

/* Welcome to the SQL mini project. You will carry out this project partly in the PHPMyAdmin interface, and partly in Jupyter via a Python connection.

This is Tier 1 of the case study, which means that there'll be more guidance for you about how to setup your local SQLite connection in PART 2 of the case study.

The questions in the case study are exactly the same as with Tier 2.

PART 1: PHPMyAdmin

You will complete questions 1-9 below in the PHPMyAdmin interface. Log in by pasting the following URL into your browser, and using the following Username and Password:

URL: <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.

In this case study, 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. */

/* QUESTIONS

/* 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  
FROM Facilities  
WHERE membercost !=0
```

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

```
SELECT COUNT( * )  
FROM Facilities  
WHERE membercost =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 > 0 and membercost < 0.2 * monthlymaintenance
```

/* 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, (  
CASE WHEN monthlymaintenance <100  
THEN 'cheap'  
ELSE 'expensive'  
END  
) AS maintenance  
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 joindate = (  
SELECT MAX( joindate )  
FROM `Members` )
```

/* 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( M.firstname, ' ', M.surname), F.name AS  
Mem_And_Court_Names
```

```

FROM Members M
LEFT JOIN Bookings B ON M.memid = B.memid
LEFT JOIN Facilities F ON F.facid = B.facid
WHERE F.name
IN (
'Tennis Court 1', 'Tennis Court 2'
)
ORDER BY M.firstname, M.surname

```

/* 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 Fac.name AS "facility name",
CONCAT(Mem.firstname, ' ', Mem.surname) AS "member name",
CASE WHEN Book.memid = 0 THEN Fac.guestcost * Book.slots
ELSE Fac.membercost * Book.slots END AS "cost"
FROM Bookings Book
LEFT JOIN Facilities Fac
ON Book.facid = Fac.facid
LEFT JOIN Members Mem
ON Book.memid = Mem.memid
WHERE (Book.starttime LIKE '2012-09-14%')
AND (CASE WHEN Book.memid = 0 THEN Fac.guestcost * Book.slots
ELSE Fac.membercost * Book.slots END > 30)
ORDER BY COST DESC

```

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

```

SELECT *
FROM (SELECT Fac.name AS "facility name",
CONCAT(Mem.firstname, ' ', Mem.surname) AS "member name",
CASE WHEN Book.memid = 0 THEN Fac.guestcost * Book.slots
ELSE Fac.membercost * Book.slots END AS "cost"
FROM Bookings Book
LEFT JOIN Facilities Fac
ON Book.facid = Fac.facid
LEFT JOIN Members Mem
ON Book.memid = Mem.memid
WHERE (Book.starttime LIKE '2012-09-14%'))
ORDER BY 3 DESC) AS subtable
WHERE subtable.cost > 30

```