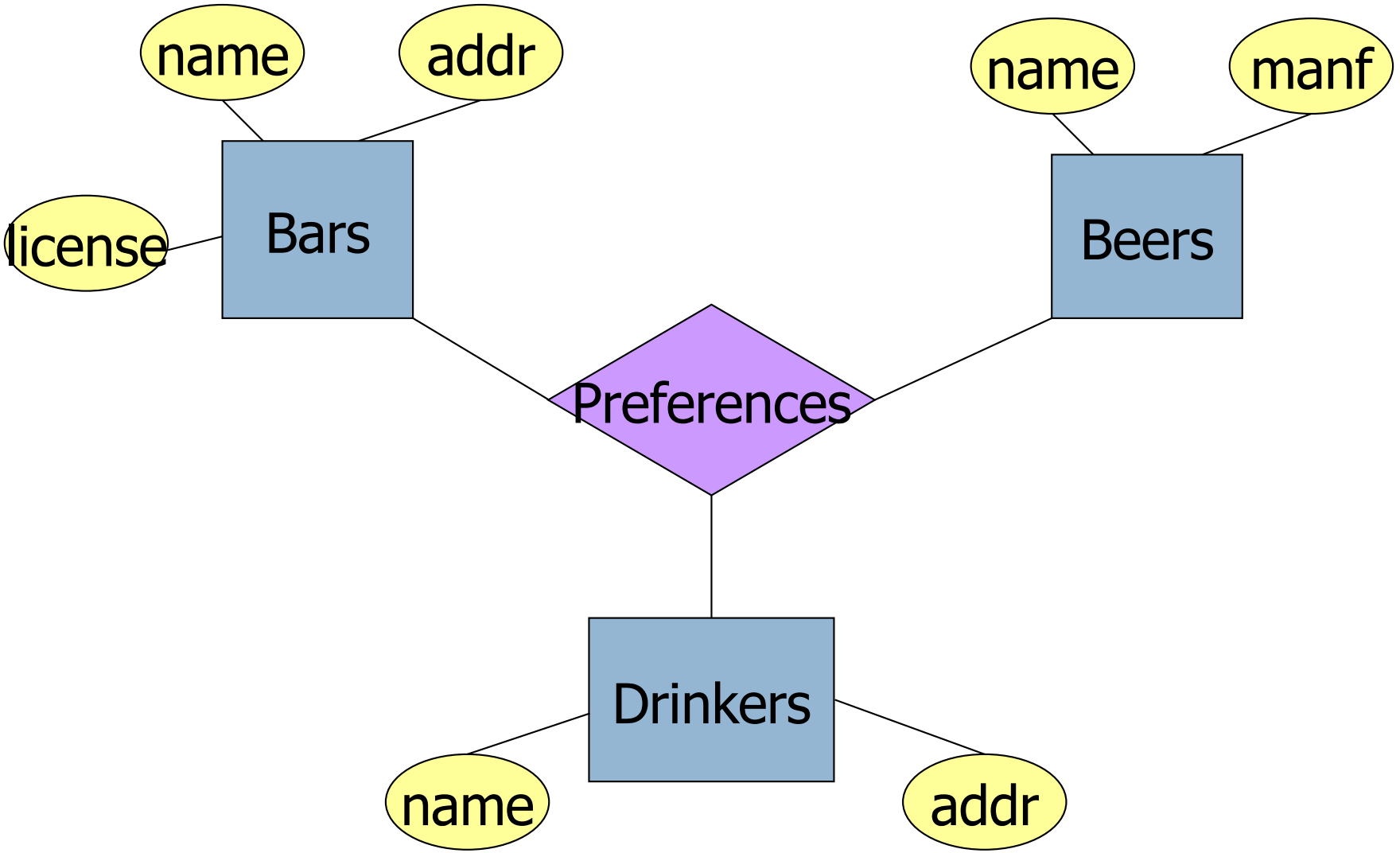
Example: 3-Way Relationship



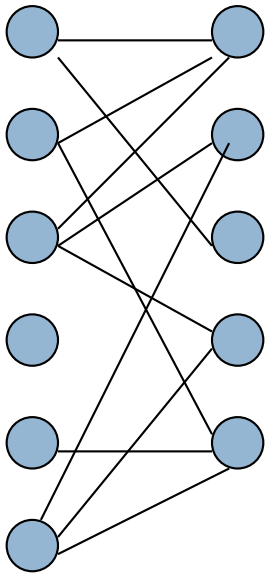
A Typical Relationship Set

Bar	Drinker	Beer
Joe's Bar	Ann	Miller
Sue's Bar	Ann	Bud
Sue's Bar	Ann	Pete's Ale
Joe's Bar	Bob	Bud
Joe's Bar	Bob	Miller
Joe's Bar	Cal	Miller
Sue's Bar	Cal	Bud Lite

Many-Many Relationships

- Focus: binary relationships, such as Sells between Bars and Beers.
- In a *many-many* relationship, an entity of either set can be connected to many entities of the other set.
  - E.g., a bar sells many beers; a beer is sold by many bars.

In Pictures:



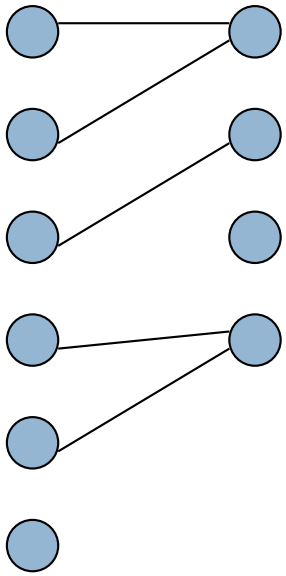
many-many

Note: each line is an instance of the binary relationship

Many-One Relationships

- Some binary relationships are *many -one* from one entity set to another.
- Each entity of the first set is connected to at most one entity of the second set.
- But an entity of the second set can be connected to zero, one, or many entities of the first set.

In Pictures:



Bars

Beers

Drinkers

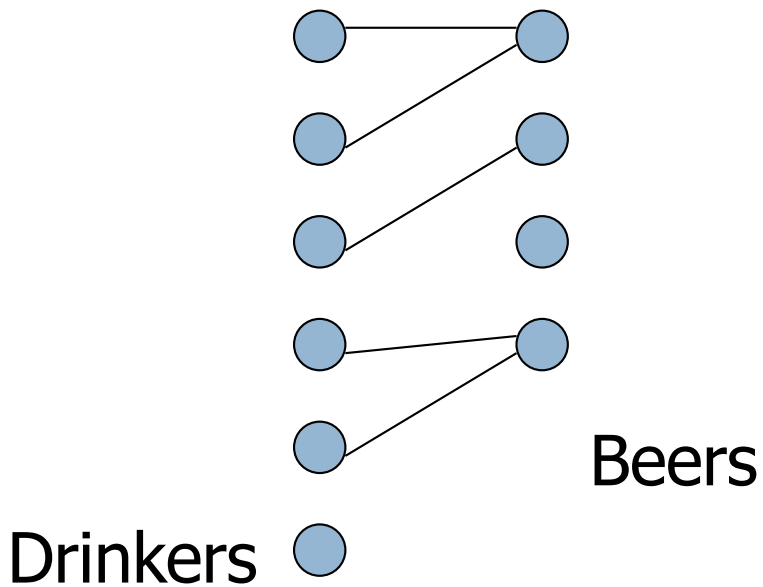
many-one

(Partial) Function on entity set

Example: Many-One Relationship

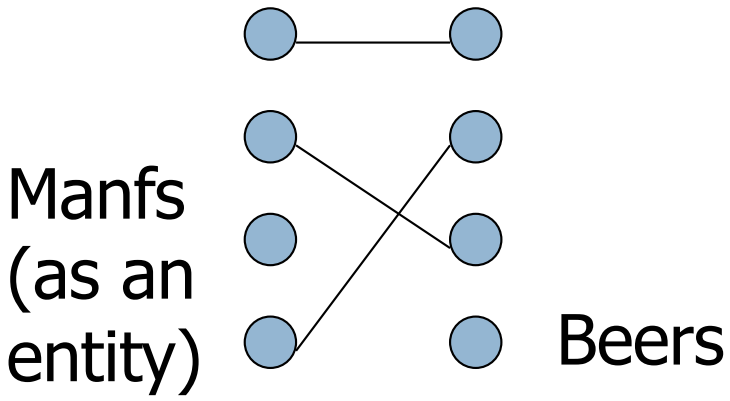
- Favourite, from Drinkers to Beers is many-one.

- A drinker has at most one favourite beer.
- But a beer can be the favorite of any number of drinkers, including zero.



One-One Relationships

- In a *one-one* relationship, each entity of either entity set is related to at most one entity of the other set.
- Example: Relationship Best-seller between entity sets Manfs (manufacturer) and Beers.
  - A beer is the best seller for 0 or 1 manufacturers, and no manufacturer can have more than one best-seller (assume no ties).

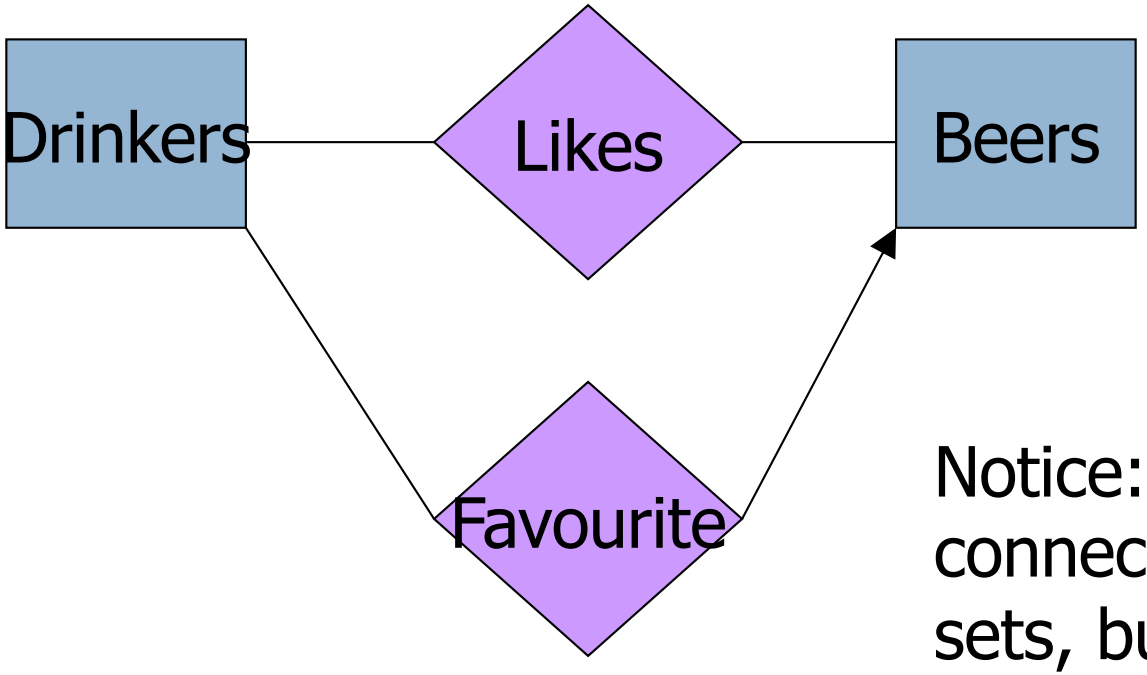


Representing "Multiplicity"

- Show a many-one relationship by an arrow entering the "one" side.
  - "at most one"
- Show a one-one relationship by arrows entering both entity sets.

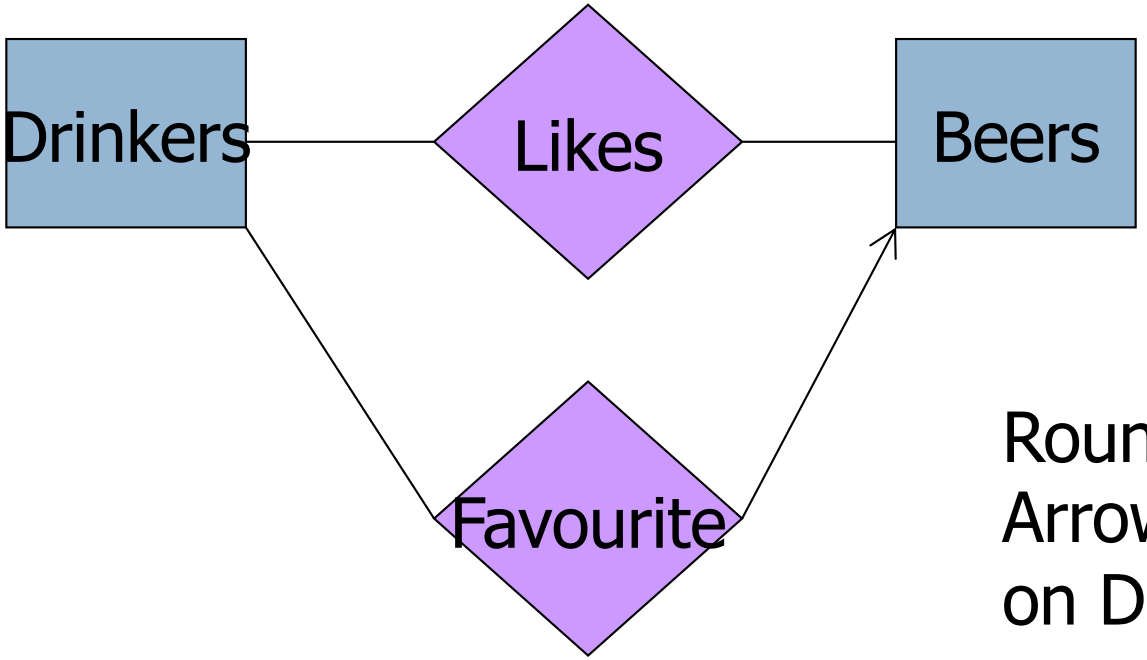
Rounded (open) arrow = "exactly one," i.e., each entity of the first set is related to exactly one entity of the target set.

Example: Many-One Relationship



Notice: two relationships connect the same entity sets, but are different.

Example: Many-One Relationship



Rounded (open)
Arrow = total function
on Drinkers

Example: One-One Relationship

- Consider Best-seller between Manfs and Beers.
- Some beers are not the best-seller of any manufacturer
- But a beer manufacturer has to have a best-seller.

In the E/R Diagram

A beer is the best-seller for 0 or 1 manufacturer.

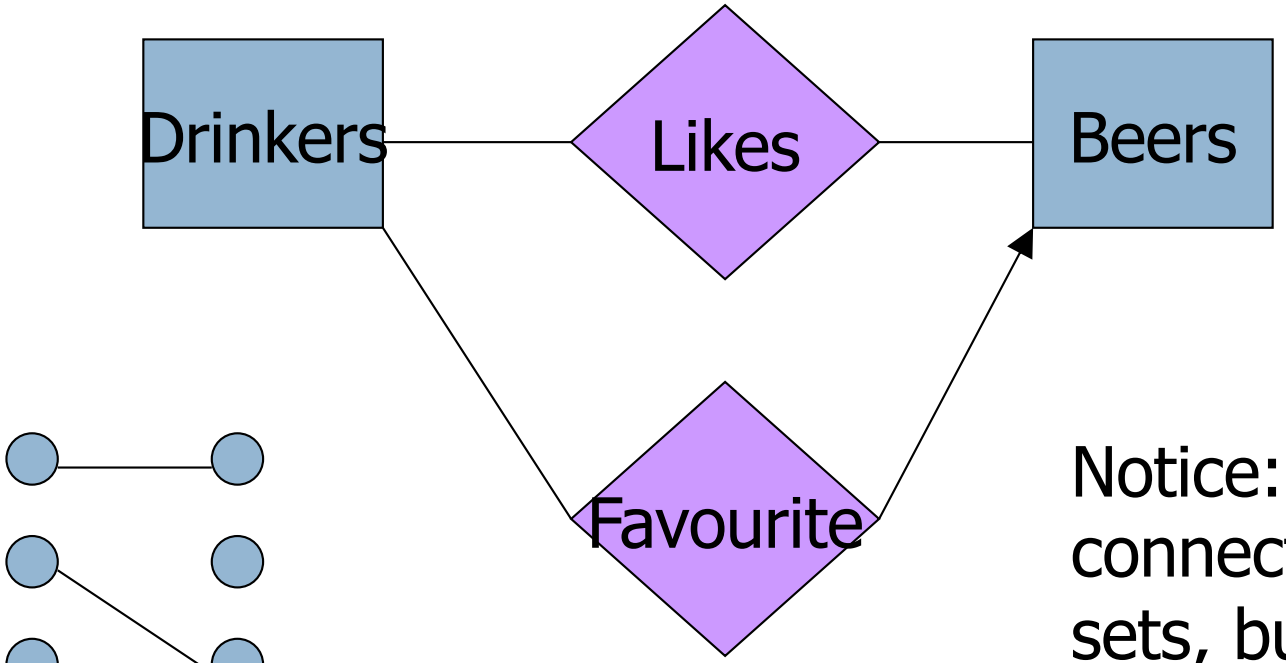
A manufacturer has exactly one best seller.

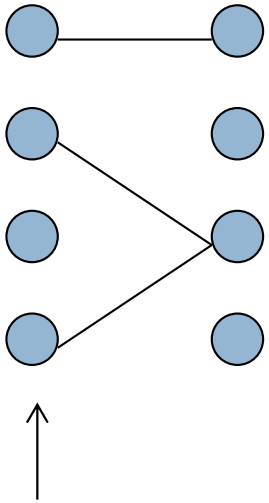
Participation Constraints

- Does every student have to take a course?
  - If so, this is a *participation constraint*: the participation of Students in Enrolled is said to be *total* (vs. *partial*).
  - Every *sid* value in Students table must appear in a row of the Enrolled table (with a non-null *sid* value!)

- Textbook notation: total participation represented by a thick (bolded) line originating from entity

Example: Many-One Relationship

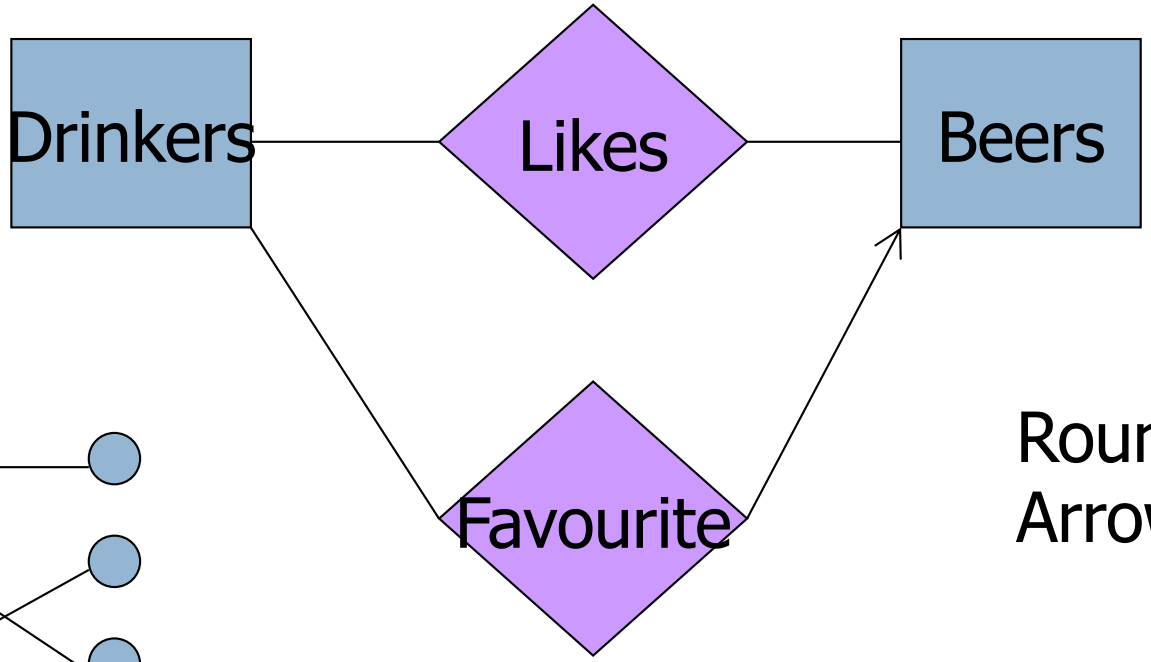


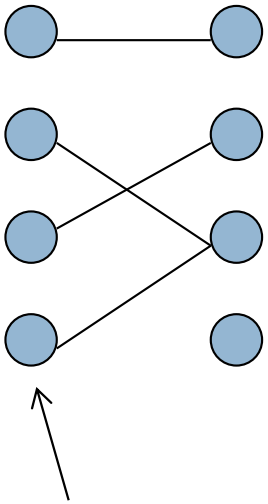


Participation of this entity is partial

Notice: two relationships connect the same entity sets, but are different.

Example: Many-One Relationship

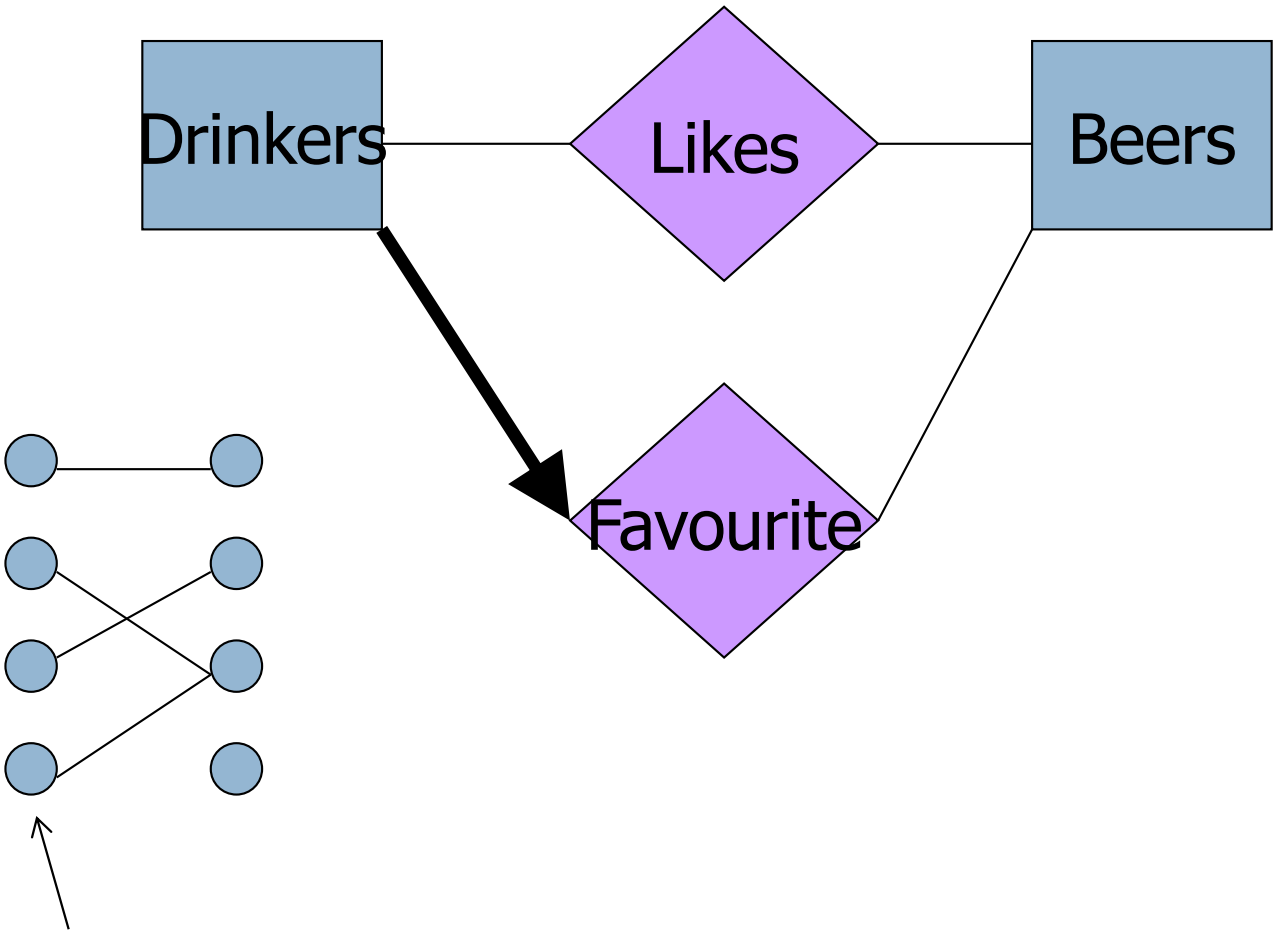




Rounded (open)
Arrow = total function

Participation of this entity is total

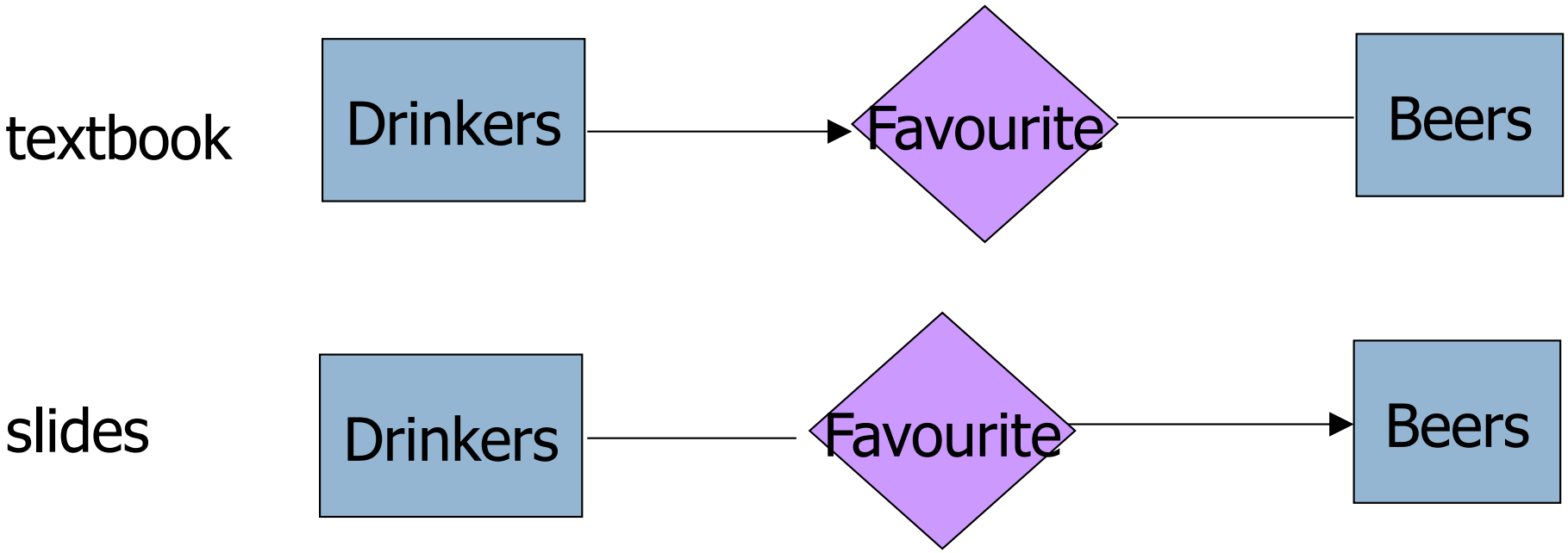
Alternative (Textbook) Notation



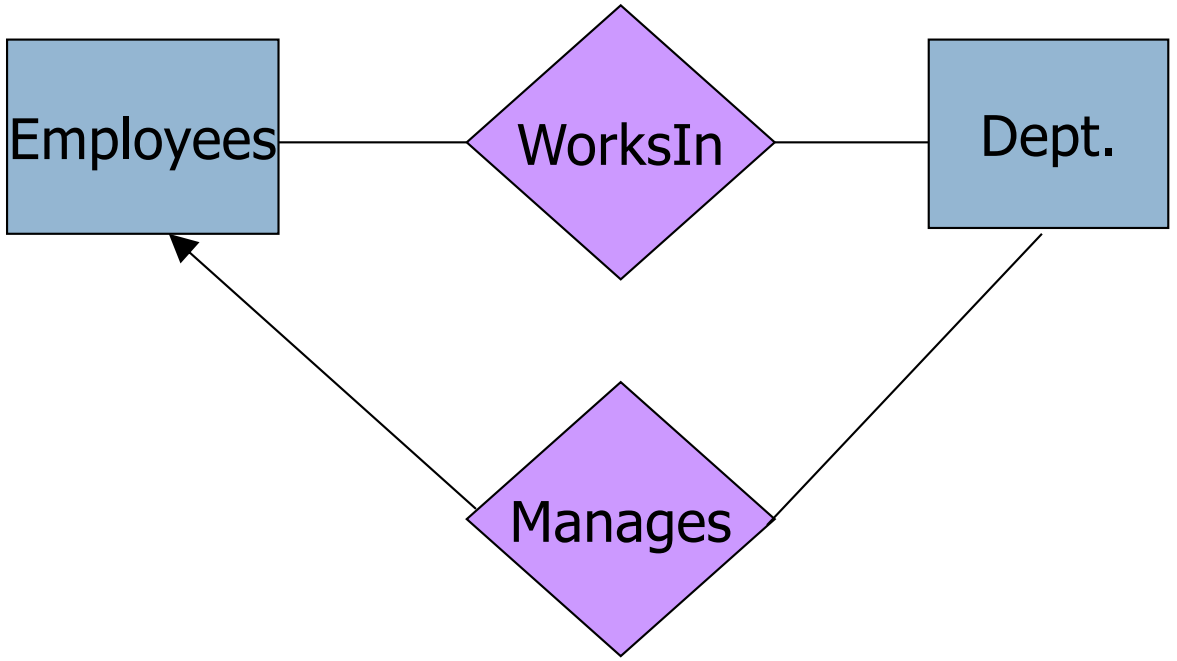
Participation of this entity is total

Notation

- Be consistent with your chosen notation!



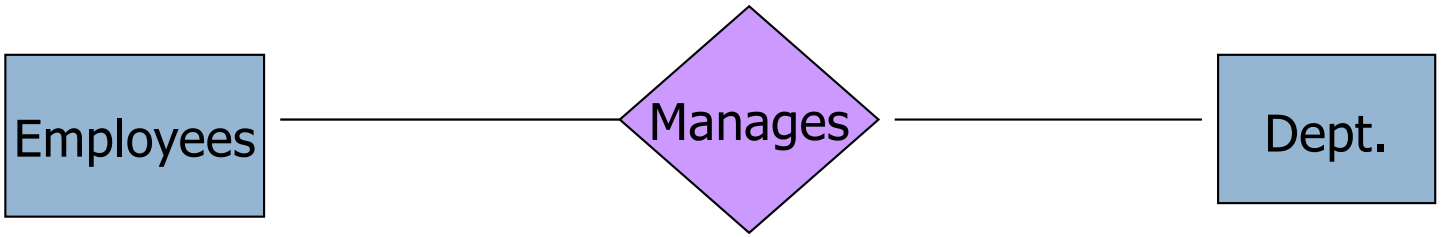
Key Constraints

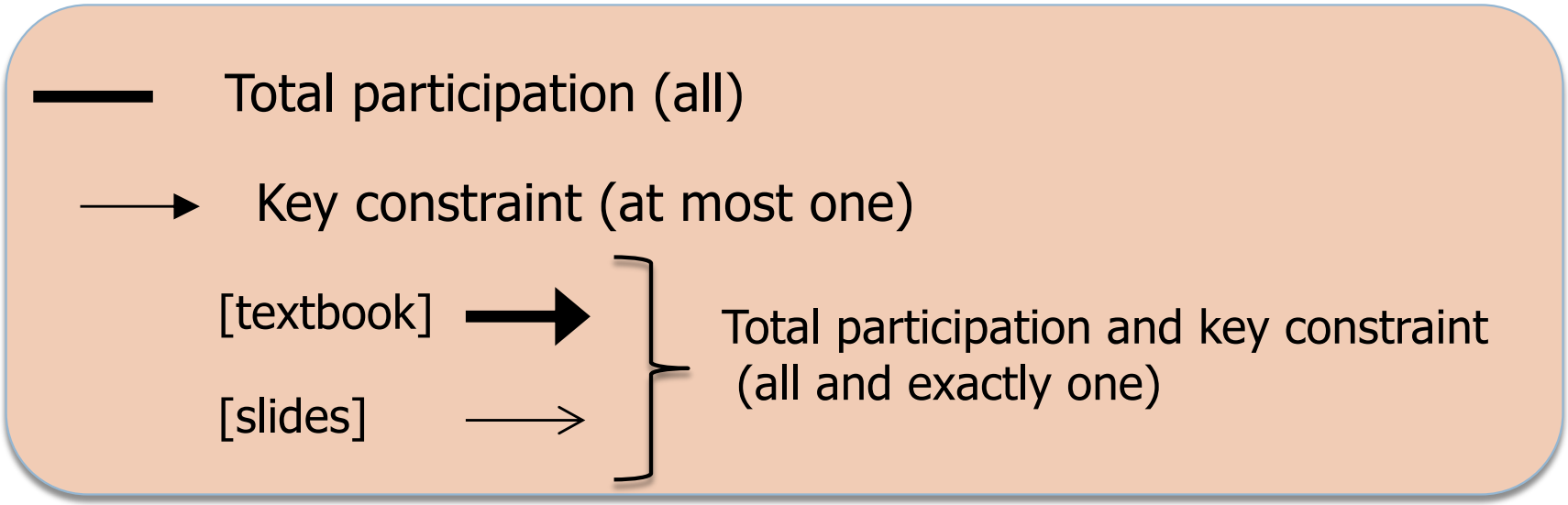


- Many-many: "An employee can work in many depts, and a dept. can have many employees
- One-many: A dept has at most one manager, and employees can manage many departments

Participation Constraints

- Does every dept. have to have a manager?
  - If yes, then every dept. must appear in the manages relation: total participation (vs. partial)

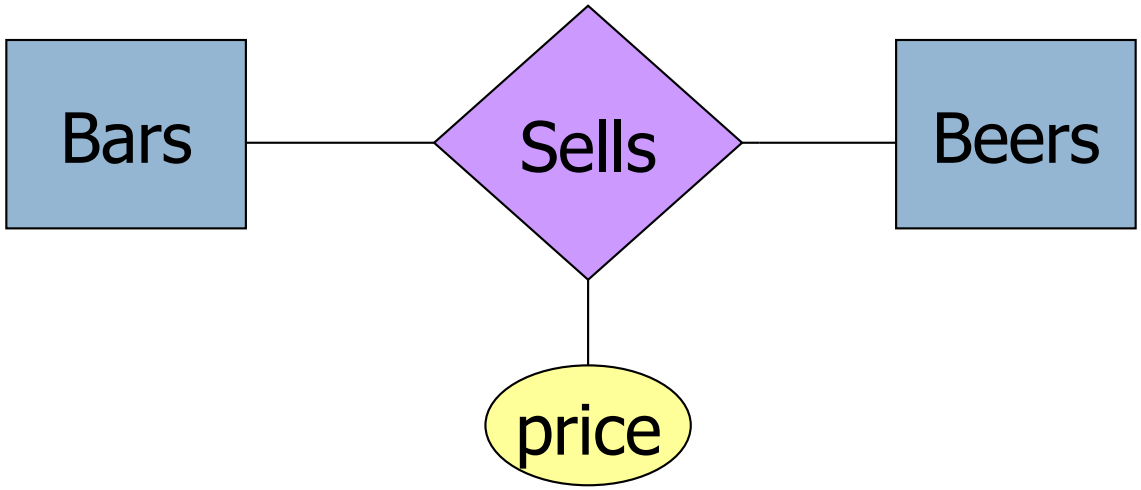




Attributes on Relationships

- Sometimes it is useful to attach an attribute to a relationship.
- Think of this attribute as a property of tuples in the relationship set.

Example: Attribute on Relationship

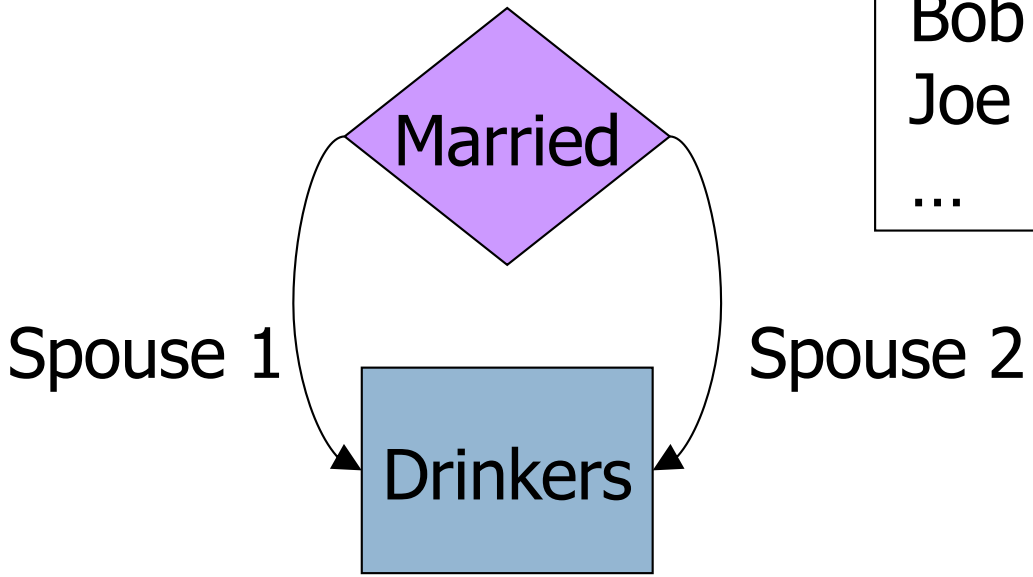


Price is a function of both the bar and the beer, not of one alone.
E.g., "The price of Miller beer at Joe's bar"

Roles

- Sometimes an entity set appears more than once in a relationship.
- Label the edges between the relationship and the entity set with names called *roles*.

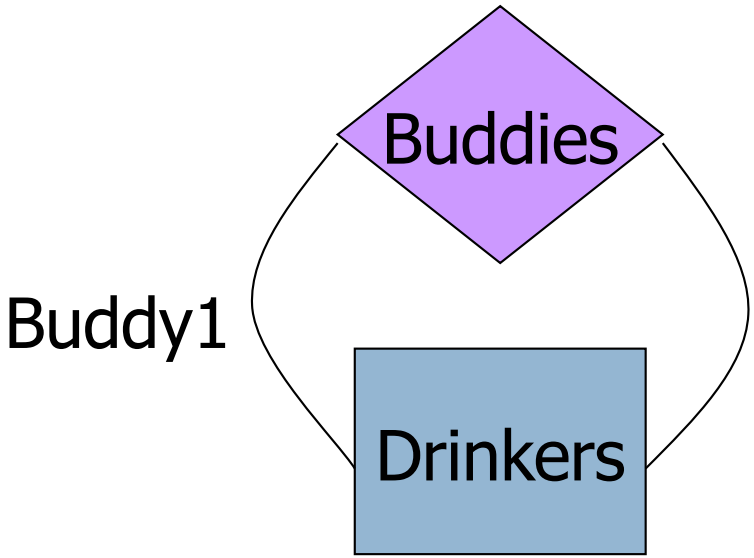
Example: Roles



Relationship Set

Spouse1	Spouse2
Bob	Ann
Joe	Sue
...	...

Example: Roles



Relationship Set

Buddy1	Buddy2
Bob	Ann
Joe	Sue
Ann	Bob
Joe	Moe
…	…

Subclasses

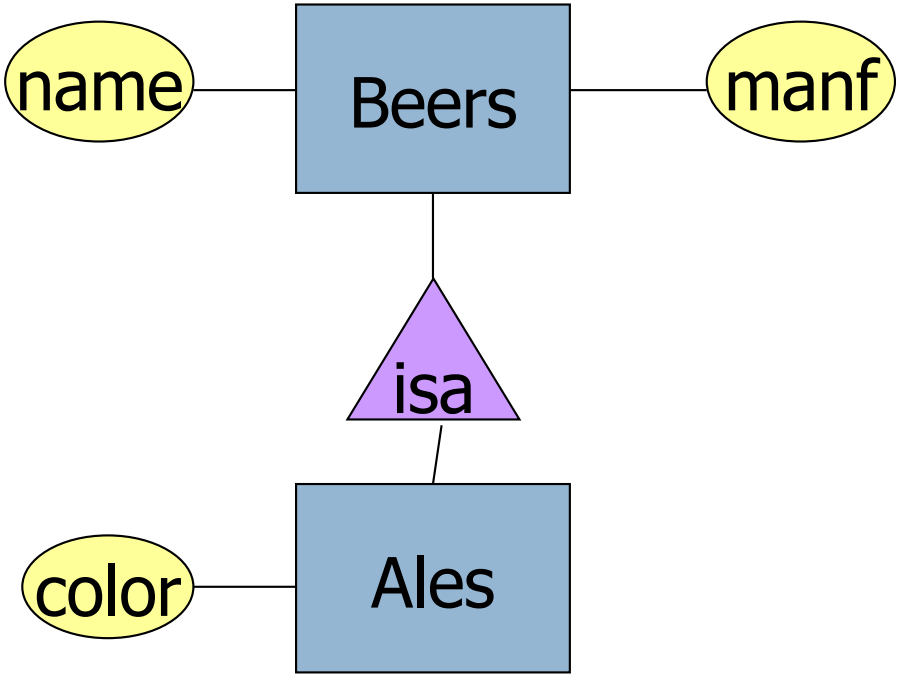
- *Subclass* = special case = more properties.
- Example: Ales are a kind of beer.
  - Not every beer is an ale, but some are.
  - Let us suppose that in addition to all the *properties* (attributes and relationships) of beers, ales also have the attribute color.

Subclasses in E/R Diagrams

- isa triangles indicate the subclass relationship.
  - Point to the superclass.

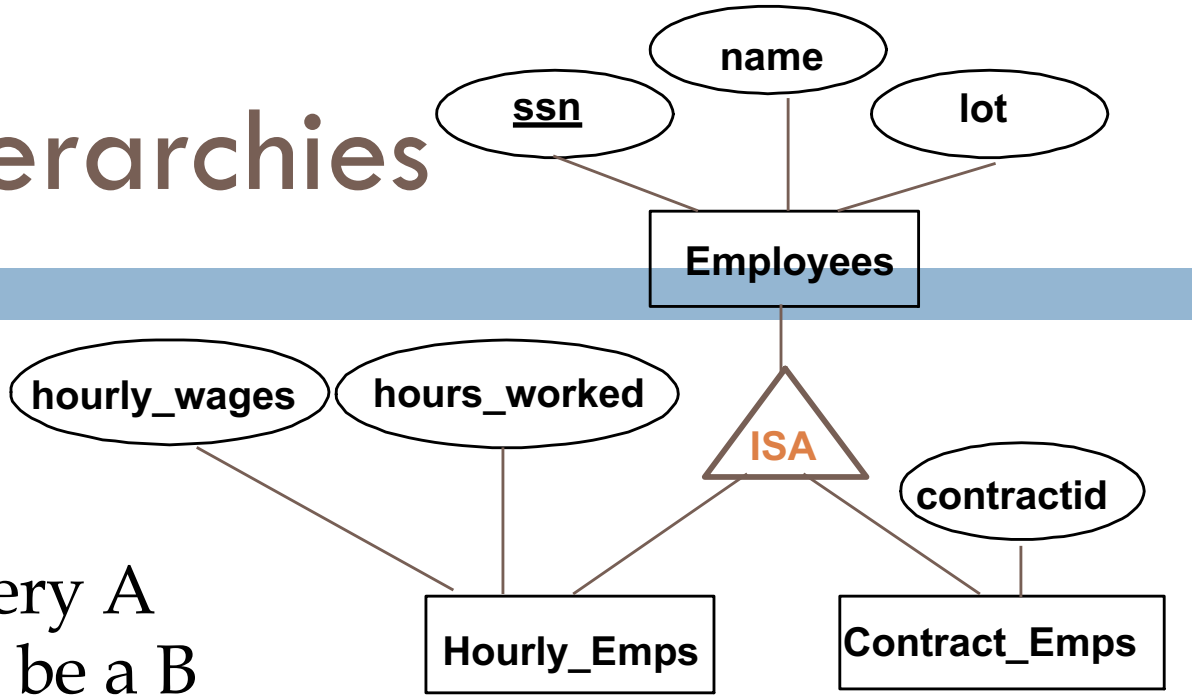
- Reasons for using isa:
  - To add descriptive attributes specific to a subclass.
  - To identify entities that participate in a relationship.

Example: Subclasses



Assume subclasses form a tree.

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.

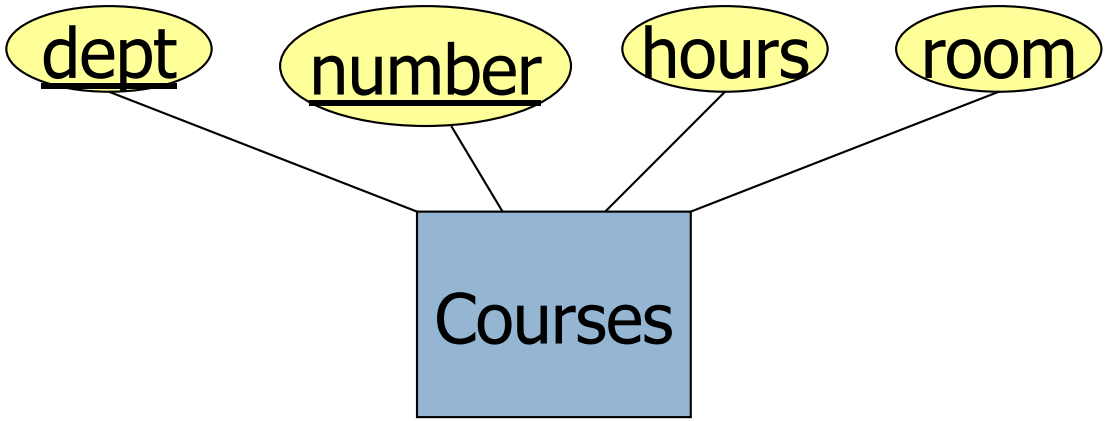
- *Overlap constraints*: Can two sub-classes contain the same entity? E.g., Can Joe be an Hourly_Emps as well as a Contract_Emps entity?

- *Covering constraints*: Does every Employees entity have to be an Hourly_Emps or a Contract_Emps entity?

Keys

- A *key* is a set of attributes for one entity set such that no two entities in this set agree on all the attributes of the key.
  - It is allowed for two entities to agree on some, but not all, of the key attributes.
- We must designate a key for every entity set.
- Underline the key attribute(s).

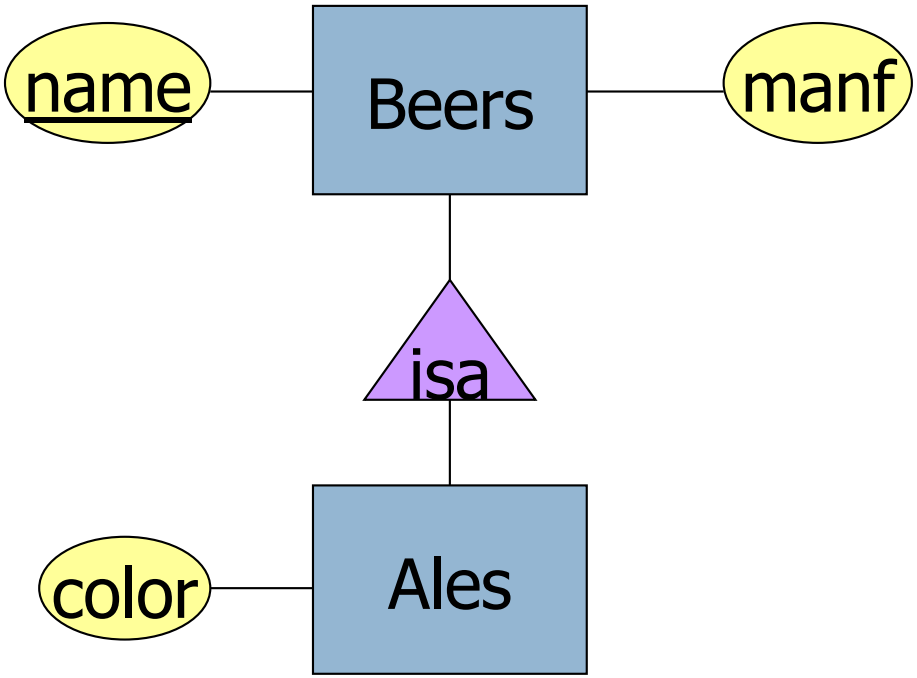
Example: a Multi-attribute Key



- Note that hours and room could also serve as a key, but we must select only one primary key.

Keys

In an Isa hierarchy, only the root entity set has a key, and it must serve as the key for all entities in the hierarchy.



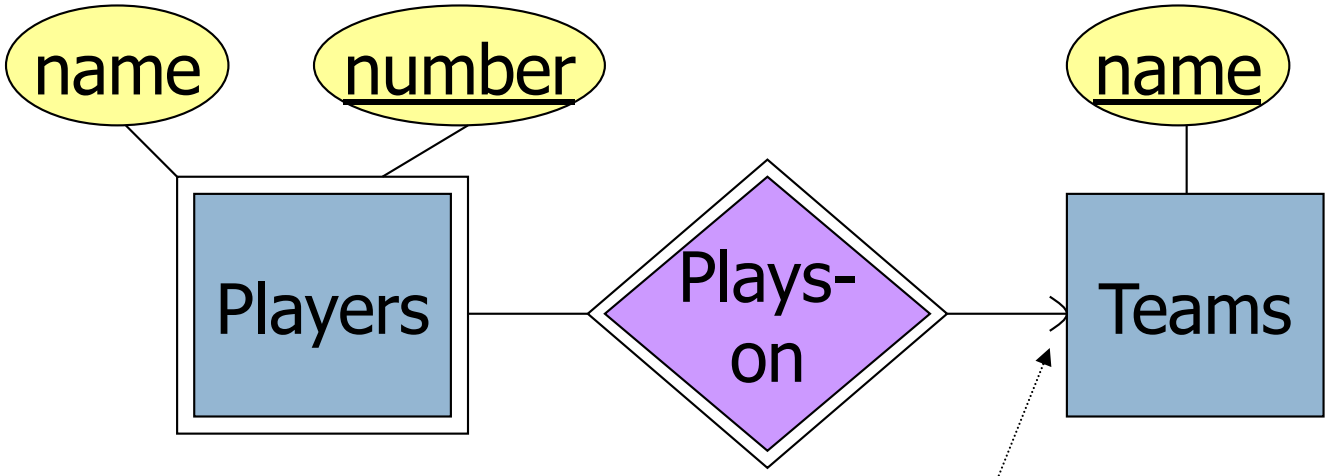
Weak Entity Sets

- Occasionally, entities of an entity set need "help" to identify them uniquely.
- Entity set *E* is said to be *weak* if in order to identify entities of *E* uniquely, we need to follow one or more many-one relationships from *E* and include the key of the related entities from the connected entity sets.

Example: Weak Entity Set

- name is almost a key for football players, but there might be two with the same name.
- number is certainly not a key, since players on two teams could have the same number.
- But number, together with the team name related to the player by Plays-on should be unique.

In E/R Diagrams



Note: must be rounded because each player needs a team to help with the key.

- Double diamond for *supporting* many-one relationship.
- Double rectangle for the weak entity set.