/* 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. */

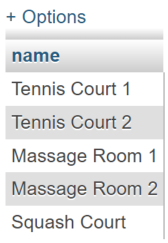
/* QUESTIONS

/* 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;
```

Output:



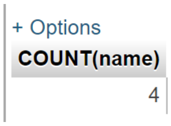
+ Options

name
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 COUNT(name)

FROM Facilities

WHERE membercost = 0;
```

Output:

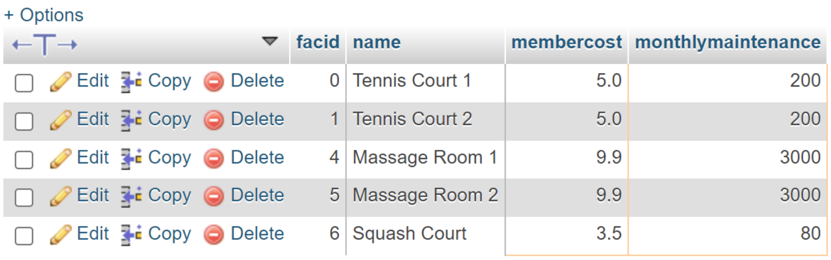


+ Options

COUNT(name)
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. */
SELECT facid, name, membercost, monthlymaintenance
FROM Facilities
WHERE membercost < 0.2*monthlymaintenance AND membercost <>0;
```

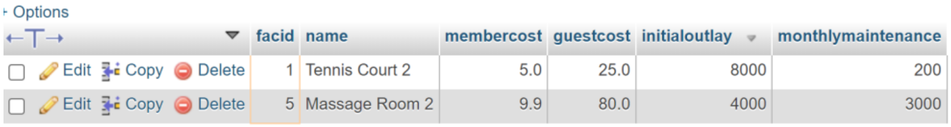
Output:



←T→	facid	name	membercost	monthlymaintenance
□ Edit Copy Delete	0	Tennis Court 1	5.0	200
□ Edit Copy Delete	1	Tennis Court 2	5.0	200
□ Edit Copy Delete	4	Massage Room 1	9.9	3000
□ Edit Copy Delete	5	Massage Room 2	9.9	3000
□ Edit Copy Delete	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. */
 SELECT *
FROM Facilities
WHERE facid IN (1,5);
```

Output:



←T→	facid	name	membercost	guestcost	initialoutlay	monthlymaintenance
□ Edit Copy Delete	1	Tennis Court 2	5.0	25.0	8000	200
□ Edit Copy Delete	5	Massage Room 2	9.9	80.0	4000	3000

```
/* 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
WHEN monthlymaintenance>= 100 THEN 'Expensive' ELSE 'Cheap' END AS Hint
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. */
```

There are two codes:

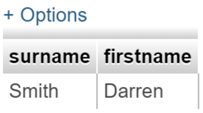
Code1:

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

Code2:

```
SELECT surname, firstname
FROM Members
WHERE memid=1000;
```

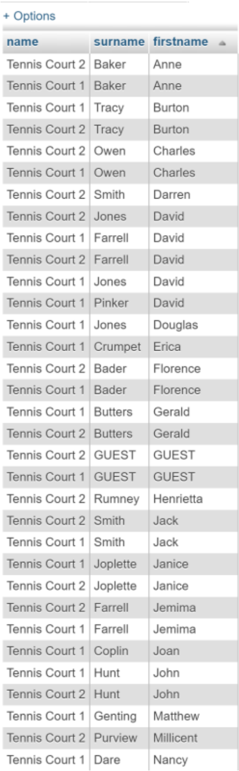
Output:



surname	firstname
Smith	Darren

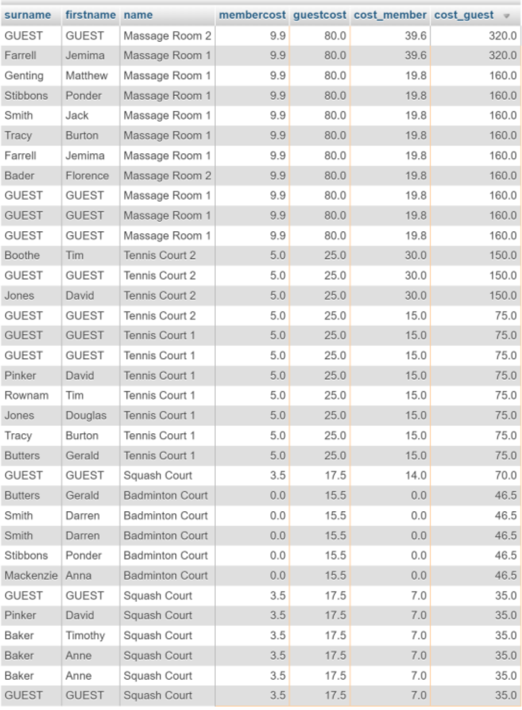
```
/* Q7: Produce a list of all members who have used a tennis court.
Include in your output the name of the court, and the name of the member
formatted as a single column. Ensure no duplicate data, and order by
the member name. */
SELECT DISTINCT name, surname, firstname
FROM Bookings
INNER JOIN Members
ON Bookings.memid=Members.memid
INNER JOIN Facilities
ON Bookings.facid=Facilities.facid
WHERE Facilities.facid IN (0,1)
ORDER BY firstname;
```

Output:



name	surname	firstname ▲
Tennis Court 2	Baker	Anne
Tennis Court 1	Baker	Anne
Tennis Court 1	Tracy	Burton
Tennis Court 2	Tracy	Burton
Tennis Court 2	Owen	Charles
Tennis Court 1	Owen	Charles
Tennis Court 2	Smith	Darren
Tennis Court 2	Jones	David
Tennis Court 1	Farrell	David
Tennis Court 2	Farrell	David
Tennis Court 1	Jones	David
Tennis Court 1	Pinker	David
Tennis Court 1	Jones	Douglas
Tennis Court 1	Crumpet	Erica
Tennis Court 2	Bader	Florence
Tennis Court 1	Bader	Florence
Tennis Court 1	Butters	Gerald
Tennis Court 2	Butters	Gerald
Tennis Court 2	GUEST	GUEST
Tennis Court 1	GUEST	GUEST
Tennis Court 2	Rumney	Henrietta
Tennis Court 2	Smith	Jack
Tennis Court 1	Smith	Jack
Tennis Court 1	Joplette	Janice
Tennis Court 2	Joplette	Janice
Tennis Court 2	Farrell	Jemima
Tennis Court 1	Farrell	Jemima
Tennis Court 1	Coplin	Joan
Tennis Court 1	Hunt	John
Tennis Court 2	Hunt	John
Tennis Court 1	Genting	Matthew
Tennis Court 2	Purview	Millicent
Tennis Court 1	Dare	Nancy

```
/* 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 surname, firstname, name, membercost, guestcost, membercost*slots AS cost_member, guestcost*slots AS cost_guest
FROM Bookings
INNER JOIN Members
ON Bookings.memid=Members.memid
INNER JOIN Facilities
ON Bookings.facid=Facilities.facid
WHERE starttime LIKE '2012-09-14%'AND (membercost*slots>30 or guestcost*slots>30)
ORDER BY cost_guest DESC;
```



surname	firstname	name	membercost	guestcost	cost_member	cost_guest
GUEST	GUEST	Massage Room 2	9.9	80.0	39.6	320.0
Farrell	Jemima	Massage Room 1	9.9	80.0	39.6	320.0
Genting	Matthew	Massage Room 1	9.9	80.0	19.8	160.0
Stibbons	Ponder	Massage Room 1	9.9	80.0	19.8	160.0
Smith	Jack	Massage Room 1	9.9	80.0	19.8	160.0
Tracy	Burton	Massage Room 1	9.9	80.0	19.8	160.0
Farrell	Jemima	Massage Room 1	9.9	80.0	19.8	160.0
Bader	Florence	Massage Room 2	9.9	80.0	19.8	160.0
GUEST	GUEST	Massage Room 1	9.9	80.0	19.8	160.0
GUEST	GUEST	Massage Room 1	9.9	80.0	19.8	160.0
GUEST	GUEST	Massage Room 1	9.9	80.0	19.8	160.0
Boothe	Tim	Tennis Court 2	5.0	25.0	30.0	150.0
GUEST	GUEST	Tennis Court 2	5.0	25.0	30.0	150.0
Jones	David	Tennis Court 2	5.0	25.0	30.0	150.0
GUEST	GUEST	Tennis Court 2	5.0	25.0	15.0	75.0
GUEST	GUEST	Tennis Court 1	5.0	25.0	15.0	75.0
GUEST	GUEST	Tennis Court 1	5.0	25.0	15.0	75.0
Pinker	David	Tennis Court 1	5.0	25.0	15.0	75.0
Rownam	Tim	Tennis Court 1	5.0	25.0	15.0	75.0
Jones	Douglas	Tennis Court 1	5.0	25.0	15.0	75.0
Tracy	Burton	Tennis Court 1	5.0	25.0	15.0	75.0
Butters	Gerald	Tennis Court 1	5.0	25.0	15.0	75.0
GUEST	GUEST	Squash Court	3.5	17.5	14.0	70.0
Butters	Gerald	Badminton Court	0.0	15.5	0.0	46.5
Smith	Darren	Badminton Court	0.0	15.5	0.0	46.5
Smith	Darren	Badminton Court	0.0	15.5	0.0	46.5
Stibbons	Ponder	Badminton Court	0.0	15.5	0.0	46.5
Mackenzie	Anna	Badminton Court	0.0	15.5	0.0	46.5
GUEST	GUEST	Squash Court	3.5	17.5	7.0	35.0
Pinker	David	Squash Court	3.5	17.5	7.0	35.0
Baker	Timothy	Squash Court	3.5	17.5	7.0	35.0
Baker	Anne	Squash Court	3.5	17.5	7.0	35.0
Baker	Anne	Squash Court	3.5	17.5	7.0	35.0
GUEST	GUEST	Squash Court	3.5	17.5	7.0	35.0

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

SELECT surname, firstname, name, membercost, guestcost, cost_member, cost_guest

FROM (SELECT membercost AS cost_member, guest AS cost_guest

    FROM Bookings

    INNER JOIN Members

    ON Bookings.memid=Members.memid

    INNER JOIN Facilities

    ON Bookings.facid=Facilities.facid

    WHERE starttime LIKE '2013-09--14' AND (memebercos*slots OR guestcost*slots>30))

ORDER BY cost_guest 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:

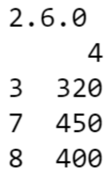
/* Q10: Produce a list of facilities with a total revenue less than 1000.

The output of facility name and total revenue, sorted by revenue. Remember

that there's a different cost for guests and members! */

```
step1=create_connection('C:/Users/shaha/Desktop/Springboard/SQL/SQLFiles Tier 1///sqlite_db_pythonsqlite.db')

curr=step1.cursor()

rs=curr.execute("SELECT * FROM Facilities")

df=pd.DataFrame(rs.fetchall())

print(df[[4]][df[4]<1000])
```

Output:

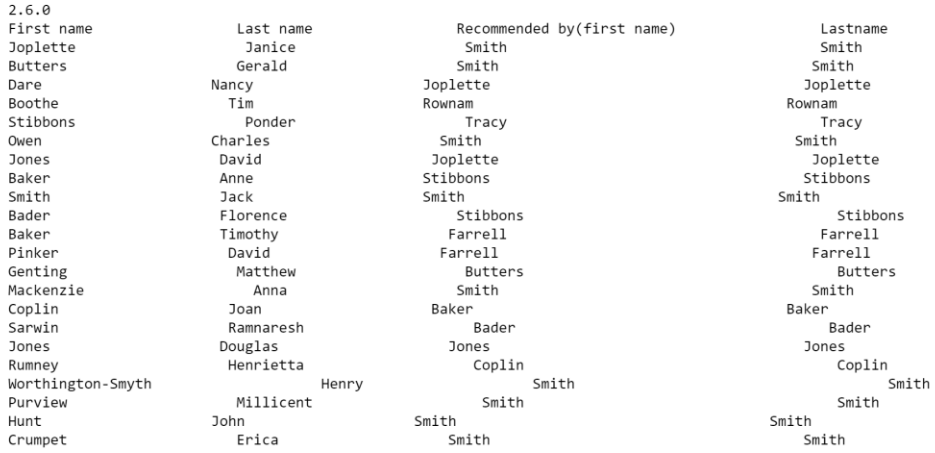


```
2.6.0
     4
3  320
7  450
8  400
```

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

```
 step1=create_connection('C:/Users/shaha/Desktop/Springboard/SQL/SQLFiles Tier 1///sqlite_db_pythonsqlite.db')

curr=step1.cursor()

rs=curr.execute("SELECT * FROM Members")

df=pd.DataFrame(rs.fetchall())

df.shape

df

for i in range(31):

   if df.iloc[i,6]=='':

       df.iloc[i,6]=0

print('First name','            ','Last name','            ','Recommended by(first name)','            ','Lastname')

for i in range(31):
   if df.iloc[i,6]!=0:
       a=int(df.iloc[i,6])
       print(df.iloc[i,1],'              ',df.iloc[i,2],'              ',df.iloc[a,1],'                            ',df.iloc[a,1])
```

Output:

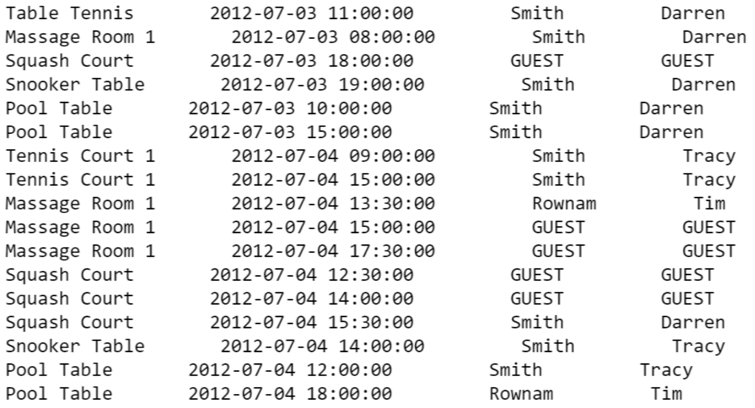


```
2.6.0
First name                 Last name                 Recommended by(first name)                 Lastname
Joplette                    Janice                    Smith                                     Smith
Butters                    Gerald                    Smith                                     Smith
Dare                    Nancy                    Joplette                                     Joplette
Boothe                    Tim                    Rownam                                     Rownam
Stibbons                    Ponder                    Tracy                                     Tracy
Owen                    Charles                    Smith                                     Smith
Jones                    David                    Joplette                                     Joplette
Baker                    Anne                    Stibbons                                     Stibbons
Smith                    Jack                    Smith                                     Smith
Bader                    Florence                    Stibbons                                     Stibbons
Baker                    Timothy                    Farrell                                     Farrell
Pinker                    David                    Farrell                                     Farrell
Genting                    Matthew                    Butters                                     Butters
Mackenzie                    Anna                    Smith                                     Smith
Coplin                    Joan                    Baker                                     Baker
Sarwin                    Ramnaresh                    Bader                                     Bader
Jones                    Douglas                    Jones                                     Jones
Rumney                    Henrietta                    Coplin                                     Coplin
Worthington-Smyth                    Henry                    Smith                                     Smith
Purview                    Millicent                    Smith                                     Smith
Hunt                    John                    Smith                                     Smith
Crumpet                    Erica                    Smith                                     Smith
```

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

```
step1=create_connection('C:/Users/shaha/Desktop/Springboard/SQL/SQLFiles Tier 1///sqlite_db_pythonsqlite.db')
curr=step1.cursor()
rs=curr.execute("SELECT * FROM Bookings INNER JOIN Members ON Bookings.memid=Members.memid INNER JOIN Facilities ON Bookings.facid=Facilities.facid")
df=pd.DataFrame(rs.fetchall())
for i in range (4042):
  print(df.iloc[i,14],'    ',df.iloc[i,3],'      ',df.iloc[i,6],'      ',df.iloc[i,7])
```

Output:



```
Table Tennis       2012-07-03 11:00:00         Smith         Darren
Massage Room 1       2012-07-03 08:00:00         Smith         Darren
Squash Court       2012-07-03 18:00:00         GUEST         GUEST
Snooker Table       2012-07-03 19:00:00         Smith         Darren
Pool Table       2012-07-03 10:00:00         Smith         Darren
Pool Table       2012-07-03 15:00:00         Smith         Darren
Tennis Court 1       2012-07-04 09:00:00         Smith         Tracy
Tennis Court 1       2012-07-04 15:00:00         Smith         Tracy
Massage Room 1       2012-07-04 13:30:00         Rownam         Tim
Massage Room 1       2012-07-04 15:00:00         GUEST         GUEST
Massage Room 1       2012-07-04 17:30:00         GUEST         GUEST
Squash Court       2012-07-04 12:30:00         GUEST         GUEST
Squash Court       2012-07-04 14:00:00         GUEST         GUEST
Squash Court       2012-07-04 15:30:00         Smith         Darren
Snooker Table       2012-07-04 14:00:00         Smith         Tracy
Pool Table       2012-07-04 12:00:00         Smith         Tracy
Pool Table       2012-07-04 18:00:00         Rownam         Tim
```

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

step1=create_connection('C:/Users/shaha/Desktop/Springboard/SQL/SQLFiles Tier 1///sqlite_db_pythonsqlite.db')

curr=step1.cursor()

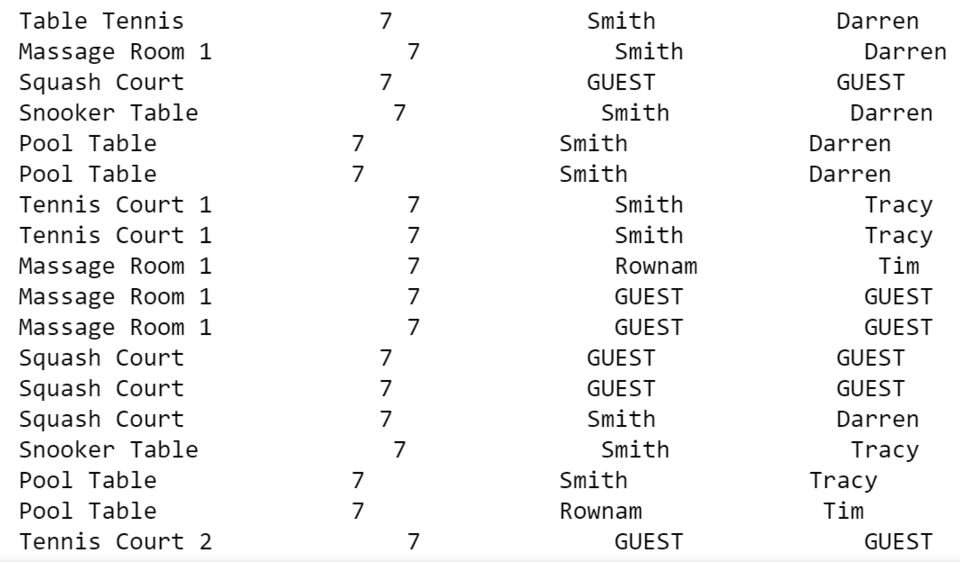
rs=curr.execute("SELECT * FROM Bookings INNER JOIN Members ON Bookings.memid=Members.memid INNER JOIN Facilities ON Bookings.facid=Facilities.facid")

df=pd.DataFrame(rs.fetchall())

df['date_month']=pd.DatetimeIndex(df[3]).month

```
for i in range (4042):
    print(df.iloc[i,14],'          ',df.iloc[i,19],'          ',df.iloc[i,6],'          ',df.iloc[i,7])
```

Output:



```
Table Tennis              7              Smith             Darren
Massage Room 1              7              Smith             Darren
Squash Court              7              GUEST             GUEST
Snooker Table               7              Smith             Darren
Pool Table              7              Smith             Darren
Pool Table              7              Smith             Darren
Tennis Court 1              7              Smith             Tracy
Tennis Court 1              7              Smith             Tracy
Massage Room 1              7              Rownam             Tim
Massage Room 1              7              GUEST             GUEST
Massage Room 1              7              GUEST             GUEST
Squash Court              7              GUEST             GUEST
Squash Court              7              GUEST             GUEST
Squash Court              7              Smith             Darren
Snooker Table              7              Smith             Tracy
Pool Table              7              Smith             Tracy
Pool Table              7              Rownam             Tim
Tennis Court 2              7              GUEST             GUEST
```