Subqueries

Lecture 11
Subsections 5.1.1 - 5.1.5

Robb T. Koether

Hampden-Sydney College

Fri, Feb 7, 2014

Outline

Subqueries

- In MySQL queries may be nested.
- For example, suppose we have a table `new_courses` that contains tuples of new courses to be added to `courses`.
- The following query will insert the new courses into the `courses` table.

Nested Queries

```
INSERT INTO employees (SELECT * FROM new_hires);
```

Subqueries

- Similarly, if we have a table `old_courses` that contains tuples of old courses to be deleted from `courses`, then the following query will delete the old courses from the `courses` table.

Nested Queries

```
DELETE FROM employees
WHERE ssn IN
(SELECT ssn FROM fired);
```

Outline

The IN Operator

The IN Operator

attribute IN *relation*

or

(*attribute_list*) IN *relation*

- The IN operator may be used in the WHERE clause to test whether a value or set of values is in a relation.
- The expression is true if the attribute or attribute list *matches* any of the tuples in the relation.
- But it must match the *entire* tuple *exactly*.

The IN Operator

The IN Operator

```
SELECT fname, lname
FROM employee
WHERE ssn IN
    (SELECT ssn
     FROM dependants);
```

- Find all employees who have dependants.

The IN Operator

The IN Operator

```
SELECT fname, lname
FROM employee
WHERE ssn IN
    (SELECT ssn
     FROM dependants);
```

- Find all employees who have dependants.
- What is another way to write the query without using IN?

The IN Operator

The IN Operator

```
SELECT fname, lname
FROM employee
WHERE ssn IN
    (SELECT mgr_ssn
     FROM departments);
```

- Find all employees who are managers.

The IN Operator

The IN Operator

```
SELECT fname, lname
FROM employee
WHERE ssn IN
    (SELECT mgr_ssn
     FROM departments);
```

- Find all employees who are managers.
- Write this another way without using IN.

The IN Operator

The IN Operator

```
SELECT proj_name, sex
FROM projects AS p NATURAL JOIN employees
WHERE ssn IN
    (SELECT ssn
     FROM works AS w
     WHERE w.proj = p.proj)
```

- Create a table of project names and sexes of all employees working on that project.

The IN Operator

- Find all projects, if any, on which at least one male is working.

The IN Operator

- Find all projects, if any, on which at least one male is working.
- Find all projects, if any, on which no male is working.

The IN Operator

- Find all projects, if any, on which at least one male is working.
- Find all projects, if any, on which no male is working.
- Find all projects, if any, on which at least one male and at least one female is working.

The IN Operator

Using the NATURAL JOIN Operator

```
SELECT proj_name
FROM projects NATURAL JOIN employees NATURAL JOIN works
WHERE sex = 'M'
GROUP BY proj
HAVING COUNT(*) > 0;
```

- Any query that can be accomplished by using IN can also be accomplished by using joins.

Outline

The ALL Operator

The ALL Operator

attribute rel_op ALL *relation*

- The ALL operator may be used to test a relationship between an attribute and all of the tuples in a relation.
- The *rel_op* is one of the relative operators =, <>, <, >, <=, and >=.
- The expression is true if the relationship holds between the specified attribute and *all* of the tuples in the relation.

The ALL Operator

The ALL Operator

```
SELECT fname, lname, salary
FROM employees
WHERE salary > AVG(salary);
```

- Find the names and salaries of all employees who earn more than the average salary of all employees at the company.
- The above query will not work. Why not?

The ALL Operator

The ALL Operator

```
SELECT fname, lname, salary
FROM employees
WHERE salary > ALL
     (SELECT AVG(salary)
      FROM employees);
```

- This query will work.

The ALL Operator

The ALL Operator

```
SELECT fname, lname, salary
FROM employees
WHERE salary < ALL
     (SELECT AVG(salary)
      FROM employees AS e, departments AS d
      WHERE e.dept = d.dept
      GROUP BY e.dept);
```

- Find the names and salaries of all employees who earn less than the average salary of each department.

The ALL Operator

The ALL Operator

```
SELECT fname, lname, salary
FROM employees
WHERE salary < ALL
    (SELECT AVG(salary)
     FROM employees AS e, departments AS d
     WHERE e.dept = d.dept
     GROUP BY e.dept);
```

- Find the names and salaries of all employees who earn less than the average salary of each department.
- Find the names and salaries of all employees who earn less than the average salary of their own department.

The ALL Operator

- Find all projects that have more than the average number of people working on them.

Outline

Assignment

Assignment

- Read Subsections 5.1.1 - 5.1.5.