

## Announcements.

- HW8 now due Tuesday (4/18).

Last modified: Wed Apr 12 13:27:41 2017

CS61A1, Lecture #30 1

## Lecture 30: Aggregation and Grouping

Last modified: Wed Apr 12 13:27:41 2017

CS61A1, Lecture #30 2

### Aggregation, Again

- We briefly saw examples of *aggregation* in a previous lecture:

```
> select max(score) from grades;
20
> select avg(score) from grades;
12.0769230769231
...
> select avg(score) from grades
... where assign="hw1";
2.0
```

name	assign	score
John Brown	hw1	2
Walt Green	hw1	3
Valerie Blue	hw1	1
Simon Red	hw2	3
John Brown	test1	20
Walt Green	test1	14
John Brown	test2	19
Valerie Blue	test1	14
Simon Red	test1	17
Walt Green	test2	12
Valerie Blue	test2	15
Sarah Tan	test2	19
Sarah Tan	test1	18

Last modified: Wed Apr 12 13:27:41 2017

CS61A1, Lecture #30 3

### Aggregation

- Sometimes, we'd like a query that groups the data into subsets and aggregates each.

- A clumsy approach:

```
> select assign, avg(score) from grades where assign="hw1" union
... select assign, avg(score) from grades where assign="hw2" union
... select assign, avg(score) from grades where assign="test1" union
... select assign, avg(score) from grades where assign="test2";
hw1|2.0
hw2|3.0
test1|16.6
test2|16.25
```

- First, groups rows with the same `assign` column value. Then runs the query on each group separately, unioning the results.

Last modified: Wed Apr 12 13:27:41 2017

CS61A1, Lecture #30 4

### Selecting Groups

- Just as we often want to filter *rows*, we may also need to filter *groups*.

- Example: I want a summary of assignments that have *at least two submissions*.

- The *where* clause isn't quite right, because it happens *before* grouping.
- So for groups, we use a new clause:

```
having:
> select assign, avg(score)
... from grades
... group by assign
... having count(*) >= 2;
hw1|3|2.0
test1|5|16.6
test2|4|16.25
```

name	assign	score
John Brown	hw1	2
Walt Green	hw1	3
Valerie Blue	hw1	1
Simon Red	hw2	3
John Brown	test1	20
Walt Green	test1	14
John Brown	test2	19
Valerie Blue	test1	14
Simon Red	test1	17
Walt Green	test2	12
Valerie Blue	test2	15
Sarah Tan	test2	19
Sarah Tan	test1	18

Last modified: Wed Apr 12 13:27:41 2017

CS61A1, Lecture #30 5

### A Bit Fancier

- I'd like average scores for each *category* of assignment:

assign	type
hw1	hw
hw2	hw
test1	test
test2	test

Last modified: Wed Apr 12 13:27:41 2017

CS61A1, Lecture #30 6

## A Bit Fancier

- I'd like average scores for each *category* of assignment:

categories	assign	type
	hw1	hw
	hw2	hw
	test1	test
	test2	test

```
> select type, avg(score) from grades, categories
... where grades.assign = categories.assign
... group by type;
hw|2.25
test|16.4444444444444
```

Last modified: Wed Apr 12 13:27:41 2017

CS61A, Lecture #30 7

## Some Bells and Whistles

- We can sort the rows presented, and can filter out duplicates:

```
> select name from grades
... order by name;
John Brown
John Brown
John Brown
Sarah Tan
Sarah Tan
Simon Red
Simon Red
Valerie Blue
Valerie Blue
Valerie Blue
Walt Green
Walt Green

> select distinct name from
grades
... order by name;
John Brown
Sarah Tan
Simon Red
Valerie Blue
Walt Green
```

Last modified: Wed Apr 12 13:27:41 2017

CS61A, Lecture #30 8

## One More Bell

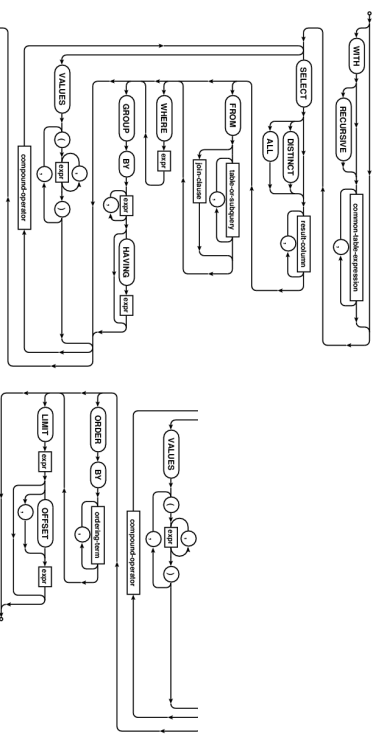
- Finally, can limit the number of responses:

```
> select name from grades order by name limit 8;
John Brown
John Brown
John Brown
Sarah Tan
Sarah Tan
Simon Red
Simon Red
Valerie Blue
```

Last modified: Wed Apr 12 13:27:41 2017

CS61A, Lecture #30 9

## Syntax of Select



Extracted from <https://www.sqlite.org/lang.html>

Last modified: Wed Apr 12 13:27:41 2017

CS61A, Lecture #30 10