- TA office hours are on website/piazza
- HW0 was due at 10am

Collaboration Policy

Read Syllabus on course site for allowed conduct

CS Dept academic honesty policies
http://www.cs.columbia.edu/education/honesty

We will not tolerate *any* cheating
immediately reported to http://studentconduct.columbia.edu/

Class Structure for Success

Goal: everyone can succeed *without others failing*

Exams will be *hard* (avg ~50% in previous years)
Curves are *always* in your favor
"Who Wants to Be a Millionaire?" lifeline on exam

Participation → rounding in your favor
Extra credit opportunities

Final grading criteria/adjustments will not be shared

Scribe Notes aka extra credit

W4111 Scribe Notes

The goal of these scribe notes is to eventually create a document that can replace and surpass the expensive textbook. These notes are meant to supplement the lecture slides, which do not include detailed information nor full examples, and address the issue that the same questions are repeatedly asked on Piazza.

- https://github.com/w4111/scribenotes/wiki

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:

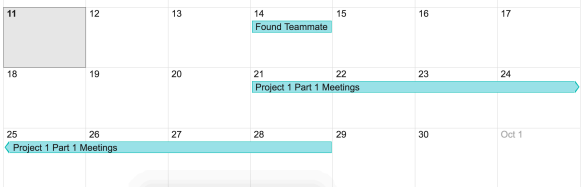
[…],SINGLE MALT SCOTCH,[…],Macallan 12 Yr Single Malt Scotch,[…]

Poll: HW0

A) Short
B) Medium
C) Long

Administrative Notes

Waiting list: >100!
First come, first served (sorry)



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

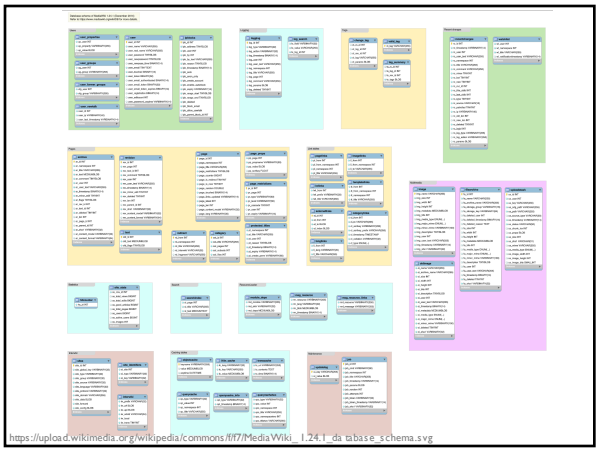
Steps for a New Application

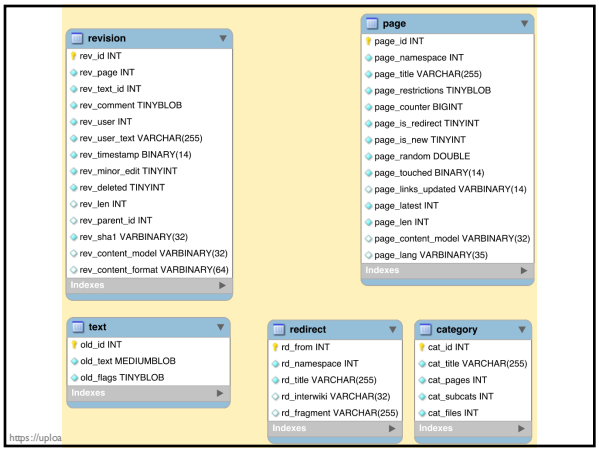
Requirements
  what are you going to build?
Conceptual Database Design
  high-level description — ER Modeling
Logical Design
  formal database schema
Schema Refinement:
  fix potential problems, normalization
Physical Database Design
  use sample of queries to optimize for speed/storage

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

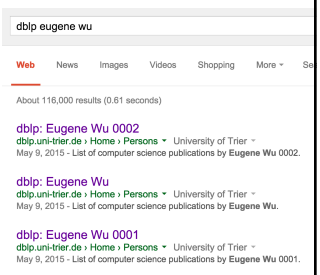




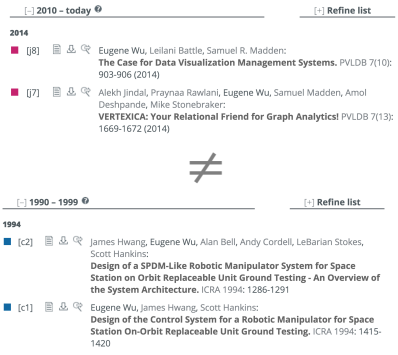
Inconsistencies/Constraint Violations

Huge amount of effort to avoid inconsistencies

DBLP is *the* site for computer science publications

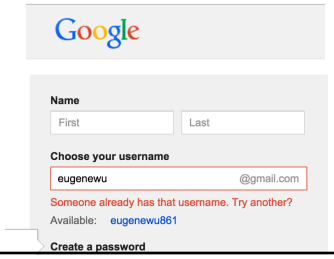


Inconsistencies/Constraint Violations

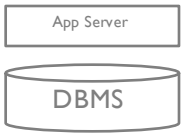


Inconsistencies/Constraint Violations

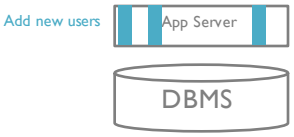
Giving me eugenewu@gmail would violate constraints



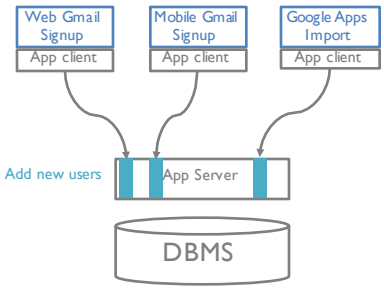
It is Hard to Design Applications



It is Hard to Design Applications

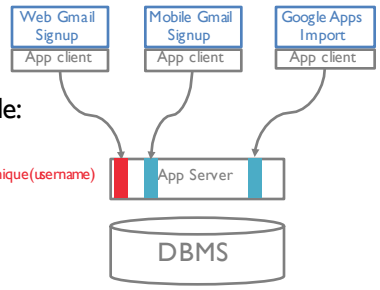


It is Hard to Design Applications



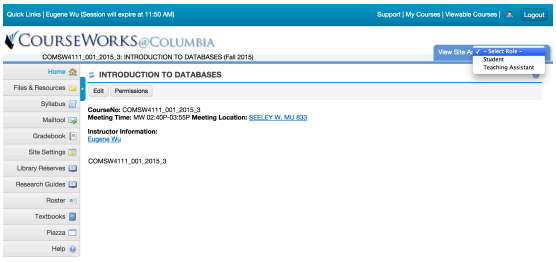
It is Hard to Design Applications

"DRY" principle:
Don't
Repeat
Yourself



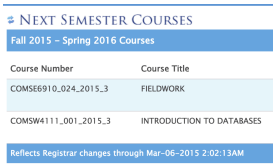
Let's make ~~a webapp~~ $$$

live exercise time



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

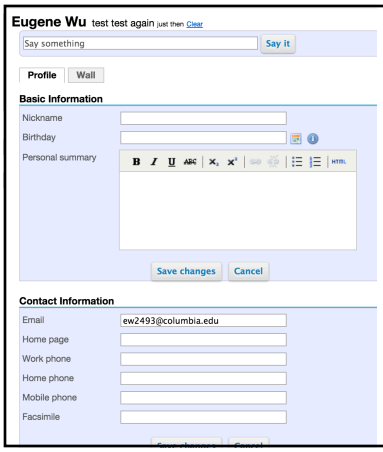

NEXT SEMESTER COURSES

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
- object distinguishable from other objects of the same "type"
- described as set of attributes and their values
- domain of an attribute: set of possible values (e.g. integers)
- (think one record)

Entity Set e.g., courses
- collection of similar entities
- all entities have same attributes (except Is-A, later)
- ≈ table

Keys

Minimal set of attributes that uniquely identify an entity
May be multiple candidate keys
- e.g. User: both uid and email may be unique
- Primary key: ?

May involve multiple attributes
- e.g. Class identified by both number and section

Primary key: designated unique identifier
Most entities have a key (an exception later)

Example: Entity

Keys are underlined
- Its values must be unique
- (think: can use as hash table key to find value)

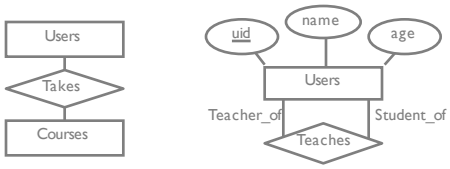
Basics: Relationships

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

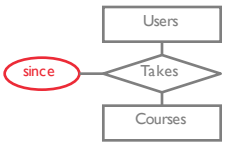
Users takes different roles in same relationships set



Basics: Relationships

Relationships sets can have descriptive attributes

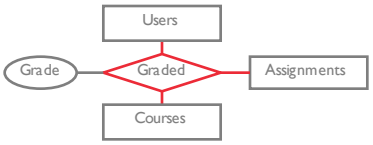
e.g., the *since* attribute of Instructs



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

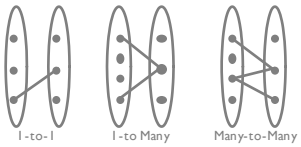
Many to many e.g., consider *Takes*

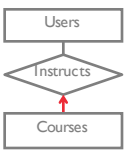
a user can take many courses

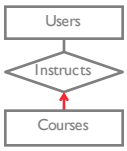
a course can have many users that take the course

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

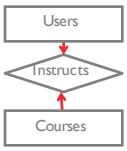
a course has at most one instructor



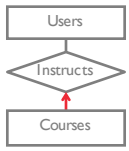




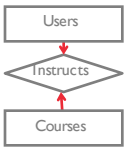
A course has at most one instructor



???



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

arrows mean "at most one"

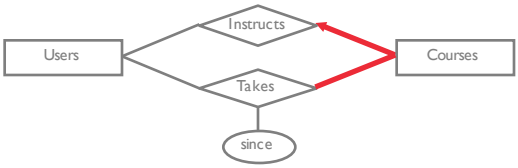
Participation Constraints

Does every course need an instructor?

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

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

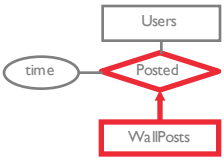
Otherwise, *partial* participation constraint

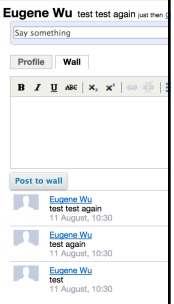


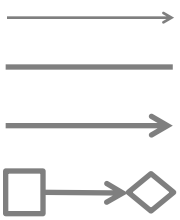
Weak Entities

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)

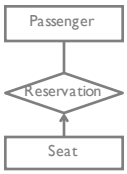
Diagram an airplane reservation with the following entities:

Passenger

Seat

What other entities might exist?

Possible solution

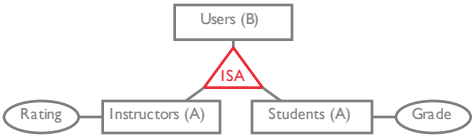


Seat: At most 1 reservation (no double booking)
Passenger: Optional: at least one reservation (thick line)

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

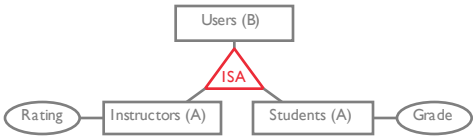


ISA (is a) Hierarchies

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

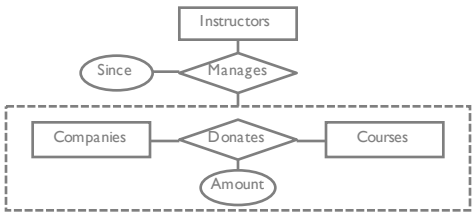
Covering Constraint
- must every user be an instructor or student? (yes/no)

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



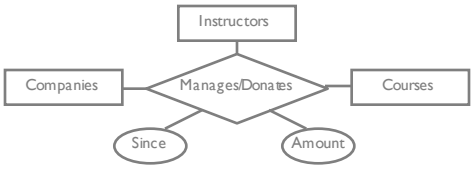
Aggregation

Relationships between (entities – relationships)
Lets us treat a Relationship Set like an Entity Set
- so it can participate in other relationships



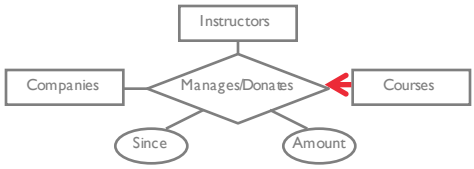
Aggregation vs Ternary Relationships

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



Is your head spinning?

Hard to be precise about what data to store!
- popularity of "schemaless" databases

entities/relationships
- which one to use depends on what you want

Survey

http://tinyurl.com/w4111