ER to Relations

Lecture #3

An Example of a Relation

Table name

Attribute names

Products:

Name	Price	Category	Manufacturer
gizmo	$19.99	gadgets	GizmoWorks
Power gizmo	$29.99	gadgets	GizmoWorks
SingleTouch	$149.99	photography	Canon
MultiTouch	$203.99	household	Hitachi

tuples

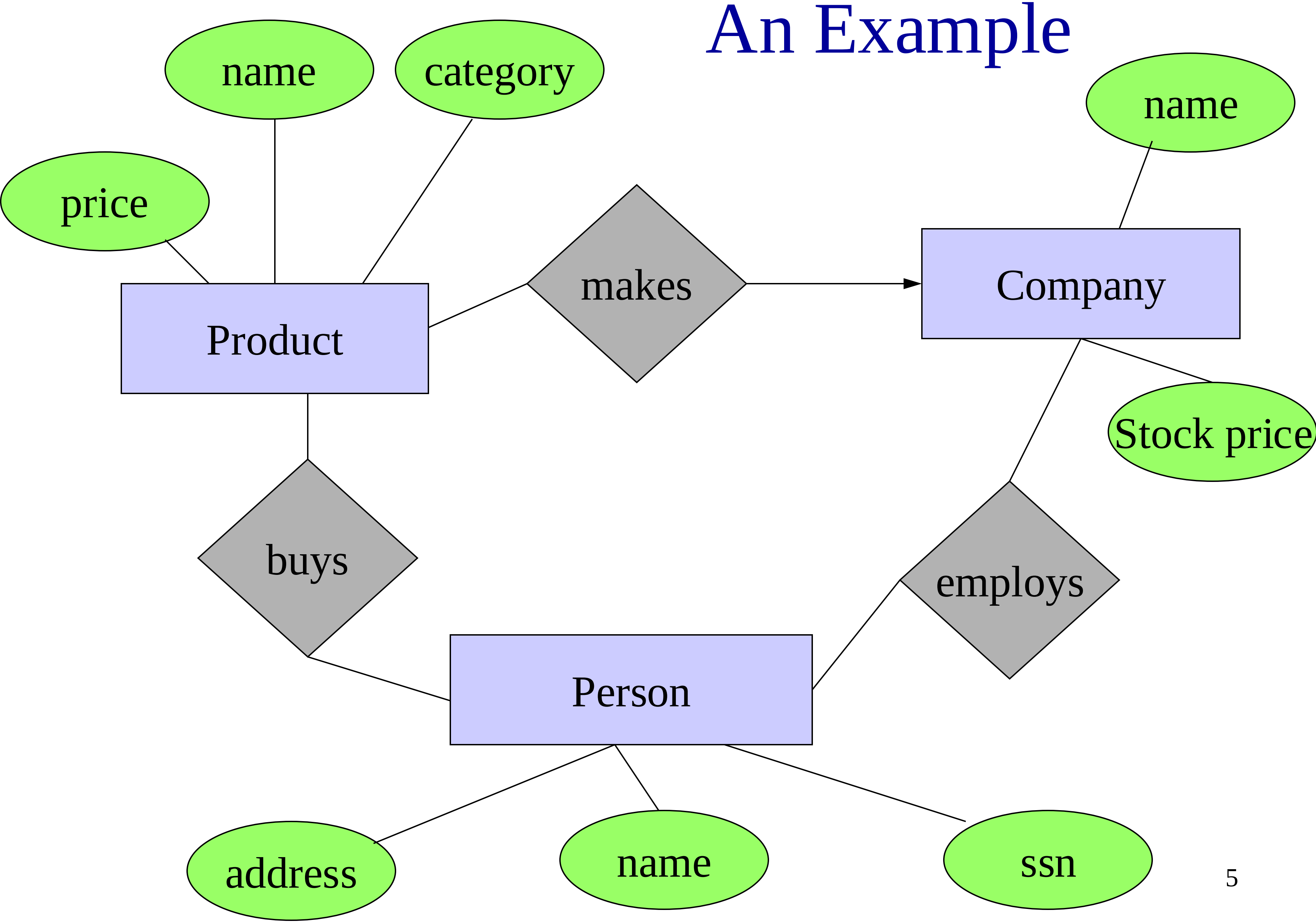
Domains

- Each attribute has a type
- Must be atomic type
- Called *domain*
- Examples:
  - Integer
  - String
  - Real
  - …

Translating ER Diagram to Rel. Design

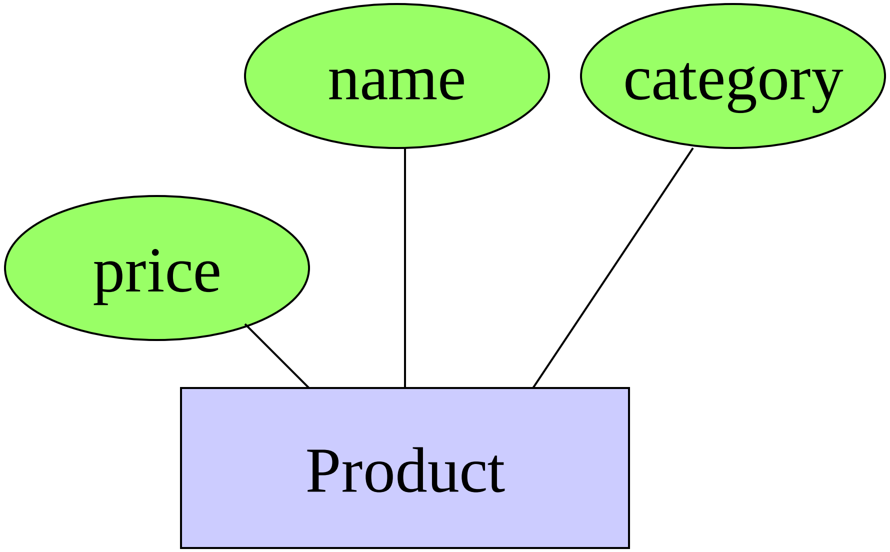
- Basic cases
  - entity set E = relation with attributes of E
  - relationship R = relation with attributes being keys of related entity sets + attributes of R
- Special cases
  - combining two relations
  - translating weak entity sets
  - translating is-a relationships and subclasses

An Example


name
category
price
Product
makes
Company
name
Stock price
buys
employs
Person
address
name
ssn

Basic cases ...

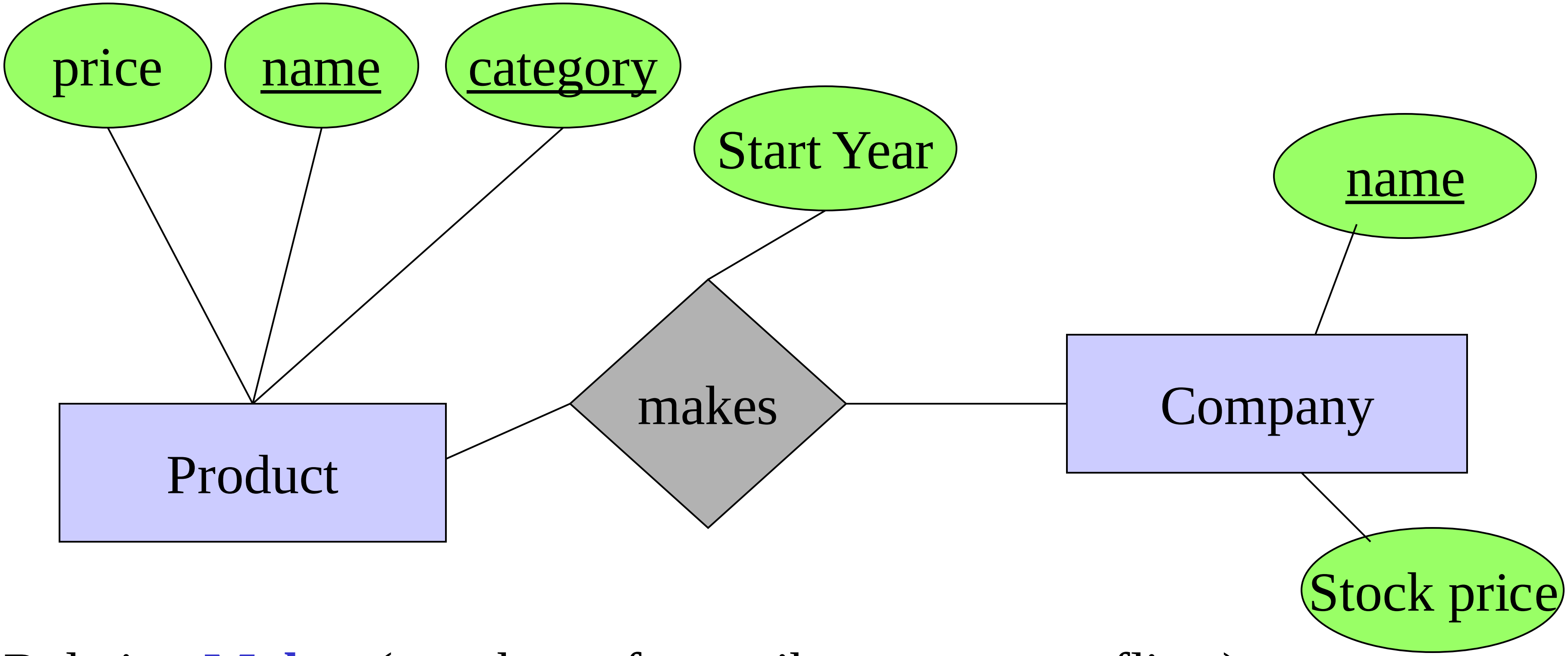
Entity Sets to Relations



Product:

Name	Category	Price
gizmo	gadgets	$19.99

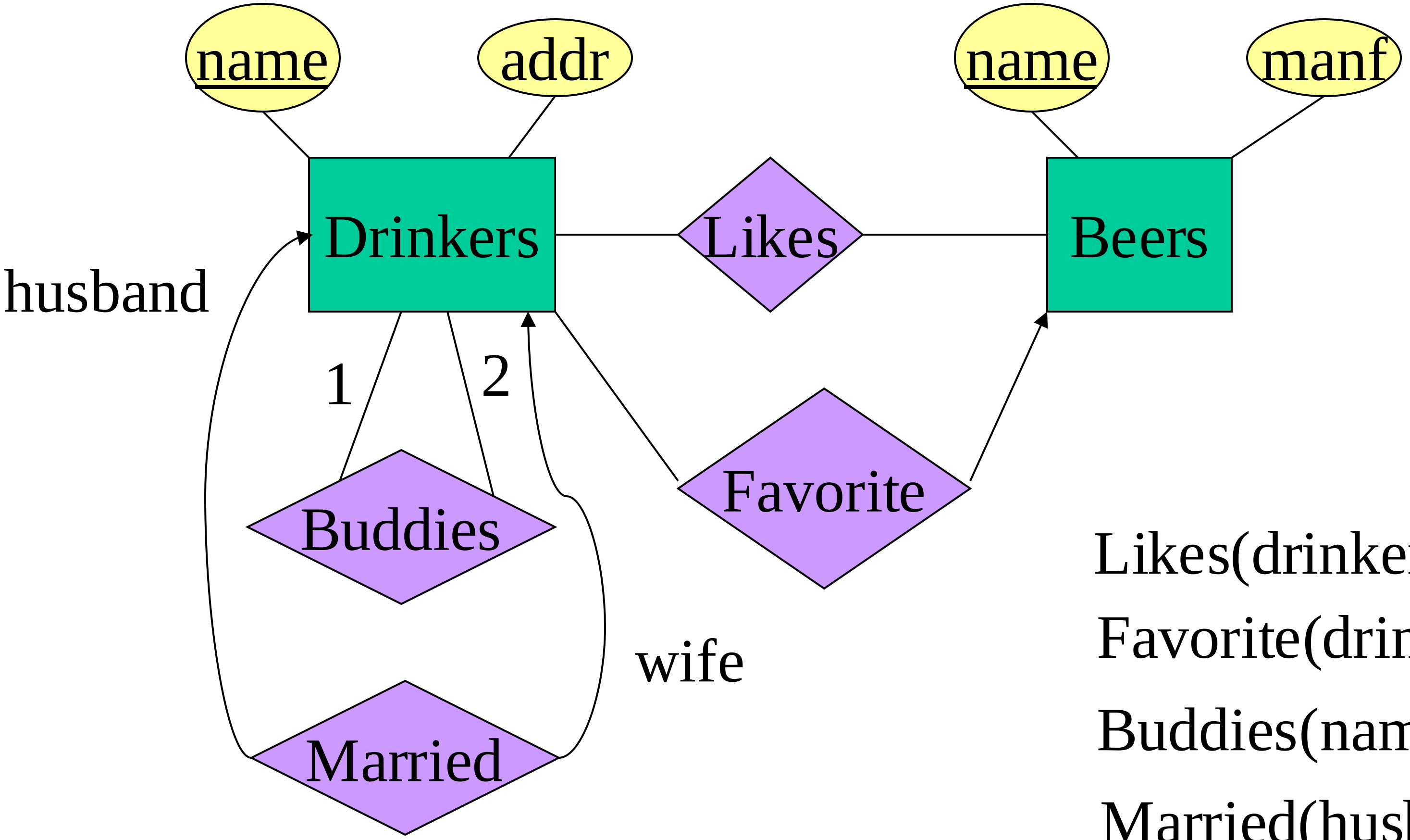
Relationships to Relations



Relation **Makes** (watch out for attribute name conflicts)

Product-name	Product-Category	Company-name	Start-year
gizmo	gadgets	gizmoWorks	1963

Relationship to Relation: Another Example



Likes(drinker, beer)

Favorite(drinker, beer)

Buddies(name1, name2)

Married(husband, wife)

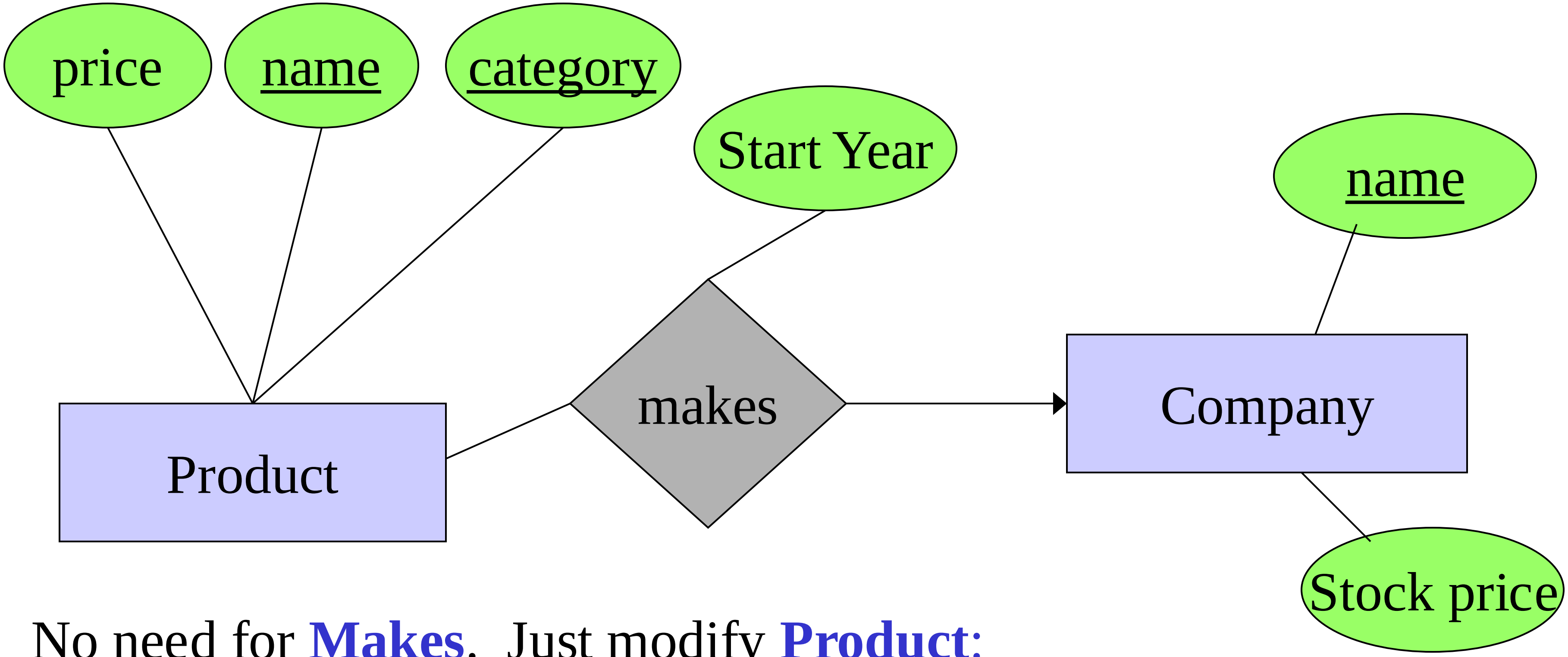
Special cases:

1) many-one relations

2) weak entity sets

3) is-a cases

Combining Two Relations



No need for **Makes**. Just modify **Product**:

name	category	price	StartYear	companyName
gizmo	gadgets	19.99	1963	gizmoWorks

Combining Relations

- It is OK to combine the relation for an entity-set *E* with the relation *R* for a many-one relationship from *E* to another entity set.

- Example: Drinkers(name, addr) and Favorite(drinker, beer) combine to make Drinker1(name, addr, favoriteBeer).

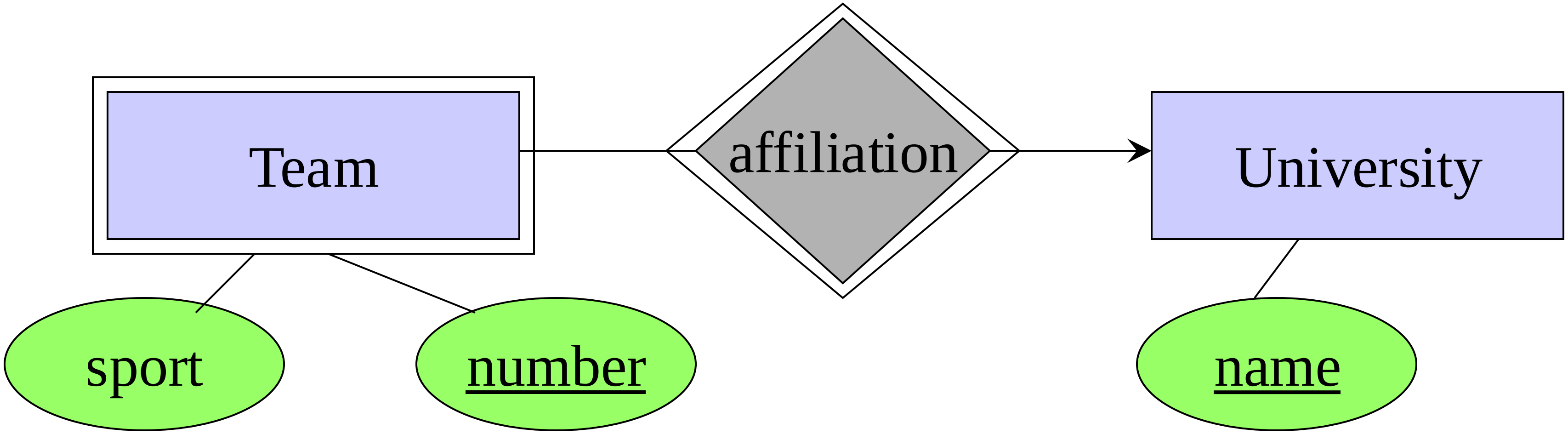
Risk with Many-Many Relationships

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

name	addr	beer
Sally	123 Maple	Bud
Sally	123 Maple	Miller

Redundancy

Handling Weak Entity Sets



Relation Team:

Sport	Number	Affiliated University
mud wrestling	15	Montezuma State U.

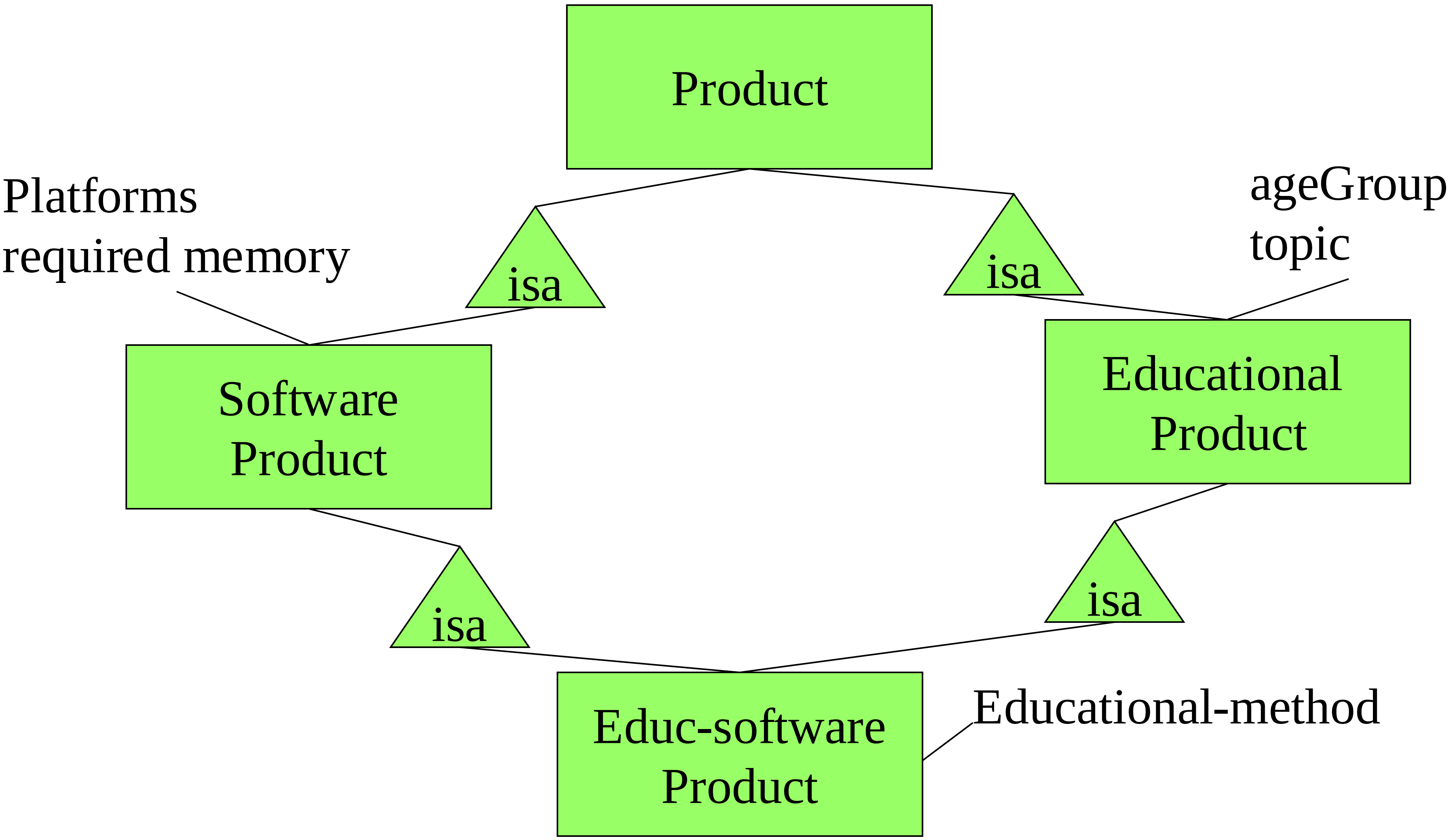
- need all the attributes that contribute to the key of Team
- don’t need a separate relation for Affiliation. (why ?)

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 (double-diamond) relationship is redundant and yields no relation.

Translating Subclass Entities



Option #1: the OO Approach

4 tables: each object can only belong to a single table

Product(name, price, category, manufacturer)

EducationalProduct(name, price, category, manufacturer,
ageGroup, topic)

SoftwareProduct(name, price, category, manufacturer,
platforms, requiredMemory)

EducationalSoftwareProduct(name, price, category, manufacturer,
ageGroup, topic, platforms,
requiredMemory)

All names are distinct

Option #2: the E/R Approach

Product(name, price, category, manufacturer)

EducationalProduct(name, ageGroup, topic)

SoftwareProduct(name, platforms, requiredMemory)

No need for a relation EducationalSoftwareProduct

Unless, it has a specialized attribute:
EducationalSoftwareProduct(name, educational-method)

Same name may appear in several relations

Option #3: The Null Value Approach

Have one table:

Product (name, price, manufacturer, age-group, topic, platforms,
required-memory, educational-method)

Some values in the table will be NULL, meaning that the attribute not make sense for the specific product.

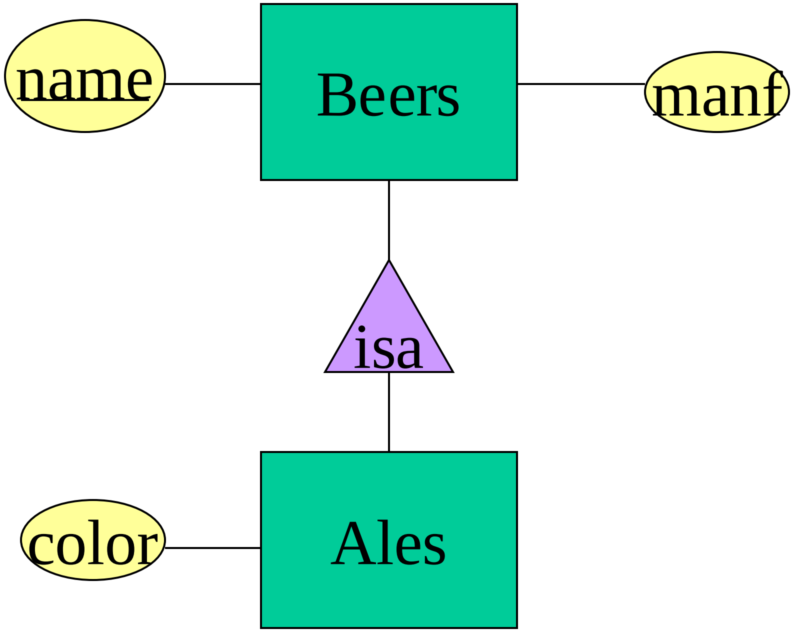
Too many meanings for NULL

Translating Subclass Entities: The Rules

Three approaches:

1. *Object-oriented* : each entity belongs to exactly one class; create a relation for each class, with all its attributes.
2. *E/R style* : create one relation for each subclass, with only the key attribute(s) and attributes attached to that E.S.; entity represented in all relations to whose subclass/E.S. it belongs.
3. *Use nulls* : create one relation; entities have null in attributes that don’t belong to them.

Example



Object-Oriented

name	manf
Bud	Anheuser-Busch

Beers

name	manf	color
Summerbrew	Pete's	dark

Ales

E/R Style

name	manf
Bud	Anheuser-Busch
Summerbrew	Pete's

Beers

name	color
Summerbrew	dark

Ales

Using Nulls

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

Beers

Comparisons

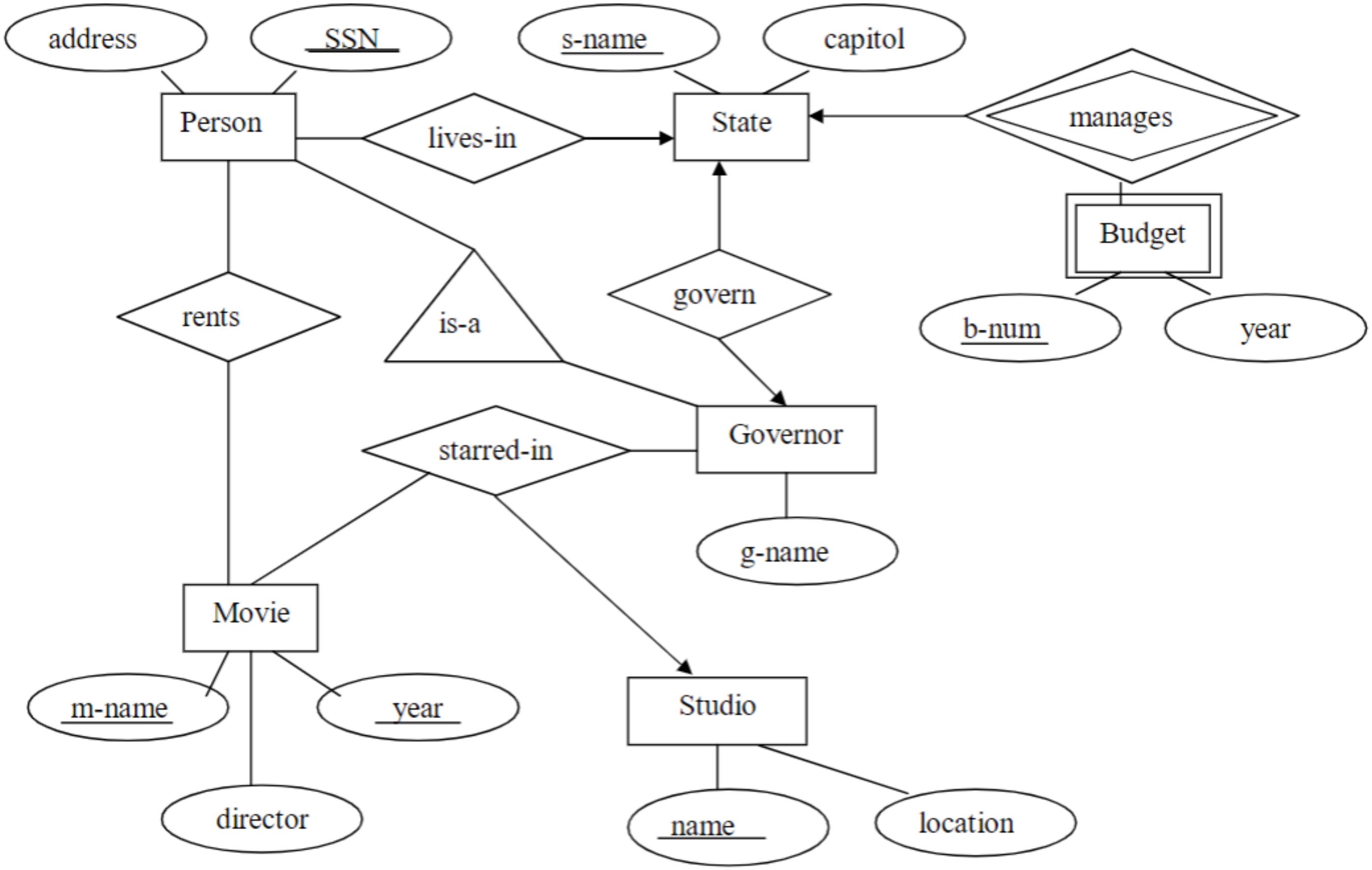
- O-O approach good for queries like “find the color of ales made by Pete’s.”
  - Just look in Ales relation.
- E/R approach good for queries like “find all beers (including ales) made by Pete’s.”
  - Just look in Beers relation.
- Using nulls saves space unless there are *lots* of attributes that are usually null.

Translation Review

- Basic cases
  - entity to table, relation to table
  - selecting attributes based on keys
- Special cases
  - many-one relation can be merged
  - merging many-many is dangerous
  - translating weak entity sets
  - translating isa hierarchy
    - 3 choices, with trade-offs

2. (15 points) ER and translation to relational model.

Translate this ER diagram into a relational schema. For each relation in your schema, specify the key of that relation. In translating a subclass hierarchy, use the ER style translation.


address
SSN
s-name
capitol
Person
lives-in
State
manages
Budget
rents
is-a
govern
b-num
year
Governor
starred-in
g-name
Movie
m-name
year
director
Studio
name
location