

Administrivia

Look for teammates

L2	8-Sep	ER Models optional: Textbook Chapter 6 except for Sections 6.7, 6.10, and 6.11.	HW1 Part 1 Project 1 Part 1	HW0 (9/11 11:59PM EST. NO LATE DAYS)
L3	13-Sep	ER Models optional: Textbook Chapter 6 except for Sections 6.7, 6.10, and 6.11.		HW 1 Part 1 (9/16 11:59PM EST) Formed Project 1 Team (no submission)
I 4	15-Sep	Relational Model		Project 1 Part 1 approval

Staff office hours will be up this weekend
Zoom links in discussion board

HW1 out today

Project 1 Part 1 out today

Find a project 1 teammate ASAP!

d

Finding Teammates Megathread #8

Eugene Wu STAFF
Now in Projects - P1Part1

UNPIN STAR WATCHING 1 VIEW

Hi all,

Please use this Megathread to find the teammate for Project 1. You will design/build a database application together. Good luck!

Comment Edit Delete Endorse ...

Add comment

2

- **Auditors OK**
 - courseworks set to institutional visibility,
 - all material on website
- **Schedule conflicts OK**
 - you are responsible for exam conflicts!

Meet the Staff



**Zachary
Huang**
Poke Bowl

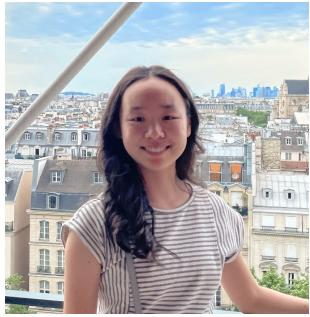


Lucas Kalejaiye
Emmy burger at Emily (West Village) is the best burger in nyc



**Weisheng
Wang**
hotpot

Meet the Staff



Jennifer Wang
Portuguese egg tarts



Lia Chen
Roti Rolls



Andrew Zheng
Uluh on 9th street and 2nd Ave.

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

Inconsistencies/Constraint Violations

Huge amount of effort to avoid inconsistencies
Can data model help us avoid automatically?

The screenshot shows a search results page with the query "dblp eugene wu" in the search bar. Below the search bar, there are tabs for "Web", "News", "Images", "Videos", "Shopping", "More", and "Search tools". The "Web" tab is selected. The results section displays three entries, each starting with "dblp: Eugene Wu". Each entry includes a link to "dblp.uni-trier.de", a date ("May 9, 2015"), and a description of the result.

Result Type	Link	Date	Description
1	dblp: Eugene Wu 0002	May 9, 2015	University of Trier - List of computer science publications by Eugene Wu 0002.
2	dblp: Eugene Wu	May 9, 2015	University of Trier - List of computer science publications by Eugene Wu.
3	dblp: Eugene Wu 0001	May 9, 2015	University of Trier - List of computer science publications by Eugene Wu 0001.

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

Eugene Wu - Columbia University

Eugene Wu received his Ph.D. from MIT, B.S. from Cal, and was a postdoc in the AMPLab. A profile, an obit. **Eugene Wu** has received the VLDB 2018 10-year test of time award, best-of-conference citations at ICDE and VLDB, the SIGMOD 2016 best demo award, the NSF CAREER, and the Google and Amazon faculty awards.

F <https://www.forbes.com/profile/eugene-wu>

Eugene Wu - Forbes

#39 **Eugene Wu** on the 2021 Taiwan's 50 Richest - **Wu** is the founder of Shin Kong Financial, one of Taiwan's largest private-sector financial companies. **Wu** stepped down as the firm's chairman in June ...

F PROFILE Finance & Investments

#39 Eugene Wu

\$1.4B

REAL TIME NET WORTH
as of 1/27/22

▲ \$11 M | 0.77%

Reflects change since 5 PM ET of
prior trading day



Inconsistencies/Constraint Violations

Check in application code!



The image shows a screenshot of a web application's sign-up form. At the top, there is a Google logo. Below it, the form has fields for 'Name' (split into 'First' and 'Last') and 'Choose your username'. A red box highlights the 'username' field containing 'eugenewu @gmail.com'. Below this, a red message says 'Someone already has that username. Try another?'. It suggests an alternative: 'Available: eugenewu861'. At the bottom, there is a 'Create a password' field.

Name

First Last

Choose your username

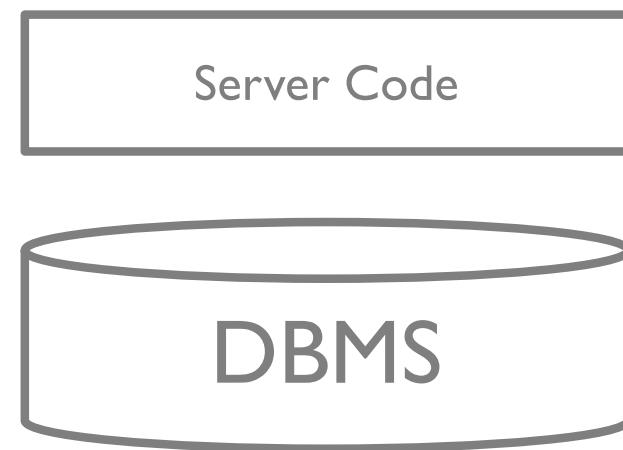
eugenewu @gmail.com

Someone already has that username. Try another?

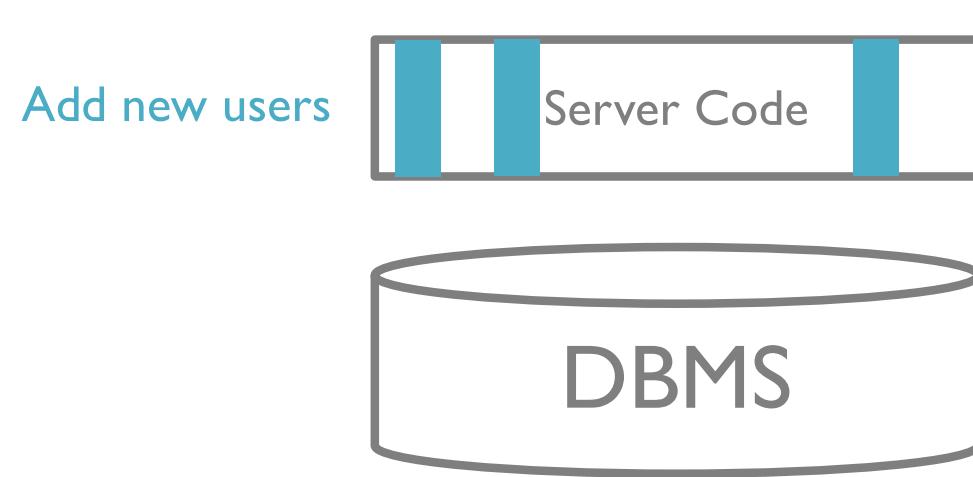
Available: eugenewu861

Create a password

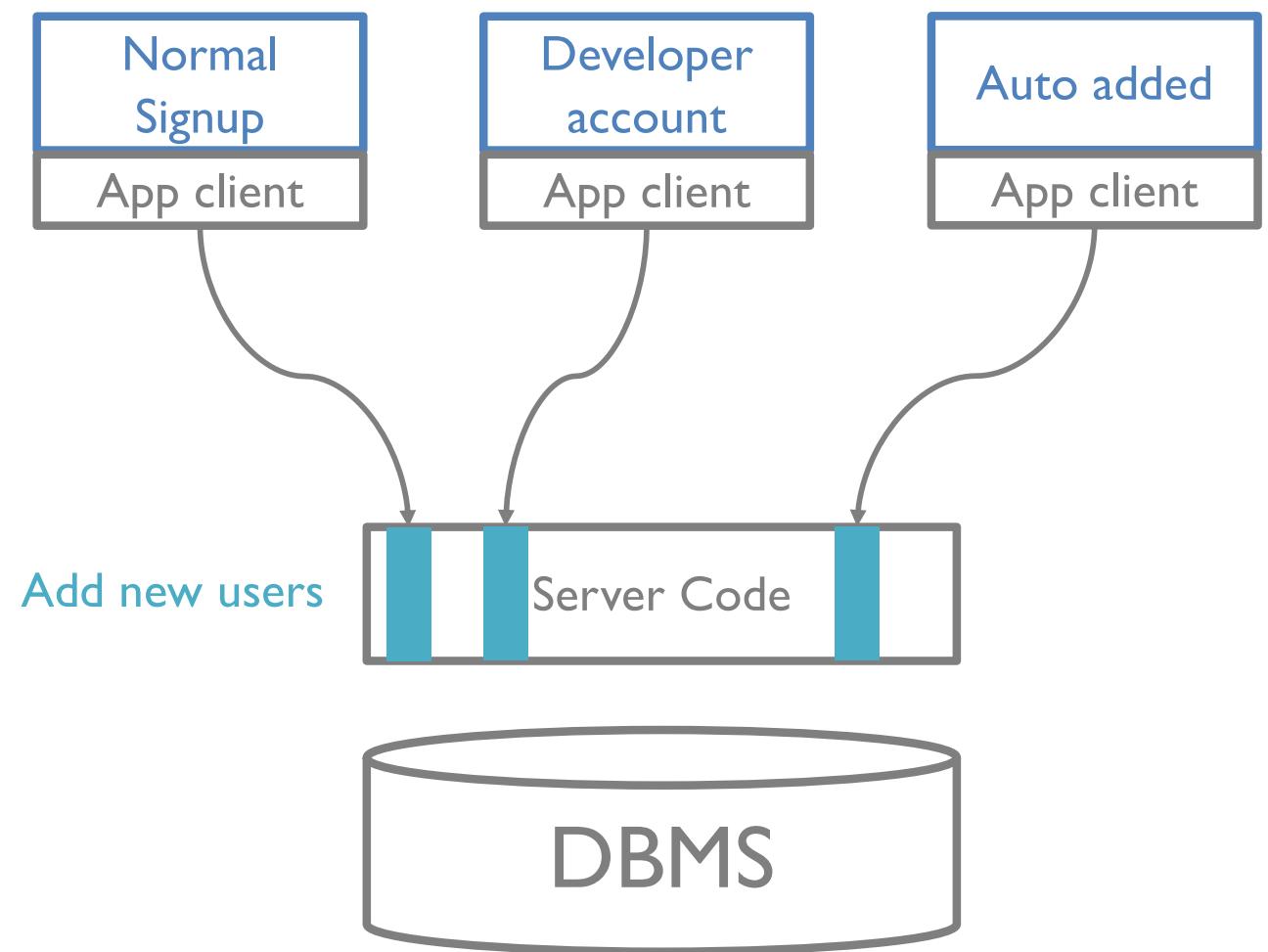
It is Hard to Design Applications



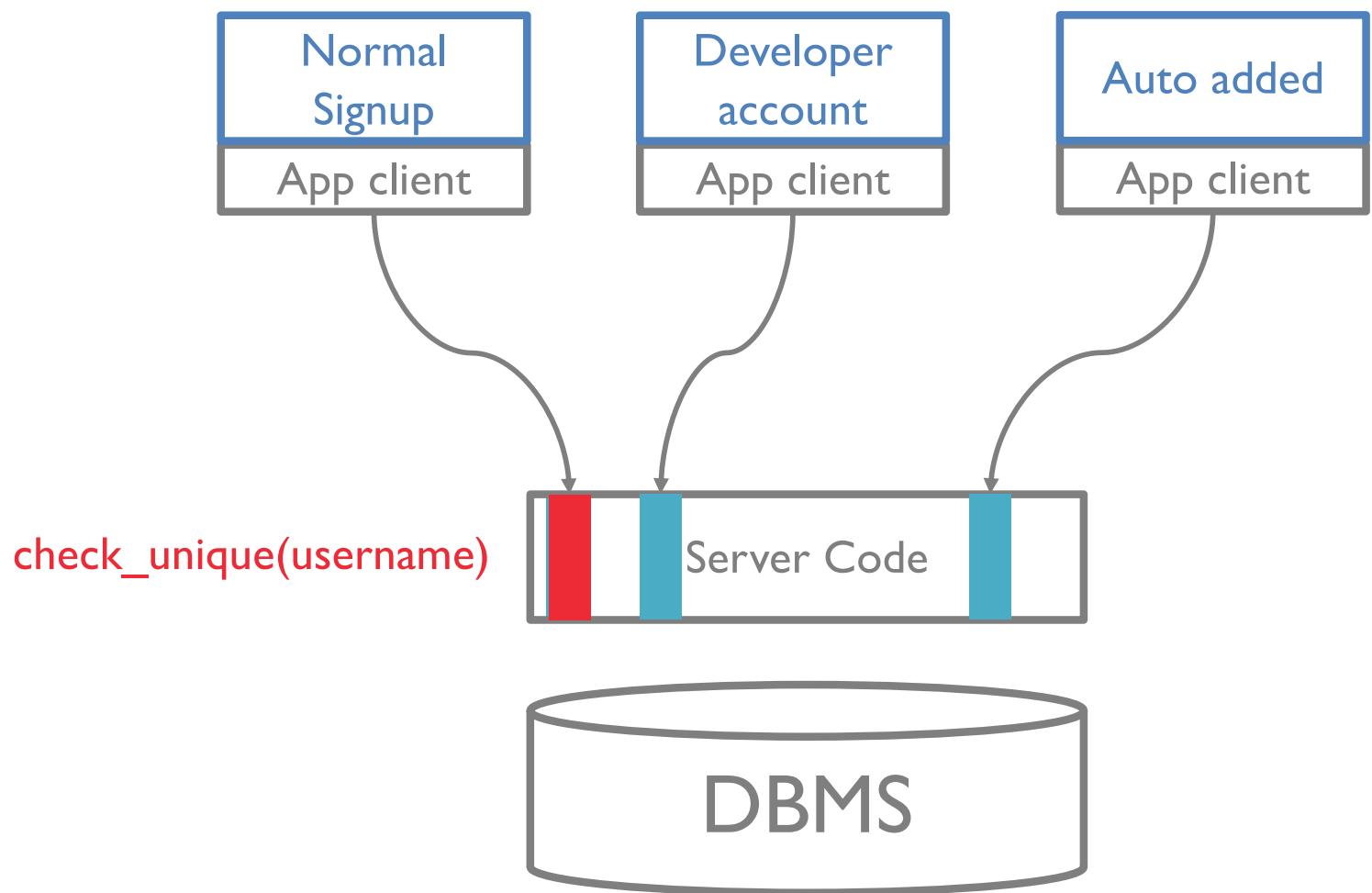
It is Hard to Design Applications



It is Hard to Design Applications



It is Hard to Design Applications



ER Diagrams

What is it?

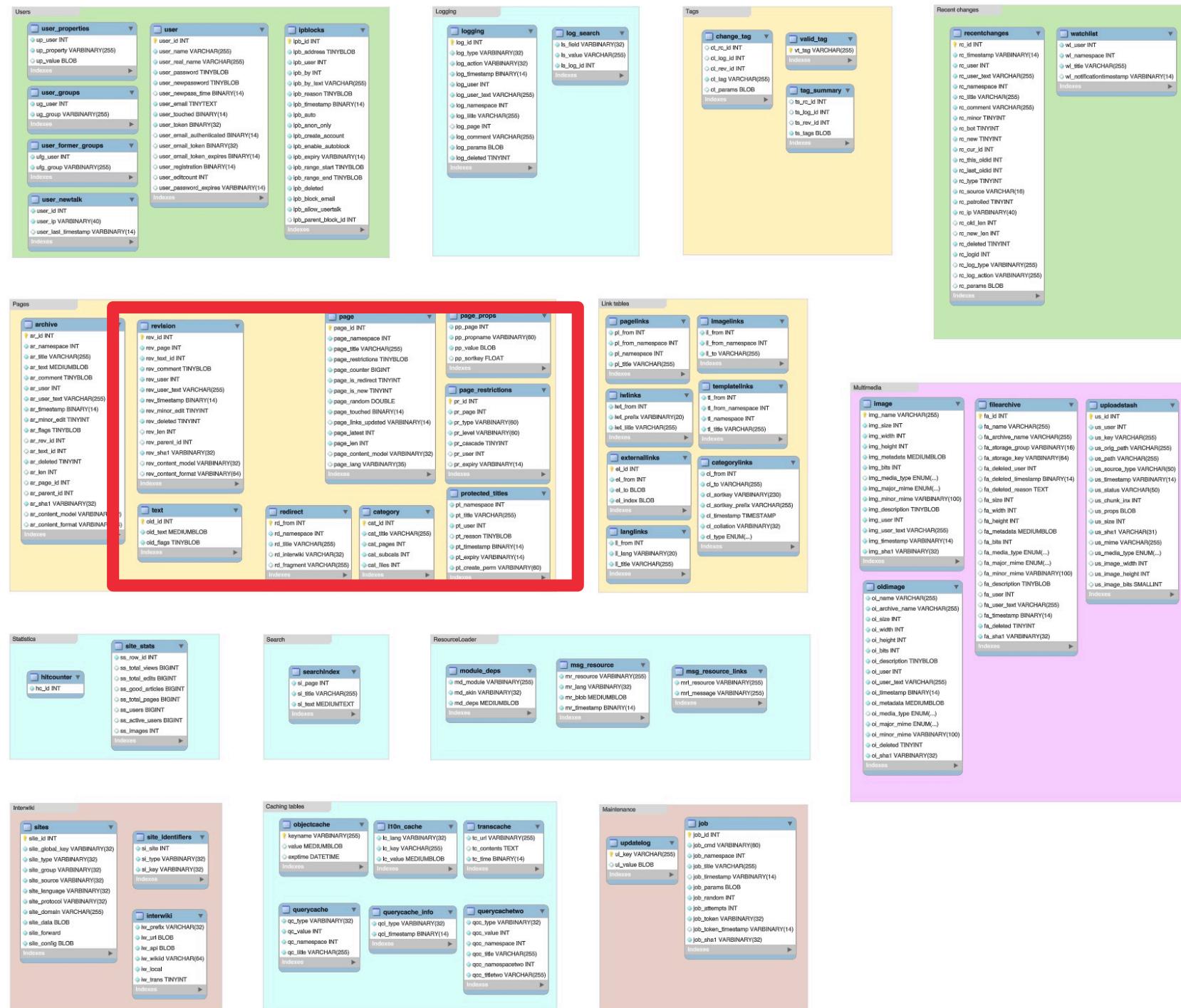
- A way to sketch the core information that your database will eventually store.
- Visually encodes important constraint information

Who cares?

- Good for “white boarding” together
- Good way to share the “gist” of your DB’s structure

```
test=# \d election
          Table "public.election"
  Column | Type   | Collation | Nullable | Default
-----+-----+-----+-----+-----+
  year   | integer |           |           |
  state  | text    |           |           |
  state_po | text    |           |           |
  state_fips | integer |           |           |
  state_cen | integer |           |           |
  state_ic  | integer |           |           |
  office   | text    |           |           |
  candidate | text    |           |           |
  party_detailed | text    |           |           |
  writein   | text    |           |           |
  candidatevotes | integer |           |           |
  totalvotes | integer |           |           |
  version   | integer |           |           |
  notes     | text    |           |           |
  party_simplified | text    |           |           |
  id        | integer |           | not null | nextval('election_id_seq'::regclass)
Indexes:
  "election_id_key" UNIQUE CONSTRAINT, btree (id)

test=# \d food
          Table "public.food"
  Column | Type   | Collation | Nullable | Default
-----+-----+-----+-----+-----+
  camis  | integer |           |           |
  dba    | text    |           |           |
  boro   | text    |           |           |
  building | integer |           |           |
  street  | text    |           |           |
  zipcode | integer |           |           |
  phone   | bigint  |           |           |
  inspection_date | date   |           |           |
  action   | text    |           |           |
  score    | integer |           |           |
  grade    | text    |           |           |
  inspection_type | text   |           |           |
  census_tract | integer |           |           |
  year     | integer |           |           |
  month    | integer |           |           |
  day      | integer |           |           |
```



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	

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	

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(36)
Indexes	

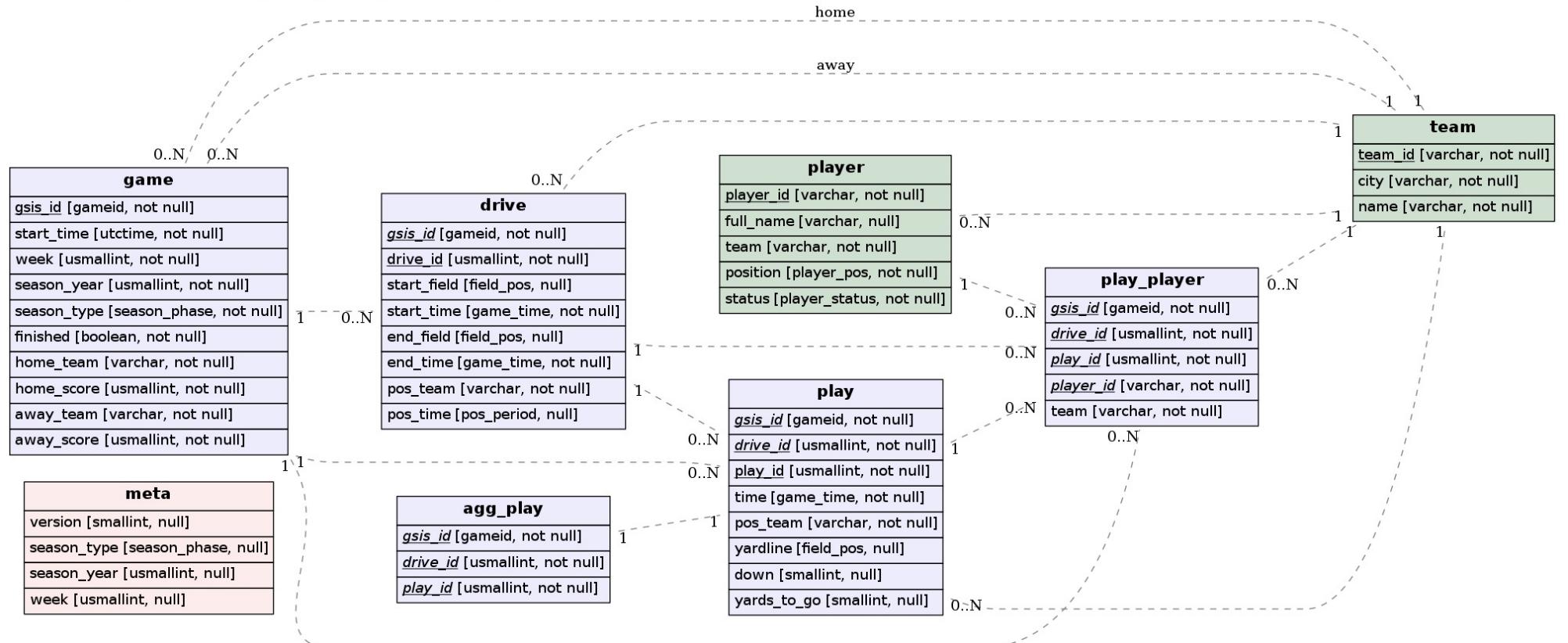
page_props	
pp_page	INT
pp_propname	VARBINARY(60)
pp_value	BLOB
pp_sortkey	FLOAT
Indexes	

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	

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

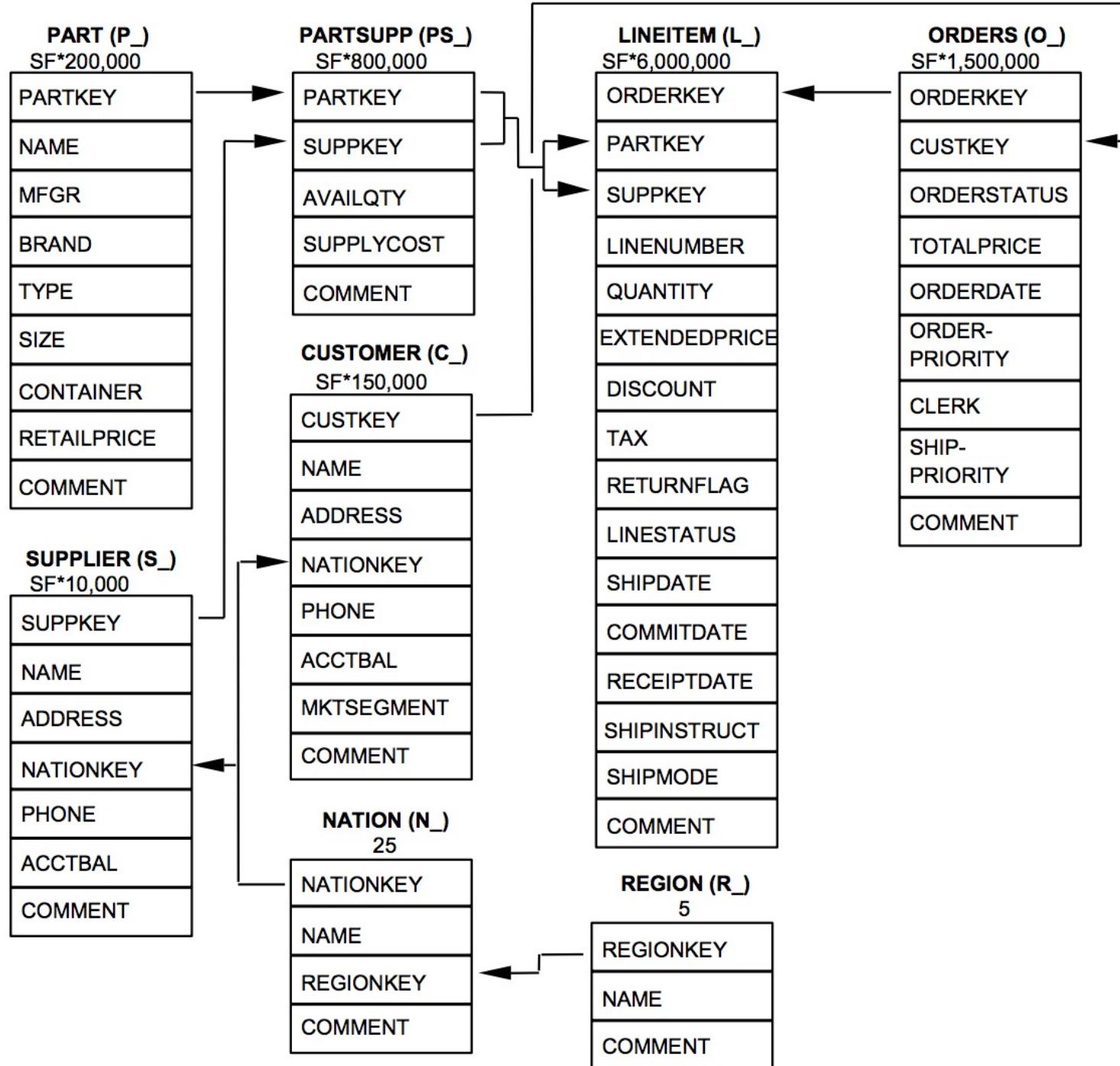
protected_titles	
pt_namespace	INT
pt_title	VARCHAR(255)
pt_user	INT
pt_reason	TINYBLOB
pt_timestamp	BINARY(14)
pt_expiry	VARBINARY(14)
pt_create_perm	VARBINARY(64)

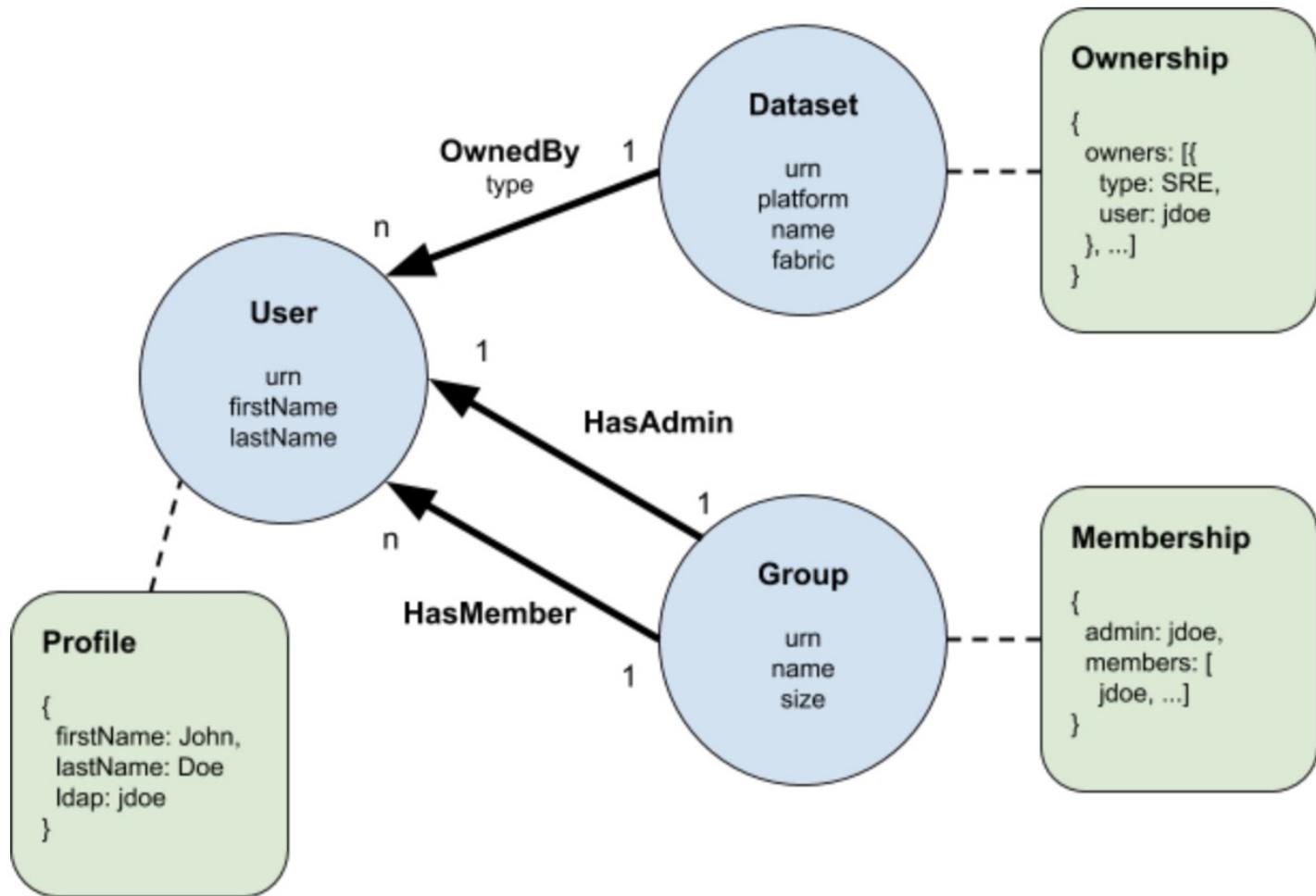
nfldb Entity-Relationship diagram (condensed)



<https://github.com/BurntSushi/nfldb/wiki/The-data-model#er-diagrams>

Figure 2: The TPC-H Schema





<https://engineering.linkedin.com/blog/2019/data-hub>

All Variations of ER diagrams

In practice, everyone uses different notations.

What matters are the core *concepts*

(in this class, we will learn a specific notation)



COMSW4111_001_2015_3: INTRODUCTION TO DATABASES (Fall 2015)

View Site As  - Select Role -

- Student
- Teaching Assistant

INTRODUCTION TO DATABASES

[Home](#) 

[Files & Resources](#) 

[Syllabus](#) 

[Mailtool](#) 

[Gradebook](#) 

[Site Settings](#) 

[Library Reserves](#) 

[Research Guides](#) 

[Roster](#) 

[Textbooks](#) 

[Piazza](#) 

[Help](#) 

COMSW4111_001_2015_3

CourseNo: COMSW4111_001_2015_3

Meeting Time: MW 02:40P-03:55P **Meeting Location:** [SEELEY W. MU 833](#)

Instructor Information:

[Eugene Wu](#)

Entity-Relationship Modeling

Entities (objects) to store and their attributes
Relationships between entities and their attrs.
Integrity constraints & business rules

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](#)

Say something

Say it

Profile Wall

Wall

Basic Information

Nickname

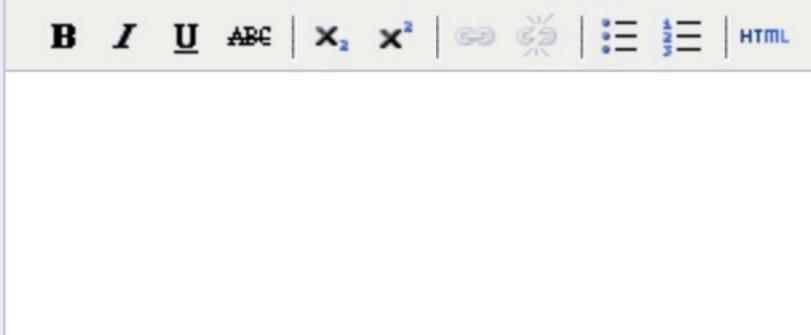
ANSWER The answer is 1000.

Birthday

[View Details](#) | [Edit](#) | [Delete](#)



Personal summary



Save changes

Cancel

Contact Information

Email

ew2493@columbia.edu

Home page

[View Details](#) | [Edit](#) | [Delete](#)

Work phone

[View Details](#) | [Edit](#) | [Delete](#)

Home phone

ANSWER The answer is 1000.

Facsimile

For more information about the study, please contact Dr. John Smith at (555) 123-4567 or via email at john.smith@researchinstitute.org.

Users

Nickname

Name

Birthday

Summary

Email

1

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

(can use as hashtable key to lookup in table)

Course
cid
name
loc
schedule

Users
uid
name
age
summary

Basics: Relationships

Relationship: association between 2 or more entities

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



Relationship Set: collection of similar relationships

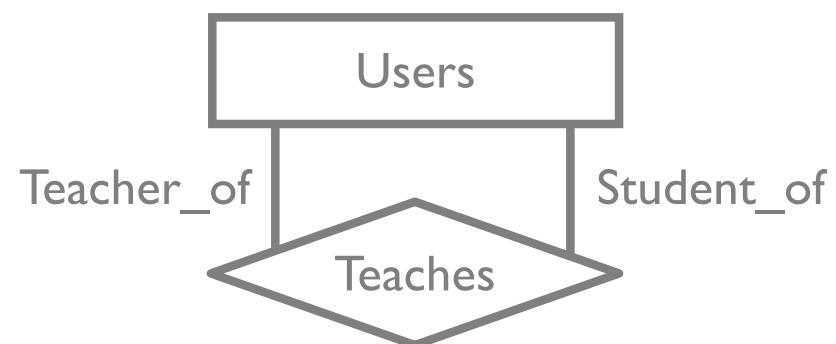
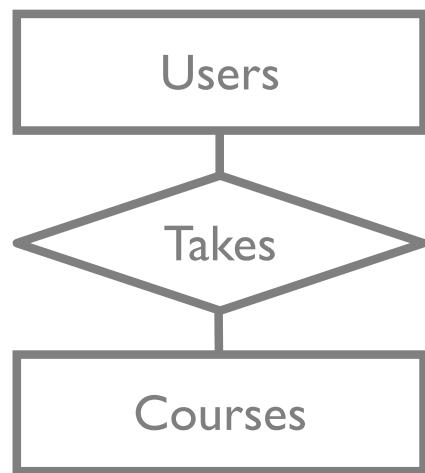
N-ary relationship set R relates N entity sets $E_1 \dots E_n$

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

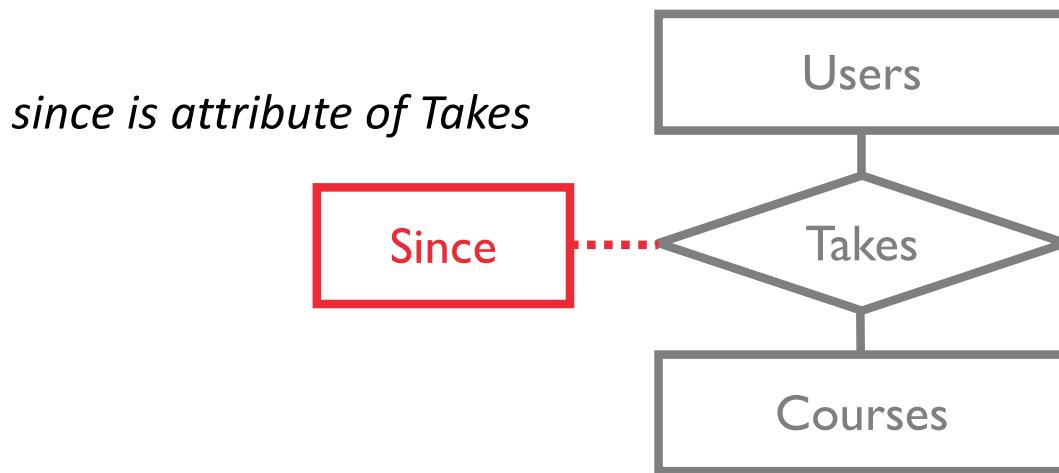
Basics: Relationships

Users can have different roles
in same relationship set



Basics: Relationships

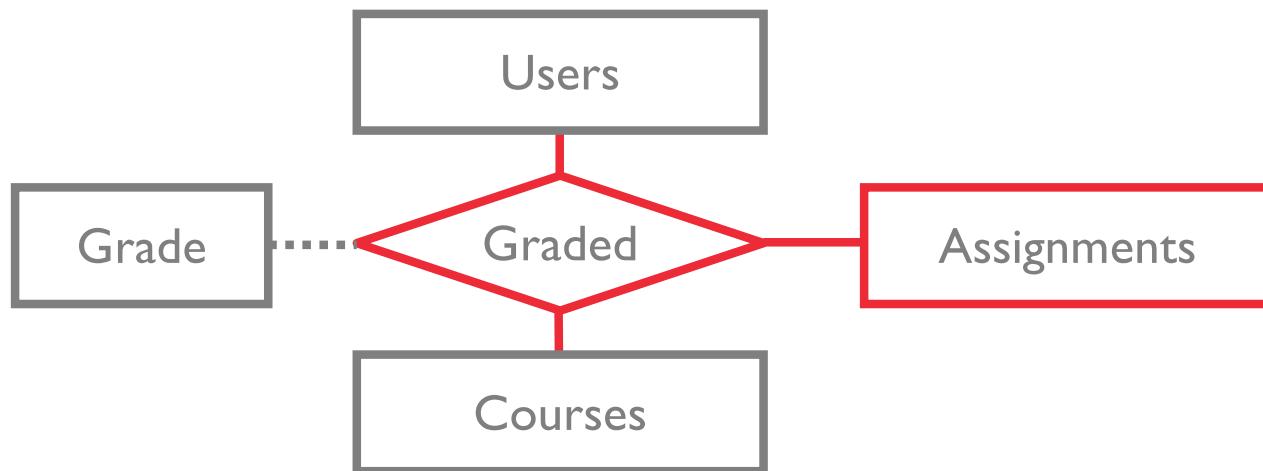
Relationships sets can have descriptive attributes
Denoted with dotted line from diamond to box



Basics: Ternary Relationships

Connects three entities

N-ary relationships possible too.



Assignments, Courses, and Users participate in the Graded relationship set

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., *Takes*

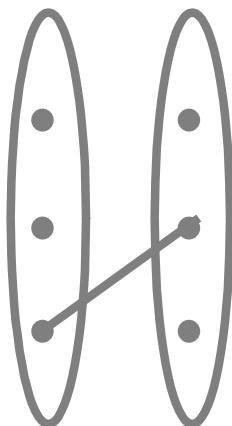
a user can take many courses

a course can have many users that take the course

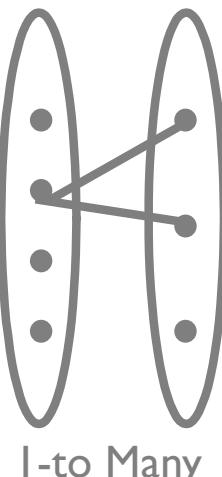
One to Many e.g., *Instructs*

a course has at most one instructor

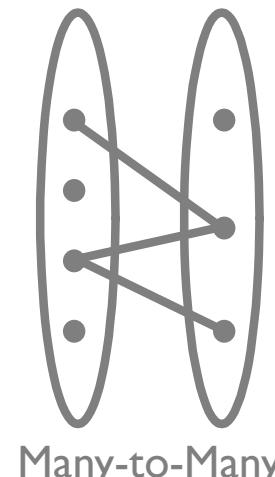
Draw arrow from diamond to box



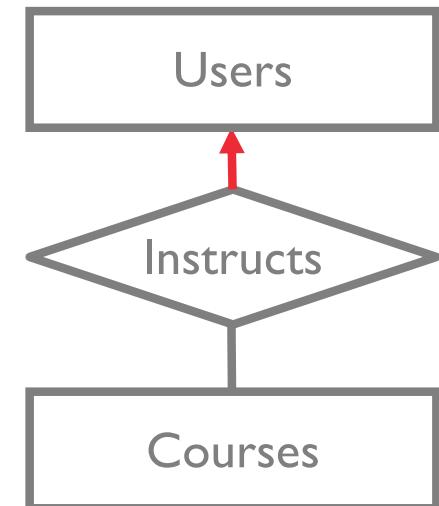
1-to-1

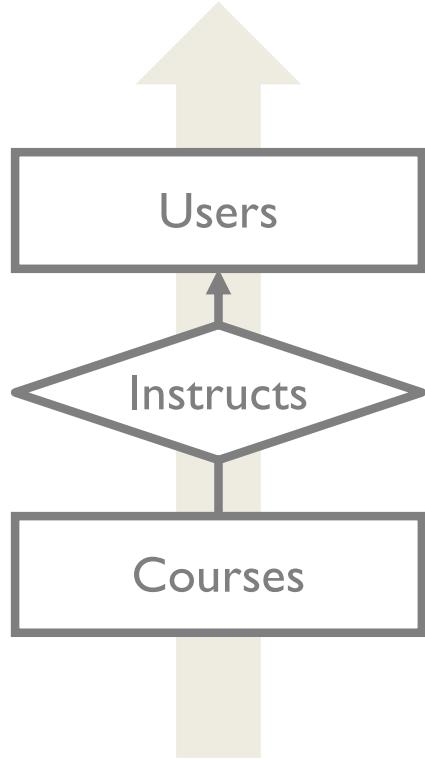


1-to Many

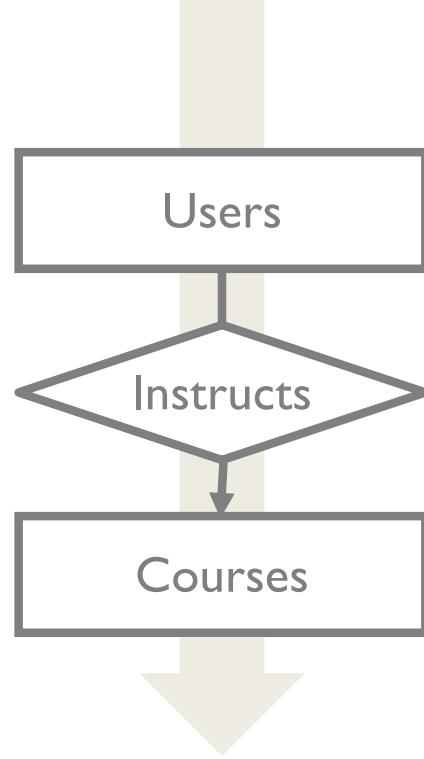


Many-to-Many

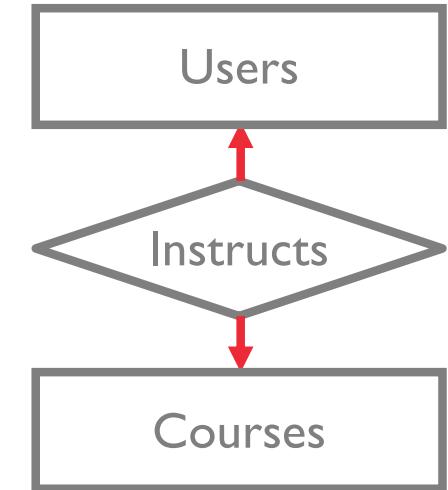




*A course is instructed by ≤ 1 user
(read along the beige arrow)*



A user instructs ≤ 1 course



*A course is instructed by ≤ 1 user
AND
A user instructs ≤ 1 course*

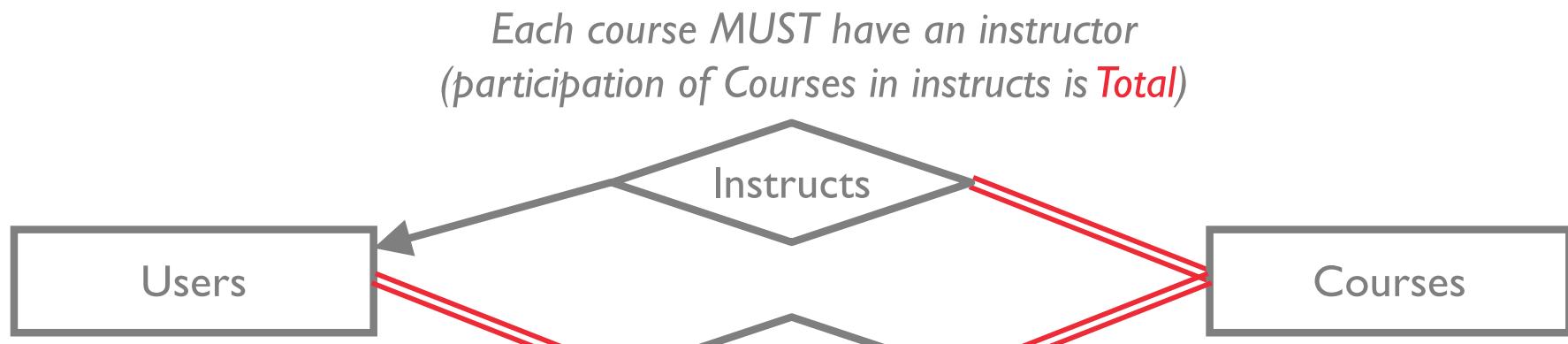
Participation Constraints

Does every course need an instructor?

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

Otherwise, **partial** participation constraint

Denoted by double line between entity set and relationship set



*Each user must take at least one course and
Each course must have at least one user (student)*

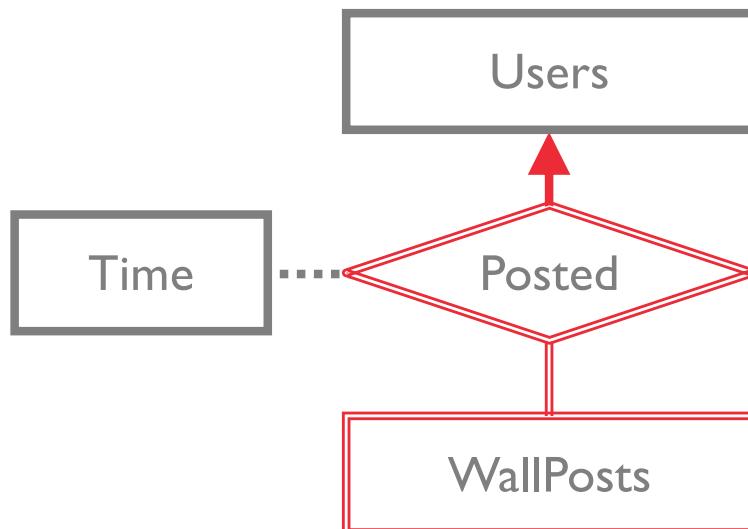
Weak Entities

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

Owner and weak entity sets must have 1-to-N relationship

Weak entity set must have total participation in this *identifying* relationships set

Denoted as double line around weak entity, set relationship set, and the edge between them; an arrow to owner entity



Say something

Profile Wall

B I U ABC | x x² | ☺ ☻ | ::

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

This screenshot shows a user profile for 'Eugene Wu' on a platform. At the top, there's a text input field labeled 'Say something'. Below it are 'Profile' and 'Wall' buttons, with 'Wall' being the active tab. Underneath are standard rich text editing tools. The main area displays three posts made by the user, each consisting of a profile picture, the user's name, the post content, and the timestamp.

General Cardinality Constraints



same as



A user instructs 0 to ∞ courses

A course has 0 to 1 instructors



Each A entity has a relationship with between x to y different B entities

Each B entity has a relationship with between n to m different A entities

Read arrows pointing in the direction from start to end

Each A is related to at most 1 B; A has N-to-1 relationship with B



Each B is related to any number of As; B has 1-to-N relationship with A



B has at most one A



B has at least one A



B has exactly one A



B is a weak entity

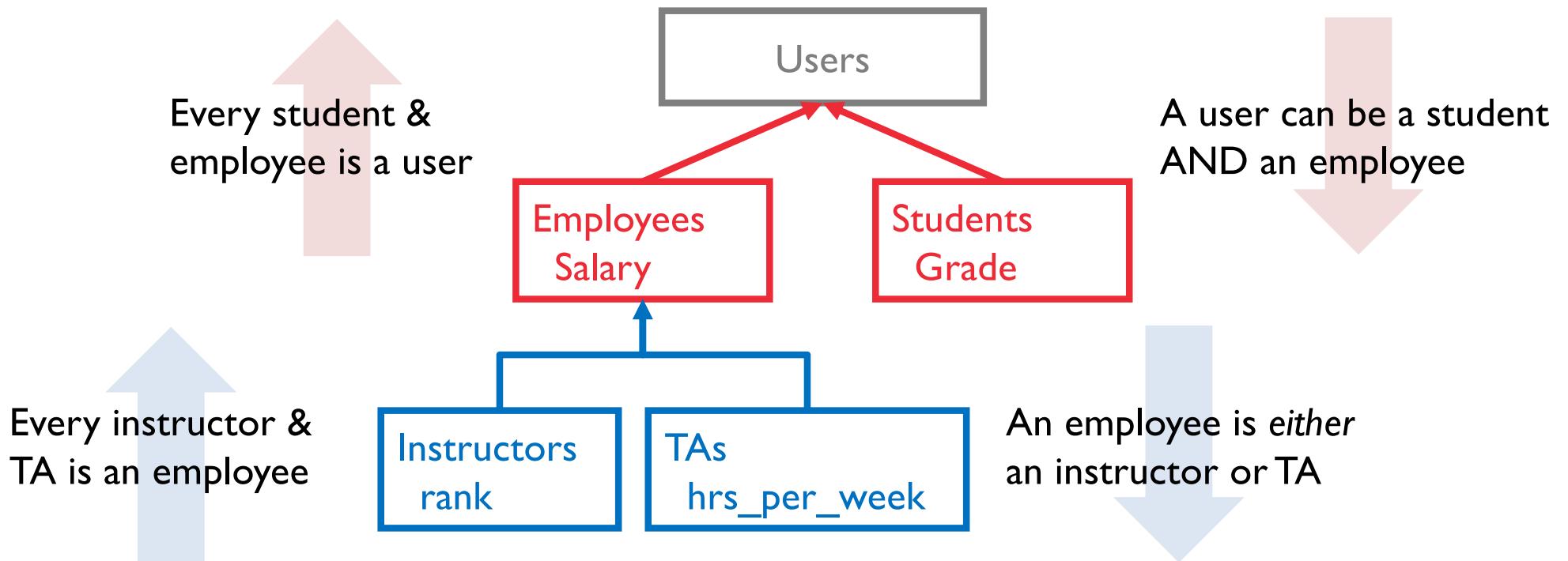
Specialization Hierarchies

Inheritance rules similar to programming languages

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

- identify entity set that participate in a relationship

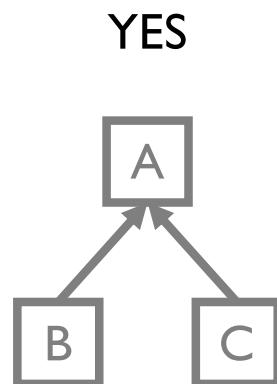
Denoted with arrow from subclass to superclass without a diamond



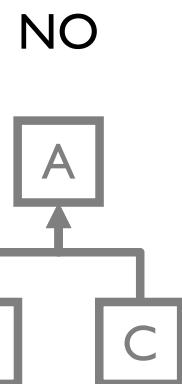
Specialization Hierarchies

Overlap Constraint

can A be a B *and* a C?

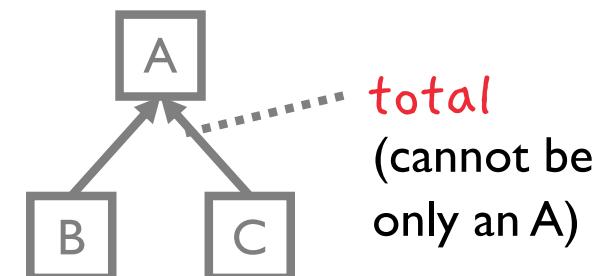


separate arrows



merged into 1 arrow

Total Specialization Constraint
must A be a B or C?
specify as the comment “total”
with dashed link to arrows

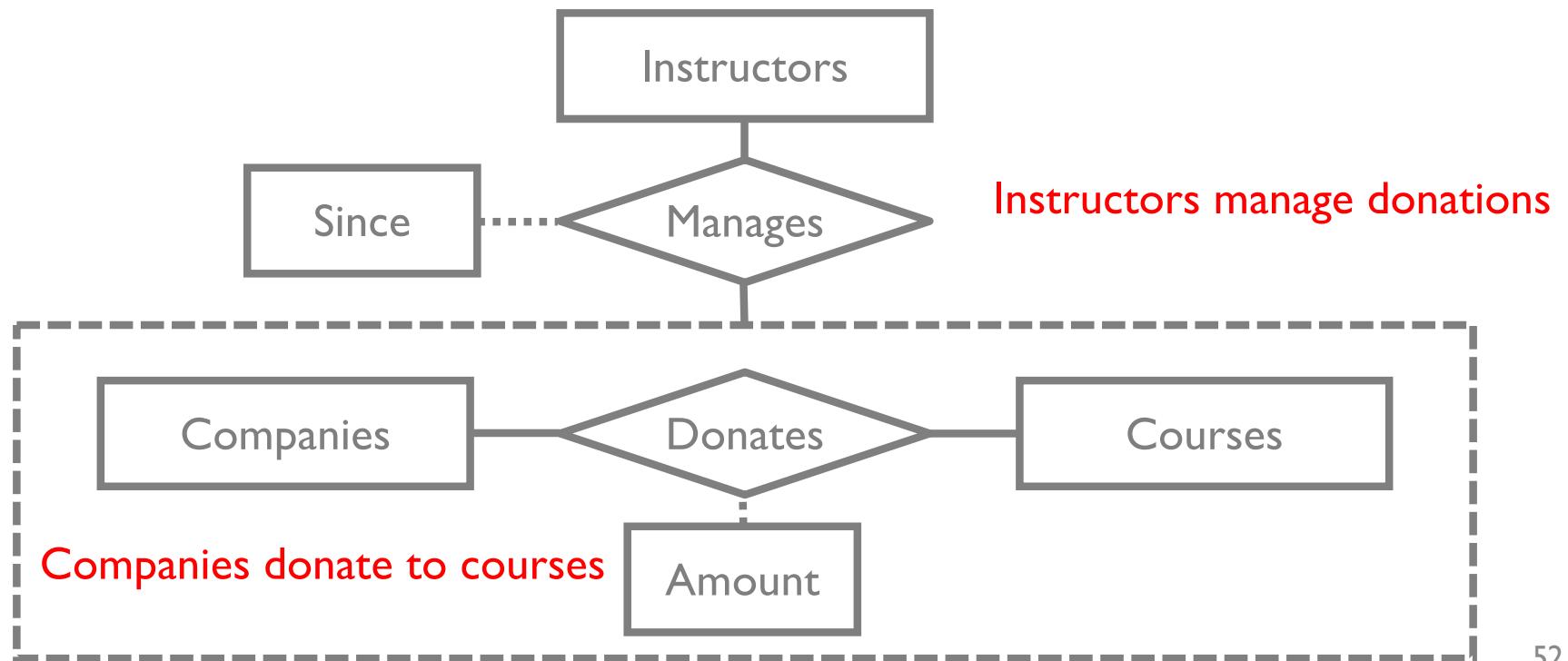


Aggregation

Relationships between (entities – relationships)

Treat Relationship Set like an Entity Set to participate in other relationships

Denoted as dashed line around the relationship set & participating entity sets



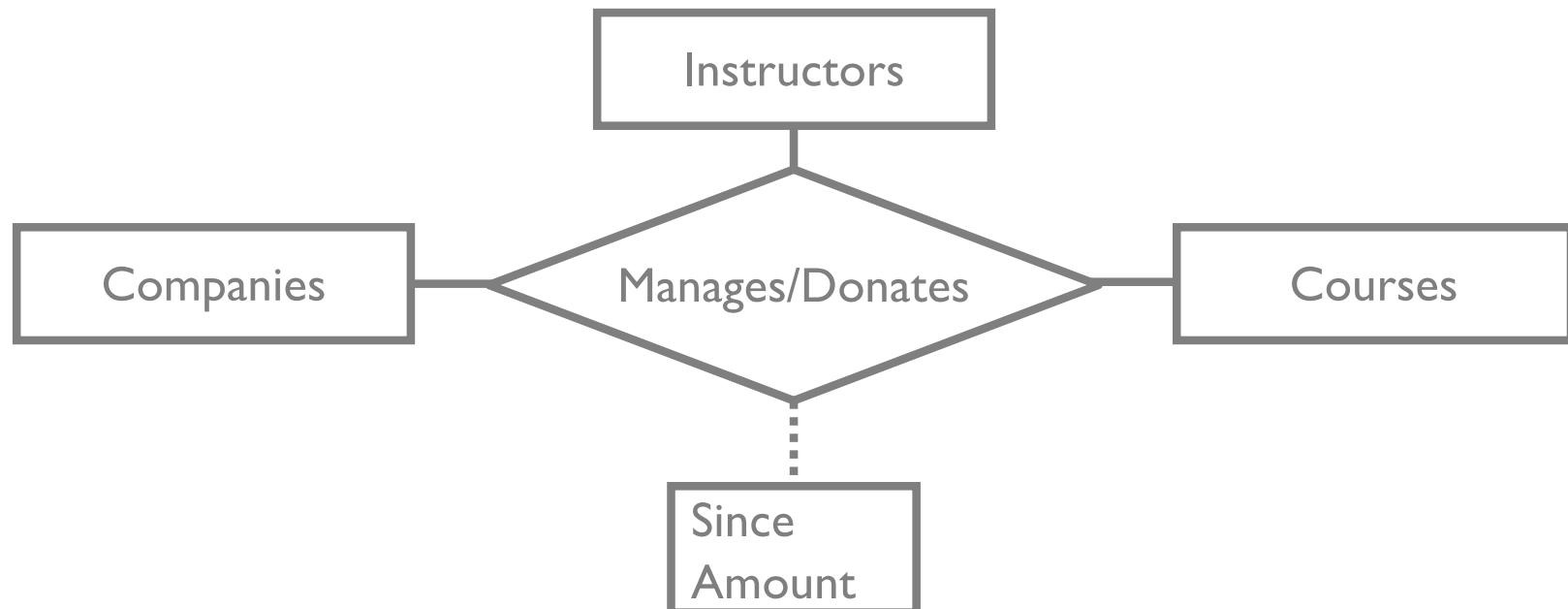
Aggregation vs Ternary Relationships

Why use aggregation?

Manages and Donates are distinct relationships with own attrs

Can define constraints on relationship sets

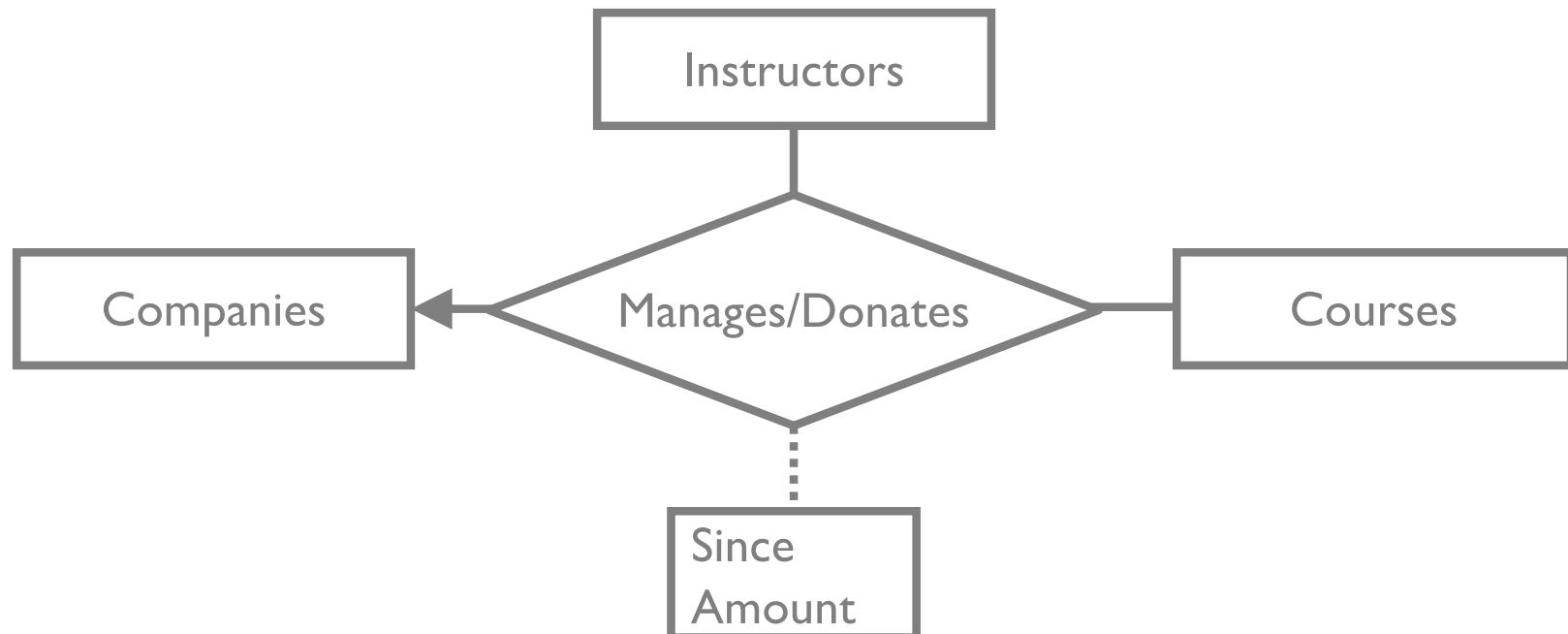
What if we modeled previous slide as ternary relationship?



Aggregation vs Ternary Relationships

Suppose we want to model “A course can have at most one donation”.
We would draw arrow from diamond to Companies.

Actually reads: “Each *instructor, course combination* can have at most one relationship with Companies” e.g., *Eugene and 4111 can have at most one donation, but Alex and 4111 can have another donation*.



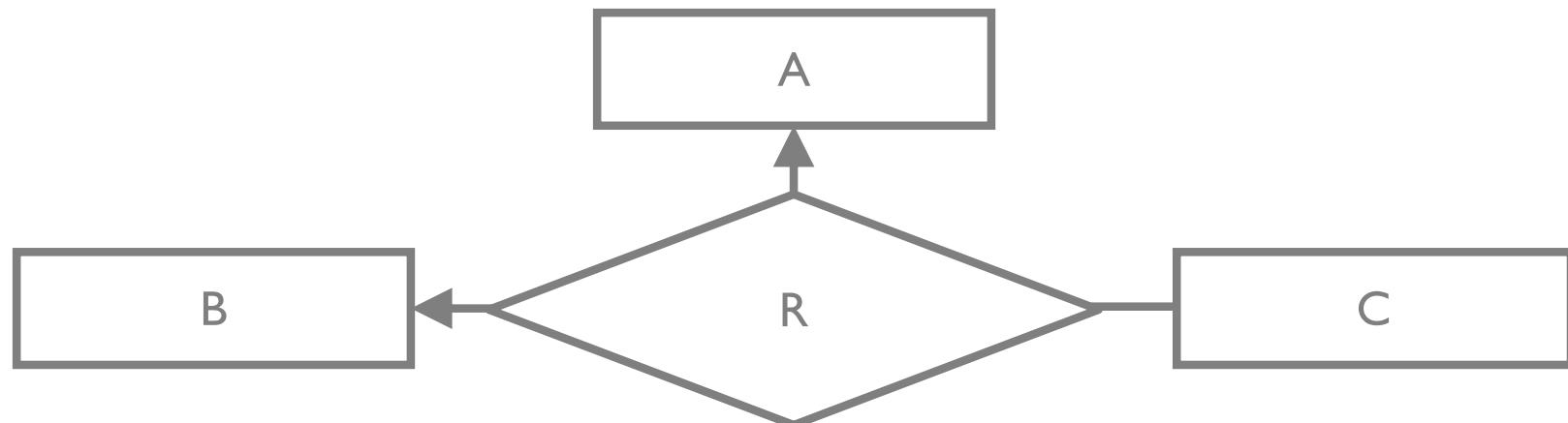
Aggregation vs Ternary Relationships

In general an N-way relationship set can have at most one "at-most-one" constraint (arrow), because multiple constraints (arrows) are ambiguous.

Below could be:

"a C has at most one relationship with a (A, B) pair" OR

"each unique (A,C) pair has at most one relationship with a B, and each unique (B,C) pair has at most one relationship with an A"



Using the ER Model

OK, we've seen the *syntax*.

How to use it involves design choices

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

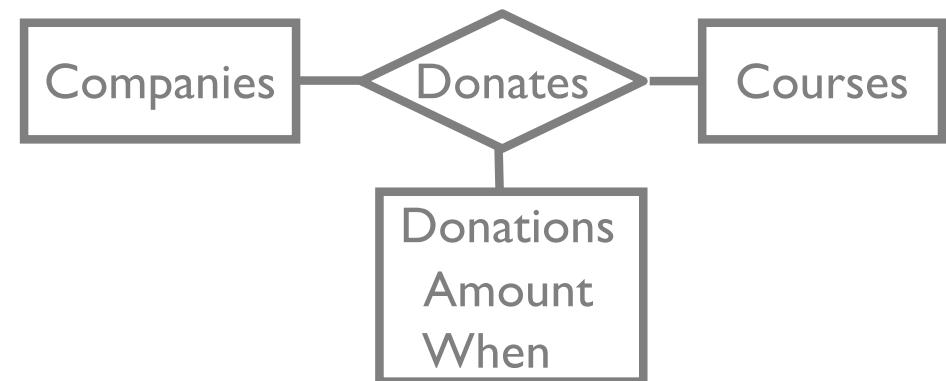
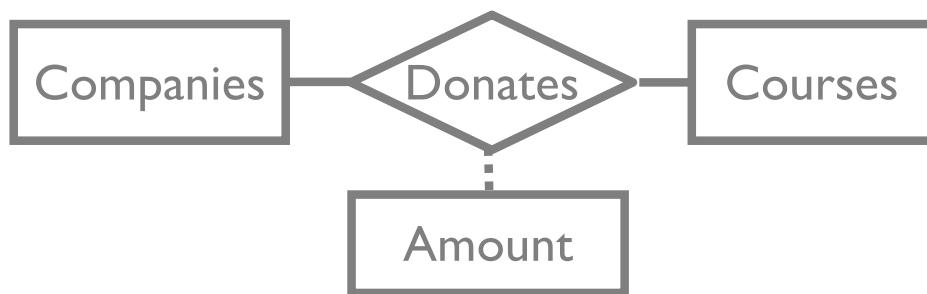
If an address has attrs (structure), must be entity

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

Entity or Attribute?

A company can't donate multiple amounts

Company can make multiple donations

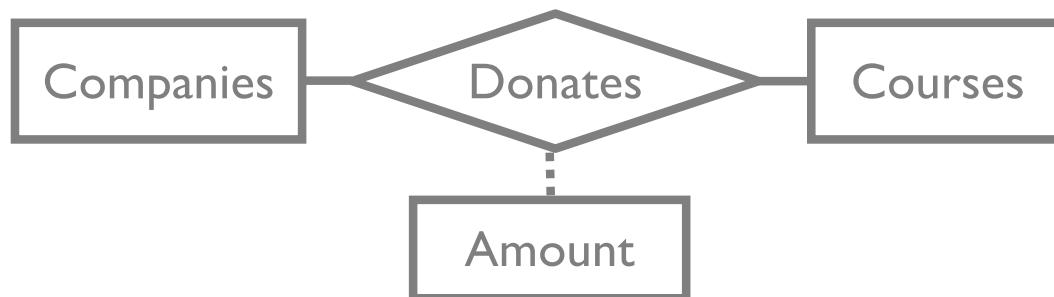


Entity or Relationship?

But 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

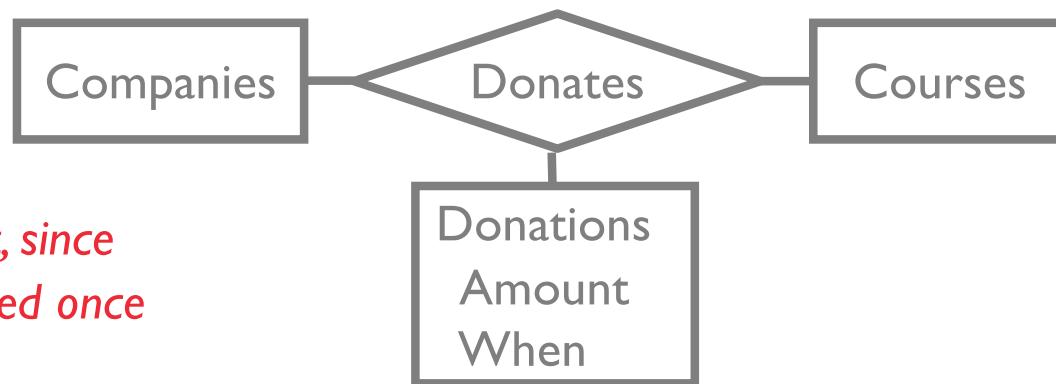


Company	Course	Amount
Amazon	4111	2000
Amazon	4112	2000
Amazon	5111	2000

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



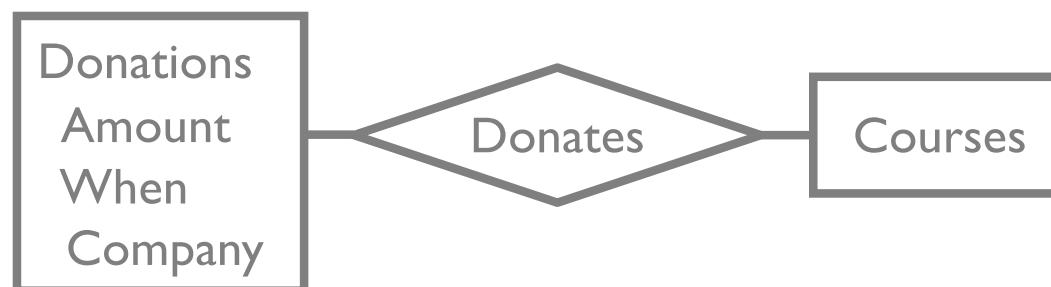
*Company redundant, since
company only donated once*

Company	Course	Donation
Amazon	4111	
Amazon	4112	
Amazon	5111	

Donation	When	Amount
	Today	2000

Entity or Relationship?

If company donates once to school for data related courses.
Refactor amount into an entity

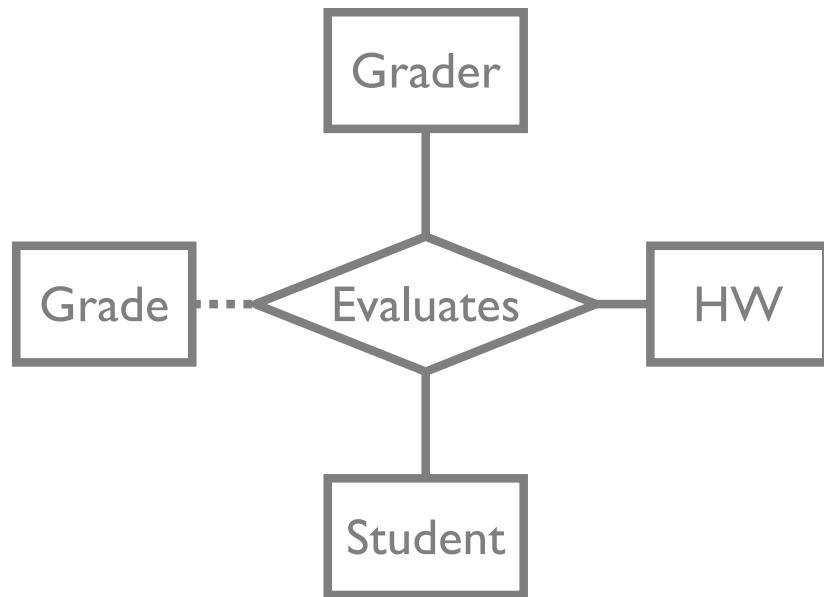


Course	Donation
4111	
4112	
5111	

Donation	When	Amount	Company
	Today	2000	Amazon

Binary or Ternary Relationship?

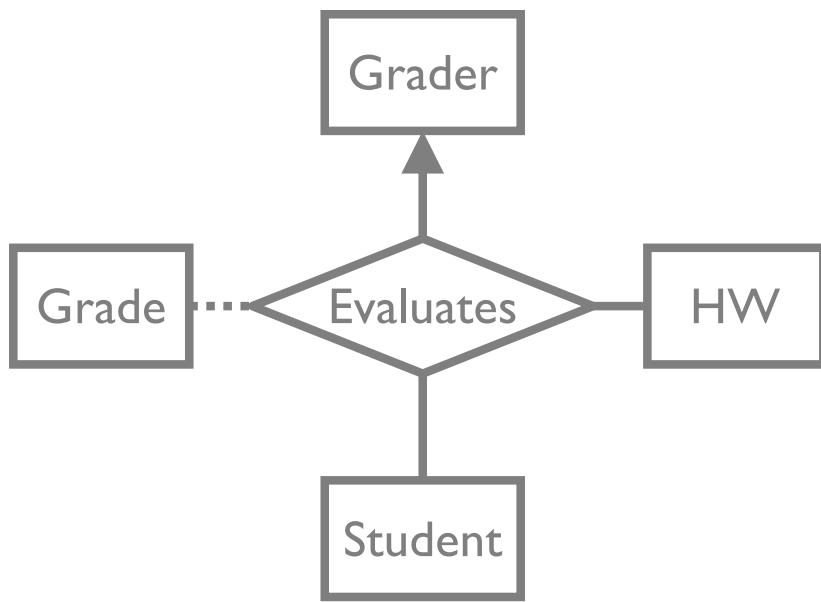
HW means a particular released HW, not a submission
What if each HW has at most one grader? (next slide)



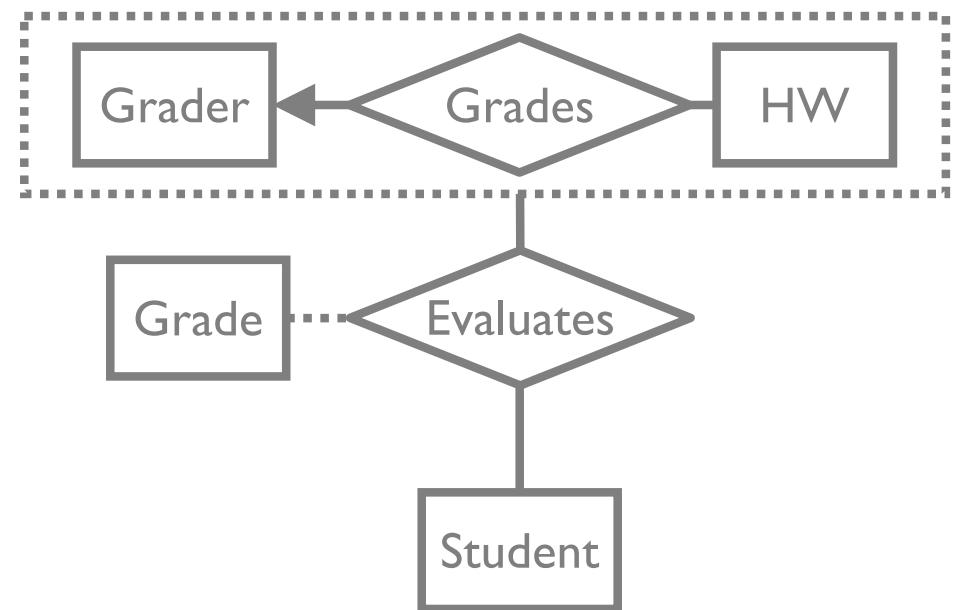
Binary or Ternary Relationship?

What if each HW has at most one grader?

Option 1: add arrow from evaluates to grader.



Option 2: aggregation



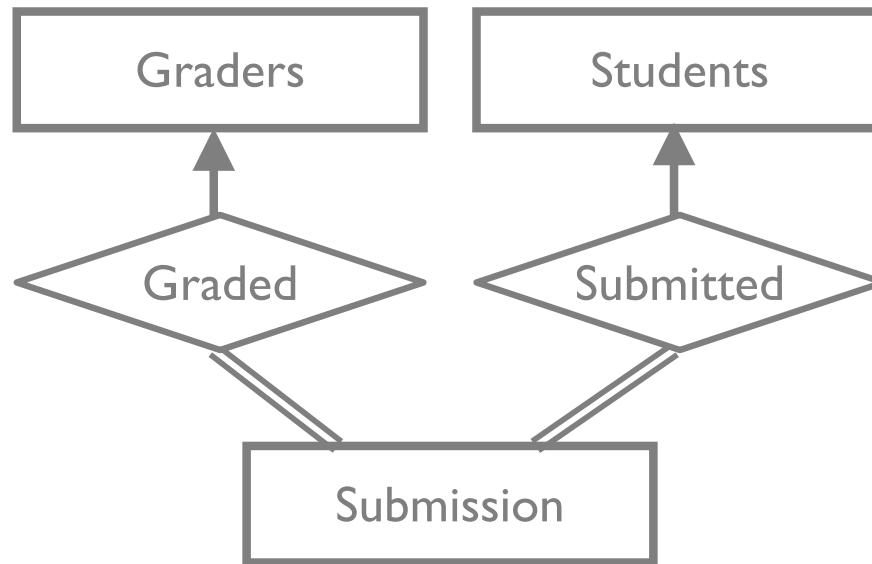
Actually says that each student's HW submission (hwid, studentid) has at most one grader

Each HW has at most 1 grader and the grader evaluates each student

Binary or Ternary Relationship?

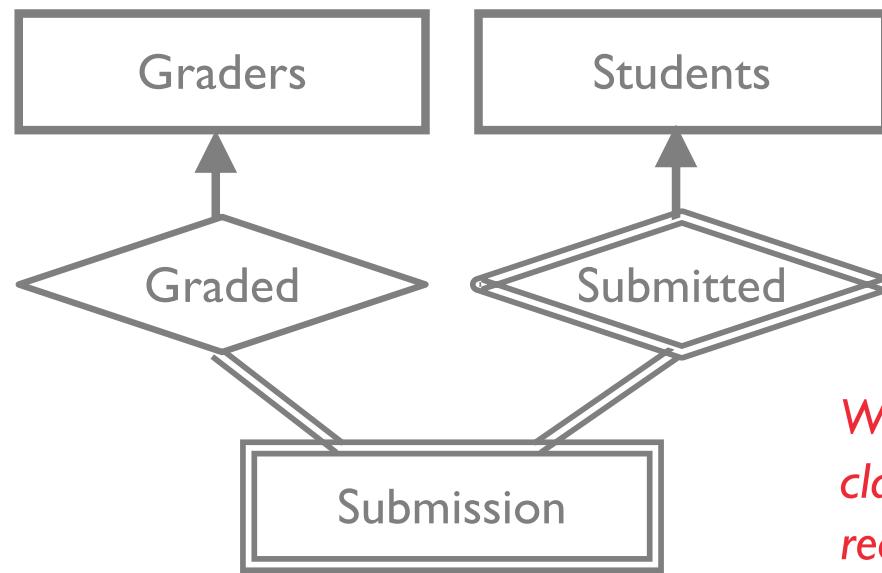
Binary relationships allows additional constraints

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



Binary or Ternary Relationship?

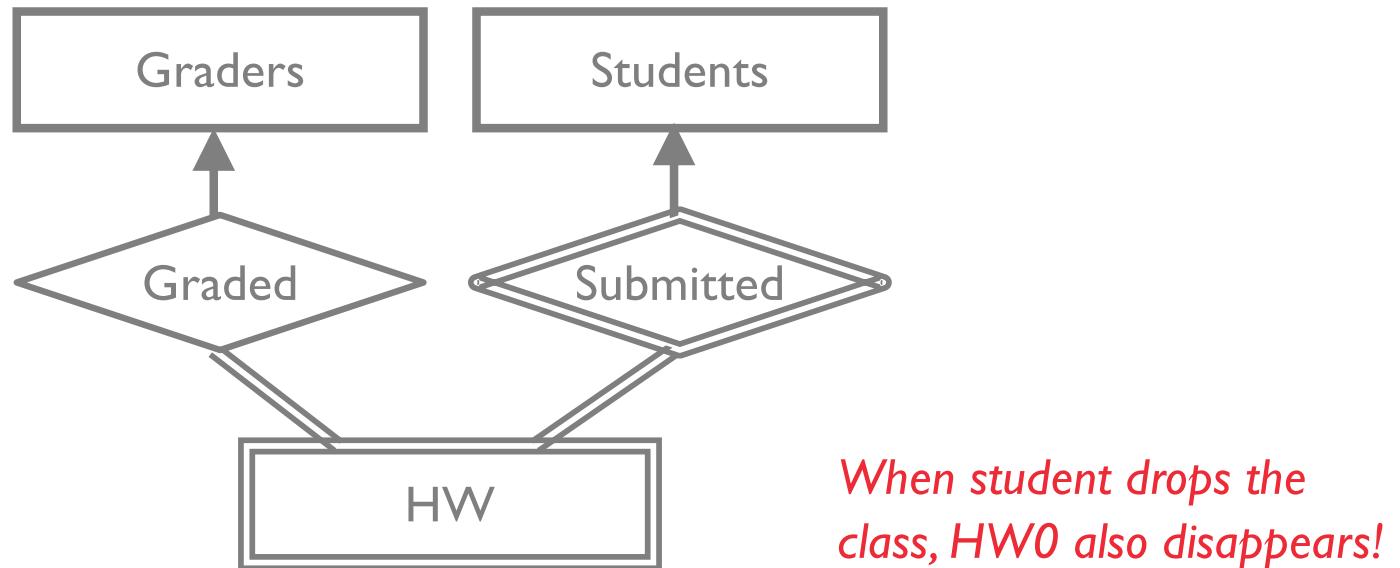
Binary relationships allows additional constraints



*When student drops the class, their submission records also are removed.
Makes sense!*

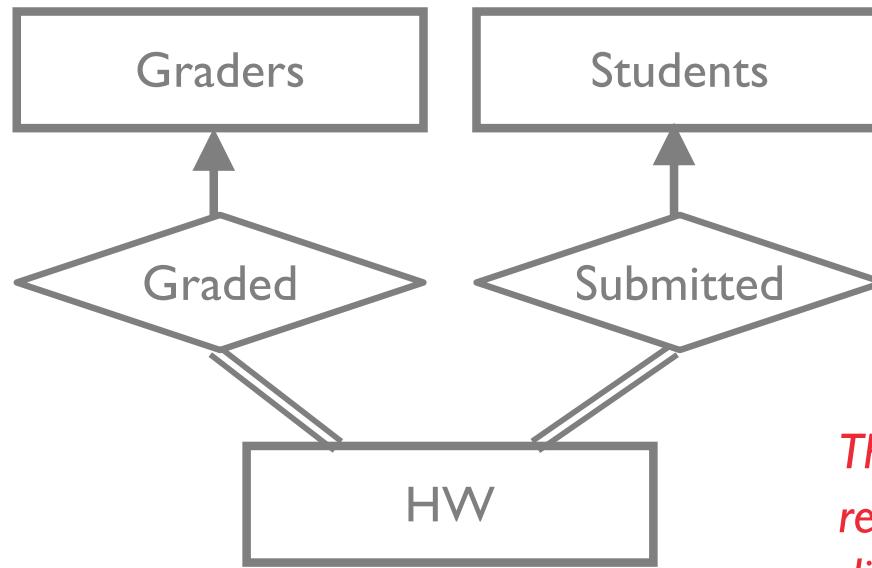
Binary or Ternary Relationship?

Binary relationships allows additional constraints
What if we model HW instead of Submission?



Binary or Ternary Relationship?

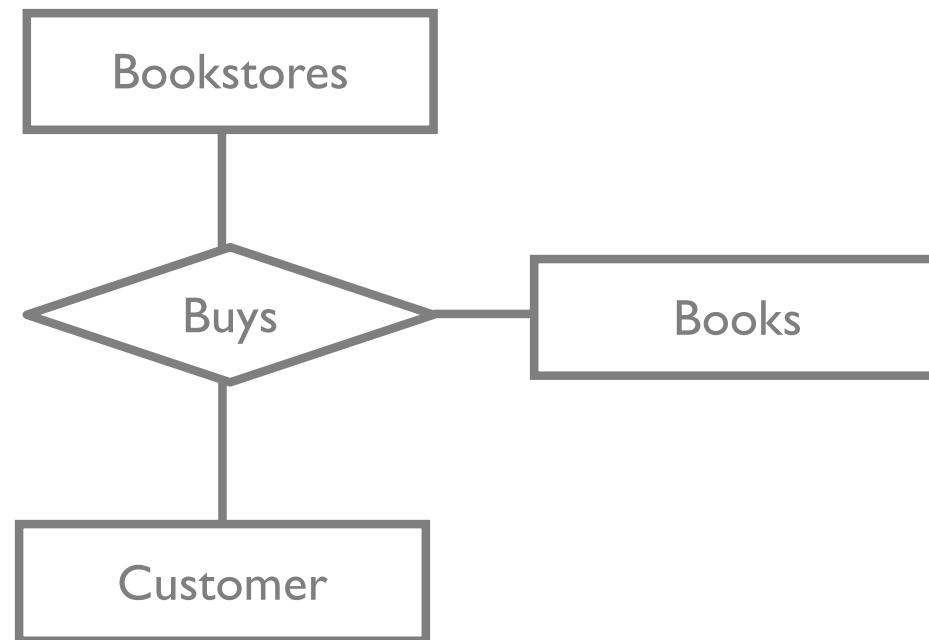
Binary relationships allows additional constraints
What if we model HW instead of Submission?



This is correct, since the relationship "submitted" will disappear if student is removed

Binary or Ternary Relationship?

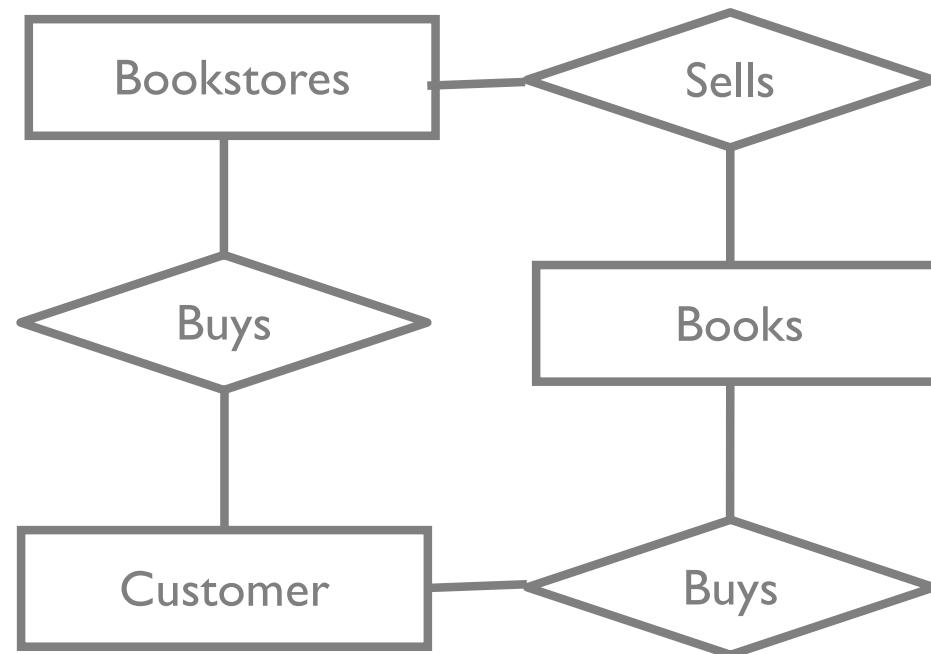
Sometimes have true ternary relationship that is defined by all three entities.



Binary or Ternary Relationship?

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

*Doesn't
Really
Work*



Advice

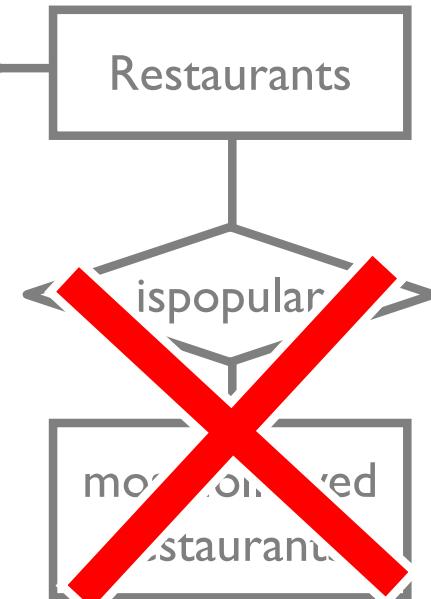
The ER diagram (and database) stores the *minimal information* needed for your application.

Everything else (e.g., stats) can be computed



Most followed restaurants computable
from Users, Restaurants, and Follows.

May still store in DB for *performance*
reasons



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

Still needs further refinement

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