/\* Q1: Some of the facilities charge a fee to members, but some do not.

Write a SQL query to produce a list of the names of the facilities that do. \*/

select name as members\_charged

from Facilities

where membercost>0.0

/\* Q2: How many facilities do not charge a fee to members? \*/

select name as charge\_free\_members

from Facilities

where membercost=0.0

/\* Q3: Write an SQL query to show a list of facilities that charge a fee to members,

where the fee is less than 20% of the facility's monthly maintenance cost.

Return the facid, facility name, member cost, and monthly maintenance of the

facilities in question. \*/

select facid, name, membercost, monthlymaintenance

from Facilities

where membercost<(monthlymaintenance\*0.2) and membercost>0

/\* Q4: Write an SQL query to retrieve the details of facilities with ID 1 and 5.

Try writing the query without using the OR operator. \*/

select \*

from Facilities

where facid in (1, 5)

/\* Q5: 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, monthlymaintenance,

IF(monthlymaintenance > 100, 'expensive', 'cheap') AS labelled

from Facilities

/\* Q6: You'd like to get the first and last name of the last member(s)

who signed up. Try not to use the LIMIT clause for your solution. \*/

select firstname, surname

from Members

where firstname != 'GUEST' or surname!= 'GUEST'

order by joindate desc

/\* Q7: 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. \*/

SELECT DISTINCT concat(surname,' ', firstname) as member, f.name as court\_name

from Members as m

INNER JOIN Bookings as b using(memid)

INNER JOIN Facilities as f using(facid)

where f.name like 'Tennis%'

order by member;

/\* Q8: 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's ID is always 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 DISTINCT concat(m.surname, ' ', m.firstname) as member, f.name,

if(memid = 0, membercost, guestcost)\*slots as cost

from Members as m

inner JOIN Bookings as b

using(memid)

inner join Facilities as f

using(facid)

where b.starttime LIKE '2012-09-14%'

and if(memid = 0, membercost, guestcost)\*slots>30

order by cost desc;

/\* Q9: This time, produce the same result as in Q8, but using a subquery. \*/

select

member,

fsub.facility,

IF(Bookings.memid=0, fsub.membercost, fsub.guestcost)\*slots as cost

from Bookings,

(select memid, concat(surname, ' ', firstname) as member

from Members) as msub,

(SELECT facid, name as facility, membercost, guestcost

from Facilities) as fsub

WHERE msub.memid = Bookings.memid

and fsub.facid = Bookings.facid

and Bookings.starttime like '2012-09-14%'

and IF(Bookings.memid=0, fsub.membercost, fsub.guestcost)\*slots>30

/\* Q10: Produce a list of facilities with a total revenue less than 1000.

The output of facility name and total revenue, sorted by revenue. Remember

that there's a different cost for guests and members! \*/

select f.name as facility,

if(memid = 0, membercost, guestcost)\*slots as revenue

from Members as m

INNER JOIN Bookings as b using(memid)

INNER JOIN Facilities as f USING(facid)

WHERE if(memid = 0, membercost, guestcost)\*slots < 1000

ORDER BY revenue DESC;

/\* Q11: Produce a report of members and who recommended them in alphabetic surname,firstname order \*/

SELECT surname, firstname, recommendedby

FROM Members

where recommendedby>0

order by surname ASC, firstname ASC

/\* Q12: Find the facilities with their usage by member, but not guests \*/

SELECT DISTINCT concat(surname, ' ',firstname) as member, f.name as facility, m.joindate

from Members as m

INNER JOIN Bookings as b USING(memid)

INNER JOIN Facilities as f USING(facid)

WHERE joindate>0

/\* Q13: Find the facilities usage by month, but not guests \*/

SELECT DISTINCT concat(surname,' ', firstname) as member, f.name as facility, EXTRACT(MONTH FROM m.joindate) as month

from Members as m

INNER JOIN Bookings as b using(memid)

INNER JOIN Facilities as f using(facid)