/\* 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 > 0.0;

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

SELECT name

FROM Facilities

WHERE membercost = 0.0;

/\* 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 < (.20 \* 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, monthlymaintenance, expense\_label

FROM Facilities

WHERE monthlymaintenance > 100;

/\* 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 ='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. \*/

SELECT CONCAT(firstname, surname) AS membername, name

FROM Members

JOIN Facilities

ON facid = memid

WHERE name LIKE 'Tennis%';

/\* 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 name, CONCAT(firstname, surname) AS membername, membercost, guestcost

FROM Facilities F

JOIN Members M

ON F.facid = M.memid

JOIN Bookings B

ON M.memid = B.memid

WHERE starttime LIKE '2012-09-14%';

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

SELECT name, CONCAT(firstname, surname) AS membername, membercost, guestcost

FROM Facilities F

JOIN Members M

ON F.facid = M.memid

WHERE EXISTS

(SELECT starttime

FROM Bookings

WHERE starttime LIKE '2012-09-14%');

/\* 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! \*/

SELECT (membercost \* initialoutlay) AS Revenue\_Member, (guestcost \* initialoutlay) AS Revenue\_Guest

FROM Facilities

WHERE (membercost \* initialoutlay) < 1000 AND (guestcost \* initialoutlay) < 10000

ORDER BY (membercost \* initialoutlay), (guestcost \* initialoutlay);

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

SELECT surname, firstname, recommendedby

FROM Members

ORDER BY surname, firstname;

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

SELECT facid, name, membercost, initialoutlay, monthlymaintenance, expense\_label

FROM Facilities

ORDER BY membercost;

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

SELECT facid, name, membercost, initialoutlay, monthlymaintenance, expense\_label

FROM Facilities

ORDER BY monthlymaintenance;