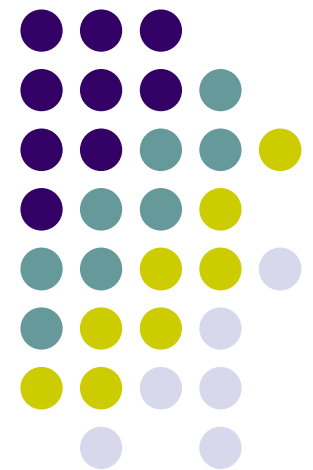


# Aggregate functions

---

**CIS 331:  
Introduction to  
Database Systems**





# Topics:

- **ORDER BY, ASC, DESC**
- **Aggregate functions**
  - **COUNT**
  - **SUM**
  - **MIN**
  - **MAX**
  - **AVG**
- **GROUP BY and HAVING**



# Order By, Asc

- **ORDER BY** sorts results of a query:

```
SELECT first_name, middle_name, last_name
      FROM professors
      ORDER BY last_name ASC
;
```

- Ascending (**ASC**) order is the default, so you do not have to specify it explicitly:

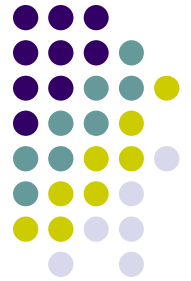
```
SELECT first_name, middle_name, last_name
      FROM professors
      ORDER BY last_name
;
```



# Order By, Desc

- You can also sort the output in descending order (**DESC**):

```
SELECT first_name, middle_name, last_name
      FROM professors
      ORDER BY first_name DESC
;
```



# Aggregate functions: Count

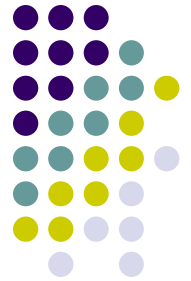
- Suppose we would just like to know the number of students, without necessarily seeing their names, etc:

```
SELECT COUNT (*)  
    FROM students  
;
```

- Or how many students attended a particular class:

```
SELECT class, COUNT (*)  
    FROM students_classes  
    GROUP BY class  
;
```

- GROUP BY groups together records which have something in common: in this case all students who attended the same class.



# Aggregate functions: Min, Max

- We can also calculate the range of grades per class:

```
SELECT class, MIN(grade), MAX(grade)
      FROM students_classes
      GROUP BY class
;
```

- Or a student's GPA:

```
SELECT student, AVG(grade)
      FROM students_classes
      GROUP BY student
      ORDER BY AVG(grade)
;
```



# Having

- Get a list of students who have a GPA over 3.5:

```
SELECT student, AVG(grade)
      FROM students_classes
      GROUP BY student
      HAVING AVG(grade) > 3.5
;
```

- Observe the HAVING clause: it is similar to WHERE, except that you can use aggregate expressions as conditions.



# Aggregate functions: Avg

- GPA of students who took more than 3 classes:

```
SELECT student, AVG(grade)
FROM students_classes
GROUP BY student
HAVING COUNT(*) > 3
ORDER BY student DESC
;
```