CS411
Database Systems

02: The Entity-Relationship Model

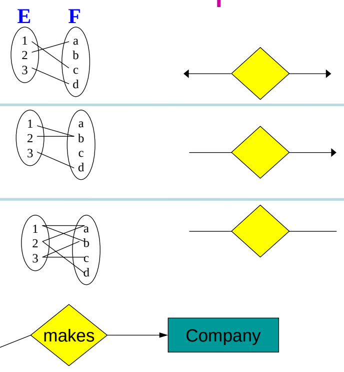
Kazuhiro Minami

We can show the cardinality of a relationship

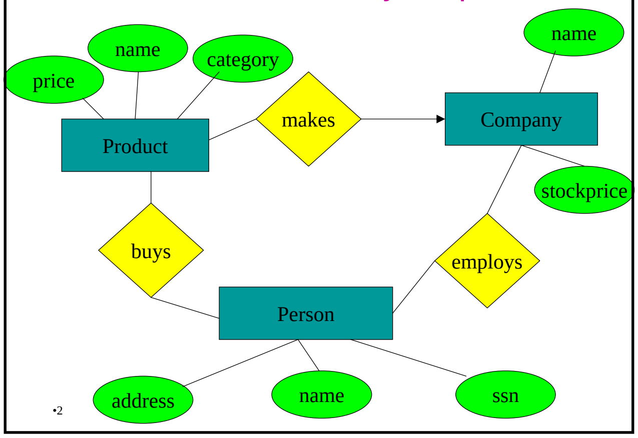
one-one

many-one

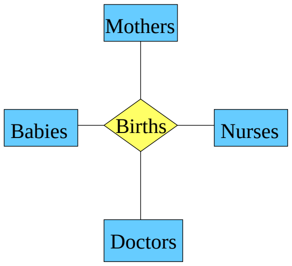
many-many



The ER model is very simple



Exercise 4.2.5
(Multiway relationships)



(baby, mother, nurse, doctor)
= (Bob, Mary, Kate, Dave)

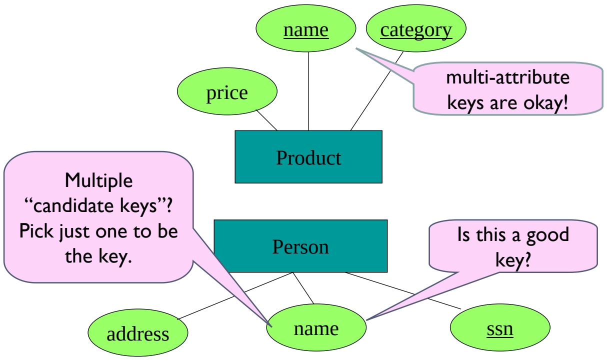
At a birth, there is one baby, one mother, any number of nurses, and any number of doctors. For each, tell how to add arrows or other elements to the E/R diagram.

a) For every baby, there is a unique mother

b) For every combination of a baby, nurse, and doctor, there is a unique mother

c) For every combination of a baby and a mother there is a unique doctor.

Underline the key for each entity set



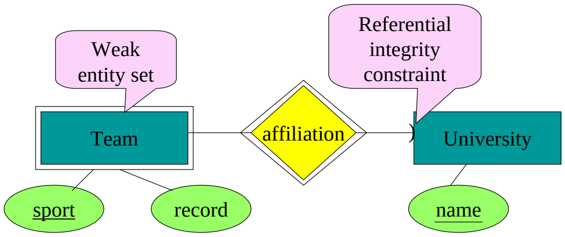
multi-attribute keys are okay!

Multiple "candidate keys"? Pick just one to be the key.

Is this a good key?

Sometimes your entity might not seem to have a key

Weak entity set: some or all of its key attributes come from other classes to which it is related.

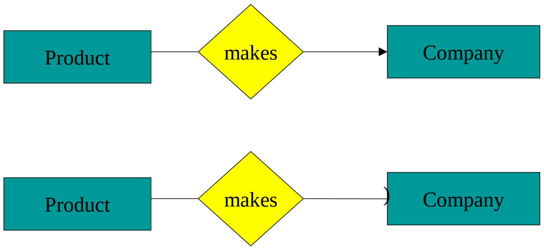


Weak entity set

Referential integrity constraint

Referential Integrity Constraints

- The reference integrity constraint on relationships explicitly requires a reference to exist
- The DB equivalent of a dangling pointer



Exercise 2

Q1. One way to represent students and the grades they get in course is to use entity sets corresponding to students, to courses, and to "enrollments." Enrollment entities form a "connecting" entity set between students and courses and can be used to represent not only the fact that a student is taking a certain course, but the grade of the student in the course. Draw an E/R diagram for this situation, indicating weak entity sets and the keys for the entity set. You can assume an appropriate set of attributes for "students" and "courses" entitysets.

Q2. Modify your solution so that we can record grades of the student for each of several assignments within a course.

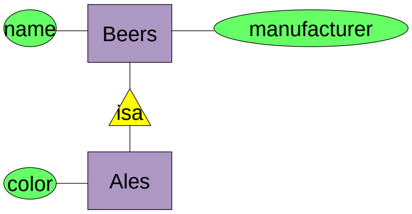
Subclasses

- Subclass = special case = fewer entities = 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*.

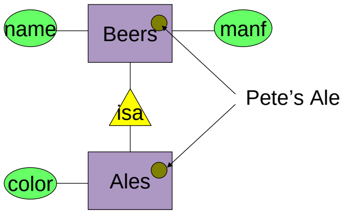
ER subclasses are different from object oriented subclasses

- In the object-oriented world, objects are in one class only.
  - Subclasses inherit properties from superclasses.
- In contrast, E/R entities have components in all subclasses to which they belong.
  - Matters when we convert to relations.

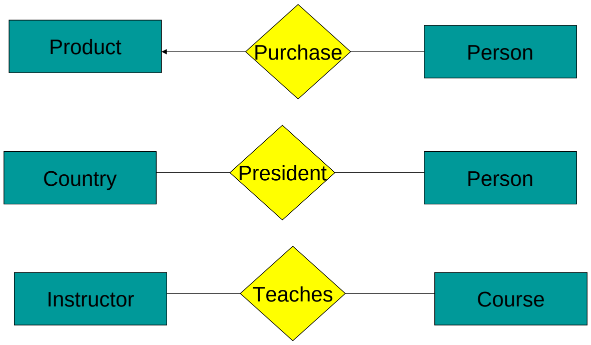
Example



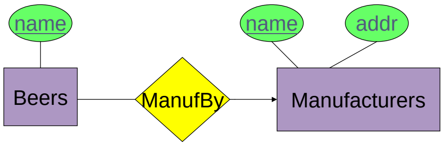
Example



ER Design Principle #1: Model your domain faithfully



Good

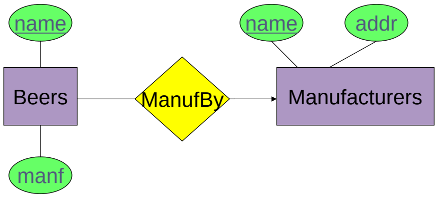


This design gives the address of each manufacturer exactly once.

Principle #2: Avoid redundancy

- Don’t say the same thing in two different ways.
- Redundancy wastes space and (more importantly) encourages inconsistency (i.e., *update anomaly*)
  - The two instances of the same fact may become inconsistent if we change one and forget to change the other, related version.

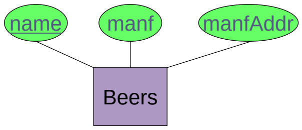
Bad



This design states the manufacturer of a beer twice: as an attribute and as a related entity.

We could use a set of attributes instead of an entity set

Q: Can we remove the Manufactures entity set?



This design repeats the manufacturer's address once for each beer; loses the address if there are temporarily no beers for a manufacturer.

Principle #3: Don't overuse entity sets

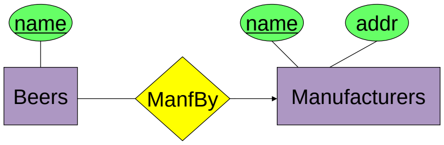
An entity set should satisfy at least one of the following conditions:

- It is more than the name of something; it has at least one non-key attribute.

or

- It is the "many" in a many-one or many-many relationship.

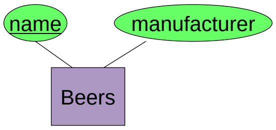
Good



Manufacturers deserves to be an entity set because of the nonkey attribute *addr*.

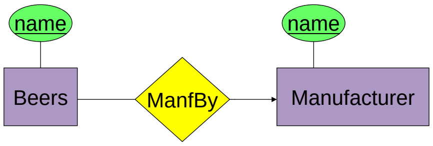
Beers deserves to be an entity set because it is the "many" of the many-one relationship *ManfBy*.

Good



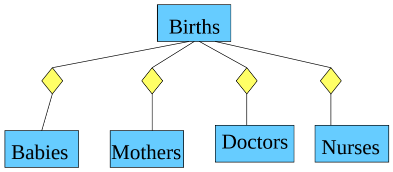
No need to make the manufacturer an entity set, because we only remember its name.

Bad



Since the manufacturer is nothing but a name, and is not at the "many" end of any relationship, it should not be an entity set.

Exercise 4.2.6



Use arrows to represent the following conditions:

a) Every baby is a result of a unique birth, and every birth is of a unique baby.
b) In addition to (a), every baby has a unique mother.
c) In addition to (a) and (b), for every birth there is a unique doctor.

In each case, what design flaws do you see?

Principle #4: Don't Overuse Weak Entity Sets

- Beginning database designers often make most entity sets weak, supported by all other entity sets to which they are linked.
- Instead, we create unique IDs for entity sets.
  - Social-security numbers, driver's license numbers, automobile VINs, …
- Only use weak entity sets when necessary.
  - Example: unique player numbers across all football teams in the world.