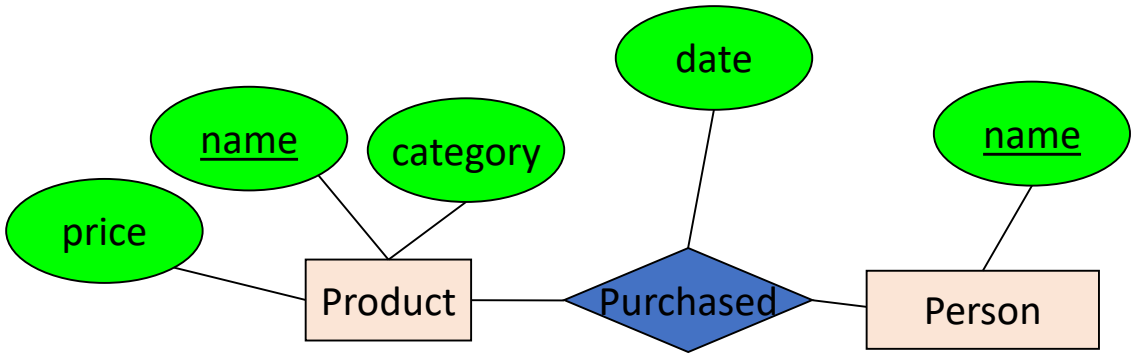
Decision: Relationship vs. Entity?

- **Q:** What does this say?

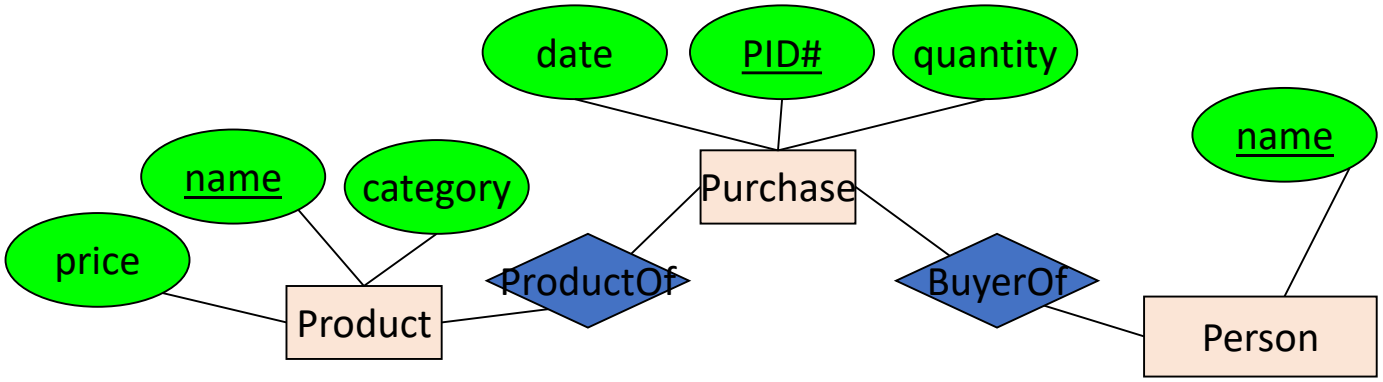


- **A:** A person can only buy a specific product once (on one date)

Modeling something as a relationship makes it unique; what if not appropriate?

Decision: Relationship vs. Entity?

- What about this way?



- *Now we can have multiple purchases per product, person pair!*

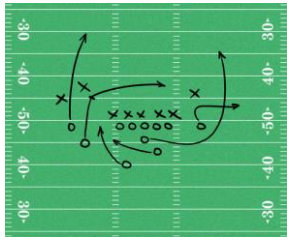
We can always use **a new entity** instead of a relationship. For example, to permit multiple instances of each entity combination!

Draw an E/R diagram for football

Use the following simplified model of a football season **(concepts to include are underlined):**

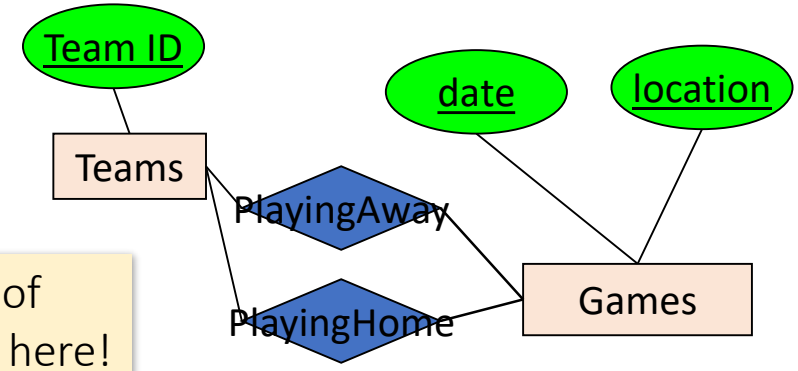
Teams play each other in **Games**. Each pair of teams can play each other multiple times

Players belong to Teams (assume no trades / changes).



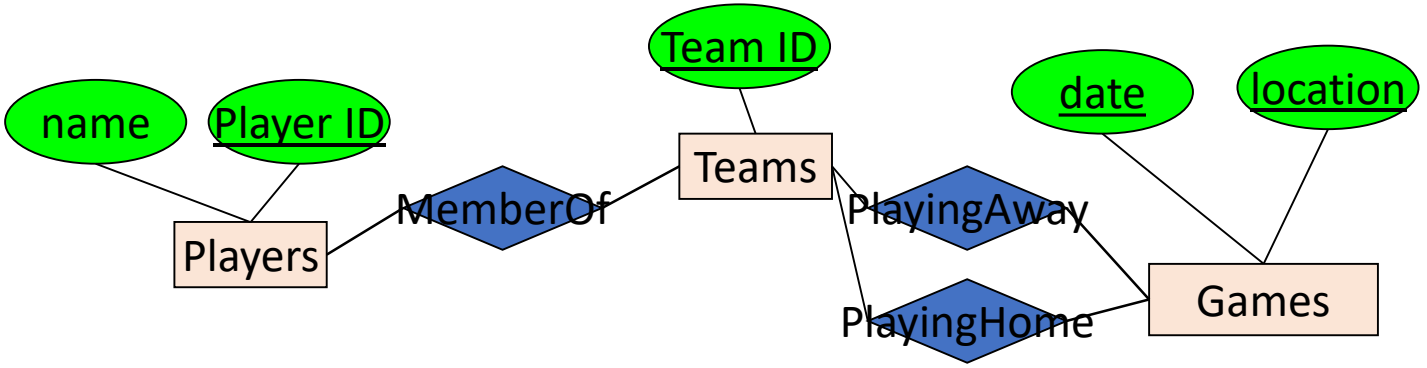
A Game is made up of **Plays** that result in a yardage gain/loss, and potentially a touchdown

A Play will contain either a **Pass** from one player to another, or a **Run** by one player

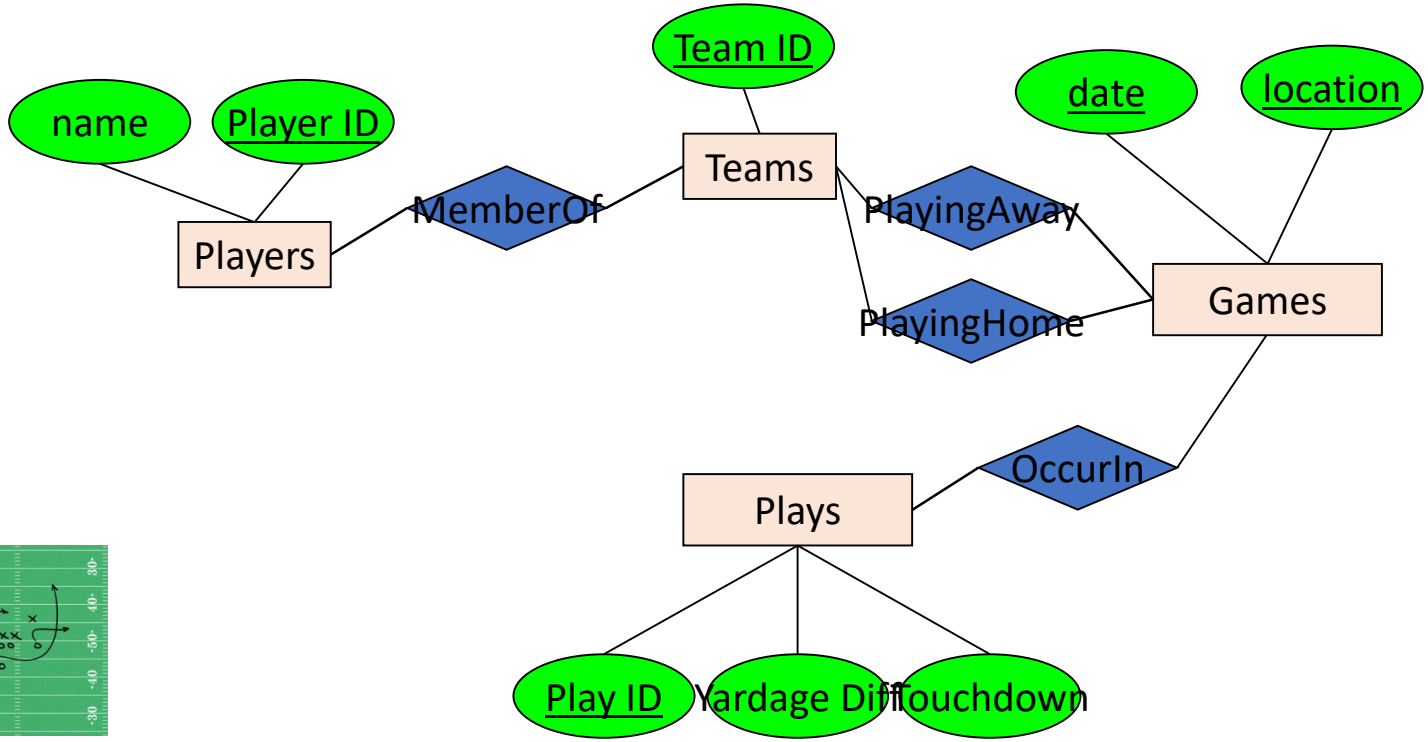


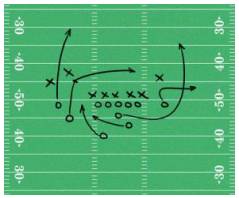
Note two copies of the Teams entity here!

Teams play each other in Games. Each pair of teams can play each other multiple times

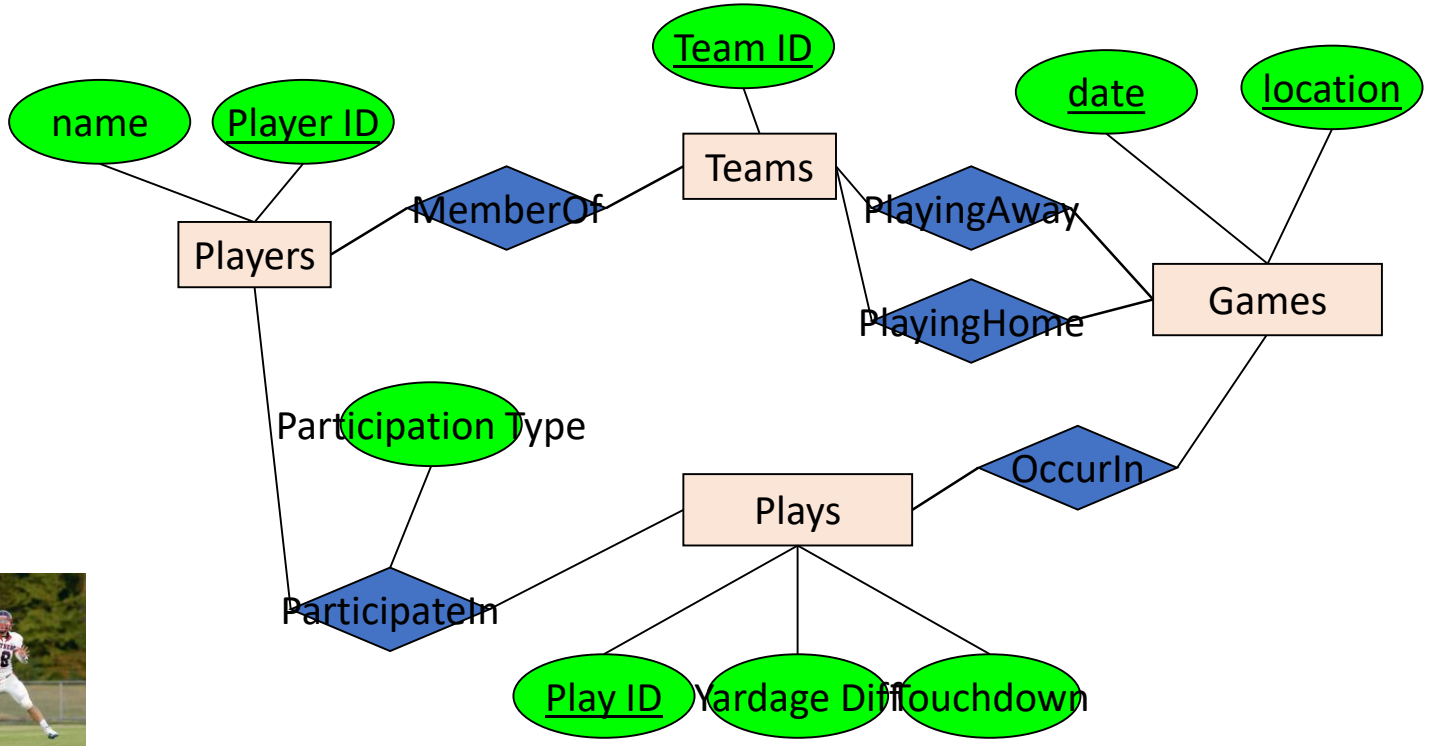


Players belong to Teams (assume no trades / changes)





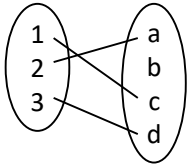
A Game is made up of Plays that result in a yardage gain/loss, and potentially a touchdown

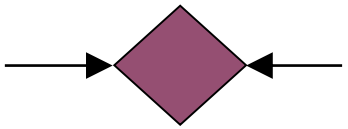


A Play will contain either a Pass from one player to another, or a Run by one player

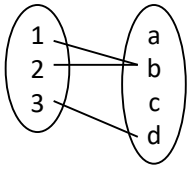
Multiplicity of E/R Relationships

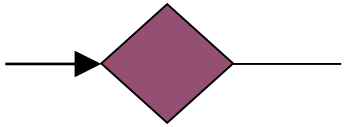
One-to-one:



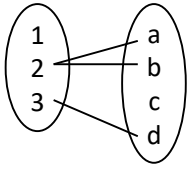


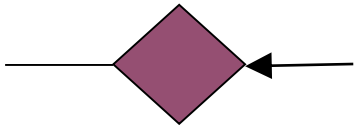
Many-to-one:



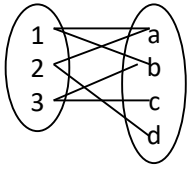


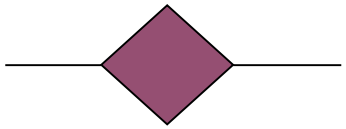
One-to-many:





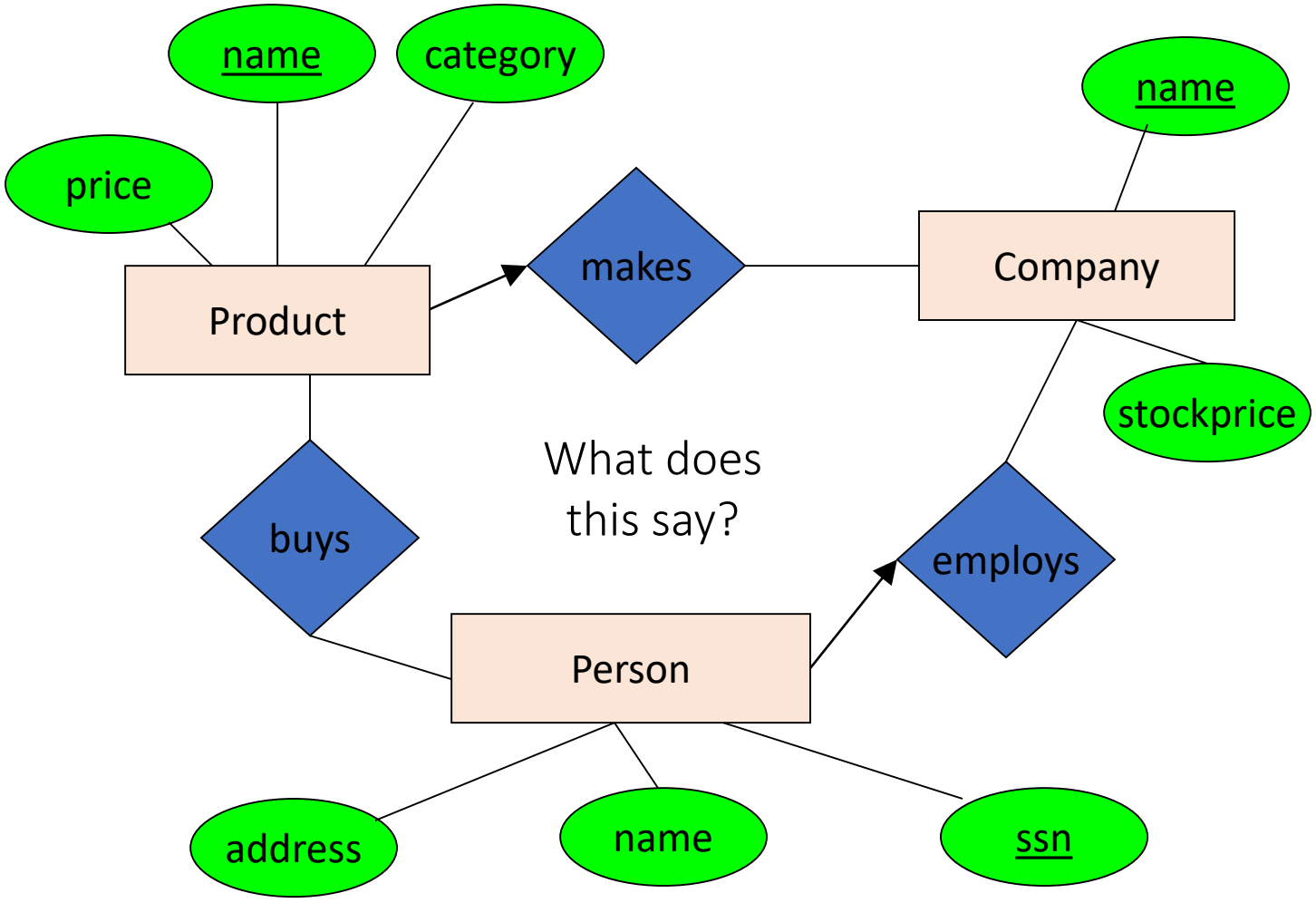
Many-to-many:





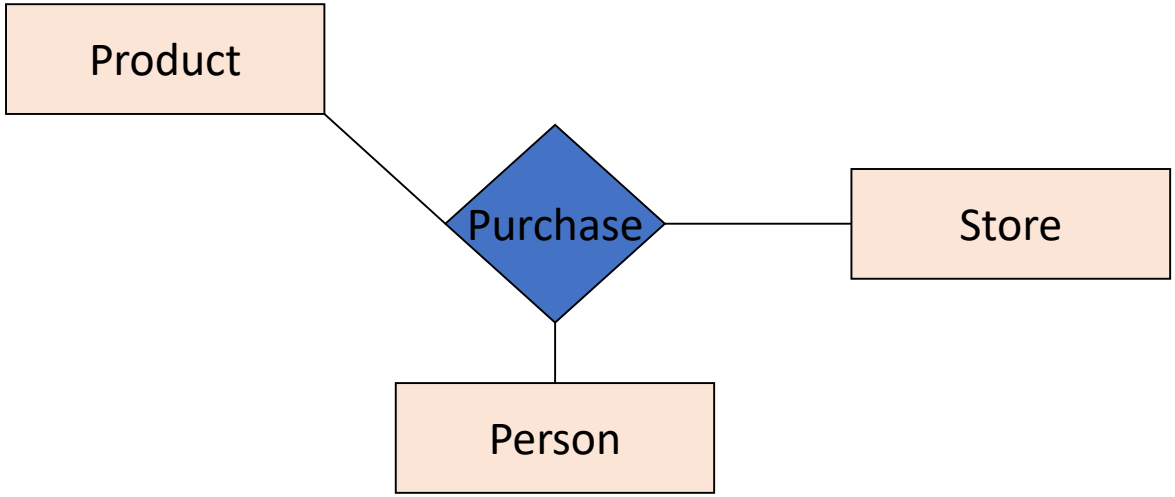
Indicated using arrows

X -> Y means **there exists a function mapping from X to Y** (*recall the definition of a function)*


name
category
price
Product
makes
Company
name
stockprice
What does
this say?
buys
employs
Person
address
name
ssn

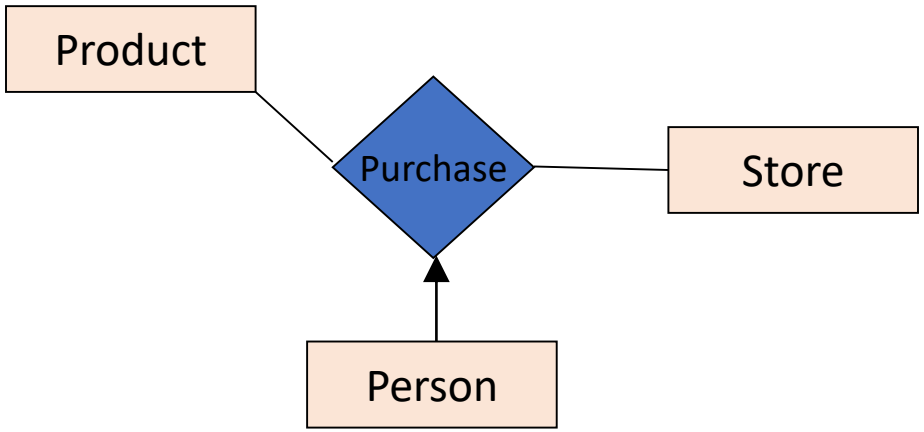
Multi-way Relationships

How do we model a purchase relationship between buyers, products and stores?



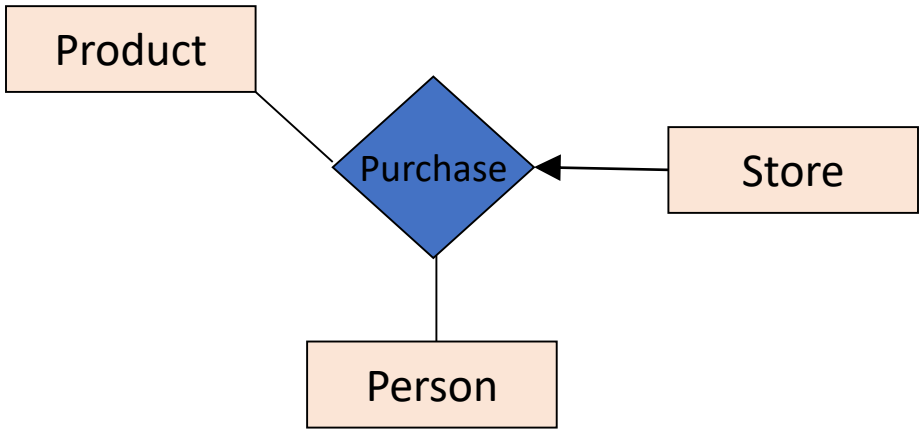
Arrows in Multiway Relationships

Q: What does the arrow mean ?

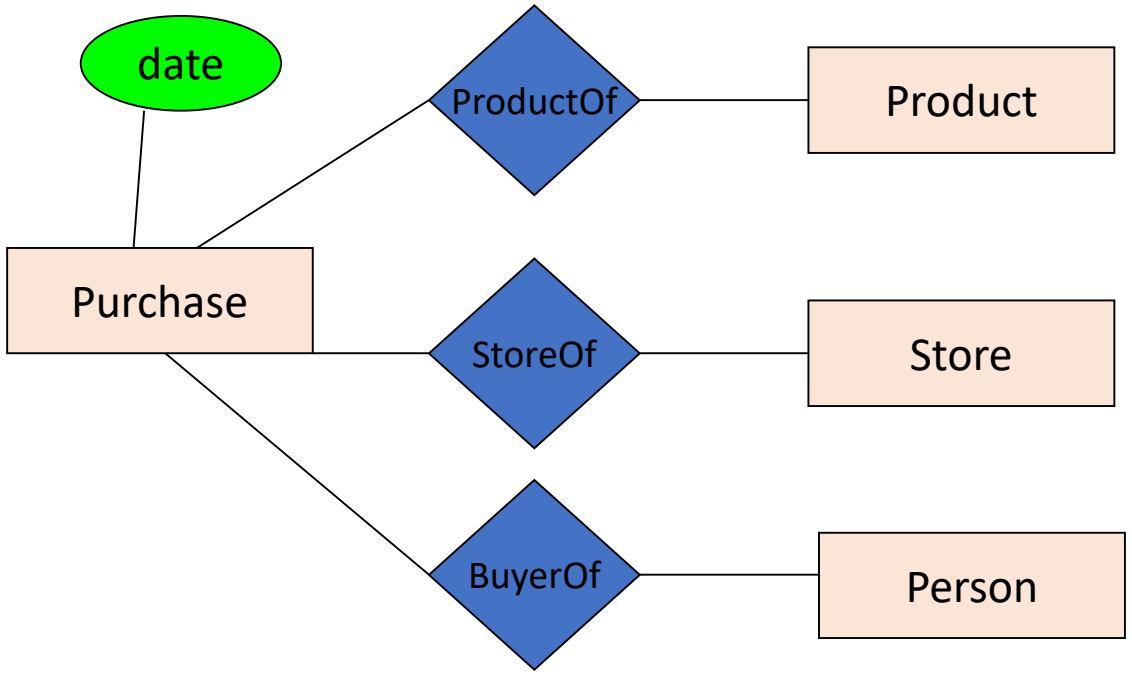


Arrows in Multiway Relationships

Q: What does the arrow mean ?

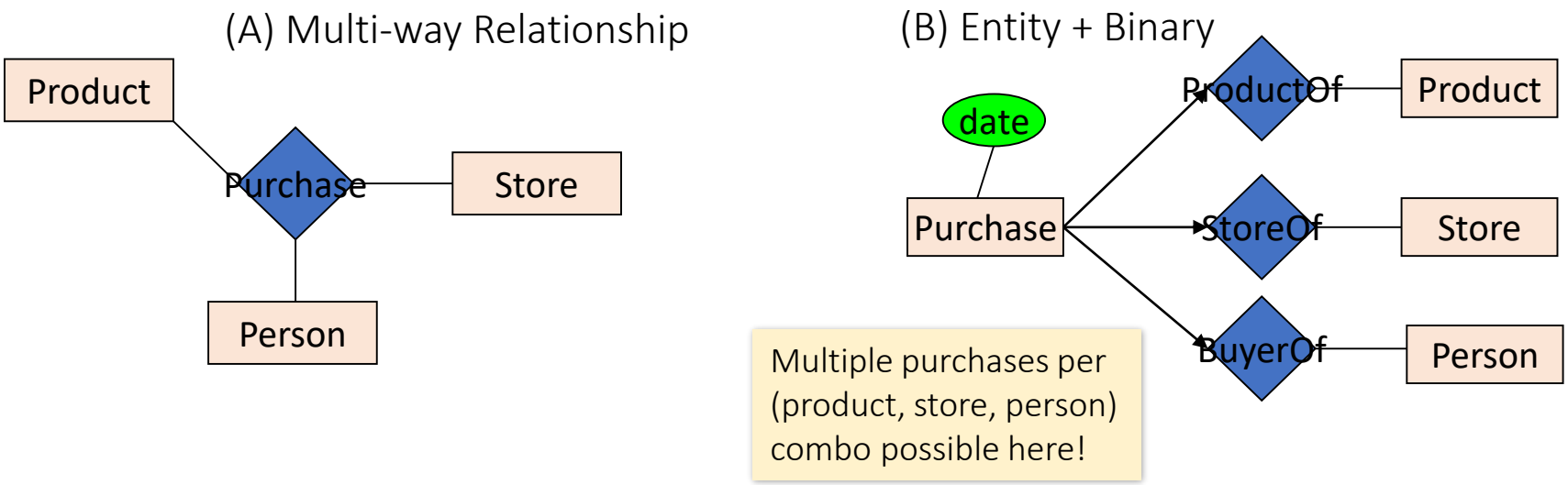


Converting Multi-way Relationships to Binary



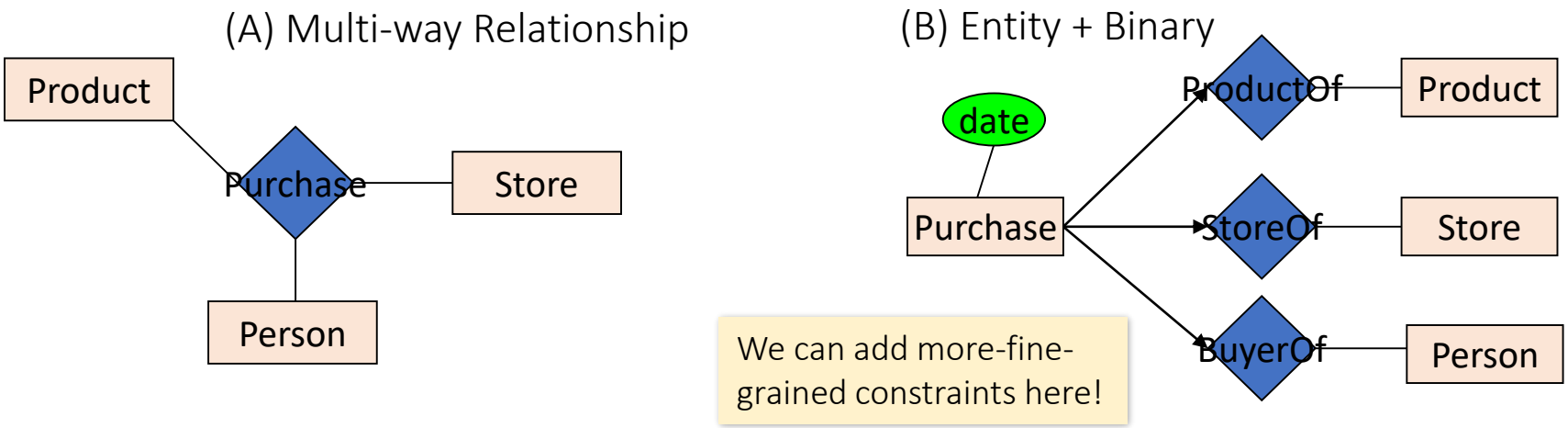
From what we had on previous slide to this - what did we do?

Decision: Multi-way or New Entity + Binary?



- *Covered earlier:* (B) is useful if we want to have multiple instances of the "relationship" per entity combination

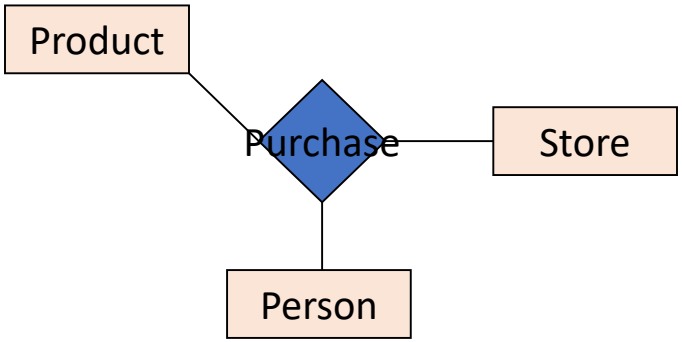
Decision: Multi-way or New Entity + Binary?



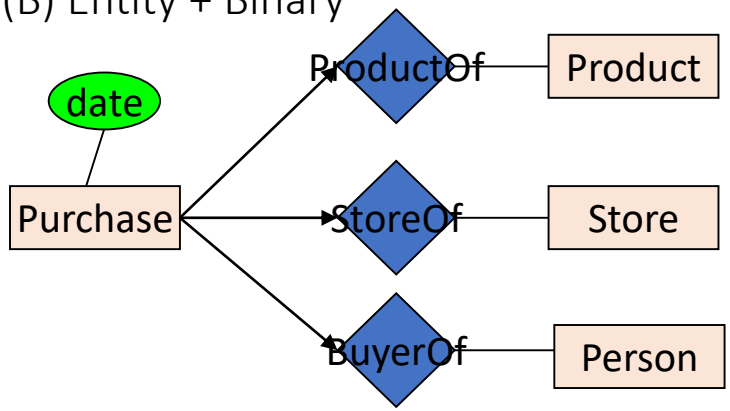
- (B) is also useful when we want to add details (constraints or attributes) to the relationship
  - "A person who shops in only one store"
  - "How long a person has been shopping at a store"

Decision: Multi-way or New Entity + Binary?

(A) Multi-way Relationship



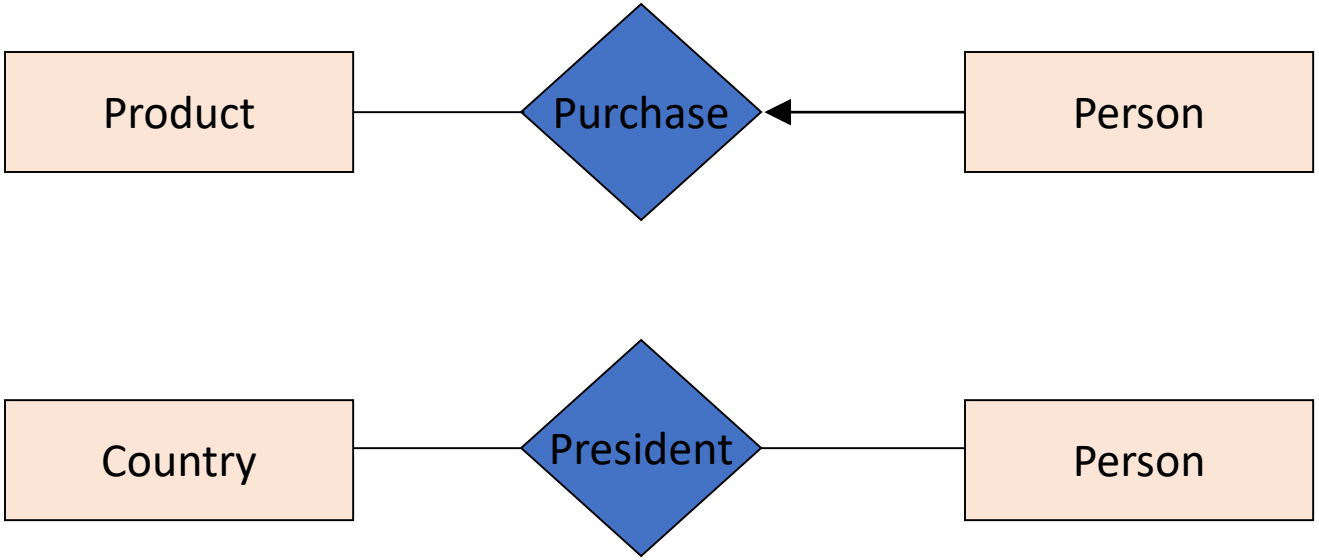
(B) Entity + Binary



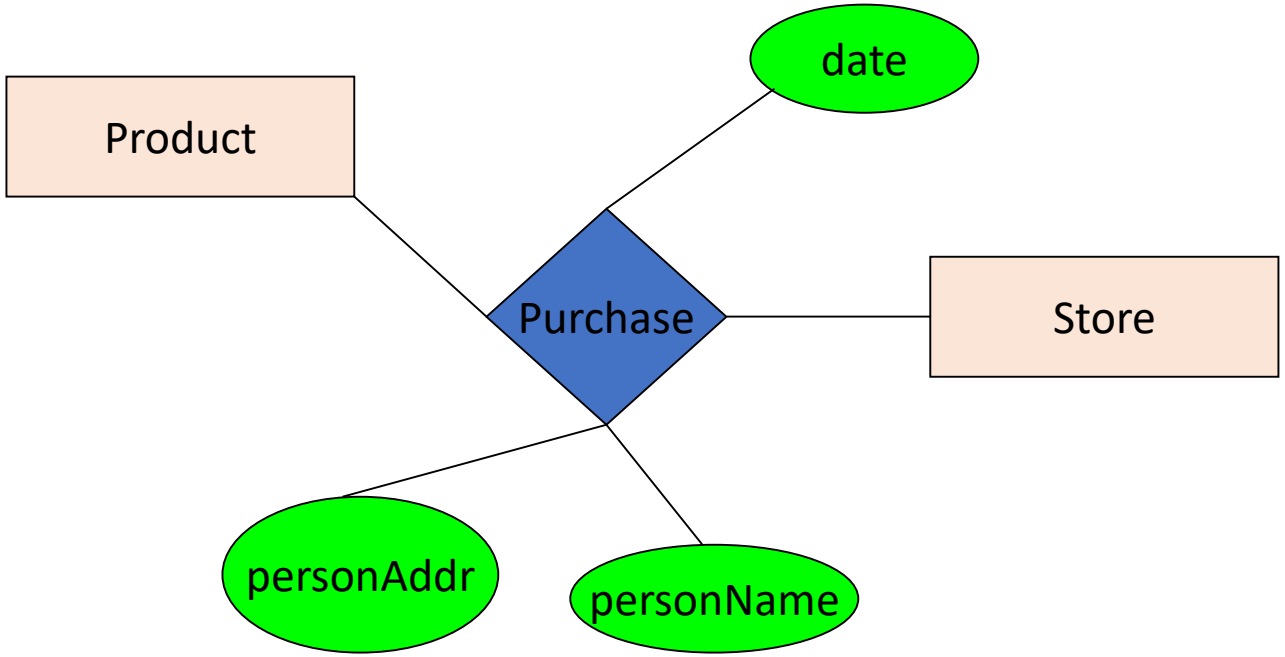
- (A) is useful when a relationship really is between multiple entities
  - *Ex: A three-party legal contract*

3. Design Principles

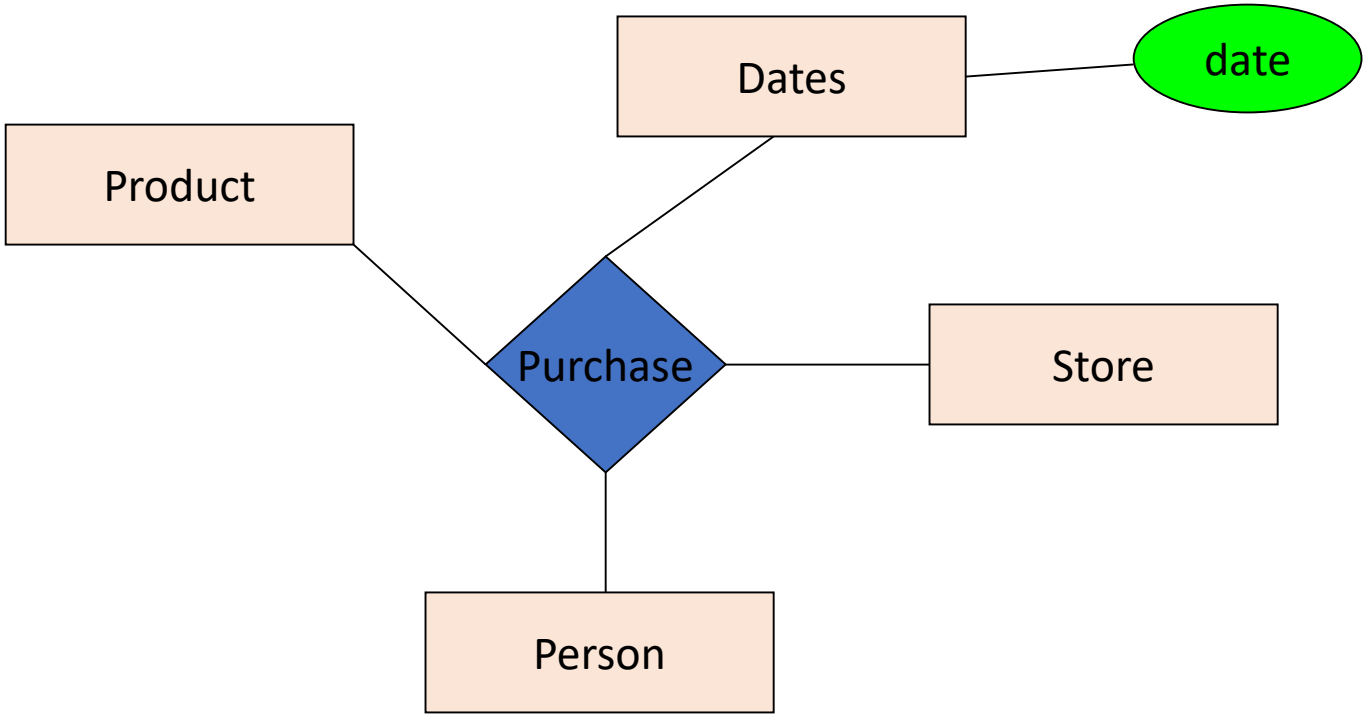
What's wrong with these examples?



Design Principles: What's Wrong?

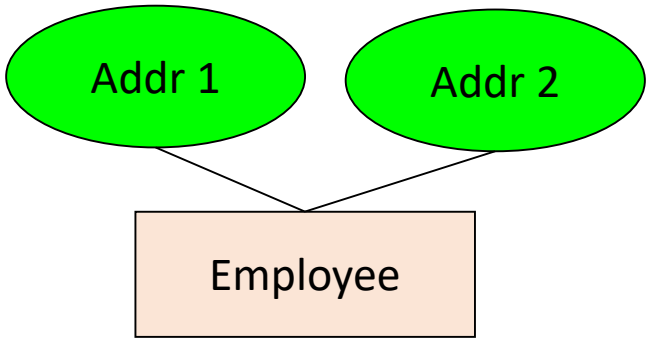


Design Principles: What's Wrong?

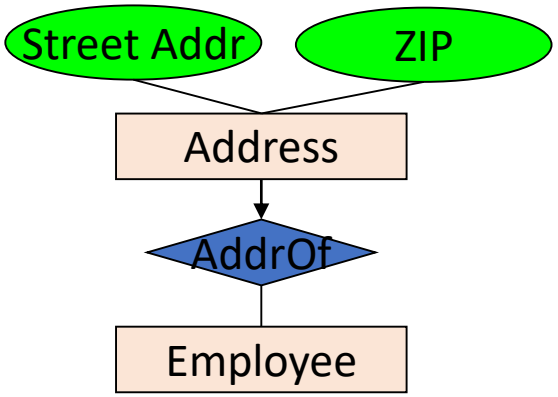


Examples: Entity vs. Attribute

Should address (A) be an attribute?

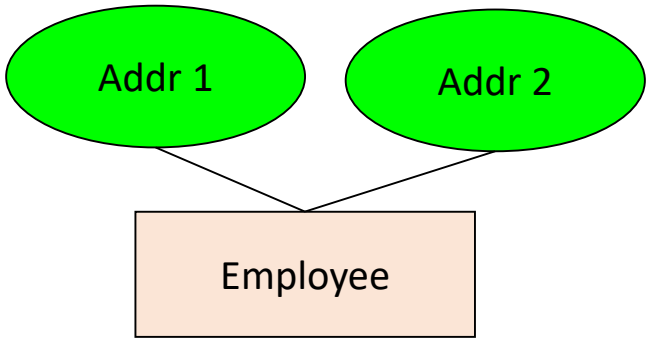


Or (B) be an entity?

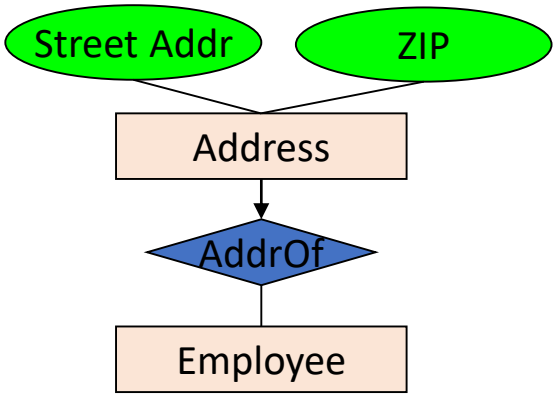


Examples: Entity vs. Attribute

Should address (A) be an attribute?



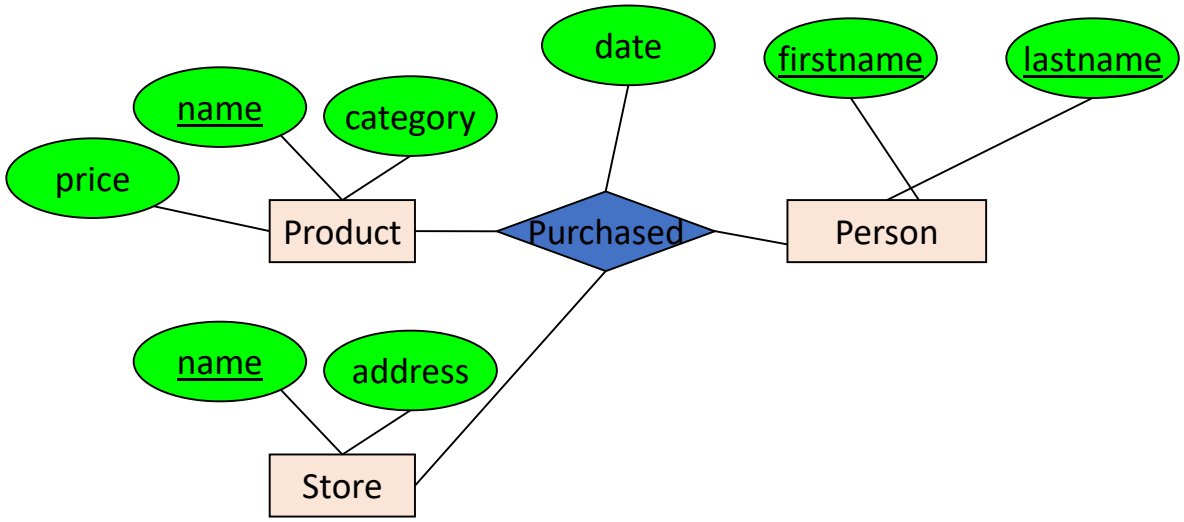
Or (B) be an entity?



In general, when we want to record several values, we choose new entity

From E/R Diagram to Relational Schema

How do we represent this as a relational schema?



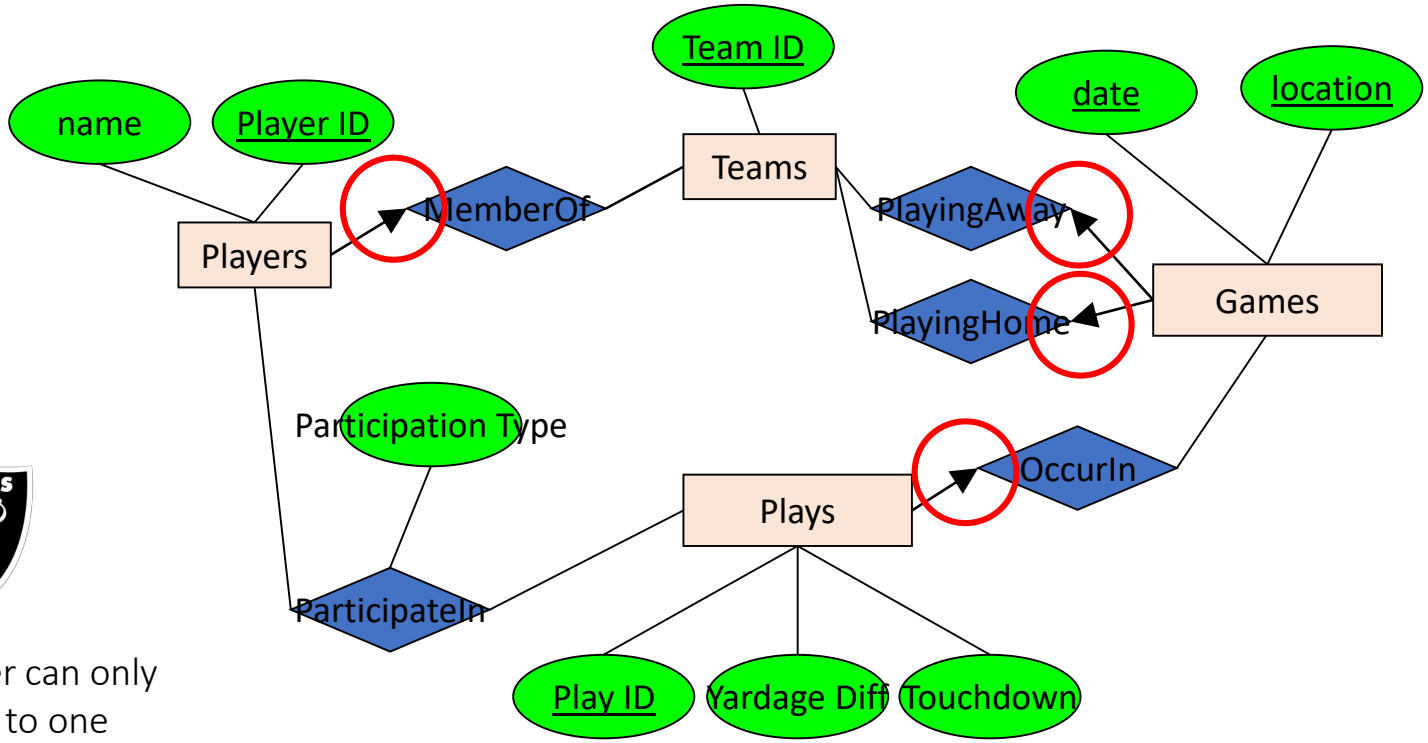
Add arrows to your E/R diagram!

Also make sure to add <u>(new concepts underlined):</u>

A player can only belong to one team, a play can only be in one game, a pass/run..?

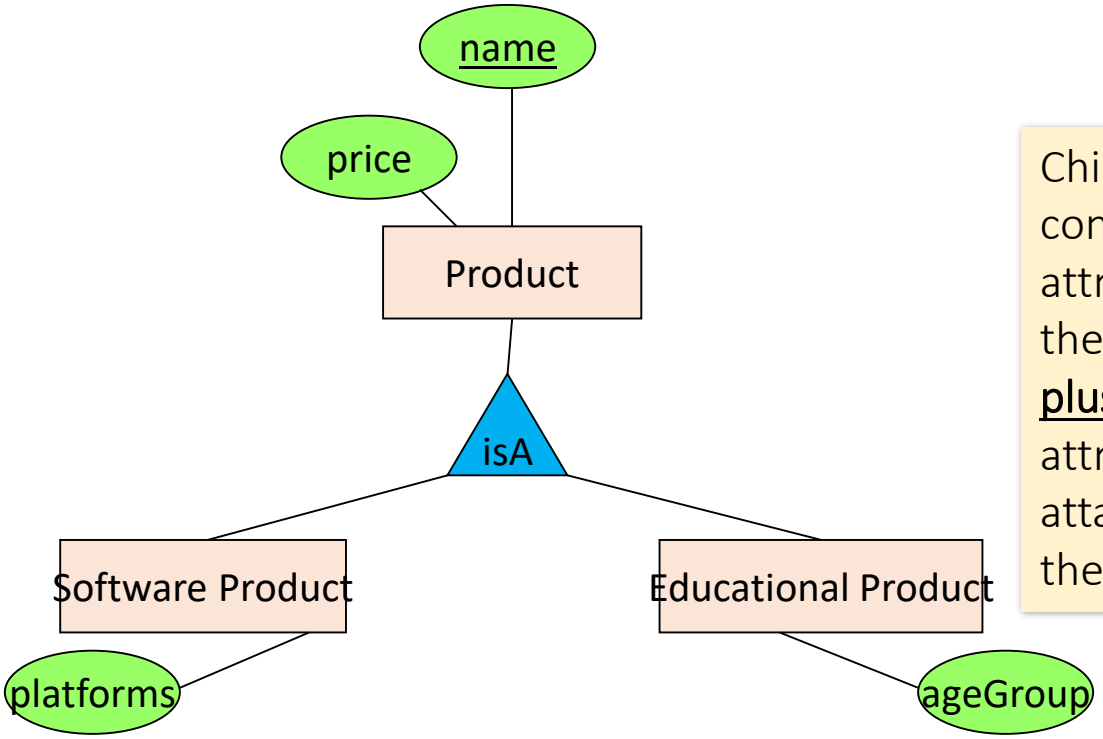
Players can achieve a <u>Personal Record</u> linked to a specific Game and Play

Players have a <u>weight</u> which changes in on vs. off-season



A player can only belong to one team, a play can only be in one game, a pass/run..?

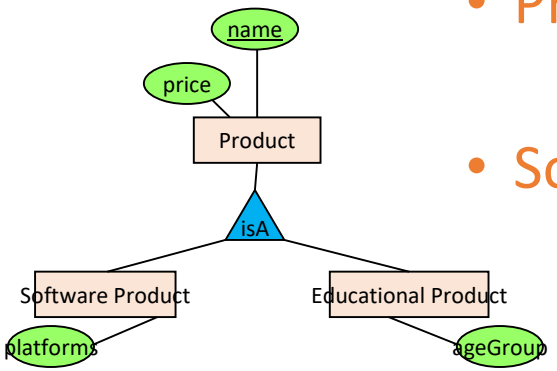
Modeling Subclasses



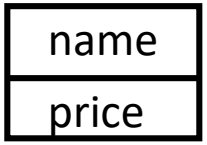
Child subclasses contain all the attributes of *all* of their parent classes **plus** the new attributes shown attached to them in the E/R diagram

Understanding Subclasses

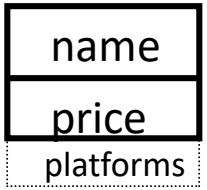
- Think in terms of records; ex:



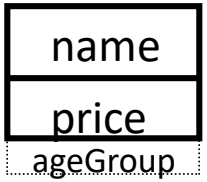
- Product



- SoftwareProduct



- EducationalProduct

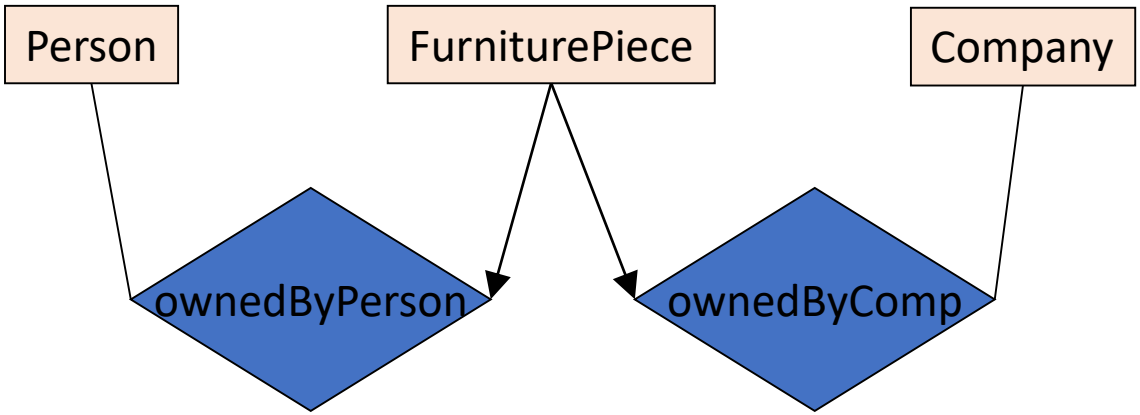


Child subclasses contain all the attributes of *all* of their parent classes **plus** the new attributes shown attached to them in the E/R diagram

Modeling Union Types with Subclasses

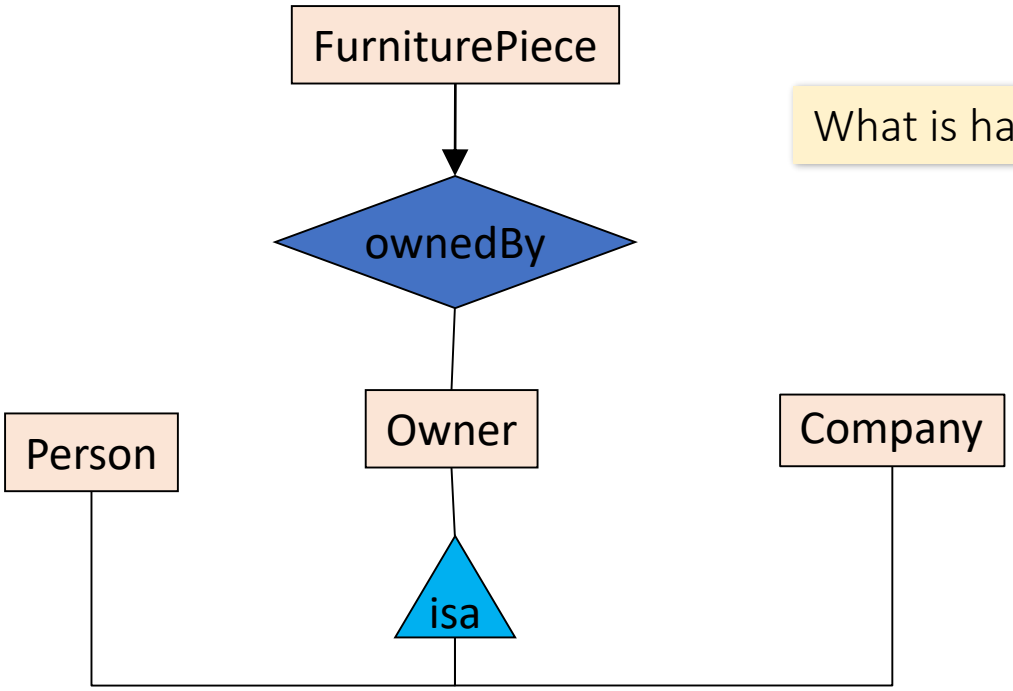
Say: each piece of furniture is owned either by a person, or by a company

Solution 1. Acceptable, but imperfect (What's wrong ?)



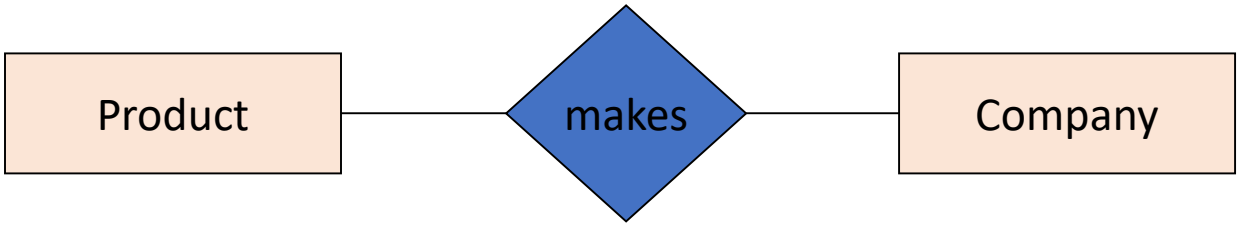
Modeling Union Types with Subclasses

Solution 2: better (though more laborious)

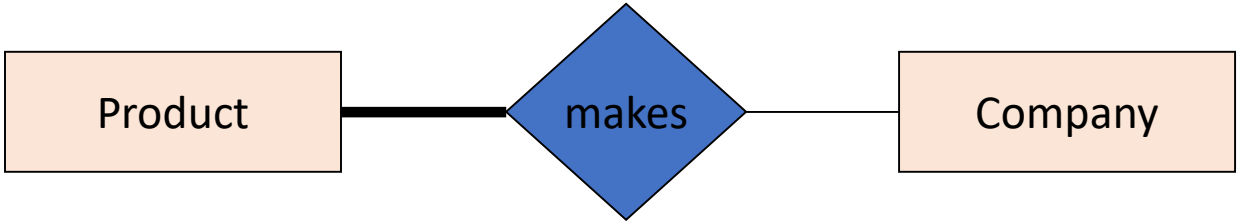


What is happening here?

Participation Constraints: Partial v. Total

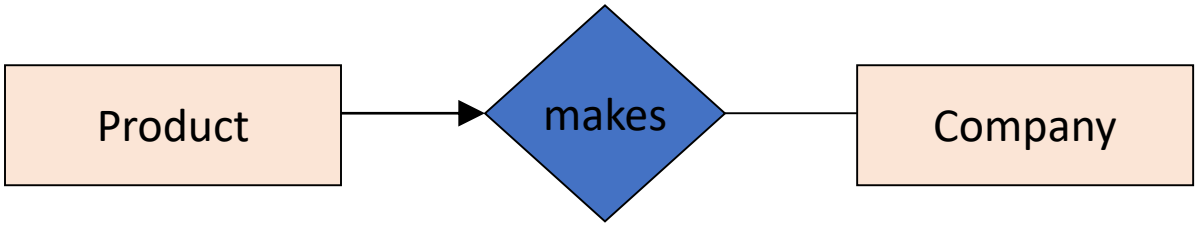


Are there products made by no company?
Companies that don't make a product?

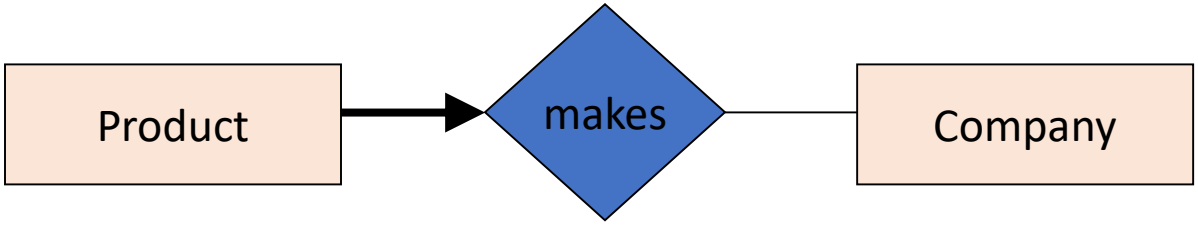


Bold line indicates *total participation* (i.e. here: all products are made by a company)

Referential Integrity Constraints



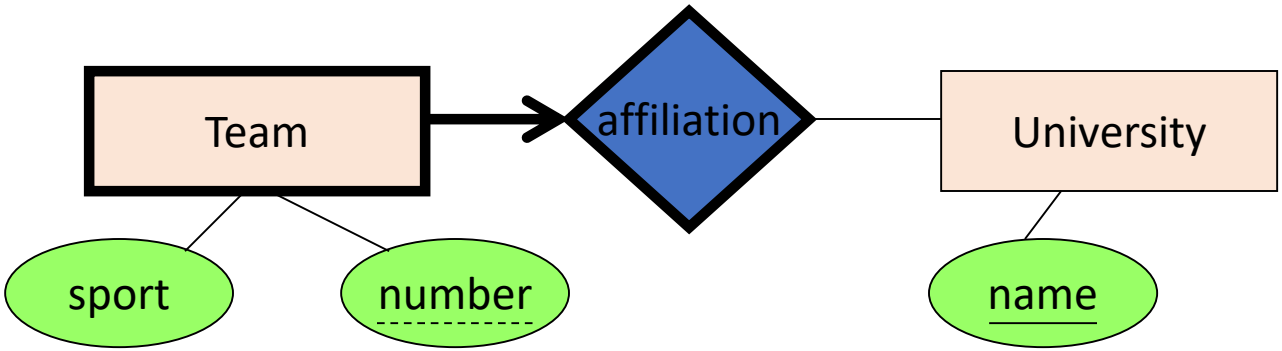
Each product made by at most one company.
Some products made by no company?



Each product made by *exactly* one company.

Weak Entity Sets

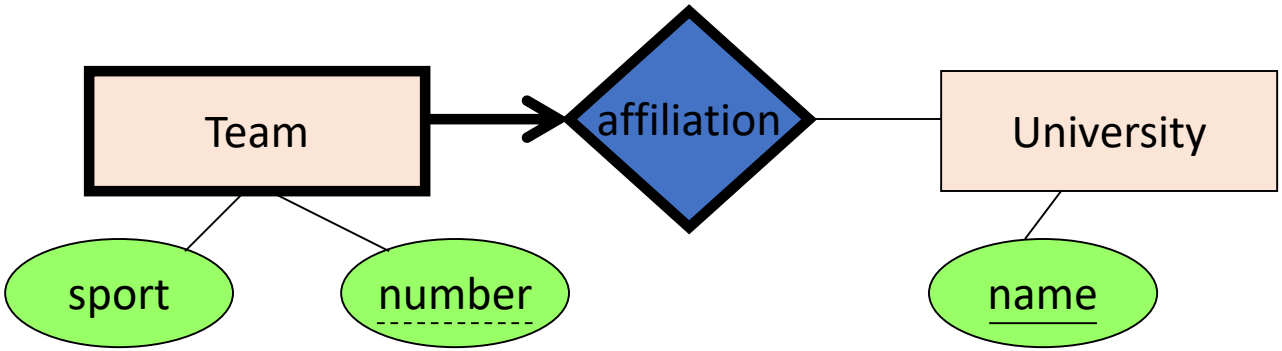
Entity sets are *weak* when their key comes from other classes to which they are related.



"Football team" v. "***The Stanford*** Football team" *(E.g., Berkeley has a football team too, sort of)*

Weak Entity Sets

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



- number is a *partial key*. (denote with dashed underline).
- University is called the *identifying owner*.
- Participation in affiliation must be total. Why?