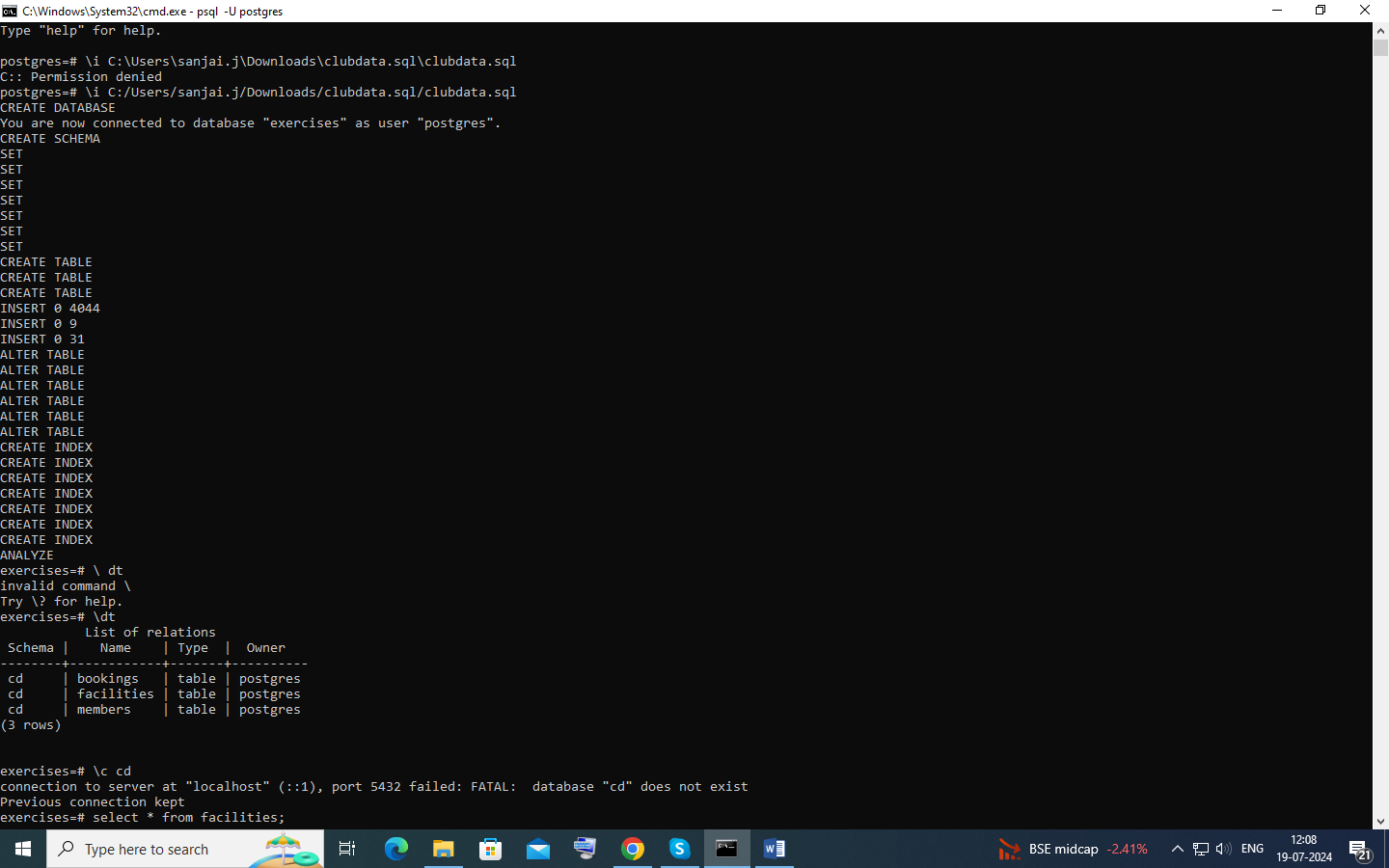
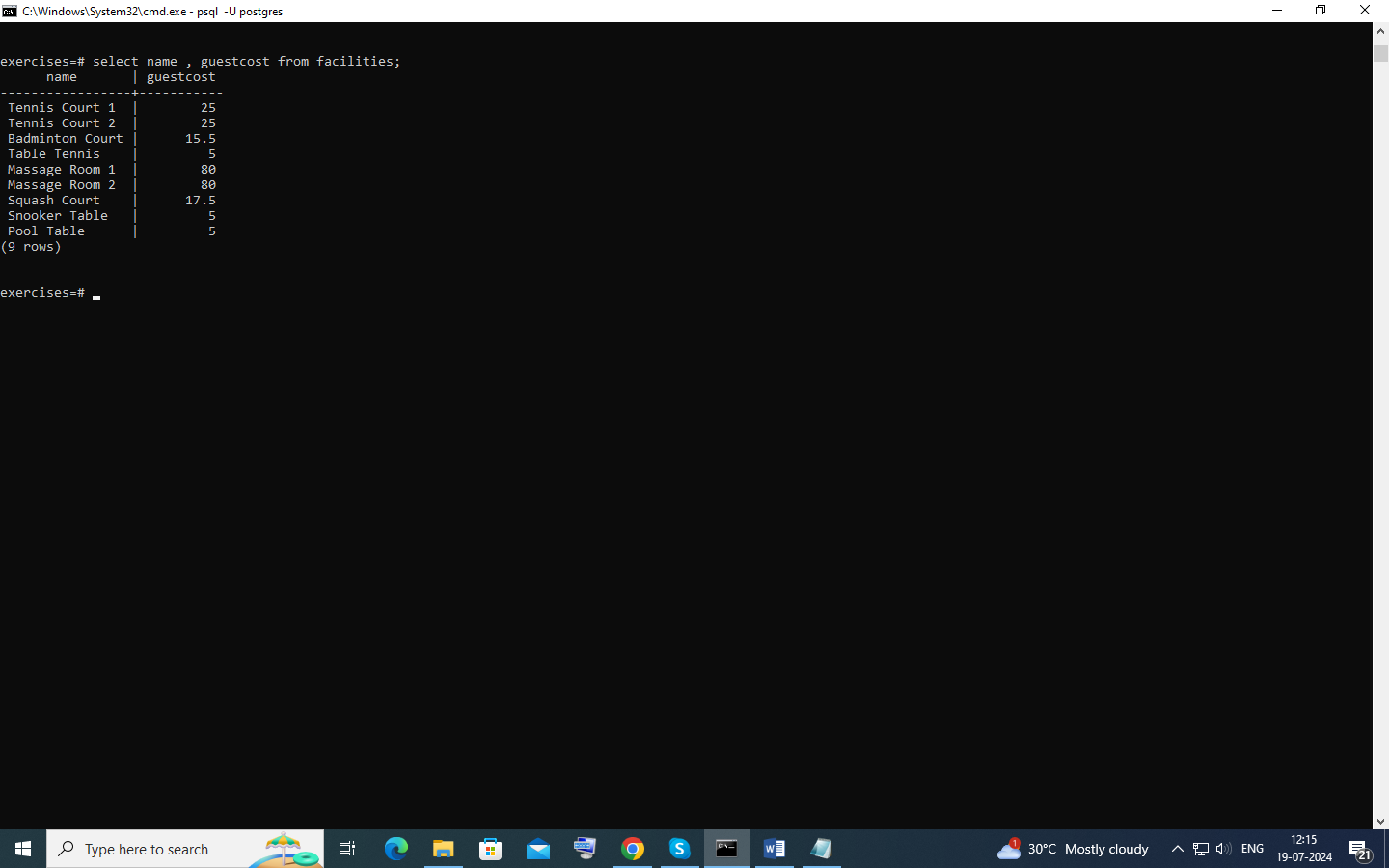
Retrieve everything from a table

\i C:/Users/sanjai.j/Downloads/clubdata.sql/clubdata.sql



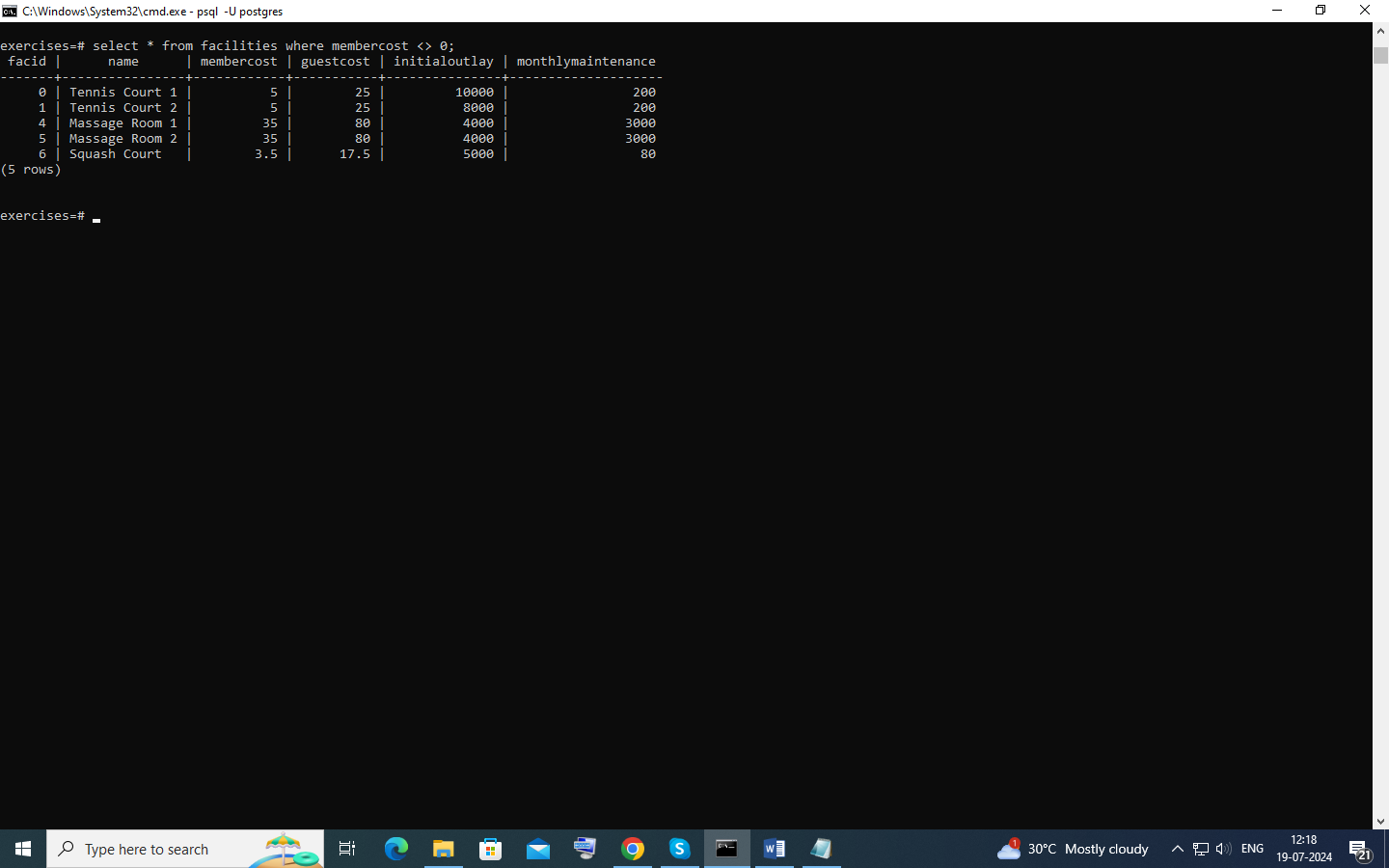
Retrieve specific columns from a table

select name , guestcost from facilities;



Control which rows are retrieved

select \* from facilities where membercost <> 0;



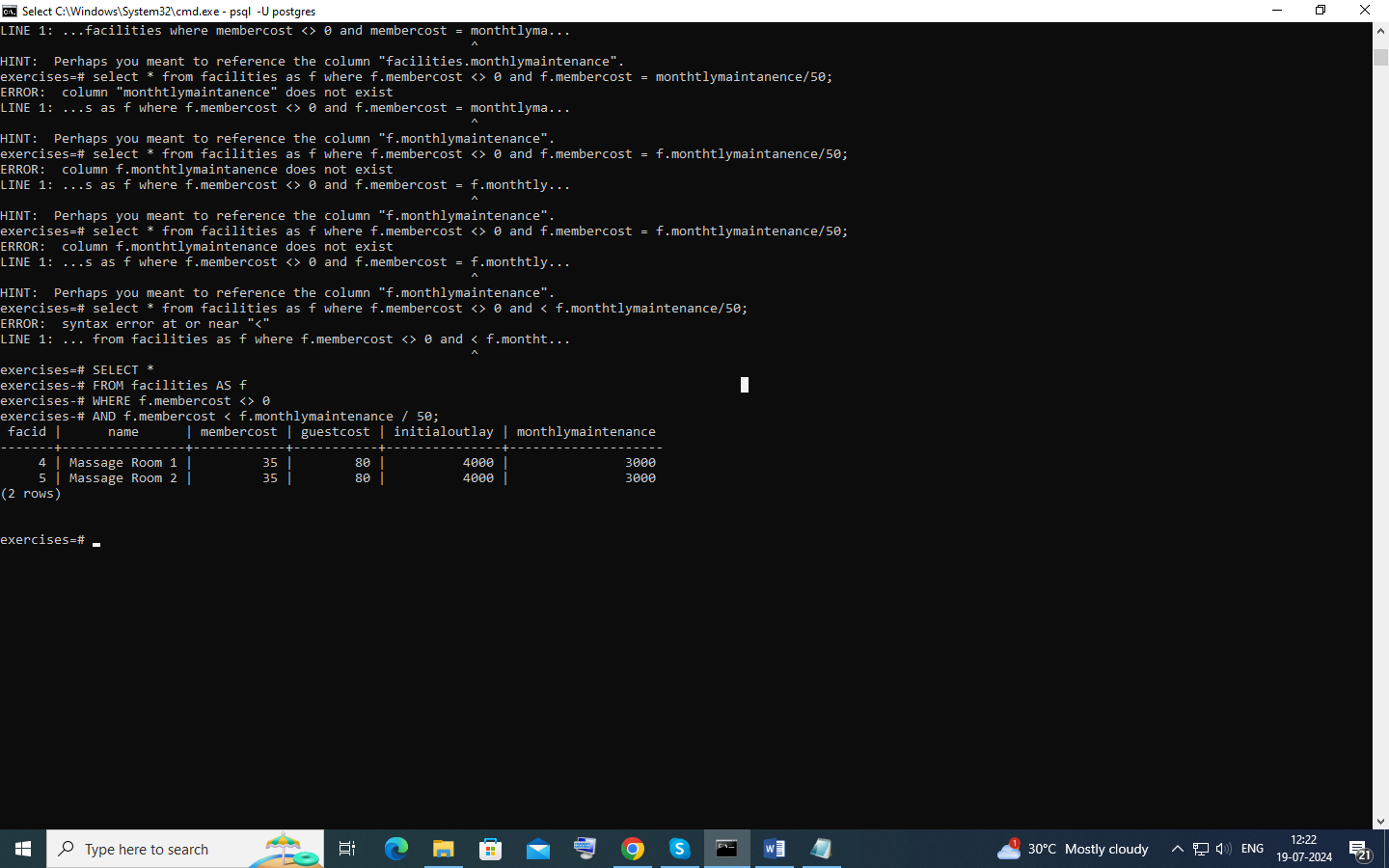
Control which rows are retrieved - part 2

SELECT \*

FROM facilities AS f

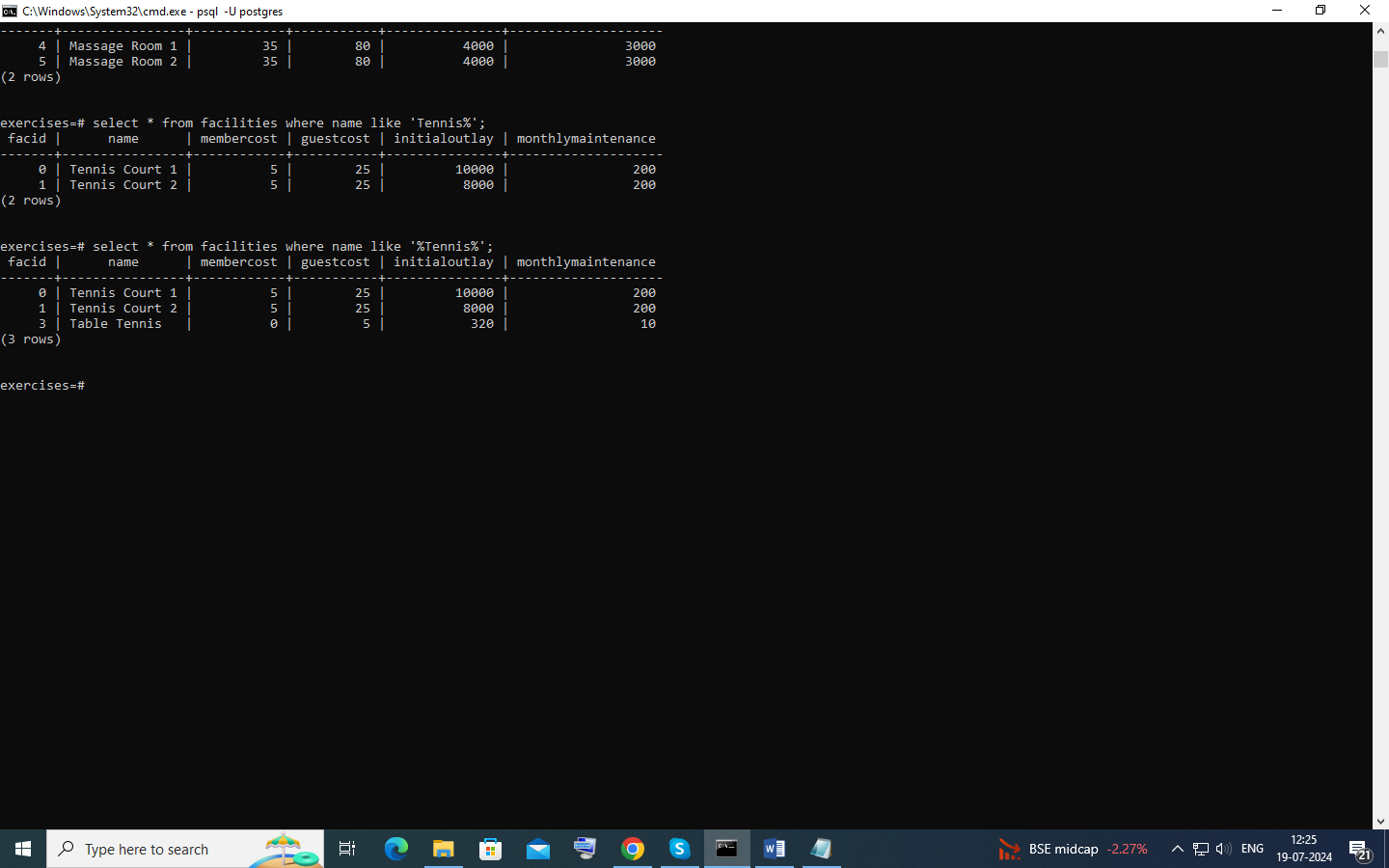
WHERE f.membercost <> 0

AND f.membercost < f.monthlymaintenance / 50;



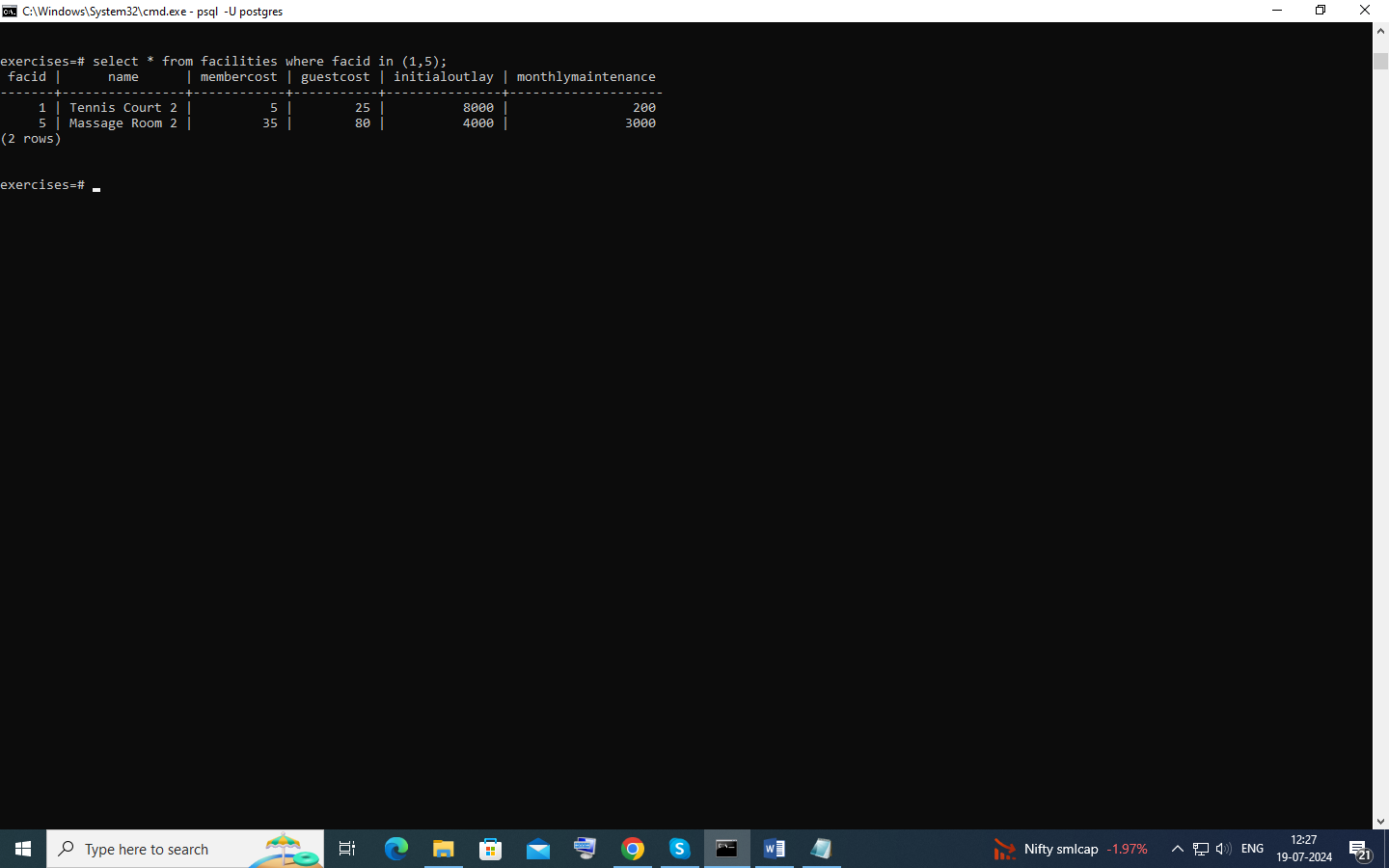
Basic string searches

select \* from facilities where name like '%Tennis%';



Matching against multiple possible values

select \* from facilities where facid in (1,5);



# Classify results into buckets

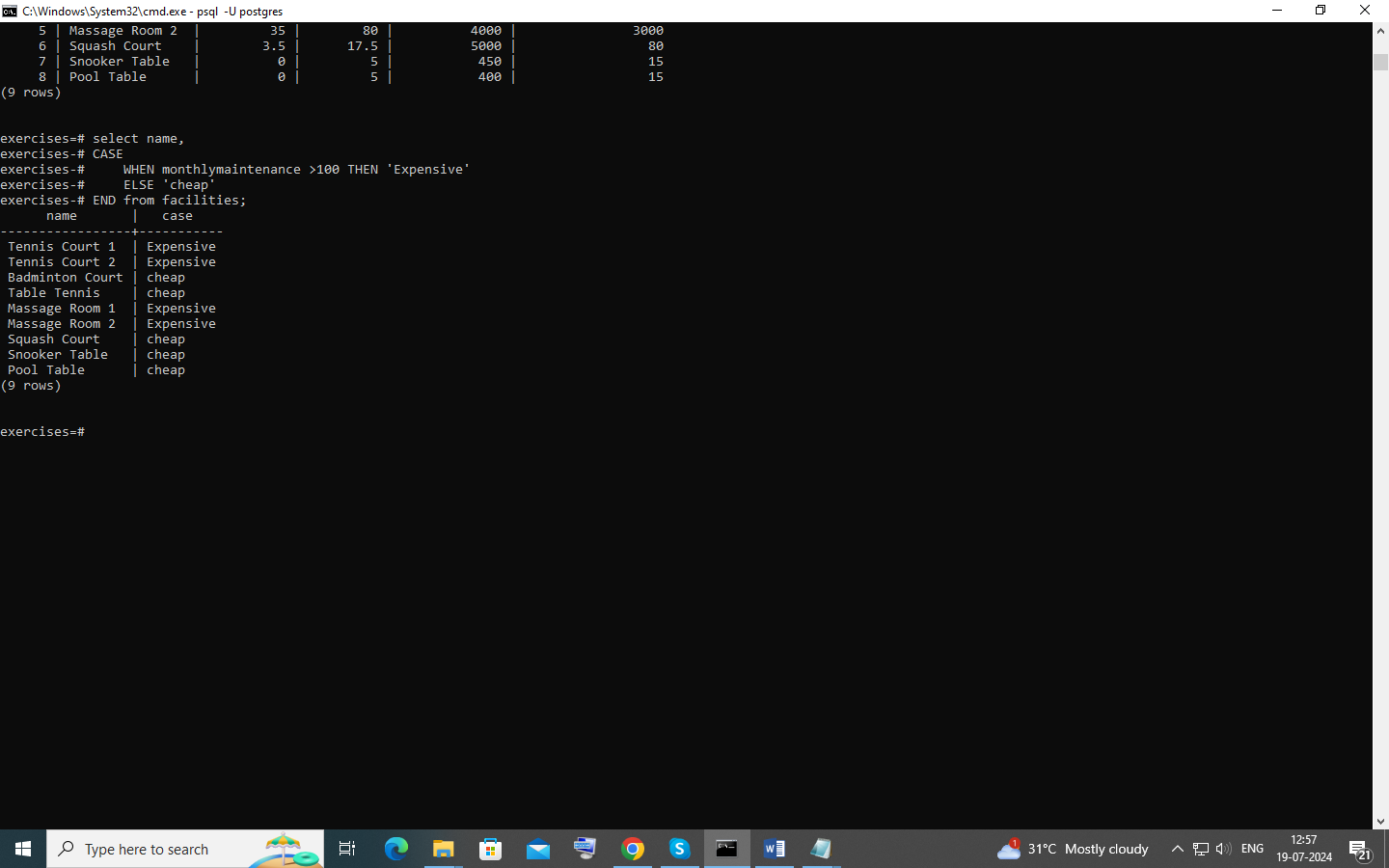
select name,

CASE

WHEN monthlymaintenance >100 THEN 'Expensive'

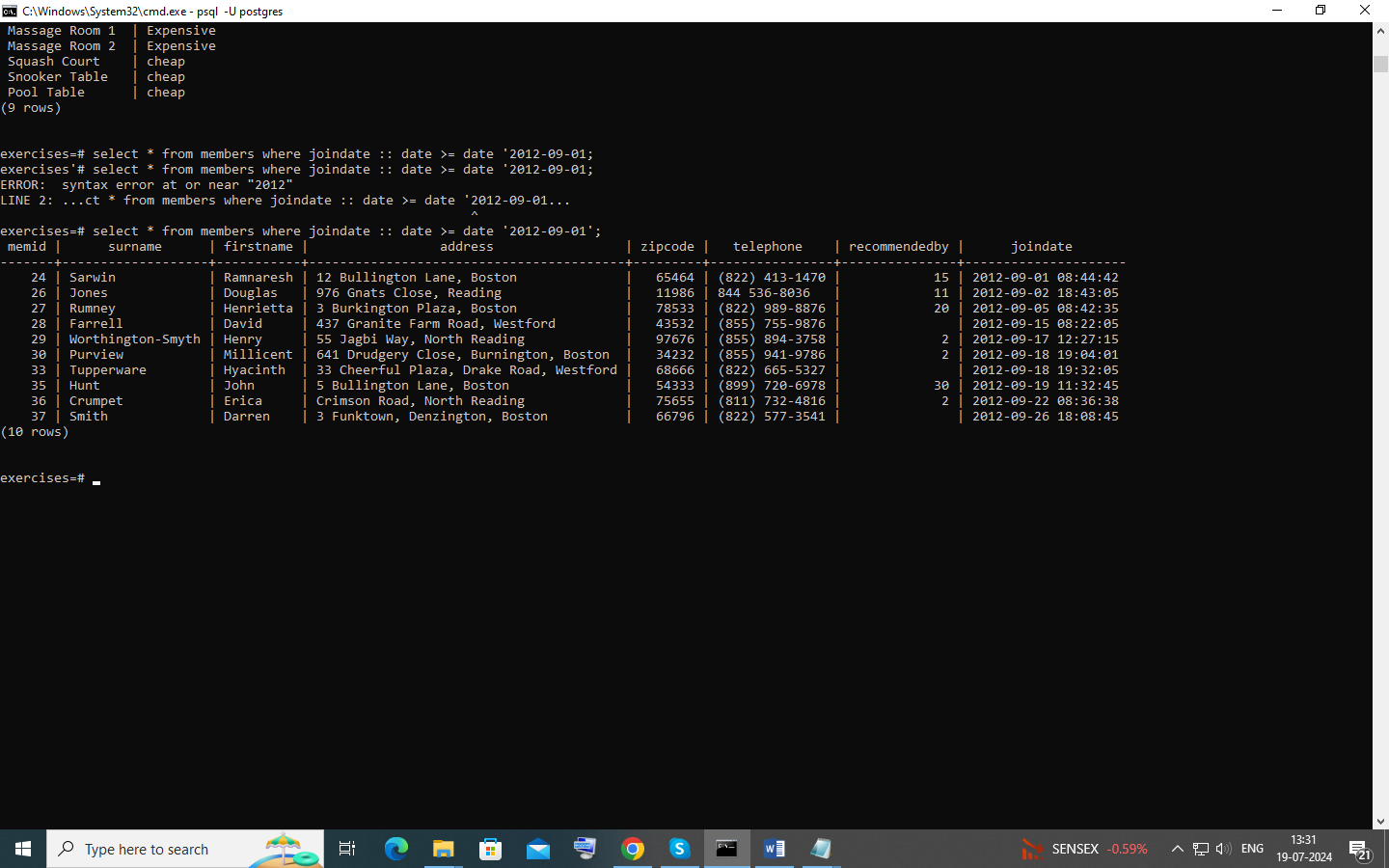
ELSE 'cheap'

END from facilities;



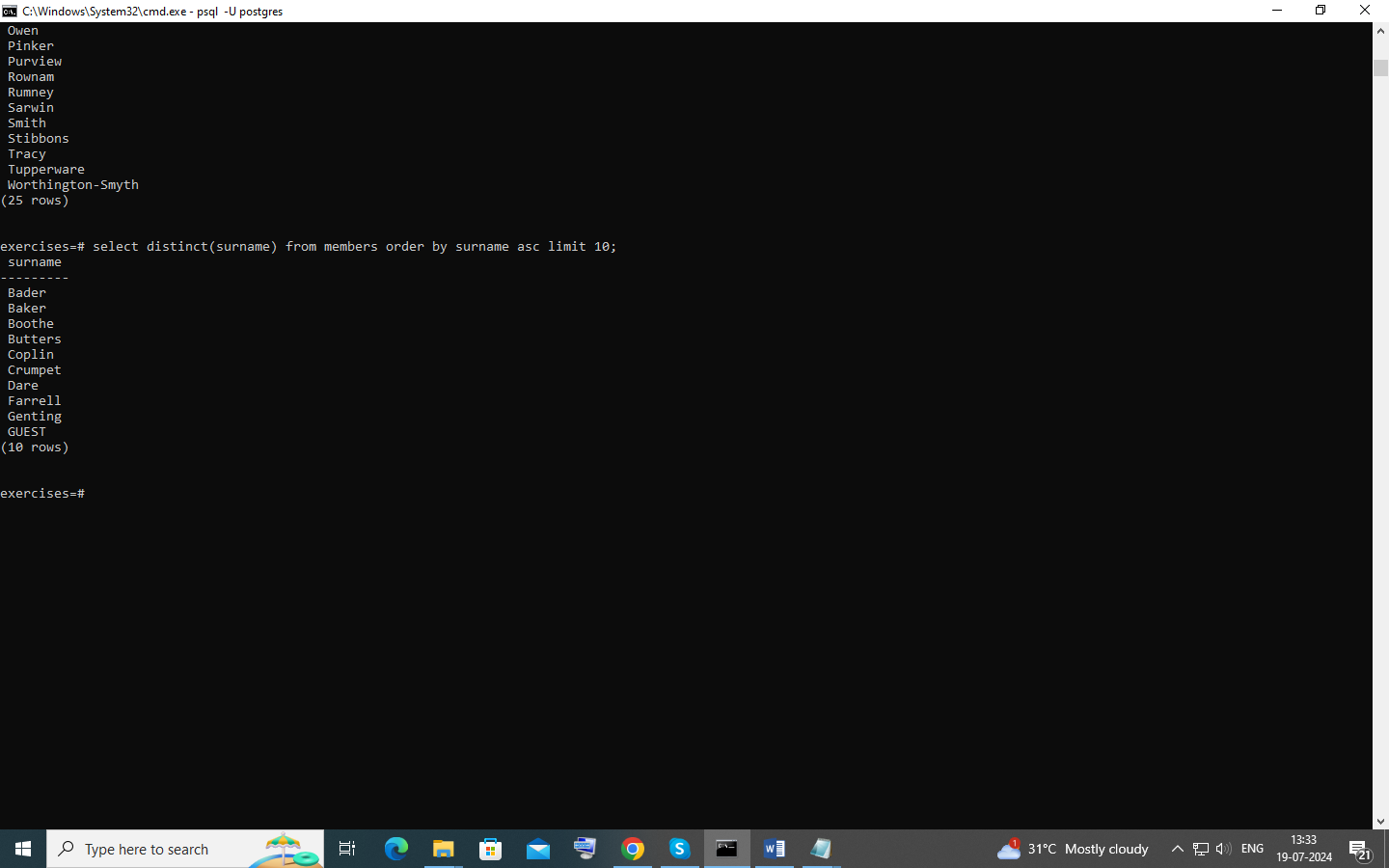
# Working with dates

select \* from members where joindate :: date >= date '2012-09-01';



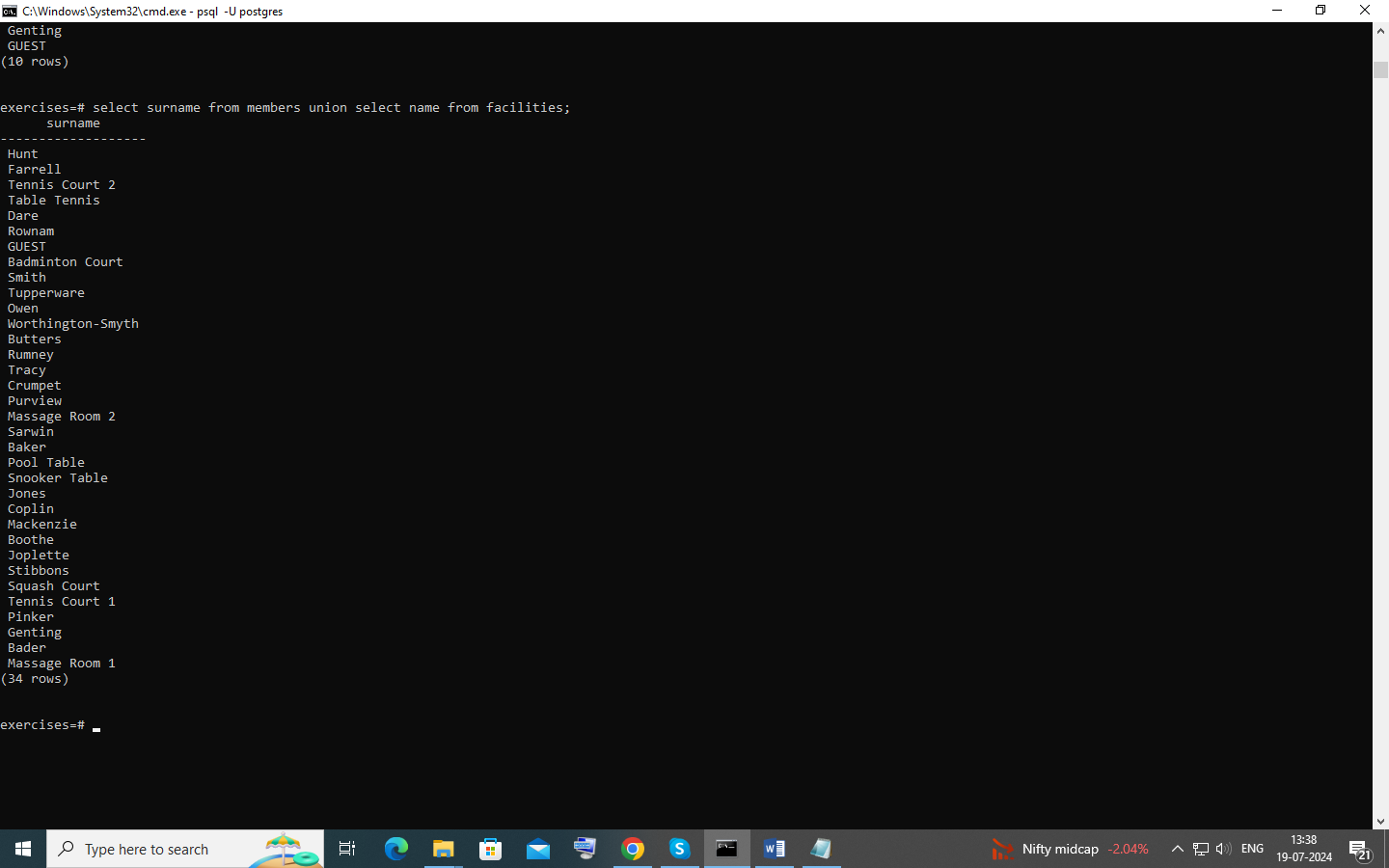
# Removing duplicates, and ordering results

select distinct(surname) from members order by surname asc limit 10;



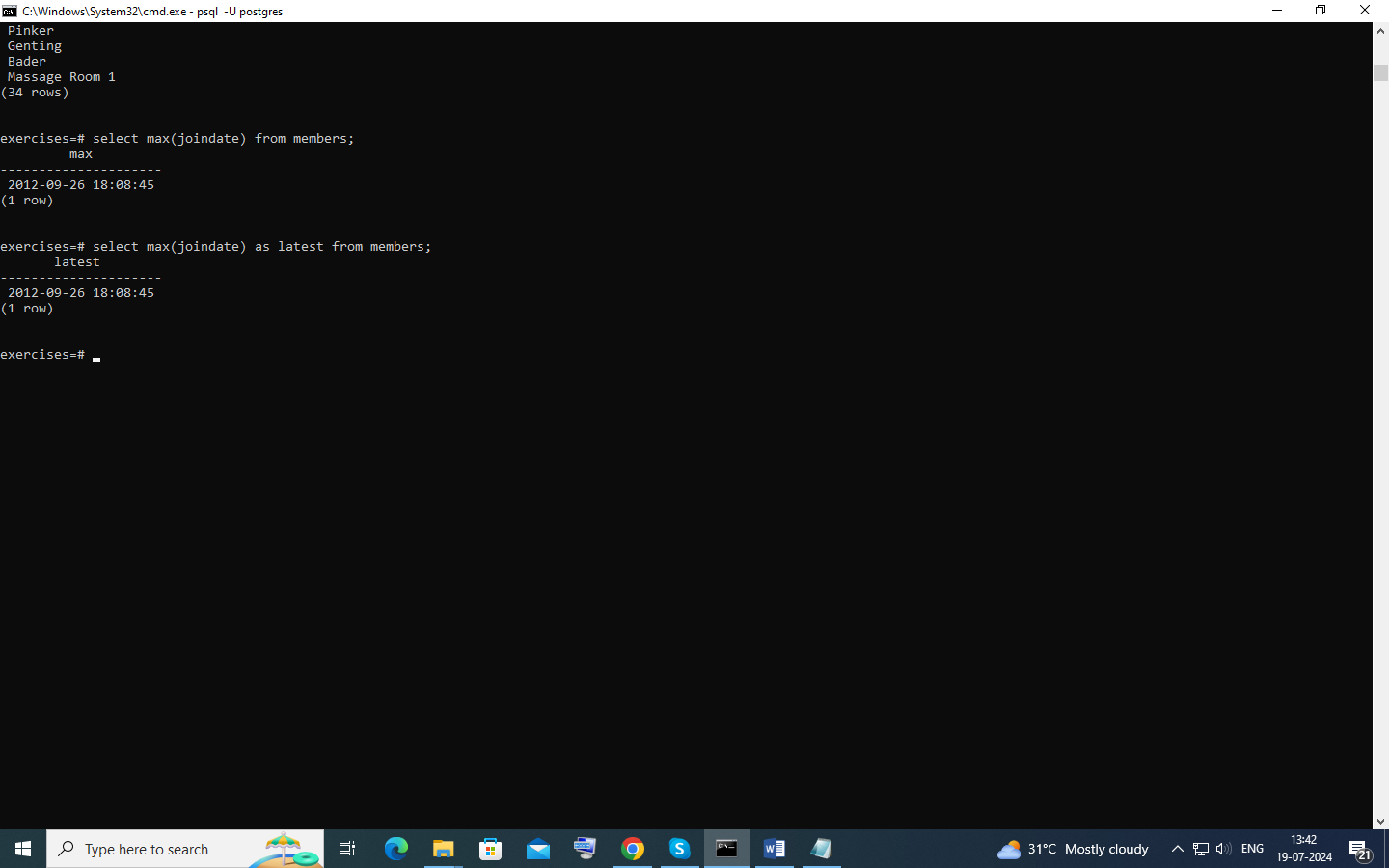
# Combining results from multiple queries

select surname from members union select name from facilities;



# Simple aggregation

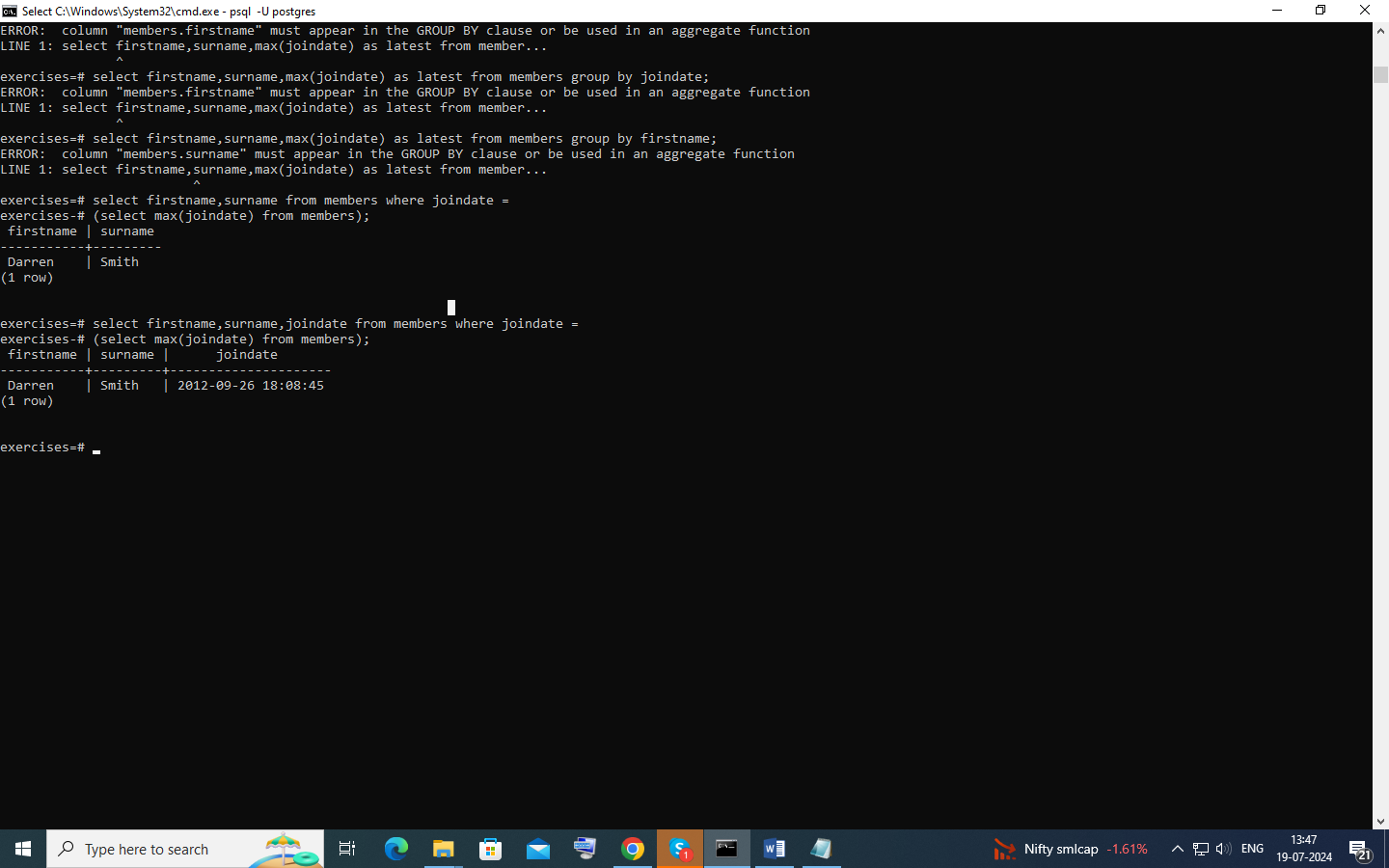
select max(joindate) as latest from members;



# More aggregation

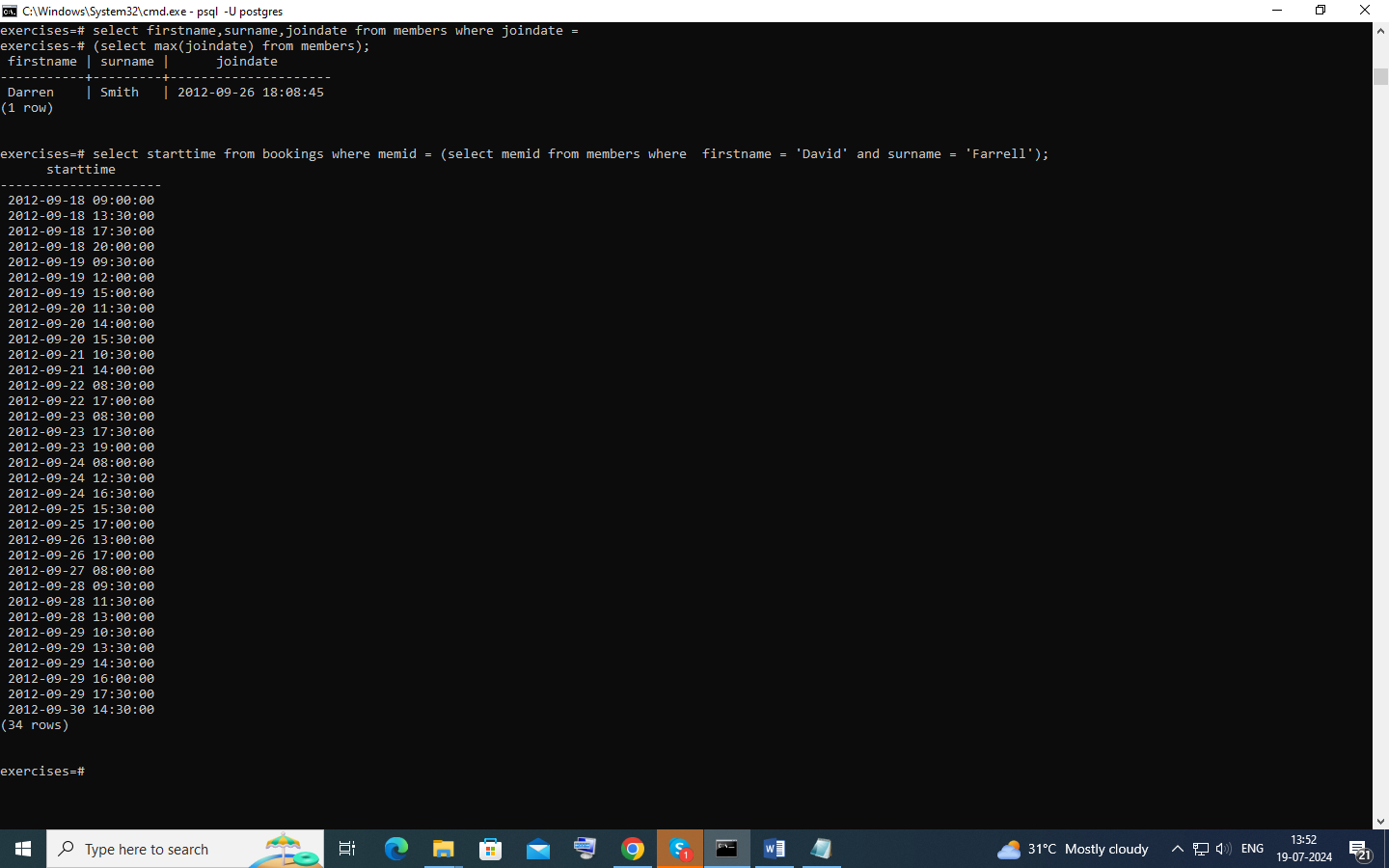
select firstname,surname,joindate from members where joindate =

(select max(joindate) from members);



# Retrieve the start times of members' bookings

select starttime from bookings where memid = (select memid from members where firstname = 'David' and surname = 'Farrell');



Work out the start times of bookings for tennis courts

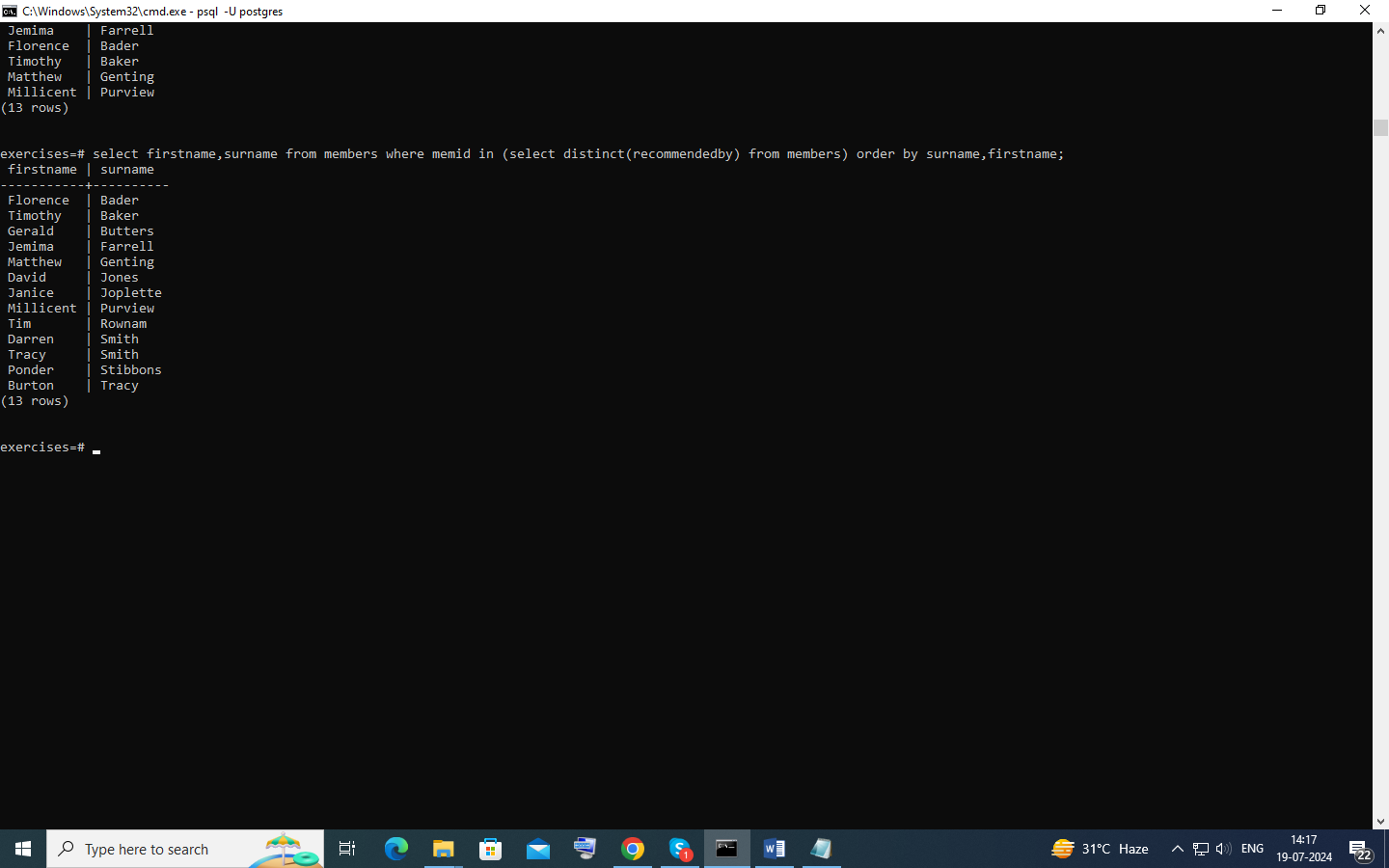
SELECT

b.starttime,f.name

FROM facilities as f

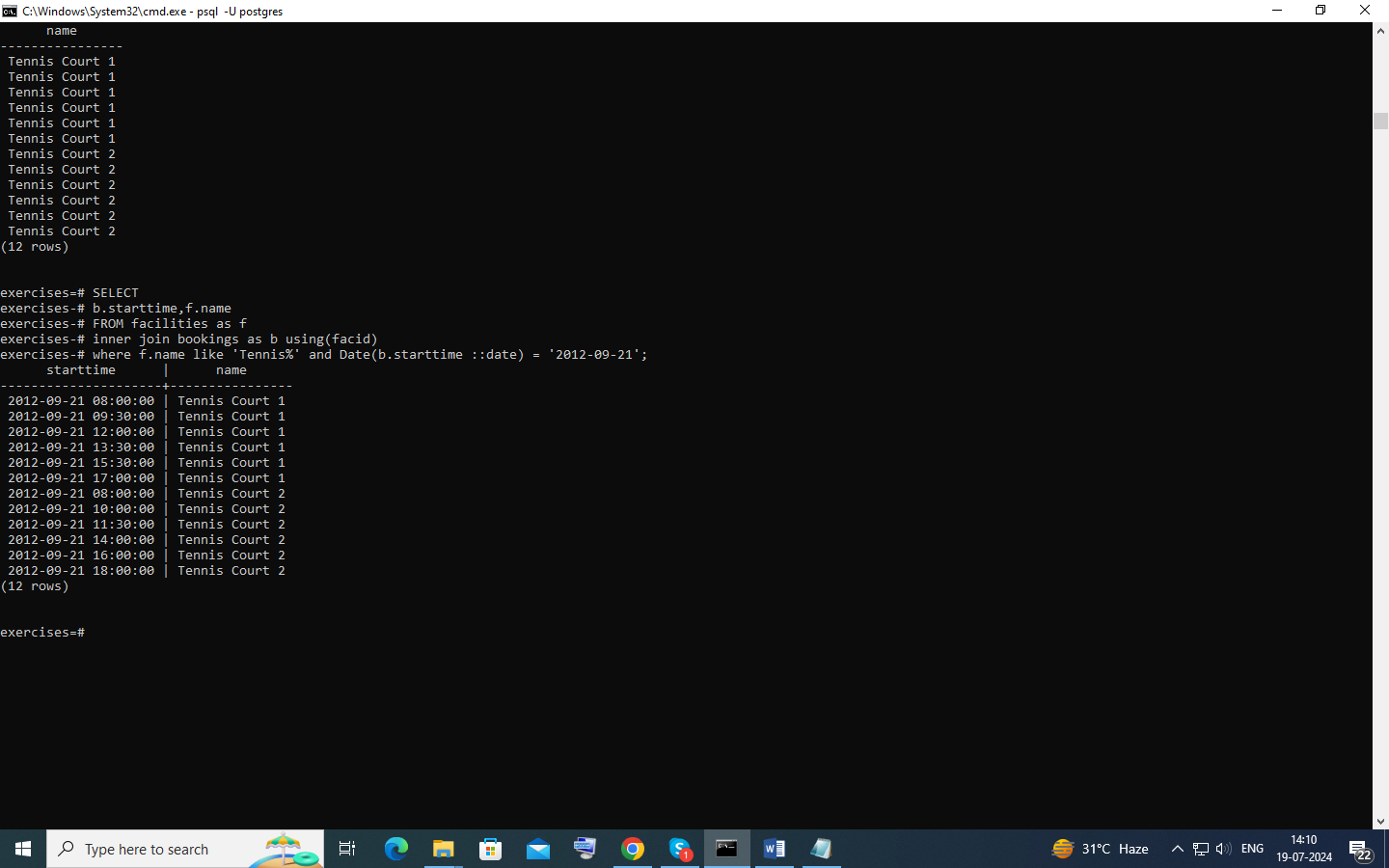
inner join bookings as b using(facid)

where f.name like 'Tennis%' and Date(b.starttime ::date) = '2012-09-21';



# Produce a list of all members who have recommended another member

select firstname,surname from members where memid in (select distinct(recommendedby) from members) order by surname,firstname;

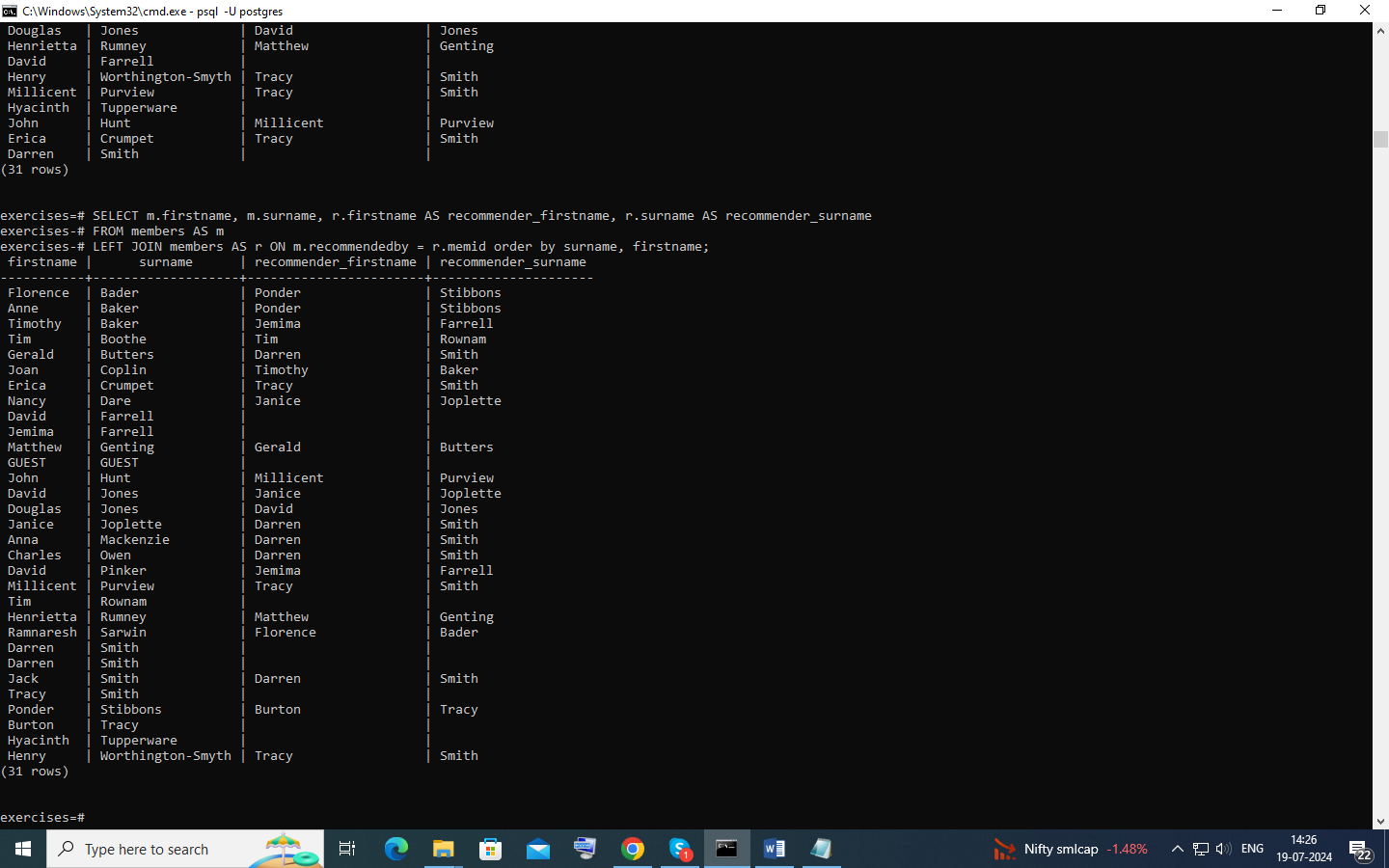


# Produce a list of all members, along with their recommender

SELECT m.firstname, m.surname, r.firstname AS recommender\_firstname, r.surname AS recommender\_surname

FROM members AS m

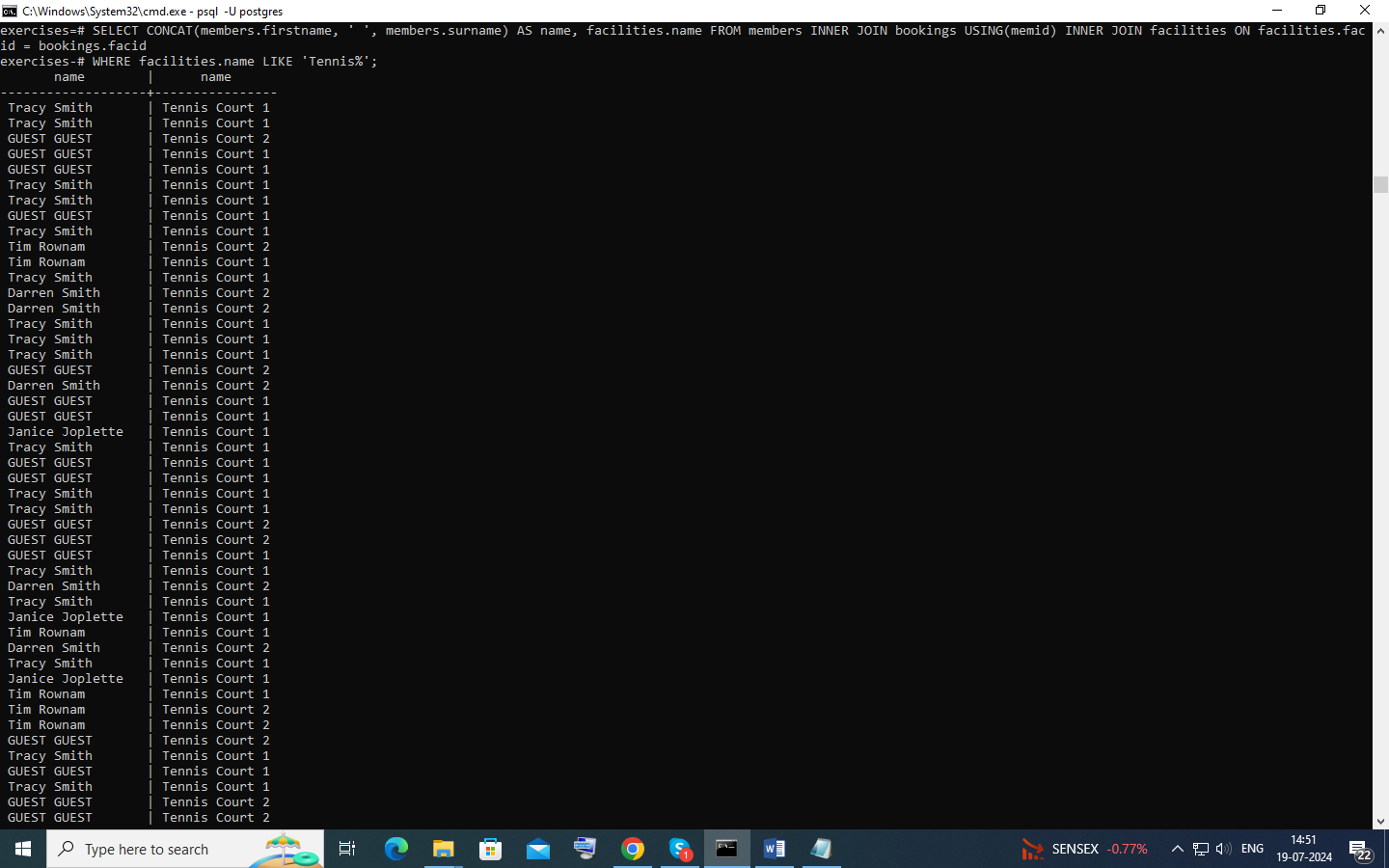
LEFT JOIN members AS r ON m.recommendedby = r.memid order by surname, firstname;



Produce a list of all members who have used a tennis court

SELECT CONCAT(members.firstname, ' ', members.surname) AS name, facilities.name FROM members INNER JOIN bookings USING(memid) INNER JOIN facilities ON facilities.facid = bookings.facid

WHERE facilities.name LIKE 'Tennis%';



# Produce a list of costly bookings

select mems.firstname || ' ' || mems.surname as member,

facs.name as facility,

case

when mems.memid = 0 then

bks.slots\*facs.guestcost

else

bks.slots\*facs.membercost

end as cost

from

cd.members mems

inner join cd.bookings bks

on mems.memid = bks.memid

inner join cd.facilities facs

on bks.facid = facs.facid

where

bks.starttime >= '2012-09-14' and

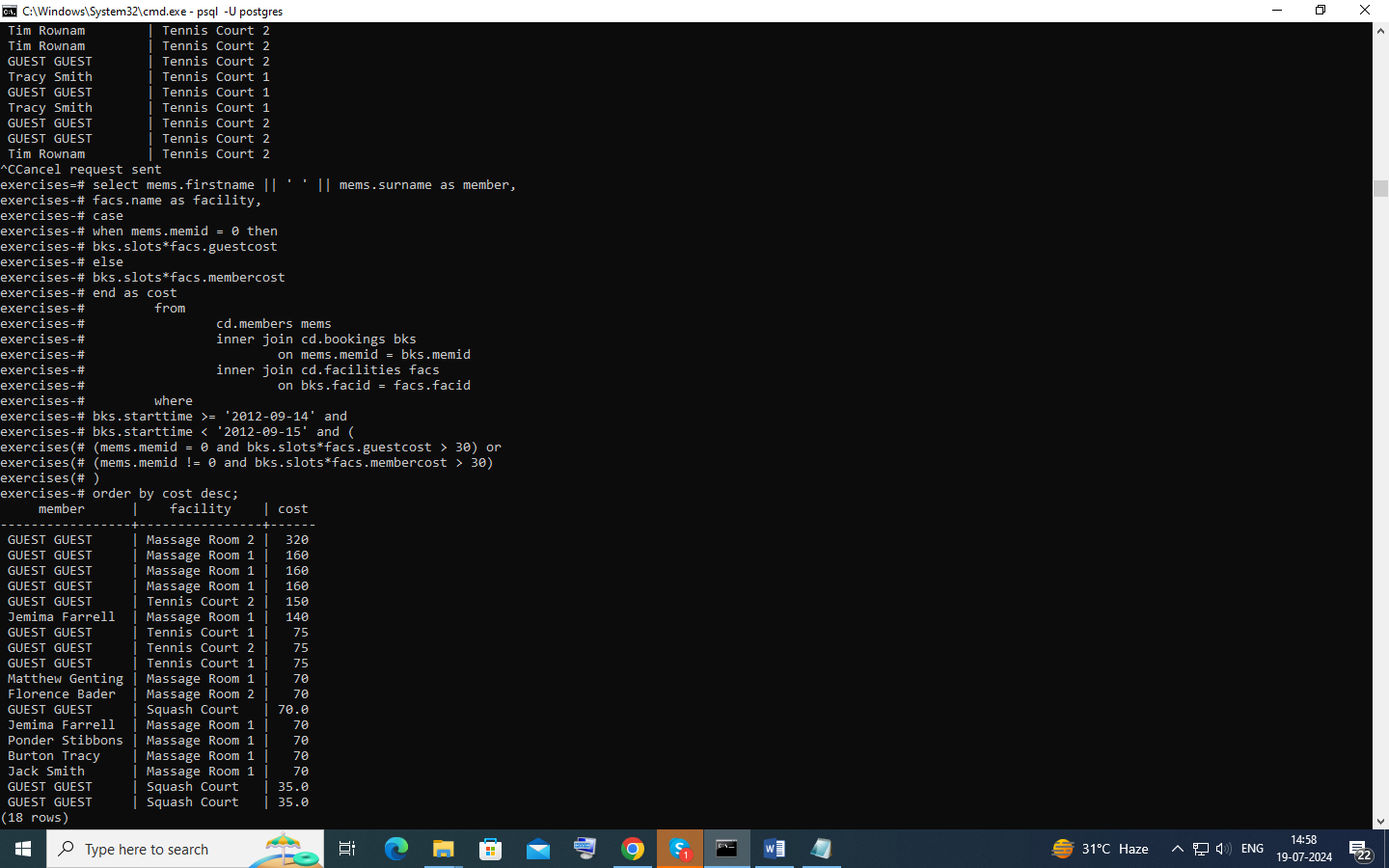
bks.starttime < '2012-09-15' and (

(mems.memid = 0 and bks.slots\*facs.guestcost > 30) or

(mems.memid != 0 and bks.slots\*facs.membercost > 30)

)

order by cost desc;



Produce a list of all members, along with their recommender, using no joins.

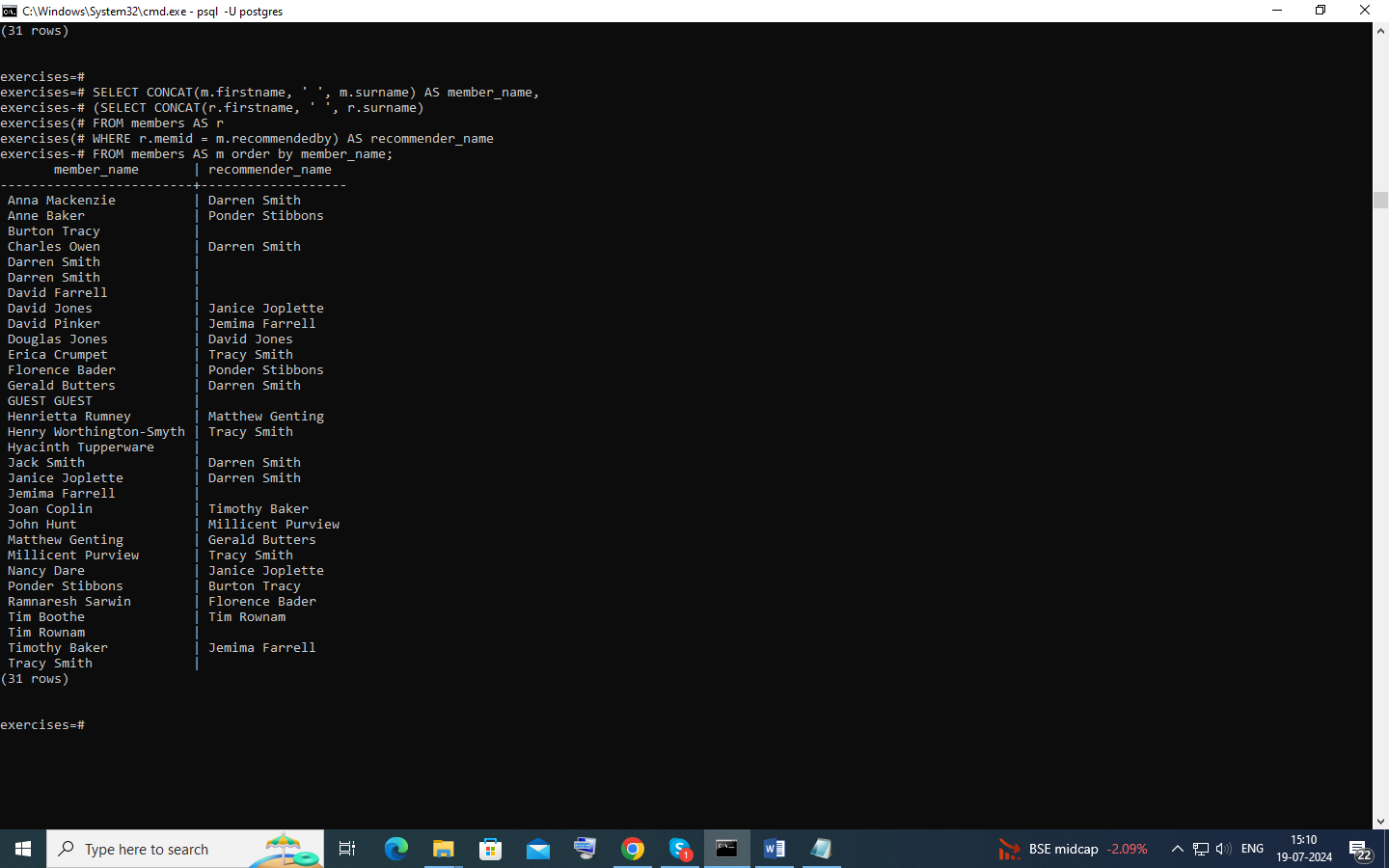
SELECT CONCAT(m.firstname, ' ', m.surname) AS member\_name,

(SELECT CONCAT(r.firstname, ' ', r.surname)

FROM members AS r

WHERE r.memid = m.recommendedby) AS recommender\_name

FROM members AS m order by member\_name;



# Produce a list of costly bookings, using a subquery

select member, facility, cost from (

select

mems.firstname || ' ' || mems.surname as member,

facs.name as facility,

case

when mems.memid = 0 then

bks.slots\*facs.guestcost

else

bks.slots\*facs.membercost

end as cost

from

cd.members mems

inner join cd.bookings bks

on mems.memid = bks.memid

inner join cd.facilities facs

on bks.facid = facs.facid

where

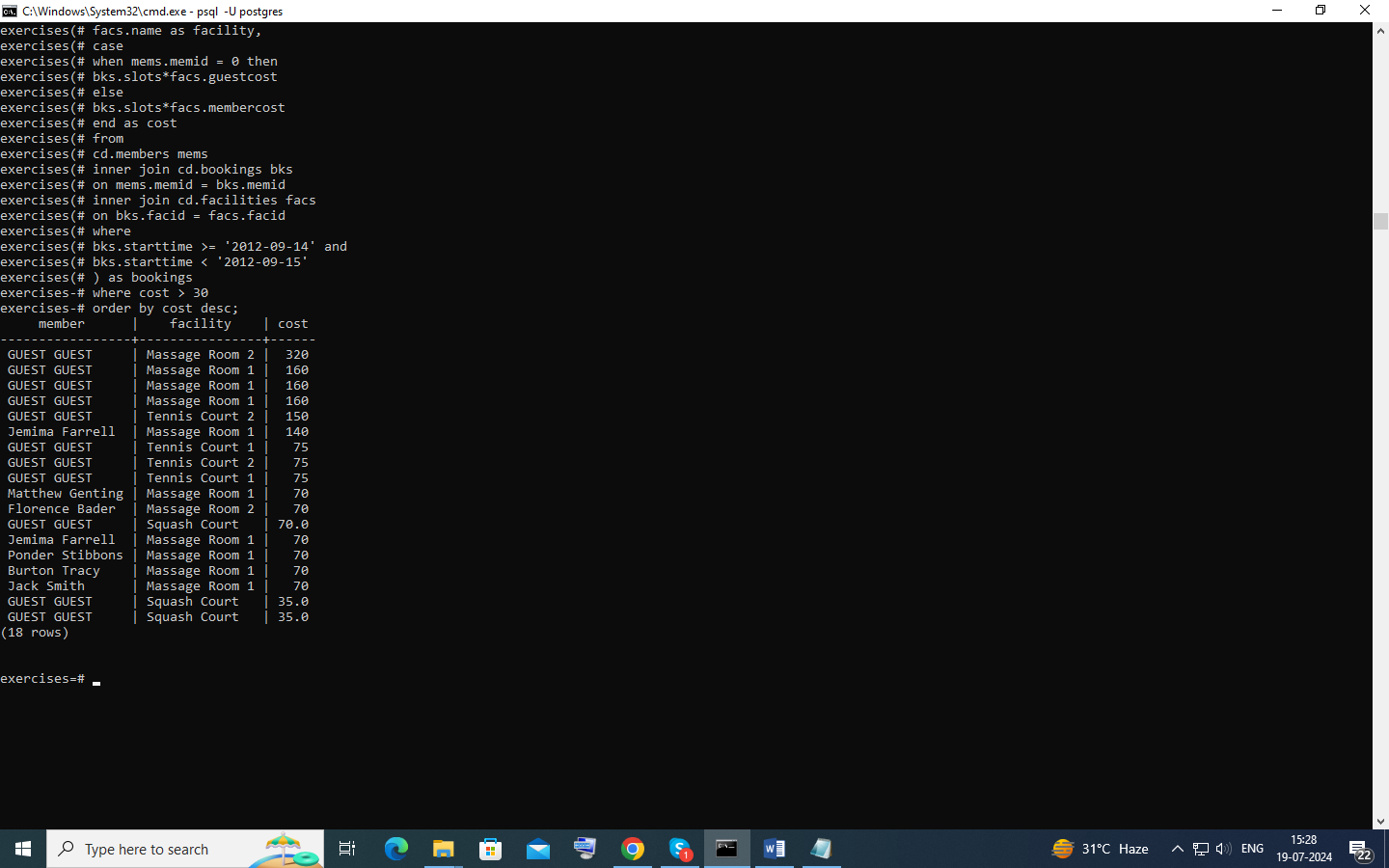
bks.starttime >= '2012-09-14' and

bks.starttime < '2012-09-15'

) as bookings

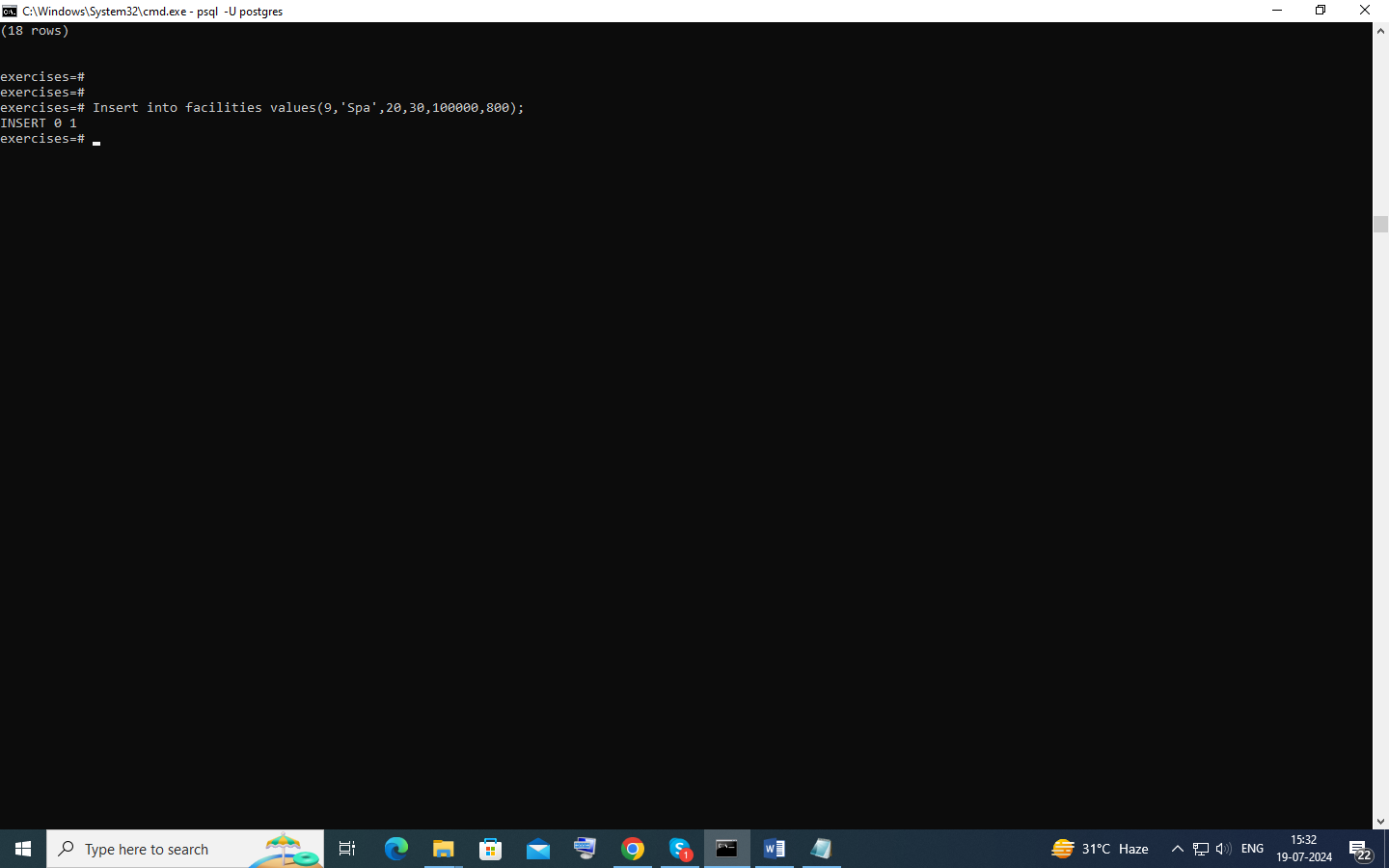
where cost > 30

order by cost desc;



# Insert some data into a table

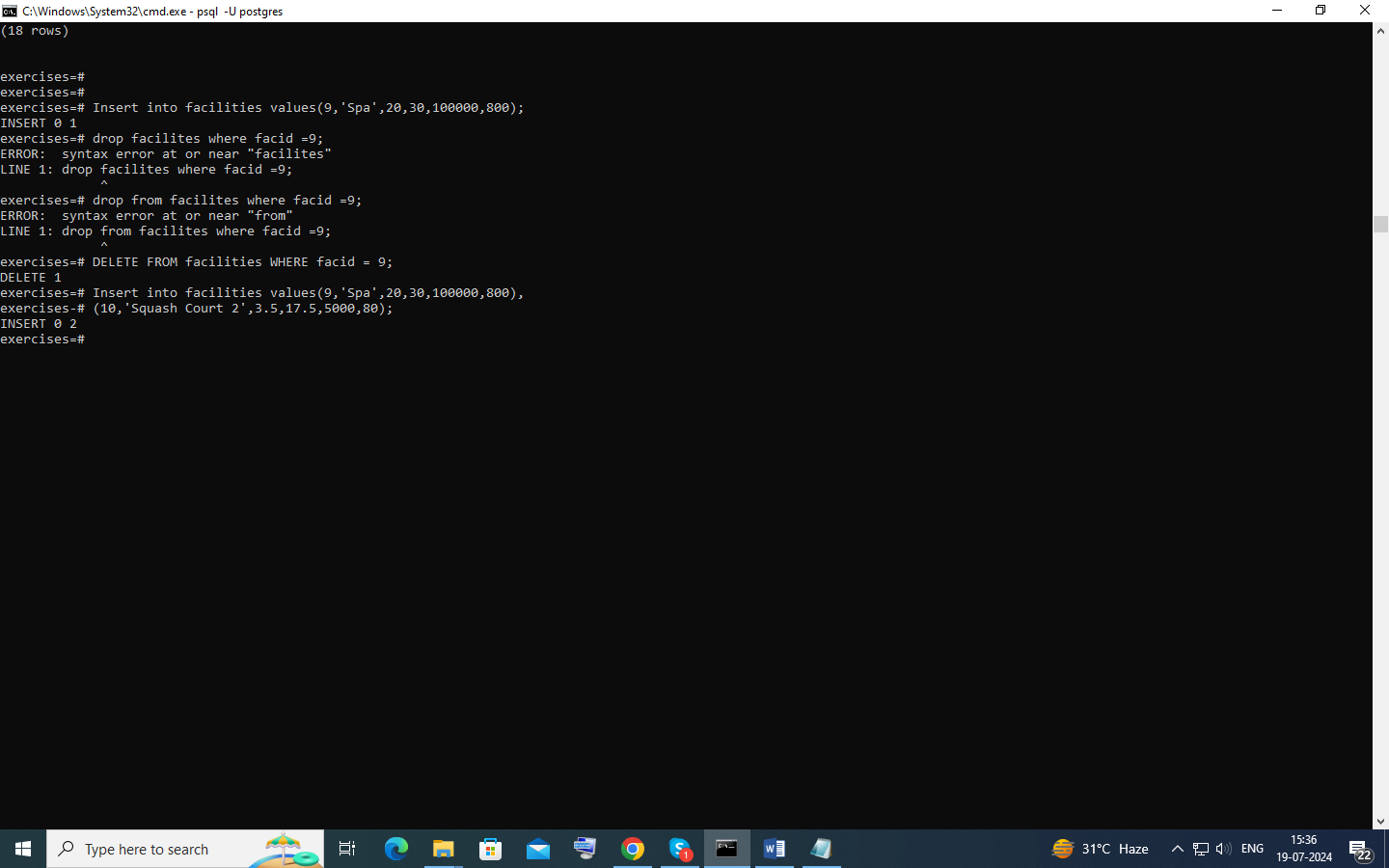
Insert into facilities values(9,'Spa',20,30,100000,800);



# Insert multiple rows of data into a table

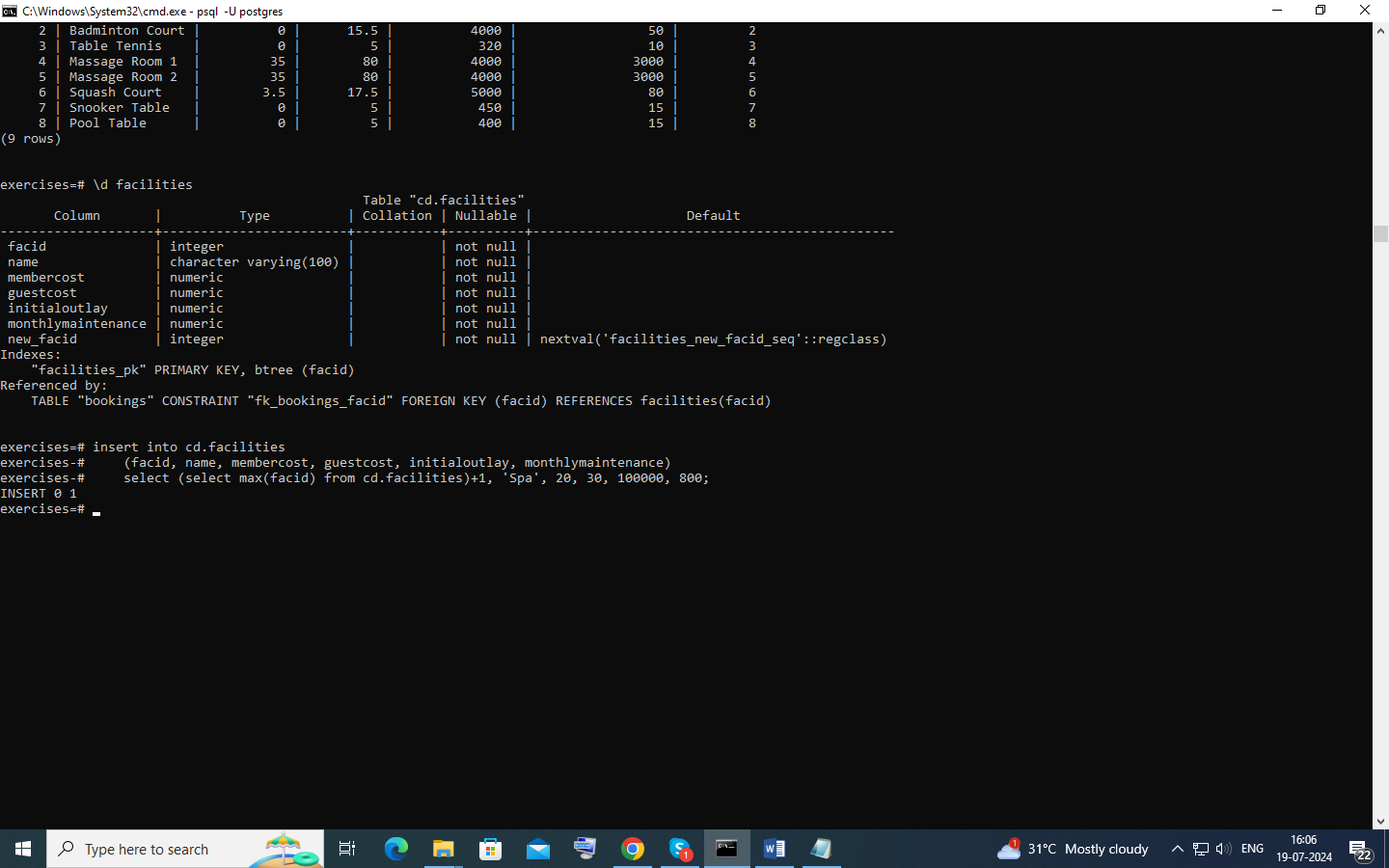
Insert into facilities values(9,'Spa',20,30,100000,800),

(10,'Squash Court 2',3.5,17.5,5000,80);



# Insert calculated data into a table

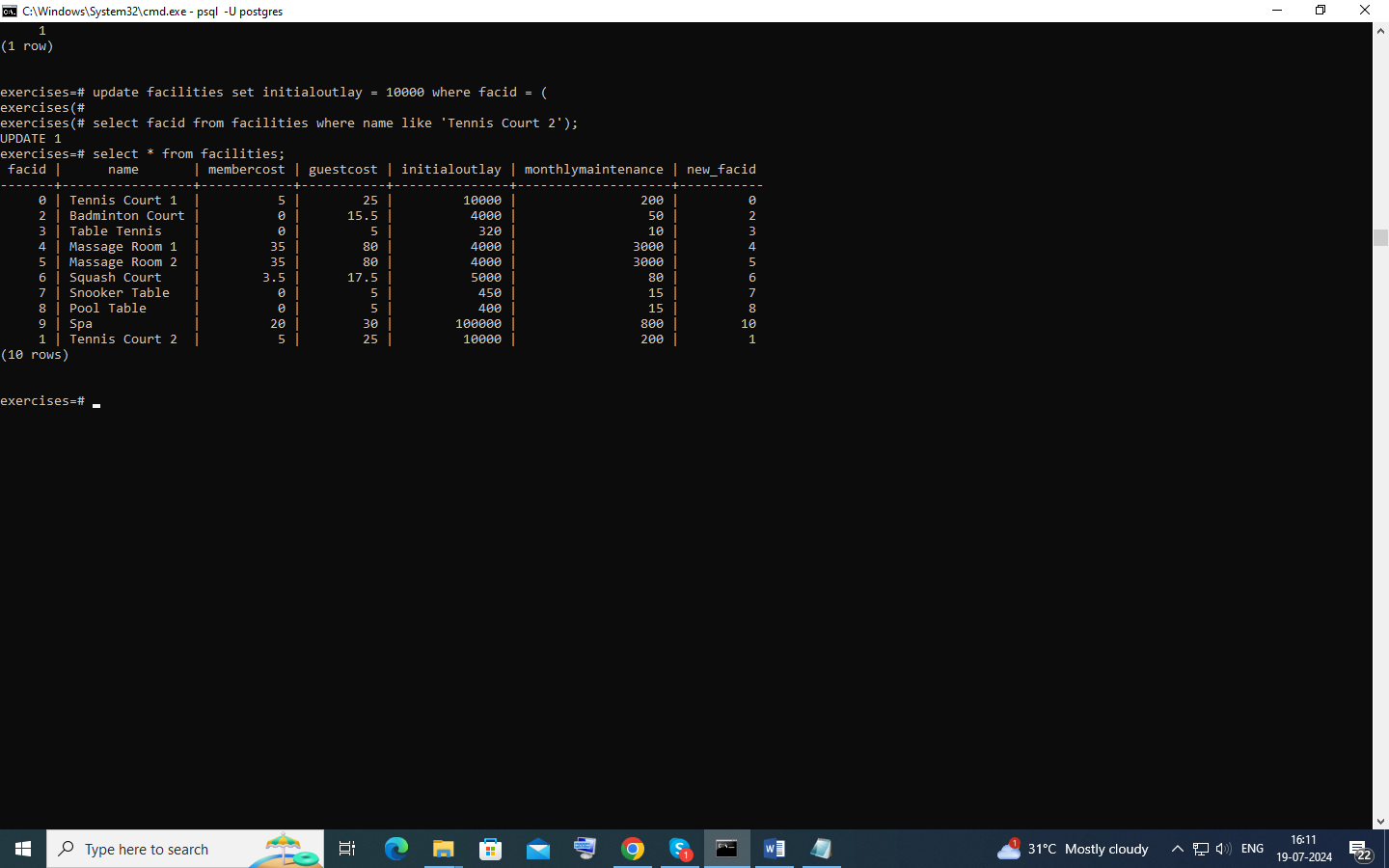
insert into cd.facilities (facid, name, membercost, guestcost, initialoutlay, monthlymaintenance) select (select max(facid) from cd.facilities)+1, 'Spa', 20, 30, 100000, 800;



# Update some existing data

update facilities set initialoutlay = 10000 where facid = (

select facid from facilities where name like 'Tennis Court 2');



# Update multiple rows and columns at the same time

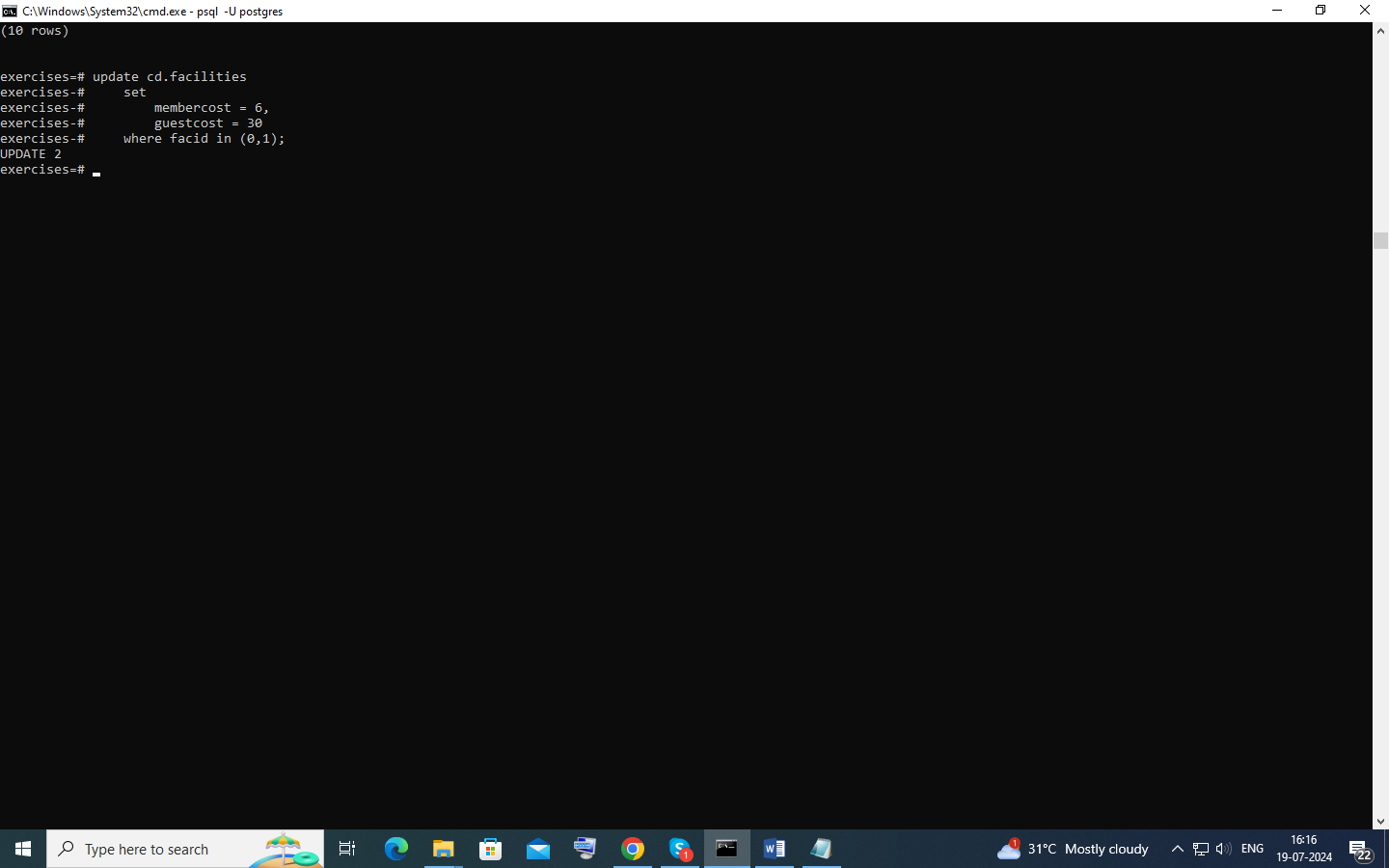
update cd.facilities

set

membercost = 6,

guestcost = 30

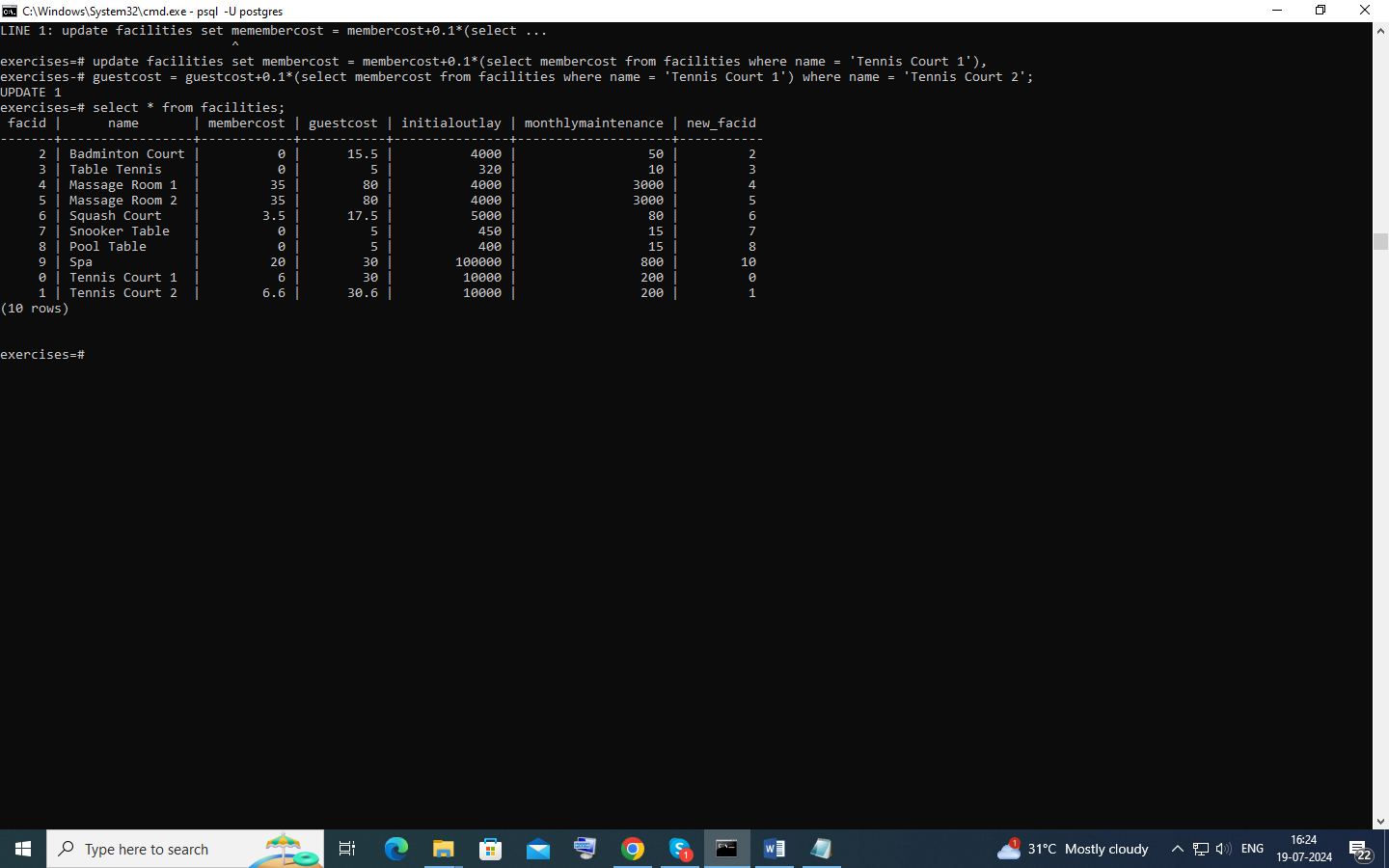
where facid in (0,1);



# Update a row based on the contents of another row

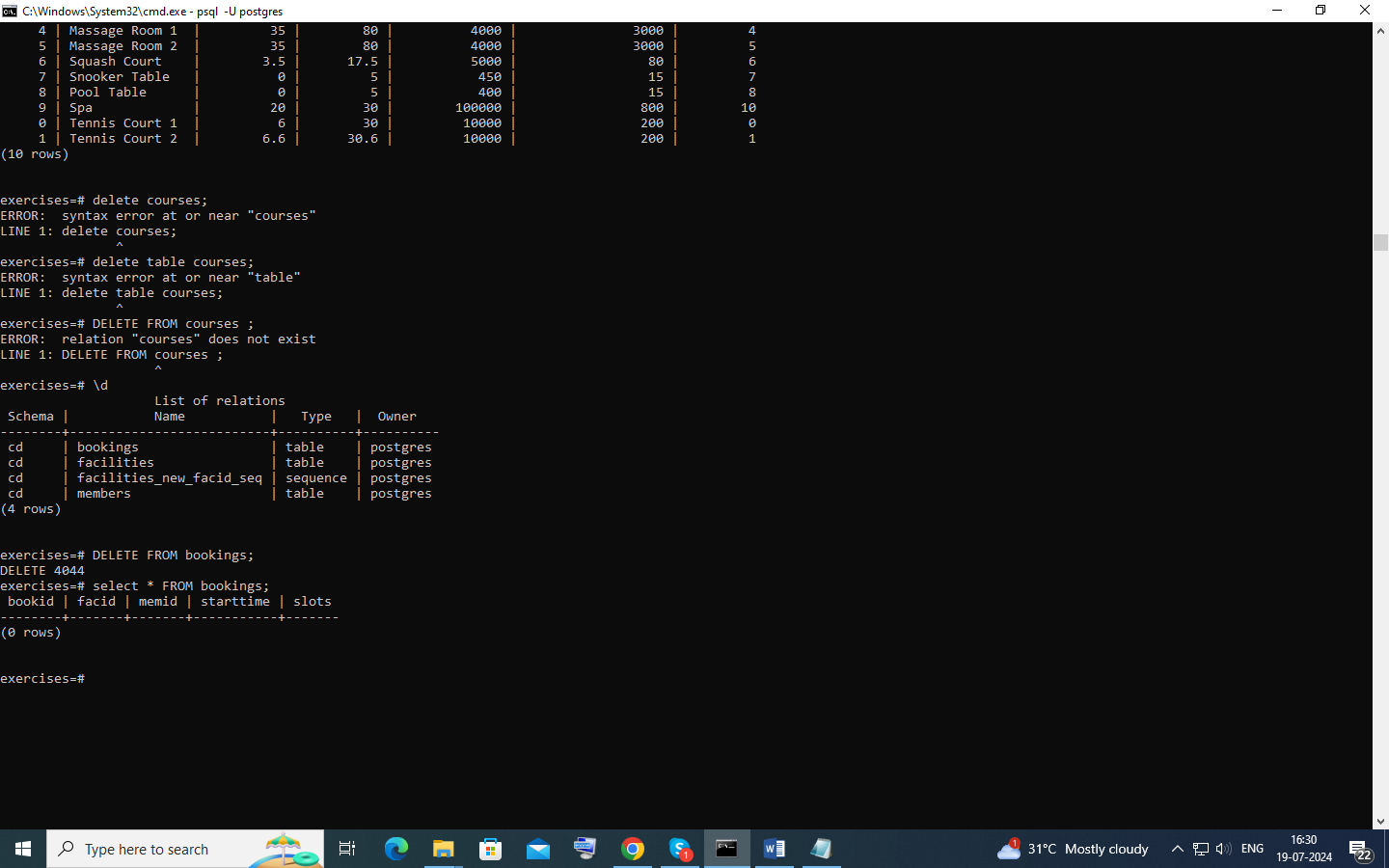
update facilities set membercost = membercost+0.1\*(select membercost from facilities where name = 'Tennis Court 1'),

guestcost = guestcost+0.1\*(select membercost from facilities where name = 'Tennis Court 1') where name = 'Tennis Court 2';



# Delete all bookings

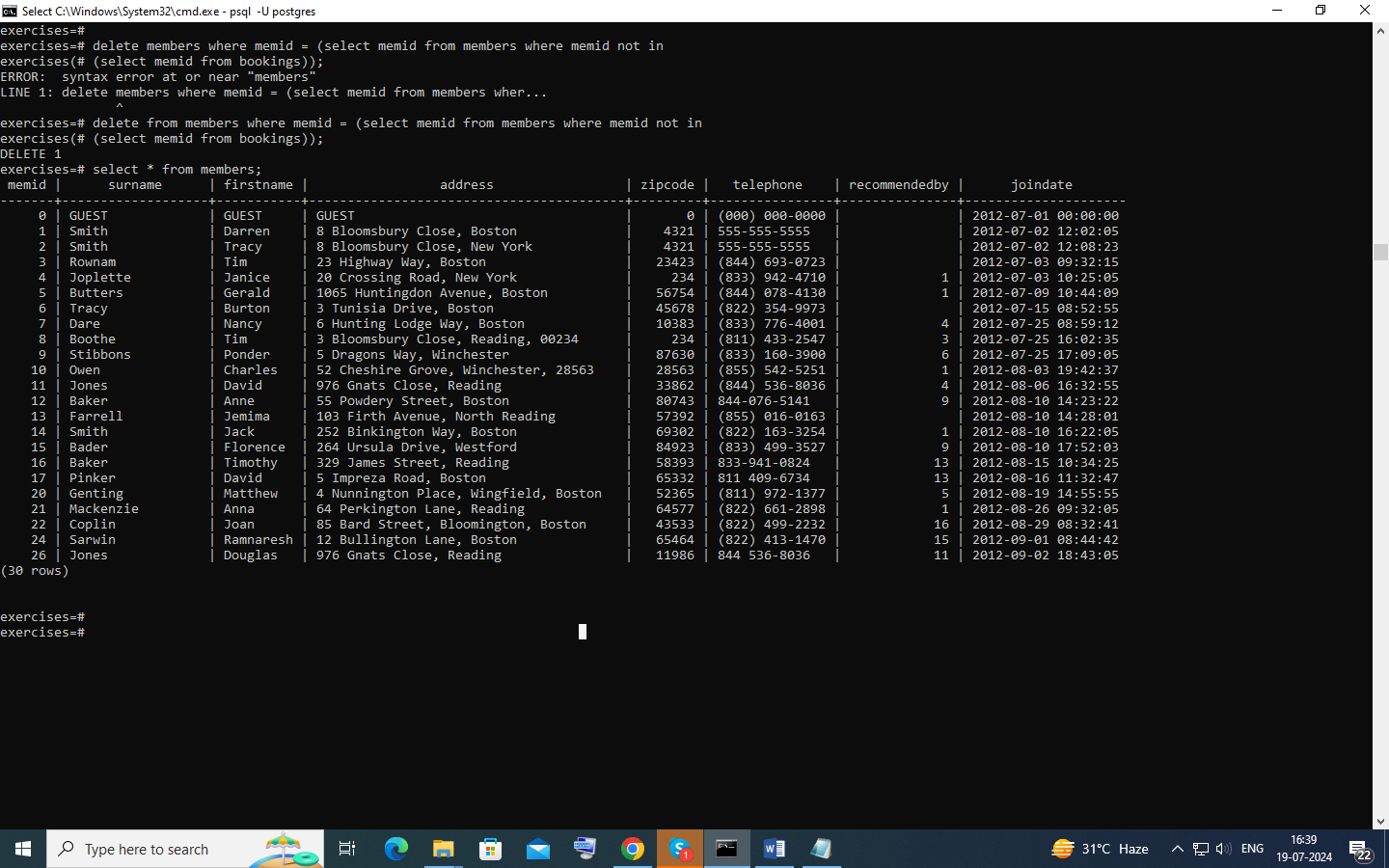
DELETE FROM bookings;



# Delete a member from the cd.members table

delete from members where memid = (select memid from members where memid not in

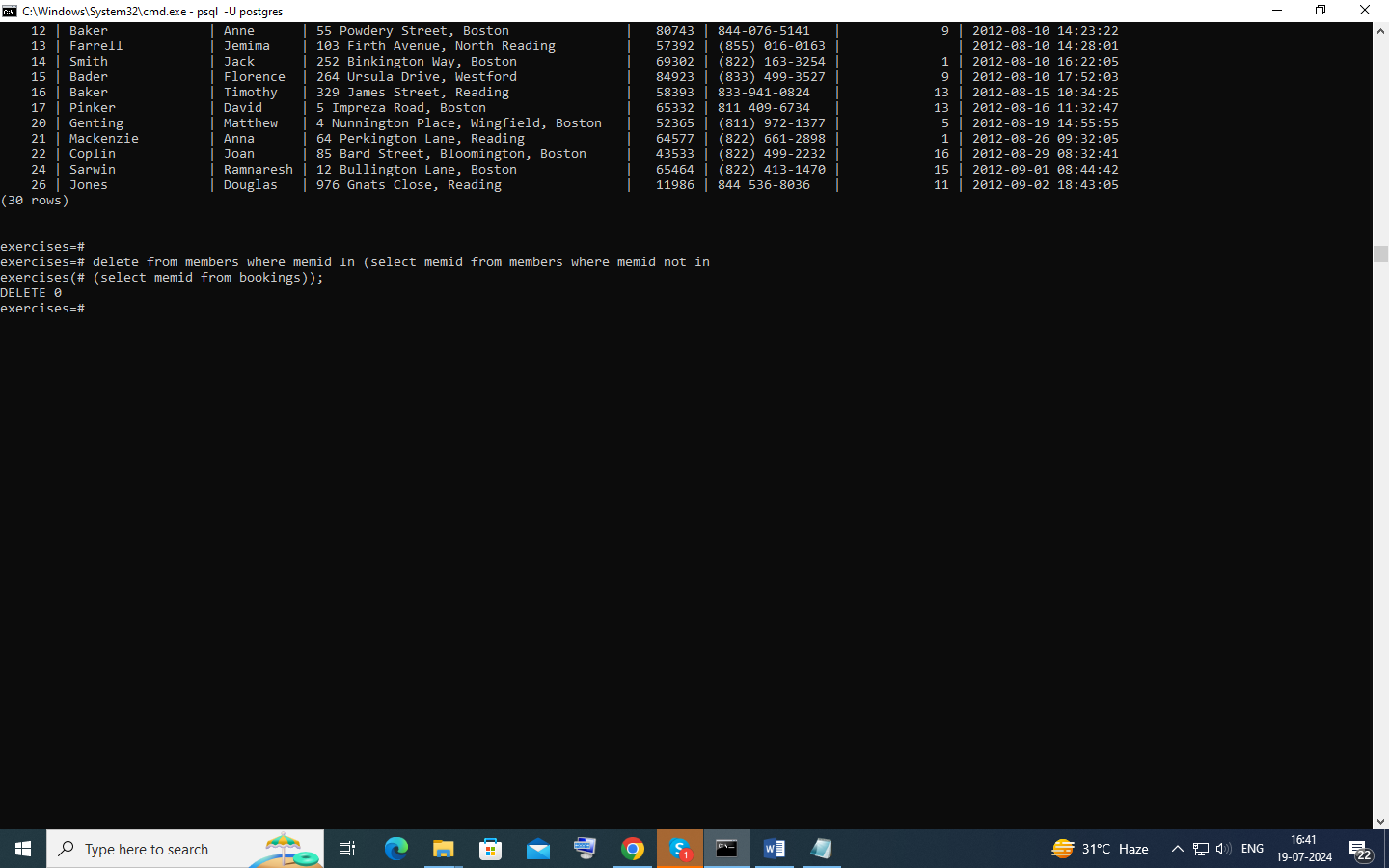
(select memid from bookings));



# Delete based on a subquery

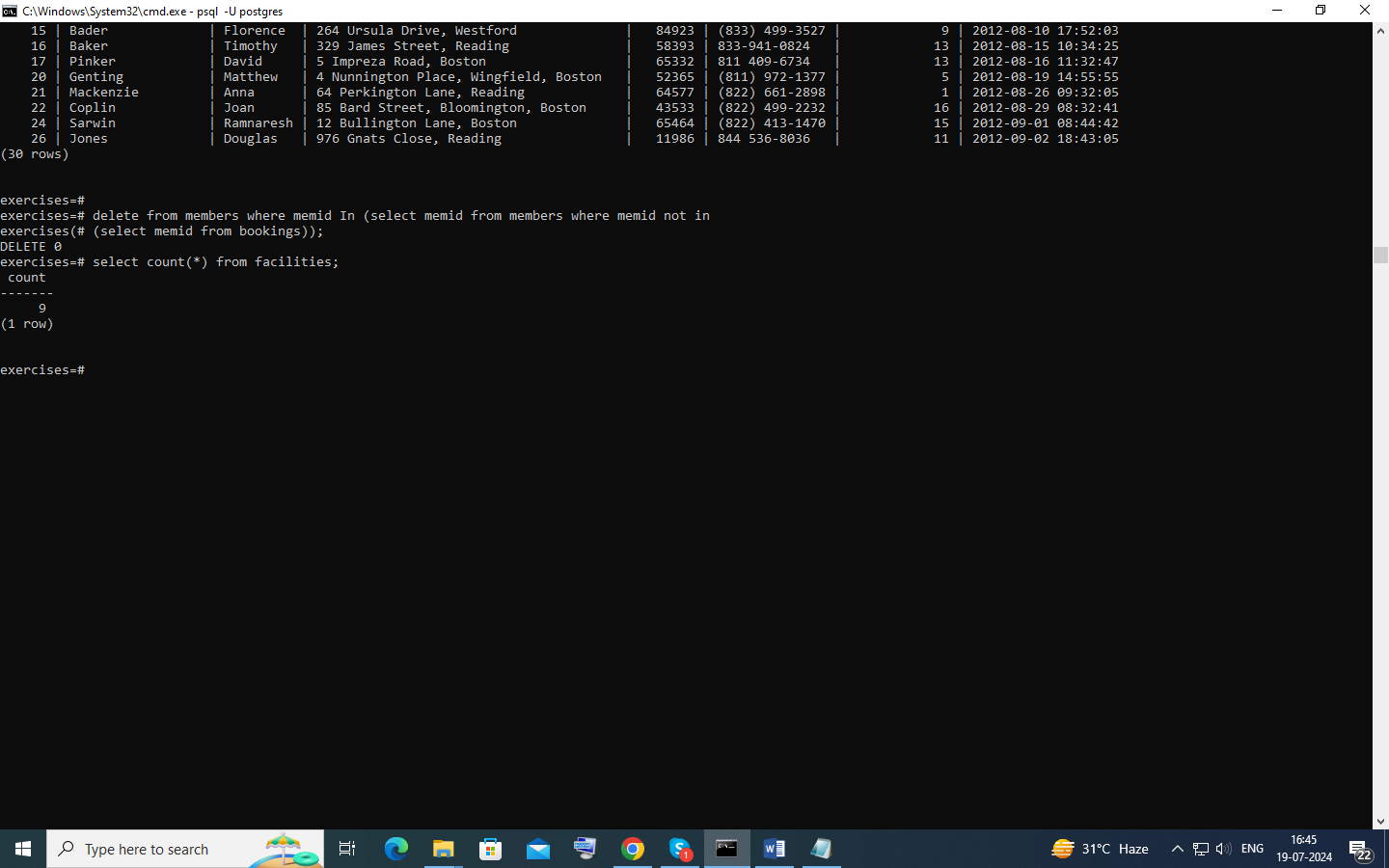
delete from members where memid In (select memid from members where memid not in

(select memid from bookings));



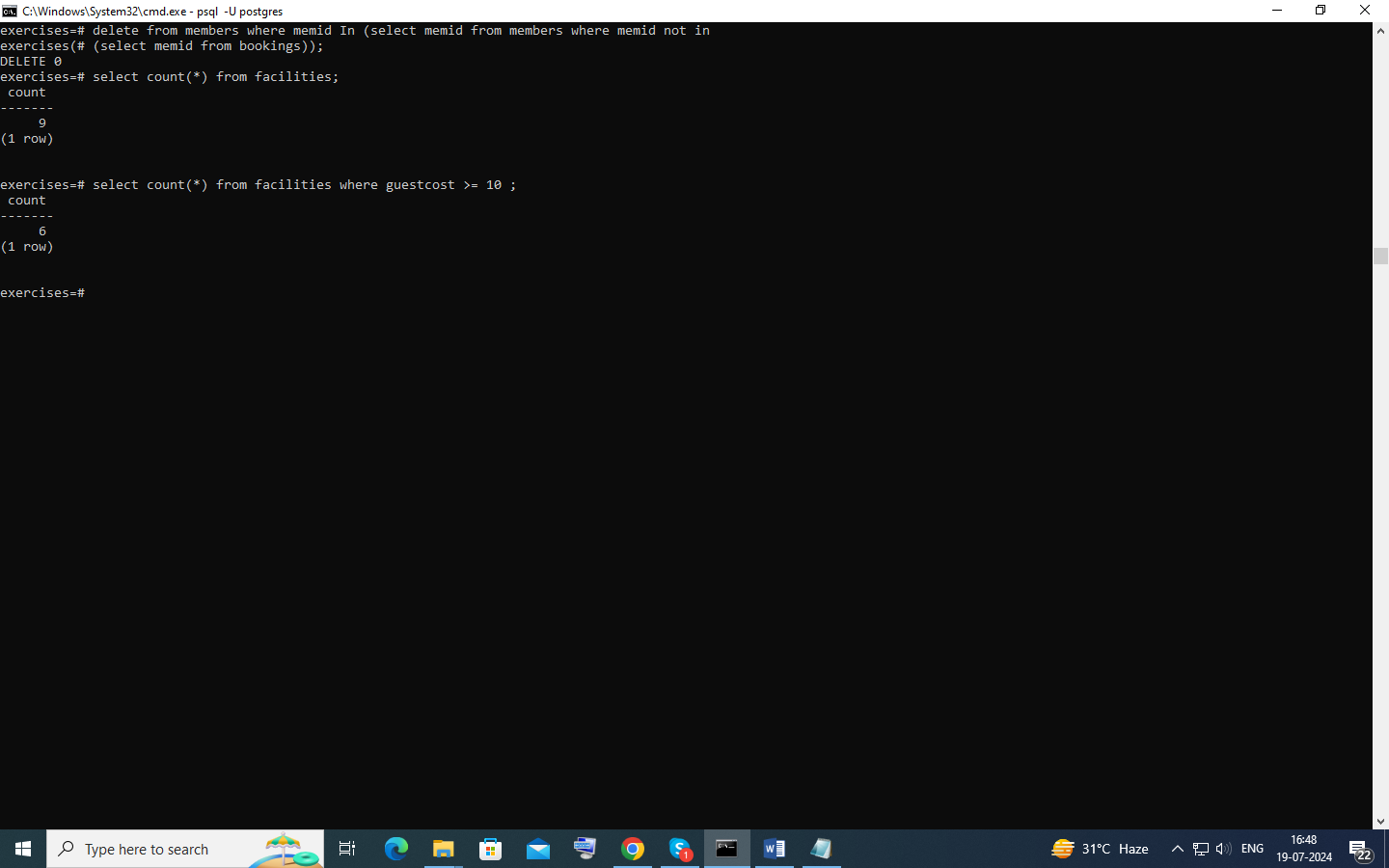
# Count the number of facilities

select count(\*) from facilities;



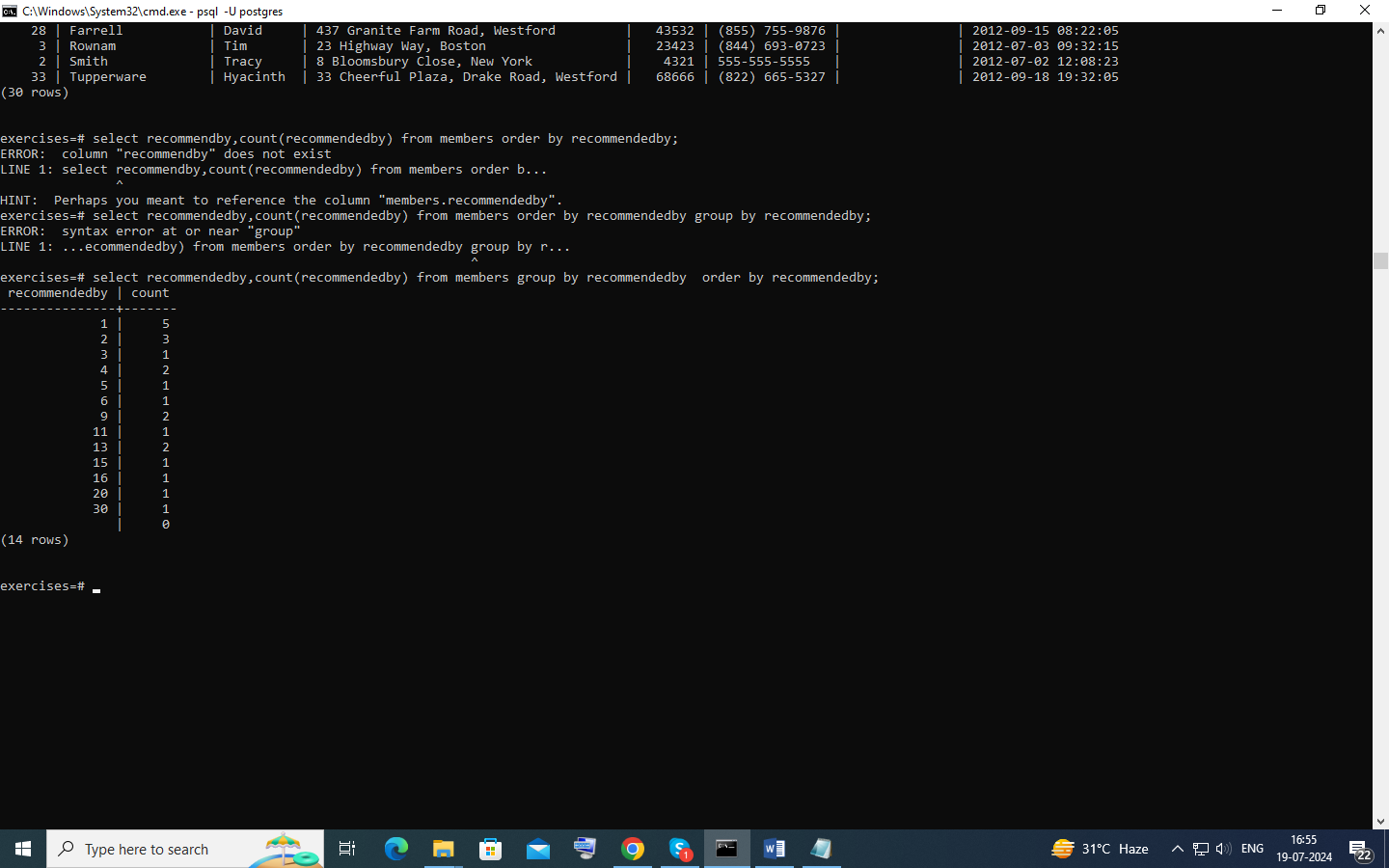
# Count the number of expensive facilities

select count(\*) from facilities where guestcost >= 10 ;



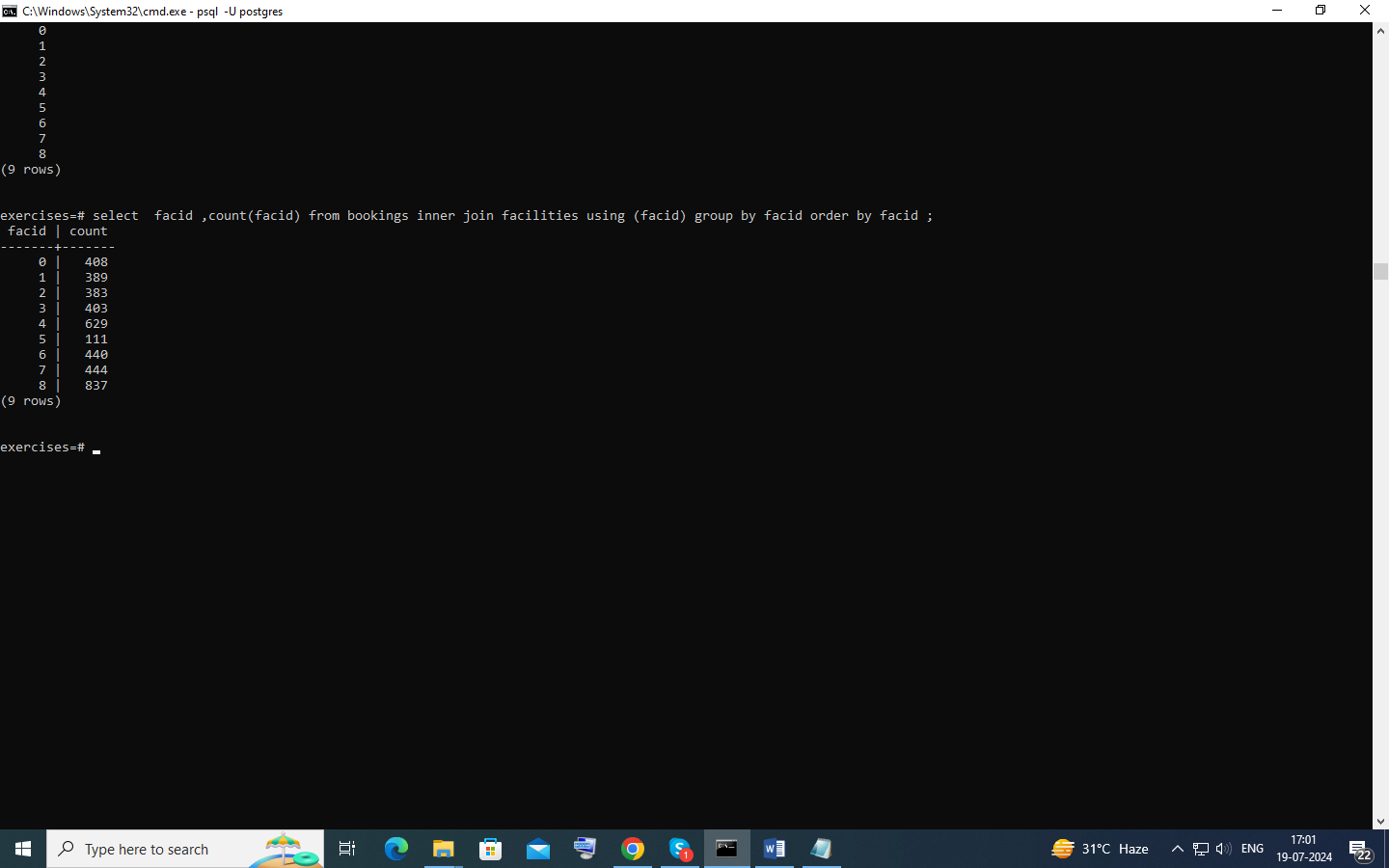
select recommendedby,count(recommendedby) from members group by recommendedby order by memid;

# Count the number of recommendations each member makes.



# List the total slots booked per facility

select facid ,count(facid) from bookings inner join facilities using (facid) group by facid order by facid ;



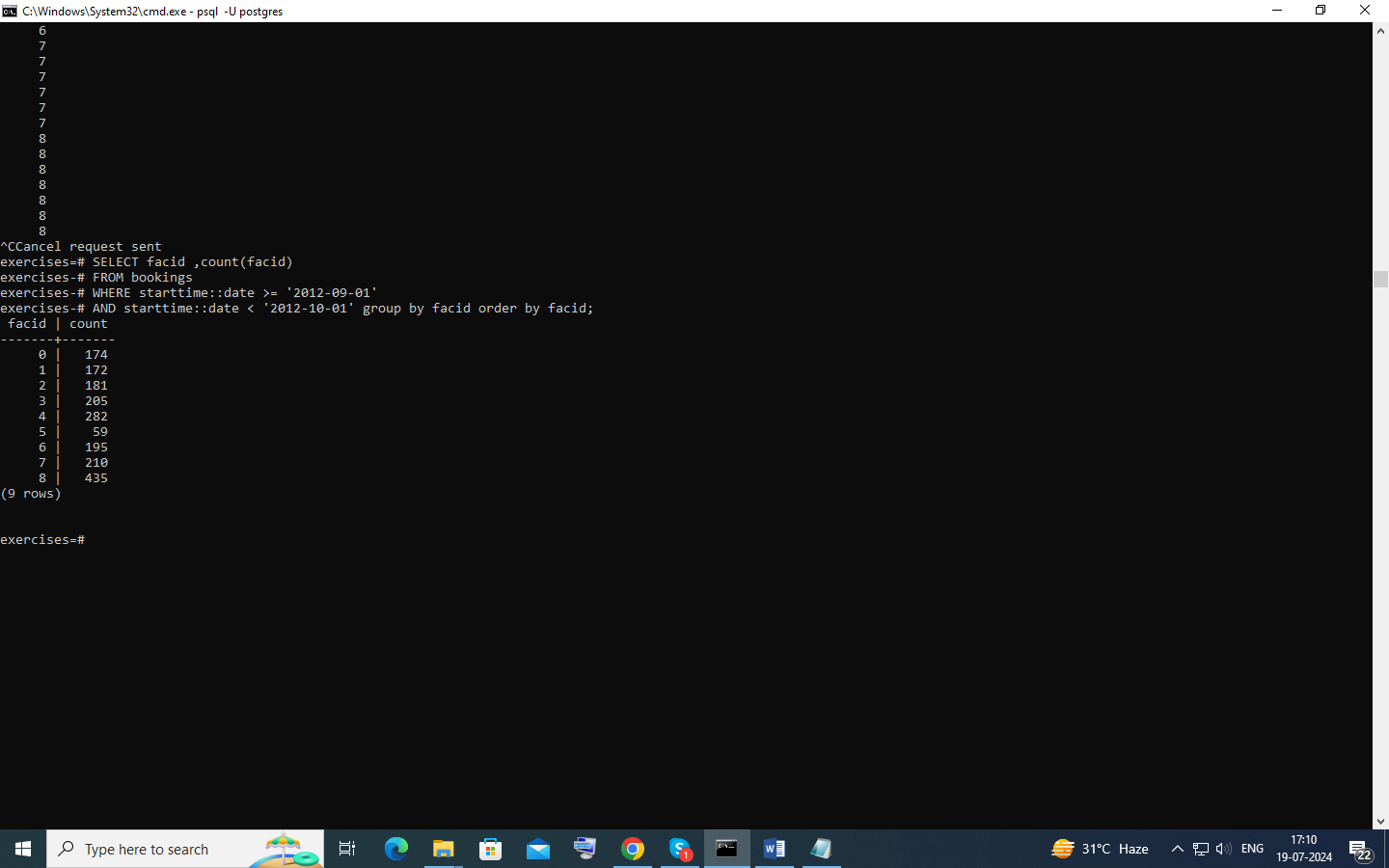
List the total slots booked per facility in a given month

SELECT facid ,count(facid)

FROM bookings

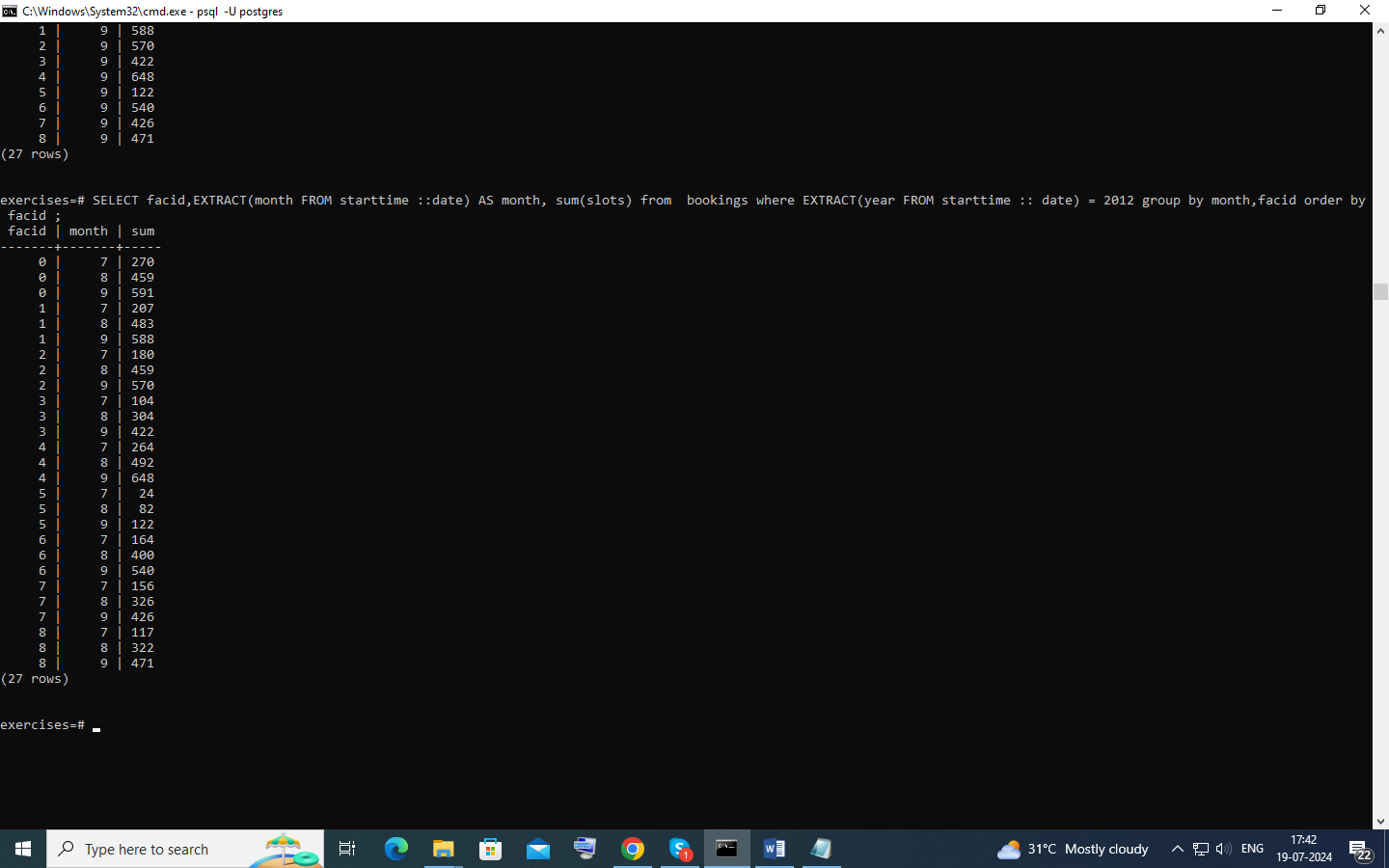
WHERE starttime::date >= '2012-09-01'

AND starttime::date < '2012-10-01' group by facid order by facid;



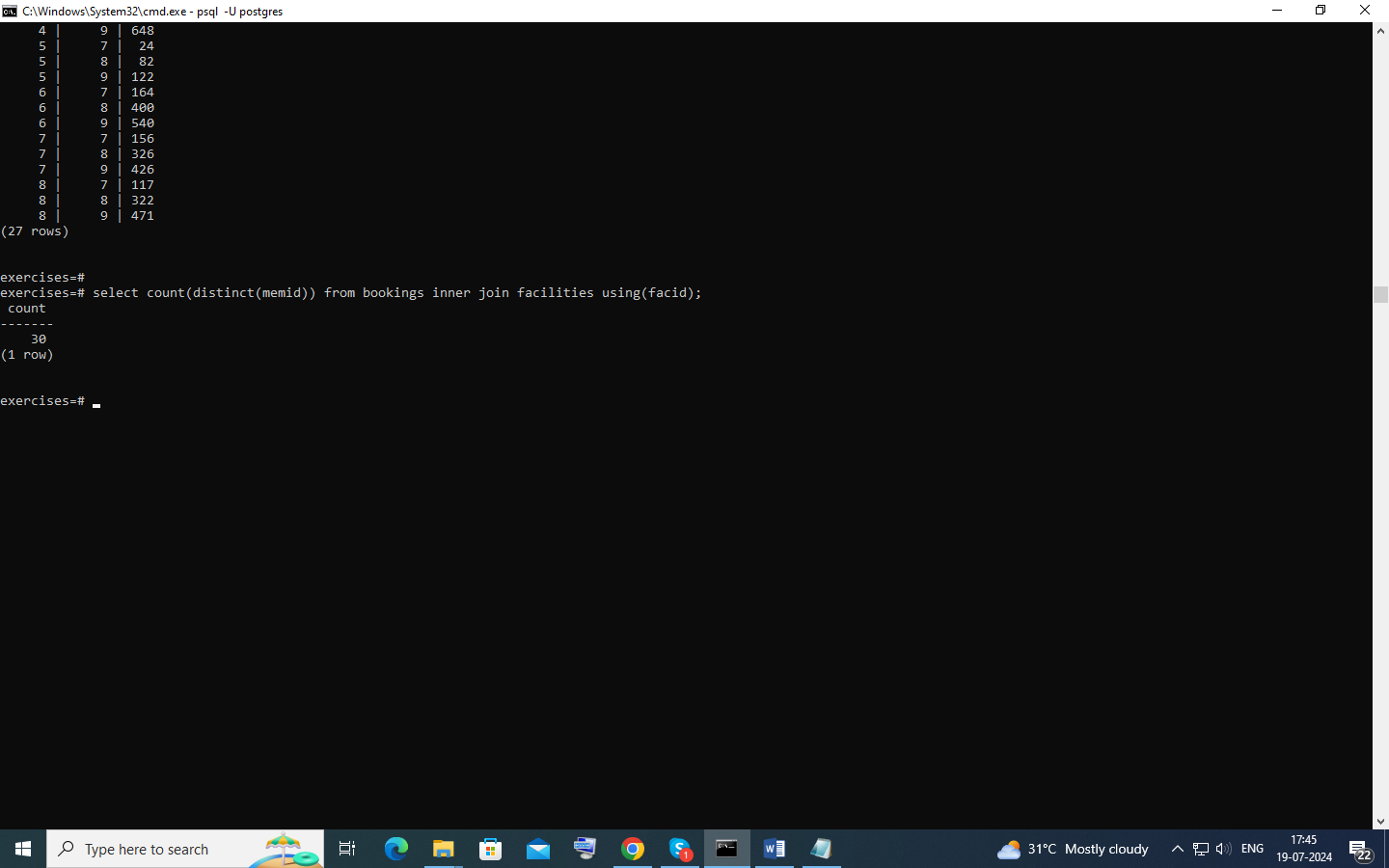
# List the total slots booked per facility per month

SELECT facid,EXTRACT(month FROM starttime ::date) AS month, sum(slots) from bookings where EXTRACT(year FROM starttime :: date) = 2012 group by month,facid order by facid ;



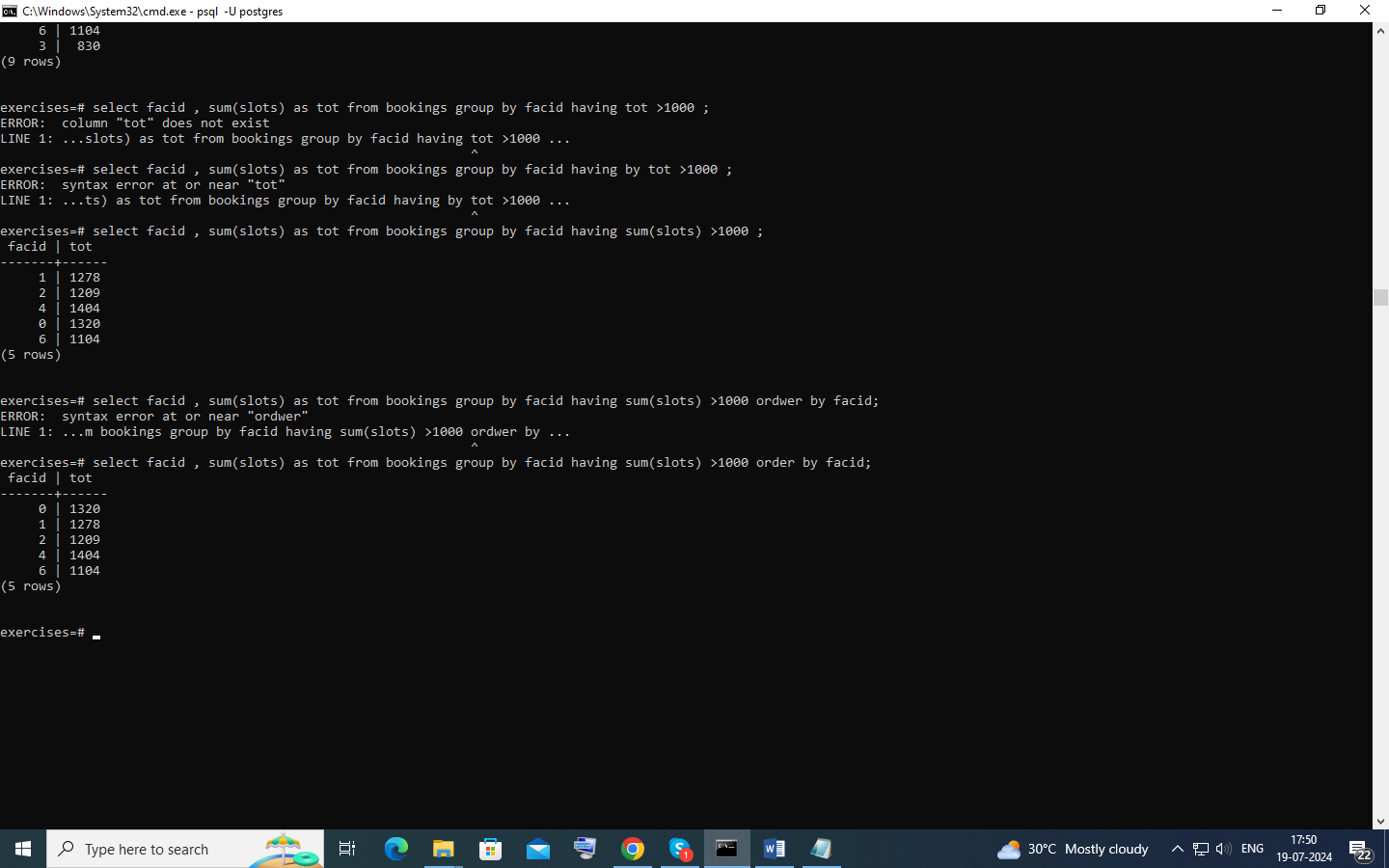
Find the count of members who have made at least one booking

select count(distinct(memid)) from bookings inner join facilities using(facid);



# List facilities with more than 1000 slots booked

select facid , sum(slots) as tot from bookings group by facid having sum(slots) >1000 order by facid;



# Find the total revenue of each facility

select name,sum( slots \*

case

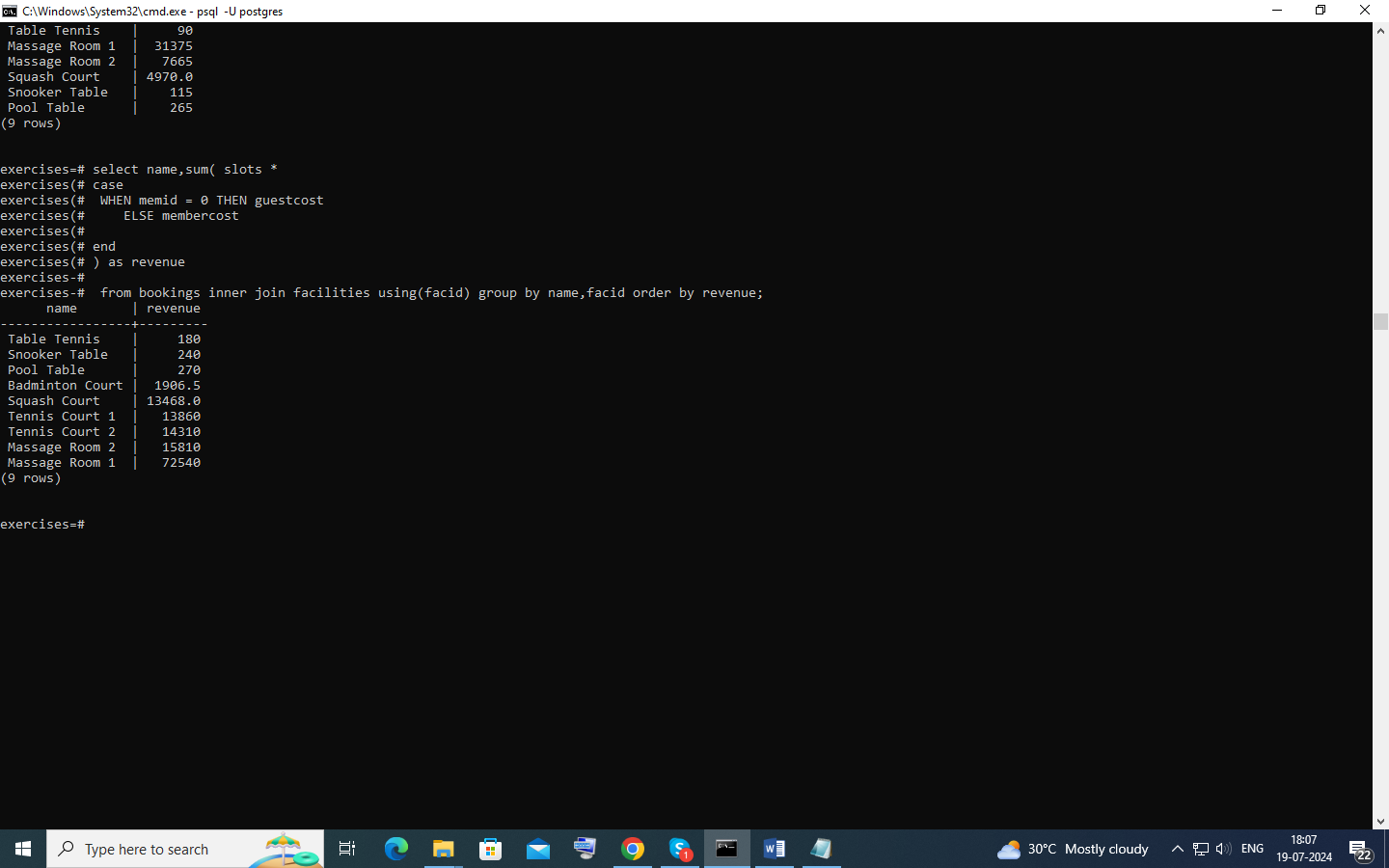
WHEN memid = 0 THEN guestcost

ELSE membercost

end

) as revenue

from bookings inner join facilities using(facid) group by name,facid order by revenue;



# Find facilities with a total revenue less than 1000

select name,sum( slots \*

case

WHEN memid = 0 THEN guestcost

ELSE membercost

end

) as revenue

from bookings inner join facilities using(facid)

group by name,facid

HAVING SUM(slots \*

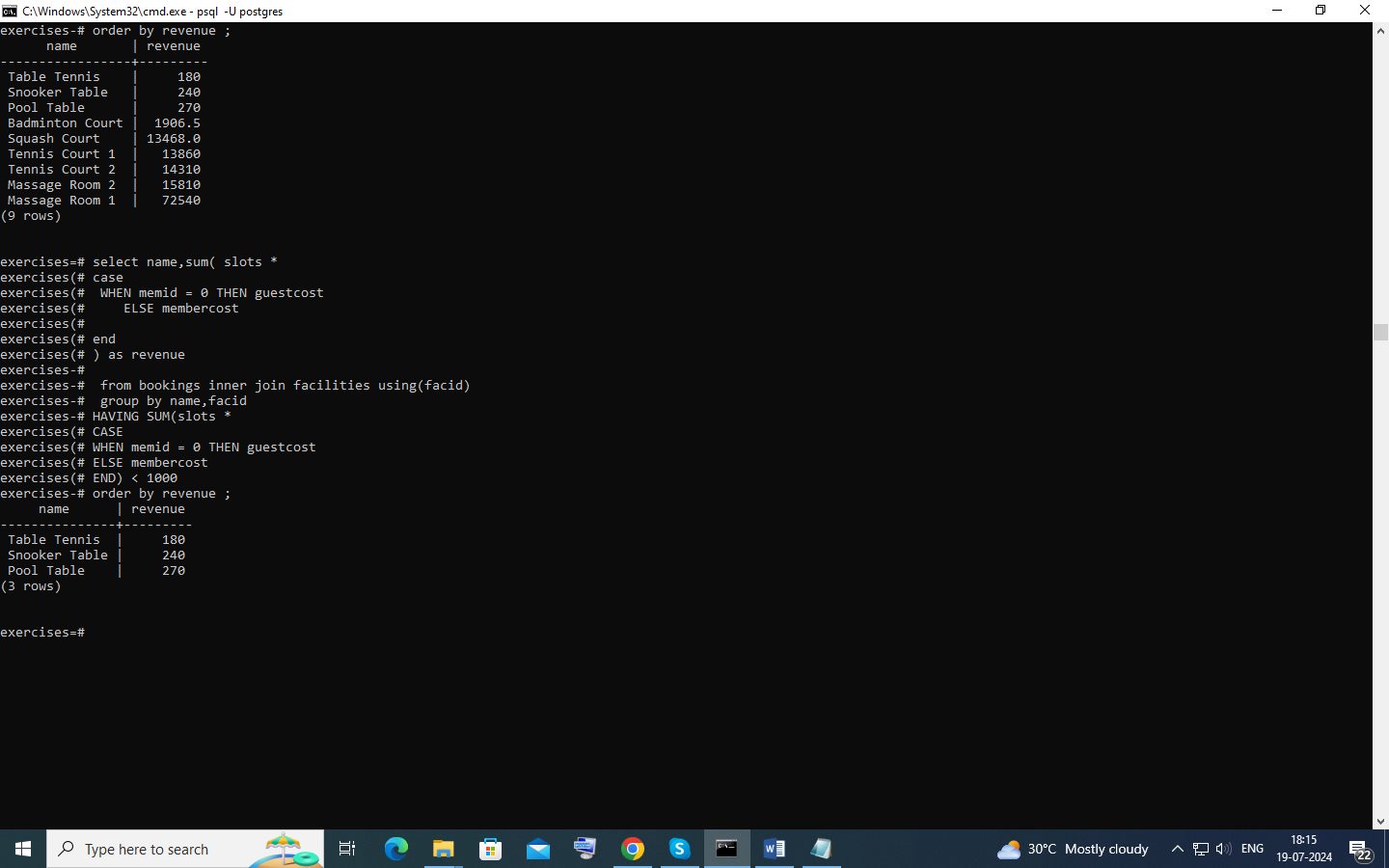
CASE

WHEN memid = 0 THEN guestcost

ELSE membercost

END) < 1000

order by revenue ;



# Output the facility id that has the highest number of slots booked

select facid,sum( slots \*

case

WHEN memid = 0 THEN guestcost

ELSE membercost

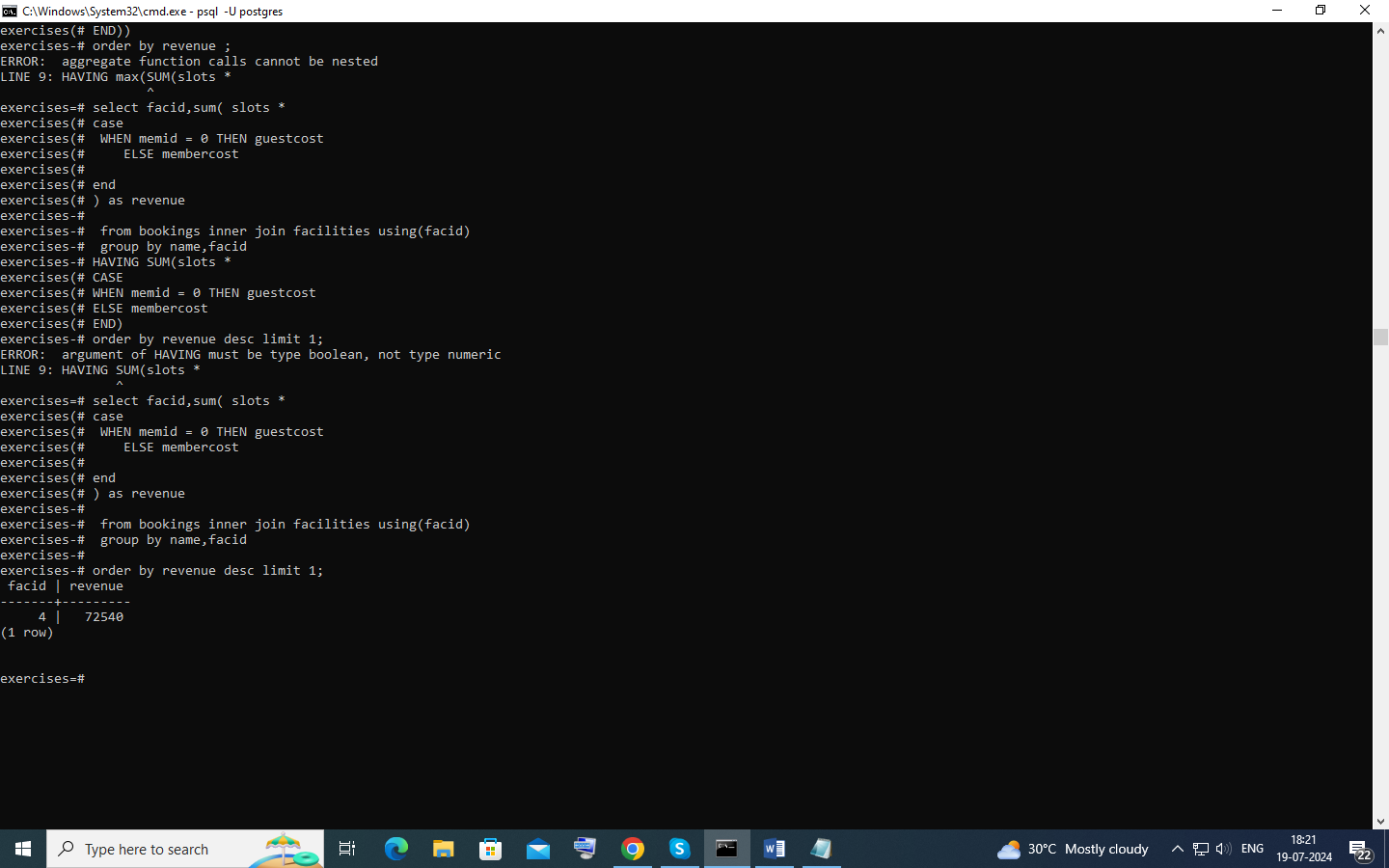
end

) as revenue

from bookings inner join facilities using(facid)

group by name,facid

order by revenue desc limit 1;



# List the total slots booked per facility per month, part 2

select facid ,extract(month FROM starttime::date) as month ,sum(slots) from bookings where starttime >= '2012-01-01'

and starttime < '2013-01-01'

group by rollup(facid ,month ) order by facid,month;

