CPSC 304
Introduction to Database Systems

Conceptual Database Design:
The Entity-Relationship Model, Part 2

Fall 2017

References:

Database Management Systems, by Ramakrishnan and Gehrke (Chapter 2 of the Textbook)

Data Modeling Essentials, by Simsion and Witt, Third Edition, Morgan Kaufman Publishers, 2005

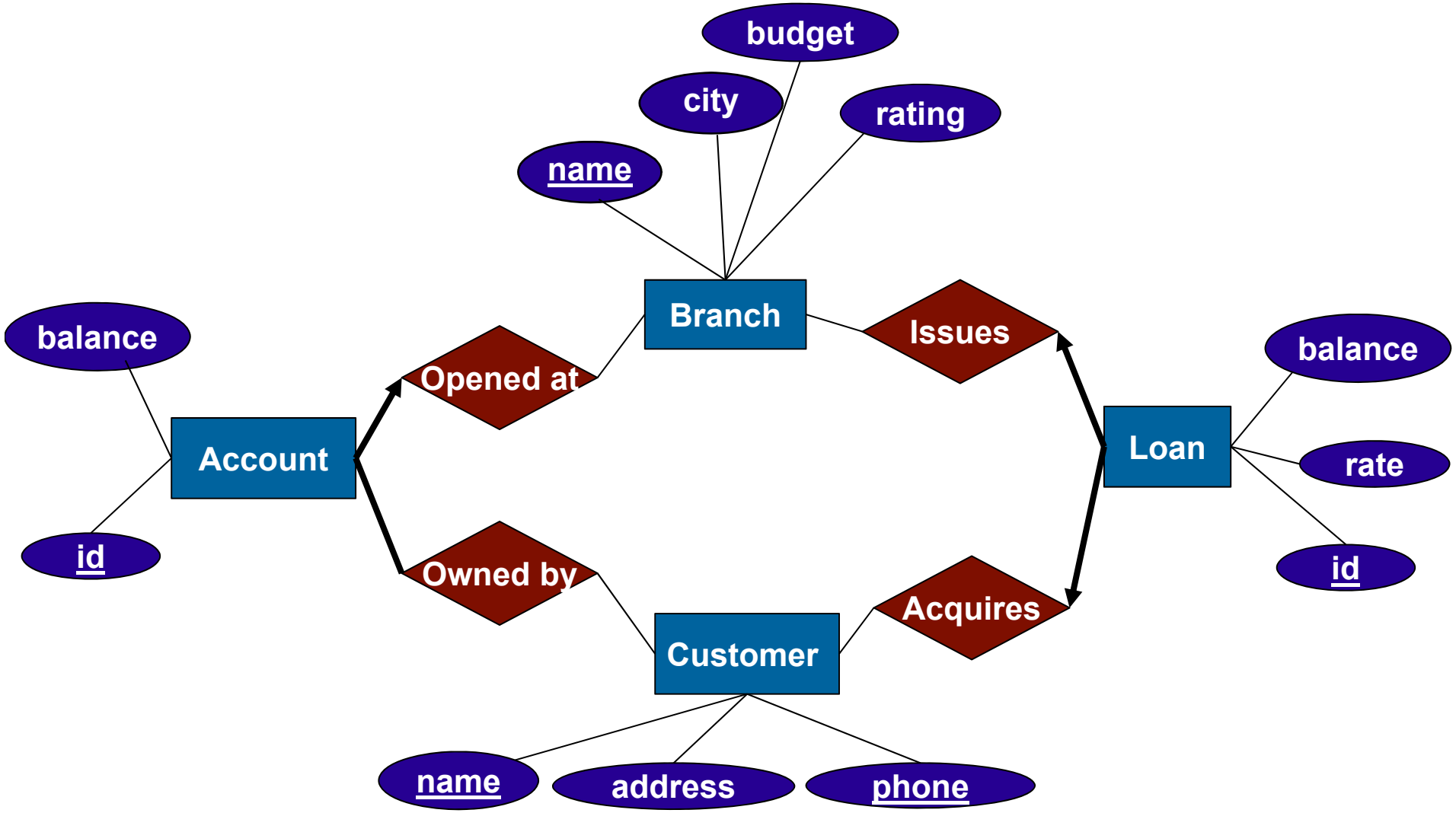
Learning Goals

- Explain the purpose of an ER diagram, and list its major components.
- Read and interpret an ER diagram.
- Given a problem description and specification, create an ER diagram. Justify the decisions you make for entities, relationships, keys, key constraints, participation constraints, weak entities, is-a relationships, aggregations, etc.
- Given a problem description, identify alternative representations of the problem concepts and evaluate the choices.
- Compare alternative ER models for the same domain and identify their strengths and weaknesses.

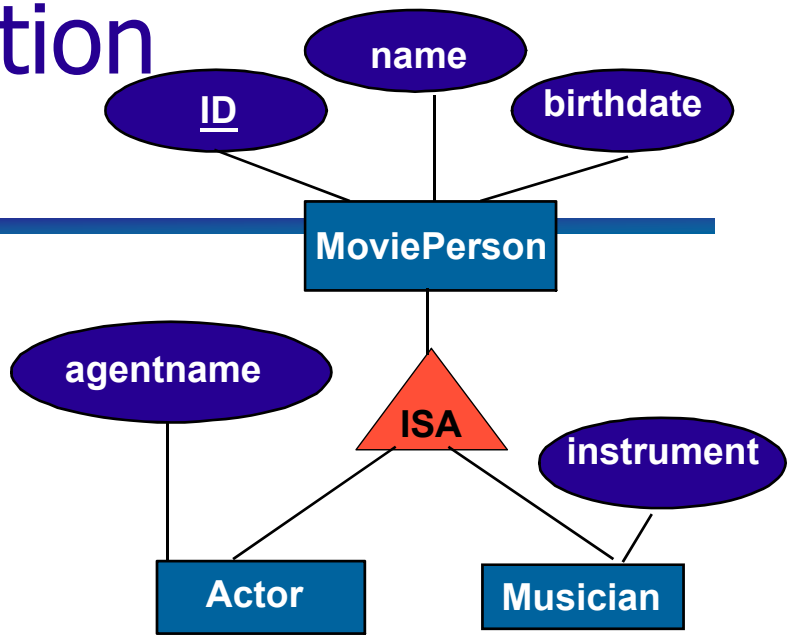
Exercise: ABC Banks

- The bank is organized into branches. Each branch is located in a particular city and is identified by a unique name. Each year the bank's board defines the yearly budget and the rating (which is a number from 1 to 10) for each branch.
- A bank customer is identified by their customer name and phone number. The bank also keeps track of each customer's current address.
- The bank offers accounts and loans to its customers. Each account and loan has a unique number and is created and maintained by a single branch.
- Each account is assigned to one or more customers and its balance can never be negative.
- A loan is always assigned to a single customer, has a fixed interest rate, and its balance cannot be negative either.

Sample Solution



Generalization/Specialization (ISA relationships)

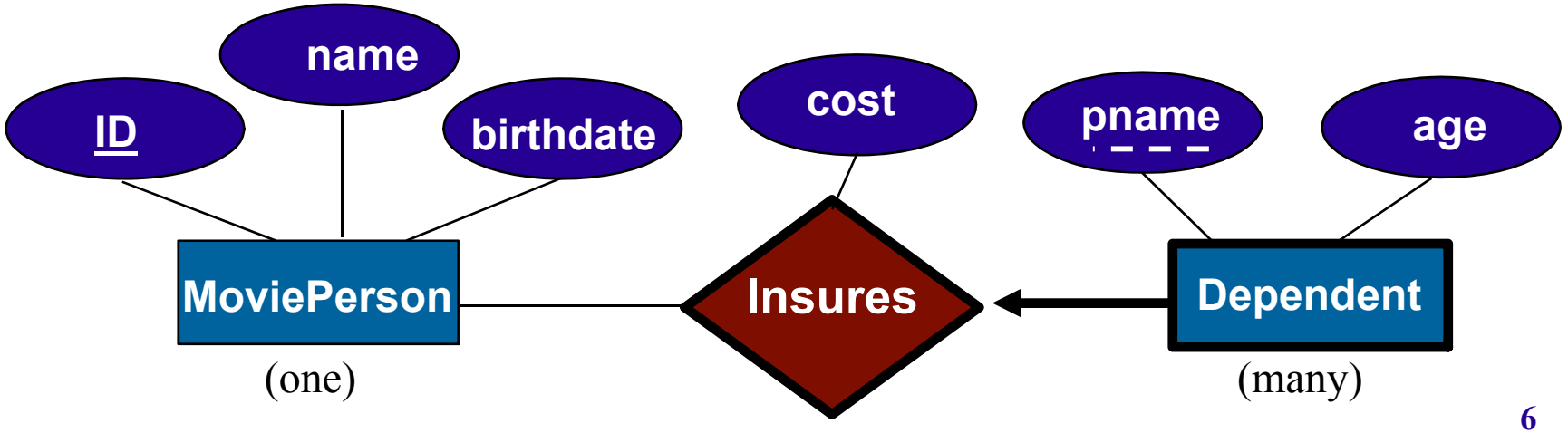


- In Java and some other languages, attributes can be inherited.
- If we declare A ***ISA*** B, every A entity is a B entity; but not vice-versa. eg. every Actor is a MoviePerson, but not every MoviePerson is an actor
- Reasons for using ISA:
  - To add descriptive attributes specific to a subclass
  - To restrict entities that participate in a relationship
- How is "ISA" different from a relationship?
  - It's hard to define a key for Actors, in our example.

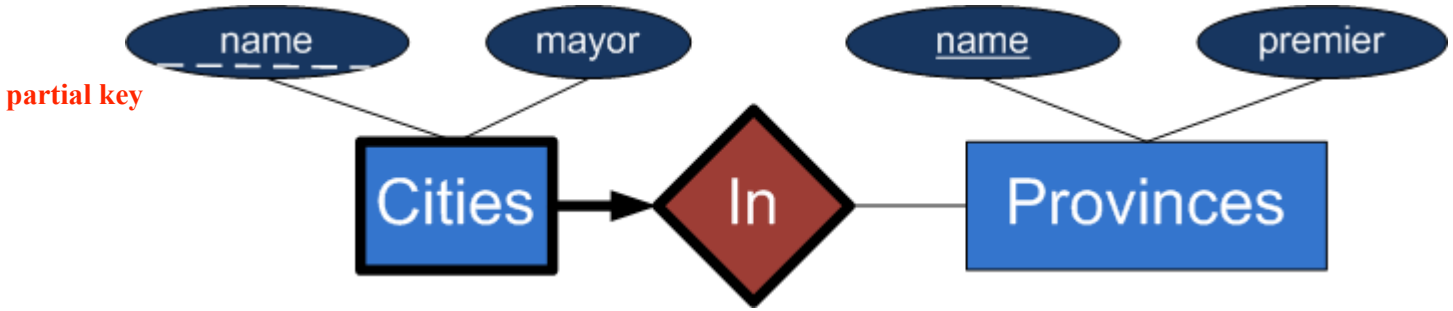
Weak Entities

dependens on another entity fo r this existence

- A ***weak entity*** can be identified uniquely only by considering the primary key of another (*owner*) entity.
  - The owner entity set and the weak entity set must participate in a one-to-many relationship set (one owner, many weak entities).
  - The weak entity set must have total participation in this ***identifying*** relationship set.
  - Think of this as a "belongs to" relationship.
- Weak entity sets and their identifying relationship sets are shown with thick lines, as in this insurance example:



Clicker Exercise

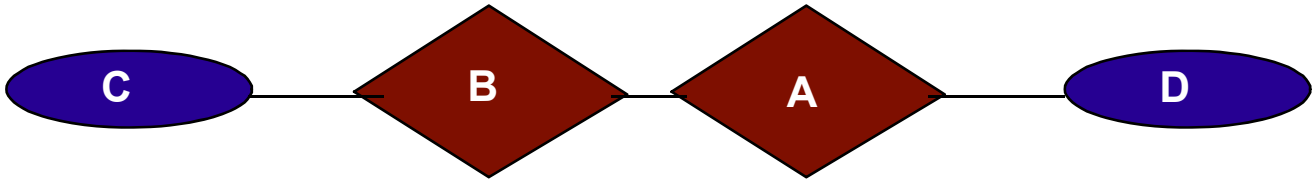


Which of the following is necessarily true?

A. No two provinces can have premiers with the same name.
B. No two cities can have mayors with the same name.
C. No two cities can have the same name.
D. No person can be the mayor of Cities in two different provinces.
E. None of the above

Aggregation

- Having a relationship between relationships is forbidden.

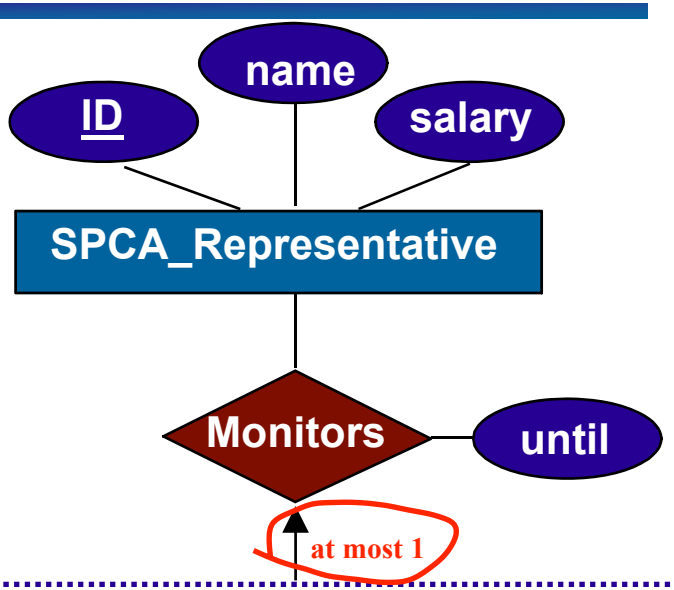


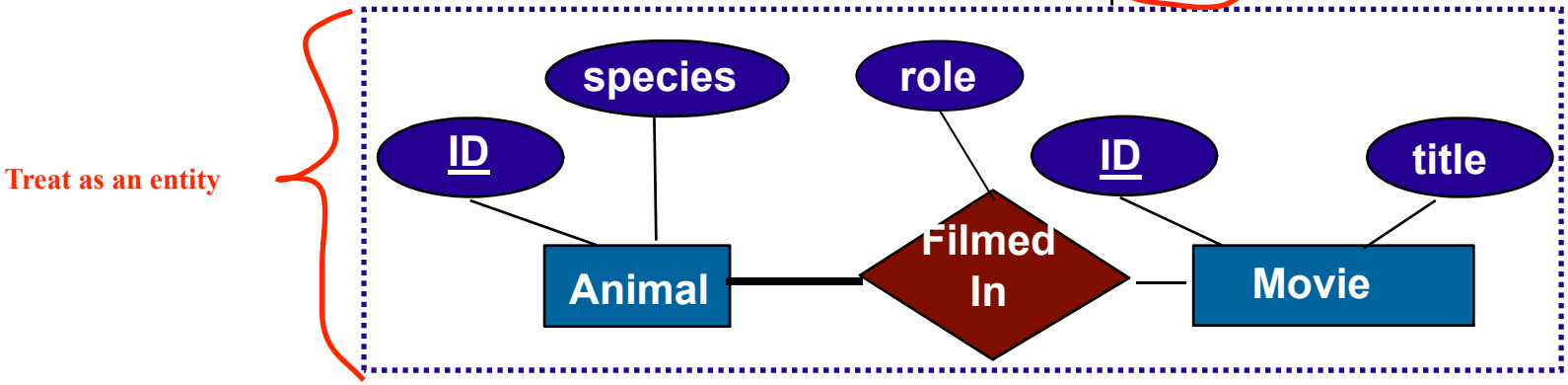
- However, **aggregation** allows us to treat a relationship set as an entity set for purposes of participation in (other) relationships.

Aggregation

SPCA = Society for the Prevention of Cruelty to Animals

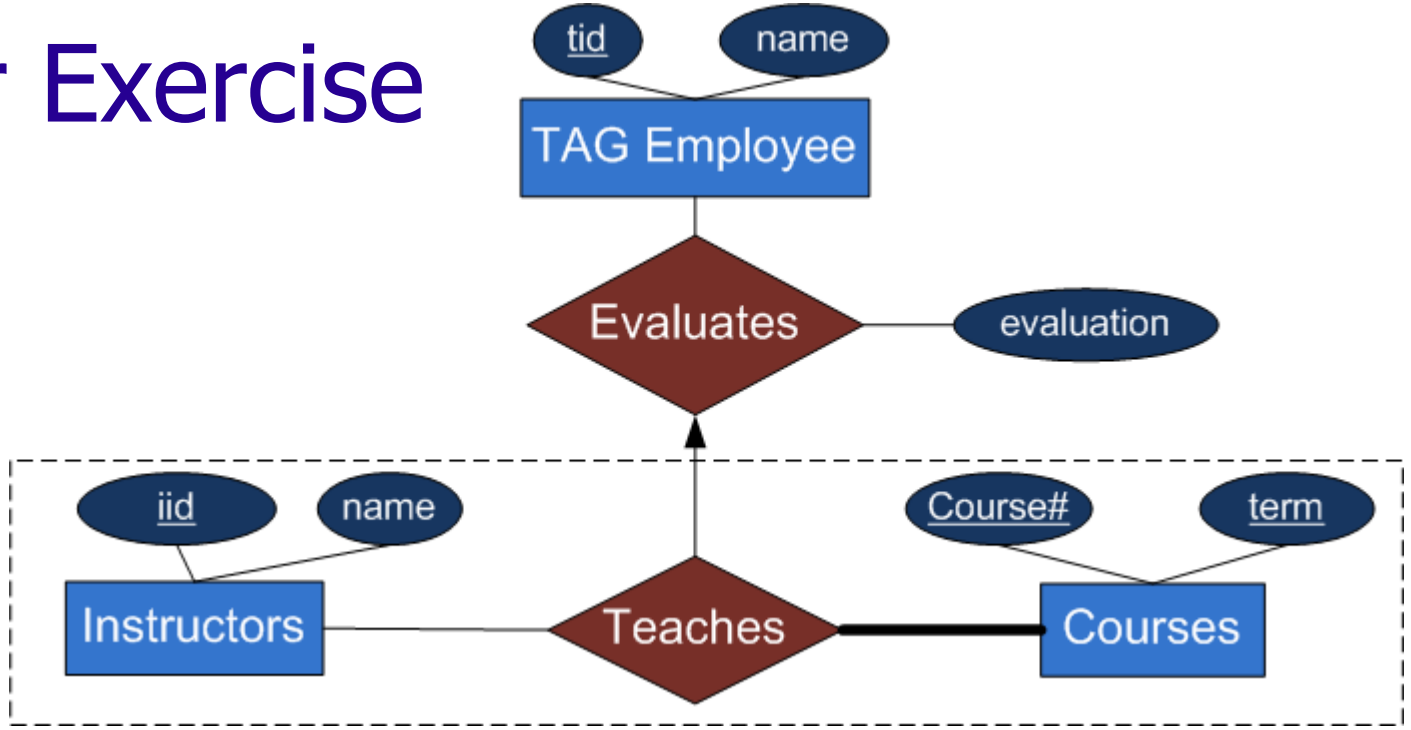
- Key for FilmedIn? Animal ID, Movie ID
- Key for Monitors? Animal ID, Movie ID





- Each sponsorship is monitored by at most one SPCA Representative.

Clicker Exercise



Choose the correct choice of (minimal) key from the options below:

A. The key of Evaluates is tid
B. The key of Evaluates is iid + Course# + term
C. The key of Evaluates is iid + Course# + term + tid
D. The key of Evaluates is iid + Course# + term + tid + evaluation
E. None of the above

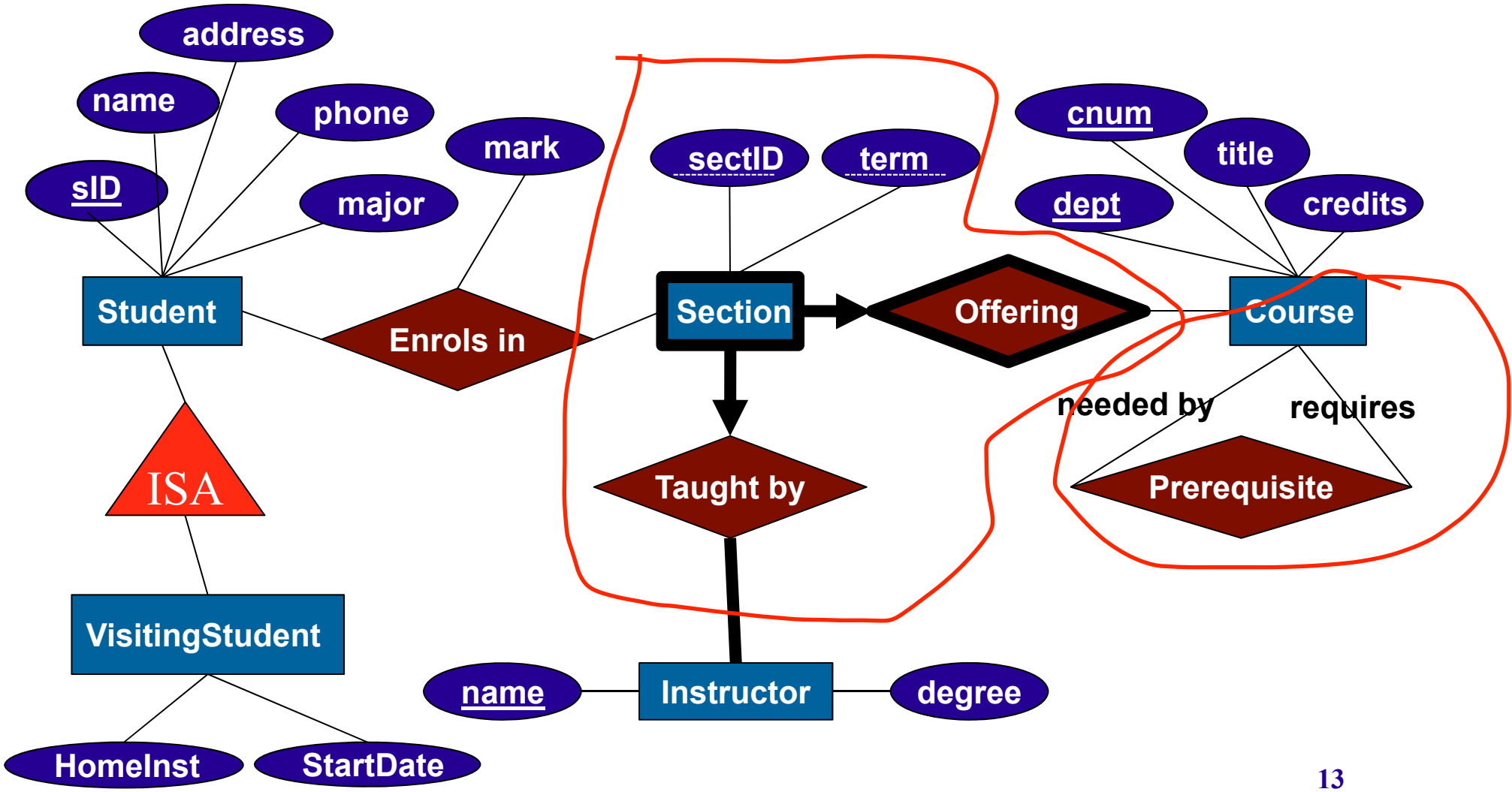
Summary of Symbols

Name	Symbol
Entity	
Attribute	
Relationship	
Generalization/Specialization	
Weak Entity	
Participation Constraint	
Key Constraint	

Exercise: UofU University. Draw an ER diagram for the following:

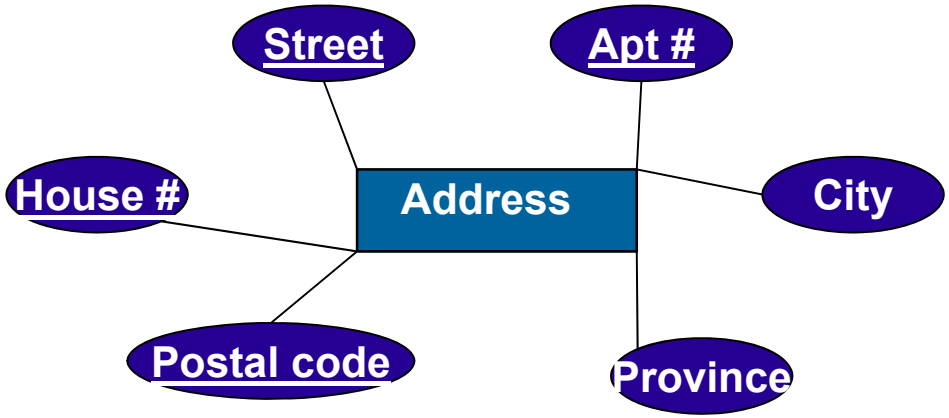
- A student is identified by a unique student ID number. A student has a name, address, and phone number. Each student is registered in a major (i.e., specialization) at UofU.
- Visiting students from another school can stay at UofU for up to one year.
- A course offered by UofU is identified by the department that offers the course and a course number which is unique within that department. Each course is listed in the calendar, with its course title and number of credits.
- A course may be offered many times, even within the same term. Each offering is assigned a section number which is unique for a given course and year, and is taught by a single instructor.
- Each instructor is responsible for some section; there are no idle instructors. Instructors have unique names, and may teach several sections of different courses. For each instructor, we like to record their highest degree.
- A student enrols in a course section and gets a mark for the course.
- A course may have any number of other courses as prerequisites.

Sample Solution



Comments about ER Modeling Techniques

- Some ER models differ in expressiveness.
- They model most concepts that designers want; but, they don't model all of them.
  - Later, we will deal with functional dependencies during logical modeling. This is when some attributes determine other attributes. For example:



Postal code determines city and province

Design Choices and Constraints

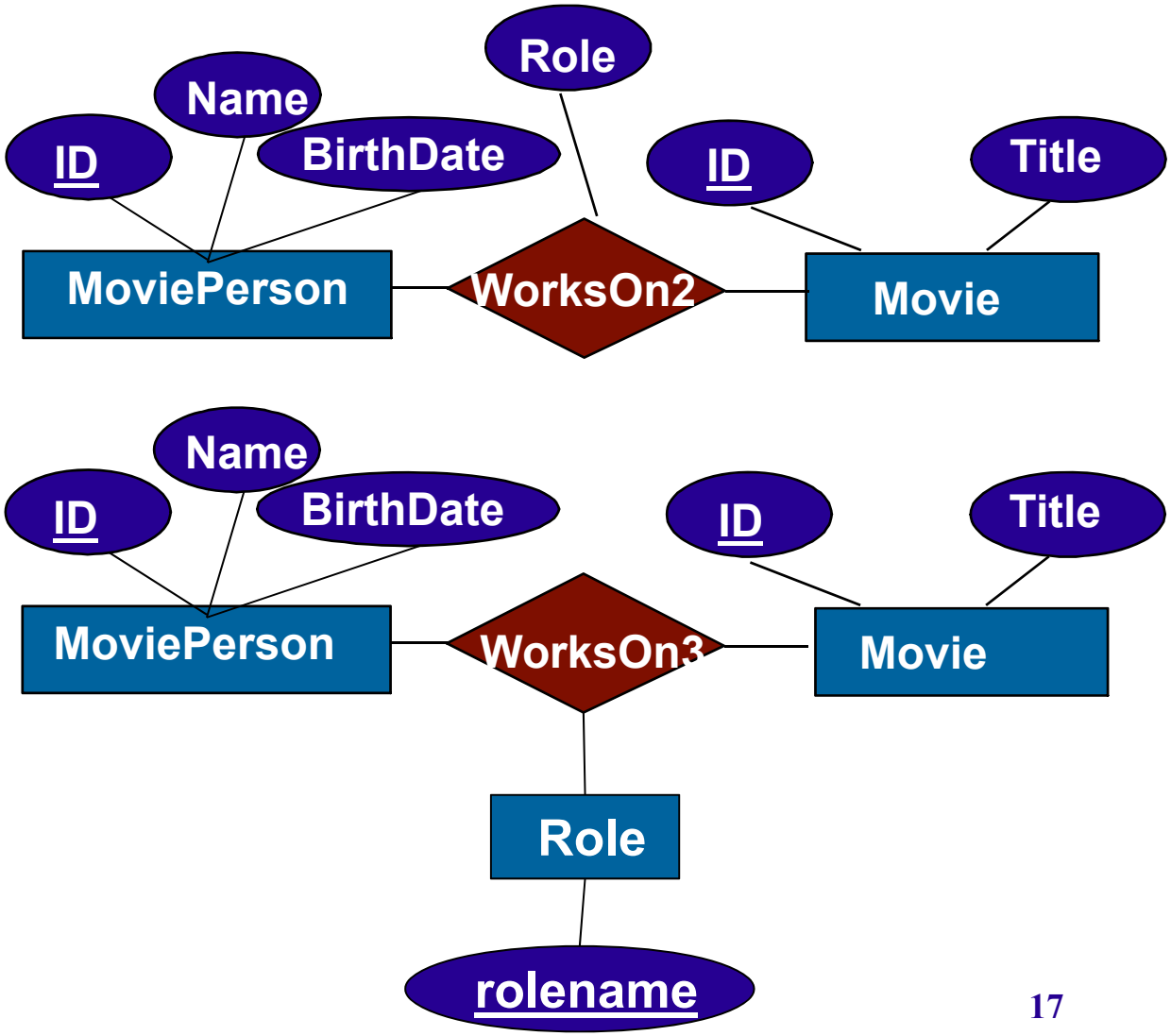
- Design Choices in the ER Model
  - Should a concept be modeled as an entity or an attribute?
    - e.g., address, phone number
  - Should a concept be modeled as an entity or a relationship?
  - Identifying relationships: binary or ternary? Aggregation?
- Constraints in the ER Model
  - A lot of data semantics can (and should) be captured.
  - But some constraints cannot be captured in ER diagrams:
    - Domain constraints
    - Dependencies

Thoughts on: Entity vs. Attribute

- Should an address be an *attribute* of MoviePerson or an *entity* (connected to MoviePerson by a relationship)?
- It depends on:
  - How we want to make use of the address information
  - The semantics of the data:
    - If we have several addresses per person, *address* must be an entity (since attributes cannot be set-valued—no "repeating groups")
    - If a person has only one street address, one city, one province, one postal code, etc., then these should simply be attributes.

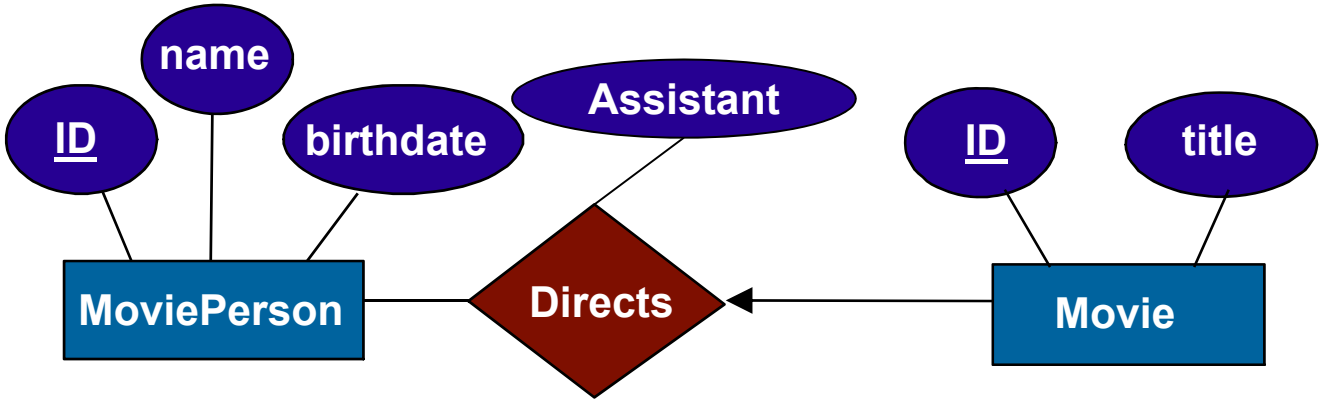
Entity vs. Attribute (cont.)

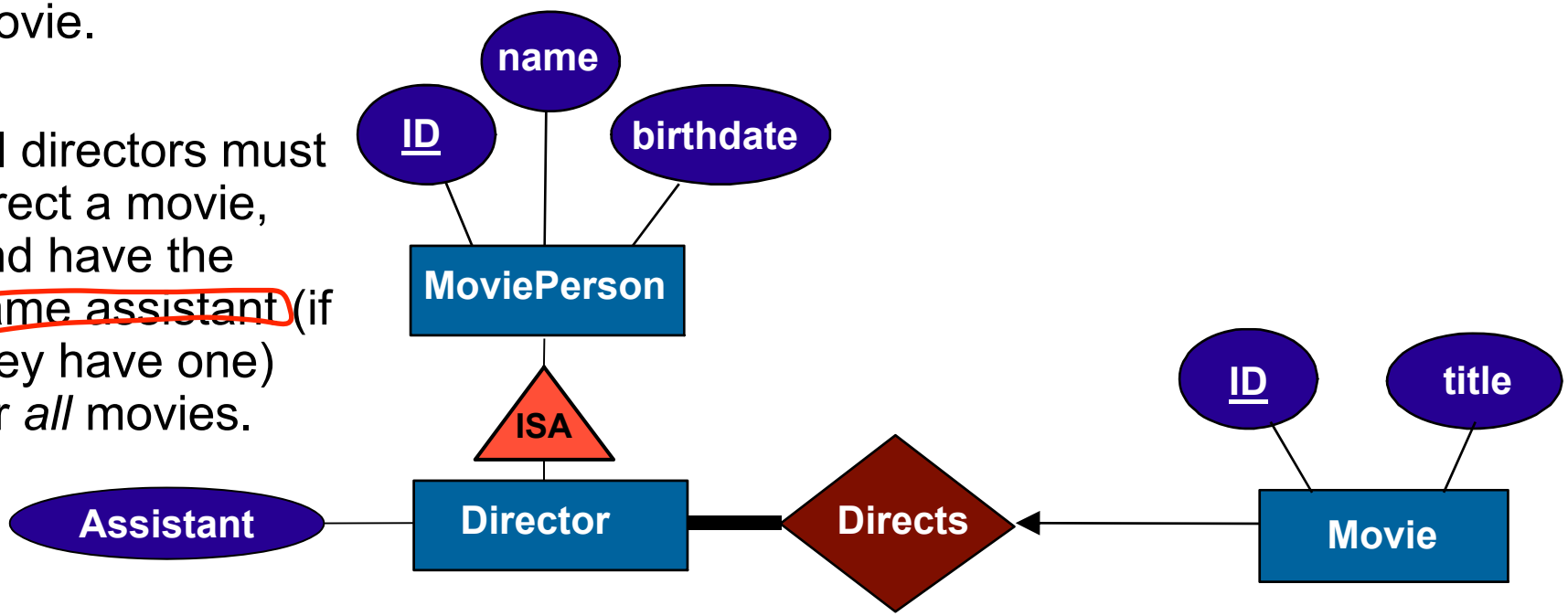
- WorksOn2 does not allow a person to have more than one role in the same movie.
- Suppose we want to associate the same (MoviePerson, Movie) pair with more than one role. How do we handle a set of values for the descriptive attributes?
- Solution: Change the descriptive attributes into entities.



Placement of Attributes

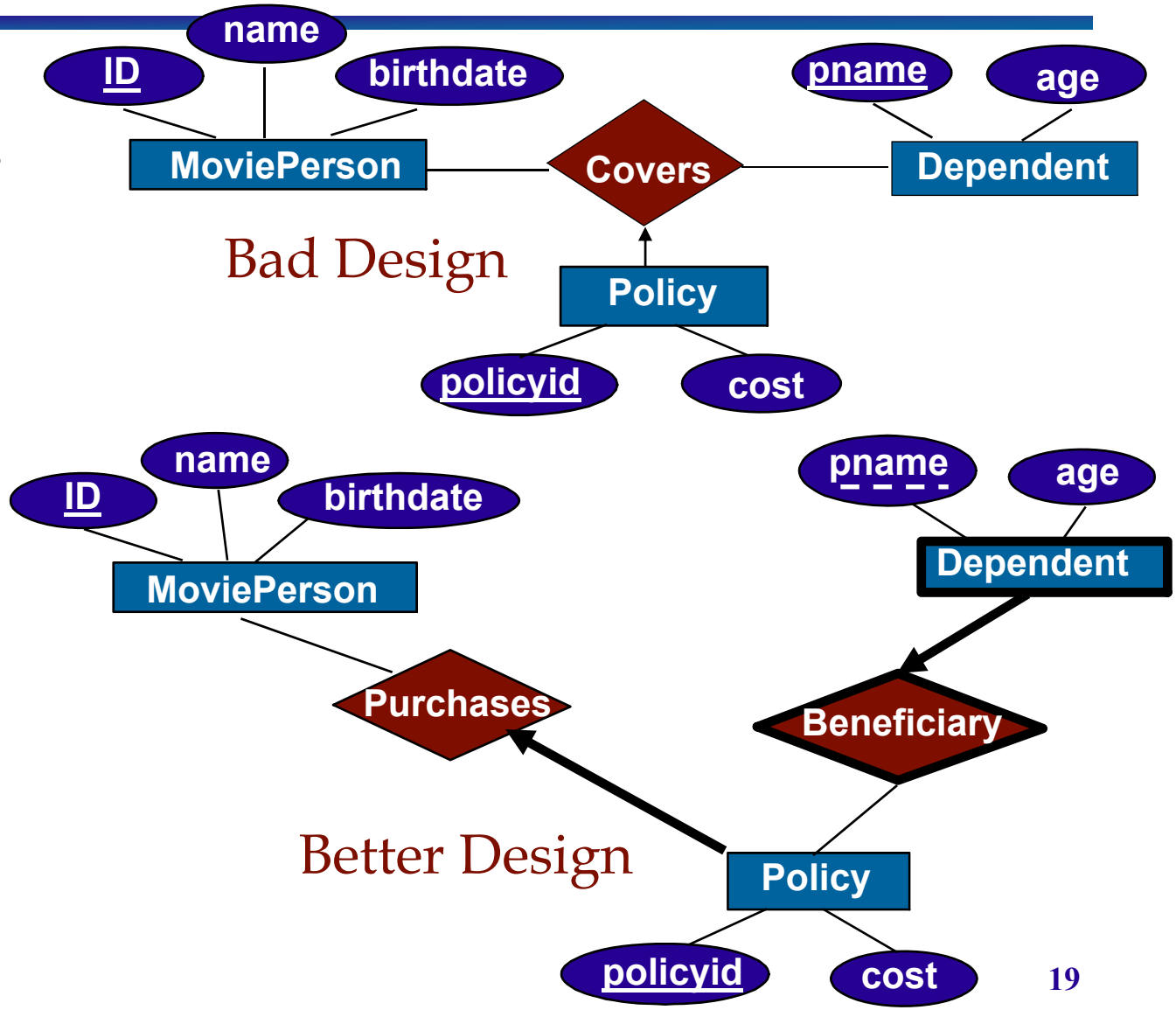
- How are these two ER models different?
- A director can get a separate assistant for each movie.
- All directors must direct a movie, and have the same assistant (if they have one) for *all* movies.





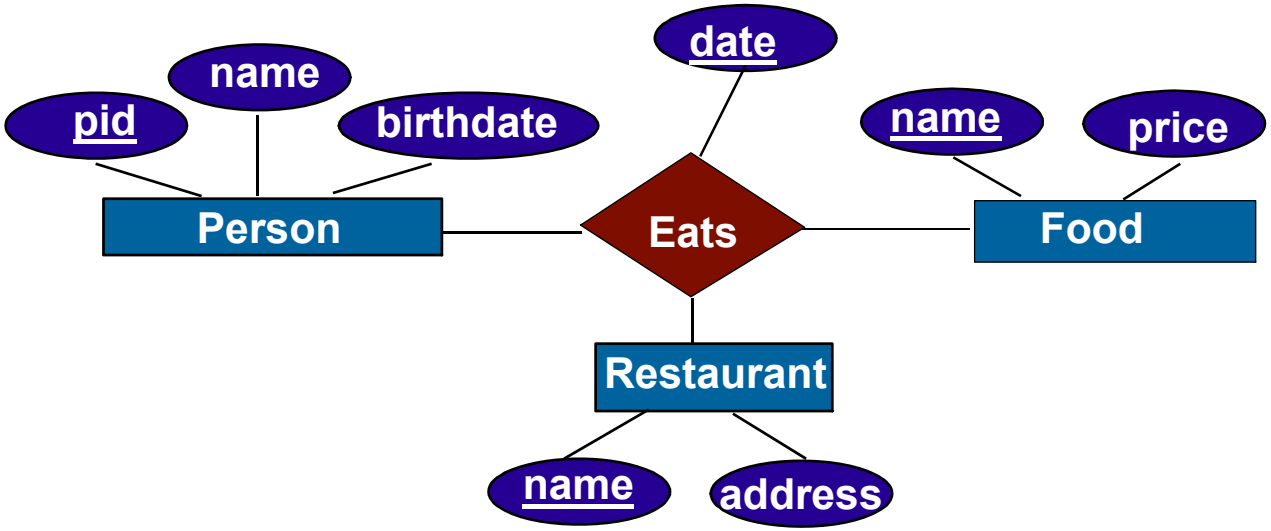
Binary vs. Ternary Relationships

- If each policy is owned by just one person:
  - Key constraint on Policies would mean policy can only cover one dependent!
- What are the additional constraints in the 2nd diagram?



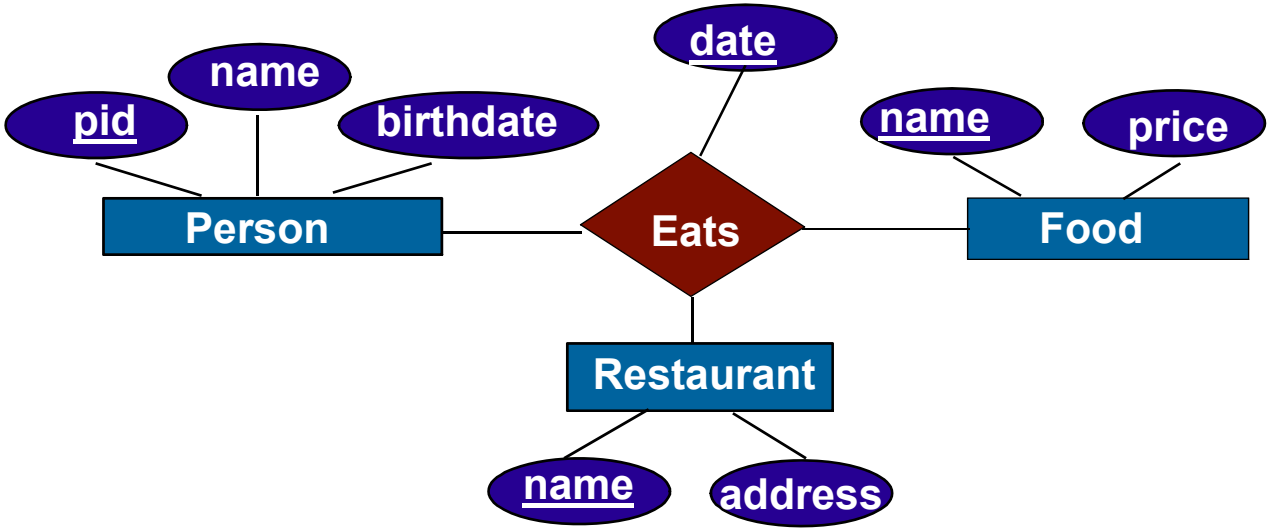
Binary vs. Ternary Relationships

- An example in the other direction: the ternary relation Eats relates entity sets Person, Food, and Restaurant, and has a descriptive attribute date.



- Can we use two binary relationships instead?

Binary vs. Ternary Relationships vs. Aggregation



- No combination of binary relationships is an adequate substitute:
  - P “likes” F, P “visits” R, and R “provides” F This does not imply that P eats F in R.
  - Also, how would we record the date of the meal?
- Aggregation can be used instead of a ternary relation if we need to impose additional constraints.
  - e.g., a person cannot have more than one meal in the same restaurant

Summary of Conceptual Design

- 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 and close to the way people think about their applications.
- Basic constructs: *entities*, *relationships*, and *attributes* (of entities and relationships).
- Some additional constructs: *weak entities*, *ISA relationships*, and *aggregation*.
- Note: There are many variations on the ER model.

Summary of ER (Cont.)

- Several kinds of integrity constraints can be expressed in the ER model: *key constraints*, *participation constraints*, and *overlap/covering constraints* for ISA relationships. Some *foreign key constraints* are also implicit in the definition of a relationship set.
  - 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.

Summary of ER (Cont.)

- ER design is *subjective*. There are often many ways to model a given scenario! 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: resulting relational schema should be analyzed and refined further.

Learning Goals Revisited

- Explain the purpose of an ER diagram, and list its major components.
- Read and interpret an ER diagram.
- Given a problem description and specification, create an ER diagram. Justify the decisions you make for entities, relationships, keys, key constraints, participation constraints, weak entities, is-a relationships, aggregations, etc.
- Given a problem description, identify alternative representations of the problem concepts and evaluate the choices.
- Compare alternative ER models for the same domain and identify their strengths and weaknesses.