Lecture 6: Manipulating Tabular Data, Part 2

COSC 480 Data Science, Spring 2017

Michael Hay

Goals for today

- Finish up tour of SQL: a *declarative* language for creating, manipulating relational data

  - Declarative = say ***what*** you want, not ***how***

Please follow submission instructions!

Lab commit messages

- completed lab 1: this lab was quite chanllenging; I've come to observe the art in data processing. Very educational experience
- completed lab1: this lab was challenging and long.
- completed everything except prelab; this lab took a lot of time...
- completed lab1: definitely done this time
- lab1 done
- working on challenge problems/cleaning up code
- completed lab1

SQL Overview

See today's handout.

Exercise 1

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

cName	state	enrollment
Colgate	NY	2700
Bucknell	PA	3650
Williams	MA	2000
Cornell	NY	21000

- Write a query to compute total college enrollments by state. On the input relation above, it would produce this output:

state	totalEnroll
MA	2000
NY	23700
PA	3650

Exercise 2

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

cName	state	enrollment
Colgate	NY	2700
Bucknell	PA	3650
Williams	MA	2000
Cornell	NY	21000

- Write a query to find states whose total college enrollment exceeds 20,000. On the input relation above, it would produce this output:

state
NY

Exercise 3

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 students who applied to colleges in NY state. The query should return the student name along with the college name.

sName	cName
Amy	Colgate
Amy	Cornell
Craig	Cornell
…	

Exercise 6

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 the largest number of CS applications received by any one school.

Hint: write a subquery in the FROM clause that computes the number of CS applications for each school.

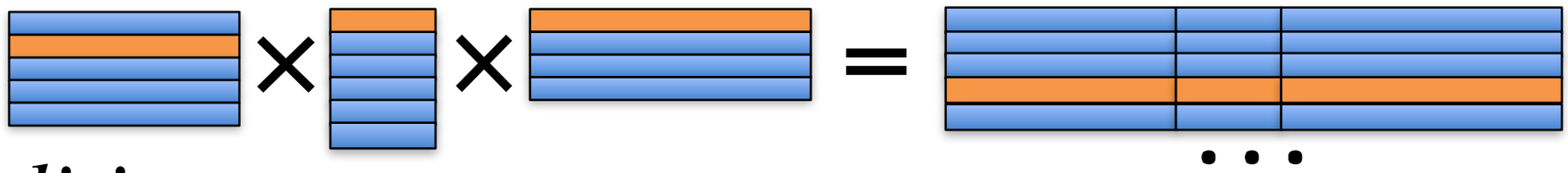
Putting it together

`SELECT` *columns or expressions*

4. 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"

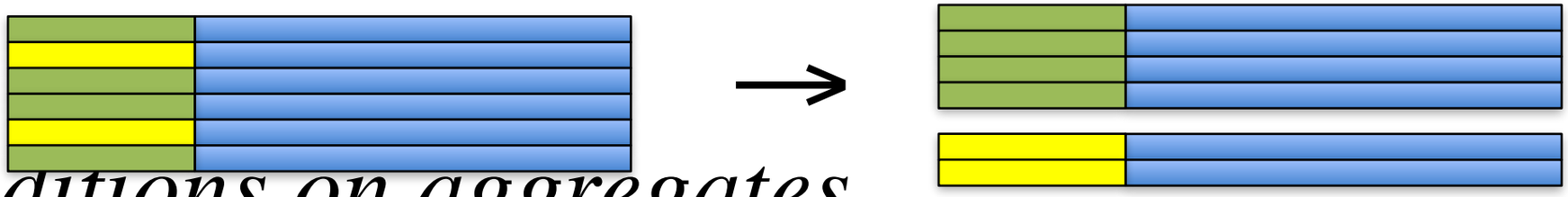


`WHERE` *conditions*

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*

5. 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

Additional exercises

Exercise 7

Instructions: ~1 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.

Exercise 4

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

We want a query that returns for each student, the student's name along with the number of colleges that student applied to. Does this query return the correct result? If not, where is the error?

A. yes, it's correct

B. error in select statement

C. error in from clause

D. error in where clause

E. error in group by clause

F. more than one error

```
select sName, count(*)
from Student S, Apply A
where S.sID = A.sID
group by sName
```

Exercise 5

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

Suppose the Student and Apply relations were as shown on the right. Consider the following query. Which student is *not* included in the result?

A. Amy

B. Bob

C. Craig

D. Doris

E. More than one is not included

```
select S.sID, sName, count(*)
from Student S, Apply A
where S.sID = A.sID
group by S.sID, sName
having count(*) < 4;
```

Student

```
123,Amy,3.9,1000
234,Bob,3.6,1500
345,Craig,3.5,500
456,Doris,3.9,1000
```

Apply

```
123,Colgate,CS,Y
123,Colgate,english,N
123,Bucknell,CS,Y
123,Cornell,english,Y
234,Bucknell,biology,N
345,Williams,chemistry,Y
```