

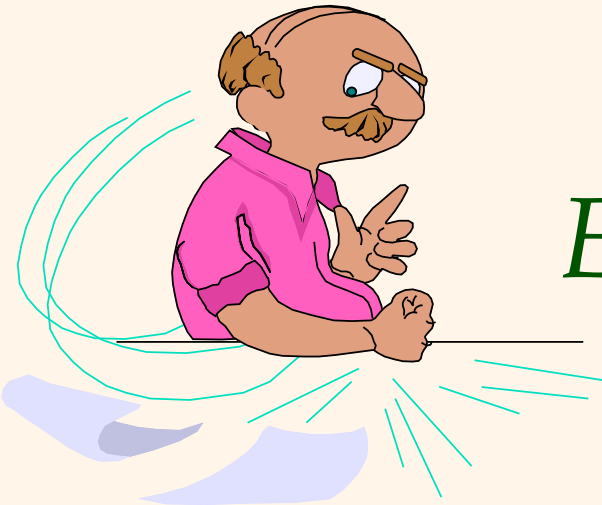


# *Introduction to Data Management*

*\*\*\* The “Flipped” Edition \*\*\**

## *Lecture #2*

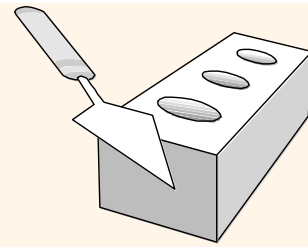
## *E-R Database Design*



Instructor: Mike Carey  
mjcarey@ics.uci.edu



# Today's Notices



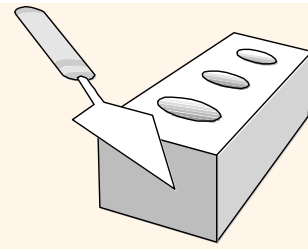
- ❖ Frequently check the course wiki page:
  - <https://grape.ics.uci.edu/wiki/asterix/wiki/cs122a-2021-fall>
- ❖ And totally camp out on the Piazza page:
  - <http://piazza.com/uci/fall2021/cs122aeecs116/home>
- ❖ Also get yourselves buddied up!
  - We'll share the purpose in the first HW assignment
- ❖ Quizzes will be open until Fridays at start of class time
  - Don't miss them – they're free points!
- ❖ Any questions?
  - Just kidding... sort of .... (😬)



# *The Database Design Process*

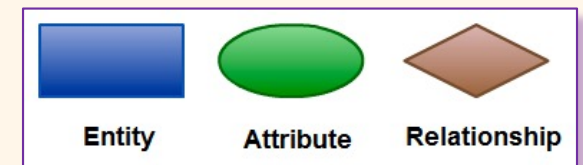
- ❖ Essentially a top-down process.
  1. Requirements gathering (interviews)
  2. Conceptual design (using E-R model)
  3. Platform selection (which DBMS?)
  4. Logical design (for target data model)
  5. Physical design (for target DBMS & workload)
  6. Implement (and test, of course 😊)
- ❖ *Notes:*
  - Expect backtracking, iteration, and then incremental changes over time
  - Our targets: Relational model & RDBMSs

# Steps 1 & 2 in Database Design



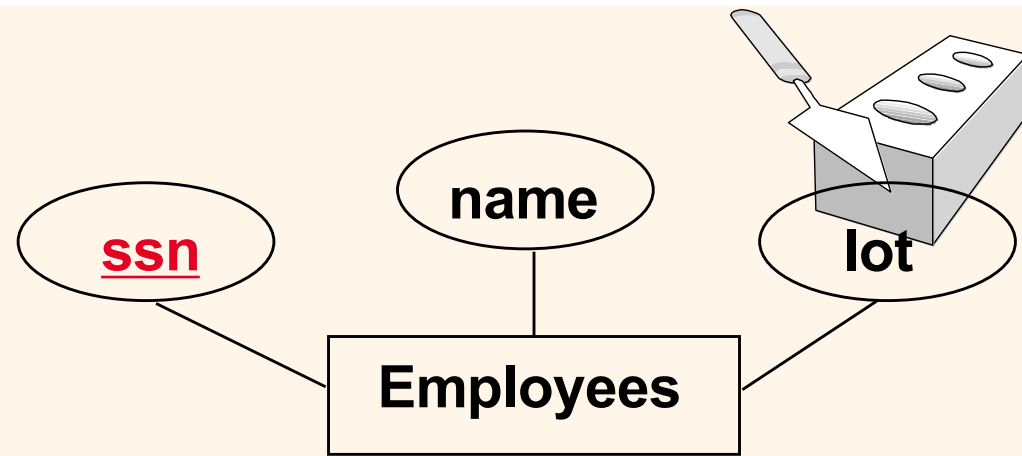
## ❖ Conceptual design (ER-based):

- What are the *entities* and *relationships* in the enterprise?
- What information about these entities and relationships should we store in the database?
- What are the *integrity constraints* or *business rules* that hold?
- A database schema in the ER Model can be represented pictorially (using an *ER diagram*).
- Can map an ER diagram into a relational schema (manually or using a design tool's automation).



# ER Model Basics

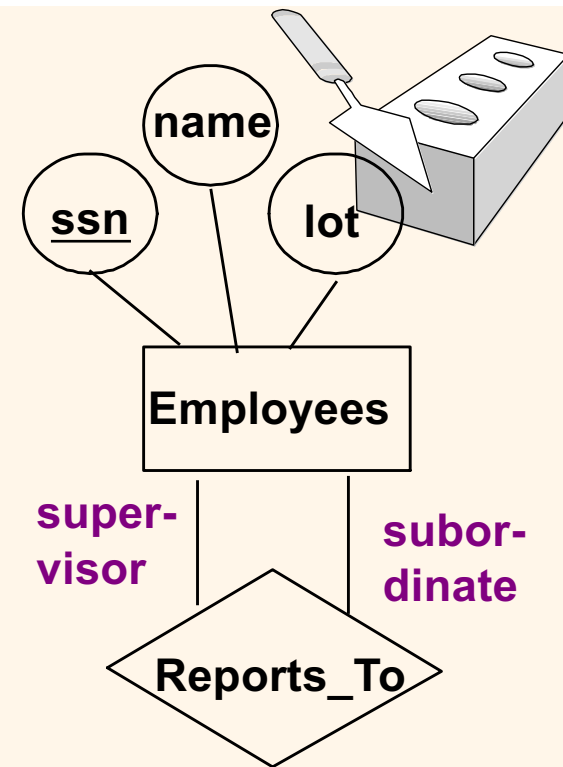
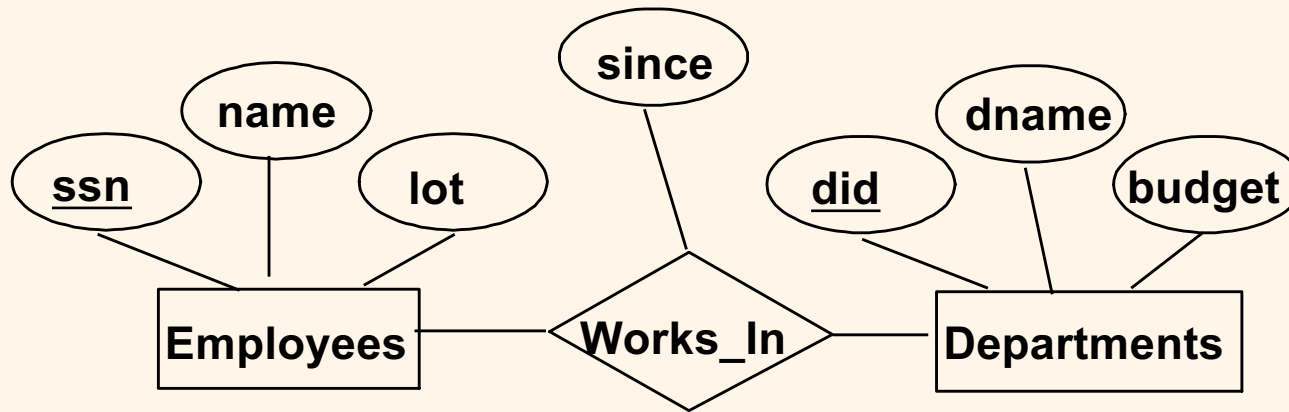
## (Entities)



- ❖ Entity: Real-world object, distinguishable from all other objects. An entity is described (in DB-land) using a set of attributes.
- ❖ Entity Set: A collection of similar entities. E.g., all employees.
  - All entities in an entity set have the same set of attributes. (Until we get to ISA hierarchies...)
  - Each entity set has a key (a unique identifier); this can be either one attribute (an “atomic” key) or several attributes (called a “composite” key)
  - Each attribute has a domain (similar to a data type).

# ER Model Basics

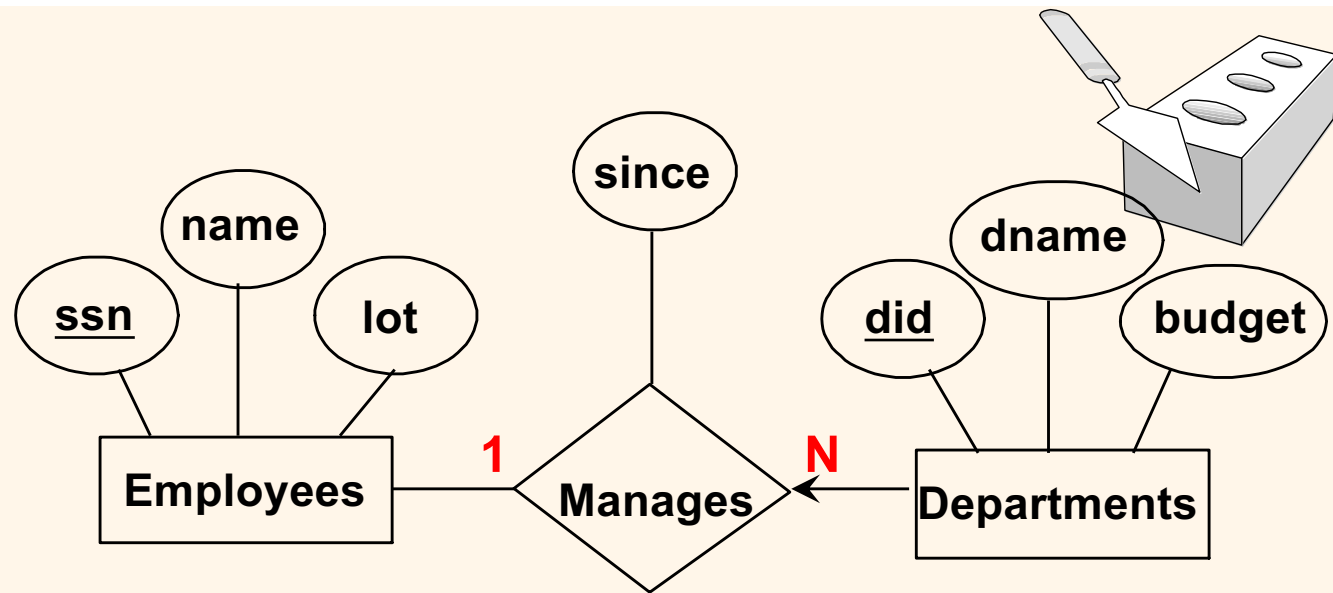
## (Relationships)



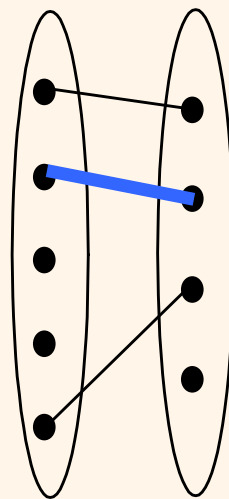
- ❖ Relationship: Association among two or more entities. E.g., Santa Claus works in the Toy department.
- ❖ Relationship Set: Collection of similar relationships.
  - An n-ary relationship set R relates n entity sets E1 ... En; each relationship in R involves entities e1:E1, ..., en:En
    - One entity set can participate in different relationship sets – or in different “roles” in the same set.

# Cardinality Constraints

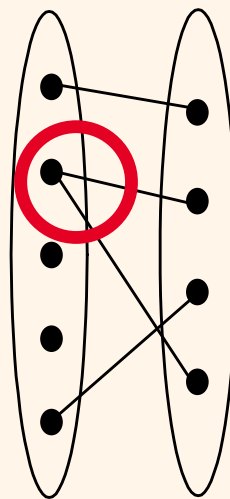
- ❖ Consider Works In:  
An employee can work in many departments; a dept can have many employees.
- ❖ In contrast, each dept has at most one manager, according to the cardinality constraint on Manages above.



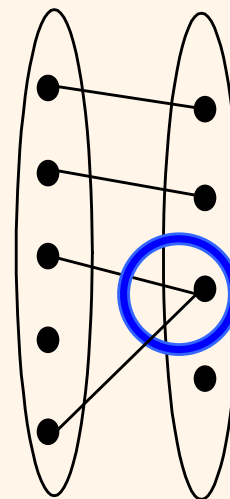
(Note: A given employee can manage several departments)



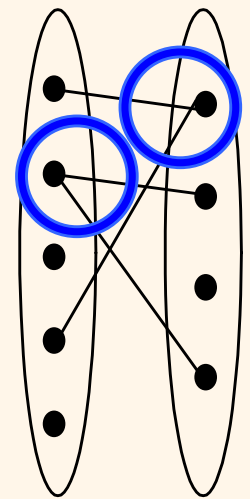
1-to-1  
(1:1)



1-to Many  
(1:N)

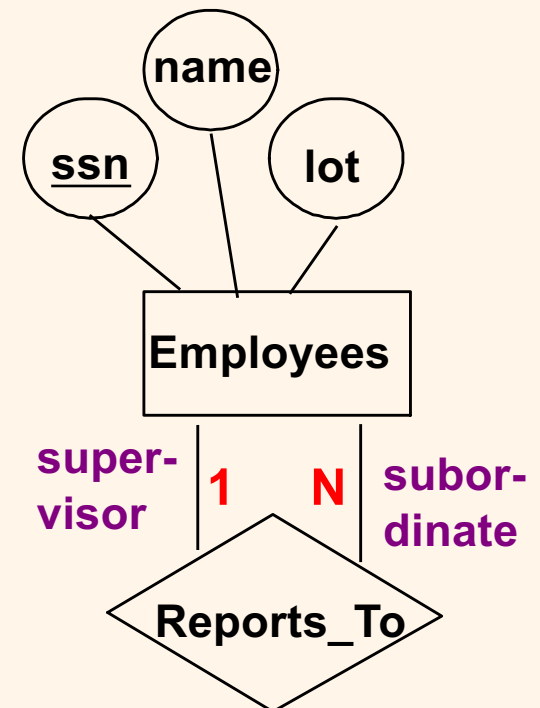
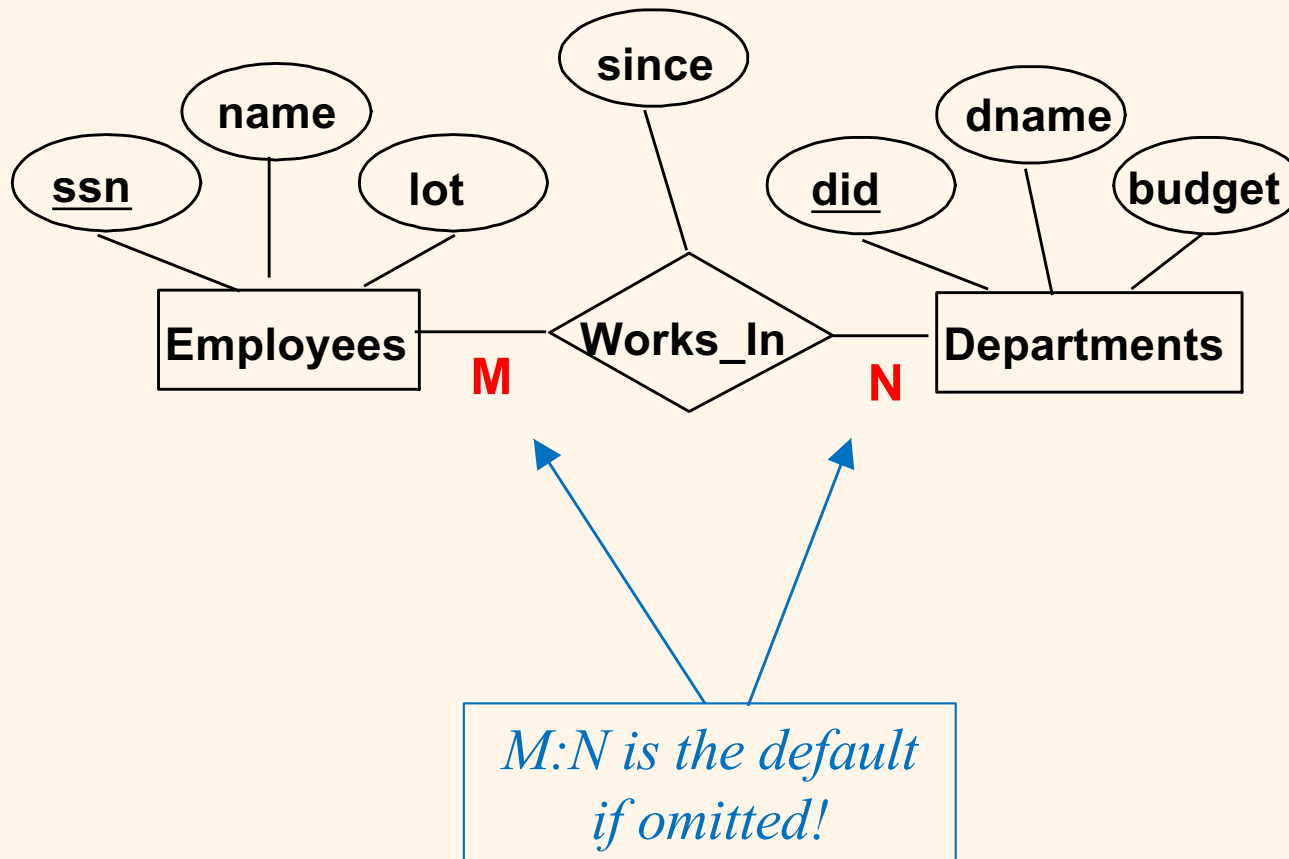
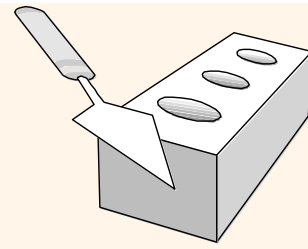


Many-to-1  
(N:1)



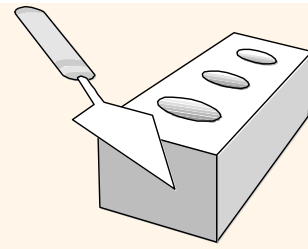
Many-to-Many  
(M:N)

# Revisiting Our Official Notation

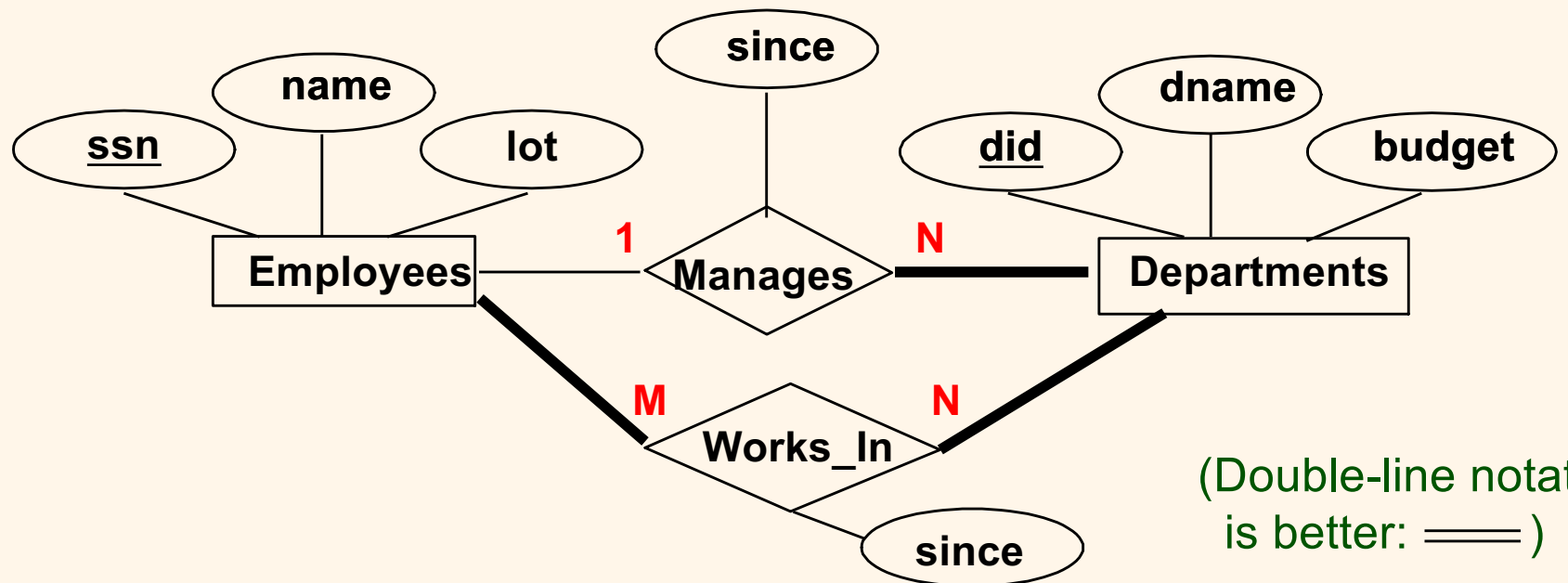




# Participation Constraints

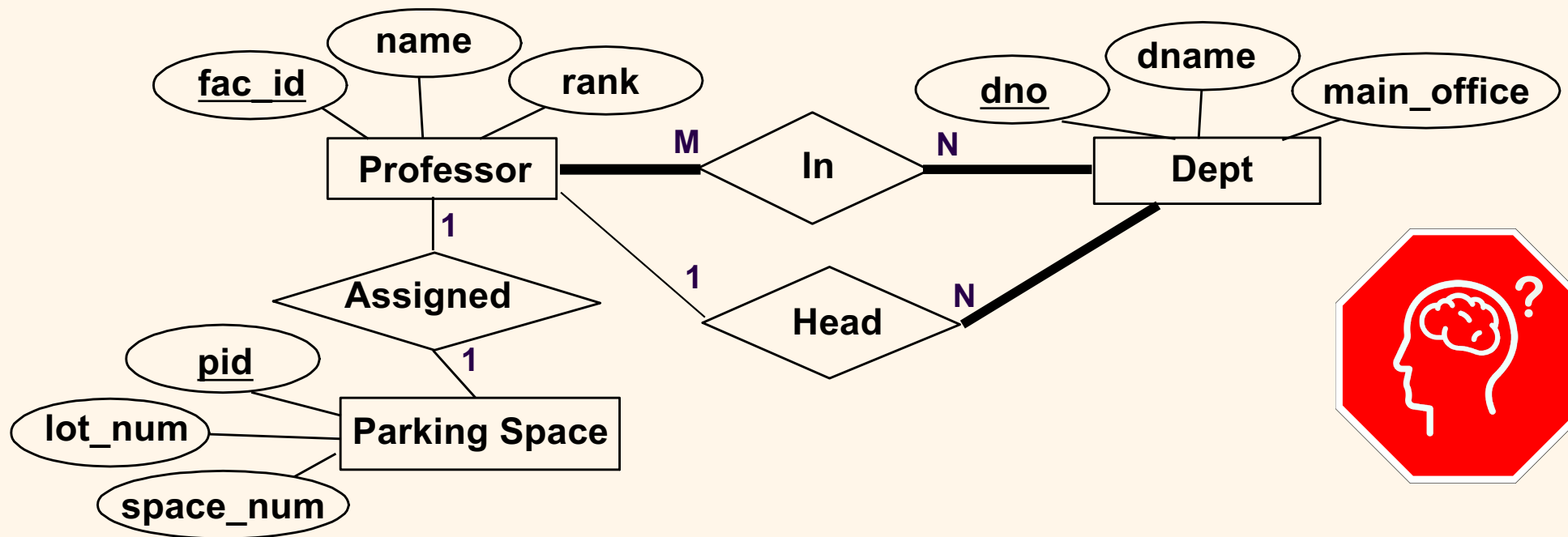
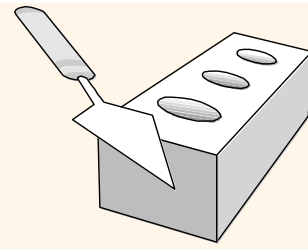


- ❖ Does every department have a manager?
  - If so, this is a *participation constraint*: the participation of Departments in Manages is said to be *total* (vs. *partial*).
    - Every Departments entity below *must* appear in an instance of the Manages relationship
    - Ditto for *both* Employees and Departments for Works\_In



(Double-line notation is better: ==)

# ER Basics: Another Example



(Note that I'm using the M:N notation, and **no**  $\longrightarrow$ 's, here.)

- ❖ Let's see if you can read/interpret the ER diagram above...! (☺)
  - What attributes are unique (i.e., identify their associated entity instances)?
  - What are the rules about (the much coveted) parking spaces?
  - What are the rules (constraints) about professors being in departments?
  - And, what are the rules about professors heading departments?



# Another Example (Answers)

## ❖ Unique attributes:

- *Professor.fac\_id, Dept.dno, Parking Space.pid*

## ❖ Faculty parking:

- 1 space/faculty, one faculty/space
- Some faculty can bike or walk (☺)
- Some parking spaces may be unused

**NOTE:** These things are all “rules of the universe” that are just being *modeled* here!

## ❖ Faculty in departments:

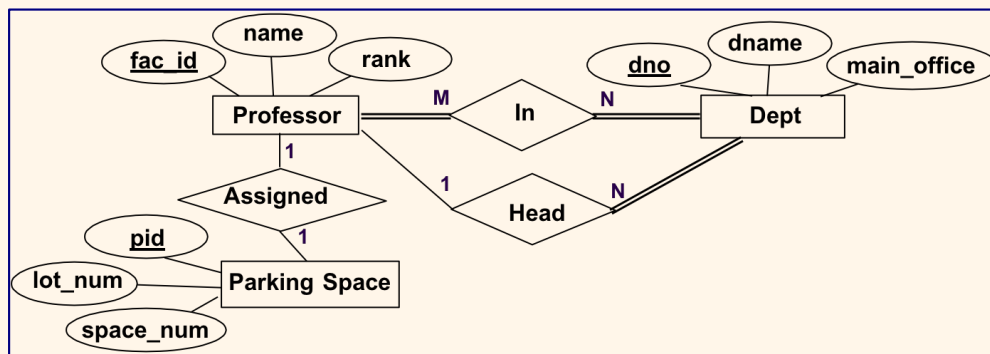
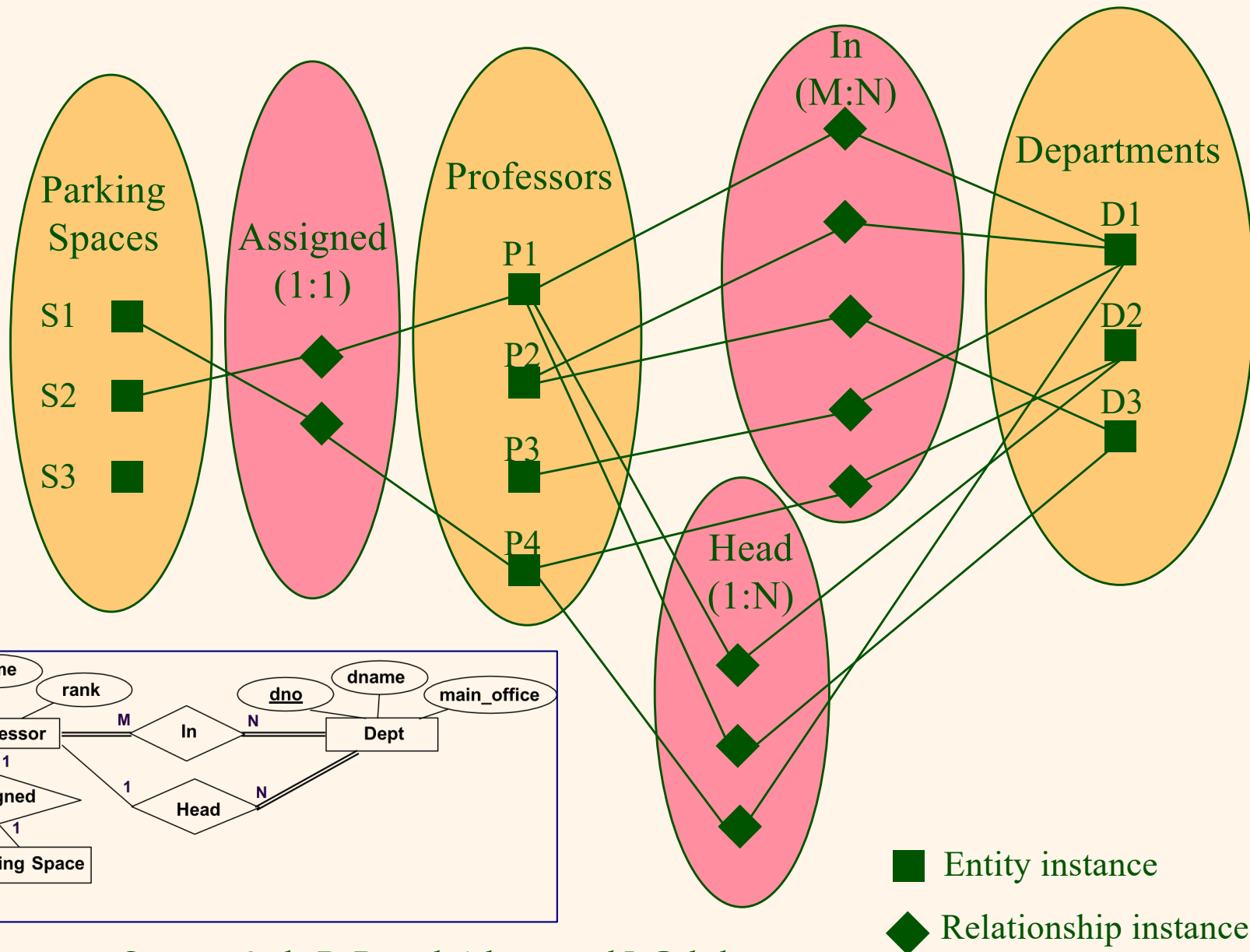
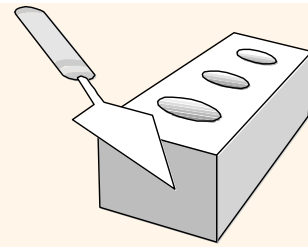
- Faculty may have appointments in multiple departments
- Departments can have multiple faculty in them
- No empty departments, and no unaffiliated faculty

## ❖ Department management:

- One head per department (exactly)
- Not all faculty are department heads

**Q:** Can a faculty member head a department that he or she isn't actually in?

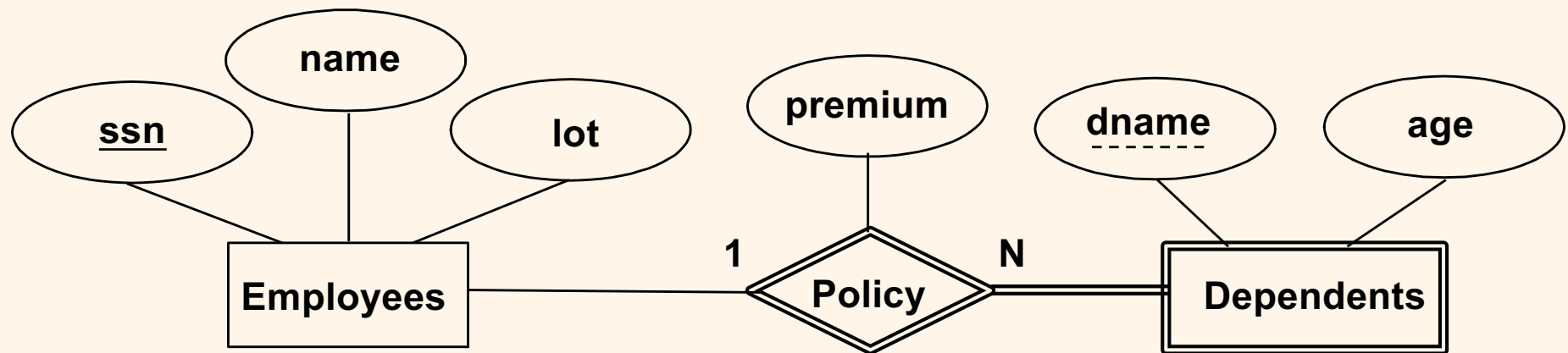
# Another Example (E's & R's)



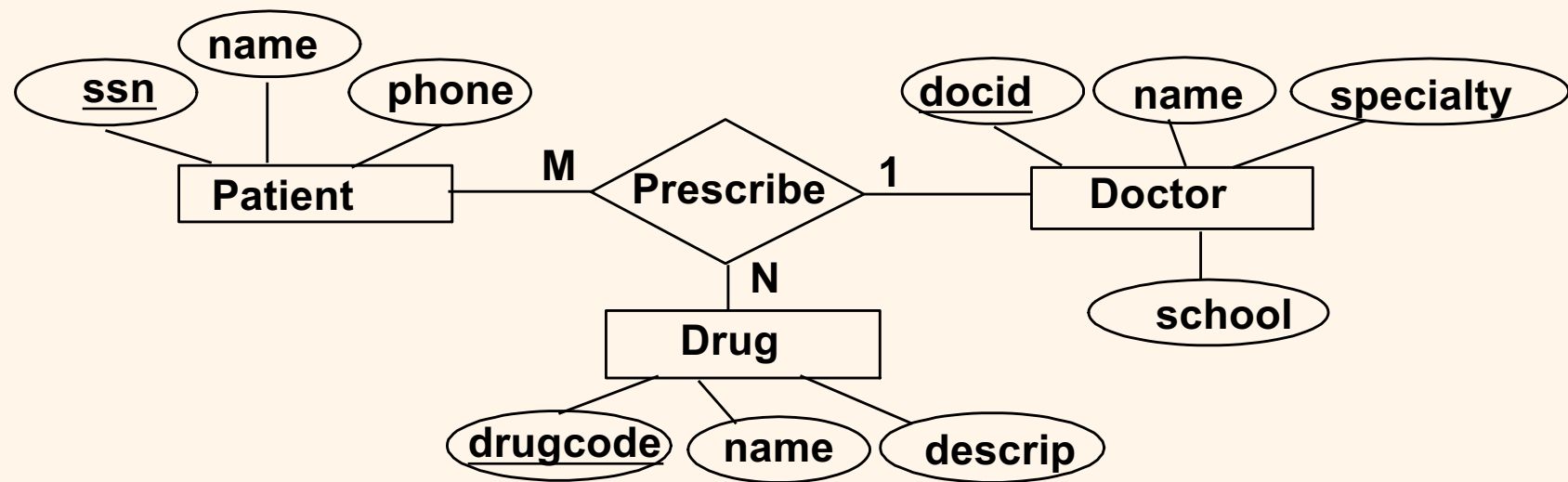
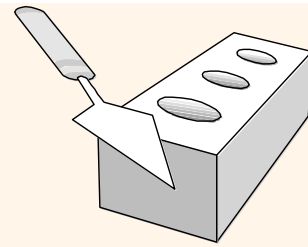


# Weak Entities

- ❖ A *weak entity* can be identified uniquely only by considering the primary key of some other (*owner*) entity.
  - Owner entity set and weak entity set must participate in a one-to-many relationship set (one owner, many weak entities).
  - Weak entity set must have *total* participation in this *identifying* relationship set.
  - Dependent identifier is unique only *within* owner context (\_\_\_\_), so its fully qualified key here is (ssn, dname)



# Ternary Relationships (and beyond)



- ❖ A prescription is a 3-way relationship between a patient, a doctor, and a drug; with the cardinality constraints above:
  - A given patient+drug will be associated with *one* doctor (1)
  - A given patient+doctor may be associated with *several* drugs (N)
  - A given doctor+drug may be associated with *several* patients (M)
- ❖ **General note:** Relationship key  $\leq$  (entity keys)

*To Be Continued...*

