DB Design:
E/R Model

Today's Lecture

1. How to communicate DBs with
    - x-teams, customers, stakeholders?

2. What are good designs?

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 <u>good</u> designs

- **Several formalisms exist**
  - We discuss few flavors of E/R diagrams

1. E/R Basics: Entities & Relations

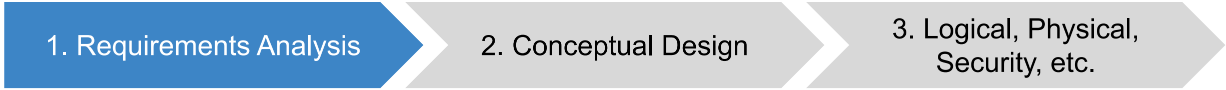
In this section

1. High-level motivation for the E/R model

2. Entities

3. Relations

Database Design Process



1. Requirements Analysis

2. Conceptual Design

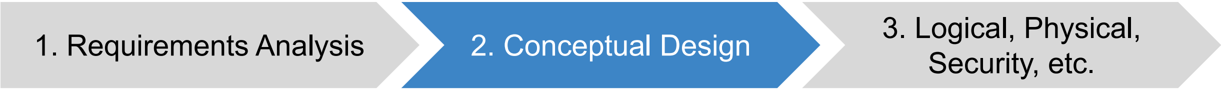
3. Logical, Physical, Security, etc.

Technical and non-technical people are involved

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?

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 reason about it
- But, not so precise that non-technical people can't participate

This is where E/R fits in.

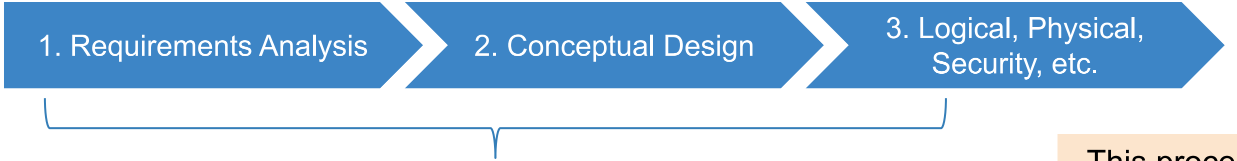
Database Design Process

1. Requirements Analysis
2. Conceptual Design
3. Logical, Physical, Security, etc.

3. More:

- Logical Database Design
- Physical Database Design
- Security Design

Database Design Process

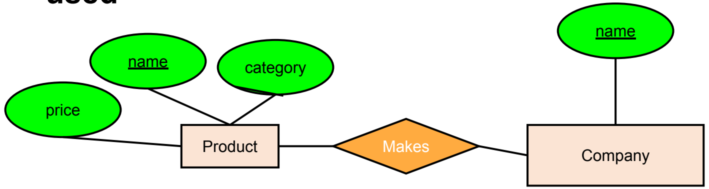


1. Requirements Analysis

2. Conceptual Design

3. Logical, Physical, Security, etc.

E/R Model & Diagrams used



This process is iterated **many** times

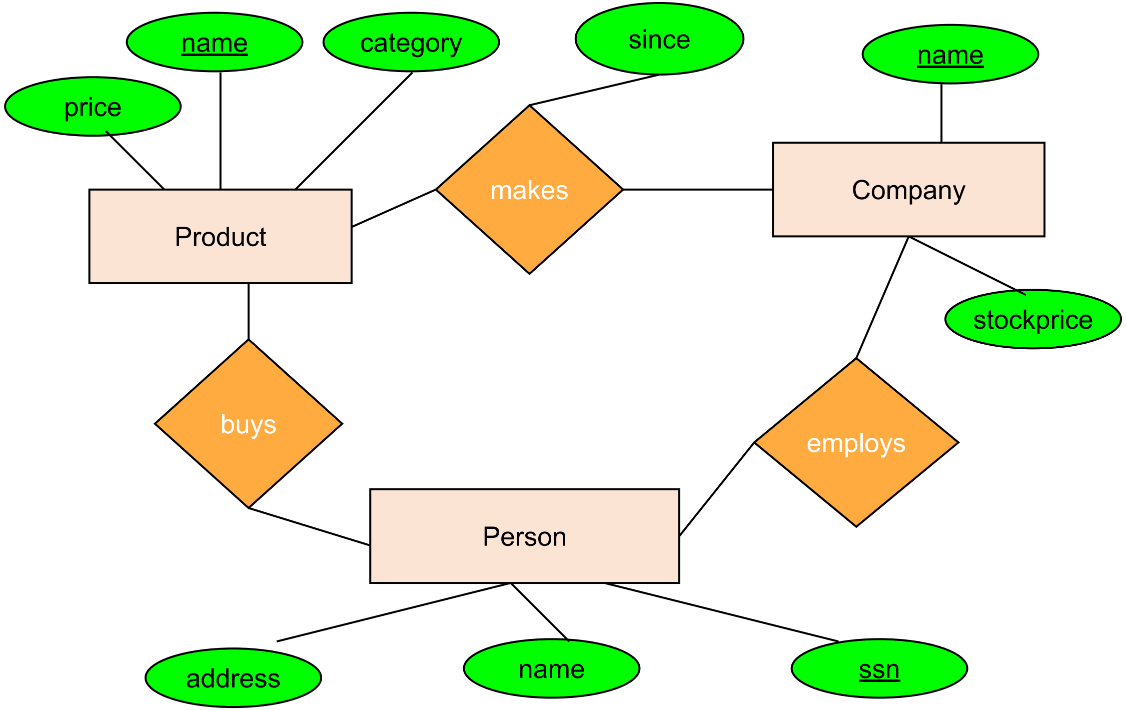
E/R is a *visual syntax* for DB design which is ***precise enough*** for technical points, but ***abstracted enough*** for non-technical people

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

An example E/R diagram

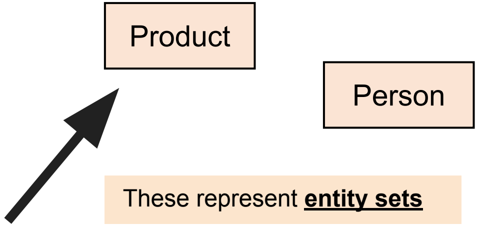


Entities and Entity Sets

- **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*

- In both Chen and Crow's Foot models, entity is represented by rectangle containing entity's name
- Entity name, a noun, is usually written in capital letters



These represent **entity sets**

More on Attributes

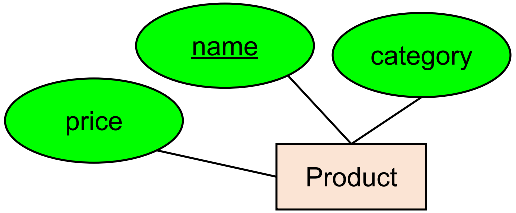
- **Required Attributes: Not Null, Must have a value**
- **Optional Attributes: Can have NULL values**
- **Domain: Attribute set of possible values (Example: Marks range)**
- **Attributes may share domain**

Entities and Entity Sets

- An entity set has **attributes**
  - <u>Represented by ovals attached to an entity set</u>

- In Chen model, attributes are represented by ovals and are connected to entity rectangle with a line
- Each oval contains the name of attribute it represents
- In Crow's Foot model, attributes are written in attribute box below entity rectangle

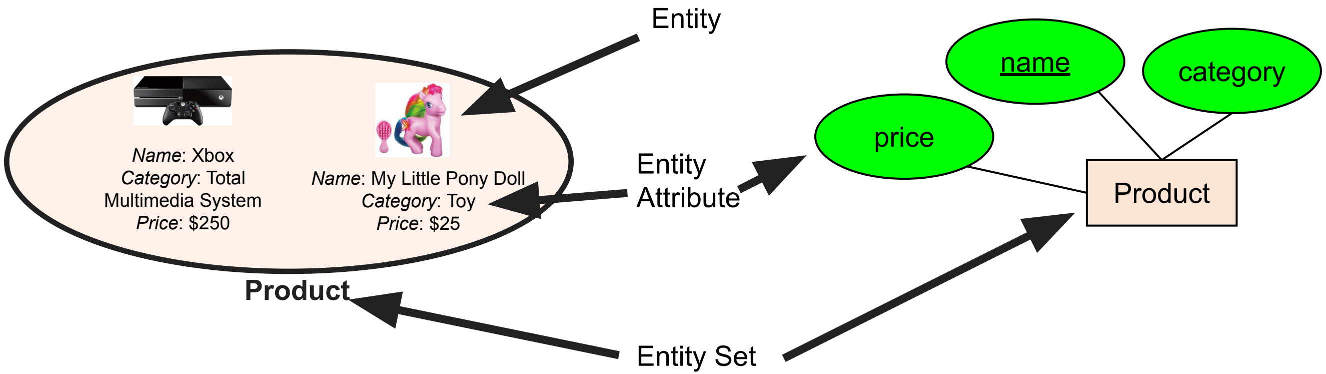
Shapes **<u>are</u>** important. Colors **<u>are not</u>.**



Entities vs. Entity Sets

Example:

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



Entity

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

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

Product

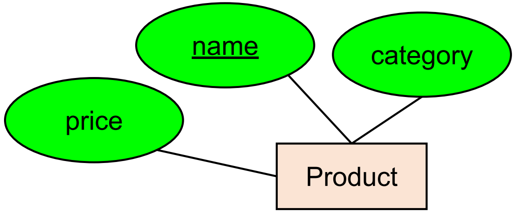
Entity Attribute

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

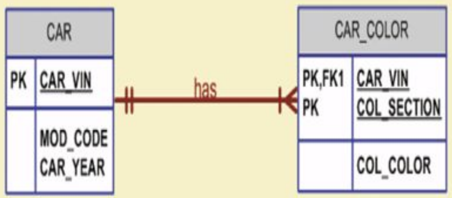
Simple and Composite Attributes

- Simple: (cannot be subdivided)
    - Age and gender etc.
- Composite: (can be divided)
        - Address

Single and Multi-valued Attributes

- Single: (can have single value)
  - CNIC and Passport number etc.
- Multivalued: (can have many values)
    - Color of car and degrees etc.
- Is single = simple?
- Do not implement multivalued attributes in relational databases.
- Can cause major structural problems
- Create new entities for multivalued attribute’s components.

FIGURE 4.5 A new entity set composed of a multivalued attribute's components



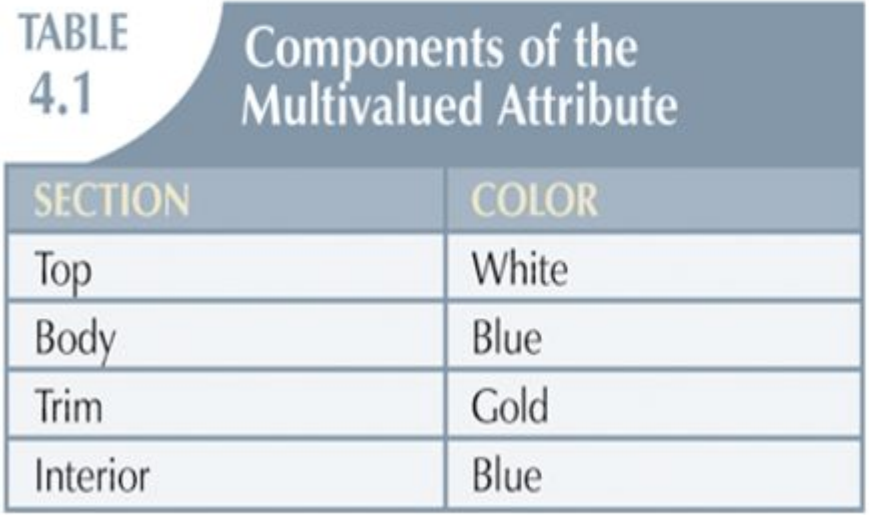


TABLE 4.1 Components of the Multivalued Attribute

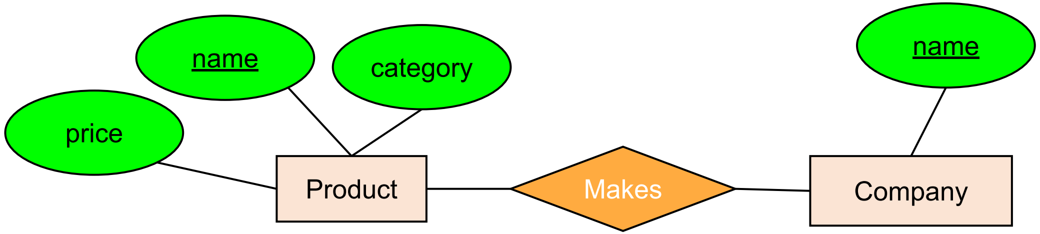
SECTION	COLOR
Top	White
Body	Blue
Trim	Gold
Interior	Blue

Derived Attribute/Computed Attribute

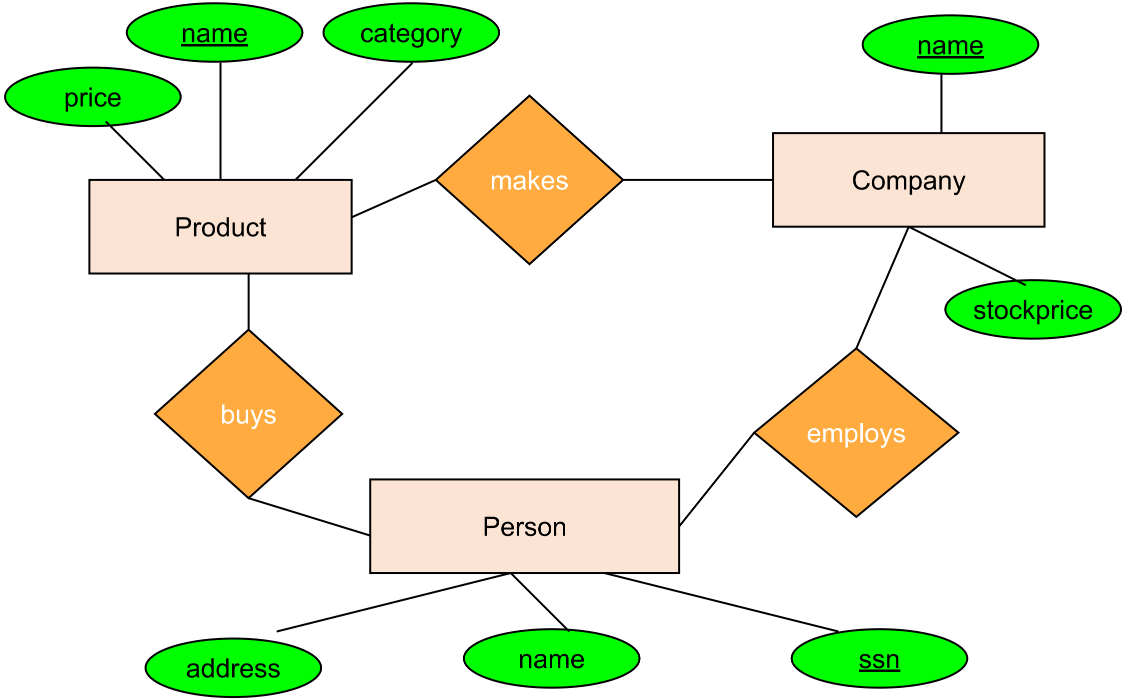
- Derived from other attributes
- Should we save it in database? (Pros/Cons)
- No method in Crow Foot to represent it.
- In Chen it is represented by dotted lines connected to entity set.

The R in E/R: Relationships

- A **relationship** is between two entities



The R in E/R: Relationships



Relationships

- Participants
- Connectivity: Relationship classification (1:1, 1:M and M:N)
- Cardinality: Minimum and Maximum number of entity occurrences associated with one occurrence of related entity

Next Lecture

- Connectivity and Cardinality
- More on relationships and entities