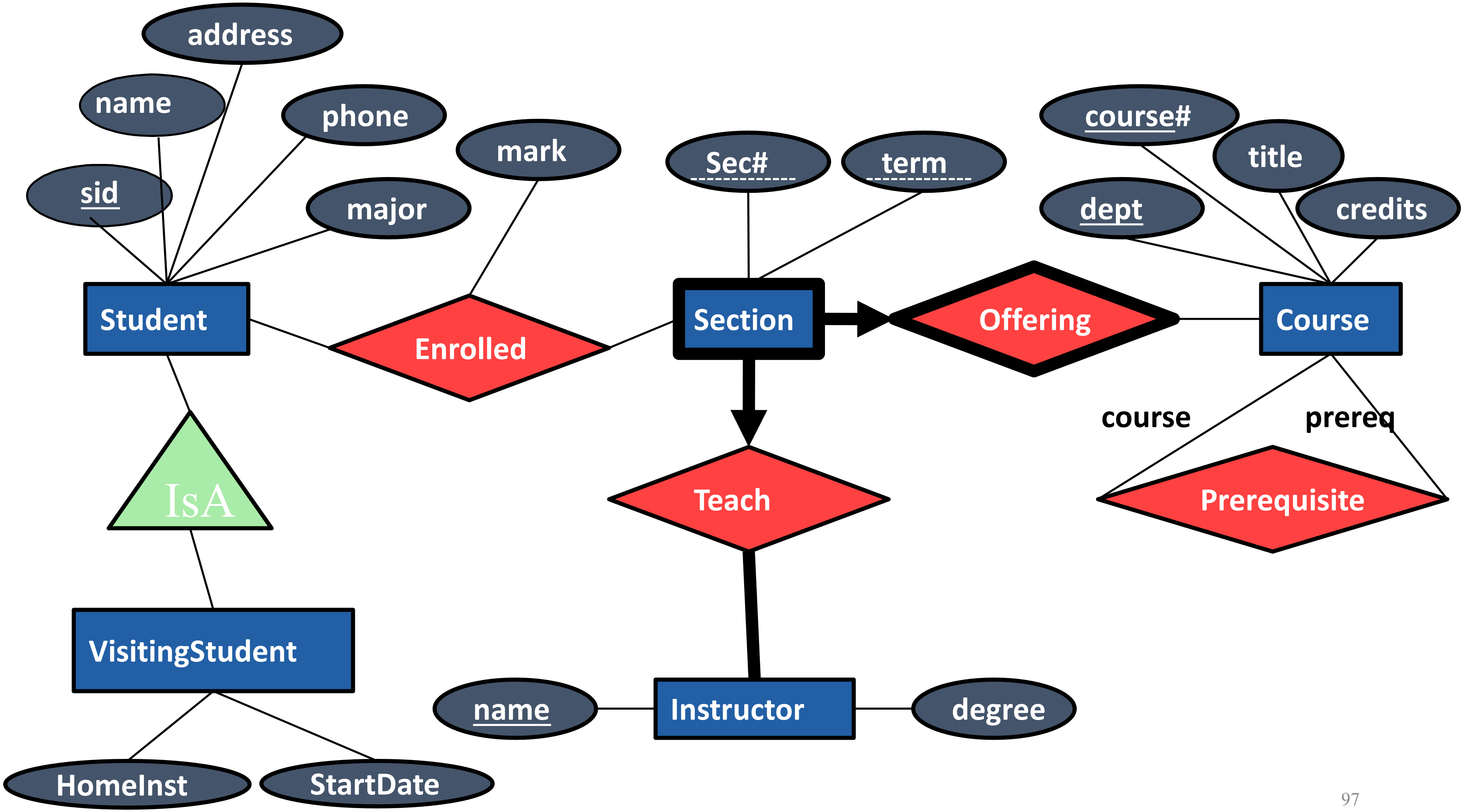
Administrative Notes- Jan 24, 2023

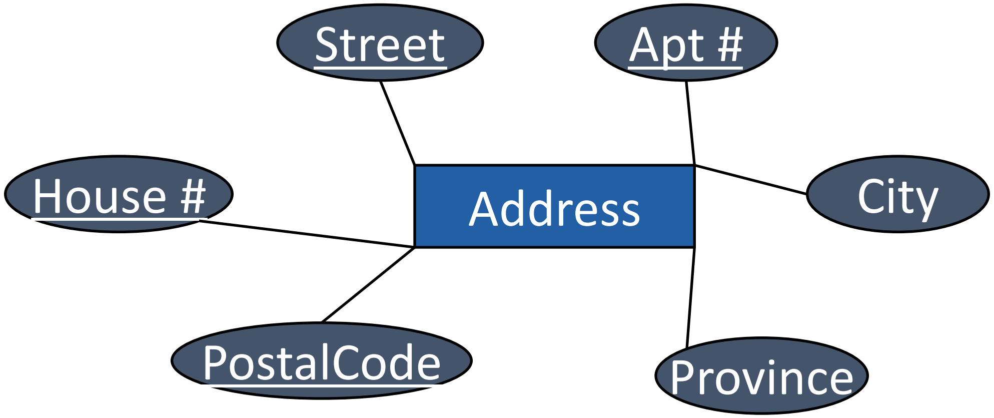
- If you haven't already, please register your clicker on Canvas
  - If you do not do this, then we will not be able to sync your iClicker grades to Canvas
- Don't forget to refer to the tentative schedule for assignment deadlines and tutorial information
- Jan 24: In-class assignment 1 due
  - Submit your own work for in-class exercises (i.e., don't submit the lecture slide with the answer)
- Jan 27: Assignment 1 due
- Assignment 2 available on Canvas
- You will need a physical student card for the class so please get one

Sample solution



That’s all there is to it

- Some ER models differ in expressiveness
- They model *most* concepts people want
- They don’t model all of them, e.g.,
  - Functional dependencies – some attributes determine some other attributes, e.g., postal code determines (only) city and province



Conceptual Design Using the ER Model

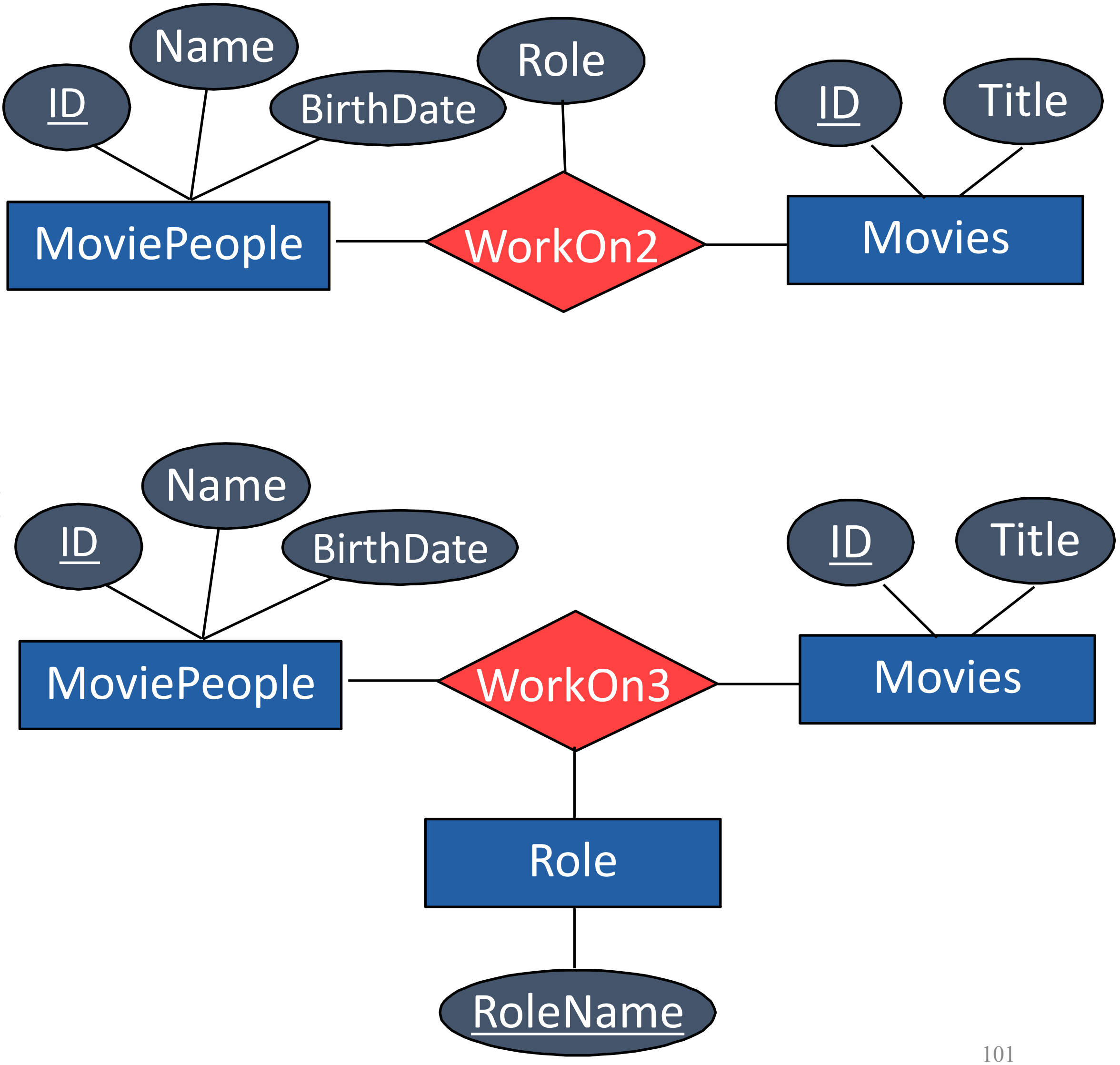
- Design choices:
  - Should a concept be modeled as an entity or an attribute?
  - Should a concept be modeled as an entity or a relationship?
  - Identifying relationships: Binary or ternary? Aggregation?
- Constraints in the ER Model:
  - A lot of data semantics can (and should) be captured.
  - But some constraints cannot be captured in ER diagrams.
    - i.e. domain constraints
    - dependencies

Entity vs. Attribute

- Should an *address* be an attribute of MoviePeople or an entity (connected to MoviePeople by a relationship)?
- Depends upon
  - the use we want to make of 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).
    - If a person has only one street address, one city, one province, one postal code, etc. then these should simply be attributes.

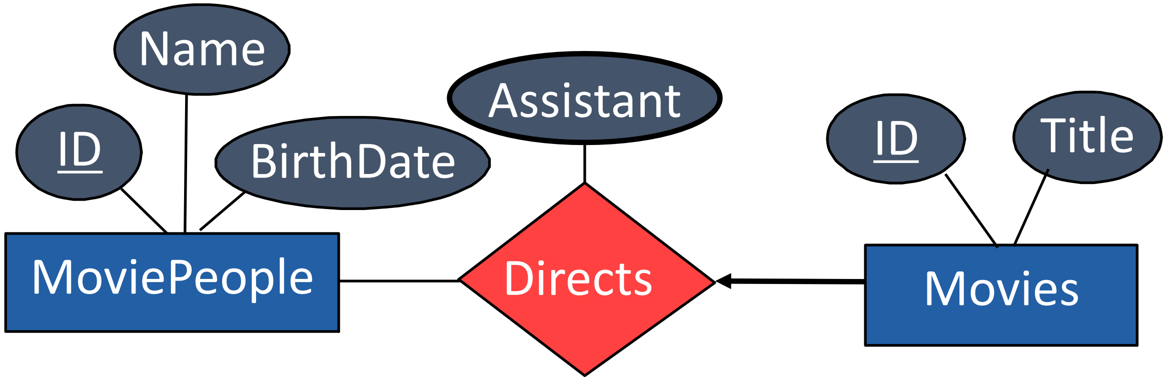
Entity vs. Attribute (Cont.)

- WorkOn2 does not allow a person to have more than one role in the same movie.
- We want to associate the same pair (MoviePerson, Movie) with more than one set of values for the descriptive attributes?
- Solution: change descriptive attributes into entities.

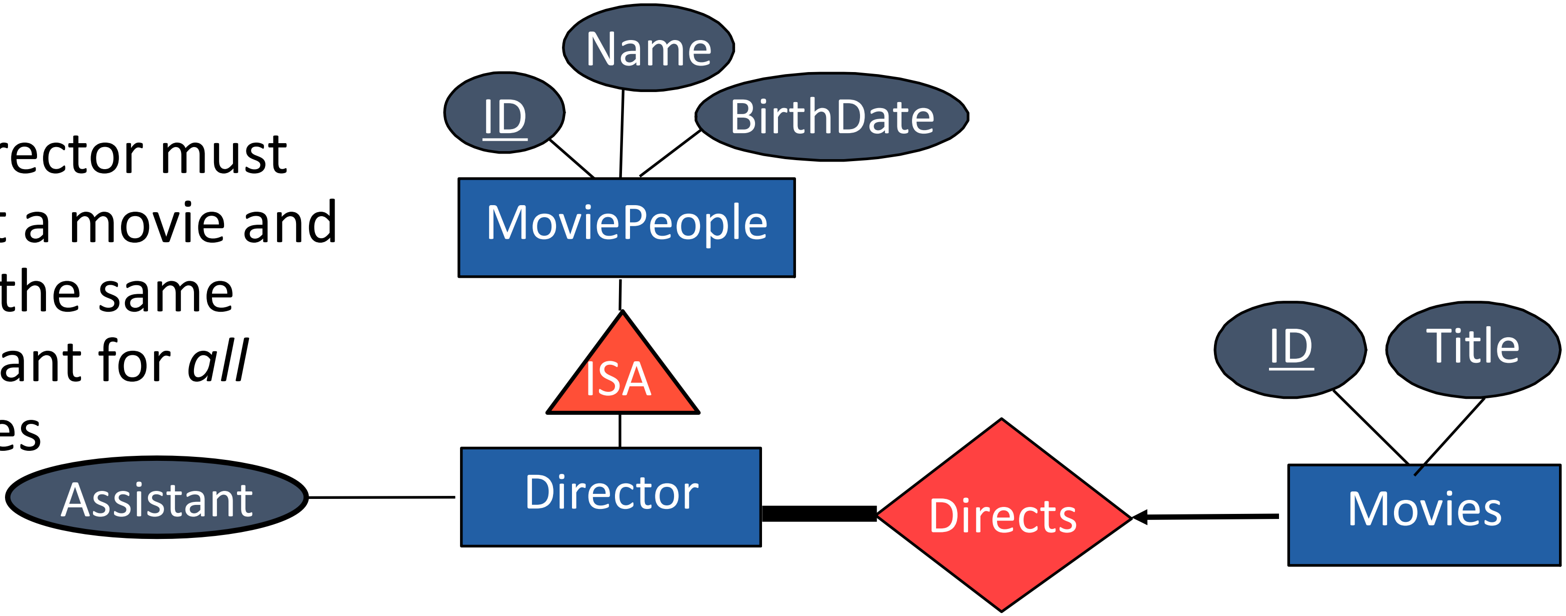


Entity vs. Relationship

- How are the two ER models different?

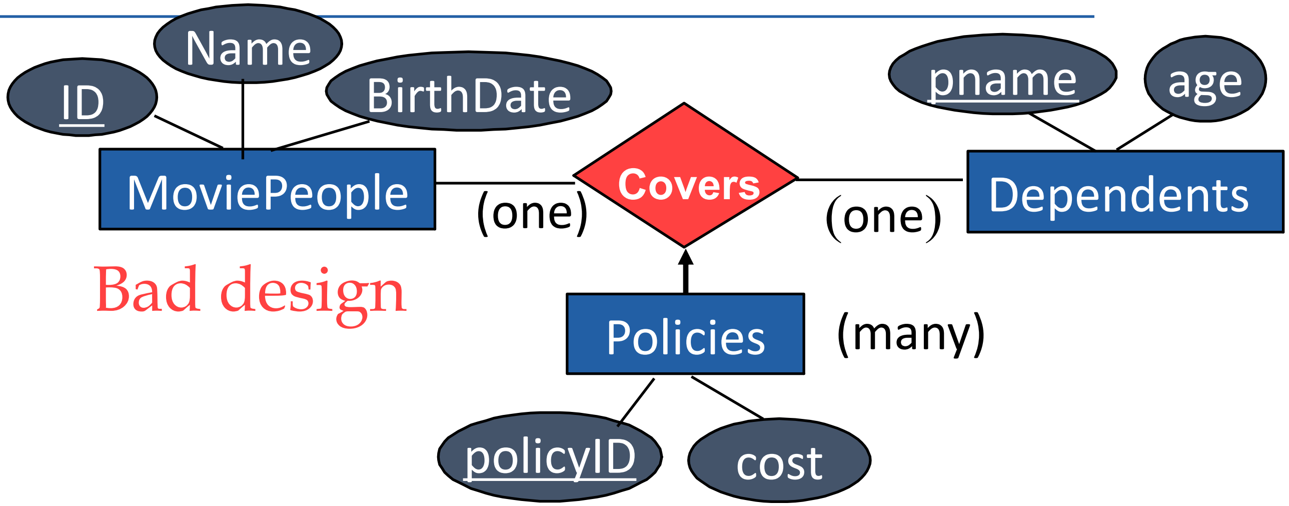


- Director can get a separate assistant for each movie.
- All director must direct a movie and have the same assistant for *all* movies

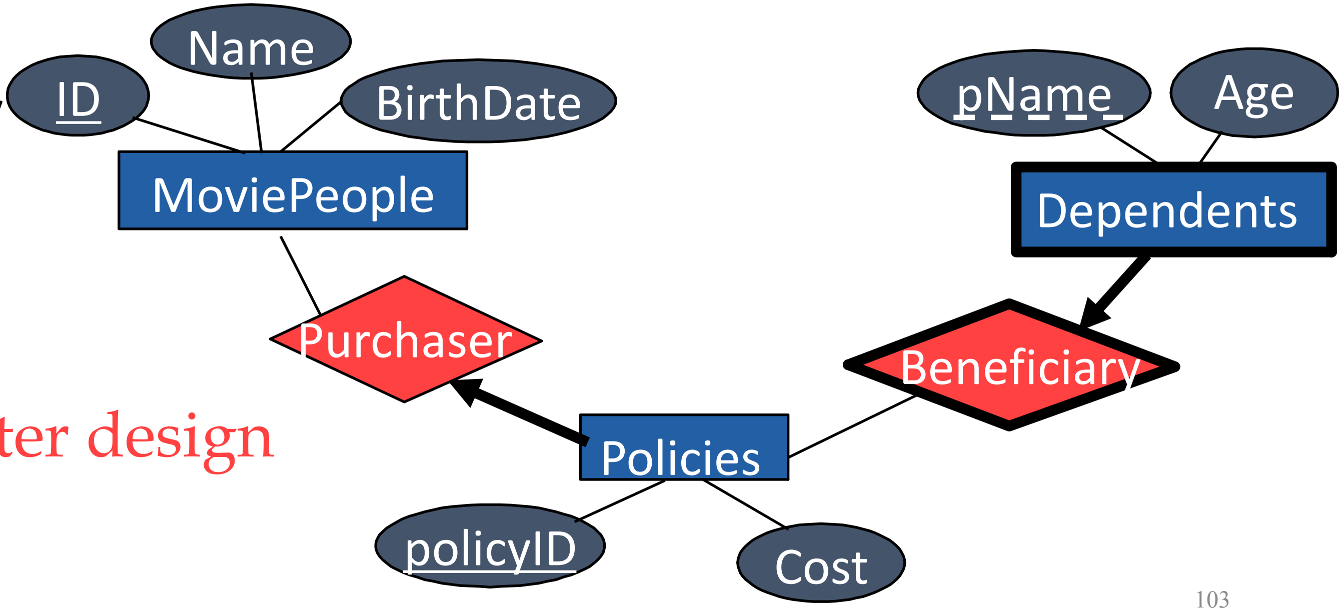


Binary vs. Ternary Relationships

- If each policy is owned by just 1 person:
  - Key constraint on Policies would mean policy can only cover 1 dependent!



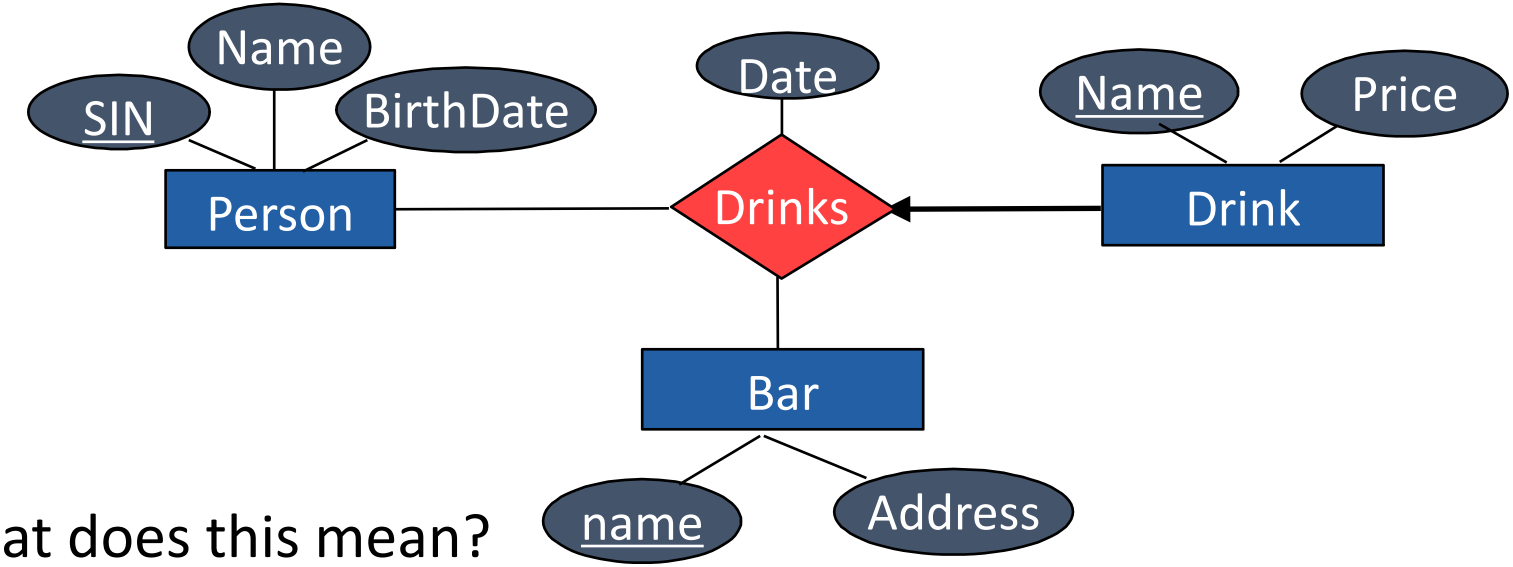
Bad design



Better design

Key constraints on non-binary relationships
Note: no guarantee this diagram makes sense

Ternary relation Drinks relates entity sets Person, Bar and Drink, and has descriptive attribute *date*.



What does this mean?

[✓ A. If I know drink, then I know both the person and the bar]

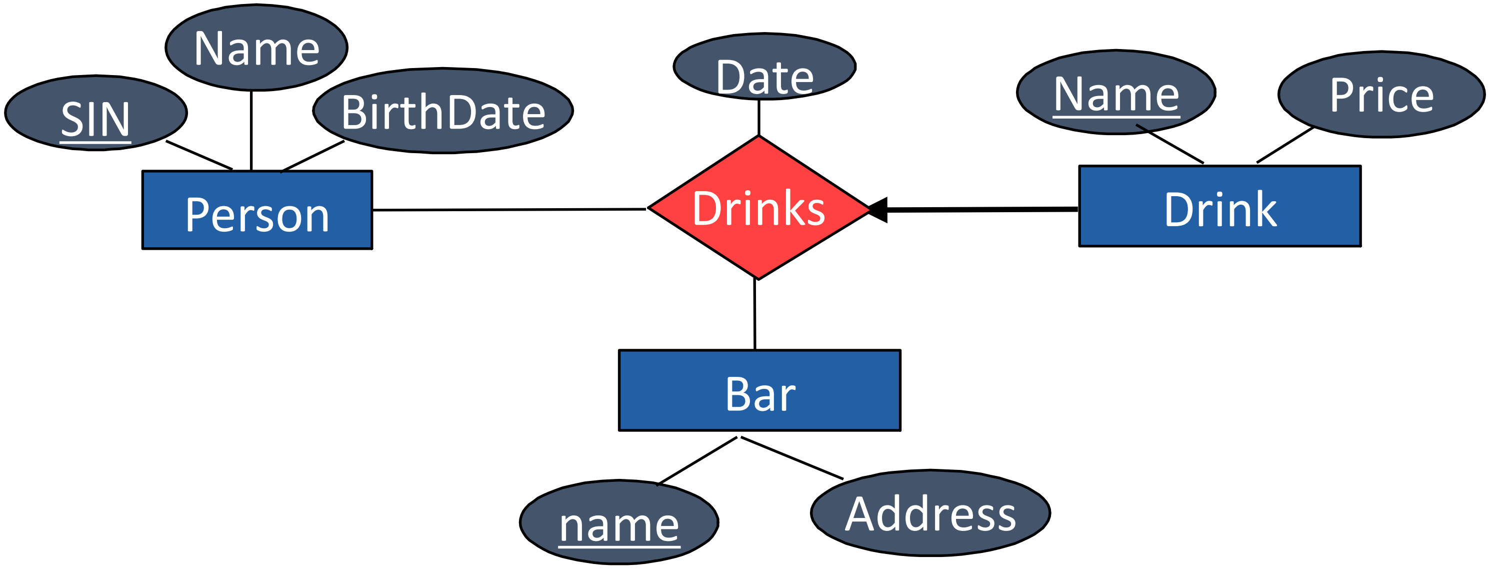
B. If I know drink I only know the person

C. If I know drink, I only know the bar

D. Other. I don’t know. My head hurts. Make it stop.

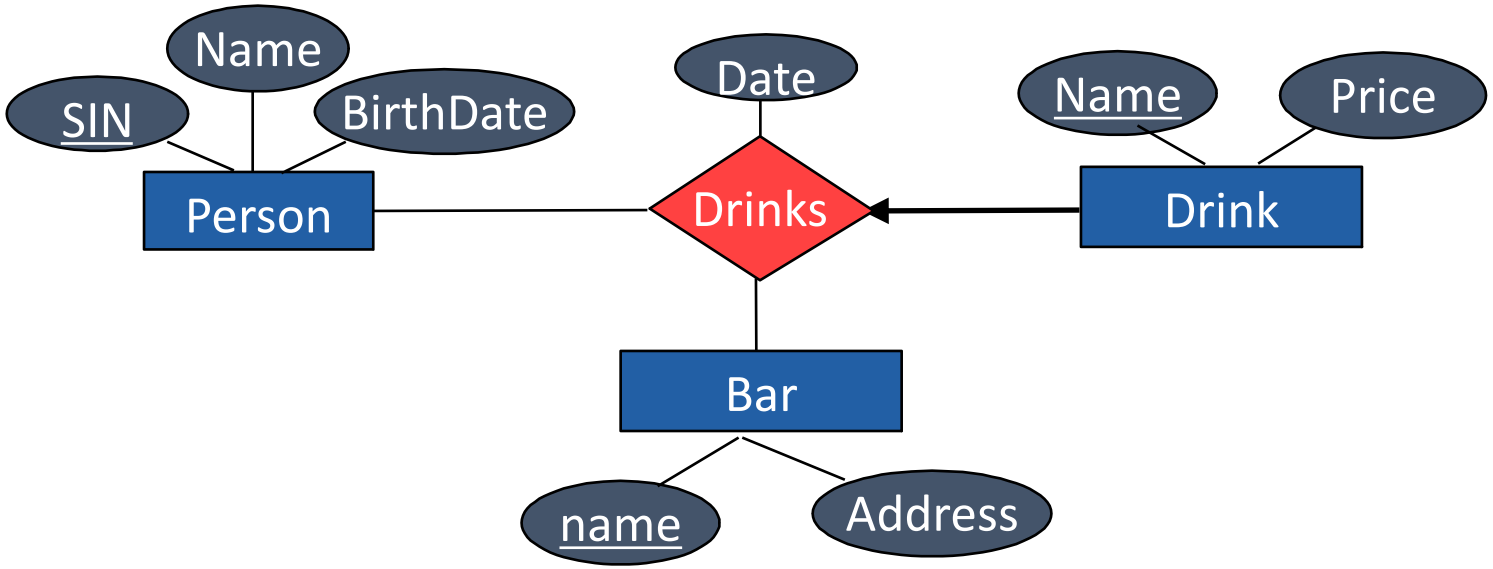
Binary vs. Ternary Relationships

- An example in the other direction: a ternary relation Drinks relates entity sets Person, Bar and Drink, and has descriptive attribute *date*.



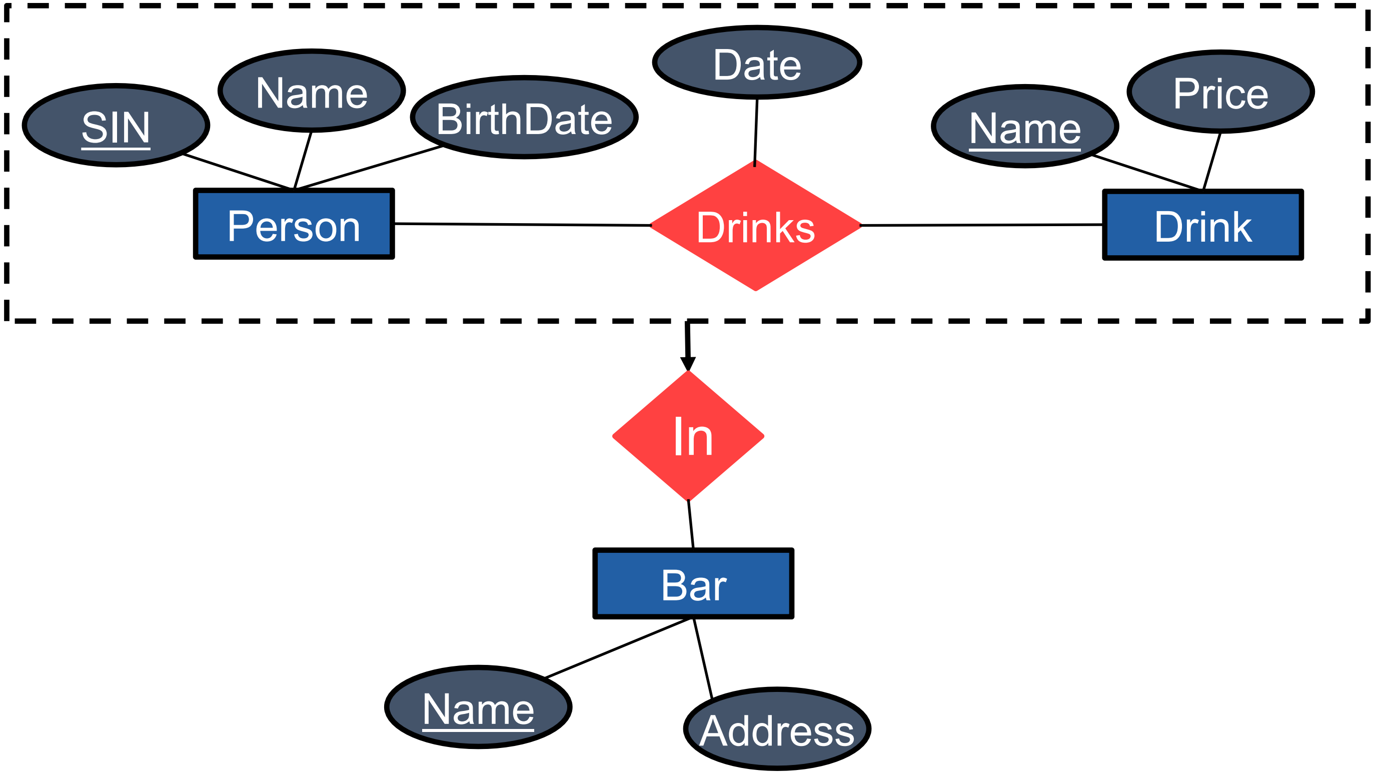
- Can we use two of binary relationships instead?

Binary vs. Ternary Relationships vs. Aggregation



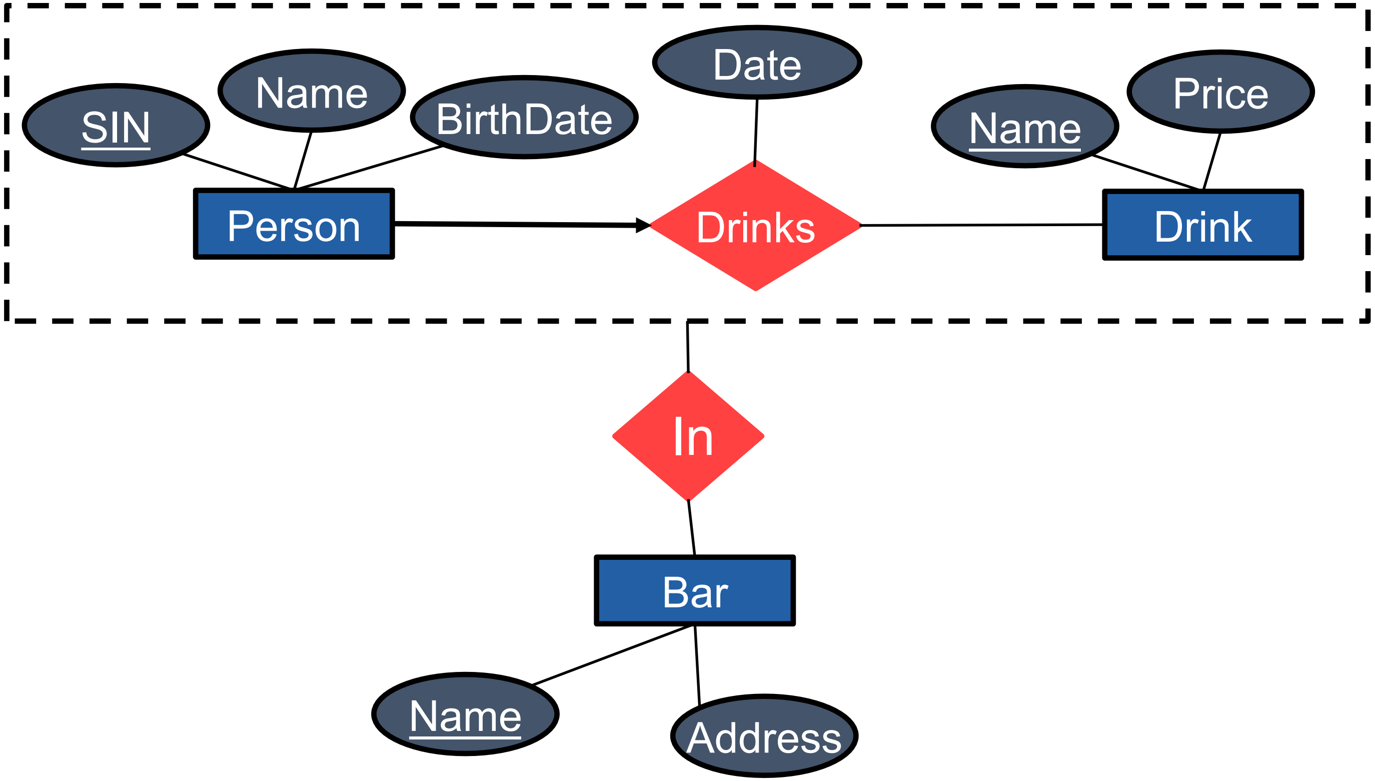
- No combination of binary relationships is an adequate substitute:
  - P “likes” D, P “visits” B, and B “provides” D does not imply that P drinks D in B.
  - Also, how would we record *date*?

Binary vs. Ternary Relationships vs. Aggregation



- Aggregation can be used instead of a ternary relation if need to impose additional constraints:
  - I.e., If you know the person and the drink, you know the bar they drank it at.

Binary vs. Ternary Relationships vs. Aggregation



- Aggregation can be used instead of a ternary relation if need to impose additional constraints:
  - I.e. A person cannot have more than one drink ever. A person only drinks that one drink at any bar.

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.
- 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 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 the major components.
- Given a problem description, create an ER diagram given a specification. Justify the decisions you make for entities, relationships, keys, key constraints, participation constraints, weak entities, is-a relationships, and aggregations.
- 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