Welcome to Data C88C!

Lecture 22: Aggregation
Thursday, July 31st, 2025
Week 6
Summer 2025
Instructor: Eric Kim (ekim555@berkeley.edu)

Announcements

- Midterm regrades: due this Friday
  - Midterm solutions doc released: [link]
- August 1st: Change Grade Option deadline
- Ants project is ongoing! Checkpoint due Mon Aug 4th
-

Lecture Overview

- More SQL
  - Aggregation, GROUP BY

Select Statements

Select: subqueries ("nested queries")

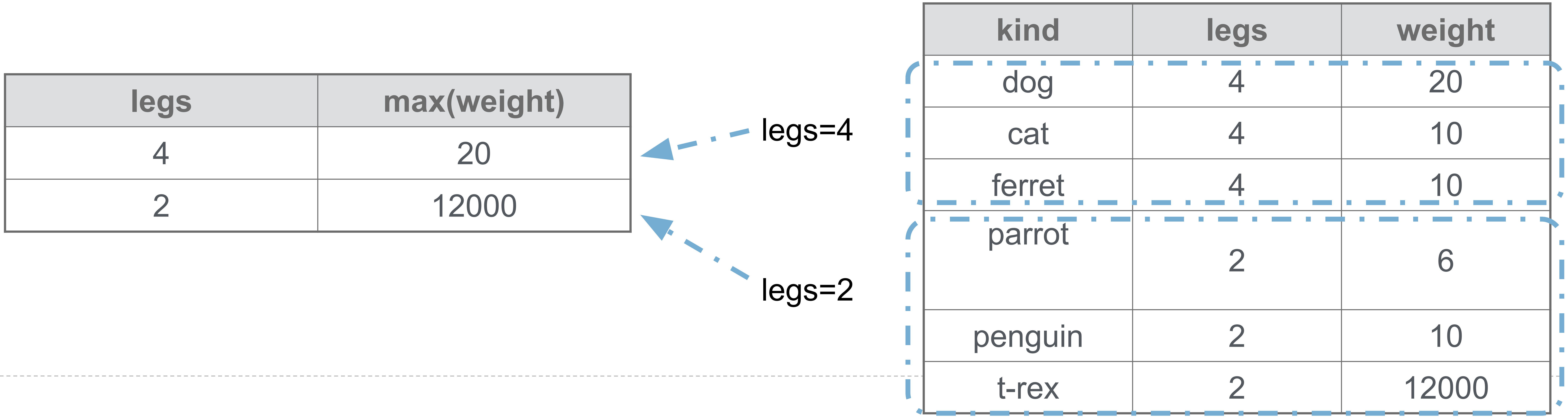
Grouping Rows

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

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

Writing Select Statements

Describe the output table:

Important: WHERE filters before aggregation, HAVING filters after aggregation

1) Determine which existing rows are needed to express the result (FROM & WHERE)

2) Form groups and determine which groups should appear as output rows (GROUP BY & HAVING)

3) Format the output rows (SELECT)

SELECT: Values each output row contains (and column labels)

- **FROM**: Source of input rows
- **WHERE**: Which input rows
- **GROUP BY**: Form output rows
- **HAVING**: Which output rows

Grouping and Aggregations

aggregator function (average)

Group keys (can be more than one!)

```
sqlite> select avg(Price), Flavor from cones group by Flavor;
```

Flavor	Color	Price
strawberry	pink	3.55
chocolate	light brown	4.75
chocolate	dark brown	5.25
strawberry	pink	5.25
bubblegum	pink	4.75

GROUP by Flavor

```
# Flavor, Price          # Flavor, Price         # Flavor, Price
strawberry, 3.55         chocolate, 4.75         bubblegum, 4.75
strawberry, 5.25         chocolate, 5.25
                         chocolate, 5.25
```

avg(Price)

```
# Flavor, Price          # Flavor, Price         # Flavor, Price
strawberry, (3.55        chocolate, (4.75 +      bubblegum,
+ 5.25) / 2              5.25 + 5.25) / 3        (4.75) / 1
```

```
# Flavor, avg(Price)
strawberry, 4.4
chocolate, 5.0833
bubblegum, 4.75
```

avg(Price) is the result of computing the average Price within each group

Group by: "Bare" columns

- Caution: when using "GROUP BY", all selected columns MUST be one of:
  - (1) included in "GROUP BY"
  - Or
  - (2) in an aggregation function

"bare"
column!

```
sqlite> select flavor, avg(price), color from cones
group by flavor;

# flavor, avg(price), color
bubblegum|4.75|pink
chocolate|5.0833333333333|light brown
strawberry|4.4|pink
```

SQL **arbitrarily** chose one of the "chocolate" group Color values: "light brown" or "dark brown". Weird!

Note: in many other SQL engines, it's an error to have a "bare" column.

Flavor	Color	Price
strawberry	pink	3.55
chocolate	light brown	4.75
chocolate	dark brown	5.25
strawberry	pink	5.25
bubblegum	pink	4.75

It's confusing to have a query where it's not clear what the output will be.
Recommendation: put selected columns in either "GROUP BY" clause OR in an aggregation function. Don't have "bare" columns!

For more info on sqlite3's "bare columns", read this interesting (and colorful) thread: https://sqlite.org/forum/forumpost/4c8a673560d7999a

Example: Select Statement Components

For each *type* of *employee*, compute the *fa23-fa18* difference in the total headcount, but include a row only for each *type* for which the headcount increased.

```
sqlite> SELECT * FROM cal;
source    type                 role                            fa08   fa13   fa18   fa23
--------  -------------------  ------------------------------  -----  -----  -----  -----
employee  Grad Student Titles  Grad St. Instructor (GSI)       1943   1925   2202   2248
                ...                                 ...     ...    ...    ...
student   Grad Student         Grad Student                    10258  10253  11666  12621
student   Undergrad            Undergrad                       25151  25951  30853  33078
```

SELECT: Values each output row contains (and column labels)

```
SELECT type, SUM(fa23) - SUM(fa18) AS increase
  FROM cal
  WHERE source = "employee"
  GROUP BY type
  HAVING SUM(fa23) > SUM(fa18);
```

FROM: Source of input rows

WHERE: Which input rows

GROUP BY: Form output rows

HAVING: Which output rows

```
type                 increase
-------------------  --------
Grad Student Titles  327
Other Faculty        352
Regular Faculty      48
Staff                454
```

(reference) SQL in C88C

- Here are all of the queries that we cover in C88C that you (the student) are responsible for
  - Note: there is more to SQL than this, but this is a good starting point (and outside scope for this course)

```
SELECT select_list

[ FROM table_source(s) ] [ WHERE search_condition ]

[ JOIN table ON join_condition ]

[ GROUP BY group_by_expression ]

[ HAVING search_condition ]

[ ORDER BY order_expression [ ASC | DESC ] ]

[ LIMIT [limit] ];


# aggregator functions (used with GROUP BY)
min(), max(), avg(), sum(), count(), count(distinct x), count(*)

# aliasing
select colA AS colA_alias ...

# subqueries ("nested" queries)
```

```
CREATE TABLE [table_name] AS
  SELECT [val1] AS [col1], [val2] AS [col2], ... UNION
  SELECT [val3], [val4], ... UNION
  SELECT [val5], [val6], ...;
```

https://learn.microsoft.com/en-us/sql/t-sql/queries/select-transact-sql?view=sql-server-ver17

Joins Practice

Discussion Question

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

Approach #1: Consider all pairs of animals.

```
SELECT _MAX(a.legs - b.legs)_____ AS difference
  FROM animals AS a, animals AS b
  _WHERE a.weight = b.weight____;
```

Approach #2: Group by weight.

```
 SELECT _MAX(legs) - MIN(legs)_____ AS difference
   FROM ______animals____________
   GROUP BY weight
   ORDER BY difference DESC
   LIMIT 1;
```

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

difference
2

Discussion Question

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

```
sqlite> SELECT * FROM animals WHERE legs = MAX(legs);
Parse error: misuse of aggregate function MAX()
```

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

Approach #1: Give the maximum number of legs a name.

```
CREATE TABLE m AS SELECT _MAX(legs) AS max_legs___ FROM animals;
SELECT kind FROM __animals, m___ WHERE legs = max_legs;
```

Approach #2: For each kind of animal, compare its legs to the maximum legs by grouping.

```
SELECT a.kind__ FROM animals AS a, animals AS b GROUP BY a.kind _HAVING a.legs = MAX(b.legs)_____;
```

Group By Practice

Spring 2023 CS 61A Final Question 7

The finals table has columns hall (strings) and course (strings), and has rows for each lecture hall in which a course is holding its final exam.

The sizes table has columns room (strings) and seats (numbers), and has one row per unique room on campus containing the number of seats in that room. All lecture halls are rooms.

Create a table with two columns, course (string) and seats (number), and with one row containing the **name of the course** and the **total number of seats in final rooms** for that course. Only include a row **for each course that uses at least two rooms for its final**.

finals:

hall	course
RSF	61A
Wheeler	61A
RSF	61B

sizes:

room	seats
RSF	900
Wheeler	700
310 Soda	40

result:

course	seats
61A	1600

```
SELECT course, _SUM(seats)_ AS seats
   FROM _finals, sizes_______
   WHERE hall=room__________
   GROUP BY __course____
   HAVING COUNT(*) > 1______;
```