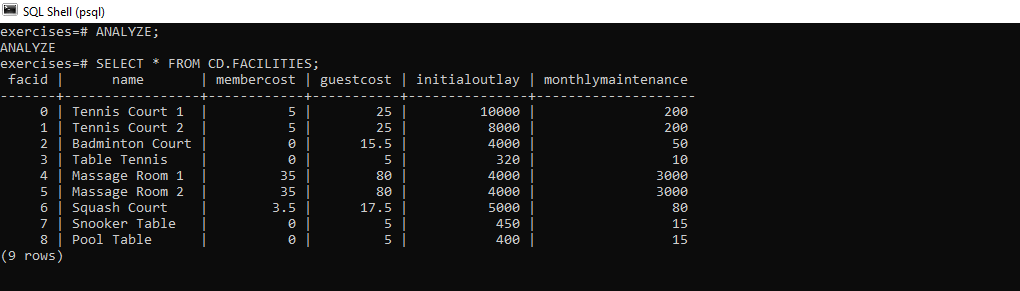
BASICS

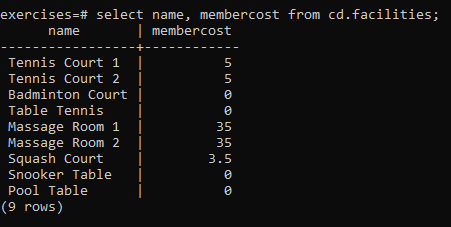
1 [Retrieve everything from a table](https://pgexercises.com/questions/basic/selectall.html)

SELECT \* FROM CD.FACILITIES;



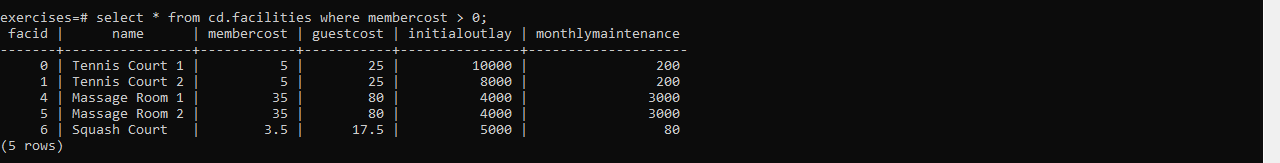
* [Retrieve specific columns from a table](https://pgexercises.com/questions/basic/selectspecific.html)

select name, membercost from cd.facilities;



* [Control which rows are retrieved](https://pgexercises.com/questions/basic/where.html)

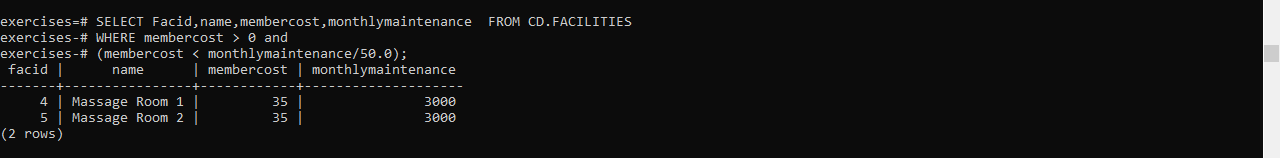
select \* from cd.facilities where membercost > 0;

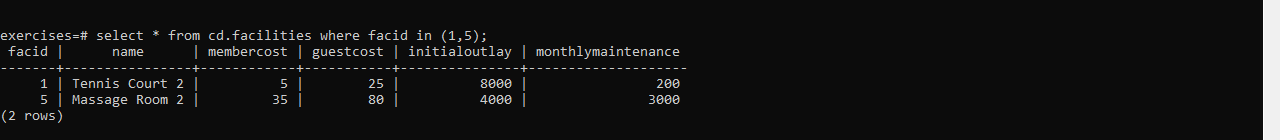


SELECT Facid,name,membercost,monthlymaintenance FROM CD.FACILITIES

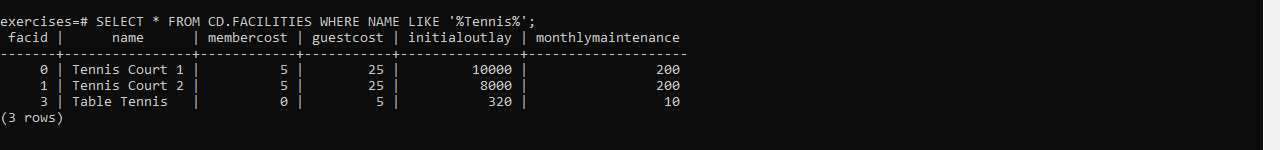
WHERE membercost > 0 and

(membercost < monthlymaintenance/50.0);

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



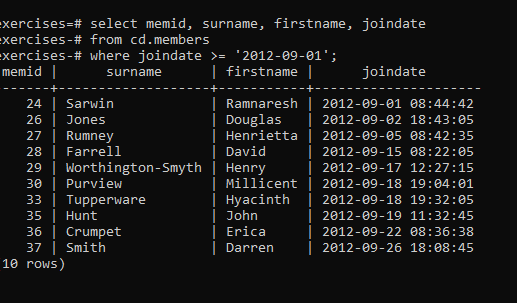
SELECT \* FROM CD.FACILITIES WHERE NAME LIKE '%Tennis%';



select memid, surname, firstname, joindate

from cd.members

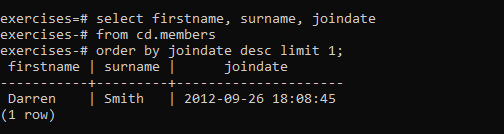
where joindate >= '2012-09-01';



select firstname, surname, joindate

from cd.members

order by joindate desc limit 1;

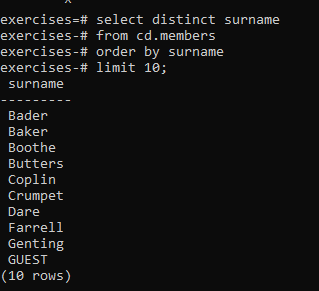


select distinct surname

from cd.members

order by surname

limit 10;



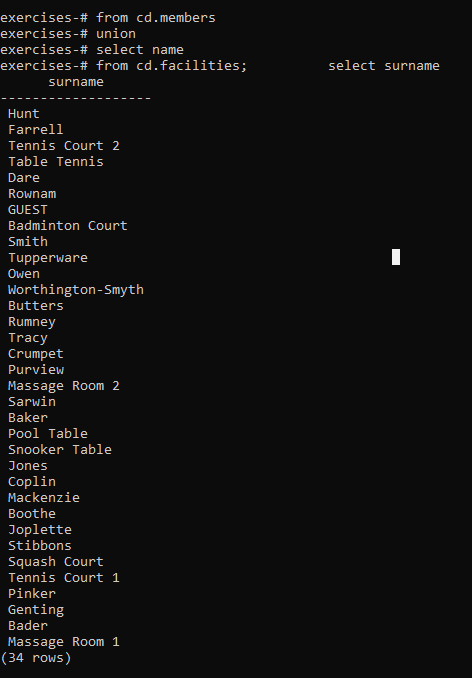
select surname

from cd.members

union

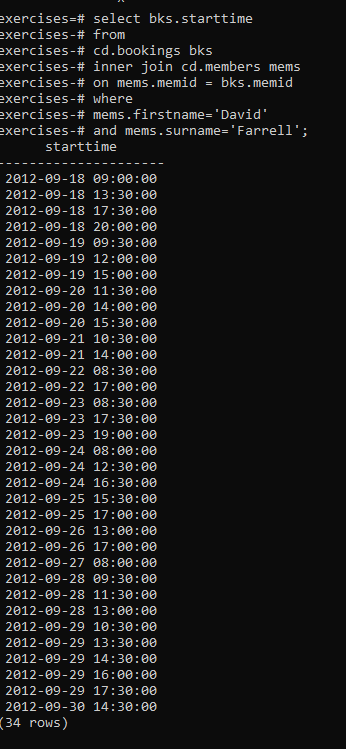
select name

from cd.facilities;

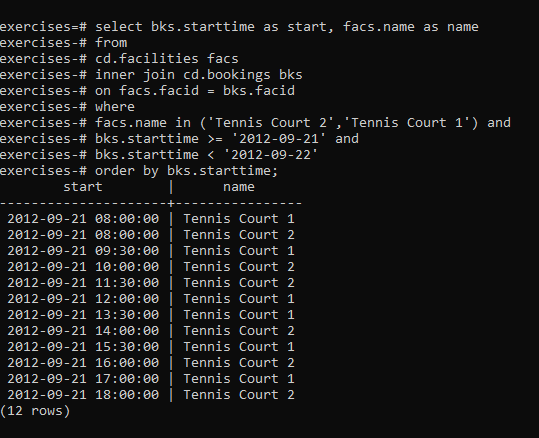


Joins and subquerry

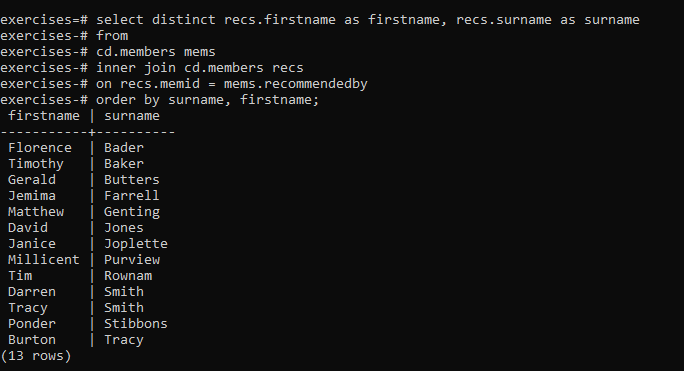
How can you produce a list of the start times for bookings by members named 'David Farrell'?



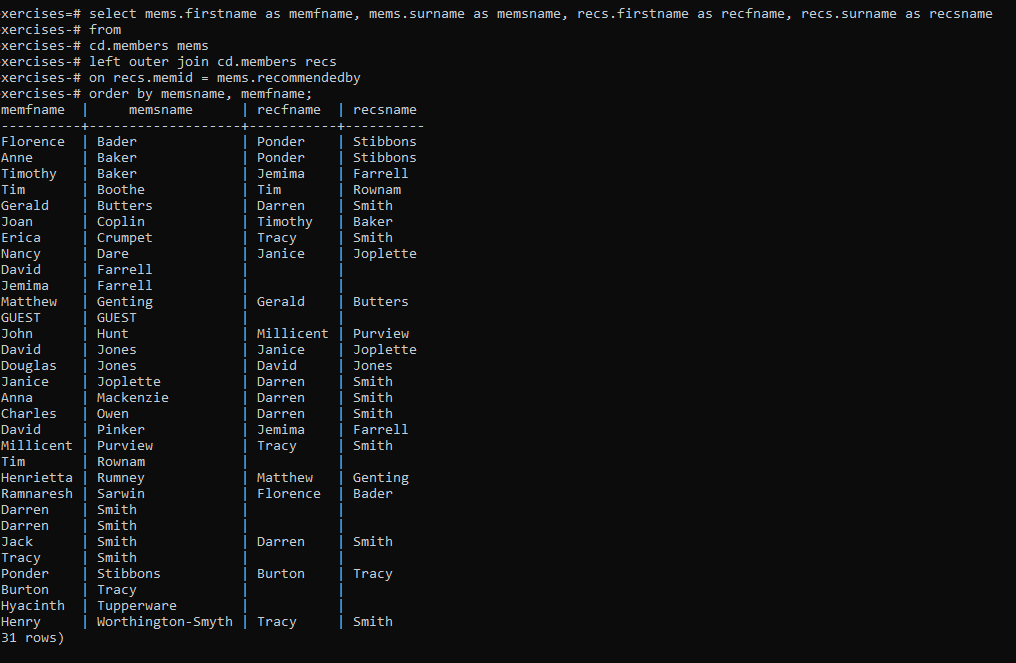
How can you produce a list of the start times for bookings for tennis courts, for the date '2012-09-21'? Return a list of start time and facility name pairings, ordered by the time.



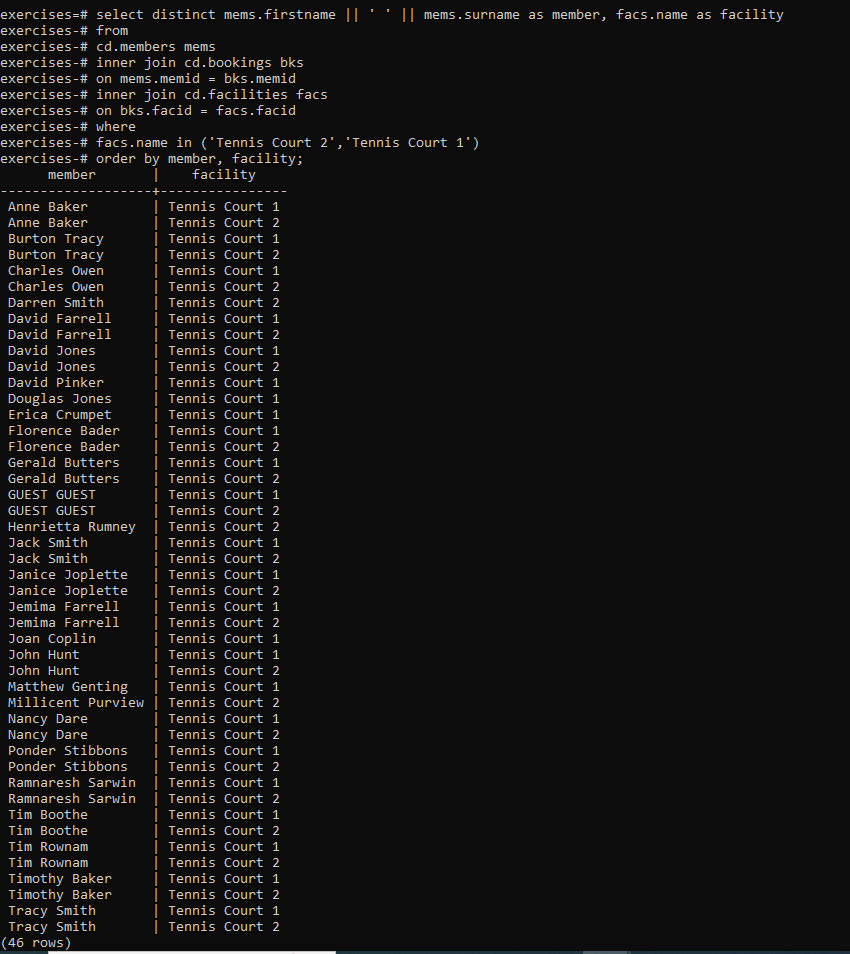
How can you output a list of all members who have recommended another member? Ensure that there are no duplicates in the list, and that results are ordered by (surname, firstname).



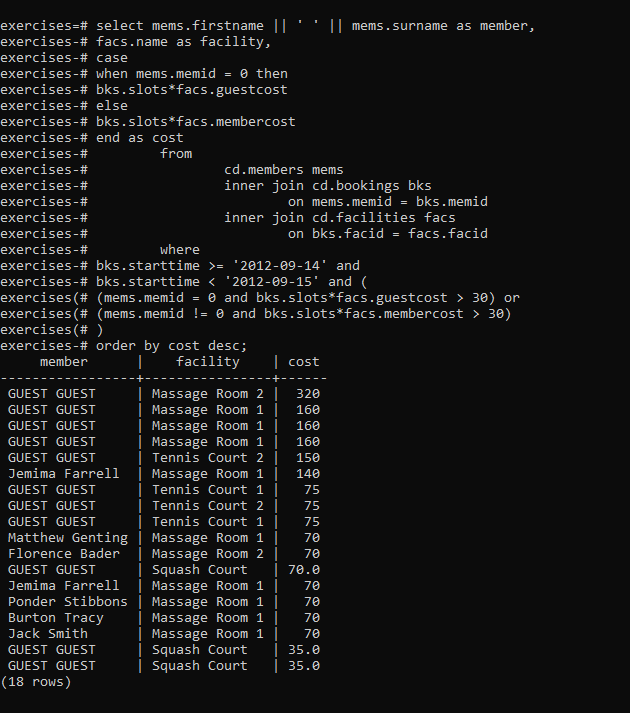
How can you output a list of all members, including the individual who recommended them (if any)? Ensure that results are ordered by (surname, firstname).



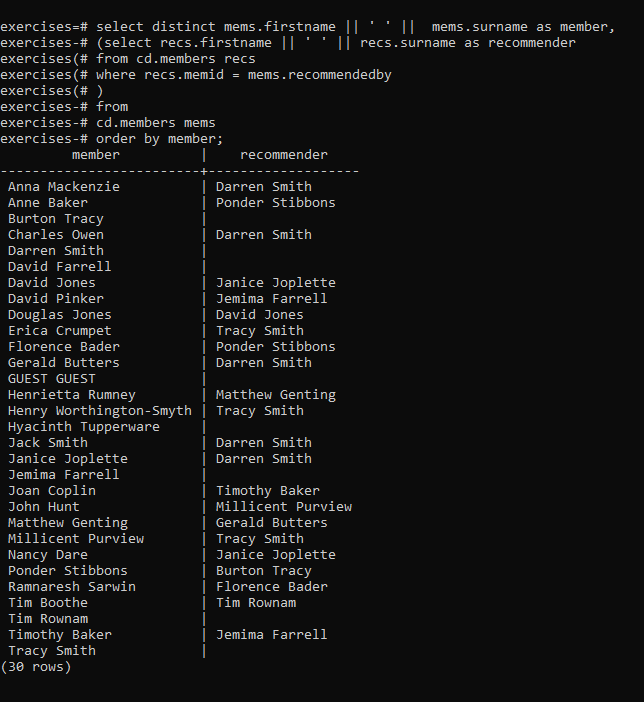
How can you produce a list of all members who have used a tennis court? Include in your output the name of the court, and the name of the member formatted as a single column. Ensure no duplicate data, and order by the member name followed by the facility name.



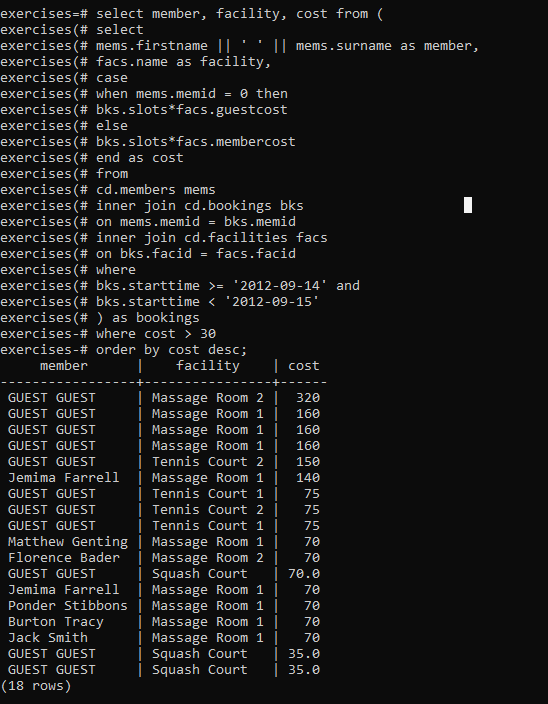
How can you produce a list of bookings on the day of 2012-09-14 which will cost the member (or guest) more than $30? Remember that guests have different costs to members (the listed costs are per half-hour 'slot'), and the guest user is always ID 0. Include in your output the name of the facility, the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



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.



Produce a list of costly bookings, using a subquery



**Modifydata**

[**Insert some data into a table**](https://pgexercises.com/questions/updates/insert.html)

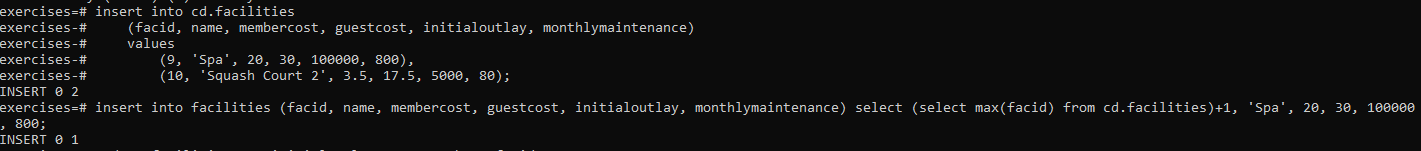
insert into facilities (facid, name, membercost, guestcost, initialoutlay, monthlymaintenance) values (9, 'Spa', 20, 30, 100000, 800);

**OUTPUT: **

[**Insert multiple rows of data into a table**](https://pgexercises.com/questions/updates/insert2.html)

insert into facilities (facid, name, membercost, guestcost, initialoutlay, monthlymaintenance) values (9, 'Spa', 20, 30, 100000, 800), (10, 'Squash Court 2', 3.5, 17.5, 5000, 80);

**OUTPUT:**

****

[**Insert calculated data into a table**](https://pgexercises.com/questions/updates/insert3.html)

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

**OUTPUT:**

****

[**Update some existing data**](https://pgexercises.com/questions/updates/update.html)

**Query:** update facilities set initialoutlay = 10000 where facid = 1; **OUTPUT:**

****

[**Update multiple rows and columns at the same time**](https://pgexercises.com/questions/updates/updatemultiple.html)

update facilities set membercost = 6, guestcost = 30 where facid in (0,1);

**OUTPUT: **

[**Update a row based on the contents of another row**](https://pgexercises.com/questions/updates/updatecalculated.html)

update facilities facs set membercost = (select membercost \* 1.1 from facilities where facid = 0), guestcost = (select guestcost \* 1.1 from facilities where facid = 0) where facs.facid = 1;

**OUTPUT: **

[**Delete all bookings**](https://pgexercises.com/questions/updates/delete.html)

delete from bookings;

**OUTPUT: **

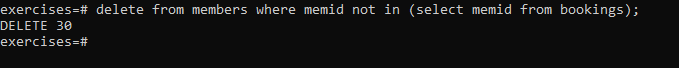
[**Delete a member from the cd.members table**](https://pgexercises.com/questions/updates/deletewh.html)

delete from members where memid = 37;

**OUTPUT: **

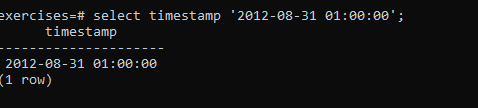
[**Delete based on a subquery**](https://pgexercises.com/questions/updates/deletewh2.html)

delete from members where memid not in (select memid from bookings);

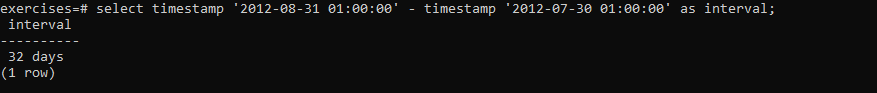
****

DATE

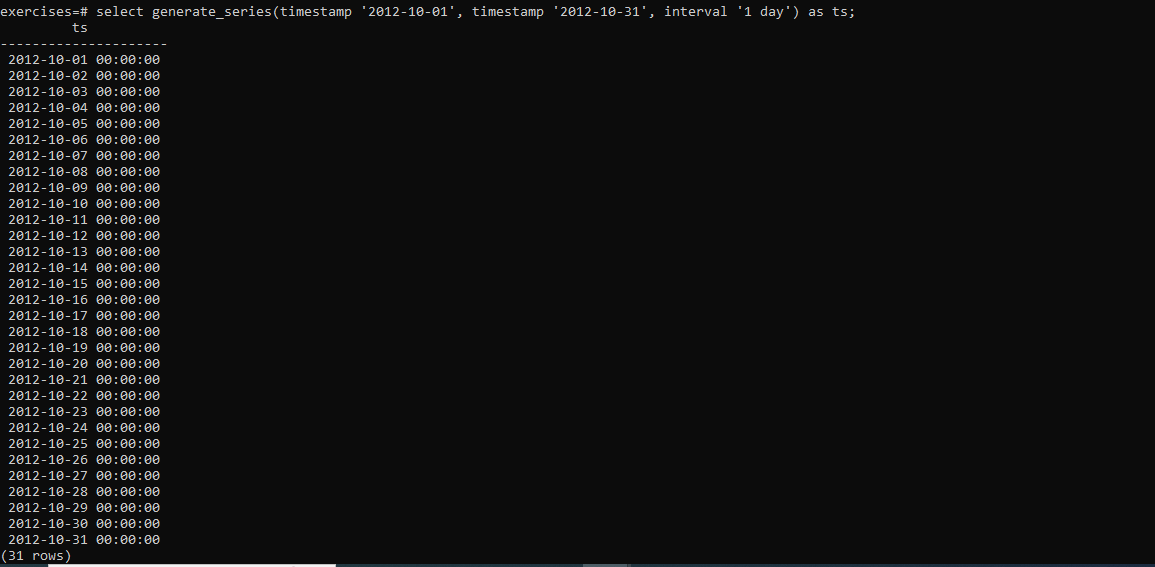
Produce a timestamp for 1 a.m. on the 31st of August 2012.



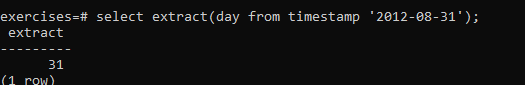
Find the result of subtracting the timestamp '2012-07-30 01:00:00' from the timestamp '2012-08-31 01:00:00'



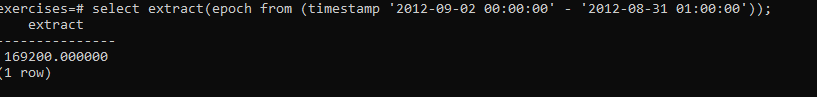
Produce a list of all the dates in October 2012. They can be output as a timestamp (with time set to midnight) or a date.



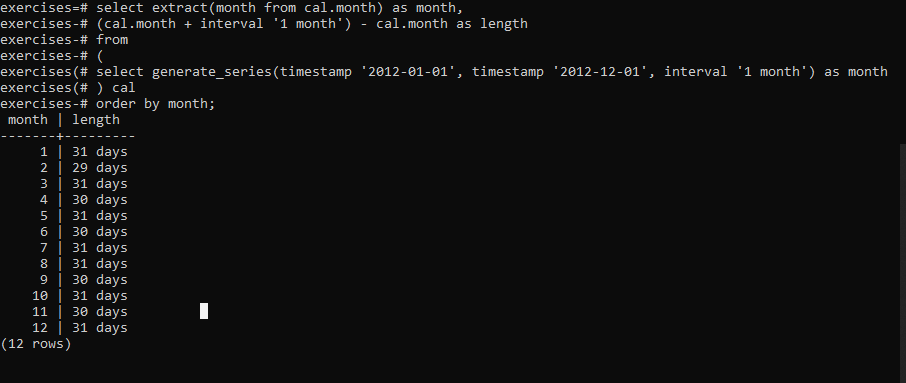
Get the day of the month from the timestamp '2012-08-31' as an integer.



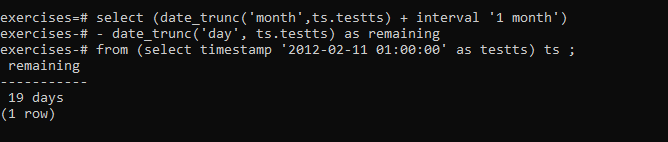
Work out the number of seconds between the timestamps '2012-08-31 01:00:00' and '2012-09-02 00:00:00'



For each month of the year in 2012, output the number of days in that month. Format the output as an integer column containing the month of the year, and a second column containing an interval data type.



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.



Return a count of bookings for each month, sorted by month

select date\_trunc('month', starttime) as month, count(\*)

from cd.bookings

group by month

order by month;

output:

| **month** | **count** |
| --- | --- |
| 2012-07-01 00:00:00 | 658 |
| 2012-08-01 00:00:00 | 1472 |
| 2012-09-01 00:00:00 | 1913 |
| 2013-01-01 00:00:00 | 1 |

Work out the utilisation percentage for each facility by month, sorted by name and month, rounded to 1 decimal place. Opening time is 8am, closing time is 8.30pm. You can treat every month as a full month, regardless of if there were some dates the club was not open.

select name, month,

round((100\*slots)/

cast(

25\*(cast((month + interval '1 month') as date)

- cast (month as date)) as numeric),1) as utilisation

from (

select facs.name as name, date\_trunc('month', starttime) as month, sum(slots) as slots

from cd.bookings bks

inner join cd.facilities facs

on bks.facid = facs.facid

group by facs.facid, month

) as inn

order by name, month

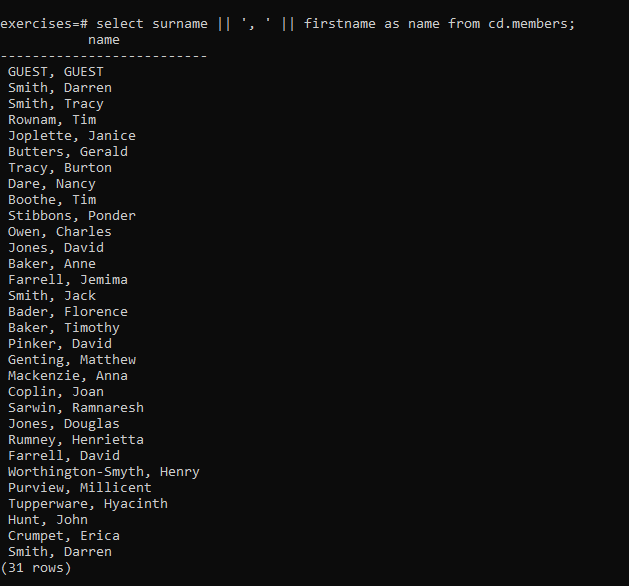
ouitput;

| **name** | **month** | **utilisation** |
| --- | --- | --- |
| Badminton Court | 2012-07-01 00:00:00 | 23.2 |
| Badminton Court | 2012-08-01 00:00:00 | 59.2 |
| Badminton Court | 2012-09-01 00:00:00 | 76.0 |
| Massage Room 1 | 2012-07-01 00:00:00 | 34.1 |
| Massage Room 1 | 2012-08-01 00:00:00 | 63.5 |
| Massage Room 1 | 2012-09-01 00:00:00 | 86.4 |
| Massage Room 2 | 2012-07-01 00:00:00 | 3.1 |
| Massage Room 2 | 2012-08-01 00:00:00 | 10.6 |
| Massage Room 2 | 2012-09-01 00:00:00 | 16.3 |
| Pool Table | 2012-07-01 00:00:00 | 15.1 |
| Pool Table | 2012-08-01 00:00:00 | 41.5 |
| Pool Table | 2012-09-01 00:00:00 | 62.8 |
| Pool Table | 2013-01-01 00:00:00 | 0.1 |
| Snooker Table | 2012-07-01 00:00:00 | 20.1 |
| Snooker Table | 2012-08-01 00:00:00 | 42.1 |
| Snooker Table | 2012-09-01 00:00:00 | 56.8 |
| Squash Court | 2012-07-01 00:00:00 | 21.2 |
| Squash Court | 2012-08-01 00:00:00 | 51.6 |
| Squash Court | 2012-09-01 00:00:00 | 72.0 |
| Table Tennis | 2012-07-01 00:00:00 | 13.4 |
| Table Tennis | 2012-08-01 00:00:00 | 39.2 |
| Table Tennis | 2012-09-01 00:00:00 | 56.3 |
| Tennis Court 1 | 2012-07-01 00:00:00 | 34.8 |
| Tennis Court 1 | 2012-08-01 00:00:00 | 59.2 |
| Tennis Court 1 | 2012-09-01 00:00:00 | 78.8 |
| Tennis Court 2 | 2012-07-01 00:00:00 | 26.7 |
| Tennis Court 2 | 2012-08-01 00:00:00 | 62.3 |
| Tennis Court 2 | 2012-09-01 00:00:00 | 78.4 |

**String Operations**

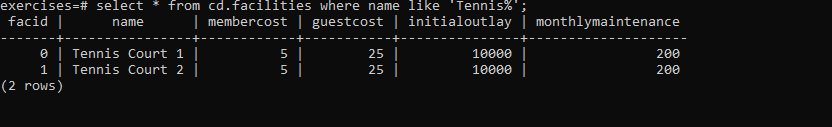
[**Format the names of members**](https://pgexercises.com/questions/string/concat.html)

select surname || ', ' || firstname as name from cd.members

**output**

Find facilities by a name prefix

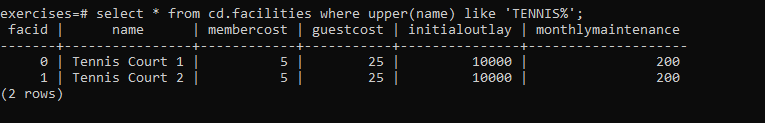
select \* from cd.facilities where name like 'Tennis%';



Perform a case-insensitive search

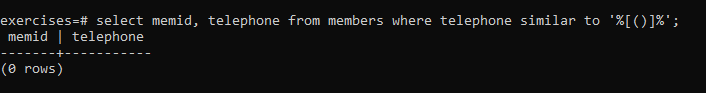
select \* from cd.facilities where upper(name) like 'TENNIS%';

output



Find telephone numbers with parentheses

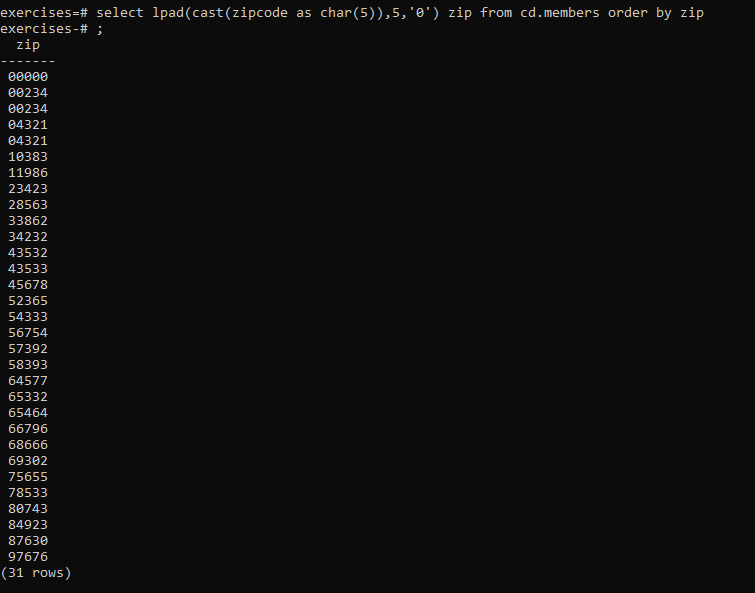
select memid, telephone from cd.members where telephone similar to '%[()]%';

**OUTPUT: **

Pad zip codes with leading zeroes

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

output

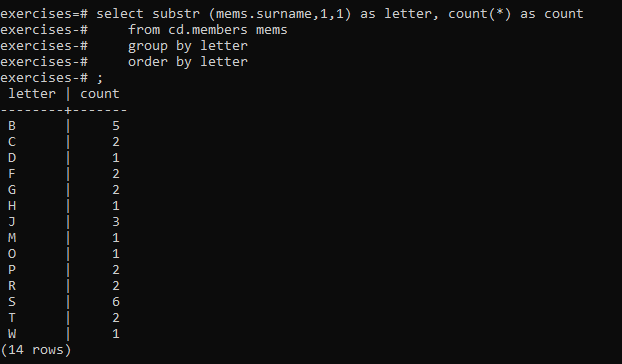
****

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

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

**from cd.members mems**

**group by letter**

**order by letter; OUTPUT: **

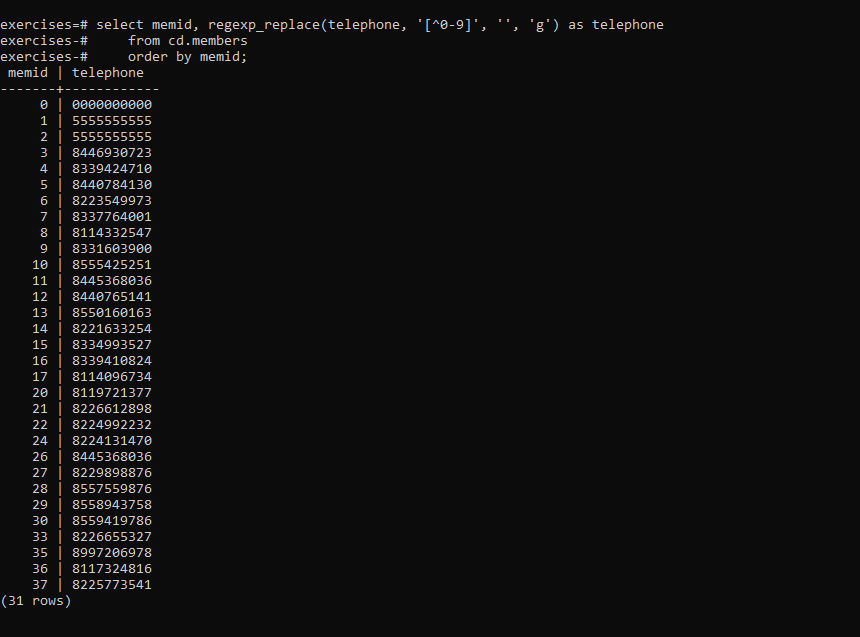
[**Clean up telephone numbers**](https://pgexercises.com/questions/string/translate.html)

select memid, regexp\_replace(telephone, '[^0-9]', '', 'g') as telephone

from cd.members

order by memid;

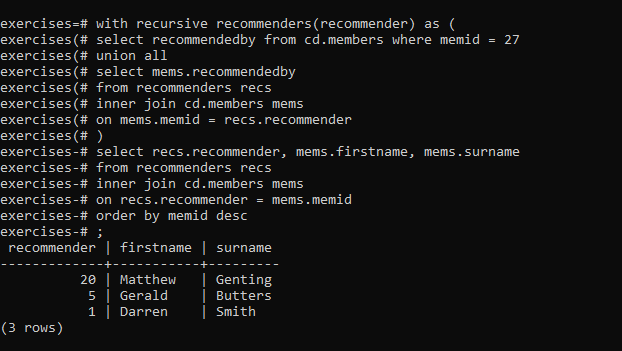
**OUTPUT:**



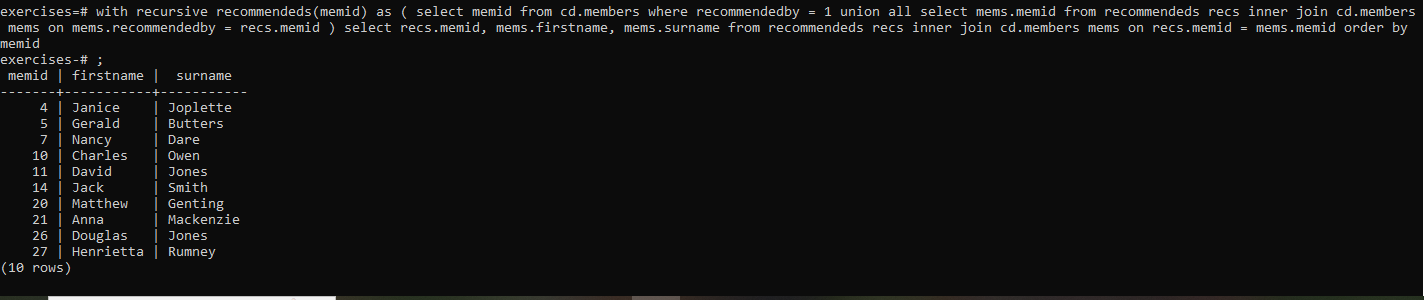
[Recursive](https://pgexercises.com/questions/recursive/)

* [Find the upward recommendation chain for member ID 27](https://pgexercises.com/questions/recursive/getupward.html)

**with** recursive recommenders(recommender) **as** ( **select** recommendedby **from** cd.members **where** memid = 27 **union** **all** **select** mems.recommendedby **from** recommenders recs **inner** **join** cd.members mems **on** mems.memid = recs.recommender ) **select** recs.recommender, mems.firstname, mems.surname **from** recommenders recs **inner** **join** cd.members mems **on** recs.recommender = mems.memid **order** **by** memid **desc;**

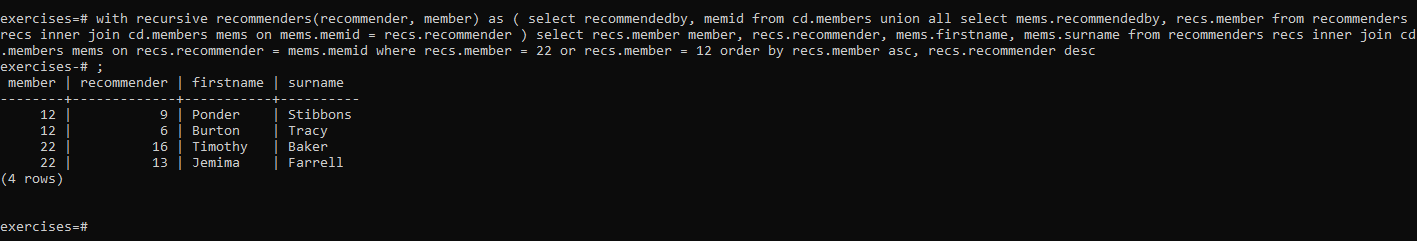
****

* [Find the downward recommendation chain for member ID 1](https://pgexercises.com/questions/recursive/getdownward.html)
* **with** recursive recommendeds(memid) **as** ( **select** memid **from** cd.members **where** recommendedby = 1 **union** **all** **select** mems.memid **from** recommendeds recs **inner** **join** cd.members mems **on** mems.recommendedby = recs.memid ) **select** recs.memid, mems.firstname, mems.surname **from** recommendeds recs **inner** **join** cd.members mems **on** recs.memid = mems.memid **order** **by** memid;



* [Produce a CTE that can return the upward recommendation chain for any member](https://pgexercises.com/questions/recursive/getupwardall.html)

**with** recursive recommenders(recommender, member) **as** ( **select** recommendedby, memid **from** cd.members **union** **all** **select** mems.recommendedby, recs.member **from** recommenders recs **inner** **join** cd.members mems **on** mems.memid = recs.recommender ) **select** recs.member member, recs.recommender, mems.firstname, mems.surname **from** recommenders recs **inner** **join** cd.members mems **on** recs.recommender = mems.memid **where** recs.member = 22 **or** recs.member = 12 **order** **by** recs.member **asc**, recs.recommender **desc;**

****