SQL case study

March 26, 2021

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;
```

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

[ ]:

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

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
AND membercost < monthlymaintenance * 0.2;
```

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.

[ ]:

```
FROM Facilities
WHERE facid
IN ( 1, 5 );
```

Q5: Produce a list of facilities, with each labelled as 'cheap' or 'expensive', depending on if their monthly maintenance cost is more than 100 dollars. Return the name and monthly maintenance of the facilities in question.

[ ]:

```
SELECT name,
CASE
WHEN monthlymaintenance >100
THEN 'expensive'
ELSE 'cheap'
END AS monthlymaintenance
```

```
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.

[ ]:

```
SELECT surname, firstname
FROM Members
WHERE joindate = (

SELECT MAX( joindate )
FROM Members
);
```

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 (
CONCAT( m.firstName, ' ', m.surname )
) AS membername, f.name
FROM Members AS m
LEFT JOIN Bookings AS b ON m.memid = b.memid
LEFT JOIN Facilities AS f ON f.facid = b.facid
WHERE f.name
IN (
'Tennis Court 1', 'Tennis Court 2'
)
ORDER BY membername;
```

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
  DISTINCT (
    CONCAT(m.firstName, ' ', m.surname)
  ) AS membername,
  f.name,
  CASE WHEN (
    b.memid = 0
    AND (b.slots * f.guestcost > 30)
  ) THEN (b.slots * f.guestcost) ELSE b.slots * f.membercost END AS cost
FROM
  Bookings AS b
  LEFT JOIN Members AS m ON m.memid = b.memid
  LEFT JOIN Facilities AS f ON f.facid = b.facid
```

```
WHERE
  (
    b.starttime >= '2012-09-14 00:00:00'
    AND b.starttime <= '2012-09-14 23:59:59'
  )
  AND CASE
    WHEN b.memid = 0 THEN (f.guestcost * b.slots)
    ELSE (f.membercost * b.slots) END > 30
ORDER BY cost DESC;
```

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

[ ]:

```
SELECT
  sub3.membername,
  sub3.facilityname,
  sub3.Cost
FROM
  (
    SELECT
      sub2.membername AS membername,
      f.name AS facilityname,
      CASE WHEN sub2.Type = 'Member'
      AND (
        sub2.slotNumber * f.membercost > 30
      ) THEN sub2.slotNumber * f.membercost WHEN sub2.Type = 'Guest'
      AND (sub2.slotNumber * f.guestcost > 30) THEN sub2.slotNumber * f.
↪guestcost END AS Cost
    FROM
      (
        SELECT
          DISTINCT (
            CONCAT(m.firstName, ' ', m.surname)
          ) AS membername,
          sub1.memid AS memberId,
          sub1.facid AS facilityId,
          sub1.slots AS slotNumber,
          sub1.UserType AS Type
        FROM
          (
            SELECT
              memId,
              facid,
              slots,
              CASE WHEN memid = 0 THEN 'Guest' WHEN memid <> 0 THEN 'Member'␣
↪END AS UserType
            FROM
              Bookings
            WHERE
              starttime >= '2012-09-14 00:00:00'
              AND starttime <= '2012-09-14 23:59:59'
          ) AS sub1
          LEFT JOIN Members m ON m.memid = sub1.memid
      ) AS sub2
      LEFT JOIN Facilities f ON f.facid = sub2.facilityId
  ) AS sub3
WHERE
  sub3.Cost > 30
ORDER BY
  sub3.Cost DESC;
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: