/* 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.

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
ORDER BY NAME
LIMIT 0 , 30
```

Tennis Court 1
Tennis Court 2
Massage Room 1
Massage Room 2
Squash Court

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

```
SELECT *
FROM Facilities
WHERE membercost = 0
LIMIT 0 , 30
```

4 facilities. Badminton Court, Snooker Table, Pool Table and Table Tennis.

/* 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`* .20
ORDER BY membercost DESC
LIMIT 0, 30
```

/* 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 )
LIMIT 0 , 30
```

Tennis Court 2 and Massage Room 2

/* Q5: Produce a list of facilities, with each labeled 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. */

Cheap

```
SELECT `name`, `monthlymaintenance` AS Cheap
FROM Facilities
WHERE `monthlymaintenance` < 100
```

Badminton Court
Table Tennis
Squash Court
Snooker Table

Pool Table

Expensive

```
SELECT `name`, `monthlymaintenance` AS Expensive
FROM Facilities
WHERE `monthlymaintenance` > 100
```

Tennis Court 1
Tennis Court 2
Massage Room 1
Massage Room 2

```
/* 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 max(joindate) as 'Last Date', firstname, surname
FROM Members
```

Darren Smith

```
/* 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 m.firstname || ' ' || surname AS fullname, f.name AS facility
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 facility like 'Tennis Court %'
ORDER BY fullname;

/* 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
  firstname || ' ' || surname AS member,
  name AS facility,
```

```
  CASE WHEN m.memid=0 THEN b.slots*f.guestcost
else b.slots*f.membercost
end as 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 DATE(b.starttime) = DATE('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
  firstname || ' ' || surname AS member,
  name AS facility,
  cost
FROM
  (
    SELECT
            firstname,
            surname,
      name,
      CASE WHEN firstname = 'GUEST' THEN guestcost * slots ELSE membercost * slots END
AS cost,
      starttime
    FROM
            members
    INNER JOIN
            bookings
    ON
            members.memid = bookings.memid
    INNER JOIN
            facilities
    ON
            bookings.facid = facilities.facid
) AS table
WHERE
  starttime >= '2012-09-14'
  AND
  starttime < '2012-09-15'
  AND
  cost > 30
ORDER BY cost DESC;
```

GUEST GUEST	Massage Room 2	320
GUEST GUEST	Massage Room 1	160
GUEST GUEST	Massage Room 1	160
GUEST GUEST	Massage Room 1	160
GUEST GUEST	Tennis Court 2	150
GUEST GUEST	Tennis Court 1	75
GUEST GUEST	Tennis Court 1	75
GUEST GUEST	Tennis Court 2	75
GUEST GUEST	Squash Court	70.0
Jemima Farrell	Massage Room 1	39.6
GUEST GUEST	Squash Court	35.0
GUEST GUEST	Squash Court	35.0

/* PART 2: SQLite
/* We now want you to jump over to a local instance of the database on your machine.

Copy and paste the LocalSQLConnection.py script into an empty Jupyter notebook, and run it.

Make sure that the SQLFiles folder containing these files is in your working directory, and that you haven't changed the name of the .db file from 'sqlite\db\pythonsqlite'.

You should see the output from the initial query 'SELECT * FROM FACILITIES'.

Complete the remaining tasks in the Jupyter interface. If you struggle, feel free to go back to the PHPMyAdmin interface as and when you need to.

You'll need to paste your query into the value of the 'query1' variable and run the code block again to get an output.

QUESTIONS:
/* 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,
        sum(CASE WHEN firstname = 'GUEST'
        THEN guestcost * slots
        ELSE membercost * slots
        END) as Total_revenue
FROM Bookings AS b
```

```
INNER JOIN Facilities ON b.facid = Facilities.facid
INNER JOIN Members AS m ON b.memid = m.memid
GROUP BY Facilities.name
HAVING Total_revenue < 1000

Pool Table          270
Snooker Table 240
Table Tennis    180

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

select *
FROM (select m2.surname || ', ' || m2.firstname AS Recommender, m1.recommendedby AS
'Recom. memid', m1.surname || ', ' || m1.firstname AS 'Member Name', m1.memid from
members AS m1
INNER JOIN Members AS m2
on m1.recommendedby = m2.memid) AS info
WHERE memid>0
ORDER BY Recommender;

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

SELECT        f.name, Subquery.Usage
FROM Facilities as f
INNER JOIN (
   SELECT facid, SUM(slots) AS Usage
         FROM Bookings
         WHERE memid >= 1
         GROUP by facid
         ORDER BY SUM(slots) DESC
   )AS Subquery USING (facid)

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

SELECT F.name AS Facility, strftime('%m', B.starttime) as month, SUM(B.slots) as Usage
FROM Bookings AS B
INNER JOIN Facilities AS F ON B.facid = F.facid
INNER JOIN Members as M ON B.memid=M.memid
WHERE M.memid>'0'
GROUP BY Facility, Month
ORDER BY usage DESC
```