# Lecture 07
# Data Modeling: E/R Diagrams

Monday, October 14, 2002

1

## Outline

- E/R diagrams (Chapter 2)
- From E/R diagrams to relations (3.2, 3.3)

2

## Database Design

- Why do we need it?
  - Agree on structure of the database before deciding on a particular implementation.
- Consider issues such as:
  - What entities to model
  - How entities are related
  - What constraints exist in the domain
  - How to achieve ***good*** designs

3

## Database Design Formalisms

1. Object Definition Language (ODL):
   - Closer in spirit to object-oriented models
   - Will not cover in class
2. Entity/Relationship model (E/R):
   - More relational in nature.
   - Very widely used

- Both can be translated (semi-automatically) to relational schemas
- ODL to OO-schema: direct transformation (C++ or Smalltalk based system).

4

## Entity / Relationship Diagrams

Objects ⟶ entities
Classes ⟶ entity sets

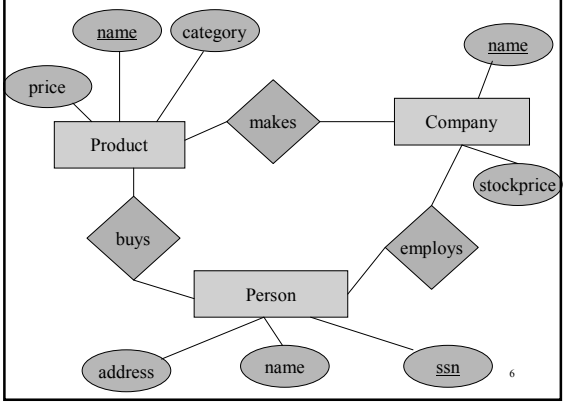
Attributes are like in ODL.

Relationships: like in ODL except

- first class citizens (not associated with classes)

- not necessarily binary

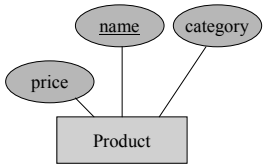
5




6

## Keys in E/R Diagrams

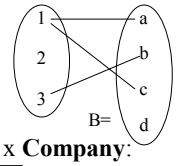
- Every entity set must have a key



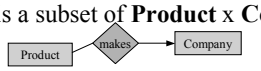

7

## What is a Relation ?

- A mathematical definition:
  - if A, B are sets, then a relation R is a subset of A x B
- A={1,2,3}, B={a,b,c,d},
  R = {(1,a), (1,c), (3,b)}




- **makes** is a subset of **Product** x **Company**:



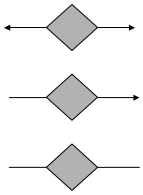

8

## Multiplicity of E/R Relations

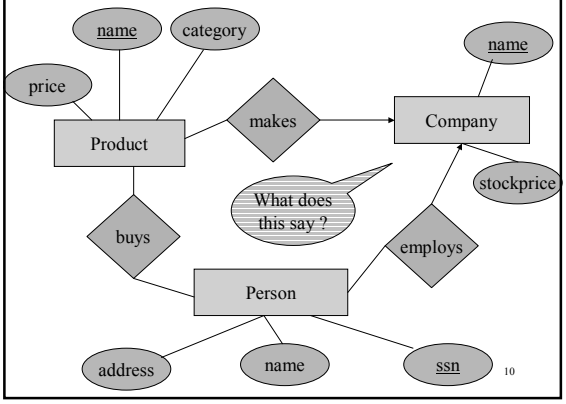
- one-one:

- many-one

- many-many



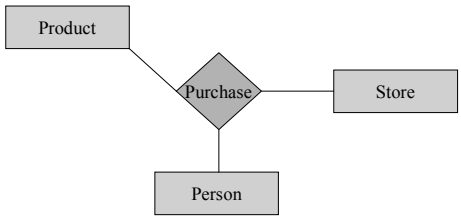
9




10

## Multi-way Relationships

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

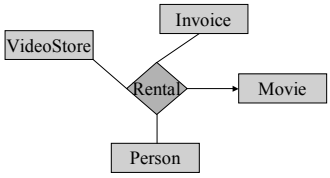



Can still model as a mathematical set (how ?)

11

## Arrows in Multiway Relationships

**Q**: what does the arrow mean ?

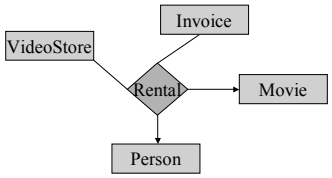



**A**: if I know the store, person, invoice, I know the movie too

12

## Arrows in Multiway Relationships

**Q**: what do these arrow mean ?



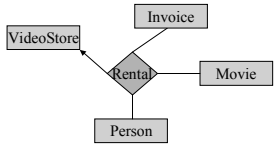

**A**: store, person, invoice determines movie and store, invoice, movie determines person

13

## Arrows in Multiway Relationships

**Q**: how do I say: "invoice determines store" ?

**A**: no good way; best approximation:

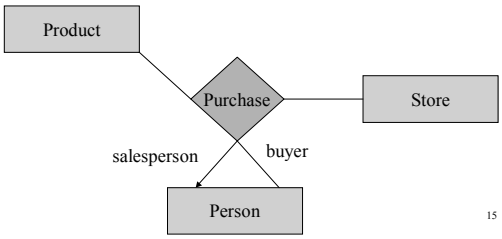



**Q**: Why is this incomplete ?

14

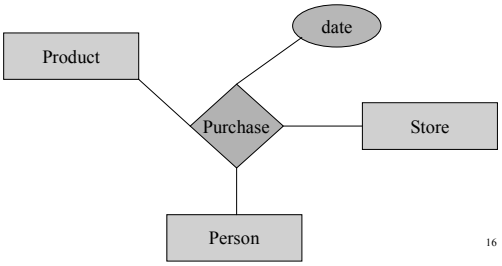
## Roles in Relationships

What if we need an entity set twice in one relationship?



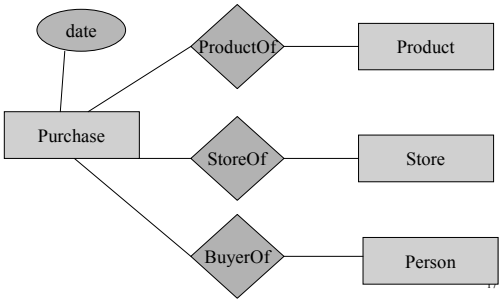

15

## Attributes on Relationships




16

## Converting Multi-way Relationships to Binary



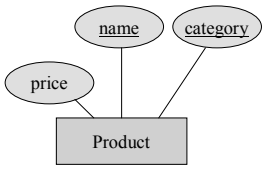

17

## From E/R Diagrams to Relational Schema

- Entity set → relation
- Relationship → relation

18

## Entity Set to Relation

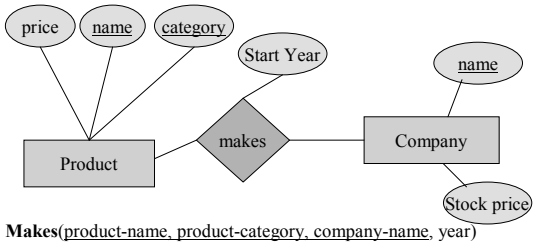



**Product**(name, category, price)

| name | category | price |
|---|---|---|
| gizmo | gadgets | $19.99 |

19

## Relationships to Relations



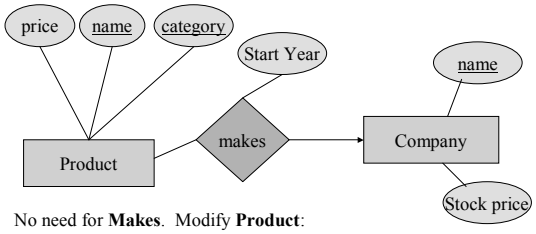

**Makes**(product-name, product-category, company-name, year)

| Product-name | Product-Category | Company-name | Starting-year |
|---|---|---|---|
| gizmo | gadgets | gizmoWorks | 1963 |

(watch out for attribute name conflicts)

20

## Relationships to Relations

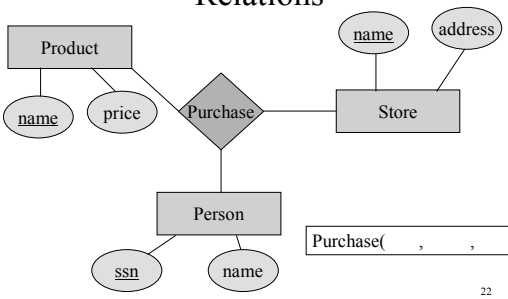



No need for **Makes**. Modify **Product**:

| name | category | price | StartYear | companyName |
|---|---|---|---|---|
| gizmo | gadgets | 19.99 | 1963 | gizmoWorks |

21

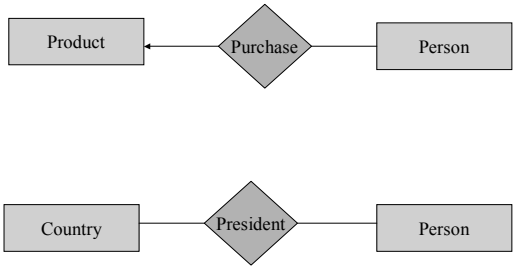
## Multi-way Relationships to Relations




22

## 3. Design Principles

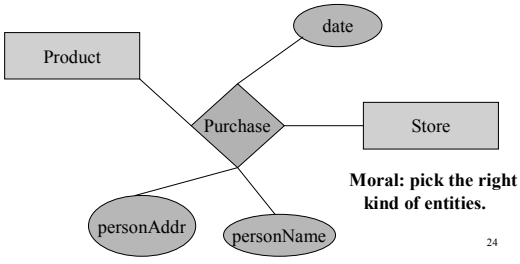
**What's wrong?**




**Moral: be faithful!**

23

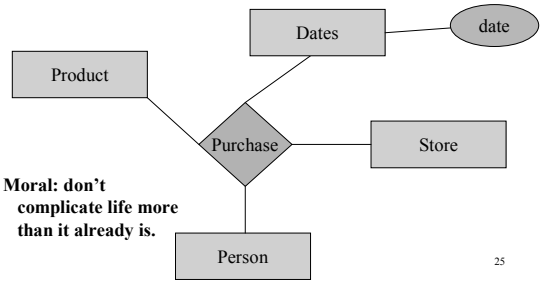
## Design Principles: What's Wrong?




**Moral: pick the right kind of entities.**

24

## Design Principles: What's Wrong?

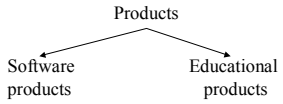



**Moral: don't complicate life more than it already is.**

25

## Modeling Subclasses

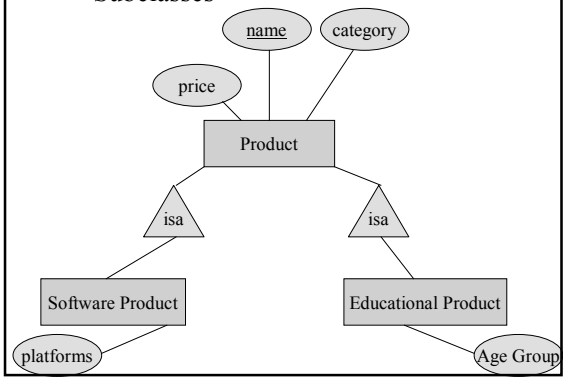
Some objects in a class may be special
- define a new class
- better: define a *subclass*




So --- we define subclasses in E/R

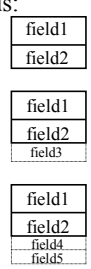
26

## Subclasses



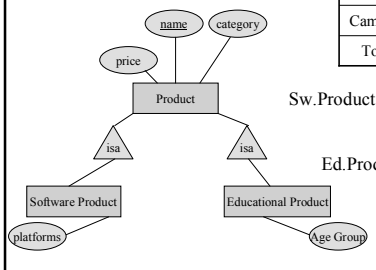

## Understanding Subclasses

- Think in terms of records:
  - Product
  - SoftwareProduct
  - EducationalProduct




28

## Subclasses to Relations




Product

| Name | Price | Category |
|---|---|---|
| Gizmo | 99 | gadget |
| Camera | 49 | photo |
| Toy | 39 | gadget |

Sw.Product

| Name | platforms |
|---|---|
| Gizmo | unix |

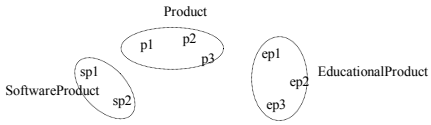
Ed.Product

| Name | Age Group |
|---|---|
| Gizmo | todler |
| Toy | retired |

29

## Difference between ODL and E/R inheritance

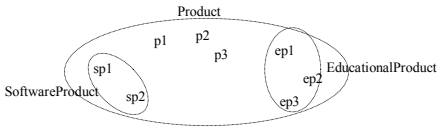
- ODL: classes are disjoint




30

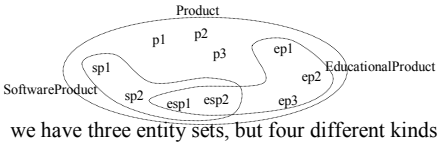
## Difference between ODL and E/R inheritance

- E/R: entity sets overlap





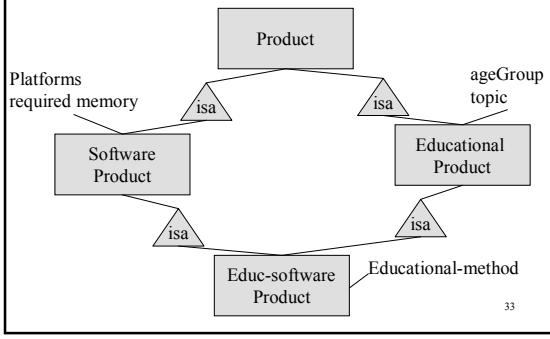

- No need for multiple inheritance in E/R




- we have three entity sets, but four different kinds of objects
- Still needed if we want extra attributes

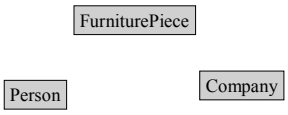


## Modeling Subclass Structure






## Modeling UnionTypes With Subclasses



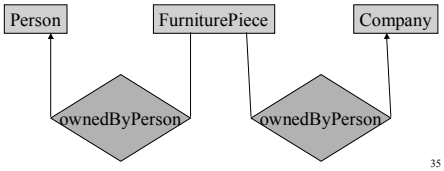

Say: each piece of furniture is owned either by a person, or by a company



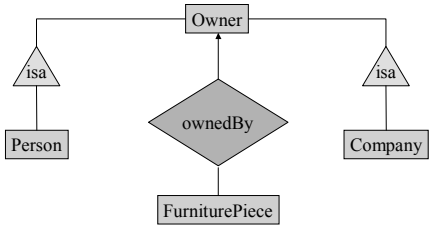
## Modeling Union Types with Subclasses

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

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






## Modeling Union Types with Subclasses

Solution 2: better, more laborious