

# Aggregations - 2/22

## Announcements

---

- Demo code is on the github snippets repository.
- Midterm is paper based similar to the quizzes and practice problems.

## Reading Quiz

---

- Q1: The `LIKE` keyword is not a function.
- Q2: In order to query for the number of rows in a table we must use the `SELECT` keyword to specify a query and `COUNT` to say we want the function that counts the number of items.
- Q3: `COUNT` does include the records with `NULL` values.
- Q5: `GROUP BY` divides rows into groups that match on column values.

## Aggregate Functions

---

### Standard Functions

- `MIN`
- `MAX`
- `SUM`
- `AVG`
- `COUNT`

What is the difference between `COUNT(*)`, `COUNT(depid)`, and `COUNT(DISTINCT depid)` ?

Note: You can have more than one aggregate function in a select statement.

### Practice Problem 1

```
SELECT COUNT(*) AS transactions, SUM(qtysold) AS tickets_sold, AVG(commission) AS avg
_commission
FROM Sales;
```

Note: You can use `AS` to alias or rename your columns that you are querying back.

## Practice Problem 2

We need a join since our data that we are querying is in a different table than our search condition. Then we use aggregate functions to find the min and the max.

```
SELECT MAX(l.priceperticket), MIN(l.priceperticket)
FROM Listing l
JOIN Event e ON l.eventid = e.eventid
WHERE e.eventname = 'Spoon';
```

## Groupings

---

Use `GROUP BY` to group together certain groups into the aggregate function's results.

So this is useful for example if you wanted to count the number of transactions per person in a ticket database. You can use group by on the buyer field so that it will give you the counts per each different buyer.

## Practice Problem 3

```
SELECT c.catid, c.catname, COUNT(e.catid) AS total_events
FROM Event e
RIGHT JOIN Category c ON e.catid = c.catid
GROUP BY c.catid, c.catname
ORDER BY c.catname;
```