

# Lecture 2

## Entity-Relationship Model

Eugene Wu

# Steps for a New Application

## Requirements

what are you going to build?

## Conceptual Database Design

pen-and-pencil description

## Logical Design

formal database schema

## Schema Refinement:

fix potential problems, normalization

## Physical Database Design

use sample of queries to optimize for speed/storage

## App/Security Design

prevent security problems

# Steps for a New Application

## Requirements

what are you going to build?

## Conceptual Database Design

pen-and-pencil 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

## App/Security Design

prevent security problems

# Database Apps Are Complicated

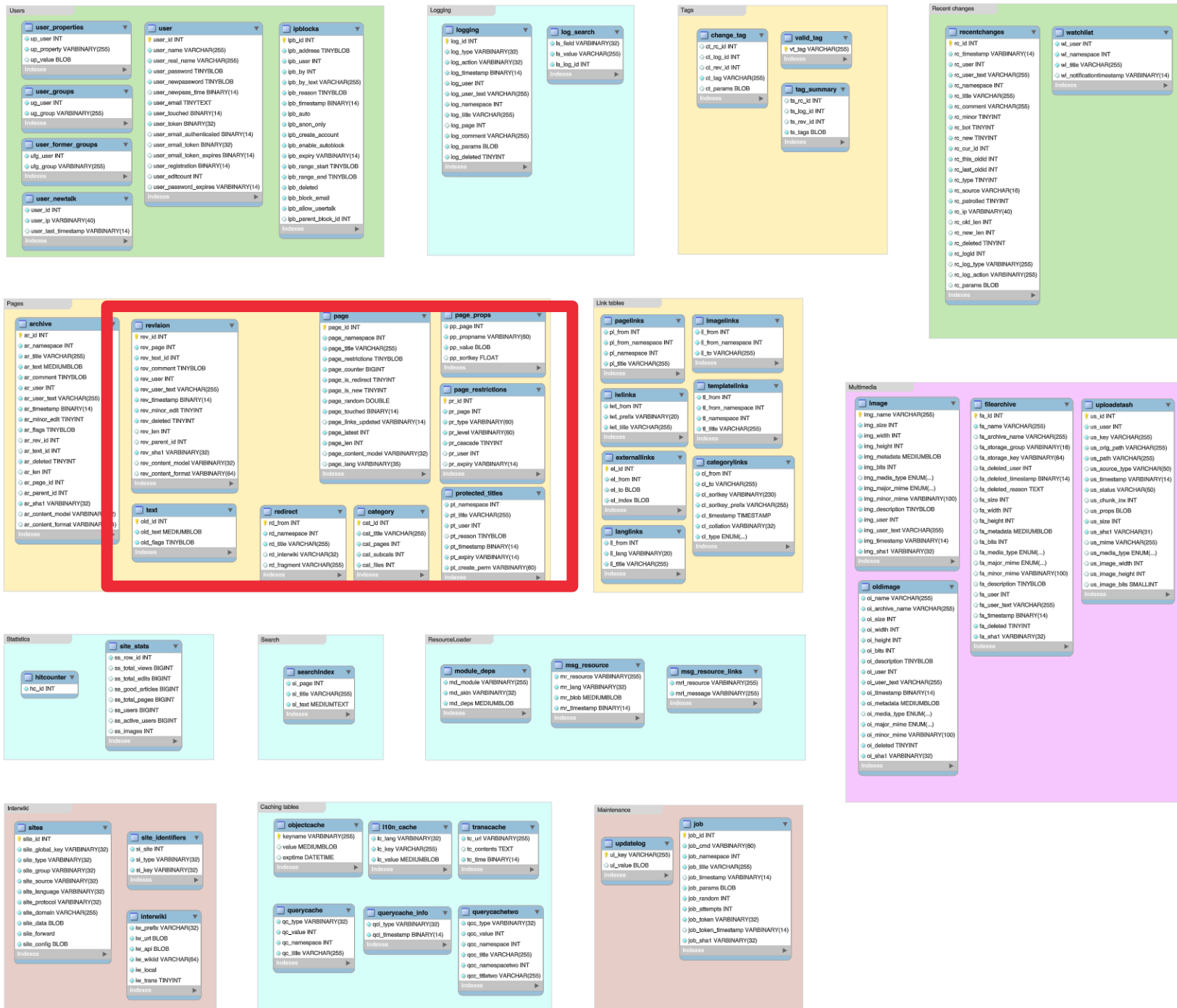
## Typical Fortune 100 Company

- ~10k different information (data) systems

- 90% relational databases (DBMSes)

- Typical database has >100 tables

- Typical table has 50 – 200 attributes



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

page_props
pp_page INT
pp_propname VARBINARY(60)
pp_value BLOB
pp_sortkey FLOAT
Indexes

page_restrictions
pr_id INT
pr_page INT
pr_type VARBINARY(60)
pr_level VARBINARY(60)
pr_cascade TINYINT
pr_user INT
pr_expiry VARBINARY(14)
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

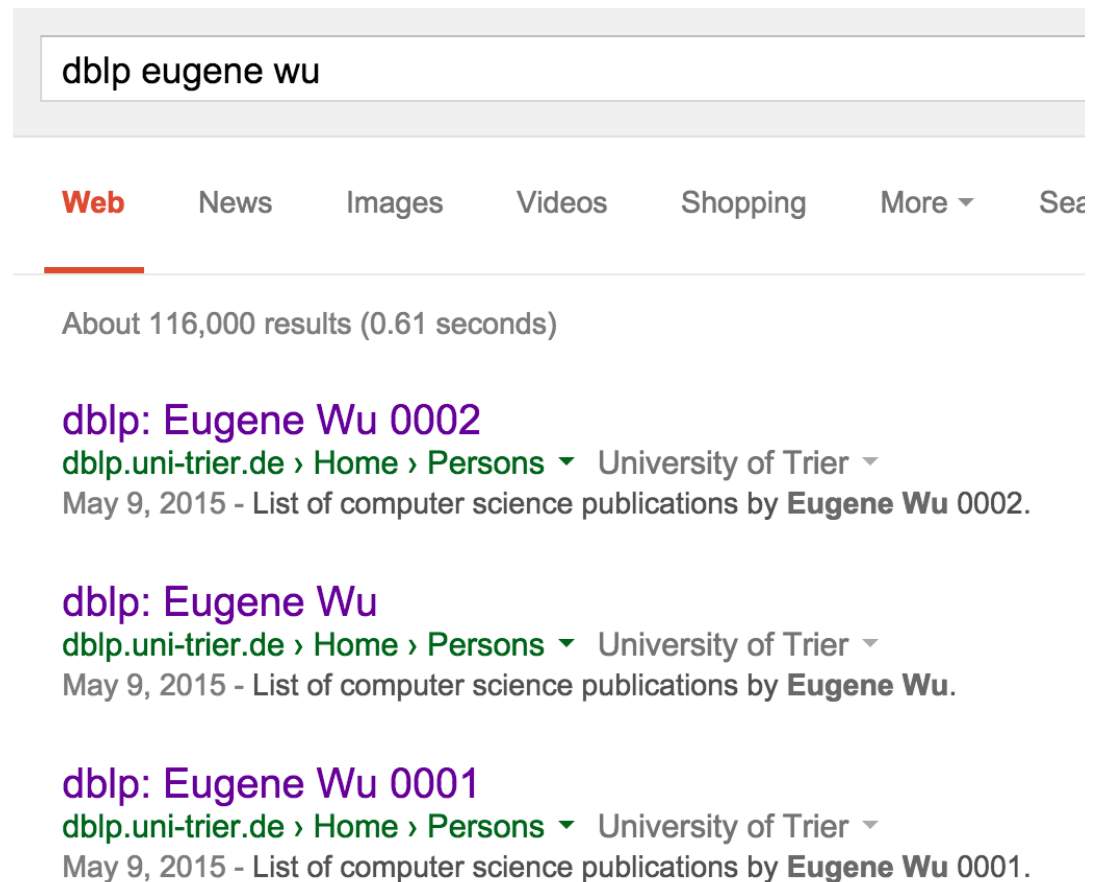
category
cat_id INT
cat_title VARCHAR(255)
cat_pages INT
cat_subcats INT
cat_files INT
Indexes

protected_titles
pt_namespace INT
pt_title VARCHAR(255)
pt_user INT
pt_reason TINYBLOB
pt_timestamp BINARY(14)
pt_expiry VARBINARY(14)
pt_create_perm VARBINARY(64)
Indexes

# Inconsistencies/Constraint Violations

*Huge* amount of effort to avoid inconsistencies

DBLP is *the* site for  
computer science  
publications



The screenshot shows a search interface with a search bar containing 'dblp eugene wu'. Below the search bar are navigation tabs: 'Web' (highlighted with a red underline), 'News', 'Images', 'Videos', 'Shopping', 'More', and 'Search'. Below the tabs, it says 'About 116,000 results (0.61 seconds)'. There are three search results listed, each with a purple title, a green breadcrumb trail, and a grey description.

dblp eugene wu

**Web** News Images Videos Shopping More Search

About 116,000 results (0.61 seconds)

**dblp: Eugene Wu 0002**  
[dblp.uni-trier.de](#) › [Home](#) › [Persons](#) › University of Trier  
May 9, 2015 - List of computer science publications by **Eugene Wu** 0002.

**dblp: Eugene Wu**  
[dblp.uni-trier.de](#) › [Home](#) › [Persons](#) › University of Trier  
May 9, 2015 - List of computer science publications by **Eugene Wu**.







**dblp: Eugene Wu 0001**  
[dblp.uni-trier.de](#) › [Home](#) › [Persons](#) › University of Trier  
May 9, 2015 - List of computer science publications by **Eugene Wu** 0001.

# 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



**Name**

First

Last

**Choose your username**

eugenewu

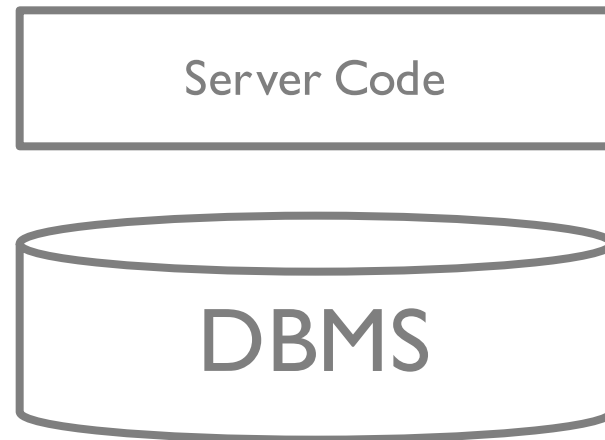
@gmail.com

Someone already has that username. Try another?

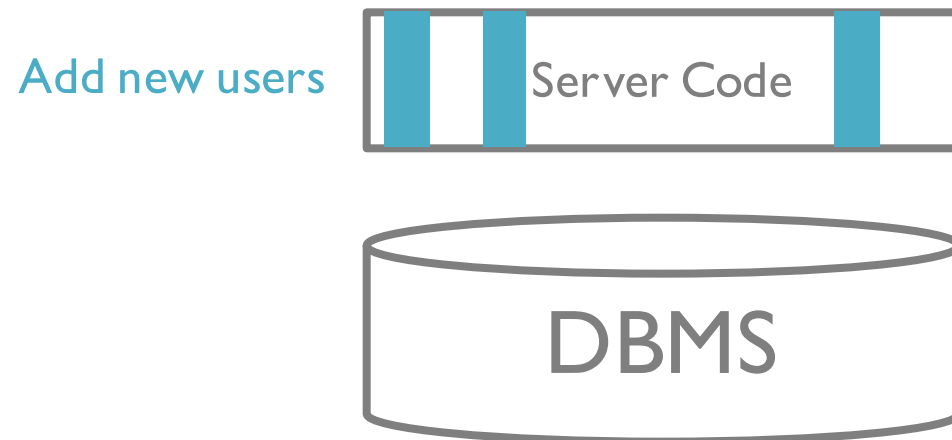
Available: eugenewu861

**Create a password**

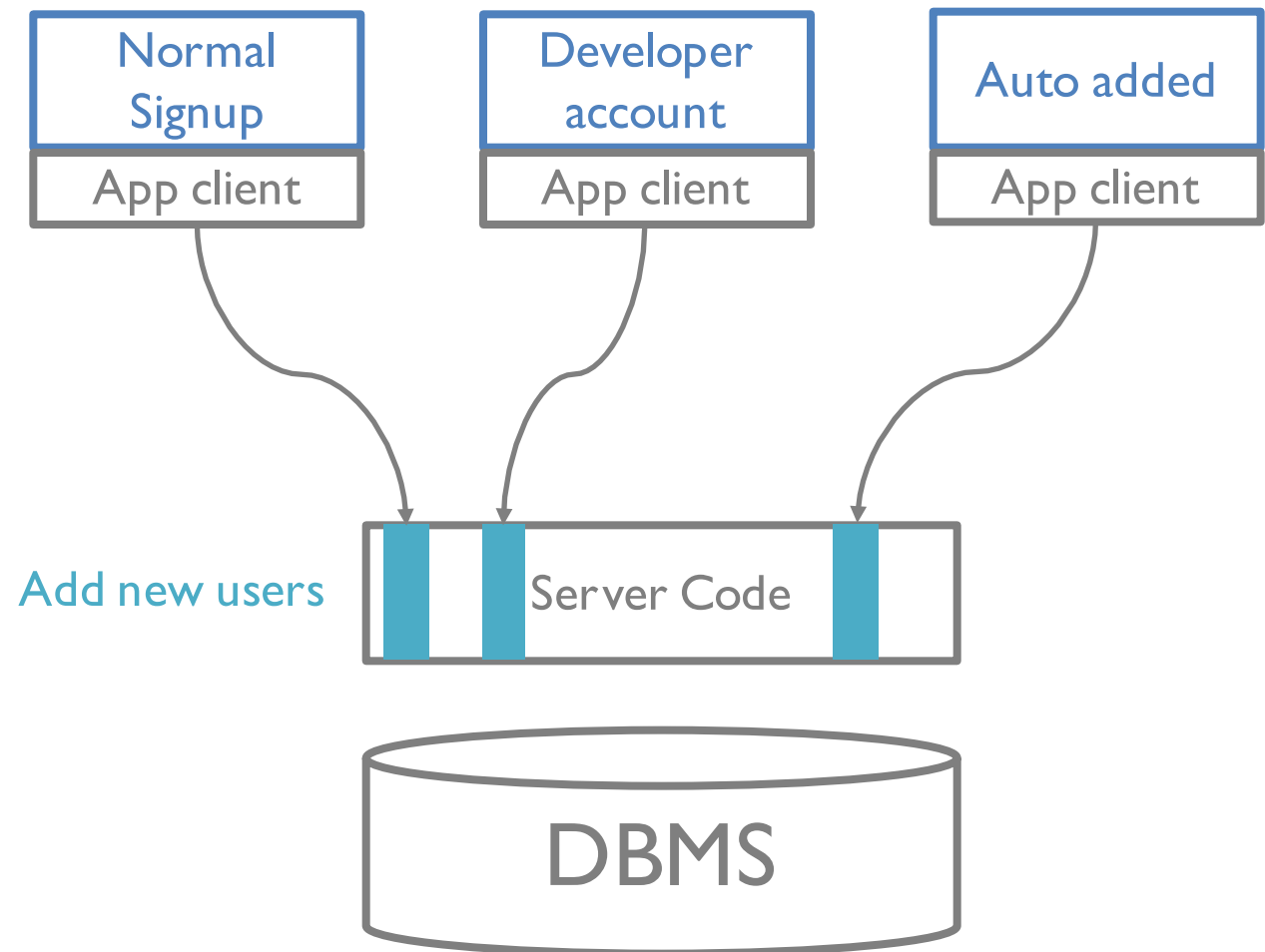
# It is Hard to Design Applications



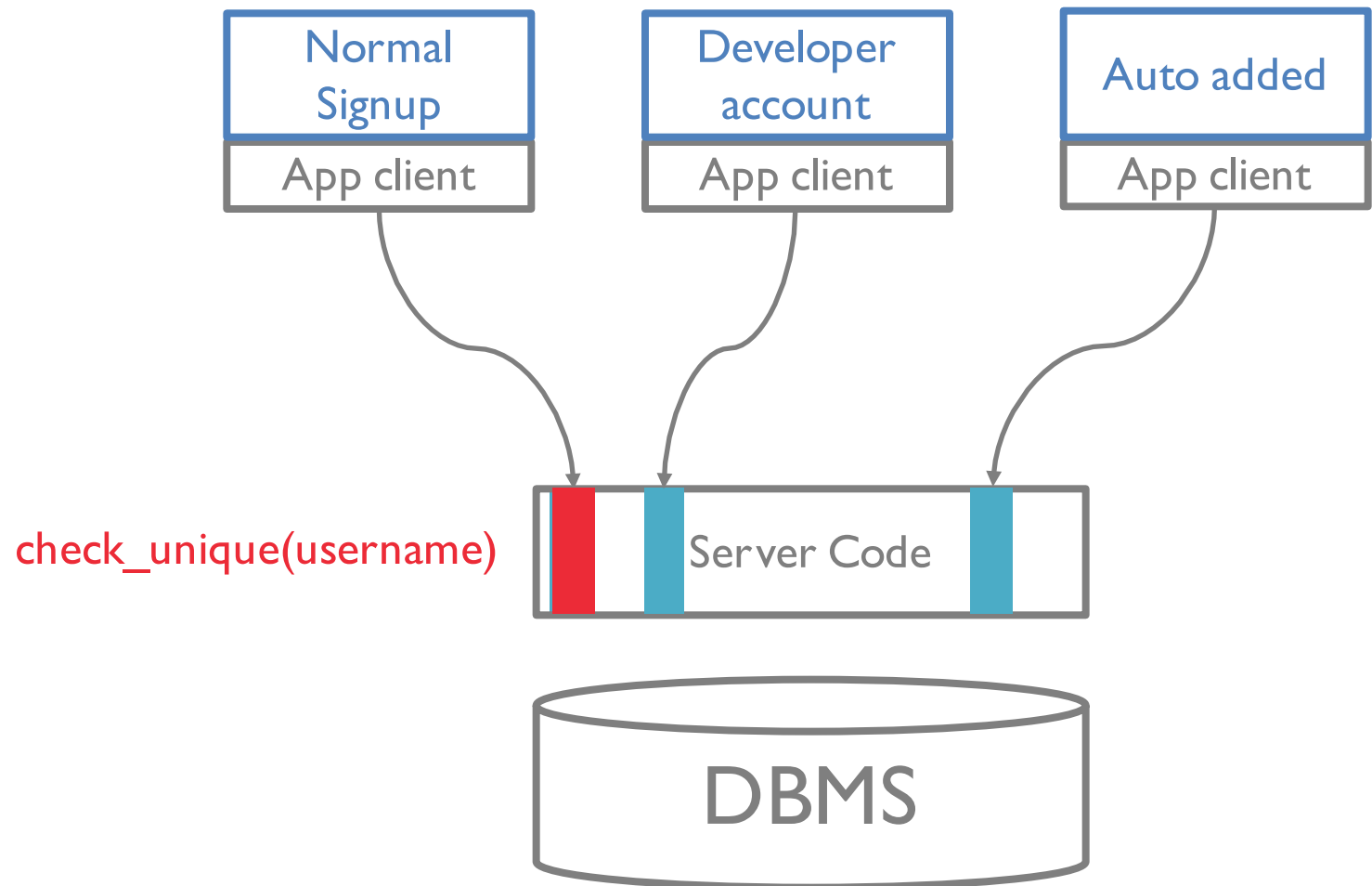
# It is Hard to Design Applications



# It is Hard to Design Applications



# It is Hard to Design Applications



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

live exercise time




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 their attrs.

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



**Eugene Wu** test test again just then [Clear](#)

### Basic Information

Nickname

Birthday



Personal summary

**B** *I* U ~~ABC~~ |  $x_2$   $x^2$  | | | [HTML](#)

### Contact Information

Email

Home page

Work phone

Home phone

Mobile phone

Facsimile

## 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

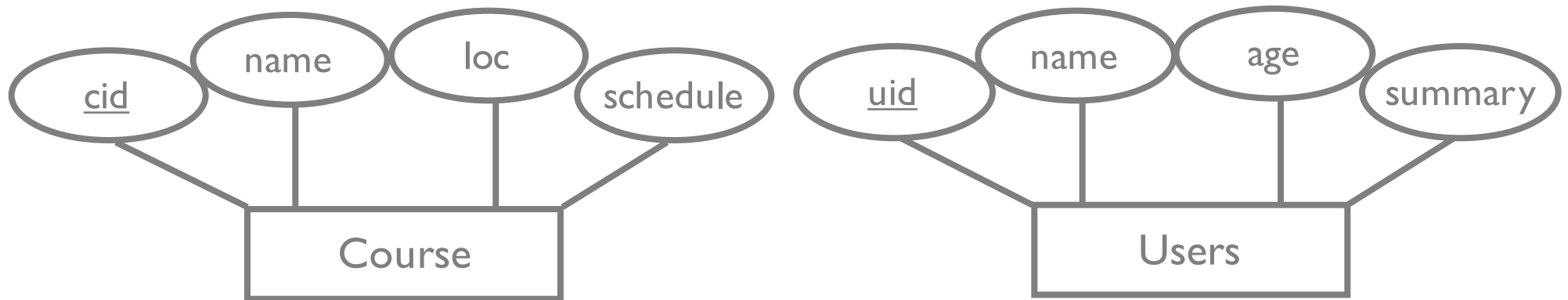
collection of similar entities  
all entities have same attributes (unless Is-A)  
must have one or more keys  
attributes have domains  
≈ table

# Example: Entity

Keys (cid, uid) are underlined

Values must be unique

(think: can use as hashtable key to lookup in table)



# Basics: Relationships

Relationship: association between 2 or more entities

e.g., alice **is taking** Introduction to DBs

Relationship Set: collection of similar relationships

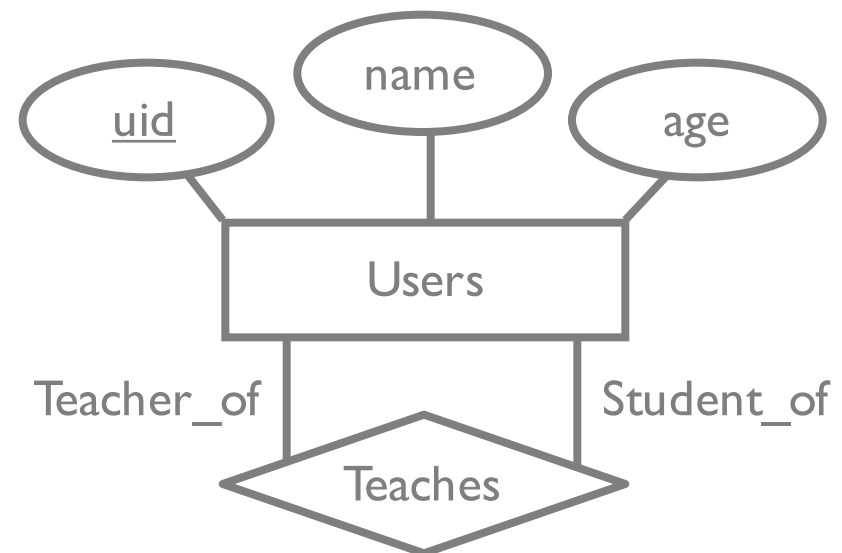
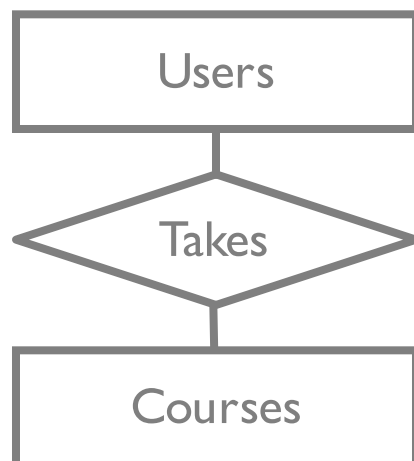
N-ary relationship set  $R$  relates  $N$  entity sets  $E_1 \dots E_n$

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

An  $E_i$  can be part of diff. relationship sets or diff. roles in same set

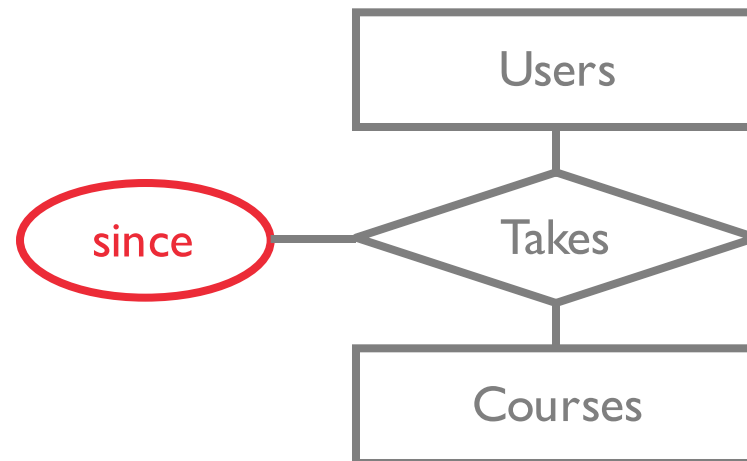
# Basics: Relationships

Users takes diff roles in  
same relationships set



# Basics: Relationships

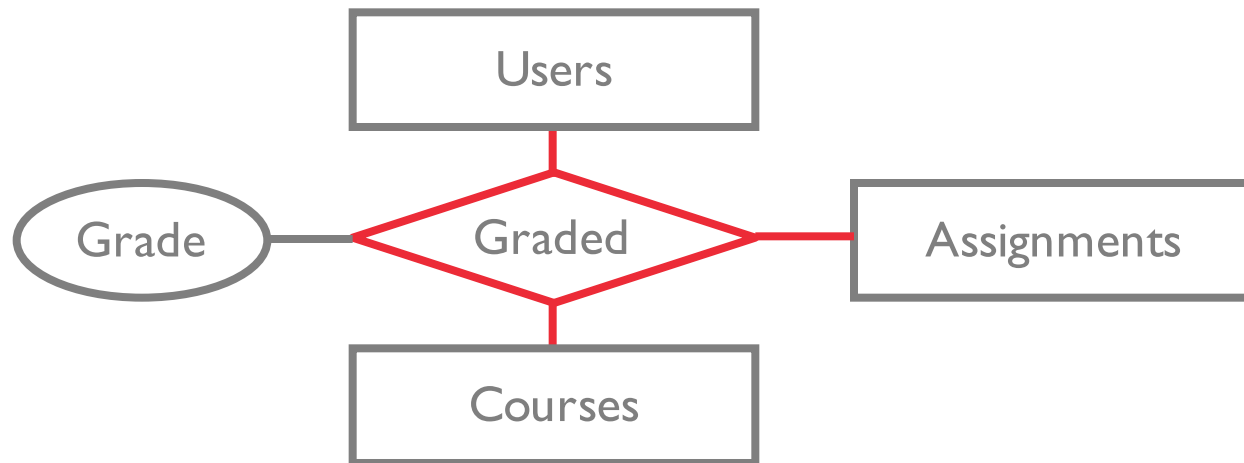
Relationships sets can have descriptive attributes  
e.g., the *since* attribute of Instructs



# 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

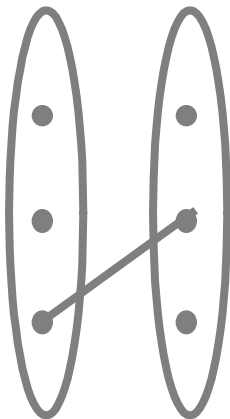
**Many to many** e.g., consider *Takes*

a user can take many courses

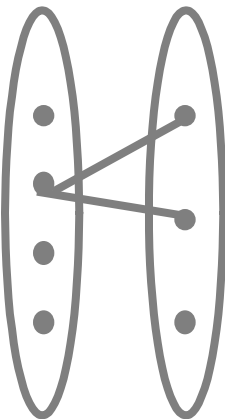
a course can have many users that take the course

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

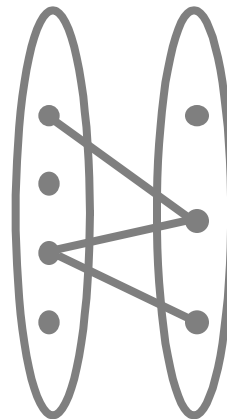
a course has at most one instructor



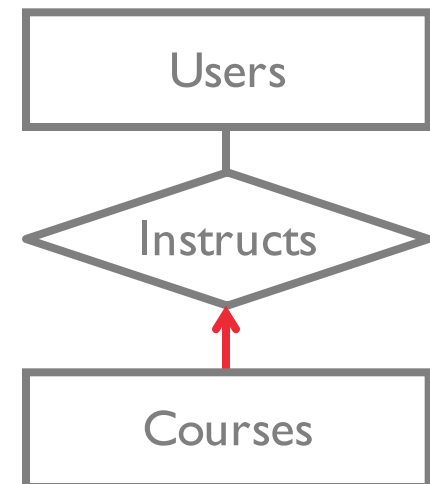
1-to-1

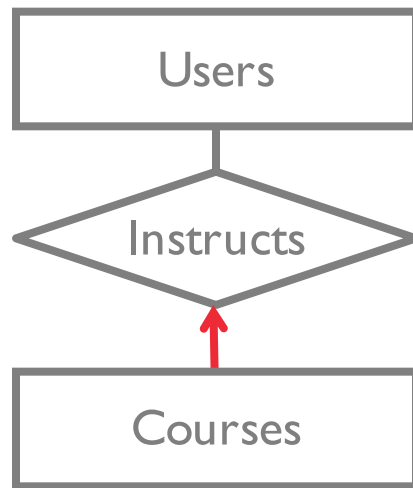


1-to Many

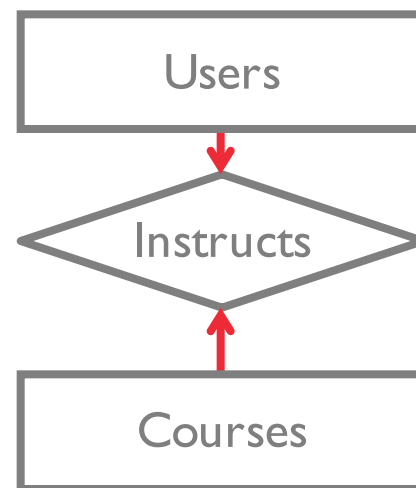


Many-to-Many





A course has at most  
one instructor



???

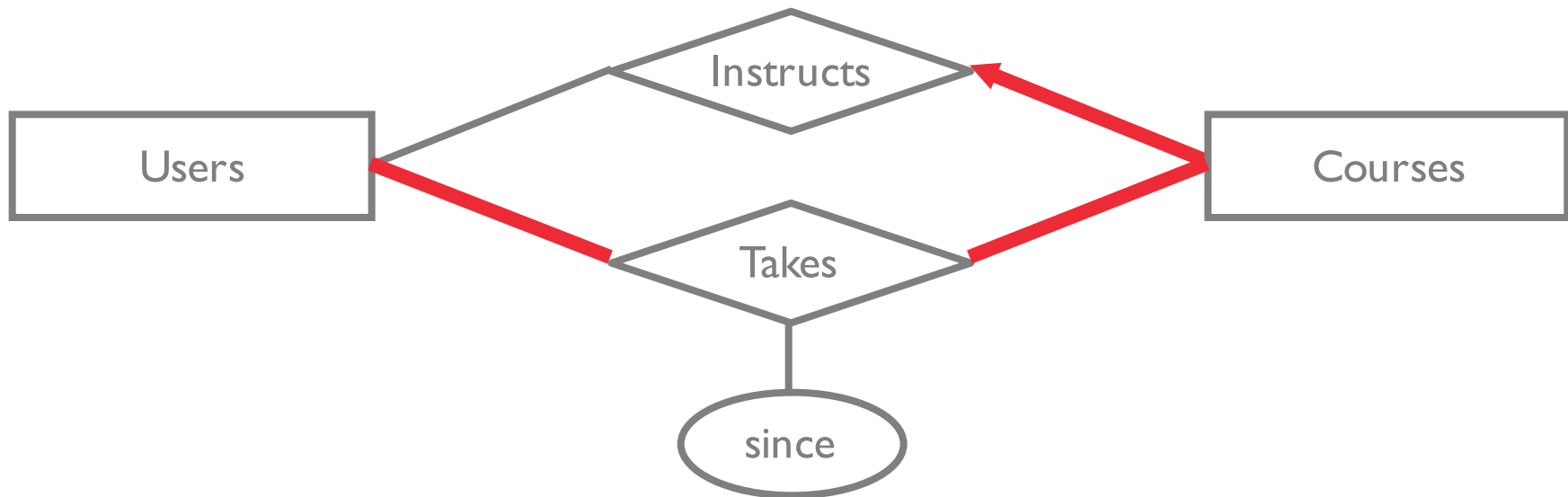
# Participation Constraints

Does every course need an instructor?

If yes, it's a **participation constraint**

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

Otherwise, *partial* participation constraint

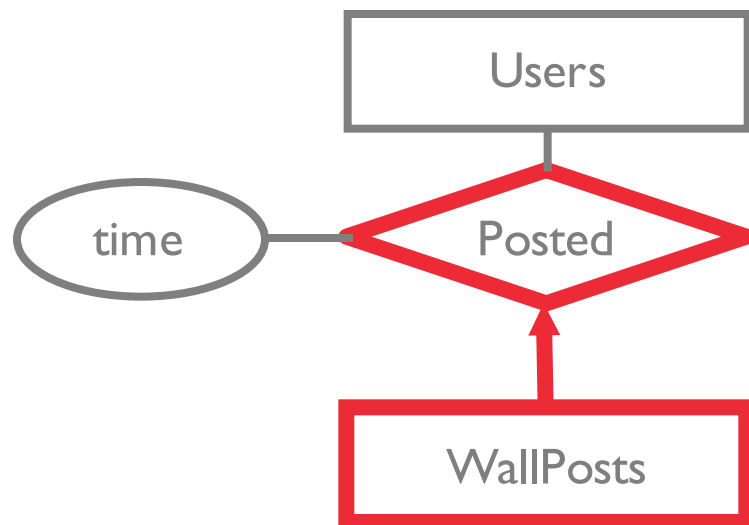


# Weak Entities

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



**Eugene Wu** test test again just then [c](#)

Profile

Wall

**B** *I* U ABC | x<sub>2</sub> x<sup>2</sup> |

Post to wall



[Eugene Wu](#)

test test again

11 August, 10:30



[Eugene Wu](#)

test again

11 August, 10:30



[Eugene Wu](#)

test

11 August, 10:30



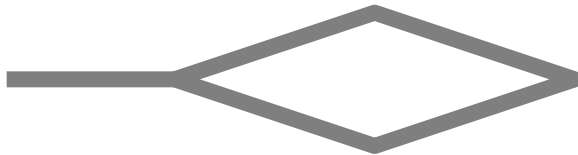
At most one



At least one



Exactly one



Weak Entity

# ISA (is a) Hierarchies

Inheritance rules similar to programming languages

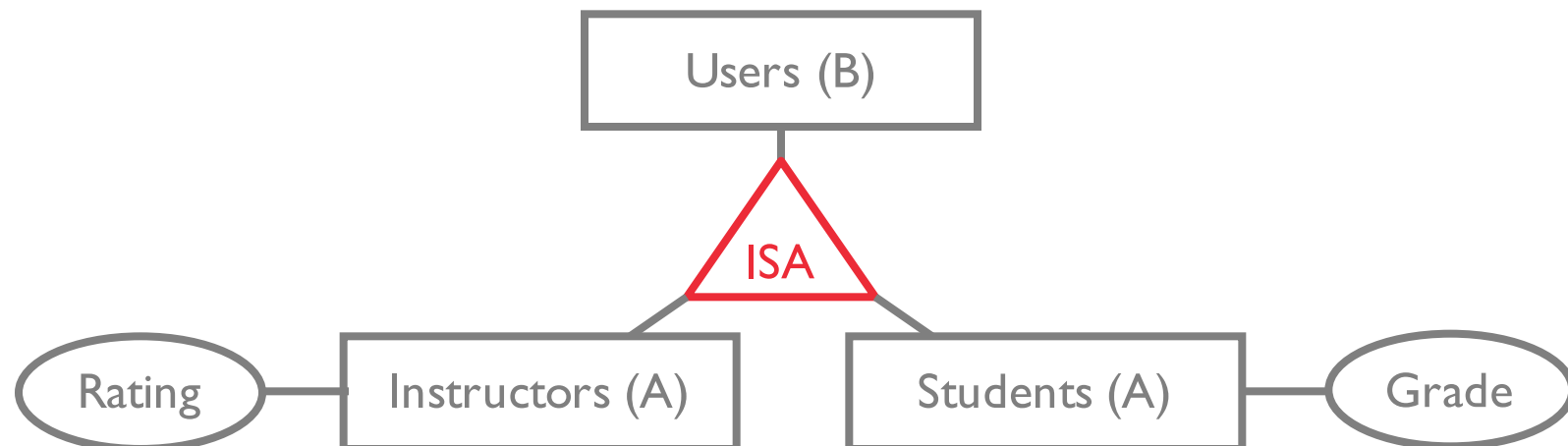
$A \text{ ISA } B \rightarrow$  every  $A$  also considered a  $B$

When querying for  $B$ s, must consider  $A$ s (unlike e.g., C++)

Why use ISA?

add descriptive attributes specific to a subclass e.g., grade

identify entities that participate in a relationship



# ISA (is a) Hierarchies

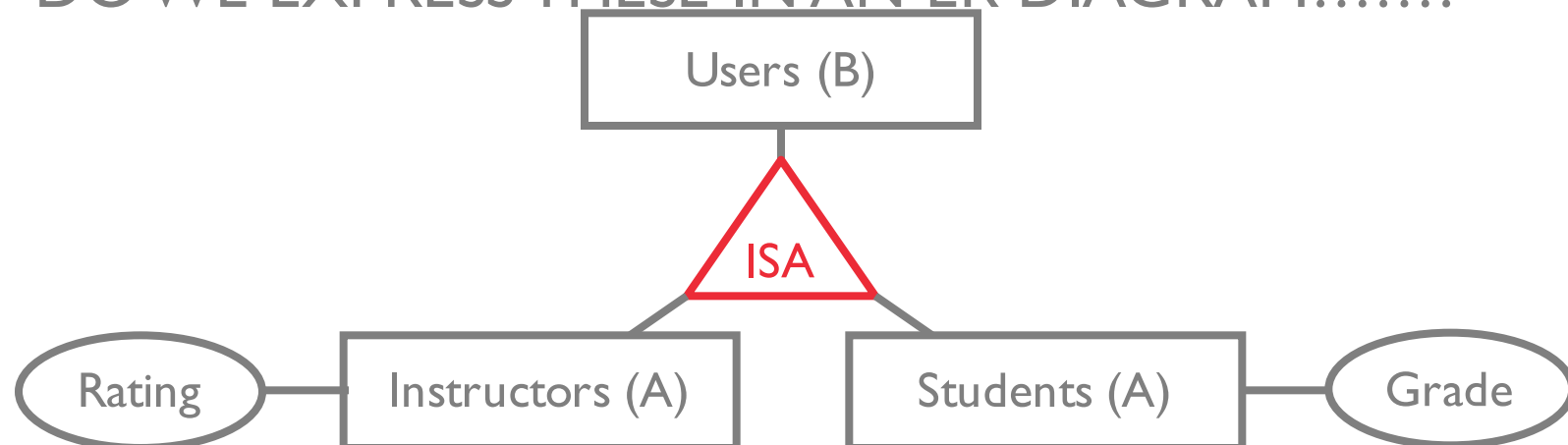
## Overlap Constraint

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

## Covering Constraint

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

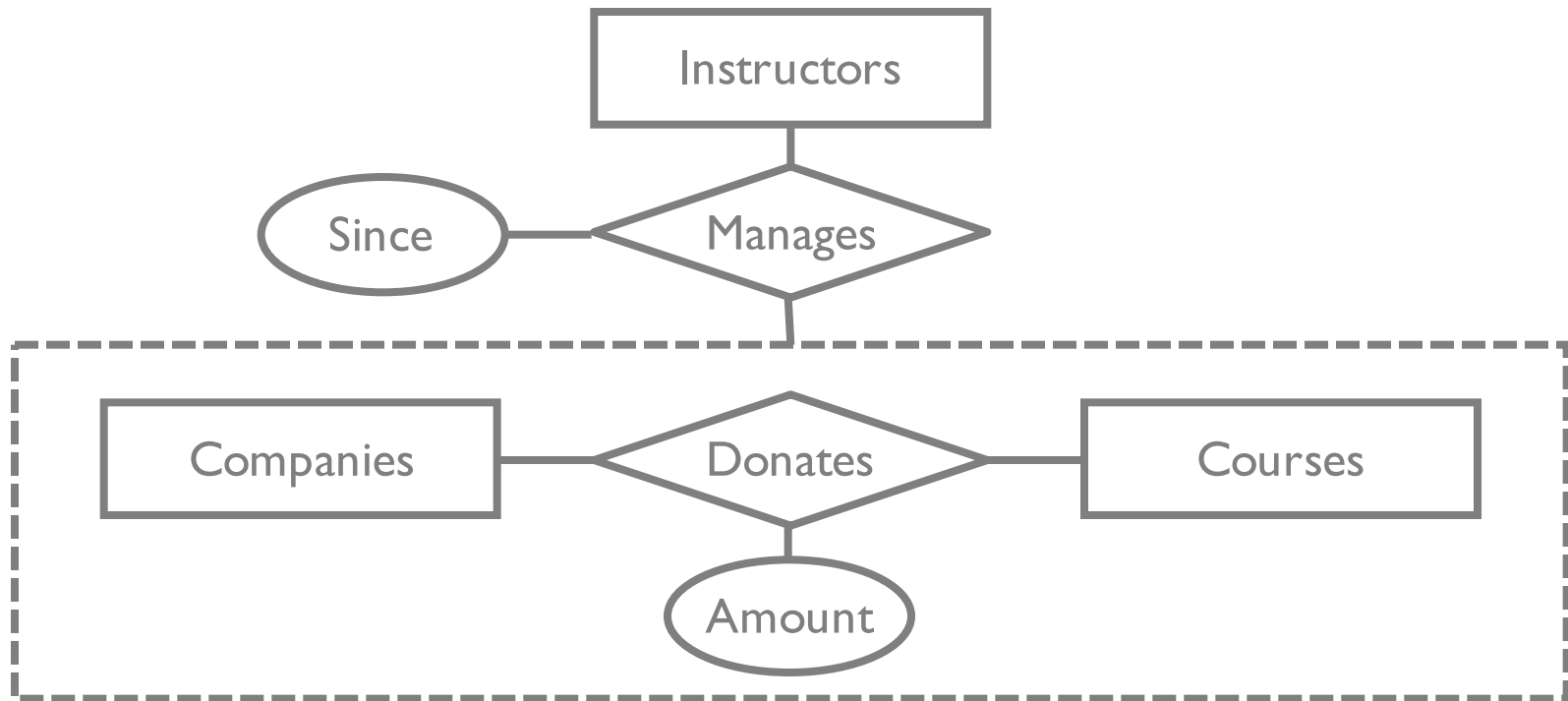
HOW DO WE EXPRESS THESE IN AN ER DIAGRAM???????



# Aggregation

Relationships between (entities – relationships)

Lets us treat a Relationship Set like an Entity Set  
so it can participate in other relationships





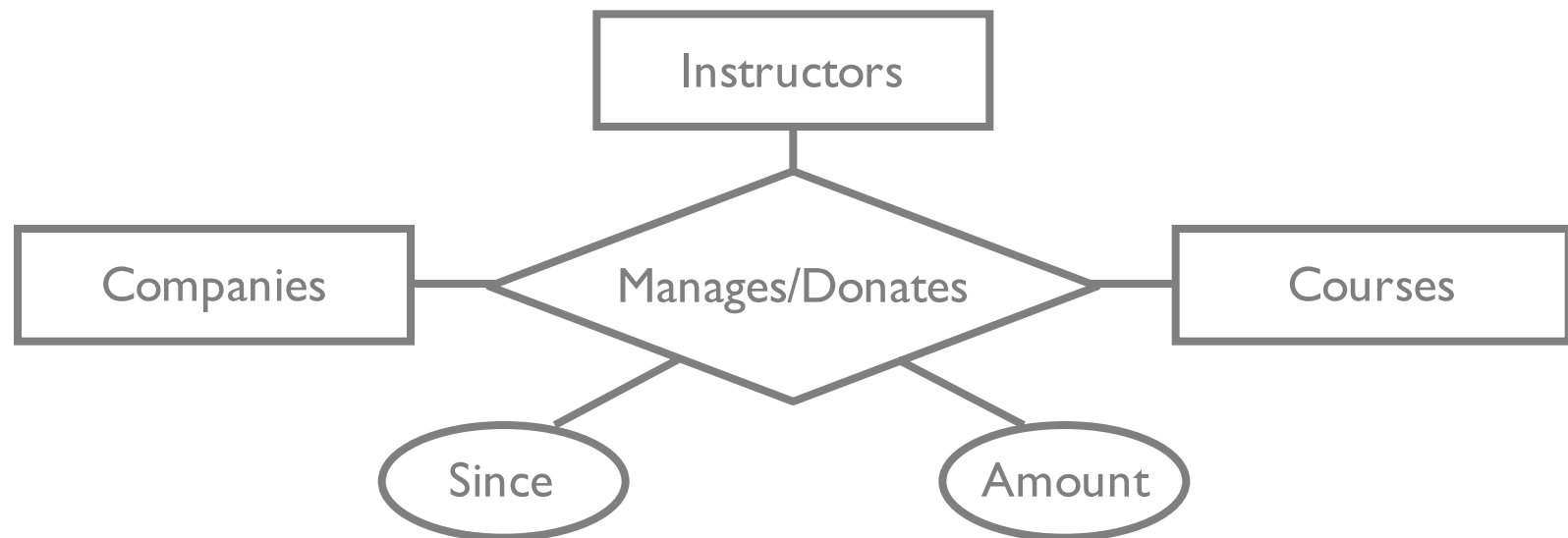
# Aggregation vs Ternary Relationships

Why use aggregation?

Manages and Donates are distinct relationships with own attrs

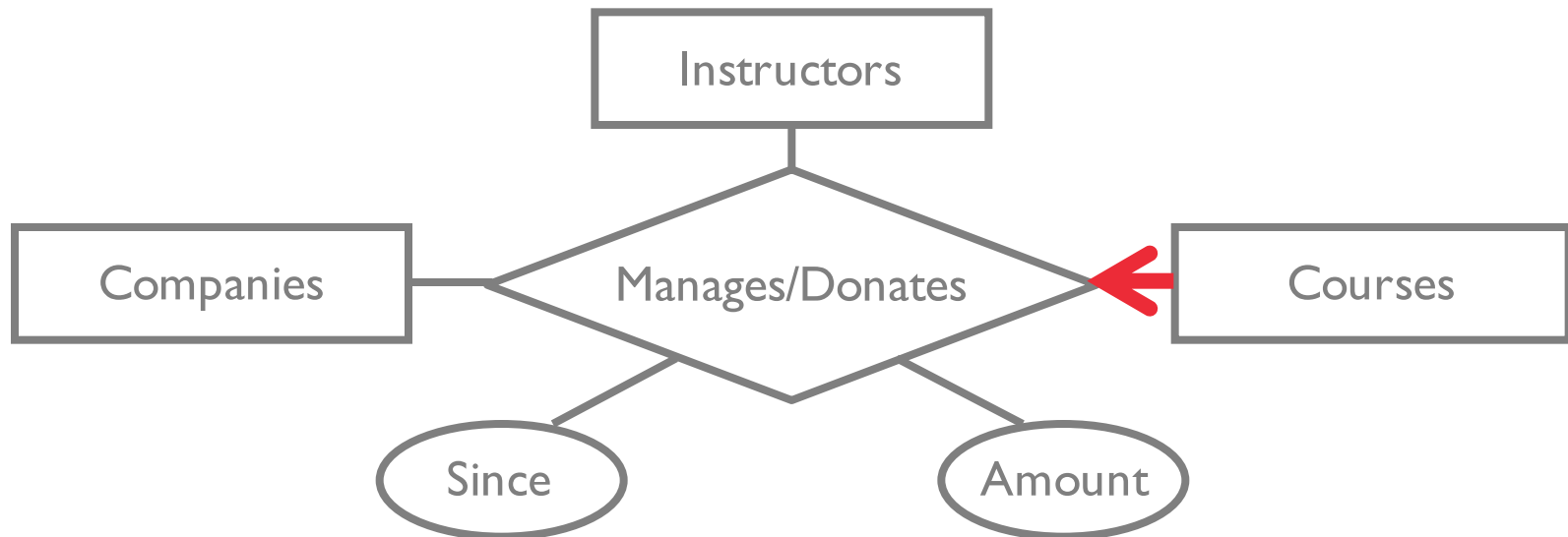
Can define constraints on relationship sets

e.g., a donation can be managed by at most one instructor



# Aggregation vs Ternary Relationships

Constraints apply to all connected entity sets



# Using the ER Model

Design Choices for a concept

Entity or Attribute?

Entity or Relationship?

Binary or Ternary relationship?

Aggregation or Ternary relationship?

# Entity or Attribute?

Is **users.address** an attribute of Users or an entity connected to Users by a relationship?

Depends (and may change over time!)

If a user has  $>1$  addresses, must be an entity

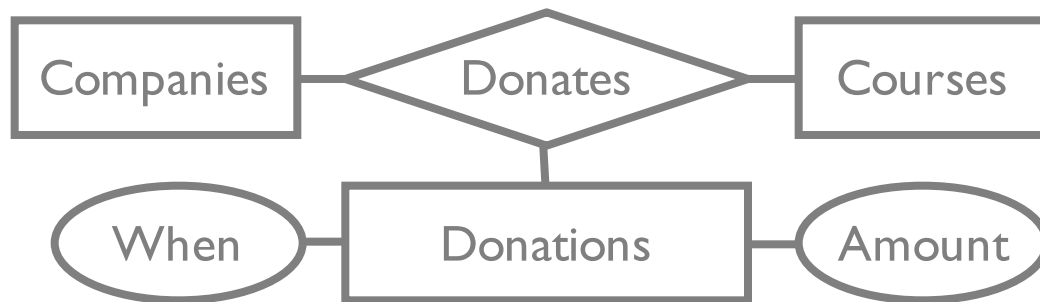
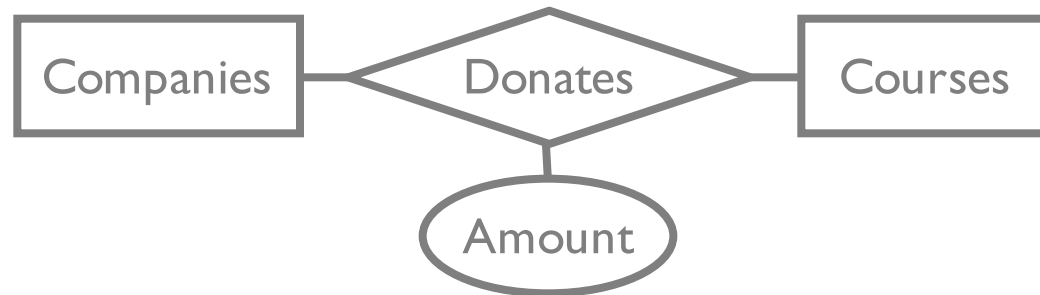
If an address has attrs (structure), must be entity

e.g., want to search for users by city, state, or zip

# Entity or Attribute?

A company can't donate multiple amounts (top fig)

Use ternary relationship (bottom fig)



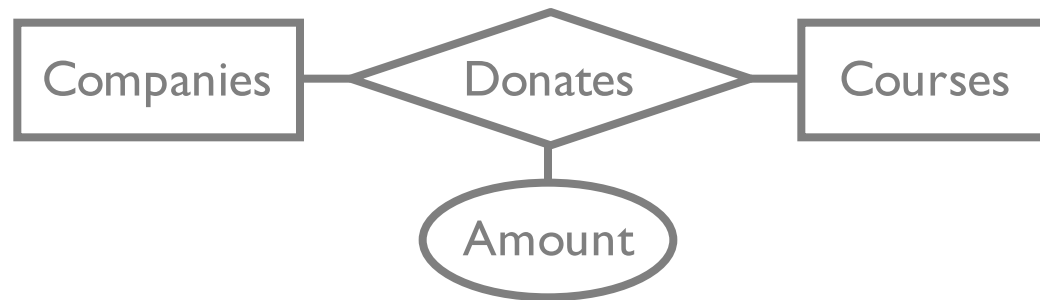
# Entity or Relationship?

OK if company donates to courses individually

What if company donates to school for all data-related courses?

**Redundancy** of *amount*, need to remember to update every one

**Misleading** implies *amount* tied to *each* donation individually



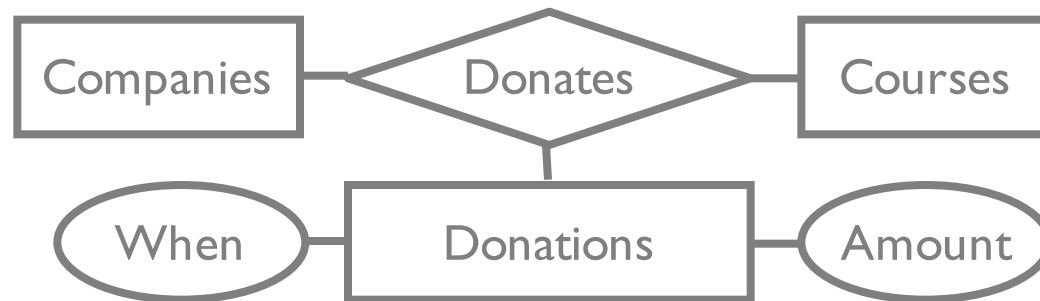
Company	Course	Amount
Amazon	4111	2000
Amazon	4112	2000
Amazon	5111	2000

} These amounts are logically the same (redundant)!

# Entity or Relationship?

If company donates once to school for data related courses.

Refactor amount into an entity

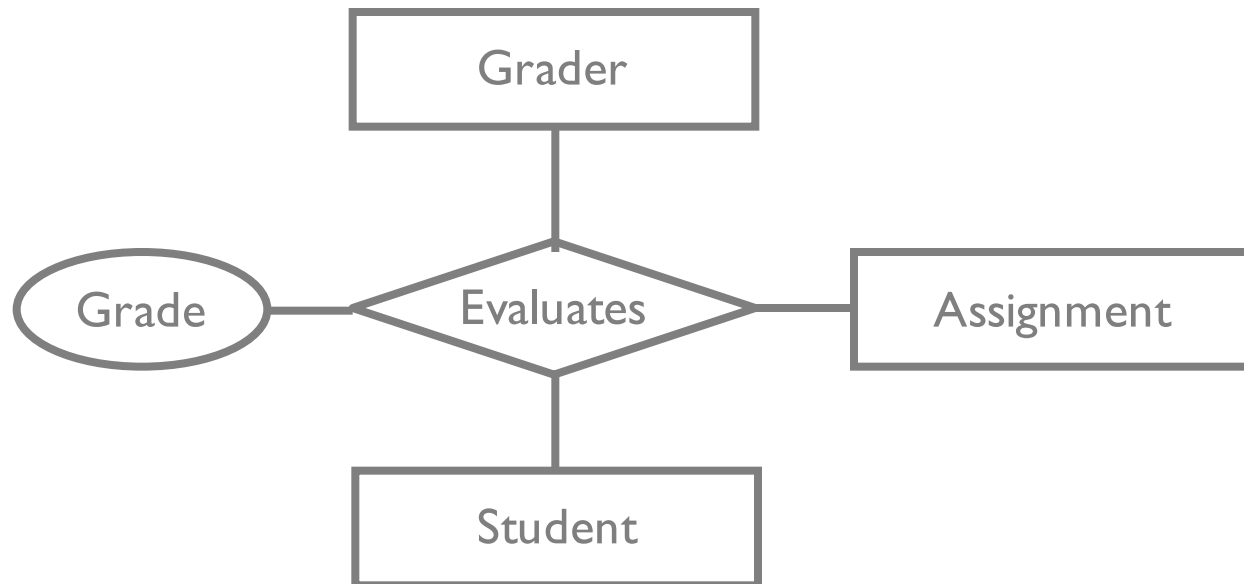


Company	Course	Donation
Amazon	4111	1
Amazon	4112	1
Amazon	5111	1

Donation	When	Amount
1	Today	2000

# Binary or Ternary Relationship?

What if assignments have at most one grader?



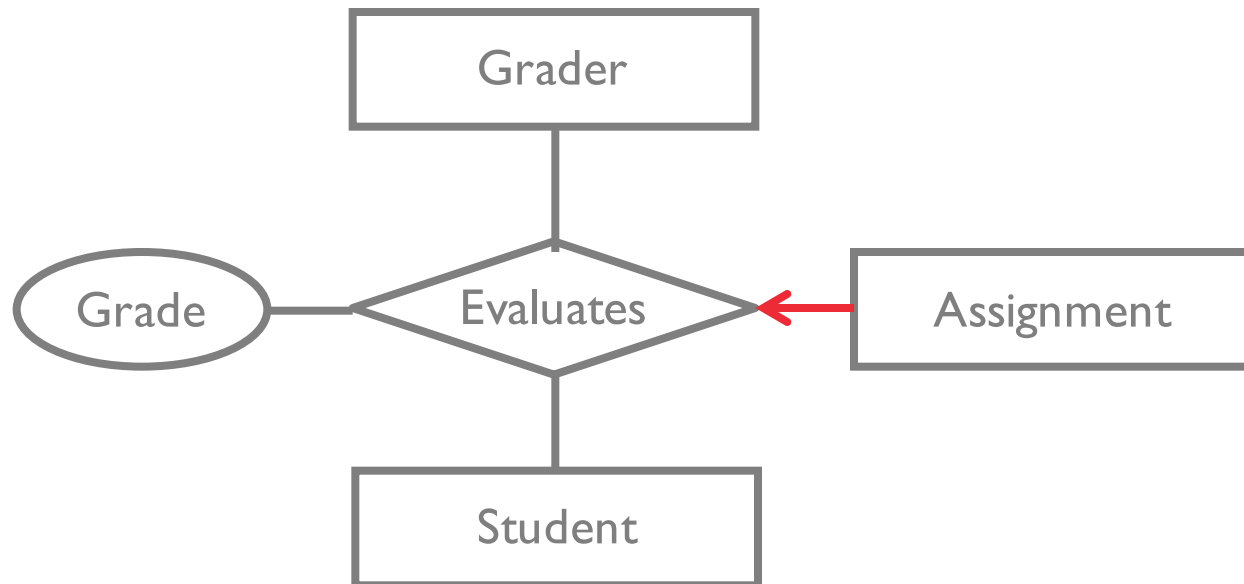


# Binary or Ternary Relationship?

What if assignments have at most one grader?

Only one student can complete HW0!

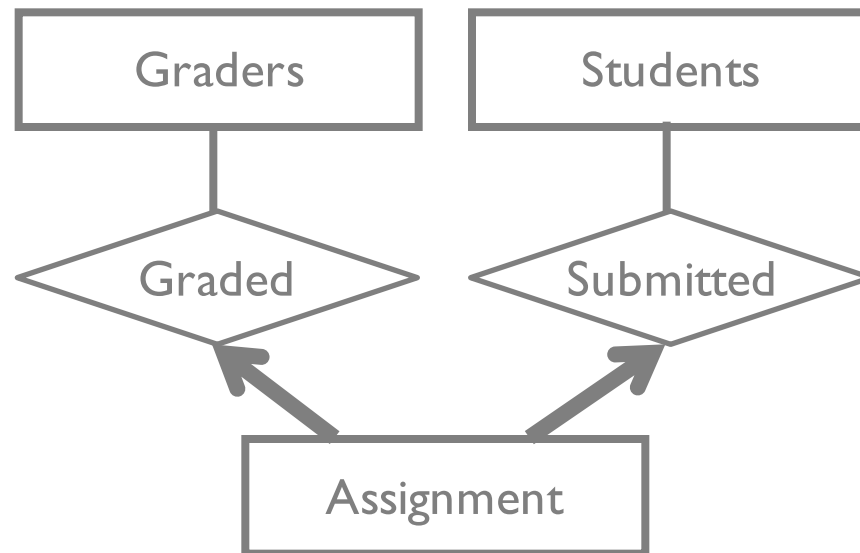
Actually two separate relationships



# Binary or Ternary Relationship?

Binary relationships allows additional constraints

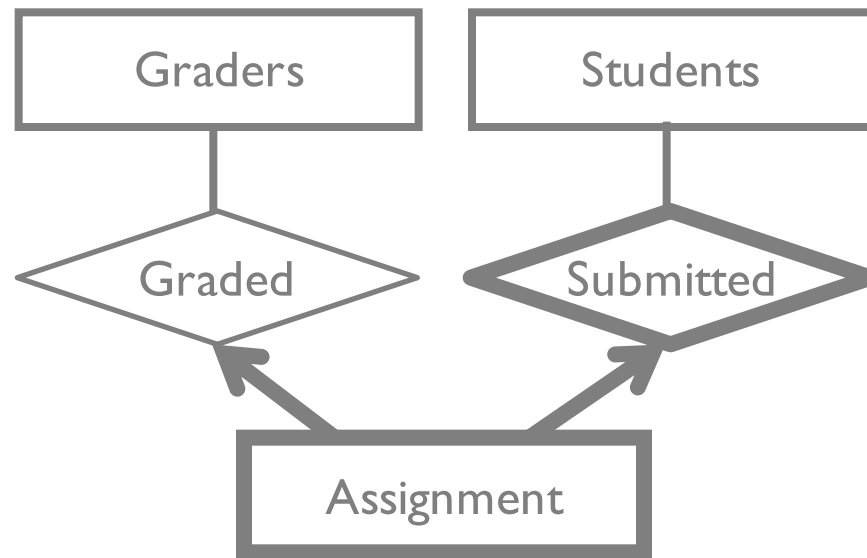
What should happen if a student drops the class? (see next slide)



# Binary or Ternary Relationship?

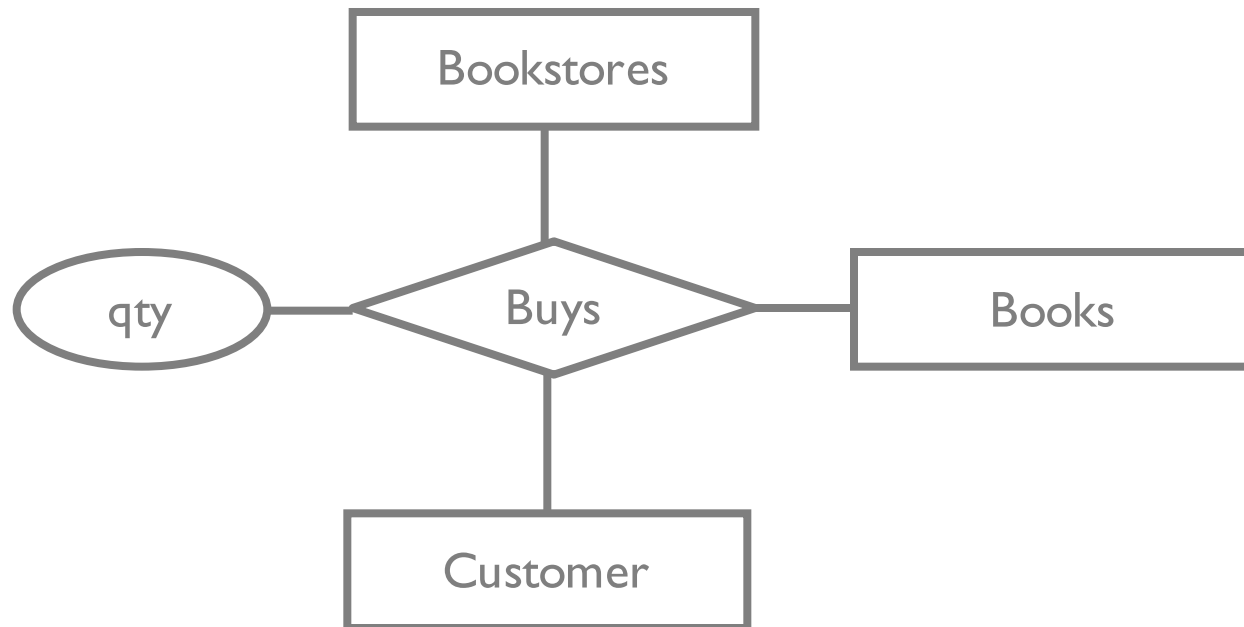
## Binary relationships allows additional constraints

What should happen if a student drops the class? The previous ER diagram is sufficient. This is wrong! You don't need a weak entity to express this. In fact, this would mean that if Jane completed HW0 and then dropped the class, then the entire HW0 (not just her submission) would disappear.



# Binary or Ternary Relationship?

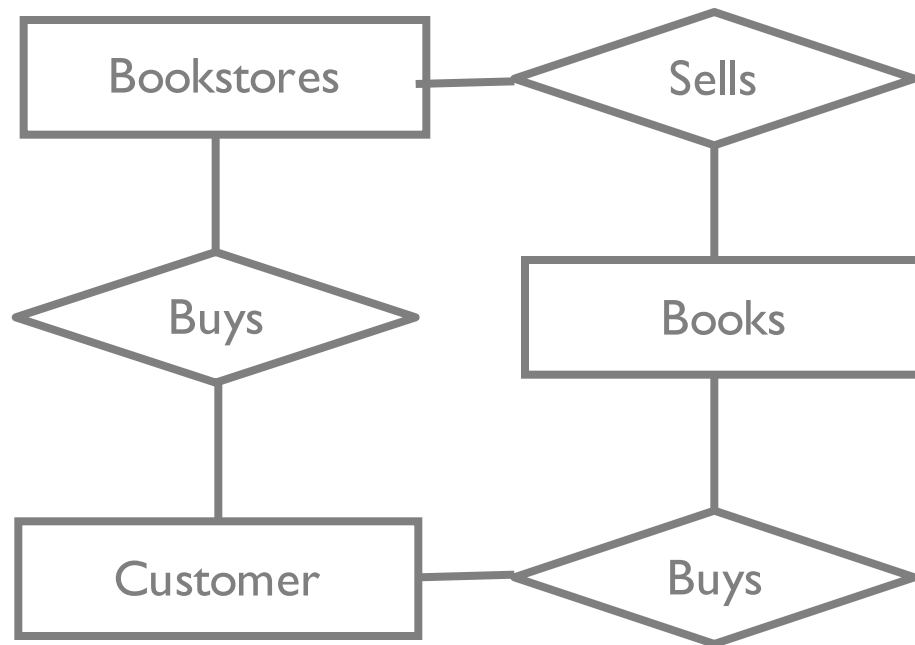
Sometimes have true ternary relationship that is defined by all three entities.



# Binary or Ternary Relationship?

Sometimes have true ternary relationship that is defined by all three entities.

Doesn't  
Really  
Work



# Using ER Modeling

## Constraints in ER Modeling

Many types of data semantics can be captured using ER

Some constraints not captured (discuss limitations later)

## Need further schema refinement

ER Model is still subjective, need further refinement after translated into relational schema

# Summary

## Requirements

what are you going to build?

## Conceptual Database Design

pen-and-pencil description

(Today) 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

## App/Security Design

prevent security problems

# Summary

Conceptual design follows *requirements analysis*

ER model helpful for conceptual design

- constraints are expressive

- matches how we often think about applications

Core constructs

- entity, relationship, attribute

- weak entities, ISA, aggregation

Many variations beyond today's discussion



# Summary

ER design is subjective based on usage+needs

Today we saw multiple ways to model same idea

ER design is not complete/perfect

Developed in an enterprise-oriented world (ER First)

Doesn't capture semantics (what does “instructor” *mean*?)

Doesn't capture e.g., processes/state machines

How to combine multiple ER models automatically?

Limitation of imagination when designing application

Open problems!

ER design is a useful way to think

# Next Time

Relational Model: de-facto DBMS standard

Set up for ER diagrams → Relational models