Administrivia

Schedule

	Date	Topic	Assigned	Due
L1	3-Sep	**Intro and Overview**	HW 0 Look for teammates	
L2	5-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 (9/8 11:59PM EST. NO LATE DAYS)
L3	10-Sep	**ER Models** optional: Textbook Chapter 6 except for Sections 6.7, 6.10, and 6.11.		HW 1 Part 1 (9/11 11:59PM EST) Formed Project 1 Team (no submission)

Staff office hours will be up this weekend

Zoom links in discussion board

HW0 Due Soon!

Read the instructions

Course Expectations

Copy each of the following statements into the associated answer box to acknowledge that you will abide by these expectations.
Include the full statement but do NOT include the "*" at the end.

I understand that students can receive a failing grade in the course if the staff find evidence of academic dishonesty.

26 responses

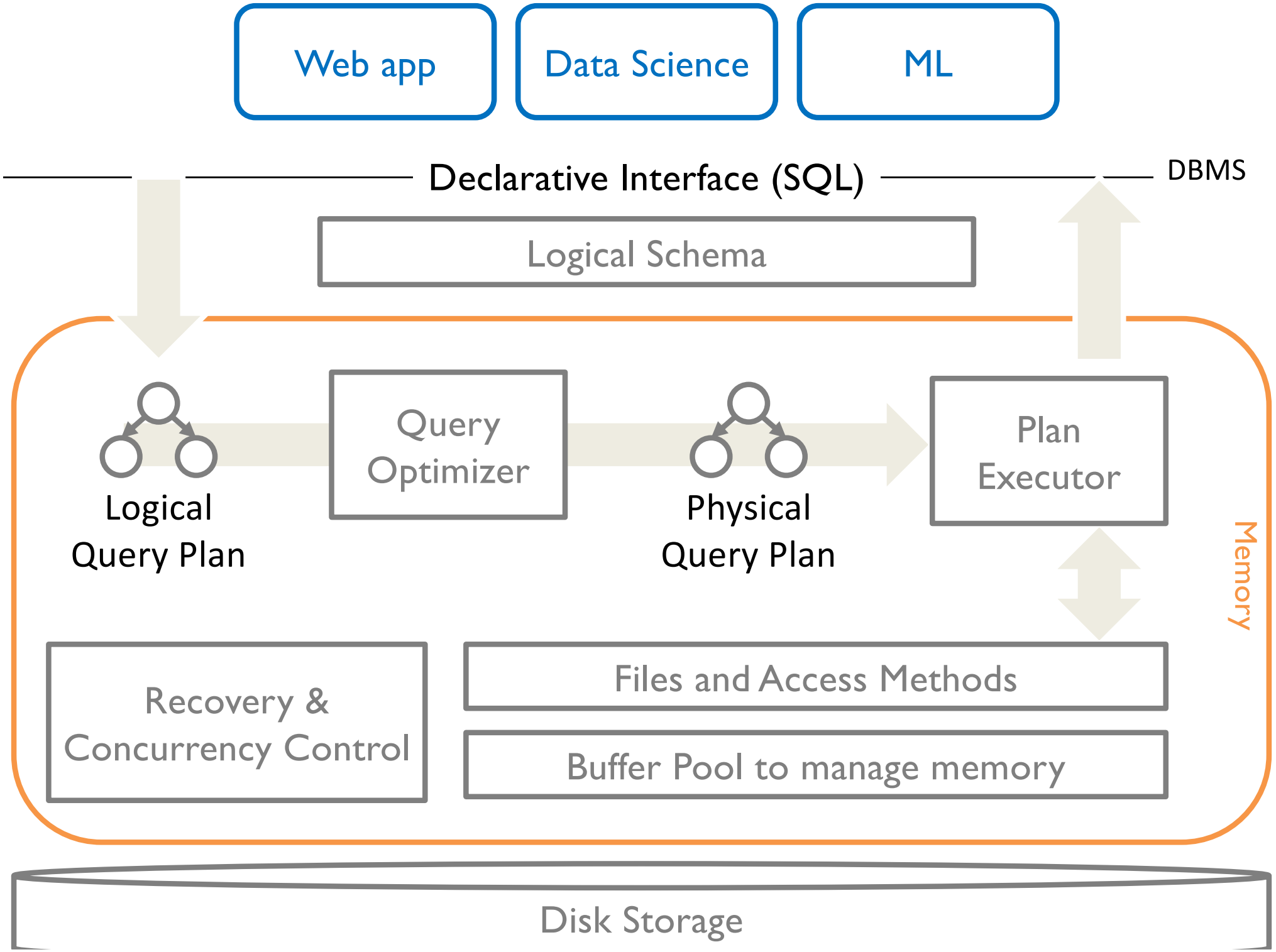
HW1 out today

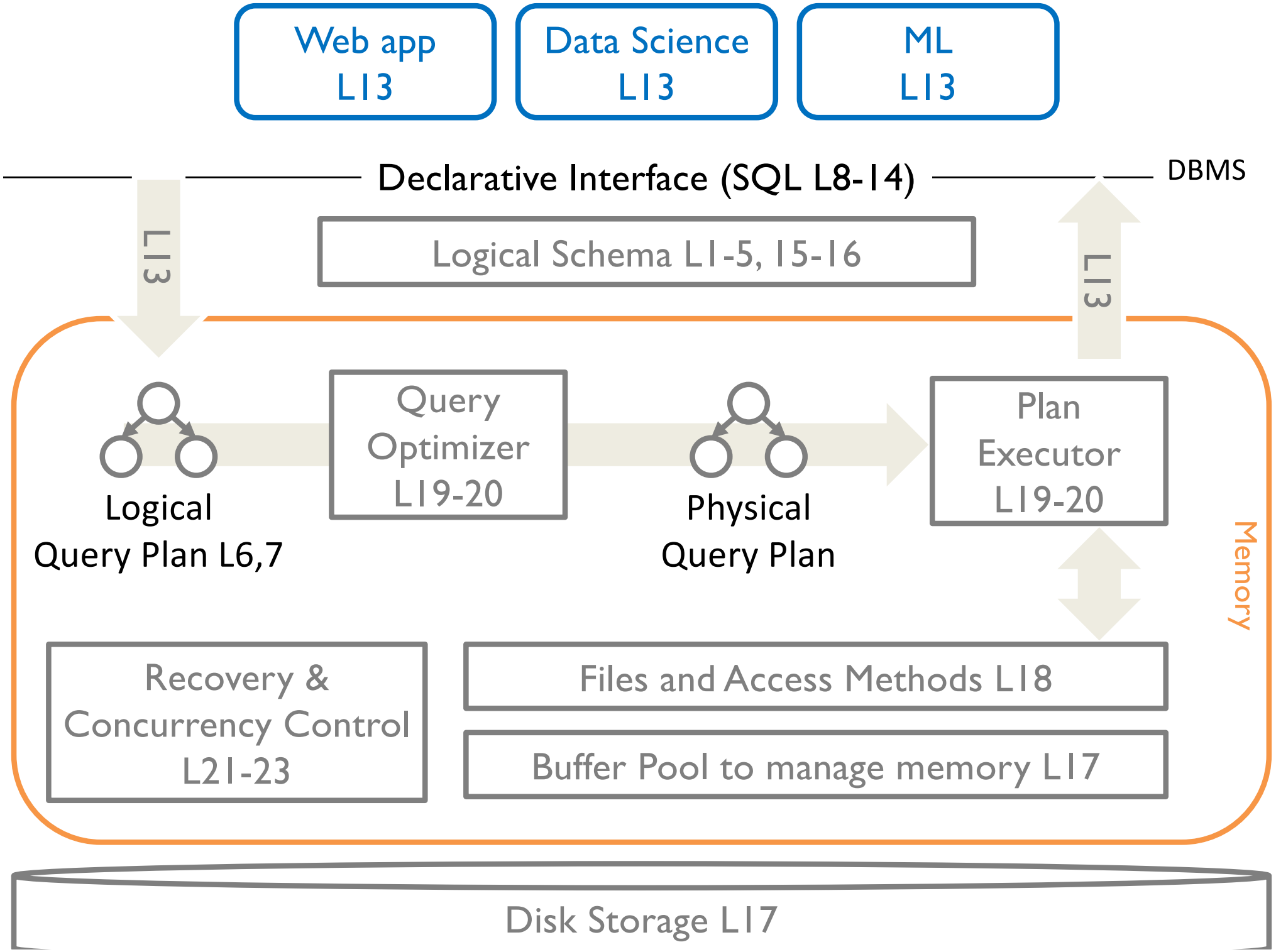
Project 1 Part 1 out today

Find a project 1 teammate *ASAP*!

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

Course Overview


Web app
Data Science
ML
Declarative Interface (SQL)
DBMS
Logical Schema
Logical
Query Plan
Query
Optimizer
Physical
Query Plan
Plan
Executor
Memory
Recovery &
Concurrency Control
Files and Access Methods
Buffer Pool to manage memory
Disk Storage


Web app
L13
Data Science
L13
ML
L13
Declarative Interface (SQL L8-14)
DBMS
L13
Logical Schema L1-5, 15-16
L13
Query
Optimizer
L19-20
Plan
Executor
L19-20
Logical
Query Plan L6,7
Physical
Query Plan
Memory
Recovery &
Concurrency Control
L21-23
Files and Access Methods L18
Buffer Pool to manage memory L17
Disk Storage L17

COMS W4111 - Intro to Databases

Prerequisites: CS3137 or CS3134; fluency in Python

Intro to DBMSes

Data Models

Relational Algebra

SQL

Applications + SQL

Normalization

Peek at DBMS internals:

- Storage and indexing
- Query optimization
- Transaction Processing

COMS W4112-Database Sys. Impl.

Prerequisites: CS3137 or CS3134; fluency in Python

Components of a Database System in Detail

Storage Methods and Indexing

Query Processing and Optimization

Materialized Views

Transaction Processing and Recovery

Parallel & Distributed DBMSes

Performance Considerations Beyond Disk I/Os

COMS E6111-Advanced Databases

Prerequisites: CS4111; fluency in Java or Python

Information Retrieval

Information Extraction

Web Search

Data Mining

Data Warehousing, OLAP, Decision Support

COMS E6xxx-Graduate Seminars

Prerequisites: CS4111; fluency in Java or Python

6113 Database Research Topics

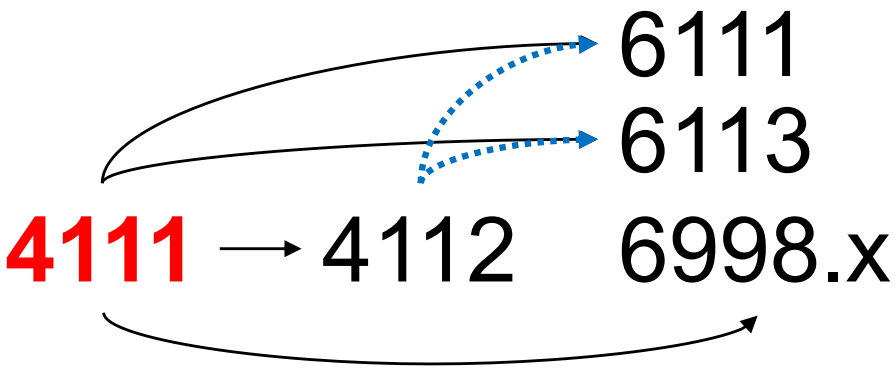
Topics vary e.g., modern databases, ML + Data

w6113.github.io

6998.002 Systems for Human Data Interaction

Topics combine HCI, visualization, and databases

columbiaviz.github.io


6111
6113
4111
4112
6998.x

On-going Feedback

C O L U M B I A U N I V E R S I T Y C O M S W 4 1 1

INTRODUCTION TO DATABASES

Information

- Tues/Thurs
  8:40-9:55AM
  301 Uris Hall
  3 units
- Syllabus
- Ed Discussion
- Provide Feedback
- Course Github
- Course Gradescope

Overview

The goal of this class is two-fold. First, to introduce you to core database concepts (design, SQL) so that you too can build a billion dollar application. Second, to teach internals (e.g., physical database design, query optimization, transaction processing why queries may be running slowly/incorrectly. We will also discuss their relevance

The Data Management Seminar invites interesting database researchers and practi or on zoom (if available). We will announce these periodical

Announcements

Schedule

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

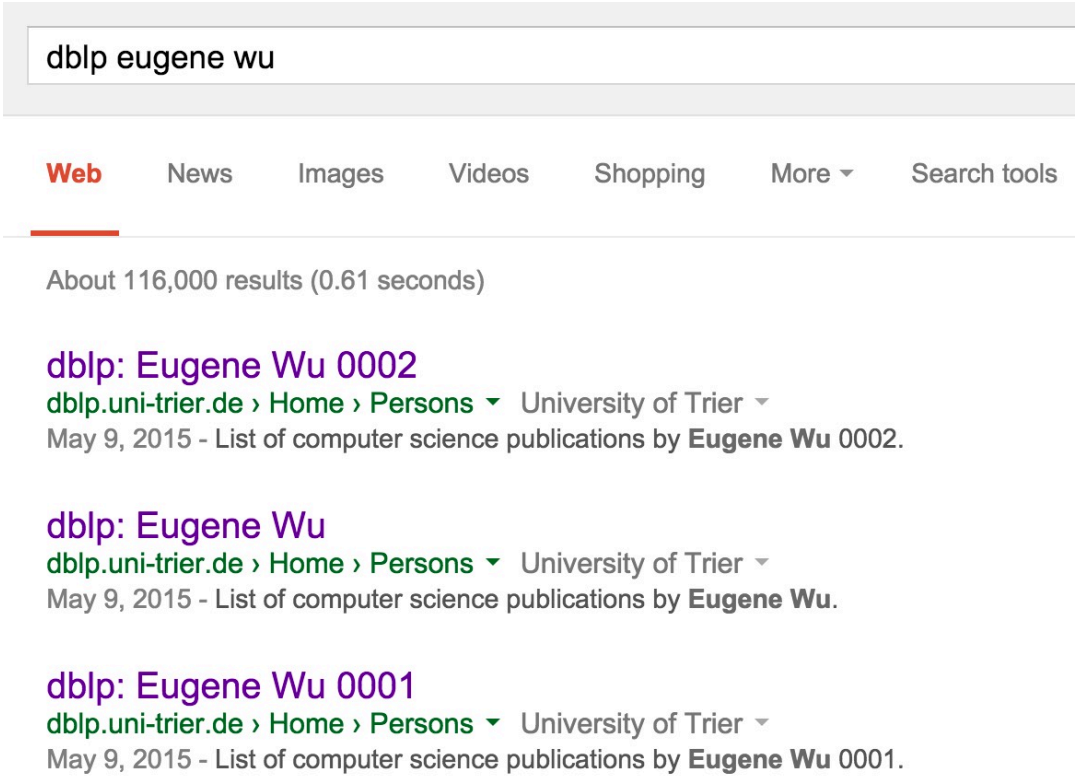
- ~10k different information (data) systems
- 90% relational databases (DBMSes)
- Typical database has >100 tables
- Typical table has 50 – 200 attributes

Inconsistencies/Constraint Violations

Huge amount of effort to avoid inconsistencies

Can data model help us avoid automatically?

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

www.cs.columbia.edu › ~ewu

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.

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 ...

PROFILE | Finance & Investments

#39 Eugene Wu

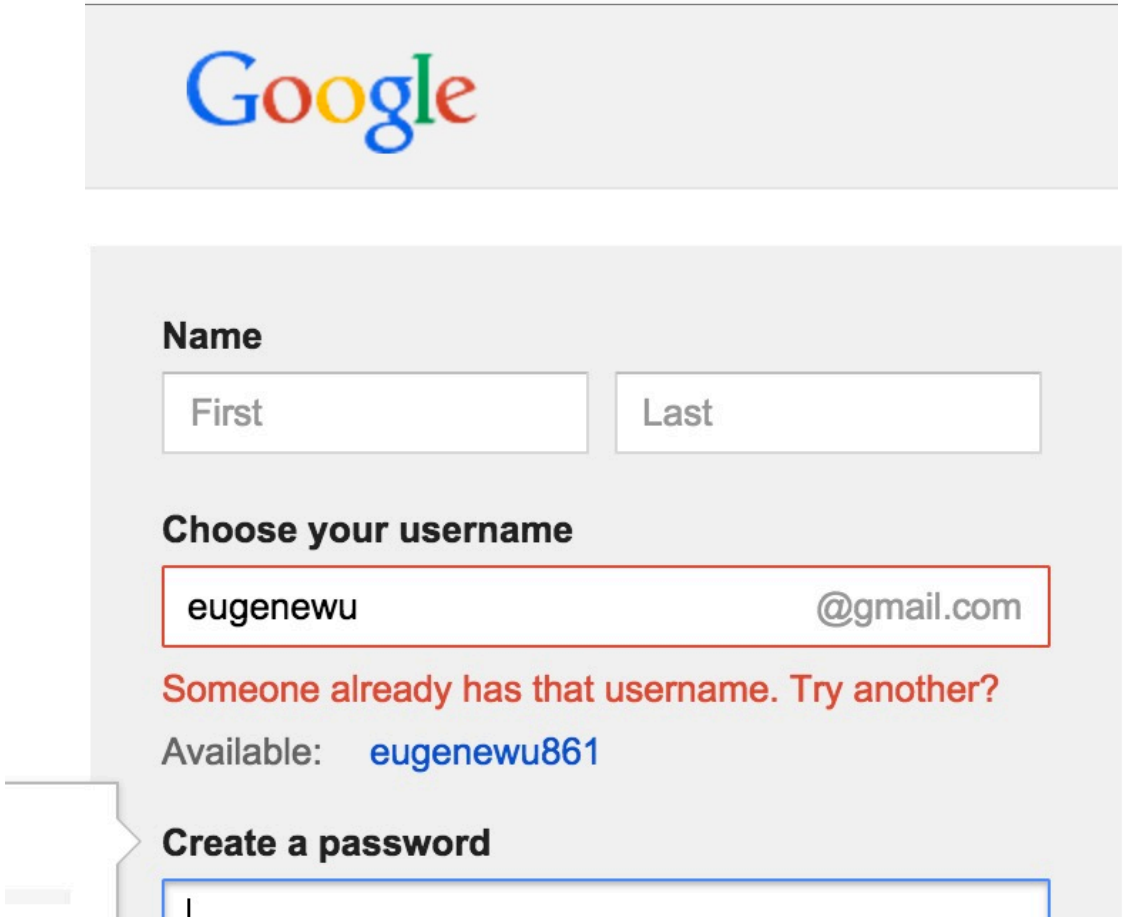
$1.4B REAL TIME NET WORTH
as of 1/27/22

▲ $11 M | 0.77%
Reflects change since 5 PM ET of prior trading day

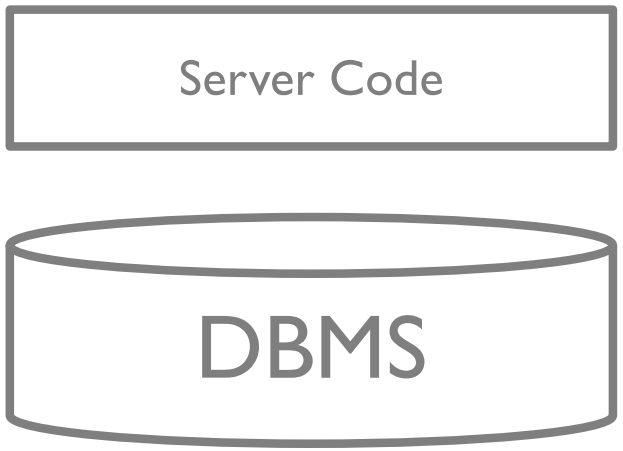
Not Teaching 4!!!!!!

Inconsistencies/Constraint Violations

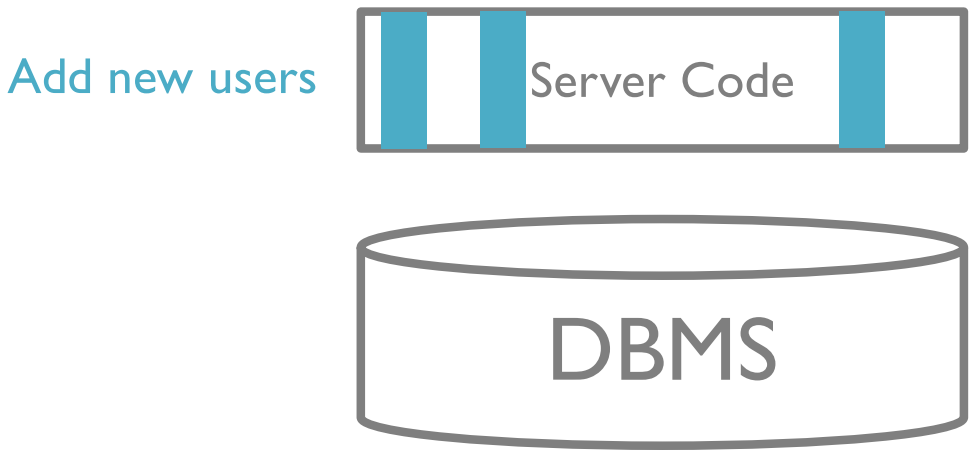
Check in application code!



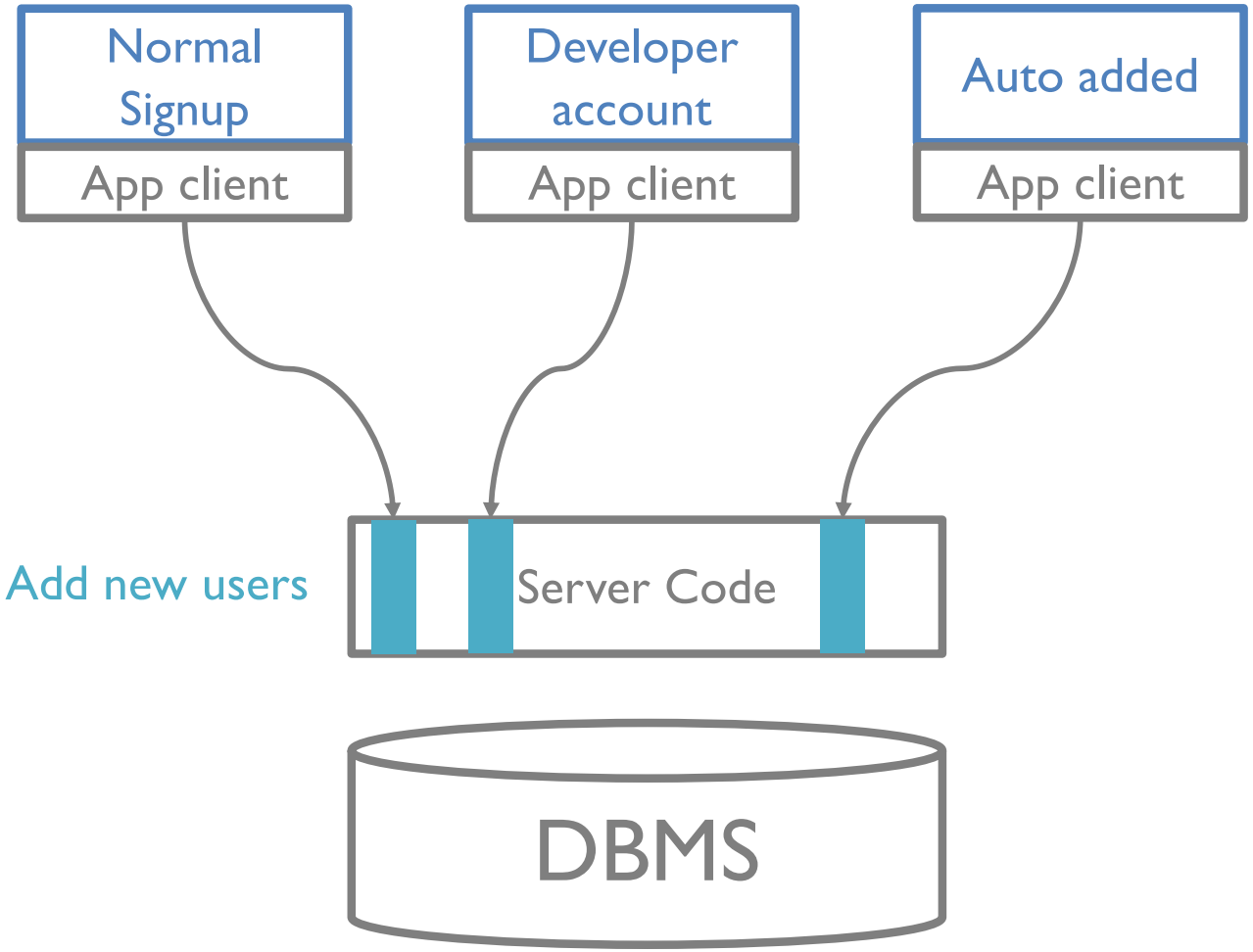
It is Hard to Design Applications



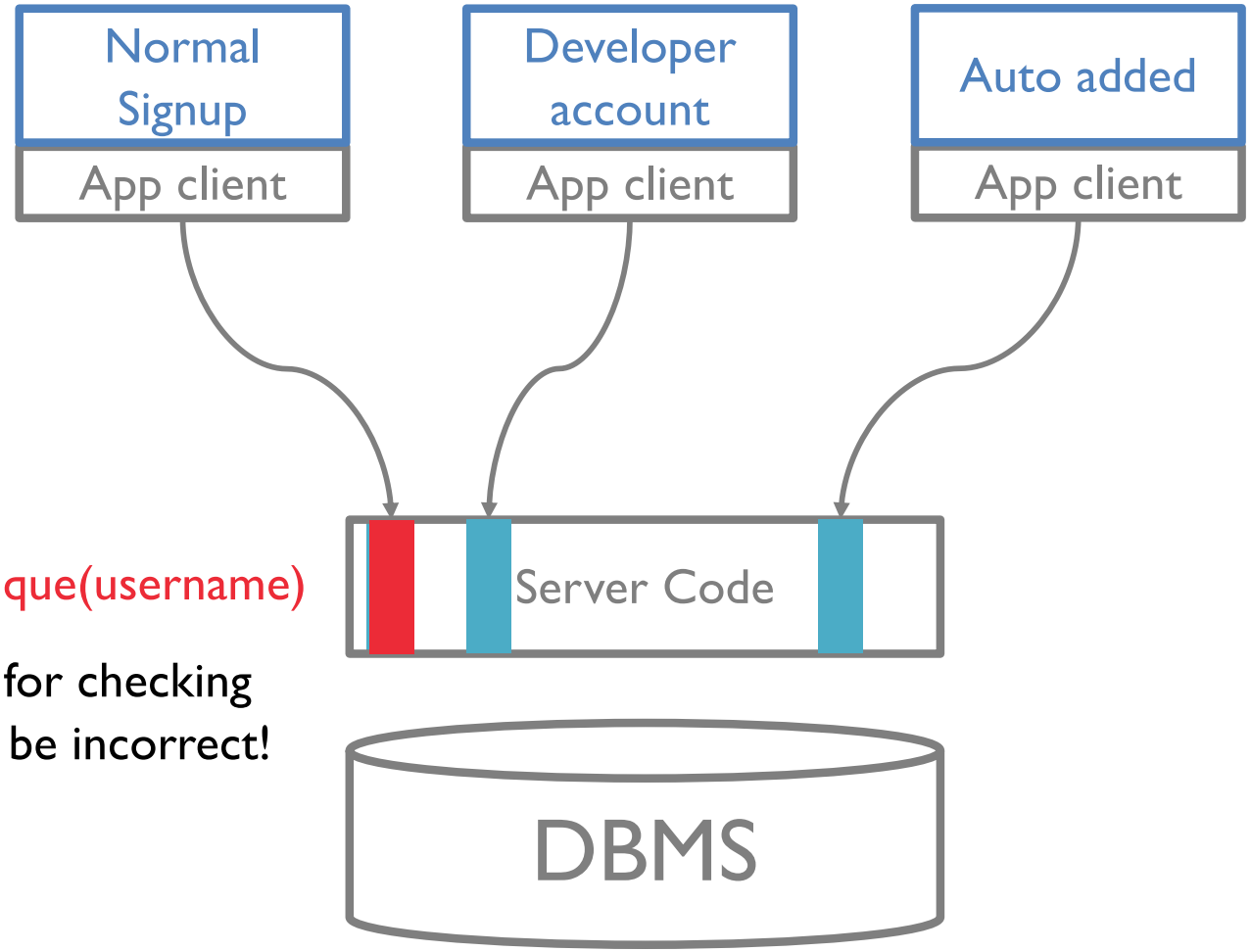
It is Hard to Design Applications



It is Hard to Design Applications



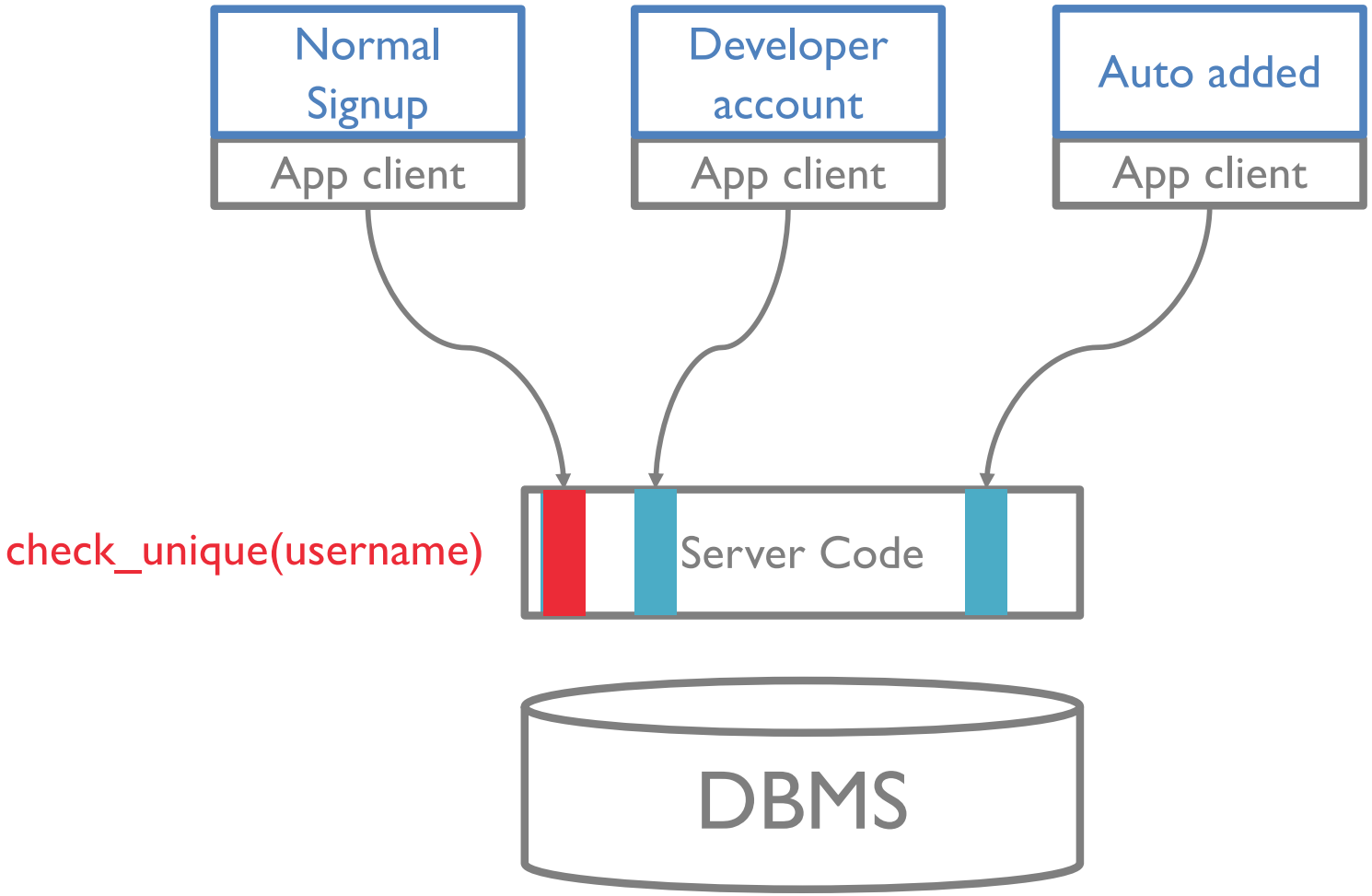
It is Hard to Design Applications



check_unique(username)

The code for checking this could be incorrect!

It is Hard to Design Applications



ER Diagrams

What is it?

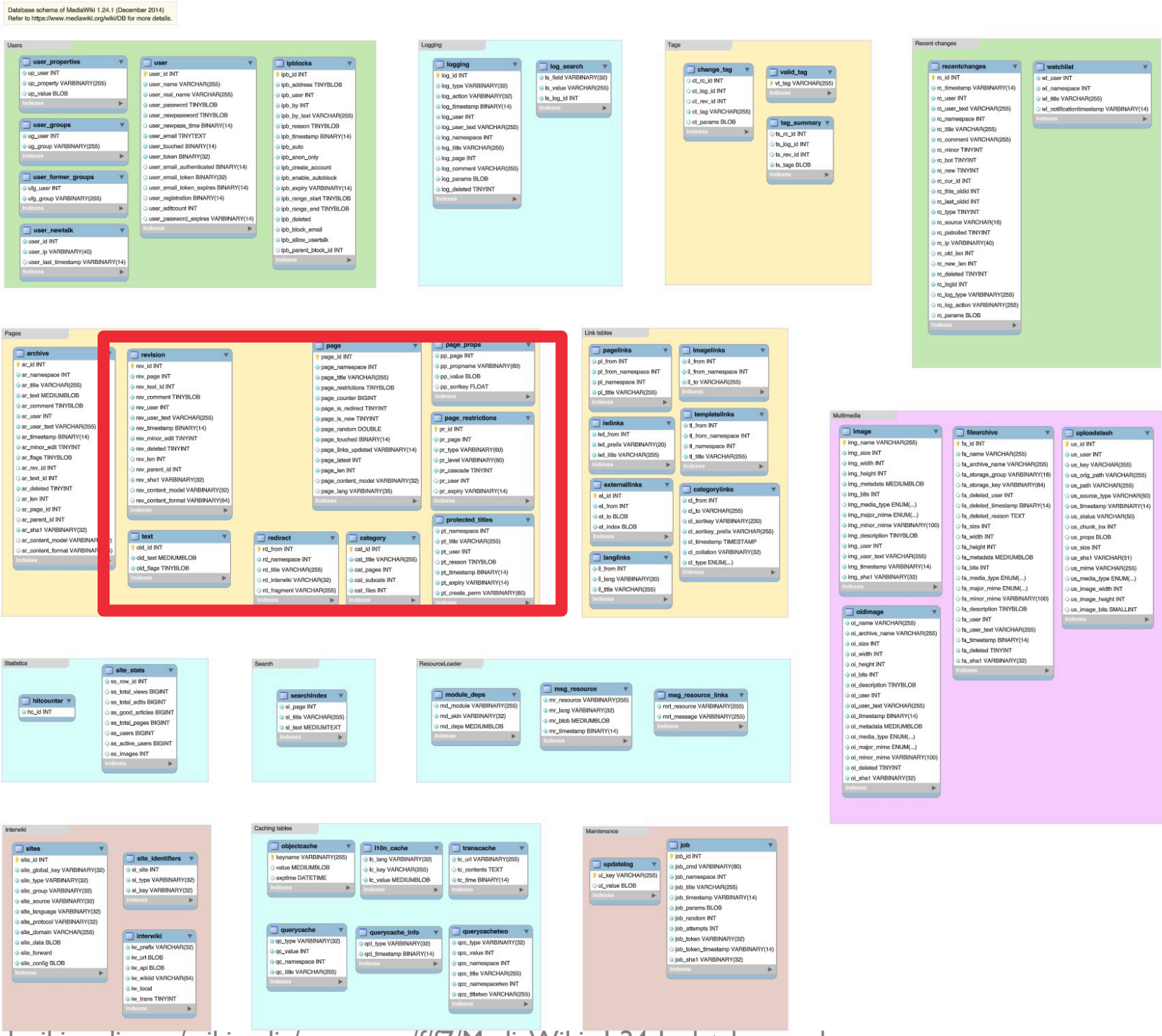
- Draws the core info your database will eventually store.
- Visually encodes important constraints

Who cares?

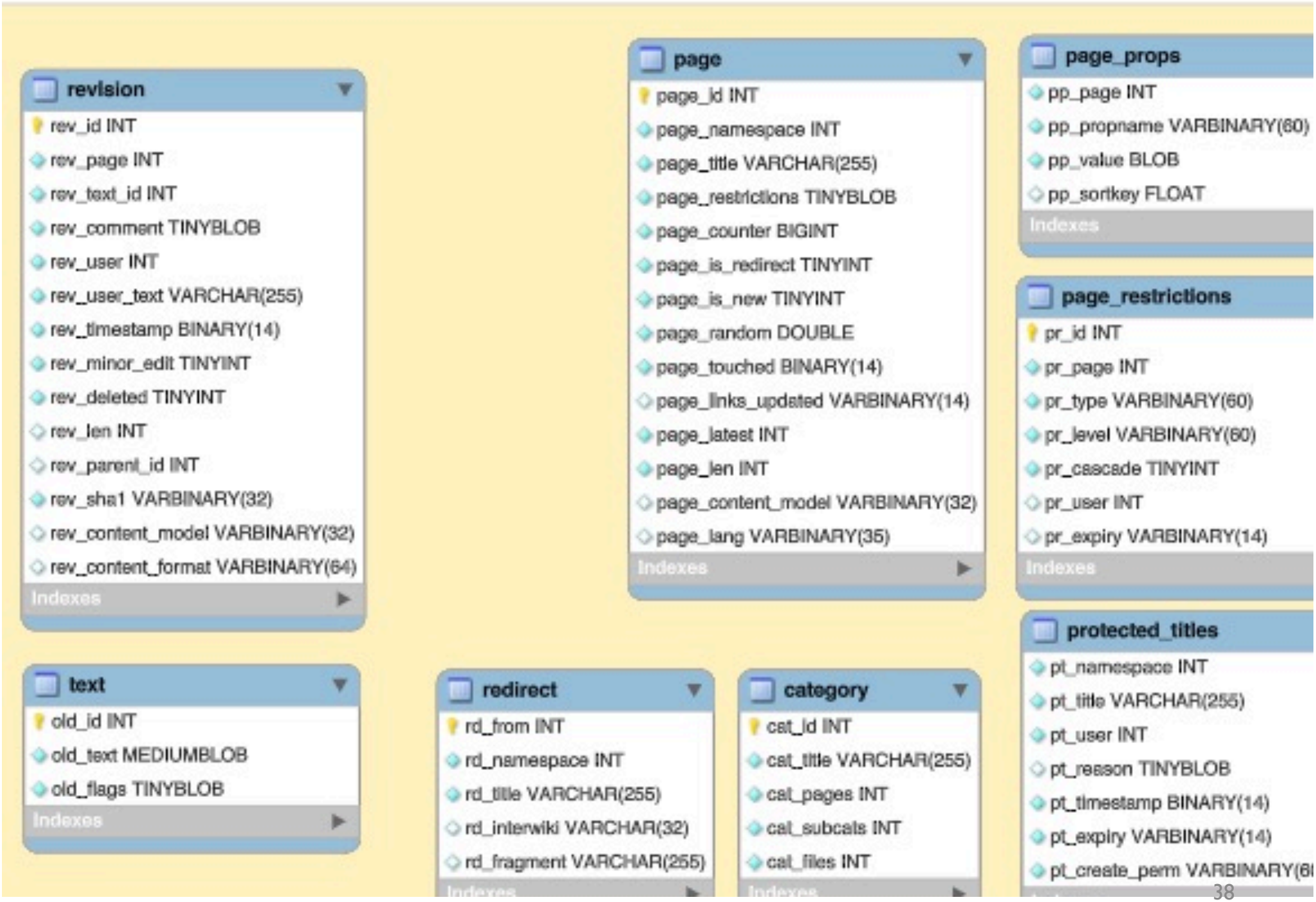
- Good for "white boarding" together
- Good way to share the "gist" of your DB's structure
- Good for thinking

```
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'::
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 |           |          |
```

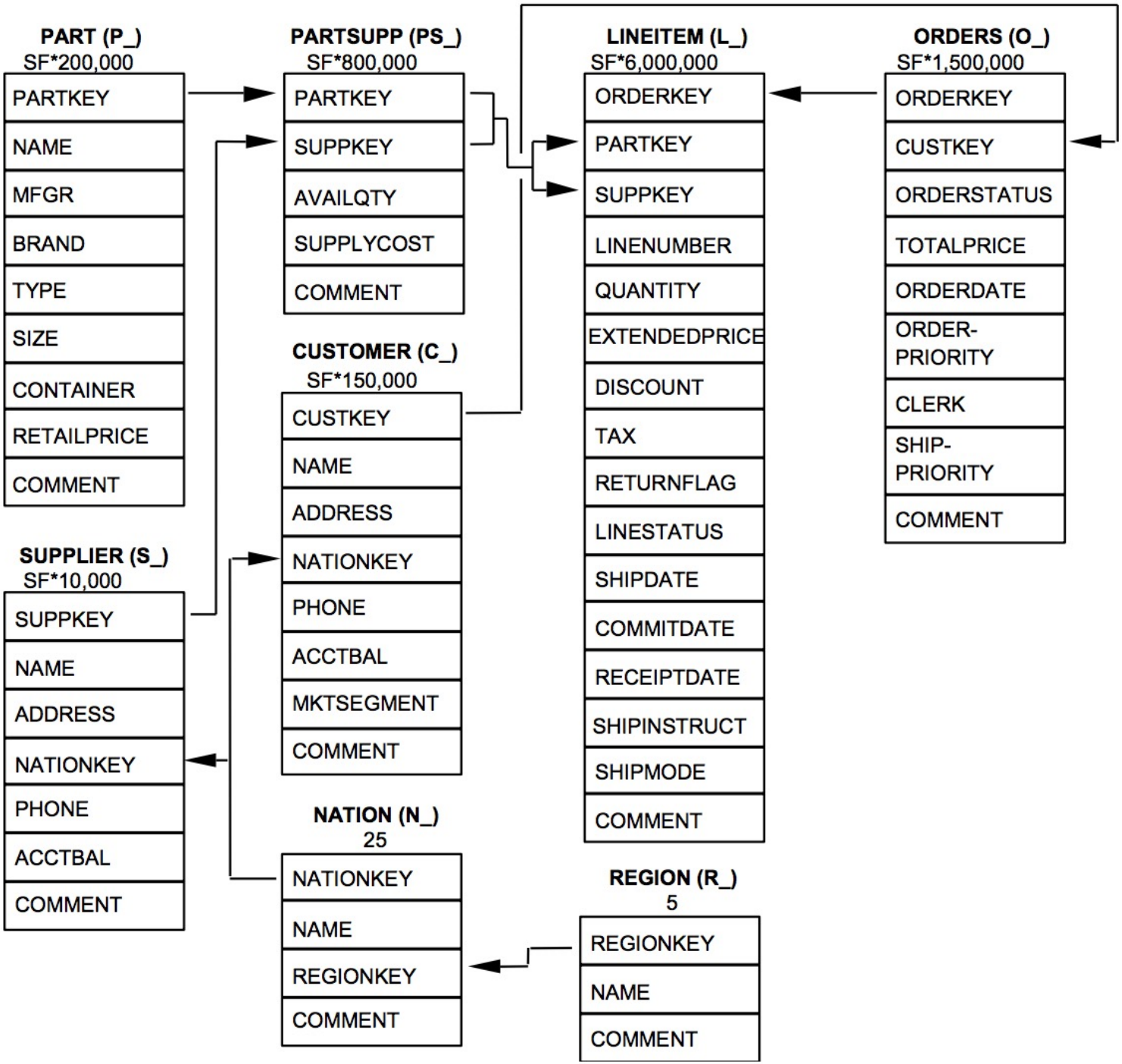


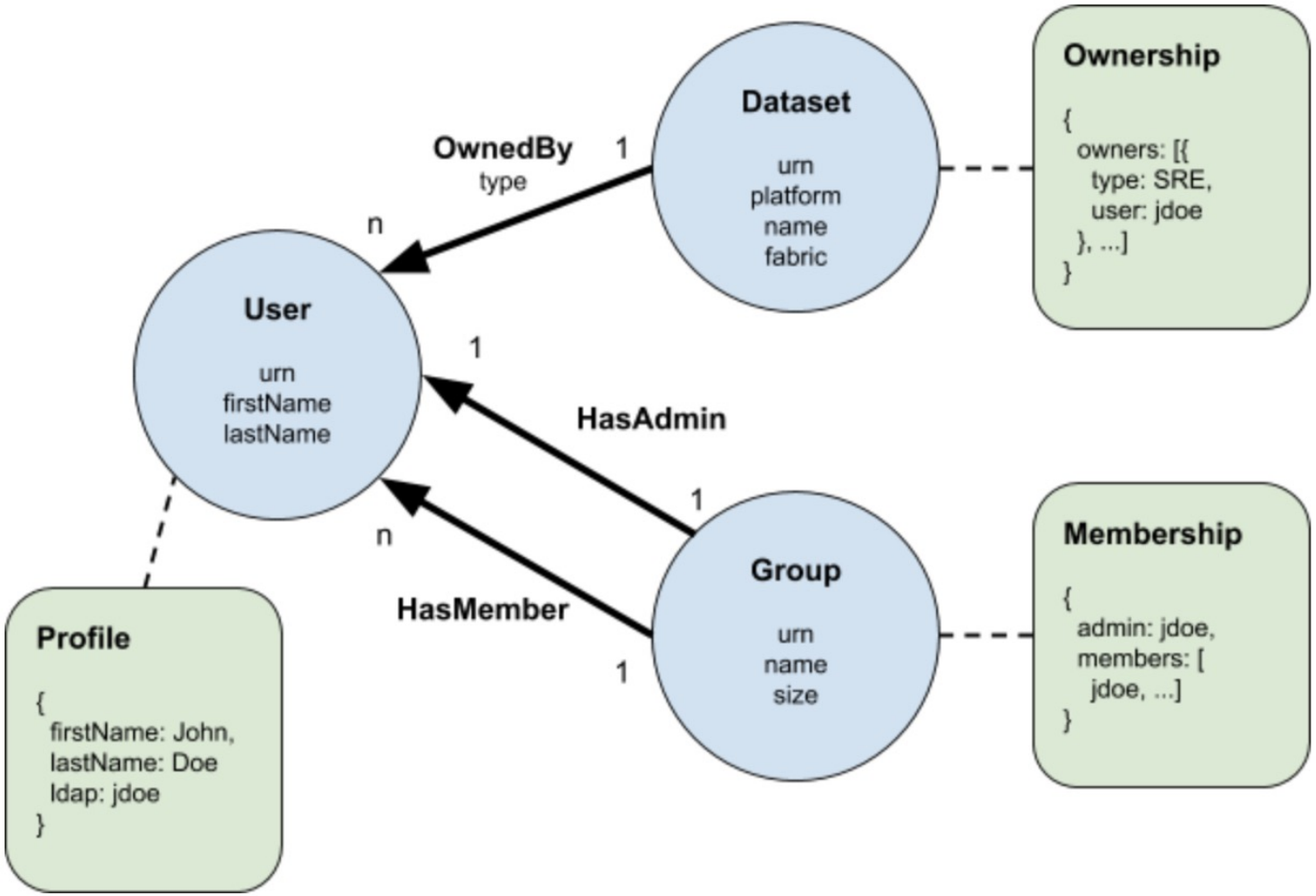
https://upload.wikimedia.org/wikipedia/commons/f/f7/MediaWiki_1.24.1_database_schema.svg



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

Figure 2: The TPC-H Schema





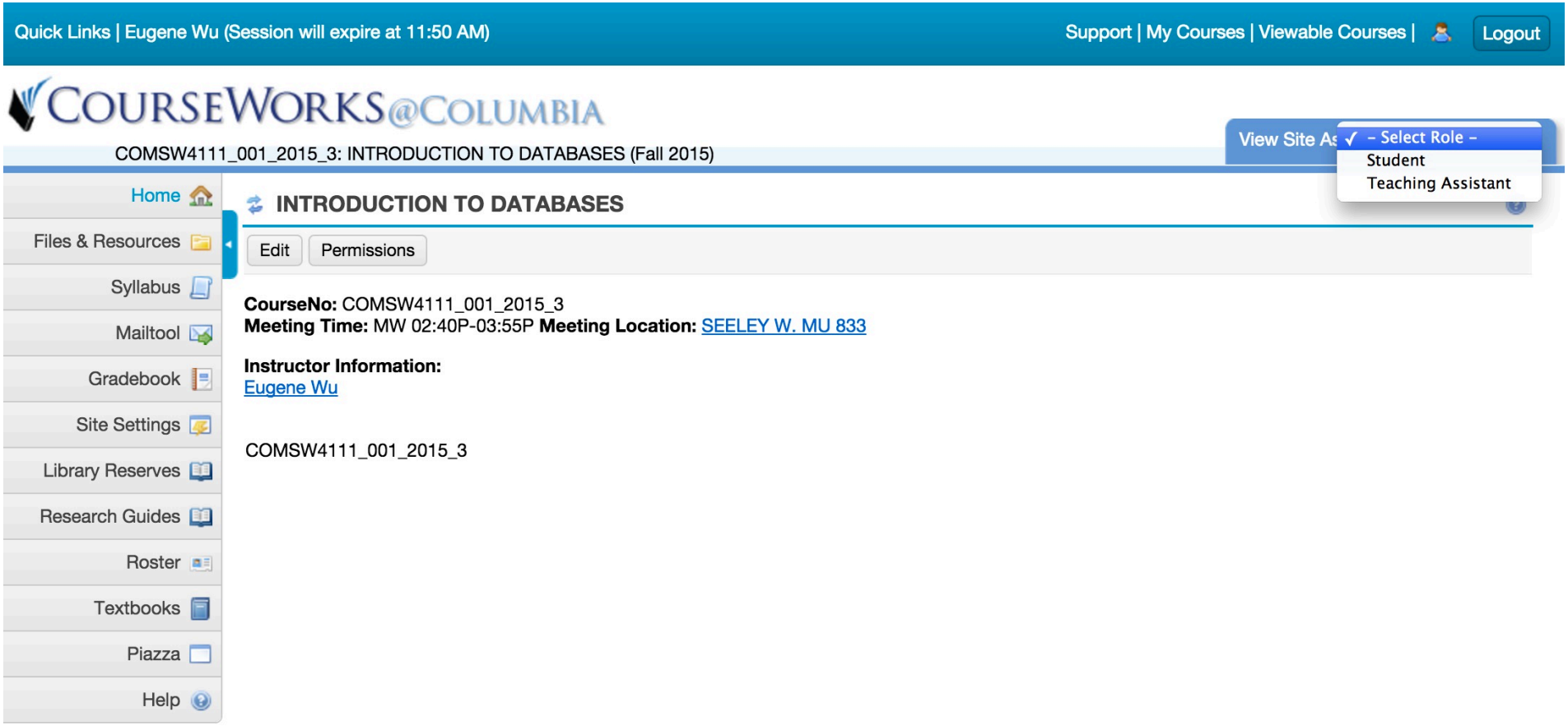
https://engineering.linkedin.com/blog/2019/data-hub

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)

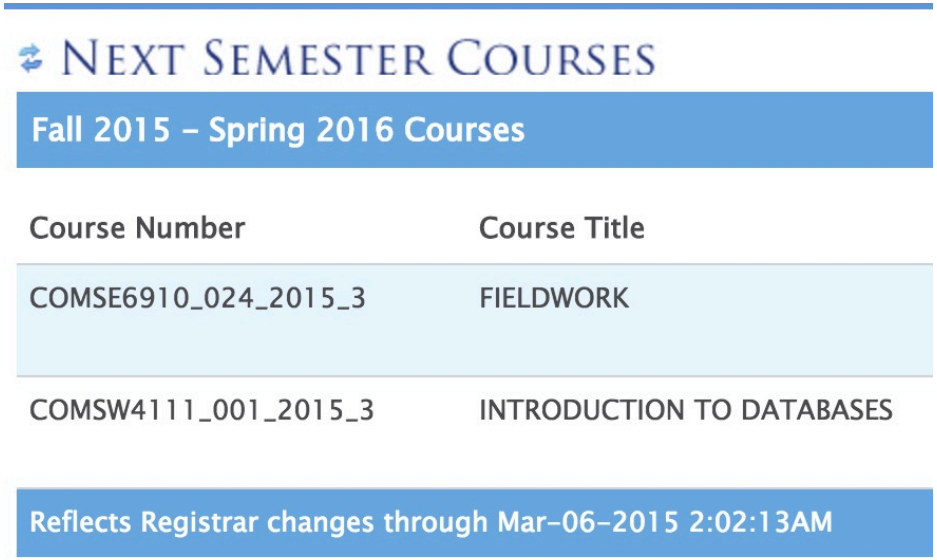

Quick Links | Eugene Wu (Session will expire at 11:50 AM)
Support | My Courses | Viewable Courses |
Logout
COURSEWORKS@COLUMBIA
COMSW4111_001_2015_3: INTRODUCTION TO DATABASES (Fall 2015)
View Site As
– Select Role –
Student
Teaching Assistant
Home
Files & Resources
Syllabus
Mailtool
Gradebook
Site Settings
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 their attrs.

Integrity constraints & business rules



NEXT SEMESTER COURSES

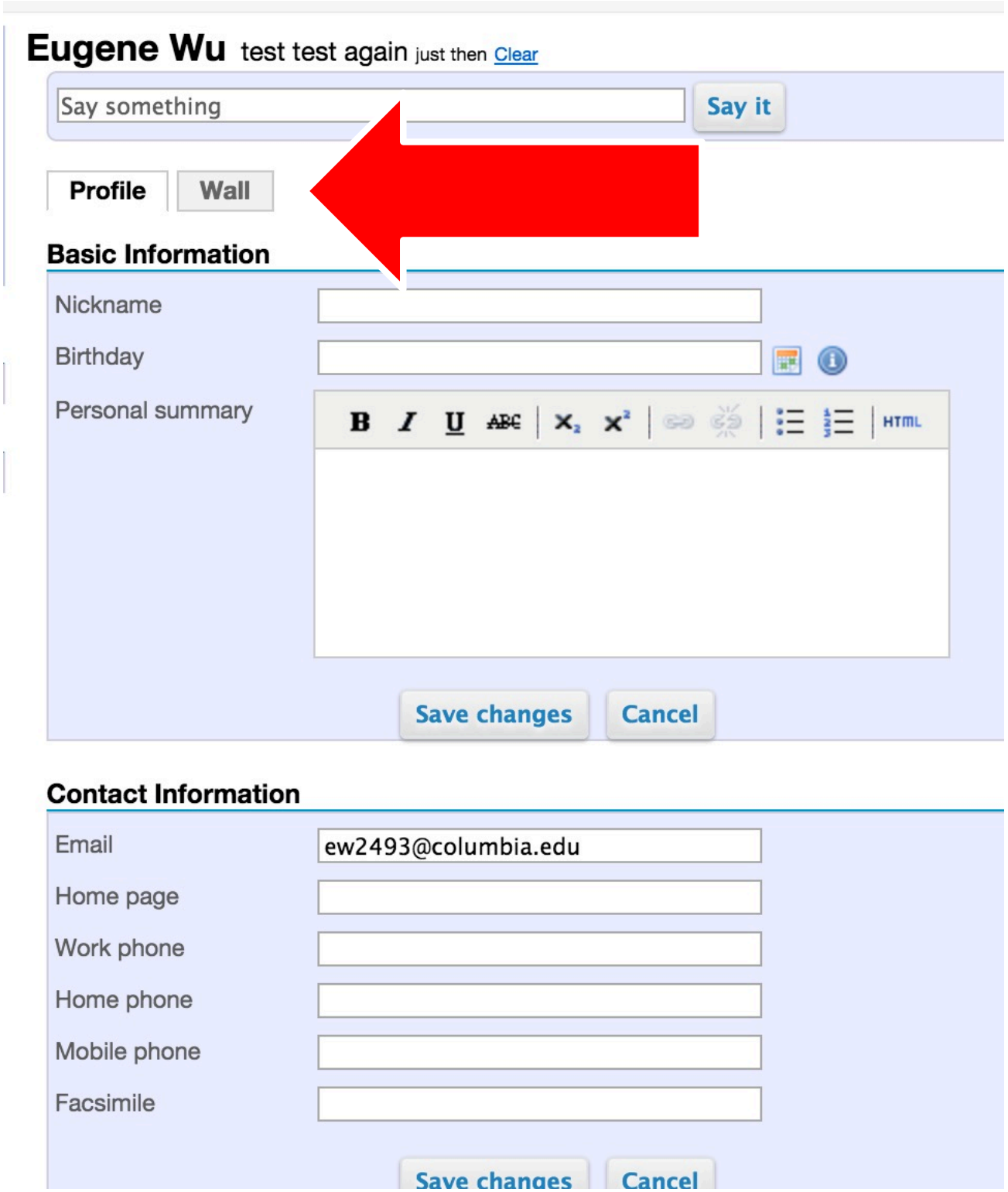
Fall 2015 – Spring 2016 Courses

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



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)

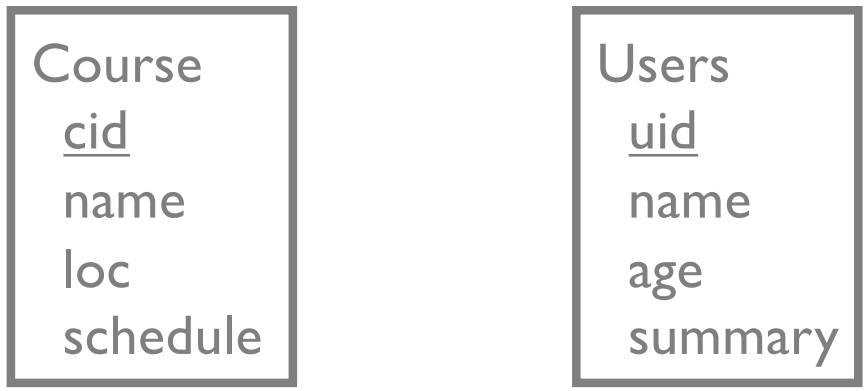
Entity Set e.g., all courses

- 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
- (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

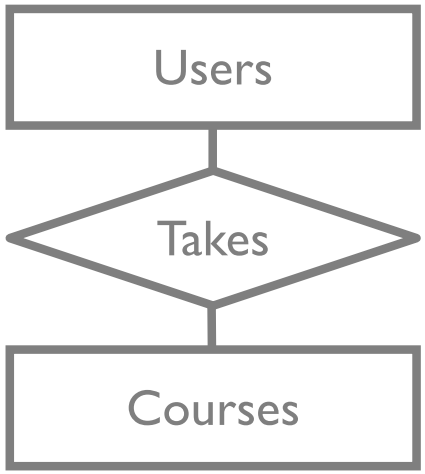
N-ary relationship set R relates N entity sets $E_1 \ldots E_n$

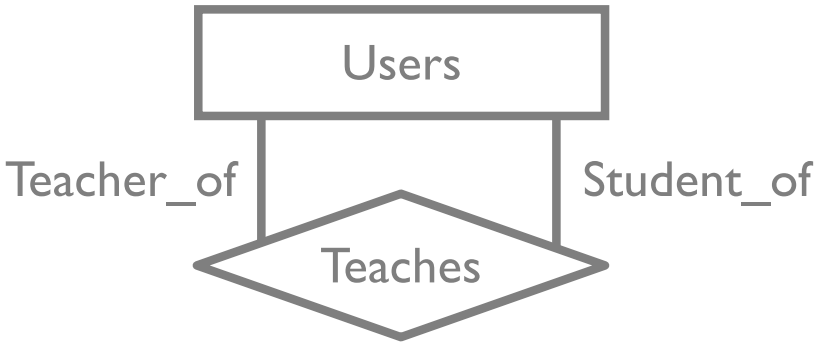
Each $r \in R$ involves entities $e_1 \ldots e_n$

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



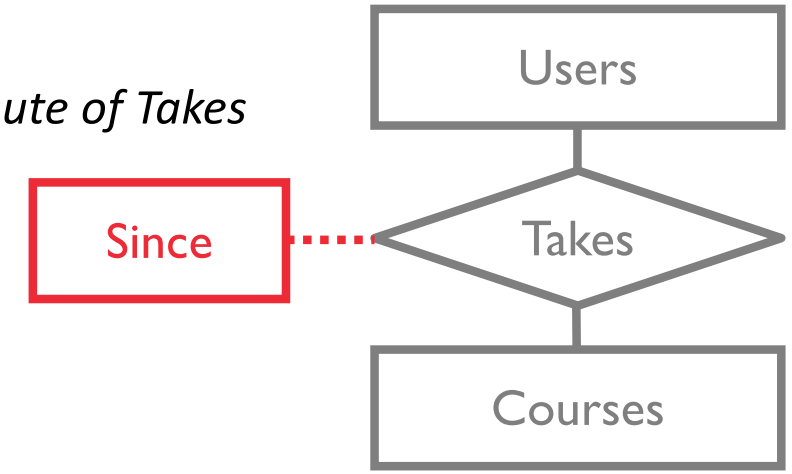


Basics: Relationships

Relationships sets can have descriptive attributes

Denoted with dotted line from diamond to box

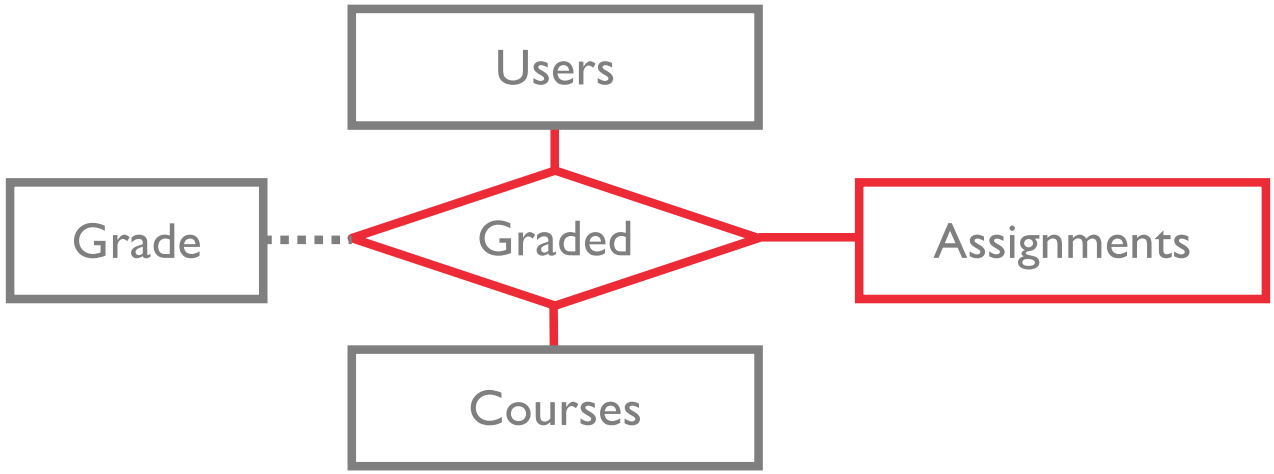
since is attribute of Takes



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

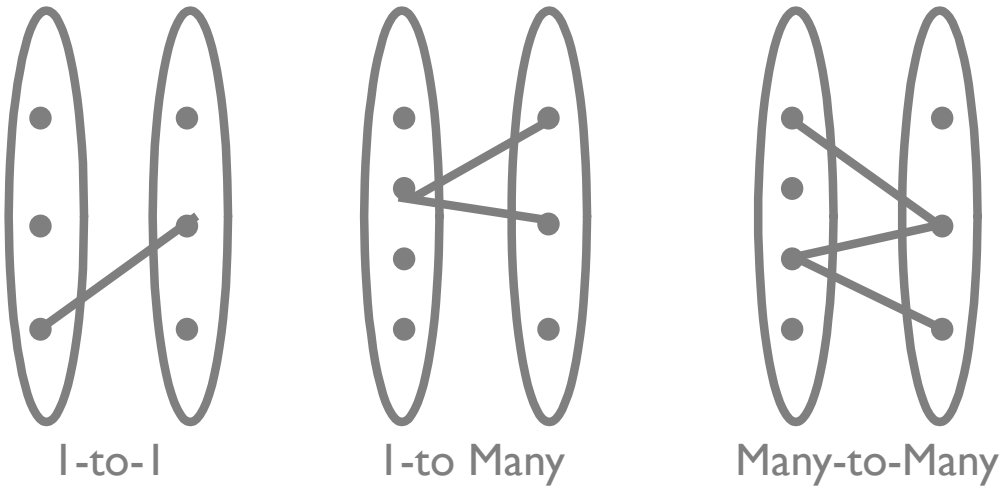
Many to many e.g., *Takes*

a user can take many courses

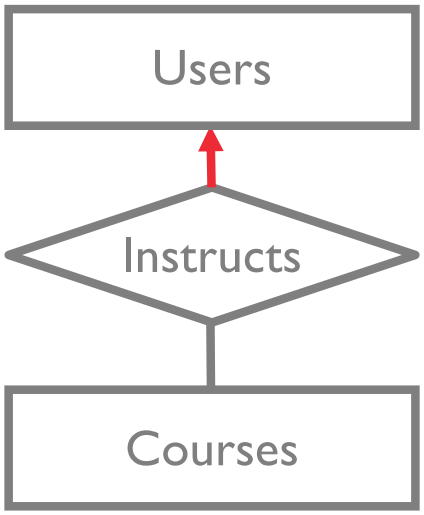
a course can have many users that take the course

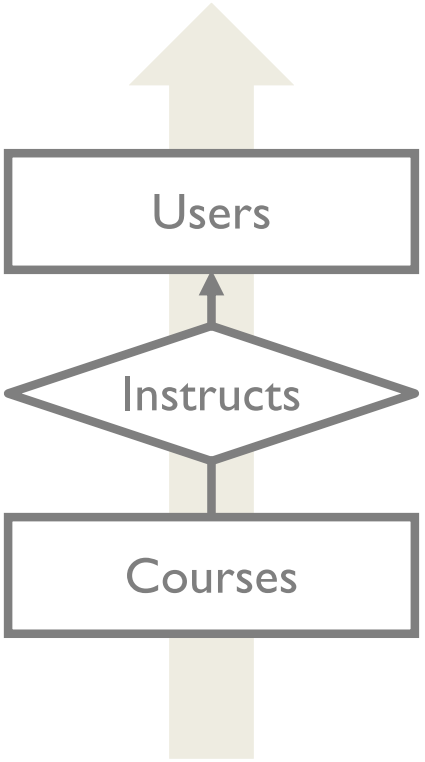
One to Many e.g., *Instructs*

a course has at most one instructor

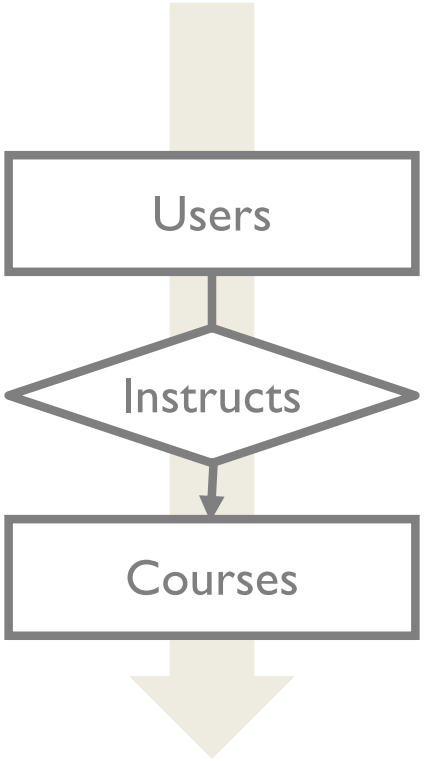


Draw arrow from diamond to box

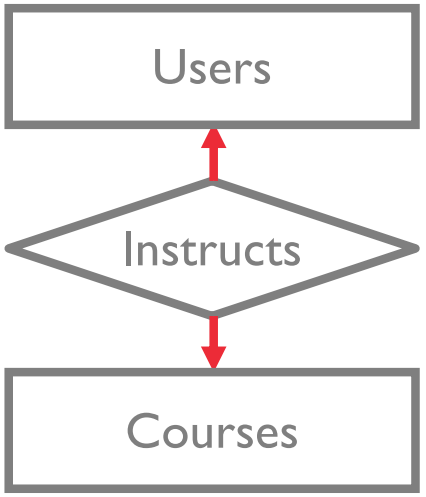




A course is instructed by ≤1 user
(read along the beige arrow)



A user instructs ≤1 course

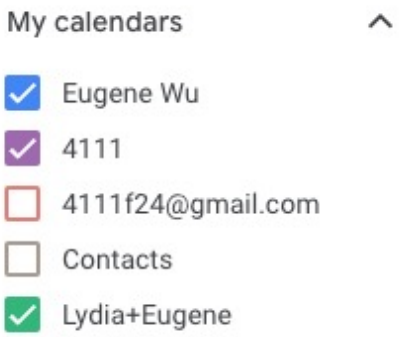


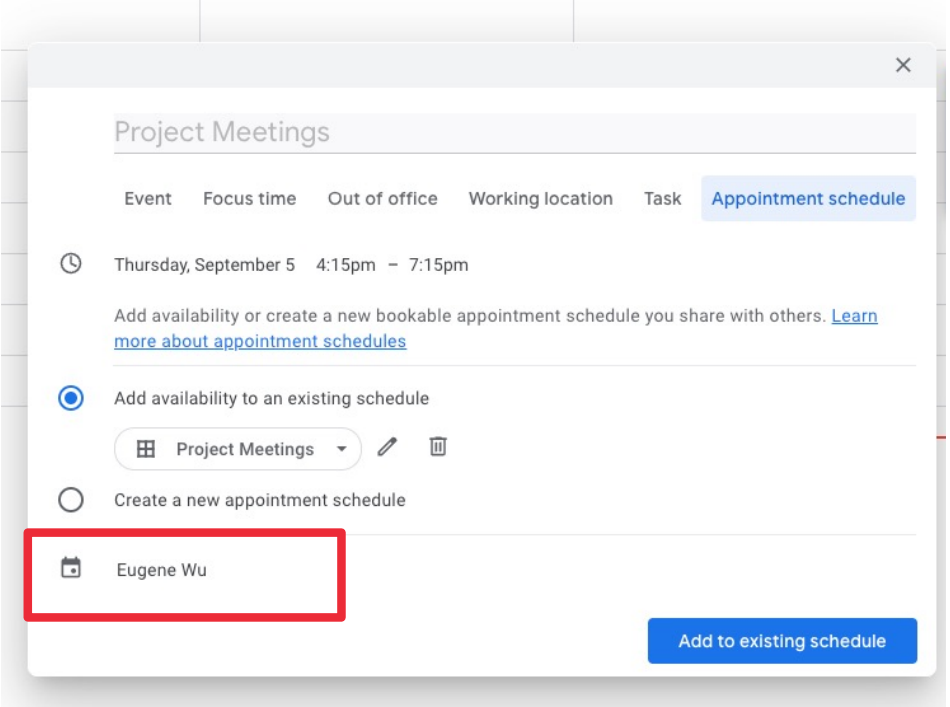
A course is instructed by ≤1 user
AND
A user instructs ≤1 course

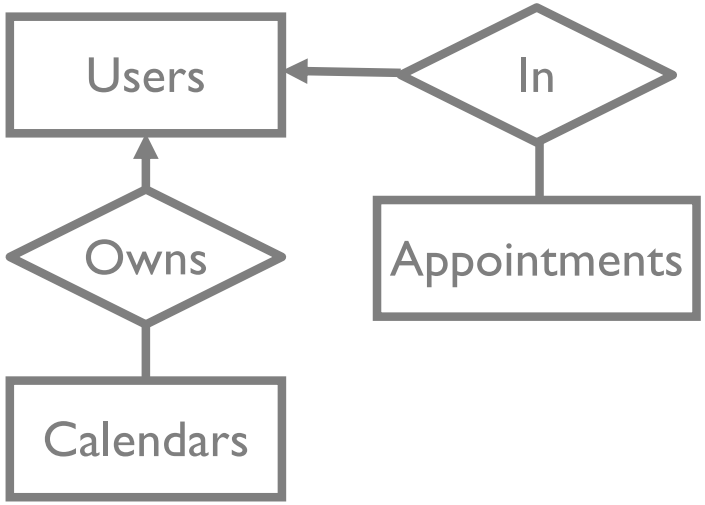
Data Models In the Wild

Try to create appointment slots on 4111 cal

But can only create them in *my* calendar







Participation Constraints

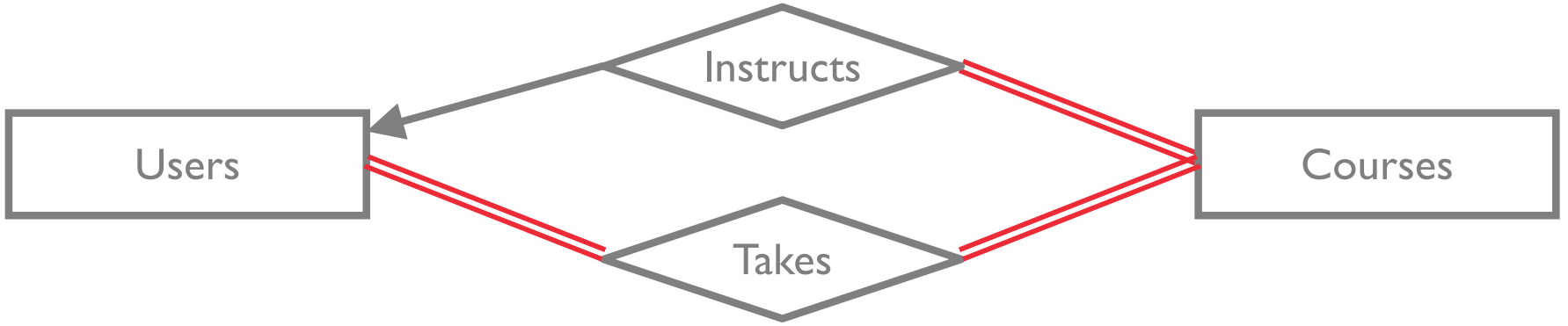
Does every course need an instructor?

If yes, it's a participation constraint (participation of course by instructor is *Total*)

Otherwise, *partial* participation constraint

Ddouble line between entity set and relationship set

Each course has at least one instructor and
each course has at most one instructor



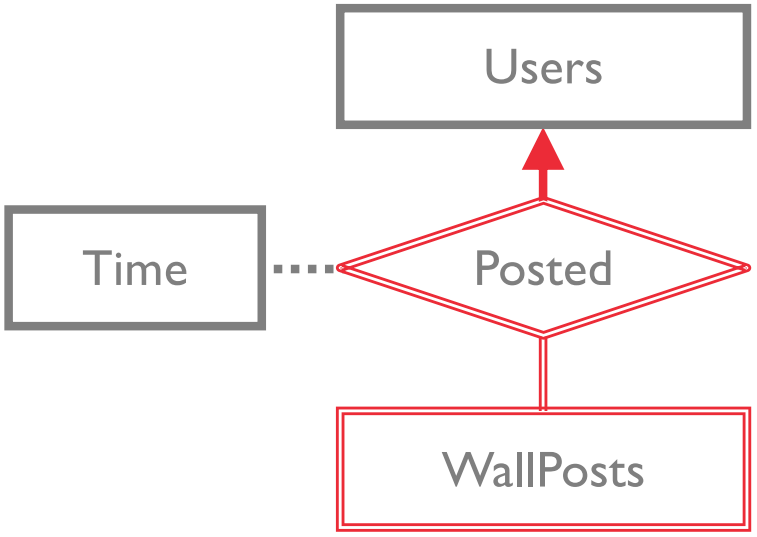
Each user must take at least one course and
Each course must have at least one user (student)

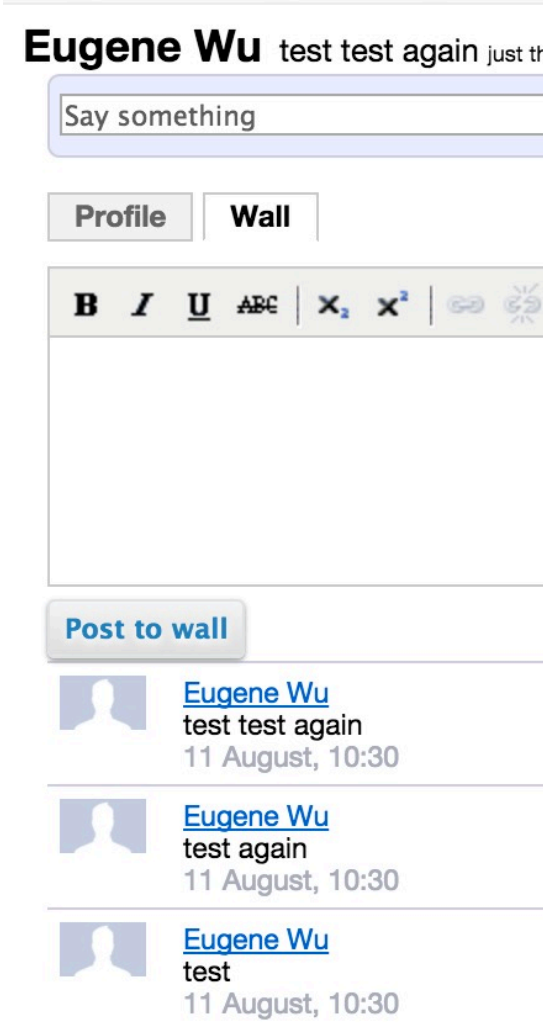
Weak Entities

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



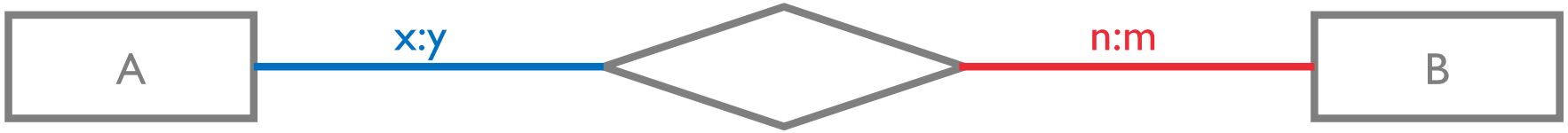


General Cardinality Constraints

same as

A user instructs 0 to ∞ courses

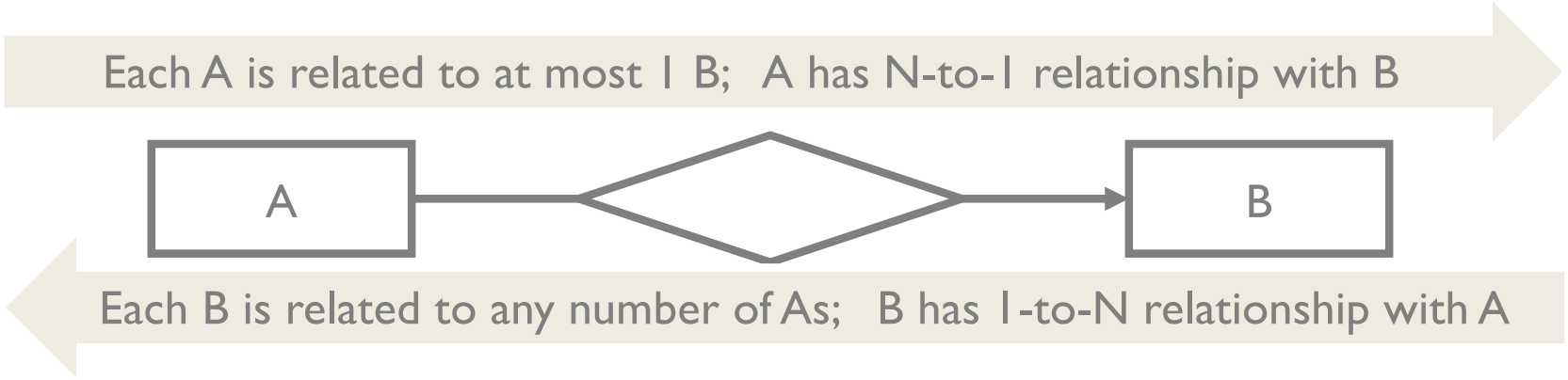
A course instructed by 0 to 1 users

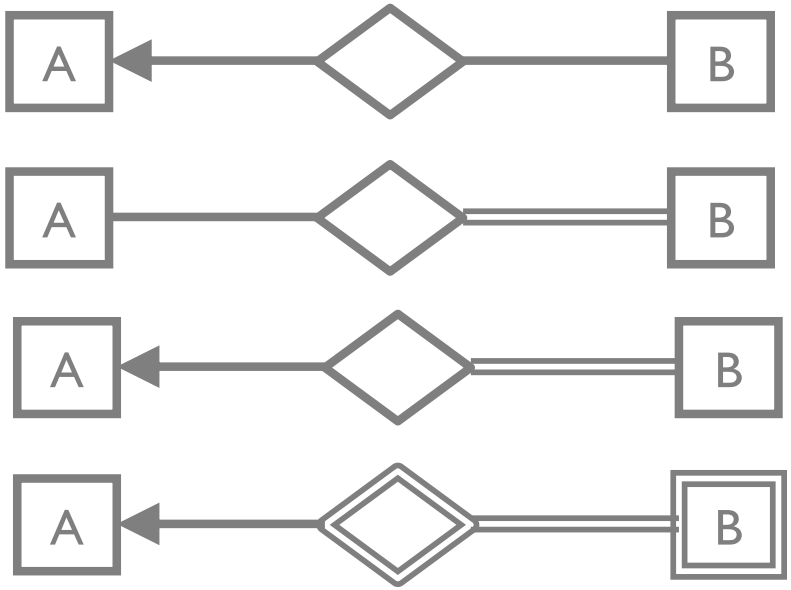


Each A entity has a relationship with between x to y different B entities

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

Read arrows pointing in the direction from start to end





B has at most one A

B has at least one A

B has exactly one A

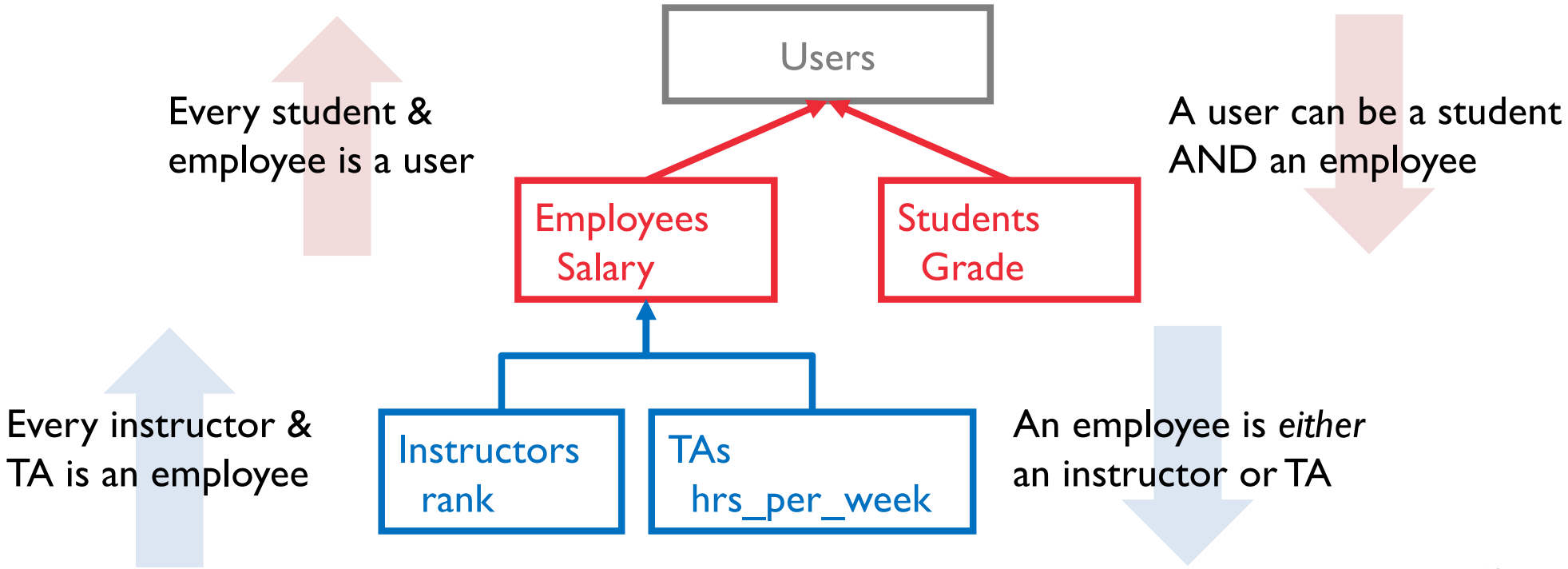
B is a weak entity

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

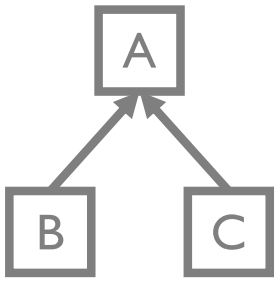


Specialization Hierarchies

Overlap Constraint

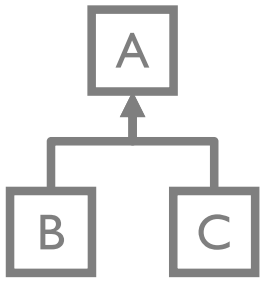
can A be a B *and* a C?

YES



separate arrows

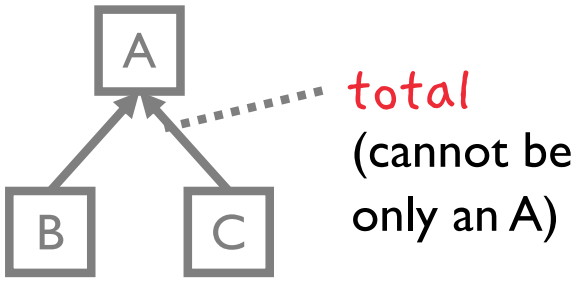
NO



merged into 1 arrow

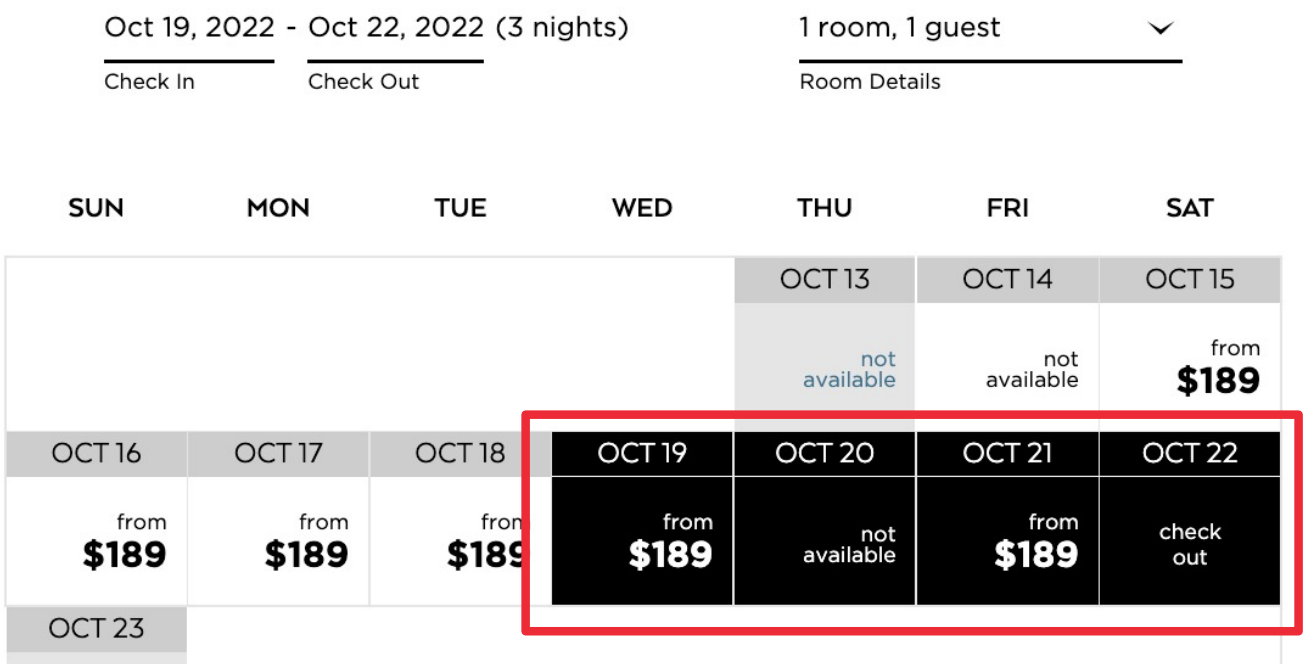
Total Specialization Constraint

must A be a B or C?
specify as the comment "total"
with dashed link to arrows



VIS 2022

the premier forum for advances in visualization and visual analytics



Want: Book Oct 19-22 with discount, pay full rate on Oct 20.

Got: 3 reservations. One for each night.

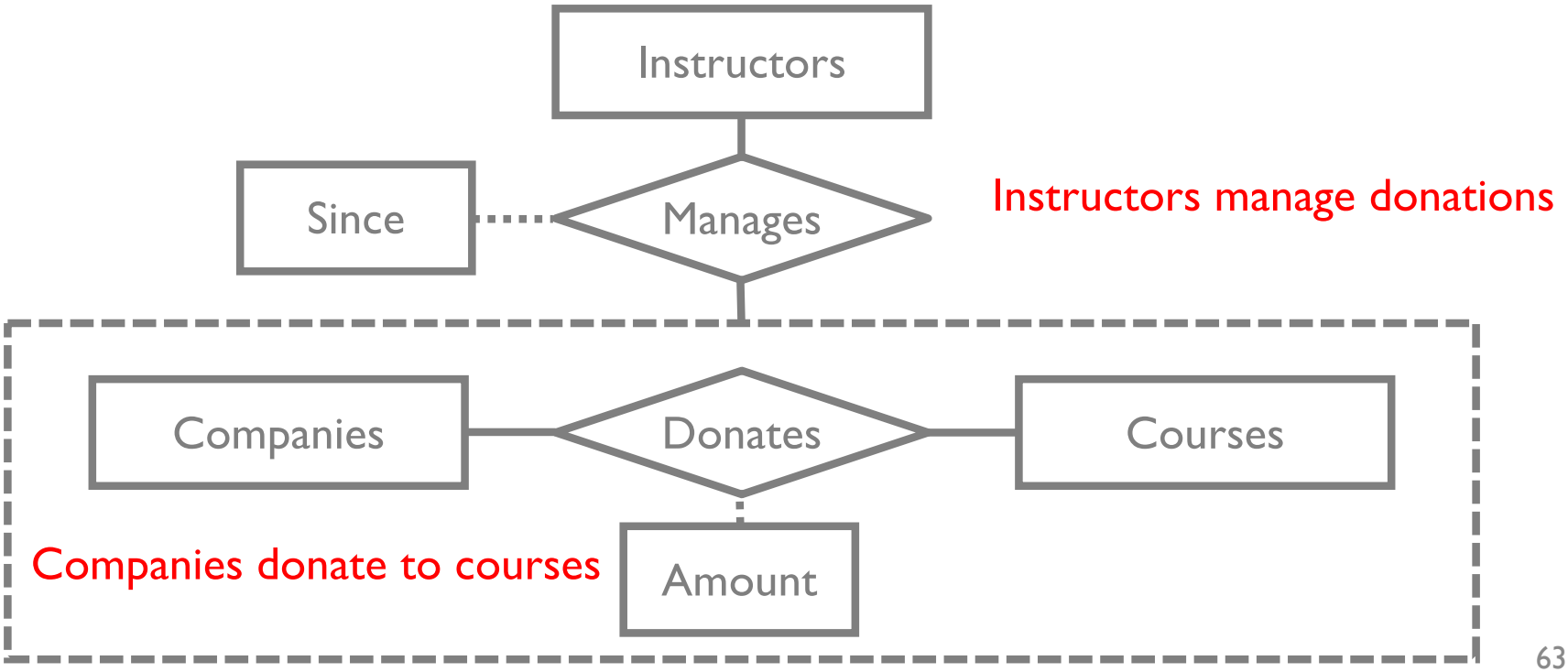
What does this imply about the data model?

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

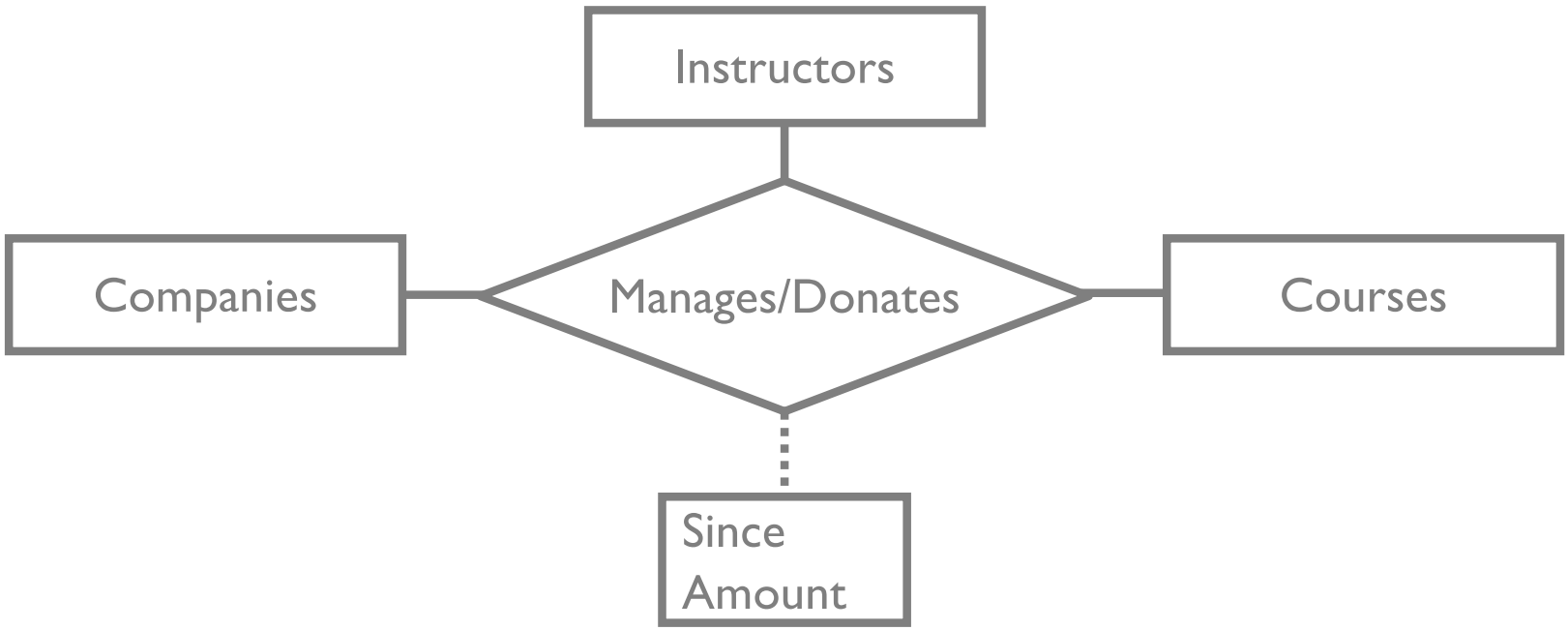


Aggregation vs Ternary Relationships

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?

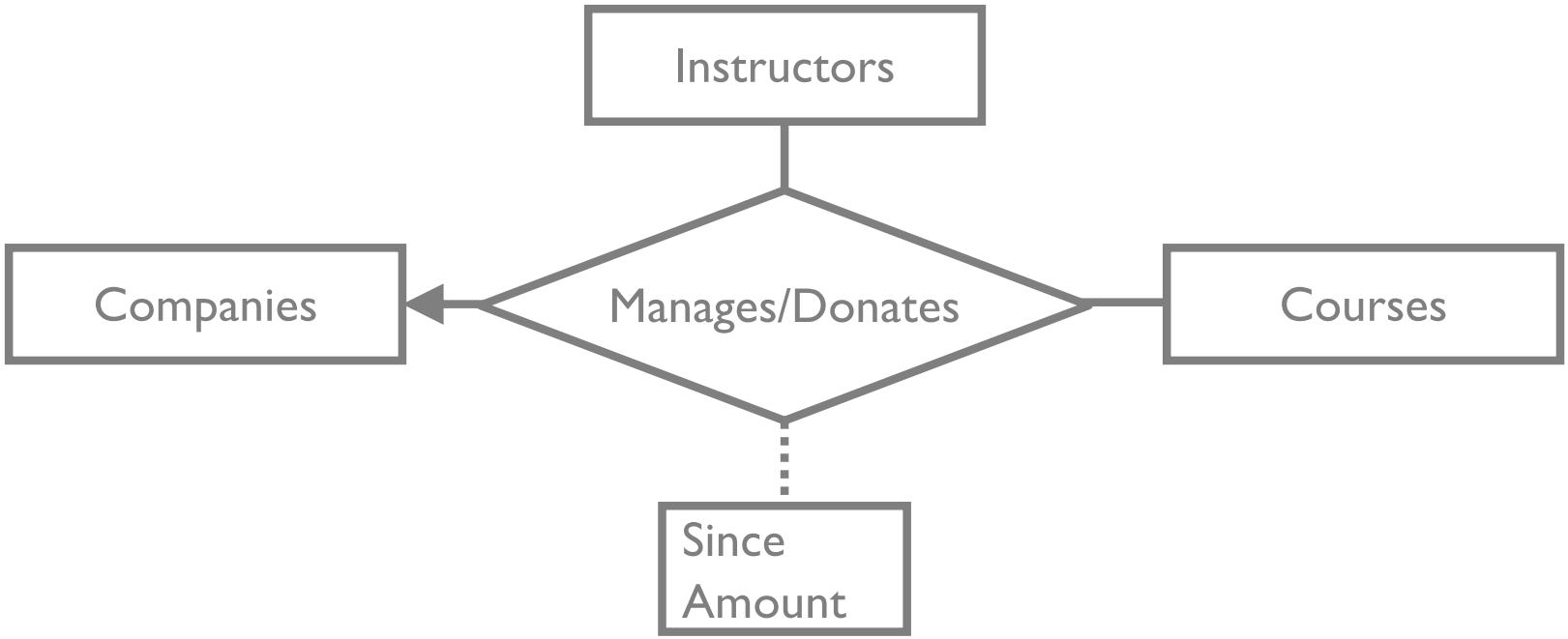


Aggregation vs Ternary Relationships

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.*



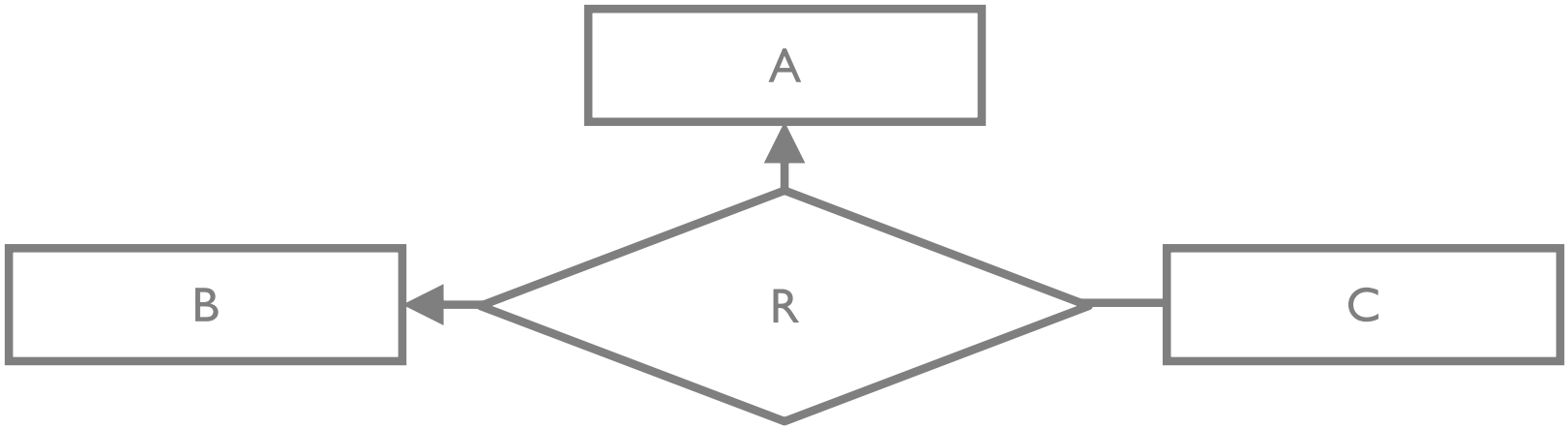
Aggregation vs Ternary Relationships

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?

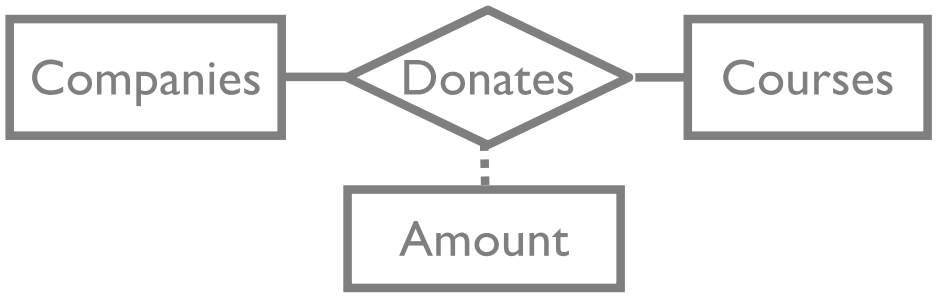
Is users.address an attribute of Users or an entity connected to Users by a relationship?

Depends (and may change over time!)

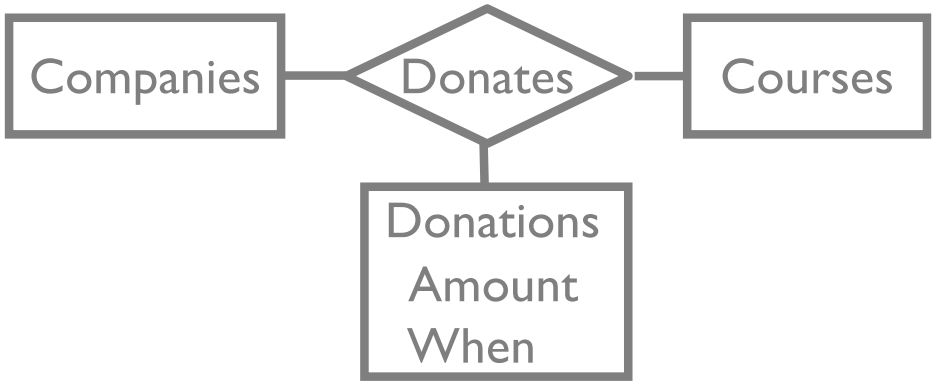
- If a user has >1 addresses, must be an entity
- If an address has attrs (structure), must be entity
  - e.g., want to search for users by city, state, or zip

Entity or Attribute?

A company can't donate multiple amounts



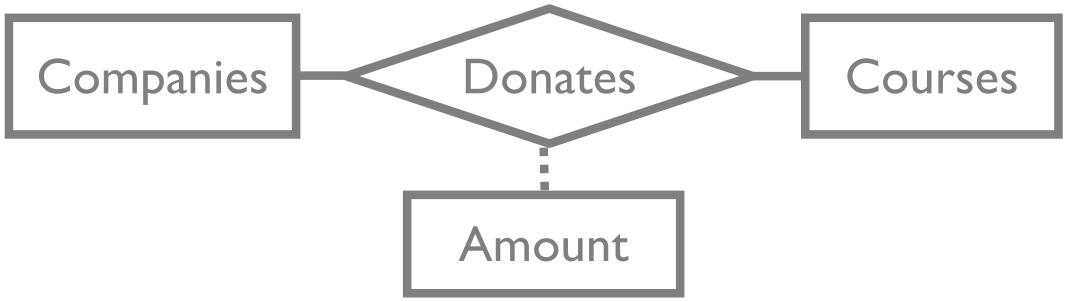
Company can make multiple donations



Entity or Relationship?

Company makes 1 donation to split across all data-related courses

- Redundancy of *amount*, need to remember to update every one
- Misleading implies *amount* tied to *each* donation individually



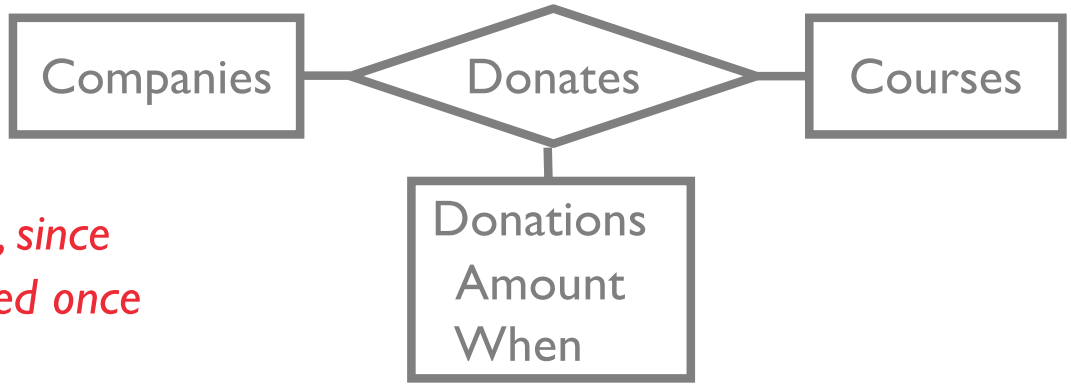
Company	Course	Amount
Amazon	4111	2000
Amazon	4112	2000
Amazon	5111	2000

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	1
Amazon	4112	1
Amazon	5111	1

Donation	When	Amount
1	Today	2000

Entity or Relationship?

If company donates once to school for data related courses.

Refactor amount into an entity (or could be an aggregation)

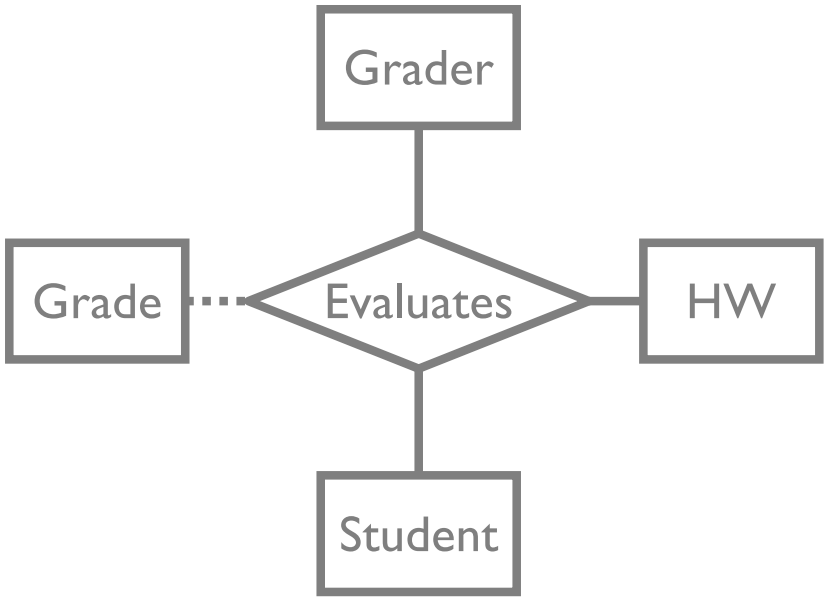
Course	Donation
4111	1
4112	1
5111	1

Donation	When	Amount	Company
1	Today	2000	Amazon

Binary or Ternary Relationship?

HW means a particular released HW, not a submission

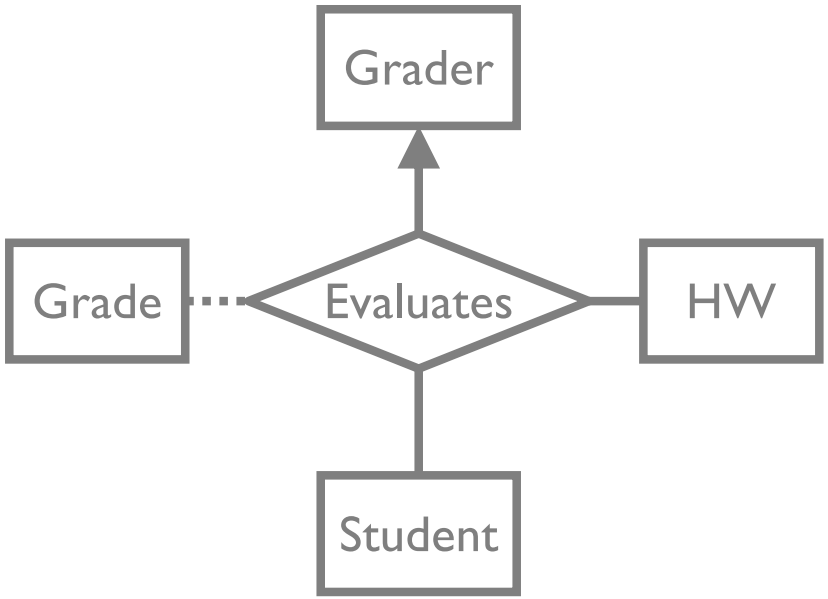
What if each HW has at most one grader? (next slide)



Binary or Ternary Relationship?

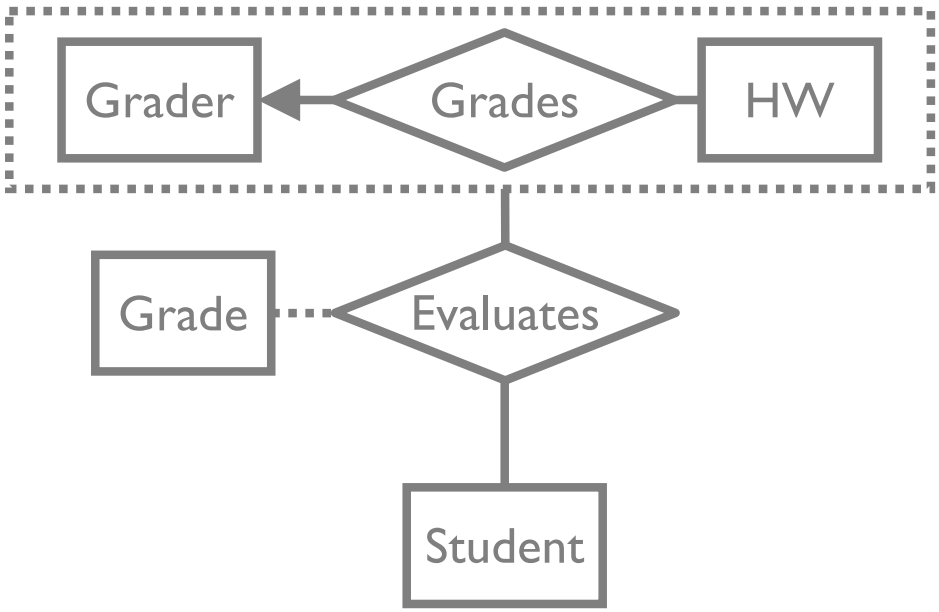
What if each HW has at most one grader?

Option 1: add arrow from evaluates to grader.



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

Option 2: aggregation

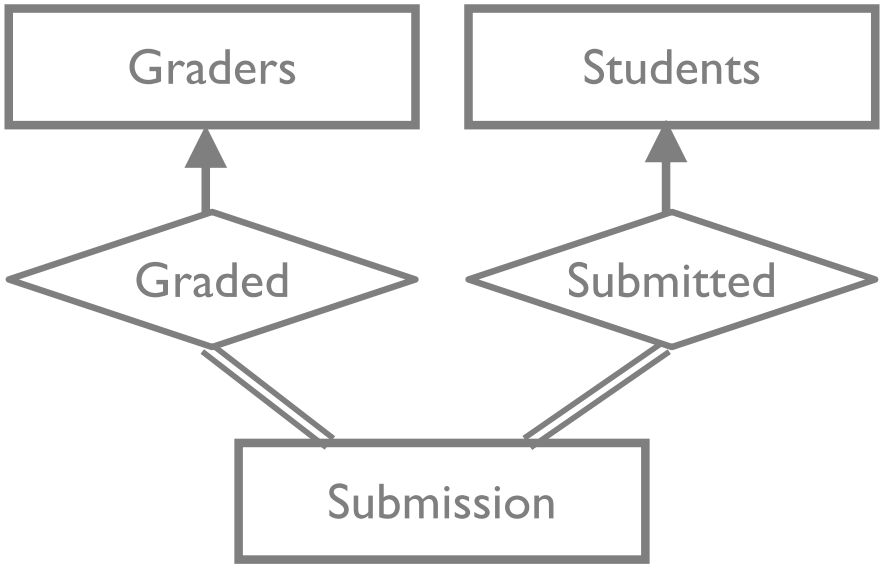


Each HW has at most 1 grader and the grader evaluates each student

Binary or Ternary Relationship?

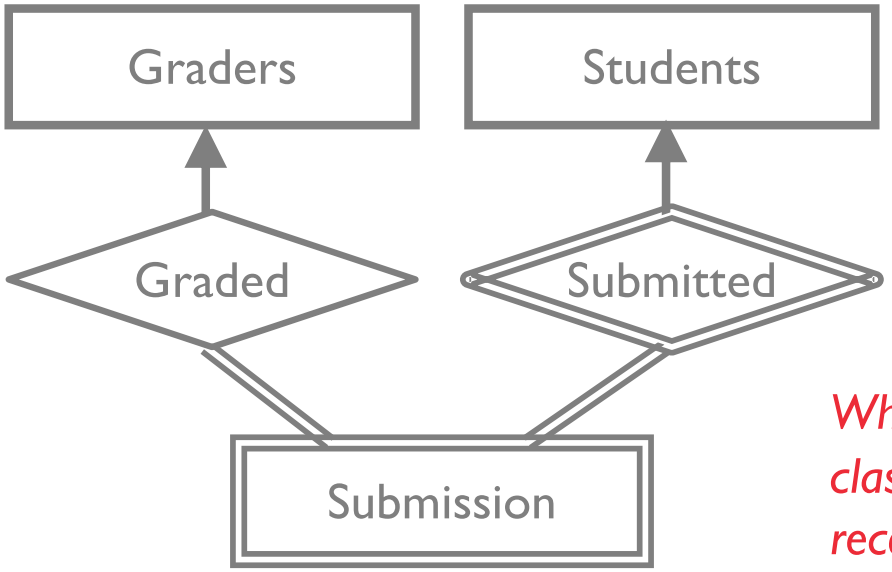
Binary relationships allows additional constraints

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



Binary or Ternary Relationship?

Binary relationships allows additional constraints

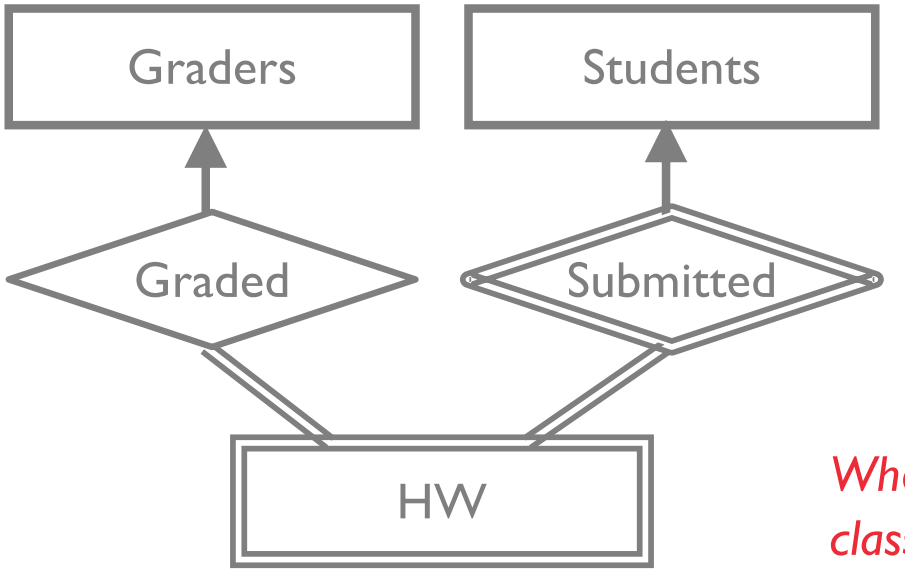


When student drops the class, their submission records also are removed. Makes sense!

Binary or Ternary Relationship?

Binary relationships allows additional constraints

What if we model HW instead of Submission?

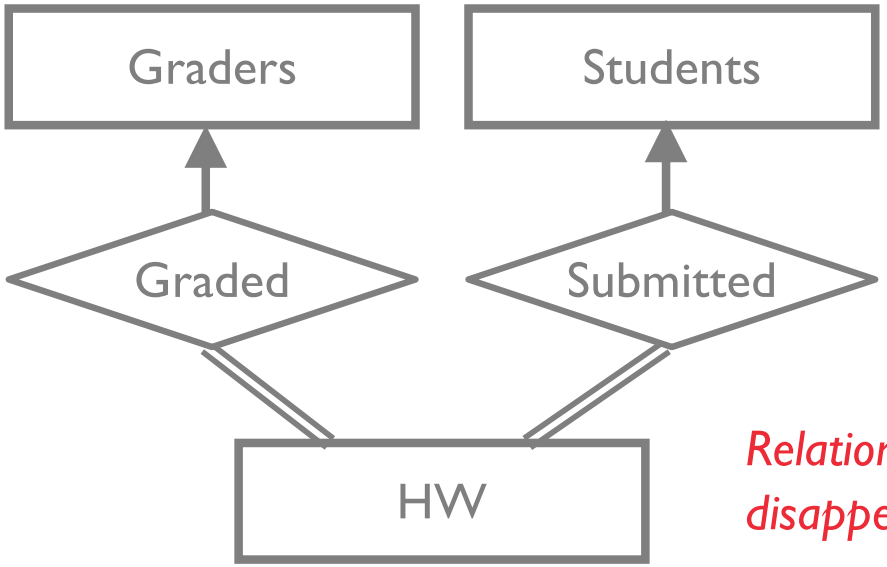


When student drops the class, HW0 also disappears!

Binary or Ternary Relationship?

Binary relationships allows additional constraints

What if we model HW instead of Submission?

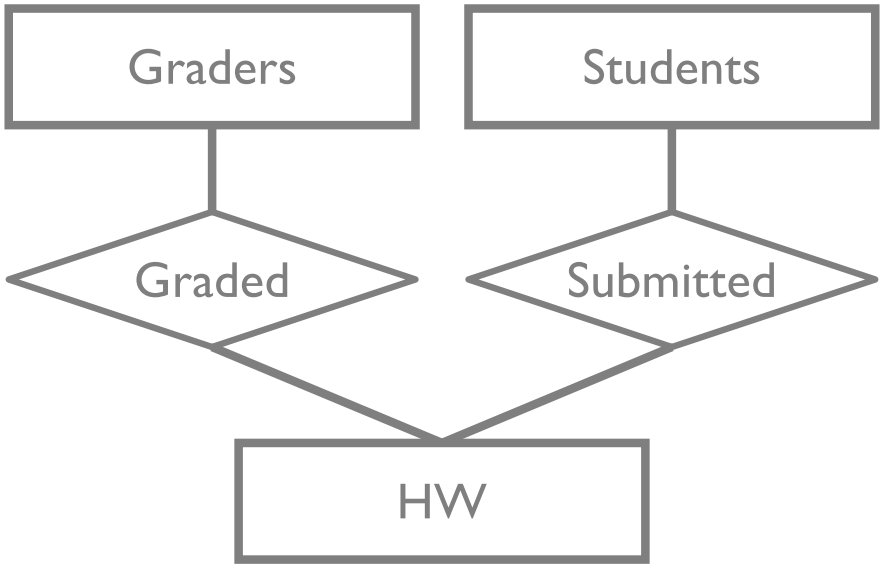


Relationship "submitted" will disappear if student is removed.

But there is still an error

Binary or Ternary Relationship?

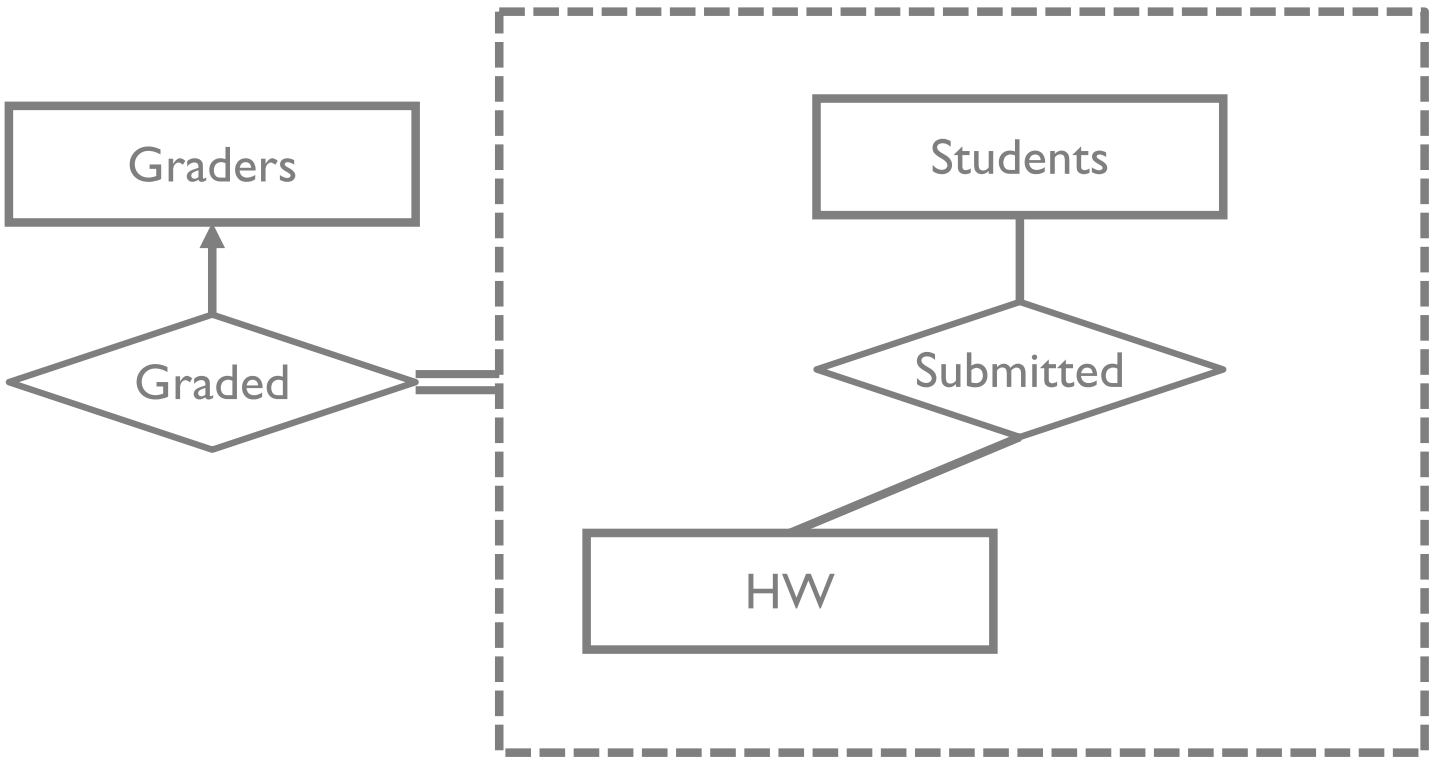
Still an issue…



Binary or Ternary Relationship?

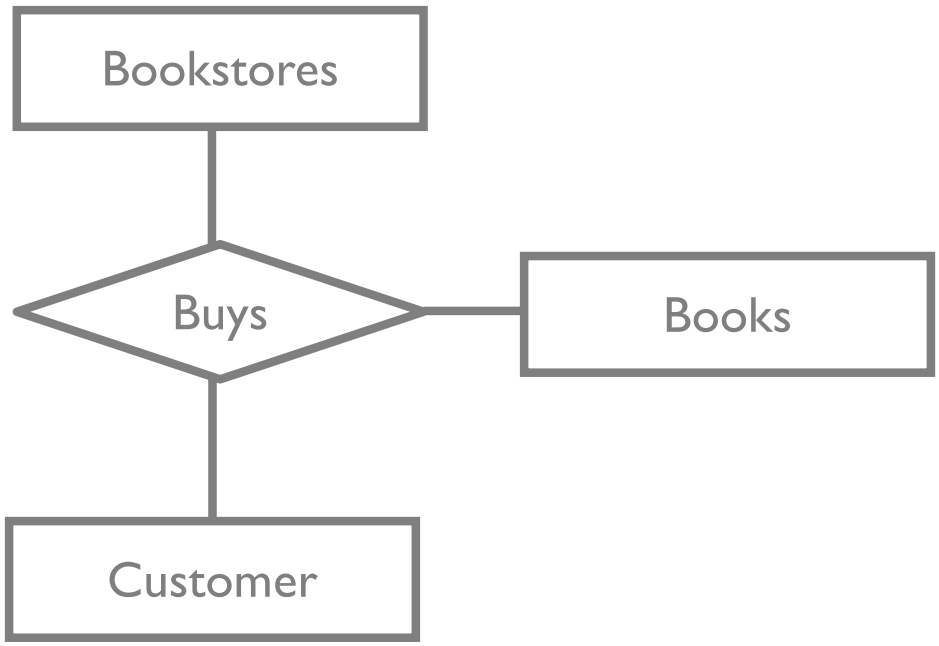
Students can submit HWs.

Each submission has exactly 1 grader



Binary or Ternary Relationship?

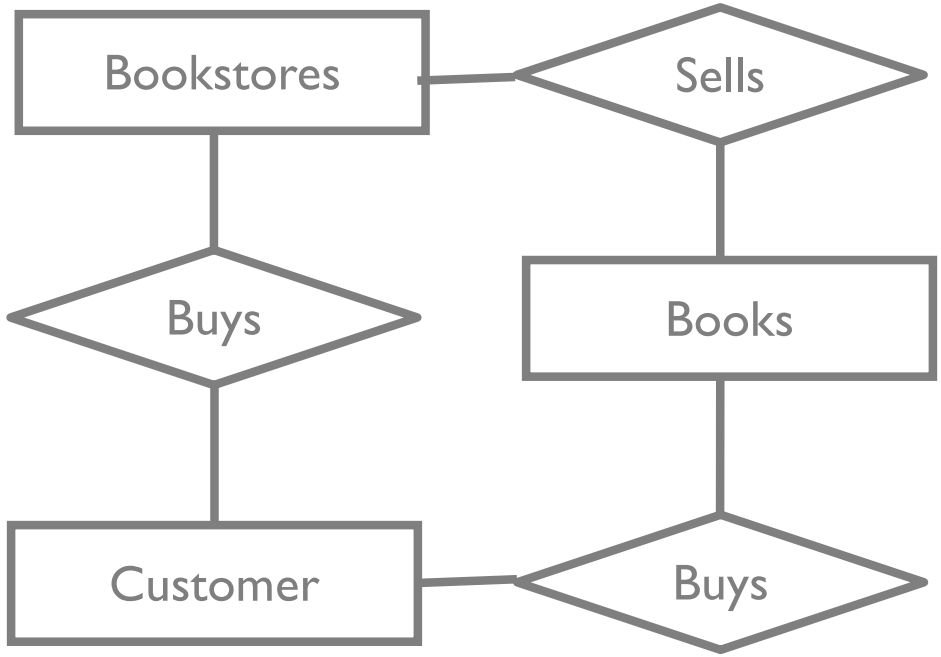
Sometimes have true ternary relationship that is defined by all three entities.



Binary or Ternary Relationship?

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

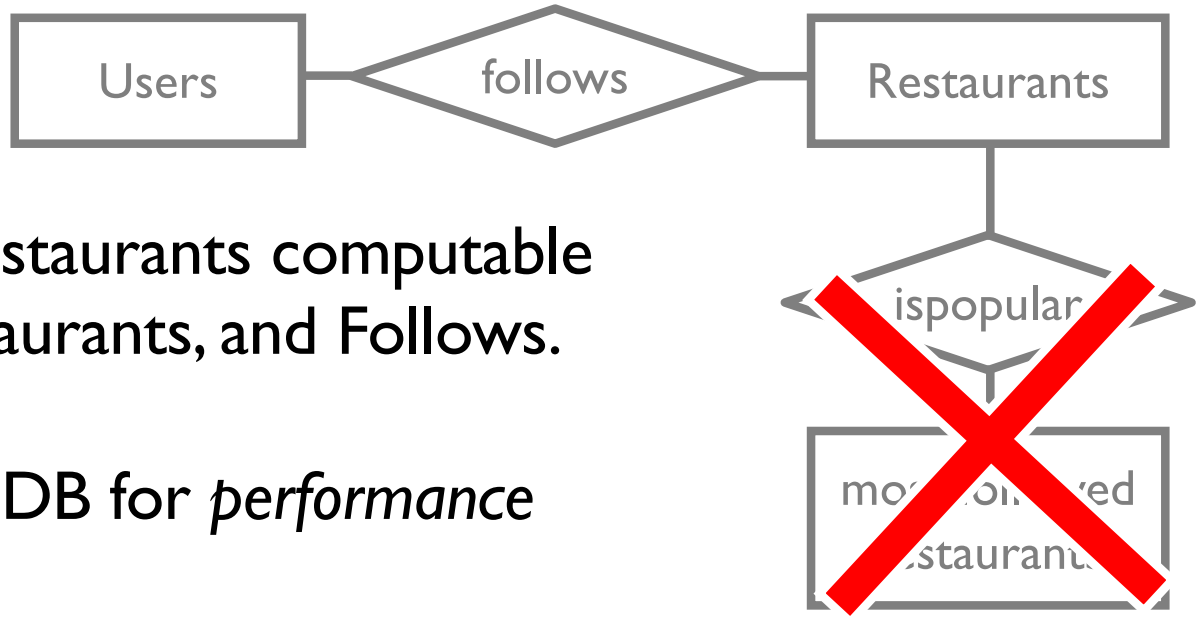
Doesn't Really Work



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