

CSE 180 - Lab 2

Subqueries

Dev

October 23 2023, Section 4



Logistics



- Lab 2 Due: **Tuesday, October 31 11:59PM PT**



Subqueries

- Subqueries are nested queries that can be placed in
 - SELECT
 - FROM
 - WHERE (most common)

Subqueries, Example

- Find all the beers that cost less than average, list their price and the bar

```
SELECT beer, bar, price
FROM sells
WHERE price < ( SELECT AVG(price)
                 FROM sells);
```

Rules for Subqueries

- Needs to be enclosed in parentheses (Subquery)
- Must be to the right of comparison operator
- Cannot use modifiers like ORDER BY to manipulate its results
- Careful when subqueries return NULLs

Comparison Operators with Subqueries

- With a comparison operator, the subquery SELECT can only return one value
- If WHERE outside the subquery contains a column, it must be join-able with the subquery
- So how do we use comparison operators with subqueries that return multiple values?

IN and NOT IN

- Subquery returns a set of 0 or more values, and the outer query checks for set membership
- IN can almost always be converted to a join, NOT IN cannot
- Eg: Find beer name, manufacturer of the beers sold by The Red Room

```
SELECT beer, manf
FROM beers
WHERE beer IN ( SELECT beer
                FROM sells
                WHERE bar='The Red Room');
```

ANY, SOME, ALL

- WHERE Value > ALL (subquery) means value is greater than all of the results returned by subquery
 - > ALL (3, 4, 5) means >5
- WHERE Value > ANY (subquery) means value is greater than at least one of the values
 - > ANY (3, 4, 5) means >3
- =ANY is equivalent to IN
- <>ALL is equivalent to NOT IN

EXISTS and NOT EXISTS

- Performs an Existence test
- There is nothing to the left of EXISTS or NOT EXISTS after WHERE
- NULL will evaluate to TRUE in EXISTS
- WHERE EXISTS (subquery)

```
SELECT beer, manf
FROM beers b
WHERE EXISTS ( SELECT *
                FROM sells s
                WHERE b.beer = s.beer
                AND bar='The Red Room');
```

Lab 2 Tips

- Break down wordy problem statements into SELECT and clauses
- Joins can be broken into subqueries
- Joins are often (but not always) done between 2 tables based on Primary Key matching Foreign Key.
- Use load data to test the queries
- EXISTS, IN can be made interchangeable, multiple solutions exist
- Getting the correct output does not guarantee that your query is correct
- Get help early!