Subqueries

Lecture 9

Robb T. Koether

Hampden-Sydney College

Fri, Feb 2, 2018

1. Subqueries
2. The `IN` Operator
3. Using `IN` and `NOT`
4. The `ALL` Operator

Subqueries

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

Nested Queries

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

The IN Operator

The IN Operator

```
SELECT attribute_list_1
FROM table
WHERE attribute_list_2 IN
(SELECT ...);
```

- The IN operator may be used in the WHERE clause to test whether a value or set of values appears in a table created by a subquery.
- The WHERE clause is true for tuples in the *table* for which the values in *attribute_list_2* match the corresponding attributes in *any* tuple returned by the subquery.

Subqueries

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

Nested Queries

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

The IN Operator

The IN Operator

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

- Find all employees who have dependents.

The IN Operator

The IN Operator

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

- Find all employees who have dependents.
- 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, fname
FROM projects AS p NATURAL JOIN employees
WHERE ssn IN
    (SELECT ssn
     FROM works_on 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_on
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.

Using `IN` and `NOT`

- In the `hardware` database, find all customers who did not order a Wrench.

Using IN and NOT

Example (Using IN and NOT)

```
SELECT name
FROM customers
WHERE cust_no NOT IN
    (SELECT cust_no
    FROM customers
        NATURAL JOIN cust_orders
        NATURAL JOIN orders
    WHERE cat_no = "1234");
```

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
     GROUP BY 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
      GROUP BY 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.