Announcements.

• HW8 now due Tuesday (4/18).

Aggregation, Again

• We briefly saw examples of aggregation in a previous lecture:

  > select max(score) from grades;
  20

  > select avg(score) from grades;
  12.076923076923

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

<table>
<thead>
<tr>
<th>name</th>
<th>assign</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Brown</td>
<td>hw1</td>
<td>2</td>
</tr>
<tr>
<td>Walt Green</td>
<td>hw1</td>
<td>3</td>
</tr>
<tr>
<td>Valerie Blue</td>
<td>hw1</td>
<td>1</td>
</tr>
<tr>
<td>Simon Red</td>
<td>hw2</td>
<td>3</td>
</tr>
<tr>
<td>John Brown</td>
<td>test1</td>
<td>20</td>
</tr>
<tr>
<td>Walt Green</td>
<td>test1</td>
<td>14</td>
</tr>
<tr>
<td>John Brown</td>
<td>test2</td>
<td>19</td>
</tr>
<tr>
<td>Valerie Blue</td>
<td>test1</td>
<td>14</td>
</tr>
<tr>
<td>Simon Red</td>
<td>test1</td>
<td>17</td>
</tr>
<tr>
<td>Walt Green</td>
<td>test2</td>
<td>12</td>
</tr>
<tr>
<td>Valerie Blue</td>
<td>test2</td>
<td>15</td>
</tr>
<tr>
<td>Sarah Tan</td>
<td>test2</td>
<td>19</td>
</tr>
<tr>
<td>Sarah Tan</td>
<td>test1</td>
<td>18</td>
</tr>
</tbody>
</table>

Aggregation

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

A Bit Fancier

• We’d like average scores for each category of assignment:

<table>
<thead>
<tr>
<th>assign</th>
<th>type</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>hw1</td>
<td>hw</td>
<td>3</td>
</tr>
<tr>
<td>hw2</td>
<td>hw</td>
<td>4</td>
</tr>
<tr>
<td>test1</td>
<td>test</td>
<td>5</td>
</tr>
<tr>
<td>test2</td>
<td>test</td>
<td>4</td>
</tr>
</tbody>
</table>

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;

<table>
<thead>
<tr>
<th>assign</th>
<th>count</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>hw1</td>
<td>3</td>
<td>2.0</td>
</tr>
<tr>
<td>test1</td>
<td>5</td>
<td>16.6</td>
</tr>
<tr>
<td>test2</td>
<td>4</td>
<td>16.25</td>
</tr>
</tbody>
</table>

A Bit Fancier

• We briefly saw examples of aggregation in a previous lecture.

Aggregation, Again
A Bit Fancier

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

categories
assign
type
hw1
hw
test1
test

- select type, avg(score) from grades, categories
- where grades.assign = categories.assign
- group by type;

hw | 2.25
---
test | 16.4444444444444

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
  Simon Red
  Valerie Blue
  Walt Green

- Finally, can limit the number of responses:

  select name from grades
  order by name
  limit 8;

  John Brown
  John Brown
  John Brown
  Sarah Tan
  Simon Red
  Valerie Blue
  Walt Green

Syntax of Select

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

- There's a method of selecting specific columns.
- You can sort the results presented.
- You can filter out duplicates.
- We can sort the rows presented, and can filter out duplicates:

  select name from grades
  order by name
  limit 8;

- Finally, can limit the number of responses:

  select name from grades
  order by name
  limit 8;