

/\* 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. \*/

CODE:

```
SELECT name, membercost FROM `Facilities` WHERE membercost > 0
```

OUTPUT:

name	membercost
Tennis Court 1	5.0
Tennis Court 2	5.0
Massage Room 1	9.9
Massage Room 2	9.9
Squash Court	3.5

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

CODE:

```
SELECT name, membercost FROM `Facilities` WHERE membercost = 0
```

OUTPUT:

name	membercost
Badminton Court	0.0
Table Tennis	0.0
Snooker Table	0.0
Pool Table	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. \*/

CODE:

```
SELECT `facid`, `name`, `membercost`, `monthlymaintenance` FROM `Facilities` WHERE `membercost` < `monthlymaintenance` * .2
```

-OR-

```
SELECT `facid`, `name`, `membercost`, `monthlymaintenance` FROM `Facilities` WHERE `membercost` < (`monthlymaintenance` * .2)
```

/\* Better code to force order of operations \*/

OUTPUT

	facid	name	membercost	monthlymaintenance
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0	Tennis Court 1	5.0	200
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Tennis Court 2	5.0	200
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Badminton Court	0.0	50
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	Table Tennis	0.0	10
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	Massage Room 1	9.9	3000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	Massage Room 2	9.9	3000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	Squash Court	3.5	80
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	Snooker Table	0.0	15
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	Pool Table	0.0	15

/\* Q4: How can you retrieve the details of facilities with ID 1 and 5? Write the query without using the OR operator. \*/

CODE

```
SELECT * FROM `Facilities` WHERE `facid` = 1 OR `facid` = 5
```

OUTPUT

	facid	name	membercost	guestcost	initialoutlay	monthlymaintenance
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Tennis Court 2	5.0	25.0	8000	200
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	Massage Room 2	9.9	80.0	4000	3000

/\* 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. \*/

```

CODE
SELECT `name`
      , CASE
        WHEN `monthlymaintenance` < 100 THEN 'Expensive'
        WHEN `monthlymaintenance` > 100 THEN 'Cheap'
      END
      , `monthlymaintenance`
FROM `Facilities`

```

## OUTPUT

name	CASE WHEN `monthlymaintenance` < 100 THEN 'Expensive' WHEN `monthlymaintenance` > 100 THEN 'Cheap' END	monthlymaintenance
Tennis Court 1	Cheap	200
Tennis Court 2	Cheap	200
Badminton Court	Expensive	50
Table Tennis	Expensive	10
Massage Room 1	Cheap	3000
Massage Room 2	Cheap	3000
Squash Court	Expensive	80
Snooker Table	Expensive	15
Pool Table	Expensive	15

/\* 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. \*/

```

CODE
SELECT `surname`, `firstname`, MAX(`joindate`) FROM `Members` WHERE `firstname` != "GUEST"

```

## OUTPUT

surname	firstname	MAX(`joindate`)
Smith	Darren	2012-09-26 18:08:45

/\* 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. \*/

```

CODE
SELECT DISTINCT Bookings.memid,
Bookings.facid,
Members.memid,
Facilities.facid,
CONCAT(Members.firstname, ' ', Members.surname) AS MemberName,
Facilities.name AS "Facility Name"
FROM Bookings
INNER JOIN Members
ON Bookings.memid = Members.memid
INNER JOIN Facilities
ON Bookings.facid = Facilities.facid
WHERE Facilities.facid = 0 OR Facilities.facid = 1

```

## OUTPUT

memid	facid	memid	facid	MemberName	Facility Name
2	0	2	0	Tracy, Smith	Tennis Court 1
0	1	0	1	GUEST, GUEST	Tennis Court 2
0	0	0	0	GUEST, GUEST	Tennis Court 1
3	1	3	1	Tim, Rownam	Tennis Court 2
3	0	3	0	Tim, Rownam	Tennis Court 1
1	1	1	1	Darren, Smith	Tennis Court 2
4	0	4	0	Janice, Joplette	Tennis Court 1
5	0	5	0	Gerald, Butters	Tennis Court 1
4	1	4	1	Janice, Joplette	Tennis Court 2
2	1	2	1	Tracy, Smith	Tennis Court 2
5	1	5	1	Gerald, Butters	Tennis Court 2
8	1	8	1	Tim, Boothe	Tennis Court 2
6	1	6	1	Burton, Tracy	Tennis Court 2
6	0	6	0	Burton, Tracy	Tennis Court 1
7	1	7	1	Nancy, Dare	Tennis Court 2
7	0	7	0	Nancy, Dare	Tennis Court 1
8	0	8	0	Tim, Boothe	Tennis Court 1
9	1	9	1	Ponder, Stibbons	Tennis Court 2
10	0	10	0	Charles, Owen	Tennis Court 1
10	1	10	1	Charles, Owen	Tennis Court 2
12	0	12	0	Anne, Baker	Tennis Court 1
11	1	11	1	David, Jones	Tennis Court 2
14	0	14	0	Jack, Smith	Tennis Court 1
12	1	12	1	Anne, Baker	Tennis Court 2
11	0	11	0	David, Jones	Tennis Court 1
16	1	16	1	Timothy, Baker	Tennis Court 2
15	1	15	1	Florence, Bader	Tennis Court 2
16	0	16	0	Timothy, Baker	Tennis Court 1
17	0	17	0	David, Pinker	Tennis Court 1
13	1	13	1	Jemima, Farrell	Tennis Court 2

/\* 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
Bookings.starttime,
Facilities.name AS "Facility Name",
CONCAT(Members.firstname, ' ', Members.surname) AS "Member Name",
Facilities.membercost,
Facilities.guestcost,
CASE
    WHEN Facilities.membercost > 30 THEN Facilities.membercost
    WHEN Bookings.memid = 0 AND Facilities.guestcost > 15 THEN Facilities.guestcost *
2
    END AS "Cost"
FROM Bookings
INNER JOIN Facilities
ON Bookings.facid = Facilities.facid
INNER JOIN Members
ON Bookings.memid = Members.memid
WHERE
Bookings.starttime LIKE '2012-09-14%'

```

--Code does not give desired output.

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

```
SELECT * FROM (
SELECT
Bookings.starttime,
Facilities.name AS "Facilty Name",
CONCAT(Members.firstname, ' ', Members.surname) AS "Member Name",
Facilities.membercost,
Facilities.guestcost,
CASE
WHEN Facilities.membercost > 30 THEN Facilities.membercost
WHEN Bookings.memid = 0 AND Facilities.guestcost > 15 THEN Facilities.guestcost *
2
END AS "Cost"
FROM Bookings
INNER JOIN Facilities
ON Bookings.facid = Facilities.facid
INNER JOIN Members
ON Bookings.memid = Members.memid
WHERE
Bookings.starttime LIKE '2012-09-14%'
)
WHERE Cost IS NOT NULL
```

--Code does not give desired output.

/\* 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
Facilities.name AS "Facilty Name",
Facilities.membercost,
Facilities.guestcost,
CASE
WHEN Facilities.membercost < 1000 THEN Facilities.membercost
WHEN Bookings.memid = 0 AND Facilities.guestcost < 500 THEN Facilities.guestcost
* 2
END AS "Revenue"
FROM Bookings
INNER JOIN Facilities
ON Bookings.facid = Facilities.facid
-- Sort by Revenue
```

--Code does not give desired output.