What is a Database Management System?

1. Manages very large amounts of data.
2. Supports efficient access to very large amounts of data.
3. Supports concurrent access to v.l.a.d.
4. Supports secure, atomic access to v.l.a.d.

Relational Model

- Based on tables, as:

acct#	name	balance
12345	Sally	1000.21
34567	Sue	285.48
...	...	...

- Today used in *most* DBMS's.

Three Aspects to Studying DBMS's

1. Modeling and design of databases.

   ❖ Allows exploration of issues before committing to an implementation.

2. Programming: queries and DB operations like update.

   ❖ SQL = "intergalactic dataspeak."

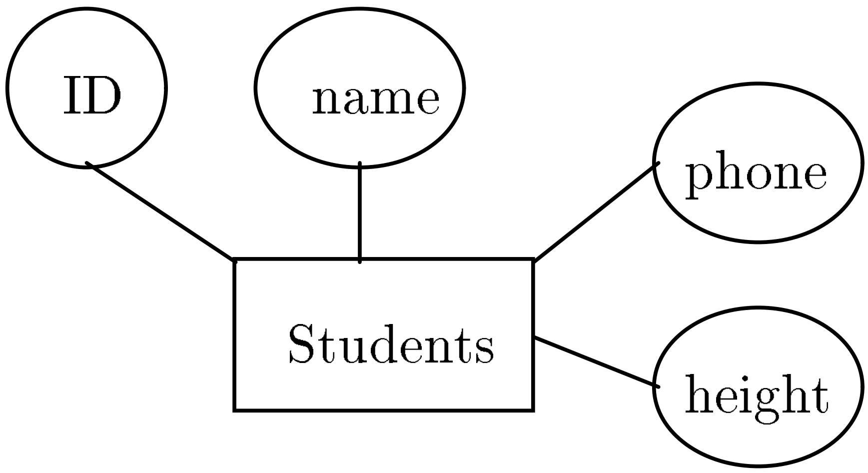
3. DBMS implementation.

CS145 = (1) + (2), while (3) is covered in CS245, CS346, CS347.

Entity/Relationship Model

Diagrams to represent designs.

- *Entity* like object, = "thing."
- *Entity set* like class = set of "similar" entities/objects.
- *Attribute* = property of entities in an entity set, similar to fields of a struct.
- In diagrams, entity set → rectangle; attribute → oval.



Relationships

- Connect two or more entity sets.
- Represented by diamonds.

Relationship Set

Think of the "value" of a relationship set as a table.

- One column for each of the connected entity sets.
- One row for each list of entities, one from each set, that are connected by the relationship.

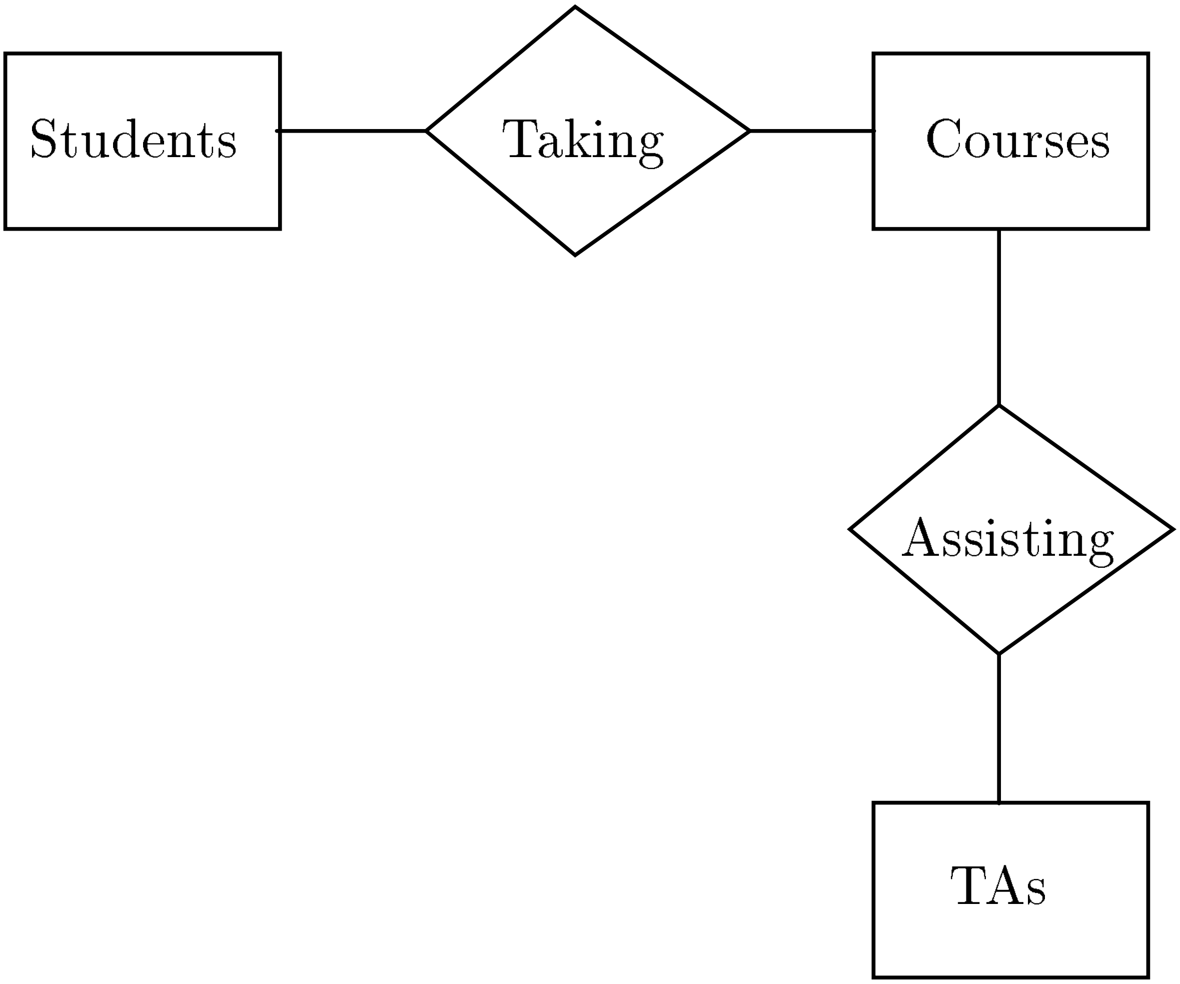
Students	Courses
Sally	CS145
Sally	CS244
Joe	CS145
...	...

Multiway Relationships

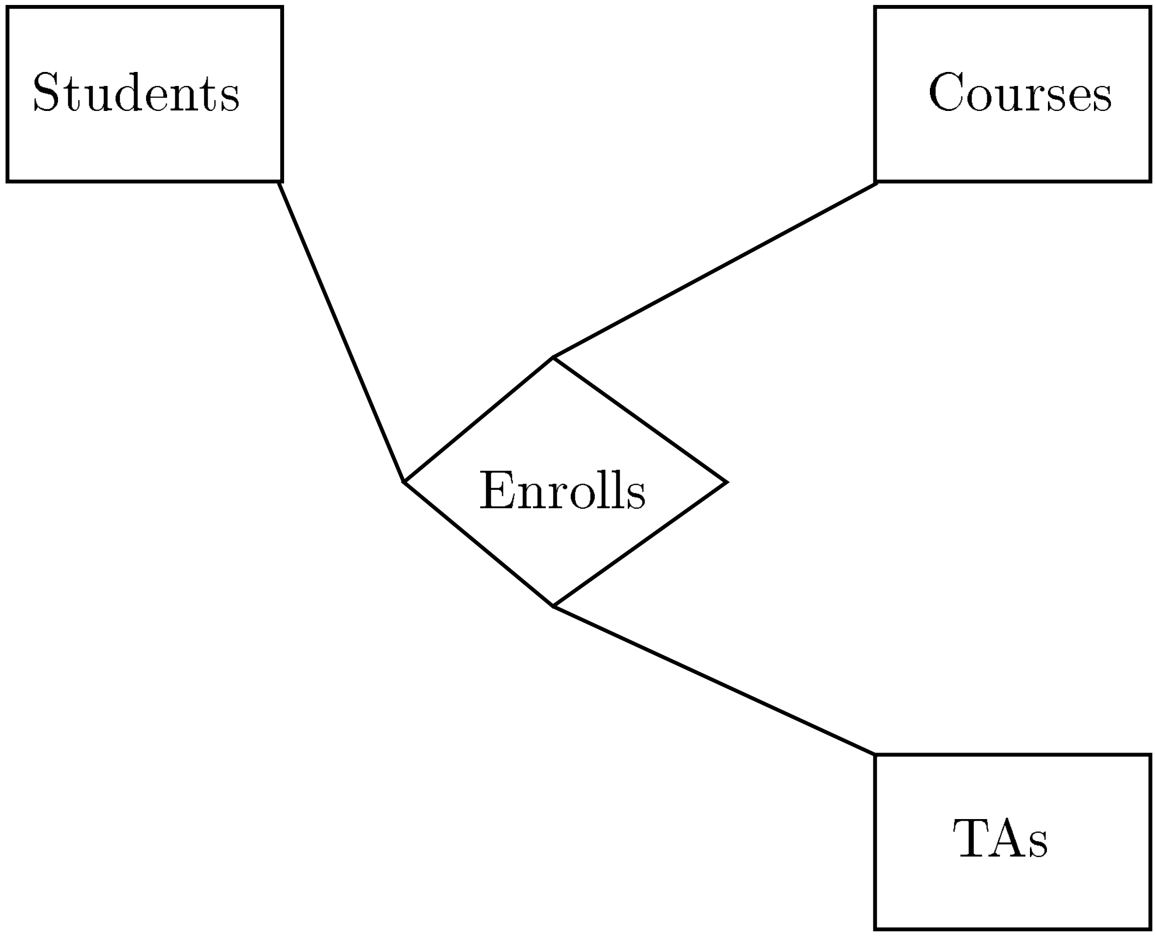
Usually *binary* relationships (connecting two E.S.) suffice.

- However, there are some cases where three or more E.S. must be connected by one relationship.

- Example: relationship among students, courses, TA's. Possibly, this E/R diagram is OK:



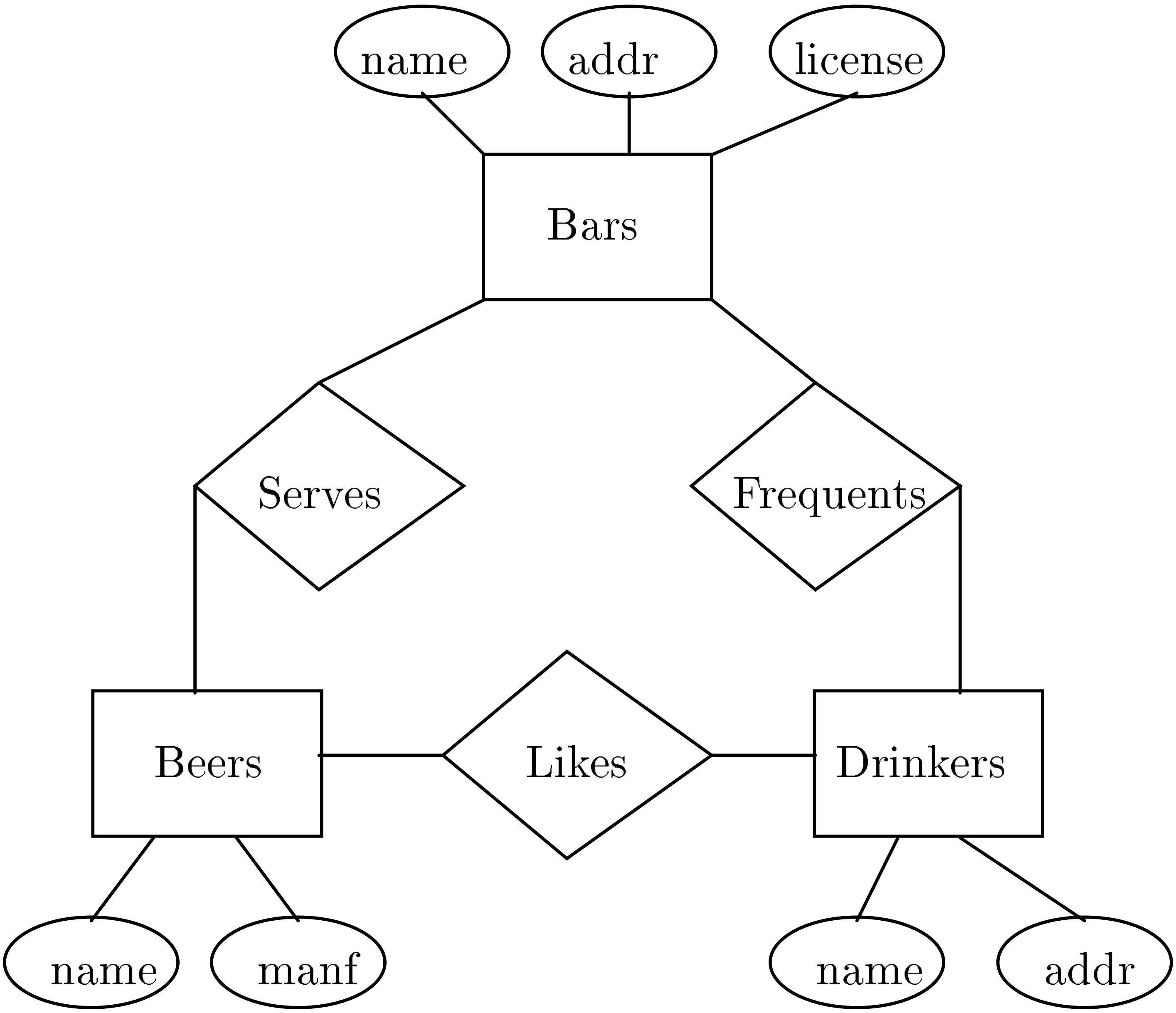
- Works in CS145, because each TA is a TA of all students. Connection student-TA is *only* via the course.
- But what if students were divided into sections, each headed by a TA?
  - Then, a student in CS145 would be related to only one of the TA's for CS145. Which one?
- Need a 3-way relationship to tell.



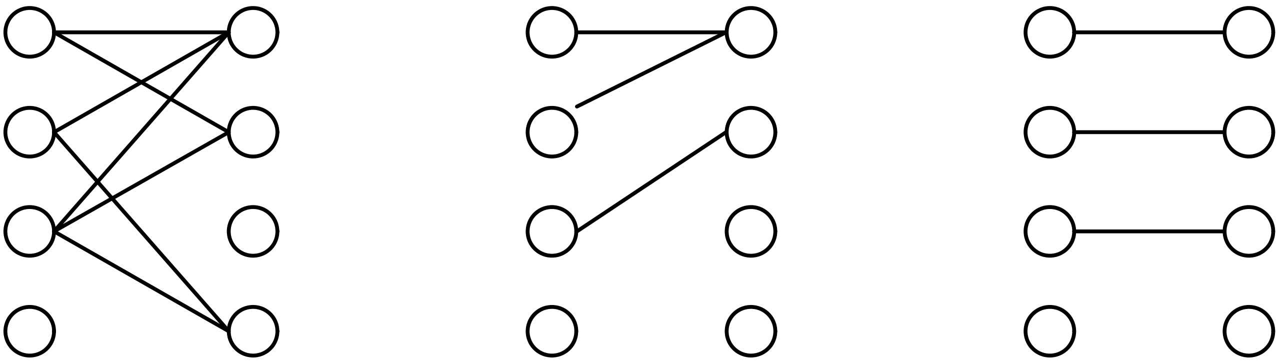
Students	Courses	TAs
Ann	CS145	Don
Bob	CS145	Edy
Cal	CS145	Don
...	...	...

Beers-Bars-Drinkers Example

- Our running example for the course.



Multiplicity of Relationships

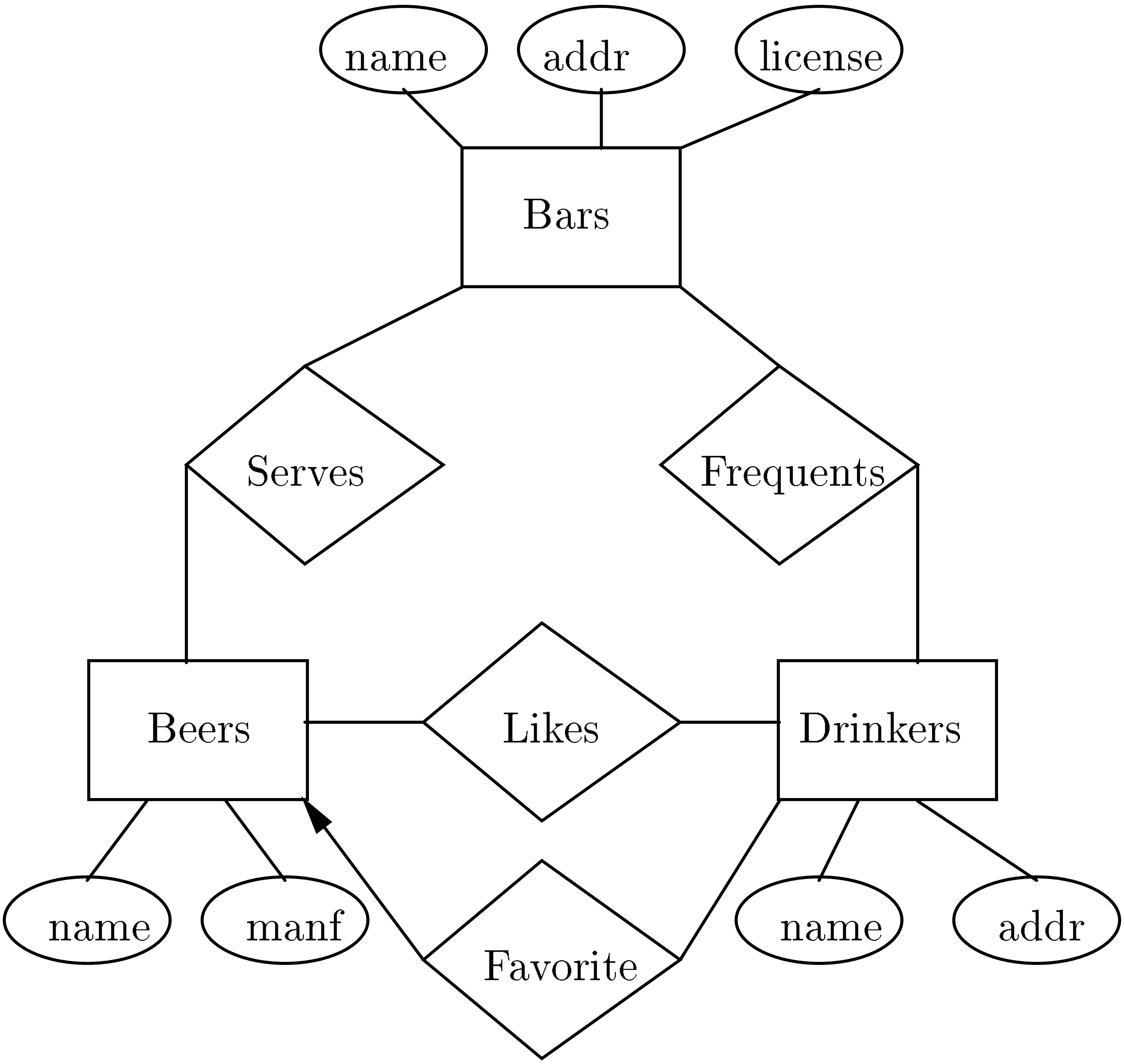


Many-many Many-one One-one

Representation of Many-One

- E/R: arrow pointing to "one."
  - ✧ Rounded arrow = "exactly one."

Example: Drinkers Have Favorite Beers



One-One Relationships

Put arrows in both directions.

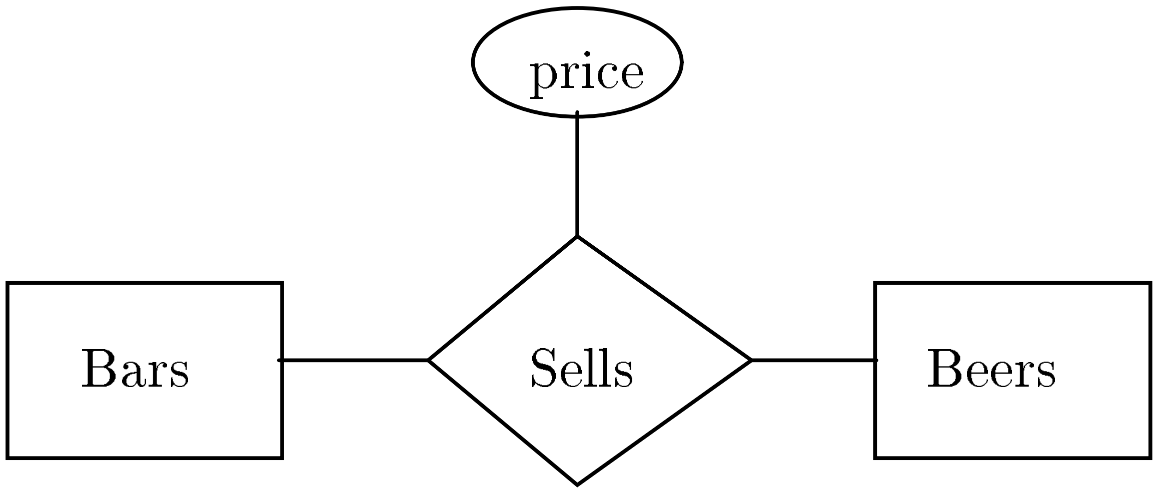
Design Issue:

Is the rounded arrow justified?

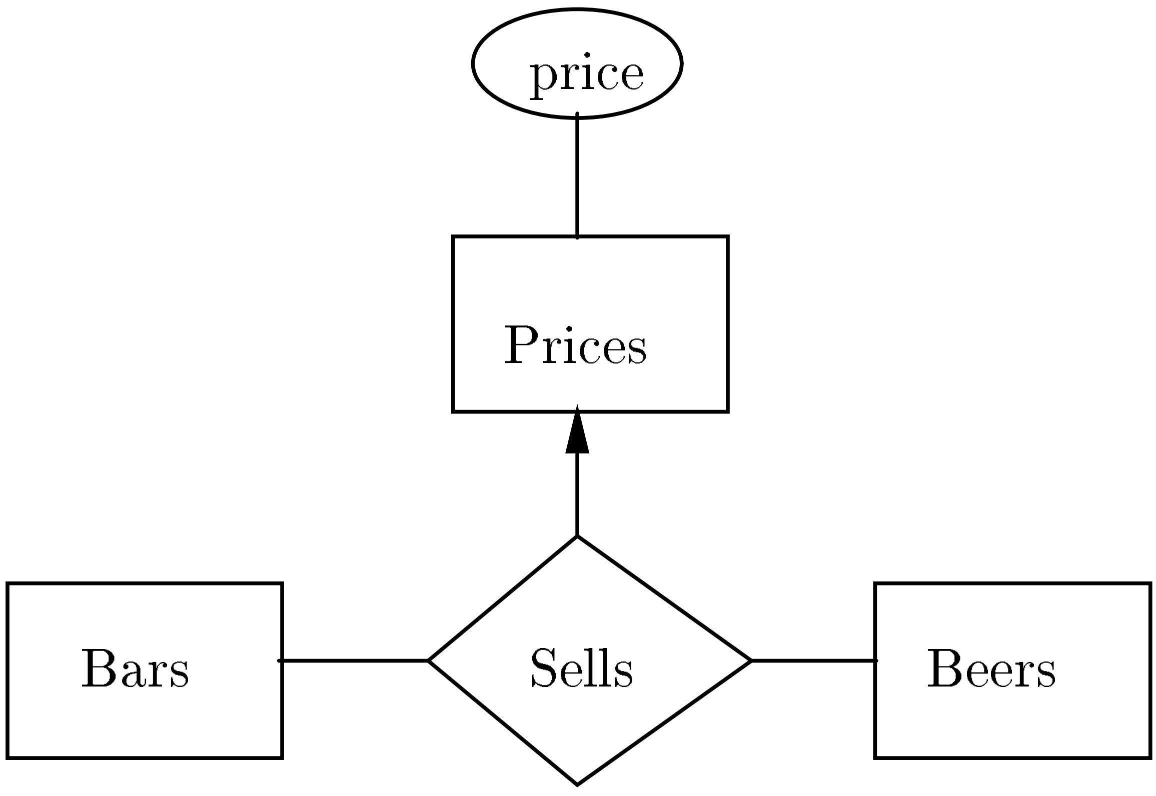
Design Issue:

Here, manufacturer is an E.S.; in earlier diagrams it is an attribute. Which is right?

Attributes on Relationships



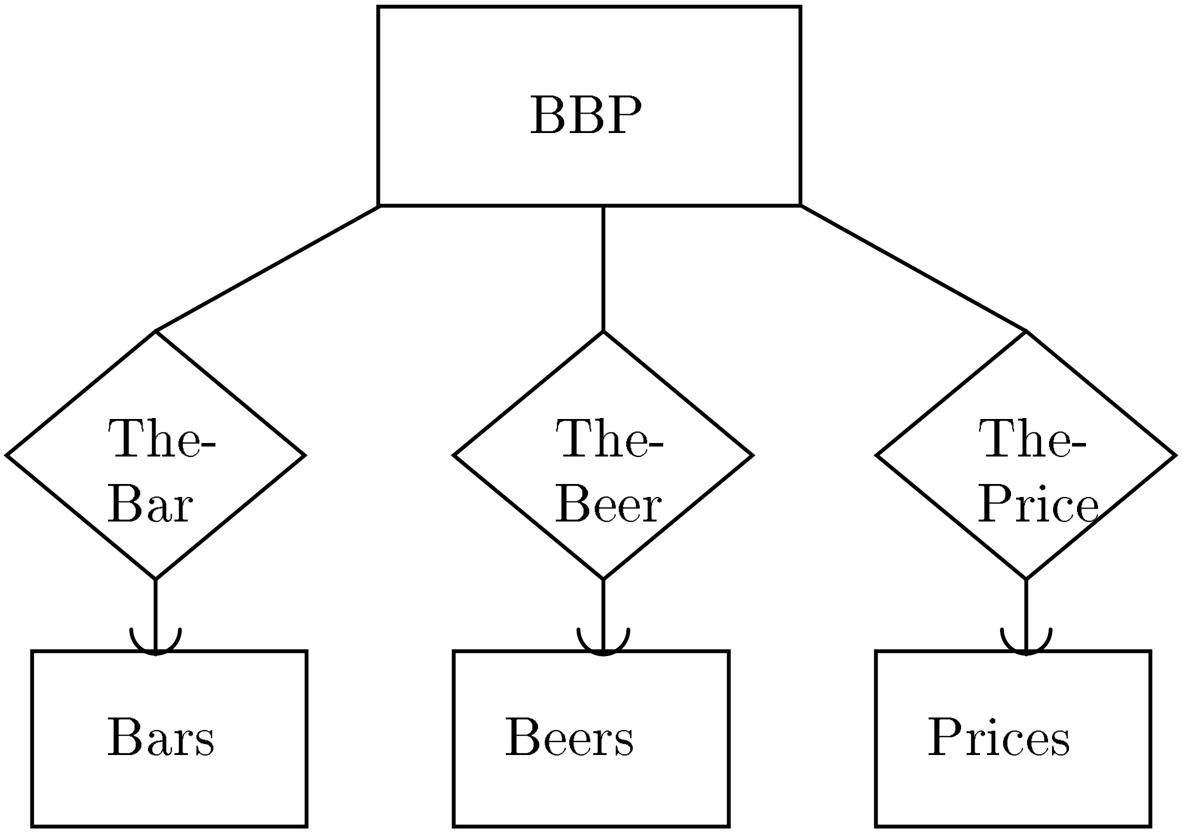
- Shorthand for 3-way relationship:



- A true 3-way relationship.
  - ❖ Price depends jointly on beer and bar.
- Notice arrow convention for multiway relationships: "all other E.S. determine one of these."
  - ❖ Not sufficiently general to express any possibility.
  - ❖ However, if price, say, depended only on the beer, then we could use two 2-way relationships: price-beer and beer-bar.
  - ❖ Or better: just make price an attribute of beer.

Converting Multiway to 2-Way

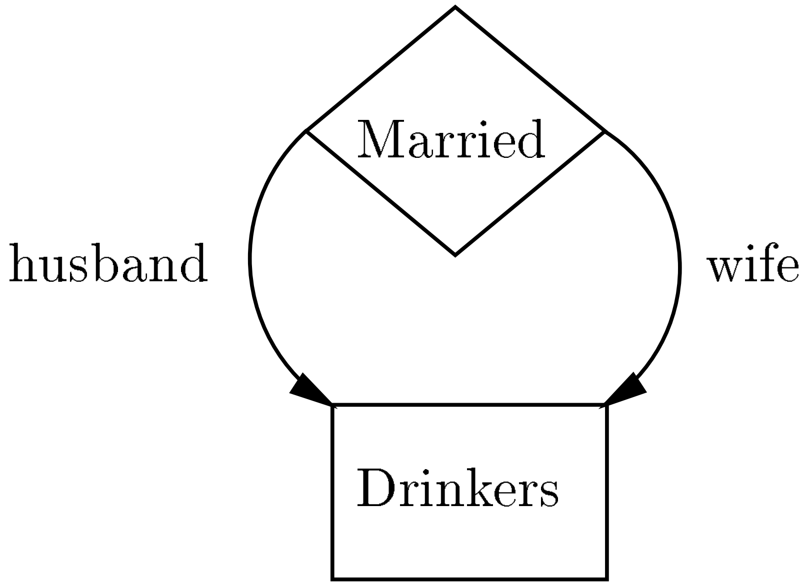
- Baroque in E/R, but necessary in certain "object-oriented" models.
- Create a new *connecting* E.S. to represent rows of a relationship set.
  - E.g., (Joe's Bar, Bud, \$2.50) for the *Sells* relationship.
- Many-one relationships from the connecting E.S. to the others.



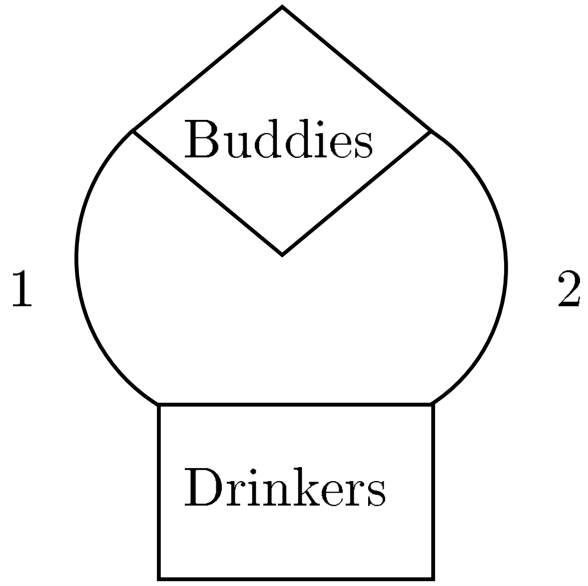
Roles

Sometimes an E.S. participates more than once in a relationship.

- Label edges with *roles* to distinguish.



Husband	Wife
d_1	d_2
d_3	d_4
$\dots$	$\dots$



Buddy1	Buddy2
d_1	d_2
d_1	d_3
d_2	d_1
d_2	d_4
$\ldots$	$\ldots$

- Notice *Buddies* is symmetric, *Married* not.
  - ✦ No way to say "symmetric" in E/R.

Design Question

Should we replace `husband` and `wife` by one relationship `spouse`?

Subclasses

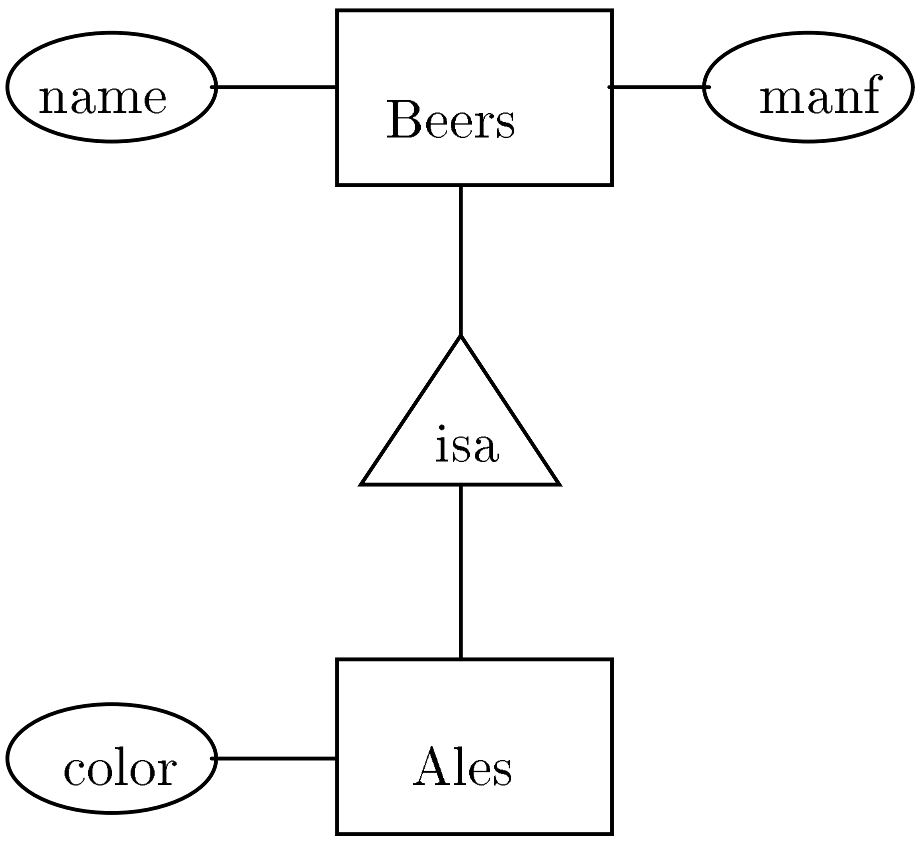
Subclass = special case = fewer entities = more properties.

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.



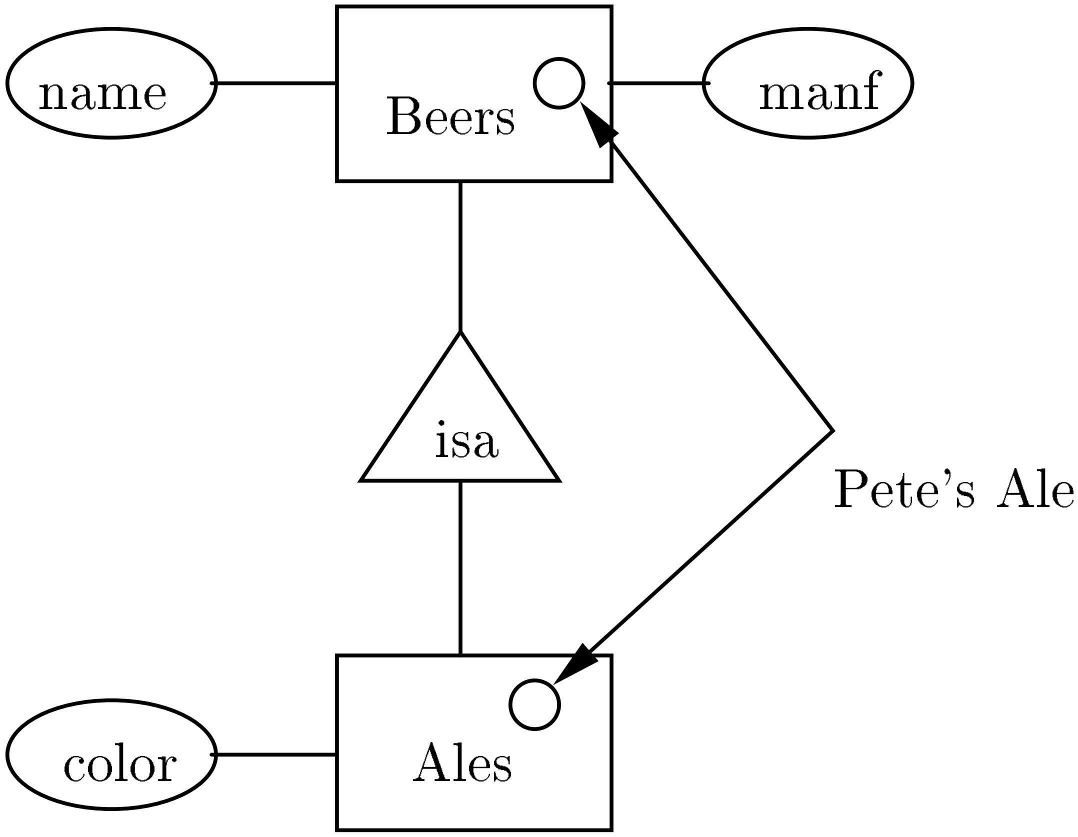
Different Subclass Viewpoints

1. *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.

2. Contrasts with *object-oriented viewpoint*: An object (entity) belongs to exactly one class.

   ✥ It *inherits* properties of its superclasses.



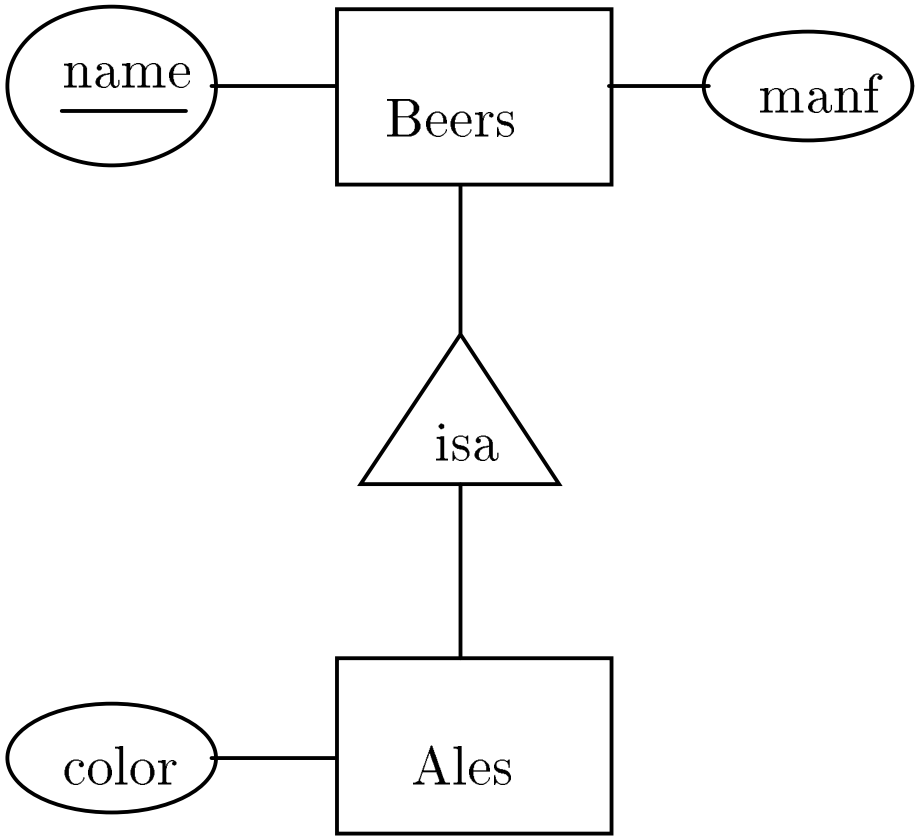
Keys

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.

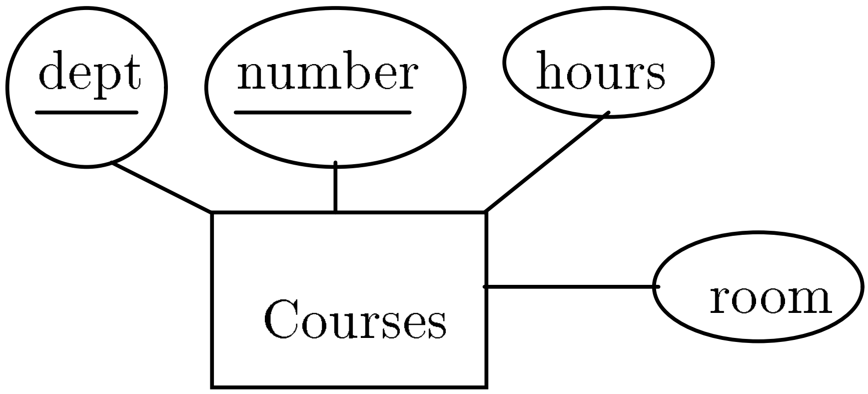
Example

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 designated it as such.