Case Study - Country Club

by: Sonjoy Das, PhD

The objective of this notebook is to illustrate the power of Structured Query Language (SQL) to solve a few business questions related to a country club. We connect to a local instance of the database containing information about the country club using Python and SQLite in this Juptyer Notebook. Depending on the business question, we retrieve particular pieces of information from the database and translate our requests into SQL queries. The output format of each querry is similar to what we would see in a typical relational database management system software such as `MySQL`. The `sqlite3` module is used here only to access the database and the `pandas` library is used only to format the output of each querry similar to what we would obtain in a standard relational database management system software such as `MySQL`. No data manipulation or analysis is carried out through python. **This notebook exclusively uses SQL querries for data processing and analysis as and when required.**

1.1 Imports

In [1]:
```
import sqlite3
import pandas as pd
```

1.2 Connect to Database

In [2]:
```
database = "sqlite_db_pythonsqlite.db"
conn = sqlite3.connect(database)
print(sqlite3.version)
```

```
2.6.0
```

1.3 Define Function to Execute SQL Querry

In [3]:
```
# Define a function to execute the SQL queery and print them as pandas's dataframe a
# the size of panda's dataframe.
def execute_print_sql_query(query):
    cursor = conn.cursor()

    rs = cursor.execute(query)
    # Extract the column names of the table
    # Reference: https://www.daniweb.com/programming/software-development/threads/12
    col_name_list = [tuple[0] for tuple in cursor.description]

    df = pd.DataFrame(cursor.fetchall())
    # Set the column name of pandas dataframe to keys or column names of table extra
    df.columns = col_name_list

    print("Size of panda's dataframe: " + str(df.shape)+"\n")
    print(df)
    cursor.close()
```

1.4 Explore Database and Tables

In [4]:
```
# Print how many tables are there in the database
query = """
        SELECT name
        FROM sqlite_master
        WHERE type='table';
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (3, 1)

         name
0    Bookings
1  Facilities
2     Members
```

There are 3 tables: `Bookings`, `Facilities`, and `Members`.

1.4.1 `facilities` Table

In [5]:
```
# Print and explore facilities table
query = """
        SELECT *
        FROM facilities;
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (9, 6)

   facid             name  membercost  guestcost  initialoutlay  \
0      0   Tennis Court 1         5.0       25.0          10000
1      1   Tennis Court 2         5.0       25.0           8000
2      2  Badminton Court         0.0       15.5           4000
3      3     Table Tennis         0.0        5.0            320
4      4   Massage Room 1         9.9       80.0           4000
5      5   Massage Room 2         9.9       80.0           4000
6      6     Squash Court         3.5       17.5           5000
7      7    Snooker Table         0.0        5.0            450
8      8       Pool Table         0.0        5.0            400

   monthlymaintenance
0                 200
1                 200
2                  50
3                  10
4                3000
5                3000
6                  80
7                  15
8                  15
```

1.4.2 `members` Table

In [6]:
```
# Print and explore members table
query = """
        SELECT *
        FROM members;
        """
execute_print_sql_query(query)
```

Size of panda's dataframe: (31, 8)

	memid	surname	firstname	address	zipcode	telephone
0	0	GUEST	GUEST	GUEST	0	(000) 000-0000
1	1	Smith	Darren	8 Bloomsbury Close, Boston	4321	555-555-5555
2	2	Smith	Tracy	8 Bloomsbury Close, New York	4321	555-555-5555
3	3	Rownam	Tim	23 Highway Way, Boston	23423	(844) 693-0723
4	4	Joplette	Janice	20 Crossing Road, New York	234	(833) 942-4710
5	5	Butters	Gerald	1065 Huntingdon Avenue, Boston	56754	(844) 078-4130
6	6	Tracy	Burton	3 Tunisia Drive, Boston	45678	(822) 354-9973
7	7	Dare	Nancy	6 Hunting Lodge Way, Boston	10383	(833) 776-4001
8	8	Boothe	Tim	3 Bloomsbury Close, Reading, 00234	234	(811) 433-2547
9	9	Stibbons	Ponder	5 Dragons Way, Winchester	87630	(833) 160-3900
10	10	Owen	Charles	52 Cheshire Grove, Winchester, 28563	28563	(855) 542-5251
11	11	Jones	David	976 Gnats Close, Reading	33862	(844) 536-8036
12	12	Baker	Anne	55 Powdery Street, Boston	80743	844-076-5141
13	13	Farrell	Jemima	103 Firth Avenue, North Reading	57392	(855) 016-0163
14	14	Smith	Jack	252 Binkington Way, Boston	69302	(822) 163-3254
15	15	Bader	Florence	264 Ursula Drive, Westford	84923	(833) 499-3527
16	16	Baker	Timothy	329 James Street, Reading	58393	833-941-0824
17	17	Pinker	David	5 Impreza Road, Boston	65332	811 409-6734
18	20	Genting	Matthew	4 Nunnington Place, Wingfield, Boston	52365	(811) 972-1377
19	21	Mackenzie	Anna	64 Perkington Lane, Reading	64577	(822) 661-2898
20	22	Coplin	Joan	85 Bard Street, Bloomington, Boston	43533	(822) 499-2232
21	24	Sarwin	Ramnaresh	12 Bullington Lane, Boston	65464	(822) 413-1470
22	26	Jones	Douglas	976 Gnats Close, Reading	11986	844 536-8036
23	27	Rumney	Henrietta	3 Burkington Plaza, Boston	78533	(822) 989-8876
24	28	Farrell	David	437 Granite Farm Road, Westford	43532	(855) 755-9876
25	29	Worthington-Smyth	Henry	55 Jagbi Way, North Reading	97676	(855) 894-3758
26	30	Purview	Millicent	641 Drudgery Close, Burnington, Boston	34232	(855) 941-9786
27	33	Tupperware	Hyacinth	33 Cheerful Plaza, Drake Road, Westford	68666	(822) 665-5327
28	35	Hunt	John	5 Bullington Lane, Boston	54333	(899) 720-6978
29	36	Crumpet	Erica	Crimson Road, North Reading	75655	(811) 732-4816
30	37	Smith	Darren	3 Funktown, Denzington, Boston	66796	(822) 577-3541

	recommendedby	joindate
0		2012-07-01 00:00:00
1		2012-07-02 12:02:05
2		2012-07-02 12:08:23
3		2012-07-03 09:32:15
4	1	2012-07-03 10:25:05
5	1	2012-07-09 10:44:09
6		2012-07-15 08:52:55
7	4	2012-07-25 08:59:12
8	3	2012-07-25 16:02:35
9	6	2012-07-25 17:09:05
10	1	2012-08-03 19:42:37
11	4	2012-08-06 16:32:55
12	9	2012-08-10 14:23:22
13		2012-08-10 14:28:01
14	1	2012-08-10 16:22:05
15	9	2012-08-10 17:52:03
16	13	2012-08-15 10:34:25
17	13	2012-08-16 11:32:47
18	5	2012-08-19 14:55:55
19	1	2012-08-26 09:32:05
20	16	2012-08-29 08:32:41
21	15	2012-09-01 08:44:42
22	11	2012-09-02 18:43:05
23	20	2012-09-05 08:42:35
24		2012-09-15 08:22:05
25	2	2012-09-17 12:27:15
26	2	2012-09-18 19:04:01
27		2012-09-18 19:32:05
28	30	2012-09-19 11:32:45
29	2	2012-09-22 08:36:38
30		2012-09-26 18:08:45

1.4.3 `bookings` Table

In [7]:

```
# Print and explore bookings table
query = """
        SELECT *
        FROM bookings;
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (4043, 5)
```

	bookid	facid	memid	starttime	slots
0	0	3	1	2012-07-03 11:00:00	2
1	1	4	1	2012-07-03 08:00:00	2
2	2	6	0	2012-07-03 18:00:00	2
3	3	7	1	2012-07-03 19:00:00	2
4	4	8	1	2012-07-03 10:00:00	1
...	...	...	...	...	...
4038	4038	8	29	2012-09-30 16:30:00	2
4039	4039	8	29	2012-09-30 18:00:00	1
4040	4040	8	21	2012-09-30 18:30:00	1
4041	4041	8	16	2012-09-30 19:00:00	1
4042	4042	8	29	2012-09-30 19:30:00	1

```
[4043 rows x 5 columns]
```

1.5 SQL Querries for Business Questions

Question 1

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.

In [8]:
```
query = """
        SELECT name, membercost
        FROM facilities
        WHERE membercost <> 0;
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (5, 2)

             name  membercost
0  Tennis Court 1         5.0
1  Tennis Court 2         5.0
2  Massage Room 1         9.9
3  Massage Room 2         9.9
4    Squash Court         3.5
```

Question 2

How many facilities do not charge a fee to members?

In [9]:
```
query = """
        SELECT COUNT(name)
        FROM facilities
        WHERE membercost = 0;
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (1, 1)

   COUNT(name)
0            4
```

Question 3

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.

In [10]:
```
query = """
        SELECT
            facid, name AS facilityname, membercost,
            0.2*monthlymaintenance,
            monthlymaintenance
        FROM facilities
        WHERE membercost < 0.2*monthlymaintenance;
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (9, 5)

   facid     facilityname  membercost  0.2*monthlymaintenance  \
0      0   Tennis Court 1         5.0                    40.0
1      1   Tennis Court 2         5.0                    40.0
2      2  Badminton Court         0.0                    10.0
3      3     Table Tennis         0.0                     2.0
4      4   Massage Room 1         9.9                   600.0
5      5   Massage Room 2         9.9                   600.0
6      6     Squash Court         3.5                    16.0
7      7    Snooker Table         0.0                     3.0
8      8       Pool Table         0.0                     3.0
```

```
   monthlymaintenance
0                 200
1                 200
2                  50
3                  10
4                3000
5                3000
6                  80
7                  15
8                  15
```

Question 4

Write an SQL query to retrieve the details of facilities with ID 1 and 5. Try writing the query without using the OR operator.

In [11]:

```
query = """
        SELECT *
        FROM facilities
        WHERE facid IN (1,5);
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (2, 6)

   facid            name  membercost  guestcost  initialoutlay  \
0      1  Tennis Court 2         5.0         25           8000
1      5  Massage Room 2         9.9         80           4000

   monthlymaintenance
0                 200
1                3000
```

Question 5

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.

In [12]:

```
query = """
        SELECT name, monthlymaintenance,
            CASE WHEN monthlymaintenance < 100 THEN 'cheap'
                 WHEN monthlymaintenance BETWEEN 100 AND 1000 THEN 'expensive'
                 ELSE 'ultra expensive' END AS 'cheap/expensive'
        FROM facilities;
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (9, 3)

              name  monthlymaintenance  cheap/expensive
0   Tennis Court 1                 200        expensive
1   Tennis Court 2                 200        expensive
2  Badminton Court                  50            cheap
3     Table Tennis                  10            cheap
4   Massage Room 1                3000  ultra expensive
5   Massage Room 2                3000  ultra expensive
6     Squash Court                  80            cheap
7    Snooker Table                  15            cheap
8       Pool Table                  15            cheap
```

Question 6

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.

In [13]:
```
query = """
        SELECT firstname, surname AS lastname, joindate, memid
        FROM members
        WHERE joindate = (SELECT MAX(joindate)
                          FROM MEMBERS);
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (1, 4)

  firstname lastname             joindate  memid
0    Darren    Smith  2012-09-26 18:08:45     37
```

Question 7

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.

Note: We include `memid` in our final output to ensure that two similar names (such as `Smith, Darren` with `memid = 1` and `memid = 37`) can be identified clearly.

In [14]:
```
query = """
        SELECT facility, (surname||', '||firstname) AS member, memid
        FROM members
        INNER JOIN
                (SELECT DISTINCT memid, facility
                 FROM bookings
                 INNER JOIN
                    (SELECT facid, name AS facility
                     FROM facilities
                     WHERE name IN ('Tennis Court 1', 'Tennis Court 2'))
                 USING(facid))
        USING(memid)
        ORDER BY member;
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (46, 3)

          facility              member  memid
0   Tennis Court 2     Bader, Florence     15
1   Tennis Court 1     Bader, Florence     15
2   Tennis Court 1         Baker, Anne     12
3   Tennis Court 2         Baker, Anne     12
4   Tennis Court 2      Baker, Timothy     16
5   Tennis Court 1      Baker, Timothy     16
6   Tennis Court 2         Boothe, Tim      8
7   Tennis Court 1         Boothe, Tim      8
8   Tennis Court 1     Butters, Gerald      5
9   Tennis Court 2     Butters, Gerald      5
10  Tennis Court 1        Coplin, Joan     22
11  Tennis Court 1      Crumpet, Erica     36
12  Tennis Court 2         Dare, Nancy      7
13  Tennis Court 1         Dare, Nancy      7
14  Tennis Court 1      Farrell, David     28
15  Tennis Court 2      Farrell, David     28
16  Tennis Court 2     Farrell, Jemima     13
17  Tennis Court 1     Farrell, Jemima     13
18  Tennis Court 2        GUEST, GUEST      0
19  Tennis Court 1        GUEST, GUEST      0
```

```
20  Tennis Court 1    Genting, Matthew     20
21  Tennis Court 1          Hunt, John     35
22  Tennis Court 2          Hunt, John     35
23  Tennis Court 2        Jones, David     11
24  Tennis Court 1        Jones, David     11
25  Tennis Court 1      Jones, Douglas     26
26  Tennis Court 1    Joplette, Janice      4
27  Tennis Court 2    Joplette, Janice      4
28  Tennis Court 1       Owen, Charles     10
29  Tennis Court 2       Owen, Charles     10
30  Tennis Court 1       Pinker, David     17
31  Tennis Court 2  Purview, Millicent     30
32  Tennis Court 2         Rownam, Tim      3
33  Tennis Court 1         Rownam, Tim      3
34  Tennis Court 2   Rumney, Henrietta     27
35  Tennis Court 2   Sarwin, Ramnaresh     24
36  Tennis Court 1   Sarwin, Ramnaresh     24
37  Tennis Court 2       Smith, Darren      1
38  Tennis Court 1         Smith, Jack     14
39  Tennis Court 2         Smith, Jack     14
40  Tennis Court 1        Smith, Tracy      2
41  Tennis Court 2        Smith, Tracy      2
42  Tennis Court 2    Stibbons, Ponder      9
43  Tennis Court 1    Stibbons, Ponder      9
44  Tennis Court 2       Tracy, Burton      6
45  Tennis Court 1       Tracy, Burton      6
```

Well, it seems only one of `Smith, Darren`'s (with `memid = 1`) used a tennis court. The other `Smith, Darren` (with `memid = 37`) did not use any tennis court. However, recall that the `Smith, Darren` (with `memid = 37`) was the last member to join (see Question 6). So, he possibly did not explore all faciltities until the time when this database was generated!

In [15]:

```
# Another way: Common Table Expression
query = """
        WITH s AS (
             SELECT memid, name AS facility
             FROM bookings AS b
             LEFT JOIN facilities AS f
             ON f.facid = b.facid
             WHERE name IN ('Tennis Court 1', 'Tennis Court 2')
        )
        SELECT DISTINCT facility, (surname||', '||firstname) AS member, memid
        FROM members
        INNER JOIN s
        USING(memid)
        ORDER BY member;
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (46, 3)

          facility              member  memid
0   Tennis Court 2     Bader, Florence     15
1   Tennis Court 1     Bader, Florence     15
2   Tennis Court 1         Baker, Anne     12
3   Tennis Court 2         Baker, Anne     12
4   Tennis Court 2      Baker, Timothy     16
5   Tennis Court 1      Baker, Timothy     16
6   Tennis Court 2         Boothe, Tim      8
7   Tennis Court 1         Boothe, Tim      8
8   Tennis Court 1     Butters, Gerald      5
9   Tennis Court 2     Butters, Gerald      5
10  Tennis Court 1        Coplin, Joan     22
11  Tennis Court 1      Crumpet, Erica     36
```

12	Tennis Court 2	Dare, Nancy	7
13	Tennis Court 1	Dare, Nancy	7
14	Tennis Court 1	Farrell, David	28
15	Tennis Court 2	Farrell, David	28
16	Tennis Court 2	Farrell, Jemima	13
17	Tennis Court 1	Farrell, Jemima	13
18	Tennis Court 2	GUEST, GUEST	0
19	Tennis Court 1	GUEST, GUEST	0
20	Tennis Court 1	Genting, Matthew	20
21	Tennis Court 1	Hunt, John	35
22	Tennis Court 2	Hunt, John	35
23	Tennis Court 2	Jones, David	11
24	Tennis Court 1	Jones, David	11
25	Tennis Court 1	Jones, Douglas	26
26	Tennis Court 1	Joplette, Janice	4
27	Tennis Court 2	Joplette, Janice	4
28	Tennis Court 1	Owen, Charles	10
29	Tennis Court 2	Owen, Charles	10
30	Tennis Court 1	Pinker, David	17
31	Tennis Court 2	Purview, Millicent	30
32	Tennis Court 2	Rownam, Tim	3
33	Tennis Court 1	Rownam, Tim	3
34	Tennis Court 2	Rumney, Henrietta	27
35	Tennis Court 2	Sarwin, Ramnaresh	24
36	Tennis Court 1	Sarwin, Ramnaresh	24
37	Tennis Court 2	Smith, Darren	1
38	Tennis Court 1	Smith, Jack	14
39	Tennis Court 2	Smith, Jack	14
40	Tennis Court 1	Smith, Tracy	2
41	Tennis Court 2	Smith, Tracy	2
42	Tennis Court 2	Stibbons, Ponder	9
43	Tennis Court 1	Stibbons, Ponder	9
44	Tennis Court 2	Tracy, Burton	6
45	Tennis Court 1	Tracy, Burton	6

Question 8

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.

In [16]:

```
query = """
        SELECT  f.name AS facilityname,
                (surname||', '||firstname) AS member,
                CASE WHEN memid = 0 THEN slots*guestcost
                     WHEN memid BETWEEN 1 AND 37 THEN slots*membercost
                     END AS cost,
                starttime
        FROM bookings
        INNER JOIN members
        USING (memid)
        INNER JOIN facilities AS f
        USING(facid)
        WHERE starttime LIKE '%2012-09-14%' AND cost > 30
        ORDER BY cost DESC;
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (12, 4)
```

	facilityname	member	cost	starttime
0	Massage Room 2	GUEST, GUEST	320.0	2012-09-14 11:00:00
1	Massage Room 1	GUEST, GUEST	160.0	2012-09-14 09:00:00

```
2   Massage Room 1     GUEST, GUEST  160.0  2012-09-14 13:00:00
3   Massage Room 1     GUEST, GUEST  160.0  2012-09-14 16:00:00
4   Tennis Court 2     GUEST, GUEST  150.0  2012-09-14 17:00:00
5   Tennis Court 1     GUEST, GUEST   75.0  2012-09-14 16:00:00
6   Tennis Court 1     GUEST, GUEST   75.0  2012-09-14 19:00:00
7   Tennis Court 2     GUEST, GUEST   75.0  2012-09-14 14:00:00
8     Squash Court     GUEST, GUEST   70.0  2012-09-14 09:30:00
9   Massage Room 1  Farrell, Jemima   39.6  2012-09-14 14:00:00
10    Squash Court     GUEST, GUEST   35.0  2012-09-14 12:30:00
11    Squash Court     GUEST, GUEST   35.0  2012-09-14 15:00:00
```

Question 9

This time, produce the same result as in Q8, but using a subquery.

In [17]:

```
query = """
        SELECT  f.name AS facilityname, (surname||', '||firstname) AS member,
                CASE WHEN memid = 0 THEN slots*guestcost
                     WHEN memid BETWEEN 1 AND 37 THEN slots*membercost
                     END AS cost,
                starttime
        FROM facilities AS f
        INNER JOIN
            (SELECT facid, memid, slots, starttime
             FROM bookings
            ) -- using a subquery
        USING(facid)
        INNER JOIN members
        USING(memid)
        WHERE starttime LIKE '%2012-09-14%' AND cost > 30
        ORDER BY cost DESC;
        """

execute_print_sql_query(query)
```

```
Size of panda's dataframe: (12, 4)

      facilityname           member   cost            starttime
0   Massage Room 2     GUEST, GUEST  320.0  2012-09-14 11:00:00
1   Massage Room 1     GUEST, GUEST  160.0  2012-09-14 09:00:00
2   Massage Room 1     GUEST, GUEST  160.0  2012-09-14 13:00:00
3   Massage Room 1     GUEST, GUEST  160.0  2012-09-14 16:00:00
4   Tennis Court 2     GUEST, GUEST  150.0  2012-09-14 17:00:00
5   Tennis Court 1     GUEST, GUEST   75.0  2012-09-14 16:00:00
6   Tennis Court 1     GUEST, GUEST   75.0  2012-09-14 19:00:00
7   Tennis Court 2     GUEST, GUEST   75.0  2012-09-14 14:00:00
8     Squash Court     GUEST, GUEST   70.0  2012-09-14 09:30:00
9   Massage Room 1  Farrell, Jemima   39.6  2012-09-14 14:00:00
10    Squash Court     GUEST, GUEST   35.0  2012-09-14 12:30:00
11    Squash Court     GUEST, GUEST   35.0  2012-09-14 15:00:00
```

In [18]:

```
# Another way: Without CASE Statement and using SET operation & Common Table Express
query = """
        WITH s AS(
            SELECT memid, name AS facilityname, slots*membercost AS cost,
                    starttime
            FROM bookings
            INNER JOIN facilities
            USING(facid)
            WHERE memid <> 0 AND starttime LIKE '%2012-09-14%' AND cost > 30 -- usin
            UNION
            SELECT memid, name AS facilityname, slots*guestcost AS cost,
                    starttime
```

```
            FROM bookings
            INNER JOIN facilities
            USING(facid)
            WHERE memid = 0 AND starttime LIKE '%2012-09-14%' AND cost > 30 -- using
        )
        SELECT facilityname, (surname||', '||firstname) AS member, cost, starttime
        FROM members
        INNER JOIN s
        USING(memid)
        ORDER BY cost DESC;
        """

execute_print_sql_query(query)
```

```
Size of panda's dataframe: (12, 4)

      facilityname           member   cost            starttime
0   Massage Room 2     GUEST, GUEST  320.0  2012-09-14 11:00:00
1   Massage Room 1     GUEST, GUEST  160.0  2012-09-14 09:00:00
2   Massage Room 1     GUEST, GUEST  160.0  2012-09-14 13:00:00
3   Massage Room 1     GUEST, GUEST  160.0  2012-09-14 16:00:00
4   Tennis Court 2     GUEST, GUEST  150.0  2012-09-14 17:00:00
5   Tennis Court 1     GUEST, GUEST   75.0  2012-09-14 16:00:00
6   Tennis Court 1     GUEST, GUEST   75.0  2012-09-14 19:00:00
7   Tennis Court 2     GUEST, GUEST   75.0  2012-09-14 14:00:00
8     Squash Court     GUEST, GUEST   70.0  2012-09-14 09:30:00
9   Massage Room 1  Farrell, Jemima   39.6  2012-09-14 14:00:00
10    Squash Court     GUEST, GUEST   35.0  2012-09-14 12:30:00
11    Squash Court     GUEST, GUEST   35.0  2012-09-14 15:00:00
```

Question 10

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!

In [19]:

```
# Using HAVING clause
query = """
        SELECT facility, SUM(revenue) AS totalrevenue
        FROM
            (SELECT name AS facility,
                CASE WHEN memid = 0 THEN slots*guestcost
                     WHEN memid BETWEEN 1 AND 37 THEN slots*membercost
                     END AS revenue
             FROM bookings
             INNER JOIN facilities
             USING(facid)
            )
        GROUP BY facility
        HAVING SUM(revenue) < 1000 -- Use HAVING clause here
--        HAVING totalrevenue < 1000 -- Use HAVING clause here
--        WHERE totalrevenue < 1000 -- WHERE clause will cause error here because it
        ORDER BY totalrevenue;
        """

execute_print_sql_query(query)
```

```
Size of panda's dataframe: (3, 2)

       facility  totalrevenue
0   Table Tennis           180
1  Snooker Table           240
2     Pool Table           270
```

In [20]:
```
# Without HAVING clause. Using WHERE clause.
query = """
        SELECT facility, (m.total_revenue+g.total_revenue) AS totalrevenue
        FROM
            (SELECT facility, SUM(revenue) AS total_revenue
             FROM (SELECT facid, name AS facility, slots*membercost AS revenue
                   FROM bookings
                   INNER JOIN facilities
                   USING(facid)
                   WHERE memid <> 0
                   )
             GROUP BY facility) AS m
        INNER JOIN
            (SELECT facility, SUM(revenue) AS total_revenue
            FROM (SELECT facid, name AS facility, slots*guestcost AS revenue
                  FROM bookings
                  INNER JOIN facilities
                  USING(facid)
                  WHERE memid = 0
                 )
            GROUP BY facility) AS g
        USING(facility)
        WHERE totalrevenue < 1000
        ORDER BY totalrevenue;
        """

execute_print_sql_query(query)
```

```
Size of panda's dataframe: (3, 2)

        facility  totalrevenue
0   Table Tennis           180
1  Snooker Table           240
2     Pool Table           270
```

Question 11

Produce a report of members and who recommended them in alphabetic surname,firstname order.

In [21]:
```
query = """
        SELECT r.memid, (r.surname||', '||r.firstname) AS member,
               (m.surname||', '||m.firstname) AS recommender, r.recommendedby
        FROM members as m
        INNER JOIN members as r
        ON m.memid = r.recommendedby
        ORDER BY member;
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (22, 4)

    memid                  member         recommender recommendedby
0      15          Bader, Florence    Stibbons, Ponder             9
1      12              Baker, Anne    Stibbons, Ponder             9
2      16            Baker, Timothy     Farrell, Jemima            13
3       8              Boothe, Tim         Rownam, Tim             3
4       5          Butters, Gerald       Smith, Darren             1
5      22             Coplin, Joan      Baker, Timothy            16
6      36           Crumpet, Erica        Smith, Tracy             2
7       7              Dare, Nancy    Joplette, Janice             4
8      20         Genting, Matthew     Butters, Gerald             5
9      35               Hunt, John  Purview, Millicent            30
10     11             Jones, David    Joplette, Janice             4
```

```
11     26            Jones, Douglas        Jones, David            11
12      4          Joplette, Janice       Smith, Darren             1
13     21           Mackenzie, Anna       Smith, Darren             1
14     10             Owen, Charles       Smith, Darren             1
15     17             Pinker, David     Farrell, Jemima            13
16     30       Purview, Millicent        Smith, Tracy             2
17     27         Rumney, Henrietta    Genting, Matthew            20
18     24         Sarwin, Ramnaresh     Bader, Florence            15
19     14               Smith, Jack       Smith, Darren             1
20      9          Stibbons, Ponder       Tracy, Burton             6
21     29  Worthington-Smyth, Henry        Smith, Tracy             2
```

Question 12

Find the facilities with their usage by member, but not guests.

In [22]:

```
query = """
        SELECT name as facility, (surname||', '||firstname) AS member, SUM(0.5*slots
        FROM bookings
        INNER JOIN facilities
        USING(facid)
        INNER JOIN members
        USING(memid)
        WHERE memid <> 0
        GROUP BY facility, member;
        """

execute_print_sql_query(query)
```

```
Size of panda's dataframe: (202, 3)

            facility            member  usagehour
0    Badminton Court   Bader, Florence       13.5
1    Badminton Court       Baker, Anne       15.0
2    Badminton Court    Baker, Timothy       10.5
3    Badminton Court       Boothe, Tim       18.0
4    Badminton Court   Butters, Gerald       31.5
..               ...               ...        ...
197   Tennis Court 2     Smith, Darren       28.5
198   Tennis Court 2       Smith, Jack        1.5
199   Tennis Court 2      Smith, Tracy        3.0
200   Tennis Court 2  Stibbons, Ponder       48.0
201   Tennis Court 2     Tracy, Burton        4.5

[202 rows x 3 columns]
```

In [23]:

```
# Another way: Easy to read but with nested subquerries
# query = """
#         SELECT name AS facility, member, usagehour
#         FROM
#             (SELECT facid, (surname||', '||firstname) AS member, usagehour
#              FROM
#                 (SELECT facid, memid, SUM(0.5*slots) AS usagehour
#                  FROM bookings
#                  WHERE memid <> 0
#                  GROUP BY facid, memid
#                 )
#              INNER JOIN members
#              USING(memid)
#              GROUP BY facid, member
#             )
#          INNER JOIN facilities
#          USING(facid)
#          GROUP BY facility, member;
```

```
#         """
# execute_print_sql_query(query)
```

Question 13

Find the facilities usage by month, but not guests.

In [24]:

```
query = """
        SELECT name as facility, strftime('%m', starttime) AS month_Y2012, SUM(0.5*s
        FROM bookings
        INNER JOIN facilities
        USING(facid)
        WHERE memid <> 0
        GROUP BY facility, month_Y2012;
        """
execute_print_sql_query(query)
```

```
Size of panda's dataframe: (27, 3)

           facility month_Y2012  usagehour
0   Badminton Court          07       82.5
1   Badminton Court          08      207.0
2   Badminton Court          09      253.5
3    Massage Room 1          07       83.0
4    Massage Room 1          08      158.0
5    Massage Room 1          09      201.0
6    Massage Room 2          07        4.0
7    Massage Room 2          08        9.0
8    Massage Room 2          09       14.0
9        Pool Table          07       55.0
10       Pool Table          08      151.5
11       Pool Table          09      221.5
12    Snooker Table          07       70.0
13    Snooker Table          08      158.0
14    Snooker Table          09      202.0
15     Squash Court          07       25.0
16     Squash Court          08       92.0
17     Squash Court          09       92.0
18     Table Tennis          07       49.0
19     Table Tennis          08      148.0
20     Table Tennis          09      200.0
21   Tennis Court 1          07      100.5
22   Tennis Court 1          08      169.5
23   Tennis Court 1          09      208.5
24   Tennis Court 2          07       61.5
25   Tennis Court 2          08      172.5
26   Tennis Court 2          09      207.0
```