King Saud University

College of Engineering

IE – 462: “Industrial Information Systems”

Spring – 2021 (2^{nd} Sem. 1442H)

Chapter 3

Data Modeling and Design – p2 – E-R Diagram

Prepared by: Ahmed M. El-Sherbeeny, PhD

Lesson Overview

- Introduction – (p1)
- **E-R Diagram** – (p2)
- Case Studies – (p3)

Lesson Overview

- **E-R Diagram**
  - **Introduction to E-R Modeling**
    - Introduction
    - Entities
    - Attributes
    - Candidate Keys and Identifiers
    - Other Attribute Types
    - Relationships

Lesson Overview

- **E-R Diagram**
  - **Conceptual Data Modeling and the E-R Model**
    - Degree of a Relationship
      - Unary Relationships
      - Binary Relationships
      - Ternary Relationships
    - Cardinalities in Relationships
      - Minimum and Maximum Cardinalities
      - Alternative Cardinality System
      - Semantic Net Diagram

Lesson Overview

- **E-R Diagram**
  - **Conceptual Data Modeling and the E-R Model**
    - Naming Relationships
    - Associative Entities

INTRODUCTION TO E-R MODELING

Introduction

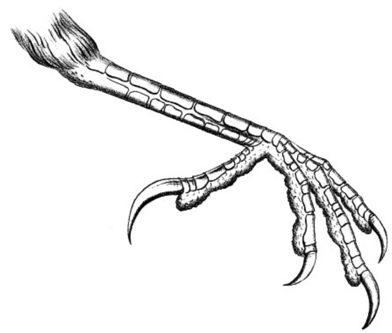
Introduction to E-R Modeling

- Purpose of E-R modeling is to design a conceptual schema (model) of entities *and* their relationships for an organization/business
- **Entity-relationship data model** (E-R model): detailed, logical representation of:
  - **data entities**
  - **relationships**, and
  - **attributes**: they represent properties of *both* the entities and their relationships

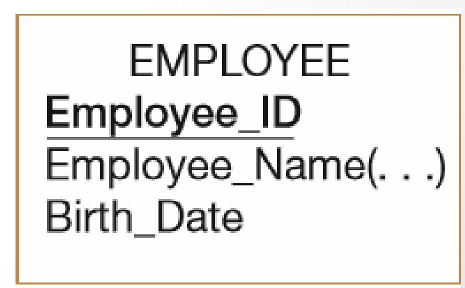
Introduction to E-R Modeling

Entity-relationship diagram (ERD):

- graphical representation of an E-R model
- utilizes several notations to show data in terms of the entities and relationships described by that data
  - o notation uses mostly "**crow's foot**" symbols
  - o another notation uses letters & numbers: "**Chen**" notation

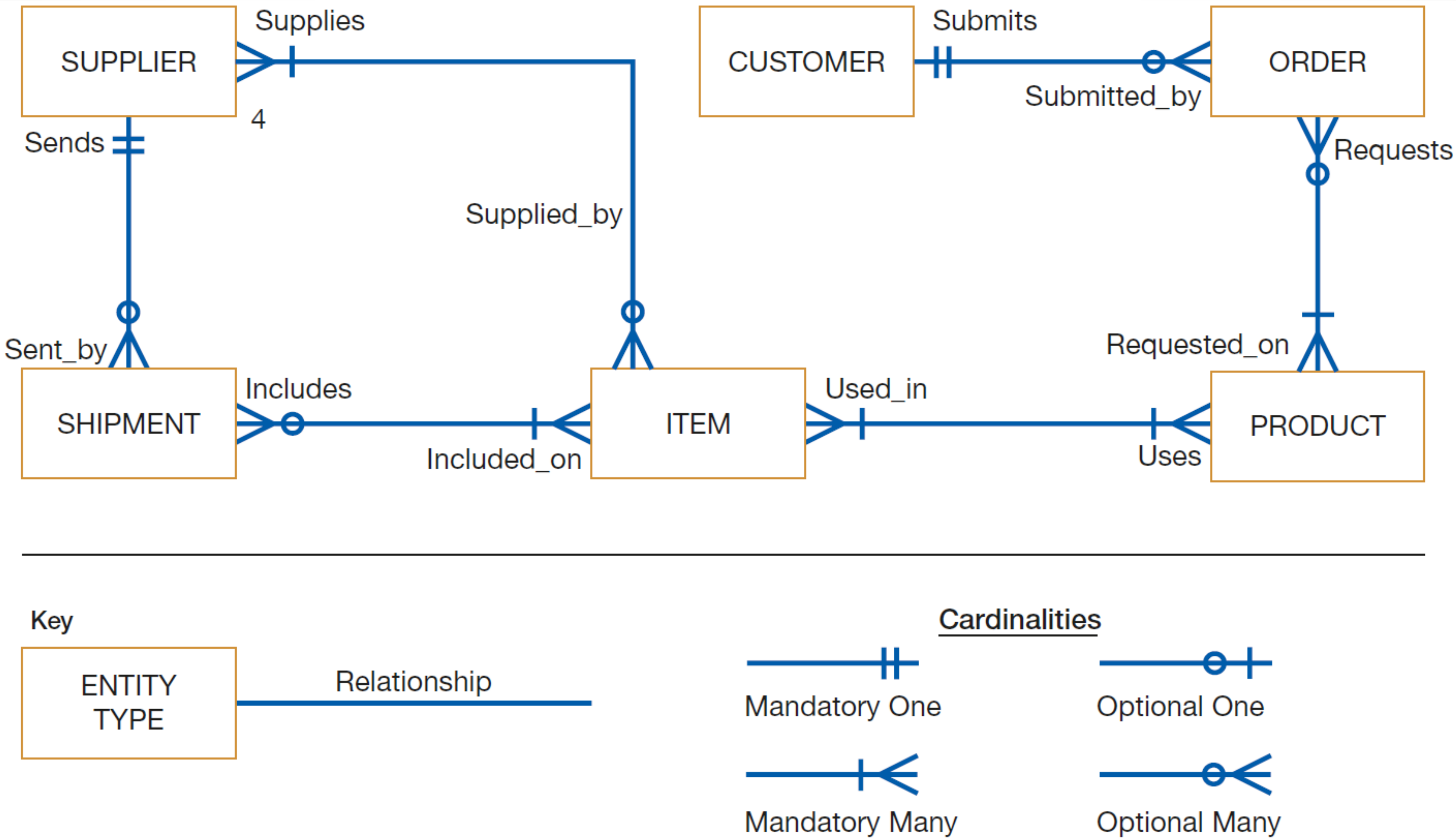


- 1st notation places data attribute names within entity rectangles

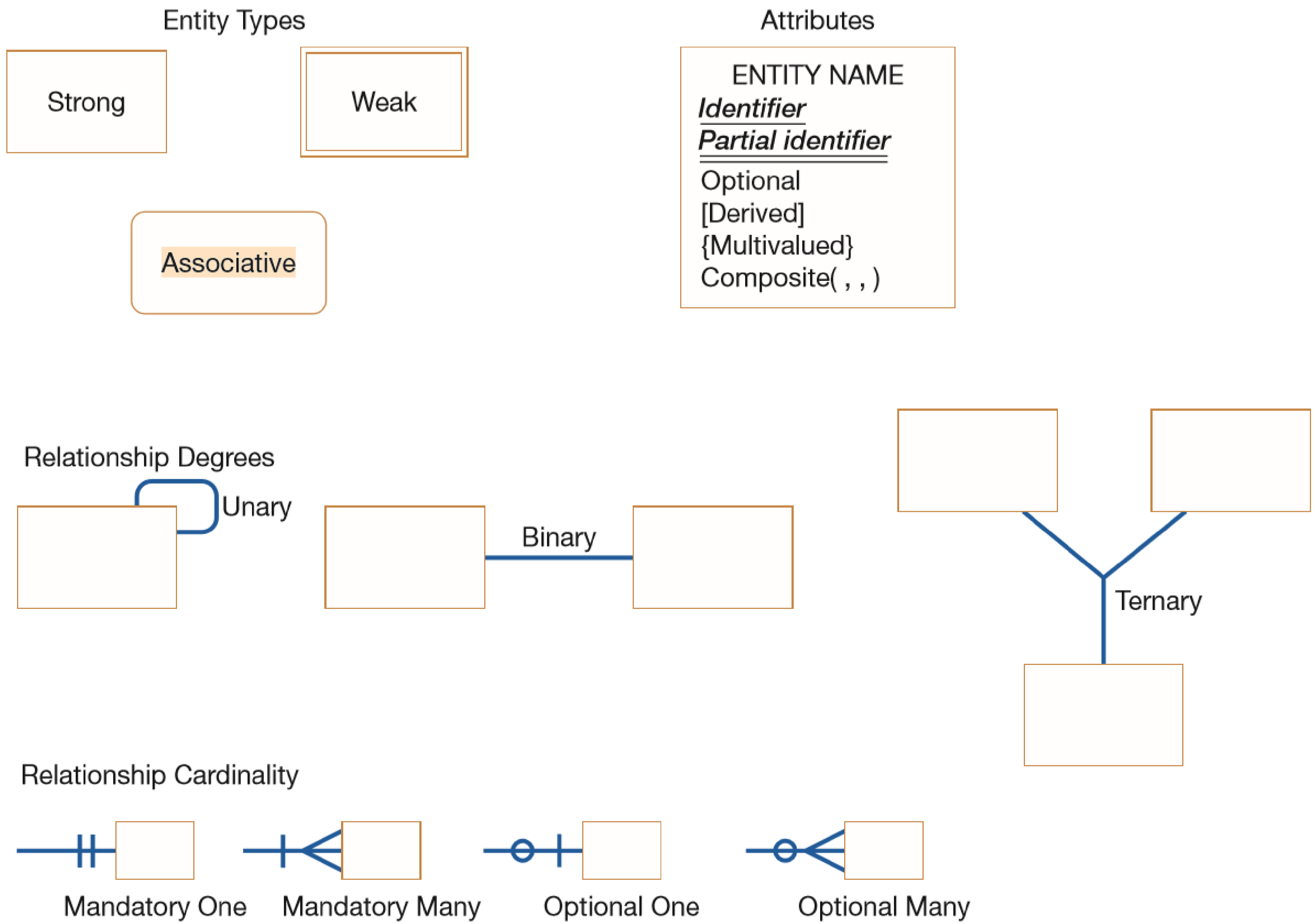


- see next 4 slides for notations, which will be explained in detail in following sections

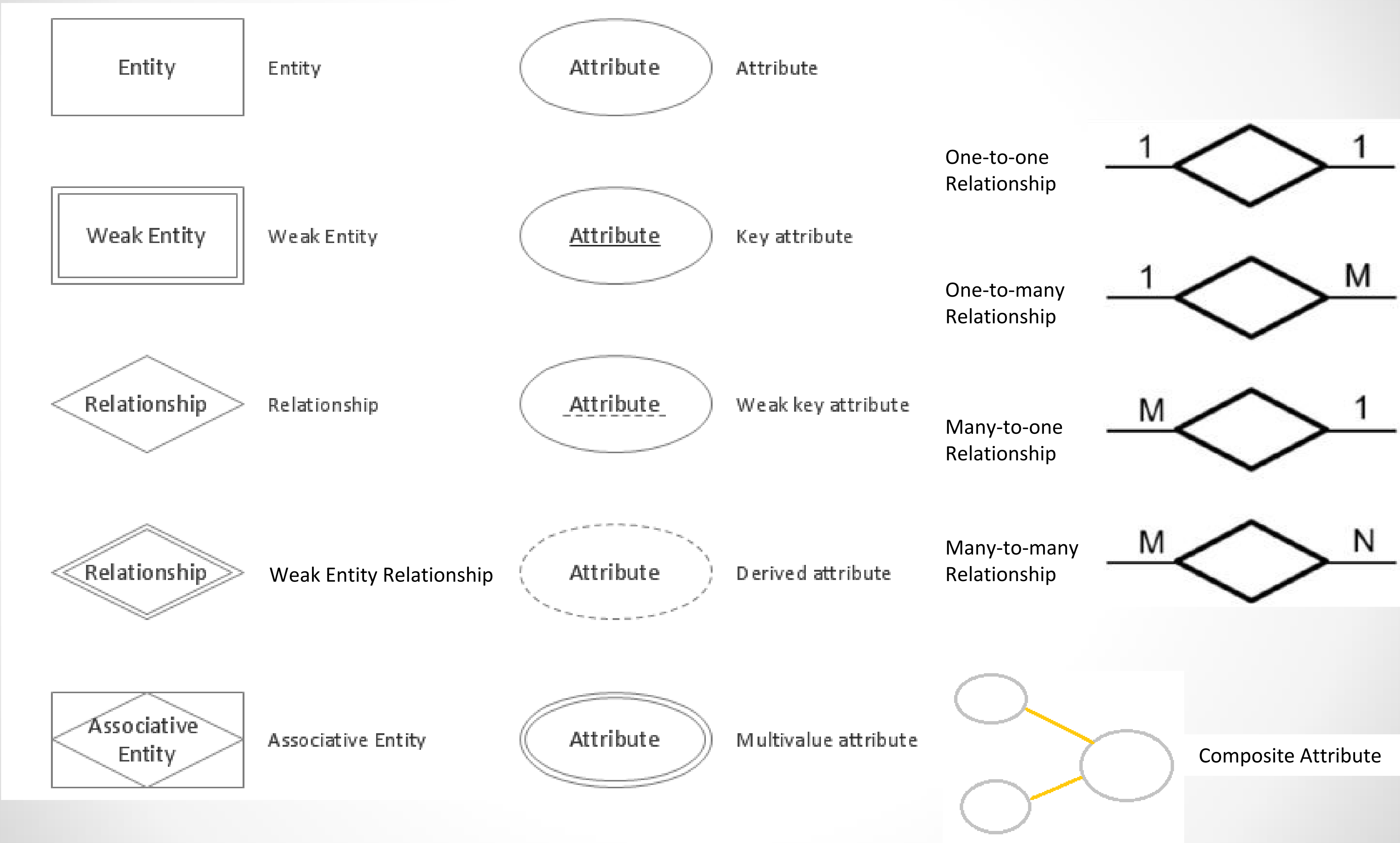
Basic E-R Notation (Crow's Foot)



Basic E-R Notation (Crow’s Foot)



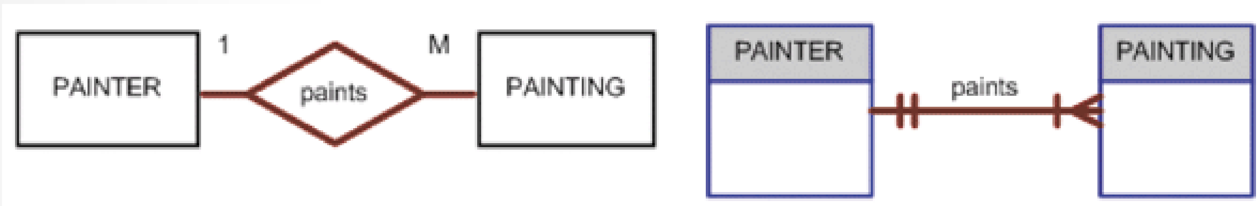
Basic E-R Notation (Chen)



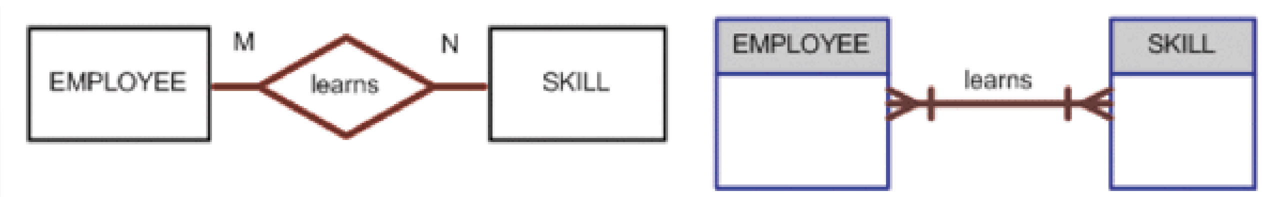
Basic E-R Notation (Crow's Foot vs Chen)

Chen Notation | *Crow's Foot Notation*

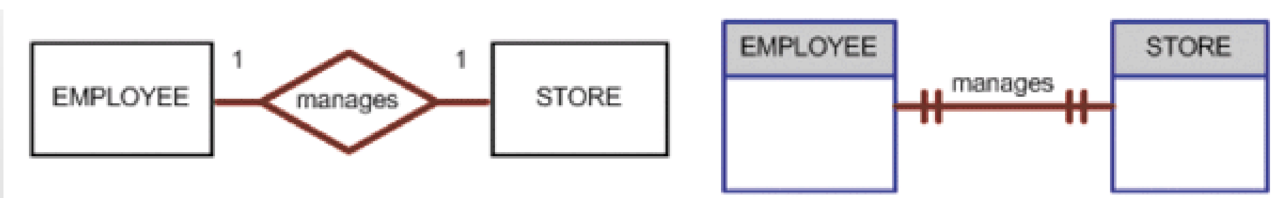
A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.



A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.



A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.



Entities

Entities

Entity:

- Class of *persons*, *places*, *objects*, *events*, or *concepts* for which the organization wishes to maintain data
- Represented by *Q1* in Table 8-1
- Each entity must have a *unique identity* that distinguishes it from each other entity

Entities

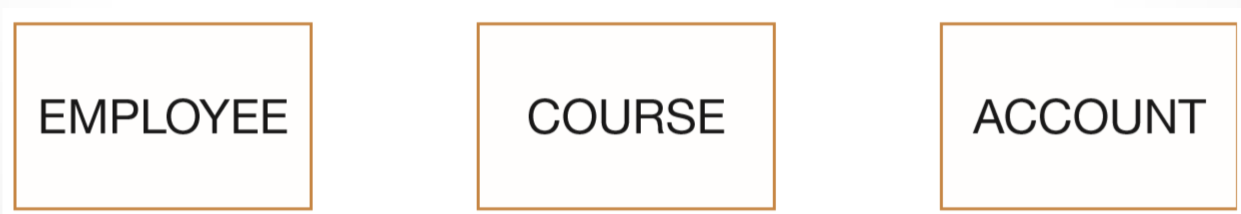
Examples of Entities (cont.):

- **Persons**: agency, contractor, customer, department, division, employee, instructor, student, supplier
- **Places**: sales region, building, room, branch office, campus, store, warehouse, state, shop floor
- **Objects**: book, machine, part, product, raw material, software license, software package, tool, vehicle model
- **Events**: application, award, cancellation, class, flight, order, registration, renewal, requisition, reservation, sale, trip, assignment
- **Concepts**: account, block of time, bond, course, fund, qualification, stock, work center

Entities

Entity *Types* vs. Entity *Instances*:

- Important to distinguish between *entity types* and *entity instances*
- **Entity type** (aka **entity class** or –simply– **entity**):
    - o collection of entities that share common properties/characteristics
    - o *each entity* type in an E-R model is given a name
    - o name is placed inside a rectangle representing the entity
    - o each entity is described just *once* in a data model



Entities

Entity *Types* vs. Entity *Instances* (cont.):

- **Entity instance** (aka **instance**):
  - a single occurrence of an entity
  - *many* instances of an entity type may be represented by data stored in the database

entity

instance

Student ID	Last Name	First Name
2144	Arnold	Betty
3122	Taylor	John
3843	Simmons	Lisa
9844	Macy	Bill
2837	Leath	Heather
2293	Wrench	Tim

Entities

Common Mistakes with Data Entities:

- Many people confuse
  - o data *entities* with *sources/sinks* or system outputs,
  - o *relationships* with *data flows*
- Avoid this problem with a simple rule:
  - o true data entity will have *many possible instances*,
  - o each instance has a distinguishing characteristic
- Example below, “sorority expense system”:
  - o do we need to keep track of data about the treasurer?

Entities

Naming Entity Types:

- Should use all *capital letters*
  - e.g. EMPLOYEE
- Should be named by a *singular noun*
  - e.g. CUSTOMER, STUDENT, or AUTOMOBILE
- Use *simple*, *concise* nouns
  - e.g. use REGISTRATION instead of STUDENT REGISTRATION FOR CLASS
- Name should be *descriptive/specific* to company
  - e.g. instead of just using ORDER, use PURCHASE ORDER (to distinguish between it and CUSTOMER ORDER)

Attributes

Attributes

Attribute:

- A named *property* or *characteristic* of an entity that is of interest to the organization
- Represented by *Q3* in Table 8-1
- Some typical entity types and associated attributes:
  - STUDENT: Student_ID, Student_Name, Home_Address, Phone_Number, Major
  - AUTOMOBILE: Vehicle_ID, Color, Weight, Horsepower
  - EMPLOYEE: Employee_ID, Employee_Name, Payroll_Address, Skill

Attributes

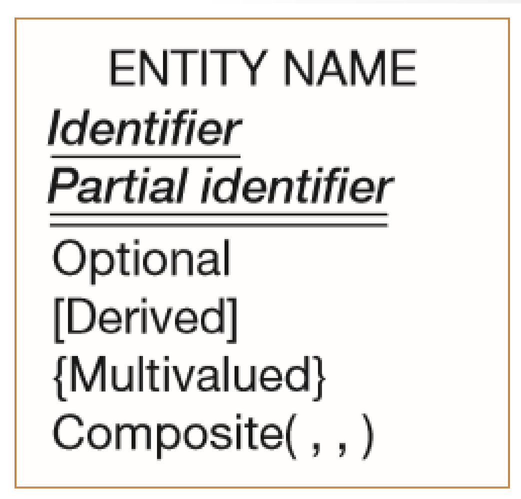
Naming Attributes:

- Use initial capital letter, followed by lowercase letters
- Use nouns for names; e.g. Age
- Use underscores to separate words (optional);
  e.g. Customer_ID, Product_Minimum_Price
- Attribute name should be *unique*:
  - no 2 attributes of same entity type may have the same name
  - preferable no two attributes have the same name (i.e. across all entity types)
- Follow a standard format for naming attributes:
  - e.g. using Student_GPA
    as opposed to GPA_of_Student

Attributes

Using Attributes in E-R Diagram:

- Place the name inside the rectangle for associated entity
- We use different notations to distinguish between different types of attributes (to be discussed next)



Candidate Keys and Identifiers

Candidate Keys and Identifiers

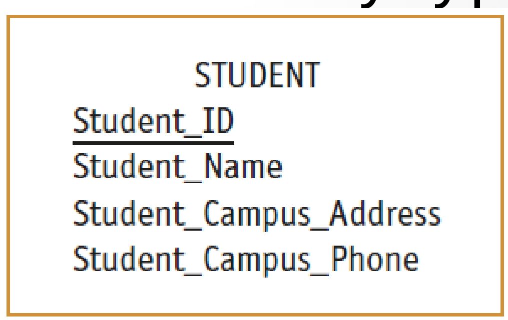
Candidate Key (aka **Primary key**):

- It's an attribute (or combination of attributes) that *uniquely* identifies each instance of an entity type
- Represented by *Q2* in Table 8-1
- e.g. candidate key for a STUDENT entity type might be Student_ID

Candidate Keys and Identifiers

Identifiers:

- Some entities may have > 1 possible candidate key; e.g. for EMPLOYEE data entity:
  - o possible candidate key: Employee_ID
  - o another possible candidate key: Employee_Name and Address (assuming no two employees with the same name live at the same address)
  - o designer must choose one of the candidate keys as identifier thus:
- Identifier: candidate key that has been selected as the *unique*, *identifying characteristic* for entity type
  - o it is represented by placing a *solid underline* below identifier



Candidate Keys and Identifiers

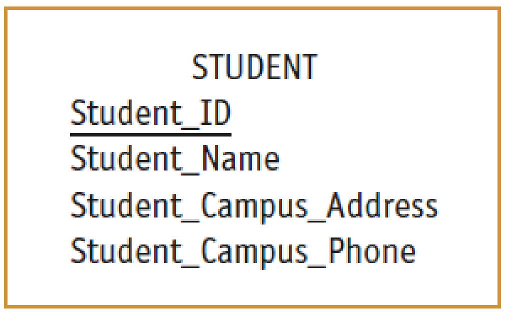
Criteria for Selecting Identifiers:

- Choose a candidate key that will *not change* its value over life of each instance of the entity type
  - e.g. don't pick identifier for EMPLOYEE: combination of Employee_Name and Payroll_Address
- Choose candidate key so that, for each instance of the entity, the attribute is guaranteed to have:
  - valid values and
  - not be 'null' (note, special controls in data entry can eliminate possibility of errors, e.g. use of '*')

Candidate Keys and Identifiers

Criteria for Selecting Identifiers (cont.):

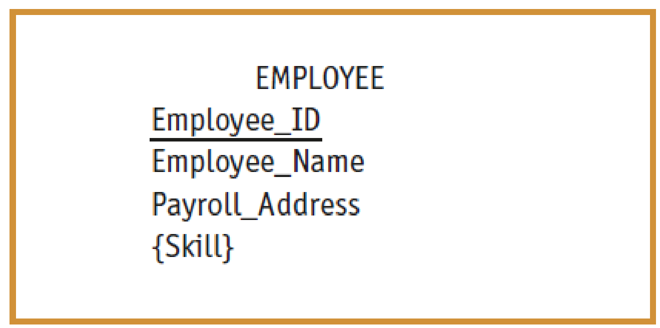
- Avoid so-called “intelligent identifiers”
  - o e.g. first 2 digits of a key for a PART entity may indicate the warehouse location
  - o note that such codes are often modified, and this would make primary key values invalid
- Example here:
  - o representation for a STUDENT entity type using E-R notation
  - o STUDENT has:
    - a simple identifier, Student_ID, and
    - 3 other simple attributes



Other Attribute Types

Other Attribute Types

- **Single-valued attribute:**
  - o attribute that may take one entry in each instance of that attribute
  - o e.g. there is only 1 employee ID number to be entered in each instance of the attribute Employee_ID
- **Multi-valued attribute**:
  - o an attribute that may take on > 1 value for each entity instance
  - o e.g. Skill is a multivalued attribute (since each employee can have > 1 skill)
  - o special symbol indicates that it is multivalued: { }



Other Attribute Types

Repeating group:

- A set of two or more multi-valued attributes that are logically related

EMPLOYEE
Employee_ID
{Dep_Name,
Dep_Age,
Dep_Relation}

Repeating group of dependent data

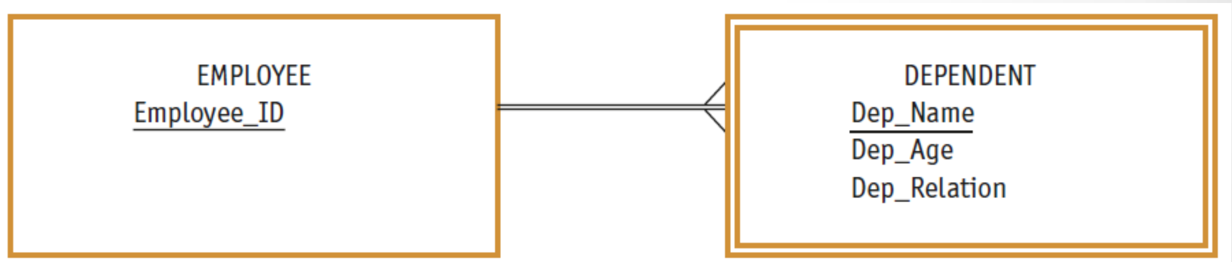
- e.g. employee entity with multivalued attributes for data about each employee's dependents:
  - data includes: dependent name, age, and relation to employee
  - dependents: spouse, child, parent, etc.
  - data are multivalued attributes about employee
  - we show this by using one set of curly brackets around the data that repeats together
  - we call this a *repeating group*

Other Attribute Types

Weak entity:

- Second approach to representing a repeating group:
  - o consider dependents as *entities*
  - o we separate the repeating data into another entity, called a *weak* (or *attributive*) *entity*
  - o weak entity is designated by:
    - *rectangle* with a *double line border* and
    - relationship to link the weak entity to its associated regular entity (using *double line*)

Weak entity for dependent data

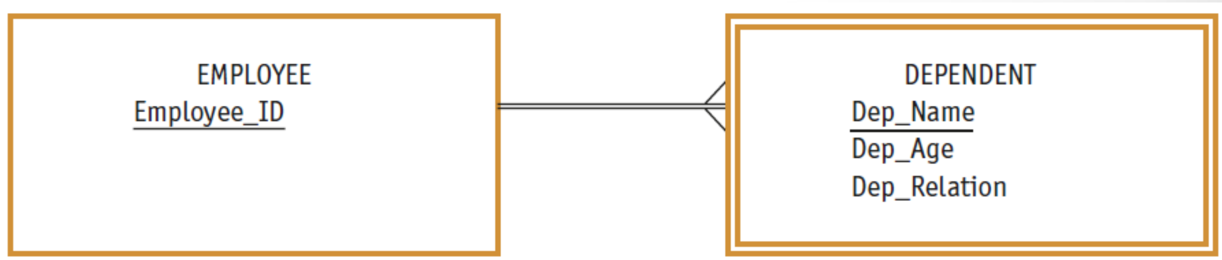


Other Attribute Types

Weak entity (cont.):

- Examine example below:
  - o use a weak entity, DEPENDENT
  - o establish relationship between DEPENDENT and EMPLOYEE
  - o crow's foot next to DEPENDENT: there may be many DEPENDENTs for the same EMPLOYEE
  - o identifier of DEPENDENT: dependent's name + ID of the employee, or use a double underline for Dep_Name to designate it as a *partial identifier*

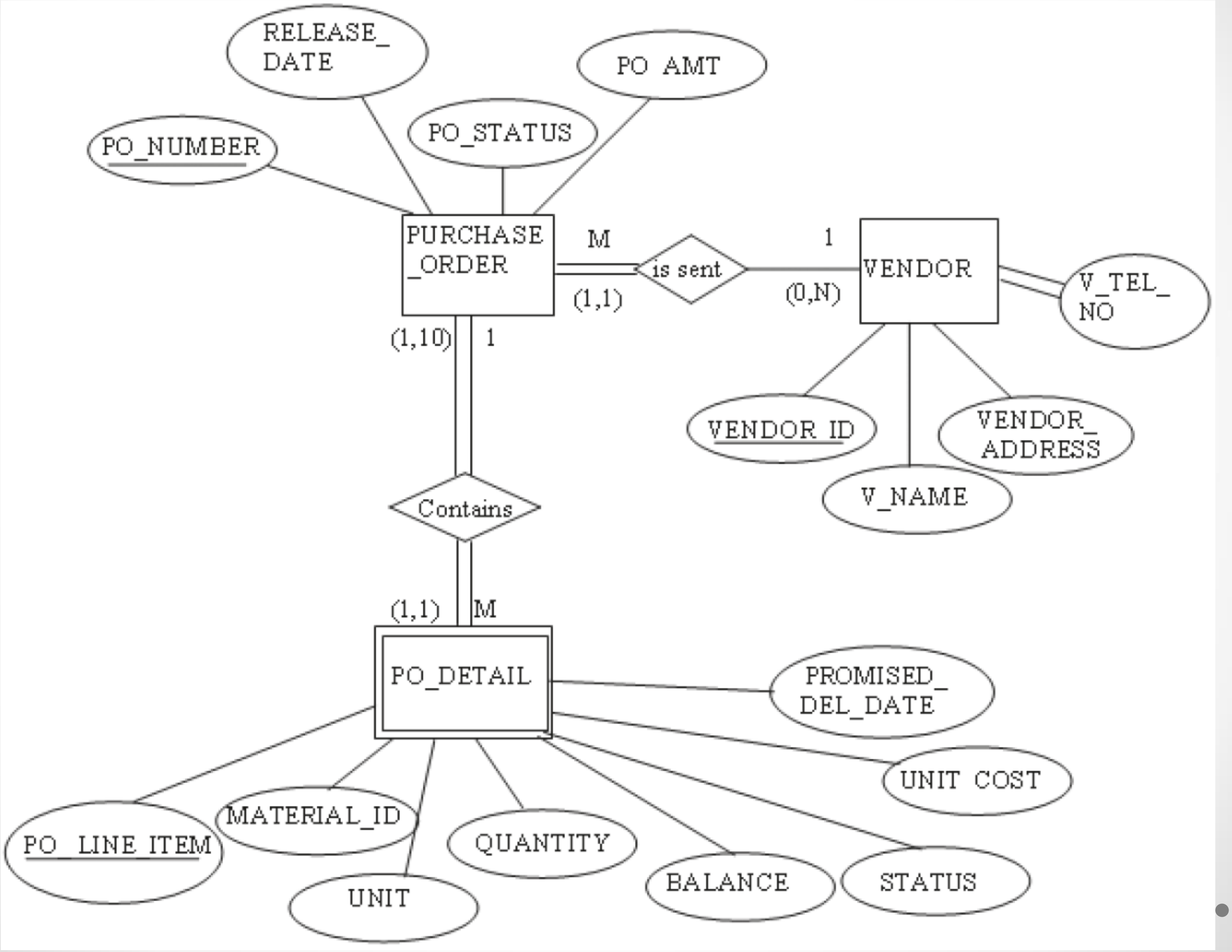
Weak entity for dependent data



Weak and Strong Entities (Example)

- E-R diagram uses a box to represent an entity set (PURCHASE_ORDER, PO_DETAIL, and VENDOR)
- E-R diagrams distinguish between *weak* and *strong* entities
  - entity is *weak* if its existence is dependent on the existence of another entity
  - e.g. of this occurs in the case of PO_DETAIL: PO_DETAIL is dependent on the existence of PURCHASE_ORDER

Weak and Strong Entities (Example) – Cont.



Other Attribute Types

- **Required attribute**: an attribute that *must* have a value for every entity instance
  - o shown in bold letters (list!)
- **Optional attribute**: an attribute that *may not* have a value for every entity instance
  - o shown in normal letters (list!)

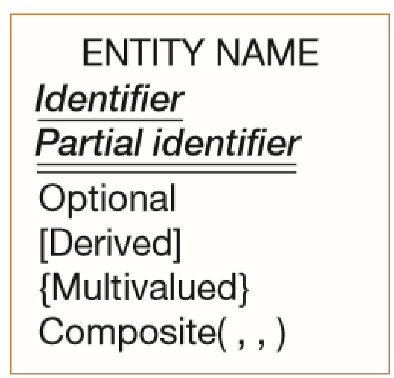
EMPLOYEE

Employee_ID
Employee_Name(First_Name, Last_Name)
Date_of_Birth
[Employee_Age]

Required, optional, composite, and derived attributes

Other Attribute Types

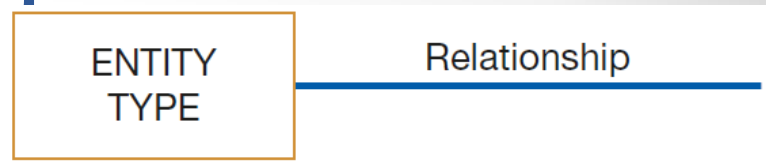
- **Composite attribute**: an attribute that has meaningful *component* parts
    - o e.g. Name or Address
    - o components are shown between brackets ( ) (list!)
- **Derived attribute**: an attribute whose value can be computed from *related attribute* values
    - o shown inside square brackets: [ ] (list!)
- Notations for each attributes type is shown below



Required, optional, composite, and derived attributes

Relationships

Relationships



Relationship:

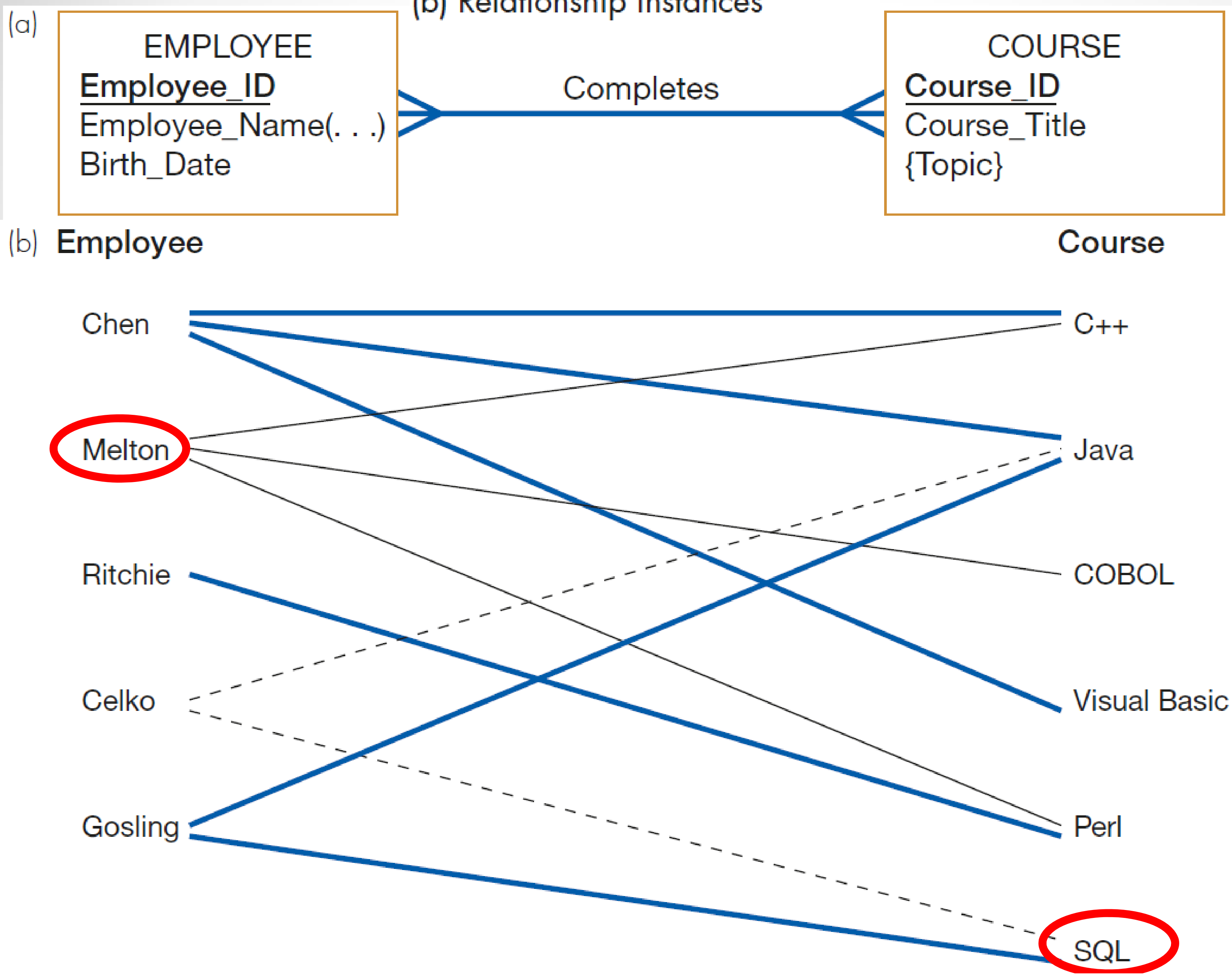
- It is association between the instances of one or more entity types that is of interest to the organization
- It is the ‘glue’ that holds together the various components of an E-R model
- Represented by *Questions 5, 7, and 8* in Table 8-1
- Labeled with verb phrases since usually means that an event has occurred
- Note, some standards use *two verb phrases* for a relationship name (so it can be read in two directions), while some only use *one verb phrase*

Relationships

Relationship (cont.):

- Consider example shown on next slide:
  - o training department in a company wants to track which training courses employees have completed
  - o this leads to a relationship called Completes between the EMPLOYEE and COURSE entity types
  - o this is a *many-to-many* relationship:
    - each *employee* may complete >1 *course*
    - each *course* may be completed by >1 *employee*
  - o we can use Completes relationship to determine:
    - courses a given employee has completed
    - identity of each employee who has completed a particular course

Relationship type and instances
(a) Relationship type (Completes)
(b) Relationship instances

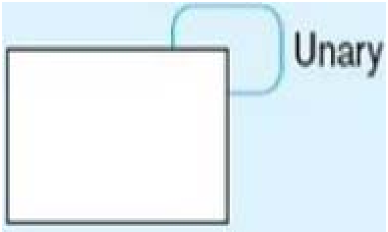


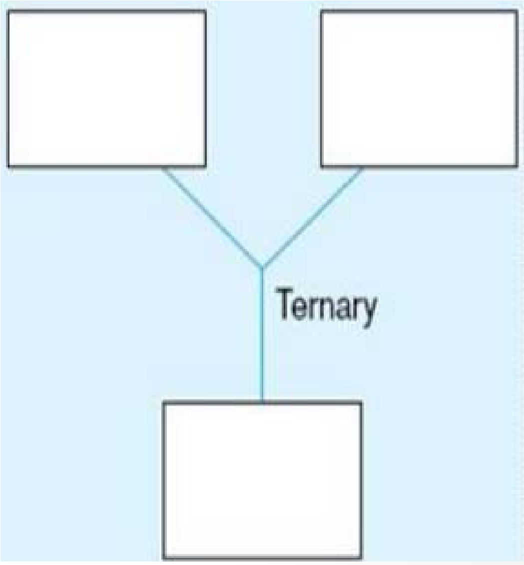
CONCEPTUAL DATA MODELING AND THE E-R MODEL

Degree of a Relationship

Degree of a Relationship

- Degree of a relationship is a measure of the number of entities sharing the same association
- There are four cases:
  - o **unary** relationships
  - o **binary** relationships
  - o **ternary** relationships
  - o **n-ary** relationships





Degree of a Relationship

Unary Relationships

- A unary relationship is a relationship between the instances of one entity type (i.e. *within a single entity*)
  - i.e. the entity has a relationship with itself
  - aka ***recursive relationship***
- Examples:
  - Is_married_to: *one-to-one* relationship between instances of PERSON entity
  - Manages: *one-to-many* relationship between instances of EMPLOYEE entity (used to identify employees who report to a particular manager and vice versa)

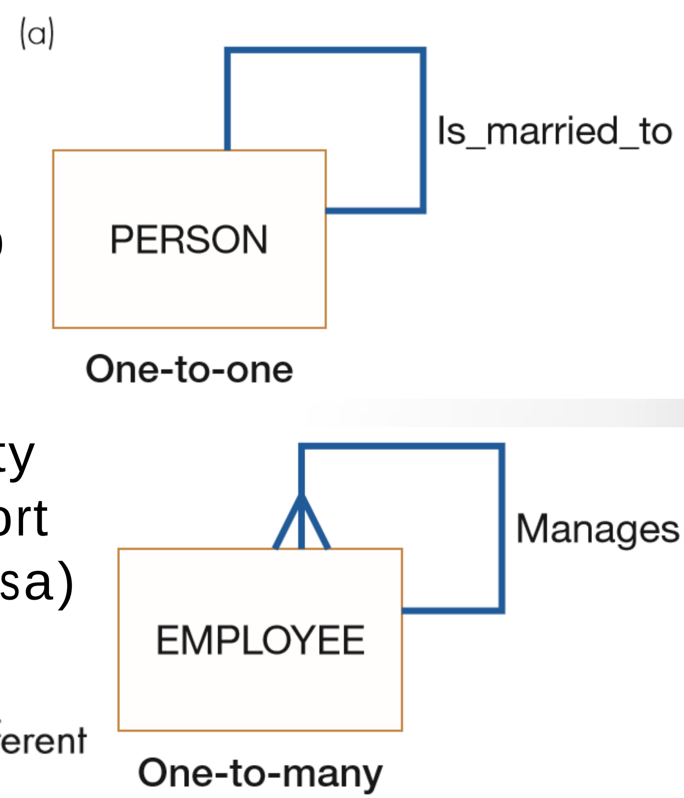


FIGURE 8-11
Examples of relationships of different degrees
(a) Unary relationships

Degree of a Relationship

Unary Relationships (cont.)

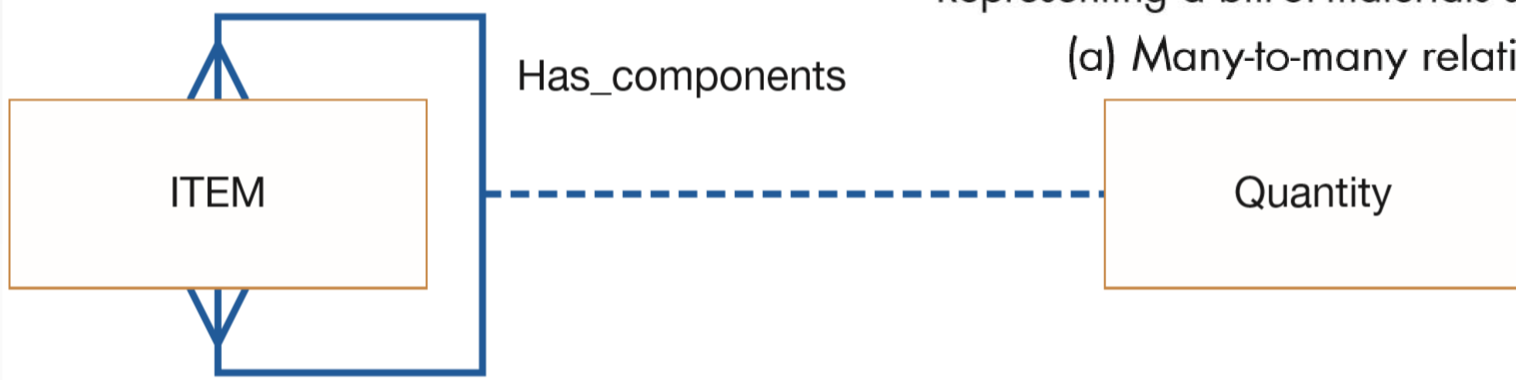
- Next slide: example of another common unary relationship: ***bill-of-materials structure***
  - o Many manufactured products are made of subassemblies
  - o Subassemblies in turn are composed of other subassemblies and parts, and so on
  - o Figure 8-12a: shows this as *many-to-many unary* relationship
    - relationship name: Has_components
    - attribute Quantity: property of the relationship; indicates # of each component that is contained in a given assembly
  - o Figure 8-12b: 2 occurrences of this structure
    - easy to see associations are in fact many-to-many
    - e.g. TX100 consists of items BR450 (Qty 2) & DX500 (Qty 1)
    - also, some components are used in several higher-level assemblies (e.g. WX240 used in item MX300 & WX340)

Degree of a Relationship

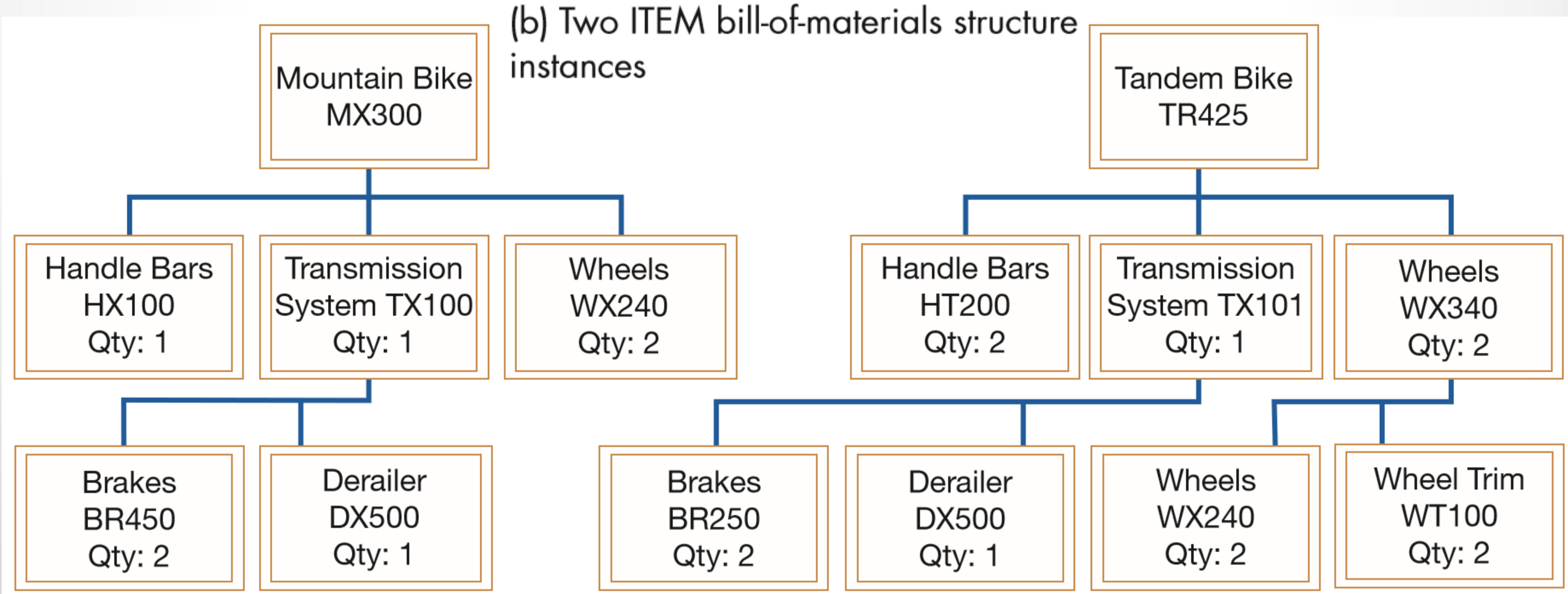
Unary Relationships (cont.)

FIGURE 8-12
Representing a bill-of-materials structure

(a) Many-to-many relationship



(b) Two ITEM bill-of-materials structure instances



Degree of a Relationship

Binary Relationships

- Binary relationship:
    - Exists when two entities have an *associated relationship* (i.e. relationship between instances of two entities)
    - It is the most common relationship used in data modeling
- Three example are shown on the next slide

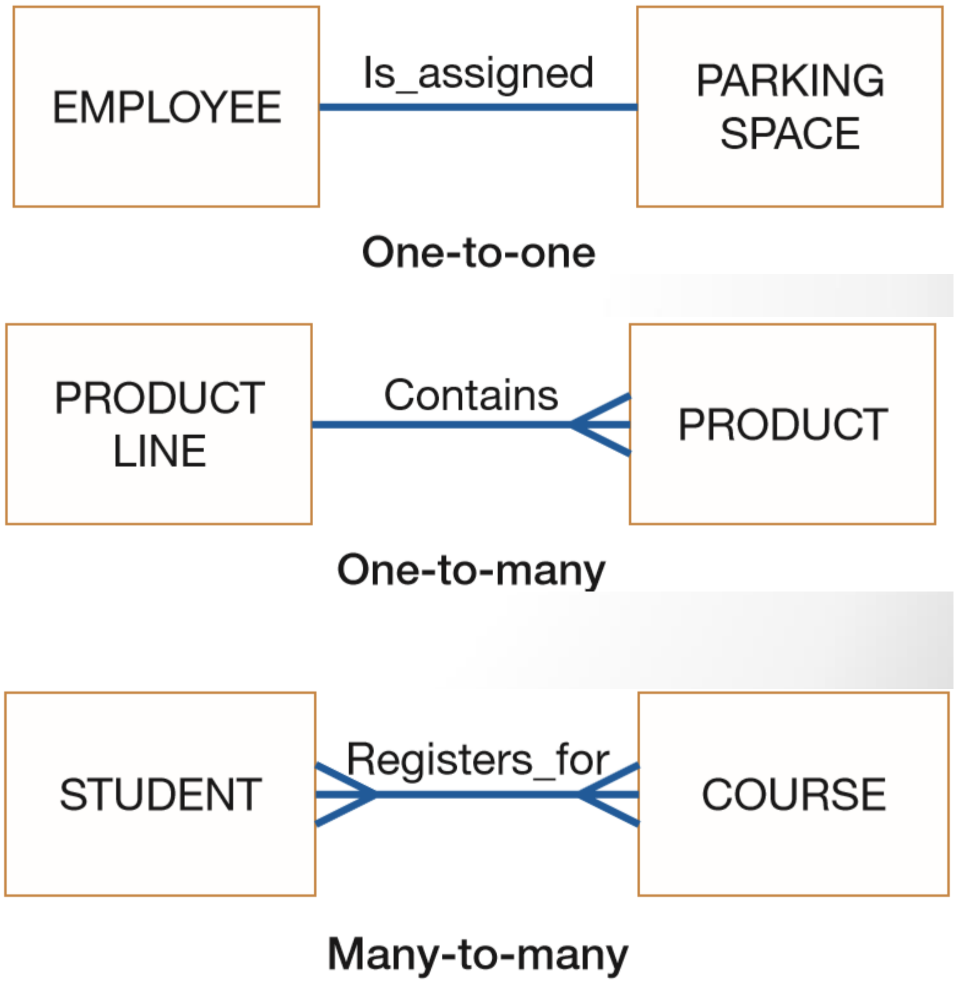
Degree of a Relationship

Binary Relationships (cont.)

- Examples (cont.)
  - *one-to-one*: employee is assigned 1 parking place, each parking place is assigned to 1 employee
  - *one-to-many*: product line may contain several products, and each product belongs to only 1 product line
  - *many-to-many*: student may register for > 1 course, each course may have many student registrants

FIGURE 8-11
Examples of relationships of different degrees

(b) Binary relationships



Degree of a Relationship

Ternary Relationships

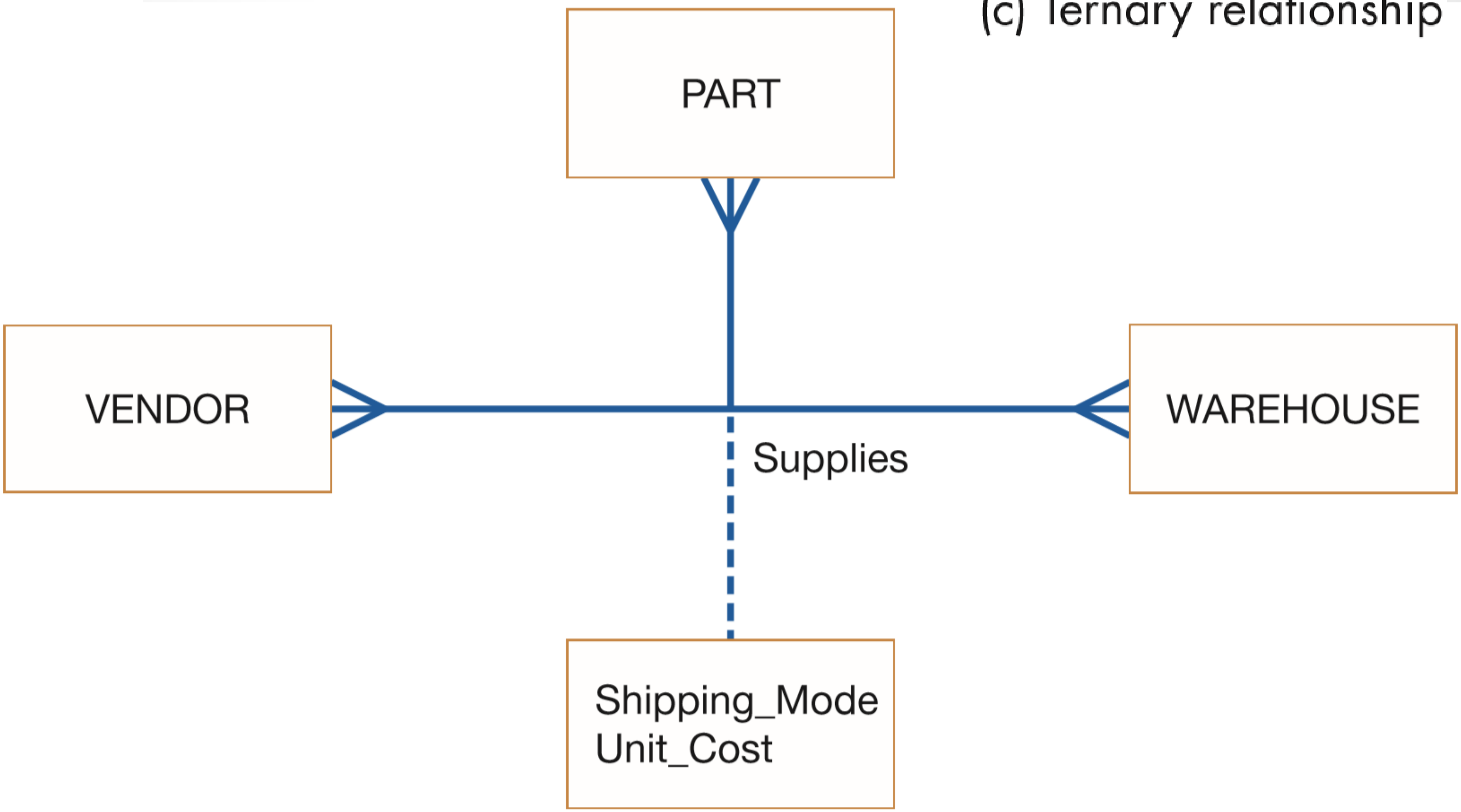
- Ternary relationship:
  - It is *simultaneous* relationship among instances of 3 entities
  - i.e. it occurs when 3 entities share a *common relationship*
- Examine example shown on the next slide:
  - Relationship: Supplies tracks the
    - quantity of a given part,
    - that is shipped by a particular vendor,
    - to a selected warehouse
  - All three entities are *many* participants (in this example)
  - Shipping_Mode
    - attribute of Supplies relationship
    - it's type of shipping carrier used for a particular PART, shipped from particular VENDOR to particular WAREHOUSE

Degree of a Relationship

Ternary Relationships (cont.)

FIGURE 8-11
Examples of relationships of different degrees

(c) Ternary relationship



Degree of a Relationship

N-ary Relationships

- Occurs when > 3 entities share a relationship
- This situation rarely occurs and can be ignored for the purposes of this course

Cardinalities in Relationships

Cardinalities in Relationships

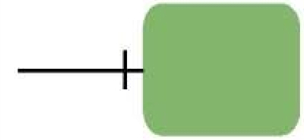
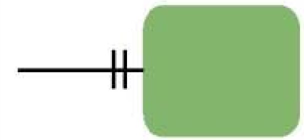
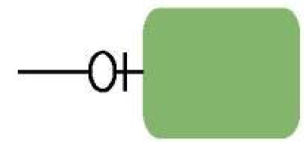
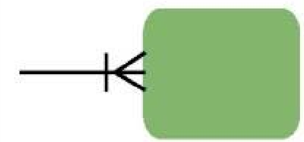
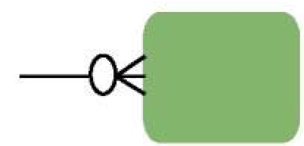
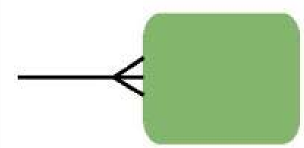
Cardinality

- This is the number of entity occurrences associated with 1 occurrence of the related entity
- Represented by Questions 5, 7, and 8 in Table 8-1
- Cardinality is indicated at the ends of the relationship arc by either
  - Symbols (crow's foot notation), or
  - Letters and numbers (Chen's notation)
- Example:
  - 2 entity types, A and B, are connected by a relationship
  - $\Rightarrow$ cardinality is number of instances of entity B that can (or must) be associated with each instance of entity A

Cardinalities in Relationships

Minimum and Maximum Cardinalities

- Consider relationship for [DVDs at a video store]:
  - Since video store may stock > 1 DVD of a given movie, it is clear that this is (basically) a "many" relationship (Fig. 8-13a)
- We use min. & max. cardinalities to more precisely indicate range of cardinalities for a relationship
  - This notation is shown below
  - A more detailed version is shown on the following slide

Cardinality Interpretation	Minimum Instances	Maximum Instances	Graphic Notation
Exactly one (one and only one)	1	1	 – or – 
Zero or one	0	1	
One or more	1	many (>1)	
Zero, one, or more	0	many (>1)	
More than one	>1	>1	

Cardinalities in Relationships

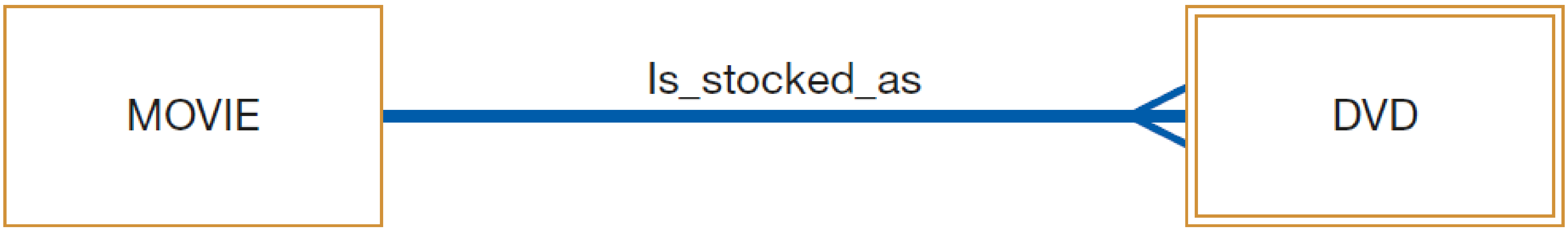
Minimum and Maximum Cardinalities (cont.)

- Minimum Cardinality:
  - This's the *minimum number* of instances of entity B that may be associated with each instance of entity A
  - In our e.g. min. # of DVDs available for a movie is 0
    ⇒ DVD is **optional participant** in the Is_stocked_as relationship
  - If minimum cardinality of a relationship = 1
    ⇒ entity B is a **mandatory participant** in the relationship

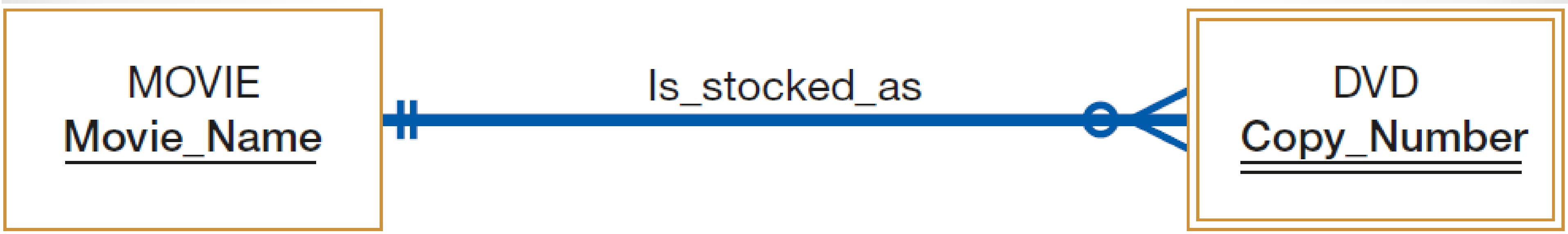
Cardinalities in Relationships

FIGURE 8-13
Introducing cardinality constraints

(a) Basic relationship



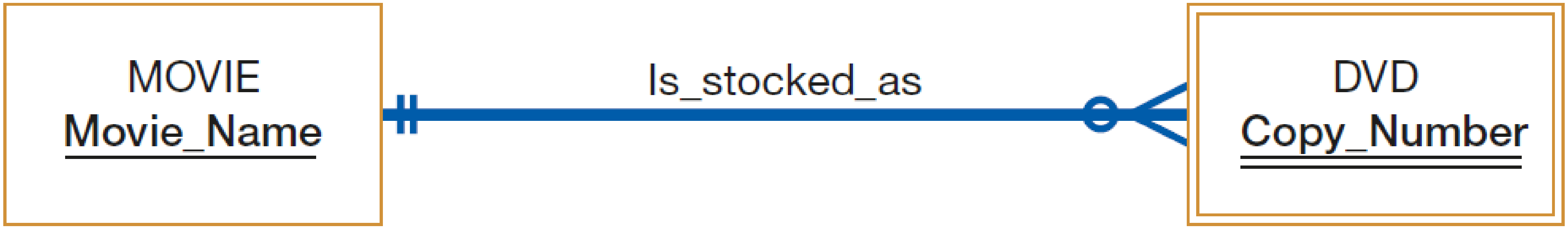
(b) Relationship with cardinality constraints



Cardinalities in Relationships

Minimum and Maximum Cardinalities (cont.)

- Maximum Cardinality:
  - This's is the *maximum number* of instances of entity B that may be associated with each instance of entity A
  - In our e.g. maximum is "many" (an unspecified number > 1)
  - 0 thru line near DVD entity means min. cardinality of zero
  - crow's foot notation means a "many" maximum cardinality
  - double underline of Copy_Number:
    - indicates that this attribute is part of the identifier of DVD
    - note, full composite identifier must also include the identifier of MOVIE, Movie_Name



Cardinalities in Relationships

Minimum and Maximum Cardinalities (cont.)

- Following are 3 relationships that show *all possible combinations* of min. & max. cardinalities

1. PATIENT Has_recorded PATIENT_HISTORY
   - o Each patient has recorded *one or more patient* histories (note, 1st patient visit is recorded as PATIENT HISTORY instance)
   - o Each instance of PATIENT HISTORY is a record for *exactly one* PATIENT

FIGURE 8-14
Examples of cardinality constraints
(a) Mandatory cardinalities

•

Cardinalities in Relationships

Minimum and Maximum Cardinalities (cont.)

2. EMPLOYEE Is_assigned_to PROJECT
   - Each PROJECT has *at least one* assigned EMPLOYEE
   - Each EMPLOYEE *may or may not* be assigned to any existing PROJECT, or may be assigned to several PROJECTs

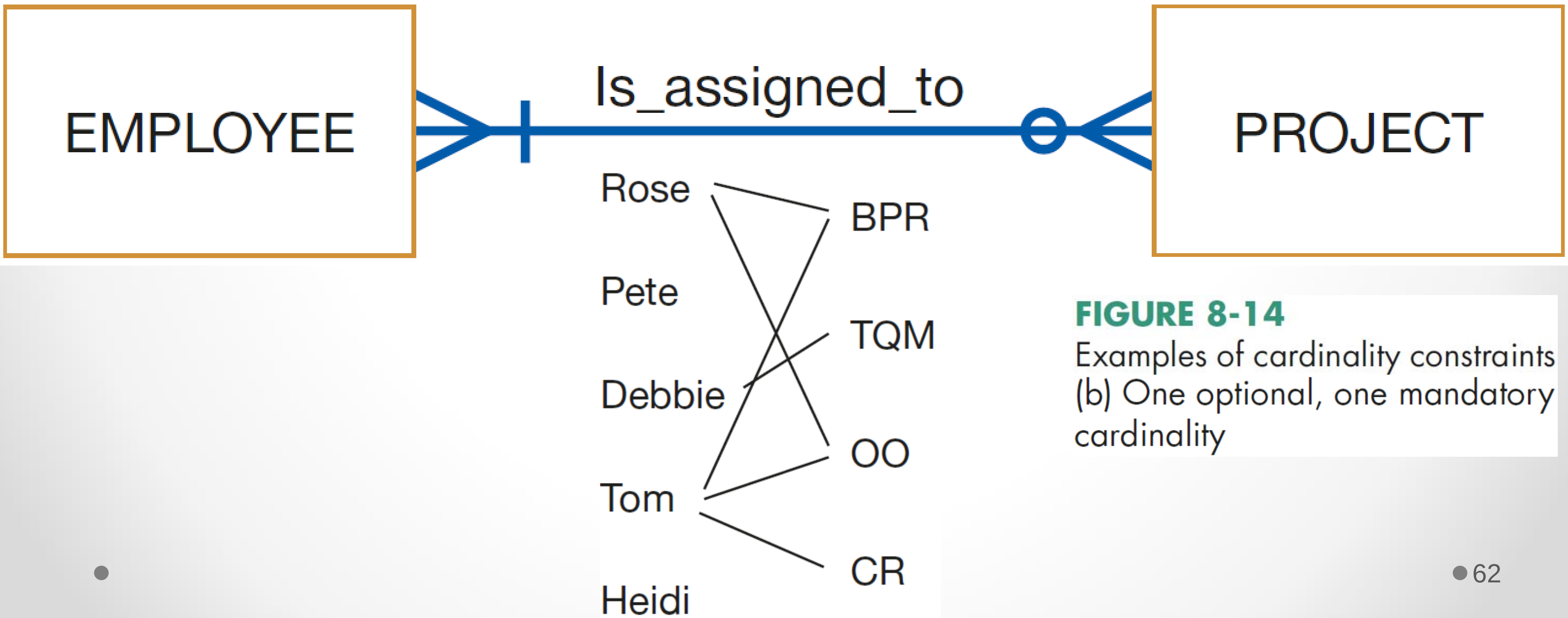


FIGURE 8-14
Examples of cardinality constraints
(b) One optional, one mandatory cardinality

Cardinalities in Relationships

Minimum and Maximum Cardinalities (cont.)

3. PERSON Is_married_to PERSON
   - This is an optional *zero or one cardinality* (in both directions)
   - i.e. person may or may not be married

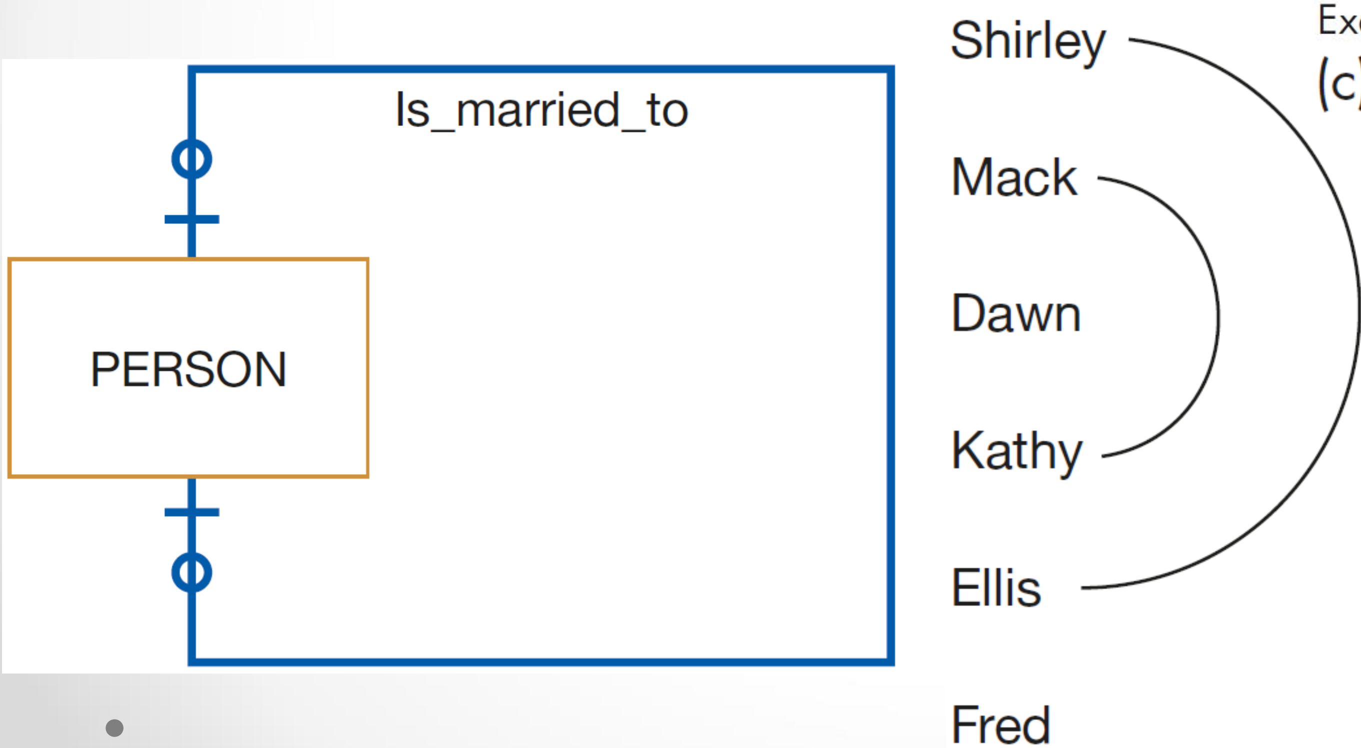
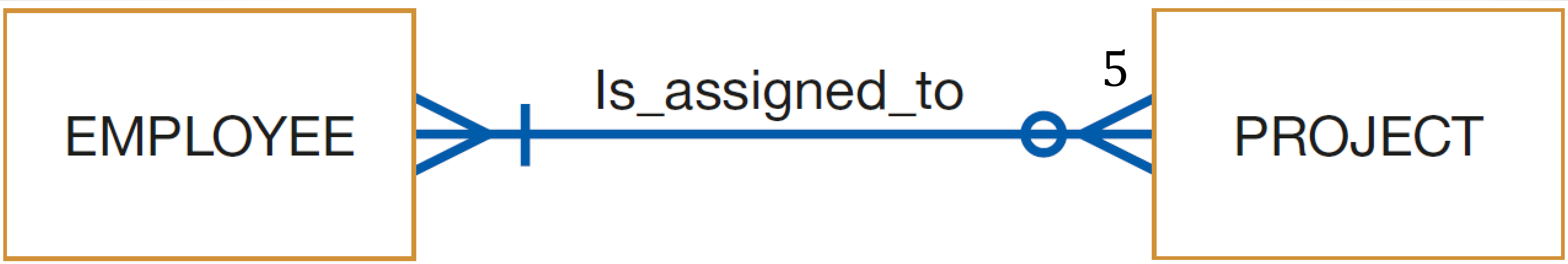


FIGURE 8-14
Examples of cardinality constraints
(c) Optional cardinalities

Cardinalities in Relationships

Minimum and Maximum Cardinalities (cont.)

- It is possible for the maximum cardinality to be a fixed number, not an arbitrary "many" value
  - o Cardinality limits are determined according to the way in which the business is operated (**business rules** of enterprise)
- e.g. suppose corporate policy states that employee may work on *at most* 5 projects at the same time
  - o We could show this business rule by placing a "5" above (or below) crow's foot next to PROJECT entity



Cardinalities in Relationships

Alternative Cardinality System (cont.)

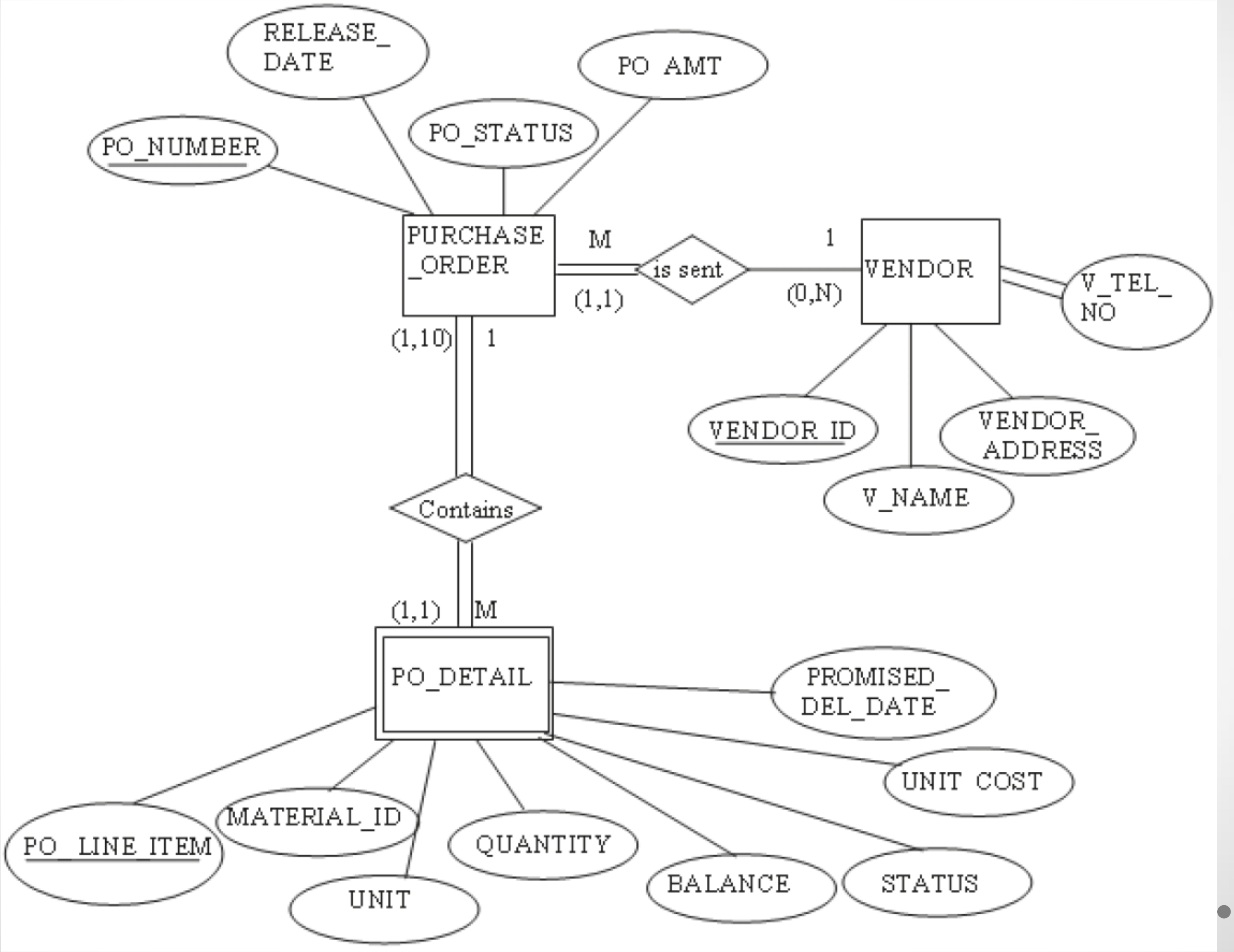
- Uses *letters* and *numbers* instead of *Crow's foot notation* (sometimes called *Chen's notation*)
- Cardinality (type of relationship) is expressed in the following way:
  - 1 : 1 (one-to-one)
  - 1 : M (one-to-many)
  - M : N (many-to-many)
- Cardinality limits are shown in parentheses (min. cardinality, max cardinality); examples:
  - (1, 1)
  - (1, 7)
  - (0, N)

Cardinalities in Relationships

Alternative Cardinality System (cont.)

- Examine e.g. of this shown on the next slide
  - o Cardinality 1 : M means:
    - each instance of an entity is associated with 0, 1, or many instances of the entity on the M side of the relationship
  - o Relationship between PURCHASE_ORDER and PO_DETAIL:
    - cardinality limits for PO_DETAIL are (1,1)
    - ⇒ each PO_DETAIL is associated with one purchase order
  - o Similarly, the cardinality limits of PURCHASE_ORDER is (1,10):
    - ⇒ a minimum of 1 and a maximum of 10 line items are associated with one purchase order
  - o Also, relationship between PURCHASE_ORDER and VENDOR:
    - cardinality limits for VENDOR are (0,N)
    - ⇒ each vendor is sent 0, 1 or many purchase orders

Cardinalities in Relationships



Cardinalities in Relationships

Alternative Cardinality System (cont.)

- Let's see if you can determine for the e.g. below:
  - o Type and degree of relationship
  - o Cardinality limits for each entity
  - o Relationship constraints

Cardinalities in Relationships

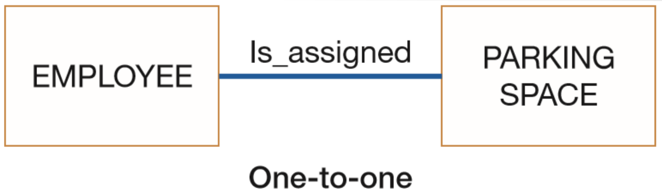
Semantic Net Diagram

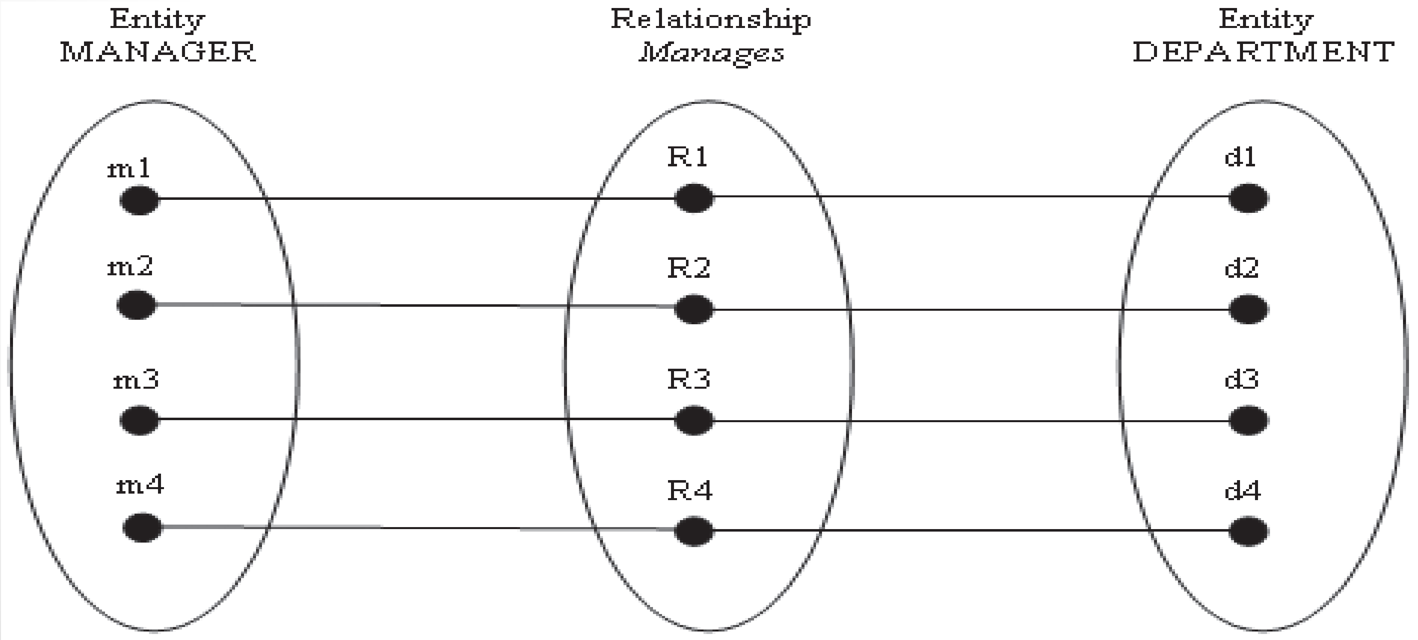
- M : N relationship is difficult to handle in a database query
  - o ⇒ it is usu. expanded into a series of 1: M relationships in the final database tables
- Semantic net diagram:
  - o Useful graphical tool that assists in visualizing the cardinality of a relationship
  - o Can be used to represent (see upcoming slides):
    1. 1 : 1 relationship
    2. 1 : M relationship
    3. M : N relationship
  - o Note how in M : N relationship, # of instances of relationship (6) is > # of related instances in an entity set (2, 4)

Cardinalities in Relationships

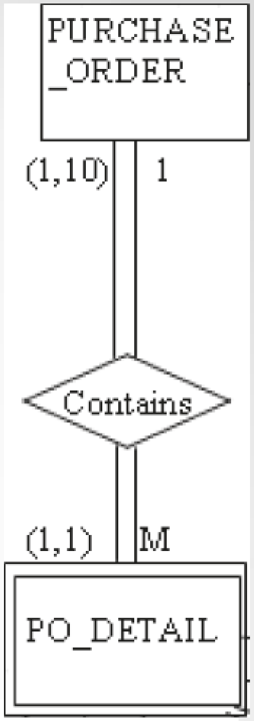
Semantic Net Diagram (cont.)

1. 1 : 1 semantic net diagram for relationship between MANAGER and DEPARTMENT



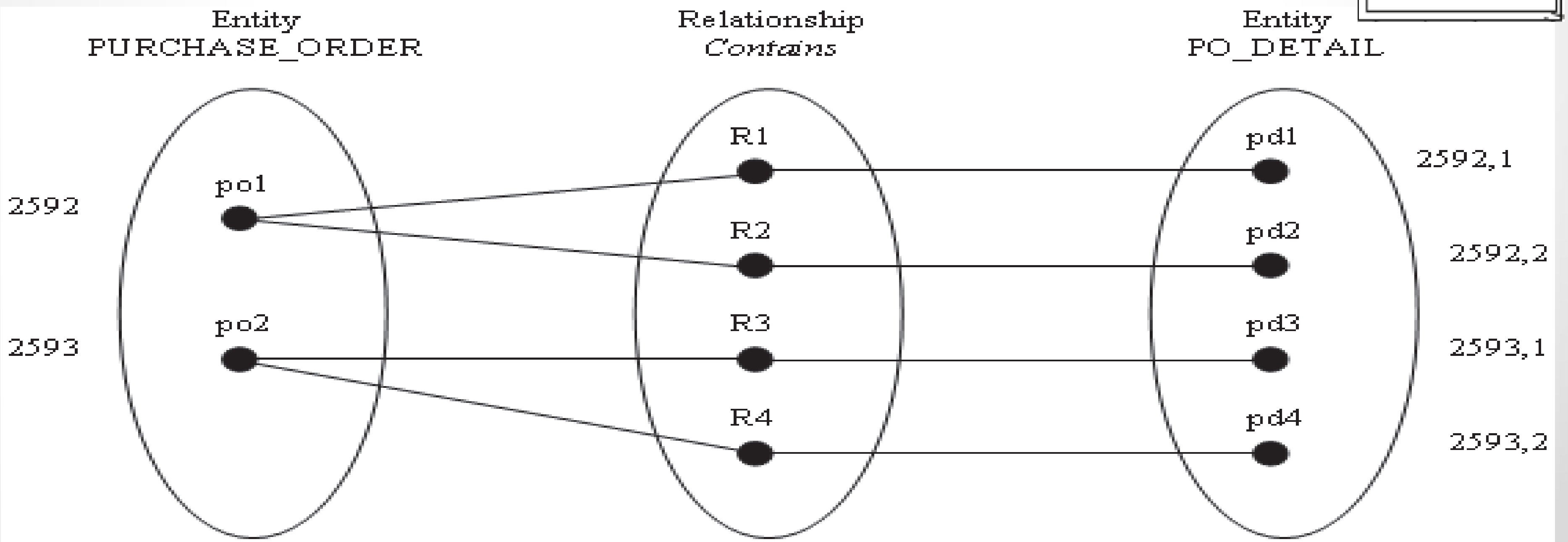


Cardinalities in Relationships



Semantic Net Diagram (cont.)

2. 1 : M semantic net diagram for relationship between PURCHASE_ORDER and PO_DETAIL

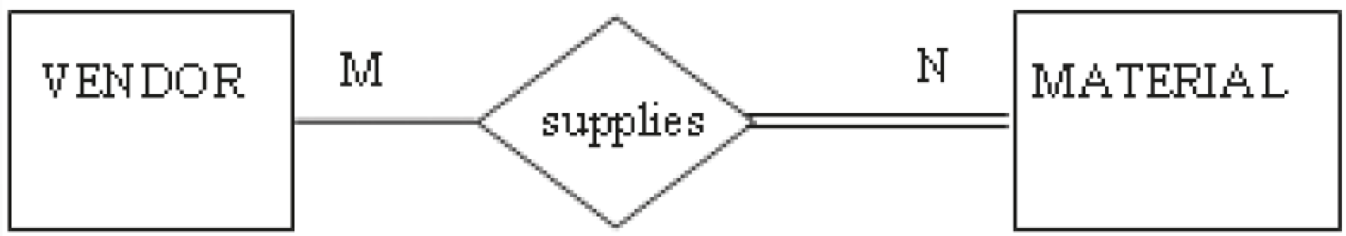


Cardinalities in Relationships

Semantic Net Diagram (cont.)

3. M : N semantic net diagram for relationship between VENDOR and MATERIAL

- Business rules:
  - o A vendor supplies material to a company
  - o 1 vendor may supply > 1 material
  - o Also, specific material may be supplied by > 1 vendor



Cardinalities in Relationships

Semantic Net Diagram (cont.)

- M : N semantic net diagram for relationship between VENDOR and MATERIAL (cont.):

Entity Set: VENDOR

VENDOR ID	V NAME	V STREET	V CITY	V STATE	V ZIP
V110	Jersey	2 Main St.	Patterson	NJ	07055
V25	General	125 Common	Boise	ID	44830
V250	Spices	25 Salty Lane	East Hampton	NY	10027
V75	Pasta Supply,	34 Henry St.	Philadelphia	PA	09098

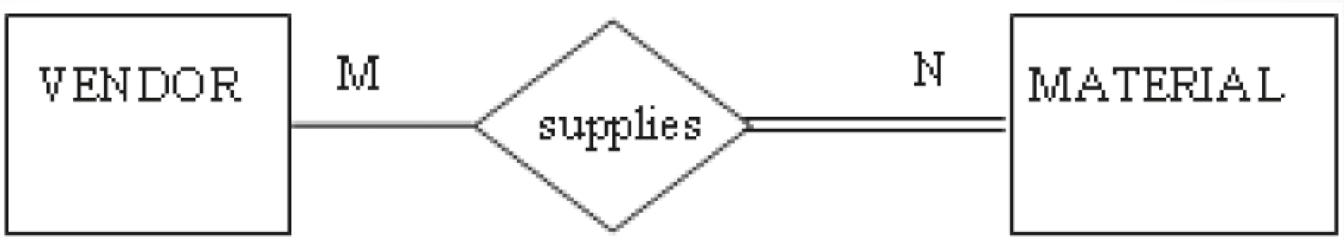
Entity Set: MATERIAL

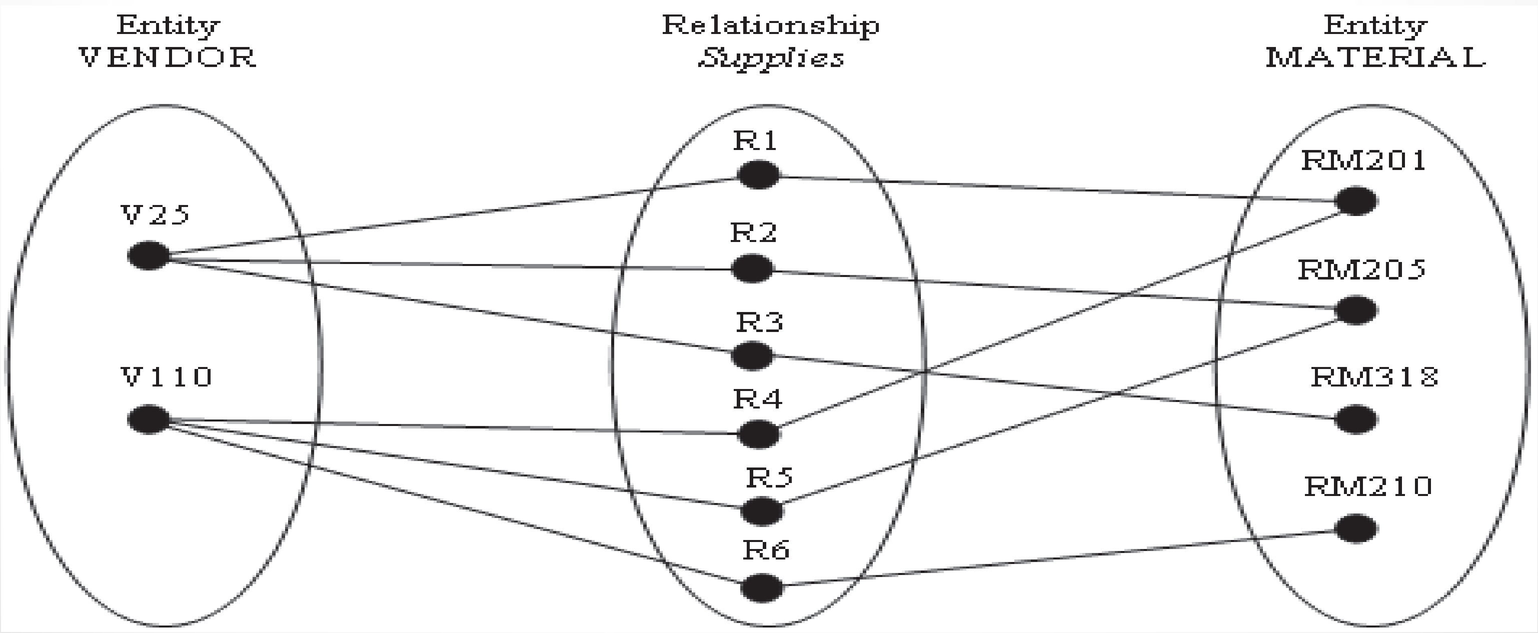
MATERIAL ID	MATL DESCRIPTION
RM201	Carrots, whole
RM202	Carrots, diced, 1/4 inch
RM205	Potatoes, Eastern,
RM210	Peas, shelled
RM211	Tomatoes, whole
RM310	Garlic, whole
RM311	Garlic powder
RM318	Salt, iodized
RM308	Onion salt
RM305	Paprika
RM340	Sugar, bulk
RM805	Olive oil
RM810	Vinegar, white

Cardinalities in Relationships

Semantic Net Diagram (cont.)

- M : N semantic net diagram for relationship between VENDOR and MATERIAL (cont.):





Naming Relationships

Naming Relationships

Few special guidelines for naming relationships:

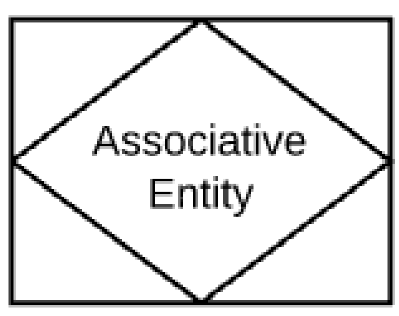
- Relationship name is a verb phrase
  - e.g. Assigned_to, Supplies, or Teaches
- Relationships represent actions, usually in the present tense
- Relationship name states the action taken, not the result of the action
  - e.g. use Assigned_to, not Assignment
- Avoid vague names
  - e.g. such as Has or Is_related_to
- Use descriptive verb phrases (action verbs)

Associative Entities

Associative Entities

- Also called composite/bridge entity
- Entity type that relates/associates instances of 1 or more entity types (e.g. Has_components, Supplies)
- Transforms M : N relationship into 1 : M relationships
- Contains attributes specific to the relationship between those entity instances

- Two different notations below:



Associative Entities

- Example 1: organization wishes to record date (month/year) that employee completes each course

Employee_ID	Course_Name	Date_Completed
549-23-1948	Basic Algebra	March 2017
629-16-8407	Software Quality	June 2017
816-30-0458	Software Quality	February 2017
549-23-1948	C Programming	May 2017

- Date_Completed is *not* a property of entity EMPLOYEE
  - e.g. 549-23-1948 completed courses on different dates
- Also, Date_Completed is *not* a property of COURSE
  - e.g. Software Quality was completed on different dates
- ⇒ Date_Completed is a property of the *relationship* between EMPLOYEE and COURSE (*next slide*)

Associative Entities

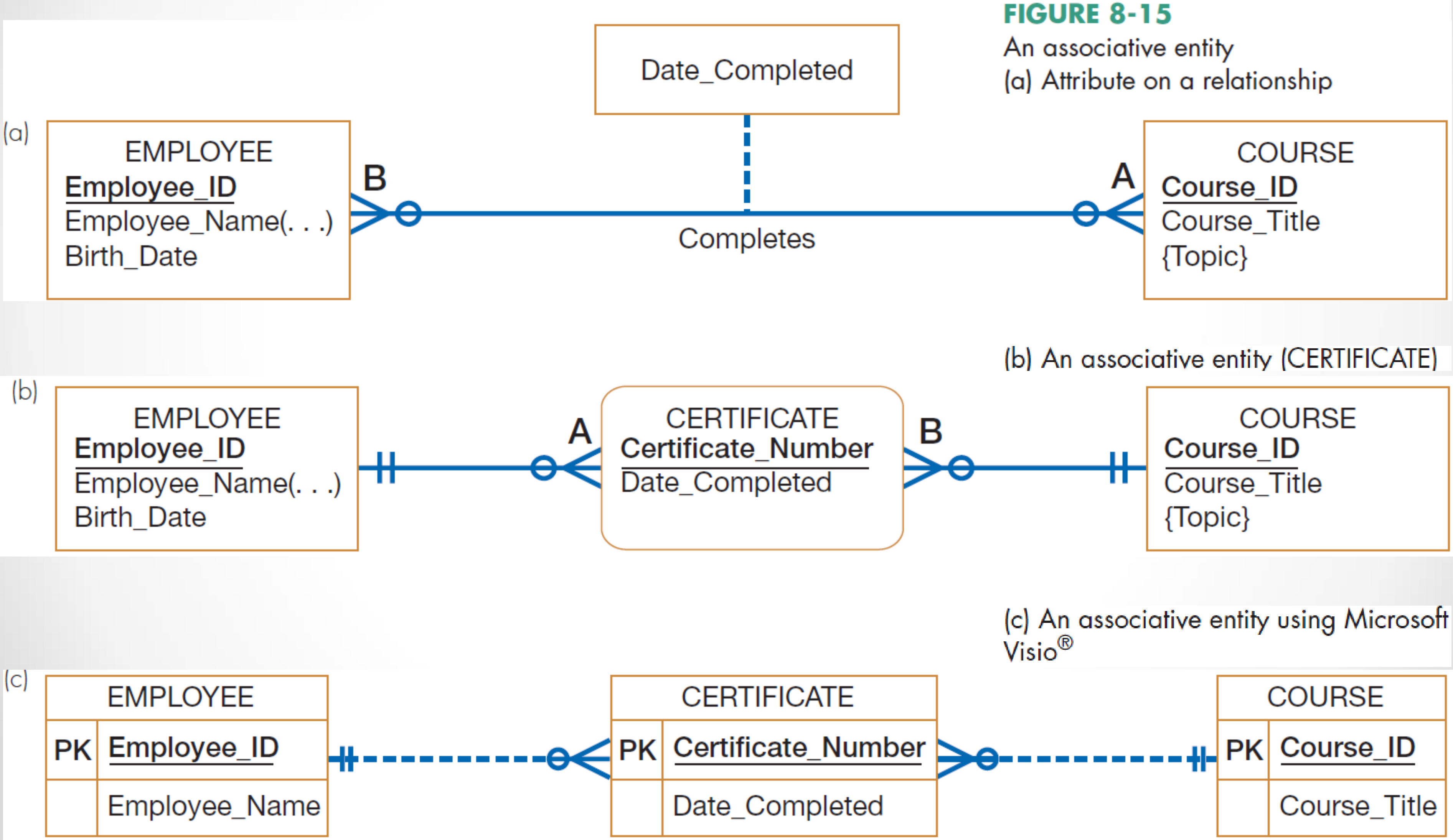


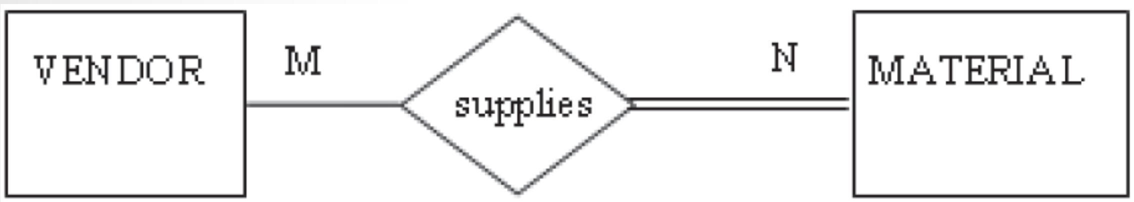
FIGURE 8-15
An associative entity
(a) Attribute on a relationship

(b) An associative entity (CERTIFICATE)

(c) An associative entity using Microsoft Visio®

Associative Entities

- Example 2: Convert M : N relationship between VENDOR and MATERIAL into associative entity



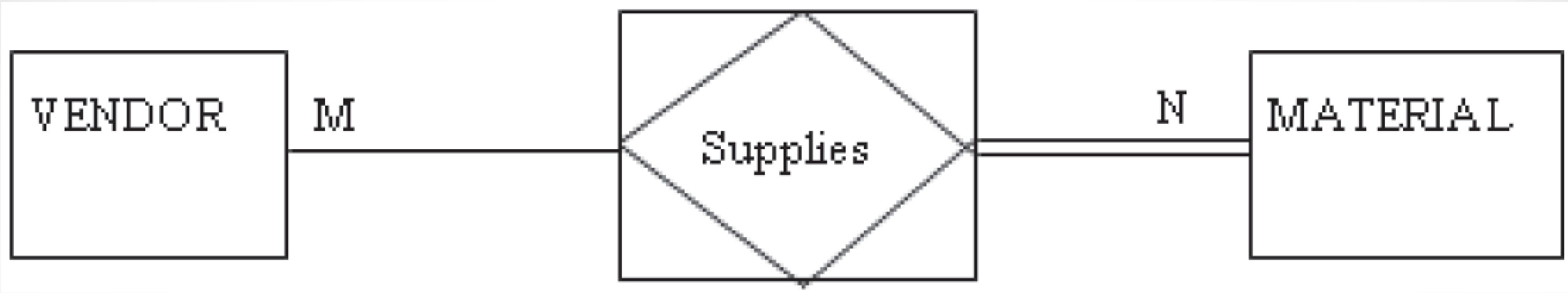
Entity Set: MATERIAL

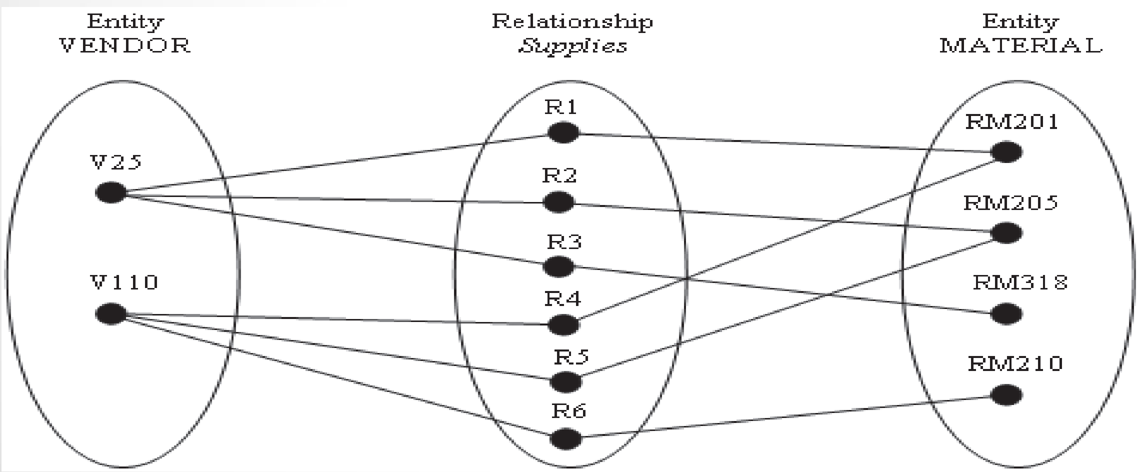
MATERIAL ID	MATL DESCRIPTION
RM201	Carrots, whole
RM202	Carrots, diced, 1/4 inch
RM205	Potatoes, Eastern,
RM210	Peas, shelled
RM211	Tomatoes, whole
RM310	Garlic, whole
RM311	Garlic powder
RM318	Salt, iodized
RM308	Onion salt
RM305	Paprika
RM340	Sugar, bulk
RM805	Olive oil
RM810	Vinegar, white

Entity Set: VENDOR

VENDOR ID	V NAME	V STREET	V CITY	V STATE	V ZIP
V110	Jersey	2 Main St.	Patterson	NJ	07055
V25	General	125 Common	Boise	ID	44830
V250	Spices	25 Salty Lane	East Hampton	NY	10027
V75	Pasta Supply,	34 Henry St.	Philadelphia	PA	09098

Associative Entities





Entity Set: VENDOR_MATL_XREF

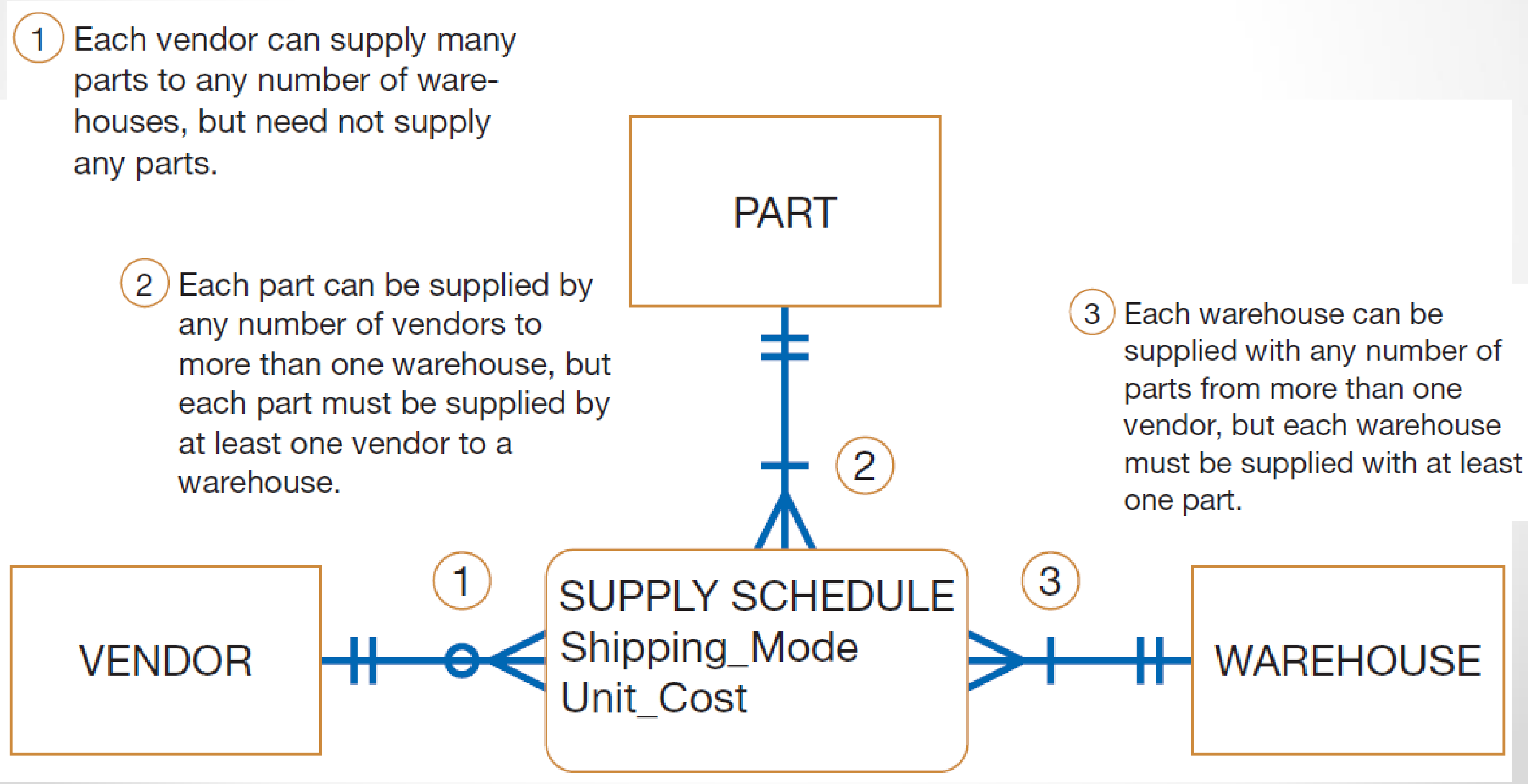
VENDOR ID	MATERIAL ID
V25	RM201
V25	RM205
V25	RM318
V25	RM340
V110	RM201
V110	RM202
V110	RM205
V110	RM210
V110	RM211
V250	RM310
V250	RM311
V250	RM318
V250	RM340
V250	RM308
V250	RM305
V25	RM805
V25	RM810

Associative Entities

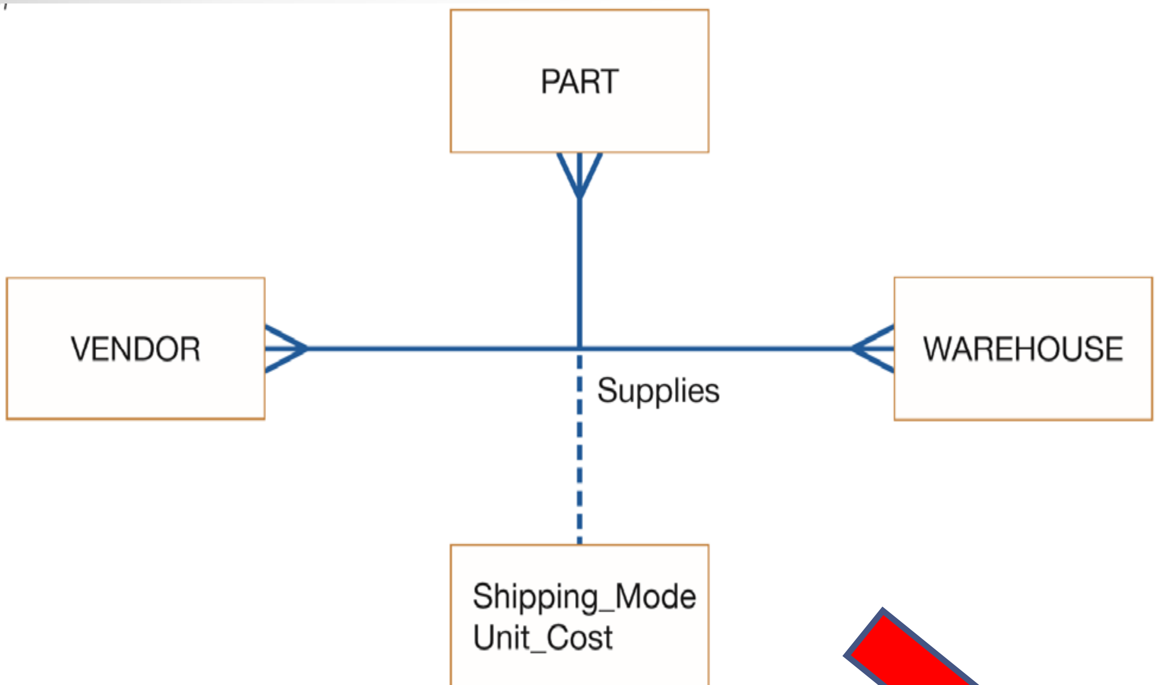
- Entity VENDOR_MATL_XREF is a bridge between the entities VENDOR and MATERIAL
  - i.e. associative/composite entity is used here to convert semantic net diagram for this M:N relationship into an entity
- Note the following:
  - each instance in VENDOR is associated with M instances of VENDOR_MATL_XREF, and
  - each instance of MATERIAL is associated with M instances of VENDOR_MATL_XREF
  - ⇒ M : N relationship is converted to two 1 : M relationships
  - VENDOR_ID/MATERIAL_ID serves as the (composite) primary key of VENDOR_MATL_XREF

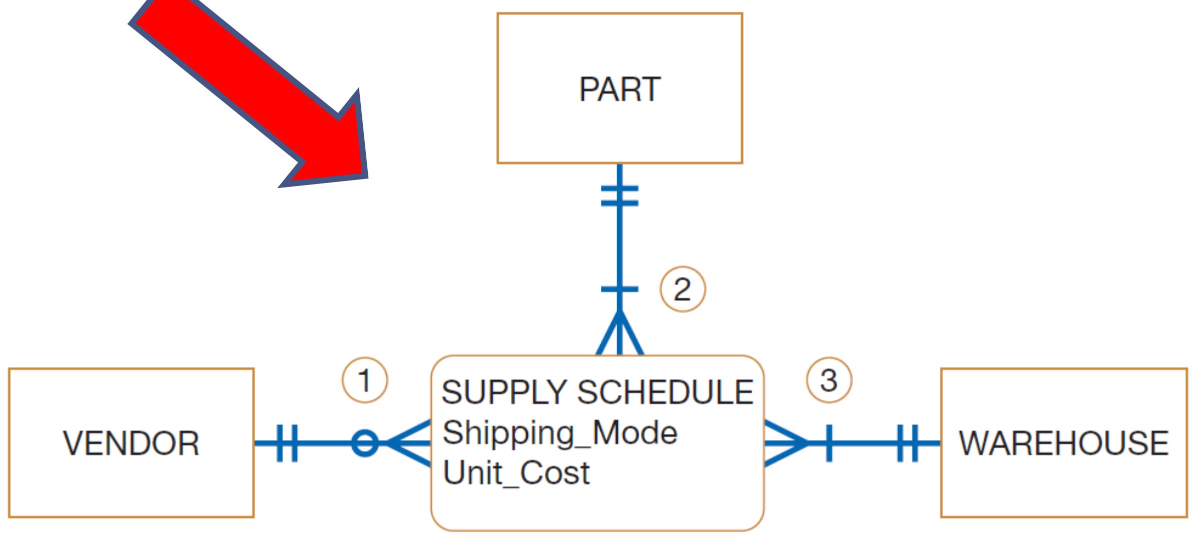
Associative Entities

- Example 3: associative entity for a ternary relationship (alternative and more explicit representation of the ternary Supplies relationship shown in Figure 8-11)



Associative Entities





Videos to Watch

- **Entity Relationship Diagram (ERD) Tutorial - Part 1**
  https://youtu.be/QpdhBUYk7Kk
- **Entity Relationship Diagram (ERD) Tutorial - Part 2**
  https://youtu.be/-CuY5ADwn24
- **Entity-Relationship Diagrams** (another system)
  https://youtu.be/c0_9Y8QAstg
- **Entity Relationship Diagram (ERD) Training Video**
  https://youtu.be/-fQ-bRllhXc

Sources

- “**Chapter 3: Database Modeling and Design**”; Slides by Dr. Sabeur Kosantini (2017)
- “**Types of Database Management Systems**” (2017) by Arjun Panwar, c-sharpcorner.com; Available at: https://www.c/sharpcorner.com/UploadFile/65fc13/types-of-database-management-systems/
- **Modern Systems Analysis and Design**. Joseph S. Valacich and Joey F. George. Pearson. Eighth Ed. 2017. Chapter 8.
- **Design of Industrial Information Systems**. Thomas Boucher, and Ali Yalcin. Academic Press. First Ed. 2006. Chapter 3.

Gathering Info. for Conceptual Data Modeling

TABLE 8-1 Requirements Determination Questions for Data Modeling

1. *What are the subjects/objects of the business?* What types of people, places, things, materials, events, etc. are used or interact in this business, about which data must be maintained? How many instances of each object might exist?—**data entities and their descriptions**
2. *What unique characteristic (or characteristics) distinguishes each object from other objects of the same type?* Might this distinguishing feature change over time or is it permanent? Might this characteristic of an object be missing even though we know the object exists?—**primary key**
3. *What characteristics describe each object?* On what basis are objects referenced, selected, qualified, sorted, and categorized? What must we know about each object in order to run the business?—**attributes and secondary keys**
4. *How do you use these data?* That is, are you the source of the data for the organization, do you refer to the data, do you modify it, and do you destroy it? Who is not permitted to use these data? Who is responsible for establishing legitimate values for these data?—**security controls and understanding who really knows the meaning of data**

Gathering Info. for Conceptual Data Modeling

TABLE 8-1 Requirements Determination Questions for Data Modeling

5. *Over what period of time are you interested in these data?* Do you need historical trends, current "snapshot" values, and/or estimates or projections? If a characteristic of an object changes over time, must you know the obsolete values?—**cardinality and time dimensions of data**
6. *Are all instances of each object the same?* That is, are there special kinds of each object that are described or handled differently by the organization? Are some objects summaries or combinations of more detailed objects?—**supertypes, subtypes, and aggregations**
7. *What events occur that imply associations among various objects?* What natural activities or transactions of the business involve handling data about several objects of the same or a different type?—**relationships and their cardinality and degree**
8. *Is each activity or event always handled the same way or are there special circumstances?* Can an event occur with only some of the associated objects, or must all objects be involved? Can the associations between objects change over time (for example, employees change departments)? Are values for data characteristics limited in any way?—**integrity rules, minimum and maximum cardinality, time dimensions of data**