Entity-Relationship Model

Linda Wu

(CMPT 354 • 2004-2)

Topics

- Overview of database design
- ER model basics
- Key constraints
- Participation constraints
- Weak entities
- Class hierarchies
- Aggregation
- Conceptual design with ER model

Overview of Database Design

- Database design process can be divided into six steps
  - Requirement analysis
  - Conceptual database design (ER model is used in this step)
  - Logical database design
  - Schema refinement
  - Physical database design
  - Application and security design

Overview of Database Design (Cont.)

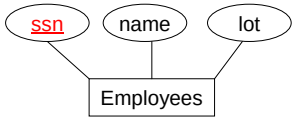
- Conceptual design
  - What are the entities and relationships in the enterprise?
  - What information about these entities and relationships should be stored in the database?
  - What are the integrity constraints or business rules that hold?
  - A database schema in the ER Model can be represented pictorially (ER diagrams)
  - An ER diagram can be mapped into a relational schema

ER Model Basics

- Entity
  - Real-world object distinguishable from other objects
  - Described in DB by 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 (if not consider ISA hierarchies)
  - Each entity set may have a key
  - Each attribute has a domain

ER Model Basics (Cont.)

- Entity set representation
  - Rectangle: entity set
  - Oval: attribute
  - Underline: attributes in the primary key

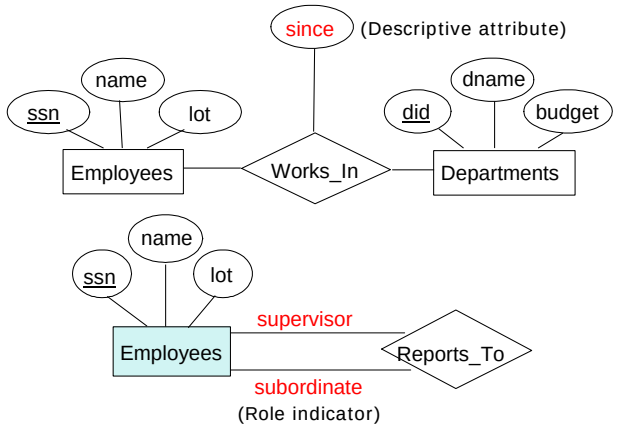


ER Model Basics (Cont.)

- Relationship
  - Association among two or more entities
  - e.g., Attishoo works in Pharmacy department
- Relationship Set
  - Collection of similar relationships
  - An n-ary relationship set R relates n entity sets E_1, ..., E_n; each relationship in R involves entities e_1, ..., e_n

    $R = \{(e_1, ..., e_n) | e_1 \in E_1, ..., e_n \in E_n\}$
  - Same entity set could participate in different relationship sets, or in different "roles" in same set

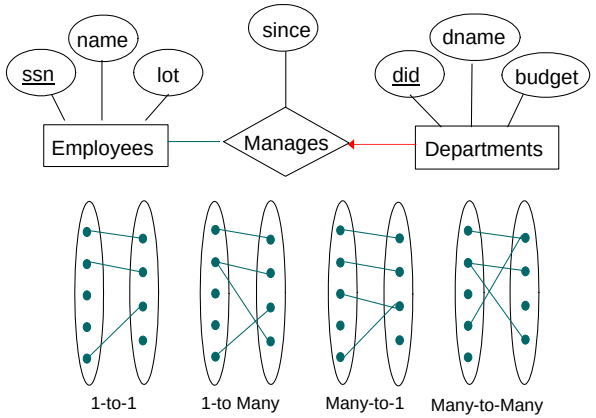
ER Model Basics (Cont.)



Key Constraints

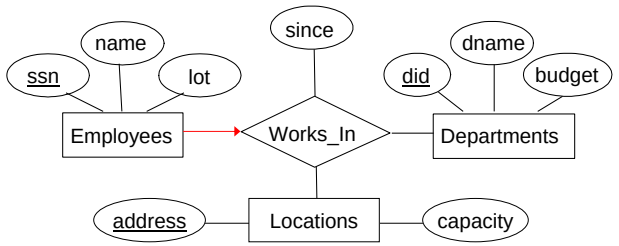
- Works_In relationship
  - An employee can work in several departments
  - A department can have several employees
- Manages relationship
  - An employee may manage more than one department
  - Each department has at most one manager (key constraint)
- Key constraint representation
  - Arrow: from Departments to Manages

Key Constraints (Cont.)



Key Constraints (Cont.)

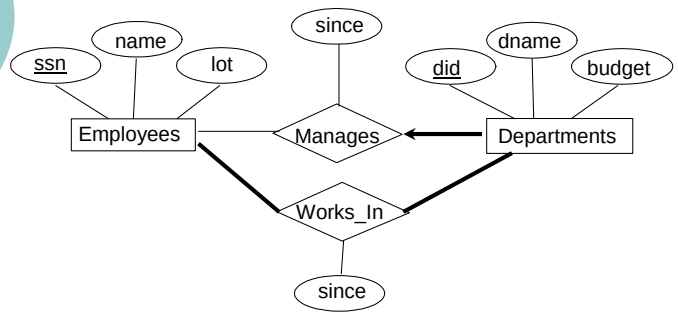
- Key constraints in ternary relationship
  - Relationship set R involves three or more entity sets
  - An entity set E has a key constraint in R
  - Representation: an arrow from E to R



Participation Constraints

- Does every department have a manager?
  - If so, the participation of Departments in Manages is said to be total
    - Every "*did*" value in Departments must appear in a tuple of the Manages relation
    - Representation: a thick line between the entity set (Departments) and the relationship set (Manages)
  - If no, the participation is partial

Participation Constraints (Cont.)

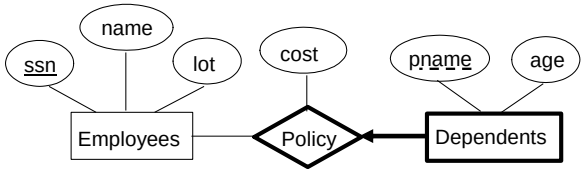


Weak Entities

- A weak entity can be identified uniquely only by considering the primary key of another (owner) entity
  - Owner entity set and weak entity set must participate in a one-to-many relationship set: identifying relationship set
  - Weak entity set must have total participation in this identifying relationship set

Weak Entities (Cont.)

- Representation
  - Dark lines: to draw weak entity set & its identifying relationship set
  - Broken line: to underline a partial key

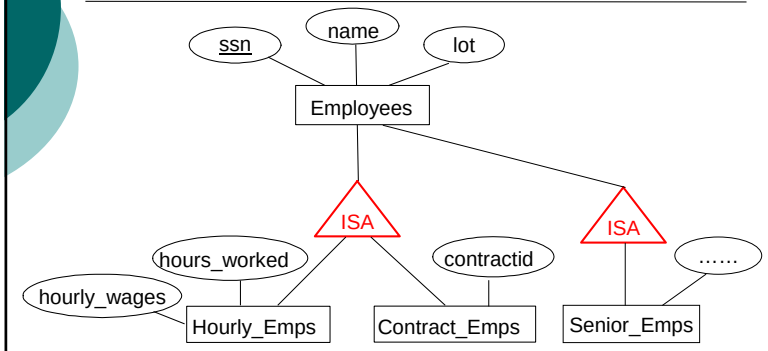


- A Dependents entity is uniquely identified by (ssn, pname)
- pname: partial key of Dependents
- policy: identifying relationship set

Class Hierarchies

- ISA (“is a”)
  - If we declare A ISA B, every A entity is also considered to be a B entity
  - As in C++, or other languages, attributes are inherited
- 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
  - To identify entities that participate in a relationship

Class Hierarchies (Cont.)



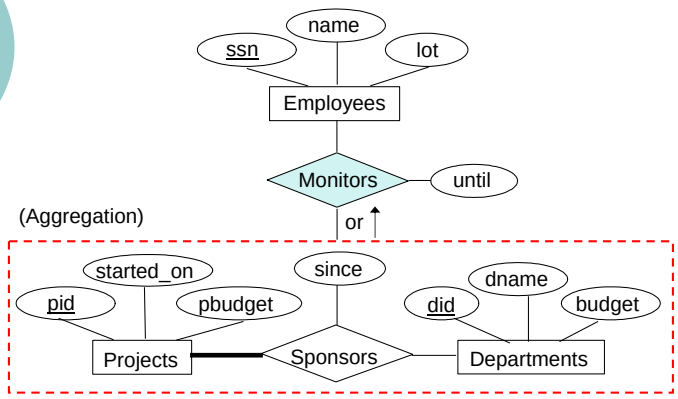
Hourly_Emps OVERLAPS Senior_Emps

Hourly_Emps AND Contract_Emps COVER Employees (?)

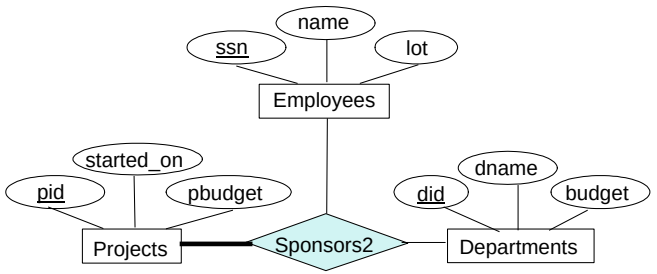
Aggregation

- Aggregation
  - To treat a relationship set as an entity set for purposes of participation in (other) relationships
  - To model a relationship involving (entity sets and) a relationship set
  - Representation: dashed box
- Aggregation vs. ternary relationship
  - Sponsors and Monitors are two distinct relationships, each with descriptive attributes of its own (*until* vs. *since*)
  - Consider the constraint that each sponsorship is monitored by at most one employee

Aggregation (Cont.)



Aggregation (Cont.)



(Ternary relationship)

Conceptual Design with ER Model

- Design choices
  - Should a concept be modeled as an entity or an attribute? (e.g., address)
  - Should a concept be modeled as an entity or a relationship? (e.g., address)
  - Binary or ternary relationships? Aggregation?
- Constraints in ER model
  - A lot of data semantics can (and should) be captured
  - But some constraints cannot be captured in ER diagrams

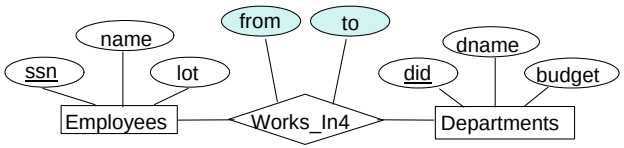
Entity vs. Attribute

- Example: address
  - An attribute of Employees?
  - Or, an entity that is connected to Employees by a relationship?
- Depends upon the use we want to make of address information, and the semantics of the data
  - If several addresses per employee: address must be an entity (attributes cannot be set-valued)
  - If the structure (city, street, etc.) is important, e.g., we want to retrieve employees in a given city: address must be modeled as an entity (attribute values are atomic)

Entity vs. Attribute (Cont.)

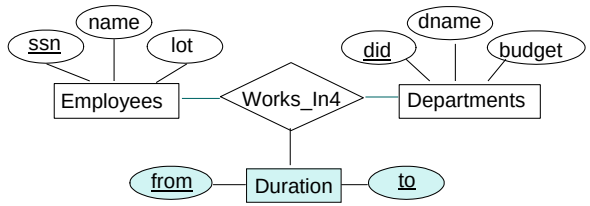
- Works_In4 does not allow an employee to work in a department two or more periods
  - A relationship is uniquely identified by the participating entities:

$$R = \{(e_1, \ldots, e_n) \mid e_1 \in E_1, \ldots, e_n \in E_n\}$$



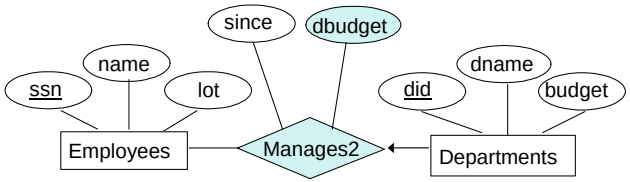
Entity vs. Attribute (Cont.)

- If wanting to record several working periods for an employee in Work_In4
  - Introduce new entity set, Duration



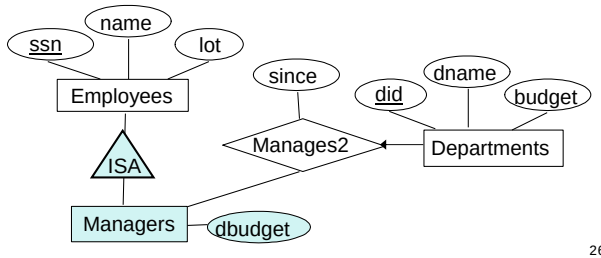
Entity vs. Relationship

- Example: A manager gets a separate discretionary budget for each department he or she manages



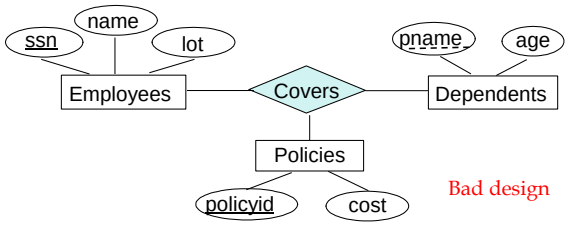
Entity vs. Relationship (Cont.)

- What if a manager gets a discretionary budget that covers *all* managed departments?
  - Redundancy: *dbudget* stored for each department managed by manager
  - Misleading: Suggests *dbudget* is associated with department-manager combination



Binary vs. Ternary Relationships

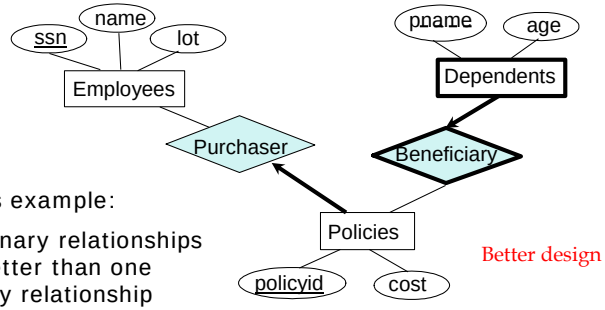
- A bad design below if:
  - Each policy is owned by just 1 employee, and,
  - Dependents is a weak entity set, and each dependent is tied to the covering policy



Bad design

Binary vs. Ternary Relationships (Cont.)

- Better design
  - Policies involves in two relationships



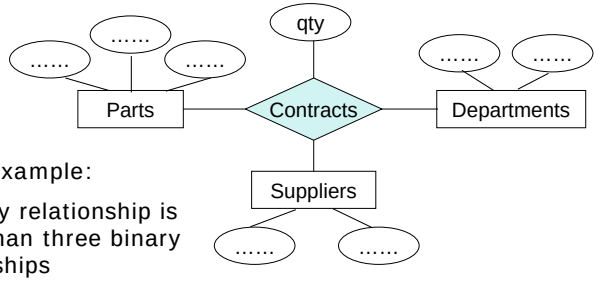
In this example:

two binary relationships are better than one ternary relationship

Better design

Binary vs. Ternary Relationships (Cont.)

- Another example
  - The contract specifies that a supplier will supply some quantity of a part to a department



In this example:

a ternary relationship is better than three binary relationships

Summary of ER Model

- Conceptual design follows requirements analysis yields a high-level description of data to be stored
- ER model is popular for conceptual design
  - Constructs are expressive
  - Close to the way people think about their applications
- Basic constructs
  - Entities, relationships, and attributes
- Some additional constructs
  - Weak entities, ISA hierarchies, aggregation
- There are many variations on ER model

Summary of ER Model (Cont.)

- Several kinds of integrity constraints can be expressed in the ER model
  - Key constraints, participation constraints, and overlap/covering constraints for ISA hierarchies
  - Some constraints (notably, *functional dependencies*) cannot be expressed in the ER model
  - Constraints play an important role in determining the best database design for an enterprise
- ER design is subjective. There are often many ways to model a given scenario!

Summary of ER Model (Cont.)

- Analyzing alternatives can be tricky, especially for a large enterprise. Common choices include:
  - Entity vs. attribute
  - Entity vs. relationship
  - Binary or n-ary relationship
  - Whether or not to use ISA hierarchies
  - Whether or not to use aggregation
- Ensuring good database design
  - Analyze and refine the resulting relational schema
  - Use FD information and normalization techniques