Data Modeling in organization

Q&A

- Lab
- TA (are they helpful?, can they solve your technical problem, explain the instruction of the perform the lab.)
- Are they willing to help you with any related questions?
- How abou the MS team?
- Feedback`1

Business Rules

- Statements that define or constrain some aspect of the business
- Assert business structure
- Control/influence business behavior
- Expressed in terms familiar to end users
- Automated through DBMS software

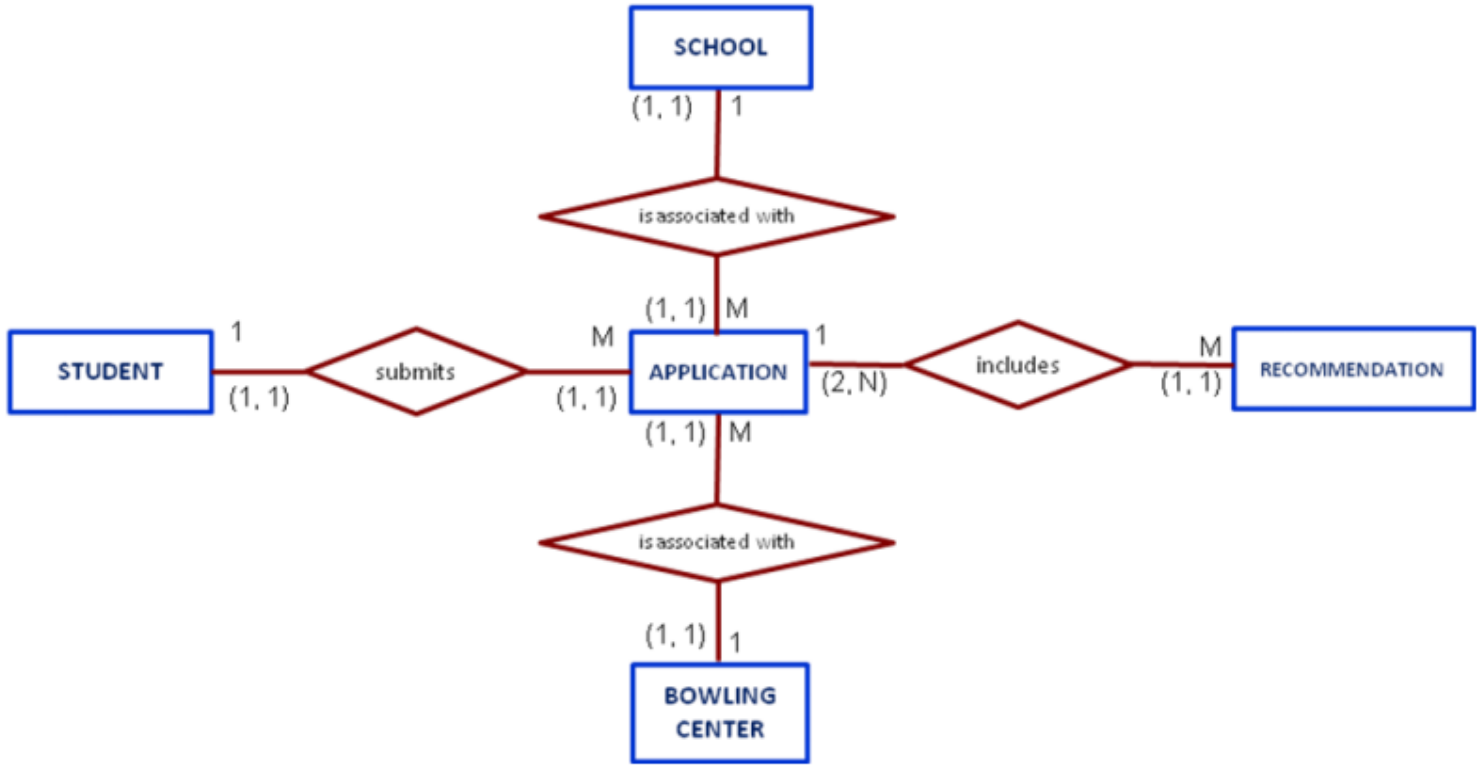
A Good Business Rule is:

- **Declarative**–what, not how
- **Precise**–clear, agreed-upon meaning
- **Atomic**–one statement
- **Consistent**–internally and externally
- **Expressible**–structured, natural language
- **Distinct**–non-redundant
- **Business-oriented–**understood by business people

Example of business rule in Bowling club

- Each applicant can submit one or more applications (one application per year for multiple years).
- Each application is submitted by only one applicant.
- Each school can be associated with one or more applications.
- Each application is associated with only one school.
- Each application must include two or more recommendations.
- Each recommendation is included with only one application.
- Each bowling center can be associated with one or more applications.
- Each application is associated with only one bowling center.

Entity Relation Diagram (ERD)



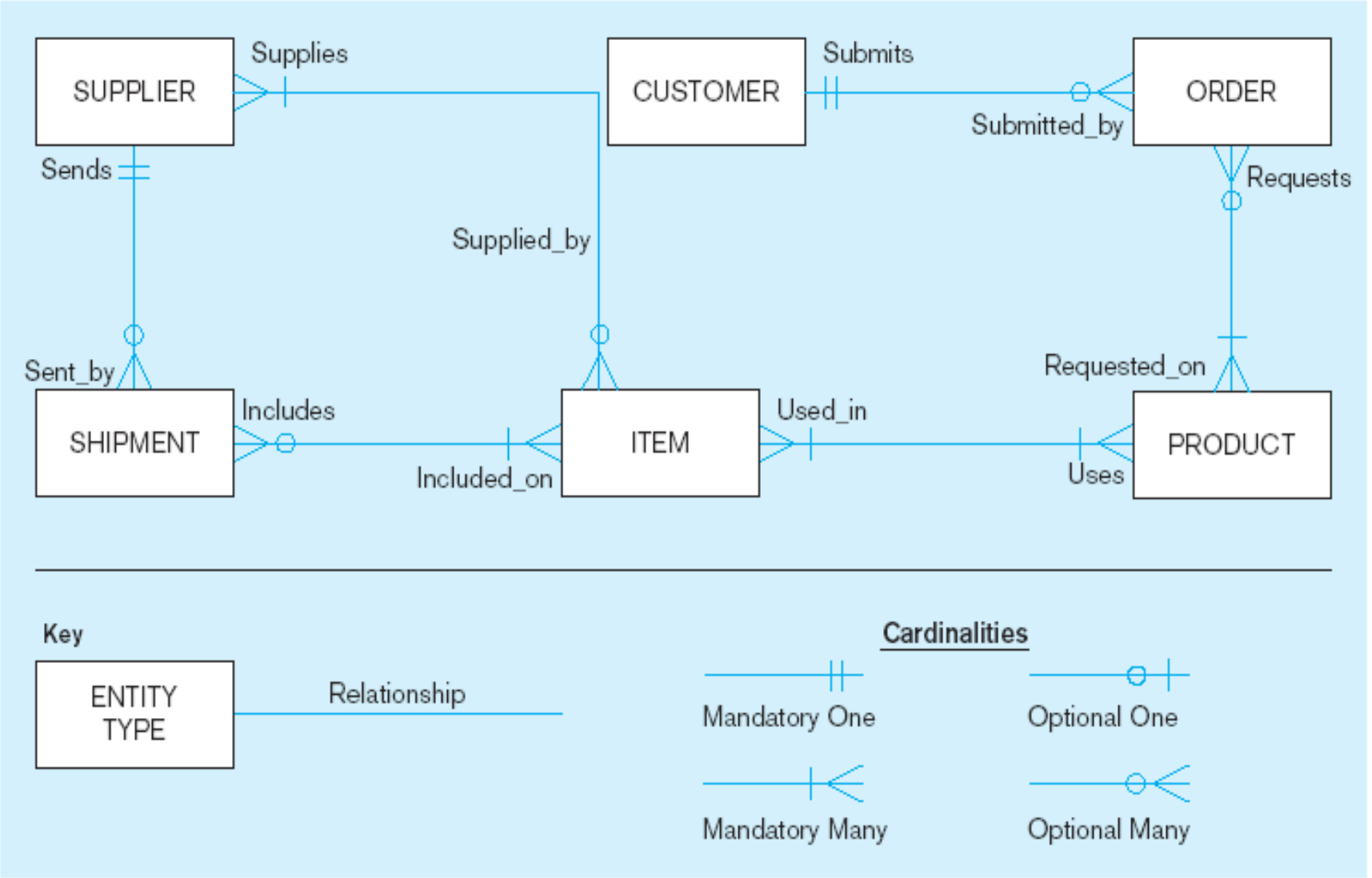
E-R Model Constructs

- **Entities:**
  - Entity instance–person, place, object, event, concept (often corresponds to a row in a table)
  - Entity Type–collection of entities (often corresponds to a table)

- **Relationships:**
  - Relationship instance–link between entities (corresponds to primary key-foreign key equivalencies in related tables)
  - Relationship type–category of relationship…link between entity types

- **Attribute**–property or characteristic of an entity or relationship type (often corresponds to a field in a table)

Sample E-R Diagram

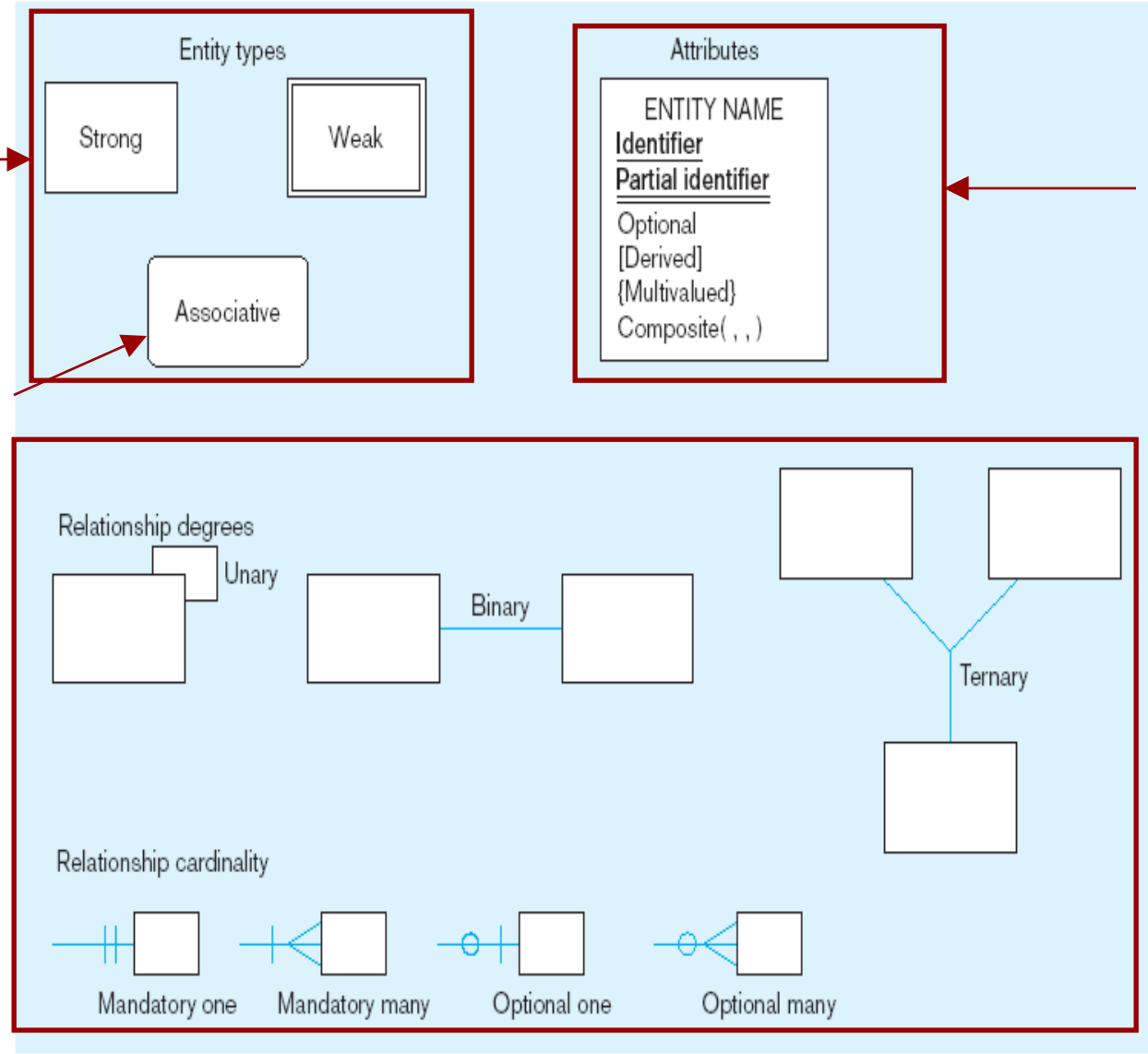


Basic E-R notation

Entity symbols

A special entity that is also a relationship

Relationship degrees specify number of entity types involved



Attribute symbols

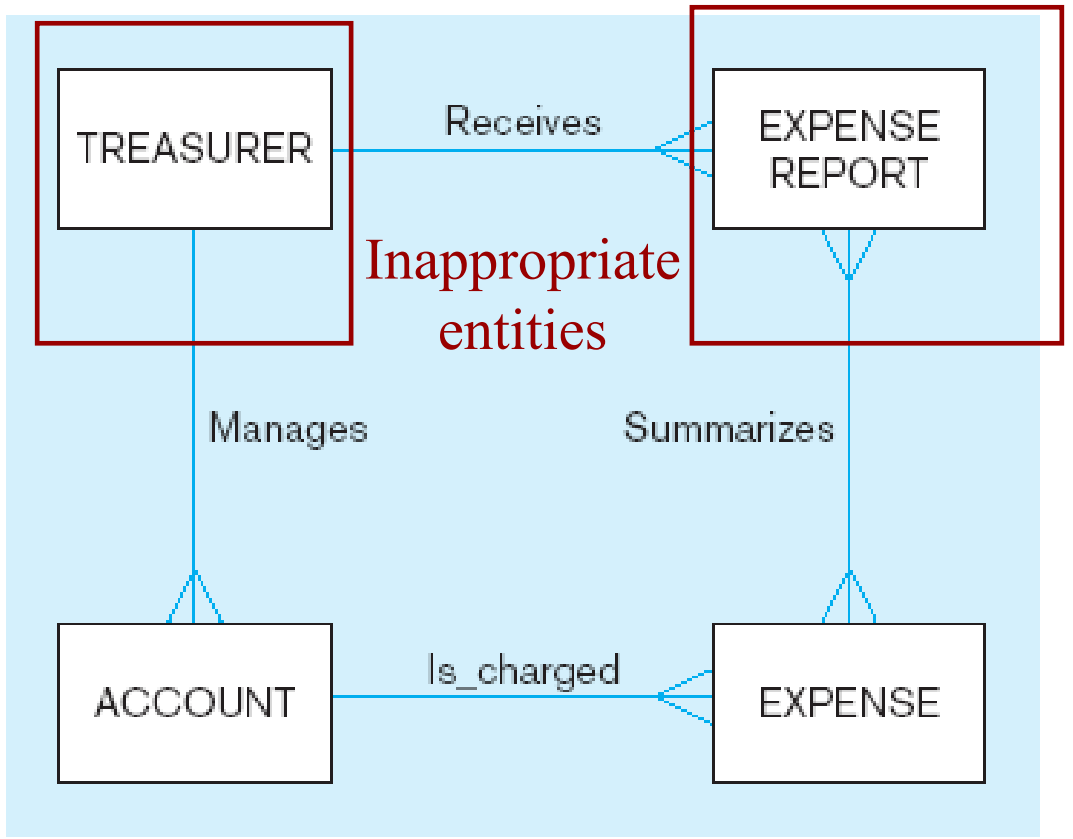
Relationship symbols

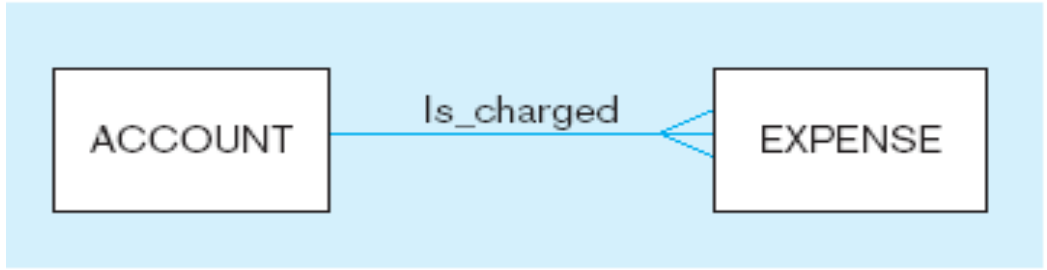
Relationship cardinalities specify how many of each entity type is allowed

What Should an Entity Be?

- **SHOULD BE:**
  - An object that will have many instances in the database
  - An object that will be composed of multiple attributes
  - An object that we are trying to model
- **SHOULD NOT BE:**
  - A user of the database system
  - An output of the database system (e.g., a report)

Example of inappropriate entities





Appropriate entities

Attributes

- Attribute–property or characteristic of an entity or relationship type
- Classifications of attributes:
  - Required vs. Optional Attributes
  - Simple vs. composite attribute
  - Single-Valued vs. Multivalued Attribute
  - Stored versus Derived Attributes
  - Identifier Attributes (PK, FK)

Identifiers (Keys)

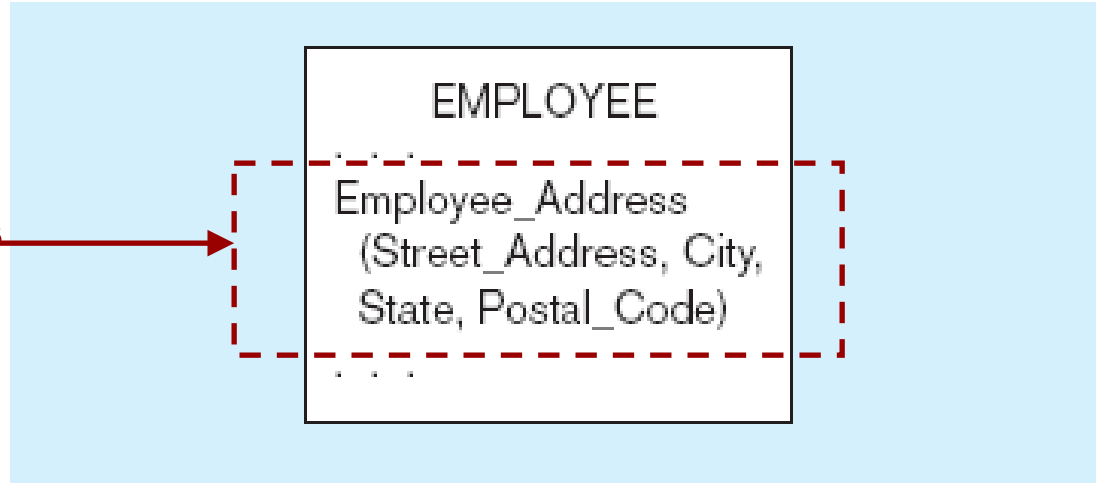
- Identifier (Key)–An attribute (or combination of attributes) that **uniquely** identifies individual instances of an entity type
- Simple versus Composite Identifier
- Candidate Identifier–an attribute that could be a key…satisfies the requirements for being an identifier

Characteristics of Identifiers

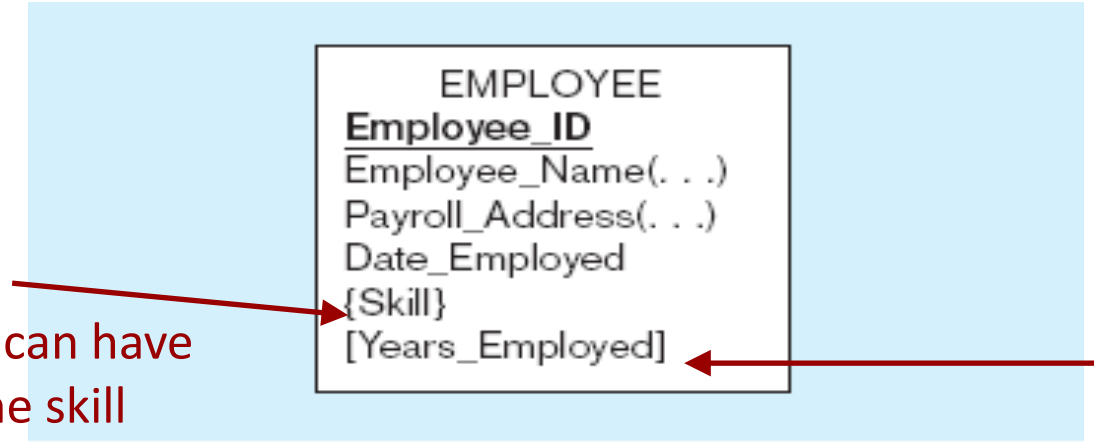
- **Will not** change in value
- **Will not** be null (can't be empty)
- No intelligent identifiers (e.g., containing locations or people that might change)
- Substitute new, simple keys for long, composite keys

A **composite** attribute

An attribute broken into component parts



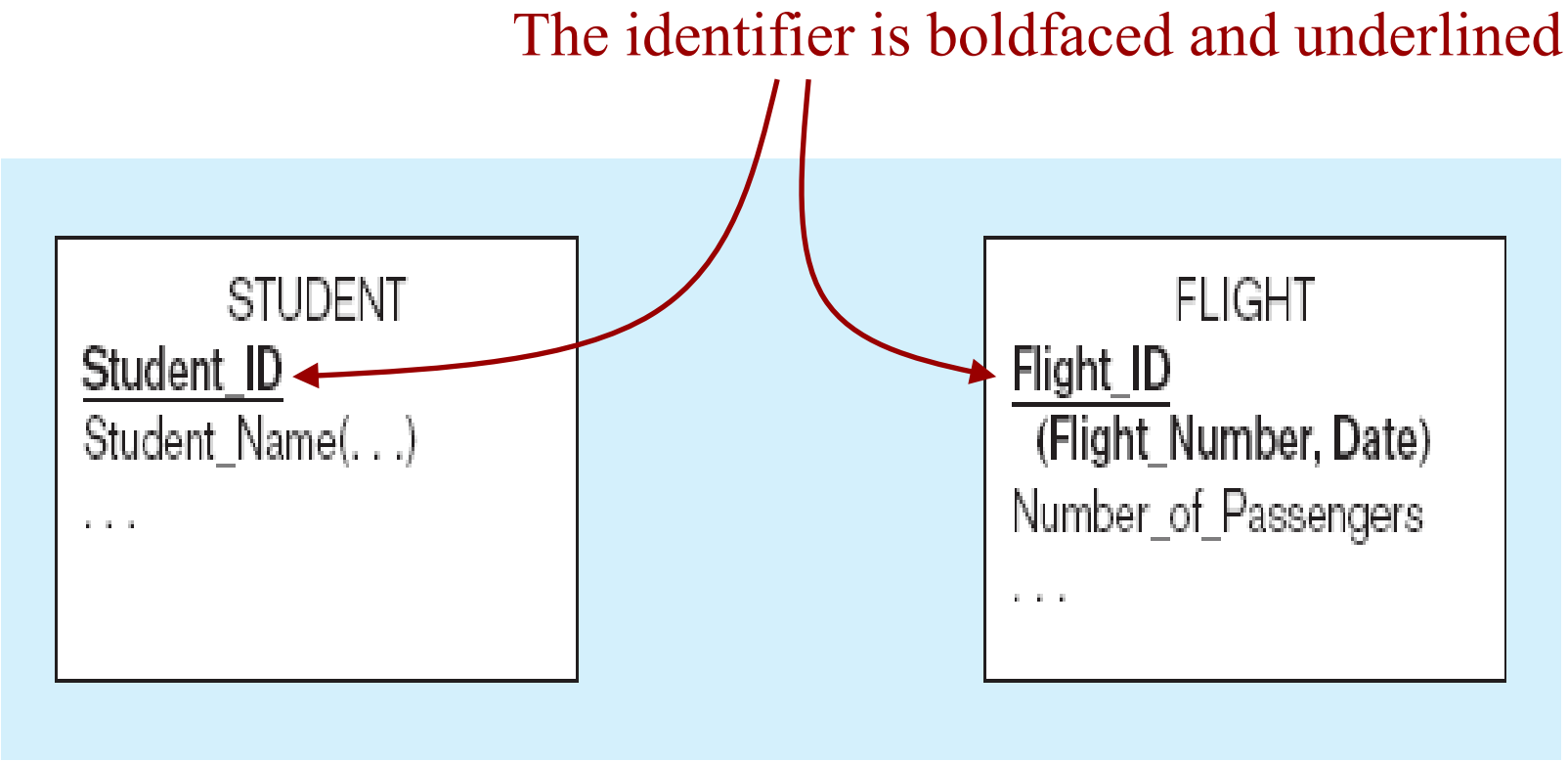
Entity with **multivalued** attribute (Skill)
and **derived** attribute (Years_Employed)



Multivalued
an employee can have more than one skill

Derived
from date employed and current date

Simple and composite identifier attributes



(a) Simple identifier attribute

(b) Composite identifier attribute

Simple example of time-stamping

This attribute that is both multivalued *and* composite

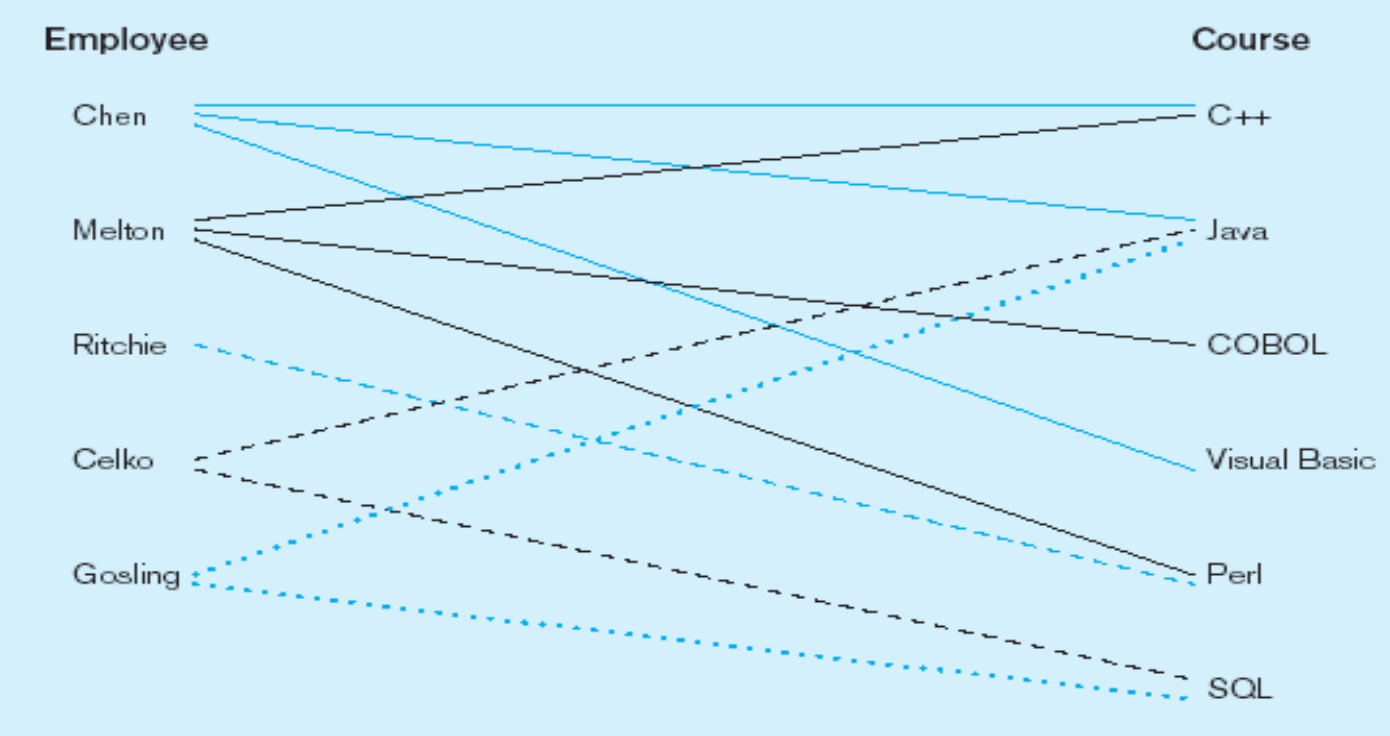
More on Relationships

- **Relationship Types vs. Relationship Instances**
  - The relationship type is modeled as lines between entity types…the instance is between specific entity instances
- Relationships can have attributes
  - These describe features pertaining to the association between the entities in the relationship
- Two entities can have more than one type of relationship between them (multiple relationships)
- Associative Entity–combination of relationship and entity

Relationship types and instances

a) Relationship type

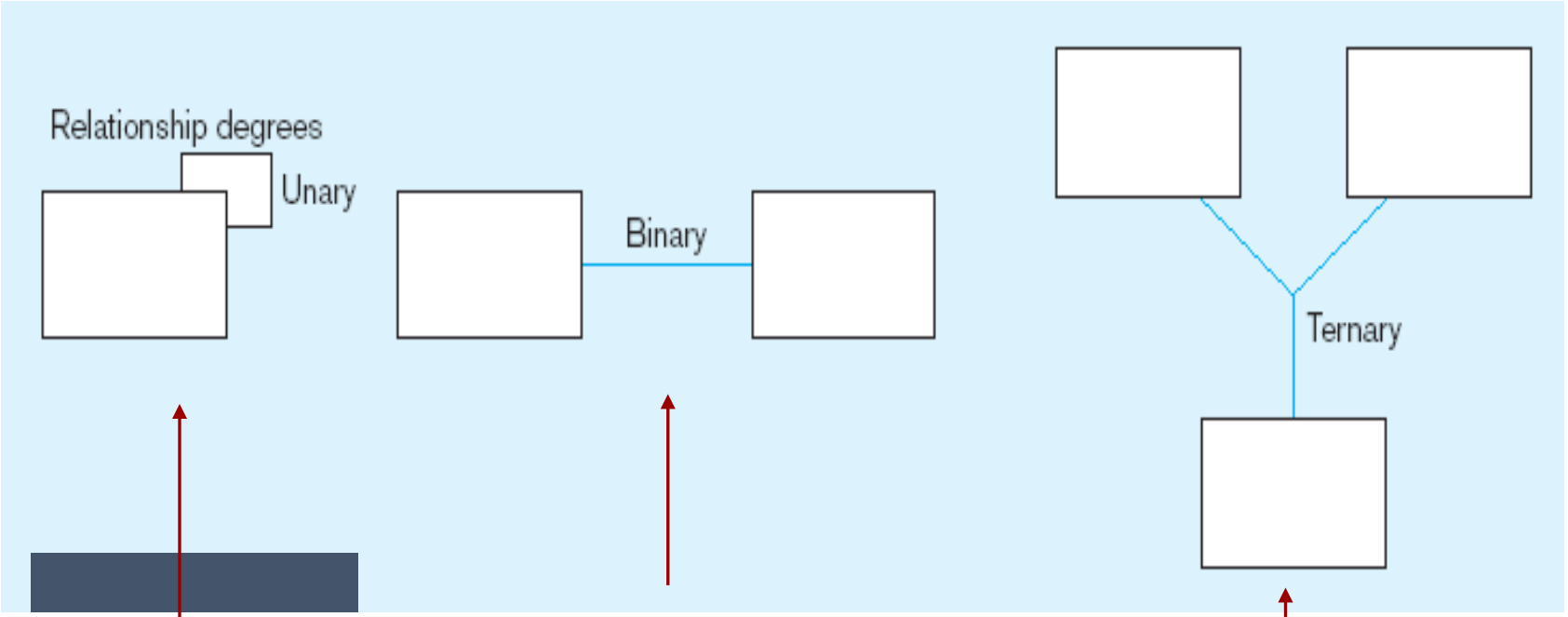
b) Relationship instances



Degree of Relationships

- Degree of a relationship is the number of entity types that participate in it
  - Unary Relationship
  - Binary Relationship
  - Ternary Relationship

Degree of relationships



One entity related to another of the same entity type

Entities of two different types related to each other

Entities of three different types related to each other

Cardinality of Relationships

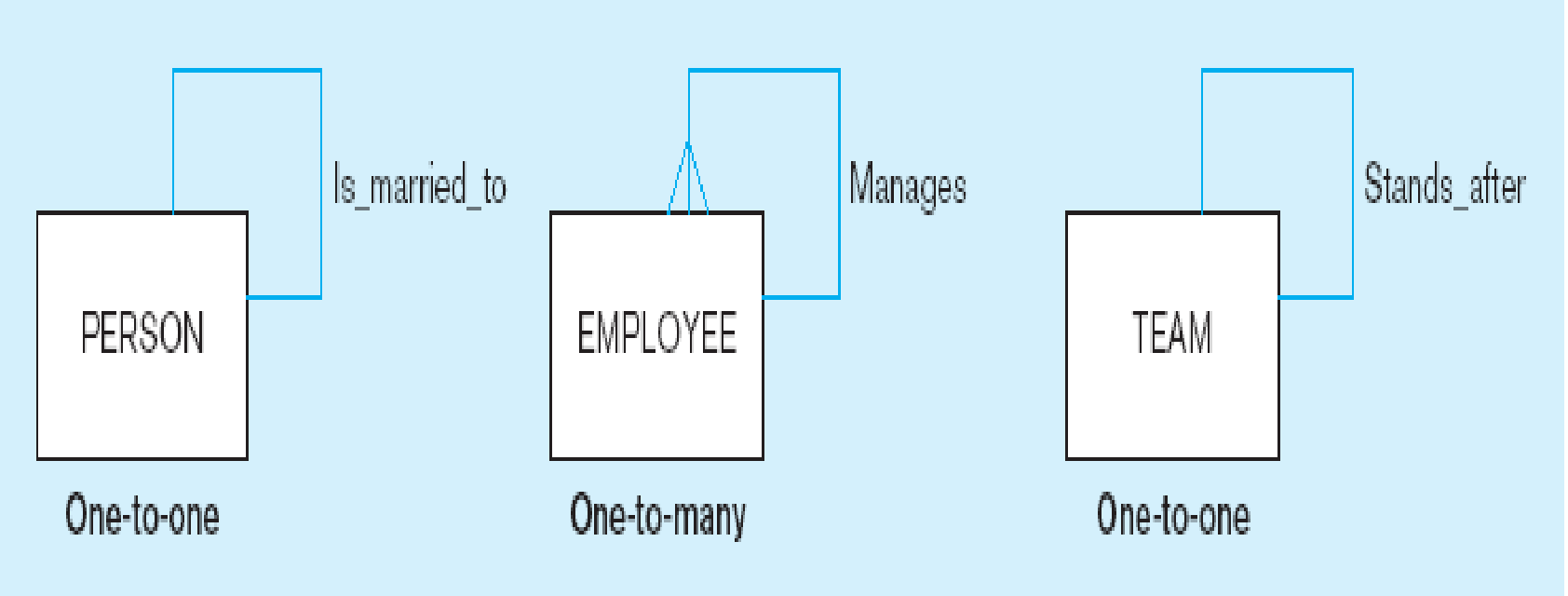
- **One-to-One**
  - Each entity in the relationship will have exactly one related entity
- **One-to-Many**
  - An entity on one side of the relationship can have many related entities, but an entity on the other side will have a maximum of one related entity
- **Many-to-Many**
  - Entities on both sides of the relationship can have many related entities on the other side

Cardinality Constraints

- Cardinality Constraints - the number of instances of one entity that can or must be associated with each instance of another entity
- Minimum Cardinality
  - If zero, then optional
  - If one or more, then mandatory
- Maximum Cardinality
  - The maximum number

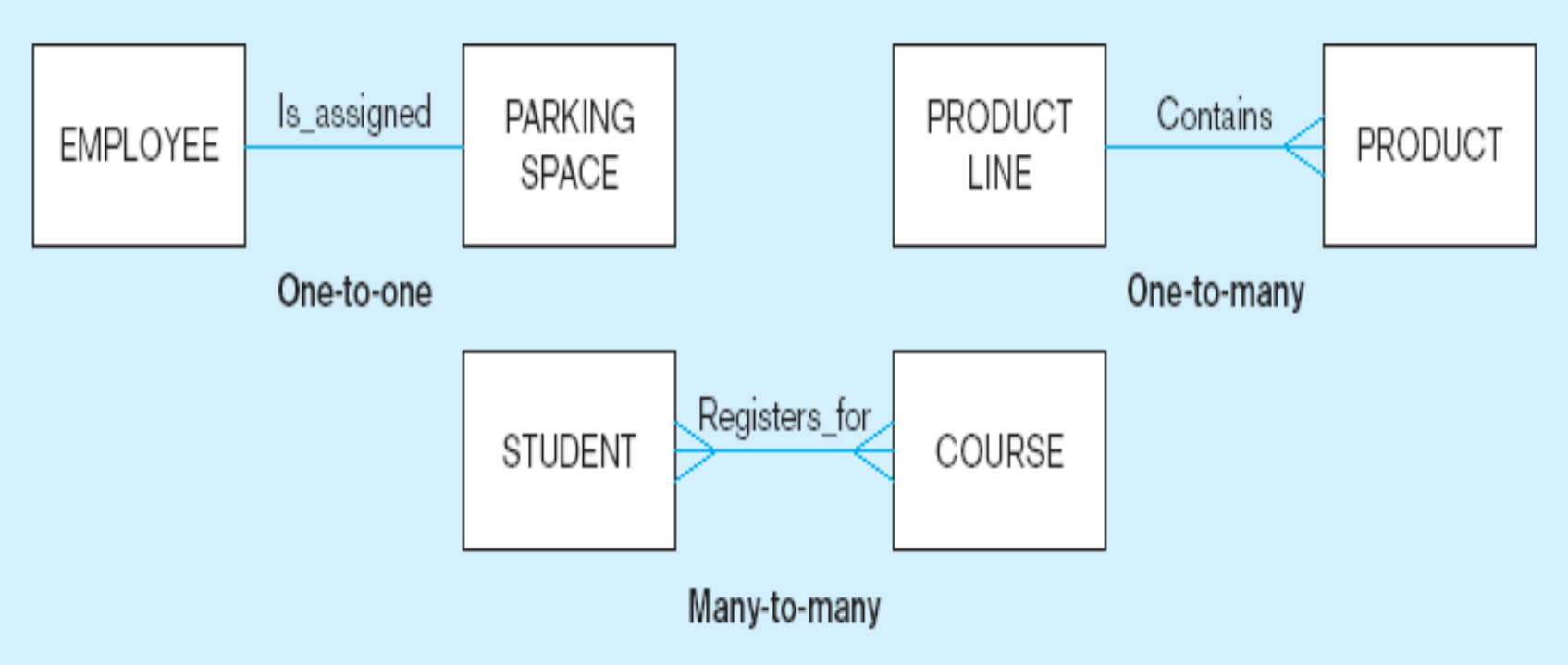
Examples of relationships of different degrees

a) Unary relationships



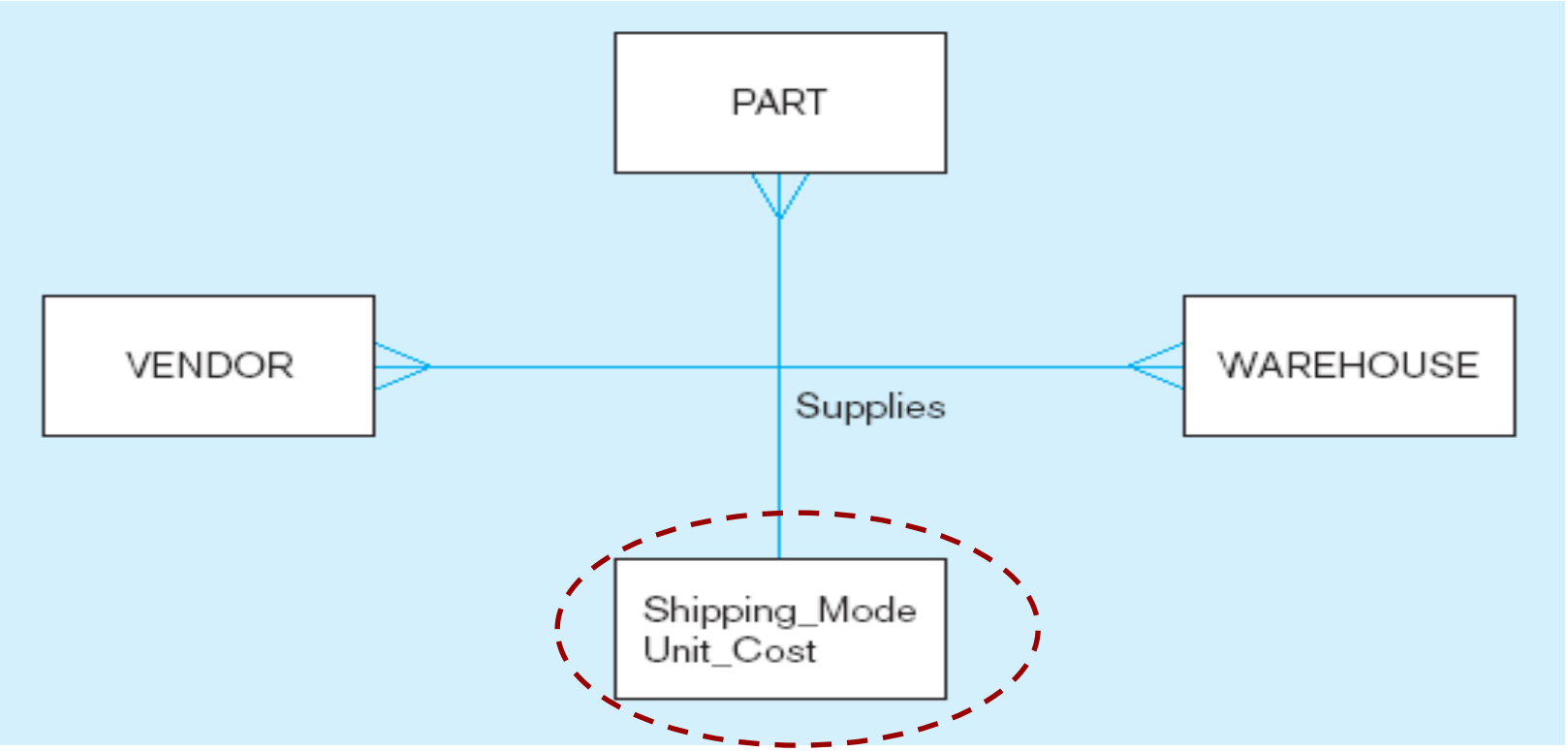
Examples of relationships of different degrees (cont.)

b) Binary relationships



Examples of relationships of different degrees (cont.)

c) Ternary relationship



Note: a relationship can have attributes of its own

Agenda 7/23

- Identifiers or Keys
- Data Integrity
- Advanced Entity + Constrains
- Announcement
  - Next week Monday and Tuesday is the holiday

4. Key

- Primary Key (PK)
- Foreign Key (FK)
- Candidate Key
- Composite Key

- Relationship != relation
- Table = relation

Primary Key (PK)

- An attribute or a combination of attributes that uniquely identifies each row in a relation

- Uniqueness and minimal

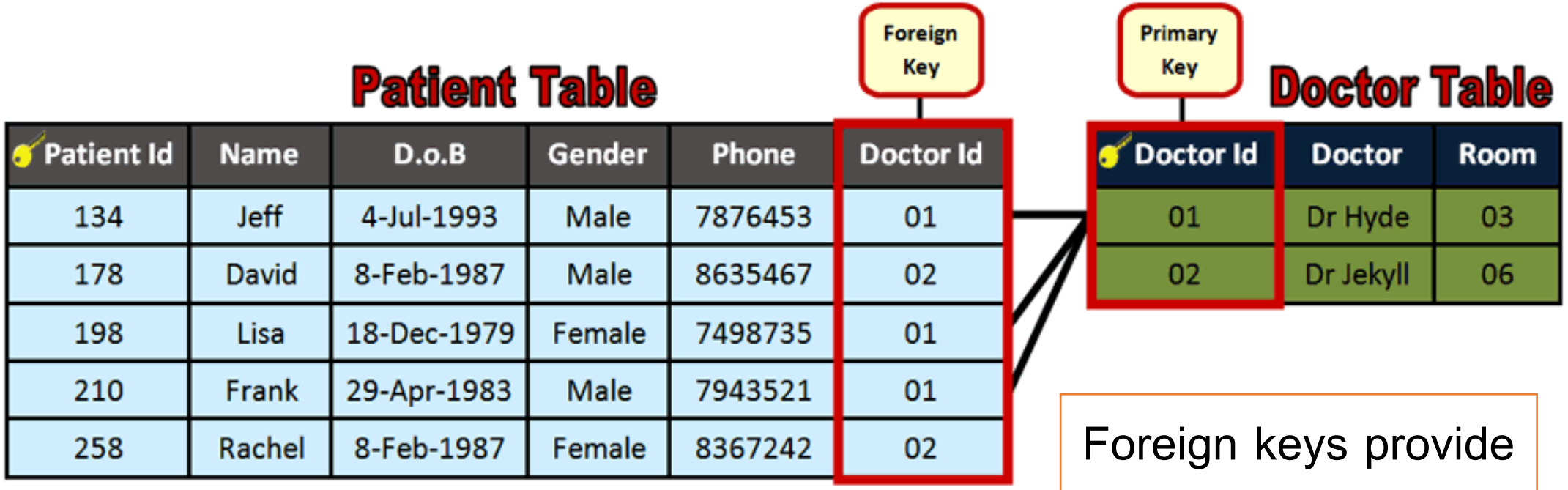
EMPLOYEE (พนักงาน)

EMPNUM	EMPNAME	HIREDATE	SALARY	POSITION	DEPNO	MGRNO
1001	SIRIWAN	06/13/93	9000	CLERK	10	1002
3001	ARLEE	08/15/93	17000	SALESMAN	30	3004
4001	WICHAI	12/26/93	33000	MANAGER	40	2002
1002	JINTANA	10/31/93	30000	CONTROLLER	10	1003
3002	MITREE	12/05/93	13000	SALESMAN	30	3004
3003	BENJAWAN	06/11/94	29000	MANAGER	30	2002
2001	CHAI	05/14/93	14000	CLERK	20	2003
1003	SURASIT	03/15/94	30000	MANAGER	10	2002
2002	KANJANA	07/10/94	50000	DIRECTOR	20	
3004	TANACHOTE	06/14/94	25000	SUPERVISOR	30	3003
1004	AMPORN	06/04/94	12000	CLERK	10	1002
3005	TAWATCHAI	07/03/94	10000	SALESMAN	30	3004
4002	THIDARAT	12/01/94	9000	CLERK	40	4001
2003	TERNJAI	11/01/94	24000	MANAGER	20	2002

รูป 4.2

Foreign key (FK)

- An attribute in a relation that serves as the primary key of another relation in the same database



Patient Id	Name	D.o.B	Gender	Phone	Doctor Id
134	Jeff	4-Jul-1993	Male	7876453	01
178	David	8-Feb-1987	Male	8635467	02
198	Lisa	18-Dec-1979	Female	7498735	01
210	Frank	29-Apr-1983	Male	7943521	01
258	Rachel	8-Feb-1987	Female	8367242	02

Doctor Id	Doctor	Room
01	Dr Hyde	03
02	Dr Jekyll	06

- database must not contain any unmatched foreign key values.

Foreign keys provide the "links" between two relations.

Candidate key

- An attribute, or combination of attributes, that uniquely identifies a row in a relation

candidate key

Employee_ID	FirstName	LastName	Email	Deartment_ ID
1000	Arun	Jayaram	arun@software developersworld.com	100
1001	Manoj	Shankar	manoj@software developersworld.com	100
1002	Syam	Sundar	syam@software developersworld.com	102

Composite key

ProductID	VendorID	AverageLeadTime	StandardPrice	LastReceiptCost
1	1	17	47.8700	50.2635
2	104	19	39.9200	41.9160
7	4	17	54.3100	57.0255
609	7	17	25.7700	27.0585
609	100	19	28.1700	29.5785

ProductVendor table

- A primary key that consists of more than one attribute
- More than one attribute is needed to make the entity unique

Quick Workshop - Registration database

Please list all of PK, FK, Candidate Key, Composite key?

Student_id	Course_id	term	Academic_year	section	grade	Lecturer_id

Student_id	name	surname	address	Telephone_no

Course_id	Course_name	Description	Unit	Type

Lecture_id	LName	LSurname	Telephone	Major_id

Major_id	Major_name

Null Values

- NULL is not the number zero.
- NULL is not the empty string value.
- NULL is the value used to represent an unknown piece of data.
- Comparisons between two **null values**, or between a **NULL** and any other **value**, return unknown because the **value** of each **NULL** is unknown.

5. Data Integrity Rules/Integrity Constraints

- The relational data model includes several types of constraints, or rules limiting accept- able values and actions, whose purpose is to facilitate maintaining the accuracy and integrity of data in the database.
- The major types of integrity constraints are domain intregrity, entity integrity, and referential integrity

Entity Integrity

- Every table must have a primary key and that the column or columns chosen to be the primary key should be unique and not null.

Code	Name
441210073	**The Intelligent Hammer**
441210093	**The Gigantic Mage**
	Captain Wacky Killer
	Venombite

Entity Integrity

- Is designed to ensure that every relation has a primary key and that the data values for that primary key are all valid
- Guarantees that every primary key attribute is non-null

Entity integrity rule

- No primary key attribute (or component of a primary key attribute) can be null

Domain Integrity

- Domain integrity means the definition of a valid set of values for an attribute. You define
  - data type,
  - length or size
  - is null value allowed
  - is the value unique or not

  for an attribute.

Pcode	Pname	Ptype	QTY	Price
001	Pen	01	10	50
002	Book	01	20	200
003	Table	03	5.5	1,600
004	Lamp	02	4	500

Domain Integrity

- All of the values that appear in a column of a relation must be from the same domain
- Domain is the set of values that may be assigned to an attribute
- Domain definition usually consists of the following components: domain name, meaning, data type, size (or length), and allowable values or allowable range (if applicable)

Domain Definitions for INVOICE Attributes

Attribute	Domain Name	Description	Domain
Cust_ID	Customer ID	Set of all possible customer ID	Character: size 5
Cust_Name	Customer Names	Set of all possible customer names	Character: size 25
Order_ID	Order ID	Set of all possible order ID	Character: size 5
OrderDate	Order Dates	Set of all possible order dates	Date: format mm/dd/yy
Product_ID	Product ID	Set of all possible product ID	Character: size 5
StandardPrice	Unit Prices	Set of all possible unit prices	Double
OrderQuantity	Quantities	Set of all possible orders quantities	Integer: 3 digits

Referential Integrity

- Any value of foreign key in the relation must be refer to the primary key value in another relation. Otherwise it must be null.

Product

Pcode	Pname	Ptype	QTY	Price
001	Pen	01	10	5
002	Pencil		20	2
003	Table	03	5	1,600
004	Lamp	02	4	500

Product Type

Ptype	Pname	Location
01	Stationary	A2
02	Electronics	A5

Referential Integrity

- Is a rule that maintains consistency among the rows of two relations
- If there is a foreign key in one relation, either each foreign key value must match a primary key value in another relation or the foreign key value must be null

Referential integrity constraint

- A rule that states that either each foreign key value must match a primary key value in another relation or the foreign key value must be null

Modification affect to Referential Integrity

- Delete the rows in the relation that their primary key are the foreign key in another relation.
- Modify the primary key value in the relation that refer to foreign key in another relation.

Modification affect to Referential Integrity

Product

Pcode	Pname	Ptype	QTY	Price
001	Pen	01	10	5
002	Pencil	01	20	2
003	Desk	02	5	1,600
004	Lamp	02	4	500

Break the rule

Product Type

Ptype	Pname	Location
01	Stationary	A2
02	Electrical	A5

Change 02 to 04

3 cases to keep Referential Integrity

1. **Cascade**

changing the primary key of a row in the primary table, the foreign key values are updated in the matching rows in the related table.

2. **Restricted**

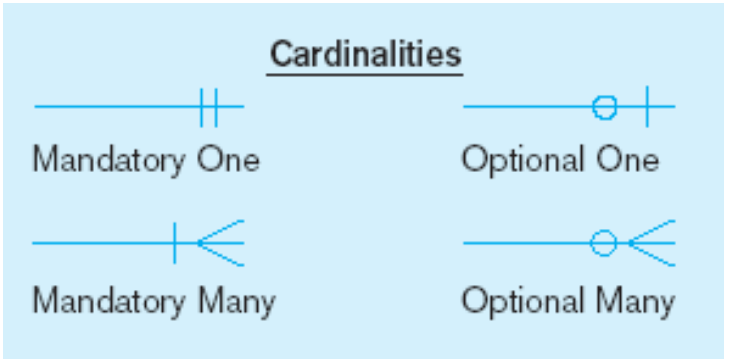
changing the primary key of a row in the primary table can be done when there not refer to another relation as foreign key. Otherwise, modification can not be done.

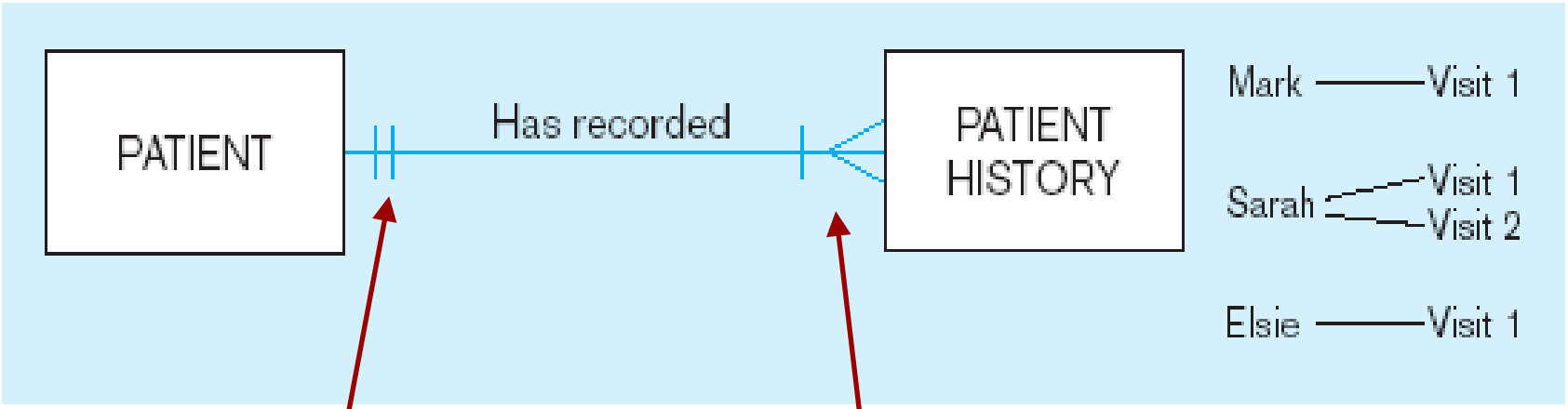
3. **Nullifies**

changing the primary key of a row in the primary table, the foreign key values are set to null.

Examples of cardinality constraints

a) Mandatory cardinalities



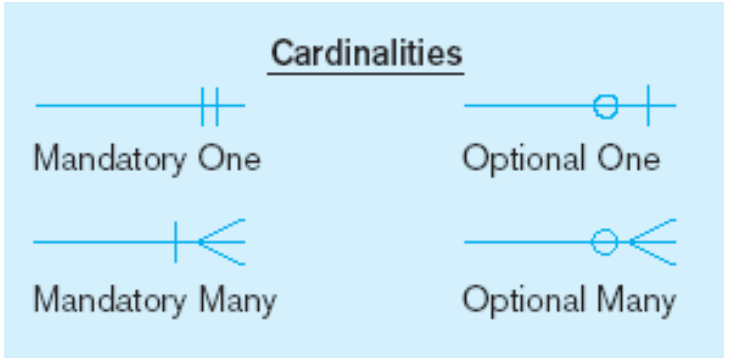


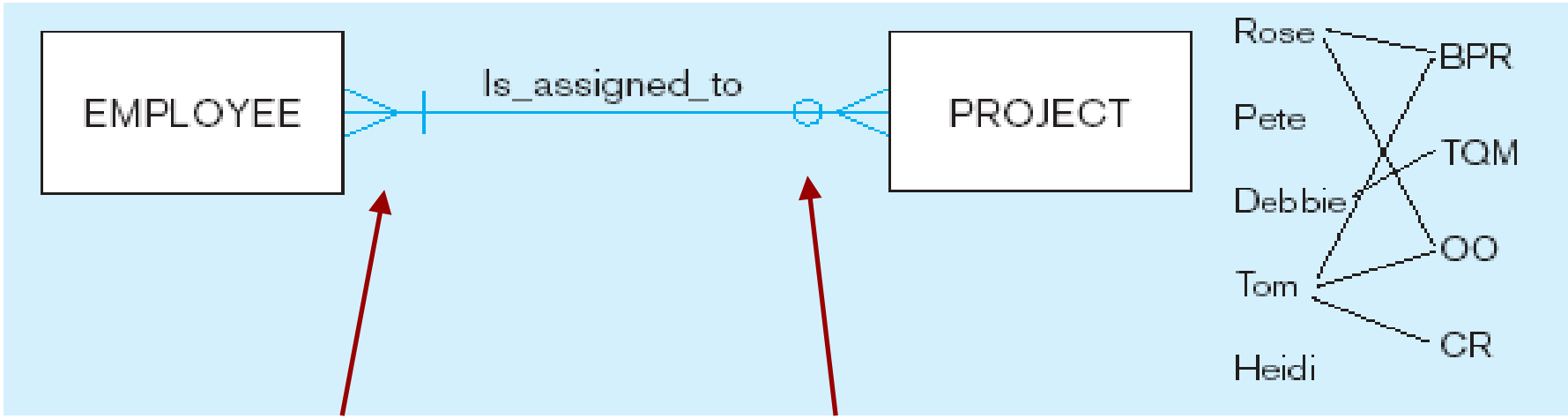
A patient history is recorded for one and only one patient

A patient must have recorded at least one history, and can have many

Figure 3-17 Examples of cardinality constraints (cont.)

b) One optional, one mandatory



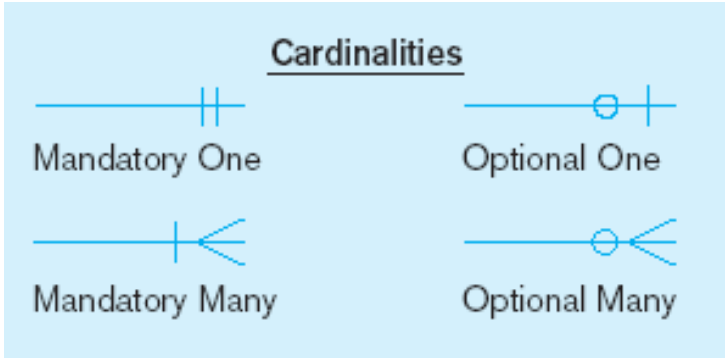


A project must be assigned to at least one employee, and may be assigned to many

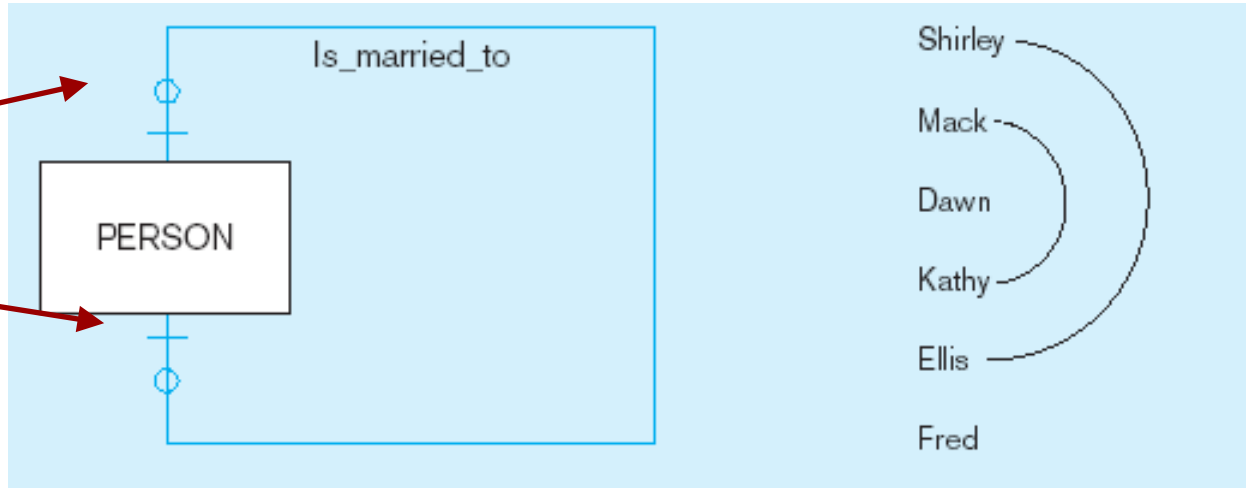
An employee can be assigned to any number of projects, or may not be assigned to any at all

Examples of cardinality constraints (cont.)

a) Optional cardinalities

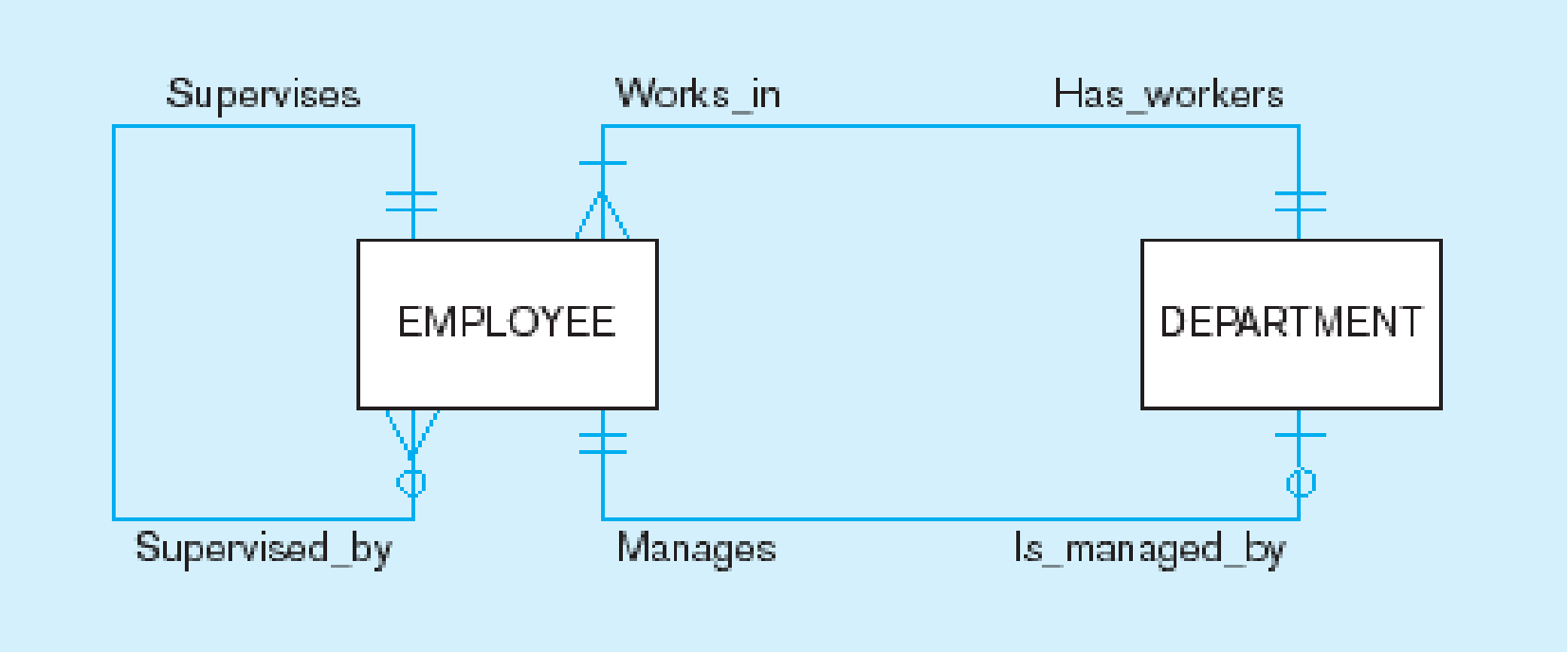


A person is is married to at most one other person, or may not be married at all



Examples of multiple relationships

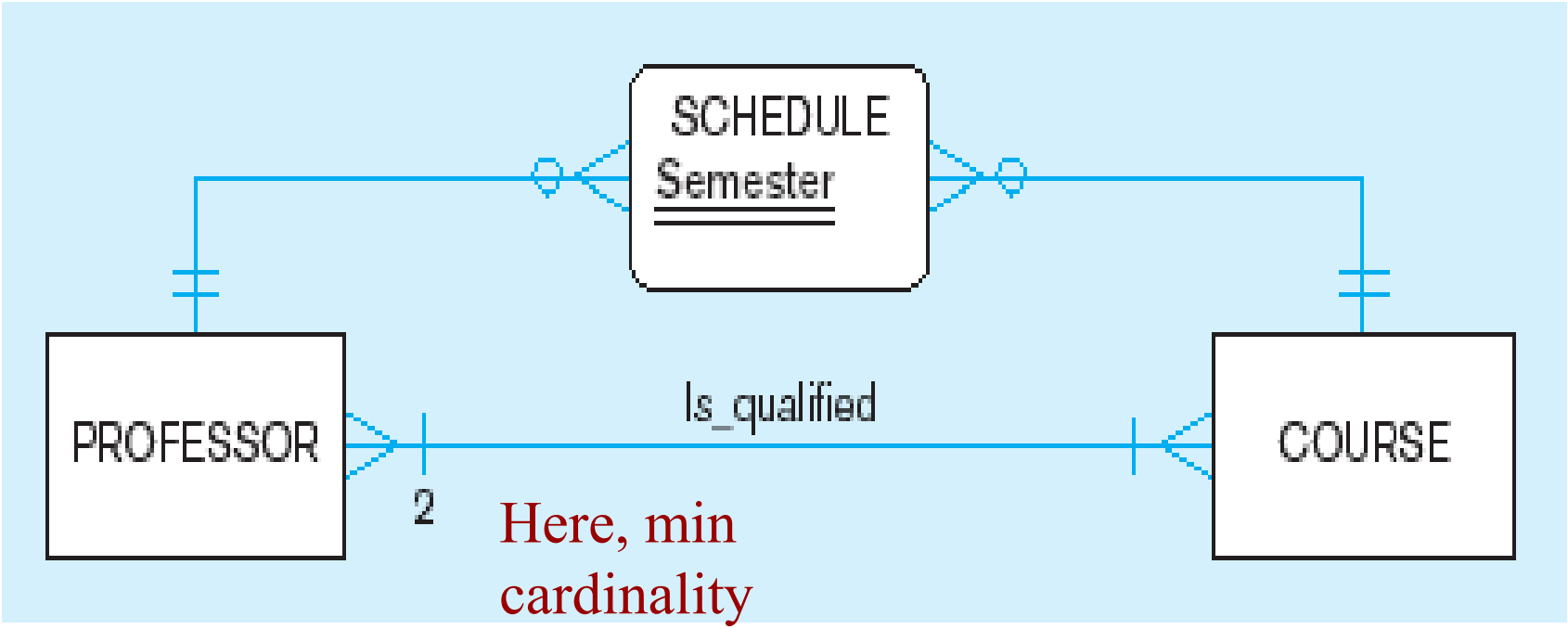
a) Employees and departments



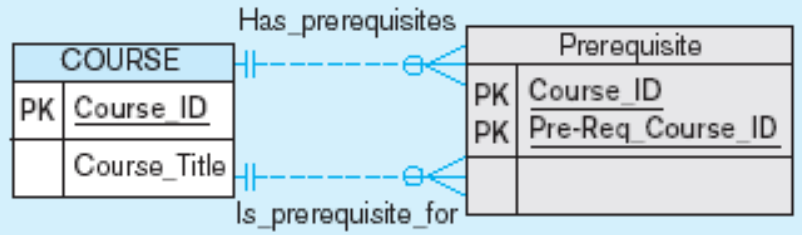
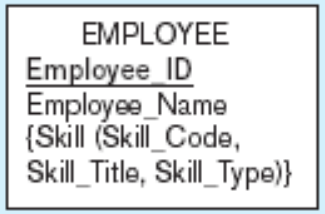
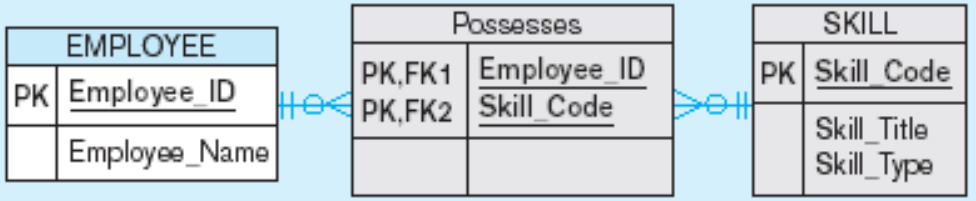
Entities can be related to one another in more than one way

Examples of multiple relationships (cont.)

b) Professors and courses (fixed lower limit constraint)



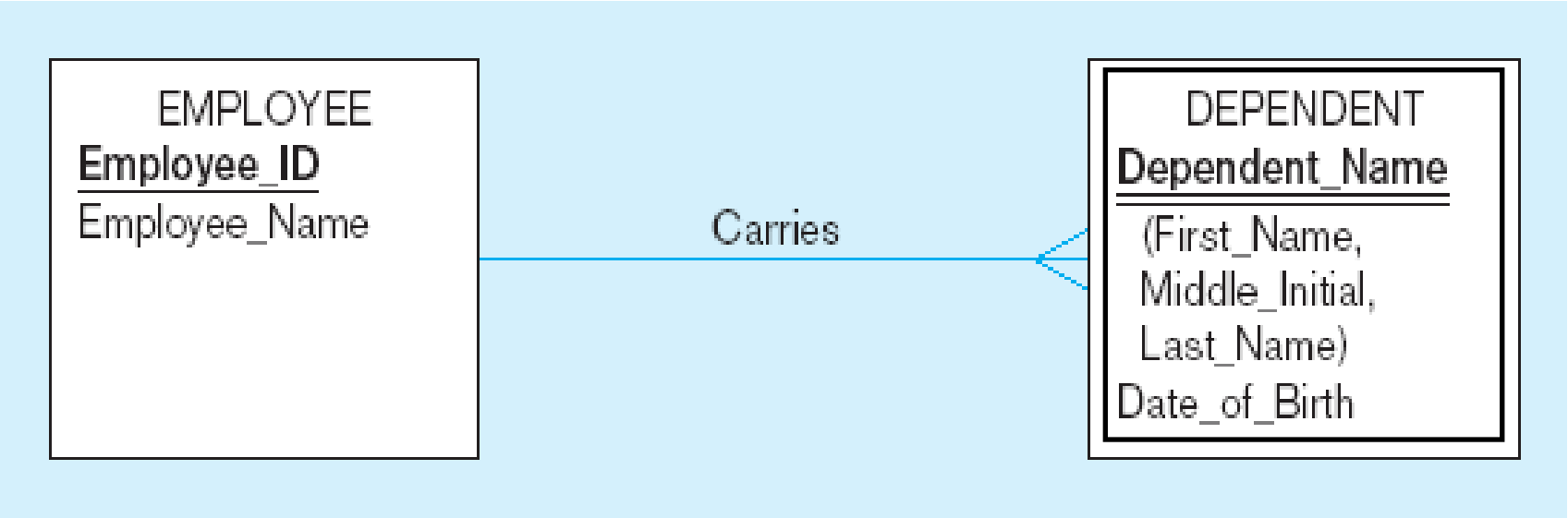
Multivalued attributes can be represented as relationships

ATTRIBUTE	RELATIONSHIP & ENTITY
simple	
 composite	

Strong vs. Weak Entities, and Identifying Relationships

- Strong entities
  - exist independently of other types of entities
  - has its own unique identifier
  - identifier underlined with single-line
- Weak entity
  - dependent on a strong entity (identifying owner)…cannot exist on its own
  - does not have a unique identifier (only a partial identifier)
  - Partial identifier underlined with double-line
  - Entity box has double line
- Identifying relationship
  - links strong entities to weak entities

Identifying relationship



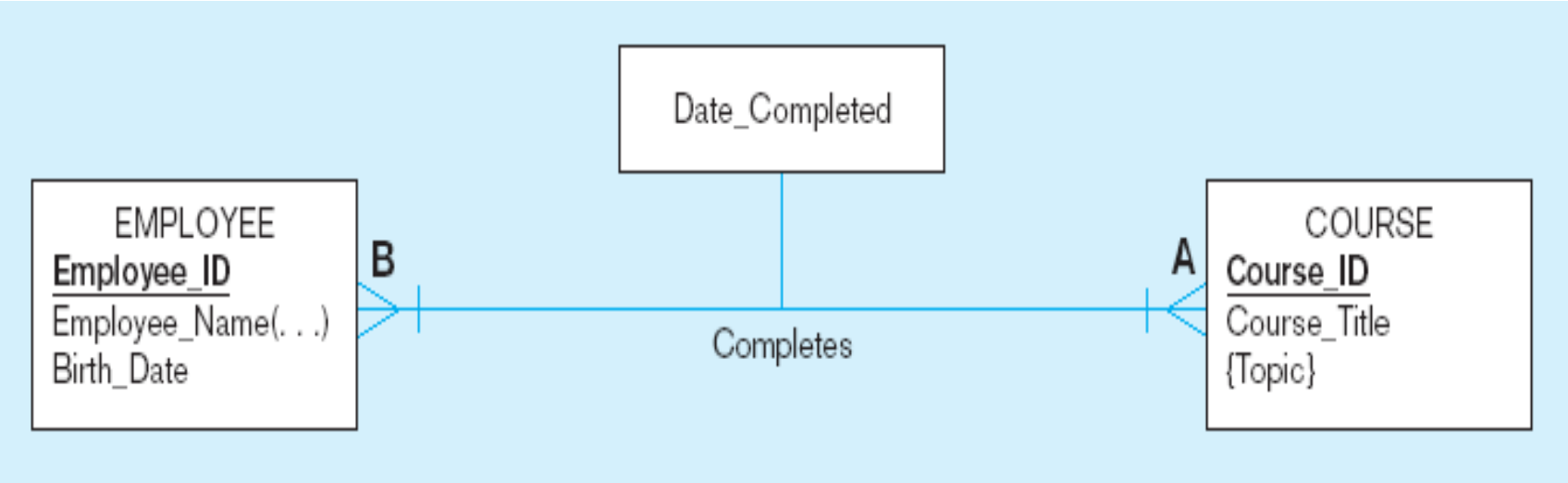
Strong entity

Weak entity

Associative Entities

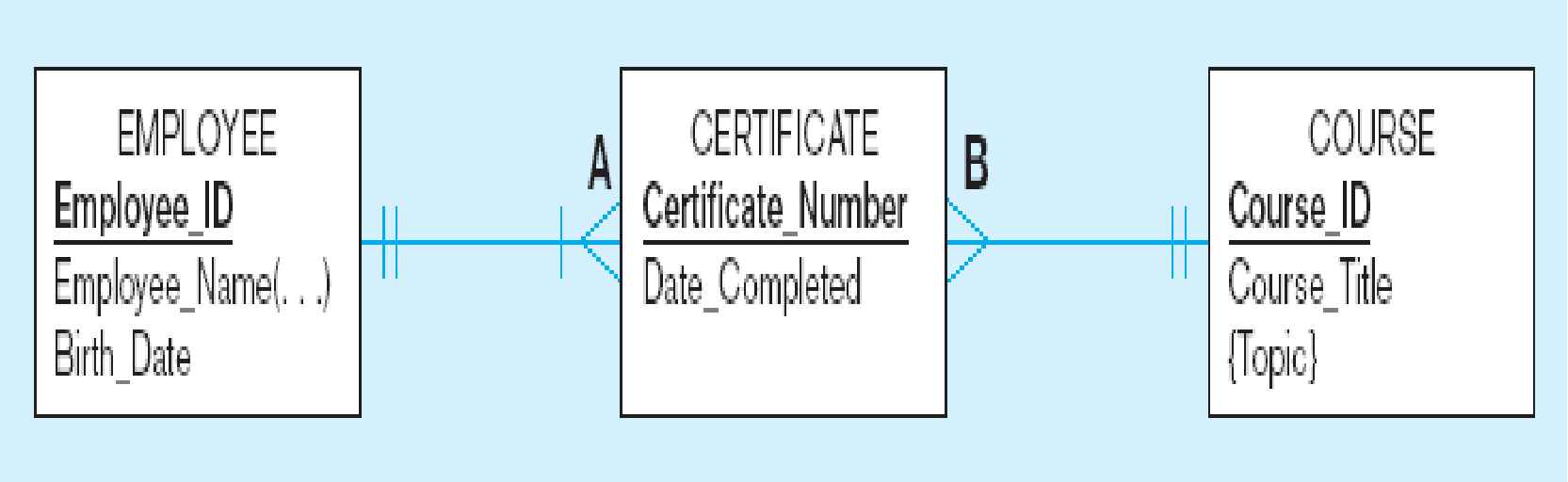
- An entity–has attributes
- A relationship–links entities together
- When should a *relationship with attributes* instead be an *associative entity*?
  - All relationships for the associative entity should be many
  - The associative entity could have meaning independent of the other entities
  - The associative entity preferably has a unique identifier, and should also have other attributes
  - The associative entity may participate in other relationships other than the entities of the associated relationship
  - Ternary relationships should be converted to associative entities

A binary relationship with an attribute



Here, the date completed attribute pertains specifically to the employee's completion of a course…it is an attribute of the *relationship*

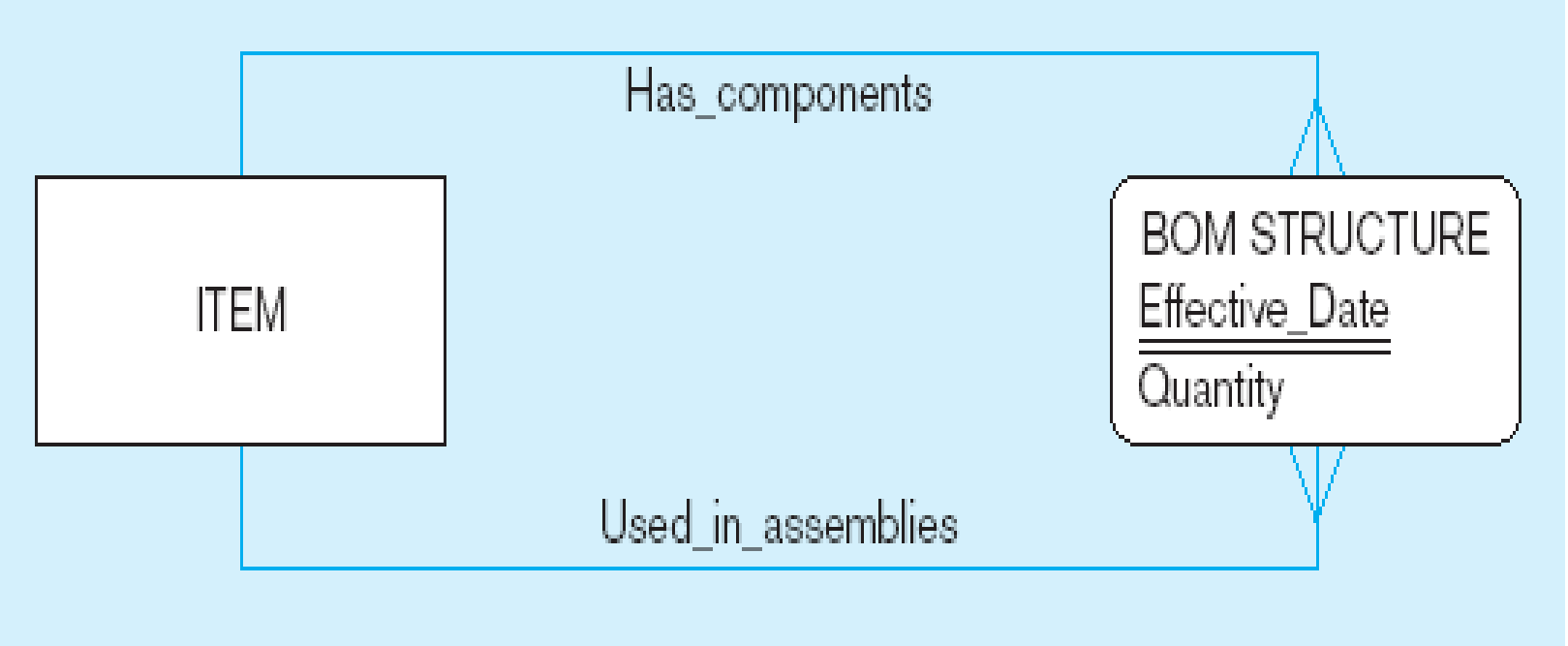
An associative entity (CERTIFICATE)



Associative entity is like a relationship with an attribute, but it is also considered to be an entity in its own right.

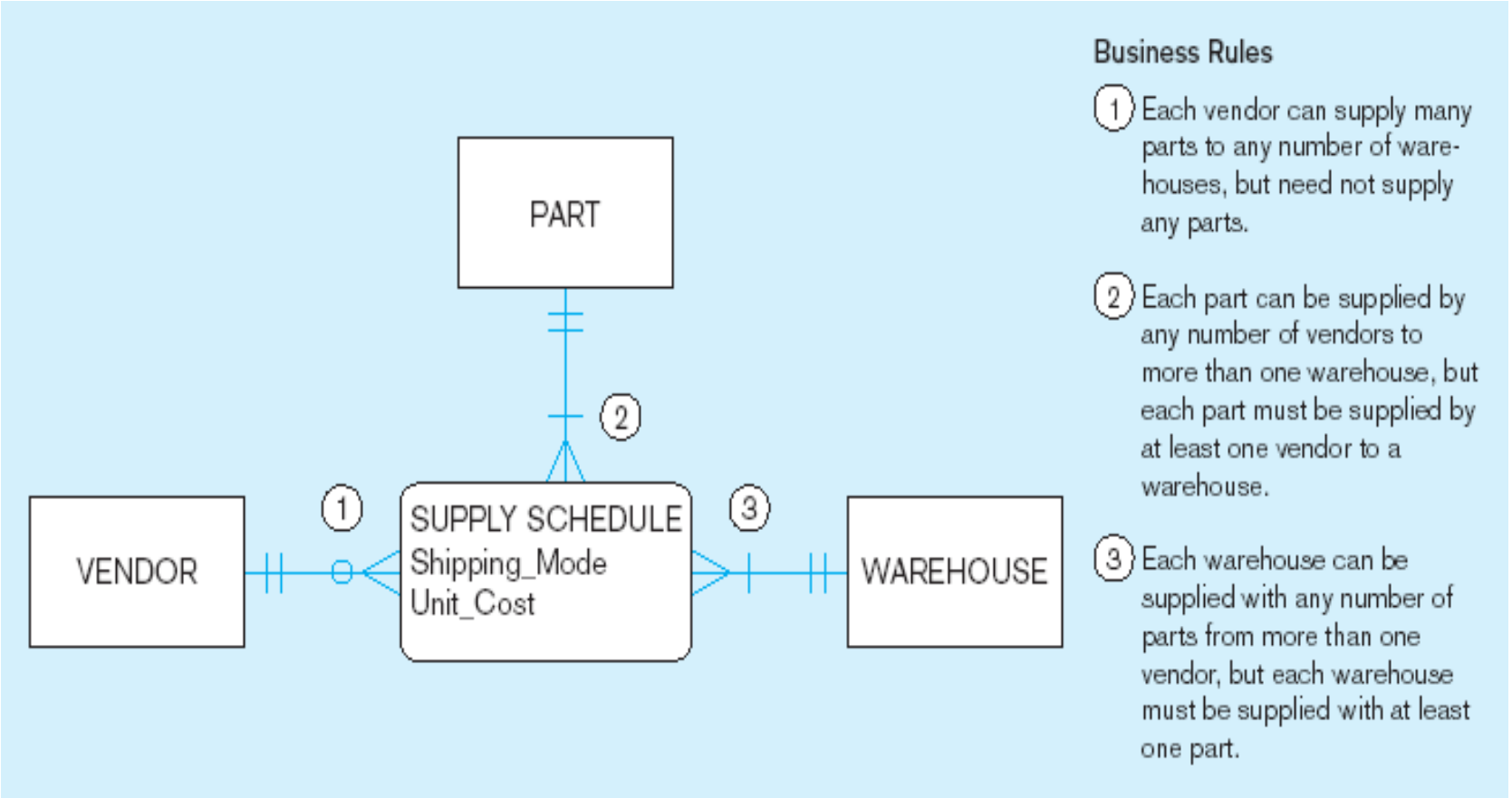
Note that the many-to-many cardinality between entities in Figure 3-11a has been replaced by two one-to-many relationships with the associative entity.

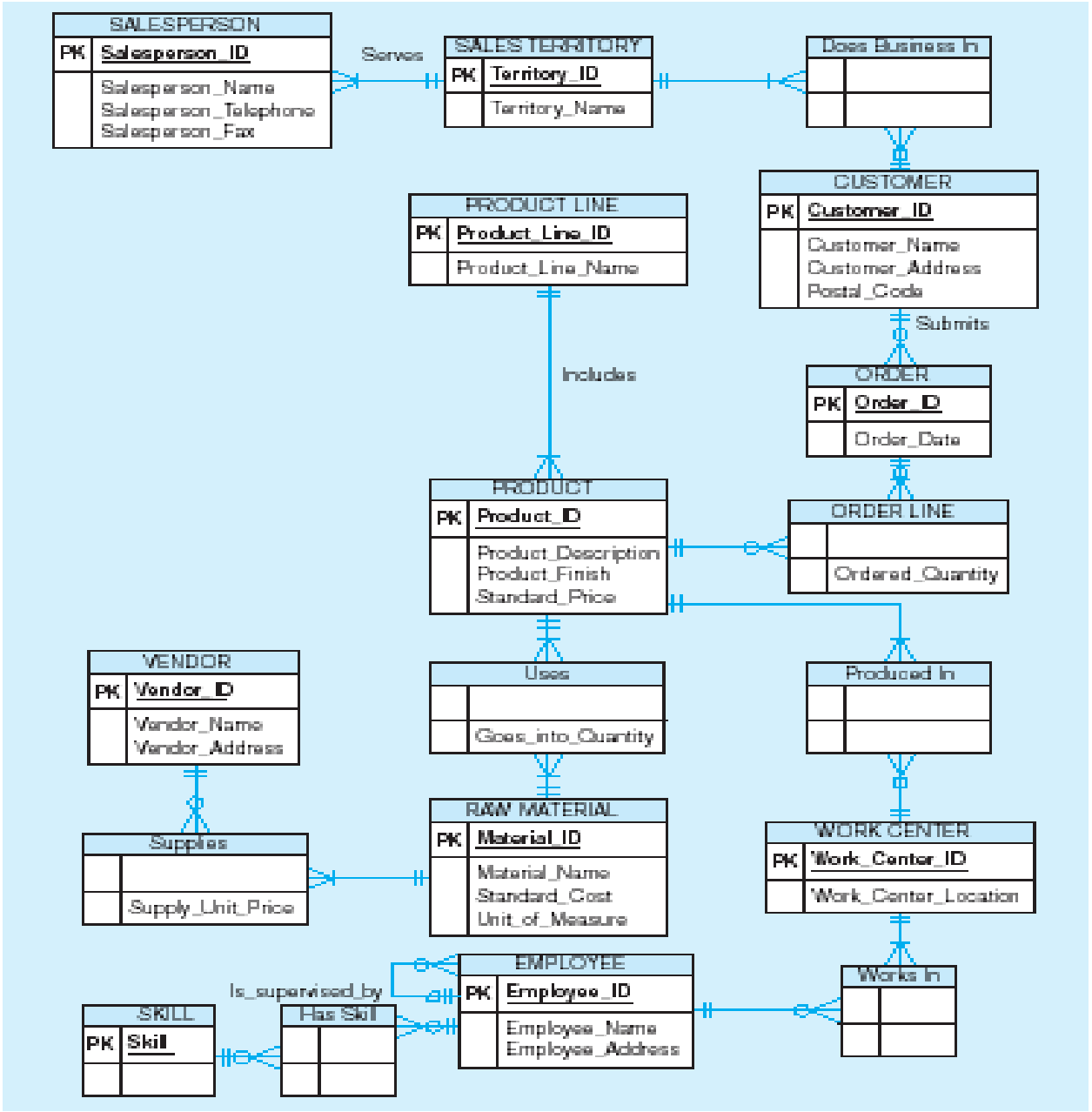
An associative entity – bill of materials structure



This could just be a relationship with attributes…it's a judgment call

Ternary relationship as an associative entity





Microsoft Visio Notation for Pine Valley Furniture E-R diagram

Different modeling software tools may have different notation for the same constructs

1, different type of keys involving primary key, foreign key, candidate
2, associative entity
3, Strong and weak entity
4, domain integrity, referential integrity, entity ingrity
5. How to convert trinary to asso. Entity
6, How to convert mutivalue attr. To entity
7 How to convert compsite attr. To entity