HW0 Puzzle: Why different results?

Result: 69

```
import csv
file = open('iowa-liquor-sample.csv')
file_reader = csv.reader(file)
n = 0
for row in file_reader:
  for el in row:
    if "single malt scotch" in el.lower():
      n += 1
print n
```

Result: 51

```
file = open('iowa-liquor-sample.csv','r')
n = 0
for line in file:
  temp = line.lower()
  if 'single malt scotch' in temp:
    n += 1
print n
```

HW0 Puzzle: Why different results?

Result: 69

```
import csv
file = open('iowa-liquor-sample.csv')
file_reader = csv.reader(file)
n = 0
for row in file_reader:
  for el in row:
    if "single malt scotch" in el.lower():
      n += 1
print n
```

Result: 51

```
file = open('iowa-liquor-sample.csv','r')
n = 0
for line in file:
  temp = line.lower()
  if 'single malt scotch' in temp:
    n += 1
print n
```

HW0: Why the different results?

Result: 69

```
import csv
file = open('iowa-liquor-sample.csv')
file_reader = csv.reader(file)
n = 0
for row in file_reader:
  for el in row:
    if "single malt scotch" in el.lower():
      n += 1
print n
```

Result: 51

```
file = open('iowa-liquor-sample.csv','r')
n = 0
for line in file:
  temp = line.lower()
  if 'single malt scotch' in temp:
    n += 1
print n
```

Example record:

```
[…],SINGLE MALT SCOTCH,[…],Macallan 12 Yr Single Malt Scotch,[…]
```

Poll: HW0

A) Short

B) Medium

C) Long

Administrative Notes

- Waiting list: 107! First come, first served (sorry); Looks like 8 should get in today?

Project 1: Build a web application!

Step 1: Find a partner

Step 2: Discuss it with an IA (next week; see web site)

Lecture 2
Entity-Relationship Model

Steps for a New Application

Requirements

- what are you going to build?

Conceptual Database Design

- high-level description

Logical Design

- formal database schema

Schema Refinement

- fix potential problems, normalization

Physical Database Design

- use sample of queries to optimize for speed/storage

Steps for a New Application

Requirements

what are you going to build?

Conceptual Database Design

high-level description

ER Modeling

Logical Design

formal database schema

Schema Refinement:

fix potential problems, normalization

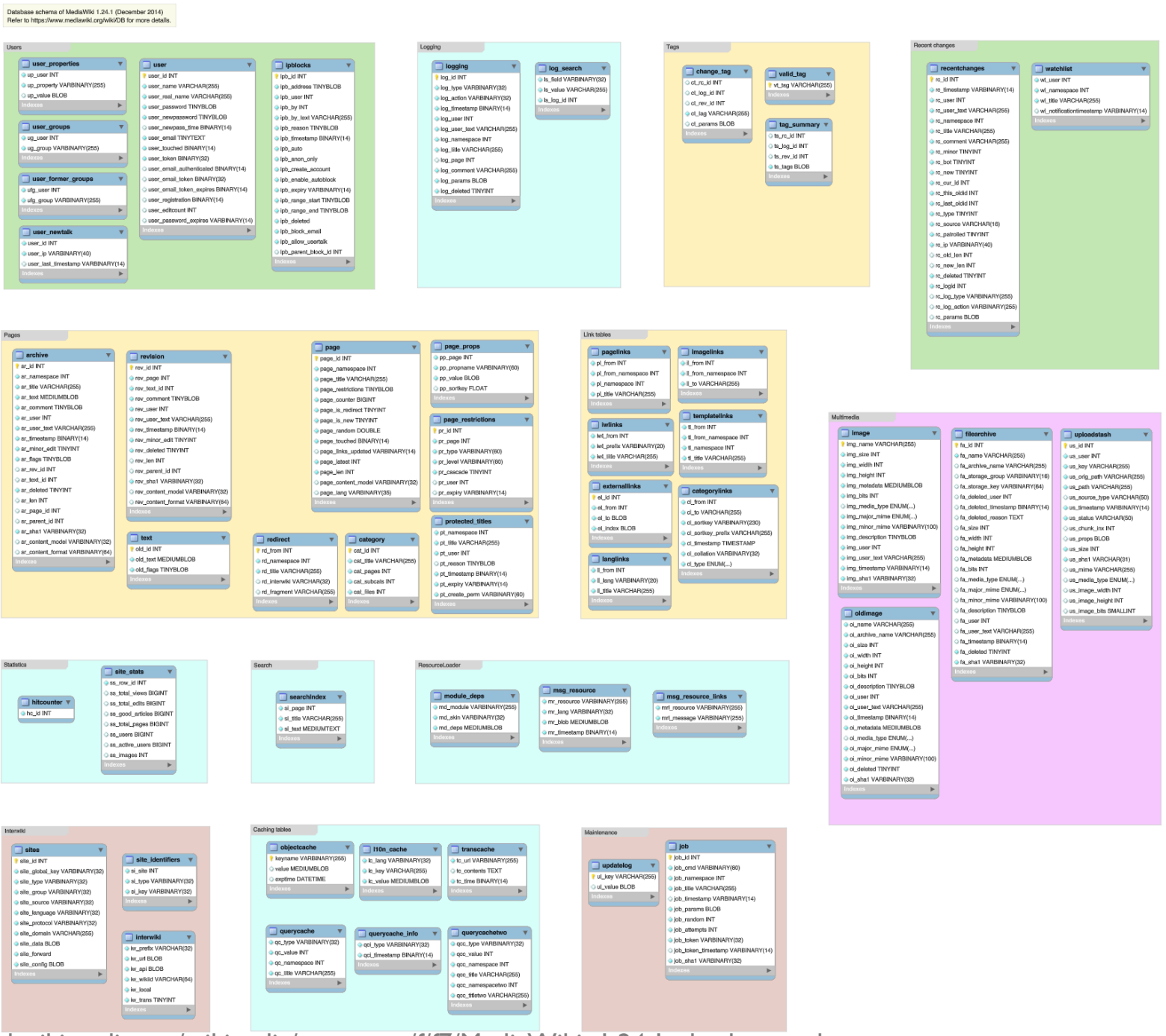
Physical Database Design

use sample of queries to optimize for speed/storage

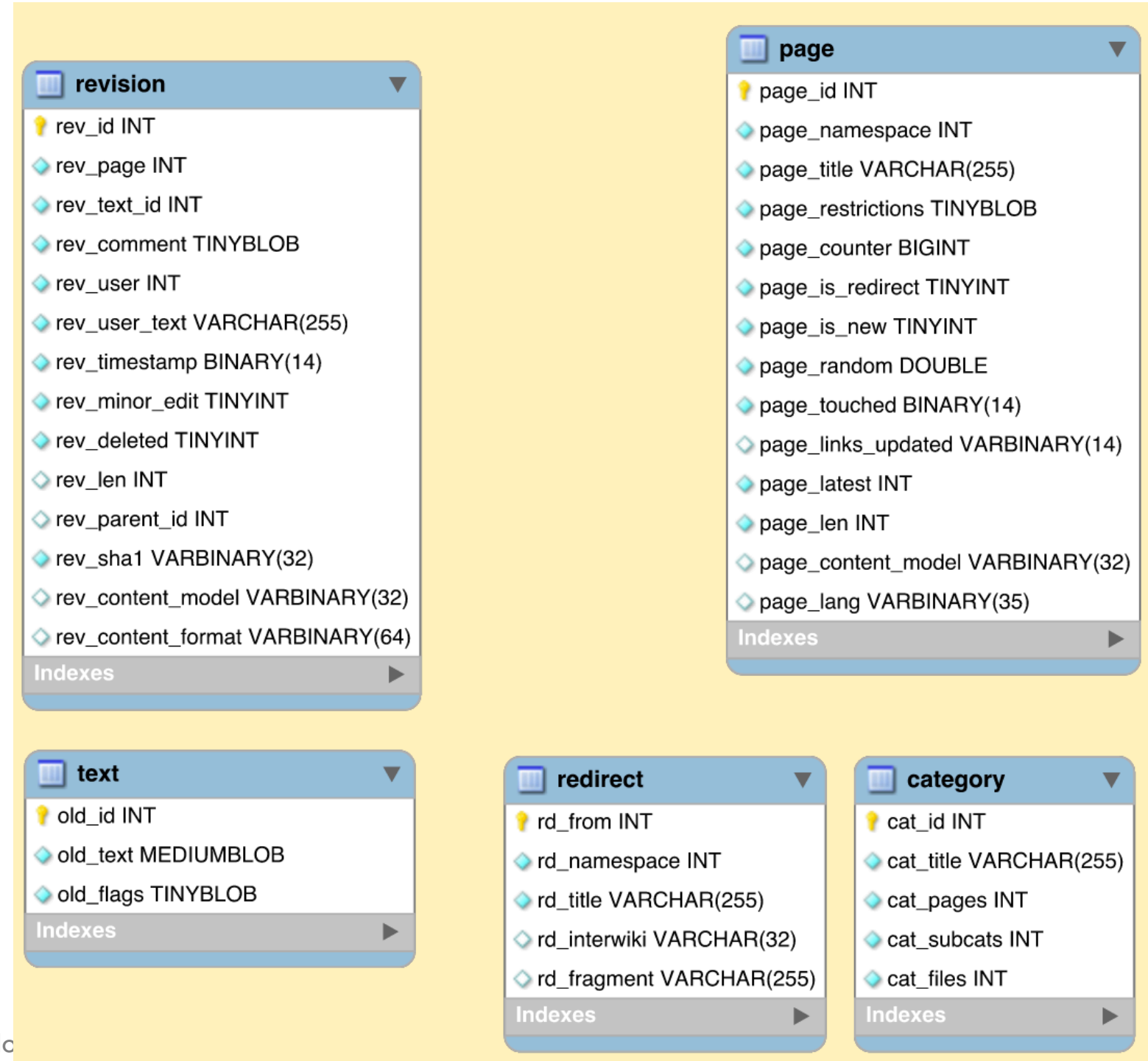
Database Apps Are Complicated

Typical Fortune 100 Company

- ~10k different information (data) systems
- 90% relational databases (DBMSes)
- Typical database has >100 tables
- Typical table has 50 – 200 attributes



https://upload.wikimedia.org/wikipedia/commons/f/f7/MediaWiki_1.24.1_database_schema.svg

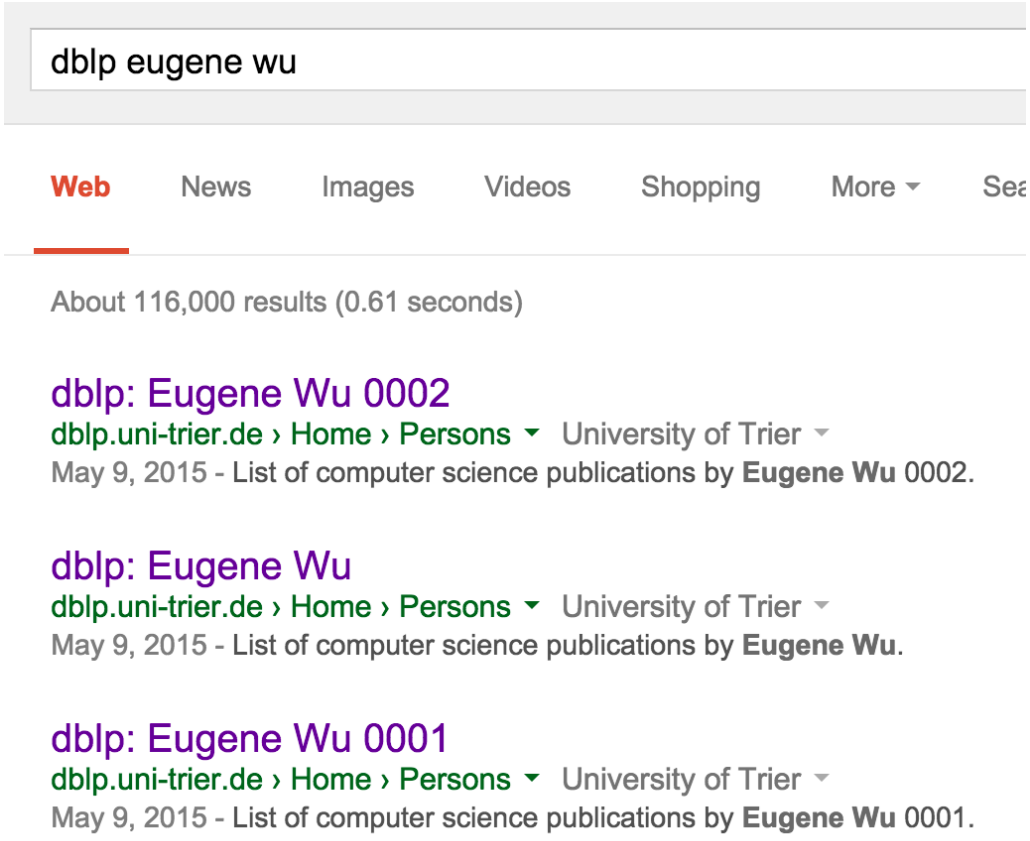

revision
rev_id INT
rev_page INT
rev_text_id INT
rev_comment TINYBLOB
rev_user INT
rev_user_text VARCHAR(255)
rev_timestamp BINARY(14)
rev_minor_edit TINYINT
rev_deleted TINYINT
rev_len INT
rev_parent_id INT
rev_sha1 VARBINARY(32)
rev_content_model VARBINARY(32)
rev_content_format VARBINARY(64)
Indexes
page
page_id INT
page_namespace INT
page_title VARCHAR(255)
page_restrictions TINYBLOB
page_counter BIGINT
page_is_redirect TINYINT
page_is_new TINYINT
page_random DOUBLE
page_touched BINARY(14)
page_links_updated VARBINARY(14)
page_latest INT
page_len INT
page_content_model VARBINARY(32)
page_lang VARBINARY(35)
Indexes
text
old_id INT
old_text MEDIUMBLOB
old_flags TINYBLOB
Indexes
redirect
rd_from INT
rd_namespace INT
rd_title VARCHAR(255)
rd_interwiki VARCHAR(32)
rd_fragment VARCHAR(255)
Indexes
category
cat_id INT
cat_title VARCHAR(255)
cat_pages INT
cat_subcats INT
cat_files INT
Indexes

https://uplc

Inconsistencies/Constraint Violations

Huge amount of effort to avoid inconsistencies

DBLP is *the* site for computer science publications



Inconsistencies/Constraint Violations

[–] **2010 – today** [+] **Refine list**

2014

- [j8] Eugene Wu, Leilani Battle, Samuel R. Madden: **The Case for Data Visualization Management Systems.** PVLDB 7(10): 903-906 (2014)
- [j7] Alekh Jindal, Praynaa Rawlani, Eugene Wu, Samuel Madden, Amol Deshpande, Mike Stonebraker: **VERTEXICA: Your Relational Friend for Graph Analytics!** PVLDB 7(13): 1669-1672 (2014)

≠

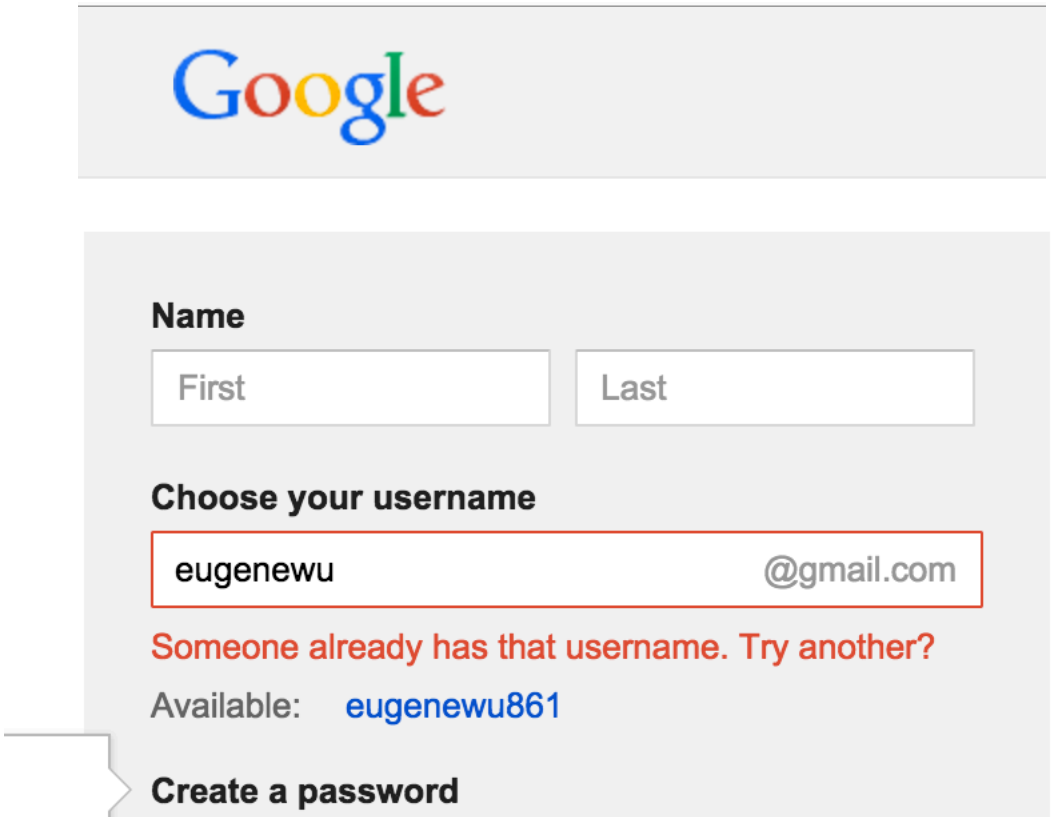
[–] **1990 – 1999** [+] **Refine list**

1994

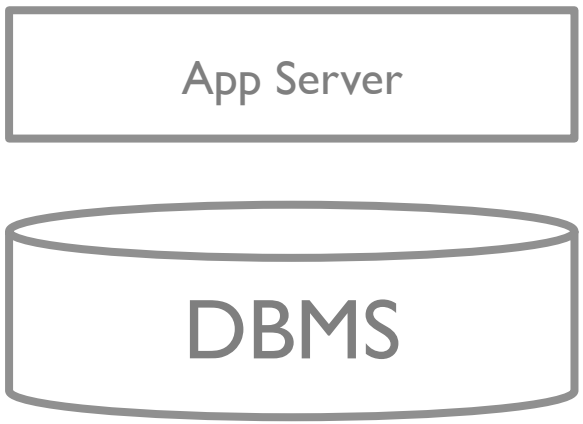
- [c2] James Hwang, Eugene Wu, Alan Bell, Andy Cordell, LeBarian Stokes, Scott Hankins: **Design of a SPDM-Like Robotic Manipulator System for Space Station on Orbit Replaceable Unit Ground Testing - An Overview of the System Architecture.** ICRA 1994: 1286-1291
- [c1] Eugene Wu, James Hwang, Scott Hankins: **Design of the Control System for a Robotic Manipulator for Space Station On-Orbit Replaceable Unit Ground Testing.** ICRA 1994: 1415-1420

Inconsistencies/Constraint Violations

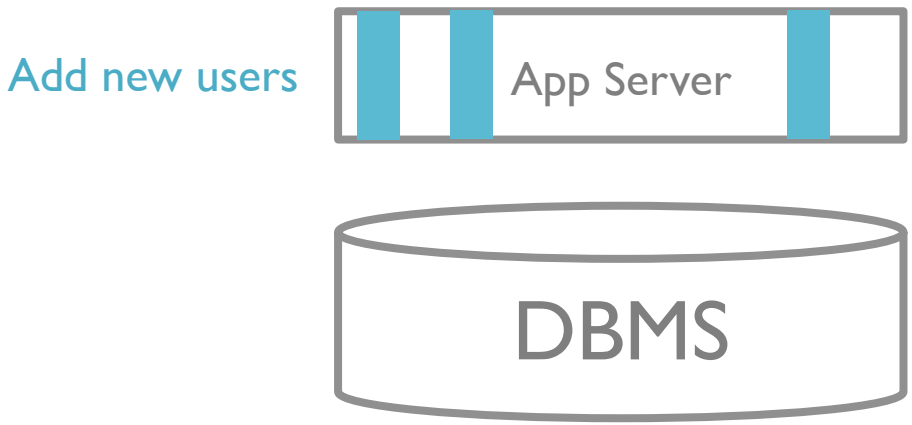
Giving me eugenewu@gmail would violate constraints



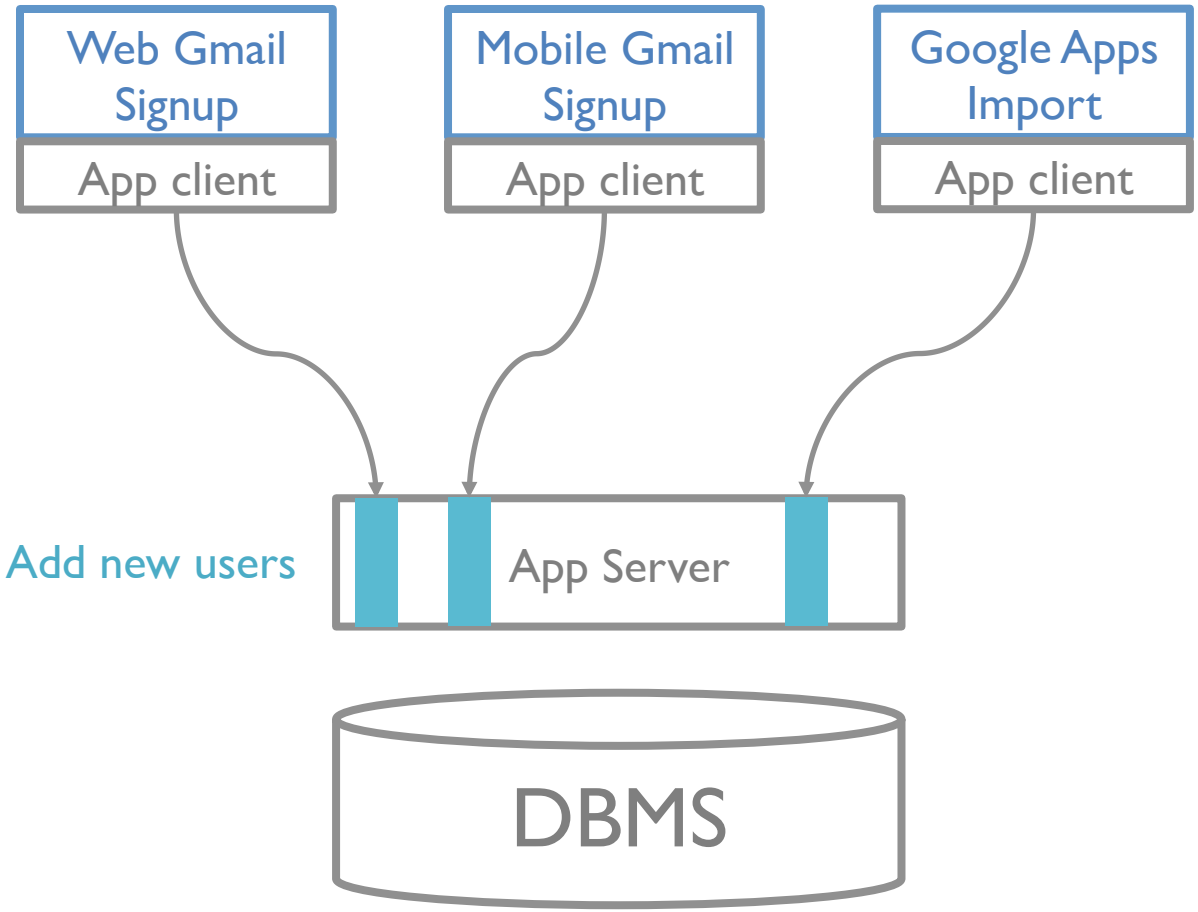
It is Hard to Design Applications



It is Hard to Design Applications

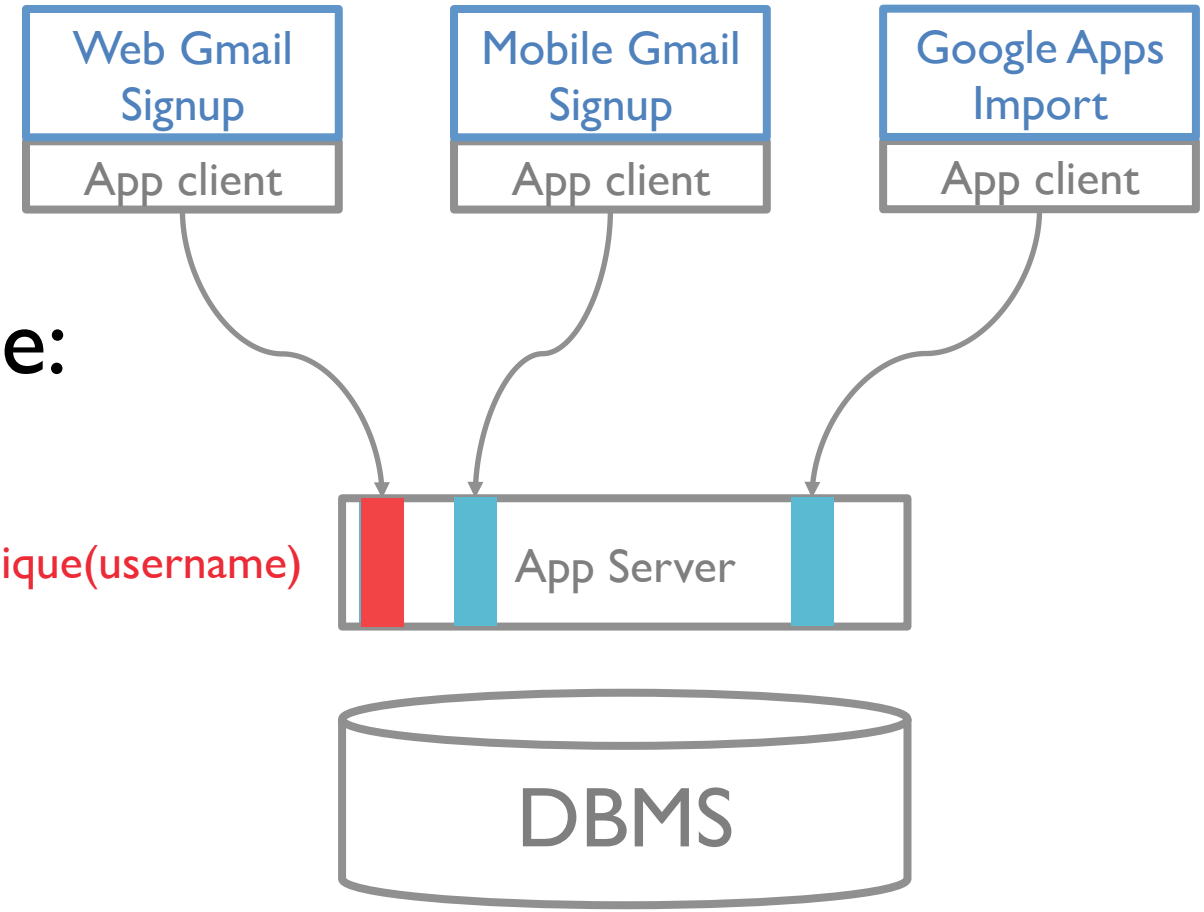


It is Hard to Design Applications



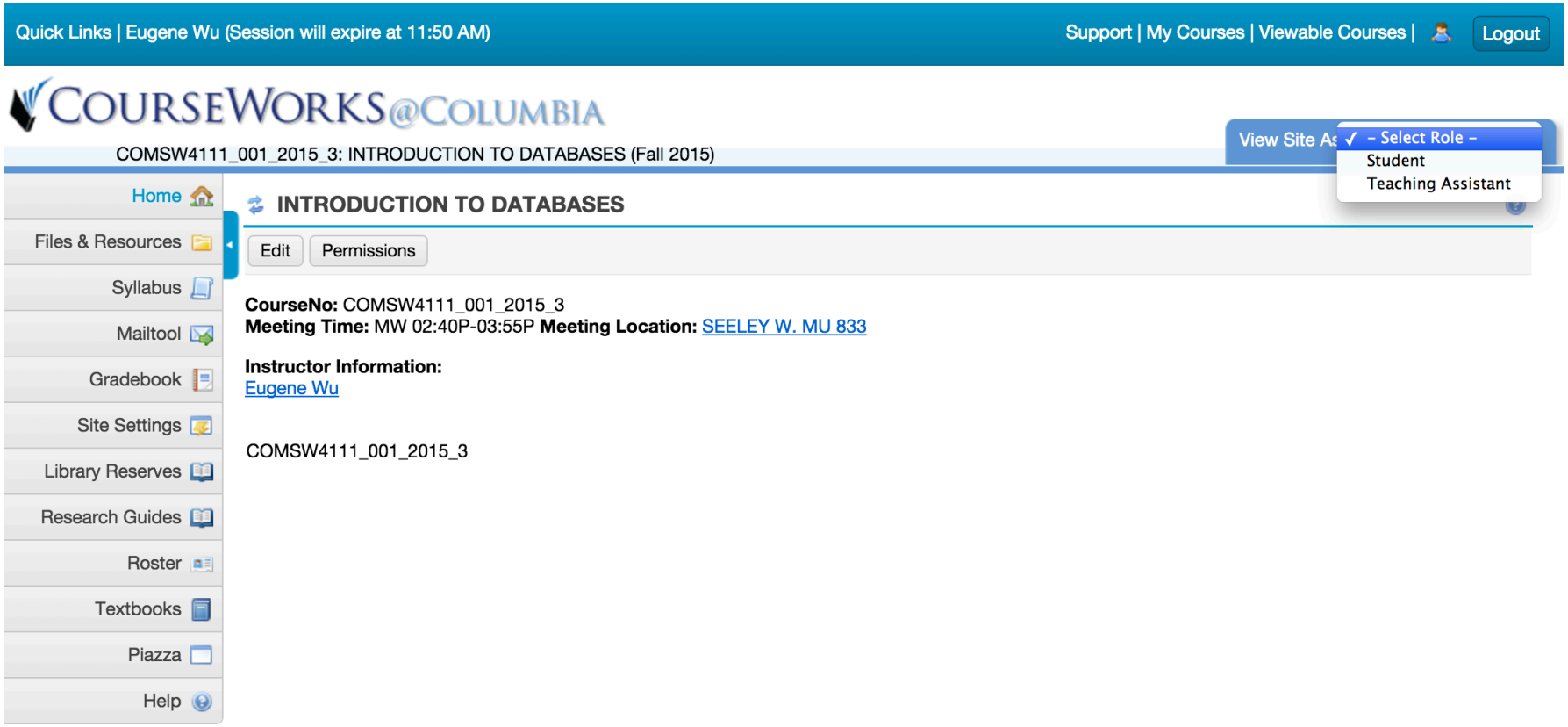
It is Hard to Design Applications

"DRY" principle:
Don't
Repeat
Yourself



Let's make ~~a webapp~~ $$$

live exercise time


Quick Links | Eugene Wu (Session will expire at 11:50 AM)
Support | My Courses | Viewable Courses |
Logout
COURSEWORKS@COLUMBIA
COMSW4111_001_2015_3: INTRODUCTION TO DATABASES (Fall 2015)
View Site As
- Select Role -
Student
Teaching Assistant
Home
Files & Resources
Syllabus
Mailtool
Gradebook
Site Settings
Library Reserves
Research Guides
Roster
Textbooks
Piazza
Help
INTRODUCTION TO DATABASES
Edit
Permissions
CourseNo: COMSW4111_001_2015_3
Meeting Time: MW 02:40P-03:55P Meeting Location: SEELEY W. MU 833
Instructor Information:
Eugene Wu
COMSW4111_001_2015_3

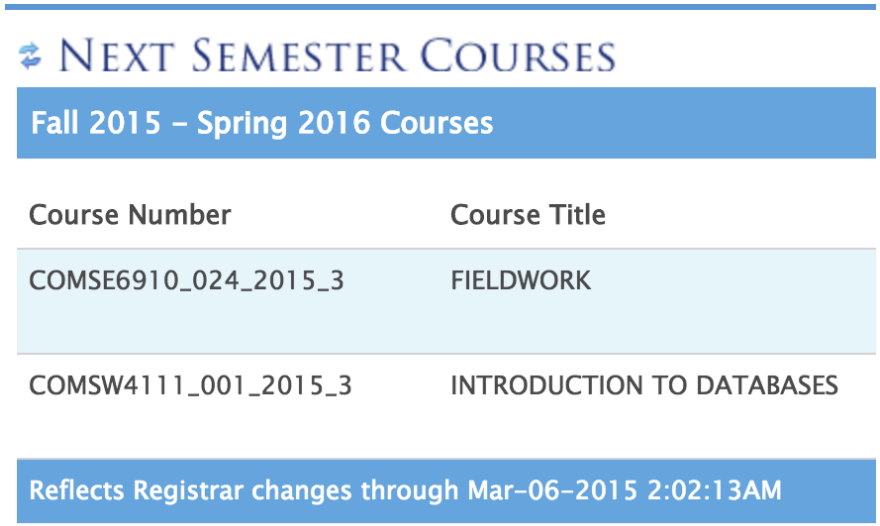
Entity-Relationship Modeling

Entities (objects) to store and their attributes

Relationships between entities and attributes

Integrity constraints & business rules

Visually modeled, easy to turn into DB schema



NEXT SEMESTER COURSES

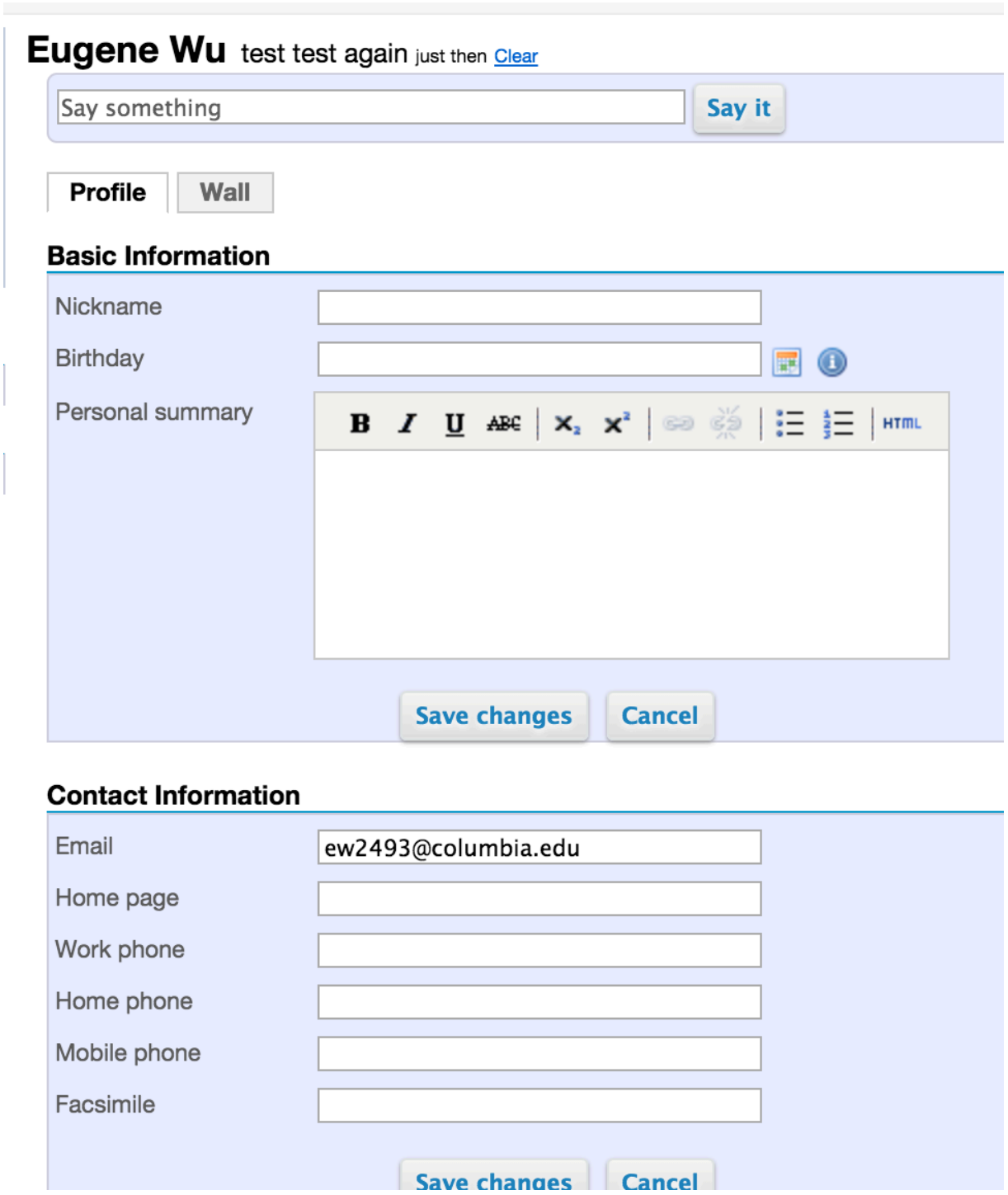
Fall 2015 – Spring 2016 Courses

Course Number	Course Title
COMSE6910_024_2015_3	FIELDWORK
COMSW4111_001_2015_3	INTRODUCTION TO DATABASES

Reflects Registrar changes through Mar-06-2015 2:02:13AM

Courses

- Course Number
- Course Title
- Year
- Semester



Users

- Nickname
- Name
- Birthday
- Summary
- Email
- …

Basics: Entities

Entity e.g., intro to databases

real-world object distinguishable from other objects

described as set of attributes & the values

(think one record)

Entity Set e.g., all courses

collection of similar entities

all entities have same attributes (except Is-A, later)

must have one or more keys

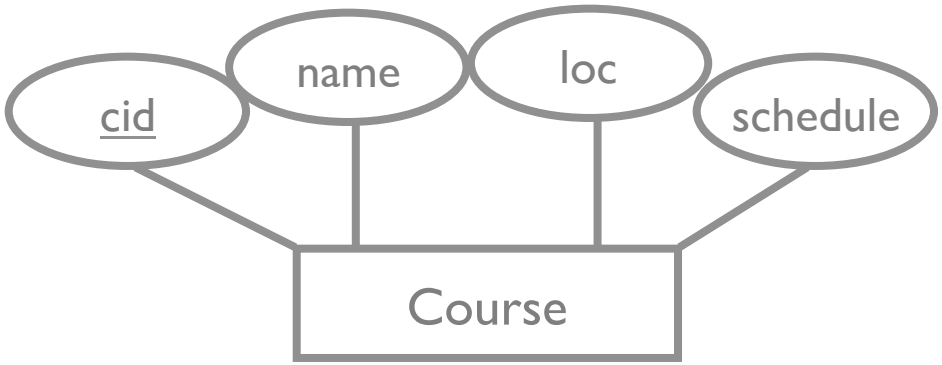
≈ table

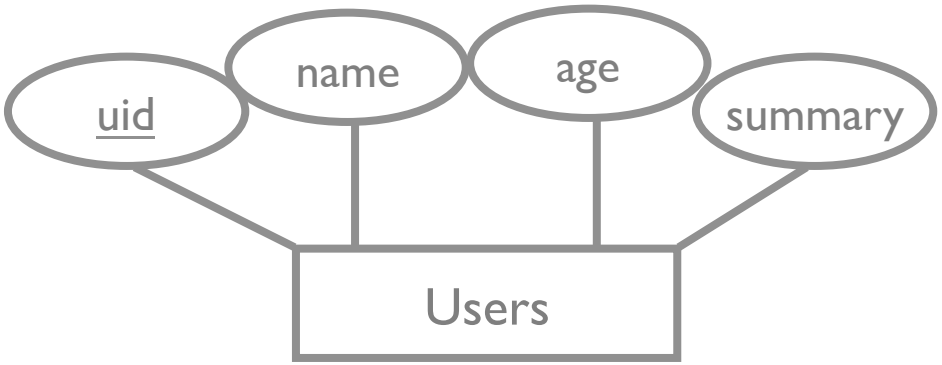
Example: Entity

Keys (cid, uid) are underlined

- Values must be unique

(think: can use as hash table key to find value)





Basics: Relationships

Relationship: association between 2 or more entities

e.g., alice is taking Introduction to DBs

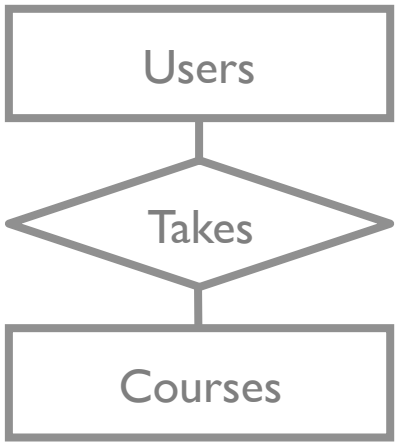
Relationship Set: collection of similar relationships

N-ary relationship set R relates N entity sets $E_1 \ldots E_n$

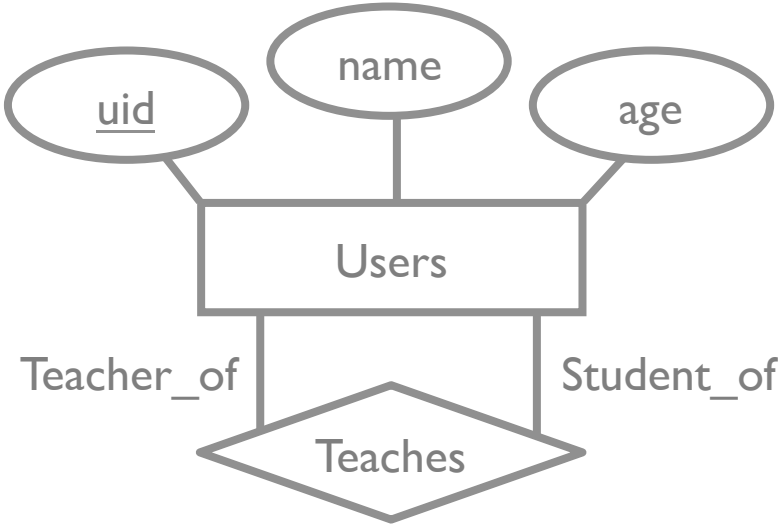
Each $r \in R$ involves entities $e_1 \ldots e_n$

An E_i can be part of multiple relationship sets or multiple roles in same set

Basics: Relationships



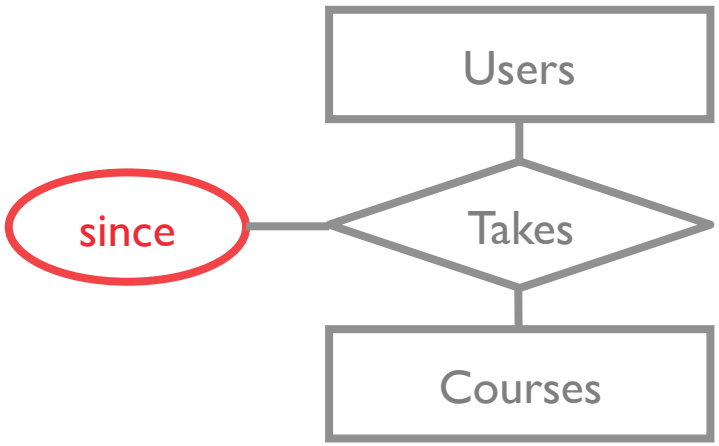
Users takes different roles
in same relationships set



Basics: Relationships

Relationships sets can have descriptive attributes

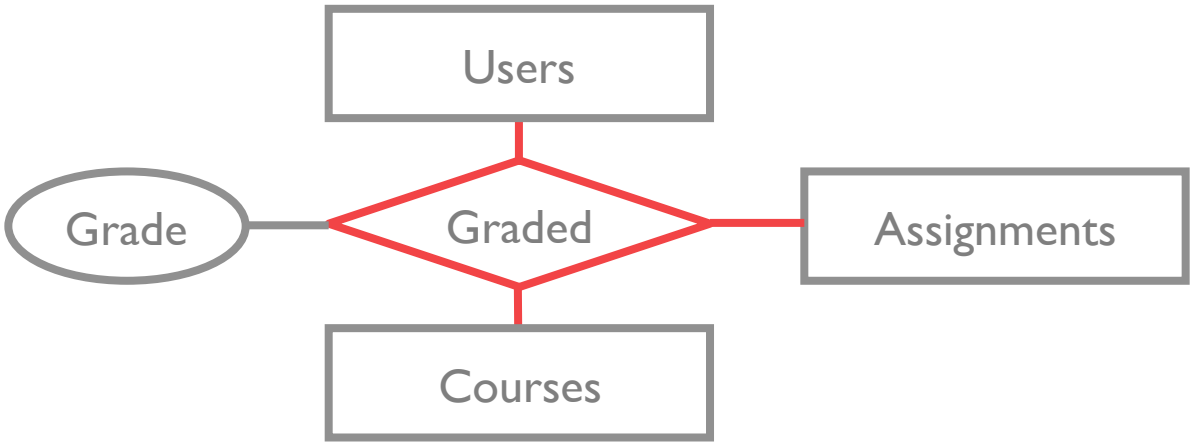
e.g., the *since* attribute of Instructs



Basics: Ternary Relationships

Connects three entities

N-ary relationships possible too.



Constraints

Help avoid corruption, inconsistencies

- Key constraints
- Participation constraints
- Weak entities
- Overlap and covering constraints

Key Constraints

Defines cardinality requirements on relationships

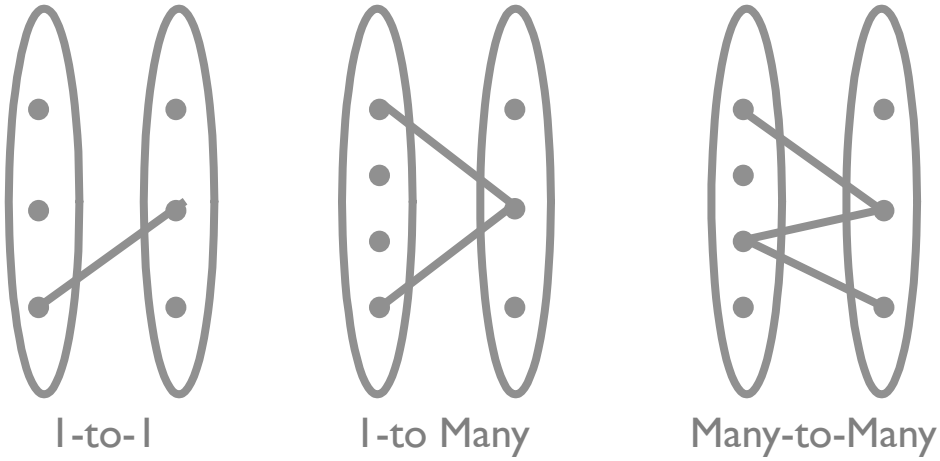
Many to many e.g., consider *Takes*

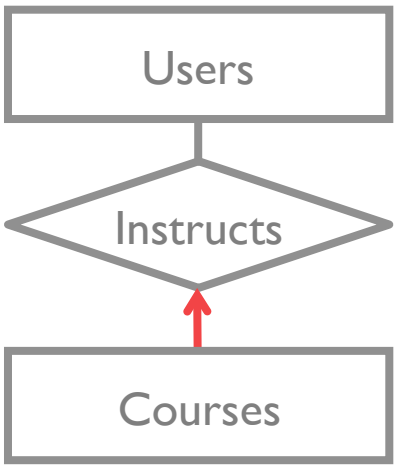
a user can take many courses

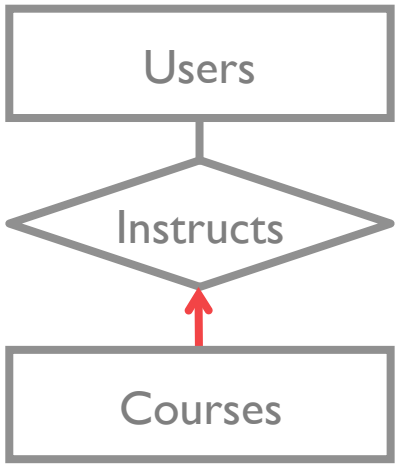
a course can have many users that take the course

One to Many e.g., consider *Instructs*

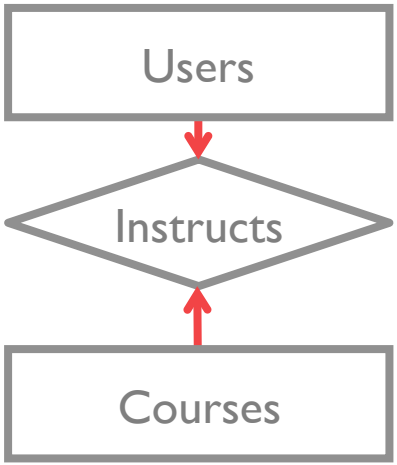
a course has at most one instructor



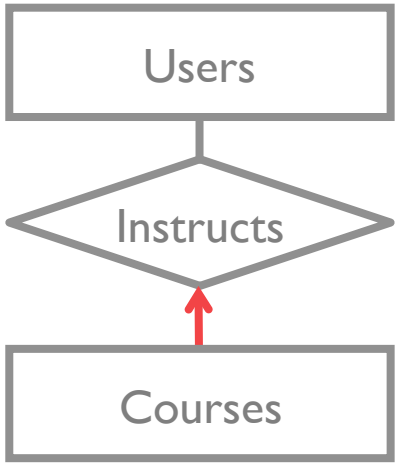




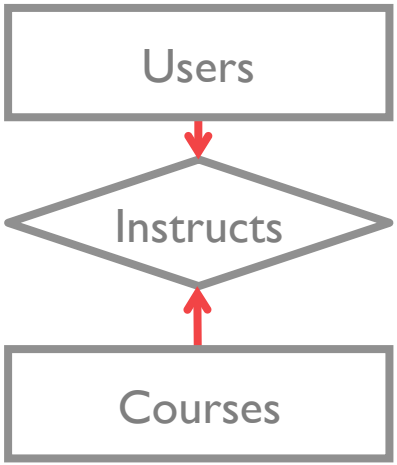
A course has at most one instructor



???



A course has at most one instructor



One-to-one:
A course has at most one instructor AND users can instruct at most one course

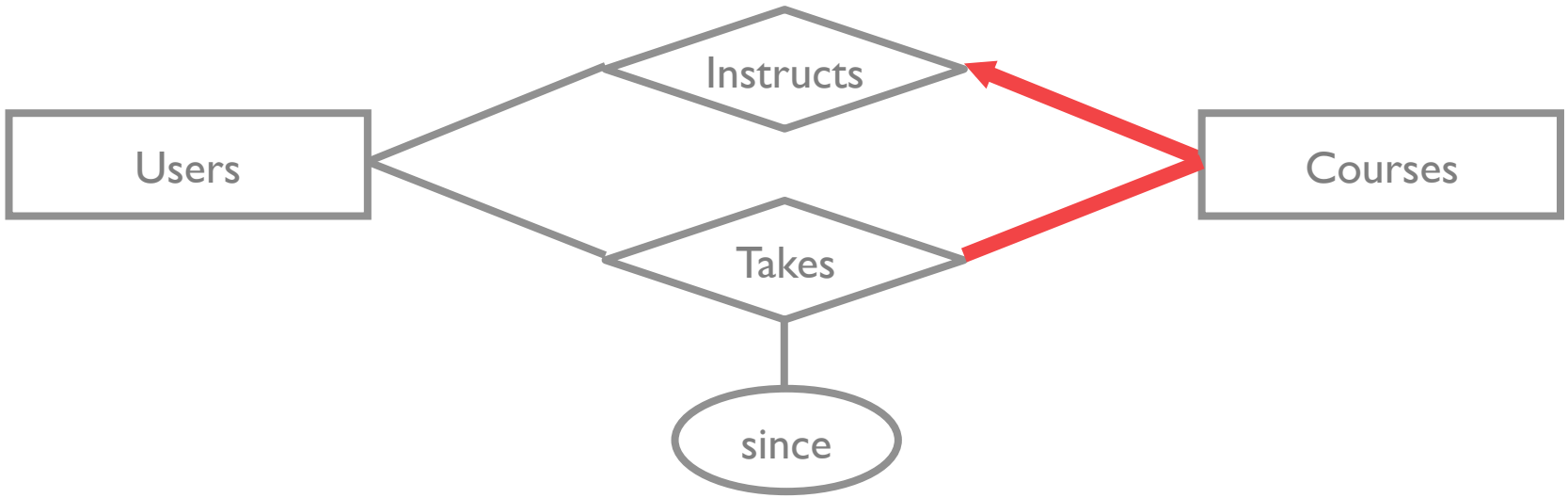
Participation Constraints

Does every course need an instructor?

If yes, it's a participation constraint (thick line)

e.g., participation of Courses in instructs is *Total*

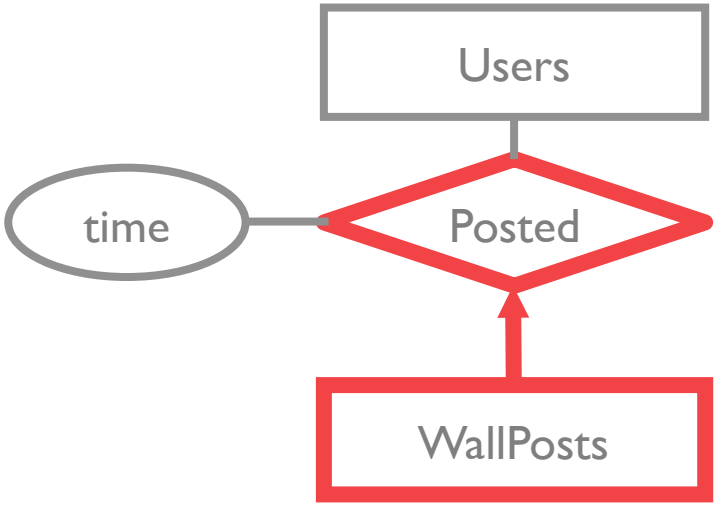
Otherwise, *partial* participation constraint

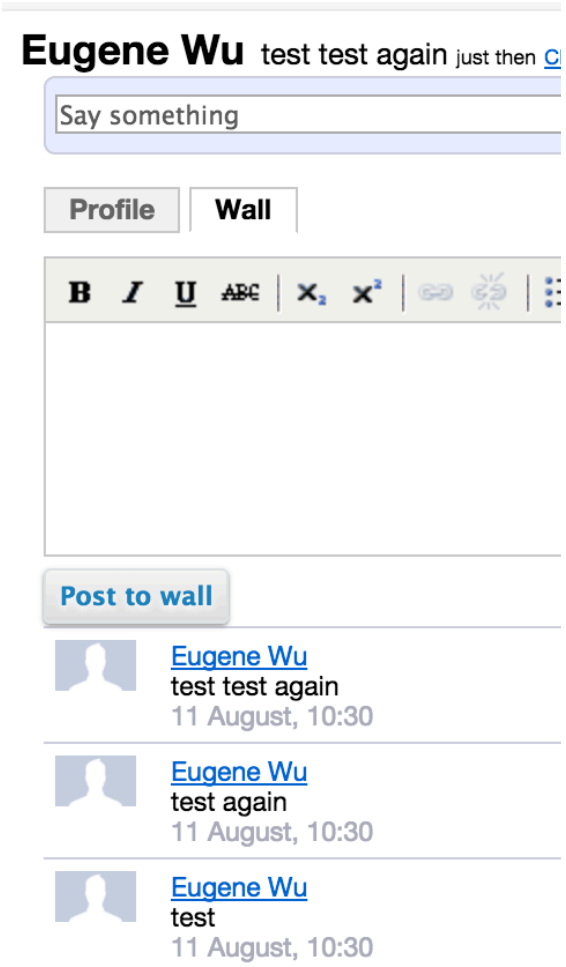


Weak Entities

A *weak entity* can only be uniquely identified by using the primary key of its owner entity

- Owner and weak entity sets must be in one to many relationship set
- Weak entity set must have total participation in this *identifying* relationships set

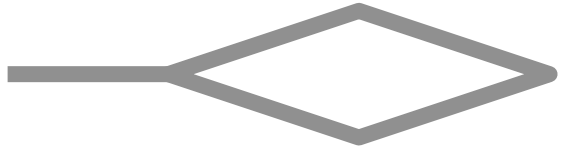




At most one

At least one

Exactly one



Weak Entity

5 min exercise (aka break)

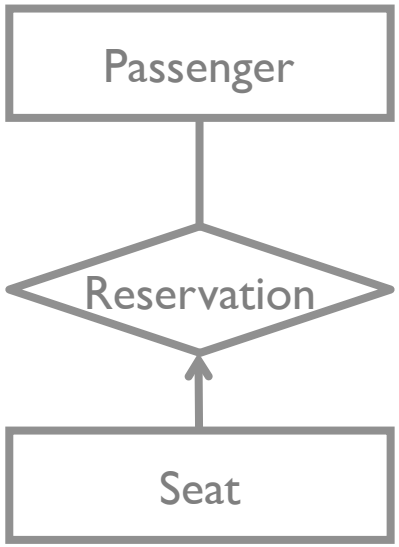
Diagram an airplane reservation with the following entities:

Passenger

Seat

What other entities might exist?

Possible solution



Seat: At most 1 reservation (no double booking)

Passenger: Optional: at least one reservation (thick line)

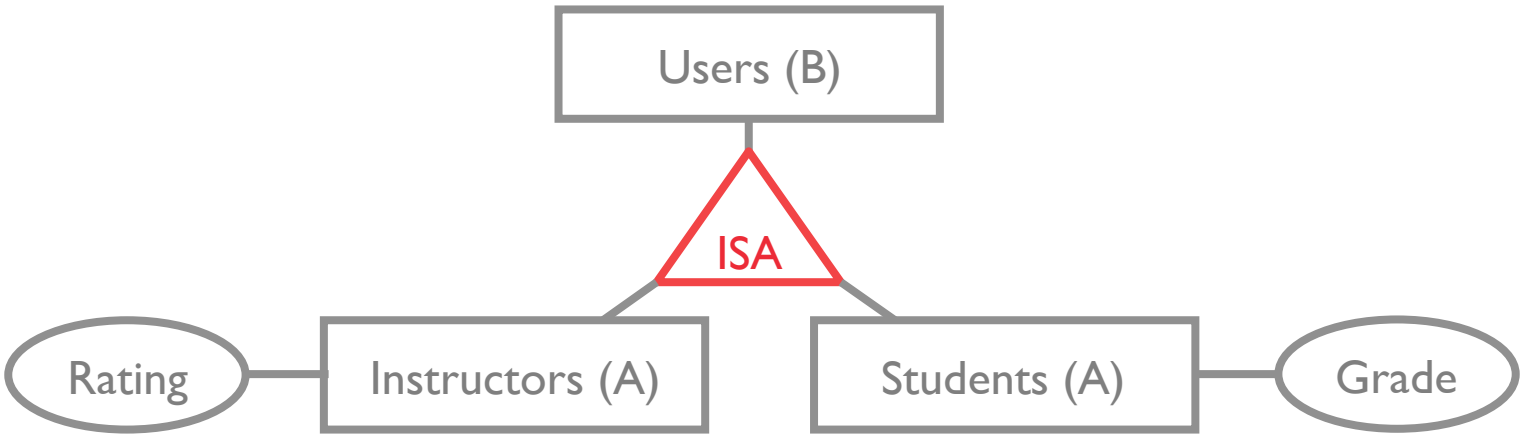
ISA (is a) Hierarchies

Inheritance rules similar to programming languages

- A ISA B → every A also considered a B
- When querying for Bs, must consider As

Why use ISA?

- add descriptive attributes specific to a subclass e.g., grade
- identify entities that participate in a relationship



ISA (is a) Hierarchies

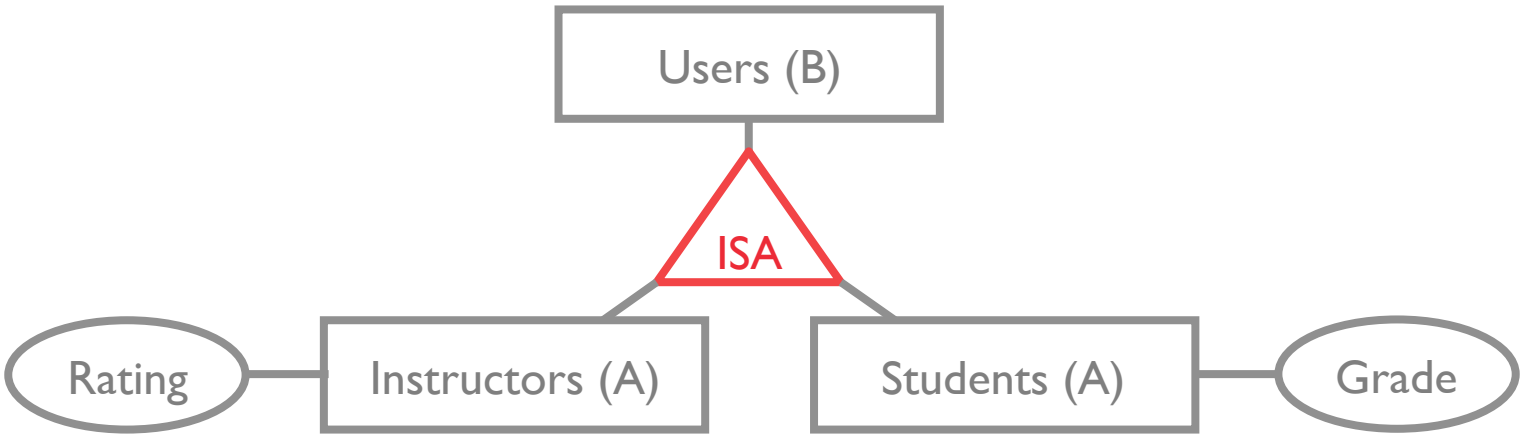
Overlap Constraint

can eugene be an instructor and a student? (allow/disallow)

Covering Constraint

must every user be an instructor or student? (yes/no)

HOW DO WE EXPRESS THESE IN AN ER DIAGRAM???????

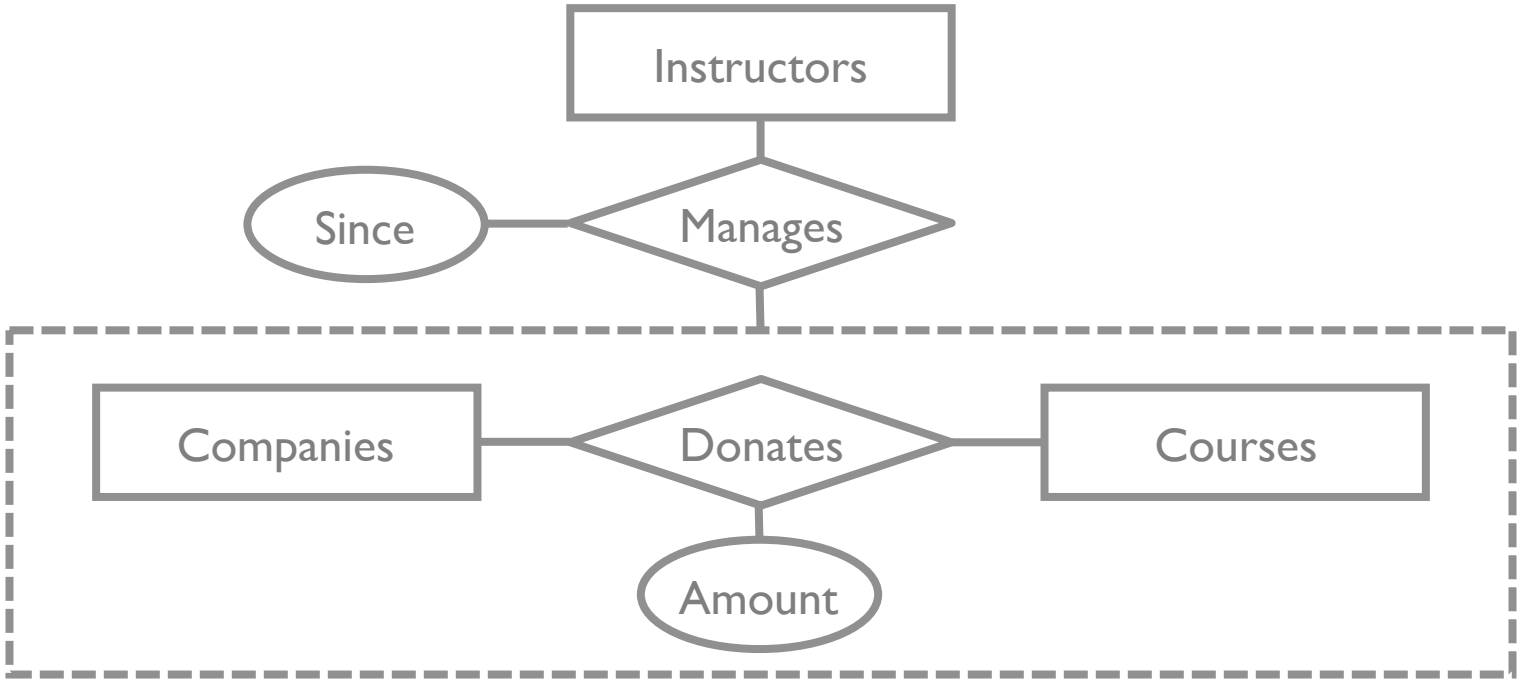


Aggregation

Relationships between (entities – relationships)

Lets us treat a Relationship Set like an Entity Set

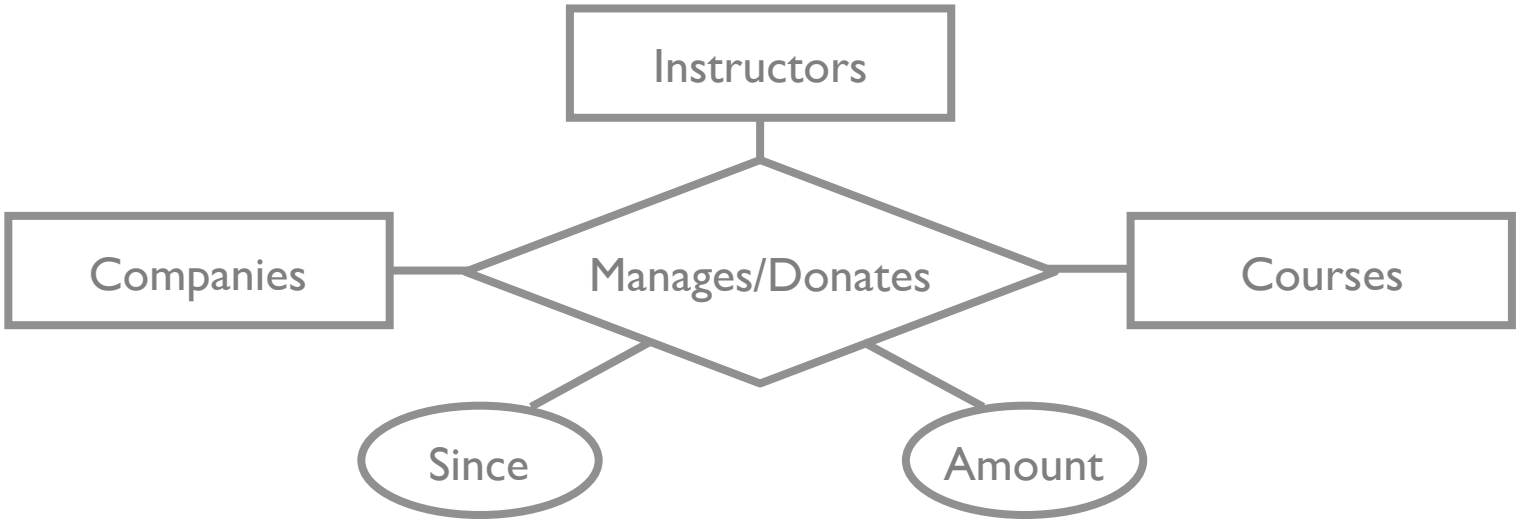
so it can participate in other relationships



Aggregation vs Ternary Relationships

Why use aggregation?

- Manages and Donates are distinct relationships with own attrs
- Can define constraints on relationship sets
  - e.g., a donation can be managed by at most one instructor



Aggregation vs Ternary Relationships

Constraints apply to all connected entity sets