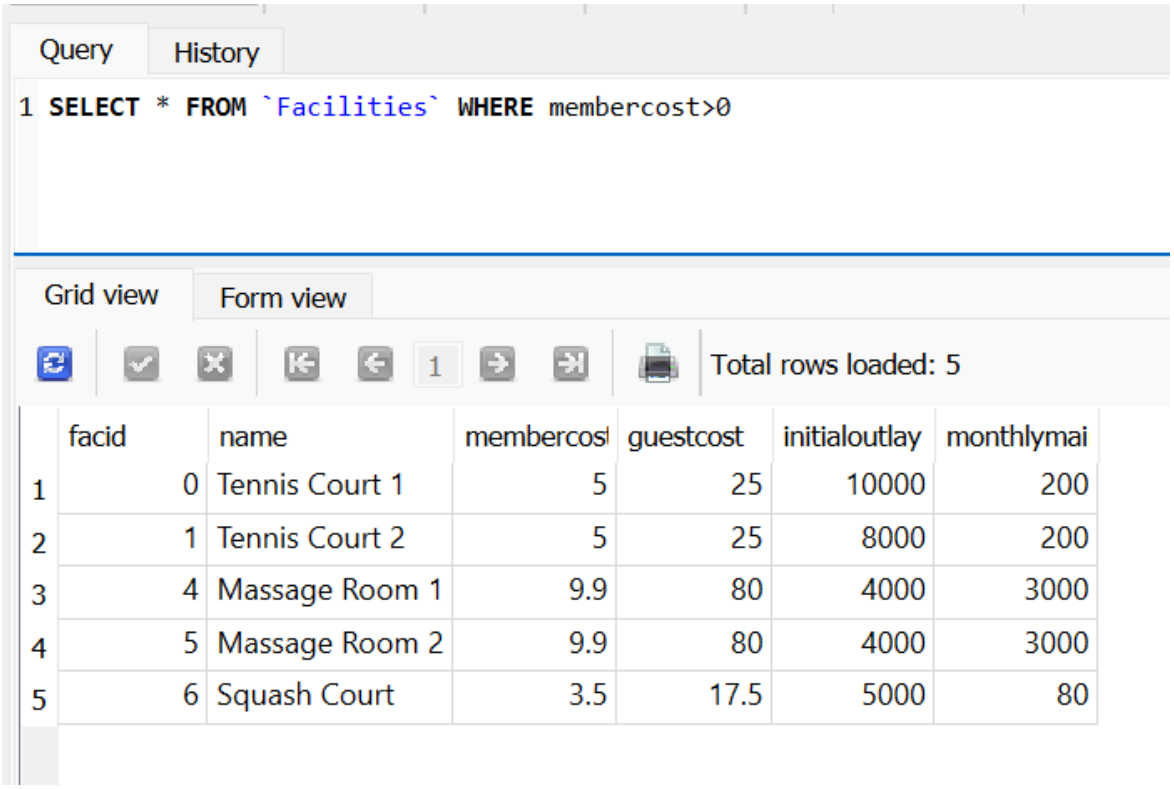
Case Study - Country Club

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.



Query | History

```
1 SELECT * FROM `Facilities` WHERE membercost>0
```

Grid view | Form view

Total rows loaded: 5

	facid	name	membercost	guestcost	initialoutlay	monthlymai
1	0	Tennis Court 1	5	25	10000	200
2	1	Tennis Court 2	5	25	8000	200
3	4	Massage Room 1	9.9	80	4000	3000
4	5	Massage Room 2	9.9	80	4000	3000
5	6	Squash Court	3.5	17.5	5000	80

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

Query History

```
1 SELECT * FROM `Facilities` WHERE membercost=0
```

Grid view Form view

Total rows loaded: 4

	facid	name	membercost	guestcost	initialoutlay	monthlymai
1	2	Badminton Court	0	15.5	4000	50
2	3	Table Tennis	0	5	320	10
3	7	Snooker Table	0	5	450	15
4	8	Pool Table	0	5	400	15

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.

Query | History

```
1  SELECT
2      facid ,
3      name AS `Facility Name`,
4      membercost AS `Mmeber Cost`,
5      monthlymaintenance AS `Monthly Maintainance`
6
7  FROM
8      `Facilities`
9  WHERE
10     membercost< ( monthlymaintenance*20 )/100 and membercost>0
```

Grid view | Form view

Total rows loaded: 5

	facid	Facility Name	Mmeber Cost	Monthly Maintainance
1	0	Tennis Court 1	5	200
2	1	Tennis Court 2	5	200
3	4	Massage Room 1	9.9	3000
4	5	Massage Room 2	9.9	3000
5	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.

Query | History

```
1 SELECT
2     *
3 FROM
4     `Facilities`
5 WHERE
6     facid in (1,5)
```

Grid view | Form view

Total rows loaded: 2

	facid	name	membercost	guestcost	initialoutlay	monthlymai
1	1	Tennis Court 2	5	25	8000	200
2	5	Massage Room 2	9.9	80	4000	3000

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.

Query | History

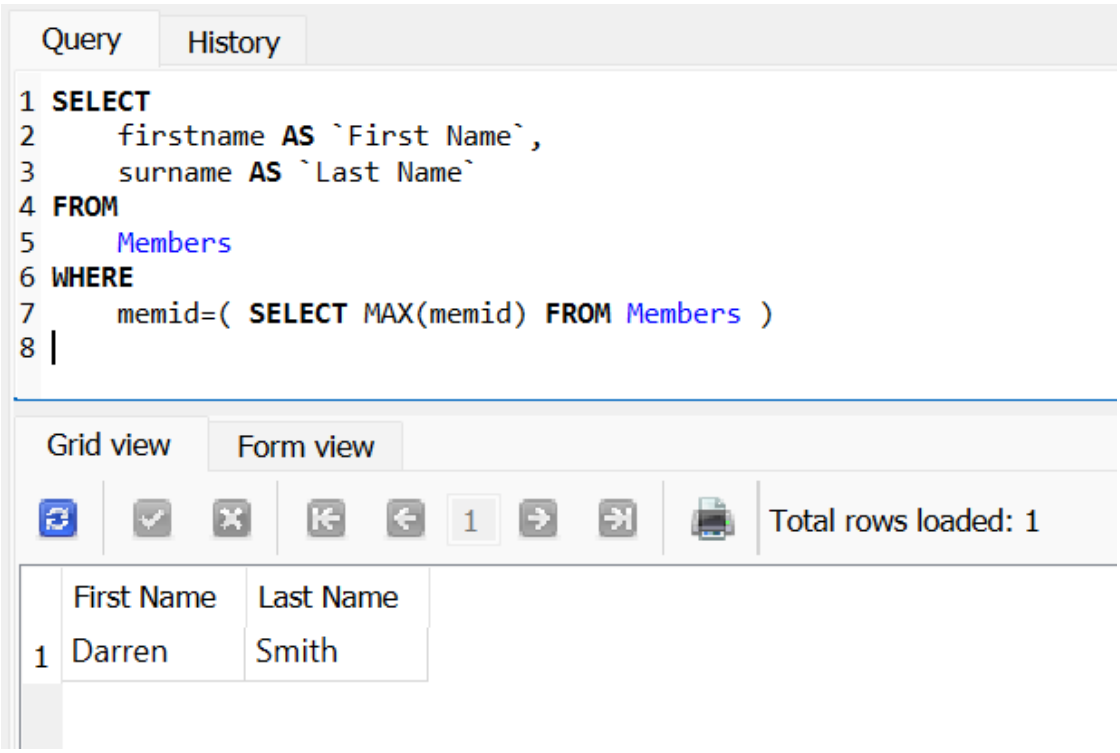
```
1  SELECT
2      name AS `Facility Name`,
3      monthlymaintenance AS `Monthly Maintainance`,
4  CASE
5      WHEN monthlymaintenance > 100 THEN 'EXPENSIVE'
6      WHEN monthlymaintenance = 100 THEN 'MID RANGE'
7      ELSE 'CHEAP'
8  END AS Status
9
10 FROM
11     Facilities
12
```

Grid view | Form view

Total rows loaded: 9

	Facility Name	Monthly Maintainance	Status
1	Tennis Court 1	200	EXPENSIVE
2	Tennis Court 2	200	EXPENSIVE
3	Badminton Court	50	CHEAP
4	Table Tennis	10	CHEAP
5	Massage Room 1	3000	EXPENSIVE
6	Massage Room 2	3000	EXPENSIVE
7	Squash Court	80	CHEAP
8	Snooker Table	15	CHEAP
9	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.



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.

Query | History

```
1  SELECT
2      DISTINCT
3       Members.firstname || ' ' || Members.surname AS `Member`,
4      Facilities.name AS `Facility`
5  FROM
6      Bookings
7  INNER JOIN
8      Members
9  ON
10     Bookings.memid=Members.memid
11
12 INNER JOIN
13     Facilities
14 ON
15     Bookings.facid=Facilities.facid
16 WHERE
17     Bookings.facid in (0,1)
18 ORDER BY Member ASC
```

Grid view | Form view

Total rows loaded: 46

	Member	Facility
1	Anne Baker	Tennis Court 1
2	Anne Baker	Tennis Court 2
3	Burton Tracy	Tennis Court 2
4	Burton Tracy	Tennis Court 1
5	Charles Owen	Tennis Court 1
6	Charles Owen	Tennis Court 2
7	Darren Smith	Tennis Court 2
8	David Farrell	Tennis Court 1
9	David Farrell	Tennis Court 2
10	David Jones	Tennis Court 2
11	David Jones	Tennis Court 1
12	David Pinker	Tennis Court 1
13	Douglas Jones	Tennis Court 1
14	Erica Crumpet	Tennis Court 1
15	Florence Bader	Tennis Court 2
16	Florence Bader	Tennis Court 1
17	GUEST GUEST	Tennis Court 2
18	GUEST GUEST	Tennis Court 1

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.

Query | History

```
SELECT
   Members.firstname || ' ' || Members.surname AS `Member`,
   Facilities.name AS `Facility Name`,
CASE
    when Members.memid = 0
    THEN
        Bookings.slots*Facilities.guestcost
    ELSE
        Bookings.slots*Facilities.membercost
  END AS cost

FROM
    Bookings
INNER JOIN
    Members
ON
    Bookings.memid=Members.memid

INNER JOIN
    Facilities
ON
    Bookings.facid=Facilities.facid


WHERE

Bookings.starttime>='2012-09-14' AND Bookings.starttime<'2012-09-15'
AND
((Members.memid = 0 and Bookings.slots*Facilities.guestcost > 30) OR (Members.memid != 0 and Bookings.slots*Facilities.membercost > 30))

ORDER BY cost DESC
```

Grid view | Form view

Total rows loaded: 12

	Member	Facility Name	cost
1	GUEST GUEST	Massage Room 2	320
2	GUEST GUEST	Massage Room 1	160
3	GUEST GUEST	Massage Room 1	160
4	GUEST GUEST	Massage Room 1	160
5	GUEST GUEST	Tennis Court 2	150
6	GUEST GUEST	Tennis Court 1	75
7	GUEST GUEST	Tennis Court 1	75
8	GUEST GUEST	Tennis Court 2	75
9	GUEST GUEST	Squash Court	70
10	Jemima Farrell	Massage Room 1	39.6
11	GUEST GUEST	Squash Court	35
12	GUEST GUEST	Squash Court	35

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

Query | History

```
SELECT
    sub.Member,
    sub.`Facility Name`,
    CASE
        WHEN sub.memid = 0 THEN sub.slots * sub.guestcost
        ELSE sub.slots * sub.membercost
    END AS cost
FROM
    (
        SELECT
            Members.memid,
            Members.firstname || ' ' || Members.surname AS `Member`,
            Facilities.name AS `Facility Name`,
            Bookings.slots,
            Facilities.membercost,
            Facilities.guestcost
        FROM
            Bookings
        INNER JOIN
            Members
        ON
            Bookings.memid = Members.memid
        INNER JOIN
            Facilities
        ON
            Bookings.facid = Facilities.facid
        WHERE
            Bookings.starttime >= '2012-09-14' AND Bookings.starttime < '2012-09-15'
    ) AS sub
WHERE
    (sub.memid = 0 AND sub.slots * sub.guestcost > 30) OR (sub.memid != 0 AND sub.slots * sub.membercost > 30)
ORDER BY
    cost DESC;
```

Grid view | Form view

Total rows loaded: 12

	Member	Facility Name	cost
1	GUEST GUEST	Massage Room 2	320
2	GUEST GUEST	Massage Room 1	160
3	GUEST GUEST	Massage Room 1	160
4	GUEST GUEST	Massage Room 1	160
5	GUEST GUEST	Tennis Court 2	150
6	GUEST GUEST	Tennis Court 1	75
7	GUEST GUEST	Tennis Court 1	75
8	GUEST GUEST	Tennis Court 2	75
9	GUEST GUEST	Squash Court	70
10	Jemima Farrell	Massage Room 1	39.6
11	GUEST GUEST	Squash Court	35

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!

```
import sqlite3
from sqlite3 import Error

def create_connection(db_file):
    """ create a database connection to the SQLite database
        specified by the db_file
    :param db_file: database file
    :return: Connection object or None
    """
    conn = None
    try:
        conn = sqlite3.connect(db_file)
        print(sqlite3.version)
    except Error as e:
        print(e)

    return conn

def select_all_tasks(conn,query1):
    """
    Query all rows in the tasks table
    :param conn: the Connection object
    :return:
    """
    cur = conn.cursor()


    cur.execute(query1)

    rows = cur.fetchall()

    for row in rows:
        print(row)
```

+ Code + Markdown

```
database = "sqlite_db_pythonsqlite.db"
    # create a database connection
conn = create_connection(database)
```

✓ 0.0s

```
#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_all_tasks(conn,"""SELECT
    Table1.name AS `facility Name`,
    Table1. `Total Revenue` AS `Total Revenue`
FROM
    ( SELECT Facilities.name,
       SUM( CASE WHEN Bookings.memid =0 THEN Bookings.slots * Facilities.guestcost
                 ELSE Bookings.slots * Facilities.membercost END )  AS `Total Revenue`
       FROM
           Facilities
        INNER JOIN
           Bookings
        ON
           Facilities.facid=Bookings.facid
        GROUP BY
           Facilities.name
      ) AS Table1
WHERE
    `Total Revenue`<1000
ORDER BY
    `Total Revenue`""")
```

✓ 0.0s

```
('Table Tennis', 180)
('Snooker Table', 240)
('Pool Table', 270)
```

Q11: Produce a report of members and who recommended them in alphabetical surname,first name order

```
#Q11: Produce a report of members and who recommended them in alphabetical surname,first name order

select_all_tasks(conn,"""SELECT
    member1.memid,
    member1.firstname || ' ' || member1.surname AS Name,
    member2.firstname || ' ' || member2.surname AS `Recommended By`
from
    Members AS member1
INNER JOIN
    Members AS member2
ON
    member1.recommendedby=member2.memid
WHERE member1.memid != 0
ORDER BY Name ASC""")
```

✓ 0.0s

```
(21, 'Anna Mackenzie', 'Darren Smith')
(12, 'Anne Baker', 'Ponder Stibbons')
(10, 'Charles Owen', 'Darren Smith')
(11, 'David Jones', 'Janice Joplette')
(17, 'David Pinker', 'Jemima Farrell')
(26, 'Douglas Jones', 'David Jones')
(36, 'Erica Crumpet', 'Tracy Smith')
(15, 'Florence Bader', 'Ponder Stibbons')
(5, 'Gerald Butters', 'Darren Smith')
(27, 'Henrietta Rumney', 'Matthew Genting')
(29, 'Henry Worthington-Smyth', 'Tracy Smith')
(14, 'Jack Smith', 'Darren Smith')
(4, 'Janice Joplette', 'Darren Smith')
(22, 'Joan Coplin', 'Timothy Baker')
(35, 'John Hunt', 'Millicent Purview')
(20, 'Matthew Genting', 'Gerald Butters')
(30, 'Millicent Purview', 'Tracy Smith')
(7, 'Nancy Dare', 'Janice Joplette')
(9, 'Ponder Stibbons', 'Burton Tracy')
(24, 'Ramnaresh Sarwin', 'Florence Bader')
(8, 'Tim Boothe', 'Tim Rownam')
(16, 'Timothy Baker', 'Jemima Farrell')
```

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

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

select_all_tasks(conn,"""SELECT
    Facilities.name,
    SUM(Bookings.slots) AS `Members Usage`
FROM
    Facilities INNER JOIN Bookings
ON
    Facilities.facid=Bookings.facid

WHERE Bookings.memid !=0
GROUP BY Bookings.facid
""")
```

✓ 0.0s

```
('Tennis Court 1', 957)
('Tennis Court 2', 882)
('Badminton Court', 1086)
('Table Tennis', 794)
('Massage Room 1', 884)
('Massage Room 2', 54)
('Squash Court', 418)
('Snooker Table', 860)
('Pool Table', 856)
```

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

```
#Q13: Find the facilities usage by month, but not guests
select_all_tasks(conn,"""SELECT
    Facilities.name,
    strftime('%m', Bookings.starttime) as Month,
    SUM(Bookings.slots) AS `Month Usage`
FROM
    Facilities INNER JOIN Bookings
ON
    Facilities.facid=Bookings.facid

WHERE Bookings.memid !=0
GROUP BY Bookings.facid,Month""")
```

✓ 0.0s

```
('Tennis Court 1', '07', 201)
('Tennis Court 1', '08', 339)
('Tennis Court 1', '09', 417)
('Tennis Court 2', '07', 123)
('Tennis Court 2', '08', 345)
('Tennis Court 2', '09', 414)
('Badminton Court', '07', 165)
('Badminton Court', '08', 414)
('Badminton Court', '09', 507)
('Table Tennis', '07', 98)
('Table Tennis', '08', 296)
('Table Tennis', '09', 400)
('Massage Room 1', '07', 166)
('Massage Room 1', '08', 316)
('Massage Room 1', '09', 402)
('Massage Room 2', '07', 8)
('Massage Room 2', '08', 18)
('Massage Room 2', '09', 28)
('Squash Court', '07', 50)
('Squash Court', '08', 184)
('Squash Court', '09', 184)
('Snooker Table', '07', 140)
('Snooker Table', '08', 316)
('Snooker Table', '09', 404)
('Pool Table', '07', 110)
('Pool Table', '08', 303)
('Pool Table', '09', 443)
```