

Nested Subqueries

- SQL provides a mechanism for the nesting of subqueries.
- A **subquery** is a **select-from-where** expression that is nested within another query.
- A common use of subqueries is to perform tests for set membership, set comparisons, and set cardinality.

Nested queries

- ▶ At the end of class last time we attempted to create a query that returns (from the university dataset):
 - The building that houses the departments with the largest average budget.
- ▶ We came up with:
 - ▶

```
select building, avg_budget
from
  (select building, avg(budget) as avg_budget from department group by building)
  as avg_budgets
where avg_budget = (select max(avg_budget) from
  (select building, avg(budget) as avg_budget from department group by building)
  as avg_budgets);
```
- ▶ Alternatively, we can use the with clause
 - ▶

```
with avg_budgets(building, avg) as
  (select building, avg(budget) from department group by building)
select building, avg
from avg_budgets
where avg = (select max(avg) from avg_budgets)
```

Multi-comparisons

- ▶ So far we've seen simple predicates in WHERE clause
 - `SELECT * from table where val = 4;`
 - `SELECT * from table where val > 4;`
- ▶ What if you want to compare a value with multiple other values?
- ▶ `select * from table where val = "at least one of" (5, 6, 7);`
 - Above example can be done with OR
 - `where val = 5 or val = 6 or val = 7`
 - But what about:
 - `select * from table`
`where val = "at least one of" (select some_att from other_table);`
 - Solution: Use the IN keyword
 - `select * from table where val IN (5, 6, 7);`
 - `select * from table where val IN (select some_att from other_table);`

Popular WHERE-clause comparisons

- ▶ IN: equals at least one of
 - ▶ NOT IN: != all of
 - ▶ EXISTS r: returns true if r is not empty
 - ▶ NOT EXISTS r: returns true if r is empty
 - ▶ SOME, ALL --- see next two slides
-
- ▶ See examples of these comparison operations in the companion SQL examples document from today

Some Clause

- $F \text{ <comp> some } r \Leftrightarrow \exists t \in r \text{ such that } (F \text{ <comp> } t)$

Where <comp> can be: <, >=, >, =, !=, <>

(5 < **some**

0
5
6

) = true

(read: 5 < some tuple in the relation)

(5 < **some**

0
5

) = false

(5 = **some**

0
5

) = true

(5 != **some**

0
5

) = true (since $0 \neq 5$)

(= **some**) \equiv in

However, (!= **some**) is not the same as **not in**

Definition of all Clause

- $F \text{ <comp> } \mathbf{all} \ r \Leftrightarrow \forall t \in r \ (F \text{ <comp> } t)$

(5 < **all**

0
5
6

) = false

(5 < **all**

6
10

) = true

(5 = **all**

4
5

) = false

(5 != **all**

4
6

) = true (since $5 \neq 4$ and $5 \neq 6$)