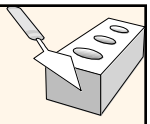
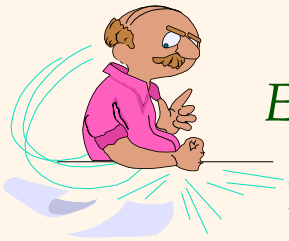
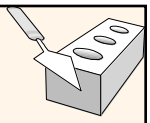


Introduction to Data Management



Lecture #3
E-R Model, Continued

Instructor: Mike Carey
mjcarey@ics.uci.edu



It's time for....

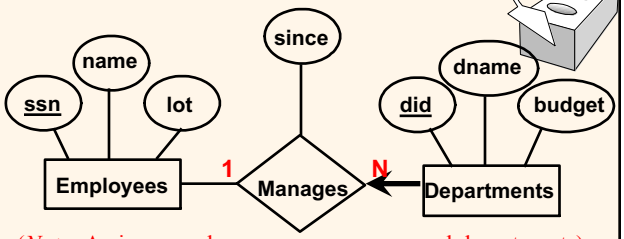
Brought to you by

Today's Reminders

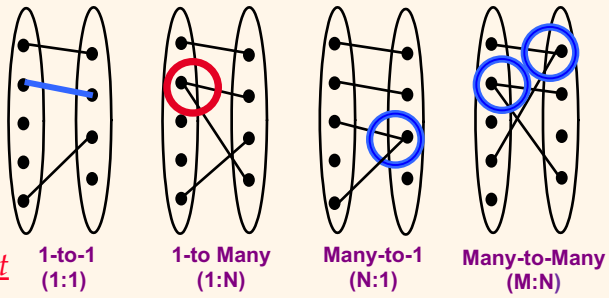
- Read (and live by!) the course wiki page:
  - http://www.ics.uci.edu/~cs122a/
- Also follow (and live by) the Piazza page:
  - https://piazza.com/uci/spring2019/cs122a/home
  - Everyone needs to get signed up!
  - The first HW assignment is now available…
  - Personal Health Logger – ***PHLOG.com*** (☺)
- I will be out of town next week (sorry!!)
  - ICDE (http://conferences.cis.umac.mo/icde2019/)
  - Shiva (and Xikui) will cover the next three lectures
  - I expect to pay attention to Piazza while in Macau

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)

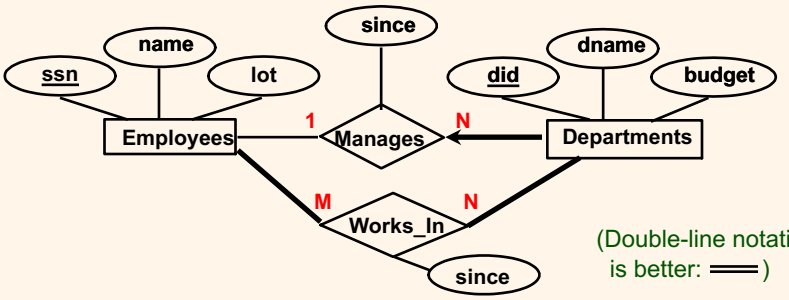


无论是哪一个，都可以是0

Participation Constraints

所以双虚线是指全部都有这种关系，比如这里，不是所有的employee都会manage department 但是所有的employees都会work in department 都需要有manage，然后箭头是指，我也不知道

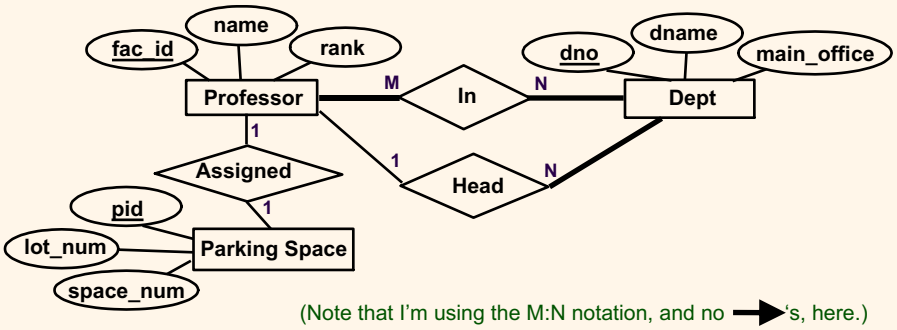
- Does every department have a manager?
  - If so, this is a *participation constraint*: the participation of Departments in Manages is said to be *total* (vs. *partial*).
    - Every Departments entity below *must* appear in an instance of the Manages relationship
    - Ditto for *both* Employees and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



(Double-line notation is better: ═══)

就是有departments的就必须先有manages

ER Basics: Another Example



(Note that I'm using the M:N notation, and no →'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 passes?
  - What are the rules (constraints) about professors being in departments?
  - And, what are the rules about professors heading departments?

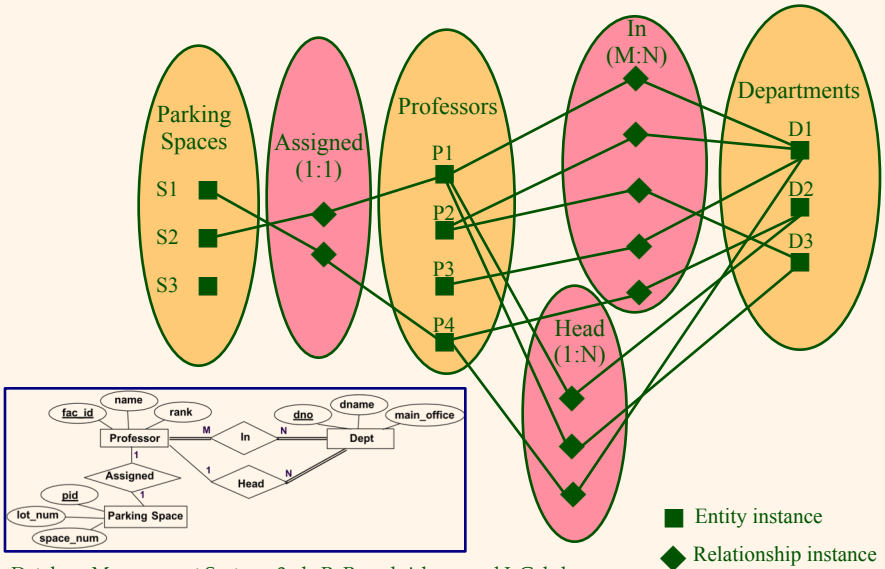
Another Example (Answers)

- 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
- 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

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

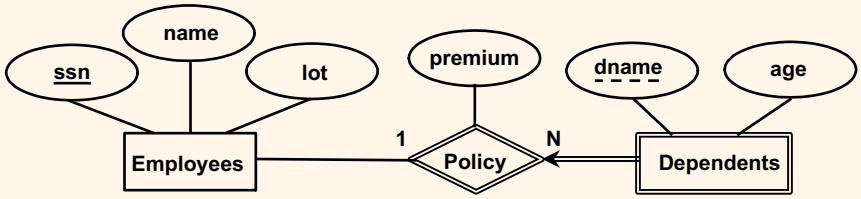
Q: Can a faculty member head a department that he or she isn't actually in?

Another Example (E's & R's)

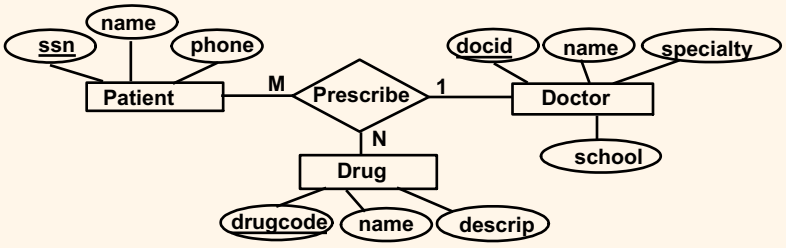


Weak Entities

- 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 (dashed), so its fully qualified key here is (ssn, dname)

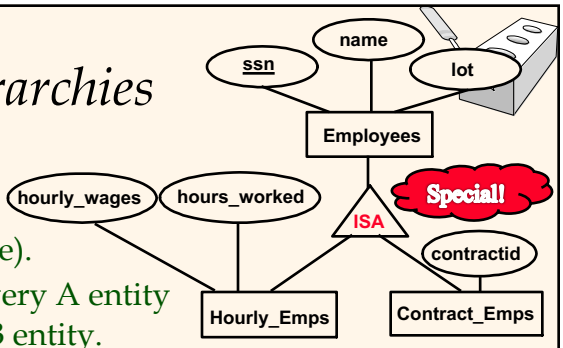


Ternary Relationships (and beyond)



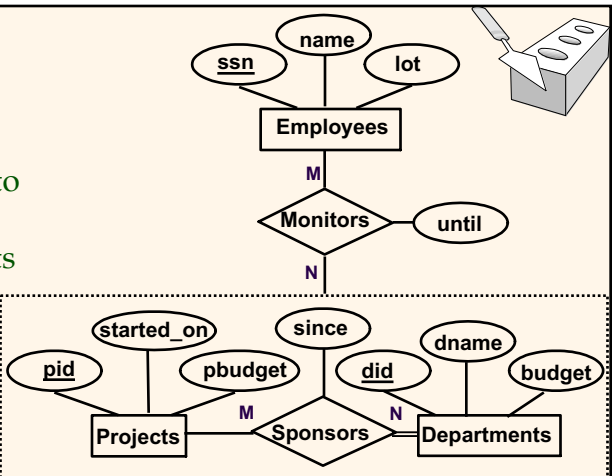
- A prescription is a 3-way relationship between a patient, a doctor, and a drug; with the cardinality constraints above:
  - A given patient+drug will be associated with *one* doctor (1)
  - A given patient+doctor may be associated with *several* drugs (N)
  - A given doctor+drug may be associated with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

ISA ("is a") Hierarchies



- As in Java or other PLs, ER attributes are inherited (including the key attribute).
- If we declare A **ISA** B, every A entity is also considered to be a B entity.
  - *Covering constraints*: *Must every Employees entity be either an Hourly_Emps or a Contract_Emps entity? (Yes* or ***no)***
    - ***Ex:*** Hourly_Emps AND Contract_Emps COVER Employees (pick 1 of **2** vs. 1 of 3)
  - *Overlap constraints: Can some Employees entity be an Hourly_Emps* ***as well as*** *a Contract_Emps entity? (Allowed* or ***disallowed)***
    - ***Ex:*** Hourly_Emps OVERLAPS Contract_Emps (else pick 1 of the **3** types)
  - Reasons for using ISA: 这是一个书面的constrints，从图看不出来
    - To add descriptive attributes specific to a subclass.
    - To identify subclasses that participate in a relationship.
  - Design: specialization (top-down), generalization (bottom-up)

Aggregation

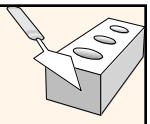


- Used when we have to model a relationship involving (entitity sets and) a *relationship set*.
  - *Aggregation* allows us to treat a relationship set as an entity set for purposes of participating in (other) relationships.

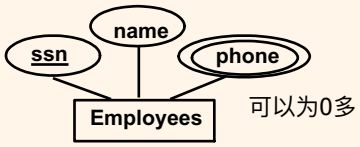
☛ *Aggregation vs. ternary relationship*:

- Monitors is a distinct relationship; even has its own attribute here.
- Each sponsorship can monitored by zero or more employees (as above).

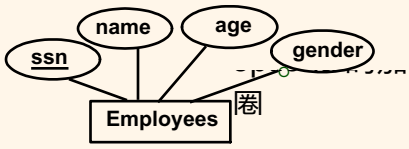
Additional Advanced ER Features



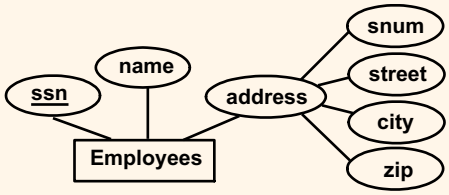
- Multi-valued (vs. single-valued) attributes



- Optional attributes



- Composite (vs. atomic) attributes



- Derived (vs. base/stored) attributes