Name :Basil Ahamed

Assignment 2 SQL

BASIC

1) Retrieve everything from a table:

Query:

```
select *
from facilities;
```

```
exercises=# select *
exercises-# from facilities;
 facid |      name       | membercost | guestcost | initialoutlay | monthlymaintenance
-------+-----------------+------------+-----------+---------------+--------------------
     0 | Tennis Court 1  |          5 |        25 |         10000 |                200
     1 | Tennis Court 2  |          5 |        25 |          8000 |                200
     2 | Badminton Court |          0 |      15.5 |          4000 |                 50
     3 | Table Tennis    |          0 |         5 |           320 |                 10
     4 | Massage Room 1  |         35 |        80 |          4000 |               3000
     5 | Massage Room 2  |         35 |        80 |          4000 |               3000
     6 | Squash Court    |        3.5 |      17.5 |          5000 |                 80
     7 | Snooker Table   |          0 |         5 |           450 |                 15
     8 | Pool Table      |          0 |         5 |           400 |                 15
(9 rows)
```

2) Retrieve specific columns from a table

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;
```

```
exercises=# select name, membercost
exercises-# from facilities;
      name       | membercost
-----------------+------------
 Tennis Court 1  |          5
 Tennis Court 2  |          5
 Badminton Court |          0
 Table Tennis    |          0
 Massage Room 1  |         35
 Massage Room 2  |         35
 Squash Court    |        3.5
 Snooker Table   |          0
 Pool Table      |          0
(9 rows)
```

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;

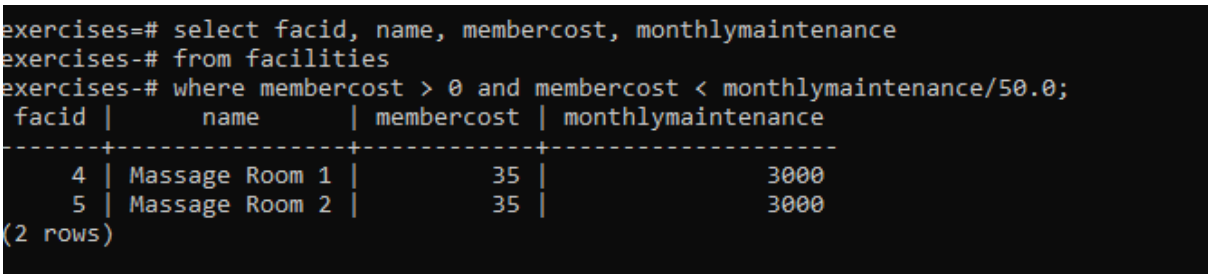
```
exercises=# select *
exercises-# from facilities
exercises-# where membercost > 0;
 facid |      name      | membercost | guestcost | initialoutlay | monthlymaintenance
-------+----------------+------------+-----------+---------------+--------------------
     0 | Tennis Court 1 |          5 |        25 |         10000 |                200
     1 | Tennis Court 2 |          5 |        25 |          8000 |                200
     4 | Massage Room 1 |         35 |        80 |          4000 |               3000
     5 | Massage Room 2 |         35 |        80 |          4000 |               3000
     6 | Squash Court   |        3.5 |      17.5 |          5000 |                 80
(5 rows)
```

4) Control which rows are retrieved - part 2

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;
```



```
exercises=# select facid, name, membercost, monthlymaintenance
exercises-# from facilities
exercises-# where membercost > 0 and membercost < monthlymaintenance/50.0;
 facid |      name      | membercost | monthlymaintenance
-------+----------------+------------+--------------------
     4 | Massage Room 1 |         35 |               3000
     5 | Massage Room 2 |         35 |               3000
(2 rows)
```

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%';
```

```
exercises=# select *
exercises-# from facilities
exercises-# where name like '%Tennis%';
 facid |      name      | membercost | guestcost | initialoutlay | monthlymaintenance
-------+----------------+------------+-----------+---------------+--------------------
     0 | Tennis Court 1 |          5 |        25 |         10000 |                200
     1 | Tennis Court 2 |          5 |        25 |          8000 |                200
     3 | Table Tennis   |          0 |         5 |           320 |                 10
(3 rows)
```

6) Matching against multiple possible values

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);

```
exercises=# select *
exercises-# from facilities
exercises-# where facid in(5,1);
 facid |      name      | membercost | guestcost | initialoutlay | monthlymaintenance
-------+----------------+------------+-----------+---------------+--------------------
     1 | Tennis Court 2 |          5 |        25 |          8000 |                200
     5 | Massage Room 2 |         35 |        80 |          4000 |               3000
(2 rows)
```

7) Classify results into buckets

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;

```
exercises=# select name,
exercises-# case
exercises-# when monthlymaintenance > 100 then 'expensive' else 'cheap'
exercises-# end as cost
exercises-# from facilities;
      name       |   cost
-----------------+-----------
 Tennis Court 1  | expensive
 Tennis Court 2  | expensive
 Badminton Court | cheap
 Table Tennis    | cheap
 Massage Room 1  | expensive
 Massage Room 2  | expensive
 Squash Court    | cheap
 Snooker Table   | cheap
 Pool Table      | cheap
(9 rows)
```

8) Working with dates

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' ;

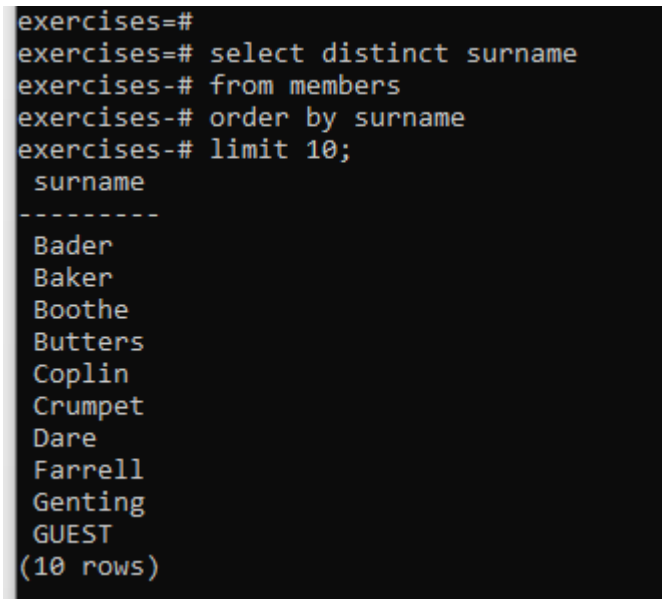
```
exercises=# select memid,  surname, firstname, joindate
exercises-# from members
exercises-# where joindate > '2012-09-01' ;
 memid |      surname      | firstname |      joindate
-------+-------------------+-----------+---------------------
    24 | Sarwin            | Ramnaresh | 2012-09-01 08:44:42
    26 | Jones             | Douglas   | 2012-09-02 18:43:05
    27 | Rumney            | Henrietta | 2012-09-05 08:42:35
    28 | Farrell           | David     | 2012-09-15 08:22:05
    29 | Worthington-Smyth | Henry     | 2012-09-17 12:27:15
    30 | Purview           | Millicent | 2012-09-18 19:04:01
    33 | Tupperware        | Hyacinth  | 2012-09-18 19:32:05
    35 | Hunt              | John      | 2012-09-19 11:32:45
    36 | Crumpet           | Erica     | 2012-09-22 08:36:38
    37 | Smith             | Darren    | 2012-09-26 18:08:45
(10 rows)
```

9) Removing duplicates, and ordering

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 queries

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;
```

```
exercises=# select surname
exercises-# from members
exercises-# union
exercises-# select name
exercises-# from facilities;
      surname
------------------
 Hunt
 Farrell
 Tennis Court 2
 Table Tennis
 Dare
 Rownam
 GUEST
 Badminton Court
 Smith
 Tupperware
 Owen
 Worthington-Smyth
 Butters
 Rumney
 Tracy
 Crumpet
 Purview
 Massage Room 2
 Sarwin
 Baker
 Pool Table
 Snooker Table
 Jones
 Coplin
 Mackenzie
 Boothe
 Joplette
-- More  --
```

11)Simple aggregation

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;

```
exercises=#
exercises=# select joindate as latest
exercises-# from members
exercises-# order by latest desc
exercises-# limit 1;
       latest
---------------------
 2012-09-26 18:08:45
(1 row)


exercises=#
```

12) More aggregation

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;

```
exercises=# select firstname, surname, joindate
exercises-# from members
exercises-# order by joindate desc
exercises-# limit 1;
 firstname | surname |      joindate
-----------+---------+---------------------
 Darren    | Smith   | 2012-09-26 18:08:45
(1 row)
```

JOINS AND SUBQUARY

1)Retrieve the start times of members' bookings

Question

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

```
exercises=# select starttime
exercises-# from bookings
exercises-# where memid = (select memid
exercises(#   from members
exercises(#   where firstname = 'David' and surname= 'Farrell' );
      starttime
---------------------
 2012-09-18 09:00:00
 2012-09-18 13:30:00
 2012-09-18 17:30:00
 2012-09-18 20:00:00
 2012-09-19 09:30:00
 2012-09-19 12:00:00
 2012-09-19 15:00:00
 2012-09-20 11:30:00
 2012-09-20 14:00:00
 2012-09-20 15:30:00
 2012-09-21 10:30:00
 2012-09-21 14:00:00
 2012-09-22 08:30:00
 2012-09-22 17:00:00
 2012-09-23 08:30:00
 2012-09-23 17:30:00
 2012-09-23 19:00:00
 2012-09-24 08:00:00
 2012-09-24 12:30:00
 2012-09-24 16:30:00
 2012-09-25 15:30:00
 2012-09-25 17:00:00
 2012-09-26 13:00:00
 2012-09-26 17:00:00
 2012-09-27 08:00:00
 2012-09-28 09:30:00
 2012-09-28 11:30:00
 2012-09-28 13:00:00
 2012-09-29 10:30:00
 2012-09-29 13:30:00
 2012-09-29 14:30:00
 2012-09-29 16:00:00
 2012-09-29 17:30:00
 2012-09-30 14:30:00
(34 rows)
```

2) Work out the start times of bookings for tennis courts

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;

```
exercises=# select b.starttime as start, f.name
exercises-# from bookings b inner join
exercises-# facilities f using(facid)
exercises-# where f.name like 'Tennis Court%' and b.starttime between '2012-09-21' and '2012-09-22'
exercises-# order by b.starttime;
        start        |      name
---------------------+---------------
 2012-09-21 08:00:00 | Tennis Court 1
 2012-09-21 08:00:00 | Tennis Court 2
 2012-09-21 09:30:00 | Tennis Court 1
 2012-09-21 10:00:00 | Tennis Court 2
 2012-09-21 11:30:00 | Tennis Court 2
 2012-09-21 12:00:00 | Tennis Court 1
 2012-09-21 13:30:00 | Tennis Court 1
 2012-09-21 14:00:00 | Tennis Court 2
 2012-09-21 15:30:00 | Tennis Court 1
 2012-09-21 16:00:00 | Tennis Court 2
 2012-09-21 17:00:00 | Tennis Court 1
 2012-09-21 18:00:00 | Tennis Court 2
(12 rows)
```

3)

Produce a list of all members who have recommended another member

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 ;
```

```
exercises=# select distinct r.firstname  , r.surname
exercises-# from members f
exercises-# inner join
exercises-# members r
exercises-# on  r.memid = f.recommendedby
exercises-# order by surname, firstname ;
 firstname | surname
-----------+----------
 Florence  | Bader
 Timothy   | Baker
 Gerald    | Butters
 Jemima    | Farrell
 Matthew   | Genting
 David     | Jones
 Janice    | Joplette
 Millicent | Purview
 Tim       | Rownam
 Darren    | Smith
 Tracy     | Smith
 Ponder    | Stibbons
 Burton    | Tracy
(13 rows)


exercises=# z
```

4) Produce a list of all members, along with their recommender

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;
```

```
exercises=# select m.firstname as memfname, m.surname as memsname,
exercises-# r.firstname as recfname, r.surname as recsname
exercises-# from members r  right outer join
exercises-# members m on r.memid = m.recommendedby
exercises-# order by m.surname, m.firstname;
 memfname  |     memsname      | recfname  | recsname
-----------+-------------------+-----------+----------
 Florence  | Bader             | Ponder    | Stibbons
 Anne      | Baker             | Ponder    | Stibbons
 Timothy   | Baker             | Jemima    | Farrell
 Tim       | Boothe            | Tim       | Rownam
 Gerald    | Butters           | Darren    | Smith
 Joan      | Coplin            | Timothy   | Baker
 Erica     | Crumpet           | Tracy     | Smith
 Nancy     | Dare              | Janice    | Joplette
 David     | Farrell           |           |
 Jemima    | Farrell           |           |
 Matthew   | Genting           | Gerald    | Butters
 GUEST     | GUEST             |           |
 John      | Hunt              | Millicent | Purview
 David     | Jones             | Janice    | Joplette
 Douglas   | Jones             | David     | Jones
 Janice    | Joplette          | Darren    | Smith
 Anna      | Mackenzie         | Darren    | Smith
 Charles   | Owen              | Darren    | Smith
 David     | Pinker            | Jemima    | Farrell
 Millicent | Purview           | Tracy     | Smith
 Tim       | Rownam            |           |
 Henrietta | Rumney            | Matthew   | Genting
 Ramnaresh | Sarwin            | Florence  | Bader
 Darren    | Smith             |           |
 Darren    | Smith             |           |
 Jack      | Smith             | Darren    | Smith
 Tracy     | Smith             |           |
 Ponder    | Stibbons          | Burton    | Tracy
 Burton    | Tracy             |           |
 Hyacinth  | Tupperware        |           |
 Henry     | Worthington-Smyth | Tracy     | Smith
(31 rows)
```

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

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;
```

```
exercises=#
exercises=# select distinct m.firstname||' '||m.surname as member, f.name as facility
exercises-# from members m inner join
exercises-# bookings b using(memid) inner join
exercises-# facilities f on b.facid = f.facid
exercises-# where f.name like 'Tennis Court%'
exercises-# order by m.firstname||' '||m.surname, f.name;
      member       |    facility
-------------------+----------------
 Anne Baker        | Tennis Court 1
 Anne Baker        | Tennis Court 2
 Burton Tracy      | Tennis Court 1
 Burton Tracy      | Tennis Court 2
 Charles Owen      | Tennis Court 1
 Charles Owen      | Tennis Court 2
 Darren Smith      | Tennis Court 2
 David Farrell     | Tennis Court 1
 David Farrell     | Tennis Court 2
 David Jones       | Tennis Court 1
 David Jones       | Tennis Court 2
 David Pinker      | Tennis Court 1
 Douglas Jones     | Tennis Court 1
 Erica Crumpet     | Tennis Court 1
 Florence Bader    | Tennis Court 1
 Florence Bader    | Tennis Court 2
 Gerald Butters    | Tennis Court 1
 Gerald Butters    | Tennis Court 2
 GUEST GUEST       | Tennis Court 1
 GUEST GUEST       | Tennis Court 2
 Henrietta Rumney  | Tennis Court 2
 Jack Smith        | Tennis Court 1
 Jack Smith        | Tennis Court 2
 Janice Joplette   | Tennis Court 1
 Janice Joplette   | Tennis Court 2
 Jemima Farrell    | Tennis Court 1
 Jemima Farrell    | Tennis Court 2
 Joan Coplin       | Tennis Court 1
 John Hunt         | Tennis Court 1
 John Hunt         | Tennis Court 2
 Matthew Genting   | Tennis Court 1
 Millicent Purview | Tennis Court 2
 Nancy Dare        | Tennis Court 1
 Nancy Dare        | Tennis Court 2
 Ponder Stibbons   | Tennis Court 1
 Ponder Stibbons   | Tennis Court 2
 Ramnaresh Sarwin  | Tennis Court 1
 Ramnaresh Sarwin  | Tennis Court 2
 Tim Boothe        | Tennis Court 1
 Tim Boothe        | Tennis Court 2
 Tim Rownam        | Tennis Court 1
 Tim Rownam        | Tennis Court 2
 Timothy Baker     | Tennis Court 1
 Timothy Baker     | Tennis Court 2
 Tracy Smith       | Tennis Court 1
 Tracy Smith       | Tennis Court 2
(46 rows)
```

6) Produce a list of costly bookings

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;
```

```
exercises=# select  m.firstname|| ' ' ||m.surname as member, f.name as facility,
exercises-# case
exercises-# when m.memid = 0 then
exercises-# b.slots*f.guestcost
exercises-# else
exercises-# b.slots*f.membercost
exercises-# end as cost
exercises-# from members m inner join
exercises-#
exercises-# bookings b using(memid) inner join
exercises-#
exercises-# facilities f on b.facid=f.facid
exercises-# where (b.starttime between '2012-09-14' and '2012-09-15') and ((m.memid = 0 and b.slots*f.guestcost >30)or
exercises(#    (m.memid != 0 and b.slots*f.membercost >30))
exercises-#    order by cost desc;
     member      |    facility    | cost
-----------------+----------------+------
 GUEST GUEST     | Massage Room 2 |  320
 GUEST GUEST     | Massage Room 1 |  160
 GUEST GUEST     | Massage Room 1 |  160
 GUEST GUEST     | Massage Room 1 |  160
 GUEST GUEST     | Tennis Court 2 |  150
 Jemima Farrell  | Massage Room 1 |  140
 GUEST GUEST     | Tennis Court 1 |   75
 GUEST GUEST     | Tennis Court 2 |   75
 GUEST GUEST     | Tennis Court 1 |   75
 Matthew Genting | Massage Room 1 |   70
 Florence Bader  | Massage Room 2 |   70
 GUEST GUEST     | Squash Court   | 70.0
 Jemima Farrell  | Massage Room 1 |   70
 Ponder Stibbons | Massage Room 1 |   70
 Burton Tracy    | Massage Room 1 |   70
 Jack Smith      | Massage Room 1 |   70
 GUEST GUEST     | Squash Court   | 35.0
 GUEST GUEST     | Squash Court   | 35.0
(18 rows)
```

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 ;

```
exercises=#
exercises=# select distinct m.firstname||' '||m.surname as member, r.firstname||' '||r.surname as recommender
exercises-# from members m left outer join
exercises-# members r on r.memid=m.recommendedby
exercises-# order by m.firstname||' '||m.surname,r.firstname||' '||r.surname ;
         member          |    recommender
-------------------------+-------------------
 Anna Mackenzie          | Darren Smith
 Anne Baker              | Ponder Stibbons
 Burton Tracy            |
 Charles Owen            | Darren Smith
 Darren Smith            |
 David Farrell           |
 David Jones             | Janice Joplette
 David Pinker            | Jemima Farrell
 Douglas Jones           | David Jones
 Erica Crumpet           | Tracy Smith
 Florence Bader          | Ponder Stibbons
 Gerald Butters          | Darren Smith
 GUEST GUEST             |
 Henrietta Rumney        | Matthew Genting
 Henry Worthington-Smyth | Tracy Smith
 Hyacinth Tupperware     |
 Jack Smith              | Darren Smith
 Janice Joplette         | Darren Smith
 Jemima Farrell          |
 Joan Coplin             | Timothy Baker
 John Hunt               | Millicent Purview
 Matthew Genting         | Gerald Butters
 Millicent Purview       | Tracy Smith
 Nancy Dare              | Janice Joplette
 Ponder Stibbons         | Burton Tracy
 Ramnaresh Sarwin        | Florence Bader
 Tim Boothe              | Tim Rownam
 Tim Rownam              |
 Timothy Baker           | Jemima Farrell
 Tracy Smith             |
(30 rows)


exercises=#
```

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;
```

```
exercises=# select  m.firstname|| ' ' ||m.surname as member, f.name as facility,
exercises-# case
exercises-# when m.memid = 0 then
exercises-# b.slots*f.guestcost
exercises-# else
exercises-# b.slots*f.membercost
exercises-# end as cost
exercises-# from members m inner join
exercises-#
exercises-# bookings b using(memid) inner join
exercises-#
exercises-# facilities f on b.facid=f.facid
exercises-# where (b.starttime between '2012-09-14' and '2012-09-15') and
exercises-# ((m.memid = 0 and b.slots*f.guestcost >30)or
exercises(#  (m.memid != 0 and b.slots*f.membercost >30))
exercises-# order by cost desc;
     member      |    facility    | cost
-----------------+----------------+------
 GUEST GUEST     | Massage Room 2 |  320
 GUEST GUEST     | Massage Room 1 |  160
 GUEST GUEST     | Massage Room 1 |  160
 GUEST GUEST     | Massage Room 1 |  160
 GUEST GUEST     | Tennis Court 2 |  150
 Jemima Farrell  | Massage Room 1 |  140
 GUEST GUEST     | Tennis Court 1 |   75
 GUEST GUEST     | Tennis Court 2 |   75
 GUEST GUEST     | Tennis Court 1 |   75
 Matthew Genting | Massage Room 1 |   70
 Florence Bader  | Massage Room 2 |   70
 GUEST GUEST     | Squash Court   | 70.0
 Jemima Farrell  | Massage Room 1 |   70
 Ponder Stibbons | Massage Room 1 |   70
 Burton Tracy    | Massage Room 1 |   70
 Jack Smith      | Massage Room 1 |   70
 GUEST GUEST     | Squash Court   | 35.0
 GUEST GUEST     | Squash Court   | 35.0
(18 rows)
```

Modifying data

Insert some data into a table

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);

```
exercises=# insert into cd.facilities(facid,name,membercost,guestcost,initialoutlay
exercises(# ,monthlymaintenance)values(9,'Spa',20,30,100000,800);
INSERT 0 1
exercises=# select *
exercises-# from facilities;
 facid |      name       | membercost | guestcost | initialoutlay | monthlymaintenance
-------+-----------------+------------+-----------+---------------+--------------------
     0 | Tennis Court 1  |          5 |        25 |         10000 |                200
     1 | Tennis Court 2  |          5 |        25 |          8000 |                200
     2 | Badminton Court |          0 |      15.5 |          4000 |                 50
     3 | Table Tennis    |          0 |         5 |           320 |                 10
     4 | Massage Room 1  |         35 |        80 |          4000 |               3000
     5 | Massage Room 2  |         35 |        80 |          4000 |               3000
     6 | Squash Court    |        3.5 |      17.5 |          5000 |                 80
     7 | Snooker Table   |          0 |         5 |           450 |                 15
     8 | Pool Table      |          0 |         5 |           400 |                 15
     9 | Spa             |         20 |        30 |        100000 |                800
(10 rows)
```

2) Insert multiple rows of data into a table

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.

```
exercises=# select *
exercises-# from facilities;
 facid |      name       | membercost | guestcost | initialoutlay | monthlymaintenance
-------+-----------------+------------+-----------+---------------+--------------------
     0 | Tennis Court 1  |          5 |        25 |         10000 |                200
     1 | Tennis Court 2  |          5 |        25 |          8000 |                200
     2 | Badminton Court |          0 |      15.5 |          4000 |                 50
     3 | Table Tennis    |          0 |         5 |           320 |                 10
     4 | Massage Room 1  |         35 |        80 |          4000 |               3000
     5 | Massage Room 2  |         35 |        80 |          4000 |               3000
     6 | Squash Court    |        3.5 |      17.5 |          5000 |                 80
     7 | Snooker Table   |          0 |         5 |           450 |                 15
     8 | Pool Table      |          0 |         5 |           400 |                 15
     9 | Spa             |         20 |        30 |        100000 |                800
    10 | Squash Court 2  |        3.5 |      17.5 |          5000 |                 80
(11 rows)
```

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;

```
exercises=# insert into cd.facilities
exercises-# (facid,name,membercost,guestcost,initialoutlay,monthlymaintenance)
exercises-# select(select facid from cd.facilities order by facid desc limit 1)+1,'Spa',20,30,100000,800;
INSERT 0 1
exercises=#
exercises=# select *
exercises-# from facilities;
 facid |      name       | membercost | guestcost | initialoutlay | monthlymaintenance
-------+-----------------+------------+-----------+---------------+--------------------
     0 | Tennis Court 1  |          5 |        25 |         10000 |                200
     1 | Tennis Court 2  |          5 |        25 |          8000 |                200
     2 | Badminton Court |          0 |      15.5 |          4000 |                 50
     3 | Table Tennis    |          0 |         5 |           320 |                 10
     4 | Massage Room 1  |         35 |        80 |          4000 |               3000
     5 | Massage Room 2  |         35 |        80 |          4000 |               3000
     6 | Squash Court    |        3.5 |      17.5 |          5000 |                 80
     7 | Snooker Table   |          0 |         5 |           450 |                 15
     8 | Pool Table      |          0 |         5 |           400 |                 15
     9 | Spa             |         20 |        30 |        100000 |                800
(10 rows)
```

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';

```
exercises=# update facilities
exercises-# set initialoutlay = 10000
exercises-# where name = 'Tennis Court 2';
UPDATE 1
exercises=# select *
exercises-# from facilities;
 facid |      name       | membercost | guestcost | initialoutlay | monthlymaintenance
-------+-----------------+------------+-----------+---------------+--------------------
     0 | Tennis Court 1  |          5 |        25 |         10000 |                200
     2 | Badminton Court |          0 |      15.5 |          4000 |                 50
     3 | Table Tennis    |          0 |         5 |           320 |                 10
     4 | Massage Room 1  |         35 |        80 |          4000 |               3000
     5 | Massage Room 2  |         35 |        80 |          4000 |               3000
     6 | Squash Court    |        3.5 |      17.5 |          5000 |                 80
     7 | Snooker Table   |          0 |         5 |           450 |                 15
     8 | Pool Table      |          0 |         5 |           400 |                 15
     9 | Spa             |         20 |        30 |        100000 |                800
     1 | Tennis Court 2  |          5 |        25 |         10000 |                200
(10 rows)
```

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%';

```
exercises=# update facilities
exercises-# set guestcost = 30,
exercises-# membercost = 6
exercises-# where name like 'Tennis Court%';
UPDATE 2
exercises=# select *
exercises-# from facilities;
 facid |      name       | membercost | guestcost | initialoutlay | monthlymaintenance
-------+-----------------+------------+-----------+---------------+--------------------
     2 | Badminton Court |          0 |      15.5 |          4000 |                 50
     3 | Table Tennis    |          0 |         5 |           320 |                 10
     4 | Massage Room 1  |         35 |        80 |          4000 |               3000
     5 | Massage Room 2  |         35 |        80 |          4000 |               3000
     6 | Squash Court    |        3.5 |      17.5 |          5000 |                 80
     7 | Snooker Table   |          0 |         5 |           450 |                 15
     8 | Pool Table      |          0 |         5 |           400 |                 15
     9 | Spa             |         20 |        30 |        100000 |                800
     0 | Tennis Court 1  |          6 |        30 |         10000 |                200
     1 | Tennis Court 2  |          6 |        30 |         10000 |                200
(10 rows)
```

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';

```
exercises=# update facilities
exercises-# set membercost = (select membercost*1.1 from cd.facilities where name = 'Tennis Court 2'),
exercises-# guestcost = (select guestcost*1.1 from cd.facilities where name = 'Tennis Court 2')
exercises-# where name = 'Tennis Court 2';
UPDATE 1
exercises=# select *
exercises-# from facilities;
 facid |      name       | membercost | guestcost | initialoutlay | monthlymaintenance
-------+-----------------+------------+-----------+---------------+--------------------
     2 | Badminton Court |          0 |      15.5 |          4000 |                 50
     3 | Table Tennis    |          0 |         5 |           320 |                 10
     4 | Massage Room 1  |         35 |        80 |          4000 |               3000
     5 | Massage Room 2  |         35 |        80 |          4000 |               3000
     6 | Squash Court    |        3.5 |      17.5 |          5000 |                 80
     7 | Snooker Table   |          0 |         5 |           450 |                 15
     8 | Pool Table      |          0 |         5 |           400 |                 15
     9 | Spa             |         20 |        30 |        100000 |                800
     0 | Tennis Court 1  |          6 |        30 |         10000 |                200
     1 | Tennis Court 2  |        6.6 |      33.0 |         10000 |                200
(10 rows)
```

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;

```
exercises=# delete from bookings;
DELETE 4044
exercises=# select *
exercises-# from bookings;
 bookid | facid | memid | starttime | slots
--------+-------+-------+-----------+-------
(0 rows)
```

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);

```
exercises=#
exercises=# delete from cd.members
exercises-# where memid not in (select distinct memid
exercises(#    from cd.bookings
exercises(#    );
DELETE 31
```

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);

```
exercises=#
exercises=# delete from cd.members
exercises-# where memid not in (select distinct memid
exercises(#    from cd.bookings
exercises(#    );
DELETE 31
```

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;
```

```
exercises=# select count(*)
exercises-# from cd.facilities;
 count
-------
    10
(1 row)
```

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;

```
exercises=# select count(*)
exercises-# from cd.facilities
exercises-# where guestcost >= 10;
 count
-------
     7
(1 row)
```

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;

```
exercises=# select recommendedby, count(*)
exercises-#   from cd.members
exercises-#   where recommendedby is not null
exercises-#   group by recommendedby
exercises-#   order by recommendedby;
 recommendedby | count
---------------+-------
             1 |     5
             2 |     3
             3 |     1
             4 |     2
             5 |     1
             6 |     1
             9 |     2
            11 |     1
            13 |     2
            15 |     1
            16 |     1
            20 |     1
            30 |     1
(13 rows)
```

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;

```
exercises=# select f.facid as facid, sum(slots) as "Total Slots"
exercises-# from cd.facilities f inner join
exercises-# cd.bookings b using(facid)
exercises-# group by f.facid
exercises-# order by facid;
 facid | Total Slots
-------+-------------
     0 |        1320
     1 |        1278
     2 |        1209
     3 |         830
     4 |        1404
     5 |         228
     6 |        1104
     7 |         908
     8 |         911
(9 rows)
```

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);

```
exercises=# select facid , sum(slots)
exercises-# from cd.facilities inner join
exercises-# cd.bookings using(facid)
exercises-# where starttime between '2012-09-01' and '2012-10-01'
exercises-# group by facid
exercises-# order by sum(slots);
 facid | sum
-------+-----
     5 | 122
     3 | 422
     7 | 426
     8 | 471
     6 | 540
     2 | 570
     1 | 588
     0 | 591
     4 | 648
(9 rows)
```

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;

```
exercises=# select facid, extract(month from starttime) as month , sum(slots)
exercises-# from cd.bookings
exercises-# where extract(year from starttime) = 2012
exercises-# group by facid, month
exercises-# order by facid, month;
 facid | month | sum
-------+-------+-----
     0 |     7 | 270
     0 |     8 | 459
     0 |     9 | 591
     1 |     7 | 207
     1 |     8 | 483
     1 |     9 | 588
     2 |     7 | 180
     2 |     8 | 459
     2 |     9 | 570
     3 |     7 | 104
     3 |     8 | 304
     3 |     9 | 422
     4 |     7 | 264
     4 |     8 | 492
     4 |     9 | 648
     5 |     7 |  24
     5 |     8 |  82
     5 |     9 | 122
     6 |     7 | 164
     6 |     8 | 400
     6 |     9 | 540
     7 |     7 | 156
     7 |     8 | 326
     7 |     9 | 426
     8 |     7 | 117
     8 |     8 | 322
     8 |     9 | 471
-- More  --
```

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);
```

```
exercises=# select count(*)
exercises-# from cd.members
exercises-# where memid in (select memid
exercises(#    from cd.bookings
exercises(#    where memid is not null);
 count
-------
    30
(1 row)
```

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;

```
exercises=# select facid, sum(slots)
exercises-# from cd.bookings
exercises-# group by facid
exercises-# having sum(slots) > 1000
exercises-# order by facid;
 facid | sum
-------+------
     0 | 1320
     1 | 1278
     2 | 1209
     4 | 1404
     6 | 1104
(5 rows)
```

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;

```
exercises=# select f.name, sum(slots * case
exercises(# when memid = 0 then f.guestcost
exercises(# else f.membercost
exercises(# end) as revenue
exercises-# from cd.bookings inner join cd.facilities f using(facid)
exercises-# group by f.name
exercises-# order by revenue;
      name       | revenue
-----------------+---------
 Table Tennis    |     180
 Snooker Table   |     240
 Pool Table      |     270
 Badminton Court |  1906.5
 Squash Court    | 13468.0
 Massage Room 2  |   15810
 Tennis Court 1  |   16632
 Tennis Court 2  | 18889.2
 Massage Room 1  |   72540
(9 rows)
```

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;
```

```
exercises=# select f.name, sum(case when memid=0 then slots*f.guestcost
exercises(# else slots*f.membercost end) as revenue
exercises-# from cd.facilities f inner join cd.bookings using(facid)
exercises-# group by f.name
exercises-# having sum(case
exercises(# when memid = 0 then slots * f.guestcost
exercises(# else slots * membercost
exercises(# end) < 1000
exercises-# order by revenue;
    name     | revenue
-------------+---------
 Table Tennis |     180
 Snooker Table |     240
 Pool Table   |     270
(3 rows)
```

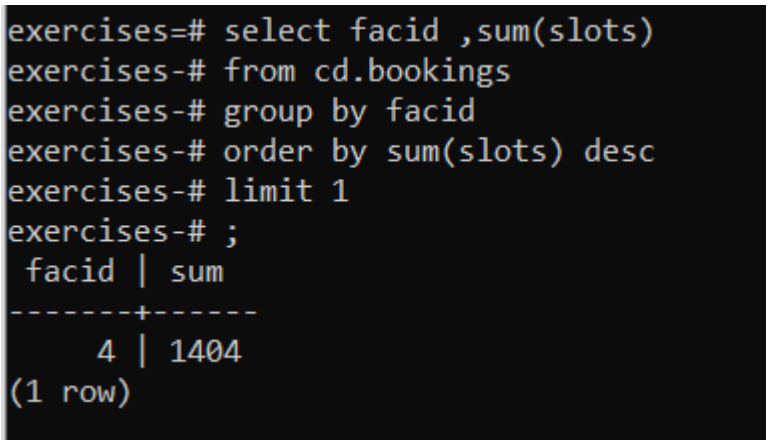
11)

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;
```

```
exercises=# select facid , extract(month from starttime) as month, sum(slots)
exercises-# from cd.bookings
exercises-# where extract(year from starttime) = 2012 or starttime = null
exercises-# group by rollup(facid, month)
exercises-# order by facid;
 facid | month | sum
-------+-------+------
     0 |     7 |  270
     0 |     8 |  459
     0 |     9 |  591
     0 |       | 1320
     1 |     7 |  207
     1 |     8 |  483
     1 |     9 |  588
     1 |       | 1278
     2 |     7 |  180
     2 |     8 |  459
     2 |     9 |  570
     2 |       | 1209
     3 |     7 |  104
     3 |     8 |  304
     3 |     9 |  422
     3 |       |  830
     4 |     7 |  264
     4 |     8 |  492
     4 |     9 |  648
     4 |       | 1404
     5 |     7 |   24
     5 |     8 |   82
```

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;

```
exercises=# select b.facid, f.name, sum(slots/2.0) as "Total Houes"
exercises-# from cd.bookings b inner join cd.facilities f using(facid)
exercises-# group by facid,f.name
exercises-# order by facid;
 facid |      name       |       Total Houes
-------+-----------------+--------------------------
     0 | Tennis Court 1  |     660.0000000000000000
     1 | Tennis Court 2  |     639.0000000000000000
     2 | Badminton Court |     604.5000000000000000
     3 | Table Tennis    | 415.00000000000000000000
     4 | Massage Room 1  | 702.00000000000000000000
     5 | Massage Room 2  | 114.00000000000000000000
     6 | Squash Court    | 552.00000000000000000000
     7 | Snooker Table   | 454.00000000000000000000
     8 | Pool Table      | 455.50000000000000000000
(9 rows)
```

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;
```

```
exercises=# select surname, firstname, memid, min(b.starttime) as strattime
exercises-# from cd.members inner join cd.bookings b using(memid)
exercises-# where b.starttime between '2012-09-01' and '2012-09-02'
exercises-# group by surname, firstname, memid
exercises-# order by memid;
  surname   | firstname | memid |      strattime
------------+-----------+-------+---------------------
 GUEST      | GUEST     |     0 | 2012-09-01 08:00:00
 Smith      | Darren    |     1 | 2012-09-01 09:00:00
 Smith      | Tracy     |     2 | 2012-09-01 11:30:00
 Rownam     | Tim       |     3 | 2012-09-01 16:00:00
 Joplette   | Janice    |     4 | 2012-09-01 15:00:00
 Tracy      | Burton    |     6 | 2012-09-01 15:00:00
 Dare       | Nancy     |     7 | 2012-09-01 12:30:00
 Boothe     | Tim       |     8 | 2012-09-01 08:30:00
 Stibbons   | Ponder    |     9 | 2012-09-01 11:00:00
 Owen       | Charles   |    10 | 2012-09-01 11:00:00
 Jones      | David     |    11 | 2012-09-01 09:30:00
 Baker      | Anne      |    12 | 2012-09-01 14:30:00
 Farrell    | Jemima    |    13 | 2012-09-01 09:30:00
 Smith      | Jack      |    14 | 2012-09-01 11:00:00
 Bader      | Florence  |    15 | 2012-09-01 10:30:00
 Baker      | Timothy   |    16 | 2012-09-01 15:00:00
 Pinker     | David     |    17 | 2012-09-01 08:30:00
 Genting    | Matthew   |    20 | 2012-09-01 18:00:00
 Mackenzie  | Anna      |    21 | 2012-09-01 08:30:00
```

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;

```
exercises=# select count(*) over(), firstname, surname
exercises-# from cd.members
exercises-# order by joindate;
 count | firstname |      surname
-------+-----------+-------------------
    31 | GUEST     | GUEST
    31 | Darren    | Smith
    31 | Tracy     | Smith
    31 | Tim       | Rownam
    31 | Janice    | Joplette
    31 | Gerald    | Butters
    31 | Burton    | Tracy
    31 | Nancy     | Dare
    31 | Tim       | Boothe
    31 | Ponder    | Stibbons
    31 | Charles   | Owen
    31 | David     | Jones
    31 | Anne      | Baker
    31 | Jemima    | Farrell
    31 | Jack      | Smith
    31 | Florence  | Bader
```

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;

```
exercises=# select row_number() over(), firstname, surname
exercises-# from cd.members;
 row_number | firstname |      surname
------------+-----------+-------------------
          1 | GUEST     | GUEST
          2 | Darren    | Smith
          3 | Tracy     | Smith
          4 | Tim       | Rownam
          5 | Janice    | Joplette
          6 | Gerald    | Butters
          7 | Burton    | Tracy
          8 | Nancy     | Dare
          9 | Tim       | Boothe
         10 | Ponder    | Stibbons
         11 | Charles   | Owen
         12 | David     | Jones
         13 | Anne      | Baker
         14 | Jemima    | Farrell
         15 | Jack      | Smith
         16 | Florence  | Bader
         17 | Timothy   | Baker
         18 | David     | Pinker
         19 | Matthew   | Genting
         20 | Anna      | Mackenzie
         21 | Joan      | Coplin
         22 | Ramnaresh | Sarwin
         23 | Douglas   | Jones
         24 | Henrietta | Rumney
```

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;

```
exercises=#
exercises=# select facid, sum(slots)
exercises-# from cd.bookings
exercises-# group by facid
exercises-# order by sum(slots) desc
exercises-# limit 1;
 facid | sum
-------+------
     4 | 1404
(1 row)
```

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;
```

```
exercises=# select firstname, surname, ((sum(b.slots)+10)/20)*10 as hours,
exercises-# rank() over (order by ((sum(b.slots)+10)/20)*10 desc) as rank
exercises-# from cd.bookings b inner join cd.members m using(memid)
exercises-# group by m.memid
exercises-# order by rank,surname,firstname;
 firstname |      surname       | hours | rank
-----------+--------------------+-------+------
 GUEST     | GUEST              |  1200 |    1
 Darren    | Smith              |   340 |    2
 Tim       | Rownam             |   330 |    3
 Tim       | Boothe             |   220 |    4
 Tracy     | Smith              |   220 |    4
 Gerald    | Butters            |   210 |    6
 Burton    | Tracy              |   180 |    7
 Charles   | Owen               |   170 |    8
 Janice    | Joplette           |   160 |    9
 Anne      | Baker              |   150 |   10
 Timothy   | Baker              |   150 |   10
 David     | Jones              |   150 |   10
 Nancy     | Dare               |   130 |   13
 Florence  | Bader              |   120 |   14
 Anna      | Mackenzie          |   120 |   14
 Ponder    | Stibbons           |   120 |   14
 Jack      | Smith              |   110 |   17
 Jemima    | Farrell            |    90 |   18
 David     | Pinker             |    80 |   19
 Ramnaresh | Sarwin             |    80 |   19
 Matthew   | Genting            |    70 |   21
```

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;
```

```
exercises-# order by rank;
      name      | rank
----------------+------
 Massage Room 1 |    1
 Tennis Court 2 |    2
 Tennis Court 1 |    3
(3 rows)
```

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'

```
exercises=# select timestamp '2012-08-31 01:00:00'
exercises-# ;
      timestamp
---------------------
 2012-08-31 01:00:00
(1 row)
```

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;

```
exercises=# select timestamp '2012-08-31 01:00:00' - timestamp '2012-07-30 01:00:00' as interval;
 interval
----------
 32 days
(1 row)
```

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;

```
exercises=# select generate_series(timestamp '2012-10-01', timestamp '2012-10-31', interval '1 day') as ts;
         ts
---------------------
 2012-10-01 00:00:00
 2012-10-02 00:00:00
 2012-10-03 00:00:00
 2012-10-04 00:00:00
 2012-10-05 00:00:00
 2012-10-06 00:00:00
 2012-10-07 00:00:00
 2012-10-08 00:00:00
 2012-10-09 00:00:00
 2012-10-10 00:00:00
 2012-10-11 00:00:00
 2012-10-12 00:00:00
 2012-10-13 00:00:00
 2012-10-14 00:00:00
 2012-10-15 00:00:00
 2012-10-16 00:00:00
 2012-10-17 00:00:00
 2012-10-18 00:00:00
 2012-10-19 00:00:00
 2012-10-20 00:00:00
 2012-10-21 00:00:00
 2012-10-22 00:00:00
 2012-10-23 00:00:00
 2012-10-24 00:00:00
```

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');

```
exercises=# select extract(day from timestamp '2012-08-31');
 extract
---------
      31
(1 row)
```

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'));

```
exercises=# select extract(epoch from (timestamp '2012-09-02 00:00:00' - '2012-08-31 01:00:00'));
    extract
---------------
 169200.000000
(1 row)
```

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

```
exercises=#
exercises=# select (date_trunc('month',ts.testts) + interval '1 month')
exercises-# - date_trunc('day', ts.testts) as remaining
exercises-# from (select timestamp '2012-02-11 01:00:00' as testts) ts
exercises-# ;
 remaining
-----------
 19 days
(1 row)
```

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;

```
exercises=# SELECT STARTTIME , STARTTIME + SLOTS*(INTERVAL '30 minutes') ENDTIME
exercises-# FROM CD.BOOKINGS
exercises-# ORDER BY ENDTIME DESC, STARTTIME DESC
exercises-# LIMIT 10;
      starttime      |       endtime
---------------------+---------------------
 2013-01-01 15:30:00 | 2013-01-01 16:00:00
 2012-09-30 19:30:00 | 2012-09-30 20:30:00
 2012-09-30 19:00:00 | 2012-09-30 20:30:00
 2012-09-30 19:30:00 | 2012-09-30 20:00:00
 2012-09-30 19:00:00 | 2012-09-30 20:00:00
 2012-09-30 19:00:00 | 2012-09-30 20:00:00
 2012-09-30 18:30:00 | 2012-09-30 20:00:00
 2012-09-30 18:30:00 | 2012-09-30 20:00:00
 2012-09-30 19:00:00 | 2012-09-30 19:30:00
 2012-09-30 18:30:00 | 2012-09-30 19:30:00
(10 rows)
```

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;

```
exercises=# SELECT SURNAME||', '||FIRSTNAME AS NAME
exercises-# FROM CD.MEMBERS;
           name
-------------------------
 GUEST, GUEST
 Smith, Darren
 Smith, Tracy
 Rownam, Tim
 Joplette, Janice
 Butters, Gerald
 Tracy, Burton
 Dare, Nancy
 Boothe, Tim
 Stibbons, Ponder
 Owen, Charles
 Jones, David
 Baker, Anne
 Farrell, Jemima
 Smith, Jack
 Bader, Florence
 Baker, Timothy
 Pinker, David
 Genting, Matthew
 Mackenzie, Anna
 Coplin, Joan
```

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%';

```
exercises=# SELECT *
exercises-# FROM CD.FACILITIES
exercises-# WHERE  NAME LIKE 'Tennis%';
 facid |      name      | membercost | guestcost | initialoutlay | monthlymaintenance
-------+----------------+------------+-----------+---------------+--------------------
     0 | Tennis Court 1 |          6 |        30 |         10000 |                200
     1 | Tennis Court 2 |        6.6 |      33.0 |         10000 |                200
(2 rows)
```

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';*

```
exercises=# select *
exercises-# from cd.facilities
exercises-# where name ~*  'Tennis Court';
 facid |      name      | membercost | guestcost | initialoutlay | monthlymaintenance
-------+----------------+------------+-----------+---------------+--------------------
     0 | Tennis Court 1 |          6 |        30 |         10000 |                200
     1 | Tennis Court 2 |        6.6 |      33.0 |         10000 |                200
(2 rows)
```

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 ~ '[()]';

```
exercises=# select memid, telephone
exercises-# from cd.members
exercises-# where telephone ~ '[()]';
 memid |   telephone
-------+----------------
     0 | (000) 000-0000
     3 | (844) 693-0723
     4 | (833) 942-4710
     5 | (844) 078-4130
     6 | (822) 354-9973
     7 | (833) 776-4001
     8 | (811) 433-2547
     9 | (833) 160-3900
    10 | (855) 542-5251
    11 | (844) 536-8036
    13 | (855) 016-0163
    14 | (822) 163-3254
    15 | (833) 499-3527
    20 | (811) 972-1377
    21 | (822) 661-2898
    22 | (822) 499-2232
    24 | (822) 413-1470
    27 | (822) 989-8876
    28 | (855) 755-9876
    29 | (855) 894-3758
```

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

```
exercises=#   select lpad(cast(zipcode as char(5)),5,'0') zip from cd.members order by zip
exercises-#
exercises-#
exercises-# ;
  zip
-------
 00000
 00234
 00234
 04321
 04321
 10383
 11986
 23423
 28563
 33862
```

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;

```
exercises=# select substr(surname,1,1)as letter, count(*)
exercises-# from cd.members
exercises-# group by letter
exercises-# order by letter;
 letter | count
--------+-------
 B      |     5
 C      |     2
 D      |     1
 F      |     2
 G      |     2
 H      |     1
 J      |     3
 M      |     1
 O      |     1
 P      |     2
 R      |     2
 S      |     6
 T      |     2
 W      |     1
(14 rows)
```

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;

```
exercises=# select memid, translate(telephone, '()- ','') as telephone
exercises-# from cd.members
exercises-# order by memid;
 memid | telephone
-------+------------
     0 | 0000000000
     1 | 5555555555
     2 | 5555555555
     3 | 8446930723
     4 | 8339424710
     5 | 8440784130
     6 | 8223549973
     7 | 8337764001
     8 | 8114332547
     9 | 8331603900
    10 | 8555425251
    11 | 8445368036
    12 | 8440765141
    13 | 8550160163
    14 | 8221633254
    15 | 8334993527
    16 | 8339410824
    17 | 8114096734
    20 | 8119721377
```