**Name :Basil Ahamed**

Assignment 2 SQL

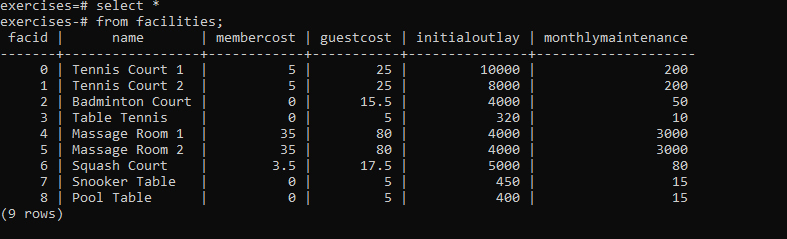
**BASIC**

1. **Retrieve everything from a table:**

Query:

select \*

from facilities;



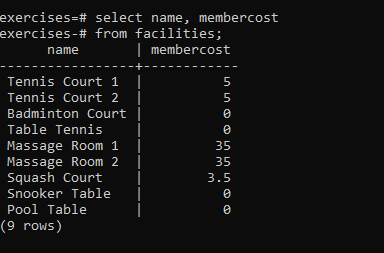
# 2) Retrieve specific columns from a tablehttps://pgexercises.com/assets/tick2.svg

**You want to print out a list of all of the facilities and their cost to members. How would you retrieve a list of only facility names and costs?**

Query:

select name, membercost

from facilities;



**3) Control which rows are retrieved**

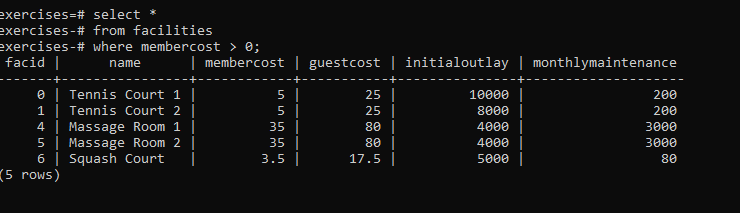
**How can you produce a list of facilities that charge a fee to members?**

Query:

select \*

from facilities

where membercost > 0;



# 4) Control which rows are retrieved - part 2https://pgexercises.com/assets/tick2.svg

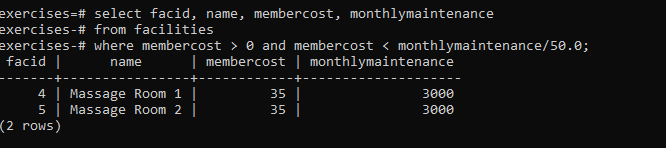
**How can you produce a list of facilities that charge a fee to members, and that fee is less than 1/50th of the monthly maintenance cost? Return the facid, facility name, member cost, and monthly maintenance of the facilities in question.**

**Query:**

select facid, name, membercost, monthlymaintenance

from facilities

where membercost > 0 and membercost < monthlymaintenance/50.0;



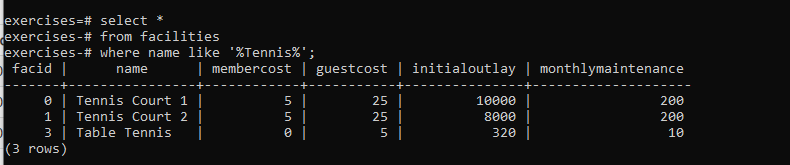
# 5) Basic string searches

**How can you produce a list of all facilities with the word 'Tennis' in their name?**

select \*

from facilities

where name like '%Tennis%';



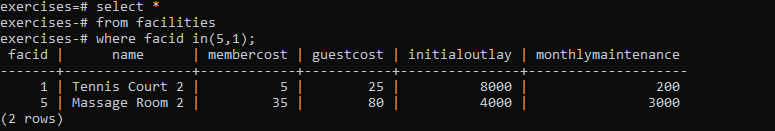
# 6) Matching against multiple possible valueshttps://pgexercises.com/assets/tick2.svg

**How can you retrieve the details of facilities with ID 1 and 5? Try to do it without using the OR operator.**

select \*

from facilities

where facid in(5,1);



# 7) Classify results into bucketshttps://pgexercises.com/assets/tick2.svg

### **Question**

**How can you produce a list of facilities, with each labelled as 'cheap' or 'expensive' depending on if their monthly maintenance cost is more than $100? Return the name and monthly maintenance of the facilities in question.**

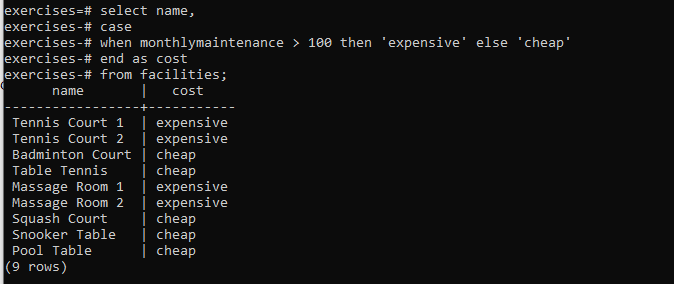
select name,

case

when monthlymaintenance > 100 then 'expensive' else 'cheap'

end as cost

from facilities;



# 8) Working with dateshttps://pgexercises.com/assets/tick2.svg

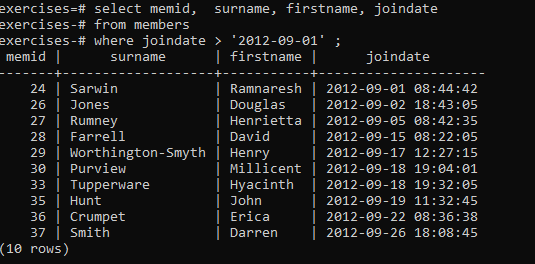
### **Question**

**How can you produce a list of members who joined after the start of September 2012? Return the memid, surname, firstname, and joindate of the members in question.**

select memid, surname, firstname, joindate

from members

where joindate > '2012-09-01' ;



# 9) Removing duplicates, and orderinghttps://pgexercises.com/assets/tick2.svg

### **Question**

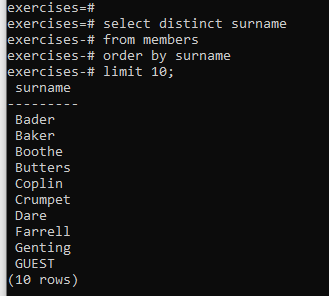
**How can you produce an ordered list of the first 10 surnames in the members table? The list must not contain duplicates.**

select distinct surname

from members

order by surname

limit 10;



# 10) Combining results from multiple querieshttps://pgexercises.com/assets/tick2.svg

### **Question**

**You, for some reason, want a combined list of all surnames and all facility names. Yes, this is a contrived example :-). Produce that list!**

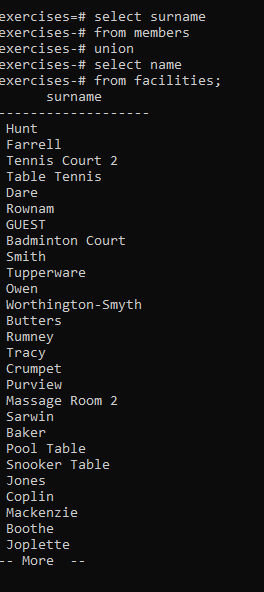
select surname

from members

union

select name

from facilities;



# 11)Simple aggregationhttps://pgexercises.com/assets/tick2.svg

### **Question**

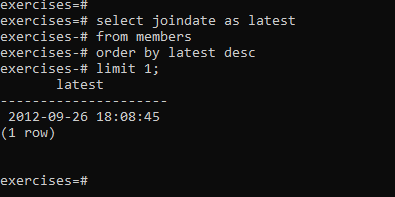
**You'd like to get the signup date of your last member. How can you retrieve this information?**

select joindate as latest

from members

order by latest desc

limit 1;



# 12) More aggregationhttps://pgexercises.com/assets/tick2.svg

### **Question**

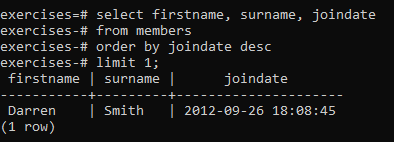
**You'd like to get the first and last name of the last member(s) who signed up - not just the date. How can you do that?**

select firstname, surname, joindate

from members

order by joindate desc

limit 1;

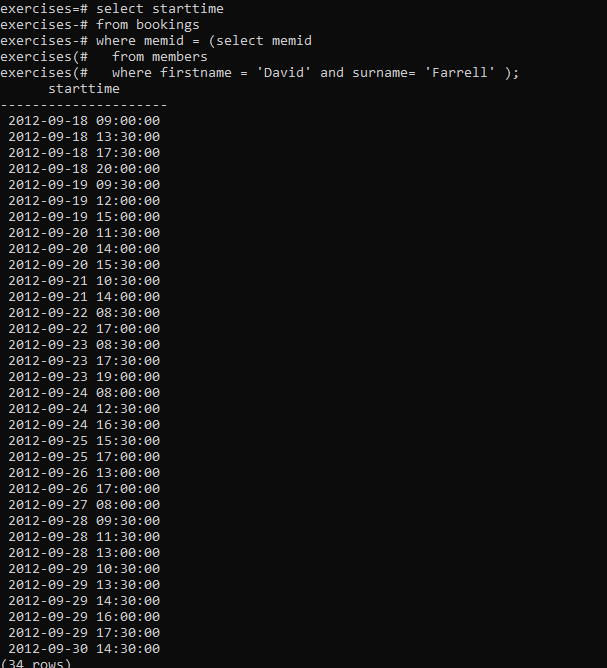


**JOINS AND SUBQUARY**

1)**Retrieve the start times of members' bookingshttps://pgexercises.com/assets/tick2.svg**

### **Question**

How can you produce a list of the start times for bookings by members named 'David Farrell'?



# 2) Work out the start times of bookings for tennis courtshttps://pgexercises.com/assets/tick2.svg

### **Question**

How can you produce a list of the start times for bookings for tennis courts, for the date '2012-09-21'? Return a list of start time and facility name pairings, ordered by the time.

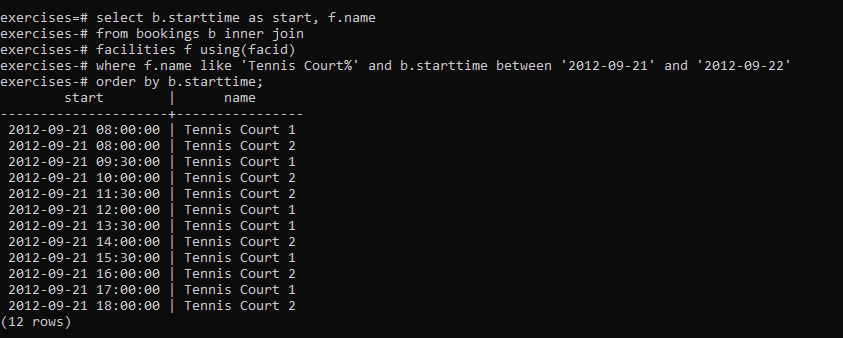
select b.starttime as start, f.name

from bookings b inner join

facilities f using(facid)

where f.name like 'Tennis Court%' and b.starttime between '2012-09-21' and '2012-09-22'

order by b.starttime;



**3)**

# Produce a list of all members who have recommended another memberhttps://pgexercises.com/assets/tick2.svg

### **Question**

How can you output a list of all members who have recommended another member? Ensure that there are no duplicates in the list, and that results are ordered by (surname, firstname).

select distinct r.firstname , r.surname

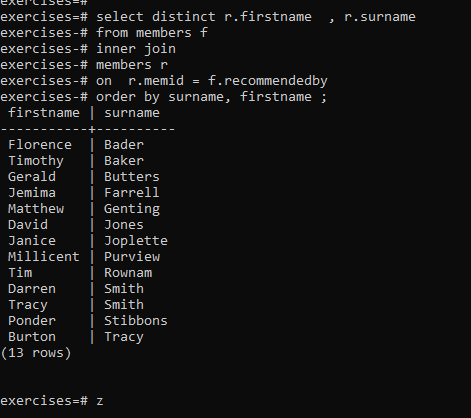
from members f

inner join

members r

on r.memid = f.recommendedby

order by surname, firstname ;



# 4) Produce a list of all members, along with their recommenderhttps://pgexercises.com/assets/tick2.svg

### **Question**

How can you output a list of all members, including the individual who recommended them (if any)? Ensure that results are ordered by (surname, firstname).

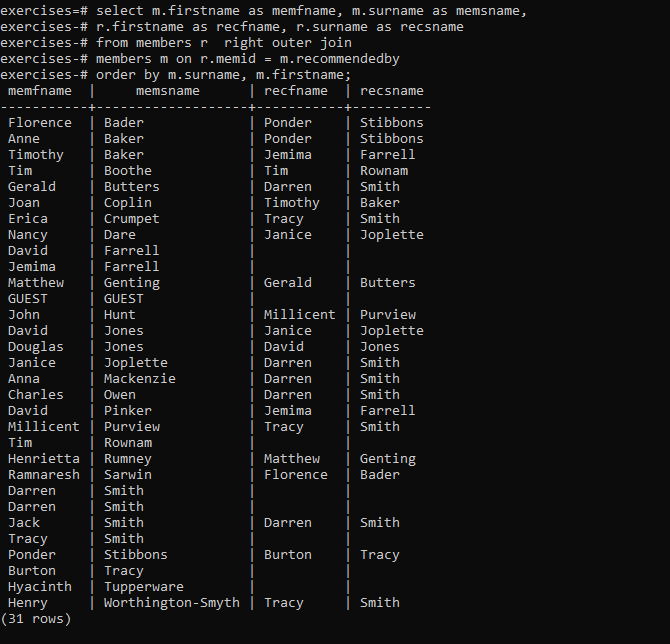
select m.firstname as memfname, m.surname as memsname,

r.firstname as recfname, r.surname as recsname

from members r right outer join

members m on r.memid = m.recommendedby

order by m.surname, m.firstname;



# 5) Produce a list of all members who have used a tennis courthttps://pgexercises.com/assets/tick2.svg

### **Question**

How can you produce a list of all members who have used a tennis court? Include in your output the name of the court, and the name of the member formatted as a single column. Ensure no duplicate data, and order by the member name followed by the facility name.

select distinct m.firstname||' '||m.surname as member, f.name as facility

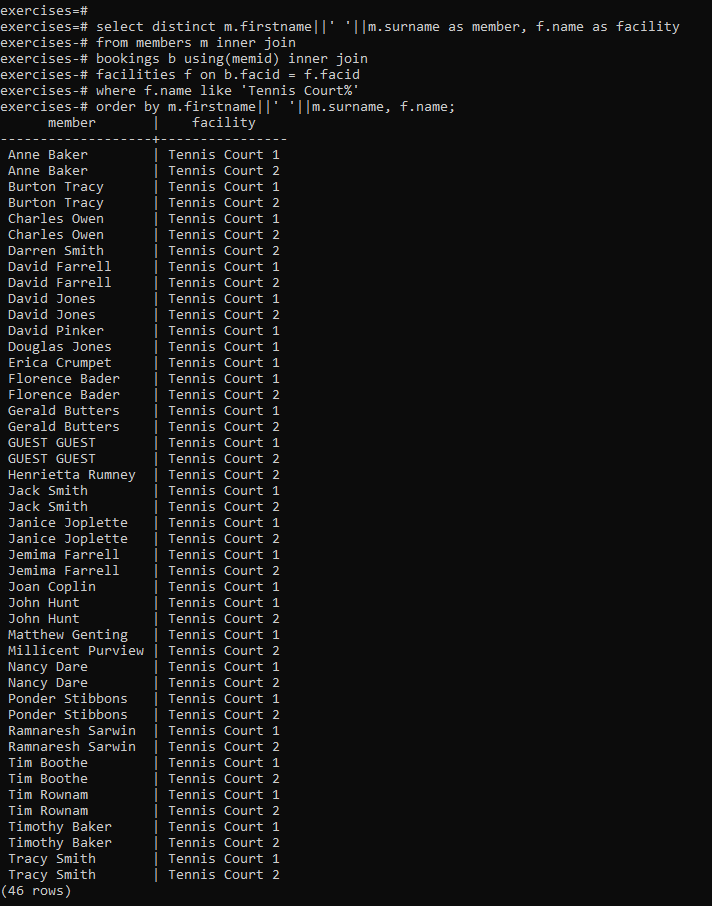
from members m inner join

bookings b using(memid) inner join

facilities f on b.facid = f.facid

where f.name like 'Tennis Court%'

order by m.firstname||' '||m.surname, f.name;



# 6) Produce a list of costly bookingshttps://pgexercises.com/assets/tick2.svg

### **Question**

How can you produce a list of bookings on the day of 2012-09-14 which will cost the member (or guest) more than $30? Remember that guests have different costs to members (the listed costs are per half-hour 'slot'), and the guest user is always ID 0. Include in your output the name of the facility, the name of the member formatted as a single column, and the cost. Order by descending cost, and do not use any subqueries

select m.firstname|| ' ' ||m.surname as member, f.name as facility,

case

when m.memid = 0 then

b.slots\*f.guestcost

else

b.slots\*f.membercost

end as cost

from members m inner join

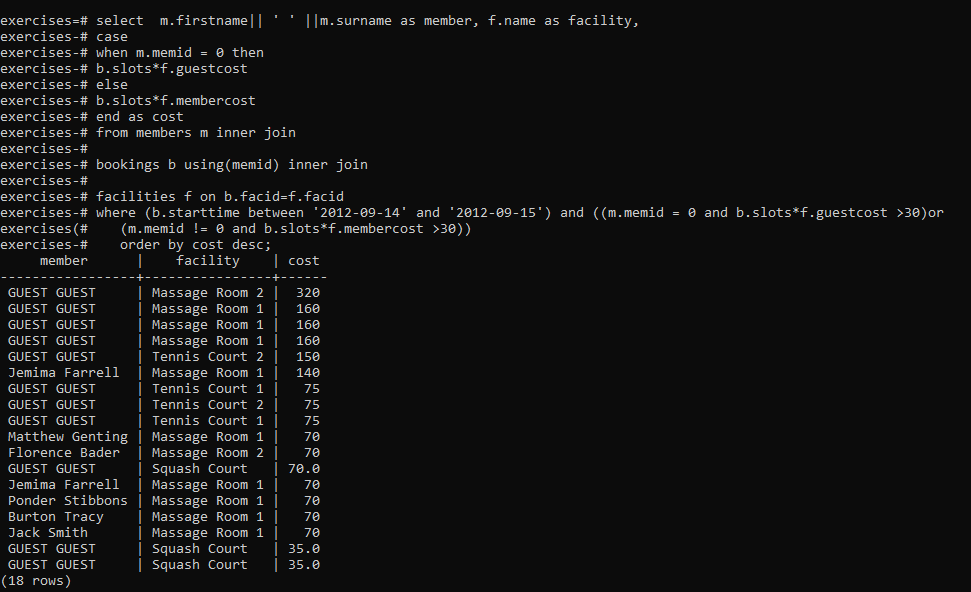
bookings b using(memid) inner join

facilities f on b.facid=f.facid

where (b.starttime between '2012-09-14' and '2012-09-15') and ((m.memid = 0 and b.slots\*f.guestcost >30)or

(m.memid != 0 and b.slots\*f.membercost >30))

order by cost desc;



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

### **Qestion**

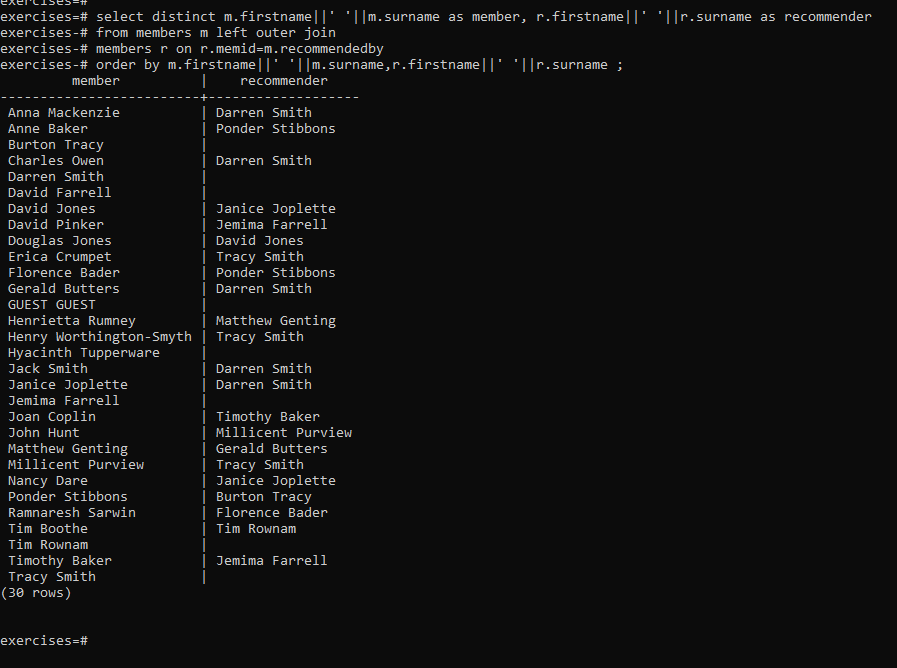
How can you output a list of all members, including the individual who recommended them (if any), without using any joins? Ensure that there are no duplicates in the list, and that each firstname + surname pairing is formatted as a column and ordered.

select distinct m.firstname||' '||m.surname as member, r.firstname||' '||r.surname as recommender

from members m left outer join

members r on r.memid=m.recommendedby

order by m.firstname||' '||m.surname,r.firstname||' '||r.surname ;



### **7)Question**

How can you produce a list of bookings on the day of 2012-09-14 which will cost the member (or guest) more than $30? Remember that guests have different costs to members (the listed costs are per half-hour 'slot'), and the guest user is always ID 0. Include in your output the name of the facility, the name of the member formatted as a single column, and the cost. Order by descending cost.

select m.firstname|| ' ' ||m.surname as member, f.name as facility,

case

when m.memid = 0 then

b.slots\*f.guestcost

else

b.slots\*f.membercost

end as cost

from members m inner join

bookings b using(memid) inner join

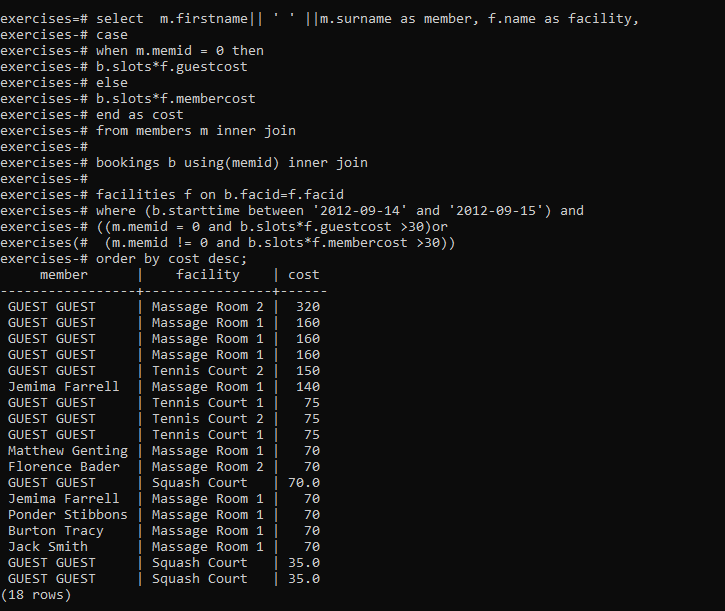
facilities f on b.facid=f.facid

where (b.starttime between '2012-09-14' and '2012-09-15') and

((m.memid = 0 and b.slots\*f.guestcost >30)or

(m.memid != 0 and b.slots\*f.membercost >30))

order by cost desc;



# Modifying data

# Insert some data into a tablehttps://pgexercises.com/assets/tick2.svg

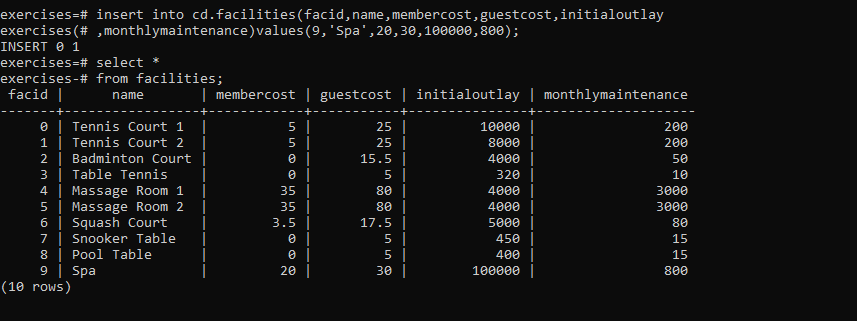
### **1)Question**

**The club is adding a new facility - a spa. We need to add it into the facilities table. Use the following values:**

* facid: 9, Name: 'Spa', membercost: 20, guestcost: 30, initialoutlay: 100000, monthlymaintenance: 800.

insert into facilities(facid,name,membercost,guestcost,initialoutlay

,monthlymaintenance)values(9,'Spa',20,30,100000,800);

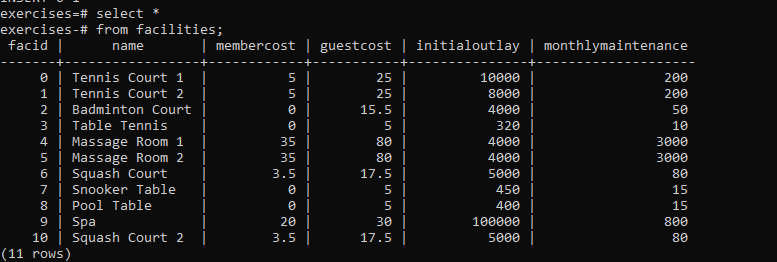


# 2) Insert multiple rows of data into a tablehttps://pgexercises.com/assets/tick2.svg

### **Question**

In the previous exercise, you learned how to add a facility. Now you're going to add multiple facilities in one command. Use the following values:

* facid: 9, Name: 'Spa', membercost: 20, guestcost: 30, initialoutlay: 100000, monthlymaintenance: 800.
* facid: 10, Name: 'Squash Court 2', membercost: 3.5, guestcost: 17.5, initialoutlay: 5000, monthlymaintenance: 80.



3)

# Insert calculated data into a table

### **Question**

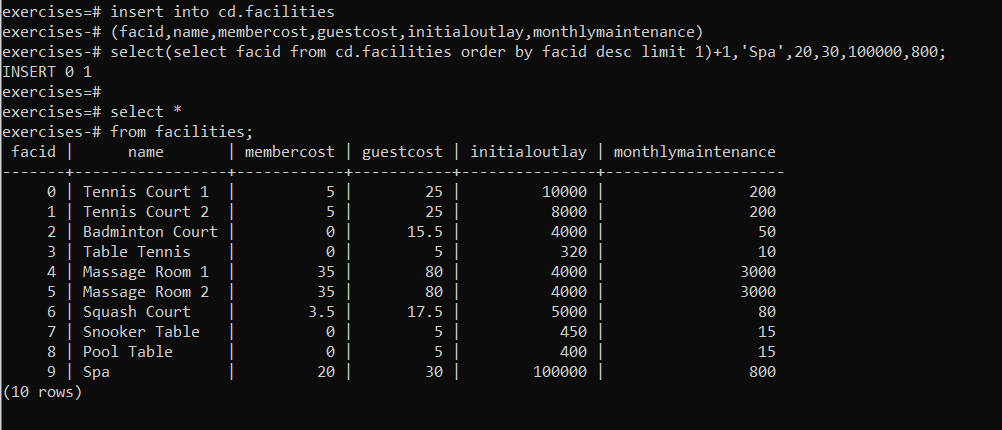
Let's try adding the spa to the facilities table again. This time, though, we want to automatically generate the value for the next facid, rather than specifying it as a constant. Use the following values for everything else:

* Name: 'Spa', membercost: 20, guestcost: 30, initialoutlay: 100000, monthlymaintenance: 800.

insert into cd.facilities

(facid,name,membercost,guestcost,initialoutlay,monthlymaintenance)

select(select facid from cd.facilities order by facid desc limit 1)+1,'Spa',20,30,100000,800;



4)Update some existing data

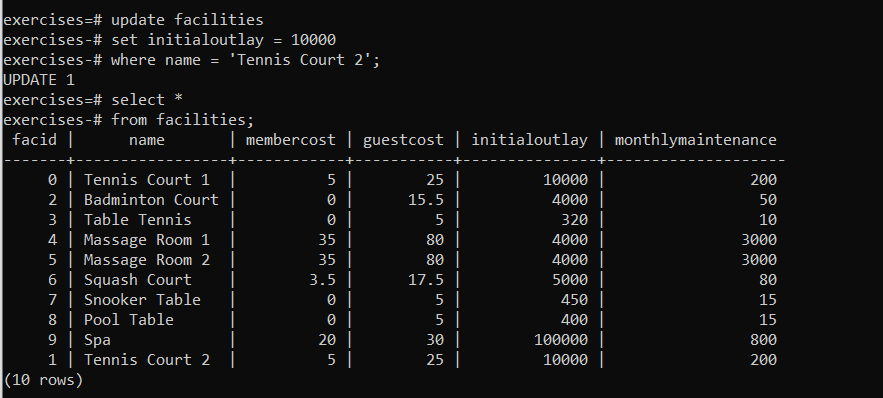
### **Question**

We made a mistake when entering the data for the second tennis court. The initial outlay was 10000 rather than 8000: you need to alter the data to fix the error.

update facilities

set initialoutlay = 10000

where name = 'Tennis Court 2';



**5) Update multiple rows and columns at the same time**

### **Question**

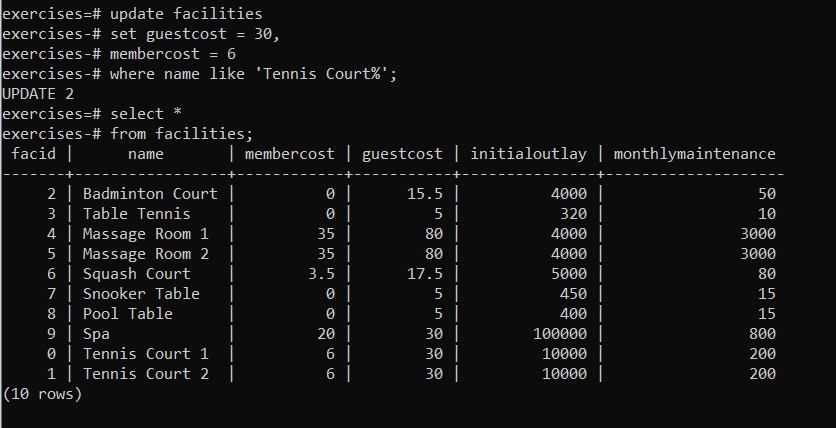
We want to increase the price of the tennis courts for both members and guests. Update the costs to be 6 for members, and 30 for guests.

update facilities

set guestcost = 30,

membercost = 6

where name like 'Tennis Court%';



6)Update a row based on the contents of another row

### **Question**

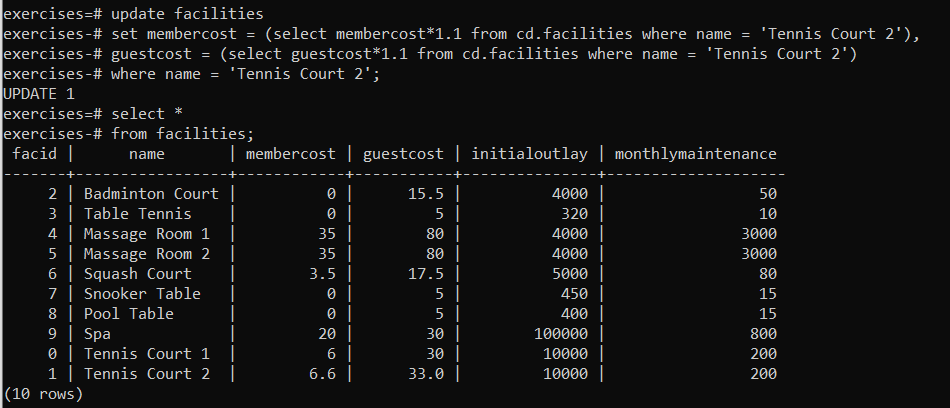
We want to alter the price of the second tennis court so that it costs 10% more than the first one. Try to do this without using constant values for the prices, so that we can reuse the statement if we want to.

update facilities

set membercost = (select membercost\*1.1 from cd.facilities where name = 'Tennis Court 2'),

guestcost = (select guestcost\*1.1 from cd.facilities where name = 'Tennis Court 2')

where name = 'Tennis Court 2';

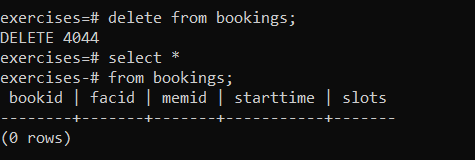


7)Delete all booking

### **Question**

As part of a clearout of our database, we want to delete all bookings from the cd.bookings table. How can we accomplish this?

delete from bookings;



8)Delete a member from the cd.members tabl

### **Question**

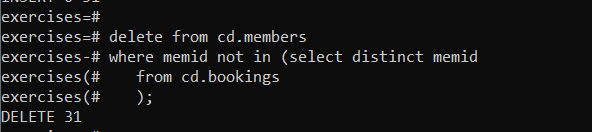
We want to remove member 37, who has never made a booking, from our database. How can we achieve that?

delete

from members

where memid not in (select distinct memid

from cd.bookings);



9)Delete based on a subquery

### **Question**

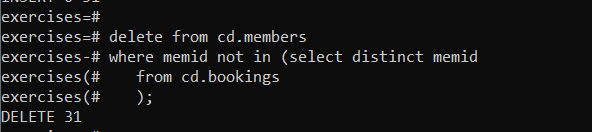
In our previous exercises, we deleted a specific member who had never made a booking. How can we make that more general, to delete all members who have never made a booking?

delete

from cd.members

where memid not in (select distinct memid

from cd.bookings);



# Aggregation

1)

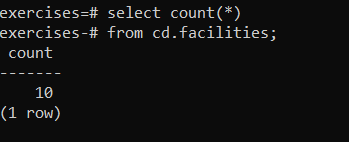
# Count the number of facilities

### **Question**

For our first foray into aggregates, we're going to stick to something simple. We want to know how many facilities exist - simply produce a total count.

select count(\*)

from cd.facilities;



2)

# Count the number of expensive facilities

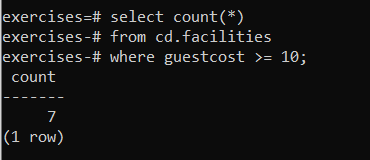
### **Question**

Produce a count of the number of facilities that have a cost to guests of 10 or more.

select count(\*)

from cd.facilities

where guestcost >= 10;



3**)**

# Count the number of recommendations each member makes

### **Question**

Produce a count of the number of recommendations each member has made. Order by member ID.

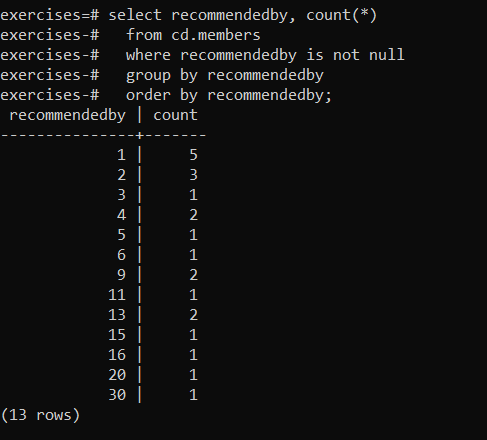
select recommendedby, count(\*)

from cd.members

where recommendedby is not null

group by recommendedby

order by recommendedby;



4)

# List the total slots booked per facility

### **Question**

Produce a list of the total number of slots booked per facility. For now, just produce an output table consisting of facility id and slots, sorted by facility id.

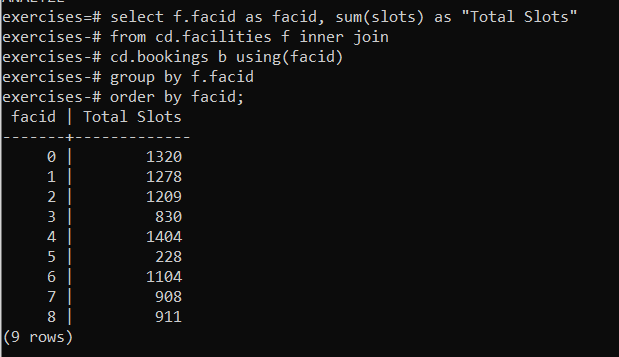
select f.facid as facid, sum(slots) as "Total Slots"

from cd.facilities f inner join

cd.bookings b using(facid)

group by f.facid

order by facid;



5)

# List the total slots booked per facility in a given month

### **Question**

Produce a list of the total number of slots booked per facility in the month of September 2012. Produce an output table consisting of facility id and slots, sorted by the number of slots.

select facid , sum(slots)

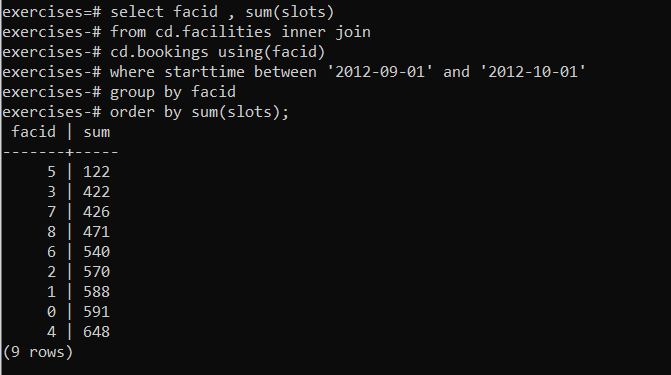
from cd.facilities inner join

cd.bookings using(facid)

where starttime between '2012-09-01' and '2012-10-01'

group by facid

order by sum(slots);



6)

# List the total slots booked per facility per month

### **Question**

Produce a list of the total number of slots booked per facility per month in the year of 2012. Produce an output table consisting of facility id and slots, sorted by the id and month.

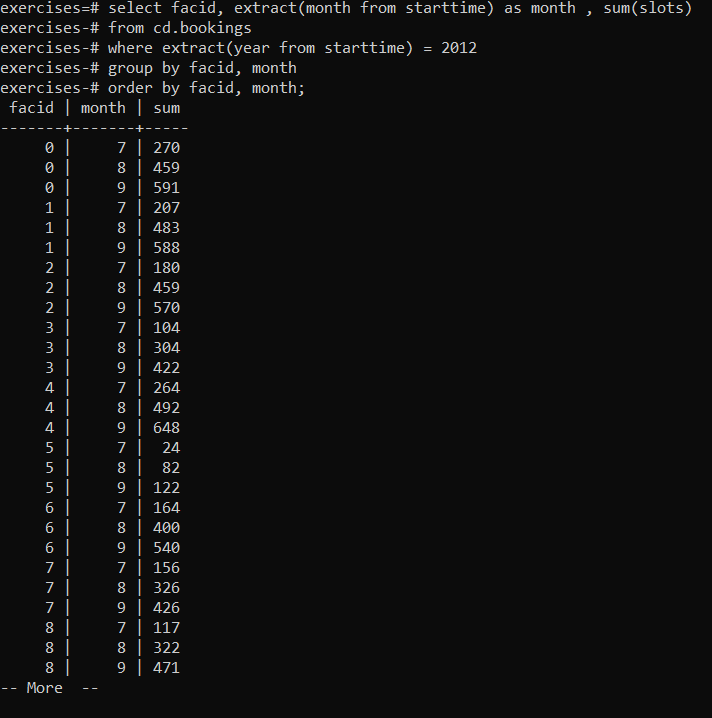
select facid, extract(month from starttime) as month , sum(slots)

from cd.bookings

where extract(year from starttime) = 2012

group by facid, month

order by facid, month;



7)

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

### **Question**

Find the total number of members (including guests) who have made at least one booking.

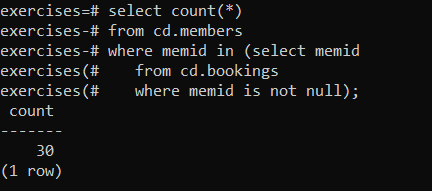
select count(\*)

from cd.members

where memid in (select memid

from cd.bookings

where memid is not null);



8)

# List facilities with more than 1000 slots booked

### **Question**

Produce a list of facilities with more than 1000 slots booked. Produce an output table consisting of facility id and slots, sorted by facility id.

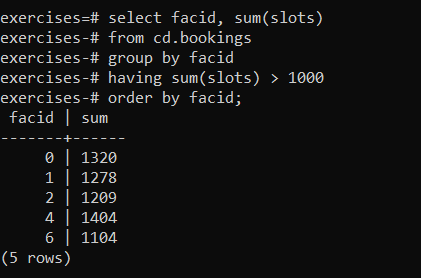
select facid, sum(slots)

from cd.bookings

group by facid

having sum(slots) > 1000

order by facid;



9)

# Find the total revenue of each facility

### **Question**

Produce a list of facilities along with their total revenue. The output table should consist of facility name and revenue, sorted by revenue. Remember that there's a different cost for guests and members!

select f.name, sum(slots \* case

when memid = 0 then f.guestcost

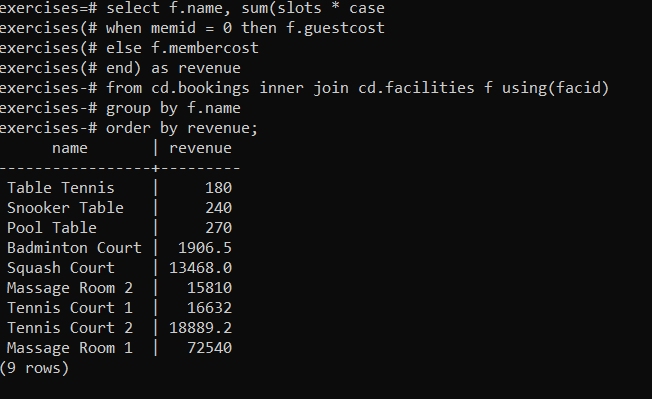
else f.membercost

end) as revenue

from cd.bookings inner join cd.facilities f using(facid)

group by f.name

order by revenue;



10)

# Find facilities with a total revenue less than 1000

### **Question**

Produce a list of facilities with a total revenue less than 1000. Produce an output table consisting of facility name and revenue, sorted by revenue. Remember that there's a different cost for guests and members!

select f.name, sum(case when memid=0 then slots\*f.guestcost

else slots\*f.membercost end) as revenue

from cd.facilities f inner join cd.bookings using(facid)

group by f.name

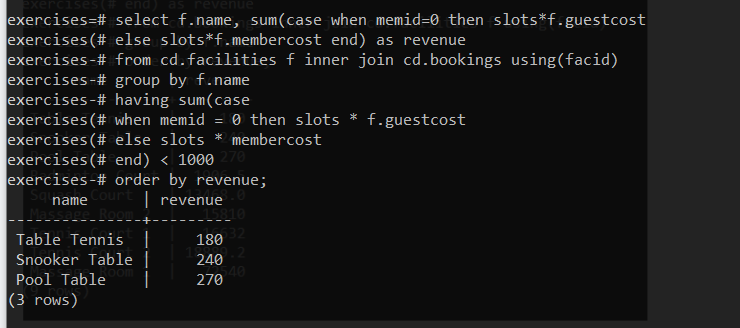
having sum(case

when memid = 0 then slots \* f.guestcost

else slots \* membercost

end) < 1000

order by revenue;



11)

# 

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

### **Question**

Output the facility id that has the highest number of slots booked. For bonus points, try a version without a LIMIT clause. This version will probably look messy!

select facid ,sum(slots)

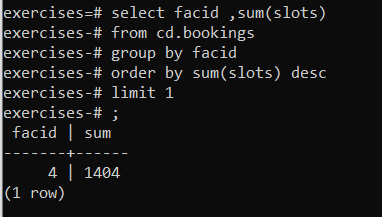
from cd.bookings

group by facid

order by sum(slots) desc

limit 1

;



12)

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

### **Question**

Produce a list of the total number of slots booked per facility per month in the year of 2012. In this version, include output rows containing totals for all months per facility, and a total for all months for all facilities. The output table should consist of facility id, month and slots, sorted by the id and month. When calculating the aggregated values for all months and all facids, return null values in the month and facid columns.

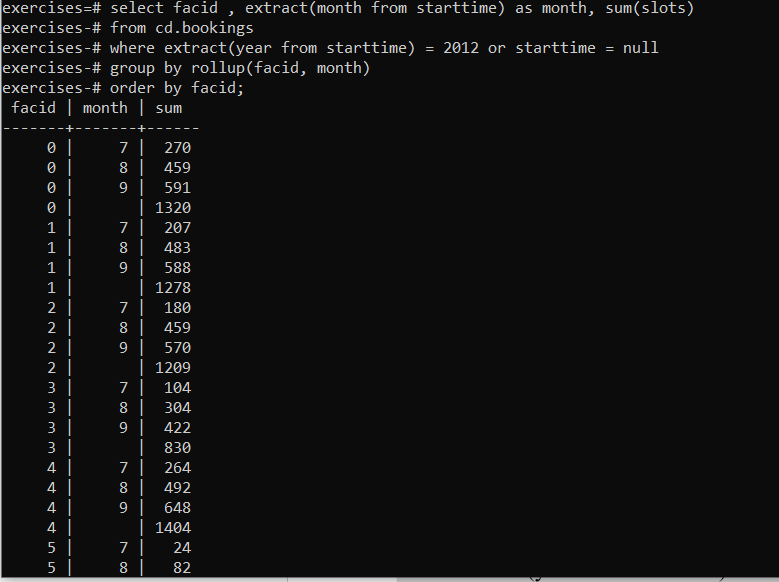
select facid , extract(month from starttime) as month, sum(slots)

from cd.bookings

where extract(year from starttime) = 2012 or starttime = null

group by rollup(facid, month)

order by facid;



13)

# List the total hours booked per named facility

### **Question**

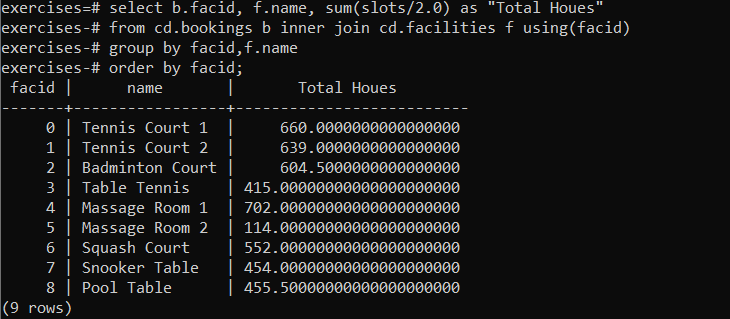
Produce a list of the total number of *hours* booked per facility, remembering that a slot lasts half an hour. The output table should consist of the facility id, name, and hours booked, sorted by facility id. Try formatting the hours to two decimal places.

select b.facid, f.name, sum(slots/2.0) as "Total Houes"

from cd.bookings b inner join cd.facilities f using(facid)

group by facid,f.name

order by facid;



14)

# List each member's first booking after September 1st 2012

### **Question**

Produce a list of each member name, id, and their first booking after September 1st 2012. Order by member ID.

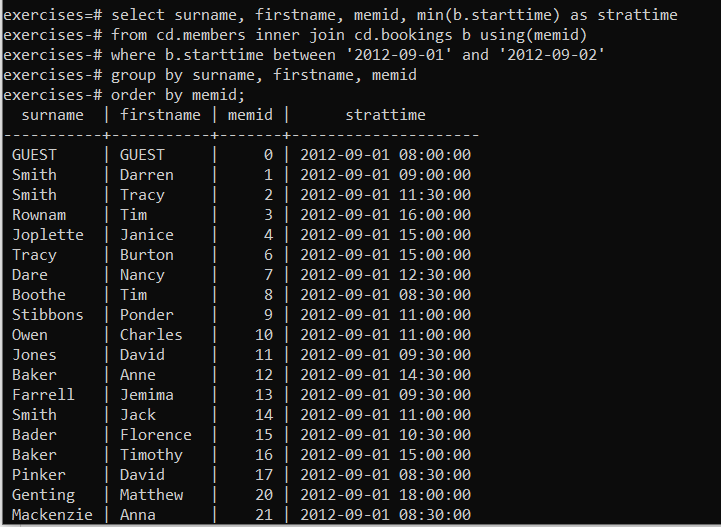
select surname, firstname, memid, min(b.starttime) as strattime

from cd.members inner join cd.bookings b using(memid)

where b.starttime between '2012-09-01' and '2012-09-02'

group by surname, firstname, memid

order by memid;



15)

# Produce a list of member names, with each row containing the total member count

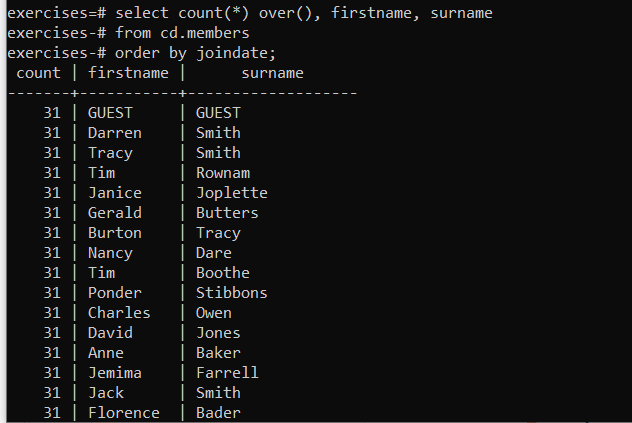
### **Question**

Produce a list of member names, with each row containing the total member count. Order by join date, and include guest members.

select count(\*) over(), firstname, surname

from cd.members

order by joindate;



16)

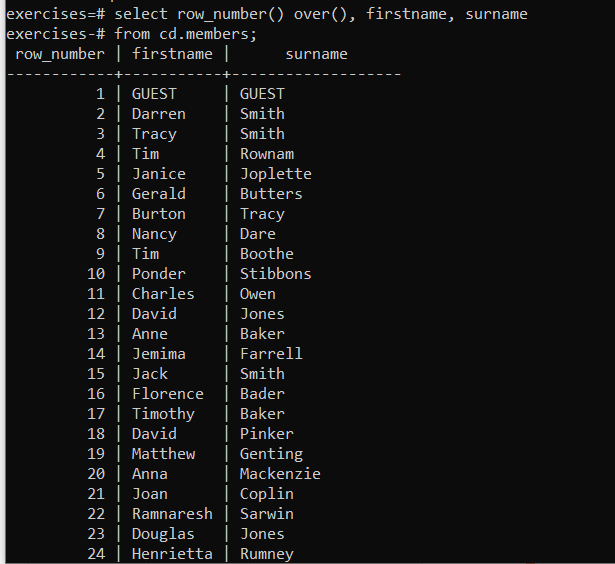
# Produce a numbered list of members

### **Question**

Produce a monotonically increasing numbered list of members (including guests), ordered by their date of joining. Remember that member IDs are not guaranteed to be sequential.

select row\_number() over(), firstname, surname

from cd.members;



17)

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

### **Question**

Output the facility id that has the highest number of slots booked. Ensure that in the event of a tie, all tieing results get output.

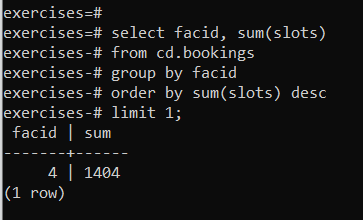
select facid, sum(slots)

from cd.bookings

group by facid

order by sum(slots) desc

limit 1;



18)

# Rank members by (rounded) hours used

### **Question**

Produce a list of members (including guests), along with the number of hours they've booked in facilities, rounded to the nearest ten hours. Rank them by this rounded figure, producing output of first name, surname, rounded hours, rank. Sort by rank, surname, and first name.

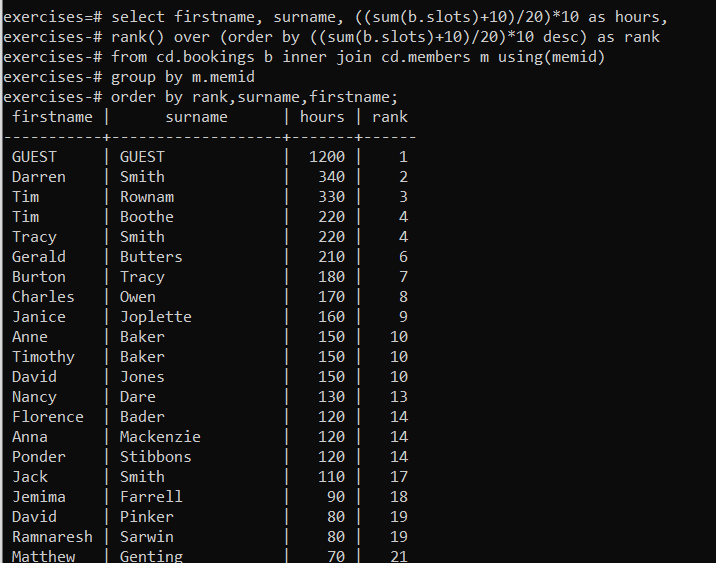
select firstname, surname, ((sum(b.slots)+10)/20)\*10 as hours,

rank() over (order by ((sum(b.slots)+10)/20)\*10 desc) as rank

from cd.bookings b inner join cd.members m using(memid)

group by m.memid

order by rank,surname,firstname;



19)

# Find the top three revenue generating facilities

### **Question**

Produce a list of the top three revenue generating facilities (including ties). Output facility name and rank, sorted by rank and facility name.

select name, rank from (

select facs.name as name, rank() over (order by sum(case

when memid = 0 then slots \* facs.guestcost

else slots \* membercost

end) desc) as rank

from cd.bookings bks

inner join cd.facilities facs

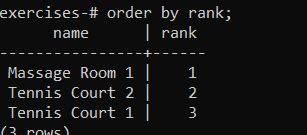
on bks.facid = facs.facid

group by facs.name

) as subq

where rank <= 3

order by rank;



DATE

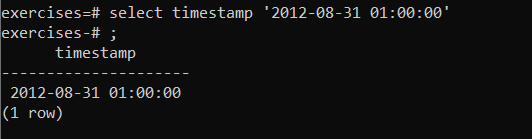
1)

# Produce a timestamp for 1 a.m. on the 31st of August 2012

### **Question**

Produce a timestamp for 1 a.m. on the 31st of August 2012.

select timestamp '2012-08-31 01:00:00'

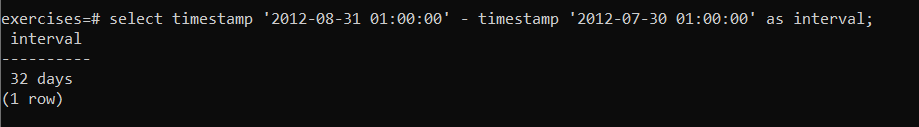


2)Subtract timestamps from each other

### **Question**

Find the result of subtracting the timestamp '2012-07-30 01:00:00' from the timestamp '2012-08-31 01:00:00'

select timestamp '2012-08-31 01:00:00' - timestamp '2012-07-30 01:00:00' as interval;



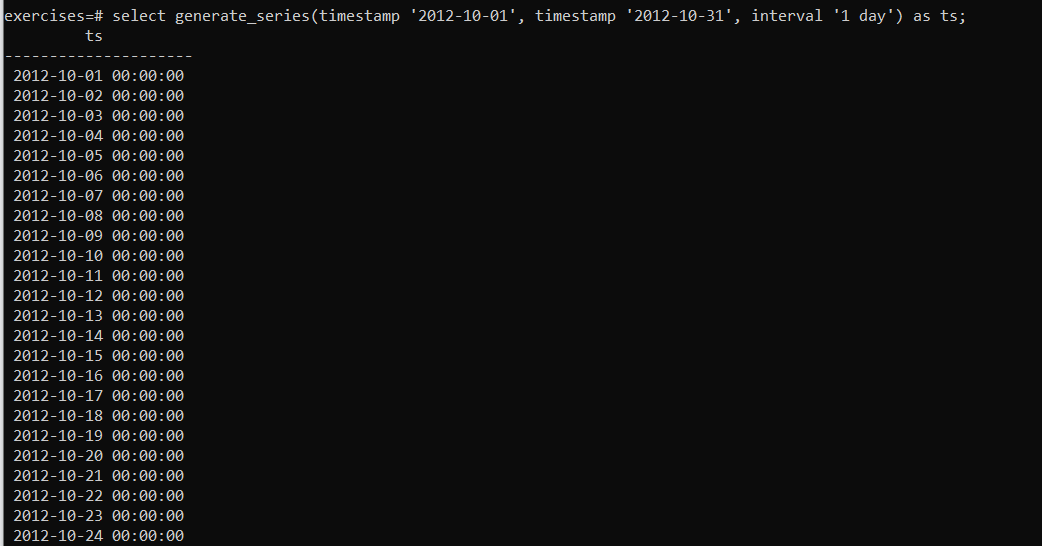
3)

# Generate a list of all the dates in October 201

### **Question**

Produce a list of all the dates in October 2012. They can be output as a timestamp (with time set to midnight) or a date.

*select generate\_series(timestamp '2012-10-01', timestamp '2012-10-31', interval '1 day') as ts;*



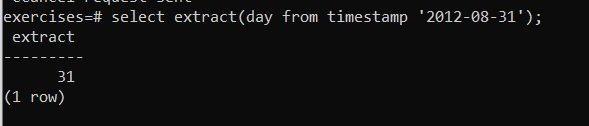
4)

# Get the day of the month from a timestamp

### **Question**

Get the day of the month from the timestamp '2012-08-31' as an integer.

*select extract(day from timestamp '2012-08-31');*

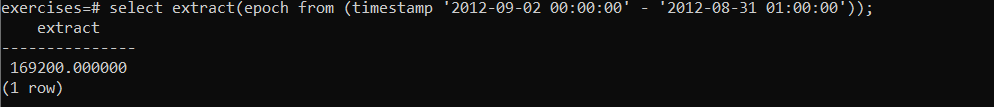


5)Work out the number of seconds between timestamps

### **Question**

Work out the number of seconds between the timestamps '2012-08-31 01:00:00' and '2012-09-02 00:00:00'

*select extract(epoch from (timestamp '2012-09-02 00:00:00' - '2012-08-31 01:00:00'));*



5)

# Work out the number of days remaining in the month

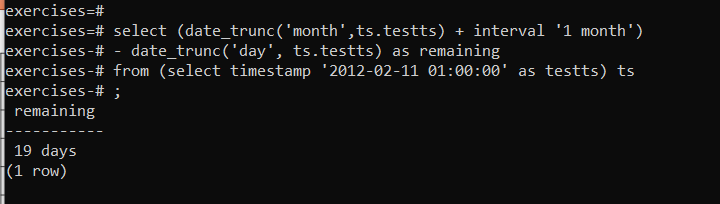
### **Question**

For any given timestamp, work out the number of days remaining in the month. The current day should count as a whole day, regardless of the time. Use '2012-02-11 01:00:00' as an example timestamp for the purposes of making the answer. Format the output as a single interval value.

*select (date\_trunc('month',ts.testts) + interval '1 month')*

*- date\_trunc('day', ts.testts) as remaining*

*from (select timestamp '2012-02-11 01:00:00' as testts) ts*



6)Work out the end time of bookings

### **Question**

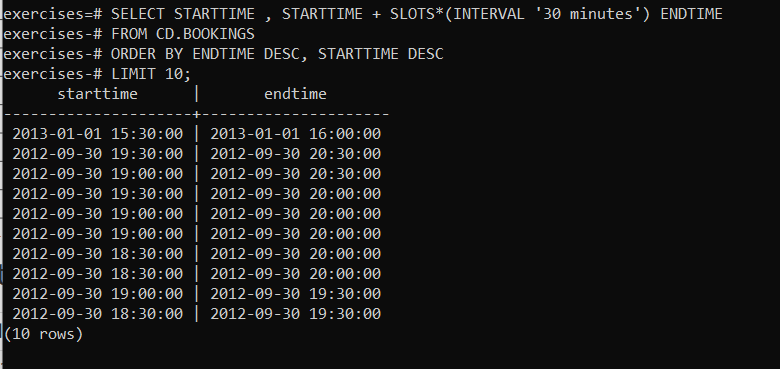
Return a list of the start and end time of the last 10 bookings (ordered by the time at which they end, followed by the time at which they start) in the system

*SELECT STARTTIME , STARTTIME + SLOTS\*(INTERVAL '30 minutes') ENDTIME*

*FROM CD.BOOKINGS*

*ORDER BY ENDTIME DESC, STARTTIME DESC*

*LIMIT 10;*



STRINGS

1)

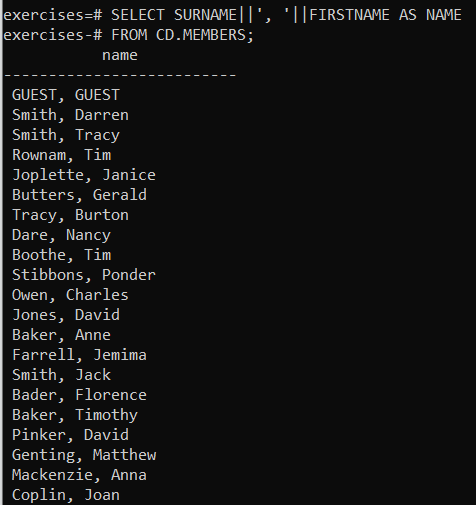
# Format the names of members

### **Question**

Output the names of all members, formatted as 'Surname, Firstname'

*SELECT SURNAME||', '||FIRSTNAME AS NAME*

*FROM CD.MEMBERS;*



2)

# Find facilities by a name prefix

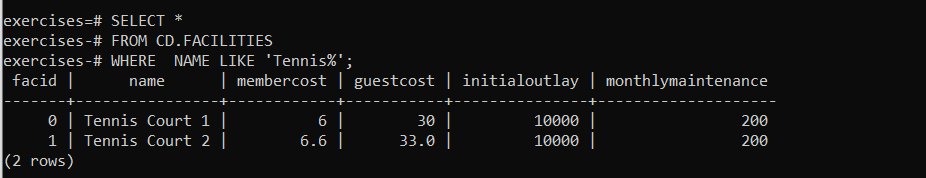
### **Question**

Find all facilities whose name begins with 'Tennis'. Retrieve all columns

*SELECT \**

*FROM CD.FACILITIES*

*WHERE NAME LIKE 'Tennis%';*



3)

# Perform a case-insensitive search

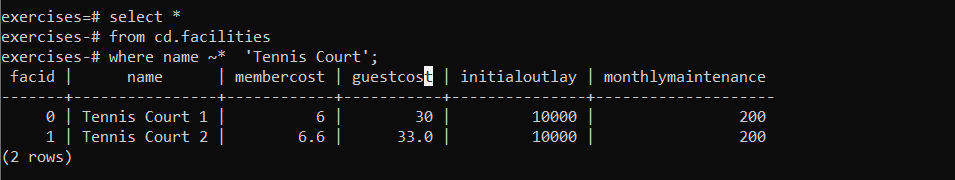
### **Question**

Perform a case-insensitive search to find all facilities whose name begins with 'tennis'. Retrieve all columns.

*select \**

*from cd.facilities*

*where name ~\* 'Tennis Court';*



4)

# Find telephone numbers with parentheses

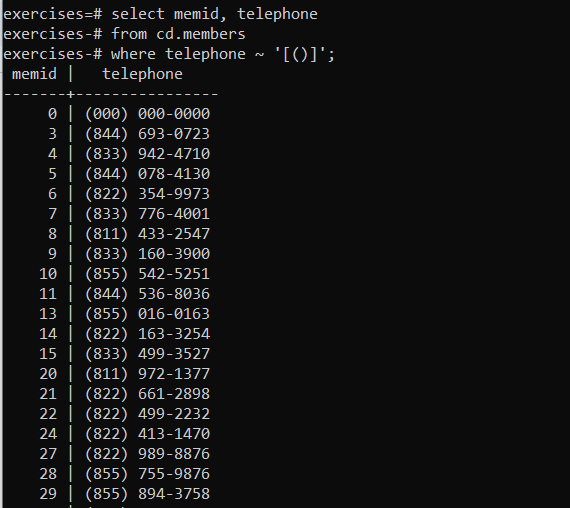
### **Question**

You've noticed that the club's member table has telephone numbers with very inconsistent formatting. You'd like to find all the telephone numbers that contain parentheses, returning the member ID and telephone number sorted by member ID.

*select memid, telephone*

*from cd.members*

*where telephone ~ '[()]';*



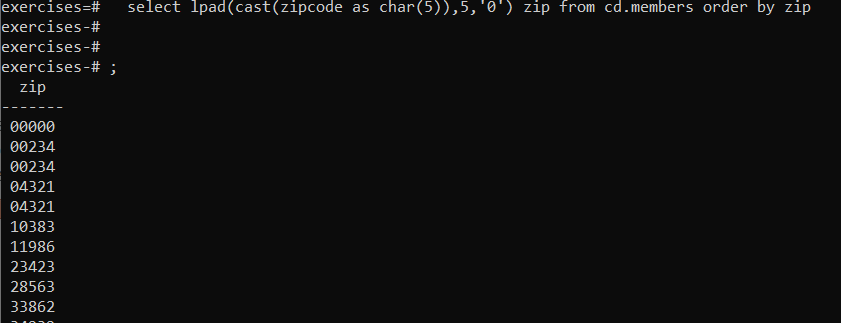
5)

# Pad zip codes with leading zeroes

### **Question**

The zip codes in our example dataset have had leading zeroes removed from them by virtue of being stored as a numeric type. Retrieve all zip codes from the members table, padding any zip codes less than 5 characters long with leading zeroes. Order by the new zip code.

*select lpad(cast(zipcode as char(5)),5,'0') zip from cd.members order by zip*



6)Count the number of members whose surname starts with each letter of the alphabet

### **Question**

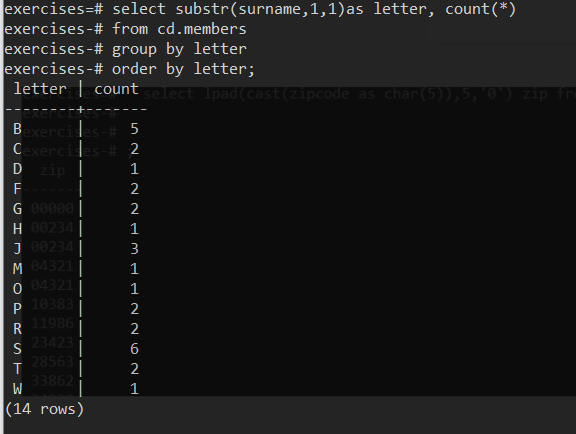
You'd like to produce a count of how many members you have whose surname starts with each letter of the alphabet. Sort by the letter, and don't worry about printing out a letter if the count is 0.

*select substr(surname,1,1)as letter, count(\*)*

*from cd.members*

*group by letter*

*order by letter;*



7)

# Clean up telephone numbers

### **Question**

The telephone numbers in the database are very inconsistently formatted. You'd like to print a list of member ids and numbers that have had '-','(',')', and ' ' characters removed. Order by member id.

*select memid, translate(telephone, '()- ','') as telephone*

*from cd.members*

*order by memid;*

