Lecture 22:

- Subqueries (cont)

Announcements:
- HW-6 due
- Project Step 4 due
- HW-7 out soon
- PS-4 out (due Tues, Nov 29)

Running Example

Schema:

```
Customer(c_num, name, addr, c_rating, c_amount, c_bal, sp_num)
Salesperson(sp_num, name, address, office)
```

with FK: customer.sp_num → salesperson.sp_num

Example Customer instance

<table>
<thead>
<tr>
<th>c_num</th>
<th>name</th>
<th>address</th>
<th>c_rating</th>
<th>c_amount</th>
<th>c_balance</th>
<th>sp_num</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
<td>xxx</td>
<td>5</td>
<td>1000</td>
<td>1000</td>
<td>101</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>yyy</td>
<td>7</td>
<td>5000</td>
<td>4000</td>
<td>101</td>
</tr>
<tr>
<td>3</td>
<td>Chuck</td>
<td>zzz</td>
<td>10</td>
<td>10000</td>
<td>1000</td>
<td>102</td>
</tr>
</tbody>
</table>
A subquery is a “nested” query

- Primarily used within WHERE and FROM clauses
- Can also be used in SELECT and HAVING clauses

```
SELECT c1.c_num, c1.name  
FROM customer c1  
WHERE c1.c_rating = (SELECT MAX(c2.c_rating)  
                      FROM customer c2);
```

- the subquery is the inner query
- the rest is the outer query

WHERE Subquery Comparators

(1) Single-valued subquery comparisons

For example:

```
SELECT c1.cust_num, c1.name  
FROM customer c1  
WHERE c1.c_rating = (SELECT MAX(c2.c_rating)  
                      FROM customer c2);
```

- comparator can be any of six standard operators: <, <=, =, !=, >=, >
- But for these, inner query must return a single value!
WHERE Subquery Comparators

(2) Single-to-multi-valued subquery comparisons

ANY comparisons

SELECT s.sp_num, s.name
FROM salesperson s
WHERE s.name = ANY (SELECT c.name FROM customer c);

• subquery can return more than one answer
• the expression (here =) must be true for at least one subquery answer
• “SOME” is equivalent to ANY

Check In: What is returned? Can it be written without a subquery?
• salespeople that have the same name as a customer
• this query can be rewritten without using subqueries

WHERE Subquery Comparators

ALL comparisons

SELECT s.name
FROM salesperson s
WHERE s.sp_num = ALL (SELECT c.sp_num
FROM customer c
WHERE c.c_rating = 3);

• expression (here =) must be true for every subquery answer

Check In: What is returned? Can it be written without a subquery?
• salespeople that have all of the customers with a credit rating of 3
• cannot be rewritten without subqueries (using constructs we've seen so far)
WHERE Subquery Comparators

Check In: What does this query do?

```
SELECT c_num
FROM customer
WHERE c_rating >= ALL (SELECT c_rating FROM customer)
```

• note: same as first MAX query

Check In: What does this query do?

```
SELECT c_num
FROM customer
WHERE c_rating <= ANY (SELECT c_rating FROM customer)
```

• note: this query is always true if there is at least one customer

3. Empty set (satisfiability) subquery checks

EXISTS/NOT EXISTS checks

```
SELECT c.name
FROM customer c
WHERE EXISTS (SELECT *
    FROM salesperson s
    WHERE s.sp_num = c.sp_num AND
    s.name = c.name);
```

• EXISTS is true if subquery returns at least one row

• NOT EXISTS is true if subquery is empty

Check In: What is returned? Can it be written w/out subqueries?

S. Bowers CPSC 321, Fall 2022
WHERE Subquery Comparators

4. Additional single-to-multi valued comparisons

IN/NOT IN checks

SELECT c.name
FROM customer c
WHERE c.name NOT IN (SELECT s.name
FROM salesperson s);

• IN is true if value(s) contained in subquery result
• NOT IN is true if value(s) not contained in subquery result

Note: This query cannot be rewritten using joins only
• but if instead we used IN, it could be

More Examples of Subqueries

Using subqueries in HAVING clauses

SELECT s.sp_num, s.name, AVG(c_balance)
FROM salesperson s JOIN customer c USING (sp_num)
WHERE c.c_rating > 5
GROUP BY s.sp_num
HAVING AVG(c.c_balance) >= ALL (SELECT AVG(c_balance)
FROM customer
WHERE c_rating > 5
GROUP BY sp_num);

Check In: What does this query do?
• Finds the salespeople whose average customer account balance for those customers with a rating over 5 is greater than the average balance of all customers with a credit rating over 5.
More Examples of Subqueries

Check In: Find the salespeople with the most customers. Return the salesperson number, name, and the number of their corresponding customers.

```sql
SELECT s.sp_num, s.name, COUNT(*)
FROM salesperson s JOIN customer c USING (sp_num)
GROUP BY s.sp_num
HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM customer GROUP BY sp_num);
```

Check In: Find customers with a higher than avg rating and lower than avg balance. Return the customer number and name

```sql
SELECT c_num, name
FROM customer
WHERE c_rating > (SELECT AVG(c_rating) FROM customer) AND c_balance < (SELECT AVG(c_balance) FROM customer);
```

Note on Using Subqueries (Wrap Up)

When possible, rewrite (unnest) your subqueries!

- Don’t get carried away with unnecessary subqueries
- Subqueries (e.g., in MySQL) may not be optimized
- For example, see: mariadb.com/kb/en/subquery-optimizations-map/
- Rewriting them into unnested queries may help optimization

Note that each query can often be written multiple ways