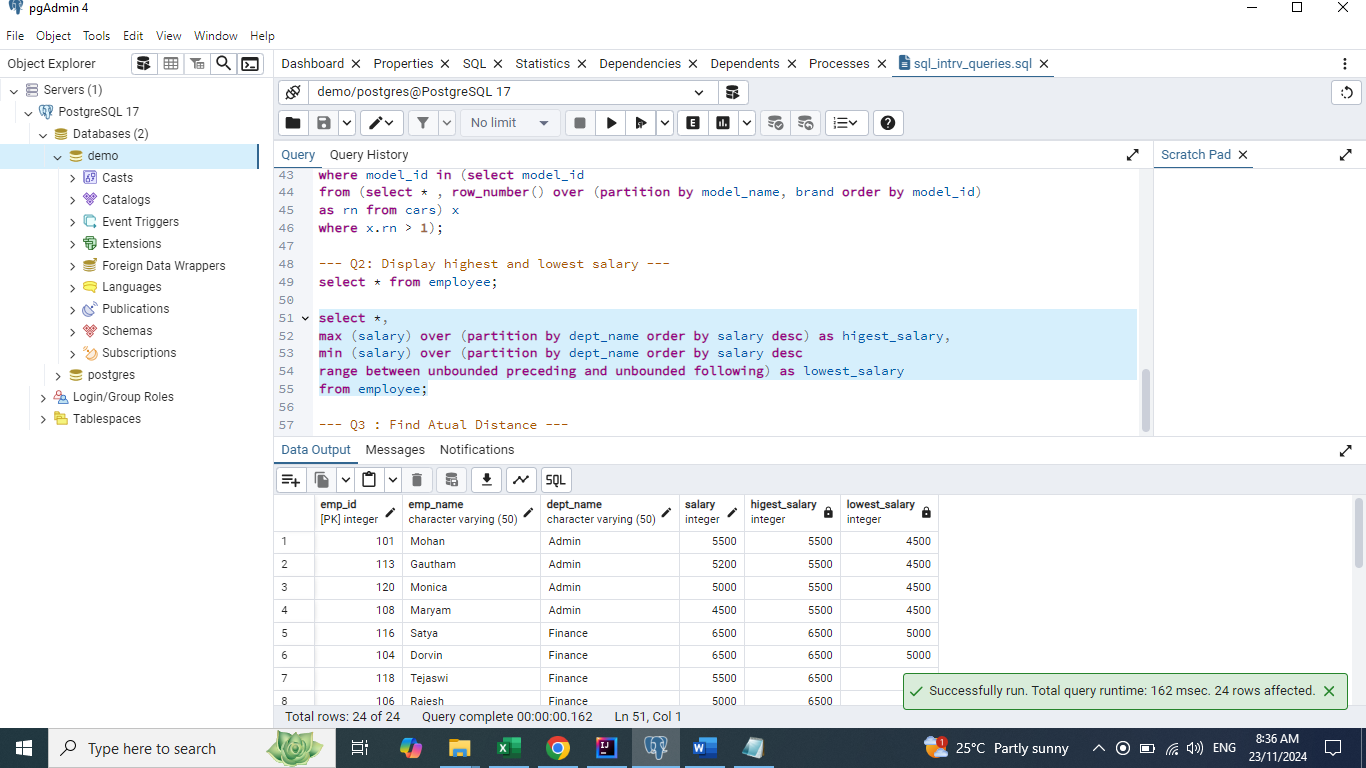
**select \*,**

**max (salary) over (partition by dept\_name order by salary desc) as higest\_salary,**

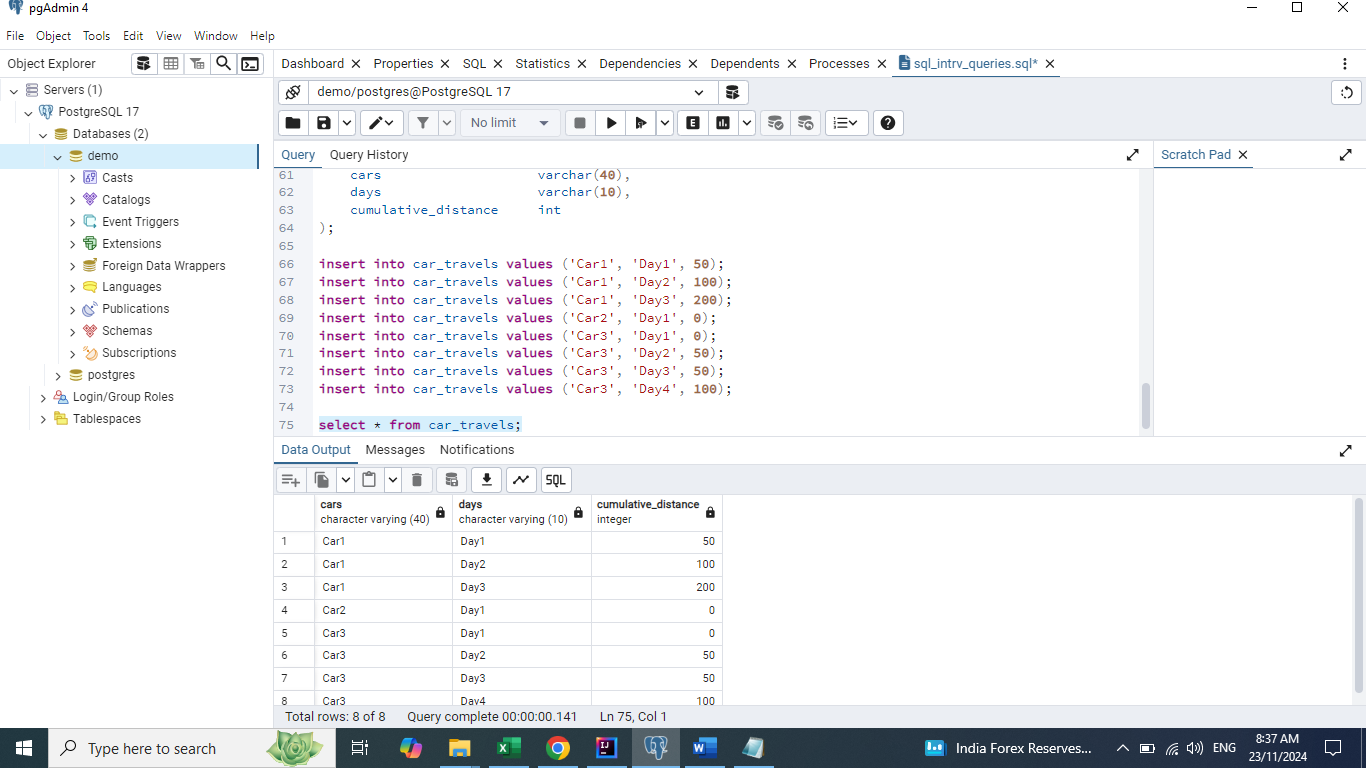
**min (salary) over (partition by dept\_name order by salary desc**

**range between unbounded preceding and unbounded following) as lowest\_salary**

**from employee;**



# Find Actual Distance

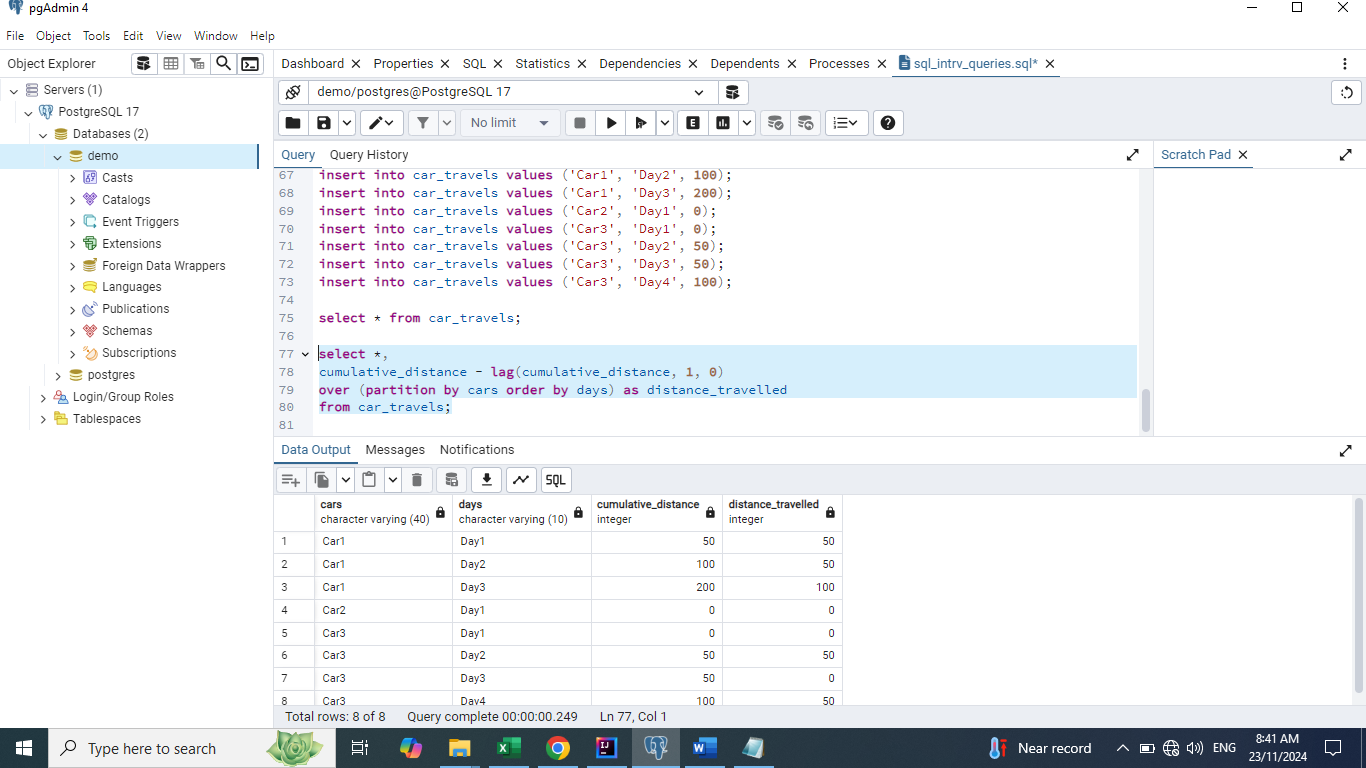


**select \*,**

**cumulative\_distance - lag(cumulative\_distance, 1, 0)**

**over (partition by cars order by days) as distance\_travelled**

**from car\_travels;**



# Convert the given input to expected output

**with cte as**

**(select \*,**

**row\_number() over() as rn**

**from src\_dest\_distance)**

**select t1.source, t1.destination, t1.distance**

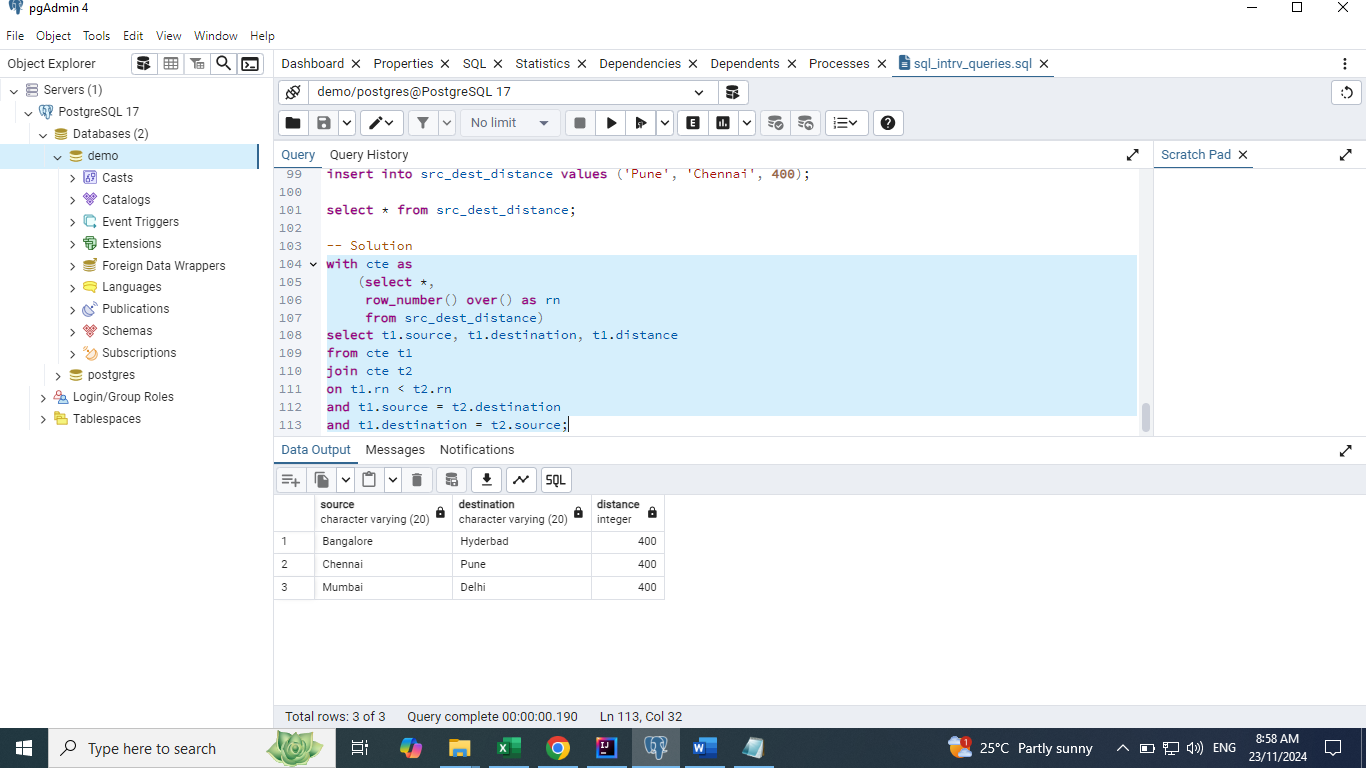
**from cte t1**

**join cte t2**

**on t1.rn < t2.rn**

**and t1.source = t2.destination**

**and t1.destination = t2.source;**



# Ungroup the given input data

**with recursive cte as**

**(select id, item\_name, total\_count, 1 as level**

**from travel\_items**

**union all**

**select cte.id, cte.item\_name, cte.total\_count - 1, level + 1 as level**

**from cte**

**join travel\_items t**

**on t.item\_name = cte.item\_name**

**and t.id = cte.id**

**where cte.total\_count > 1)**

**select id, item\_name, level**

**from cte**

**order by 3;**

