Chapter 4 High-level Database Models

- Entity/Relationship Models (E/R diagram)
- How to transfer to a relational model
- Unified Modeling Language (UML)
- Object Definition Language (ODL)
- Hot to Transfer them to a relational model

Introduction

- 现实世界：客观存在的世界。
- 信息世界：现实世界在人们头脑中的反映。
- 机器世界：信息世界的信息在机器世界中以数据的形式存放。

---

reality-》information world-》 machine world

E-R data model　　relational model√

UML　　object-relational model

ODL　　object-oriented model

Purpose of E/R Model

- The E/R model allows us to sketch database schema designs.
  - Includes some constraints, but not operations.
- Designs are pictures called *entity-relationship diagrams*.
- Later: convert E/R designs to relational DB designs.

Framework for E/R

- Design is a serious business.
- The “boss” knows they want a database, but they don’t know what they want in it.
- Sketching the key components is an efficient way to develop a working database.

Entity/Relationship Model

- Entity like objects, =things
- Entity set like class = set of similar Entity or objects
- Attribute=property of entities in an entity set, similar to fields of a struct.
- Relation=connect two or more entity set

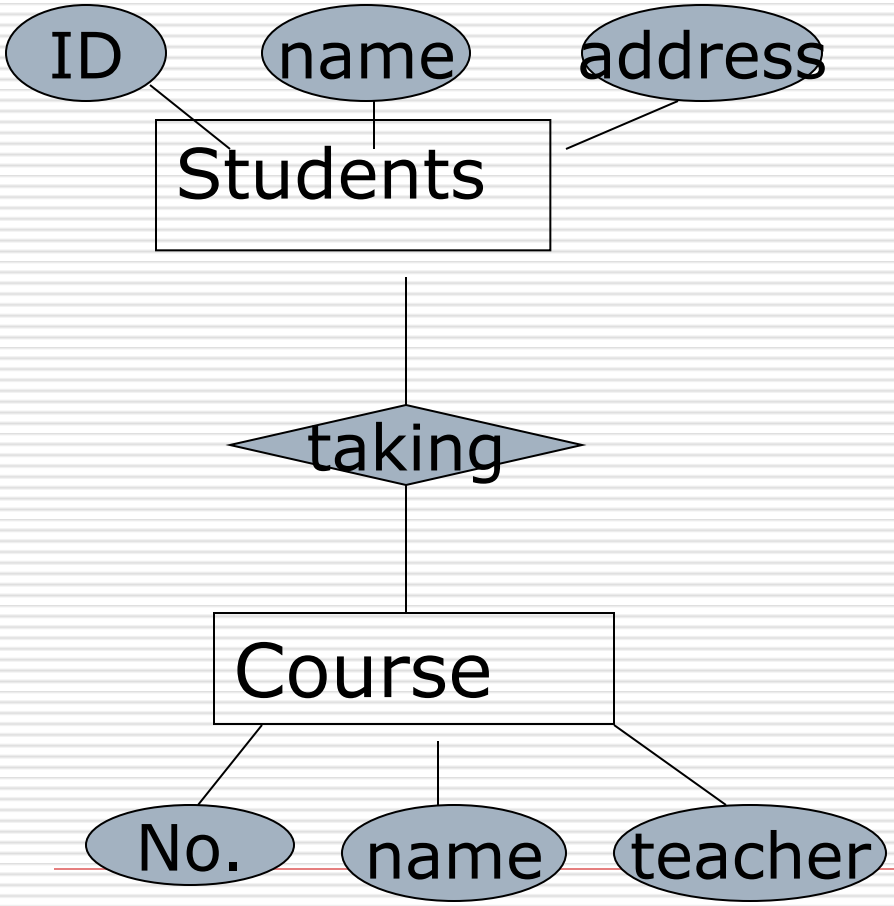
In diagrams,

entity set : rectangle;

attribute: oval,

relation: diamonds

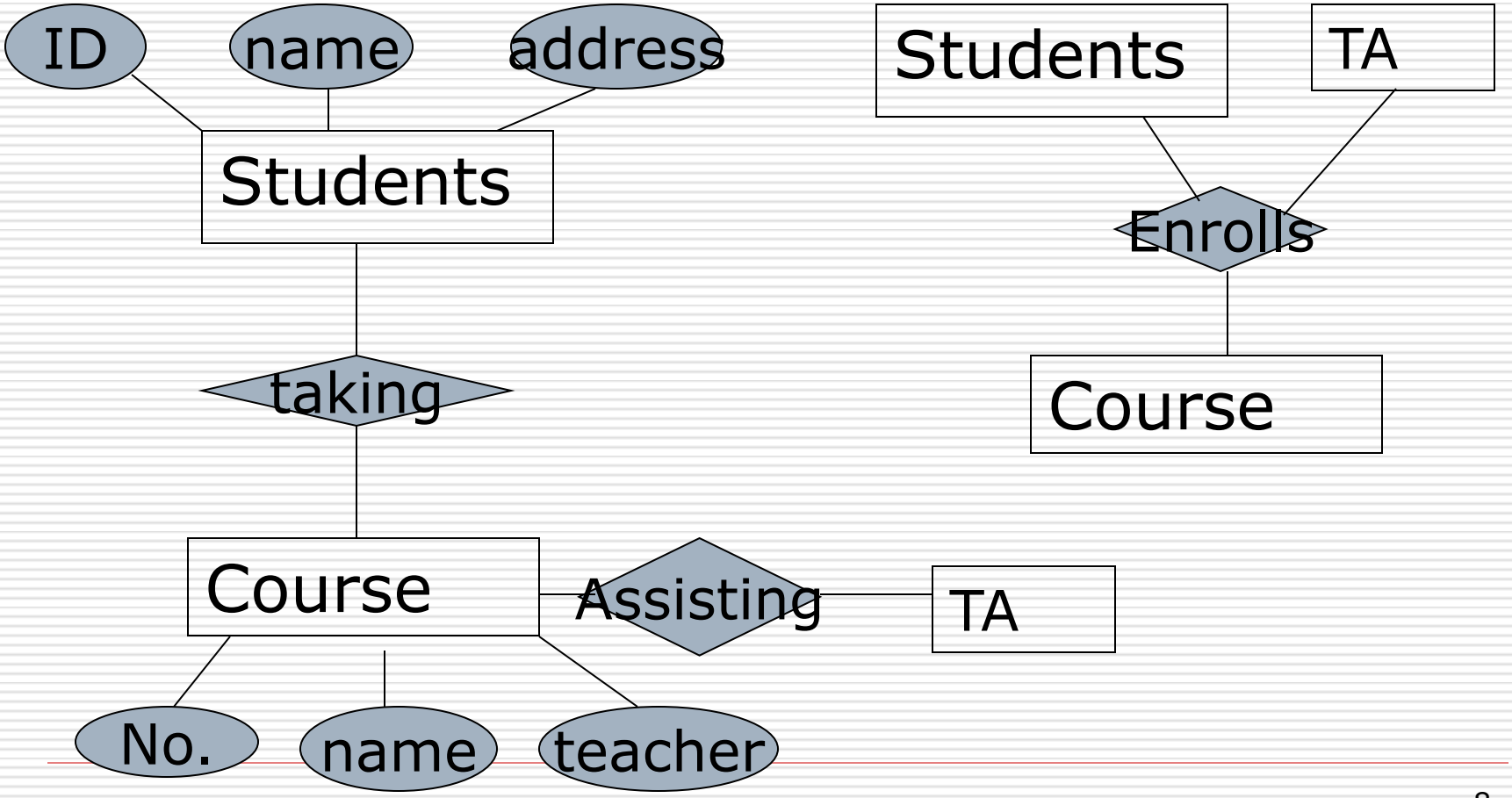
Entity/Relationship Diagrams: example



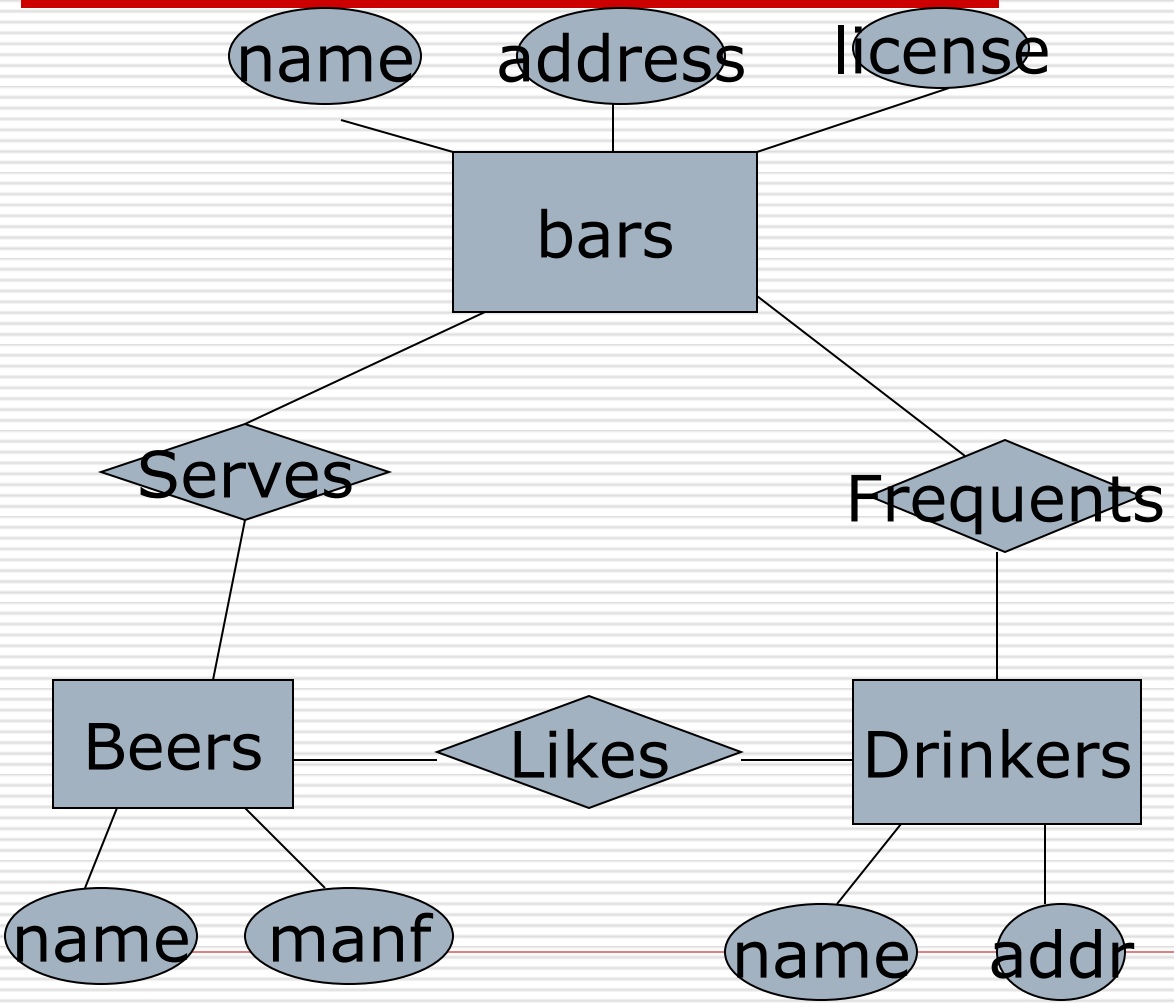
Relationships

- Binary (relation between two entity sets)
- Multiway (relation between more than two entity sets)
- Multiplicity of relationships: Express the number of entities to which another entity can be associated via a relationship set.

Binary & Multiway Relationships



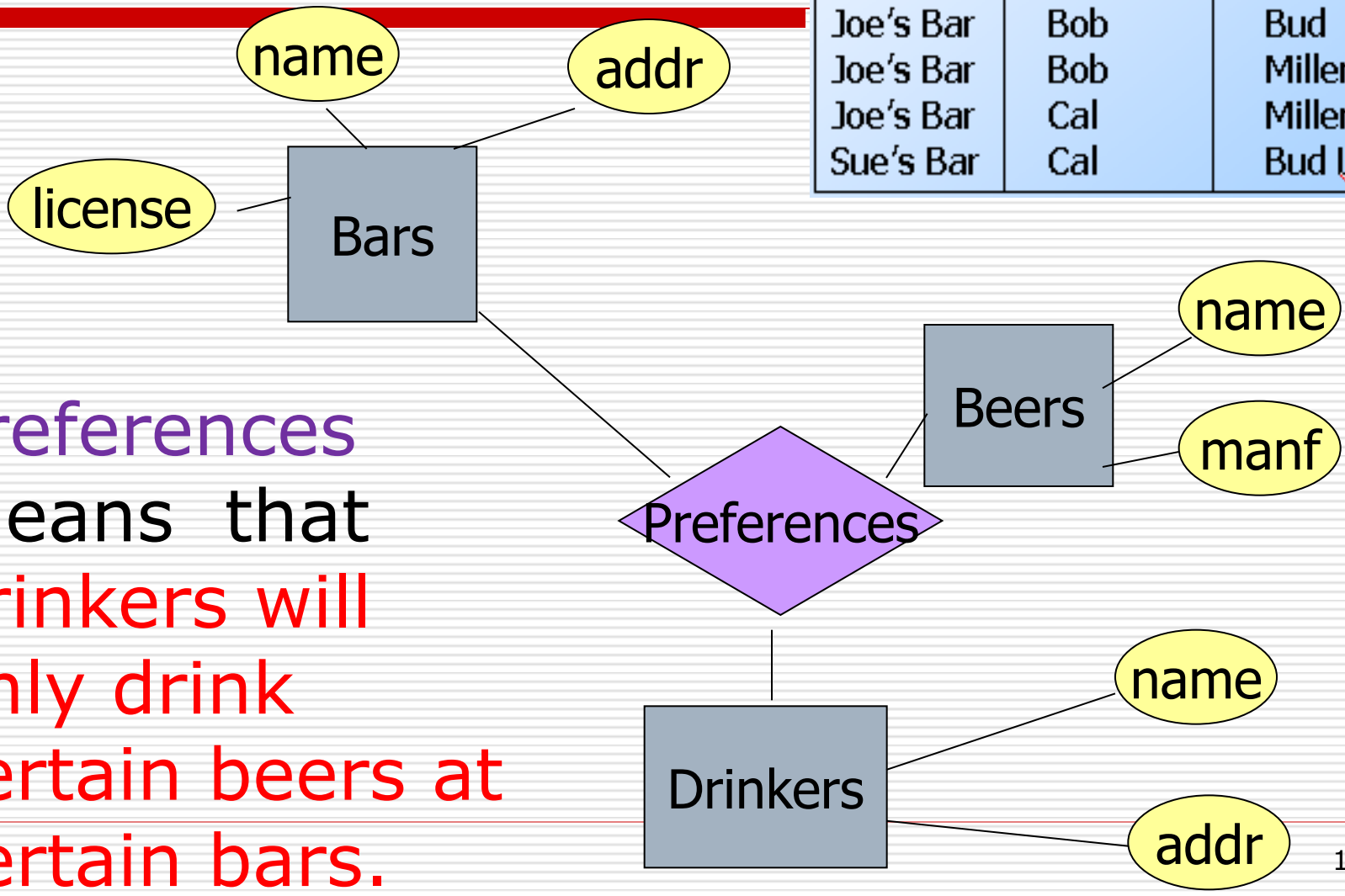
Beers-Bars-Drinkers Example



If John likes Bud in Joe's bar, he does not like the Bud in Marry's bar?

Example: 3-Way Relationship

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

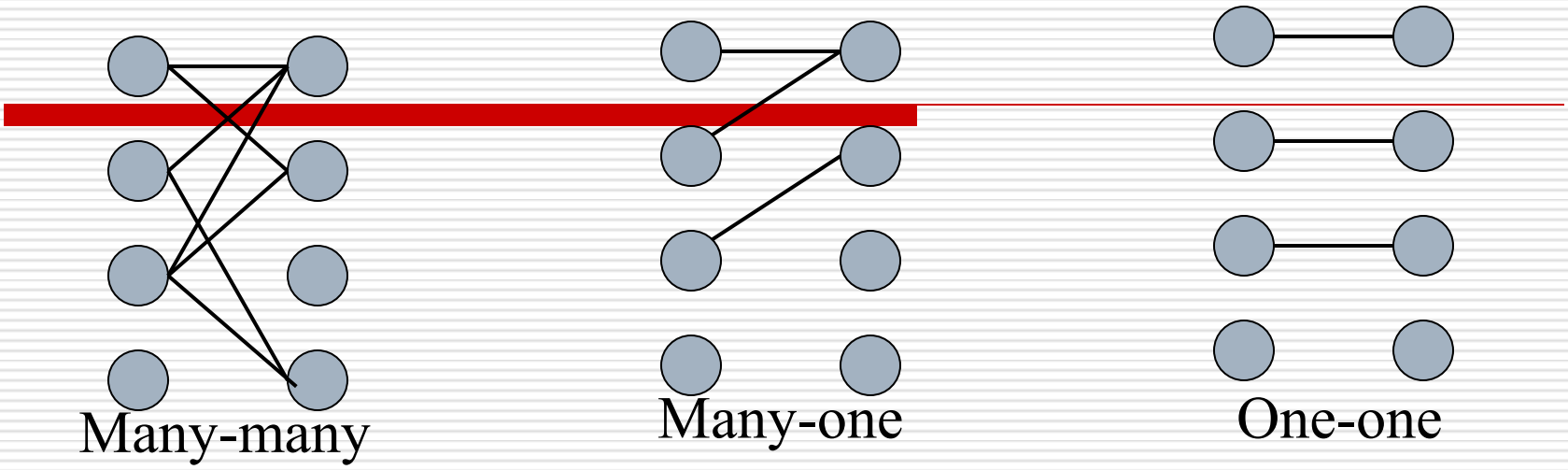


Preferences means that drinkers will only drink certain beers at certain bars.

Relationship Set

- The current "value" of an entity set is the set of entities that belong to it.
  - Example: the set of all bars in our database.
- The "value" of a relationship is a *relationship set*, a set of tuples with one component for each related entity set.

Multiplicity of Relationships

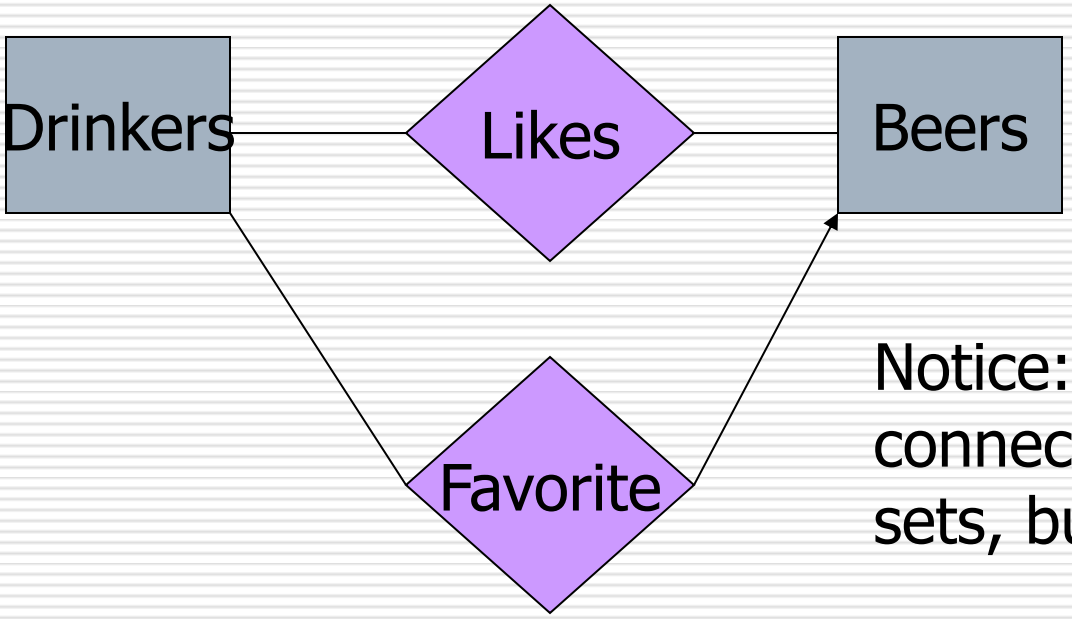


- In a *many-many* relationship, an entity of either set can be connected to many entities of the other set. E.g., a bar sells many beers; a beer is sold by many bars.
- In a many-one relationship, each entity of the first set is connected to at most one entity of the second set. But an entity of the second set can be connected to zero, one, or many entities of the first set.
- In a *one-one* relationship, each entity of either entity set is related to at most one entity of the other set.

Representing "Multiplicity"

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



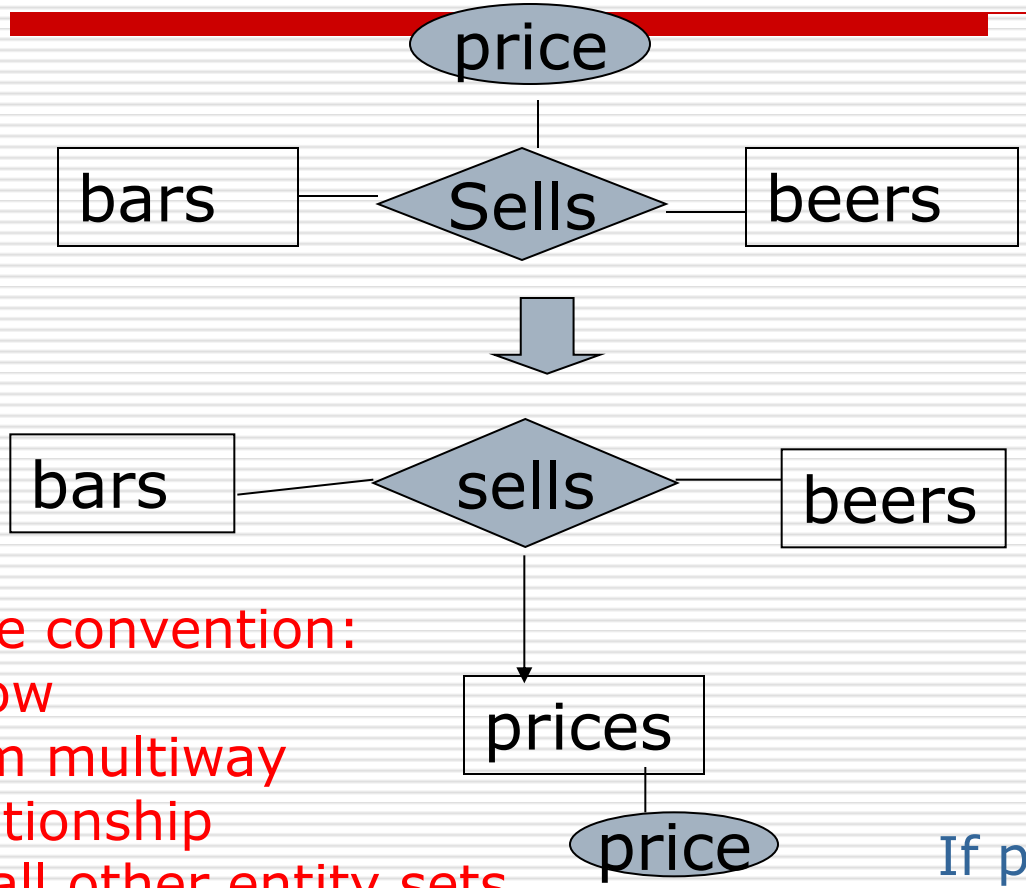
Notice: two relationships connect the same entity sets, but are different.

Example: One-One Relationship

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

A manufacturer has exactly one best seller.

Attributes on Relationships



Price depends jointly on bar and beer

- Create an entity set representing values of the attribute.
- Make that entity set participate in the relationship.

Note convention: arrow from multiway relationship = "all other entity sets together determine a unique one of these."

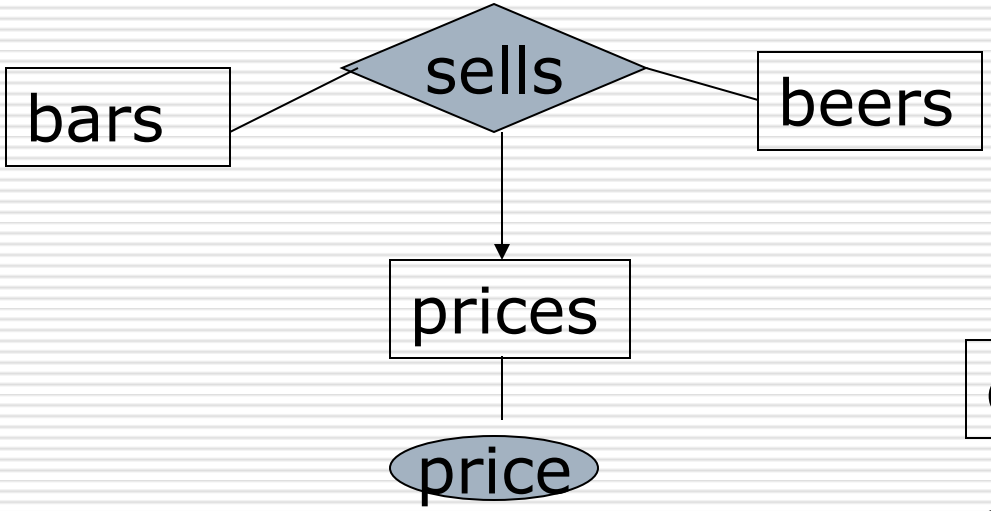
If price depends only on beers, what should we do?

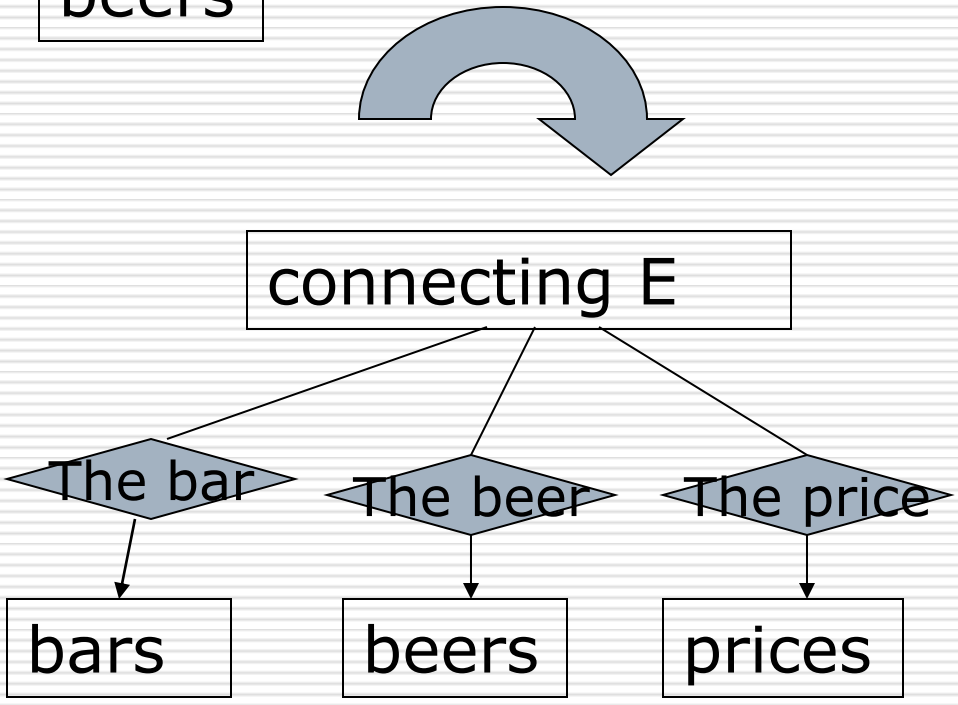
Converting Multiway to 2 way

- Creating a new connecting E.S. to represent the rows of a relationship set
- Many-one relationships from the connecting E.S. to the others

Converting Multiway to 2 way: example

Bars	Beers	Prices
Joe's bar	A. B.	5.
Joe's bar	Bud	4
Mary's bar	Bud	3
Mary's bar	A.B.	4





For each relationship (bar_i , $price_i$, $beer_i$) in Sells, create:

1. a new entity e_i in the entity set *E*
2. add (e_i , bar_i) to TheBar

3. add (e_i , $price_i$) to *ThePrice*

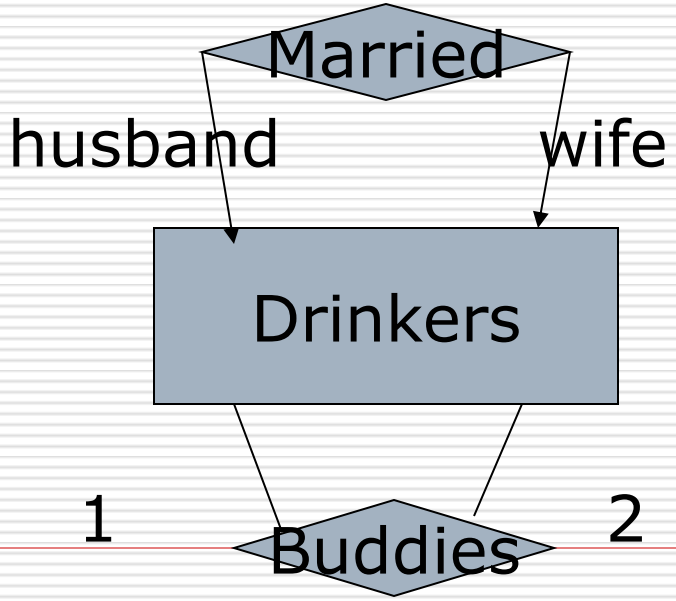
4. add (e_i , $beer_i$) to TheBeer

Roles

- Sometimes an E.S.participates more than once in a relationship.
- Label edges with roles to distinguish.

Relationship Set

Husband	Wife
Bob	Ann
Joe	Sue
…	…



Relationship Set

Buddy1	Buddy2
Bob	Ann
Joe	Sue
Ann	Bob
Joe	Moe
…	

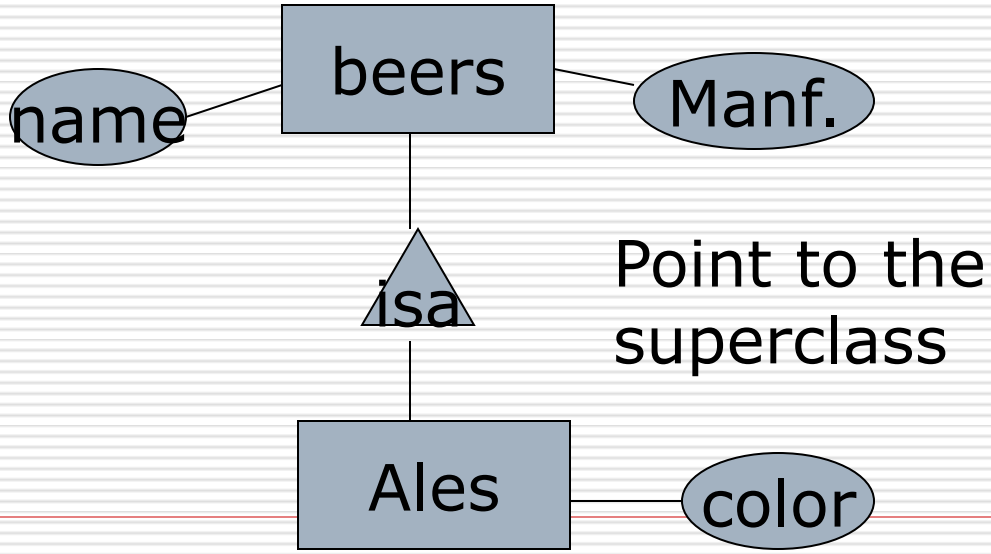
Subclasses

- Subclass = special case = fewer entities = more properties
- Example

Ales are a kind of beer. In addition to the properties (= attributes and relationships) of beers, there is a “color”attribute for ales.

E/R Subclasses

- Assume subclasses form a tree (no multiple inheritance)
- Isa triangles indicate the subclass relation.

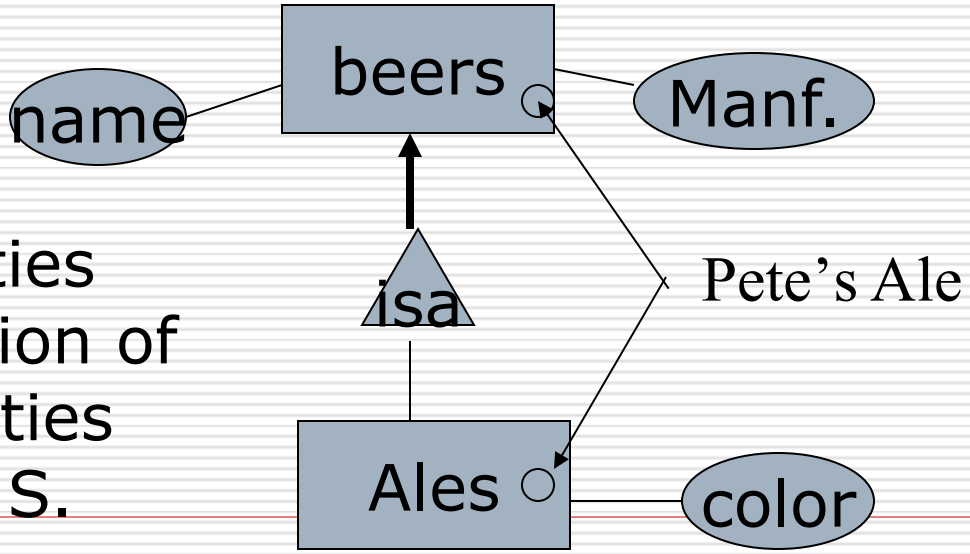


Different Subclass Viewpoints

- E/R viewpoint: E/R entities have *representatives* in all subclasses to which they belong
  - ✓ Rule: if entity *e* is represented in a subclass, then *e* is represented in the superclass. Its properties are the union of the properties of these E.S.
- Object-oriented viewpoint: An object (entity) belongs to exactly one class. It inherits properties of its superclasses.

Example, from E/R Viewpoint

- An entity has a component in each entity set to which it logically belongs.



Its properties are the union of the properties of these E.S.

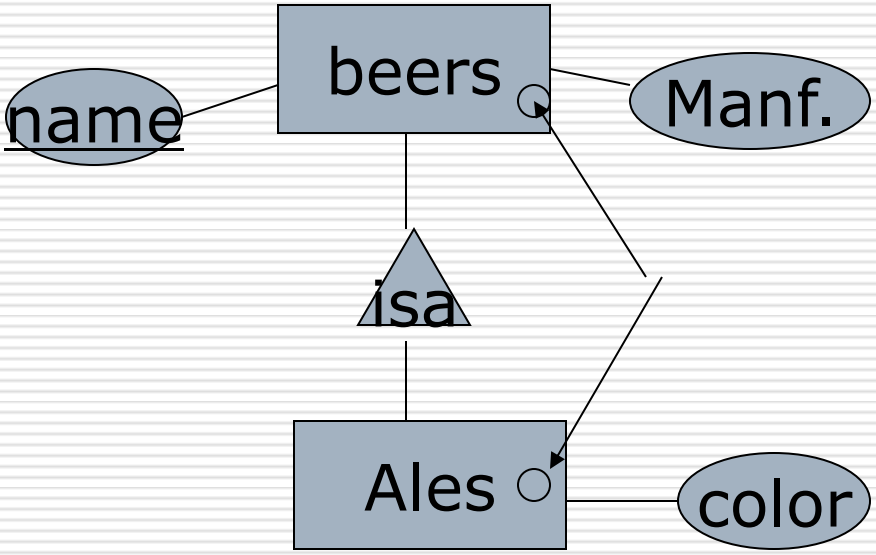
Keys

A key is a set of attributes such that no two entities agree on all these attributes.

- In E/R model, every E.S. must have a key. It could have more than one key, but one set of attributes is the “designated”key.
- In E/R diagrams, you should underline all attributes of the designated key.

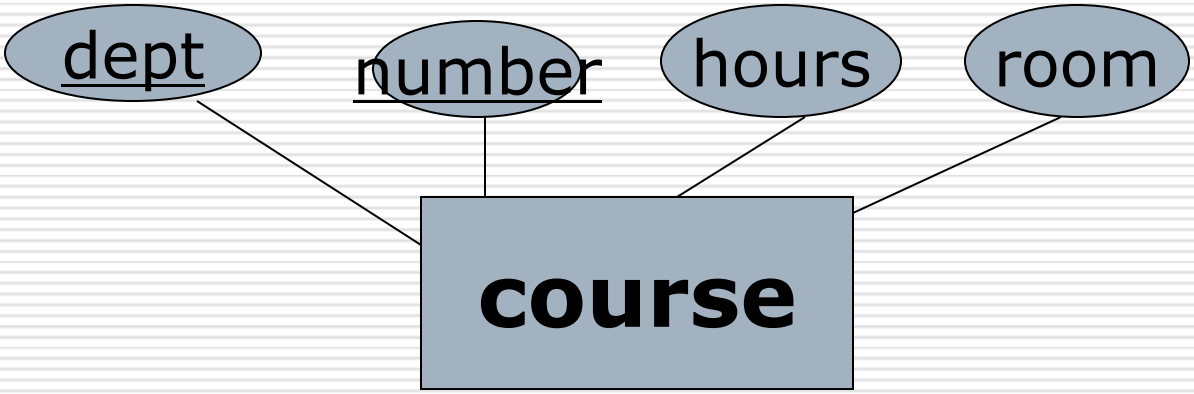
Example

Suppose name is key for Beers



Beer name is also key for Ales. In general, key at root is key for all.

Example: A Multiattribute Key



Possibly, hours+room also forms a key, but we have not designed it as such.

Weak Entity Sets

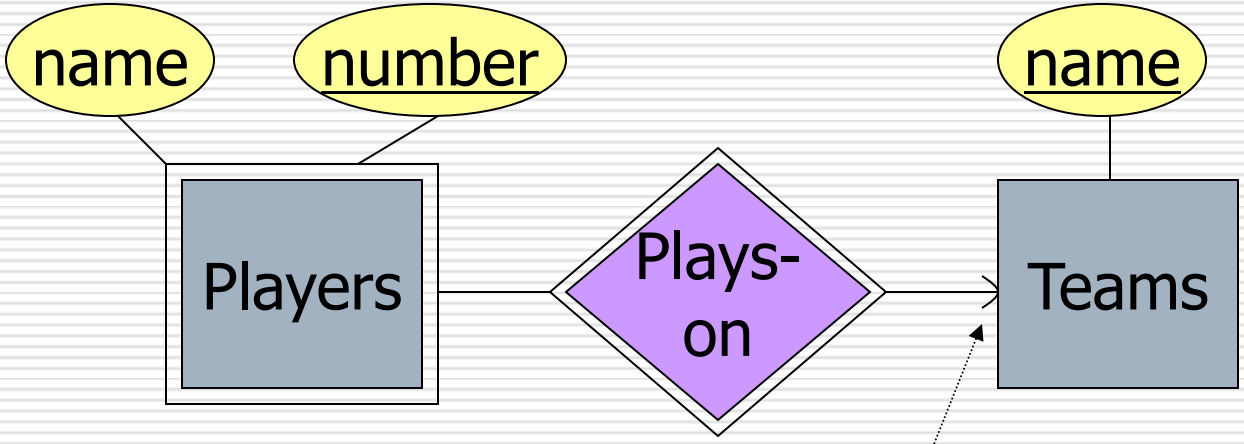
Sometimes an E.S. E's key comes not (completely) from its own attributes, but from the keys of one or more E.S's to which E is linked by a supporting many-one relationship.

- Called a **weak E.S**.
- Represented by putting double rectangle around E and a double diamond around each supporting relationship.
- Many-one-ness of supporting relationship (includes 1-1) essential. With many-many, we would not know which entity provided the key value.
- "Exactly one" also essential, or else we might not be able to extract key attributes by following the supporting relationship.

Example of Weak Entity Sets

- 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



(players) number and
(teams) name is a key
for Players

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.

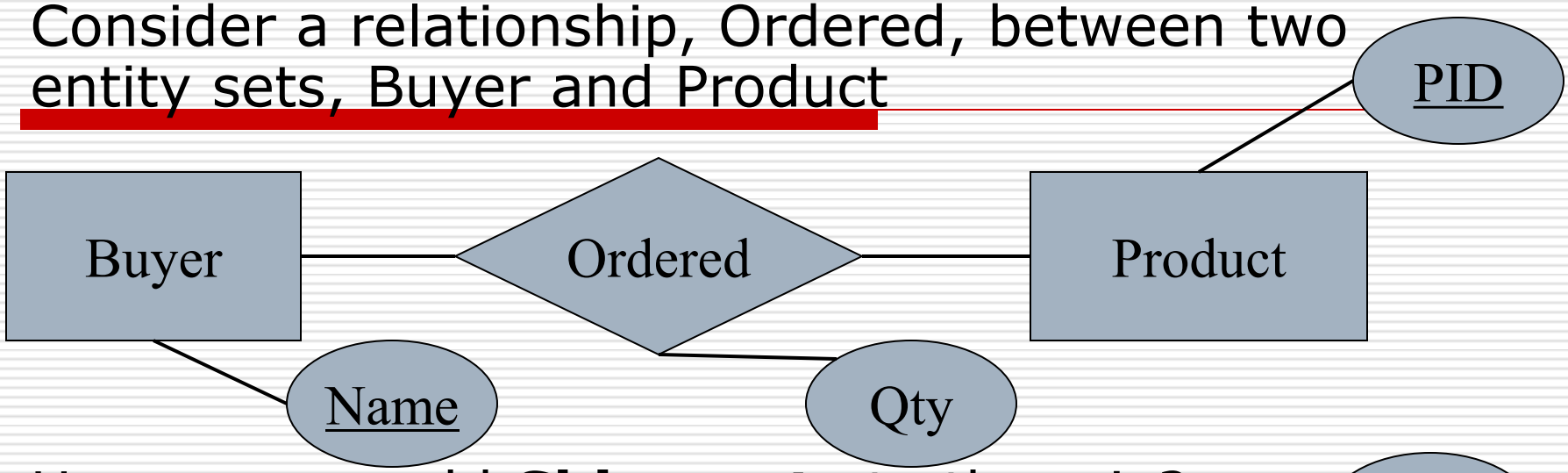
Summarization of weak entity set

Suppose E is a weak entity set, R is a supporting relationship, F is the another entity set.

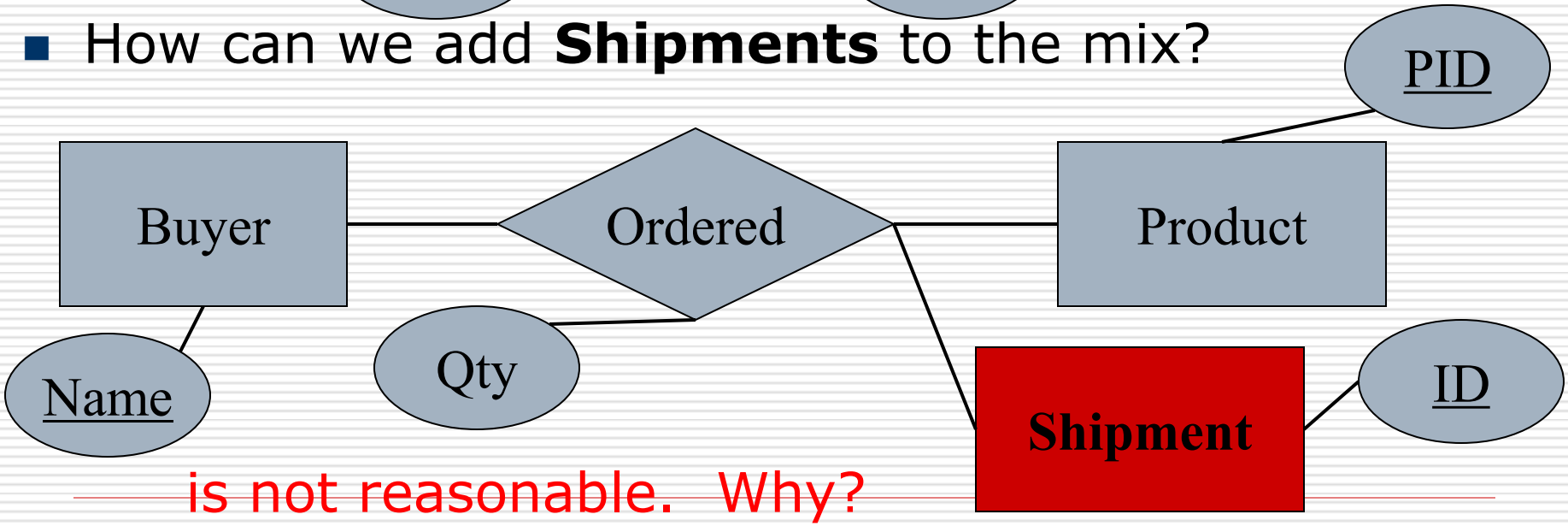
- The key of E consists of its own attributes and key attributes of F.
- R must be a many-one relationship.
- The attributes that F supplies for the key of E must be key attributes of F.
- Weakness can be chained.
- connecting entity set is a weak entity set.

Relationship To Weak Entities

- Consider a relationship, Ordered, between two entity sets, Buyer and Product



- How can we add **Shipments** to the mix?

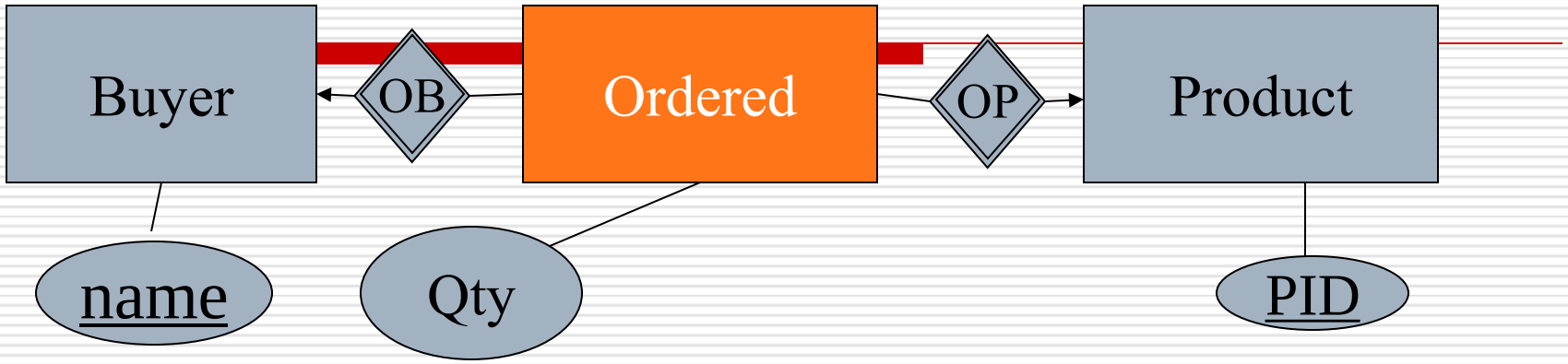


is not reasonable. Why?

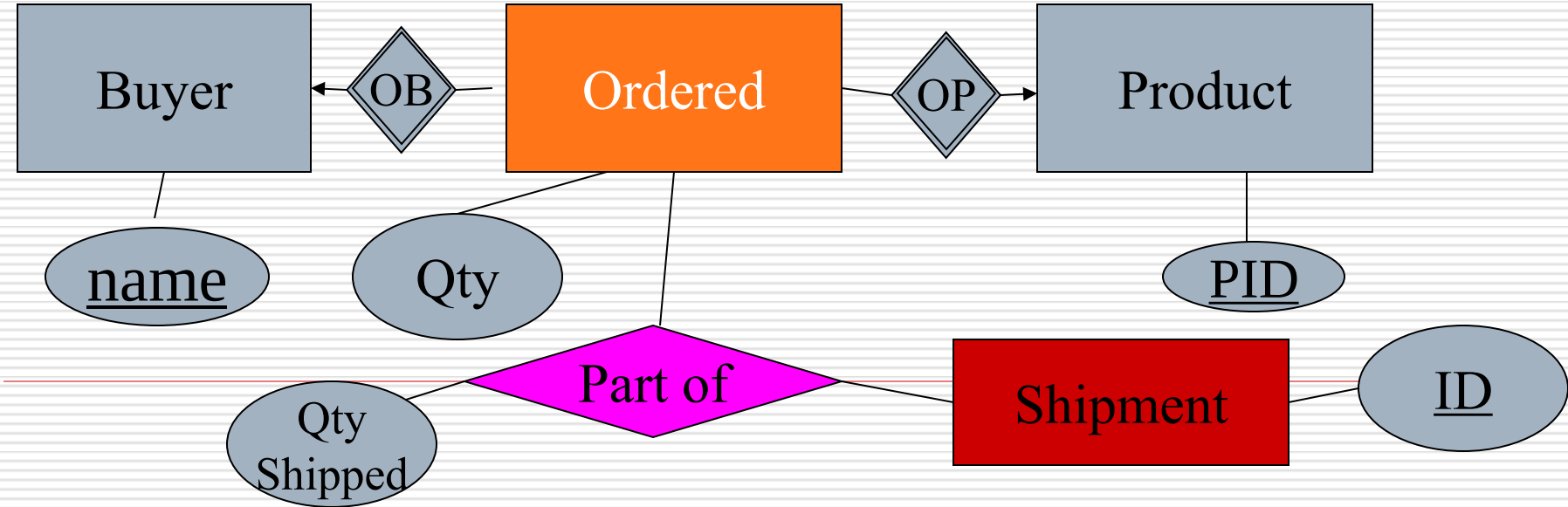
Classroom Discussion

- What is the solution to the problem?

•Solution: make Ordered into a weak entity set.



•And then add Shipment.



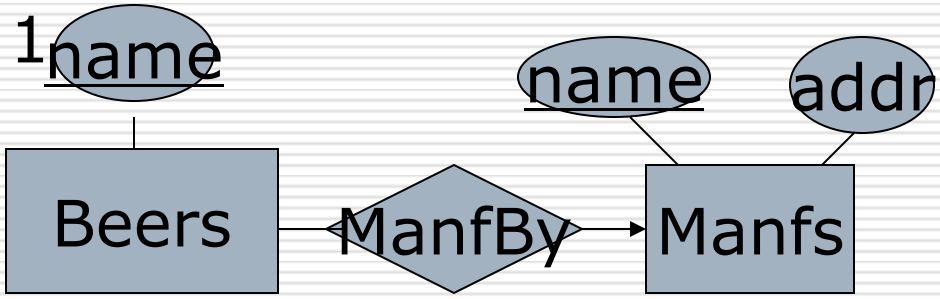
Design Techniques

1. Avoid redundancy.
2. Limit the use of weak entity sets.
3. Don’t use an entity set when an attribute will do.

Avoiding Redundancy

- *Redundancy* = saying the same thing in two (or more) different ways.
- Wastes space and (more importantly) encourages inconsistency.
  - Two representations of the same fact become inconsistent if we change one and forget to change the other.
  - Recall anomalies due to FD's.

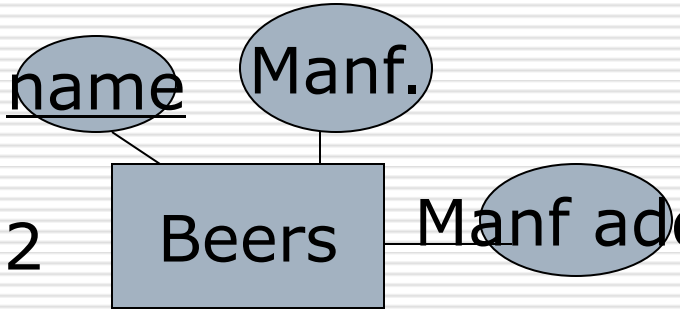
Example: Which one is better?

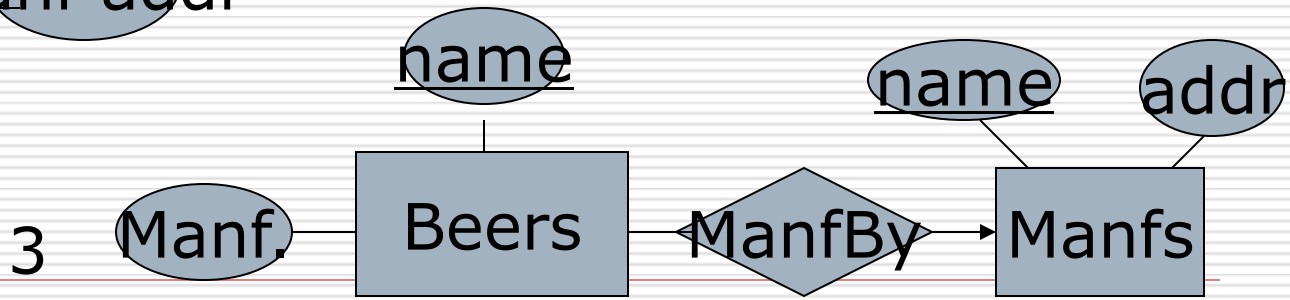


1:good

2:repeats manufacturer address for each beer they manufacture.

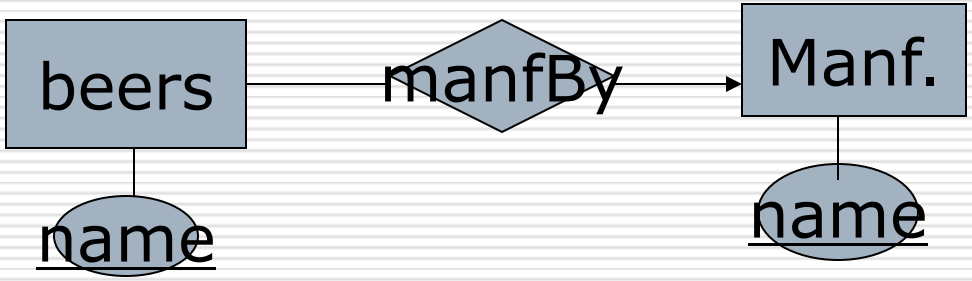
3:manufacturer's name said twice.



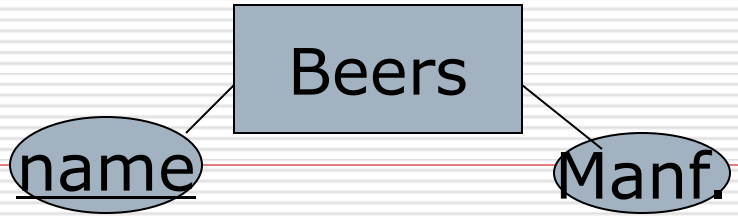


Entity Sets Vs. Attributes

You may be unsure which concepts are worthy of being entity sets, and which are handled more simple as attributes.



Wrong ??



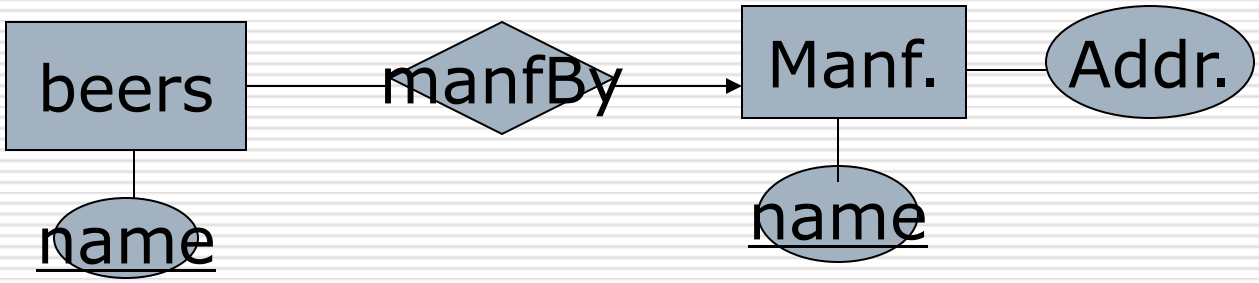
Right !!

Intuitive Rule for E.S.Vs. Attribute

Make an entity set only if it either:

1. Is more than a name of something; i.e., it has nonkey attributes or relationships with a number of different entity sets, or
2. Is the “many” in a many-one relationship.

Example



- Manf. Deserves to be an E.S. because we record addr, nonkey attribute.
- Beers deserves to be an E.S. because it is at the end of the "many" end.

Don’t overuse Weak E.S.

- There is a tendency to feel that no E.S. has its entities uniquely determined without following some relationships.
- However, in practice, we almost always create unique ID’s to compensate: social-security numbers, VIN’s, etc.

Don’t overuse Weak E.S.

- The only times weak E.S.’s seem necessary are when:

1. We can not easily create such ID’s; e.g., no one is going to accept a “species ID” as part of the standard nomenclature(术语) (species is a weak E.S supported by membership in a genus)
2. There is no global authority to create them, e.g., crews and studios.

Constraints in the E/R Model

- Key constraints
- Single-value constraints
- Referential integrity constraints
- Other constraints

Constraints are part of the model.

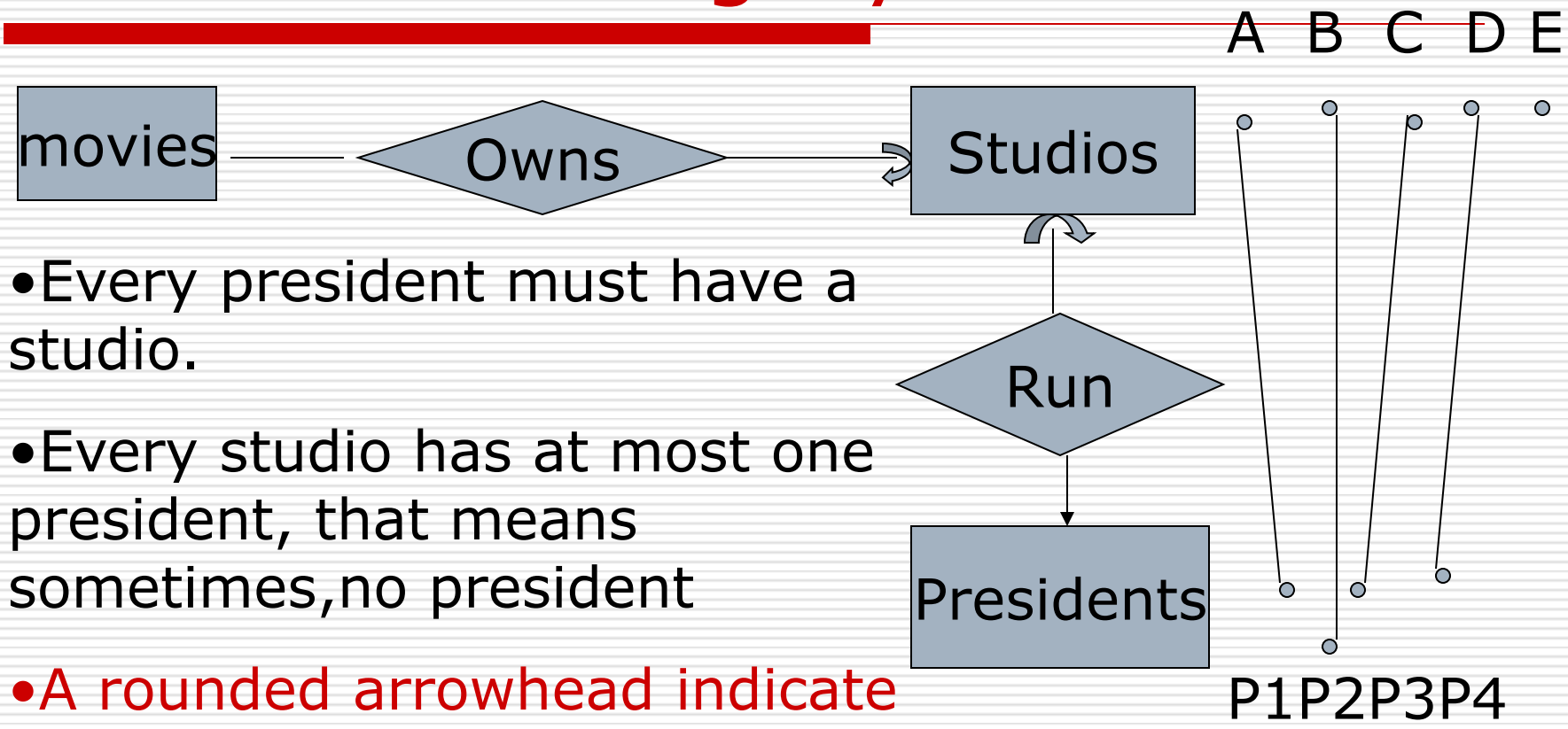
Key constraints

- No two entities may agree in their values for all of the attributes that constitute a key.
- A key may consist of more than one attribute.
- There can also be more than one possible key for an entity set.

Single-value constraints

- Many ways to express:
- Each attribute of an entity set has a single value. (not null)
- A relationship R that is many-one from entity set E to entity set F implies a single-value constraint. (at most one, or exactly one)

Referential integrity constraints

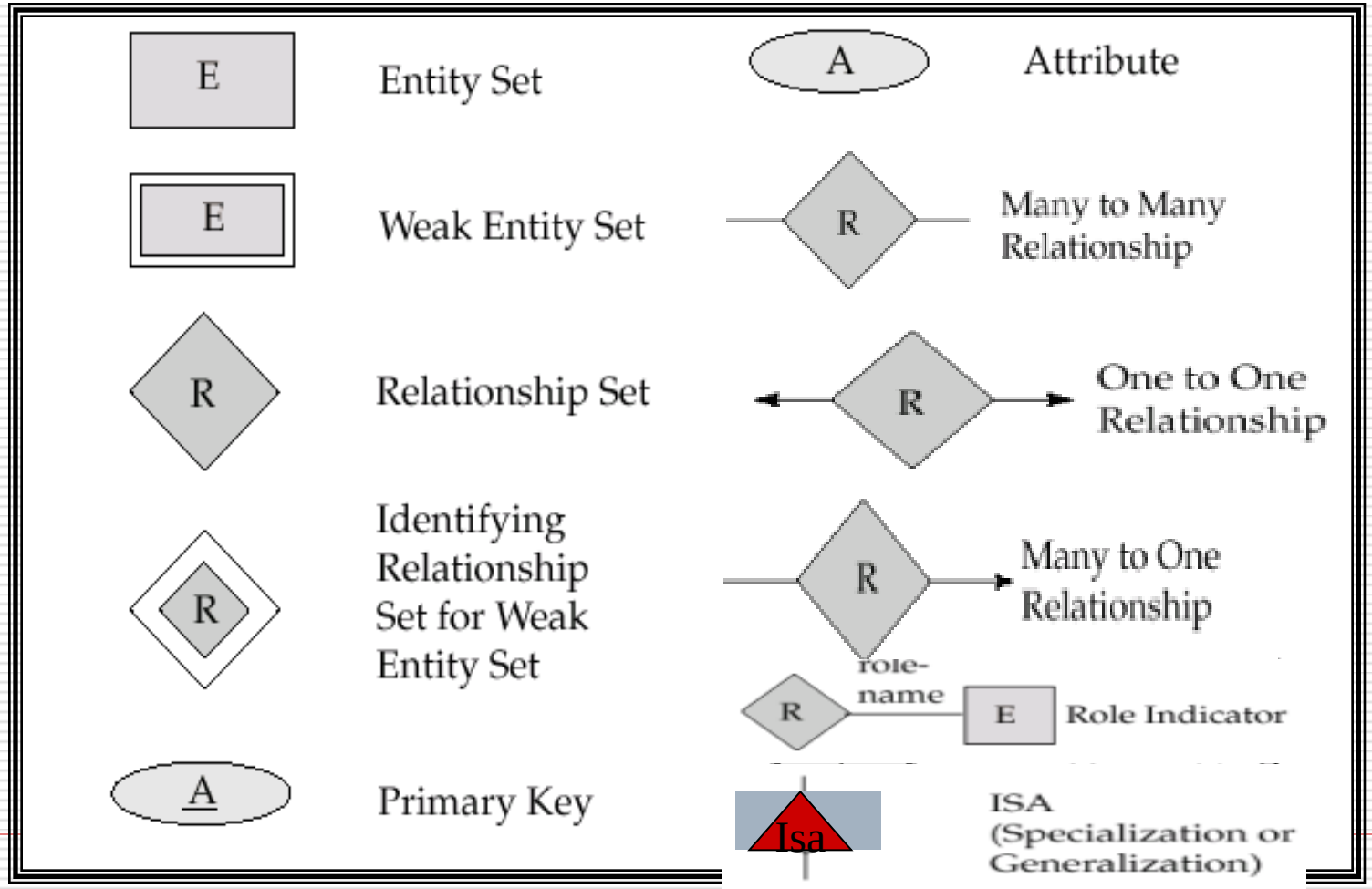


- Every president must have a studio.
- Every studio has at most one president, that means sometimes,no president
- A rounded arrowhead indicate also that every president runs a studio (A,B,C,D) that exists in the studios entity set.

Other constraints

- Domain constraints restrict the value of an attribute to be in a limited set.
- General constrains, such as placing a constraint on the degree of a relationship, number constraints and so on.

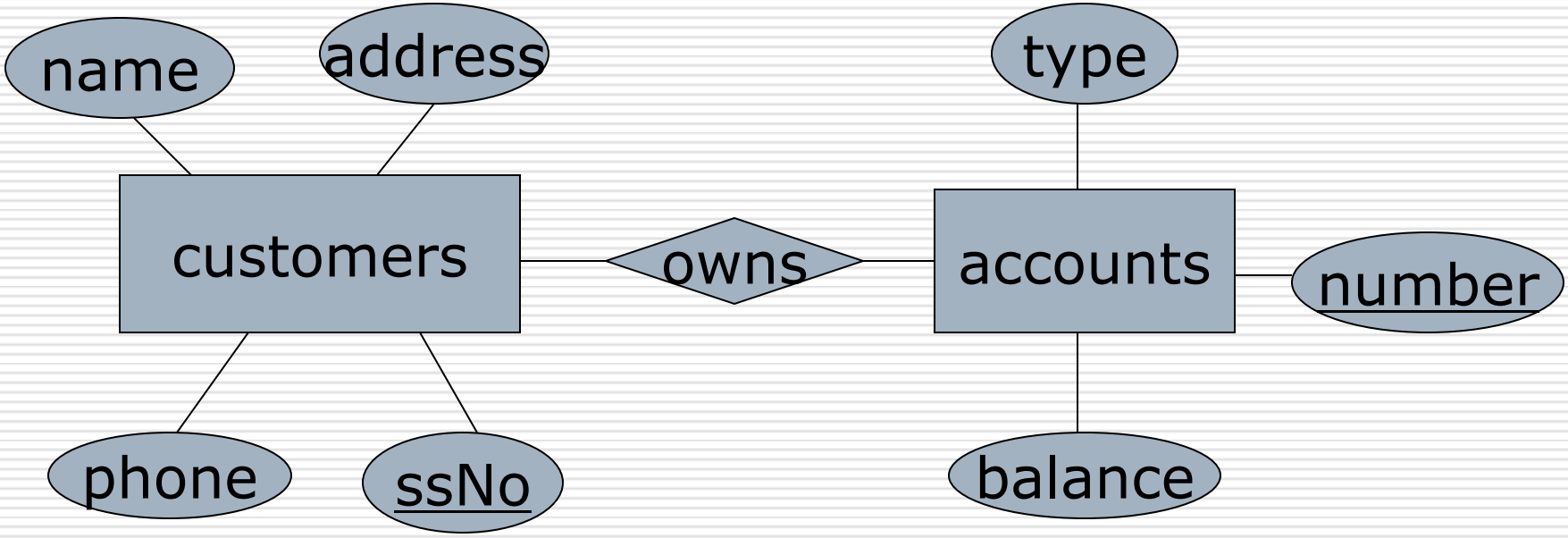
Summary of Symbols Used in E-R Notation



Exercises for E-R diagram

- Let us design a database for a bank, including information about customers and their accounts. Information about a custom includes their name, address, phone, and Social Security number. Accounts have numbers, types (e.g., savings,checking) and balances. We also need to record the customer who own an account. Be sure to include arrows where appropriate, to indicate the multiplicity of a relationship.

E-R Diagram

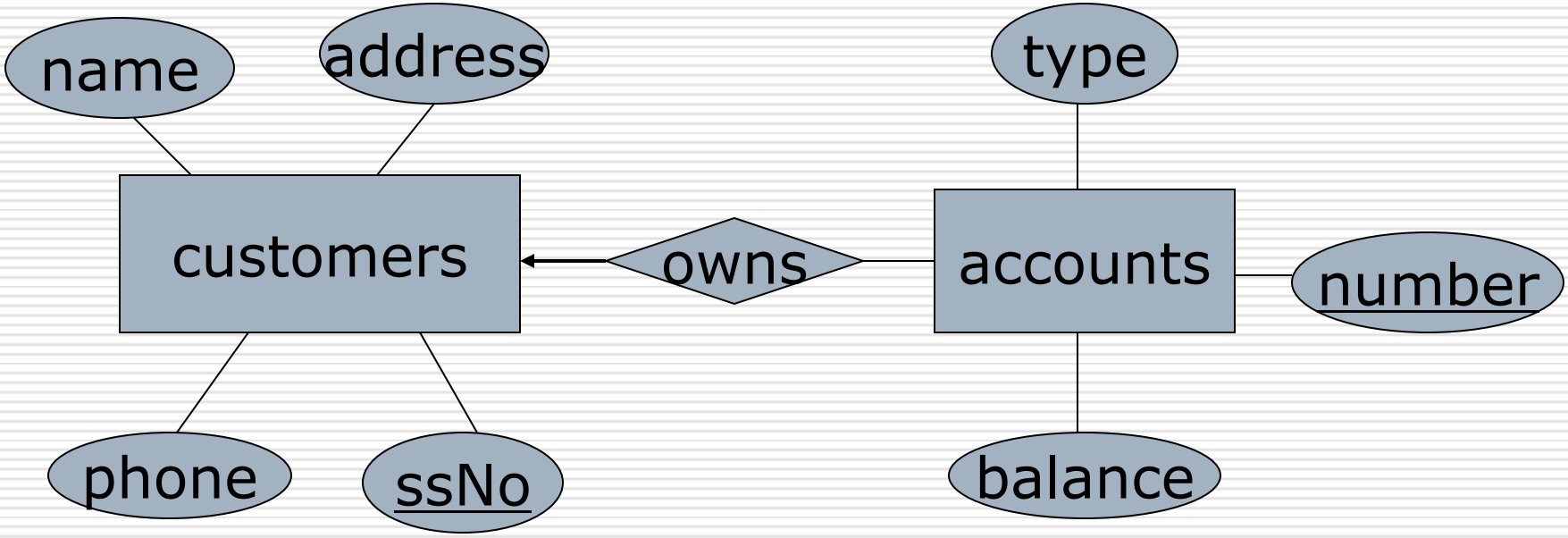

name
address
customers
phone
ssNo
owns
type
accounts
number
balance

Some modifications on the example

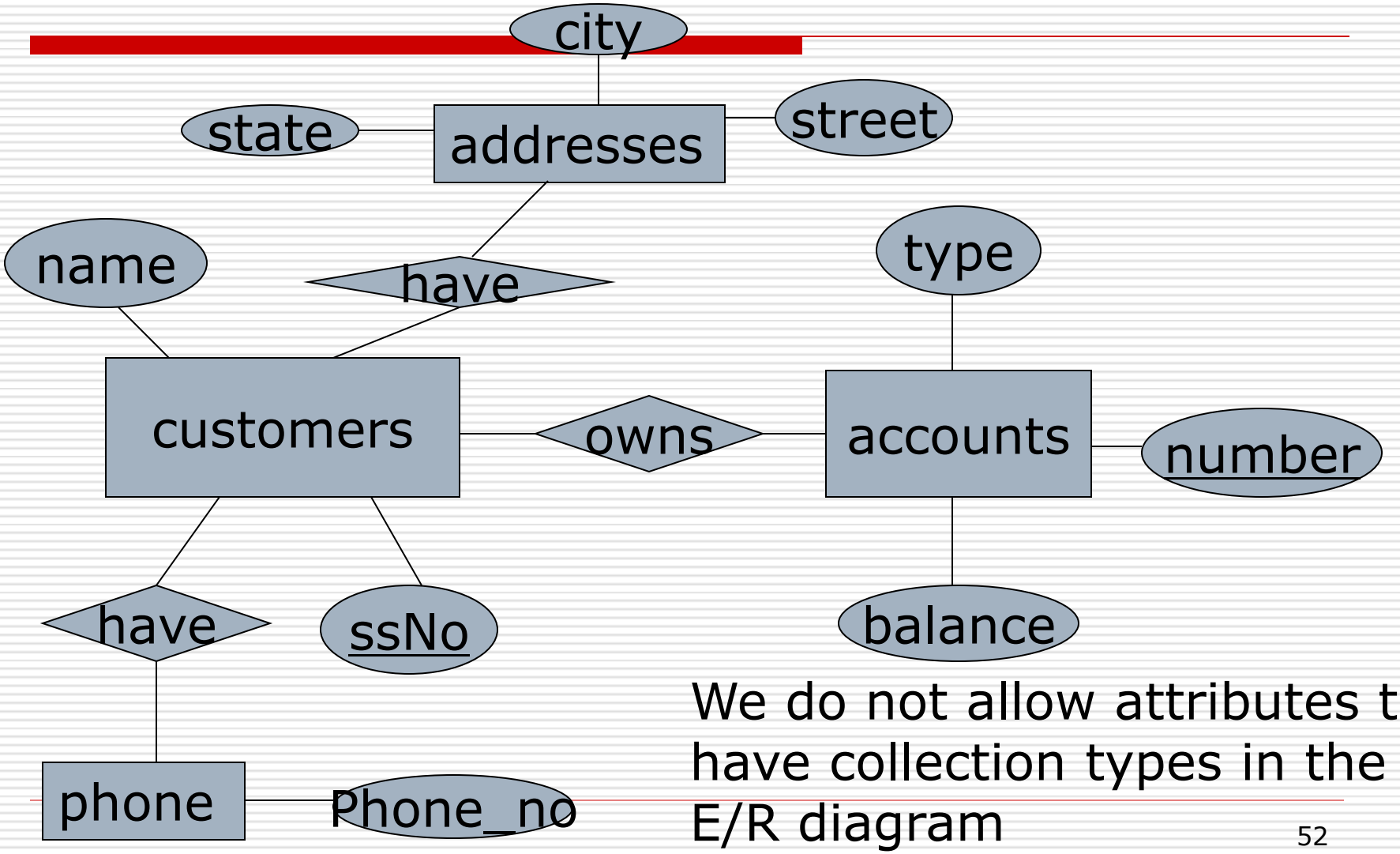
- Suppose an account can have only one customer.
- A customer can have a set of addresses (which is street-city-state triples) and a set of phones.

What are the ER diagram?

Suppose an account can have only one customer.



A customer can have a set of addresses (which is street-city-state triples) and a set of phones



We do not allow attributes to have collection types in the E/R diagram

Question?

- Modify the diagram so that customers can have a set of addresses, and at each address there is a set of phones.

Summary of E/R diagram

- Entity-Relationship Diagrams

Entities & Attributes & Relationships
Binary and Multiway of Relationships
Multiplicity, role of relationship
Weak Entity Sets, Supporting Relationship
Subclasses, key

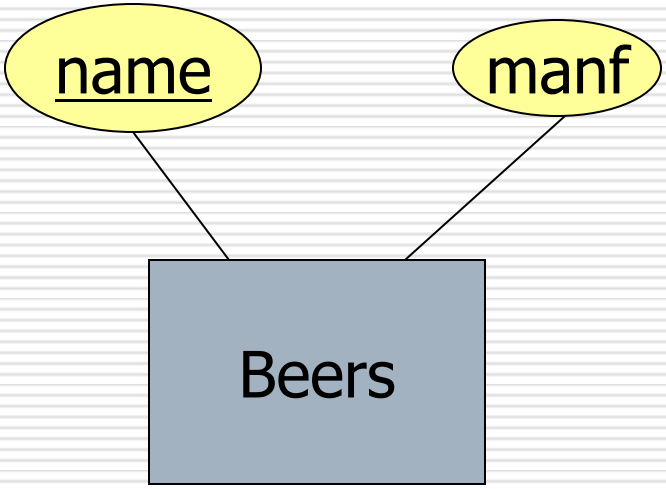
- Good Design

Faithfully represent
Avoid redundancy
Choose appropriate elements

From E/R Diagrams to Relations

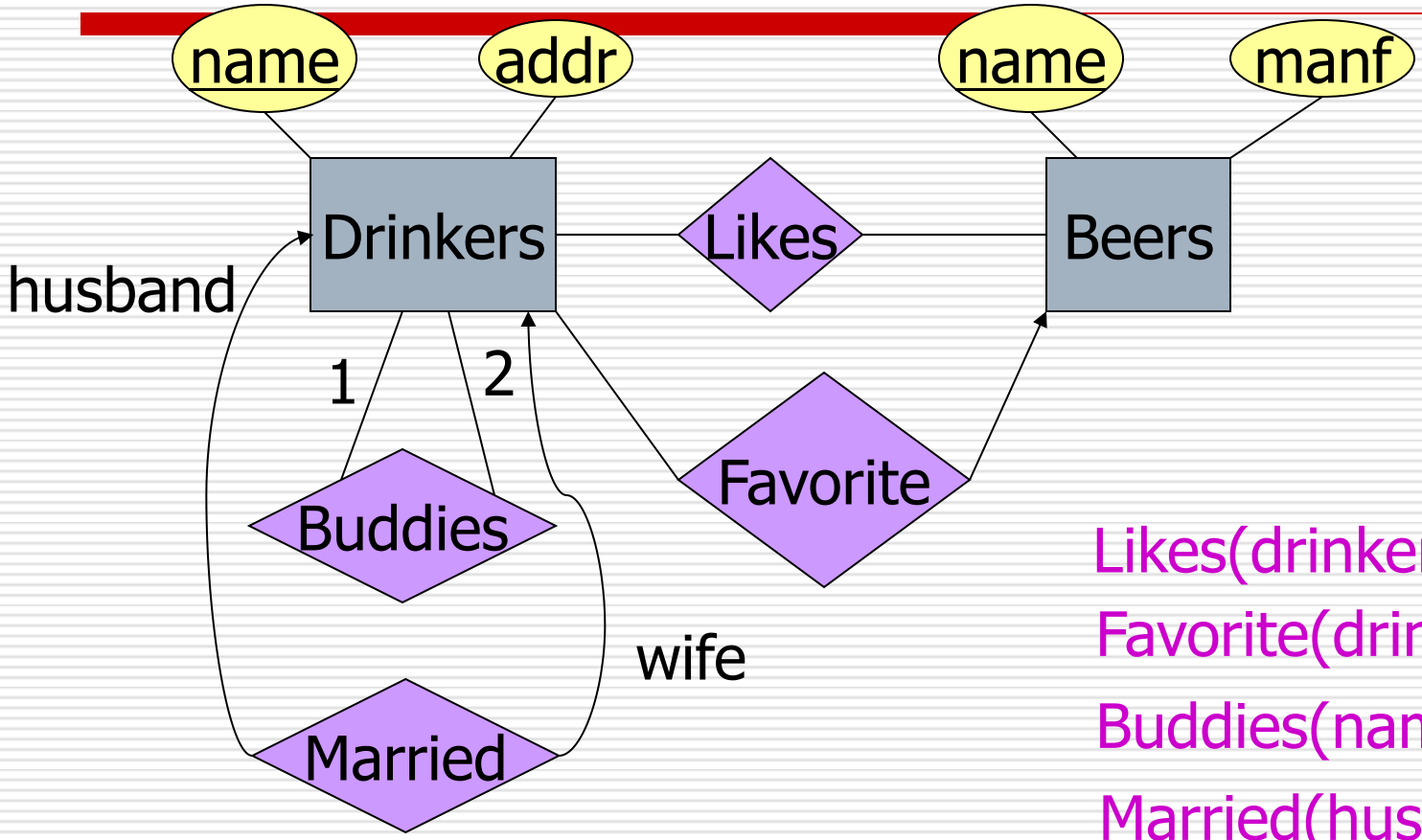
- Entity set -> relation.
  - Attributes -> attributes.
- Relationships -> relations whose attributes are only:
  - The keys of the connected entity sets.
  - Attributes of the relationship itself.

Entity Set -> Relation



Relation: Beers(name, manf)

Relationship -> Relation



Likes(drinker, beer)

Favorite(drinker, beer)

Buddies(name1, name2)

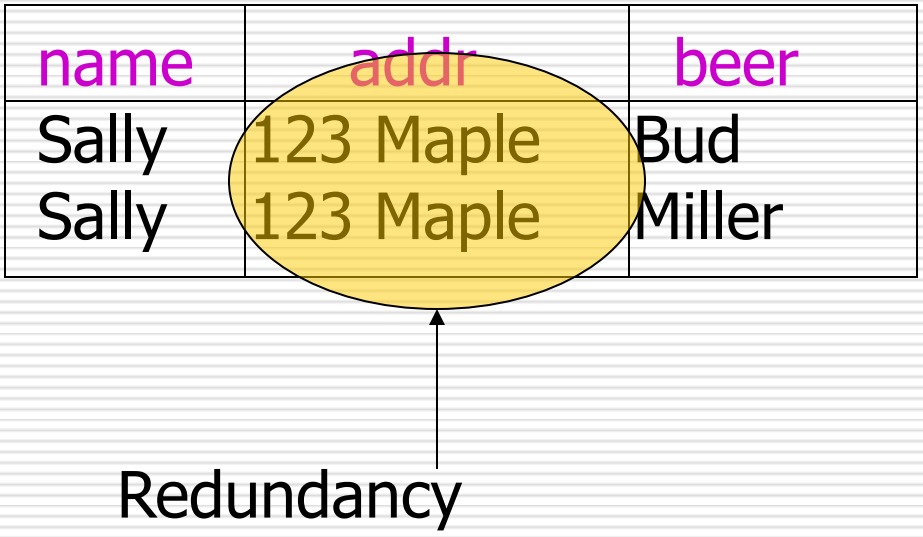
Married(husband, wife)

Combining Relations

- OK to combine into one relation:
  1. The relation for an entity-set *E*
  2. The relations for many-one relationships of which *E* is the "many."
- Example: Drinkers(name, addr) and Favorite(drinker, beer) combine to make Drinker1(name, addr, favBeer).

Risk with Many-Many Relationships

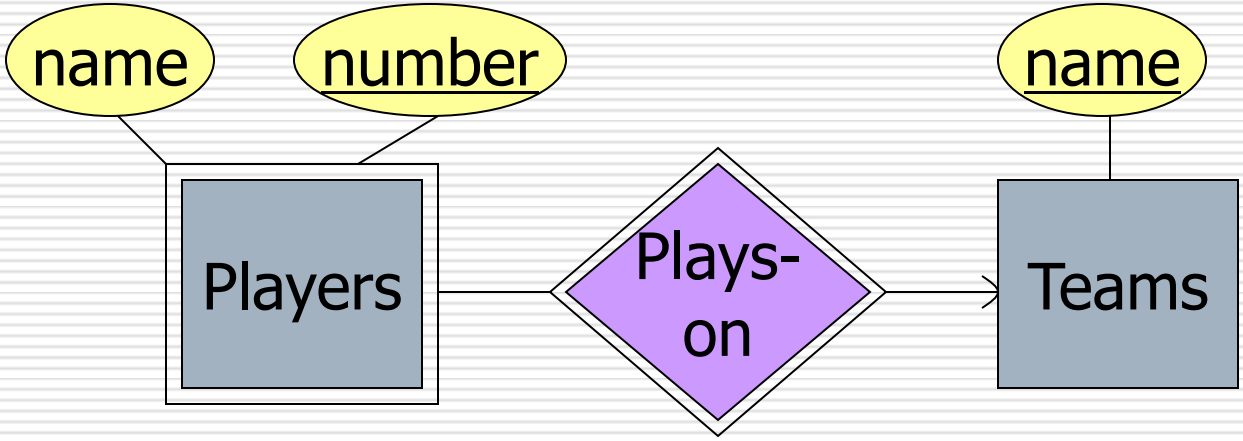
- Combining Drinkers with Likes would be a mistake. It leads to redundancy, as:



Handling Weak Entity Sets

- Relation for a weak entity set must include attributes for its complete key (including those belonging to other entity sets), as well as its own, nonkey attributes.
- A supporting relationship is redundant and yields no relation (unless *it* has attributes).

Example: Weak Entity Set -> Relation



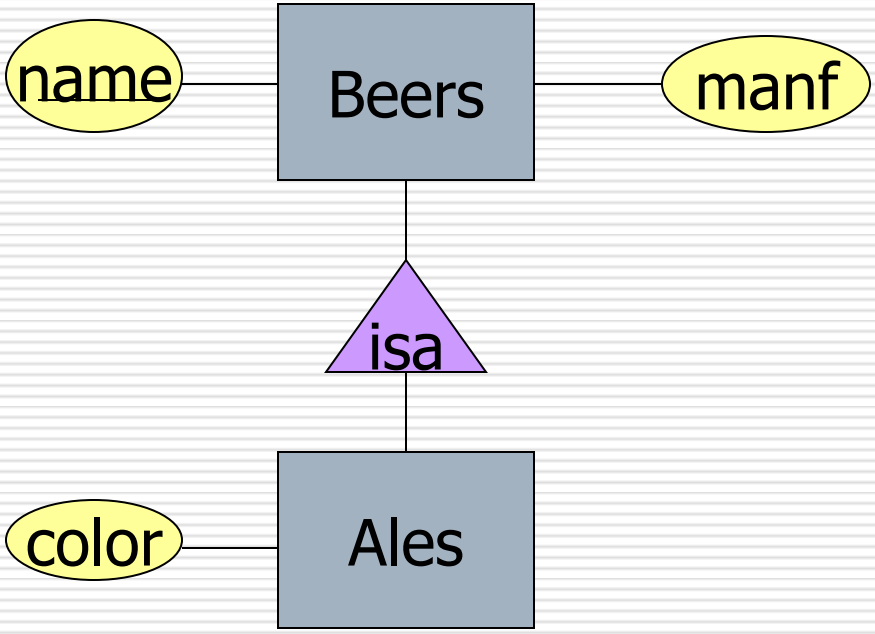
- Players(name,number,TeamName)
- Teams(name)
- ~~Plays-on(TeamName,number,name)~~

Plays-on becomes part of players

Subclasses: Three Approaches

1. *Object-oriented* : One relation per subset of subclasses, with all relevant attributes.
2. *Use nulls* : One relation; entities have NULL in attributes that don't belong to them.
3. *E/R style* : One relation for each subclass:
   - Key attribute(s).
   - Attributes of that subclass.

Example: Subclass -> Relations



Object-Oriented

name	manf
Bud	Anheuser-Busch

Beers

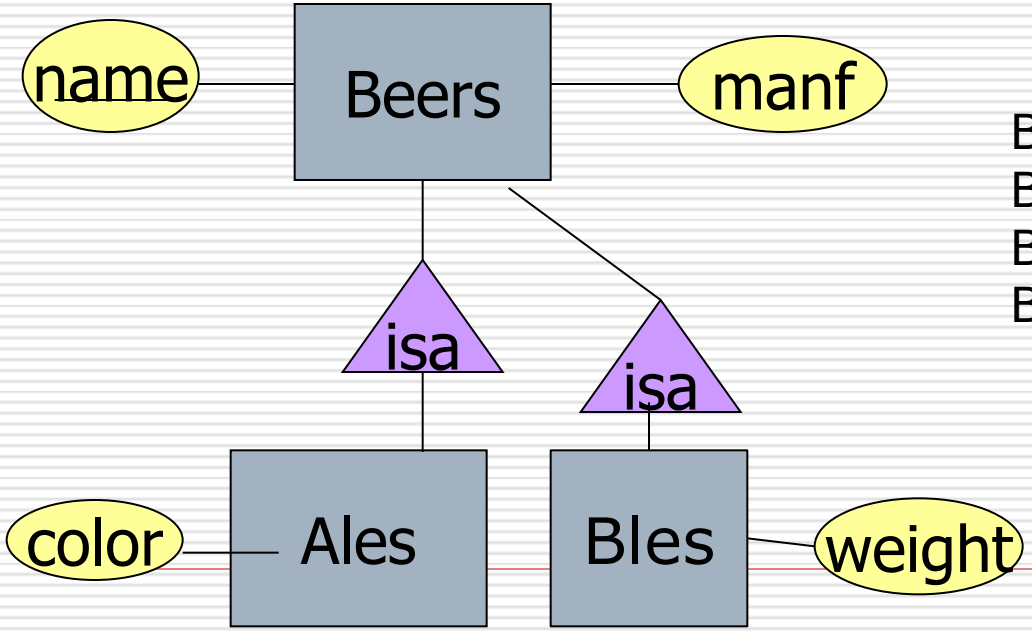
name	manf	color
Summerbrew	Pete's	dark

Ales

Good for queries like "find the color of ales made by Pete's."

Object-Oriented (cont.)

- Converting isa-hierarchies to relations is to enumerate all the possible subtrees of the hierarchy.



Beers(name,manf)
BeersA(name,manf,color)
BeersB(name,manf,weight)
BeersAB(name,manf,color,weight)

E/R Style

name	manf
Bud	Anheuser-Busch
Summerbrew	Pete's

Beers

name	color
Summerbrew	dark

Ales

Good for queries like "find all beers (including ales) made by Pete's."

Using Nulls

name	manf	color
Bud	Anheuser-Busch	NULL
Summerbrew	Pete's	dark

Beers

Saves space unless there are *lots* of attributes that are usually NULL.

Homework

- Read section 4.6.1 and 4.6.2