# CSCI 403 Database Management

11 – Grouping & Aggregation

CS@Mines

# AGGREGATE FUNCTIONS CS@Mines

## **COUNT**

One of the most heavily used functions...

How many rows are in a table?

SELECT COUNT(\*) FROM tablename;

How many rows match a condition?
 SELECT COUNT(\*) FROM tablename
 WHERE conditions;

CS@Mines

# **COUNT** is a Summary

- COUNT is an aggregate function
  - Doesn't act row-by-row
  - Gives a summary of all rows
- COUNT cannot be SELECTed with regular columns...
  - ... would be somewhat meaningless...
  - Until we learn how to group! SELECT COUNT(x), x FROM foo; ERROR!

CS@Mines

### COUNT and NULL foo You can count individual columns. Note: NULL columns don't count! apple banana SELECT COUNT(\*) AS rows, cherry COUNT(1) AS ones, COUNT(x) AS xes, COUNT(y) AS ys pear <null> <null> FROM foo; rows | ones | xes | ys 5 | 5 | 4 | 3 CS@Mines

# Other Aggregate Functions

SUM – adds up non-NULL numeric values

MAX – finds maximum non-NULL entry

MIN - finds minimum non-NULL entry

AVG – calculates the average of non-NULL numbers

STDDEV SAMP – Sample standard deviation

Note these apply to expressions, not rows!

 $See \ \underline{https://www.postgresql.org/docs/9.5/static/functions-aggregate.html} \\ for more!$ 

CS@Mines

# GROUPING CS@Mines

# Grouping

- Compute aggregates on subsets of rows
  - Rows organized by equal values of subset of columns
  - Organizing columns listed in GROUP BY clause
  - Aggregates and grouping columns only in SELECT
- Example:

```
SELECT instructor, count(*)
FROM mines_courses
GROUP BY instructor
ORDER BY count(*) DESC;
```

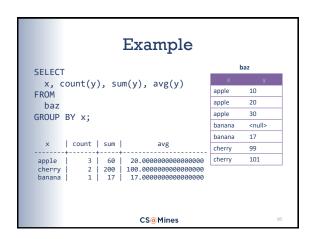
CS@Mines

## How To Think About Grouping

Suppose you group by columns c1 & c2:

- Find all unique combinations of c1 & c2
- Put all rows matching each unique combination into their own group
- Compute aggregate functions on each group
- Return results for each group

CS@Mines



# Grouping and Ordering

- You can combine ordering with grouping:
  - ORDER BY always comes at the end of the query
  - ORDER BY any aggregates or grouping columns/expressions (even if not SELECTed)
- Example:

```
SELECT
substr(course_id, 1, 4) AS subject, count(*)
FROM mines_courses
GROUP BY subject
ORDER BY avg(length(title)) DESC, subject;
```

CS@Mines

# Filtering and Grouping

- WHERE clause applies before grouping
  - Filters rows only on expressions/columns
  - Cannot filter on aggregate functions (not yet computed!)
- HAVING clause applies after grouping
  - Filters group results
  - Can filter on aggregate functions (or expressions/columns)

CS@Mines

