

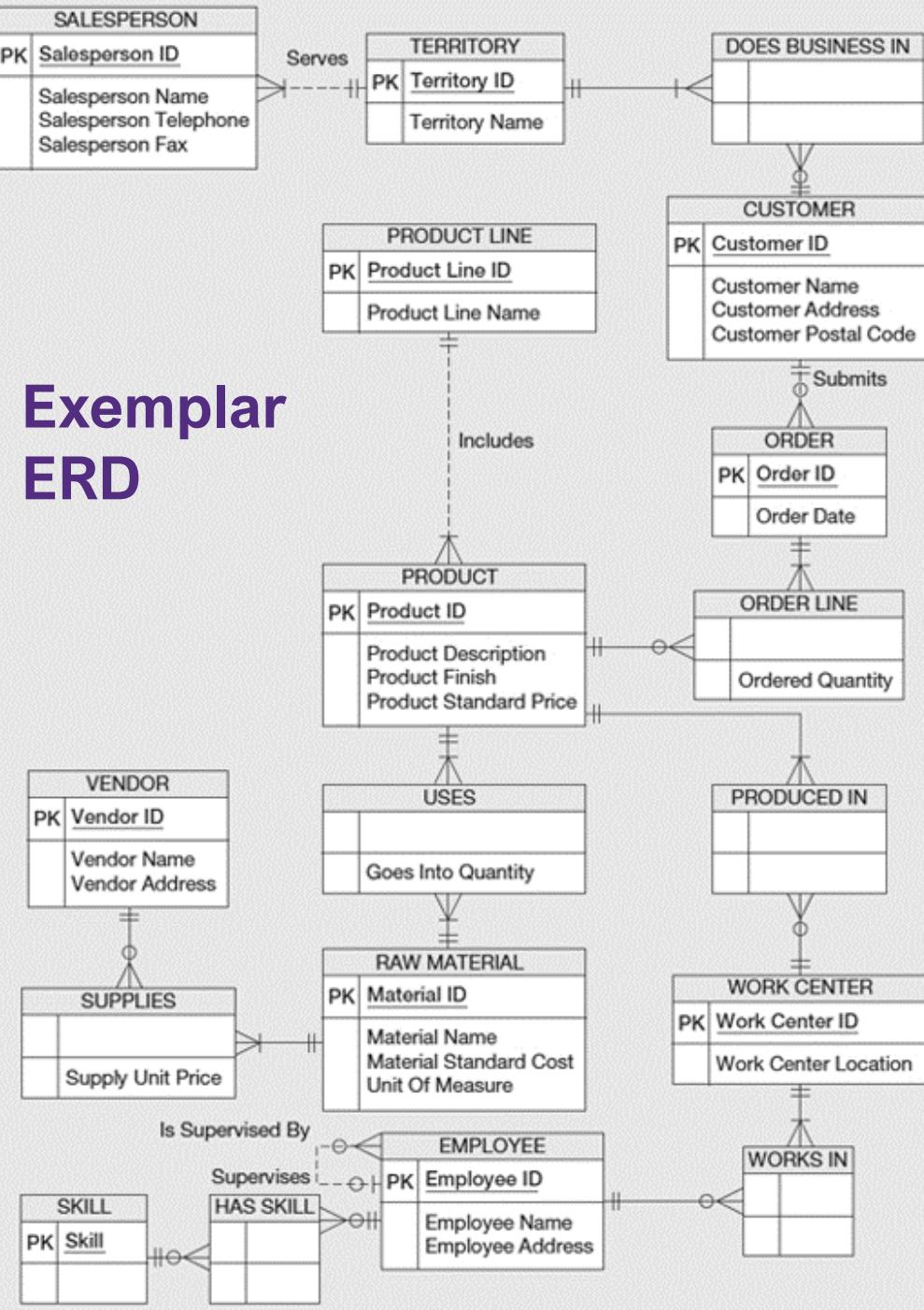
Clemson Means

**BUSINESS**

# Chapter 2

# Conceptual Data Modeling

Instructor: He Li



**Entity-Relationship (E-R) Model:**  
a detailed, logical representation  
of the data for an organization or  
for a business area.

*graphical  
representation*

**Entity-Relationship Diagram  
(ERD)**

# Business Rules

- define or constrain some aspect of the business; derived from policies, procedures, events, functions
- assert business structure; control/influence business behavior
- expressed in terms familiar to end users; automated through DBMS

Characteristic	Explanation
Declarative	A business rule is a statement of policy, not how policy is enforced or conducted; the rule does not describe a process or implementation, but rather describes what a process validates.
Precise	With the related organization, the rule must have only one interpretation among all interested people, and its meaning must be clear.
Atomic	A business rule marks one statement, not several; no part of the rule can stand on its own as a rule (that is, the rule is indivisible, yet sufficient).
Consistent	A business rule must be internally consistent (that is, not contain conflicting statements) and must be consistent with (and not contradict) other rules.
Expressible	A business rule must be able to be stated in natural language, but it will be stated in a structured natural language so that there is no misinterpretation.
Distinct	Business rules are not redundant, but a business rule may refer to other rules (especially to definitions).
Business-oriented	A business rule is stated in terms businesspeople can understand, and because it is a statement of business policy, only businesspeople can modify or invalidate a rule; thus, a business rule is owned by the business.

Source: Based on Gottesdiener (1999) and Plotkin (1999).

# A Good Data Name is:

- Related to business, not technical, characteristics
- Meaningful and self-documenting
- Unique
- Readable
- Composed of words from an approved list
- Repeatable
- Written in standard syntax

# Data Definitions

- Explanation of a term or fact
  - Term – word or phrase with specific meaning
  - Fact – association between two or more terms
- Guidelines for good data definition
  - A concise description of essential data meaning
  - Gathered in conjunction with systems requirements
  - Accompanied by diagrams
  - Achieved by consensus and iteratively refined

**ENTITY:** a person, a place, an object, an event, or a concept in the user environment about which the organization wishes to maintain data

**Entity Type:** a collection of entities that share common properties or characteristics

Entity type: EMPLOYEE			
Attributes	Attribute Data Type	Example Instance	Example Instance
Employee Number	CHAR (10)	64217836	53410197
Name	CHAR (25)	Michelle Brady	David Johnson
Address	CHAR (30)	100 Pacific Avenue	450 Redwood Drive
City	CHAR (20)	San Francisco	Redwood City
State	CHAR (2)	CA	CA
Zip Code	CHAR (9)	98173	97142
Date Hired	DATE	03-21-1992	08-16-1994
Birth Date	DATE	06-19-1968	09-04-1975

**Entity Instance:**  
a single occurrence of an entity type

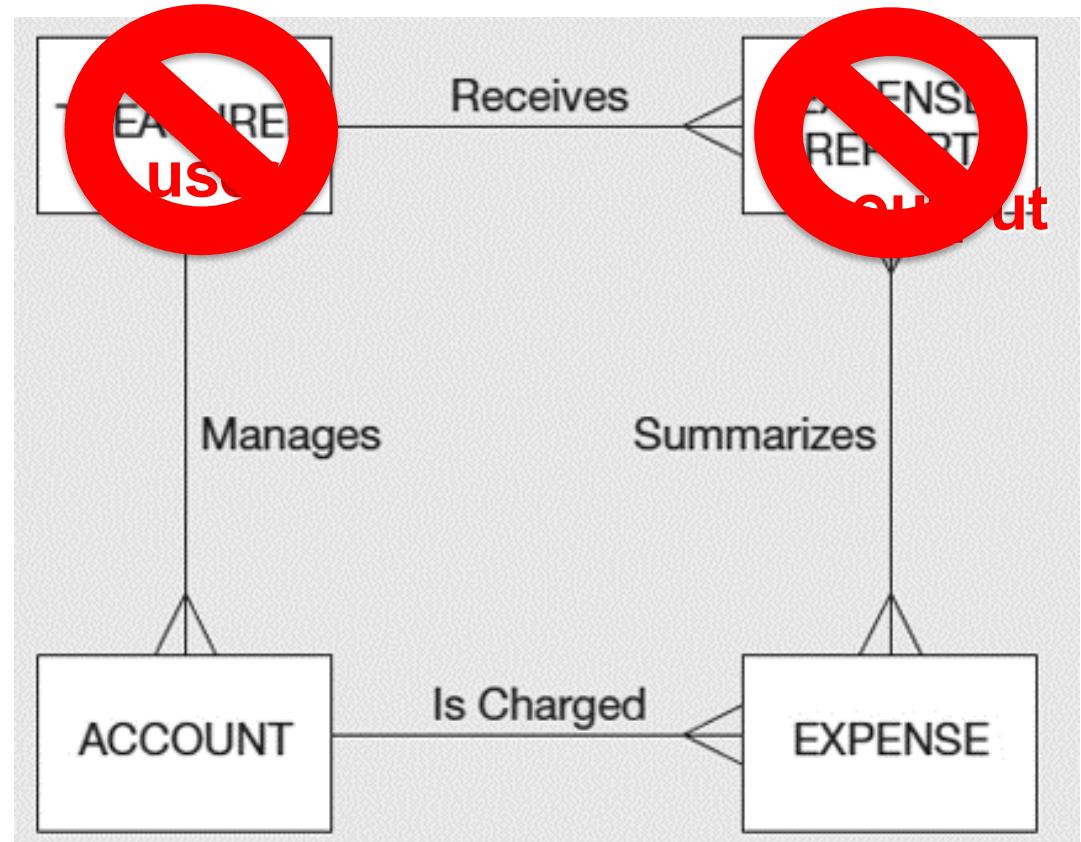
# An Entity...

## Should Be an object that ...

- will have many instances in the database
- will be composed of multiple attributes
- we are trying to model

## Should Not Be:

- a **user** of the database system
- an **output** of the database system (e.g., a report)



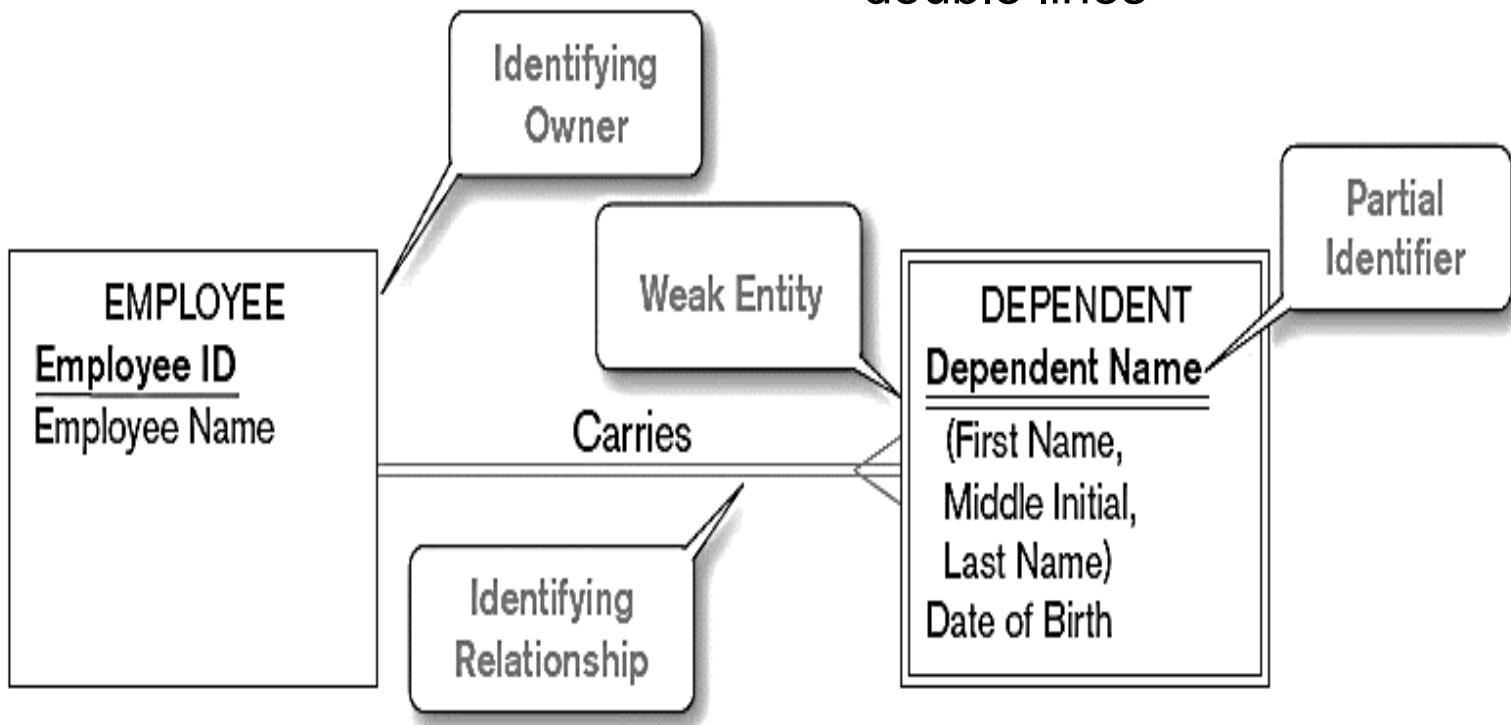
# Guidelines for Naming Entities

- **Guidelines for Naming Entities:**
  - Singular noun
  - Specific to organization
  - Concise, or abbreviation
  - For event entities, the result not the process
  - Name consistent for all diagrams
- **Guidelines for Defining Entity Types:**
  - “An X is...”
  - Describe unique characteristics of each instance
  - Explicit about what is and is not the entity
  - When an instance is created or destroyed
  - Changes to other entity types
  - History that should be kept

# Strong v.s. Weak Entities

## Strong Entity

- exists independently of other types of entities
- has its own unique identifier (*underlined with single line*)



## Weak Entity

- dependent on a strong entity (identifying owner)
- does not have a unique identifier (only a partial identifier)
- entity box and partial identifier have double lines

# Attributes

- Attribute – property or characteristic of an entity or relationship type

---

STUDENT      Student ID, Student Name, Home Address, Phone Number, Major

AUTOMOBILE    Vehicle ID, Color, Weight, Horsepower

EMPLOYEE      Employee ID, Employee Name, Payroll Address, Skill

---

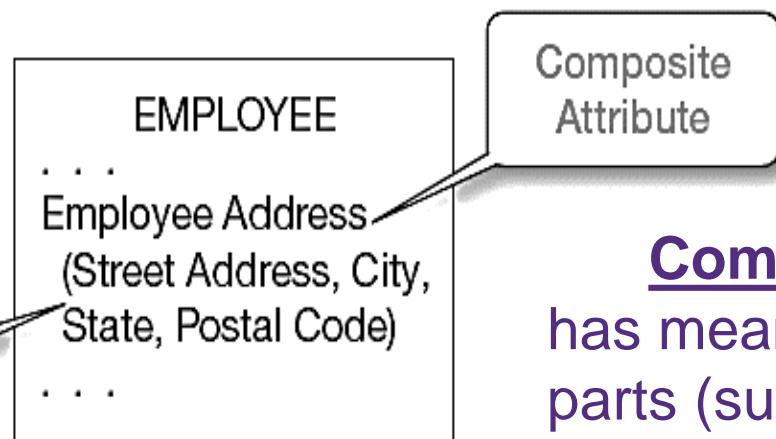
- Classifications of attributes:
  - Required versus Optional
  - Simple versus Composite
  - Single-Valued versus Multivalued
  - Stored versus Derived
  - Identifier

# Required versus. Optional Attributes

**Required:** must have a value for every entity (or relationship) instance with which it is associated

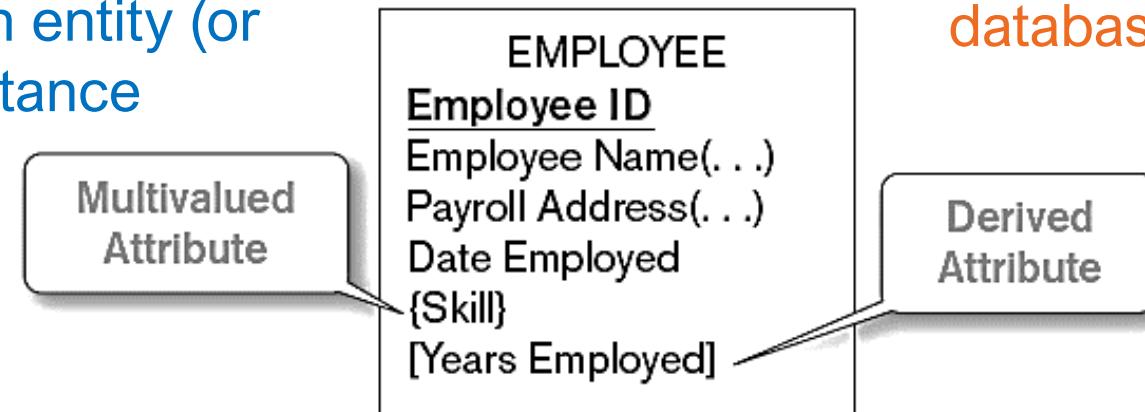
Entity type: STUDENT				
Attributes	Attribute Data Type	Required or Optional	Example Instance	Example Instance
Student ID	CHAR (10)	Required	28-618411	26-844576
Student Name	CHAR (40)	Required	Michael Grant	Melissa Kraft
Home Address	CHAR (30)	Required	314 Baker St.	1422 Heft Ave
Home City	CHAR (20)	Required	Centerville	Miami
Home State	CHAR (2)	Required	OH	FL
Home Zip Code	CHAR (9)	Required	45459	33321
Major	CHAR (3)	Optional	MIS	

**Optional:** may not have a value for every entity (or relationship) instance with which it is associated



**Composite Attribute:**  
has meaningful component parts (sub-attributes)

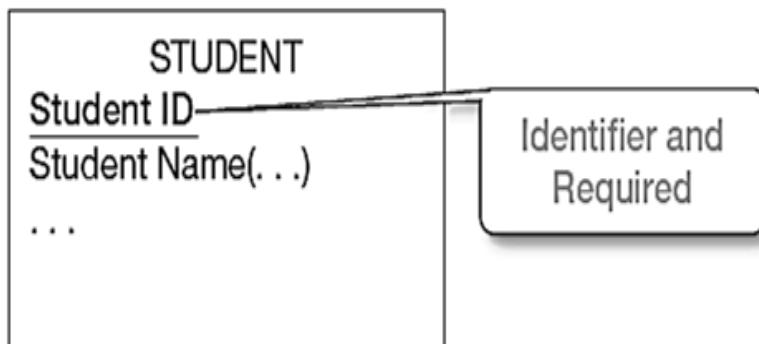
**Multivalued Attribute:**  
May take on more than one value for a given entity (or relationship) instance



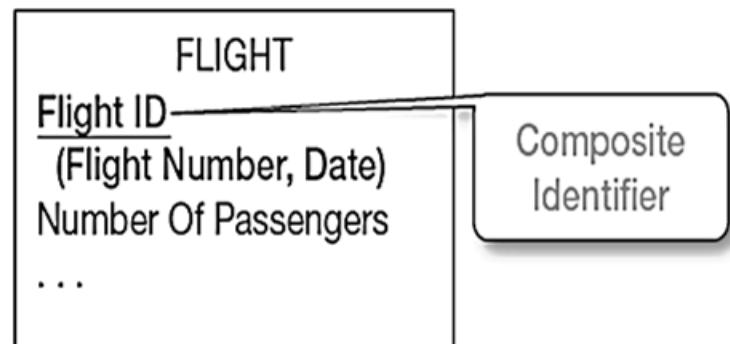
**Derived Attribute:**  
Values can be calculated from related attribute values (not physically stored in the database)

# Identifiers (Keys)

- **Identifier (Key)** – attribute(s) that uniquely identifies individual instances
- **Candidate Identifier** – an attribute that satisfies the requirements for being an identifier
- **Criteria for Identifier**
  - Choose Identifiers that: (1) will not change in value; (2) will not be null
  - Avoid intelligent identifiers (e.g., containing locations or people that might change)
  - Substitute new, simple keys for long, composite keys

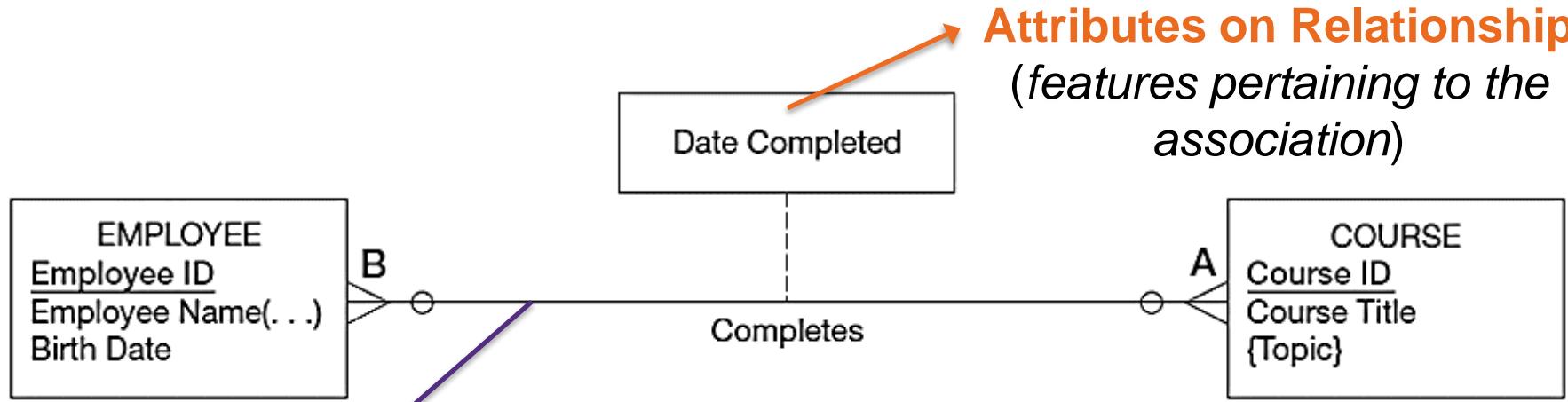


simple identifier attribute



composite identifier attribute

# Modeling Relationships



**Relationship Types**  
(lines between entity types)

**Relationship Instances**  
(between specific entity instances)

**Attributes on Relationship**  
(features pertaining to the association)

Employee Name	Course Title	Date Completed
Chen	C++	06/2009
Chen	Java	09/2009
Melton	C++	06/2009
Melton	COBOL	02/2010
Melton	SQL	03/2009
Ritchie	Perl	11/2009

# Associative Entity – *combination of relationship and entity*



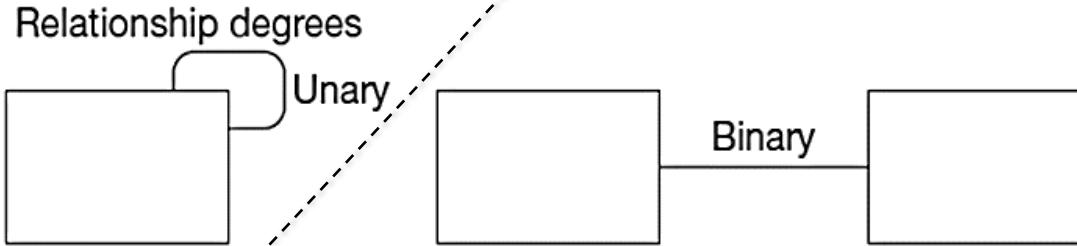
## When Do We Use Associative Entity?

- All the relationships for the participating entity types are “many” relationships
- Associative entity has independent meaning & can be identified with a single-attribute identifier
- Associative entity has attributes in addition to identifier
- Associative entity participates in other relationships beyond the associated relationship

# Degree of Relationships – *the number of participated entity types*

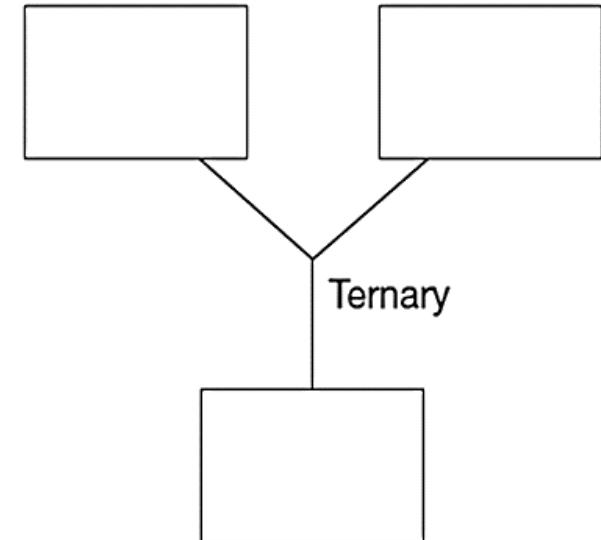
## Unary Relationship:

entities of the same entity type related to each other



## Binary Relationship:

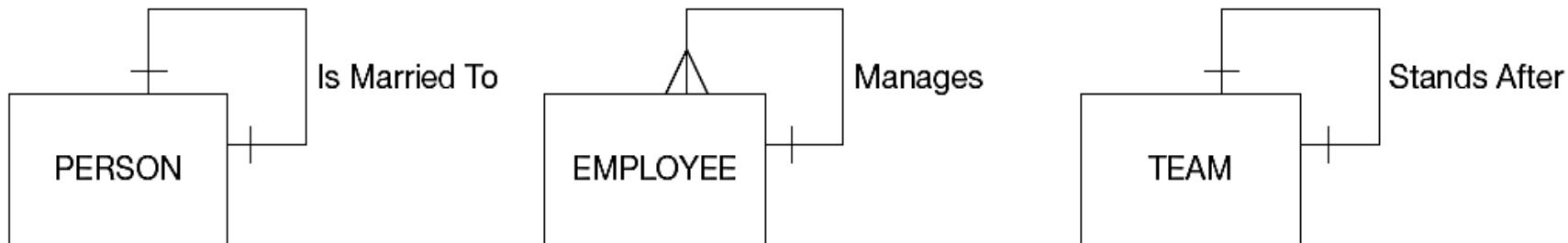
entities of one type related to entities of another



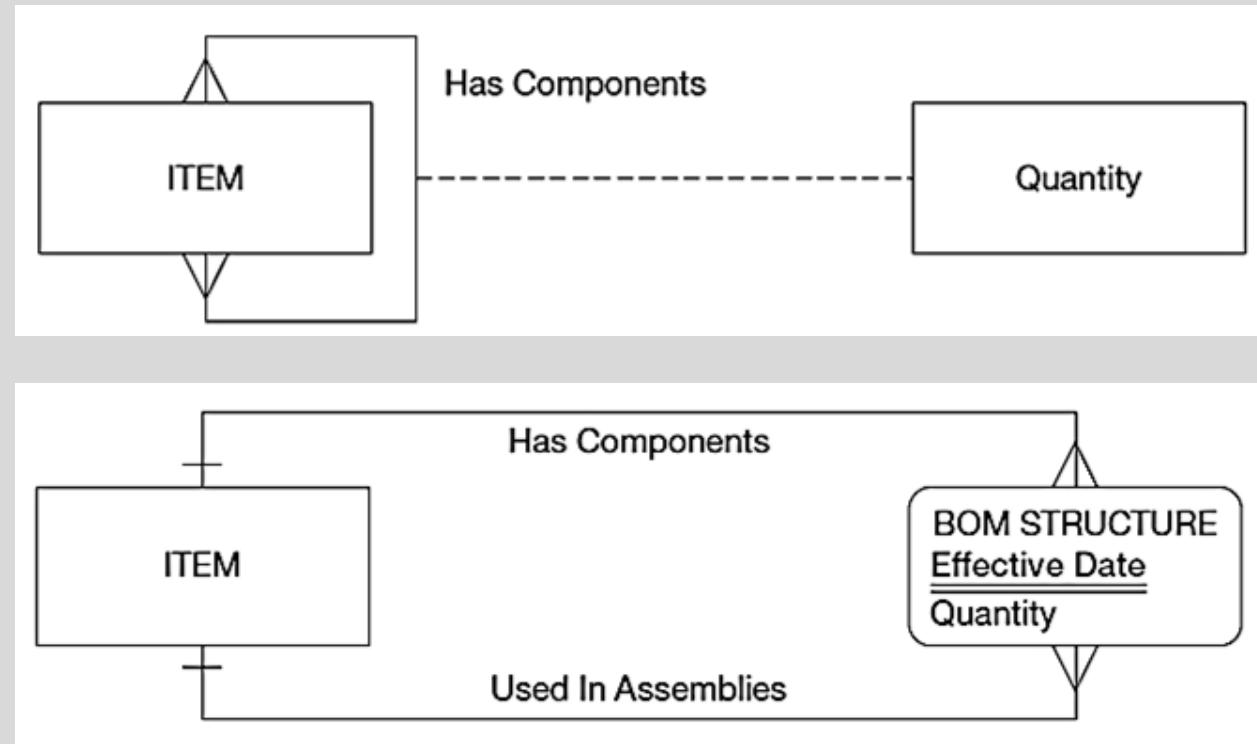
## Ternary Relationship:

entities of three different types involved in the same relationship

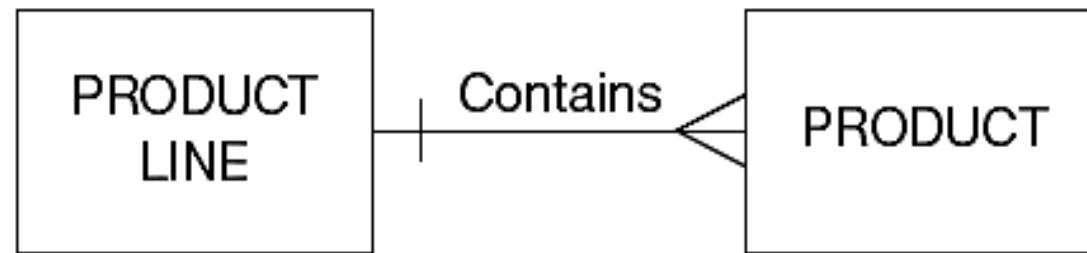
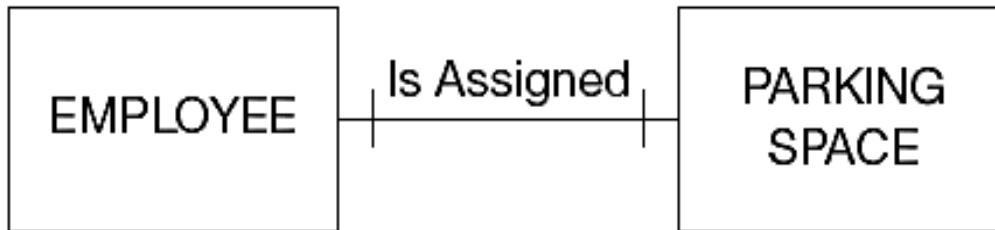
# Examples of Unary Relationships



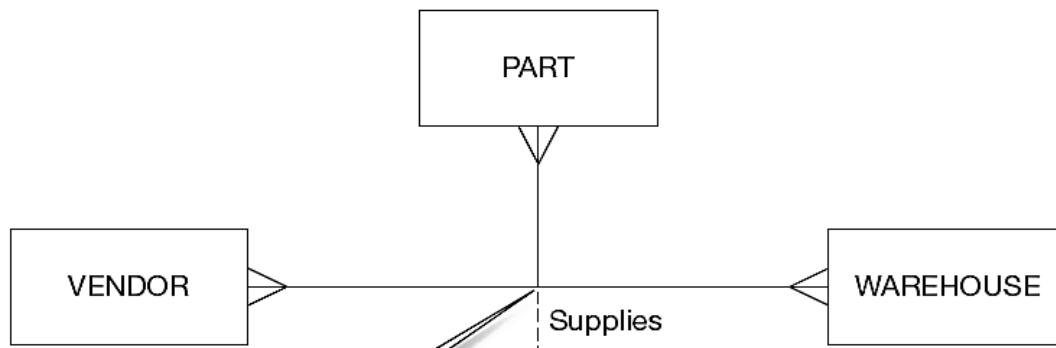
## Bill-of-Materials Structure



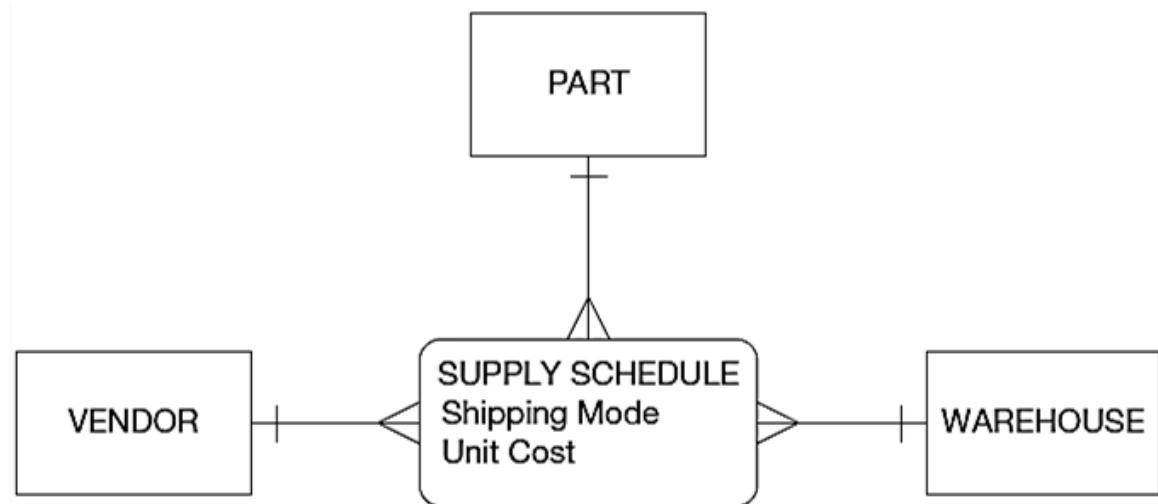
# Examples of Binary Relationships



# Examples of Ternary Relationships



For example, an instance is Vendor X Supplies Part C to Warehouse Y with a Shipping Mode of "next-day air" and a Unit Cost of \$5

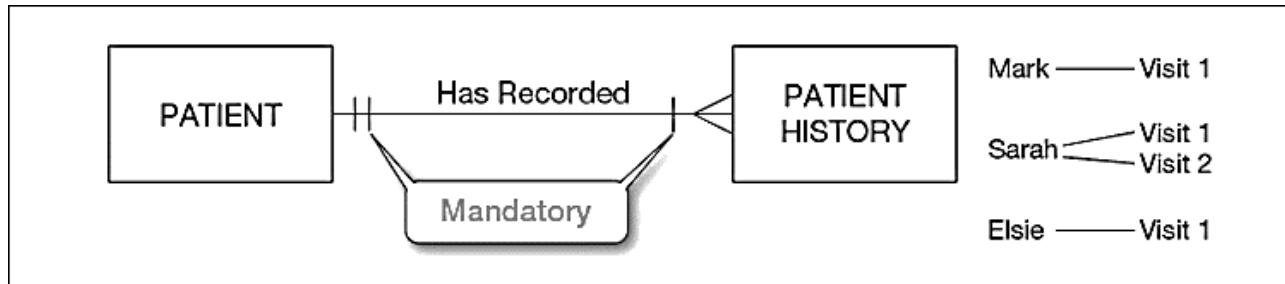


# Cardinality Constraints – *the number of instances of the relationship*

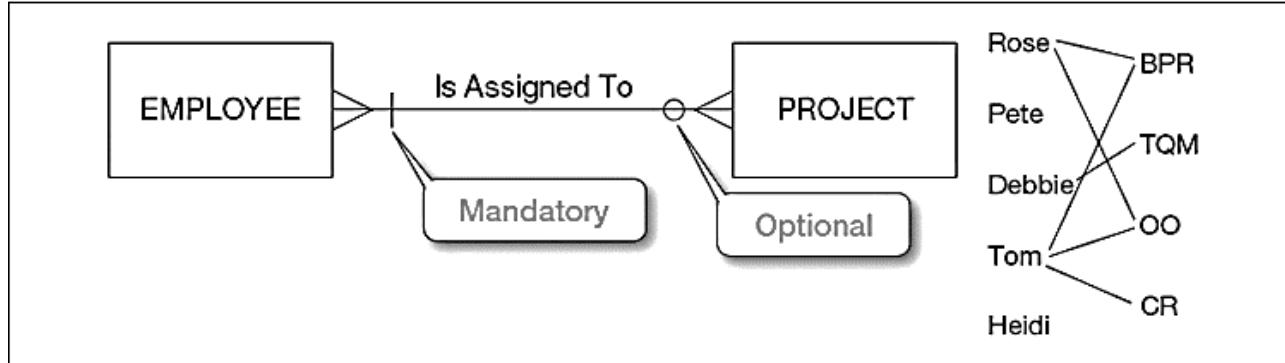
- Minimum Cardinality
  - If zero, then optional
  - If one or more, then mandatory
- Maximum Cardinality
  - The maximum number



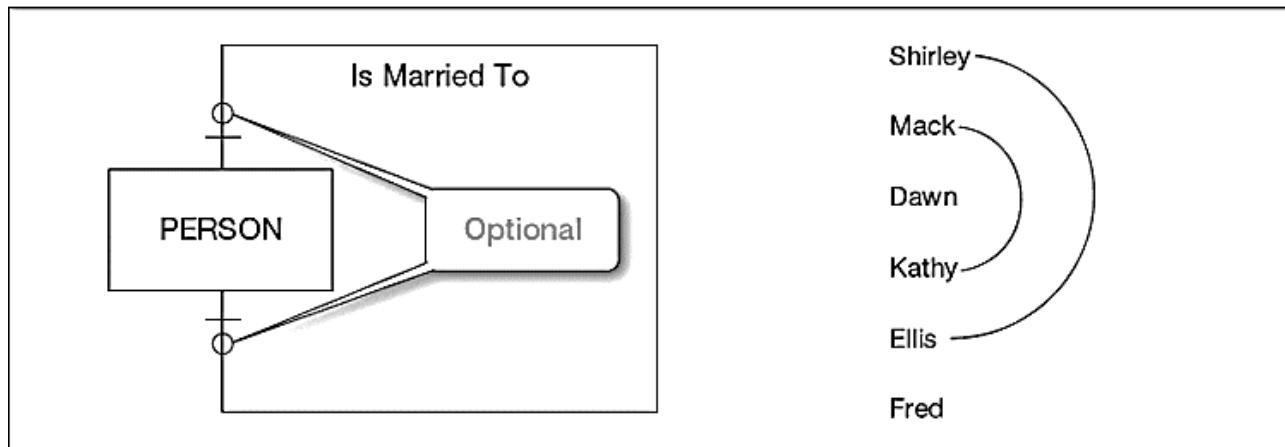
# Examples of Cardinalities



(a) Mandatory cardinalities

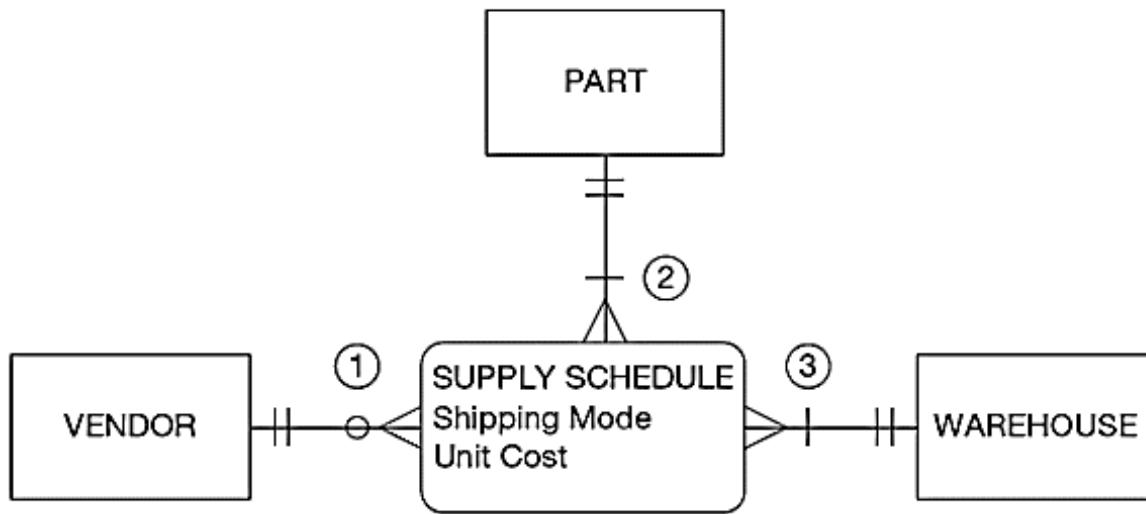


(b) One optional, one mandatory cardinality



(c) Optional cardinalities

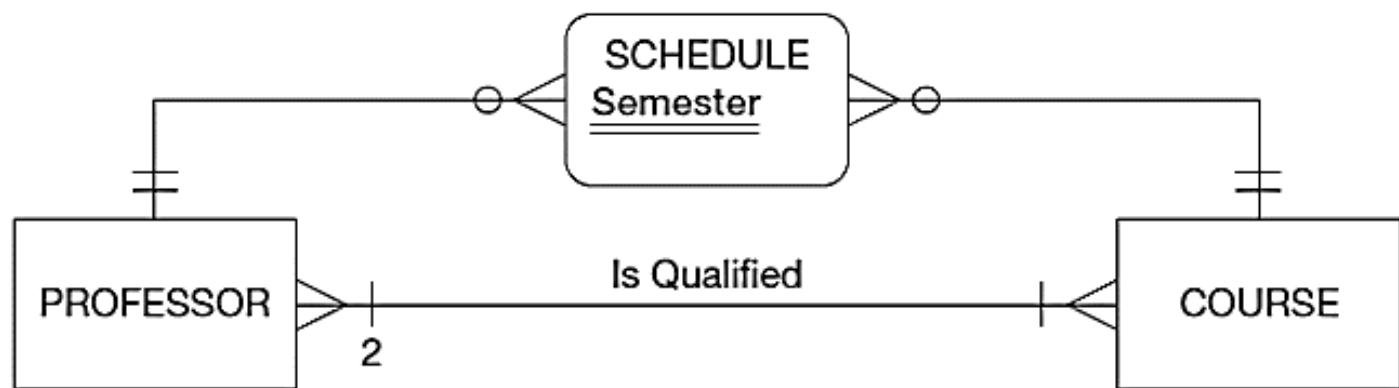
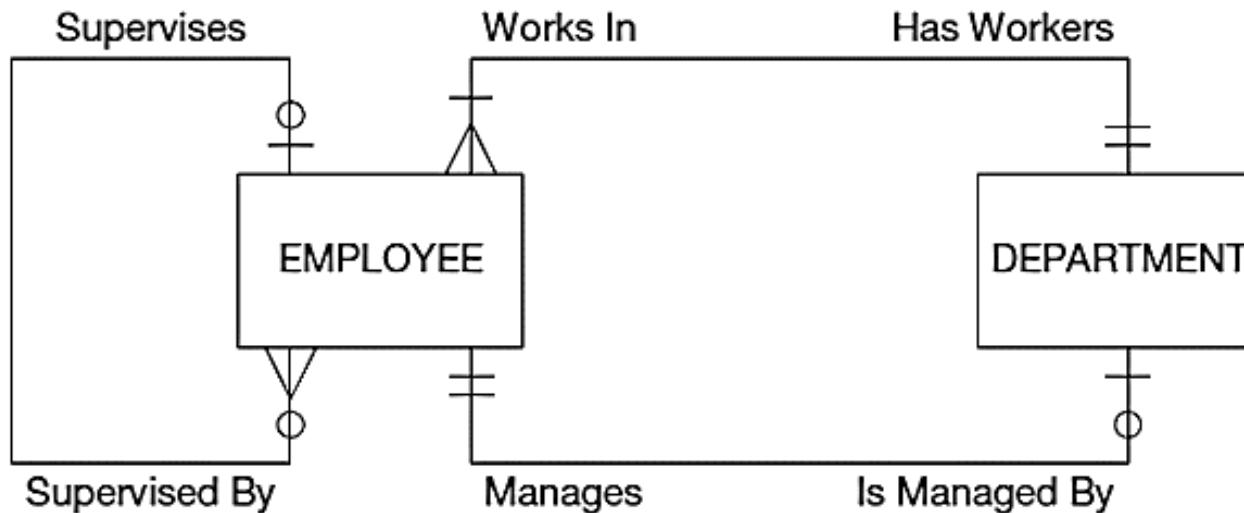
# Cardinality Constraints in a Ternary Relationship



## Business Rules

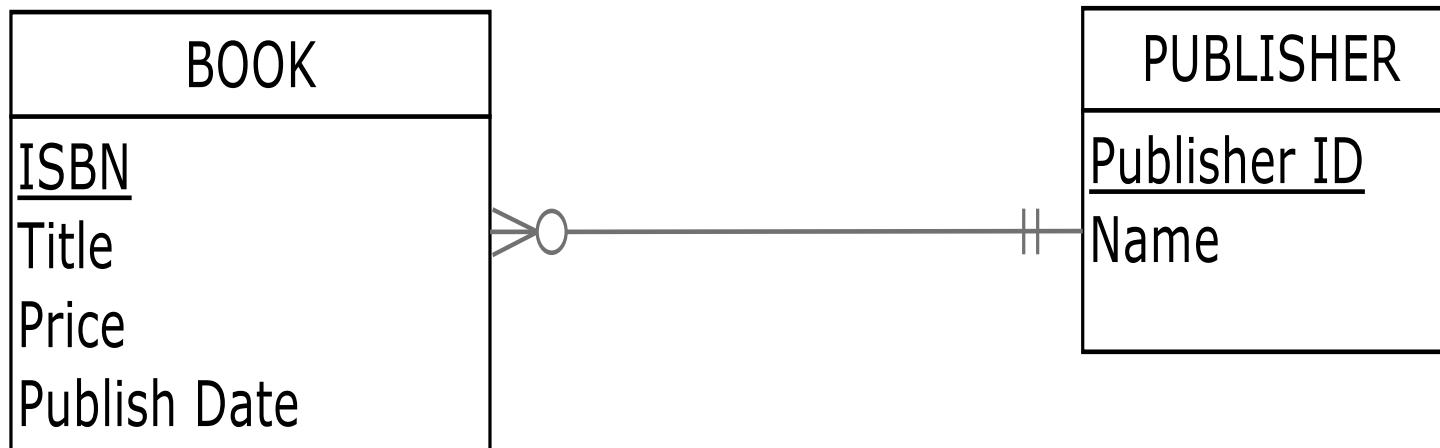
- ① Each vendor can supply many parts to any number of warehouses but need not supply any parts.
- ② Each part can be supplied by any number of vendors to more than one warehouse, but each part must be supplied by at least one vendor to a warehouse.
- ③ Each warehouse can be supplied with any number of parts from more than one vendor, but each warehouse must be supplied with at least one part.

# Modeling Multiple Relationships



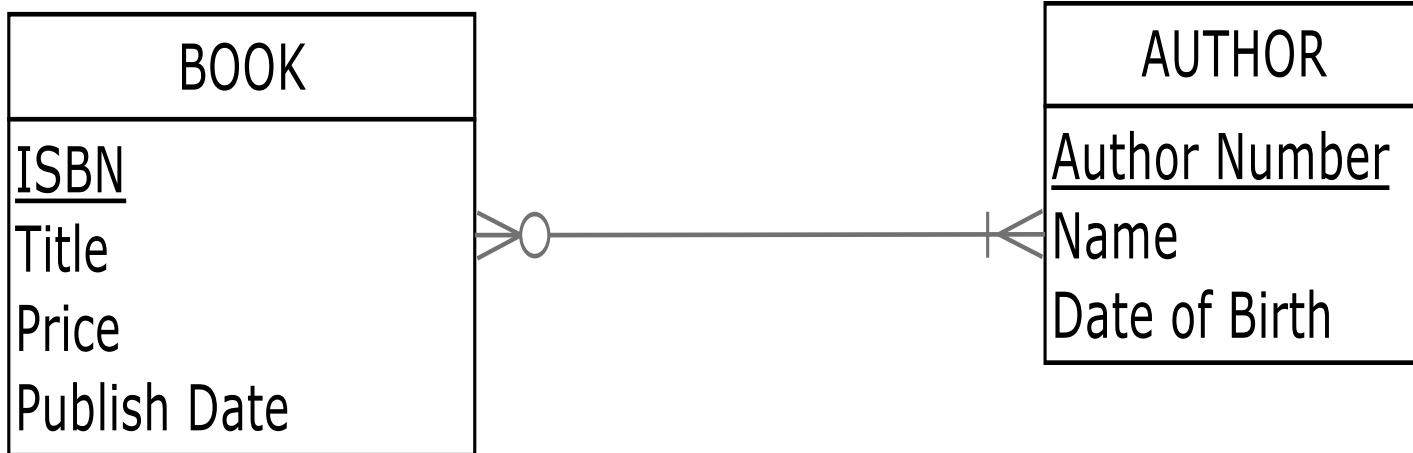
# Draw an ERD ....

- a. A book is identified by its ISBN, and it has a title, a price, and a date of publication. It is published by a publisher, each of which has its own ID number and a name. Each book has exactly one publisher, but one publisher typically publishes multiple books over time.



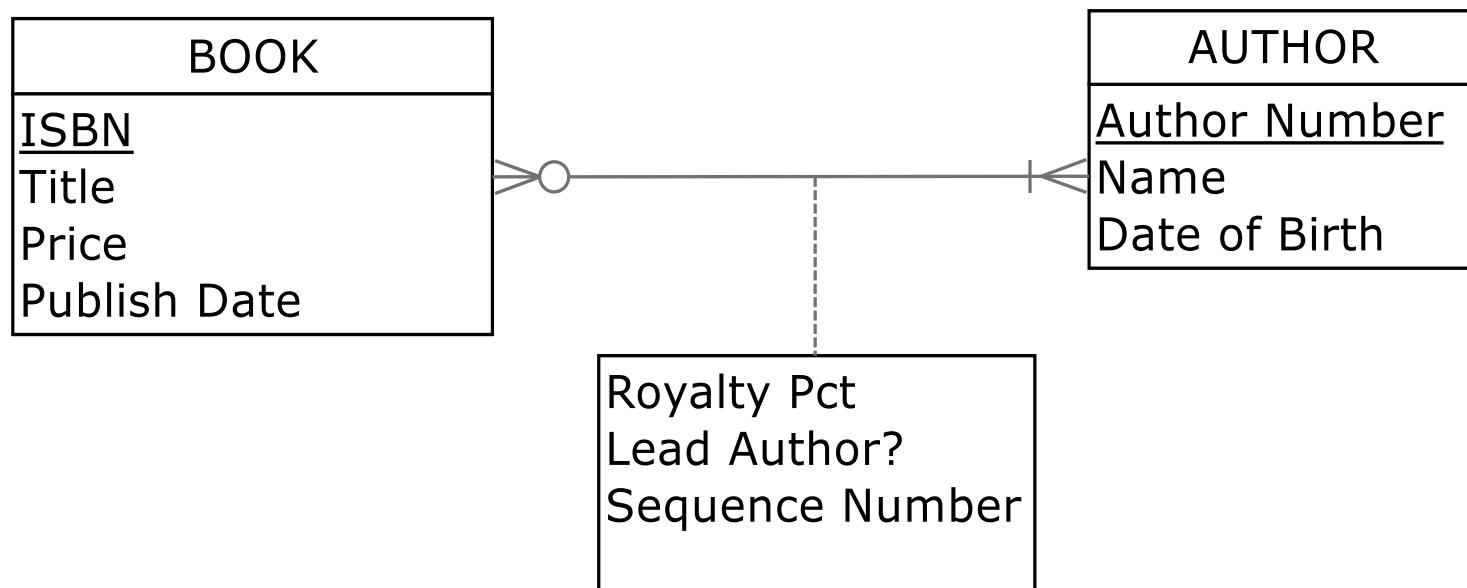
# Draw an ERD ....

b. A book (see above in (a)) is written by one or multiple authors. Each author is identified by an author number and has a name and date of birth. Each author has either one or multiple books; in addition, occasionally data are needed also regarding prospective authors who have not yet published any books.



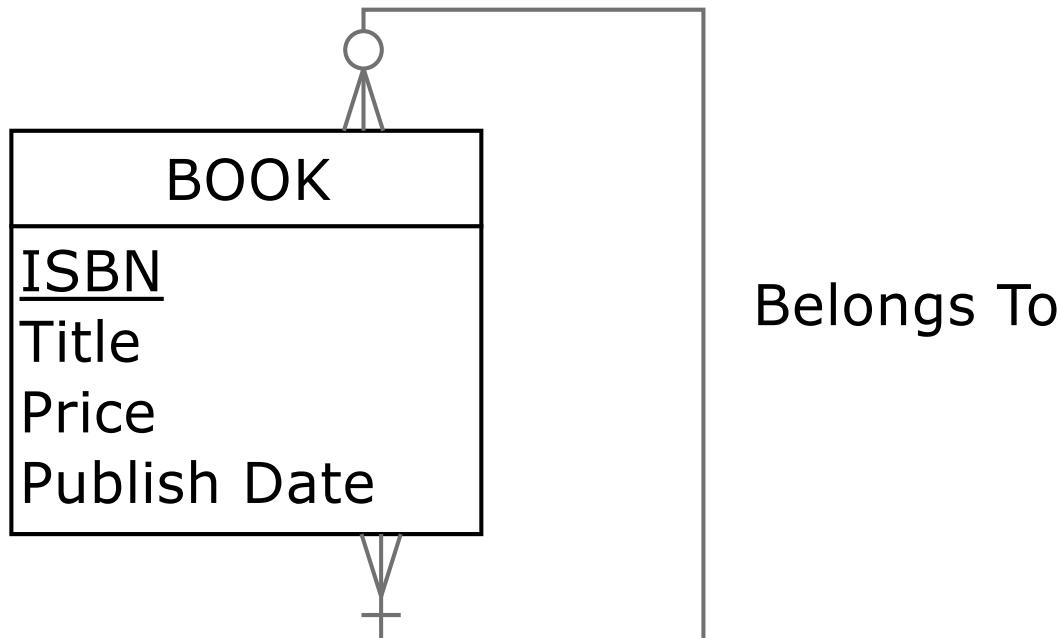
# Draw an ERD ....

c. In the context specified above in (a) and (b), better information is needed regarding the relationship between a book and its authors. Specifically, it is important to record the percentage of the royalties that belong to a specific author, whether or not a specific author is a lead author of the book, and each author's position in the sequence of the book's authors.



# Draw an ERD ....

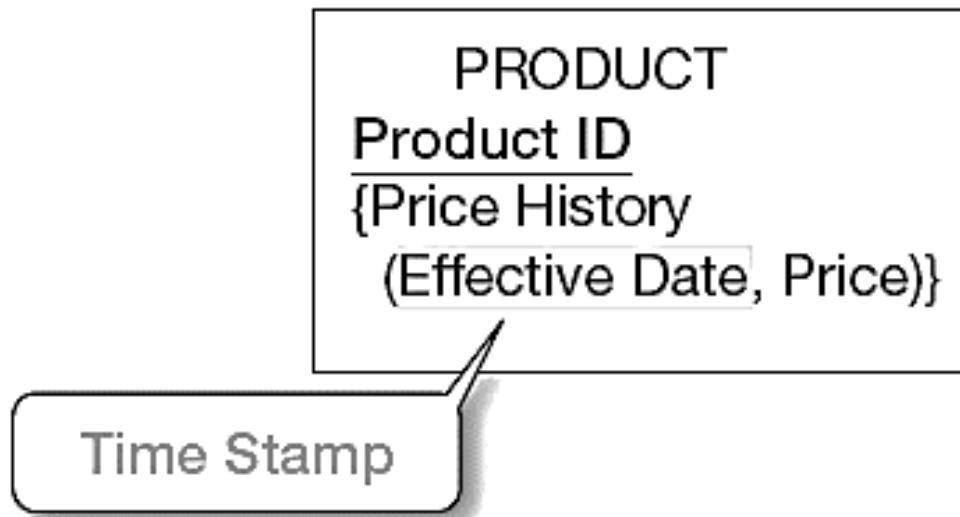
d. A book (see (a) above) can be part of a series, which is also identified as a book and has its own ISBN number. One book can belong to several sets and a set consists of at least one but potentially many books.



# A Simple Example of Time-Stamping

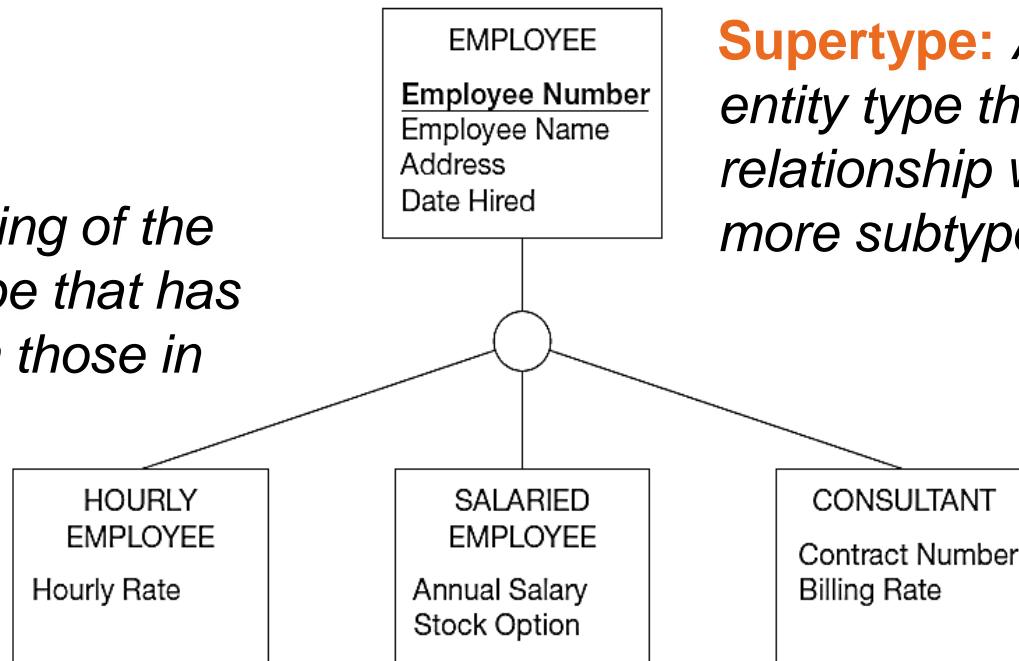
**Time stamp** – a time value that is associated with a data value, often indicating when some event occurred that affected the data value.

The Price History attribute is both multivalued and composite.



# Enhanced E-R (EER) Model

**Subtype:** A subgrouping of the entities in an entity type that has attributes distinct from those in other subgroupings

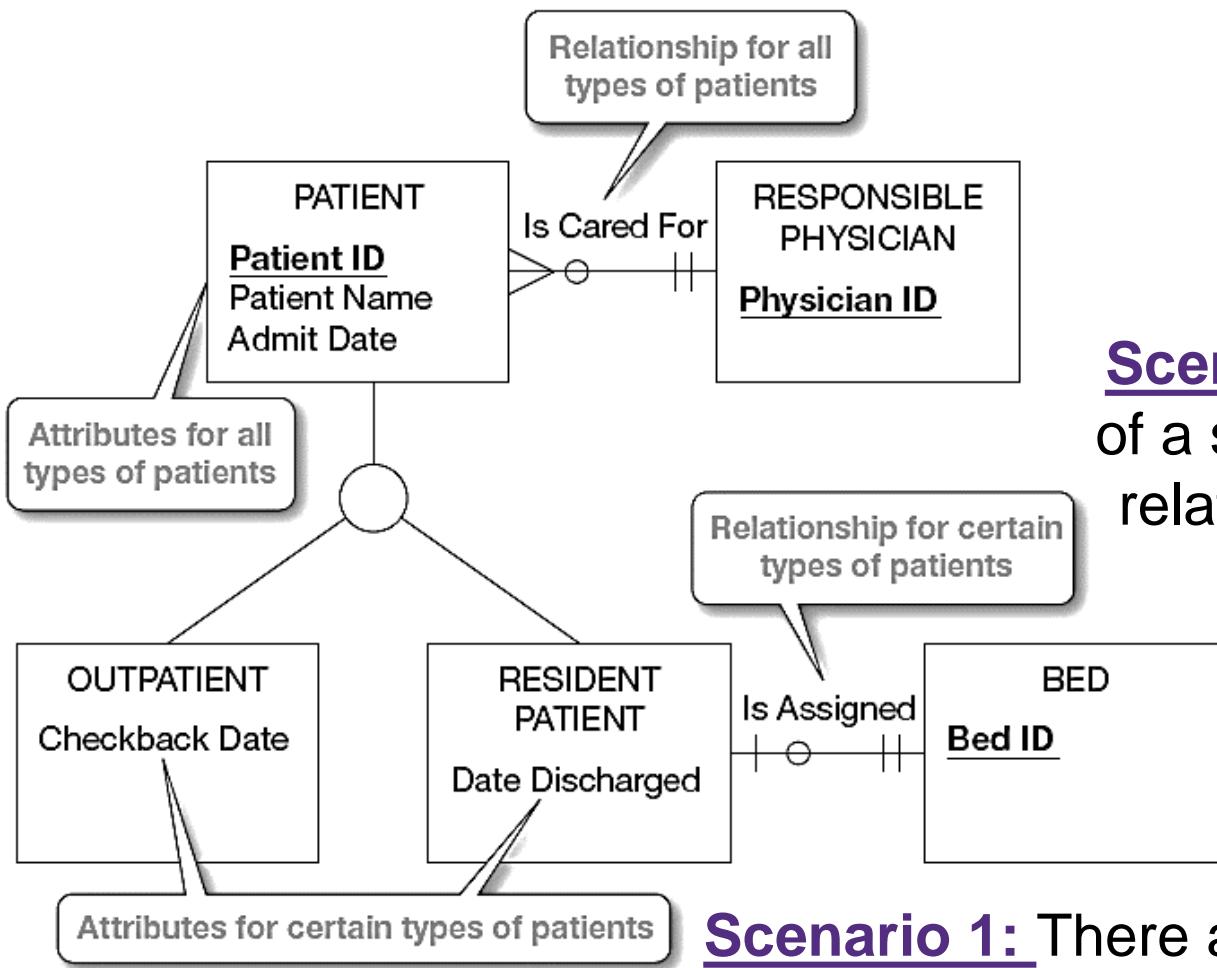


**Supertype:** A generic entity type that has a relationship with one or more subtypes

## Attribute Inheritance:

- Subtype entities inherit all attributes and relationships of the supertype
- An instance of a subtype is also an instance of the supertype

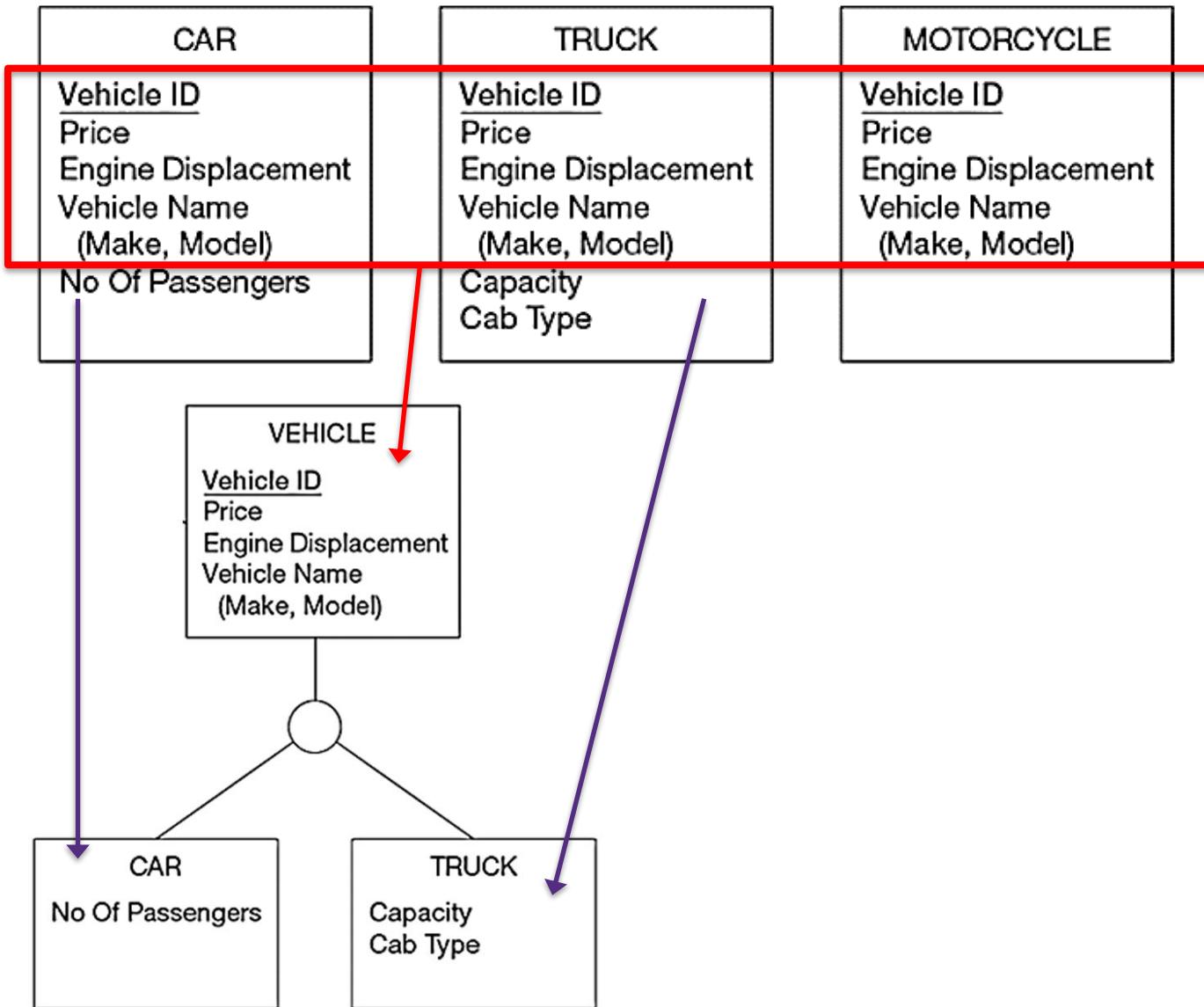
# When To Use Supertype/Subtype?



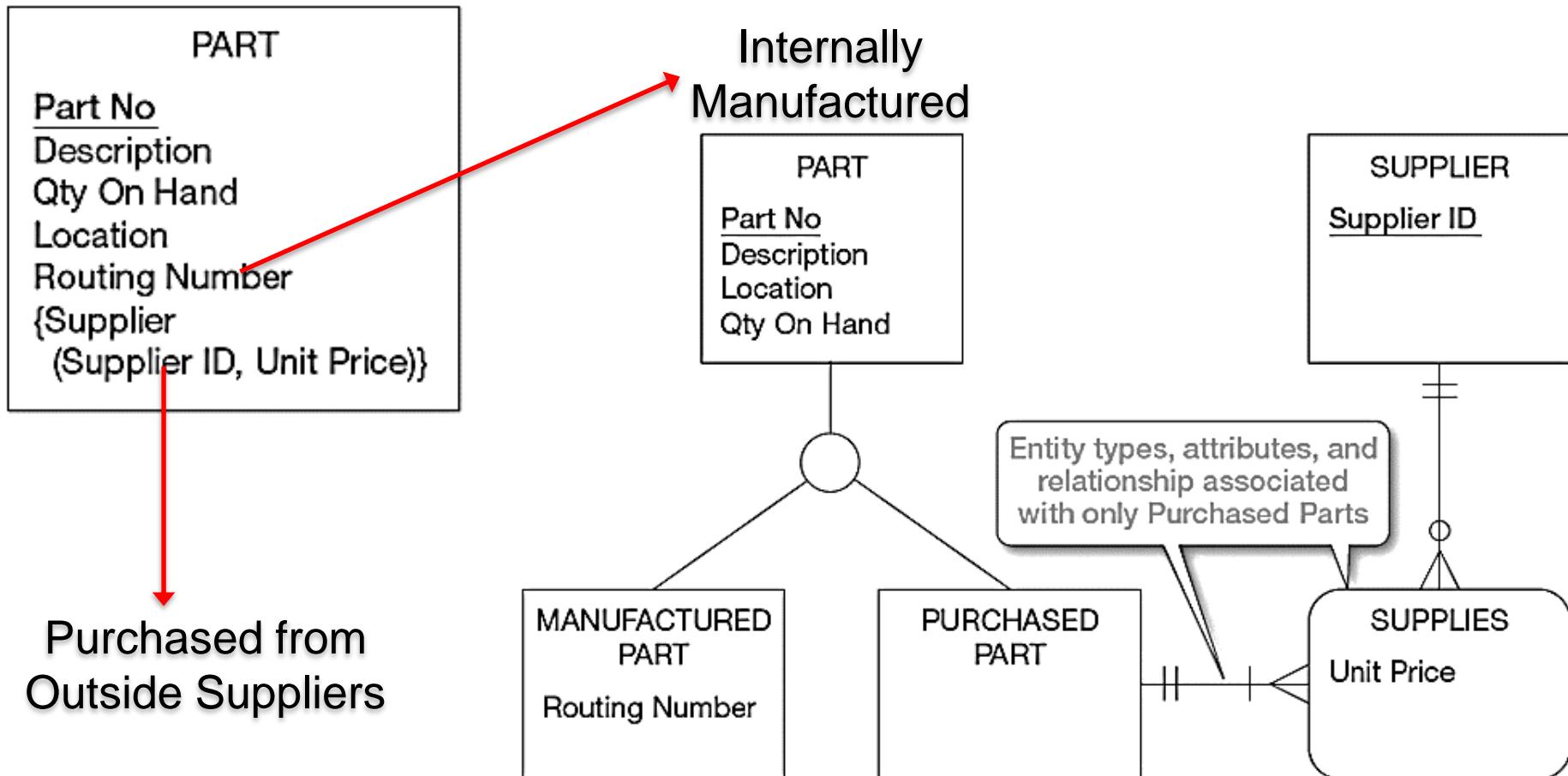
**Scenario 2:** The instances of a subtype participate in a relationship unique to that subtype.

**Scenario 1:** There are attributes that apply to some (but not all) instances of an entity type.

# Generalization – *defining supertypes from subtypes (bottom-up)*



# Specialization – *defining subtypes from supertypes (top-down)*



# Completeness Constraints

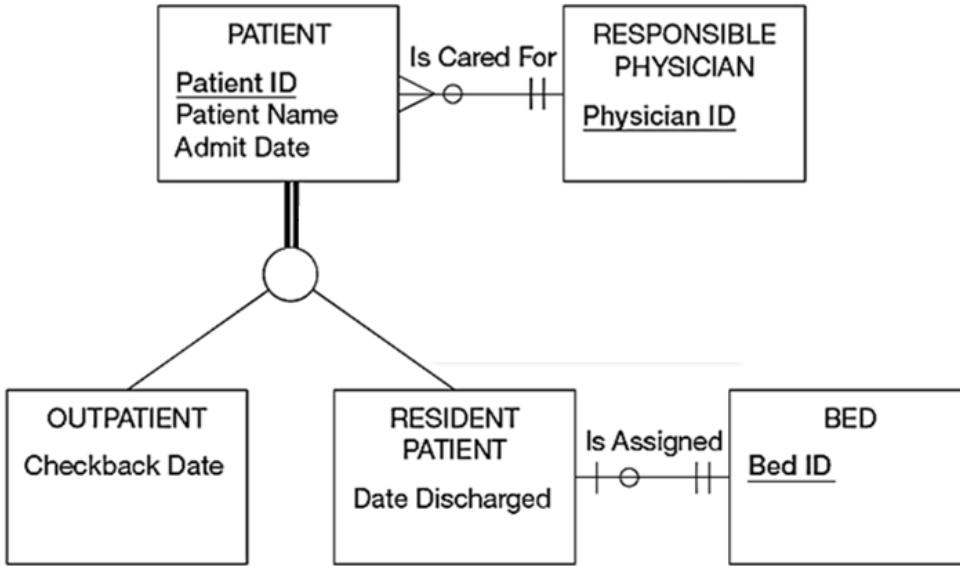
an instance of a supertype **must** also be a member of at least one subtype???

Yes

No

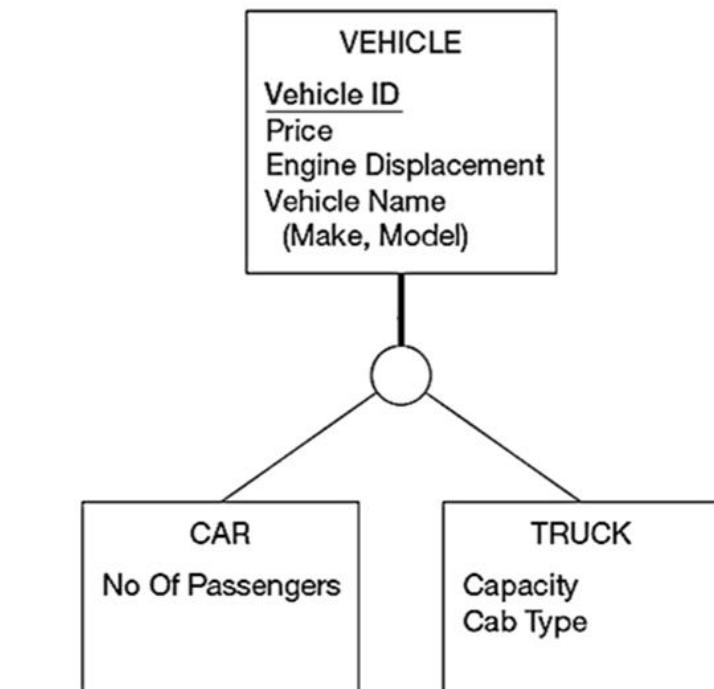
## Total Specialization

(e.g., A Patient must be either an outpatient or a Resident Patient)



## Partial Specialization

(e.g., A Vehicle can be a Car, or a Truck, but does not have to be either)



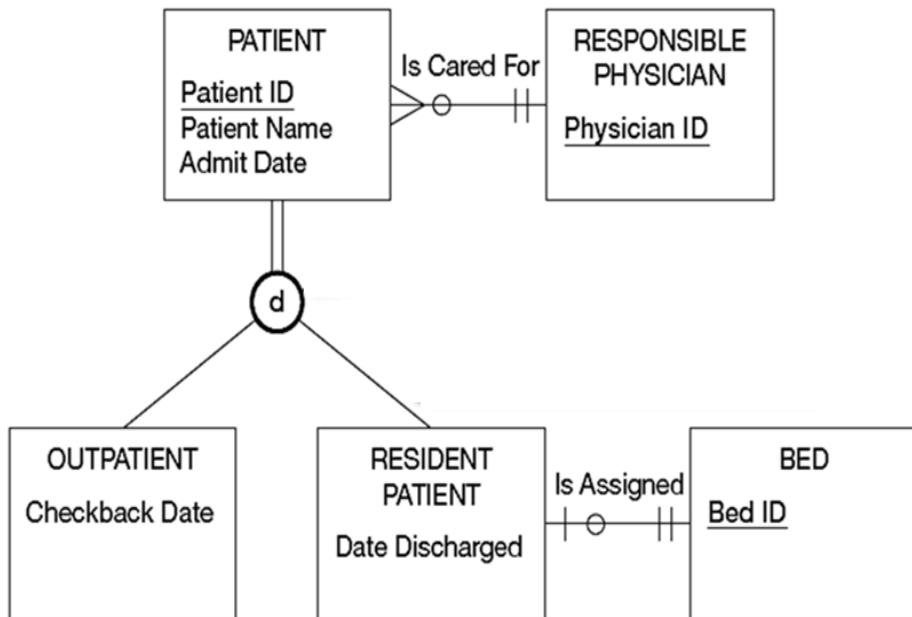
# Disjointness Constraints

an instance of a supertype may **simultaneously** be a member of multiple subtypes???

**Only One**

**Disjoint**

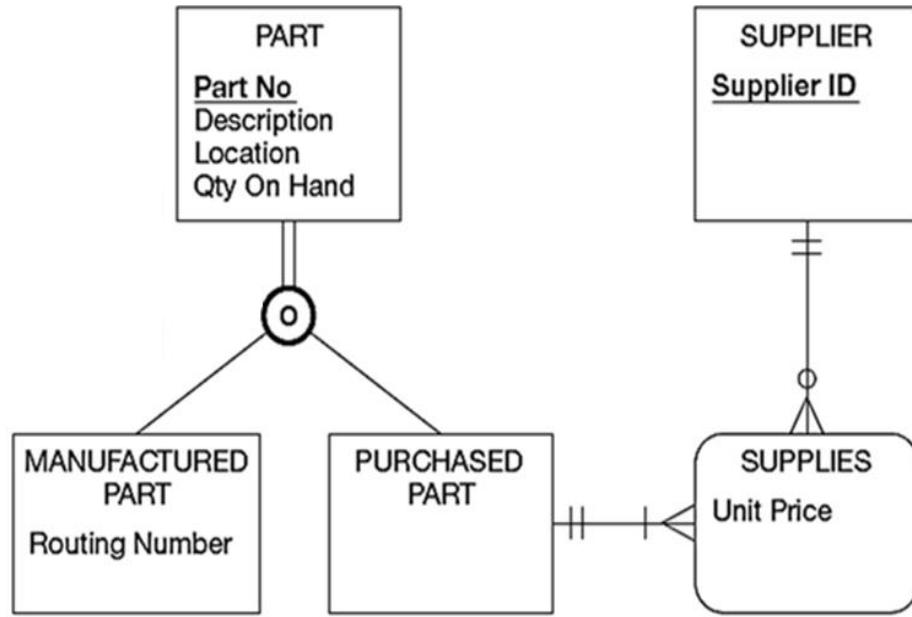
(e.g., A Patient can not be both Outpatient and Resident Patient at the same time)



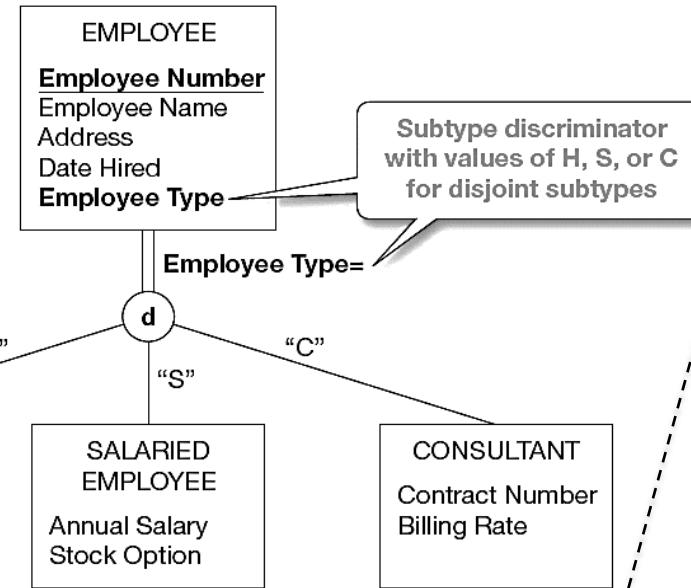
**More Than One**

**Overlap**

(e.g., A Part may be both Manufactured and Purchased at the same time)

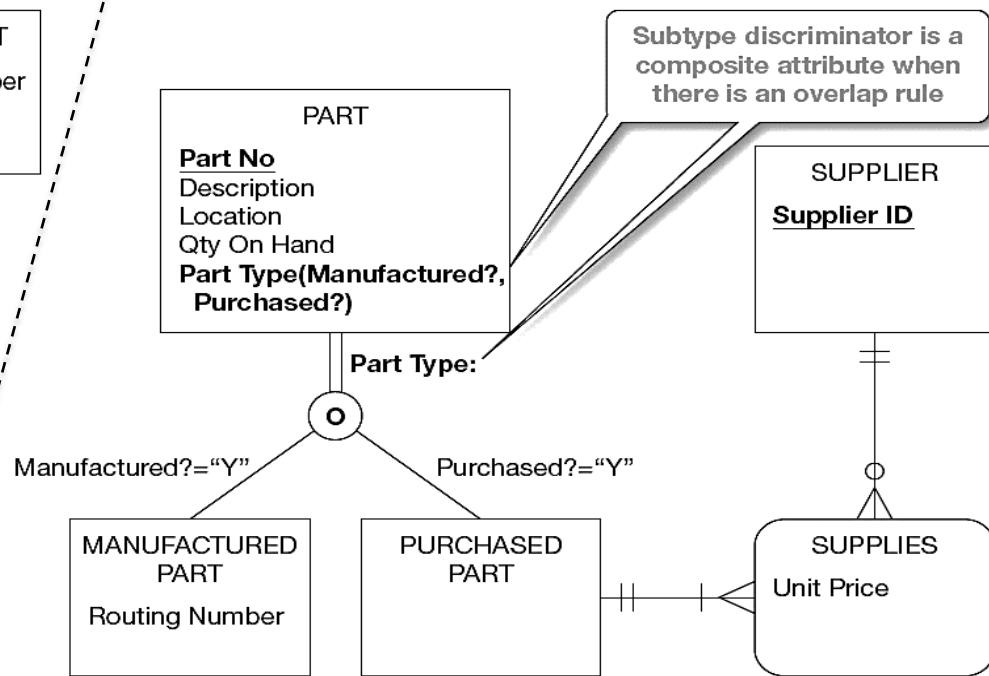


# Subtype Discriminator – *an attribute of the supertype whose values determine the target subtype(s)*

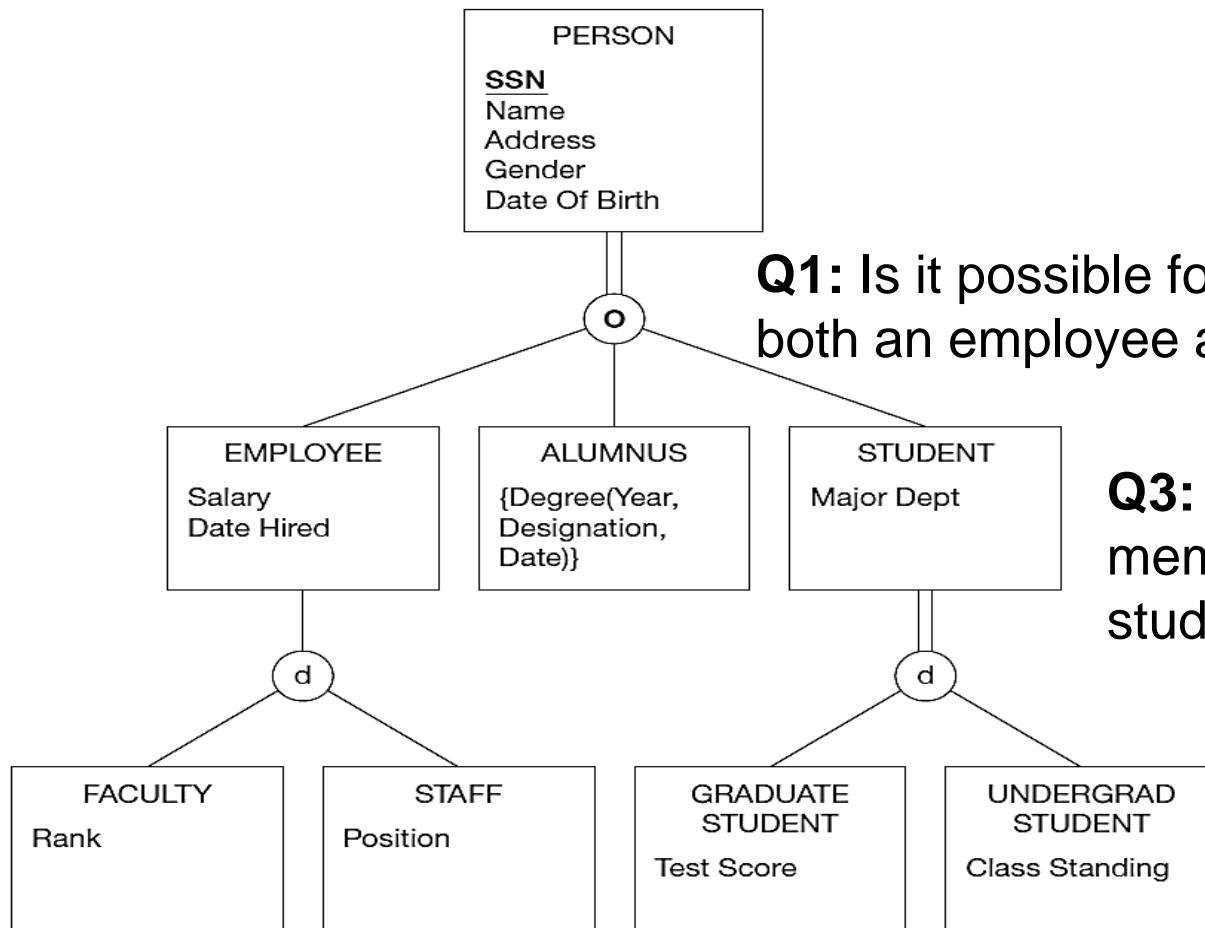


**Disjoint** – a **simple** attribute with alternative values to indicate the possible subtypes

**Overlapping** – a **composite** attribute whose subparts pertain to different subtypes. Each subpart contains a Boolean value to indicate whether the instance belongs to the associated subtype



# Supertype/Subtype Hierarchy



**Q2:** Is it possible for an employee to be something other than Faculty or Staff?

**Q1:** Is it possible for a person to be both an employee and a student?

**Q3:** Is it possible for a staff member to also be a graduate student?

**Q4:** Is it possible for someone to have more than one degree from this university?

# Steps of Designing the EER Model

1. Identify the entity types
2. Identify the relationship types and assert their degree
3. Assert the cardinality ratios and participation constraints (total versus partial participation)
4. Identify the attribute types and assert whether they are simple or composite, single- or multi-valued, derived or not
5. Link each attribute type to an entity type or a relationship type
6. Denote the key attribute type(s) of each entity type
7. Identify the weak entity types and their partial keys
8. Apply supertype/subtype through generalization/specialization, and assert the constraints such as disjoint or overlapping, total or partial

# Other Types of ERD Notations

