

## 61A Lecture 34

Wednesday, April 22

## Announcements

- Project 4 due Thursday 4/23 @ 11:59pm
  - Early point #2: All questions (including Extra Credit) by Wednesday 4/22 @ 11:59pm
- Recursive Art Contest Entries due Monday 4/27 @ 11:59pm
  - Email your code & a screenshot of your art to [cs61a-tae@mail.eecs.berkeley.edu](mailto:cs61a-tae@mail.eecs.berkeley.edu) (Albert)
- Homework 9 merged with Homework 10; both are due Wednesday 4/29 @ 11:59pm
- Quiz 4 (SQL) released on Tuesday 4/28 is due Thursday 4/30 @ 11:59pm

## Aggregation

## Aggregate Functions

So far, all SQL expressions have referred to the values in a single row at a time

```
[expression] as [name], [expression] as [name], ...
```

```
select [columns] from [table] where [expression] order by [expression];
```

An aggregate function in the [columns] clause computes a value from a group of rows

```
create table animals as
select "dog" as kind, 4 as legs, 20 as weight union
select "cat"      , 4      , 10      union
select "ferret"   , 4      , 10      union
select "parrot"   , 2      , 6       union
select "penguin" , 2      , 10      union
select "t-rex"    , 2      , 12000;
```

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

```
select max(legs) from animals;
```

max(legs)
4

 (Demo)

## Mixing Aggregate Functions and Single Values

An aggregate function also selects a row in the table, which may be meaningful

```
select max(weight), kind from animals;
select min(kind), kind from animals;
select max(legs), kind from animals;
select avg(weight), kind from animals;
```

(Demo)

```
create table animals as
select "dog" as kind, 4 as legs, 20 as weight union
select "cat"      , 4      , 10      union
select "ferret"   , 4      , 10      union
select "parrot"   , 2      , 6       union
select "penguin" , 2      , 10      union
select "t-rex"    , 2      , 12000;
```

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

## Groups

## Grouping Rows

Rows in a table can be grouped, and aggregation is performed on each group

```
[expression] as [name], [expression] as [name], ...
```

```
select [columns] from [table] group by [expression] having [expression];
```

The number of groups is the number of unique values of an expression

```
select legs, max(weight) from animals group by legs;
```

legs	max(weight)
4	20
2	12000

(Demo)

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

## Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

```
[expression] as [name], [expression] as [name], ...
```

```
select [columns] from [table] group by [expression] having [expression];
```

A **having** clause filters the set of groups that are aggregated

```
select weight/legs, count(*) from animals group by weight/legs having count(*)>1;
```

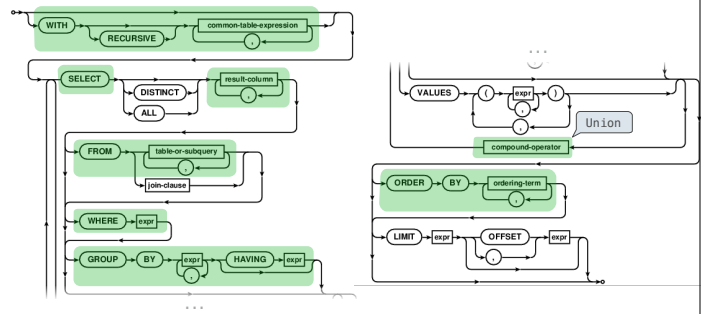
weight/legs	count(*)
5	2
2	2

weight/legs=5  
weight/legs=2  
weight/legs=2  
weight/legs=3  
weight/legs=5  
weight/legs=6000

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

## Select Grammar

## SQL Select Statements



[http://www.sqlite.org/lang\\_select.html](http://www.sqlite.org/lang_select.html)