Mapping the world in 3D will let us paint streets with augmented reality

`www.technologyreview.com/s/613311/mapping-the-world-in-3d-will-let-us-paint-streets-with-augmented-reality/`

"If you believe tech optimists, 10 years from now self-driving cars will be ubiquitous, drones will deliver our parcels, and robots will bring us our groceries. And one day soon, our cities will be painted with augmented reality that feels as if it belongs to the street corner where it was placed. Whether or not any of that comes to pass, one piece of the puzzle will be crucial to this future: ultra-precise location technology. GPS and the wandering blue dot on smartphone mapping apps are useful for a human navigating an unfamiliar city, but that just won't cut it for machines. They will need to know where things are down to the centimeter.

London-based startup Scape reckons that's what it can provide. The firm's visual positioning service uses GPS and multiple camera images to work out exactly where you are within two or three seconds, according to cofounder Edward Miller. **It has collected over two billion street images to precisely 3D-map more than 100 cities around the world, including London, San Francisco, Paris, Moscow, and Tokyo**. Some of the data was collected by employees cycling around the cities with cameras attached to their bicycles, but Scape's platform can process images from any source."

Announcements

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;
```

animals:

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 some row in the table to supply the values of columns that are not aggregated. In the case of max or min, this row is that of the max or min value. Otherwise, it is arbitrary.

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

select min(kind), 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;
```

animals:

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

Discussion Question

What are all the kinds of animals that have the maximal number of legs?

Groups

Grouping Rows

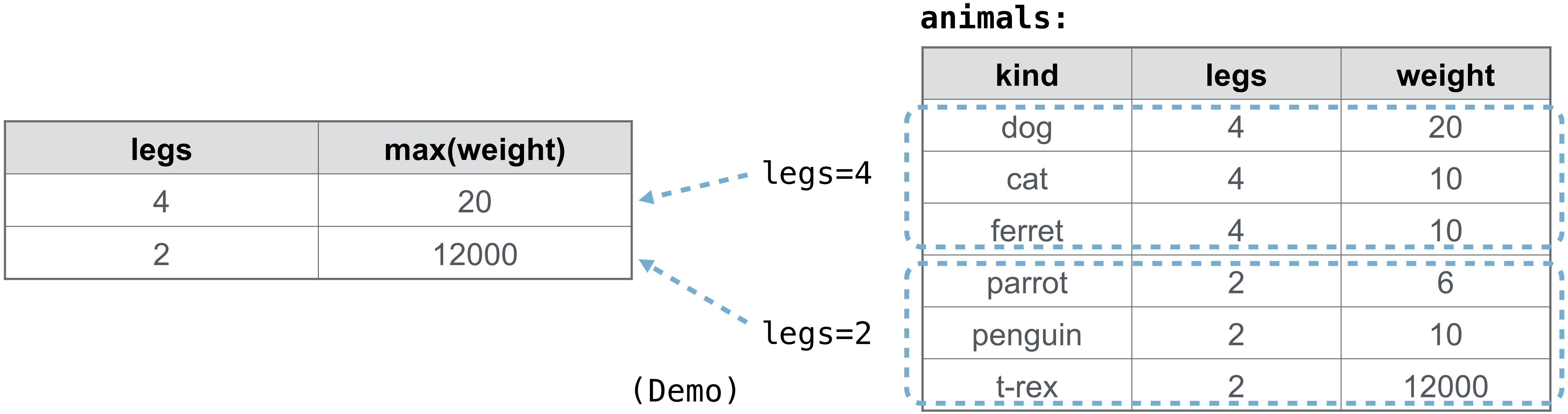
Rows in a table can be grouped, and aggregation is performed on each group

```
[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

animals:

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

(Demo)

Selecting Groups

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

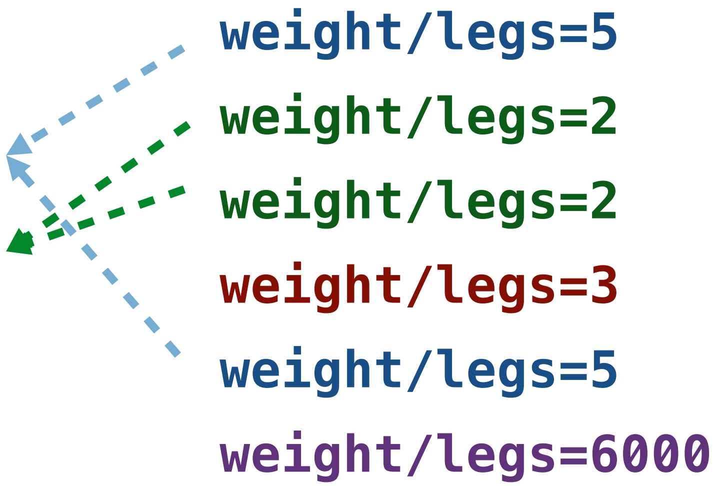
```
[expression] as [name], [expression] as [name], ...
```

```
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

animals:

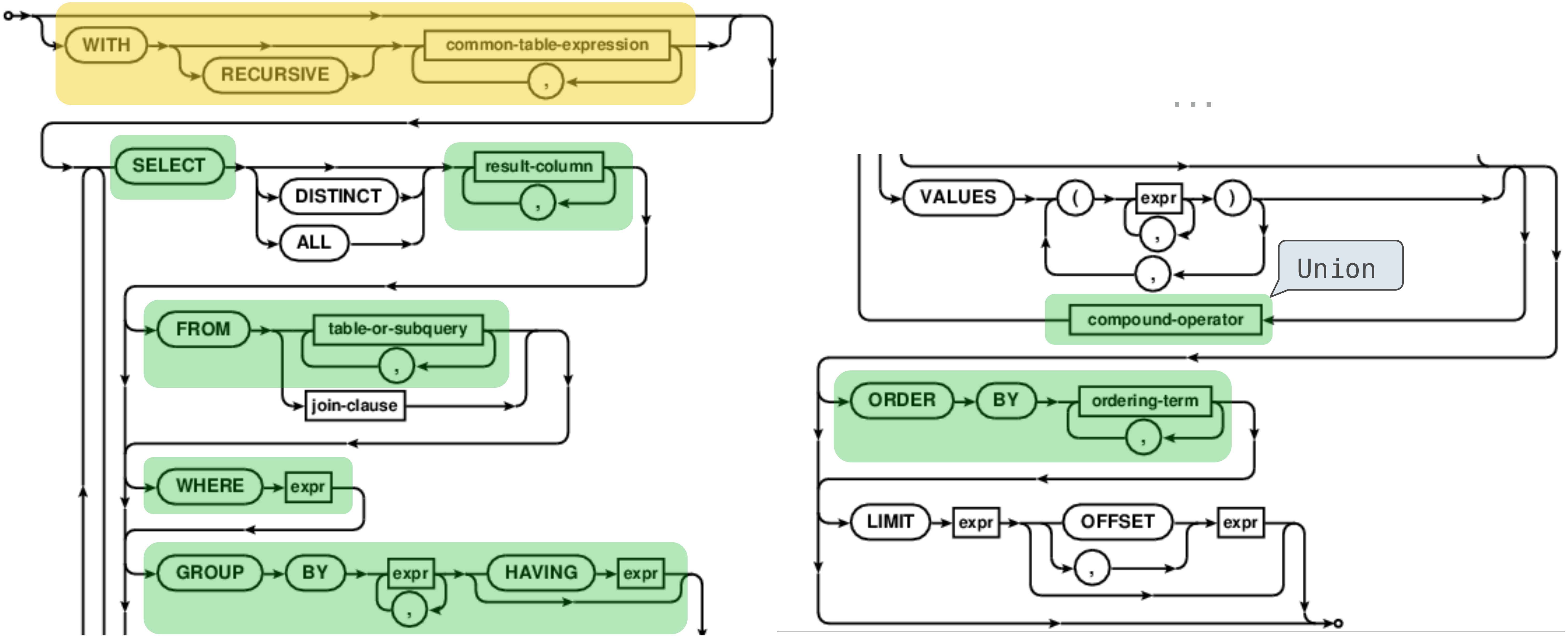
kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

Discussion Question

```
What's the maximum difference between leg count for two animals with the same weight?
```

SQL Grammar

SQL Select Statements



Union

. . .

. . .

http://www.sqlite.org/lang_select.html