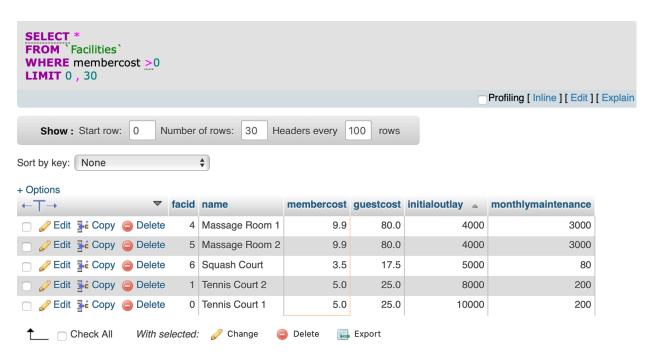
Country Club Case Study SQL

1.



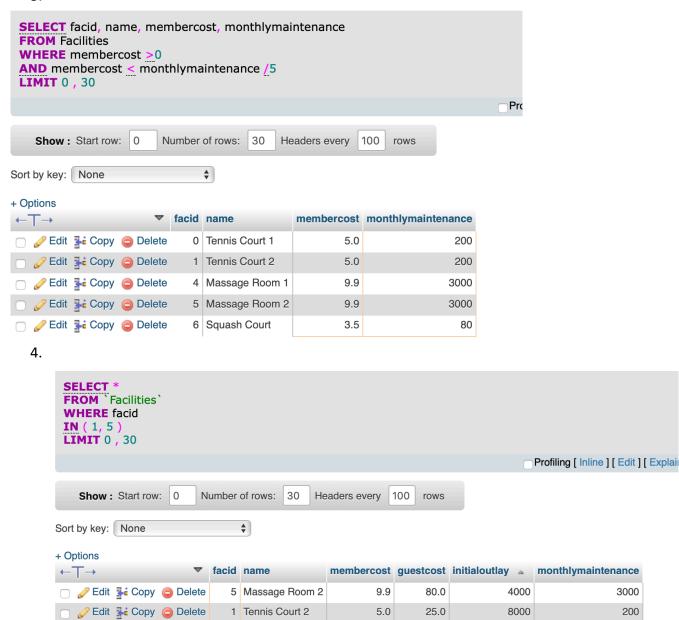
2.

```
SELECT COUNT(*)
FROM `Facilities`
WHERE membercost =0

Profiling [ Inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

+ Options
COUNT(*)
4
```

3.



Delete

Export

5.

Check All

With selected:

Change

SELECT name, facid,

CASE
WHEN (
monthlymaintenance > 100
)
THEN 'expensive'
WHEN (

Sort by key: None \$

+ Options

name	facid	cost	
Tennis Court 1	0	expensive	
Tennis Court 2	1	expensive	
Badminton Court	2	cheap	
Table Tennis	3	cheap	
Massage Room 1	4	expensive	
Massage Room 2	5	expensive	
Squash Court	6	cheap	
Snooker Table	7	cheap	
Pool Table	8	cheap	

6.



FROM Members
INNER JOIN Bookings ON Members.memid = Bookings.memid
INNER JOIN Facilities ON Bookings.facid = Facilities.facid
WHERE Bookings.facid
IN (0, 1)
ORDER BY Members.firstname, Members.surname
LIMIT 0, 30

1 \$ Show all > >> Show: Start row: 30 Number of rows: 30 Headers ev

+ Options

firstname	surname	name
Anne	Baker	Tennis Court 1
Anne	Baker	Tennis Court 2
Burton	Tracy	Tennis Court 2
Burton	Tracy	Tennis Court 1
Charles	Owen	Tennis Court 1
Charles	Owen	Tennis Court 2
Darren	Smith	Tennis Court 2
David	Farrell	Tennis Court 1
David	Farrell	Tennis Court 2
David	Jones	Tennis Court 1
David	Jones	Tennis Court 2

David	Pinker	Tennis Court 1
Douglas	Jones	Tennis Court 1
Erica	Crumpet	Tennis Court 1
Florence	Bader	Tennis Court 2
Florence	Bader	Tennis Court 1
Gerald	Butters	Tennis Court 2
Gerald	Butters	Tennis Court 1
GUEST	GUEST	Tennis Court 1
GUEST	GUEST	Tennis Court 2
Henrietta	Rumney	Tennis Court 2
Jack	Smith	Tennis Court 1
Jack	Smith	Tennis Court 2
Janice	Joplette	Tennis Court 1
Janice	Joplette	Tennis Court 2
Jemima	Farrell	Tennis Court 1
Jemima	Farrell	Tennis Court 2
Joan	Coplin	Tennis Court 1
John	Hunt	Tennis Court 2
John	Hunt	Tennis Court 1

```
8. SELECT CONCAT( Members firstname, '
   ', Members.surname ) , Facilities.name,
   CASE WHEN Members memid !=0
   THEN Facilities.membercost * Bookings.slots
   ELSE Facilities guestcost * Bookings slots
   END AS total_cost
   FROM Bookings
   LEFT JOIN Members ON Bookings.memid = Members.memid
   LEFT JOIN Facilities ON Bookings.facid = Facilities.facid
   WHERE Bookings.starttime LIKE '2012-09-14%'
   AND (
Members.memid =0
AND Facilities.membercost * Bookings.slots >30
   OR (
   Members.memid =0
   AND Facilities guestcost * Bookings slots
   ORDER BY total_cost DESC
   LIMIT 0 , 30
```

+ Options

CONCAT(Members.firstname, ' ', Members.surname)	name	total_cost 🔻
GUEST GUEST	Massage Room 2	320.0
GUEST GUEST	Massage Room 1	160.0
GUEST GUEST	Massage Room 1	160.0
GUEST GUEST	Massage Room 1	160.0
GUEST GUEST	Tennis Court 2	150.0
GUEST GUEST	Tennis Court 1	75.0
GUEST GUEST	Tennis Court 2	75.0
GUEST GUEST	Tennis Court 1	75.0
GUEST GUEST	Squash Court	70.0
GUEST GUEST	Squash Court	35.0
GUEST GUEST	Squash Court	35.0

9.

SELECT * FROM (

SELECT Facilities.name AS facility, CONCAT(Members.firstname, ', Members.surname) AS name,

CASE WHEN Bookings.memid =0

THEN Facilities.guestcost * Bookings.slots

ELSE Facilities.membercost * Bookings.slots

END AS total_cost

FROM Bookings

LEFT JOIN Facilities ON Bookings.facid = Facilities.facid

AND Bookings.starttime LIKE '2012-09-14%'

LEFT JOIN Members ON Bookings.memid = Members.memid

)new_table

WHERE new_table.total_cost >30

ORDER BY new_table.total_cost DESC

+ Uptions

facility	name	total_cost 💌
Massage Room 2	GUEST GUEST	320.0
Massage Room 1	GUEST GUEST	160.0
Massage Room 1	GUEST GUEST	160.0
Massage Room 1	GUEST GUEST	160.0
Tennis Court 2	GUEST GUEST	150.0
Tennis Court 2	GUEST GUEST	75.0
Tennis Court 1	GUEST GUEST	75.0
Tennis Court 1	GUEST GUEST	75.0
Squash Court	GUEST GUEST	70.0
Massage Room 1	Jemima Farrell	39.6
Squash Court	GUEST GUEST	35.0
Squash Court	GUEST GUEST	35.0

import sqlite3from 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):
  Query all rows in the tasks table
  :param conn: the Connection object
  :return:
  *****
  cur = conn.cursor()
```

```
query1 = """
    SELECT *
    FROM FACILITIES
  cur.execute(query1)
  rows = cur.fetchall()
  for row in rows:
    print(row)
def main():
  database = "sqlite\db\pythonsqlite.db"
  # create a database connection
  conn = create connection(database)
  with conn:
    print("2. Query all tasks")
    select all tasks(conn)
if __name__ == '__main__':
  main()
 %%sql
select merged.facility_name, merged.total_revenue
from (select fac.name as facility name,
      sum(case when mem.memid = 0 then fac.questcost*slots else fac.member
cost end) as total revenue
from public."springboard/Bookings" as book
join public. "springboard/Facilities" as fac on fac.facid = book.facid
join public."springboard/Members" as mem on mem.memid = book.memid group b
y facility name) merged
where total revenue < 1000
order by total revenue desc;
```

facility_name	total_revenue	
Pool Table	270.0	
Snooker Table	240.0	
Table Tennis	180.0	