Database System Concepts Chapters 3.6-3.8

CMSC 383

Angie Yang

3.6 Null Values

- Arithmetic expressions with any **null** input value is **null**
  - 5 + **null** returns **null**
- Comparisons with **null** yield **unknown**
  - 1 > **null** returns **unknown (Boolean operation)**
- Predicates in **where** clauses can include **and**, **or**, and **not** Boolean operations, so the definitions of Boolean operators need to deal with **unknown**
  - **and**: The result of *true* **and** *unknown* is *unknown*, *false* **and** *unknown* is *false*, while *unknown* **and** *unknown* is *unknown*.
  - **or**: The result of *true* **or** *unknown* is *true*, *false* **or** *unknown* is *unknown*, while *unknown* **or** *unknown* is *unknown*.
  - **not**: The result of **not** *unknown* is *unknown*.

Null Predicates

- **is null** can be used in a predicate to test for a null value
  - Find the names of instructors who have salaries that are null

```
select name
from instructor
where salary is null;
```

- **is not null** would yield the names of instructors who have salaries that are not null

3.7 Aggregate Functions

- input : collection of values, output : single value
    - Average: **avg**
    - Minimum: **min**
    - Maximum: **max**
    - Total: **sum**
    - Count: **count**

Basic Aggregation

- Find the average salary of instructors in the Computer Science department

```
select avg (salary)
from instructor
where dept_name = 'Comp. Sci.';
```

- Find the total number of (distinct) instructors who teach a course in the Spring 2018 semester

```
select count (distinct ID)
from teaches
where semester = 'Spring' and year = 2018;
```

- Find the number of tuples in the *course* relation

```
select count (*)
from course;
```

Aggregation with Grouping

- Find the average salary in each department

```
select dept_name, avg (salary) as avg_salary
from instructor
group by dept_name;
```

dept_name	*avg_salary*
Biology	72000
Comp. Sci.	77333
Elec. Eng.	80000
Finance	85000
History	61000
Music	40000
Physics	91000

ID	*name*	*dept_name*	*salary*
76766	Crick	Biology	72000
45565	Katz	Comp. Sci.	75000
10101	Srinivasan	Comp. Sci.	65000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
22222	Einstein	Physics	95000

Aggregate with Grouping Cont.

- Unaggregated attributes that are in the **select** clause have to be included in the **group by** clause

```
/* erroneous query */
select dept_name, ID, avg (salary)
from instructor
group by dept_name;
```

  - ID does not appear in **group by** and is not aggregated, but appears in **select**
    - Each instructor in a department could have a unique ID, but only one tuple is allowed as output for each department

The Having Clause

- Applies to groups (from **group by** clause) rather than tuples
  - Find departments where the average salary of the instructors is more than $42,000

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

- **having** must appear after **group by** - data must be aggregated first
- If a **where** clause is present, it must appear before **group by** - tuples that satisfy **where** then are put into groups

3.8 Nested Subqueries

Subquery: **select-from-where** expression nested inside another query

- Perform tests for set membership, compare sets, etc…
- In **where** clauses or **from** clauses

Set Membership

- Find all the courses taught in both the Fall 2017 and Spring 2018 semesters

```
select distinct course_id
from section
where semester = 'Fall' and year= 2017 and
      course_id in (select course_id
                    from section
                    where semester = 'Spring' and year= 2018);
```

- Generate distinct classes taught in Fall 2017 that were also taught in Spring 2018
  - Instead of intersecting two separate sets
- Can also use operator **not in**

Set Comparison

- Find the names of all instructors whose salary is greater than at least one instructor in the Biology department

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

- Subquery outputs set of all salary values of every instructor in Biology department
- Name is only selected if salary value for an instructor is higher than at least one instructor in Biology
- **some** can be replaced with **all** to mean “greater than all instructors in Biology”

Test for Empty Relations

- Find all courses taught in both the Fall 2017 semester and in the Spring 2018 semester

```
select course_id
from section as S
where semester = 'Fall' and year= 2017 and
        exists (select *
                from section as T
                where semester = 'Spring' and year= 2018 and
                        S.course_id= T.course_id);
```

- **exists** returns true if argument subquery is nonempty
  - If match in subquery, outer query will select and output
- Correlated subquery (S) can be used in a subquery in a **where** clause

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

- Subquery generates a relation of department names and each department's average instructor salary
- **as** clause renames the result relation as *dept_avg*

The End