CONSTRUCTOR UNIVERSITY

Database Design

Core Database Design Steps [ANSI 1975]

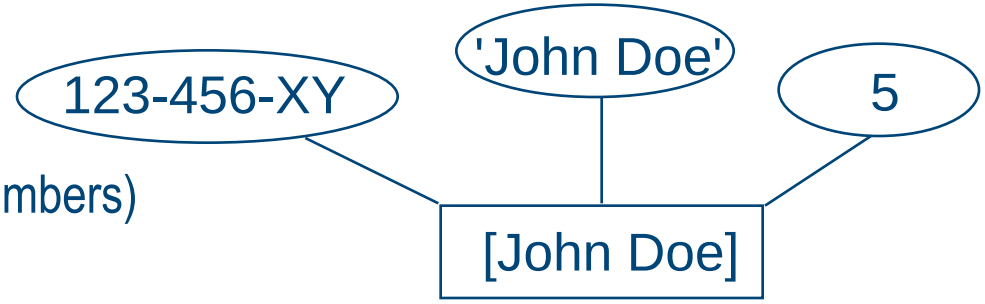
- Conceptual design ← Our focus in this section
  - Construct a description of the information used in an enterprise
  - *Focus on documenting customer intention, disregard technology*
- Logical design
  - Construct a description based on a specific data model (e.g., relational)
  - *Focus on abstract tech, disregard implementation*
- Physical design
  - Describe implementation using a particular DBMS, file structures, indexes, security, …

Issues in Conceptual Design

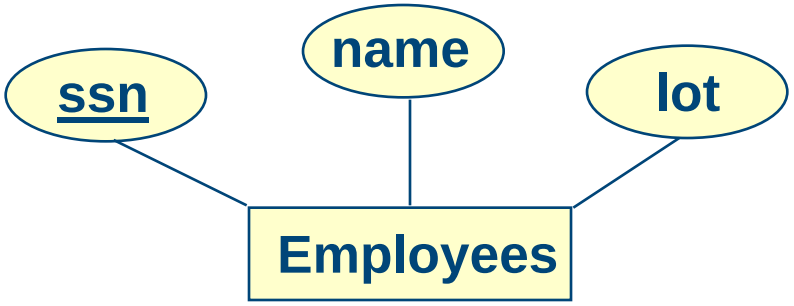
- Conceptual design: (we use ER Model at this stage)
  - What are the entities and relationships in the enterprise?
  - What information about these entities and relationships should we store in the database?
  - What are the integrity constraints or business rules that hold?
- database `schema' in the ER Model represented pictorially = ER diagrams
  - Can map an ER diagram into a relational schema
  - Actually lack of textual equivalent is shortcoming
  - … also: no formal semantics (originally)

Entity-Relationship Model: Basics

- Entity: Real-world object distinguishable from other objects
  - entity described (in DB) using a set of attributes
  - Simple attribute values (strings, numbers)

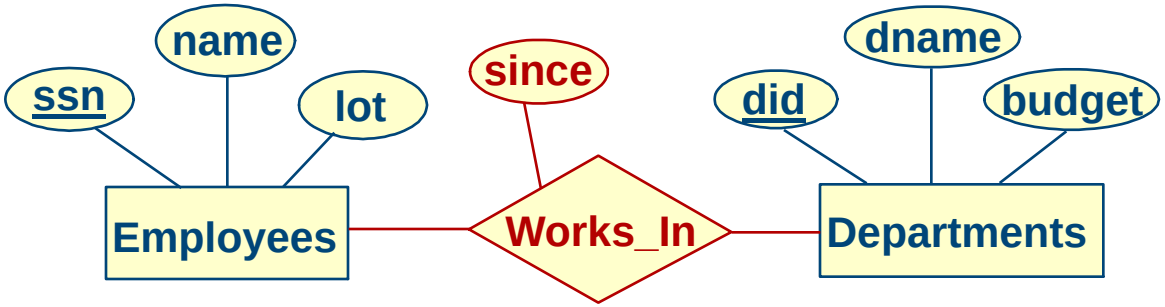


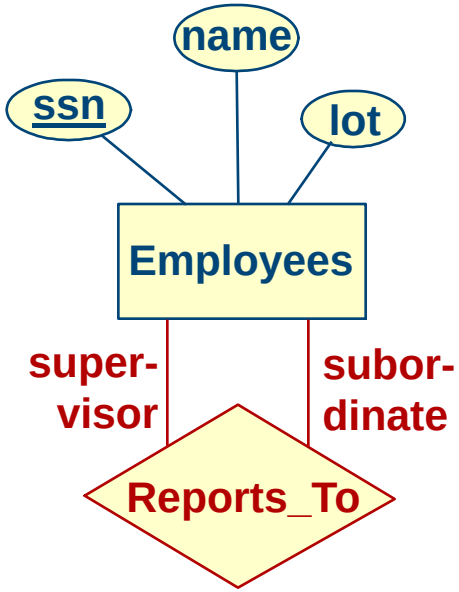
- Entity set: collection of similar entities
  - E.g., all employees
  - All entities in an entity set have the same set of attributes
    - *Until we consider ISA hierarchies, anyway!*
  - Each entity set has a key
  - Each attribute has a domain = data type



ER Model Basics (Contd.)

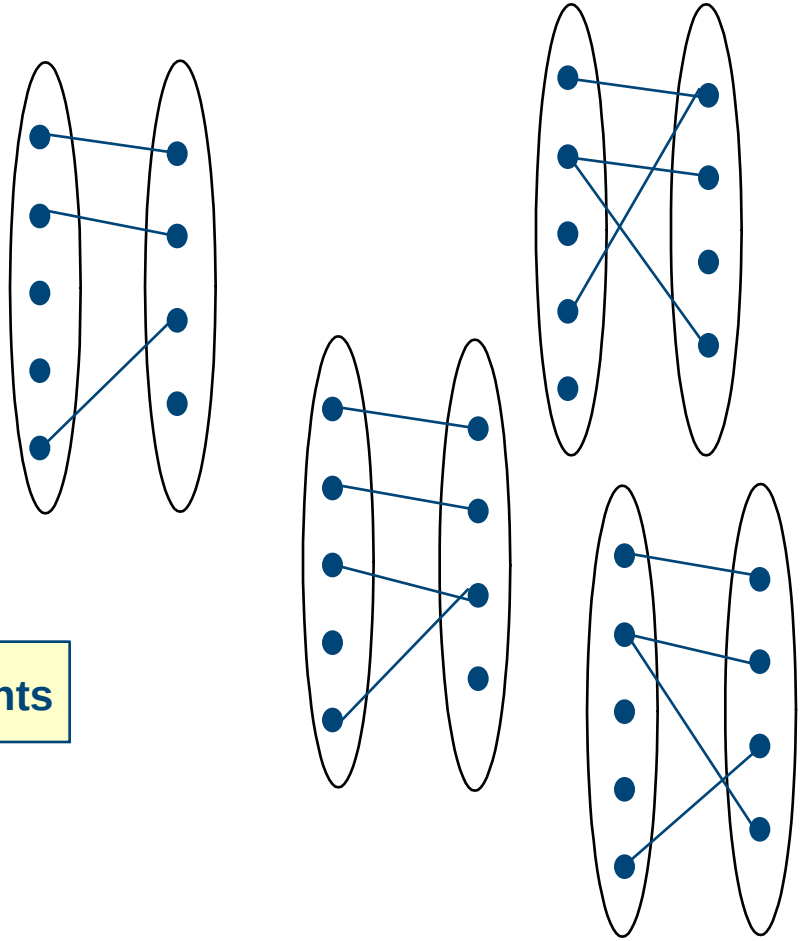
- Relationship: (unique!) association among two or more entities
  - E.g., Attishoo works_in Pharmacy department
- Relationship Set: Collection of similar relationships
  - An n-ary (binary, ternary, …) relationship set R relates n entity sets E1 ... En
  - each relationship in R involves entities e1∈E1, ..., en∈En
  - Same entity set can participate in different relationship sets, or even in the same set (but then in different roles)





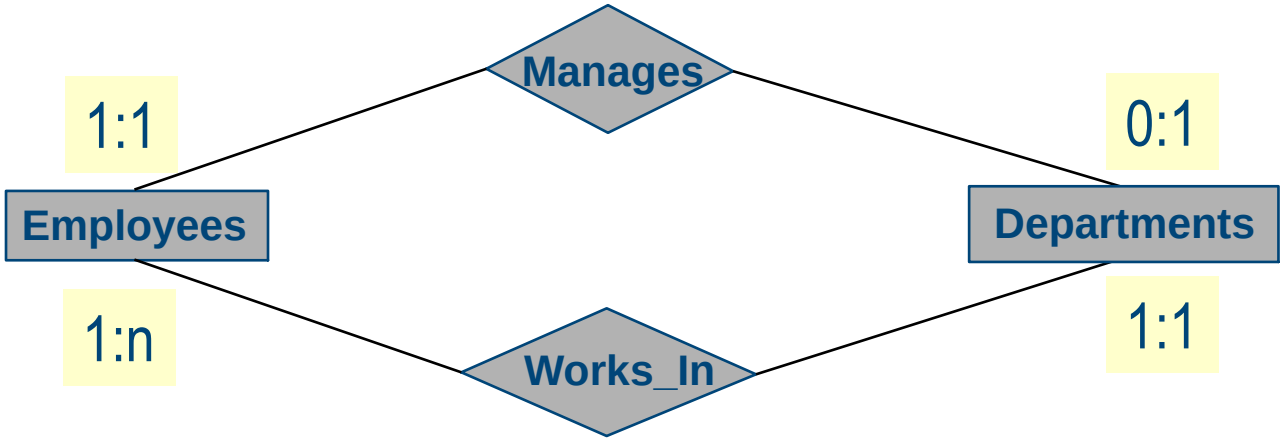
Key Constraints

- Multiplicity indicators:
  - One-to-one "1:1"
  - One-to-many "1:n"
  - Many-to-many "m:n"



More Detail Wanted!

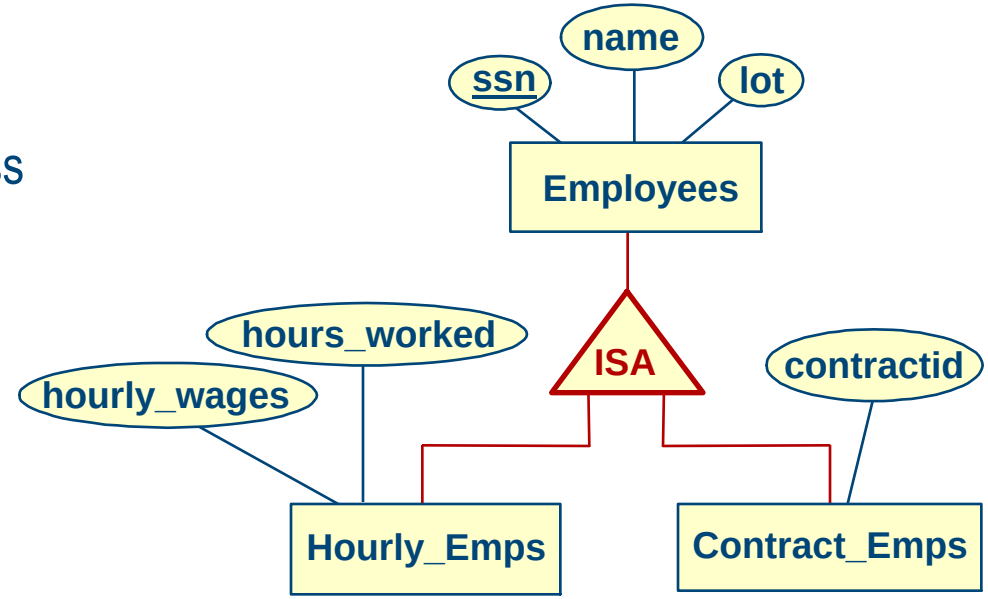
- Want to refine further: *how many connections on each leg of relship?*
- Attach intervals to leg:



- Read as:
  - *„an Employee sees, through its Manages tunnel, none or one Department“*
  - *„a Department sees, through its Works_In tunnel, at least one Employee“*

ISA (`is a') Hierarchies

- A ISA B: every A entity is also a B entity ("A inherits from B")
  - A is called subclass, B superclass
- Purpose:
  - add attributes specific to a subclass
  - identify specific entitities that participate in a relationship
- Constraints:
  - Overlap constraints
  - Covering constraints



Conceptual Design Using the ER Model

- Design choices:
  - concept modeled as entity or attribute?
  - concept modeled as entity or 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 – *comment your design!*
- *Let's see…*

UML™

- UML = Unified Modeling Language [www.uml.org]
  - "a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system."
  - UML class diagram ~ ER diagram