/\* Welcome to the SQL mini project. For this project, you will use

Springboard' online SQL platform, which you can log into through the

following link:

https://sql.springboard.com/

Username: student

Password: learn\_sql@springboard

The data you need is in the "country\_club" database. This database

contains 3 tables:

i) the "Bookings" table,

ii) the "Facilities" table, and

iii) the "Members" table.

Note that, if you need to, you can also download these tables locally.

In the mini project, you'll be asked a series of questions. You can

solve them using the platform, but for the final deliverable,

paste the code for each solution into this script, and upload it

to your GitHub.

Before starting with the questions, feel free to take your time,

exploring the data, and getting acquainted with the 3 tables. \*/

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

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

SELECT \*

FROM Facilities

WHERE membercost > 0

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

SELECT Count(\*)

FROM Facilities

WHERE membercost = 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. \*/

SELECT facid, name, membercost, monthlymaintenance

FROM Facilities

WHERE membercost > 0 AND membercost < (monthlymaintenance \* 0.2)

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

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

SELECT \*

FROM Facilities

WHERE facid IN (1,5)

/\* 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. \*/

SELECT name, monthlymaintenance,

CASE

WHEN monthlymaintenance > 100 THEN 'expensive'

ELSE 'cheap'

END

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. \*/

SELECT surname, firstname, joindate

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. \*/

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

FROM Members m

JOIN Bookings b ON b.memid = m.memid

JOIN Facilities f ON f.facid = b.facid

WHERE f.name like 'Tennis Court %'

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. \*/

SELECT CONCAT(m.surname , ' ' , m.firstname) 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 cost

FROM Members m

JOIN Bookings b on b.memid = m.memid

JOIN Facilities f on f.facid = b.facid

WHERE b.starttime LIKE ('2012-09-14 %') AND cost > 30

ORDER BY cost DESC

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

SELECT CONCAT(m.surname , ' ' , m.firstname) as member\_name, subquery.facility\_name, subquery.cost

FROM

(SELECT b.memid, f.name as facility\_name, b.starttime,

CASE

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

ELSE f.membercost \* b.slots

END

as cost

FROM Bookings b

JOIN Facilities f on f.facid = b.facid

) as subquery

JOIN Members m on subquery.memid = m.memid

WHERE subquery.starttime LIKE ('2012-09-14 %')

HAVING subquery.cost > 30

ORDER BY subquery.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! \*/

SELECT f.name as facility\_name, sum(

CASE

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

ELSE f.membercost \* b.slots

END

) as revenue

FROM Members m

JOIN Bookings b on b.memid = m.memid

JOIN Facilities f on f.facid = b.facid

GROUP BY facility\_name

HAVING revenue < 1000

ORDER BY revenue