CS460: Intro to Database Systems

Class 3: *The Entity-Relationship Model*

Instructor: Manos Athanassoulis

https://bu-disc.github.io/CS460/

The Entity-Relationship Model

Basic ER modeling concepts

Readings: Chapters 2.1-2.3

Constraints

Complex relationships

Conceptual Design

Databases Model the Real World

"Data Model" allows us to translate real world things into structures that a computer can store

Many models: Relational, ER, O-O, Network, Hierarchical, etc.

Relational

Rows & Columns

Keys & Foreign Keys to link Relations

Enrolled

sid	cid	grade
53666	Carnatic101	5
53666	Reggae203	5.5
53650	Topology112	6
53666	History105	5

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	5.4
53688	Smith	smith@eecs	18	4.2
53650	Smith	smith@math	19	4.8

Database Design

Requirements Analysis

user needs; what must database do?

Conceptual Design

high level description (often done w/ ER model)

Logical Design

translate ER into DBMS data model

Schema Refinement

consistency, normalization

Physical Design

indexes, disk layout

Security Design

who accesses what

Database Design

Requirements Analysis

user needs; what must database do?

Conceptual Design

high level description (often done w/ ER model)

Logical Design

translate ER into DBMS data model

Schema Refinement

consistency, normalization

Physical Design

indexes, disk layout

Security Design

who accesses what

Conceptual Design

entities and *relationships*

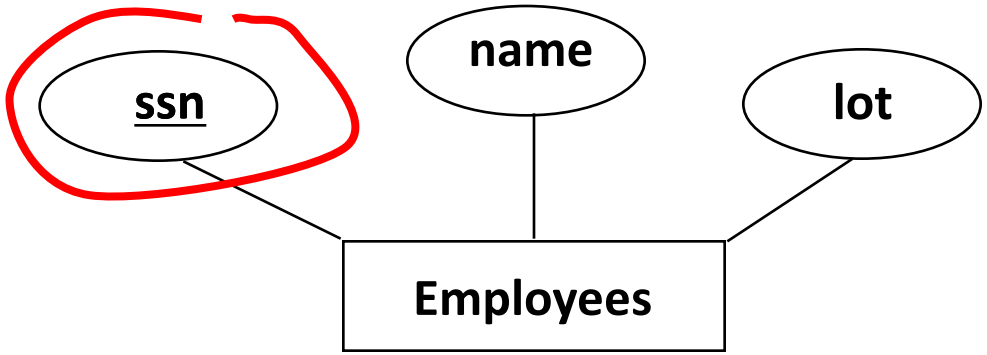
what should we store for each? 

what are the *constraints* that can be held? 

a database "schema" in the ER Model can be represented pictorially (*ER diagrams*)

ER diagrams are mapped to relational schemas

ER Model Basics



Entity: real-world object,
described (in DB) using a set of *attributes*

key?

Entity Set: a collection of similar entities

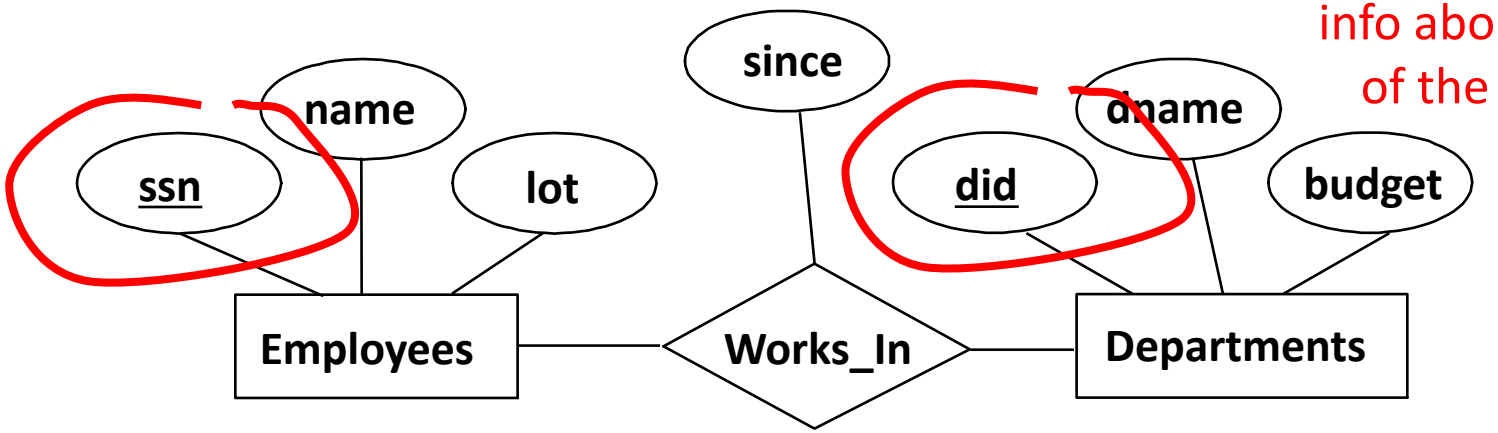
(all employees)

- entities in an entity set have the same attributes
- each entity set has a *key*
- each attribute has a *domain*

ER Model Basics (Contd.)

info about the beginning of the appointment?

key?

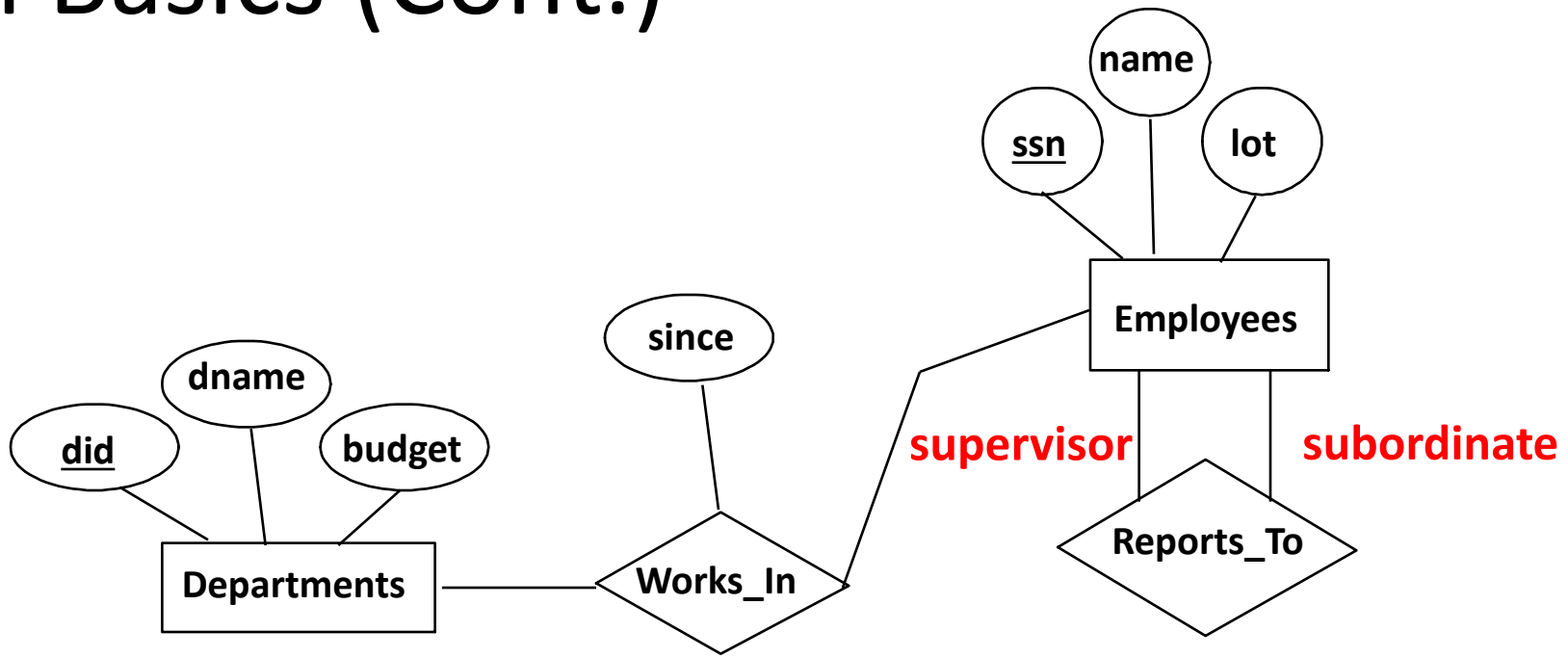


Relationship: association among two or more entities:
"Fred works in Pharmacy department"

relationships can have their own attributes

Relationship Set: collection of (similar) relationships

ER Model Basics (Cont.)



entity set can participate in different relationship sets

or

in different “roles” in the same set

The Entity-Relationship Model

Basic ER modeling concepts

Constraints

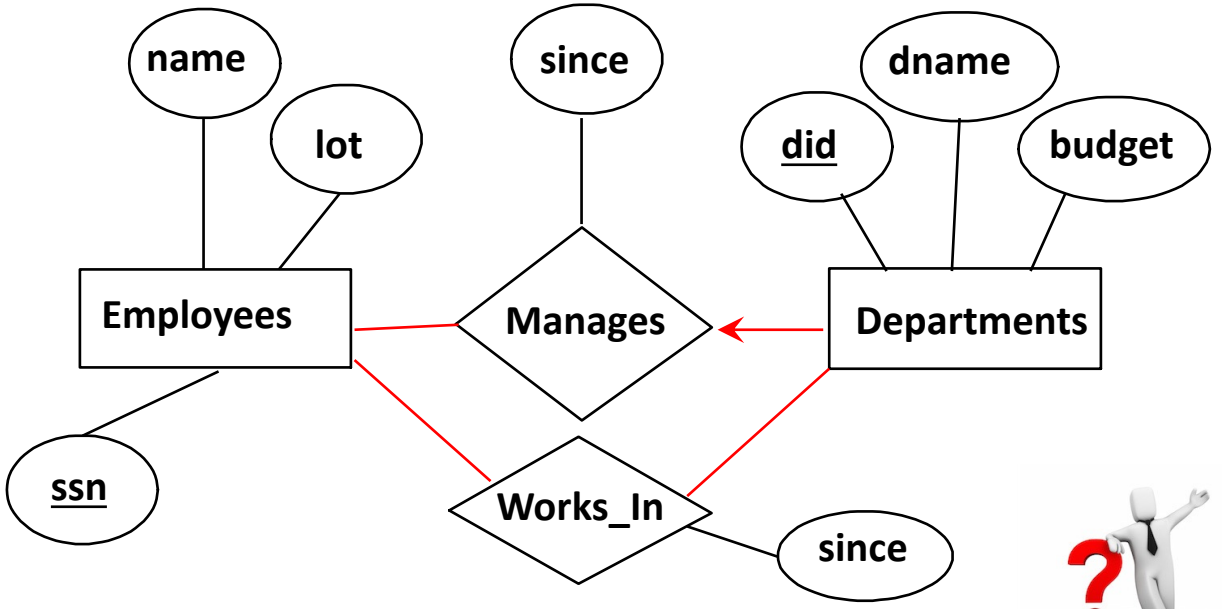
Readings: Chapters 2.4-2.4.3, 2.5.3

Complex relationships

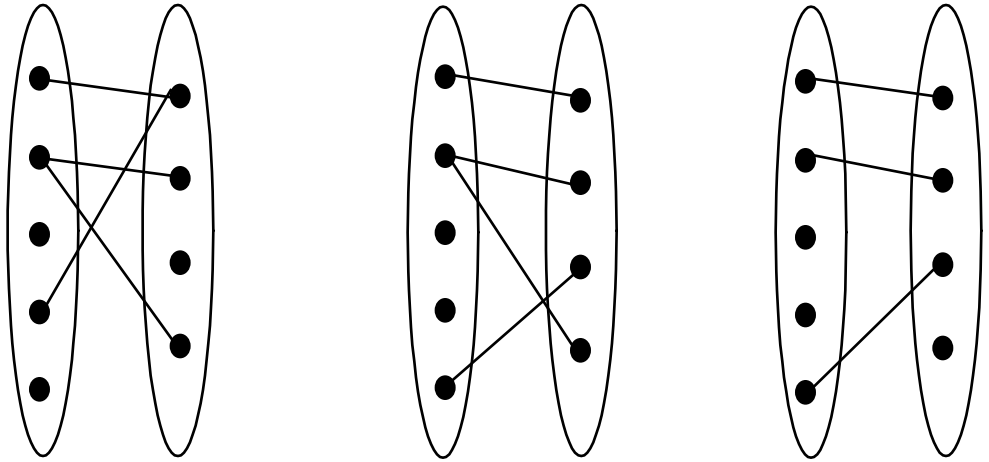
Conceptual Design

Key Constraints

An employee can work in many departments; a department can have many employees



In contrast, each department has at most one manager, according to the *key constraint* on Manages



Many-to-Many **1-to Many** **1-to-1**

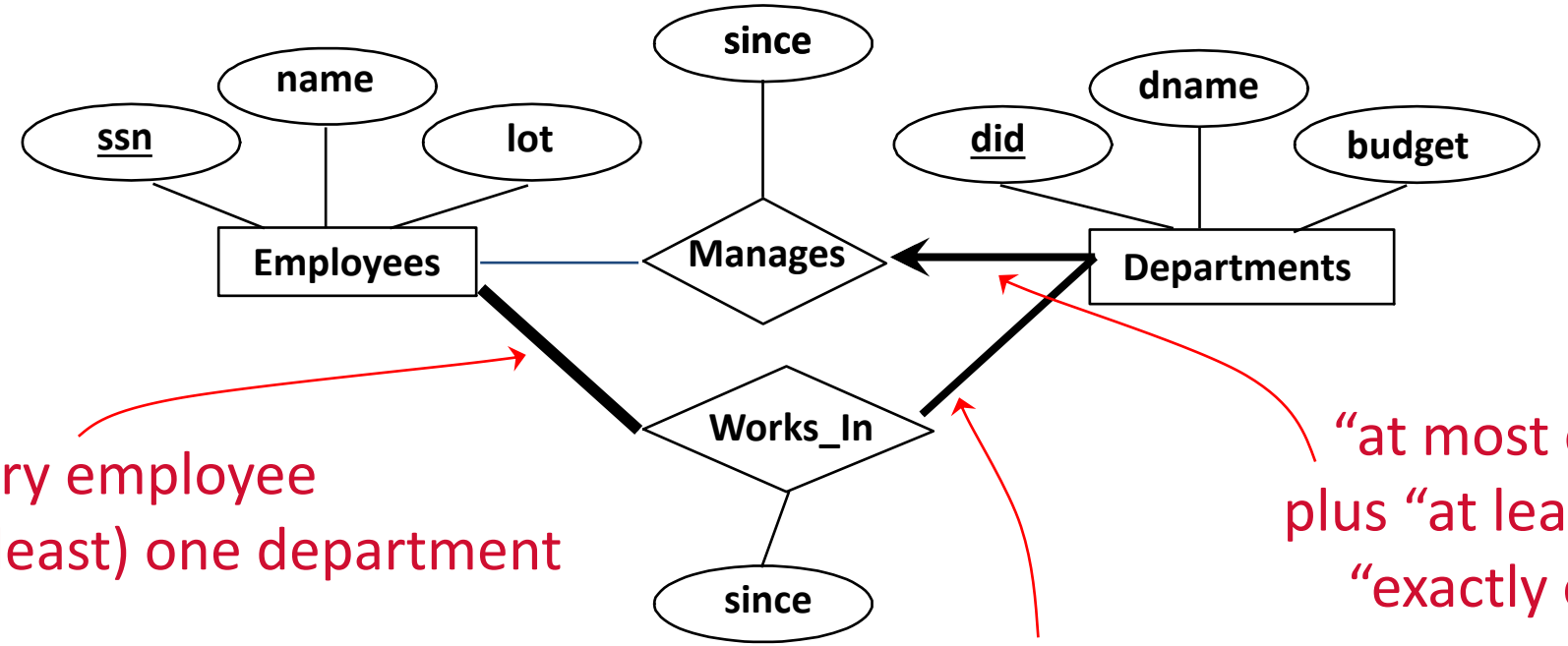
Participation Constraints

does every employee work in a department?

If so, this is a *participation constraint*

the participation is said to be *total* (vs. *partial*)

Basically means "at least one"



every employee
works in (at least) one department

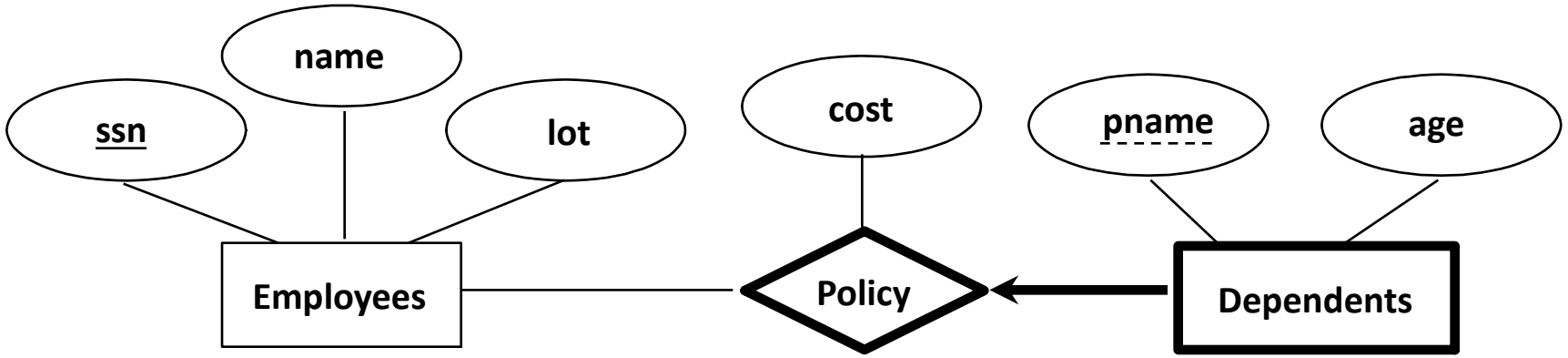
"at most one"
plus "at least one"
"exactly one"

every department
has (at least) one employee

Weak Entities

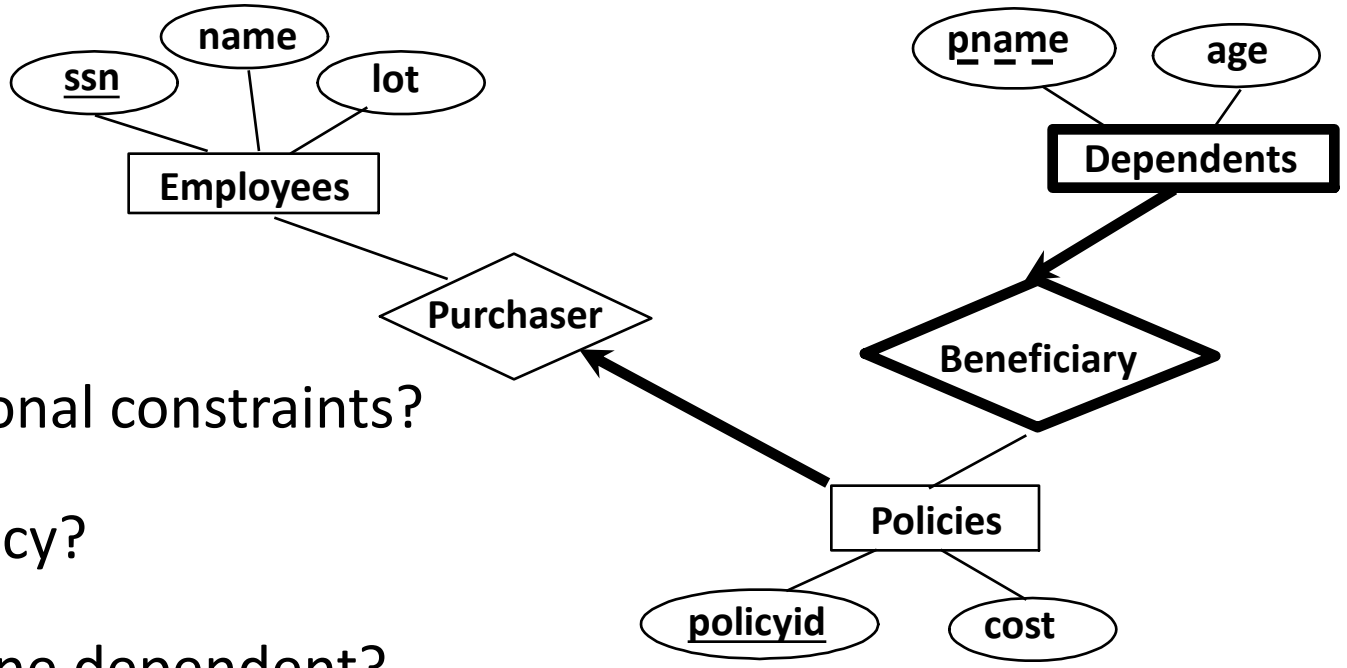
A *weak entity* can be identified uniquely by the primary key of another (*owner*) entity (+ some of its attributes)

- 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



Weak entities have only a “partial key” (dashed underline)

More Elaborate (and Realistic) Example

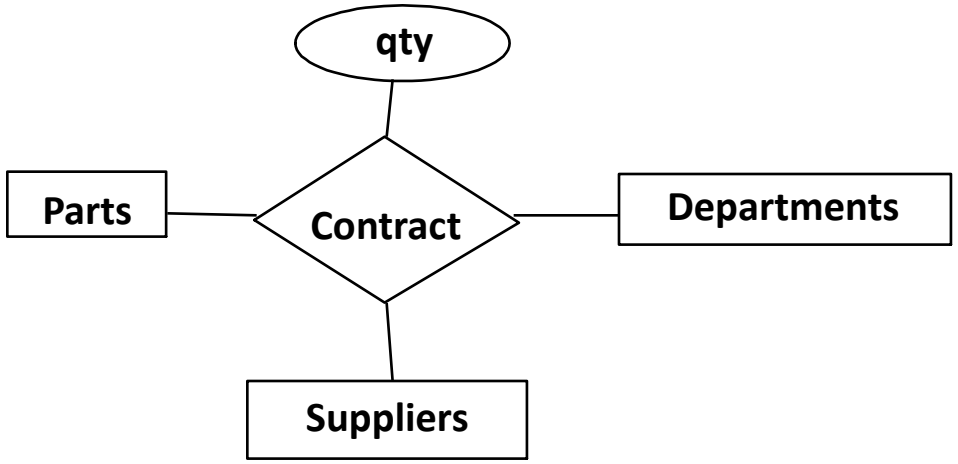


Should we add any additional constraints?

every employee a policy?

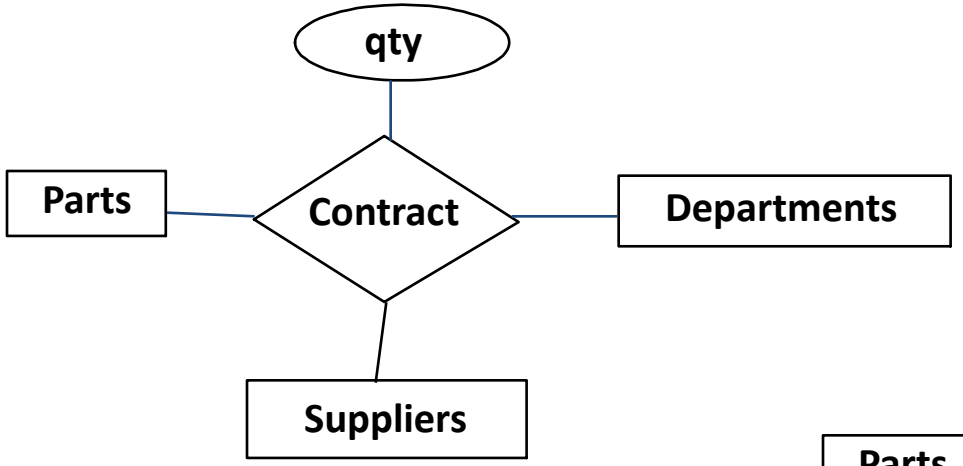
every policy at least one dependent?

Ternary Relationships

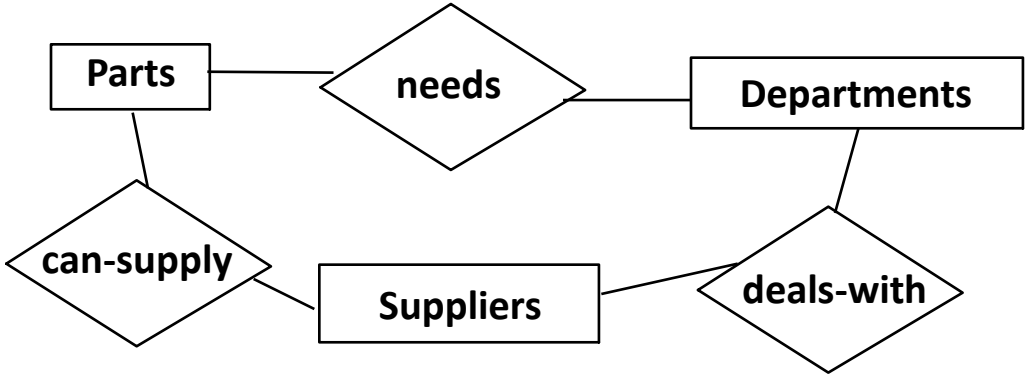


in general, **n**-ary relationships

Ternary vs. Binary Relationships



vs.



S "can-supply" P, D "needs" P, and D "deals-with" S

does it imply that D has agreed to buy P from S?

if so, how do we record *qty*?

Now you try

[You speak, I am drawing!]

University database schema

Entities: Courses, Students, Instructors

Each course has id, name, time, room #

Make up suitable attributes for students, instructor

Each course has <u>exactly one</u> instructor

Students have a grade for each course

Now … keep track of multiple semesters!

each instructor teaches exactly one course offering

track student transcripts across entire enrollment period

track history of courses taught by each instructor

The Entity-Relationship Model

Basic ER modeling concepts

Constraints

Complex relationships

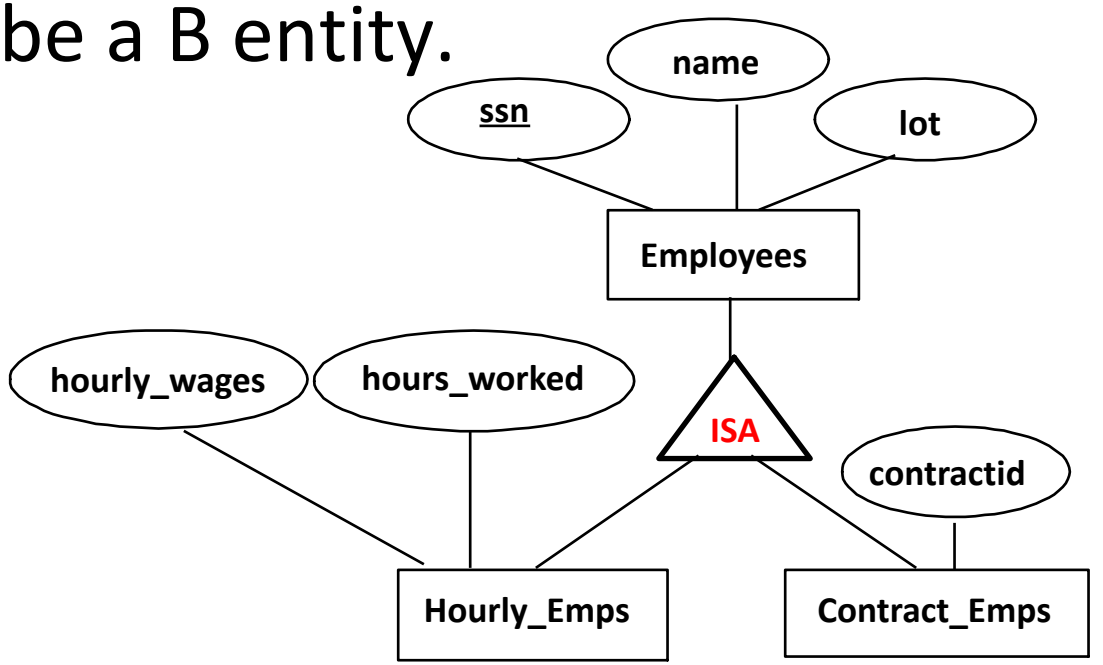
Readings: Chapters 2.4.4-2.4.5

Conceptual Design

ISA (‘is a’) Hierarchies

as in C++, or other PLs, attributes are inherited

if we declare A **ISA** B, every A entity is also considered to be a B entity.



ISA (‘is a’) Hierarchies

Overlap constraints: Can Joe be an Hourly_Emps as well as a Contract_Emps entity? (*Allowed/Disallowed*)

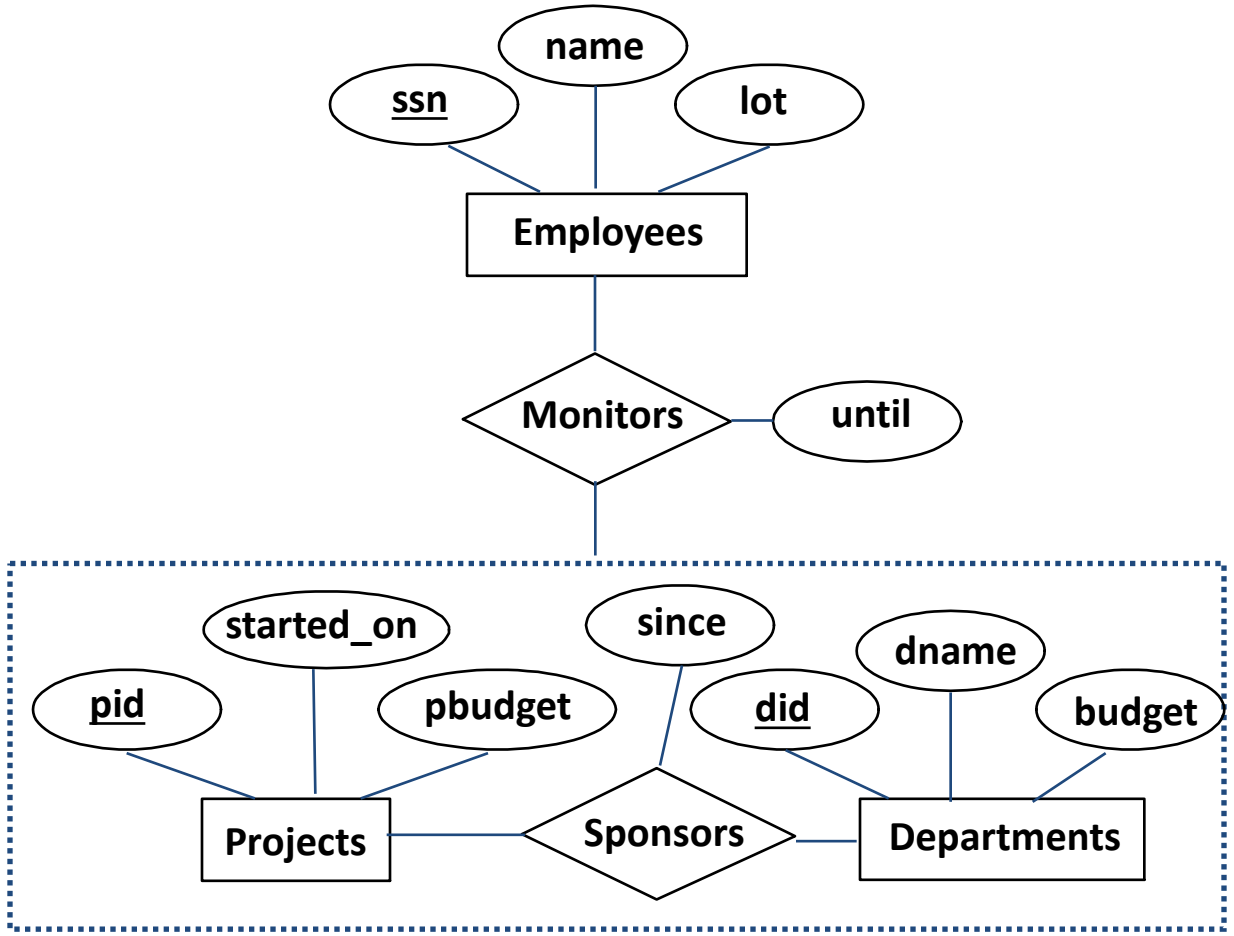
Covering constraints: Does every Employees entity also have to be an Hourly_Emps or a Contract_Emps entity? *(Yes/No)*

Reasons for using ISA:

- to add descriptive attributes specific to a subclass
  - → we do not keep “hours worked” for everybody
- to identify entities that participate in a particular relationship
  - → manager can be only a “contract employee”

Aggregation

used for a relationship involving another *relationship set*



treats a relationship set as an entity set

[for purposes of participation in (other) relationships]

Aggregation vs. ternary relationship?

- Monitors is a distinct relationship, with a descriptive attribute
- Also, can say that each sponsorship is monitored by at most one employee

The Entity-Relationship Model

Basic ER modeling concepts

Constraints

Complex relationships

Conceptual Design

Readings: Chapter 2.5

Conceptual Design Using the ER Model

Design choices:

Should a concept be modeled as an entity or an attribute?

Should a concept be modeled as an entity or a relationship?

Identifying relationships: binary or ternary? Aggregation?

Constraints in the ER Model:

A lot of data semantics can (and should) be captured

But some constraints cannot be captured in ER diagrams

Entity vs. Attribute

Should *address* be an attribute of Employees or an entity (related to Employees)?

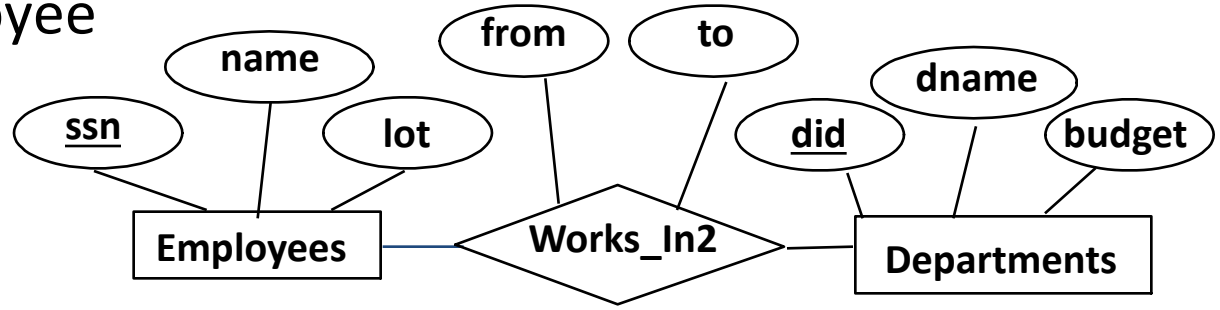
Depends upon how we want to use address information, and the semantics of the data:

If we have several addresses per employee,
address must be an entity (since attributes cannot be set-valued)

If the structure (city, street, etc.) is important, *address* must be modeled as an entity (since attribute values are atomic)

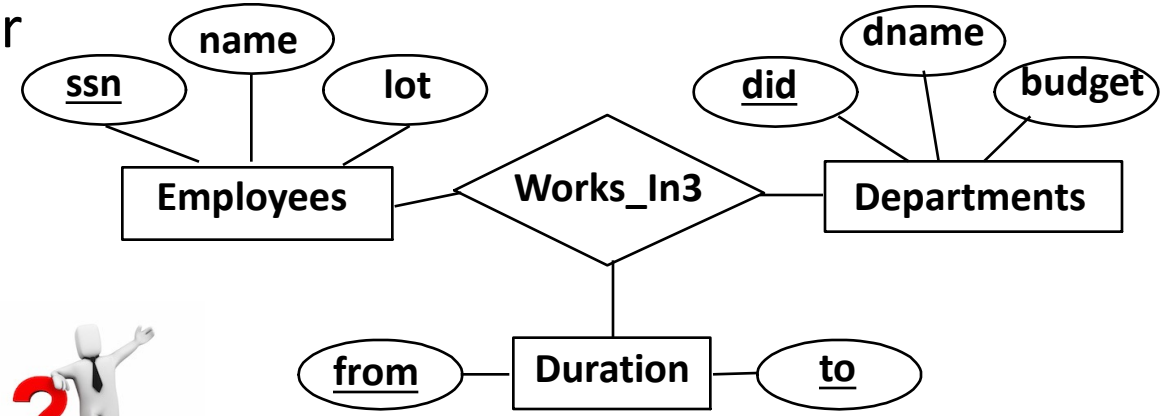
Entity vs. Attribute (Cont.)

Works_In2 does not allow an employee to work in a department for two or more periods



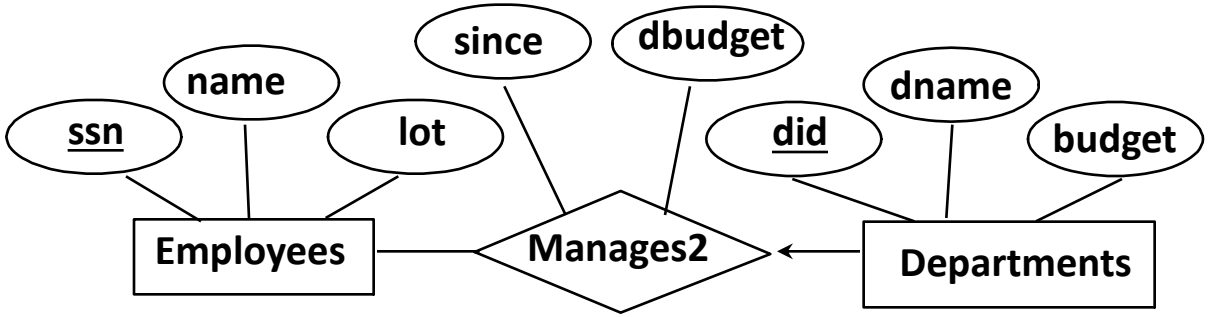
why?

Approach: Similar to the problem of wanting to record several addresses for an employee: we want to record *several values of the descriptive attributes for each instance of this relationship*

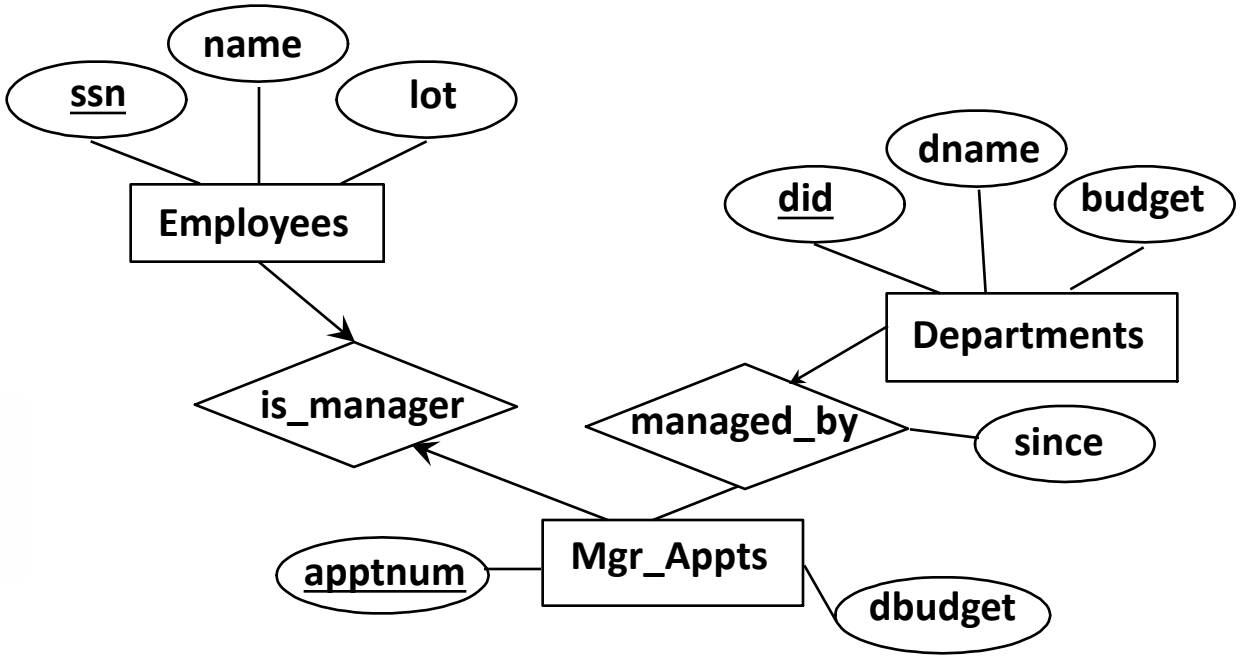


Entity vs. Relationship

OK as long as a manager gets a separate discretionary budget (*dbudget*) for each department



What if manage's *dbudget* covers *all* managed departments? (can repeat value, but such redundancy is problematic)



Notes on the ER design

ER design is *subjective*

many "correct" ways to model a given scenario!

analyzing alternatives can be tricky

common dilemmas: entity vs. attribute, entity vs. relationship, binary or n-ary relationship, whether to use ISA hierarchies, aggregation

many types of *constraints cannot be expressed* (notably, *functional dependencies*)

[although constraints play an important role in determining the best database design for an enterprise]

Context: Overall Database Design Process

Requirements Analysis

user needs; what must database do?

Today ➡ Conceptual Design

high level description (often done w/ER model)

Next time: Logical Design

translate ER into DBMS data model

Later: Schema Refinement

consistency, normalization

Physical Design

indexes, disk layout

Security Design

who accesses what

Written Assignments

Tentative Dates

#	Topic	Out on	Due on
WA1	ER-Diagram & Relational Model	9/10	9/19
WA2	Relational Algebra	9/20	9/26
WA3	Functional Dependencies & Normalization	9/27	10/3
WA4	SQL	10/4	10/10
WA5	Storage Layer & B+ Tree	10/11	10/17
WA6	Query Optimization & Processing	TBA	TBA
WA7	Locking & Logging	TBA	TBA

Useful for the midterm

Summary of Conceptual Design

Conceptual design follows *requirements analysis*

Yields a high-level description of data to be stored

ER model popular for conceptual design

Constructs are expressive, close to the way people think about their applications

Originally proposed by Peter Chen, 1976

Note: there are many variations on ER model

Basic constructs: ***entities***, ***relationships***, and ***attributes***
(of entities and relationships)

Some additional constructs: ***weak entities***, ***ISA hierarchies***, and ***aggregation***