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

Please list the names of the facilities that do. \*/

**Ans 1: SELECT name**

**FROM `Facilities`**

**WHERE membercost > 0.0;**

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

**Ans 2: SELECT count(name)**

**FROM `Facilities`**

**WHERE membercost = 0.0;**

/\* Q3: How can you produce 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. \*/

**Ans 3: SELECT facid , name, membercost, monthlymaintenance**

**FROM Facilities**

**WHERE membercost > 0 AND membercost < (.2 \* monthlymaintenance);**

/\* Q4: How can you retrieve the details of facilities with ID 1 and 5?

Write the query without using the OR operator. \*/

**Ans 4: SELECT \* FROM Facilities WHERE facid = 1**

**UNION**

**SELECT \* FROM Facilities WHERE facid = 5;**

/\* Q5: 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. \*/

**Ans 5: SELECT name, monthlymaintenance,**

**CASE**

**WHEN monthlymaintenance > 100 Then 'expensive'**

**ELSE 'cheap'**

**END**

**AS 'label'**

**FROM Facilities;**

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

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

**Ans 6: SELECT firstname , surname**

**FROM `Members`**

**WHERE joindate = (SELECT MAX(joindate) from Members);**

/\* Q7: 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. \*/

**Ans 7:** **SELECT DISTINCT CONCAT(m.firstname,' ', m.surname) as member\_name, f.name as court\_name**

**FROM Members as m**

**INNER JOIN Bookings as b ON m.memid=b.memid**

**inner join Facilities as f ON b.facid=f.facid**

**WHERE f.name in ('Tennis Court 1' ,'Tennis Court 2')**

**ORDER BY member\_name;**

/\* Q8: 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'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. \*/

**Ans 8: SELECT CONCAT(m.firstname,' ',m.surname) as member\_name , f.name as facility\_name,**

**CASE**

**WHEN m.memid=0 THEN f.guestcost\*b.slots**

**ELSE f.membercost\*b.slots**

**END**

**AS total\_cost**

**FROM Members as m**

**INNER JOIN Bookings as b ON m.memid=b.memid**

**INNER JOIN Facilities as f ON f.facid=b.facid**

**WHERE (b.starttime LIKE '2012-09-14%') AND**

**((m.memid=0 AND f.guestcost\*b.slots >30) OR**

**(m.memid<>0 AND f.membercost\*b.slots>30))**

**ORDER BY total\_cost DESC;**

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

**Ans 9:** **SELECT \* FROM**

**( SELECT CONCAT(m.firstname,' ',m.surname) as member\_name , f.name as facility\_name,**

**CASE**

**WHEN m.memid=0 THEN f.guestcost\*b.slots**

**ELSE f.membercost\*b.slots**

**END**

**AS total\_cost**

**FROM Members as m**

**INNER JOIN Bookings as b ON m.memid=b.memid**

**INNER JOIN Facilities as f ON f.facid=b.facid**

**WHERE b.starttime LIKE '2012-09-14%' ) as t**

**WHERE t.total\_cost >30**

**ORDER BY t.total\_cost DESC;**

/\* 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! \*/

**ANS 10:** **SELECT \* FROM**

**( SELECT f.name AS facility\_name ,**

**SUM(CASE**

**WHEN b.memid = 0 THEN f.guestcost\* b.slots**

**ELSE f.membercost\*b.slots**

**END)**

**AS total\_revenue**

**FROM Bookings as b**

**INNER JOIN Facilities as f ON b.facid=f.facid**

**GROUP BY facility\_name ) as t**

**WHERE t.total\_revenue < 1000**

**ORDER BY t.total\_revenue;**