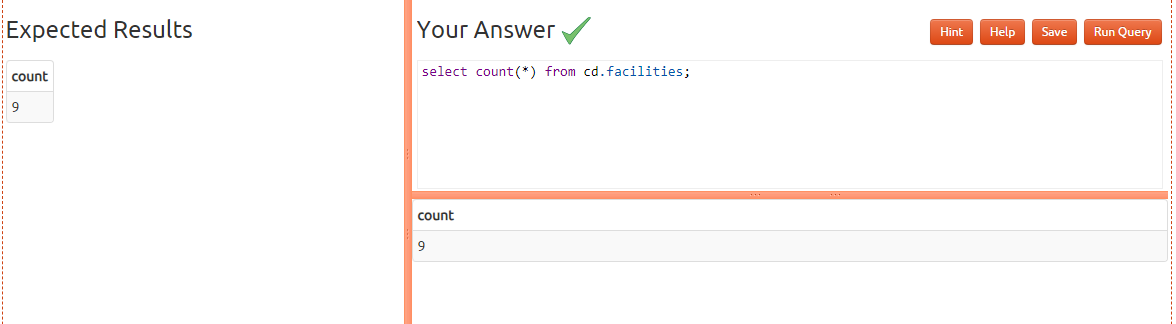
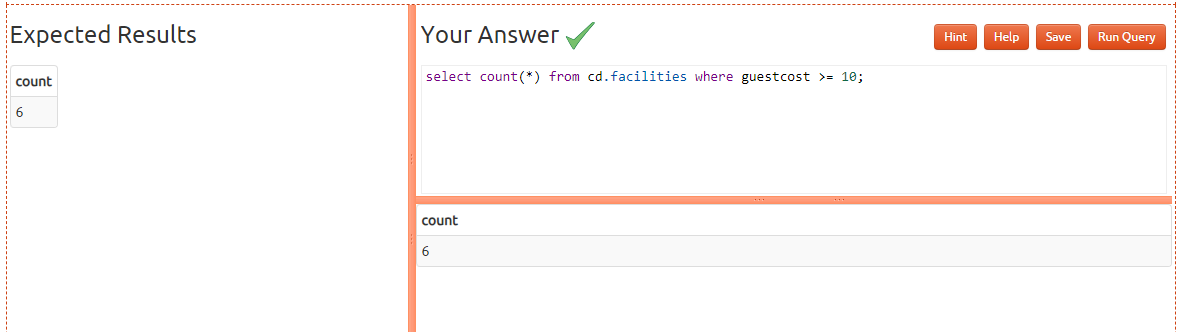
[Count the number of facilities](https://pgexercises.com/questions/aggregates/count.html)

select count(\*) from cd.facilities;



[Count the number of expensive facilities](https://pgexercises.com/questions/aggregates/count2.html)

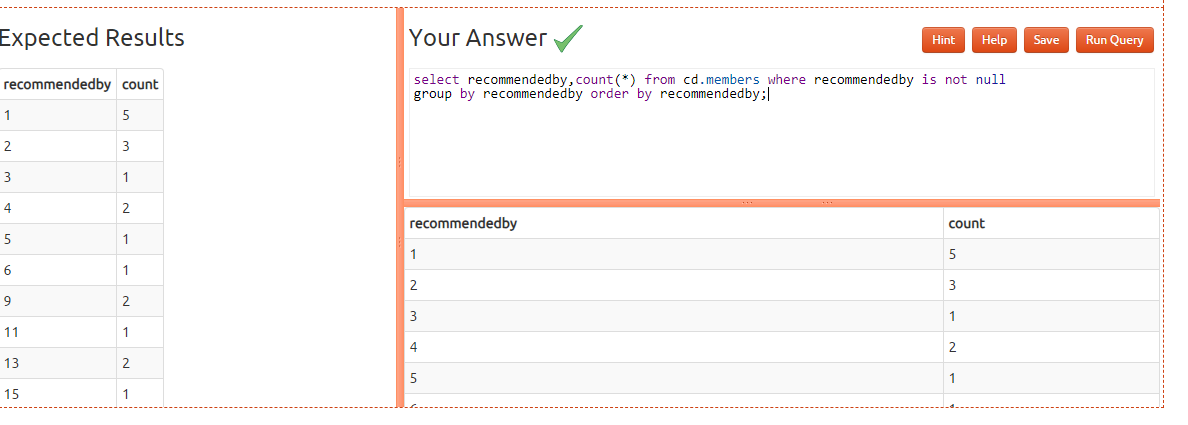
select count(\*) from cd.facilities where guestcost >= 10;



[Count the number of recommendations each member makes.](https://pgexercises.com/questions/aggregates/count3.html)

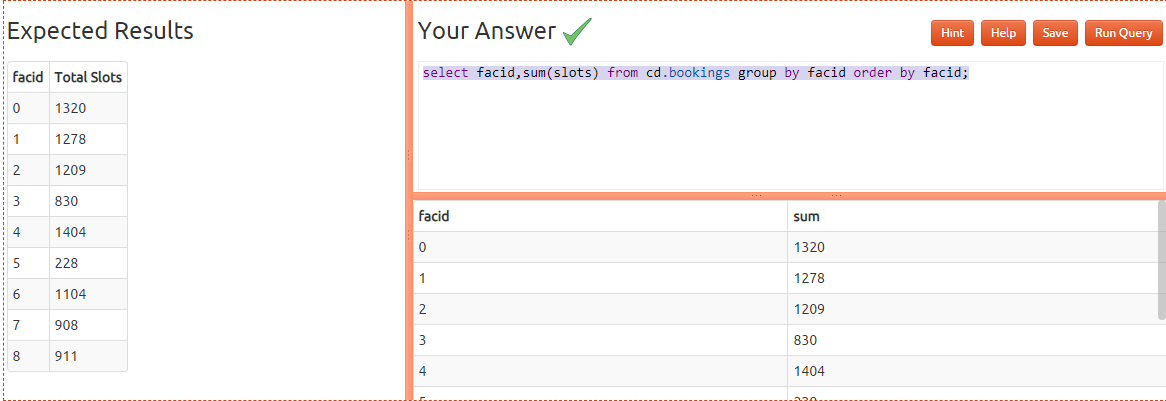
select recommendedby,count(\*) from cd.members where recommendedby is not null

group by recommendedby order by recommendedby;



[List the total slots booked per facility](https://pgexercises.com/questions/aggregates/fachours.html)

select facid,sum(slots) from cd.bookings group by facid order by facid;



[List the total slots booked per facility in a given month](https://pgexercises.com/questions/aggregates/fachoursbymonth.html)

SELECT facid,sum(slots) as Total\_Slots from cd.bookings

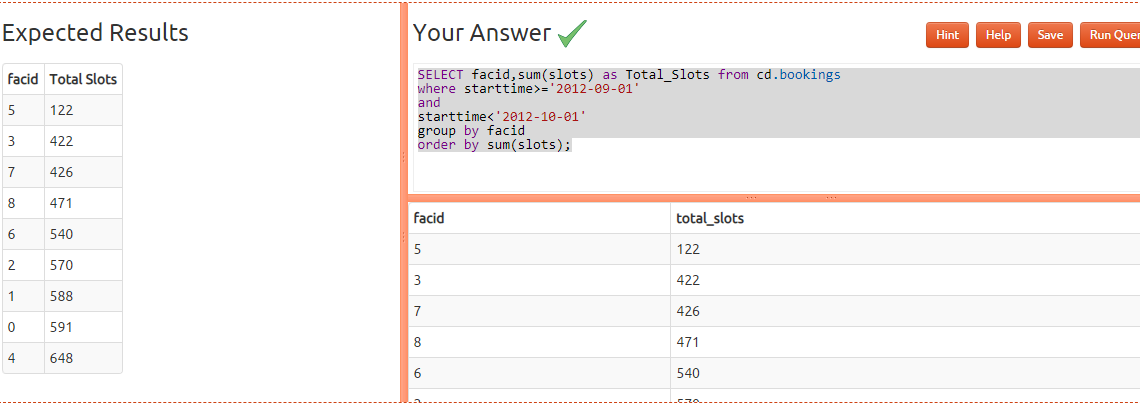
where starttime>='2012-09-01'

and

starttime<'2012-10-01'

group by facid

order by sum(slots);

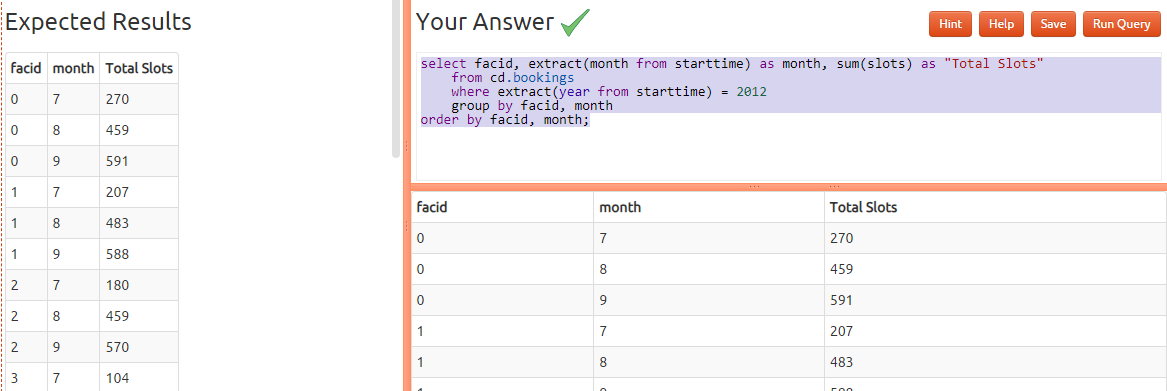
[List the total slots booked per facility per month](https://pgexercises.com/questions/aggregates/fachoursbymonth2.html)

select facid, extract(month from starttime) as month, sum(slots) as "Total Slots" from cd.bookings

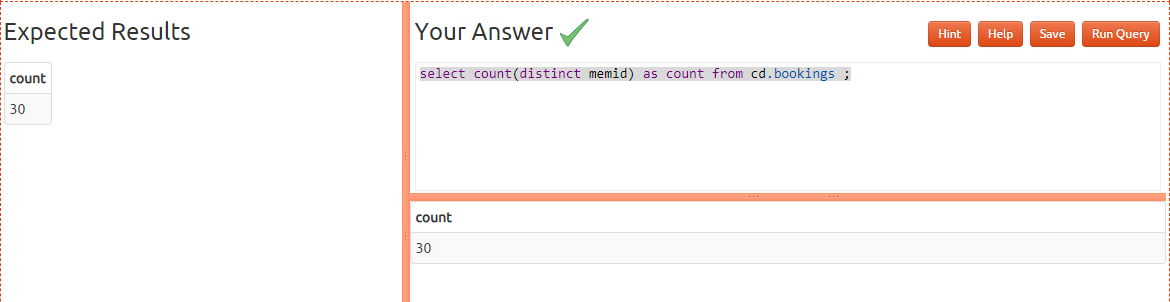
where extract(year from starttime) = 2012

group by facid, month

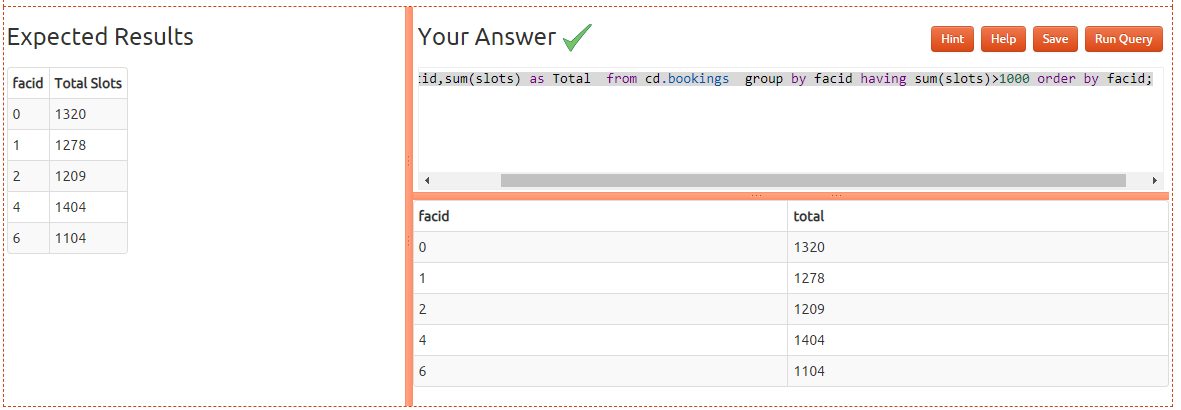
order by facid, month;



* [Find the count of members who have made at least one booking](https://pgexercises.com/questions/aggregates/members1.html)

select count(distinct memid) as count from cd.bookings ; 

* [List facilities with more than 1000 slots booked](https://pgexercises.com/questions/aggregates/fachours1a.html)

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

* [Find the total revenue of each facility](https://pgexercises.com/questions/aggregates/facrev.html)

SELECT f.name AS facility\_name,

SUM(b.slots \* CASE

WHEN b.memid = 0 THEN f.guestcost

ELSE f.membercost

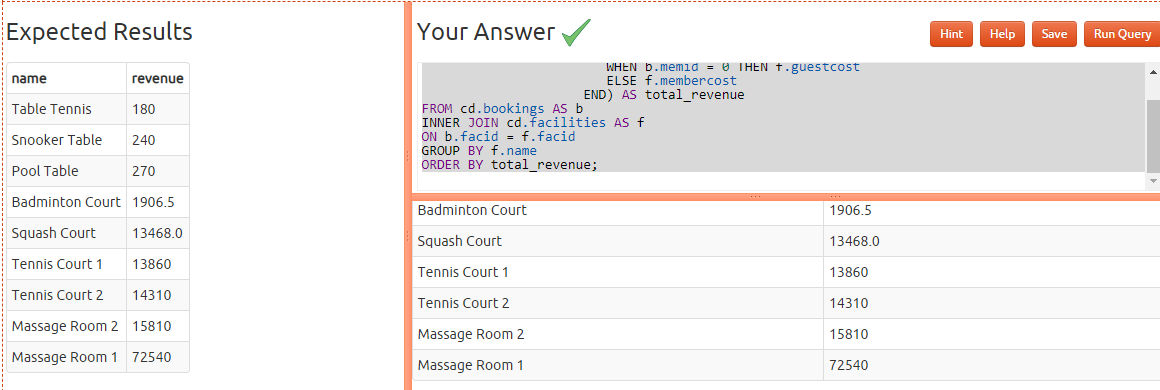
END) AS total\_revenue

FROM cd.bookings AS b

INNER JOIN cd.facilities AS f

ON b.facid = f.facid

GROUP BY f.name

ORDER BY total\_revenue; 

* [Find facilities with a total revenue less than 1000](https://pgexercises.com/questions/aggregates/facrev2.html)

select name, revenue from (

select facs.name, sum(case

when memid = 0 then slots \* facs.guestcost

else slots \* membercost

end) as revenue

from cd.bookings bks

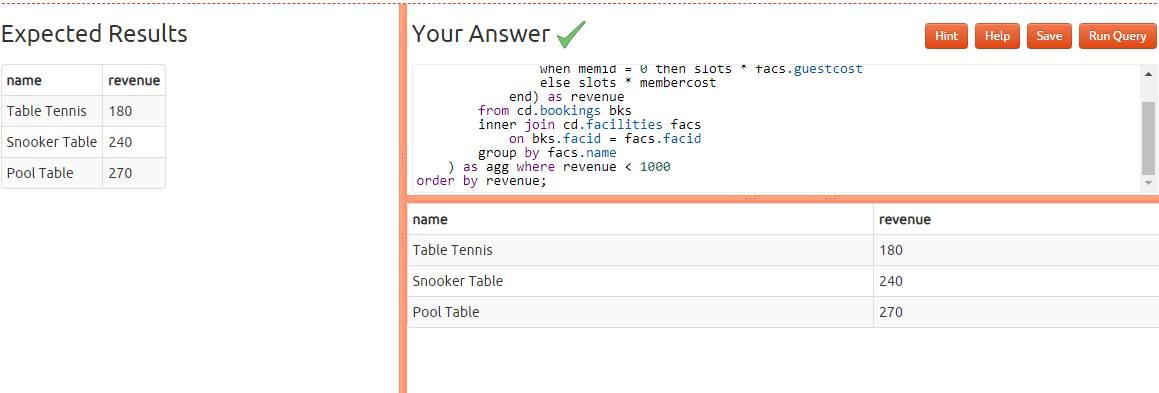
inner join cd.facilities facs

on bks.facid = facs.facid

group by facs.name

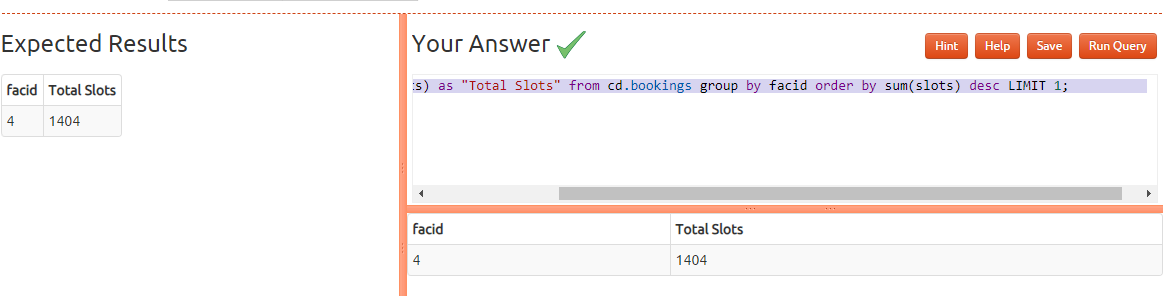
) as agg where revenue < 1000

order by revenue;



* [Output the facility id that has the highest number of slots booked](https://pgexercises.com/questions/aggregates/fachours2.html)

select facid, sum(slots) as "Total Slots" from cd.bookings group by facid order by sum(slots) desc LIMIT 1;



* [List the total slots booked per facility per month, part 2](https://pgexercises.com/questions/aggregates/fachoursbymonth3.html)

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

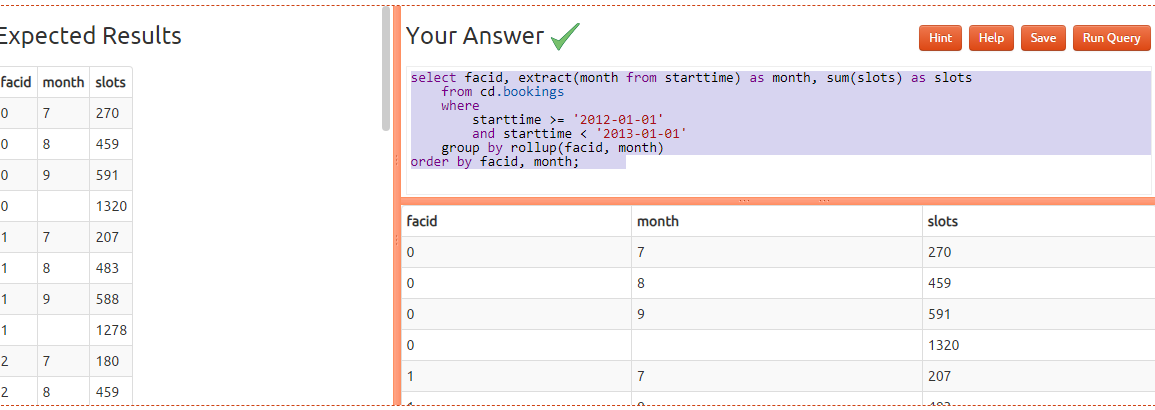
from cd.bookings

where

starttime >= '2012-01-01'

and starttime < '2013-01-01'

group by rollup(facid, month)

order by facid, month; 

* [List the total hours booked per named facility](https://pgexercises.com/questions/aggregates/fachours3.html)

select facs.facid, facs.name,

trim(to\_char(sum(bks.slots)/2.0, '9999999999999999D99')) as "Total Hours"

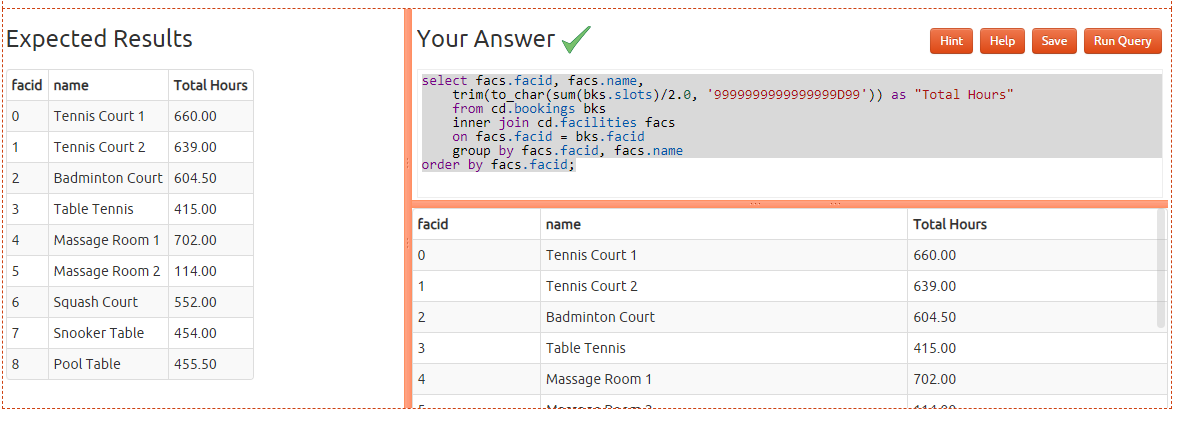
from cd.bookings bks

inner join cd.facilities facs

on facs.facid = bks.facid

group by facs.facid, facs.name

order by facs.facid;



* [List each member's first booking after September 1st 2012](https://pgexercises.com/questions/aggregates/nbooking.html)

select mems.surname, mems.firstname, mems.memid, min(bks.starttime) as starttime

from cd.bookings bks

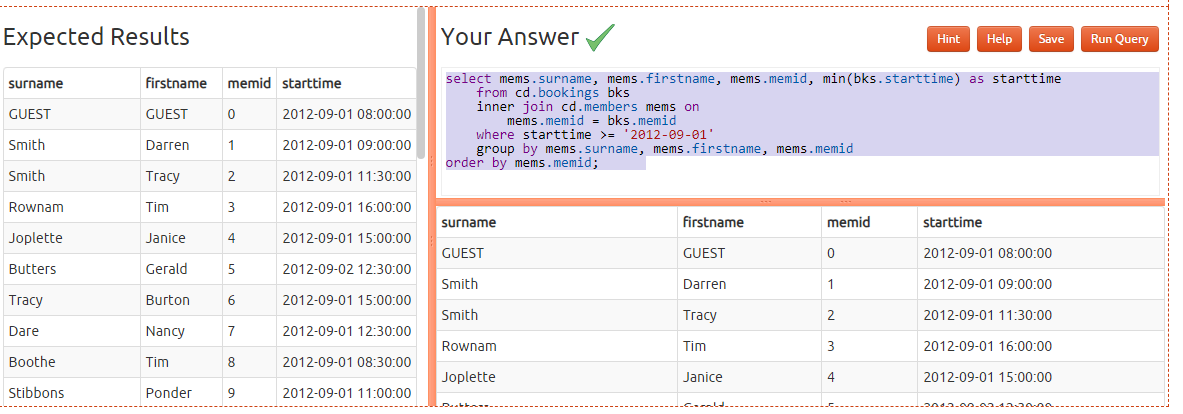
inner join cd.members mems on

mems.memid = bks.memid

where starttime >= '2012-09-01'

group by mems.surname, mems.firstname, mems.memid

order by mems.memid;

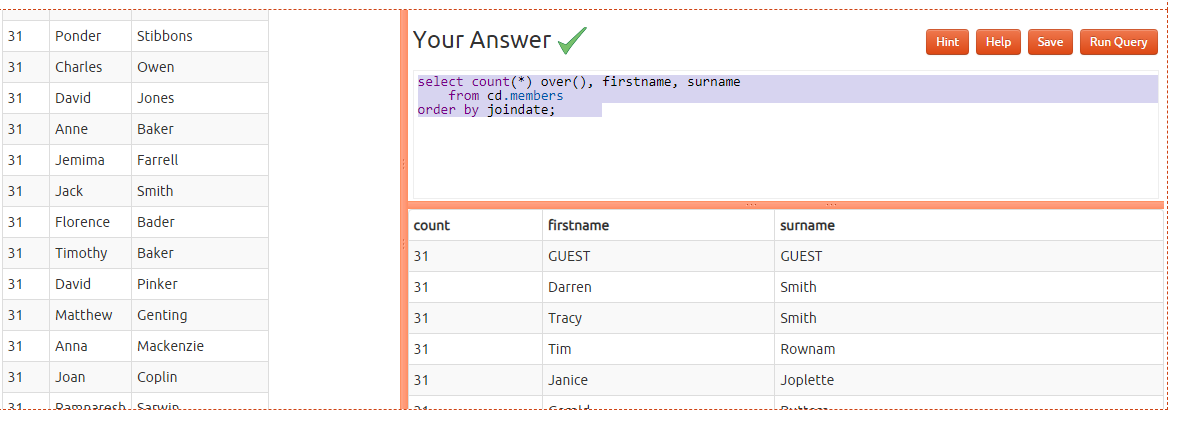


* [Produce a list of member names, with each row containing the total member count](https://pgexercises.com/questions/aggregates/countmembers.html)

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

from cd.members

order by joindate;

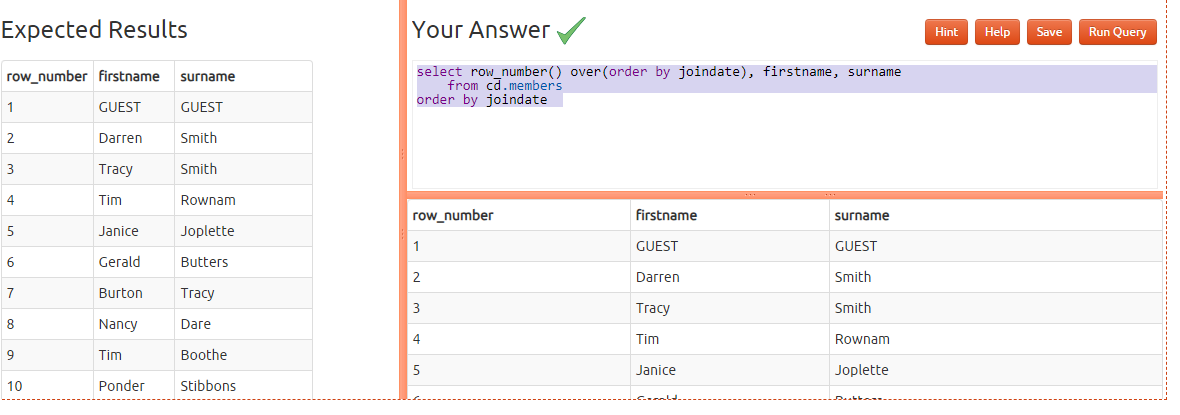


* [Produce a numbered list of members](https://pgexercises.com/questions/aggregates/nummembers.html)

select row\_number() over(order by joindate), firstname, surname

from cd.members

order by joindate

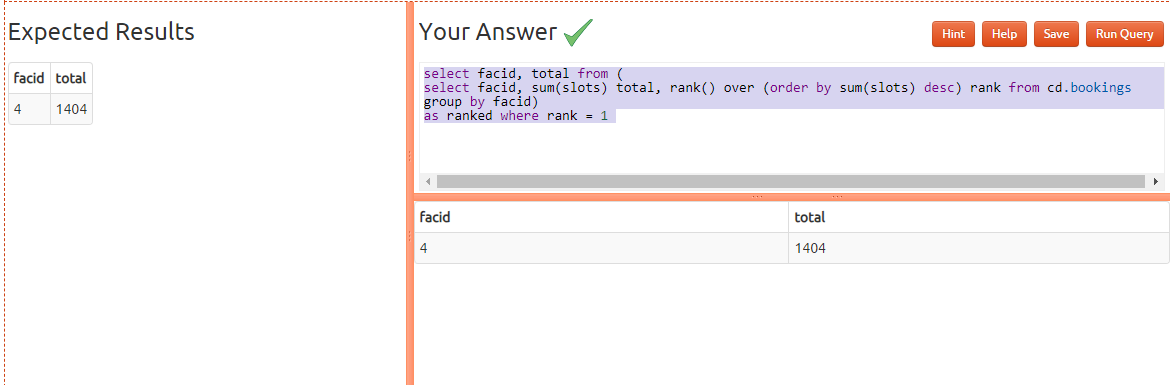


* [Output the facility id that has the highest number of slots booked, again](https://pgexercises.com/questions/aggregates/fachours4.html)

select facid, total from (

select facid, sum(slots) total, rank() over (order by sum(slots) desc) rank from cd.bookings

group by facid)

as ranked where rank = 1 

* [Rank members by (rounded) hours used](https://pgexercises.com/questions/aggregates/rankmembers.html)

select firstname, surname,

((sum(bks.slots)+10)/20)\*10 as hours,

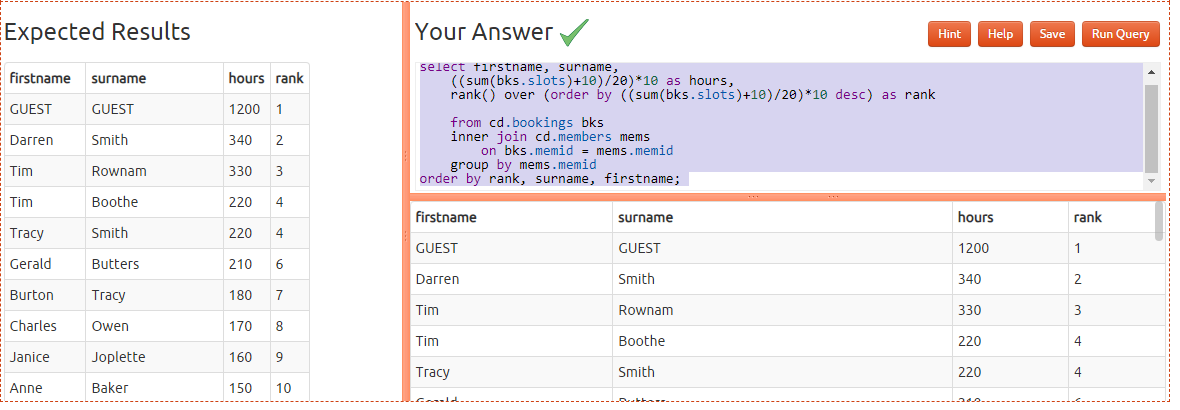
rank() over (order by ((sum(bks.slots)+10)/20)\*10 desc) as rank

from cd.bookings bks

inner join cd.members mems

on bks.memid = mems.memid

group by mems.memid

order by rank, surname, firstname; 

* [Find the top three revenue generating facilities](https://pgexercises.com/questions/aggregates/facrev3.html)

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; 

* [Classify facilities by value](https://pgexercises.com/questions/aggregates/classify.html)

select name, case when class=1 then 'high'

when class=2 then 'average'

else 'low'

end revenue

from (

select facs.name as name, ntile(3) over (order by sum(case

when memid = 0 then slots \* facs.guestcost

else slots \* membercost

end) desc) as class

from cd.bookings bks

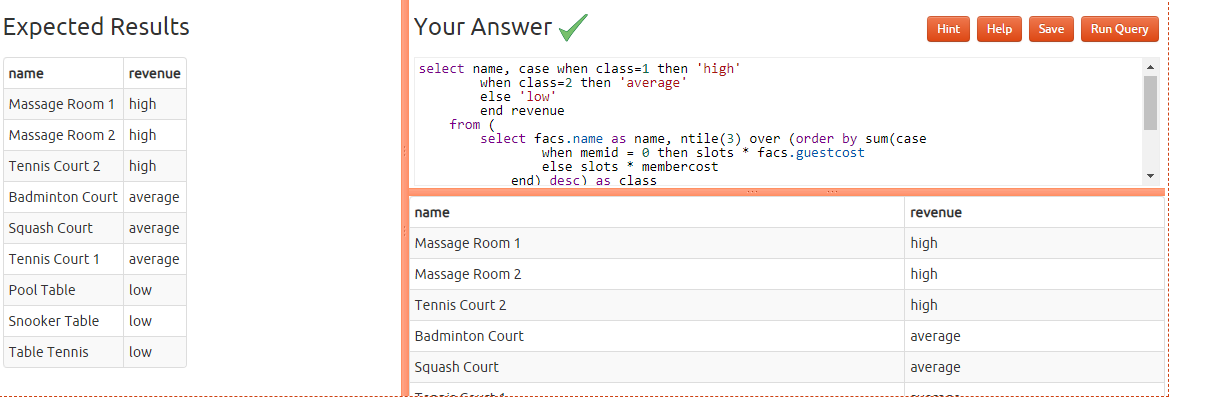
inner join cd.facilities facs

on bks.facid = facs.facid

group by facs.name

) as subq

order by class, name;



* [Calculate the payback time for each facility](https://pgexercises.com/questions/aggregates/payback.html)

select facs.name as name,

facs.initialoutlay/((sum(case

when memid = 0 then slots \* facs.guestcost

else slots \* membercost

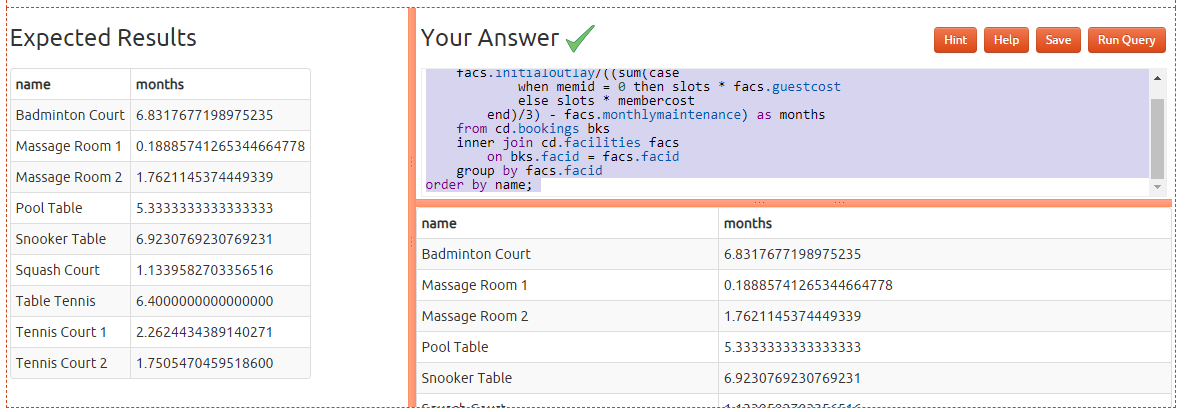
end)/3) - facs.monthlymaintenance) as months

from cd.bookings bks

inner join cd.facilities facs

on bks.facid = facs.facid

group by facs.facid

order by name; 

* [Calculate a rolling average of total revenue](https://pgexercises.com/questions/aggregates/rollingavg.html)

select dategen.date,

(

-- correlated subquery that, for each day fed into it,

-- finds the average revenue for the last 15 days

select sum(case

when memid = 0 then slots \* facs.guestcost

else slots \* membercost

end) as rev

from cd.bookings bks

inner join cd.facilities facs

on bks.facid = facs.facid

where bks.starttime > dategen.date - interval '14 days'

and bks.starttime < dategen.date + interval '1 day'

)/15 as revenue

from

(

-- generates a list of days in august

select cast(generate\_series(timestamp '2012-08-01',

'2012-08-31','1 day') as date) as date

) as dategen

order by dategen.date; 