/* 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. */

CODE:

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

ANSWER:

     Tennis Courts 1 & 2
     Massage Rooms 1 & 2
     Squash Court

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

CODE:

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

ANSWER: 4

/* 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. */

CODE:

```
SELECT
      facid,
      name AS facility_name,
      membercost,
      monthlymaintenance
FROM Facilities
WHERE
      membercost > 0
      AND membercost < 0.2 * monthlymaintenance;
```

ANSWER:

```
      facid,facility_name,membercost,membermaintenance
      0,Tennis Court 1,5.0,200
      1,Tennis Court 2,5.0,200
      4,Massage Room 1,9.9,3000
      5,Massage Room 2,9.9,3000
      6,Squash Court,3.5,80
```

/* 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. */

CODE:

```
SELECT *
FROM Facilities
WHERE facid IN (1, 5)
```

ANSWER: N/A

/* 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. */

CODE:

```
SELECT
      name AS facility_name,
      monthlymaintenance,
      CASE
            WHEN monthlymaintenance > 100 THEN 'expensive'
            ELSE 'cheap'
      END AS cost_category
FROM Facilities
```

ANSWER:

```
facility_name    monthlymaintenance    cost_category
Tennis Court 1   200   expensive
Tennis Court 2   200   expensive
Badminton Court  50    cheap
Table Tennis     10    cheap
Massage Room 1   3000  expensive
Massage Room 2   3000  expensive
Squash Court     80    cheap
Snooker Table    15    cheap
Pool Table 15    cheap
```

/* 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. */

CODE:

```
SELECT
      firstname,
      surname,
      joindate
FROM
      Members
WHERE
      joindate = (SELECT MAX(joindate) FROM Members)
```

ANSWER: Darren Smith 2012-09-26 18:08:45

/* 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. */

CODE:

```
SELECT DISTINCT
      CONCAT(m.firstname, " ", m.surname) AS member_name
      f.name AS court_name
FROM
      Bookings AS b

JOIN Members AS m ON b.memid = m.memid
JOIN Facilities AS f ON b.facid = f.facid

WHERE
      f.name LIKE 'Tennis Court%'
ORDER BY
      member_name
```

/* 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. */

CODE:

```
SELECT
    f.name AS facility_name,
    CONCAT(m.firstname, ' ', m.surname) AS member_name,
    CASE
        WHEN b.memid = 0 THEN f.guestcost * b.slots
        ELSE f.membercost * b.slots
    END AS cost
FROM
    Bookings AS b
JOIN
    Members AS m ON b.memid = m.memid
JOIN
    Facilities AS f ON b.facid = f.facid
WHERE
    b.starttime >= '2012-09-14' AND b.starttime < '2012-09-15'
    AND (
        (b.memid = 0 AND f.guestcost * b.slots > 30)
        OR (b.memid != 0 AND f.membercost * b.slots > 30)
```

```
    )
ORDER BY
    cost DESC;


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

CODE:

WITH subquery AS (
    SELECT
        b.facid,
        b.memid,
        f.name,
        m.firstname,
           m.surname,
        CASE
            WHEN b.memid = 0 THEN f.guestcost * b.slots
            ELSE f.membercost * b.slots
        END AS cost,
        b.starttime
    FROM
        country_club.Bookings AS b
    JOIN
        country_club.Members AS m ON b.memid = m.memid
    JOIN
        country_club.Facilities AS f ON b.facid = f.facid
    WHERE
        b.starttime >= '2012-09-14' AND b.starttime < '2012-09-15'
)

SELECT
    f.name AS facility_name,
    CONCAT(m.firstname, ' ', m.surname) AS member_name,
    cost
FROM
      subquery
WHERE
    cost > 30
ORDER BY
    cost DESC;

/* 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 thes 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 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! */

CODE:

```
         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
             FACILITIES AS f
         LEFT JOIN
             BOOKINGS AS b ON f.facid = b.facid
         GROUP BY
             f.name
         HAVING
             total_revenue < 1000
         ORDER BY
             total_revenue;
```

ANSWER:

```
       2.6.0
2. Query all tasks
('Table Tennis', 180)
('Snooker Table', 240)
('Pool Table', 270)
```

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

CODE:

```
          SELECT
             m1.surname || ', ' || m1.firstname AS member_name,
             m1.recommendedby AS recommended_by,
             m2.surname || ', ' || m2.firstname AS recommender_name
```

```
        FROM
            MEMBERS AS m1
        LEFT JOIN
            MEMBERS AS m2 ON m1.recommendedby = m2.memid
        ORDER BY
            member_name;
```

ANSWER:

```
('Bader, Florence', '9', 'Stibbons, Ponder')
('Baker, Anne', '9', 'Stibbons, Ponder')
('Baker, Timothy', '13', 'Farrell, Jemima')
('Boothe, Tim', '3', 'Rownam, Tim')
('Butters, Gerald', '1', 'Smith, Darren')
('Coplin, Joan', '16', 'Baker, Timothy')
('Crumpet, Erica', '2', 'Smith, Tracy')
('Dare, Nancy', '4', 'Joplette, Janice')
('Farrell, David', '', None)
('Farrell, Jemima', '', None)
('GUEST, GUEST', '', None)
('Genting, Matthew', '5', 'Butters, Gerald')
('Hunt, John', '30', 'Purview, Millicent')
('Jones, David', '4', 'Joplette, Janice')
('Jones, Douglas', '11', 'Jones, David')
('Joplette, Janice', '1', 'Smith, Darren')
('Mackenzie, Anna', '1', 'Smith, Darren')
('Owen, Charles', '1', 'Smith, Darren')
('Pinker, David', '13', 'Farrell, Jemima')
('Purview, Millicent', '2', 'Smith, Tracy')
('Rownam, Tim', '', None)
('Rumney, Henrietta', '20', 'Genting, Matthew')
('Sarwin, Ramnaresh', '15', 'Bader, Florence')
('Smith, Darren', '', None)
('Smith, Darren', '', None)
('Smith, Jack', '1', 'Smith, Darren')
('Smith, Tracy', '', None)
('Stibbons, Ponder', '6', 'Tracy, Burton')
('Tracy, Burton', '', None)
('Tupperware, Hyacinth', '', None)
('Worthington-Smyth, Henry', '2', 'Smith, Tracy')
```

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

CODE:

```
        SELECT
            f.name AS facility_name,
            COUNT(b.bookid) AS member_usage
        FROM
            FACILITIES AS f
        JOIN
            BOOKINGS AS b ON f.facid = b.facid
        WHERE
            b.memid != 0
        GROUP BY
            f.name
```

```
        ORDER BY
            member_usage DESC;
ANSWER:
      ('Pool Table', 783)
      ('Snooker Table', 421)
      ('Massage Room 1', 421)
      ('Table Tennis', 385)
      ('Badminton Court', 344)
      ('Tennis Court 1', 308)
      ('Tennis Court 2', 276)
      ('Squash Court', 195)
      ('Massage Room 2', 27)


/* Q13: Find the facilities usage by month, but not guests */
CODE:
        SELECT
            f.name AS facility_name,
            strftime('%m', b.starttime) AS month,
            COUNT(b.bookid) AS usage_count
        FROM
            FACILITIES AS f
        JOIN
            BOOKINGS AS b ON f.facid = b.facid
        WHERE
            b.memid != 0
        GROUP BY
            f.name, month
        ORDER BY
            month, usage_count DESC;
ANSWERS:
      ('Pool Table', '07', 103)
      ('Massage Room 1', '07', 77)
      ('Snooker Table', '07', 68)
      ('Tennis Court 1', '07', 65)
      ('Badminton Court', '07', 51)
      ('Table Tennis', '07', 48)
      ('Tennis Court 2', '07', 41)
      ('Squash Court', '07', 23)
      ('Massage Room 2', '07', 4)
      ('Pool Table', '08', 272)
      ('Snooker Table', '08', 154)
      ('Massage Room 1', '08', 153)
      ('Table Tennis', '08', 143)
      ('Badminton Court', '08', 132)
      ('Tennis Court 1', '08', 111)
      ('Tennis Court 2', '08', 109)
      ('Squash Court', '08', 85)
      ('Massage Room 2', '08', 9)
      ('Pool Table', '09', 408)
      ('Snooker Table', '09', 199)
      ('Table Tennis', '09', 194)
      ('Massage Room 1', '09', 191)
```

```
('Badminton Court', '09', 161)
('Tennis Court 1', '09', 132)
('Tennis Court 2', '09', 126)
('Squash Court', '09', 87)
('Massage Room 2', '09', 14)
```