Evan Nussbaum

Country Club Case Study SQL

1.

```
SELECT *
FROM `Facilities`
WHERE membercost >0
LIMIT 0 , 30
```

Profiling [Inline] [Edit] [Explain

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

Sort by key: None

+ Options

	facid	name	membercost	guestcost	initialoutlay	monthlymaintenance
Edit Copy Delete	4	Massage Room 1	9.9	80.0	4000	3000
Edit Copy Delete	5	Massage Room 2	9.9	80.0	4000	3000
Edit Copy Delete	6	Squash Court	3.5	17.5	5000	80
Edit Copy Delete	1	Tennis Court 2	5.0	25.0	8000	200
Edit Copy Delete	0	Tennis Court 1	5.0	25.0	10000	200

Check All *With selected:* Change Delete Export

2.

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

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

+ Options

COUNT(*)
4

3.

```
SELECT facid, name, membercost, monthlymaintenance
FROM Facilities
WHERE membercost >0
AND membercost < monthlymaintenance /5
LIMIT 0 , 30
```

Pro

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

Sort by key: None

+ Options

	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

4.

```
SELECT *
FROM `Facilities`
WHERE facid
IN ( 1, 5 )
LIMIT 0 , 30
```

Profiling [Inline] [Edit] [Explai

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

Sort by key: None

+ Options

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

Check All *With selected:* Change Delete Export

5.

```
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.

```
SELECT firstname, surname, MAX( joindate ) AS latest_join
FROM Members
```

Profiling [

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

+ Options

firstname	surname	latest_join
GUEST	GUEST	2012-09-26 18:08:45

7.

```
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
```

+ Options

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

10.

```
# 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):
    """
    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.guestcost*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

11.