Lecture 8: SQL Review

Core 109S IDWT?, Spring 2017
Michael Hay

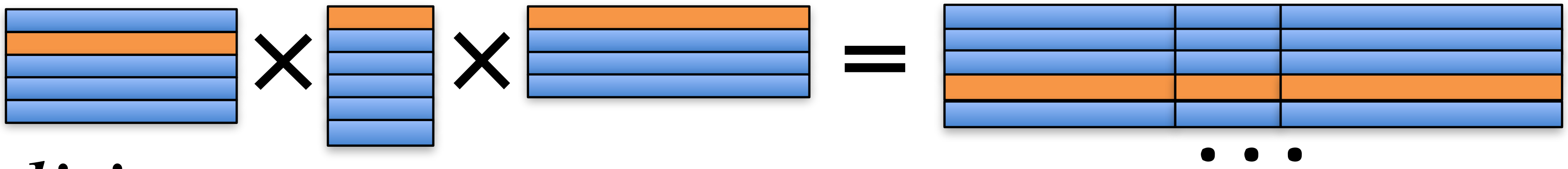
Putting it all together

SELECT *columns or expressions*

5. Compute one output row for each "wide row" (or for each group of them if query has grouping/aggregation)

FROM *tables*

1. Generate all combinations of rows, one from each table; each combination forms a "wide row"

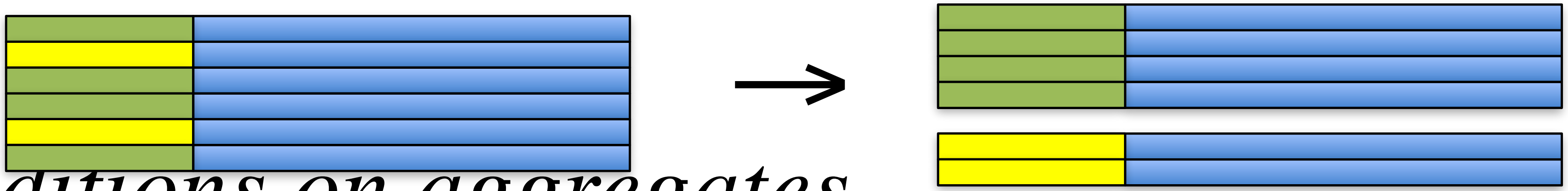


WHERE *conditions*

2. Filter—keep only "wide rows" satisfying *conditions*

GROUP BY *columns*

3. Group—"wide rows" with matching values for *columns* go into the same group



HAVING *conditions on aggregates*

4. Filter groups based on aggregates

ORDER BY *output columns*; 6. Sort the output rows

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

SQL Continued…

- Let's start with a review the handout from previous class.

Exercise 0

Instructions: ~2 minute to think/ answer on your own; then discuss with neighbors; then I will call on one of you

- Remember that you can perform basic math operations on numerical attributes. You can do this in the select clause and also the where clause.
- Write a query that returns names of students who are applying to a school that is at least 10x larger than their high school. Each name should appear at most once.

```
-- ANSWER
select distinct sname
from student s, apply a, college c
where s.sid = a.sid and a.cname = c.cname
and s.sizehs * 10 <= c.enrollment;
```

Exercise 1

Instructions: ~2 minute to think/ answer on your own; then discuss with neighbors; then I will call on one of you

- Write a query that finds the names of students who submitted at least 3 applications.
- Bonus: recall that a student can apply more than once to the same school. How would your answer change if we wanted students who applied to at least 3 different schools?

```
-- ANSWER
-- without a subquery
select sname
from student s, apply a
where s.sid = a.sid
group by s.sid, sname
having count(*) >= 3;

-- with a subquery
select sname
from student s, (select sid, count(*) as
numapps
from apply
group by sid
having count(*) >= 3) t
where s.sid = t.sid;
```

Exercise 2

Instructions: ~2 minute to think/ answer on your own; then discuss with neighbors; then I will call on one of you

```
select cName
from Apply
group by cName
having count(distinct sID) < 5;
```

The HAVING clause is convenient. However it is an *unnecessary* feature of the SQL language because you can accomplish the same thing by using a subquery in the FROM clause or using the WITH clause.

Rewrite this query so it does not use a HAVING clause. The query finds colleges with fewer than 5 distinct applicants.

```
-- ANSWER
with applicantCounts as
(select cName, count(distinct sID) as
numstudents
from Apply
group by cName)
select cName
from applicantCounts
where numstudents < 5;
```

Exercise 3

Instructions: ~2 minute to think/answer on your own; then discuss with neighbors; then I will call on one of you

- Write a query that finds the average number of applications to a college. The result should be a single number (a table with one row and one column).
- Bonus: how would your answer change if we wanted instead the average number of *applicants* rather than *applications?*

```
-- ANSWER
with applicantCounts as
(select cName, count(*) as numstudents
from Apply
group by cName)
select avg(numstudents)
from applicantCounts;
```

Exercise 4

Instructions: ~1 minute to think/ answer on your own; then discuss with neighbors; then I will call on one of you

What does this query compute?

```
select s1.sname, s2.sname
from student s1, student s2
where s1.gpa = s2.gpa
and s1.sizehs = s2.sizehs
and s1.sid != s2.sid;
```

Note: there is a subtle "mistake" in this query that we will fix after we run it.

Exercise 5

Instructions: ~1 minute to think/answer on your own; then discuss with neighbors; then I will call on one of you

- Write a query that finds pairs of colleges located in the same state.
- For reference, here is the last query we looked at:

```
select s1.sname, s2.sname
from student s1, student s2
where s1.gpa = s2.gpa
and s1.sizehs = s2.sizehs
and s1.sid != s2.sid;
```

```
-- ANSWER
select c1.cname, c2.cname
from college c1, college c2
where c1.state = c2.state
and c1.cname < c2.cname;
```

Exercise 6

Instructions: ~2 minute to think/ answer on your own; then discuss with neighbors; then I will call on one of you

- Write a query that finds the maximum total number of applications received in a *state*. The result should be a single number (a table with one row and one column).

```
-- ANSWER
with stateapps as
(select state, count(*) as numapps
from apply a, college c
where a.cname = c.cname
group by state)
select max(numapps) from stateapps;
```