

3. Advanced E/R Concepts

What you will learn about in this section



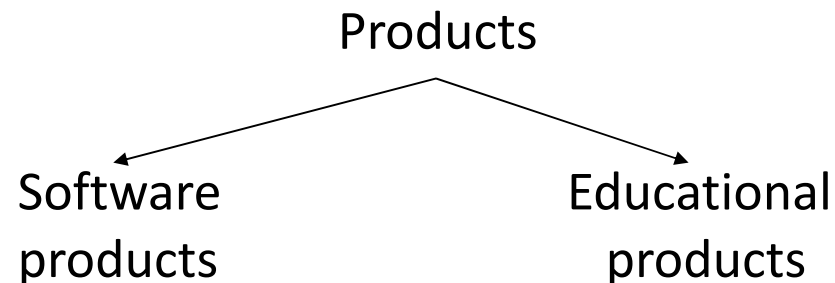
Subclasses & connection to OO

Constraints

Weak entity sets

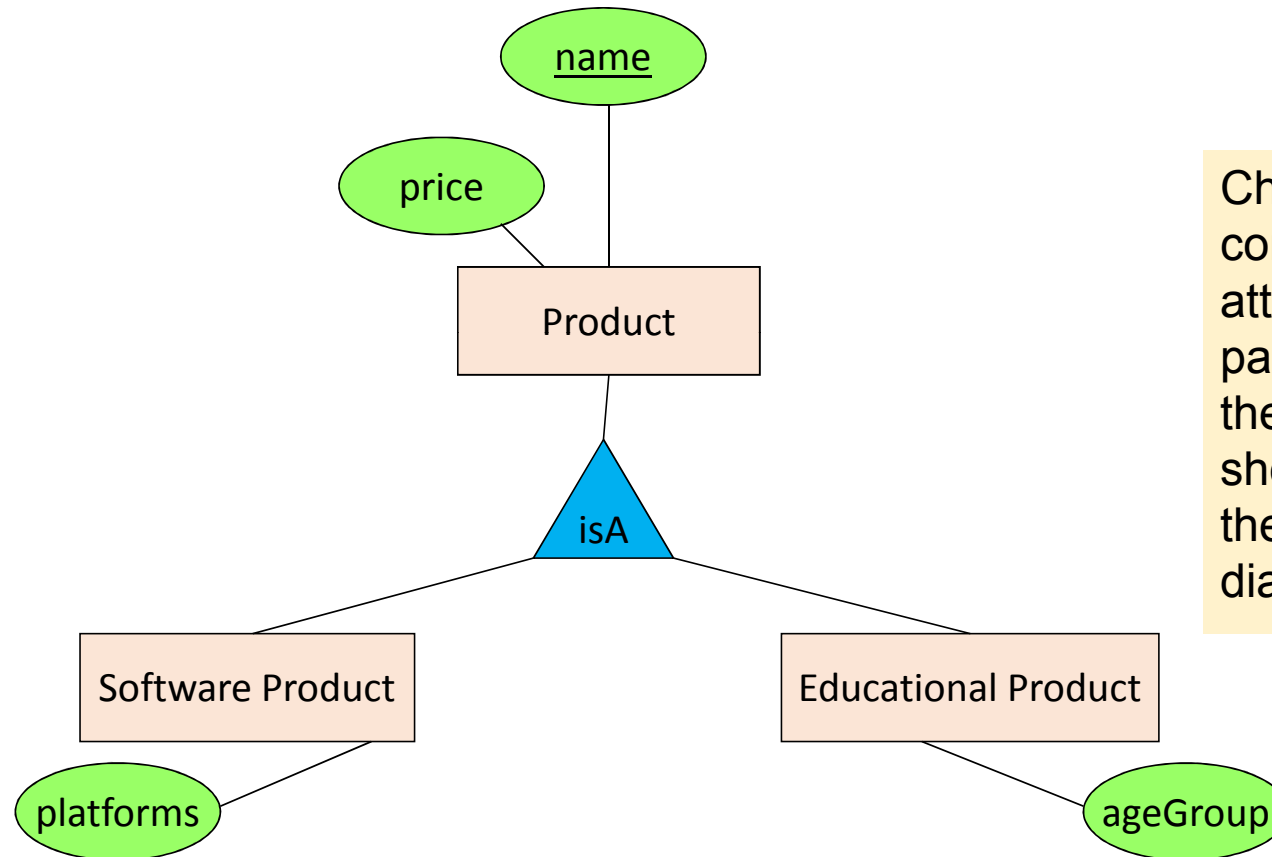
Modeling Subclasses

- Some objects in a class may be special, i.e. worthy of their own class
 - Define a new class?
 - *But what if we want to maintain connection to current class?*
 - Better: define a *subclass*
 - *Ex:*



We can define **subclasses** in E/R!

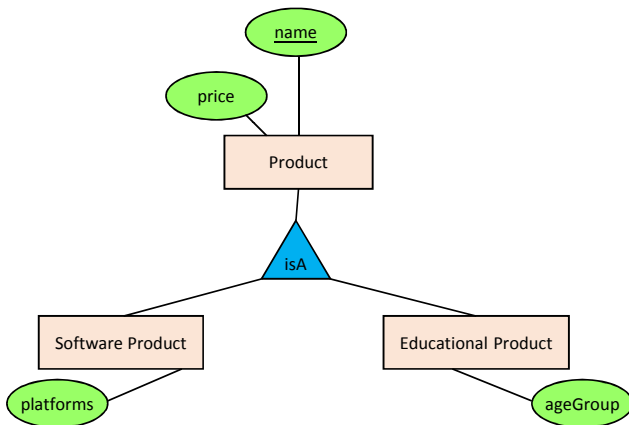
Modeling Subclasses



Child subclasses contain all the attributes of *all* of their parent classes **plus** the new attributes shown attached to them in the E/R diagram

Understanding Subclasses

- Think in terms of records; ex:



- Product

name
price

- SoftwareProduct

name
price
platforms

- EducationalProduct

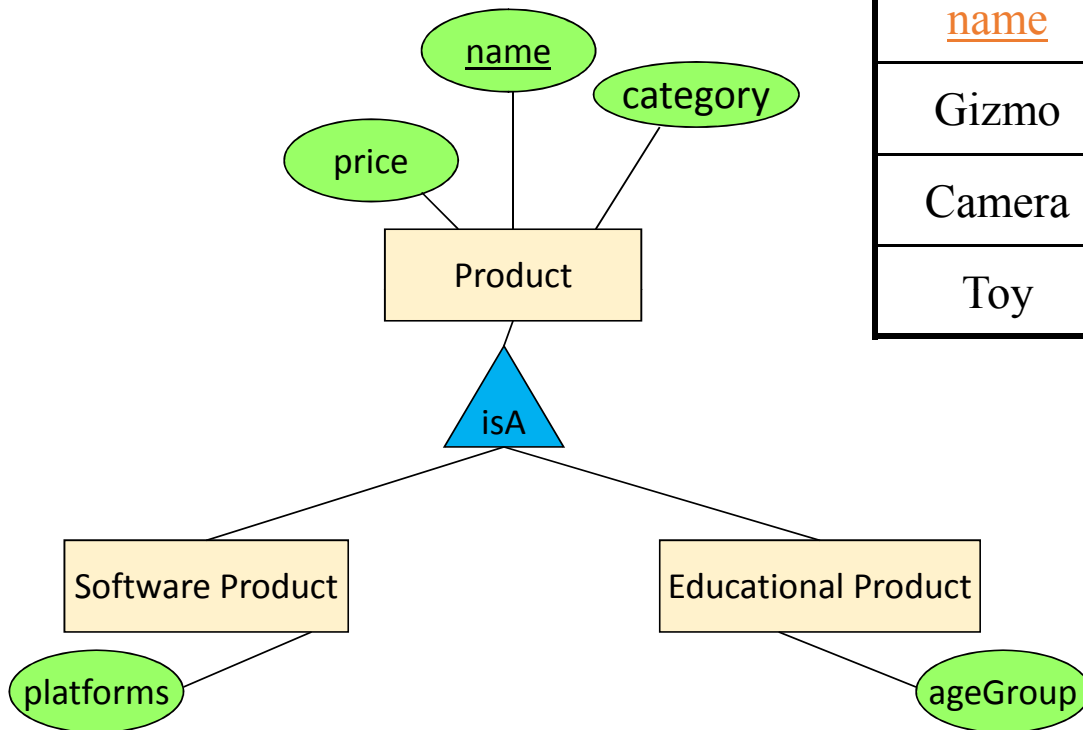
name
price
ageGroup

Child subclasses contain all the attributes of *all* of their parent classes **plus** the new attributes shown attached to them in the E/R diagram

How to convert E/R to tables...

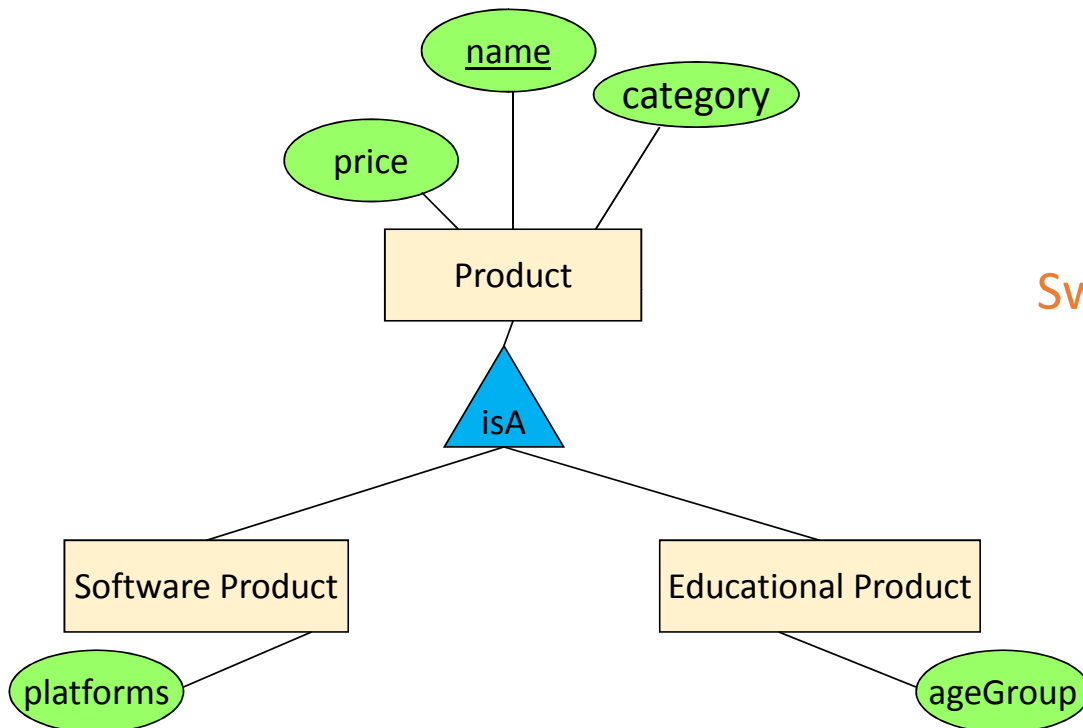
Product

<u>name</u>	price	category	platforms	ageGroup
Gizmo	99	gadget	unix	null
Camera	49	photo	null	null
Toy	39	gadget	null	retired



Many null values!!

Better option: one table for each subclass...



Product

<u>name</u>	price	category
Gizmo	99	gadget
Camera	49	photo
Toy	39	gadget

Sw.Product

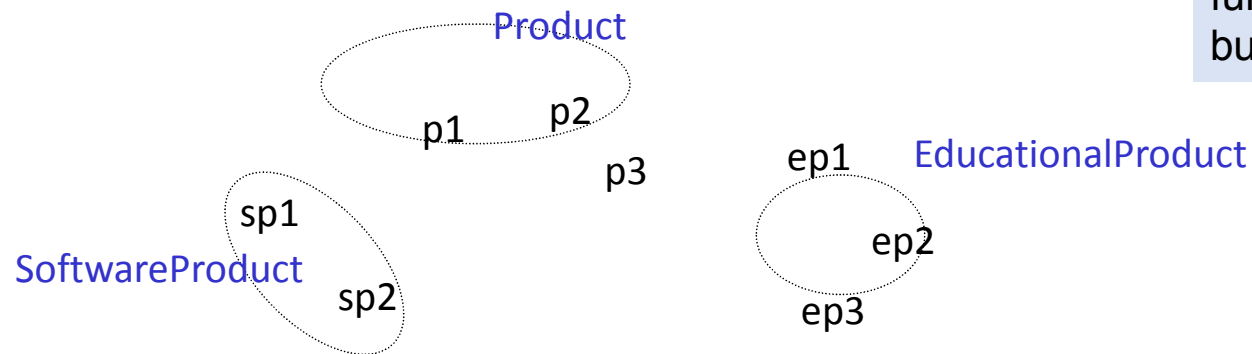
<u>name</u>	platforms
Gizmo	unix

Ed.Product

<u>name</u>	ageGroup
Toy	retired

Difference between OO and E/R inheritance

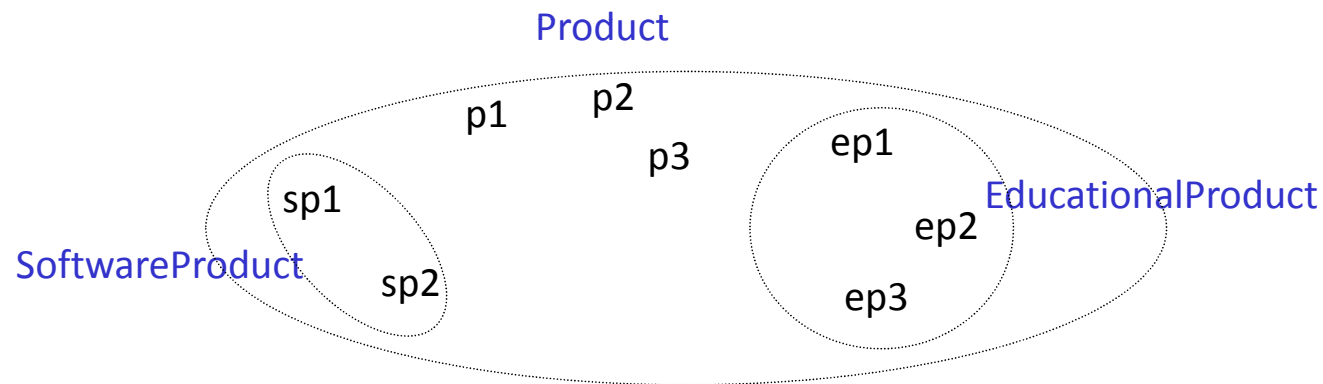
- OO: Classes are disjoint (same for Java, C++)



OO = **Object Oriented**. E.g. classes as fundamental building block, etc...

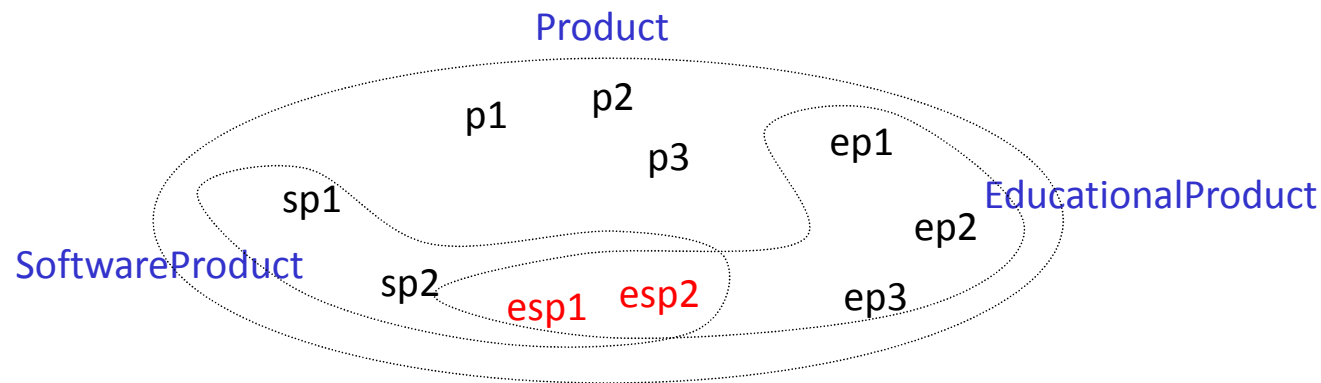
Difference between OO and E/R inheritance

- E/R: entity sets overlap



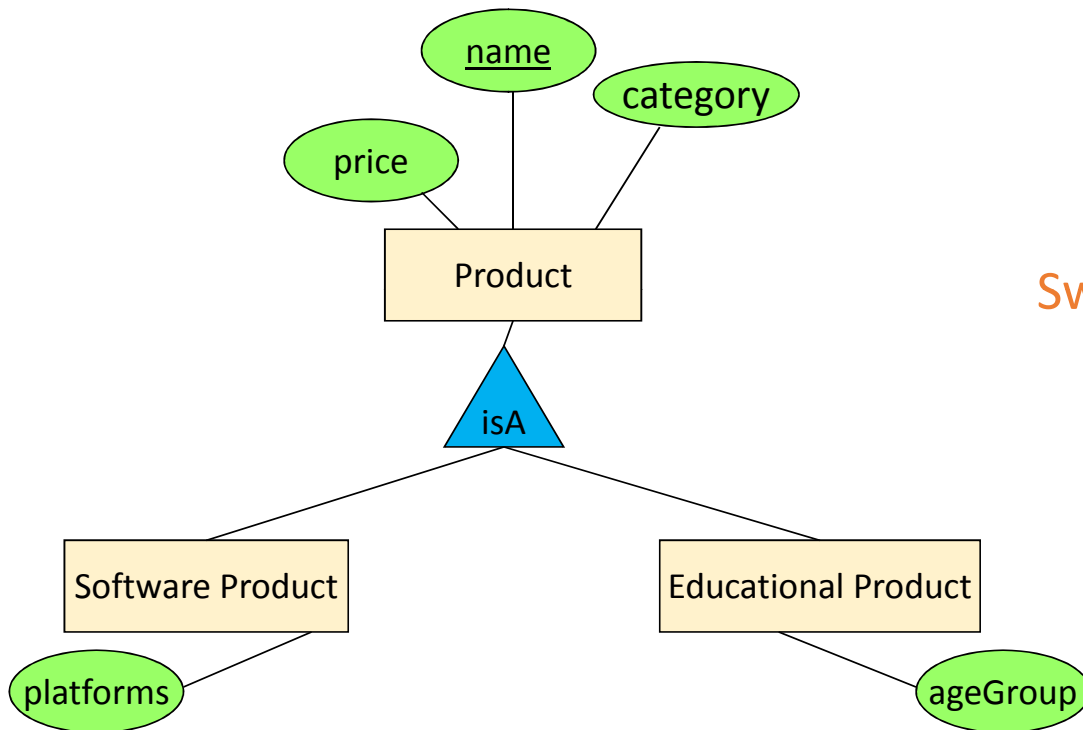
Difference between OO and E/R inheritance

We have three entity sets, but four different kinds of objects



No need for multiple inheritance in E/R

Suppose we have a product KIDPIX which is an educational software... Where do we keep the information?



Product

<u>name</u>	price	category
Gizmo	99	gadget
Camera	49	photo
Toy	39	gadget

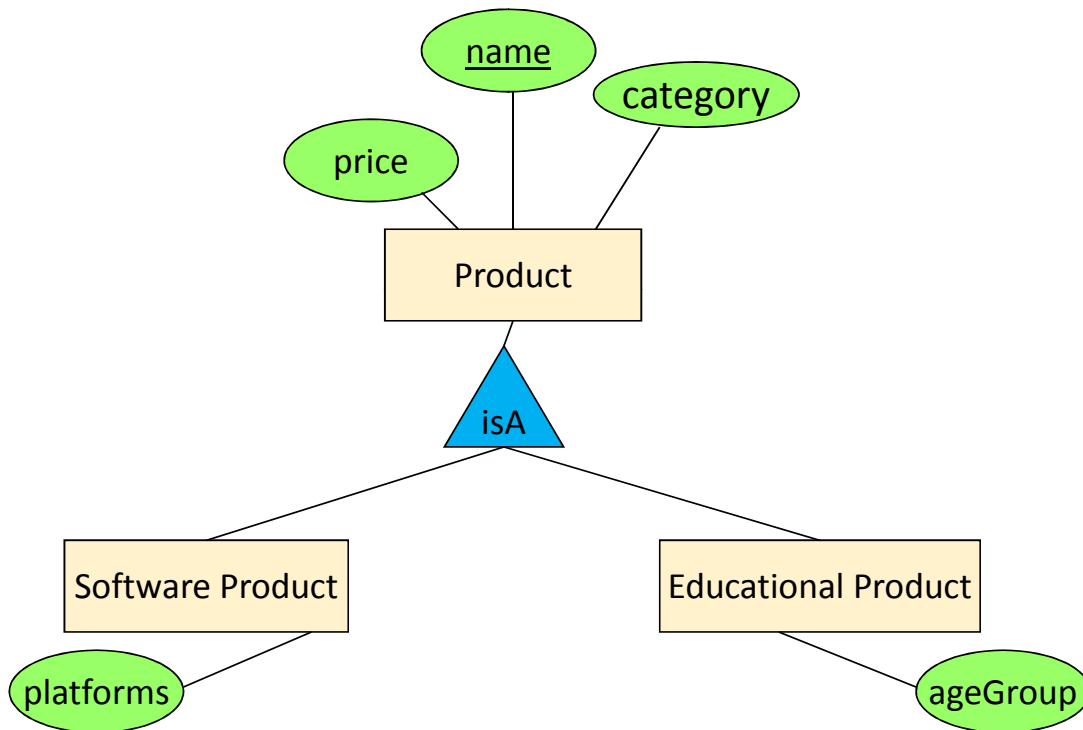
Sw.Product

<u>name</u>	platforms
Gizmo	unix

Ed.Product

<u>name</u>	ageGroup
Gizmo	todler
Toy	retired

Suppose we have a product KIDPIX which is an educational software... Where do we keep the information?



Product

<u>name</u>	price	category
Gizmo	99	gadget
Camera	49	photo
Toy	39	gadget
KidPix	100	gadget

<u>name</u>	platforms
Gizmo	unix
KidPix	Macintosh

Sw.Product

<u>name</u>	ageGroup
Toy	retired
KidPix	todler

Ed.Product

IsA Review

- If we declare ***A IsA B*** then every **A** is a **B**
- We use IsA to
 - Add descriptive attributes to a subclass
 - To identify entities that participate in a relationship
 - No table for IsA relationship
- **No need for multiple inheritance**

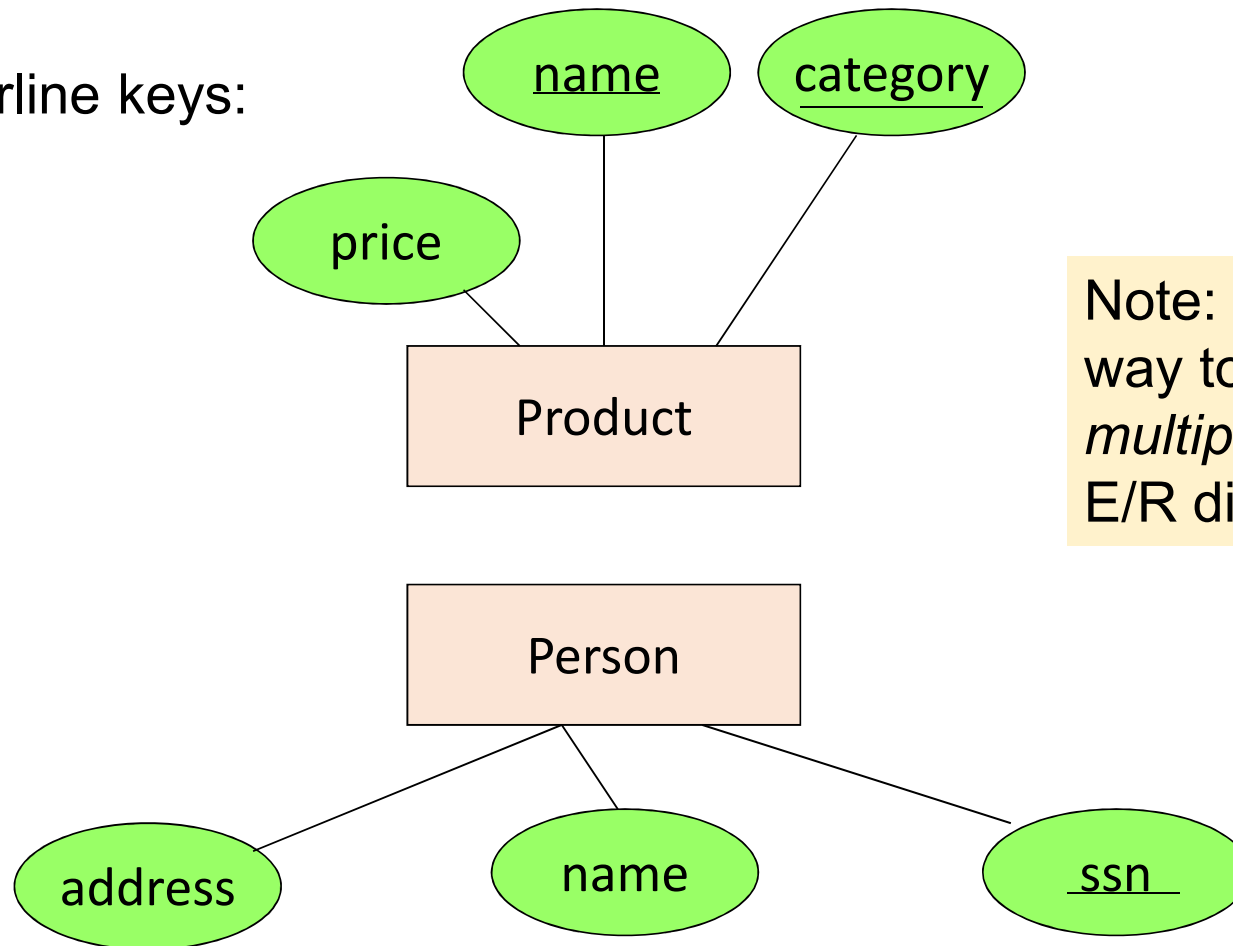
Constraints in E/R Diagrams

- Finding constraints is part of the E/R modeling process. Commonly used constraints are:
 - Keys: Implicit constraints on uniqueness of entities
 - *Ex: An SSN uniquely identifies a person*
 - Single-value constraints:
 - *Ex: a product can be made by only one company*
 - Referential integrity constraints: Referenced entities must exist
 - *Ex: if you work for a company, it must exist in the database*
 - Degree constraints:
 - *Ex: A movie may not have more than 10 actors*

Recall
FOREIGN
KEYS!

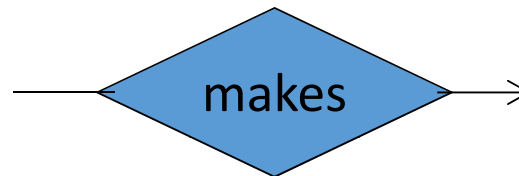
Keys in E/R Diagrams

Underline keys:

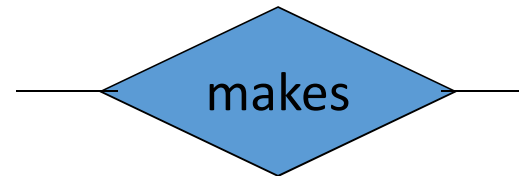


Note: no formal way to specify *multiple* keys in E/R diagrams...

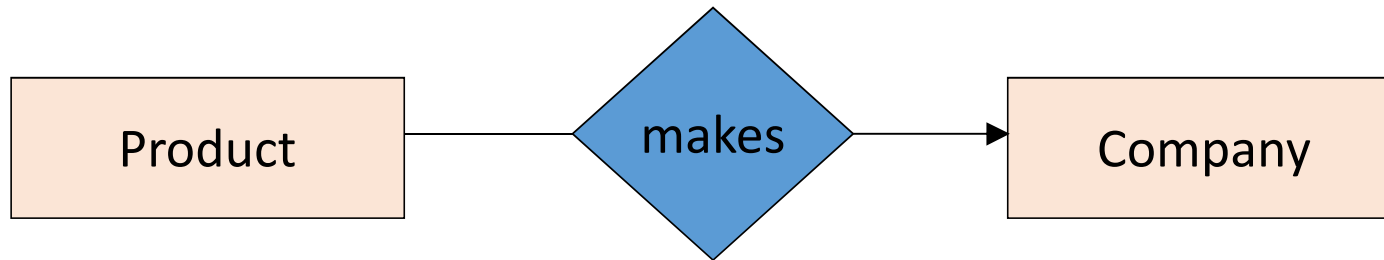
Single Value Constraints



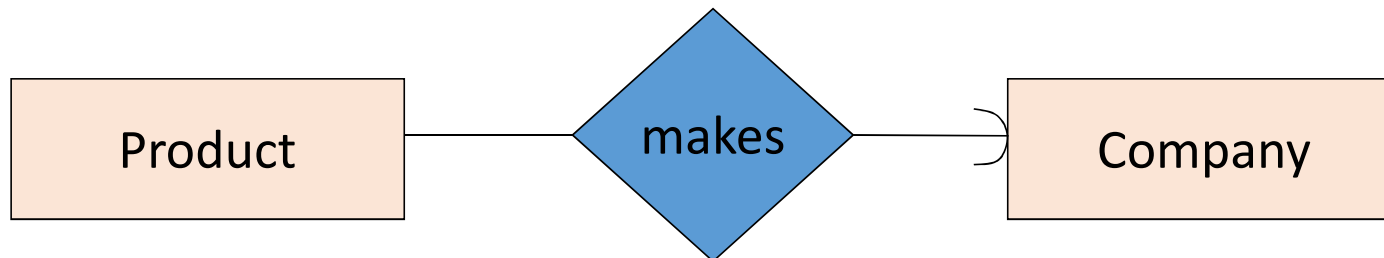
v. s.



Referential Integrity Constraints

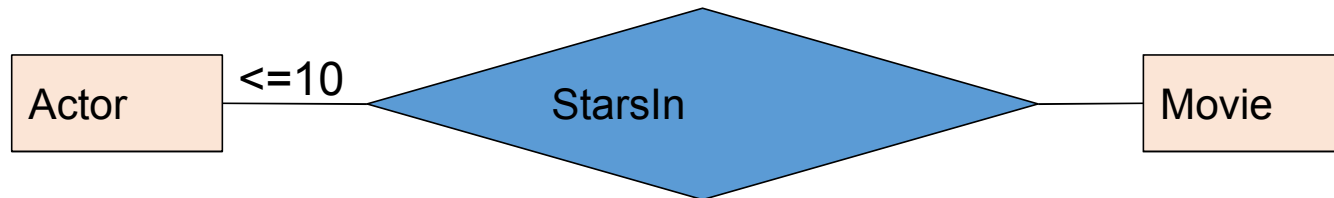


Each product made by at most one company.
Some products made by no company?



Each product is made by exactly one company.

Degree Constraints



- We can attach a bounding number to edges to indicate limits on the number of entities that can be connected to a single entity via a relationship set
- In the example above, a movie has at most 10 stars
- Note: a regular arrow is the constraint ≤ 1
- Note: a rounded arrow is the constraint $= 1$

Weak Entity Sets

The entity set that can not exist without another entity set

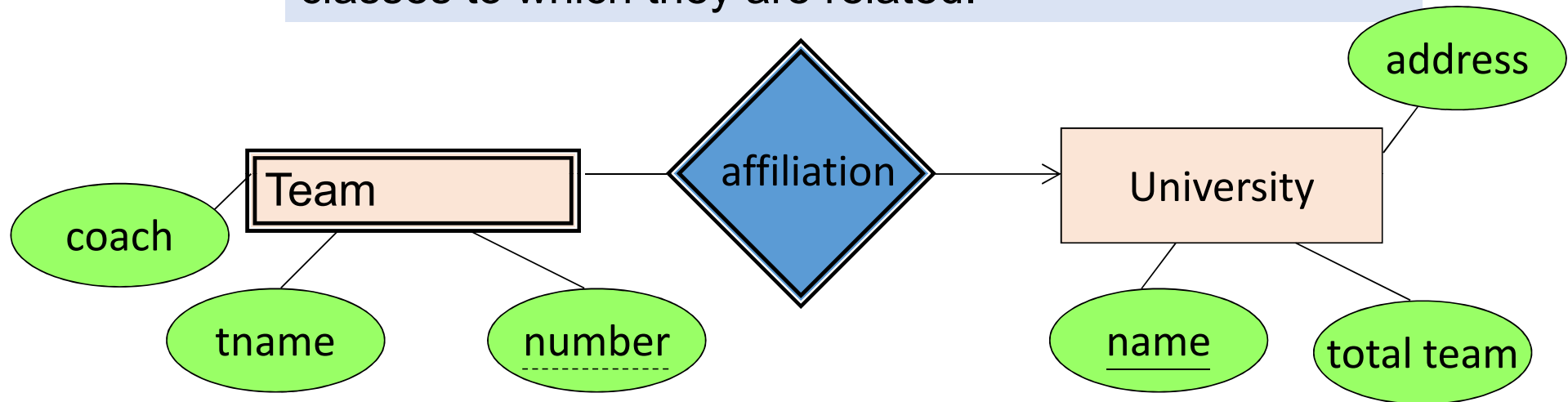
a ROOM can only exist in a BUILDING

a TIRE is considered as a strong entity because it can exist without being attached to a CAR

Question is strong entity type and answer is weak. Question is always there, but an answer requires a question to exist.

Weak Entity Sets

Entity sets are weak when their key comes from other classes to which they are related.



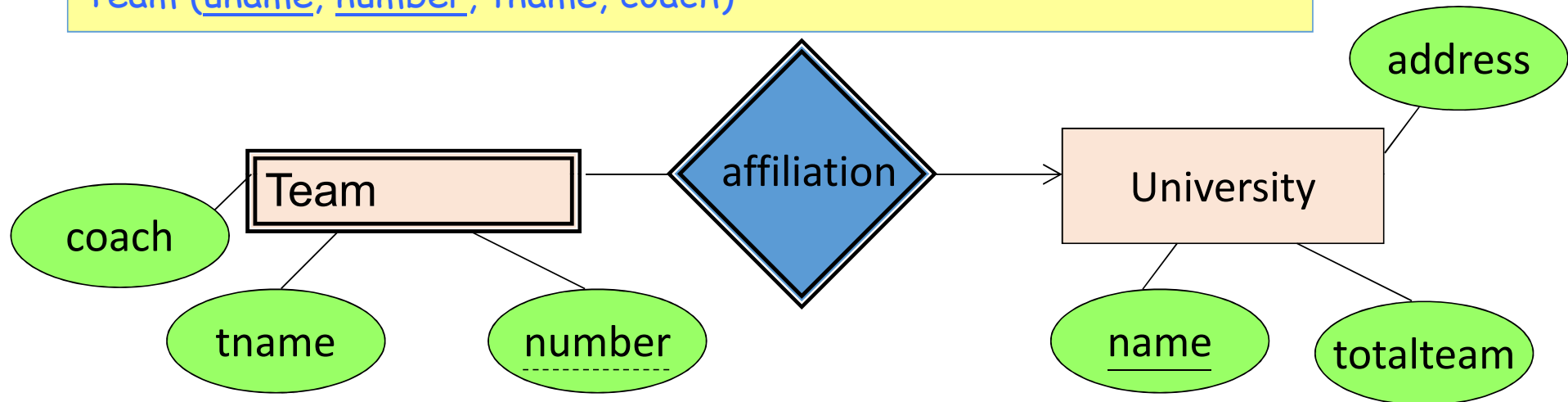
“Programming contest team” v. “**BUET Programming contest team**” (E.g., Dhaka university has a programming contest team too)

Weak Entity Sets

Only 2 tables are required, no table for the relationship is required

University (name, totalTeam, address)

Team (uname, number, tname, coach)



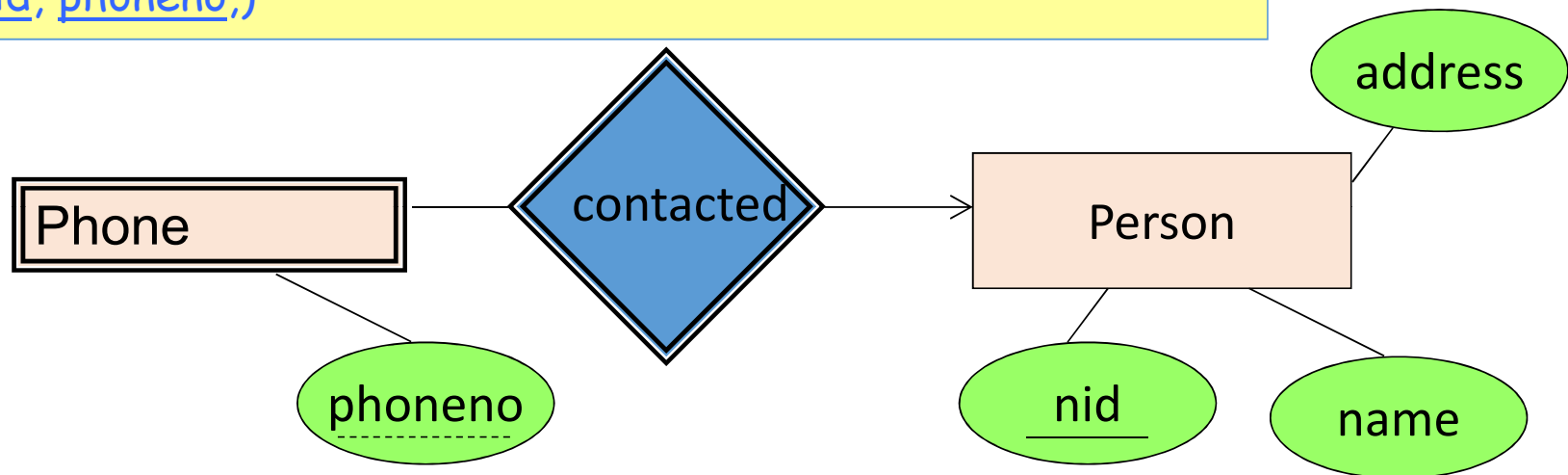
- number is a *partial key*. (denote with dashed underline).
- University is called the *identifying owner*.

Weak Entity Sets

Only 2 tables are required, no table for the relationship is required

Person (nid, name, address)

Phone (nid, phoneno,)



- Phone no is a *partial key*. (denote with dashed underline).
- Person is called the *identifying owner*.

E/R Summary

- E/R diagrams are a visual syntax that allows technical and non-technical people to talk
 - For conceptual design
- Basic constructs: **entity, relationship, and attributes**
- A good design is faithful to the constraints of the application

The bigger picture of what we have learned

