Discussion 1

ER Diagrams & Project 1 Intro
+ Associated Tools

EECS 484

Why is this course useful?

- We’re surrounded by databases.
    - Examples: mobile apps
- In the first half of this course, we will learn how to use databases.
    - How to model data, how to store and retrieve info, …
- In the second half, we will focus on the internal design of databases, which helps us use and design them efficiently.

Course Overview

- Intro to database systems
- Entity Relationship (ER) diagrams and the relational model
- Structured Query Language (SQL)
  - Will be spending a lot of time working with (coding assignments)
- Relational Algebra
  - Query language for expressing plans in a mathematical form
- Normalization
  - “Good” way to design relations
- Indexing
  - B+ trees and hash tables
- Query optimization
- Transactions
- Recovery

Logistics

- 5 Discussion sections (4 in-person and 1 virtual) a week on Friday
  - Will cover material from this week's lectures
  - Attendance strongly encouraged
    - Some topics not covered in lecture
    - The week a homework is due, some homework problems will be reviewed in more detail in discussion only
      - This will not be included in the recording
  - If you have to miss your section one week, attend another.
- Homework 1
  - Released on Aug 29th
  - Due Sep 13th, 11:45 PM ET
  - Individual, No Group!
  - Homeworks are good prep for the exam
- Project 1
  - Released on Aug 29th
  - Due Sep 24th, 11:45 PM ET
  - Groups of 2. Make sure to add each other as a group before submitting to the Autograder
  - Part 1 is due on Gradescope
  - Parts 2-4 are due on the Autograder

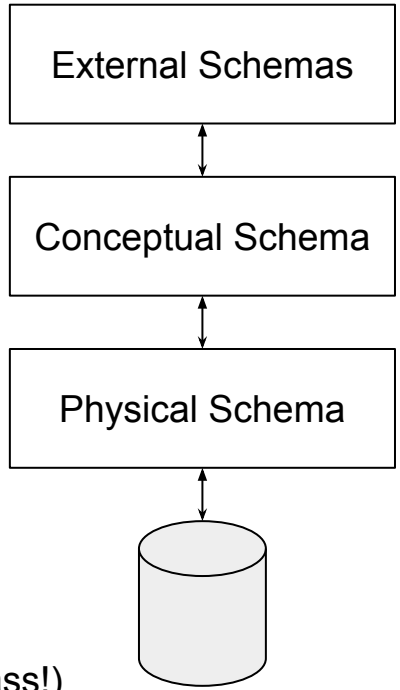
Database Basics

DBMS

- DBMS = Database Management System
  - Oracle SQL, PostgreSQL, MySQL, Transact SQL (Microsoft SQL), etc.
  - Provides declarative system to store data
    - We tell it what we want
    - As opposed to imperative (we don’t care how the DBMS stores the data in files)
- Relational database systems
  - Collection of relations (think tables)
  - Defined by a schema
    - Relation name and columns (data type and names)
    - Any other attributes

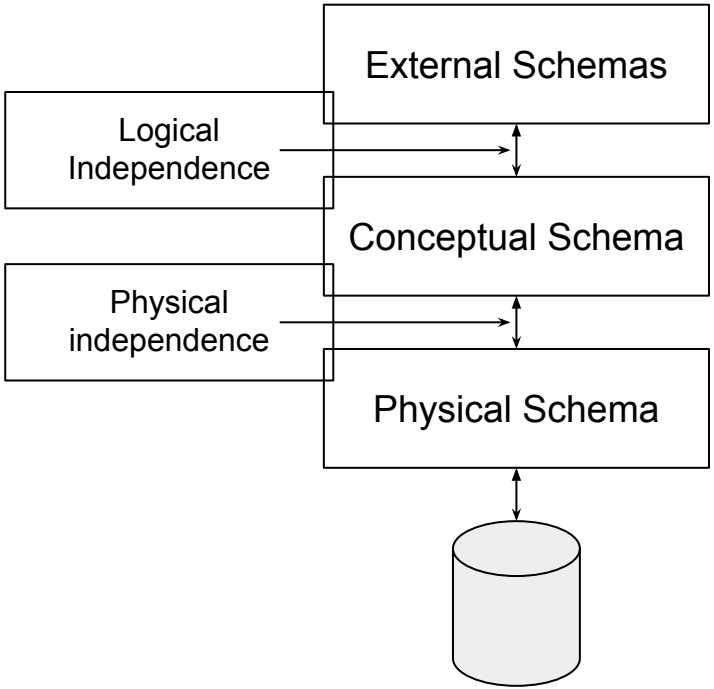
Abstraction

- Different types of schemas
  - Physical schema - how data is stored in memory with files
    - Example - Files for each relation
  - Conceptual schema - what is the the logical structure in terms of the data model
    - Example - Student relation with columns:
      - umid (number)
      - grade percentage (number)
      - name (string)
  - External schema - how is the data represented to a viewer
    - There can be multiple external schemas!
    - Example - Grader view:
      - Grader can see umid, grade (want to obscure name)
    - Example 2 - Canvas coursepage view:
      - Students can see names only (see which friends are in the class!)



Data Independence

- Logical data independence - protection from changes in logical structure of the model
  - Columns in a table within the registrar's database change but instructors don't know anything changed
- Physical data independence - protection from changes in physical structure of the model
  - Oracle releases an update changing how the database is stored on your computer but we don't notice any changes in our pre-existing database
- Logical data independence is hard to achieve!
  - If I change some of the fields, APIs that depend on the data could behave incorrectly
    - Changing primary key from SSN to user_id



Pop Quiz :D

1. In a relational data model, a schema provides what information?
    a. The total size of your table
    b. The data in your table
    c. The data types and the names of the fields

Pop Quiz :D

1. In a relational data model, a schema provides what information?
    a. The total size of your table
    b. The data in your table
    c. The data types and the names of the fields

Pop Quiz the SQL! :D

1. What type of schema abstraction would we use for each of the following:
    a. The schema a CAEN admin sees when upgrading the student information database
    b. The schema Canvas displays when showing you the other students in the course
    c. The binary files living in CAEN somewhere that contains your student personal info
    d. The schema you see in Wolverine Access when editing your personal information

Remember, the types of schemas are external, conceptual, and physical

Pop Quiz the SQL! :D

1. What type of schema abstraction would use for each of the following:
    a. The schema a CAEN admin sees when upgrading the student information database
    Conceptual Schema
    b. The schema Canvas displays when showing you the other students in the course

    c. The binary files living in CAEN somewhere that contains your student personal info

    d. The schema you see in Wolverine Access when editing your personal information

Remember, the types of schemas are external, conceptual, and physical

Pop Quiz the SQL! :D

1. What type of schema abstraction would use for each of the following:
    a. The schema a CAEN admin sees when upgrading the student information database
    Conceptual Schema
    b. The schema Canvas displays when showing you the other students in the course
    External Schema
    c. The binary files living in CAEN somewhere that contains your student personal info

    d. The schema you see in Wolverine Access when editing your personal information

Remember, the types of schemas are external, conceptual, and physical

Pop Quiz the SQL! :D

1. What type of schema abstraction would use for each of the following:
    a. The schema a CAEN admin sees when upgrading the student information database
    Conceptual Schema
    b. The schema Canvas displays when showing you the other students in the course
    External Schema
    c. The binary files living in CAEN somewhere that contains your student personal info
    Physical Schema
    d. The schema you see in Wolverine Access when editing your personal information

Remember, the types of schemas are external, conceptual, and physical

Pop Quiz the SQL! :D

1. What type of schema abstraction would use for each of the following:
    a. The schema a CAEN admin sees when upgrading the student information database
    Conceptual Schema
    b. The schema Canvas displays when showing you the other students in the course
    External Schema
    c. The binary files living in CAEN somewhere that contains your student personal info
    Physical Schema
    d. The schema you see in Wolverine Access when editing your personal information
    External Schema

Remember, the types of schemas are external, conceptual, and physical

Pop Quiz the 3QL!??! :D

2. If I am updating the DMV database, replacing their eye color with whether a student prefers coffee or tea, which type of data independence will I be most concerned about?

a. Logical independence
b. Physical independence
c. Probably a bit more concerned about the laws at play
d. Pls make it stop, I’m tired of the quiz D:

Pop Quiz the 3QL!??! :D

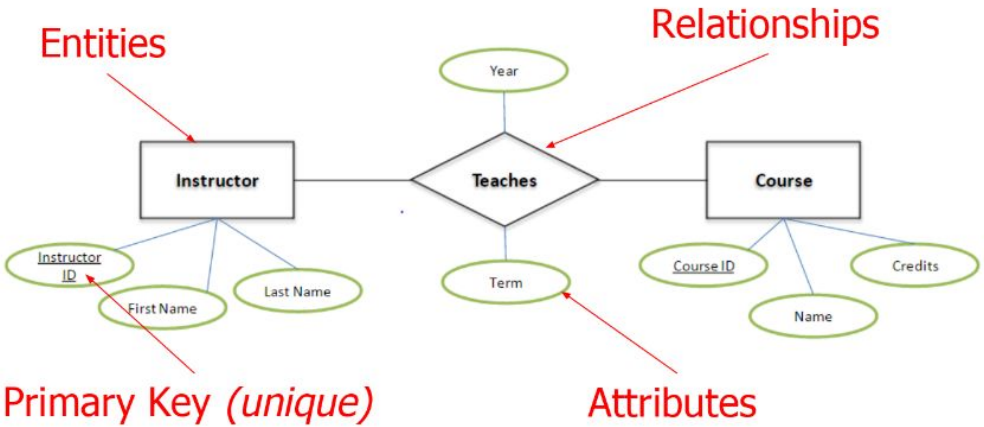
2. If I am updating the DMV database, replacing their eye color with whether a student prefers coffee or tea, which type of data independence will I be most concerned about?

a. Logical independence
b. Physical independence
c. Probably a bit more concerned about the laws at play
d. Pls make it stop, I'm tired of the quiz D:

ER Diagrams

ER Diagram Basics

- Data model that describes database schema/design
  - Entities are things (Actors, Movies, Citizens, Presidents, Types of Tea)
  - Relationships are are actions/verbs/states (Acted in, Lives in, Is president of, Drinks)
  - Attributes are characteristics (Eye color, Rating, SSN, Political Party, Plant derived from)
    - Primary key is unique identifier (can consist of multiple attributes or just one)



Key Constraints (“at most one”)

Many-to-Many

(a)

One-to-One

(b)

(c)

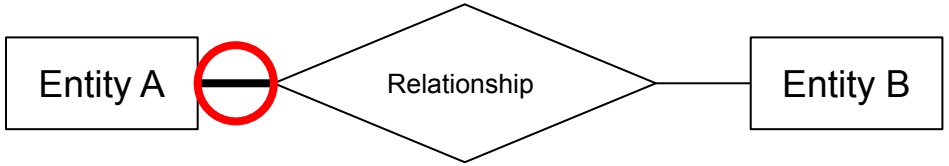
Many-to-One
Many actors can be in one movie

(d)

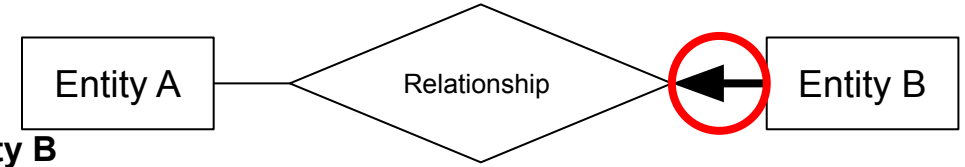
One-to-Many
Many movies can have the same actor

Participation Constraints (“at least one”)

- Heavy line denotes each Entity A must participate in a relationship with at least one Entity B
  - Could participate with more than one
  - No restriction on Entity B
    - Example: 5 Entity B’s do not participate with any Entity A

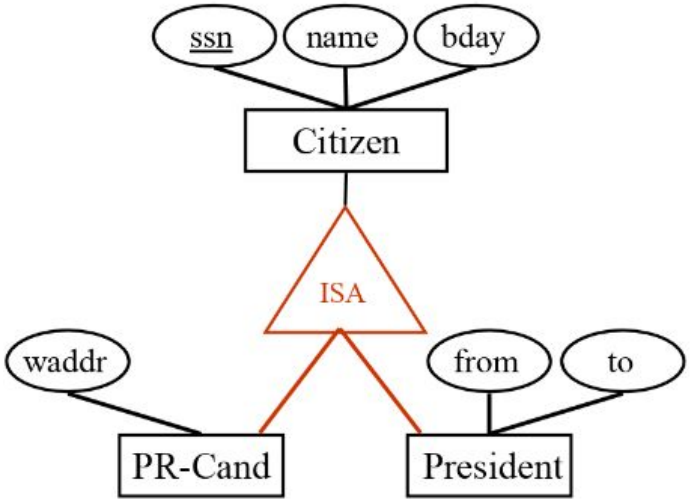


- Heavy line denotes each Entity B must participate in a relationship with at least one Entity A
  - But we know from earlier this arrow means that a single Entity B can relate to at most one Entity A
  - **Net result: 1 and only 1 Entity A per Entity B**



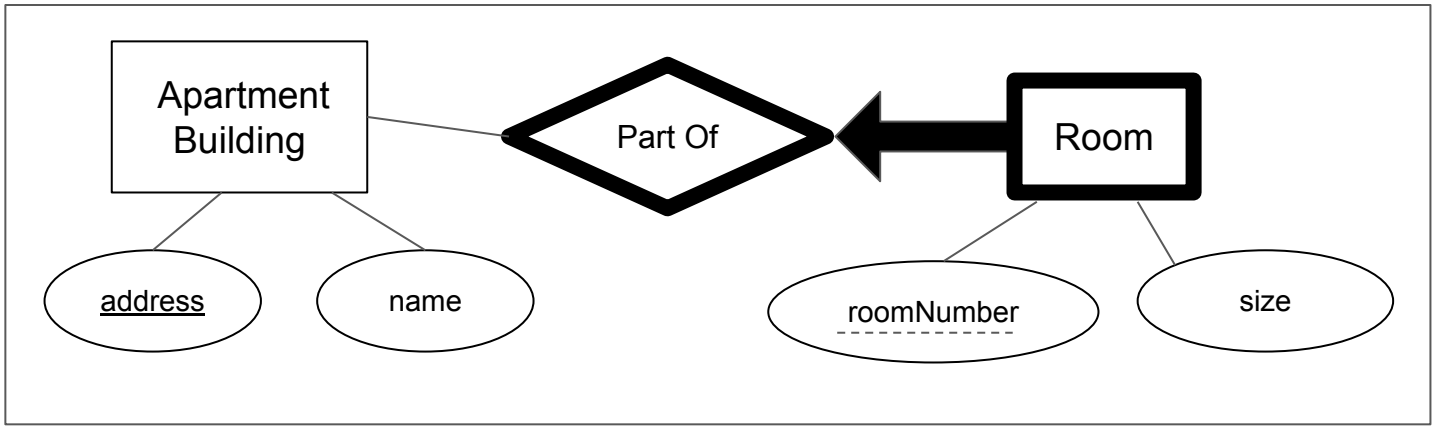
ISA (‘is a’) Hierarchies

- Equivalent of subclasses
  - All attributes from superclass are in subclasses
- Overlapping vs Disjoint
  - Overlapping if two subclasses can contain the same entity. Otherwise disjoint
    - Example A: Each president was a presidential candidate at some point (overlapping)
    - Example B: A student can either be a graduate or an undergraduate (disjoint)
- Covering vs Partial
  - Is the union of all the subclasses the same as the super class?
    - Example A: Are all citizens either presidential candidates or presidents (no - partial)
    - Example B: Are all students either graduate or undergraduates (yes- covering)



Weak Entities

- Weak Entity: Room
  - Partial key: roomNumber
    - Primary key: ApartmentBuilding.address and Room.roomNumber
  - Without a building, you can’t have a room



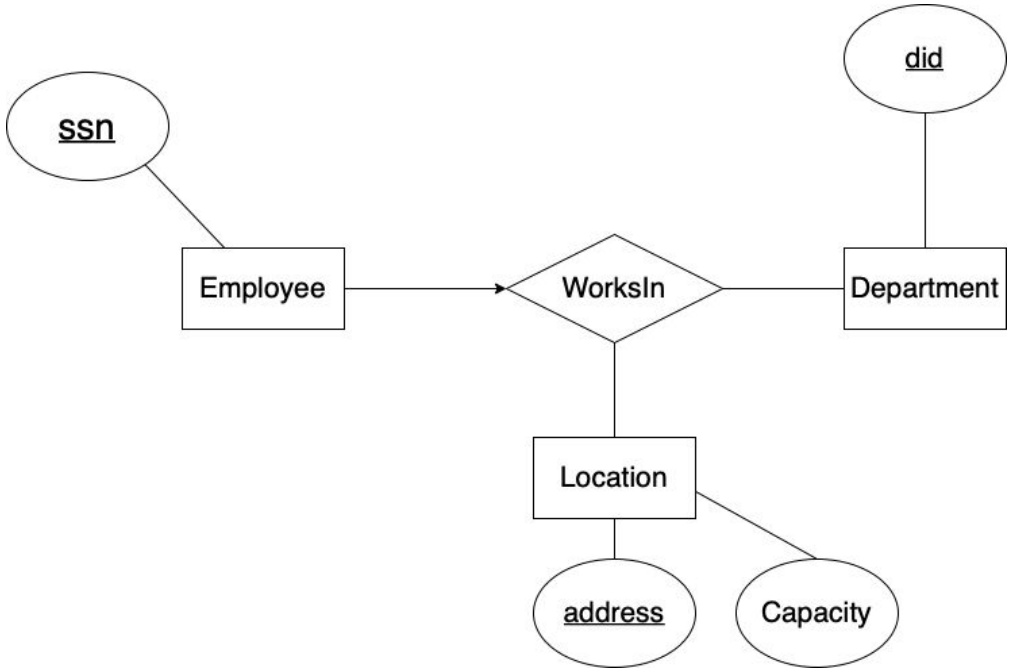
(Fun fact: the only way to convince people that an arrow is bold is to make it comically large)

Key constraints in a ternary relationship

Each employee works in **at most one** combination of department and location.

Example:

1. (E1, D1, L1)
2. ~~(E1, D1, L1)~~
3. ~~(E1, D2, L1)~~
4. ~~(E1, D1, L2)~~
5. (E2, D1, L1)

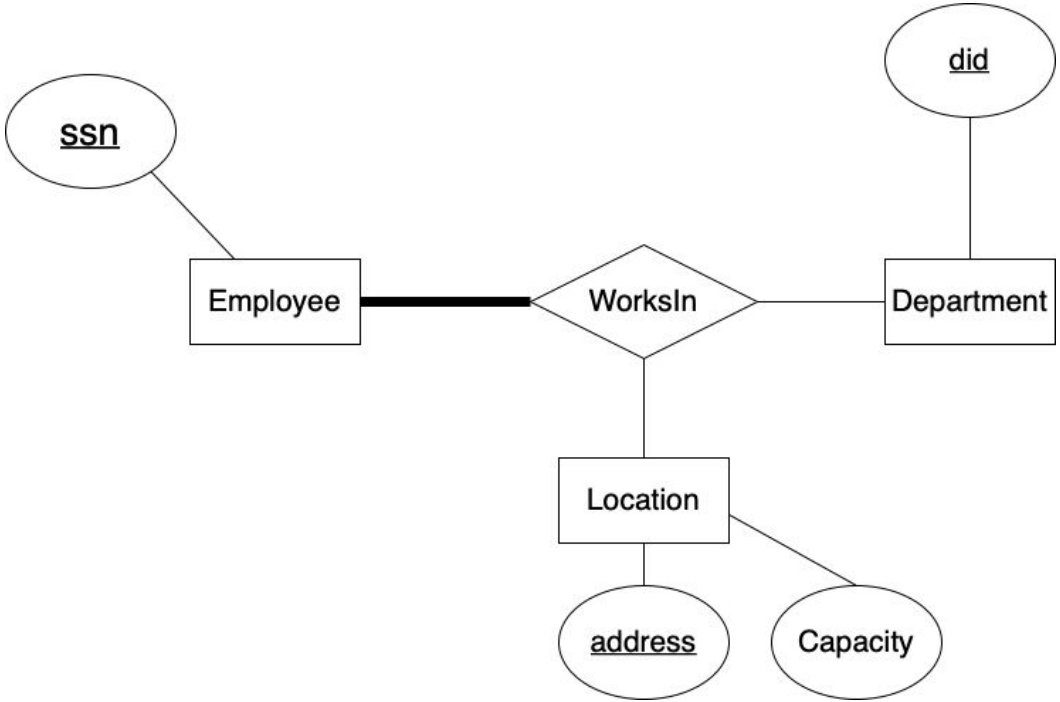


Participation constraints in a ternary relationship

Each employee works in **at least one** combination of department and location.

Example:

1. E has E1, E2
2. (E1, D1, L1)
3. Need (E2, Dx, Ly)



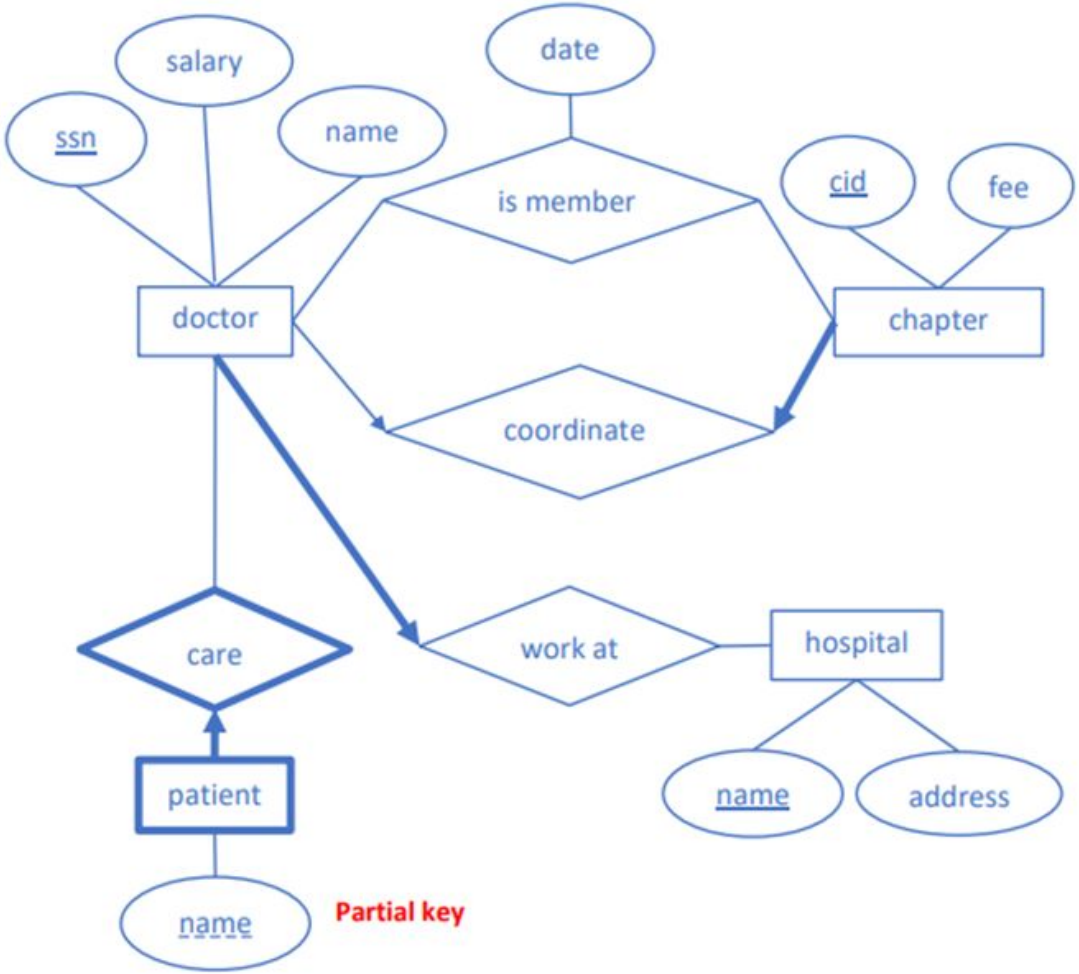
Creating ER Diagrams

- Three key steps to success
  - Start with the entities and relationships
    - Make your entity squares **(no attributes yet)**
    - Make your relationship diamonds between squares
      - Don't worry about constraints yet
    - Handle ISA hierarchies
  - Add in attributes
    - Determine if they should belong to entity or relationship
    - Determine primary keys
  - Resolve constraints
    - Handle key and participation constraints
    - Determine what weak entities exist
    - Check relationships for potential ternary relationships

ER Diagram Example - Hospital

- Each doctor works at **exactly one** hospital. Doctors have name, salary, and a **unique** ssn. Hospitals have address and a **unique** name.
- Each patient **must** be associated with **exactly one** doctor, and no two patients of a given doctor have the same name (though two patients of the different doctors can have the same name).
- In the database, patient tuples should be automatically deleted if the corresponding doctor tuple is deleted.
- Doctors can join zero or more chapters in the American Medical Association. Each chapter should have a **unique** cid, and a membership fee. It is important to maintain the date on which a doctor joined a chapter.
- Each chapter has **exactly one** coordinator, and only doctors can serve as chapter coordinators. No doctor can coordinate more than one chapter.

Solution


salary
ssn
name
date
is member
cid
fee
doctor
chapter
coordinate
care
work at
hospital
patient
name
address
name
Partial key

Project 1 Setup

Helpful Tools

- We’re going to be using SQLPlus for the next two projects
  - Tool to connect to Oracle Databases
  - Need to connect to CAEN to be able to use the SQLPlus CLI
- Requires CAEN account
  - CoE students have default
  - If you do not have one go here

SSH + SCP/rsync

- SSH (Secure Shell)
  - Linux and Mac users will have SSH built into their terminals
  - Windows users can install Windows Subsystem for Linux (WSL) to use SSH (recommended)
  - Can install CAEN VNC Client instead
- SCP (Secure Copy Protocol) and rsync (Remote Sync)
  - Ways to upload files from your local machine to a server
  - Not necessary if you do all of your development on CAEN
- Need to connect to login.engin.umich.edu for both SSH and SCP
  - Will need Duo

SSH + SCP/rsync (Command line commands)

ssh uniqname@login.engin.umich.edu

scp -r [source file/dir] uniqname@login.engin.umich.edu:[target dir]/[target name]

rsync -rtv [source file/dir] uniqname@login.engin.umich.edu:[target dir]/[target name]

Oracle SQL

- To access the Oracle DBMS
  - SSH into CAEN or use the CAEN VNC Client
  - **module load eecs484**
    - Loads some of the tools and programs needed for this class
    - Append this to your ~/.bash_profile
      - You don't need to type this command in every time then!
  - Launch SQLPlus with **rlwrap sqlplus**
  - Enter username/password (next slide)
  - Congrats! You can now run SQL commands!
- Documentation on the various commands here
  - Each DBMS differs slightly from each other

Oracle SQL

- An Oracle account within the CAEN servers has been created for you
  - Username: Your uniqname
  - Initial password: eecsclass
  - SQLPlus will prompt you to change password when you login the first time
  - **DO NOT use quotes or @ in your password**
- Make a post on Piazza or email us at eecs484f24staff@umich.edu if you don't have one for any reason
  - Adding class late, etc.

Associated Tools

https://eecs484db.github.io/f24/tools

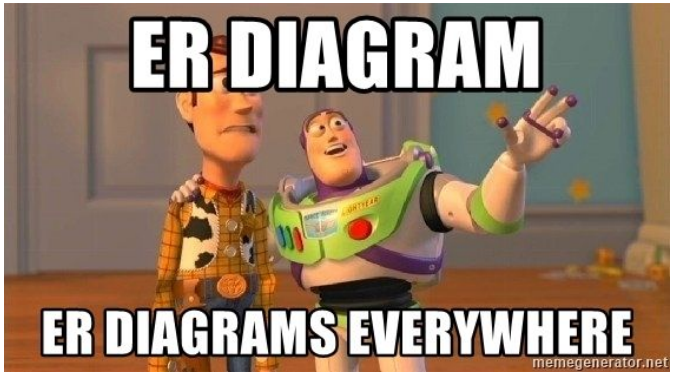
Project 1 Overview

Project 1

- You have recently been contacted by Clark Huckelburg
  - CEO of FakeBook, “the world’s fakest social media platform” ™
  - Need to design a database for them to migrate to
- 4 Parts
  - Draw ER diagram
  - Translate ER diagram and specifications into relational tables
  - Populate database with existing data
  - Create views to make it easier to look at aggregate data
  - Best to go one part a time in order
    - *some* parts will need to be worked on at the same time

Project 1 - Part 1

- Read the spec carefully!
  - Reading later parts will help you with your ER diagram
  - Draw neatly on paper and then scan or use computer tools - LucidChart, draw.io, ...
  - Try the suggested steps to creating an ER diagram (next slide)
    - Determine what's an entity, relationship, and an attribute
    - Figure out which relationships are binary vs ternary



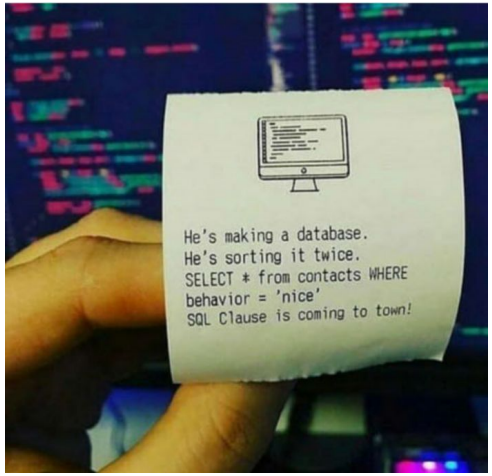
Project 1 - Part 2

- Create the schema!
  - Turn ER diagrams into tables using SQL
    - createTables and dropTables SQL scripts
    - Will also need to create and manage triggers and sequences
  - Need to make sure all constraints are captured
    - Is it NOT NULL? UNIQUE? PRIMARY KEY? FOREIGN KEY?
- Example CREATE statement

```
CREATE TABLE Students (
        student_id INTEGER PRIMARY KEY,
        name VARCHAR(200) NOT NULL
);
```

- Example DROP statement

```
DROP TABLE Students CASCADE CONSTRAINTS;
```



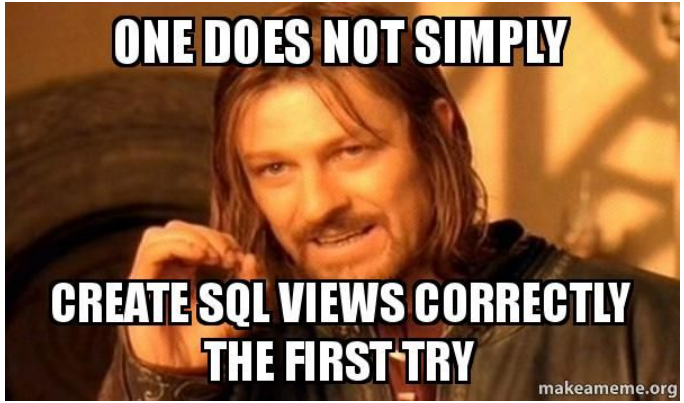
Project 1 - Part 3

- Populate the database
  - Take public data and insert it into your tables

```
INSERT INTO Users
SELECT user_id
FROM project1.Public_User_Information;
```

  - project1 is the schema in which all the data lives
  - Will need to perform unions, joins, and other relational logic to insert data

Project 1 - Part 4



- Create views
  - External schema - Designed to mimic public data set
  - CREATE VIEW Instructor_Name AS
    SELECT I.last_name
    FROM INSTRUCTOR I
    WHERE I.first_name = ‘Bob’;
    - Shows only the last name from instructors where the first name is Bob
- You can check this before submitting
  - SELECT * FROM project1.Public_User_Information
    MINUS
    SELECT * FROM View_User_Information
  - SELECT * FROM View_User_Information
    MINUS
    SELECT * FROM project1.Public_User_Information
  - Checking to make sure it is identical to the public dataset you read from

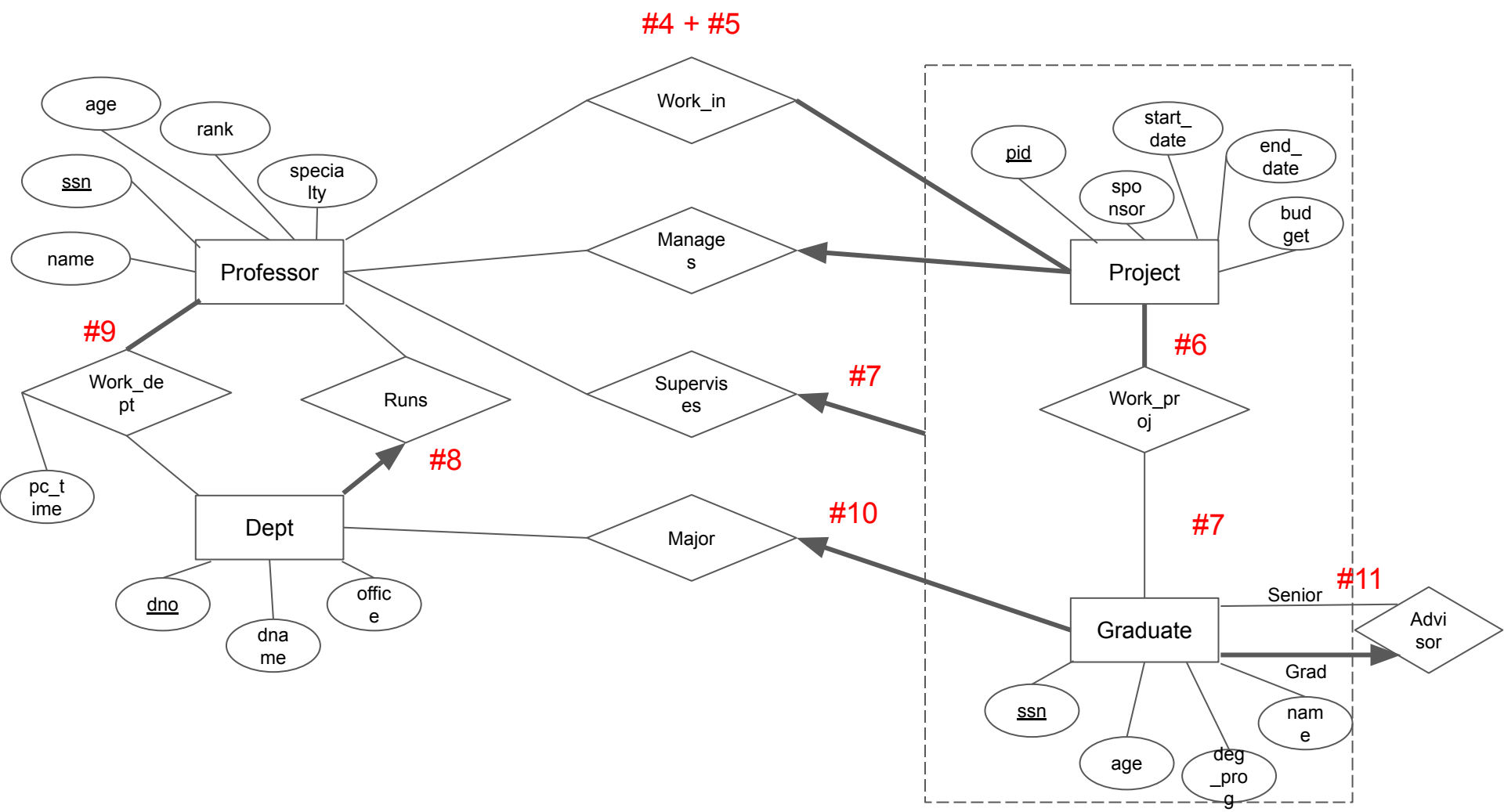
Bonus ER Diagram Practice

ER diagram practice

1. Professors have a unique SSN, a name, an age, a rank, and a research specialty.
2. Projects have a unique project number, a sponsor name, a starting date, an ending date, and a budget.
3. Graduate students have a unique SSN, a name, an age, and a degree program (e.g., M.S. or Ph.D.).
4. Each project is managed by one professor (known as the project's principal investigator) and is worked on by one or more professors (known as the project's co-investigators).
5. Professors can manage and/or work on multiple projects.
6. Each project is worked on by one or more graduate students (known as the project's research assistants).

ER diagram practice

7. When graduate students work on a project, their work on the project must be supervised by exactly one professor. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one.
8. Departments have a unique department number, a department name, and a main office. Departments must have one professor (known as the chairman) who runs the department.
9. Professors work in one or more departments, and for each department that they work in, a time percentage is associated with their job.
10. Graduate students have one major department in which they are working on their degree.
11. Each graduate student has exactly one more senior graduate student (known as a student advisor) who advises him or her what courses to take.


#4 + #5
age
rank
ssn
specia
lty
name
Professor
Work_in
Manage
s
Supervis
es
Work_de
pt
Runs
pc_t
ime
Dept
dno
dna
me
offic
e
#9
#8
#7
#10
pid
start_
date
end_
date
spo
nsor
bud
get
Project
#6
Work_pr
oj
#7
#11
Graduate
Senior
Advi
sor
Grad
ssn
age
deg
_pro
g
nam
e
Major

Get started with HW1!

We’re here if you need any help!!

- Office Hours: Schedule is here, both virtual and in person offered
- Piazza
- Next week’s discussion!!!