Lecture 4: The E/R Model

Today’s Lecture

1. E/R Basics: Entities & Relations
   - ACTIVITY: Crayon time!

2. E/R Design considerations
   - ACTIVITY: Crayon time pt. II

3. Advanced E/R Concepts
   - ACTIVITY: E/R Translation

1. E/R Basics: Entities & Relations

Database Design

- **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

- **Several formalisms exist**
  - We discuss one flavor of E/R diagrams

Database Design Process

1. Requirements Analysis

2. Conceptual Design

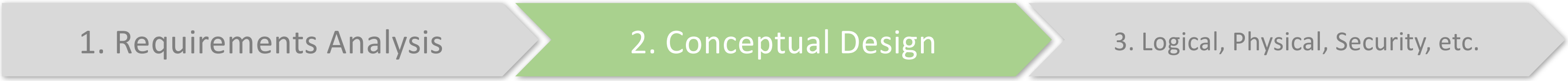
3. Logical, Physical, Security, etc.

1. Requirements analysis

- What is going to be stored?
- How is it going to be used?
- What are we going to do with the data?
- Who should access the data?

Technical and non-technical people are involved

Database Design Process



1. Requirements Analysis

2. Conceptual Design

3. Logical, Physical, Security, etc.

2. Conceptual Design

- A high-level description of the database
- Sufficiently precise that technical people can understand it
- But, not so precise that non-technical people can't participate

This is where E/R fits in.

Database Design Process

1. Requirements Analysis

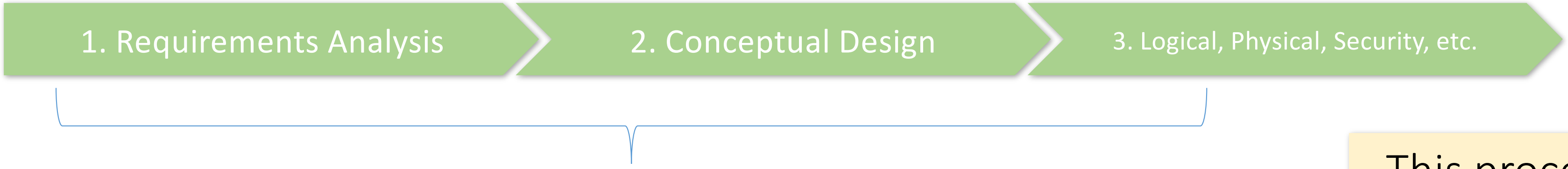
2. Conceptual Design

3. Logical, Physical, Security, etc.

3. More:

- Logical Database Design
- Physical Database Design
- Security Design

Database Design Process

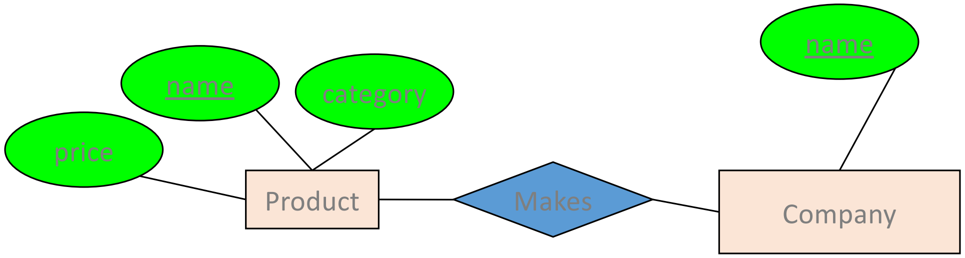


1. Requirements Analysis

2. Conceptual Design

3. Logical, Physical, Security, etc.

E/R Model & Diagrams used



This process is iterated **many** times

E/R is a *visual syntax* for DB design which is ***precise enough*** for technical points, but ***abstracted enough*** for non-technical people

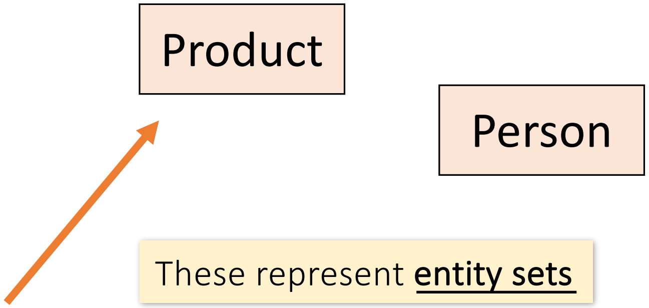
Interlude: Impact of the ER model

- The E/R model is one of the most cited articles in Computer Science
    - *"The Entity-Relationship model – toward a unified view of data"* Peter Chen, 1976

- Used by companies big and small
    - You'll know it soon enough

Entities and Entity Sets

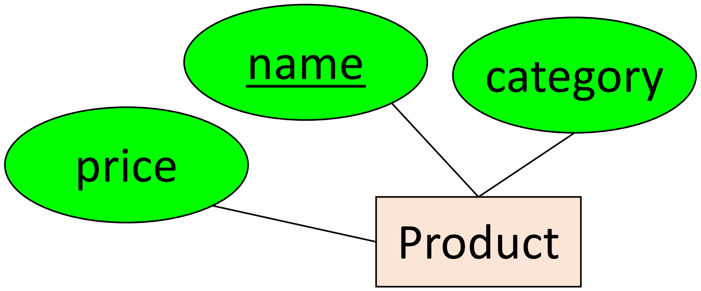
- **Entities** & **entity sets** are the primitive unit of the E/R model
  - Entities are the individual objects, which are members of entity sets
    - Ex: A specific person or product
  - Entity sets are the *classes* or *types* of objects in our model
    - Ex: Person, Product
    - *These are what is shown in E/R diagrams - as rectangles*
    - *Entity sets represent the sets of all possible entities*



These represent entity sets

Entities and Entity Sets

- An entity set has **attributes**
  - Represented by ovals attached to an entity set

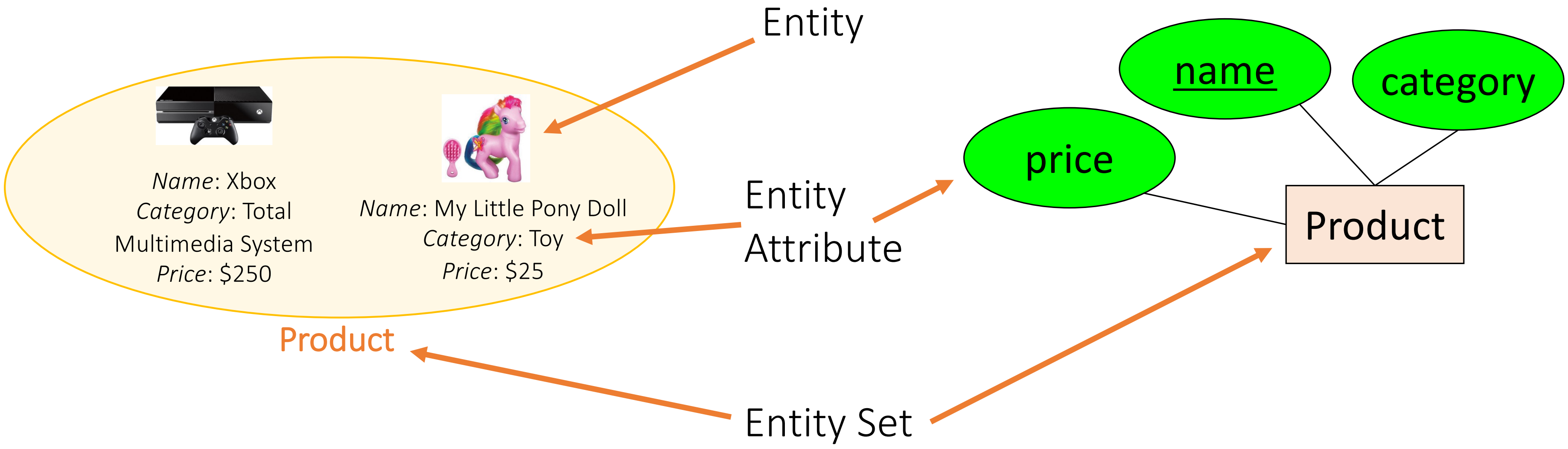


Shapes **are** important.
Colors **are not**.

Entities vs. Entity Sets

Example:

Entities are **not** explicitly represented in E/R diagrams!



Name: Xbox
Category: Total Multimedia System
Price: $250

Name: My Little Pony Doll
Category: Toy
Price: $25

Product

Entity

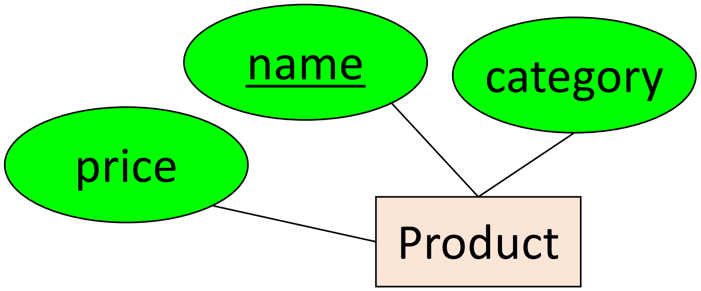
Entity Attribute

Entity Set

Keys

- A *key* is a **minimal** set of attributes that uniquely identifies an entity.

Denote elements of the primary key by underlining.



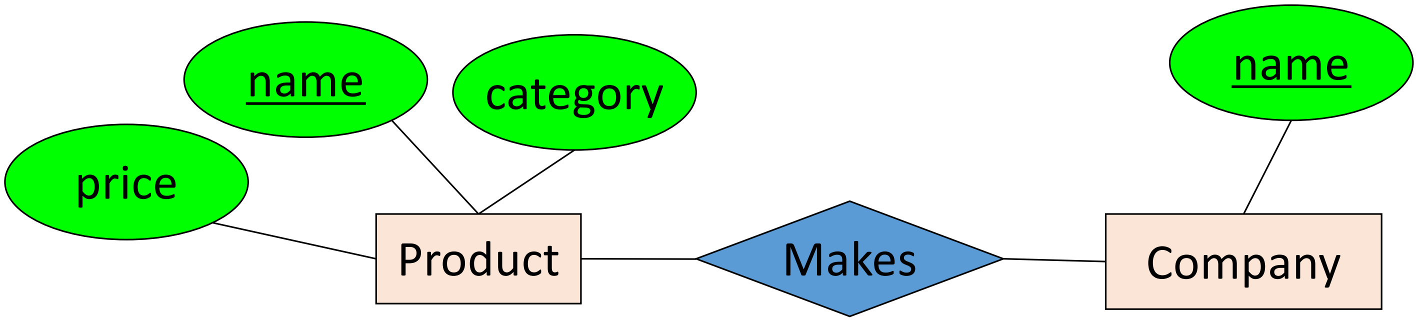
Here, {name, category} is **not** a key (it is not *minimal*).

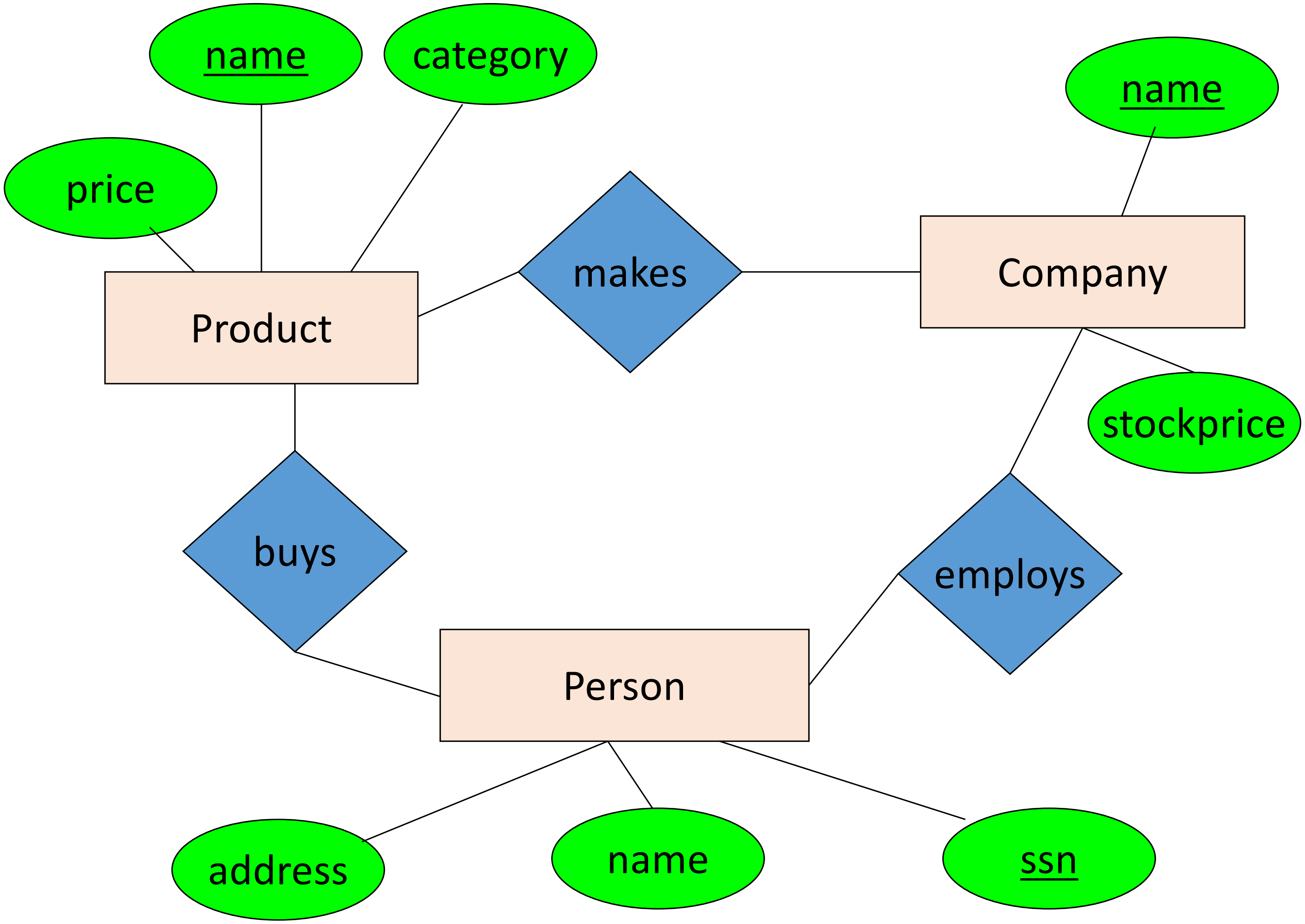
If it were, what would it mean?

The E/R model forces us to designate a single **primary** key, though there may be multiple candidate keys

The R in E/R: **Relationships**

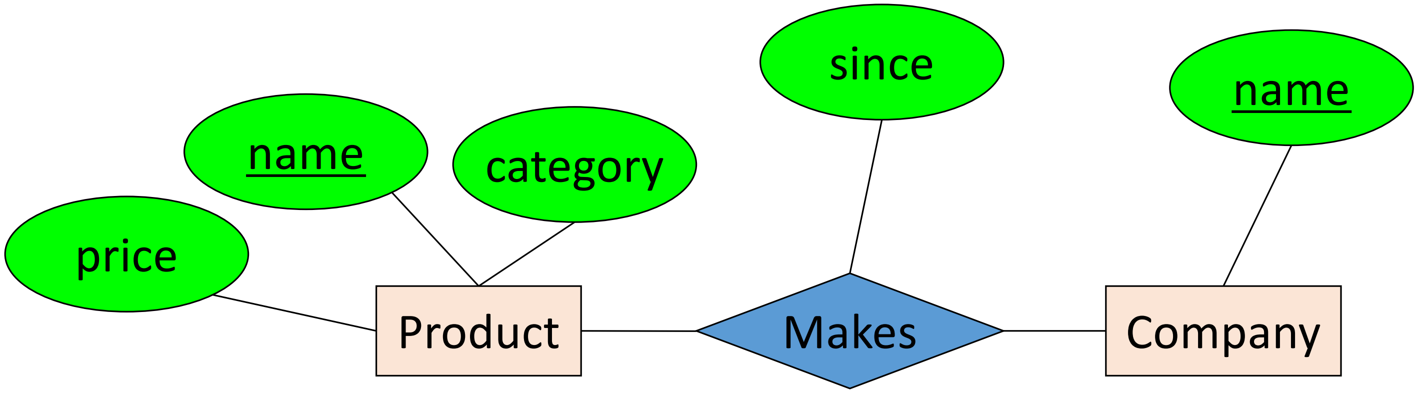
- A **relationship** is between two entities




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

Relationships and Attributes

- Relationships may have attributes as well.



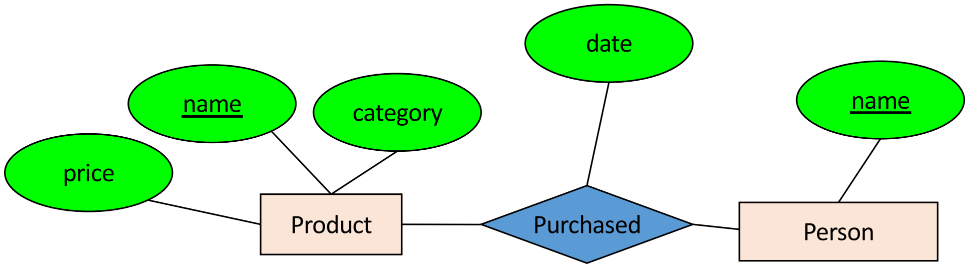
For example: "since" records when company started making a product

Note: *"since"* is implicitly unique per pair here! Why?

Note #2: Why not "how long"?

Decision: Relationship vs. Entity?

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

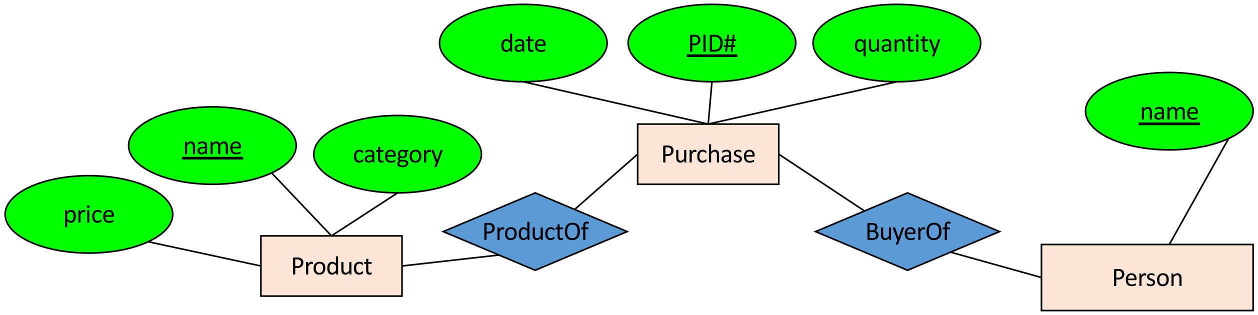


- **A:** A person can only buy a specific product once (on one date)

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!*

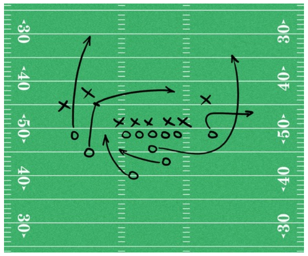
We can always use **a new entity** instead of a relationship. For example, to permit multiple instances of each entity combination!

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).



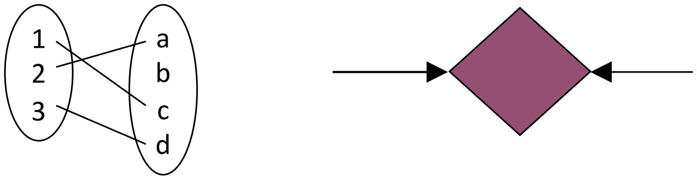
A Game is made up of Plays that result in a yardage gain/loss, and potentially a touchdown

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

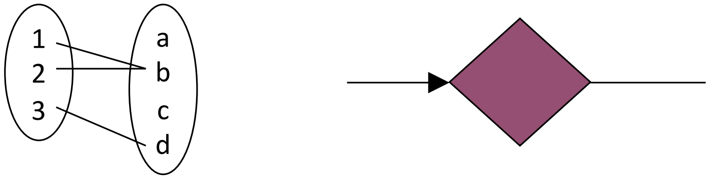
2. E/R Design Considerations

Multiplicity of E/R Relationships

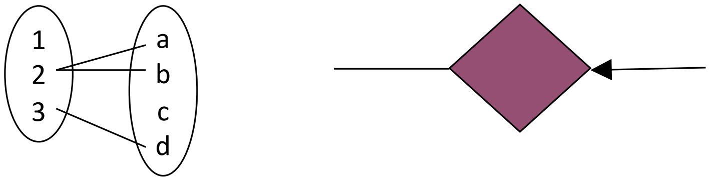
One-to-one:



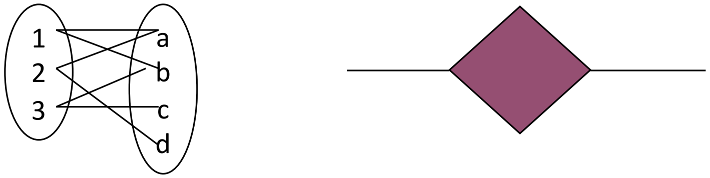
Many-to-one:



One-to-many:



Many-to-many:

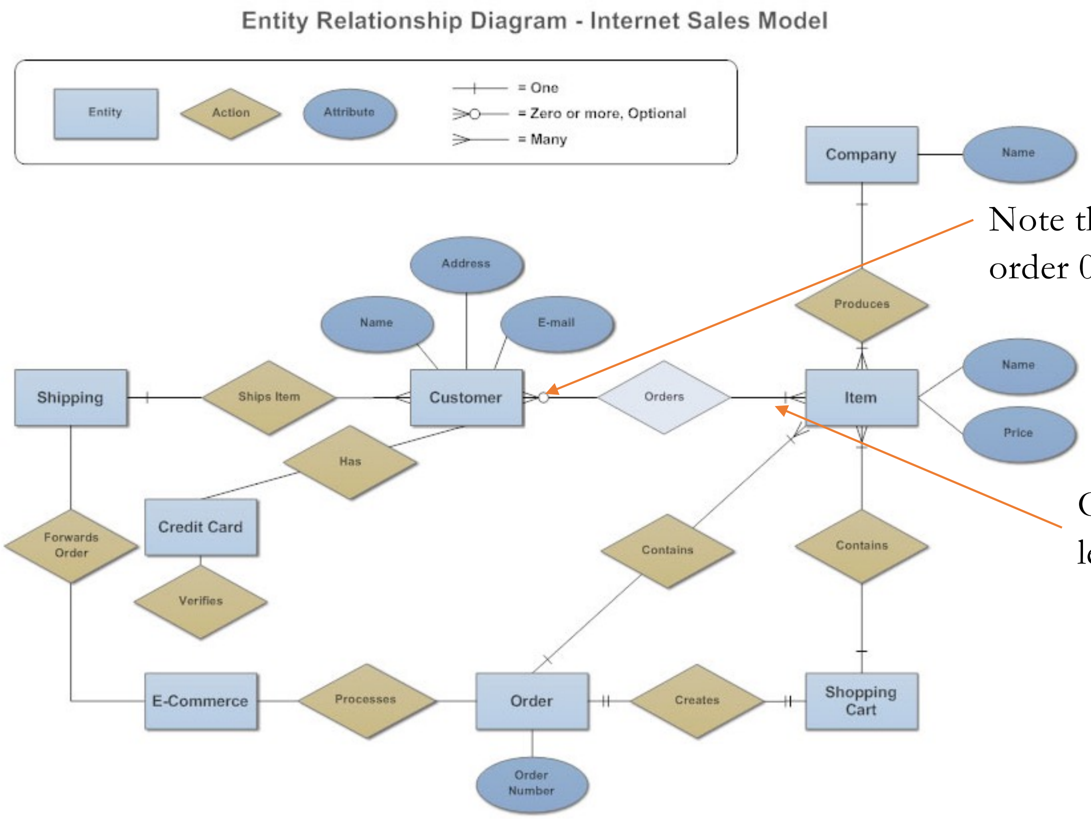


Arrow points to the “one” side

Indicated using arrows

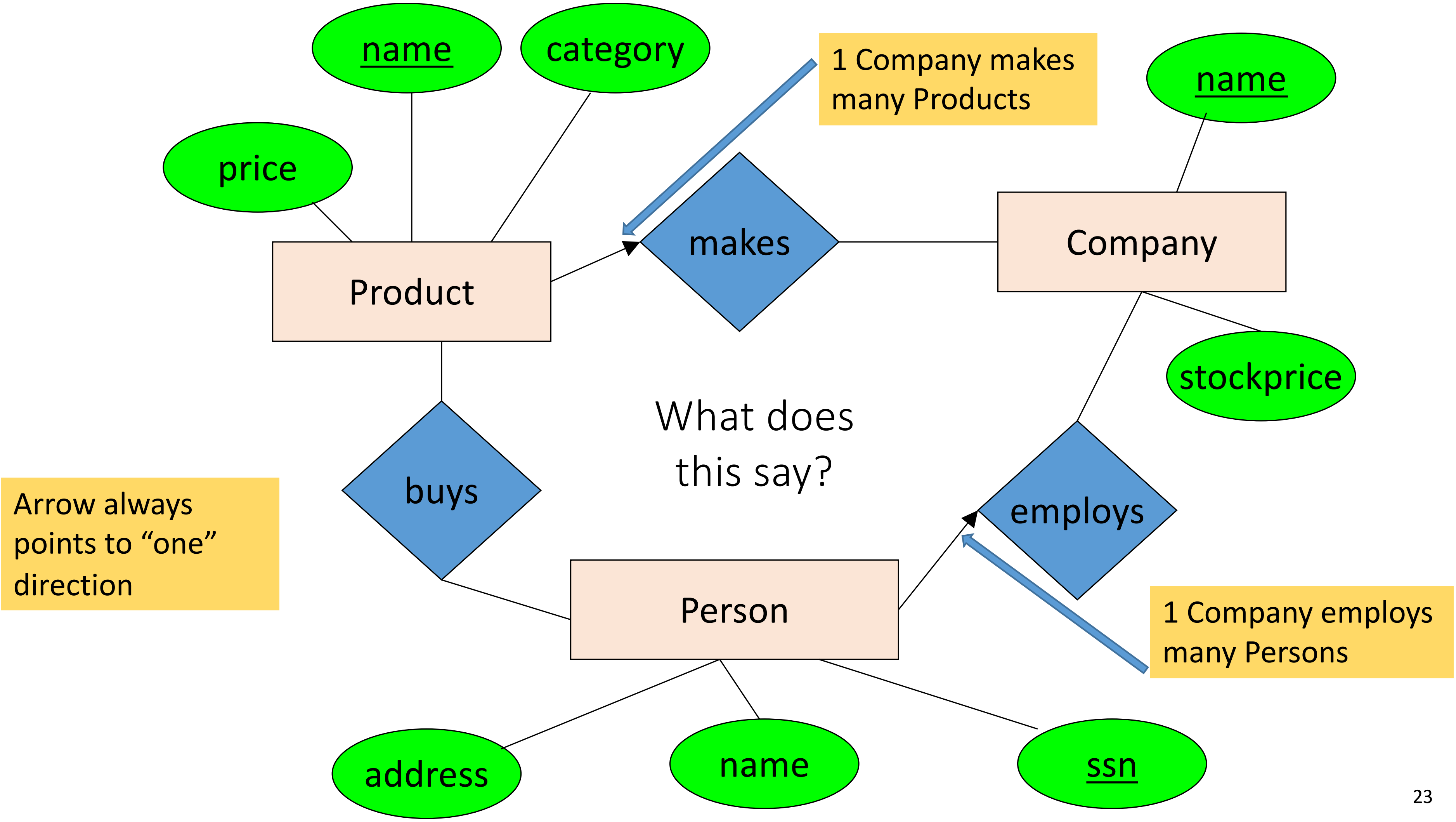
X -> Y means <u>there exists a function mapping from X to Y</u> (*recall the definition of a function)*

Another Way of Drawing ER Diagrams



Note that a customer can order 0 or more items

One item needs to have at least one customer



1 Company makes many Products

What does this say?

Arrow always points to “one” direction

1 Company employs many Persons

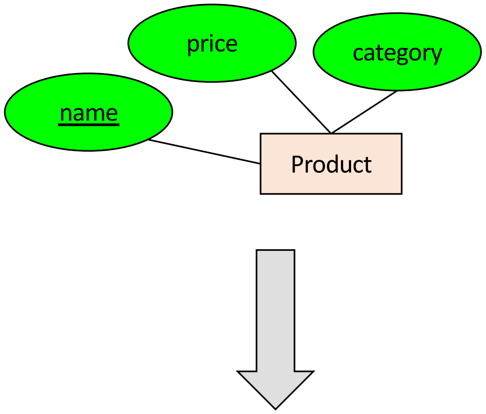
From E/R Diagrams to Relational Schema

- Key concept:

  Both ***Entity sets*** and ***Relationships*** become relations (tables in RDBMS)

From E/R Diagrams to Relational Schema

- An entity set becomes a relation (multiset of tuples / table)
  - Each tuple is one entity
  - Each tuple is composed of the entity's attributes, and has the same primary key

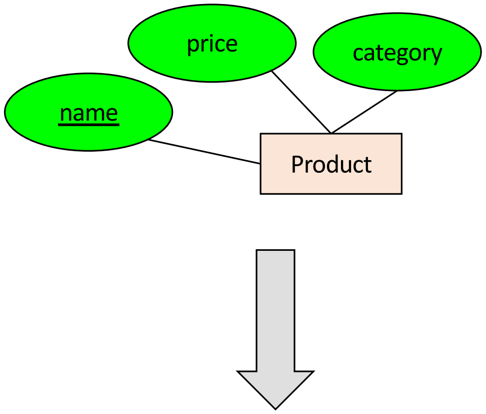


Product

<u>name</u>	price	category
Gizmo1	99.99	Camera
Gizmo2	19.99	Edible

From E/R Diagrams to Relational Schema

```
CREATE TABLE Product(
  name     CHAR(50) PRIMARY KEY,
  price    DOUBLE,
  category VARCHAR(30)
)
```

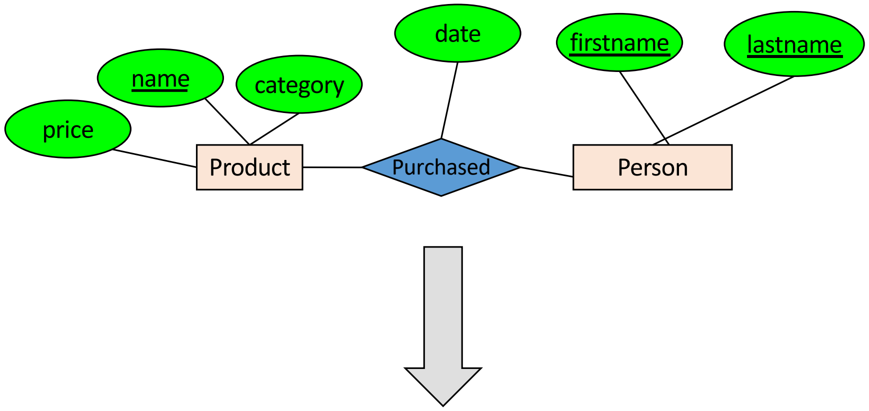


Product

name	price	category
Gizmo1	99.99	Camera
Gizmo2	19.99	Edible

From E/R Diagrams to Relational Schema

- A relation between entity sets $A_1, \ldots, A_N$ *also* becomes a multiset of tuples / a table
  - Each row/tuple is one relation, i.e. one unique combination of entities $(a_1,\ldots,a_N)$
  - Each row/tuple is
    - composed of the **union of the entity sets' keys**
    - has the entities' primary keys as foreign keys
    - has the union of the entity sets' keys as primary key

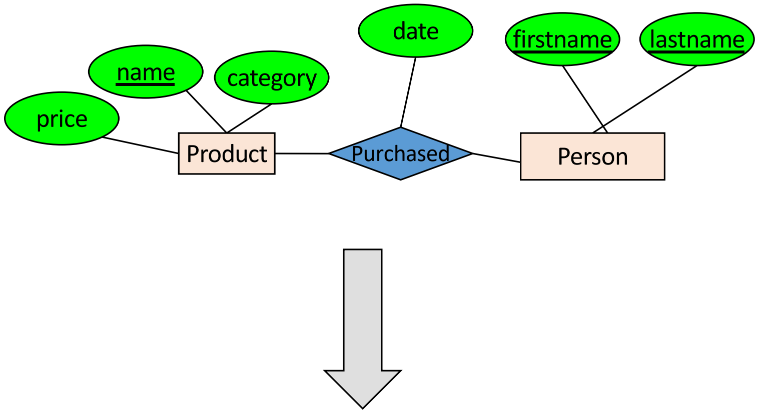


Purchased

name	firstname	lastname	date
Gizmo1	Bob	Joe	01/01/15
Gizmo2	Joe	Bob	01/03/15
Gizmo1	JoeBob	Smith	01/05/15

From E/R Diagrams to Relational Schema

```
CREATE TABLE Purchased(
  name      CHAR(50),
  firstname CHAR(50),
  lastname  CHAR(50),
  date      DATE,
  PRIMARY KEY (name, firstname, lastname),
  FOREIGN KEY (name)
      REFERENCES Product,
  FOREIGN KEY (firstname, lastname)
      REFERENCES Person
)
```



Purchased

name	firstname	lastname	date
Gizmo1	Bob	Joe	01/01/15
Gizmo2	Joe	Bob	01/03/15
Gizmo1	JoeBob	Smith	01/05/15

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, but not overzealous