SQL Case Study

Created	@November 7, 2023 4:15 PM
Class	SQL
Type	Project
Reviewed	☐

QUESTIONS:

Q1: Some of the facilities charge a fee to members, 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 name
FROM facilities
WHERE member cost = 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 < (monthlymaintenance * .20);
```

Q4: Write a SQL query to retrieve the details of facilities with ID 1 and 5.
Try writing the query without using the OR operator.

```
SELECT *
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. Return the name and monthly maintenance of the facilities in question.

```
SELECT name, monthlymaintenance,
  CASE WHEN monthlymaintenance < 100 THEN 'Cheap'
    WHEN monthlymaintenance > 100 THEN 'Expensive'
    ELSE 'Tie' END AS Cost
FROM Facilities;
```

Q6: You'd like to get the first and last name of the last member(s) who signed up.

```
SELECT firstname, surname, joindate
FROM Members
ORDER BY joindate DESC
LIMIT 1;
```

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(firstname, ' ', surname) AS Member_Name, f.name AS Facility_Name
  FROM Members AS m
INNER JOIN Bookings AS b
  ON m.memid = b.memid
INNER JOIN Facilities AS f
  ON f.facid = b.facid
WHERE f.name = 'Tennis Court 1' OR f.name = 'Tennis Court 2'
ORDER BY Member_Name;
```

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 CONCAT(m.firstname,' ', m.surname) AS Mem_Name, f.name AS Facil_Name, IF(m.memid > 0, b.slots * f.membercost, b.slots * f.guestc
  FROM Members AS m
INNER JOIN Bookings AS b
  ON m.memid = b.memid
INNER JOIN Facilities AS f
  ON b.facid = f.facid
WHERE IF(m.memid > 0, b.slots * f.membercost, b.slots * f.guestcost) > 30 AND starttime BETWEEN '2012-09-14 00:00:00' AND '2012-09-14 S
ORDER BY IF(m.memid > 0, b.slots * f.membercost, b.slots * f.guestcost) DESC;

OR

SELECT f.name AS facility_name,
       CONCAT(m.firstname, ' ', m.surname) AS member_name,
       CASE
          WHEN b.memid = 0 THEN b.slots * f.guestcost
          ELSE b.slots * f.membercost
       END AS cost
  FROM Bookings AS b
  INNER JOIN Facilities AS f
```

```
    ON b.facid = f.facid
  INNER JOIN Members AS m
    ON b.memid = m.memid
  WHERE b.starttime BETWEEN '2012-09-14 00:00:00' AND '2012-09-14 99:99:99'
  AND (b.memid = 0 AND b.slots * f.guestcost > 30 OR b.memid != 0 AND b.slots * f.membercost > 30)
  ORDER BY cost DESC;
```

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

```
SELECT CONCAT(m.firstname,' ', m.surname) AS member_name,
    CASE
      WHEN b.memid = 0 THEN b.slots * f.guestcost
      ELSE b.slots * f.membercost
      END AS cost,
      f.name AS facility_name
FROM Bookings AS b
INNER JOIN Facilities AS f ON b.facid = f.facid
INNER JOIN Members AS m ON b.memid = m.memid
WHERE date(b.starttime) = '2012-09-14'
AND (
(b.memid = 0 AND b.slots * f.guestcost > 30)
OR (b.memid != 0 AND b.slots * f.membercost > 30)
)
ORDER BY cost DESC;
```

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 f.name,
  SUM(CASE WHEN b.memid = 0 THEN b.slots * f.guestcost ELSE b.slots * f.membercost END) AS total_revenue
FROM Facilities AS f
INNER JOIN Bookings AS b ON f.facid = b.facid
GROUP BY f.facid
HAVING total_revenue < 1000
ORDER BY total_revenue;
```

Q11: Produce a report of members and who recommended them in alphabetic surname,firstname order.

```
SELECT CONCAT(m1.surname,' ', m1.firstname) AS member_name, CONCAT(m2.surname,' ', m2.firstname) AS recommended_by
FROM Members AS m1
LEFT JOIN Members AS m2 ON m1.memid = m2.recommendedby
WHERE m1.memid <> 0
ORDER BY m1.surname, m1.firstname;
```

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

```
SELECT f.name, COUNT(b.bookid) AS usage_totals
FROM Facilities AS f
INNER JOIN Bookings AS b
ON f.facid = b.facid
WHERE b.memid <> 0
GROUP BY f.name;
```

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

```
SELECT f.name, MONTH(b.starttime) AS month, COUNT(*) AS usage_count
FROM Bookings AS b
JOIN Facilities AS f
ON b.facid = f.facid
JOIN Members AS m
ON b.memid = m.memid
GROUP BY f.facid, month
ORDER BY f.facid, month;
```