Lecture 06

Subqueries

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.
- The nesting can be done in the following SQL query

```
select A1, A2, ..., An
from r1, r2, ..., rm
where P
```

- as follows:
  - Ai can be replaced be a subquery that generates a single value.
  - ri can be replaced by any valid subquery
  - P can be replaced with an expression of the form:
  - B <operation> (subquery)
  - Where B is an attribute and <operation> to be defined later.

Subqueries in the Where Clause

Subqueries in the Where Clause

- A common use of subqueries is to perform tests:
  - For set membership
  - For set comparisons
  - For set cardinality.

Set Membership

- Find courses offered in Fall 2009 and in Spring 2010

```
select distinct course_id from section where semester =
'Fall' and year= 2009 and  course_id in (select course_id
from section where semester = 'Spring' and year= 2010);
```

- Find courses offered in Fall 2009 but not in Spring 2010

```
select distinct course_id from section where semester =
'Fall' and year= 2009 and course_id  not in (select
course_id from section where semester = 'Spring' and year=
2010);
```

Set Comparison – “some” Clause

- Find names of instructors with salary greater than that of some (at least one) instructor in the Biology department.

```
select distinct T.name
from instructor as T, instructor as S
where T.salary > S.salary and S.dept_name = 'Biology’;
```

- Same query using > **some** clause

```
select distinct name
from instructor
where salary > some (select salary from instructor  where
dept_name = 'Biology');
```

Definition of "some" Clause

- F <comp> some r $\Leftrightarrow \exists$ t $\in$ r such that (F <comp> t) Where <comp> can be: $<, \leq, >, =, \neq$

$$(5 < \textbf{some} \begin{array}{|c|}\hline 0 \\ \hline 5 \\ \hline 6 \\ \hline \end{array}) = \text{true}$$

(read: 5 < some tuple in the relation)

$$(5 < \textbf{some} \begin{array}{|c|}\hline 0 \\ \hline 5 \\ \hline \end{array}) = \text{false}$$

$$(5 = \textbf{some} \begin{array}{|c|}\hline 0 \\ \hline 5 \\ \hline \end{array}) = \text{true}$$

$$(5 \neq \textbf{some} \begin{array}{|c|}\hline 0 \\ \hline 5 \\ \hline \end{array}) = \text{true (since } 0 \neq 5)$$

(= **some**) $\equiv$ **in**

However, ($\neq$ **some**) $\not\equiv$ **not in**

Set Comparison – “all” Clause

- Find the names of all instructors whose salary is greater than the salary of all instructors in the Biology department.

    -
      ```
      select name from instructor where salary > all (select
      salary from instructor where dept_name = 'Biology');
      ```

Definition of “all” Clause

- F <comp> all r $\Leftrightarrow$ $\forall$ t $\in$ r (F <comp> t)

$$(5 < \textbf{all} \begin{array}{|c|}\hline 0 \\ \hline 5 \\ \hline 6 \\ \hline \end{array}) = \text{false}$$

$$(5 < \textbf{all} \begin{array}{|c|}\hline 6 \\ \hline 10 \\ \hline \end{array}) = \text{true}$$

$$(5 = \textbf{all} \begin{array}{|c|}\hline 4 \\ \hline 5 \\ \hline \end{array}) = \text{false}$$

$$(5 \neq \textbf{all} \begin{array}{|c|}\hline 4 \\ \hline 6 \\ \hline \end{array}) = \text{true (since } 5 \neq 4 \text{ and } 5 \neq 6)$$

($\neq$ **all**) $\equiv$ **not in**

However, (= **all**) $\not\equiv$ **in**

Test for Empty Relations

- The exists construct returns the value true if the argument subquery is nonempty.
- exists r $\Leftrightarrow$ r $\neq$ Ø
- not exists r $\Leftrightarrow$ r = Ø

Use of “exists” Clause

- Yet another way of specifying the query “Find all courses taught in both the Fall 2009 semester and in the Spring 2010 semester”

```
select course_id from section as S where semester = 'Fall'
and year = 2009 and exists (select *  from section as
T  where semester = 'Spring' and year= 2010 );
```

- **Correlation name** – variable S in the outer query
- **Correlated subquery** – the inner query

Use of “not exists” Clause

- Find all students who have taken all courses offered in the Elec. Eng. department.

```
select distinct S.ID, S.name
from student as S
where not exists ( (select course_id
                    from course
                    where dept_name = Elec. Eng.’)
                  except
                    (select T.course_id
                      from takes as T
                      where S.ID = T.ID));
```

- First nested query lists all courses offered in Biology
- Second nested query lists all courses a particular student took

□ Note that $X - Y = \emptyset \;\Leftrightarrow\; X \subseteq Y$

□ *Note:* Cannot write this query using = **all** and its variants

Test for Absence of Duplicate Tuples

- The unique construct tests whether a subquery has any duplicate tuples in its result.
- The unique construct evaluates to “true” if a given subquery contains no duplicates .
- Find all courses that were offered at most once in 2009

```
select T.course_id
from course as T
where unique (select R.course_id from section as R where
T.course_id= R.course_id  and R.year = 2009);
```

Subqueries in the Form Clause

Subqueries in the From Clause

- SQL allows a subquery expression to be used in the from clause
- Find the average instructors' salaries of those departments where the average salary is greater than $42,000."

```
select dept_name, avg_salary
from
    (select dept_name, avg (salary) as avg_salary
     from instructor group by dept_name)
     as dept_avg
where avg_salary > 42000;
```

  - Note that we do not need to use the having clause

Subqueries in the From Clause

- SQL allows a subquery expression to be used in the from clause
- Find the average instructors' salaries of those departments where the average salary is greater than $42,000."
- Another way to write above query

```
select dept_name, avg_salary
from
(select dept_name, avg (salary) from instructor  group by
dept_name) as
dept_avg (dept_name, avg_salary)
where avg_salary > 42000;
```