INFORMATION SYSTEM DEPARTMENT
INFORMATION TECHNOLOGY FACULTY – HCM
UNIVERSITY OF SCIENCE

ADVANCED DATABASE

Chapter 02

DATA MODELS, BASIC ERD

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

fit@hcmus

Outline

- Data Modeling & Data Models
- The Evolution of Data Models
- Data Abstraction Levels
- Business Rules
- Entity Relationships Model (Basic)

Outline

- **Data Modeling & Data Models**
- The Evolution of Data Models
- Data Abstraction Levels
- Business Rules
- Entity Relationships Model (Basic)

cdio 4.0
fit@hcmus

Data Modeling & Data Models

- **Data modeling**: Iterative and progressive process of creating a specific data model for a determined problem domain
- **Data model**: Simple representation of complex real-world data structures
  - Useful for supporting a specific problem domain
- **Model**: Abstraction of a more complex real-world object or event

(Coronel and Morris, 2015) (Hoffer et al., 2019)

The Importance of Data Models

- Facilitates communication
- Gives an overview of the database
- Organizes data for various users
- Provides an abstraction for the creation of a good database

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Data Model Basic Building Blocks

- **Entity**: Unique and distinct object used to collect and store data
  - **Attribute**: characteristic of an entity
- **Relationship:** Association among entities
  - One-to-many (1:M OR 1..*)
  - Many-to-many (M:N or *..*)
  - One-to-one (1:1 OR 1..1)
- **Constraint:** Restrictions or rules on data
  - Ensures data integrity

(Coronel and Morris, 2015) (Hoffer et al., 2019)

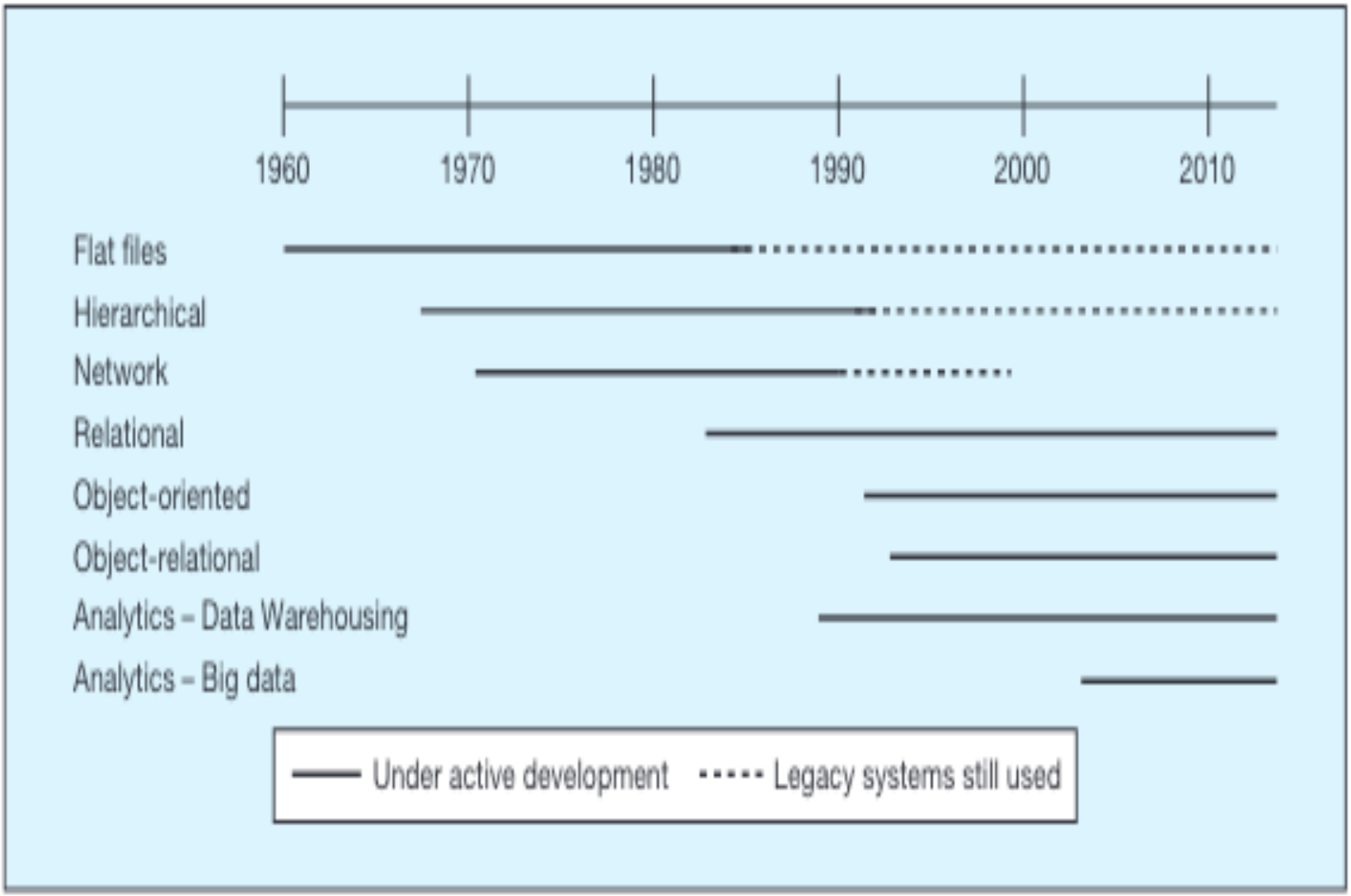
fit@hcmus

Outline

- Data Modeling & Data Models
- **The Evolution of Data Models**
- Data Abstraction Levels
- Business Rules
- Entity Relationships Model (Basic)
- Suggestions for ERD Modeling

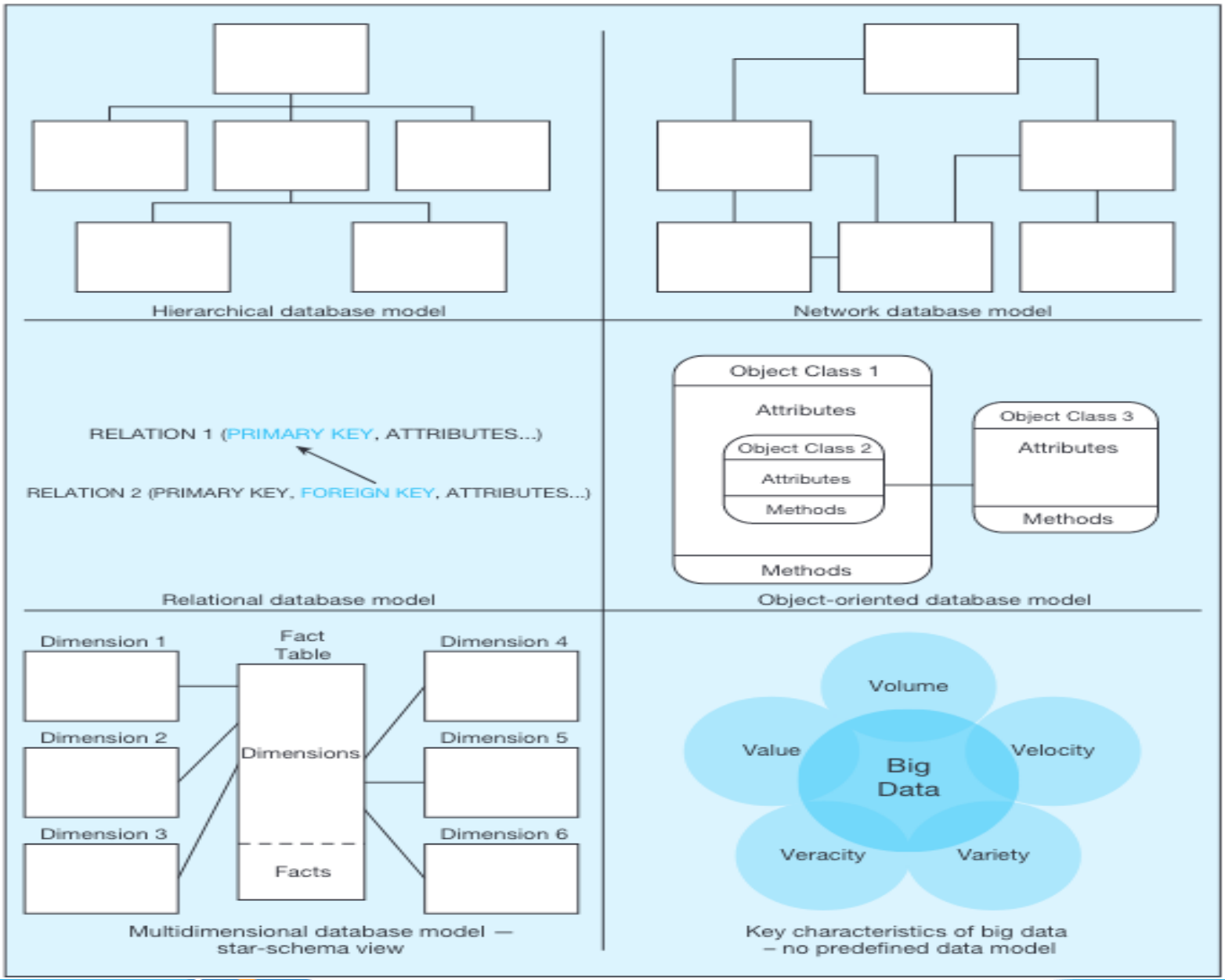
(Coronel and Morris, 2015) (Hoffer et al., 2019)

The Evolution of Data models



(Coronel and Morris, 2015) (Hoffer et al., 2019)

The Evolution of Data models



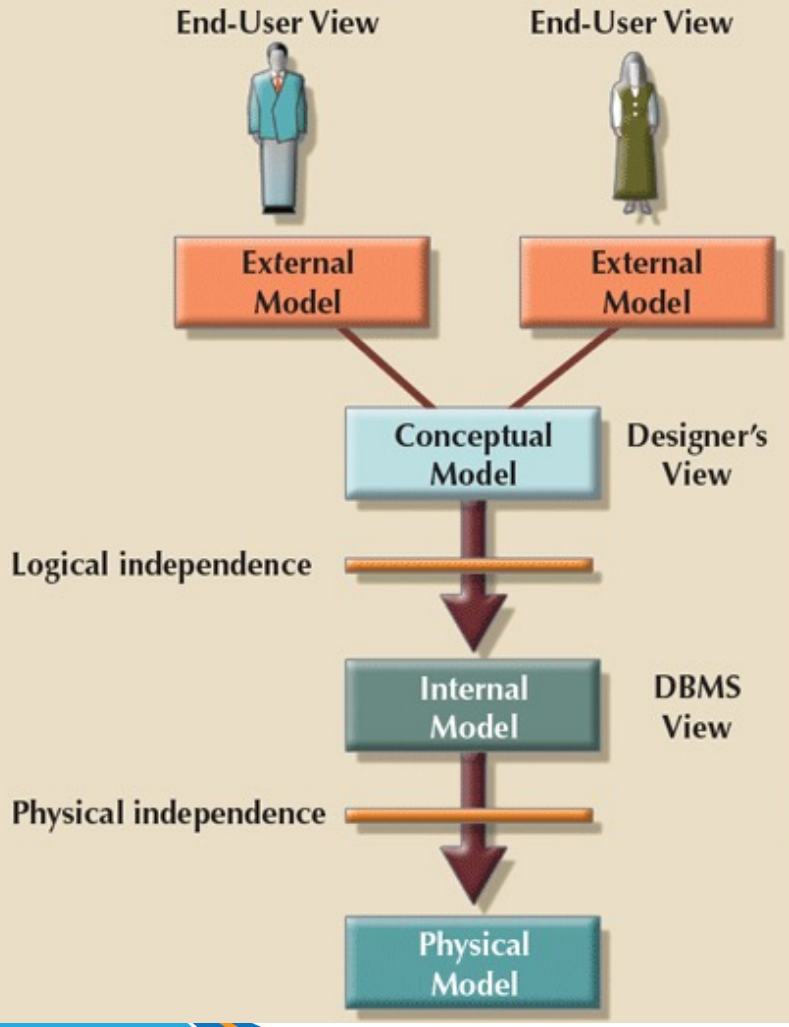
(Coronel and Morris, 2015) (Hoffer et al., 2019)

Outline

- Data Modeling & Data Models
- The Evolution of Data Models
- Data Abstraction Levels
- Business Rules
- Entity Relationships Model (Basic)

Degrees of Data Abstraction

FIGURE 2.6 DATA ABSTRACTION LEVELS



Degree of Abstraction		Characteristics
High	ER Object-Oriented	Hardware-independent Software-independent
Medium	Relational	Hardware-independent Software-dependent
Low	Network Hierarchical	Hardware-dependent Software-dependent

(Coronel and Morris, 2015) (Hoffer et al., 2019)

cdio 4.0

fit@hcmus

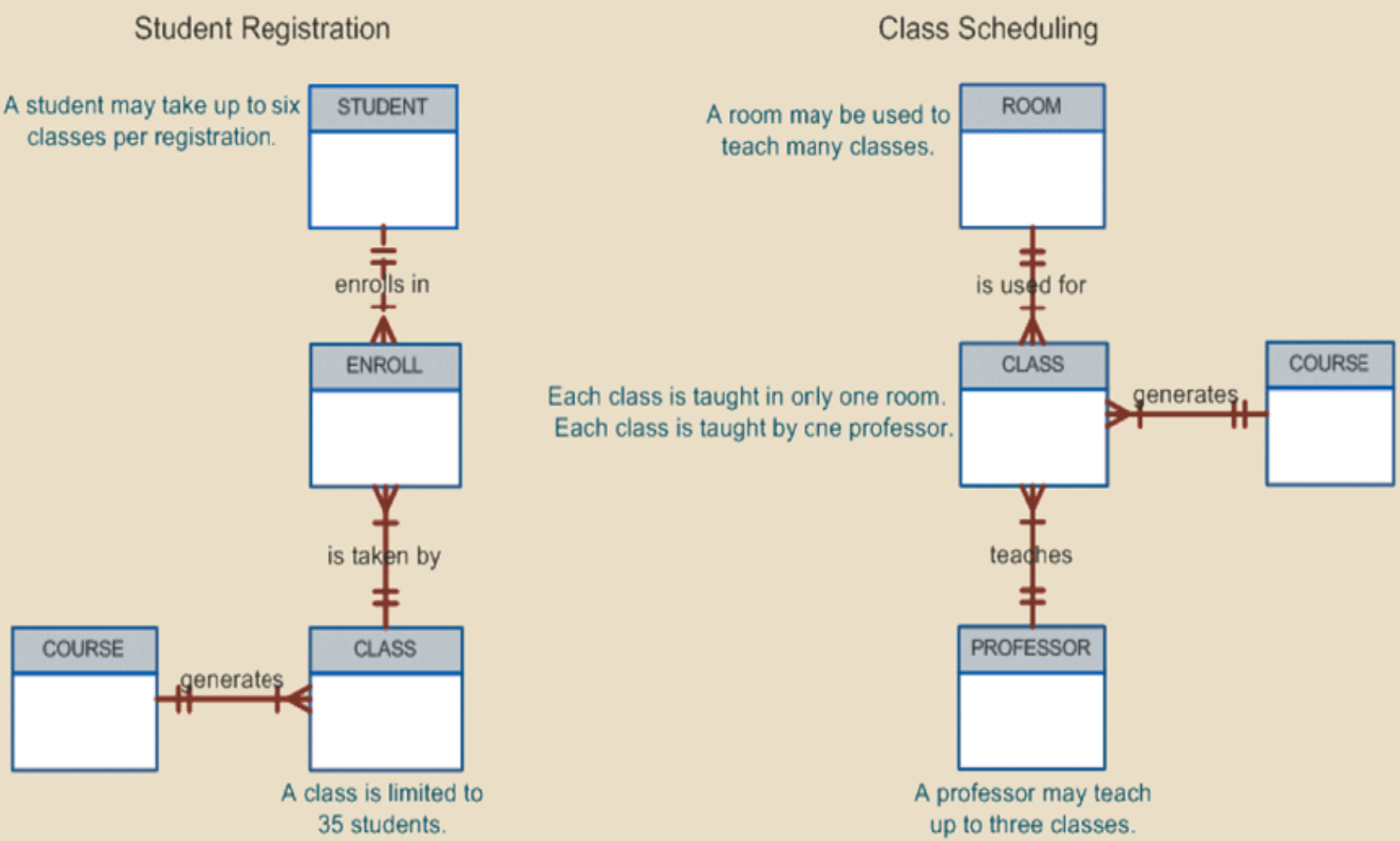
The External Model

- **External model**: Represents **end-user views**
  - End users usually operate in an environment in which an application has a specific business unit focus.
- **External schema:** A specific representation of an external view is known as an
  - Each external schema includes the appropriate entities, relationships, processes, and constraints imposed by the business unit.
  - Using **ERD** to express external schema

(Coronel and Morris, 2015) (Hoffer et al., 2019)

The External Model

FIGURE 2.7 EXTERNAL MODELS FOR TINY COLLEGE



(Coronel and Morris, 2015) (Hoffer et al., 2019)

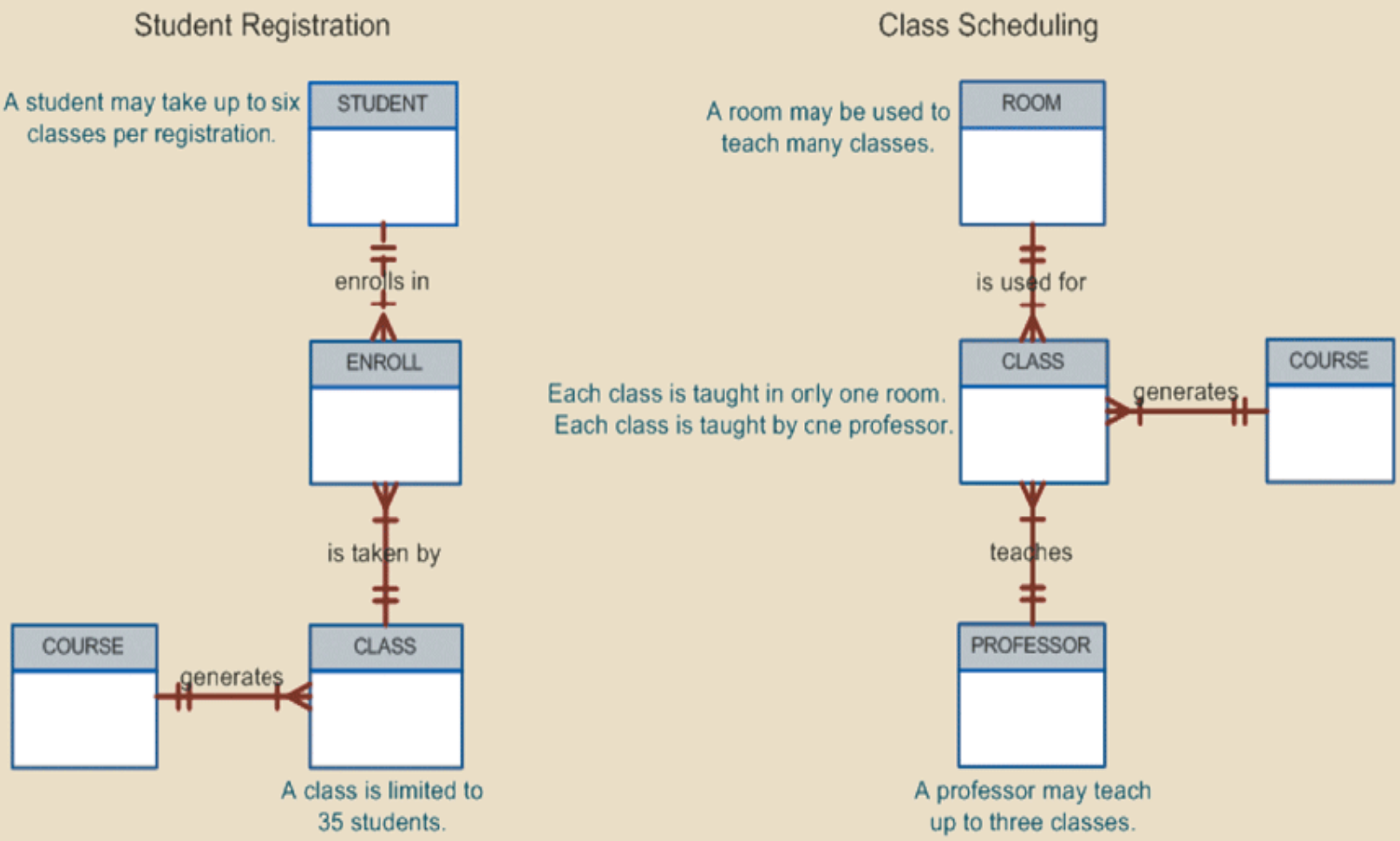
The Conceptual Model

- **Conceptual model**: Represents **a global view** of the entire database by the entire organization
  - Macro-level view of data environment
  - Software and hardware independent
- **Conceptual schema**: Basis for the identification and high-level description of the main data objects
  - Using **ERD** to express conceptual schema

(Coronel and Morris, 2015) (Hoffer et al., 2019)

The Conceptual Model

FIGURE 2.7 EXTERNAL MODELS FOR TINY COLLEGE



(Coronel and Morris, 2015) (Hoffer et al., 2019)

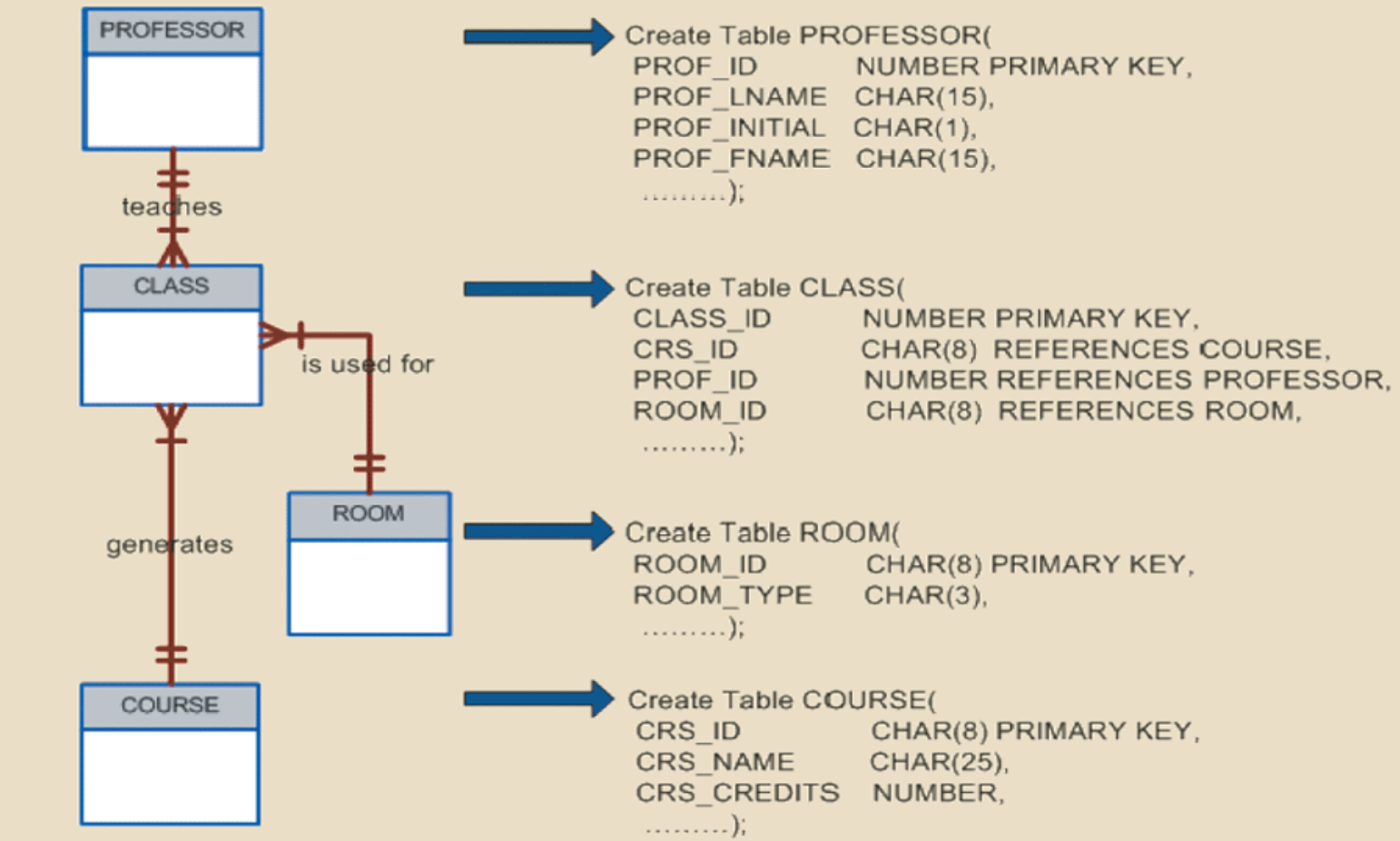
The Internal Model

- **Internal Model**: Represents database as seen by the DBMS mapping conceptual model to the DBMS.
  - Logical independence: changing internal model without affecting the conceptual model
  - Hardware independent: unaffected by the type of computer on which the software is installed
- **Internal schema:** Specific representation of an internal model.
  - Using the database constructs or (database models) supported by the chosen database.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

The Internal Model

FIGURE 2.9 INTERNAL MODEL FOR TINY COLLEGE



(Coronel and Morris, 2015) (Hoffer et al., 2019)

The Physical Model

- **Physical model**: Describes the way data are saved on storage media such as magnetic, solid state, or optical media at the lowest abstraction
  - Physical independence: Changes in physical model do not affect internal model
  - Software and hardware dependent
- Requires the definition of **physical storage** and **data access methods**
- Relational model aimed at logical level
  - Does not require physical-level details

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Summary of Abstraction Levels

TABLE 2.4

LEVELS OF DATA ABSTRACTION

MODEL	DEGREE OF ABSTRACTION	FOCUS	INDEPENDENT OF
External	High	End-user views	Hardware and software
Conceptual	↕	Global view of data (database model independent)	Hardware and software
Internal		Specific database model	Hardware
Physical	Low	Storage and access methods	Neither hardware nor software

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Outline

- Data Modeling & Data Models
- The Evolution of Data Models
- Data Abstraction Levels
- Business Rules
- Entity Relationships Model (Basic)

Business Rules (BR)

- Brief, precise, and unambiguous description (or statement) of a policy, procedure, or principle
- Control/Influence business structures or behaviors
- BR from the database perspective:
  - Ensures the accuracy, completeness, and reliability of data within a business or IT system
  - Describes main and characteristics of the data
  - Enables defining basic data model components (entities, relationships, etc.)

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Sources of Business Rules

- Descriptions of business functions, events, policies, units, stakeholders, and other objects.
  - Interview notes with end users
  - Written documentation
- Process
  - Identifying rules (who, what, why, when, where, how)
  - Clarifying initial statements of rules
  - Formulating rules through an iterative inquiry process

(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Example of Business Rules

- Each student must be affiliated with only one department.
- Student records must have a valid student ID, name, and date of birth.
- A course must be managed by a department.
- Each class section has a maximum enrollment limit of 30 students.
- All university operations must comply with state and federal education laws and accreditation standards.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

The Importance of Business Rules

- Help standardize company's view of data
- Facilitate communication between users and designers
- Assist designers to:
  - Understand the nature, role, scope of data, and business processes
  - Develop appropriate relationship participation rules and constraints
  - Create an accurate data model

(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Translating Business Rules into Data Model Components

- **Nouns** translate into **entities**
- **Verbs** translate into **relationships** among entities
- Relationships are **bidirectional**
- Questions to identify the **relationship type**:
  - How many instances of B are related to one instance of A?
  - How many instances of A are related to one instance of B?

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Outline

- Data Modeling & Data Models
- The Evolution of Data Models
- Data Abstraction Levels
- Business Rules
- Entity Relationships Model (Basic)

(Coronel and Morris, 2015) (Hoffer et al., 2019)

The Entity Relationship Model

- **ER**: Forms the basis of an entity relationship diagram (ERD)
- **ERD**: Depicts **database components**

  (A conceptual database as viewed by end user**)**
  - **Entities**: Indicates entity types, not a single instance
  - **Attributes**: Describes characteristics/properties
  - **Relationships**: Links between entities (entity types)

(Coronel and Morris, 2015) (Hoffer et al., 2019)

cdio 4.0
fit@hcmus

The Entity Relationship Model

- **Advantages**
  - Visual modeling/representation:
    - Produces conceptual simplicity
    - Served as effective communication tools.
  - Integrated with the dominant relational model
- **Disadvantages**
  - Limited constraint & relationship representation due to the lack of data manipulation languages
  - Loss of information content occurs when attributes are removed from entities to avoid crowded displays

(Coronel and Morris, 2015) (Hoffer et al., 2019)

The Entity Relationship Model

- **Entities**:
  - Entity Instance: A person, place, object, event, or concept (~ a row in a table)
  - Entity Type/Set: A set of entity instances which have the same characteristics (~ a table)
- **Relationships**:
  - Relationship Instance: Links between entity instances (~ PK, FK in related tables)
  - Relationship Type: A collection of relationship instances to link between entity types
- **Attributes**:
  - Properties or characteristics of an entity or relationship type (~ a field in a table)

(Coronel and Morris, 2015) (Hoffer et al., 2019)

The Entity Relationship Model

fit@hcmus

Entity Instance vs. Entity (Set/Type)

Entity type: STUDENT				
Attributes	**Attribute Data Type**	**Required or Optional**	**Example Instance**	**Example Instance**
Student ID	CHAR (10)	Required	876-24-8217	822-24-4456
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	

(Coronel and Morris, 2015) (Hoffer et al., 2019)

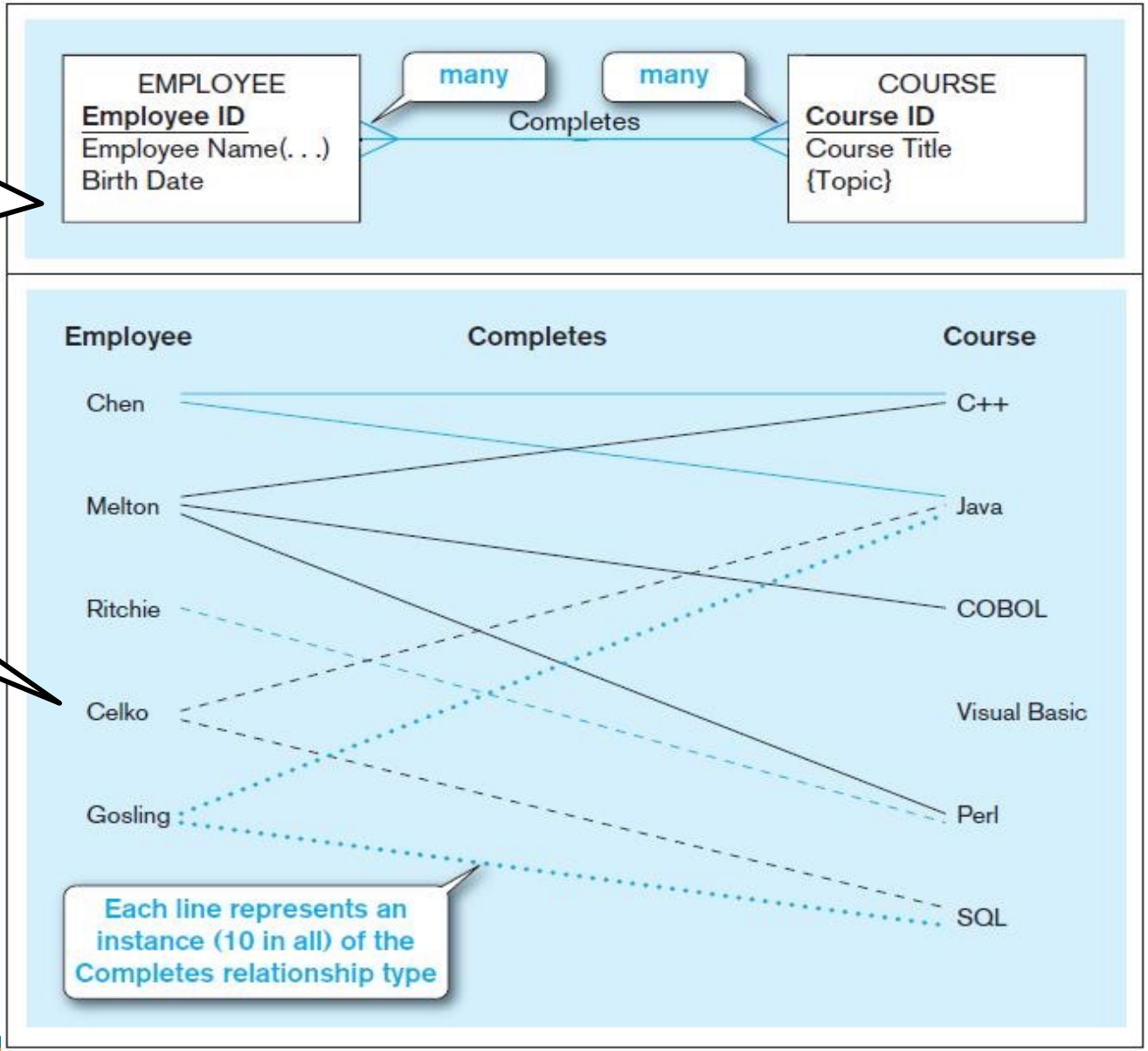
33

The Entity Relationship Model

fit@hcmus

Entity/Relationship Instance vs. Entity/Relationship (Set/Type)

Entity Type:
Employee, Course
Relationship Type:
Completes



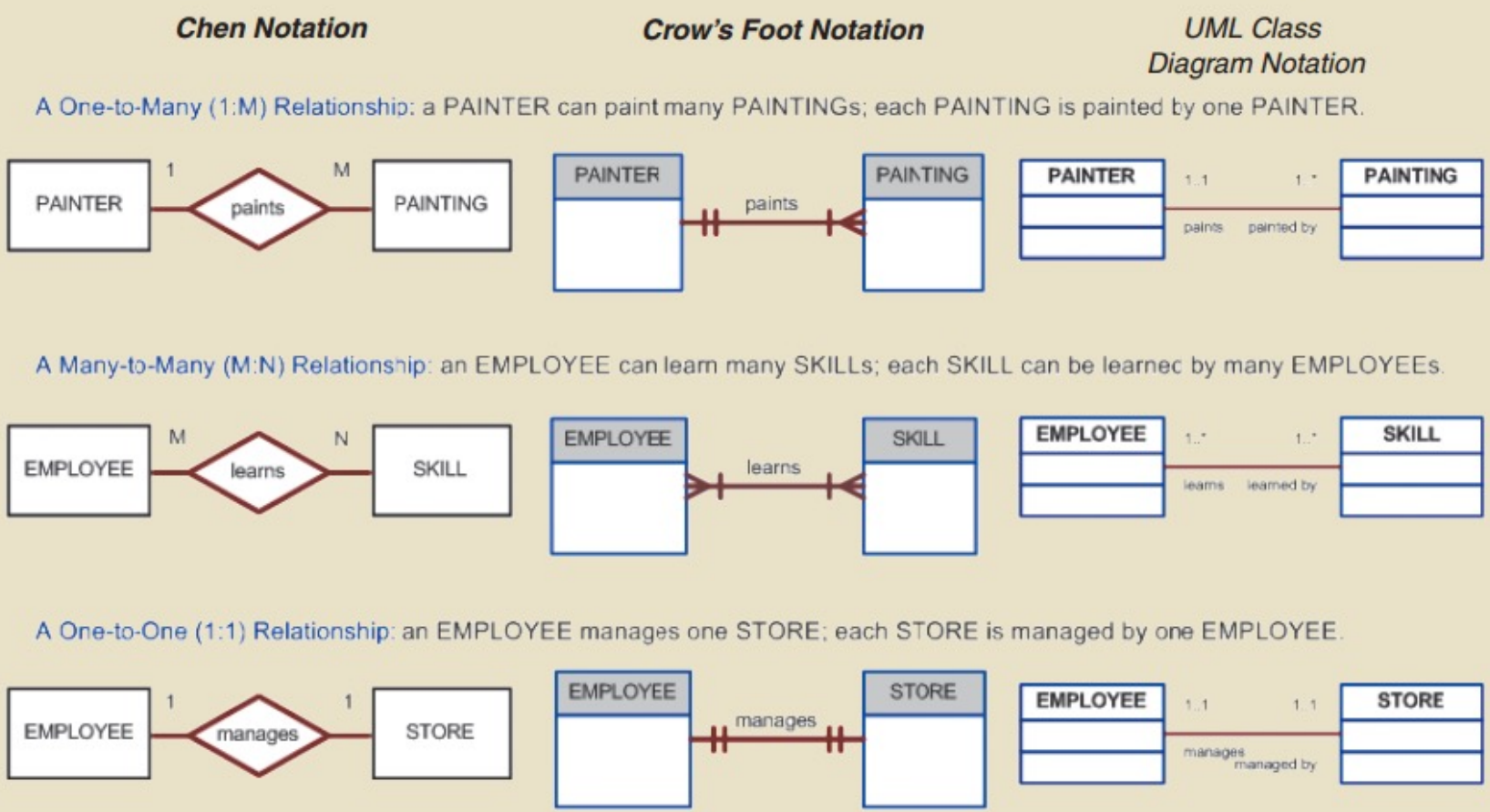
Each **employee** represents an **entity instance** (Chen, …)

(Coronel and Morris, 2015) (Hoffer et al., 2019)

34

The Entity Relationship Model

FIGURE 2.3 THE ER MODEL NOTATIONS



https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model#Crow's_foot_notation

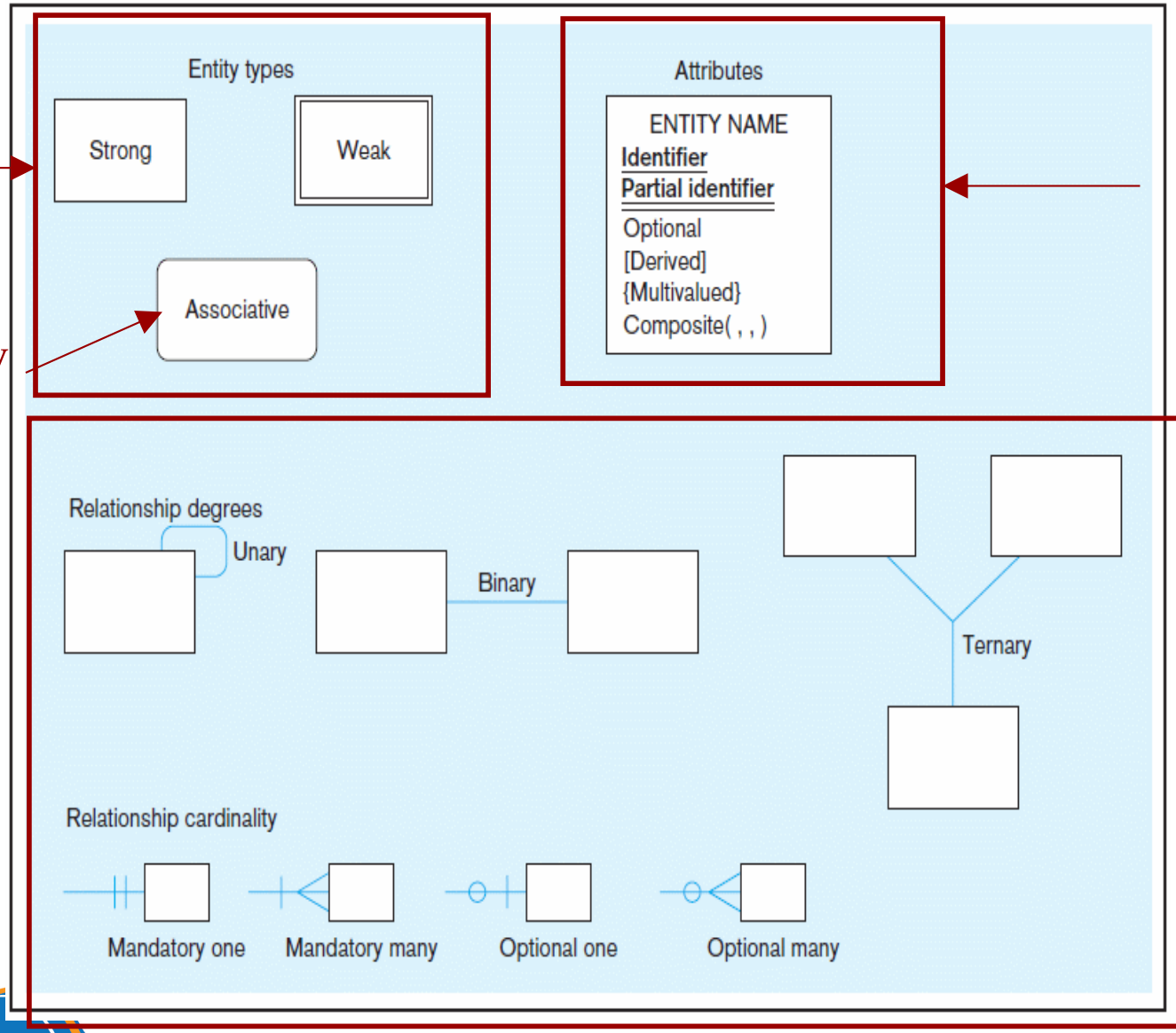
(Coronel and Morris, 2015) (Hoffer et al., 2019)

Basic ER Notations

Entity symbols

A special entity that is also a relationship

Relationship degrees specify number of entity types involved



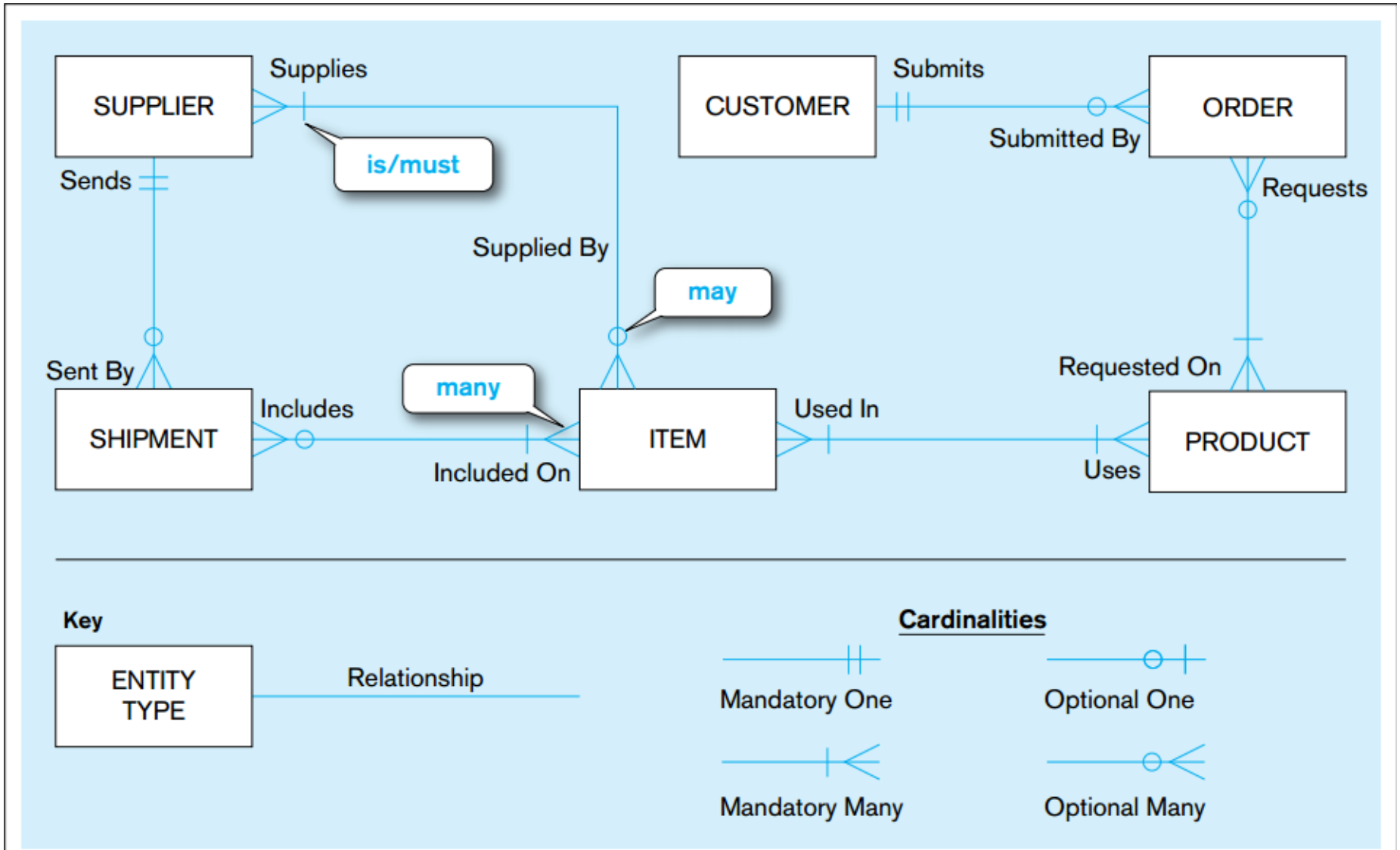
Attribute symbols

Relationship symbols

Relationship cardinalities specify how many of each entity type is allowed

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Sample ERD (ER Diagram)



(Coronel and Morris, 2015) (Hoffer et al., 2019)

Sample ERD (ER Diagram)

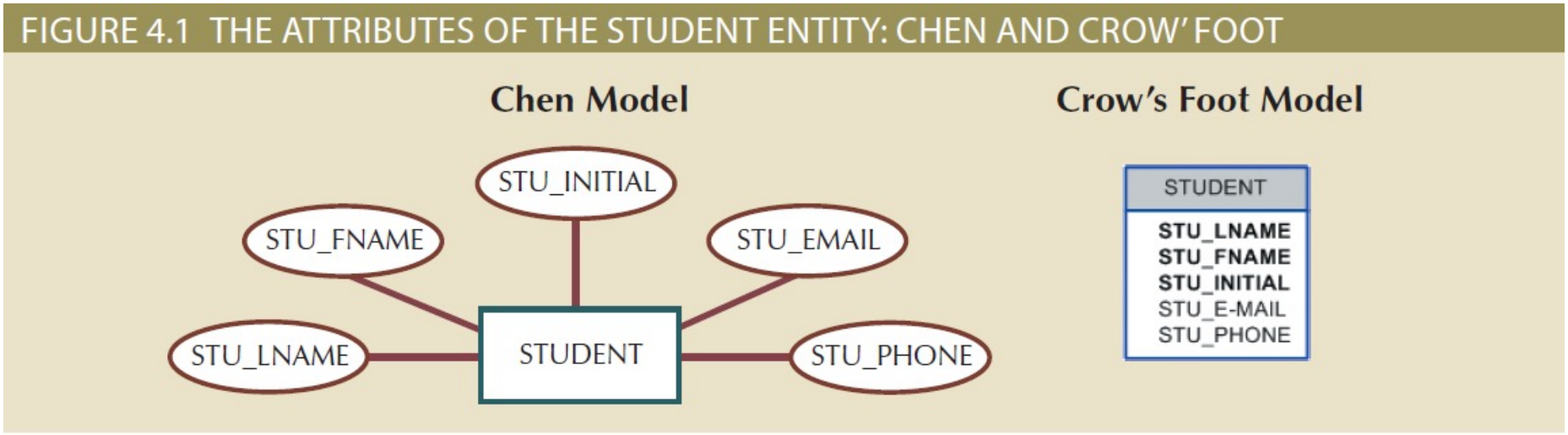
Business rules enable the formation of the preceding diagram

- A **SUPPLIER** may supply many **ITEMs**. Each **ITEM** is supplied by any number of **SUPPLIERs**.
- Each ITEM must be used in the assembly of at least one PRODUCT and may be used in many products.
- A SUPPLIER may send many SHIPMENTs. However, each shipment must be sent by exactly one SUPPLIER.
- A SUPPLIER may be able to supply an item but may not yet have sent any shipments of that item.
- A SHIPMENT must include one (or more) ITEMs. An ITEM may be included on several SHIPMENTs.
- A CUSTOMER may submit any number of ORDERs. However, each ORDER must be submitted by exactly one CUSTOMER.
- An ORDER must request one (or more) PRODUCTs. A given PRODUCT may not be requested on any ORDER or may be requested on one or more orders.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Entities

- Physical object/concept of interest to the end user
- Indicate an entity set, not a single entity occurrence
- Example: Student, Store, Building, Sale, Account



(Coronel and Morris, 2015) (Hoffer et al., 2019)

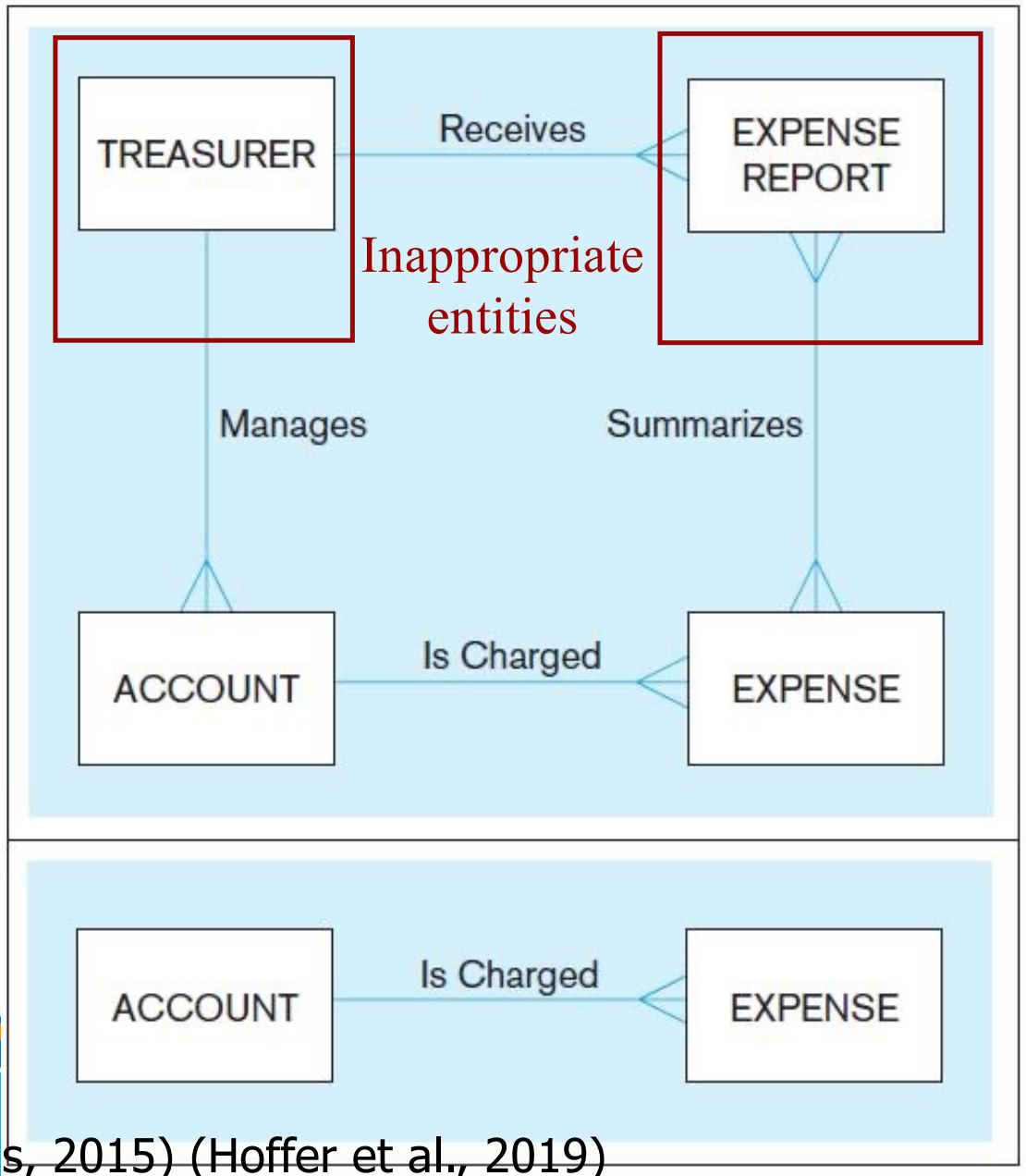
cdio 4.0
fit@hcmus

What Should an Entity Be?

- An entity SHOULD BE an **object** or **concept** that:
  - Have many instances in the database
  - Be characterized by various attributes/properties
  - The organization wishes to maintain data
- An entity SHOULD NOT BE:
  - A user of the database system
  - An output of the database system (e.g., a report)
  - A single-instance object

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Example of Inappropriate Entities



(Coronel and Morris, 2015) (Hoffer et al., 2019)

Strong vs. Weak Entities, and Identifying Relationships

- **Strong** (or Independent) **Entity**
  - Exists independently of other entities
  - Has its own unique identifier (underlined with single line)
- **Weak** (or Dependent) **Entity**
  - Exists in dependence on other entities
    - Has no business meaning without the entity on which it depends ((Identifying) **Owner**).
  - Has only a partial identifier (underlined with double lines), which is combined with its owner's identifier to form a full identifier.
- **Identifying Relationship**
  - Links between strong and weak entities

(Coronel and Morris, 2015) (Hoffer et al., 2019)

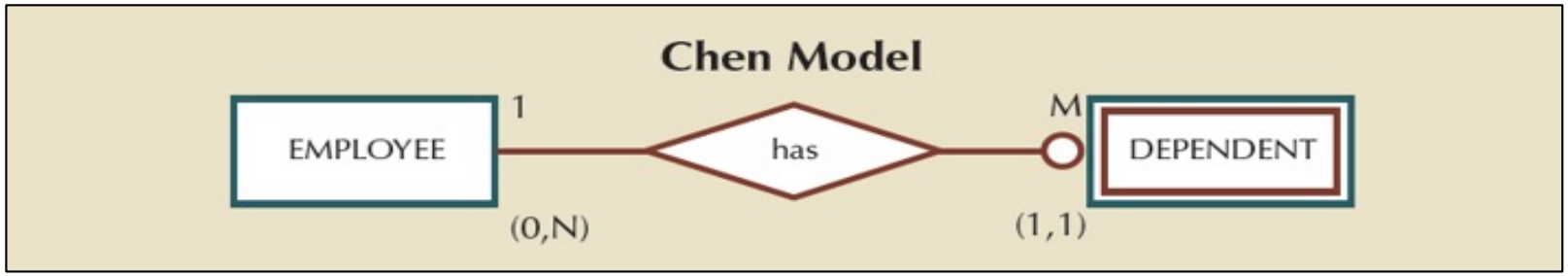
Existence dependence vs. independence

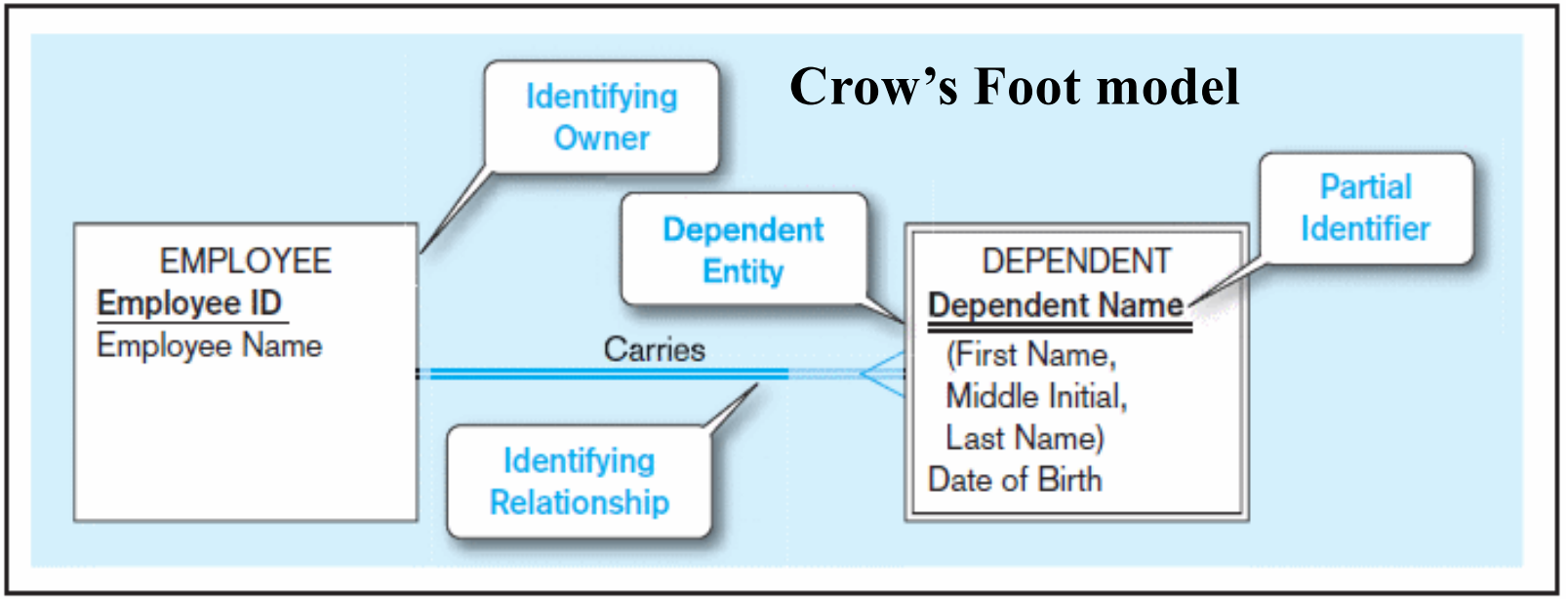
- Existence dependence
  - Entity exists in the database only when it is associated with another related entity occurrence
- Existence independence
  - Entity exists apart from all of its related entities
  - Referred to as a strong entity or regular entity

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Example of Weak Identity and Identifying Relationship

Strong entity **Weak entity**





(Coronel and Morris, 2015) (Hoffer et al., 2019)

cdio 4.0

fit@hcmus

Entity Naming Conventions

- An entity name should be:
  - A singular noun or noun phrase
  - Specific to the organization (e.g., Customer or Client)
  - Clear and descriptive
  - Familiar to the users
  - Concise (as few words as possible)
  - Consistent
- An abbreviation (short name) could be used?
  - Possible if it adheres to rules and sufficiently specifies an entity name.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Guidelines for Defining Entities

- A definition statement for an entity (type) should:
  - Start with "*An X is …*"
  - Include an identifier declaration *"…is identified by…"*
  - Include statements of:
    - What instances are included/not included?
    - When an instance is created, deleted or changed?
    - What history is to be kept about instances?

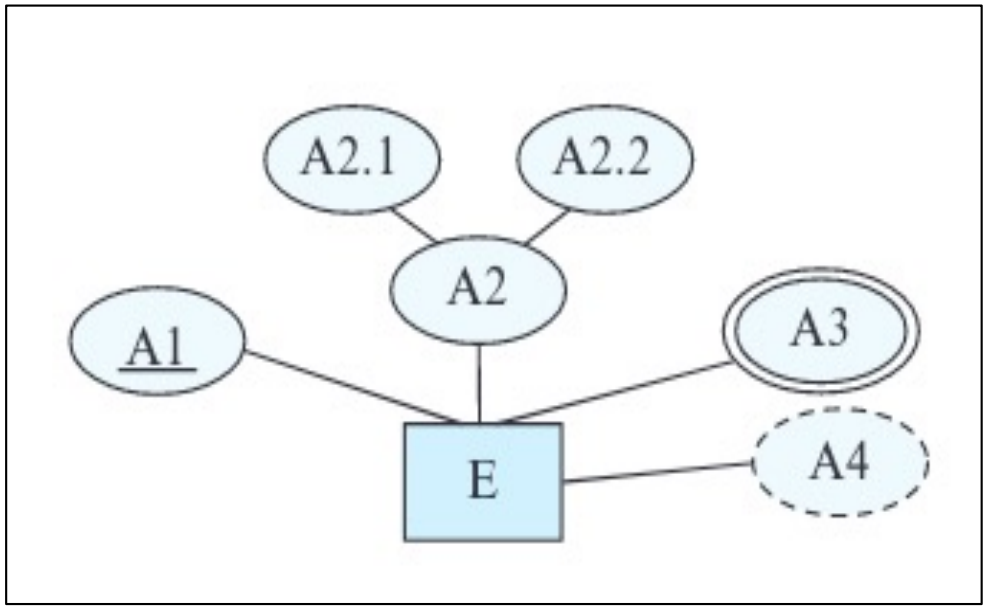
Example (cf. Hoffer et al, 2019):
A bid is a legal offer by our organization to do work for a customer. A bid is created when an officer of our company signs the bid document; a bid becomes an instance of contract when we receive a copy of the bid signed by an officer of the customer.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

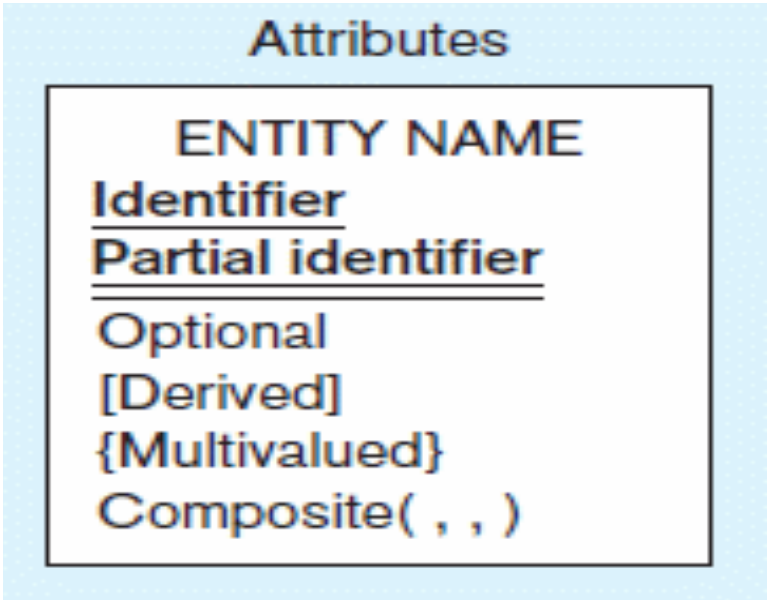
Attributes

- Property/Characteristic of an entity (or relationship) type
- **Notation**:

Chen model



Crow's Foot model

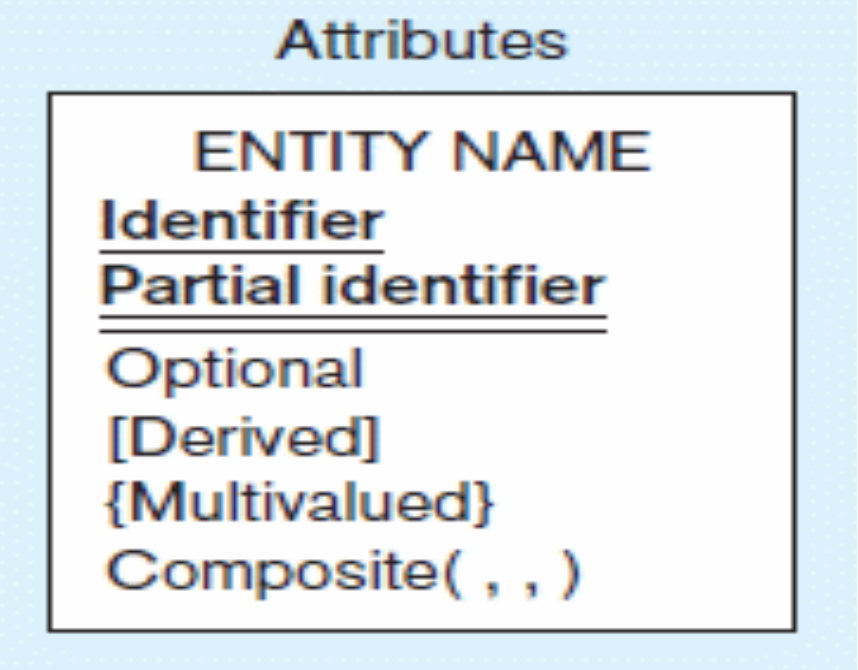


(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Classifications of Attributes

- **Required** versus **Optional** Attributes
- **Simple** versus **Composite** Attributes **(A1, …)**
- **Single-Valued** versus **Multivalued** Attributes **{A}**
- **Stored** versus **Derived** Attributes **[A]**
- **Identifier Attributes**



(Coronel and Morris, 2015) (Hoffer et al., 2019)

Required vs. Optional Attributes

- **Required attribute:** Must have a value for every entity/relationship instance with which it is associated.
- **Optional attribute**: May not have a value for every entity/relationship instance with which it is associated.

Entity type: STUDENT

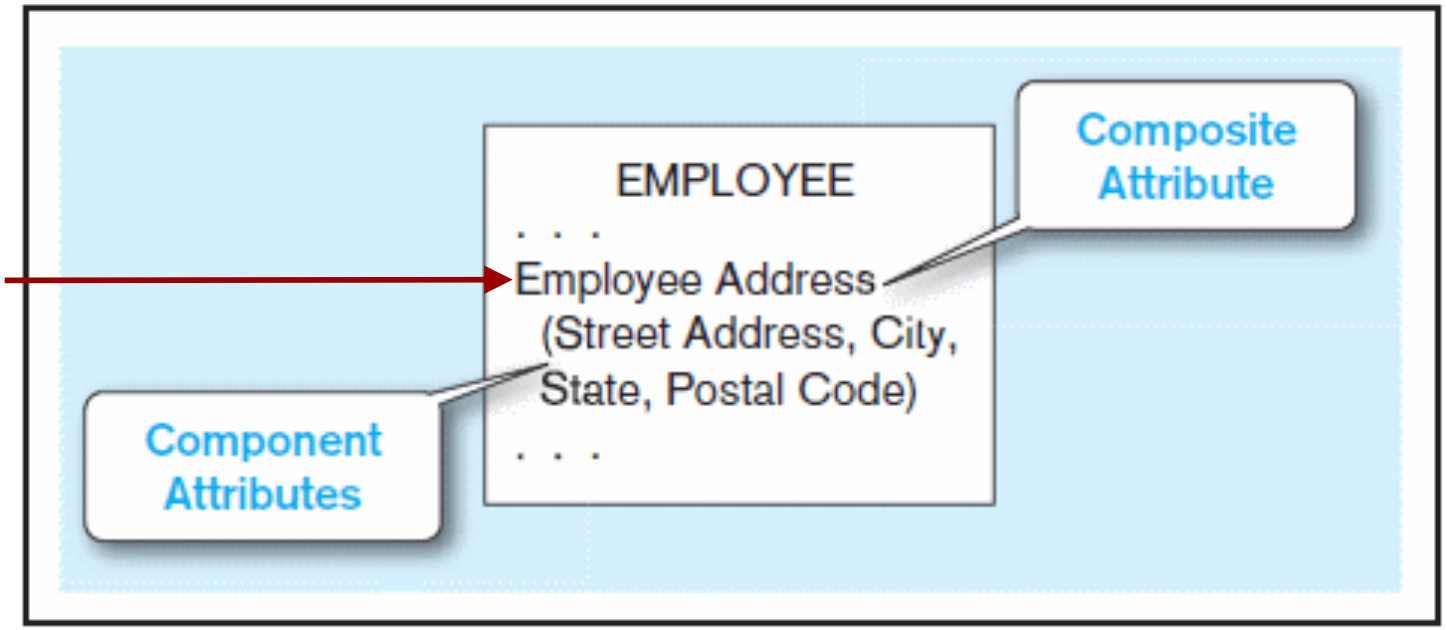
Attributes	Attribute Data Type	Required or Optional	Example Instance	Example Instance
Student ID	CHAR (10)	Required	876-24-8217	822-24-4456
Student Name	CHAR (40)	Required	Michael Grant	Melissa Kraft
Home Address	CHAR (30)	Required	314 Baker St.	1422 Heft Ave
Home City	CHAR (20)	Required	Centerville	Miami
Home State	CHAR (2)	Required	OH	FL
Home Zip Code	CHAR (9)	Required	45459	33321
Major	CHAR (3)	Optional	MIS	

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Simple vs. Composite Attributes

- **Composite attribute**: Be subdivided into meaningful component parts (also named **component attributes**).
- **Simple** (or atomic) **attribute**: Cannot be subdivided.
- **Notation**: **Composite Attribute (Component 1, …)**

The address is broken into component parts



(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Single-valued vs. Multivalued Attributes

- **Single-valued attribute:** Only have one value for a given instance (frequent occurrences in most cases)
- **Multivalued attributes**: May have many values for a given instance
- **Notation**: **{ Multivalued Attribute }**
- Multivalued vs. Composite attributes?

(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Stored vs. Derived Attributes

- **Stored attribute:** Be stored physically in the database
- **Derived attribute**: Be calculated from other stored attributes and may not be stored physically
- **Notation**: **[Derived Attribute]**
- Question:

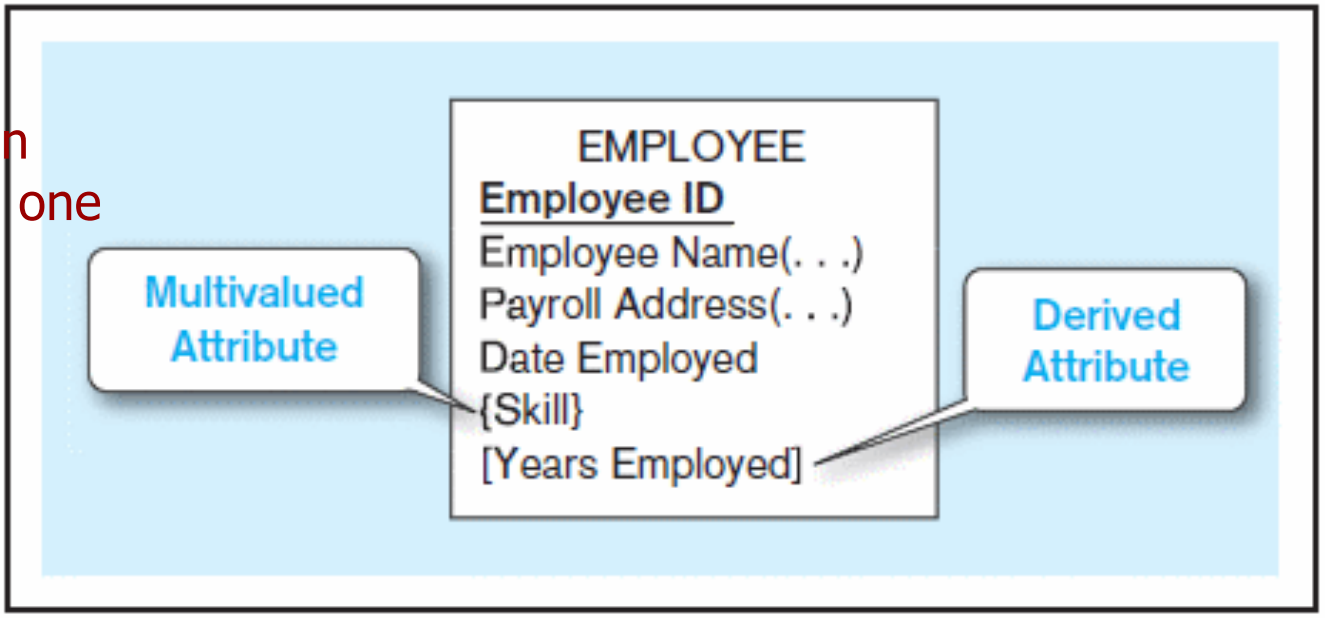
  Derived attribute should be stored or calculated?

(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Example of Multi-valued and Derived Attributes

Multivalued
an employee can have more than one skill



Derived
Calculated from date employed and current date

Entity with **multivalued** attribute (Skill) and **derived** attribute (Years Employed)

(Coronel and Morris, 2015) (Hoffer et al., 2019)

53

cdio 4.0
fit@hcmus

Derived Attribute Storage

Advantages and Disadvantages of Storing Derived Attributes		
	Derived Attribute: Stored	**Derived Attribute: Not Stored**
Pros	• CPU processing saving • Data access time saving • Data value is readily available • Keep track of historical data	• Storage space saving • Computation/Calculation always yields current values
Cons	• Requires constant maintenance to ensure derived value is current, especially if any values used in the calculation change	• CPU processing cycles increasing • Data access time increasing • Requires complex queries

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Identifiers (Keys)

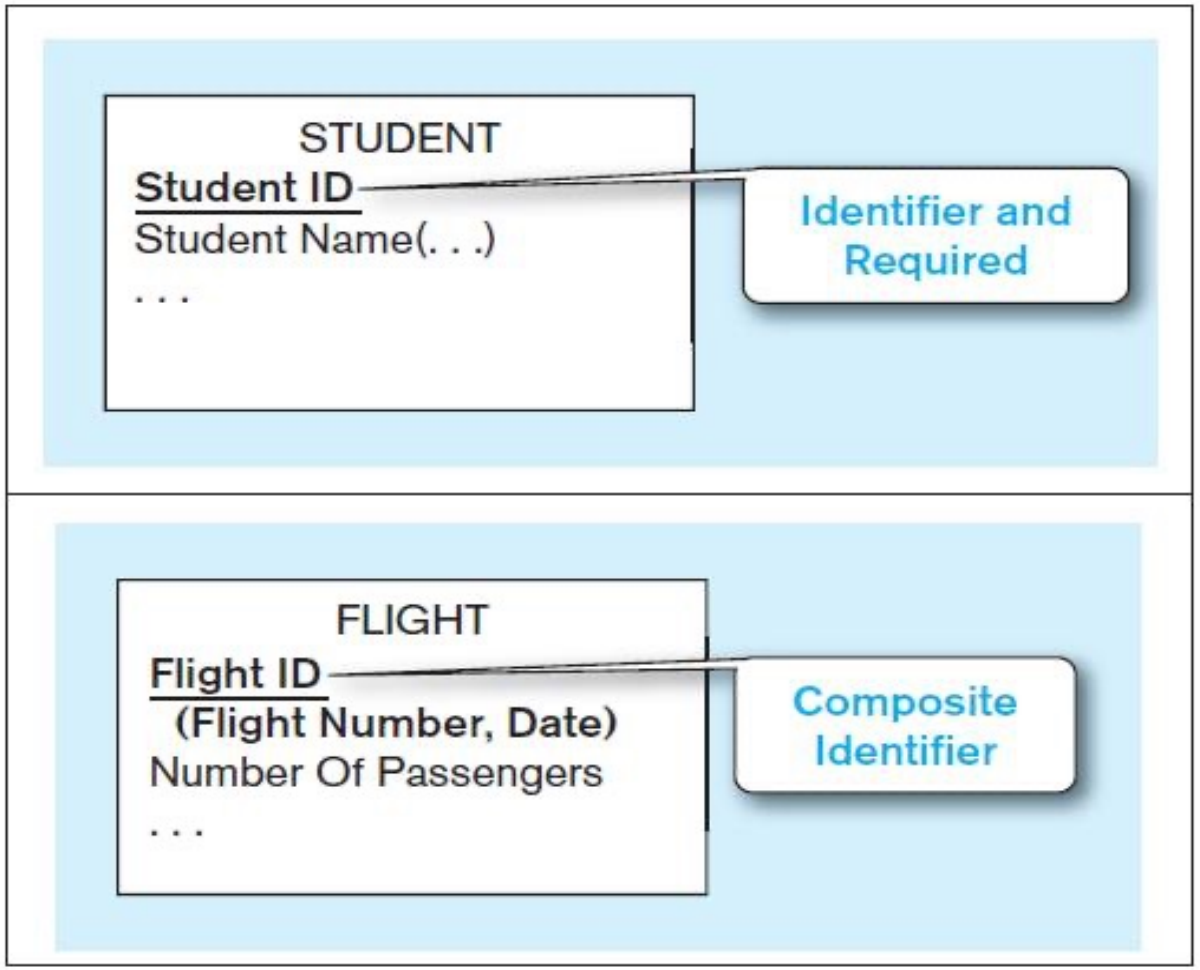
- **Identifier** (Key): An attribute (or a combination of attributes) that uniquely identifies individual instances of an entity type.

- **Candidate Identifier:** An attribute (or a combination of attributes) that could be a key (i.e., satisfies the requirements for being an identifier)

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Identifiers (Keys)

- **Simple** versus **Composite Identifier**



(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Criteria for Selecting Identifiers

- The attributes chosen as Identifiers should:
  - Not change over the life of entity instances
  - Be guaranteed to have valid values and not be null
  - Not include classifications or locations that might change over time (so-called intelligent identifiers)

- Consider substituting single-attribute surrogate identifiers for large composite identifiers.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Attribute Naming Conventions

- An attribute name should be:
  - A singular noun or noun phrase
  - Unique for clarity purposes ➔ Follow a standard format

[Entity type name { [Qualifier] }] Class

[…] an optional clause

{…} a repeatable

  - Defined using the same qualifiers and classes with other similar attributes of different entity types
  - **Example**: Emp ID, Emp Birth Date, Emp Hired Date

(Coronel and Morris, 2015) (Hoffer et al., 2019)

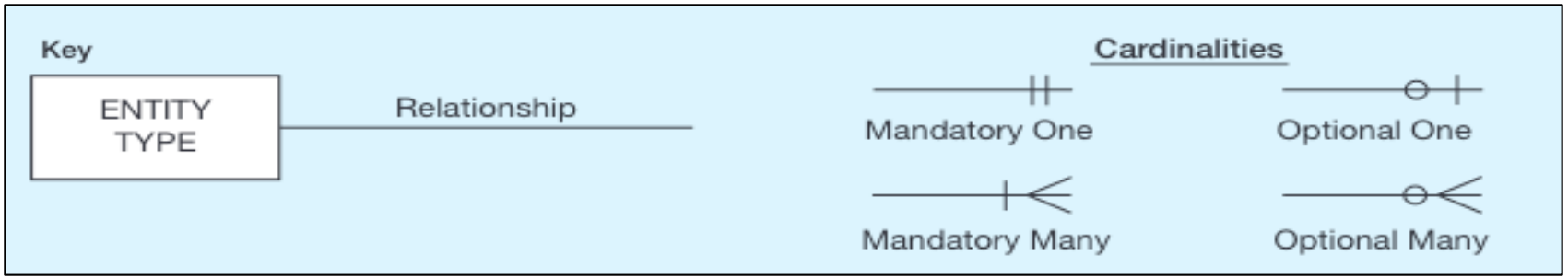
Guidelines for Defining Attributes

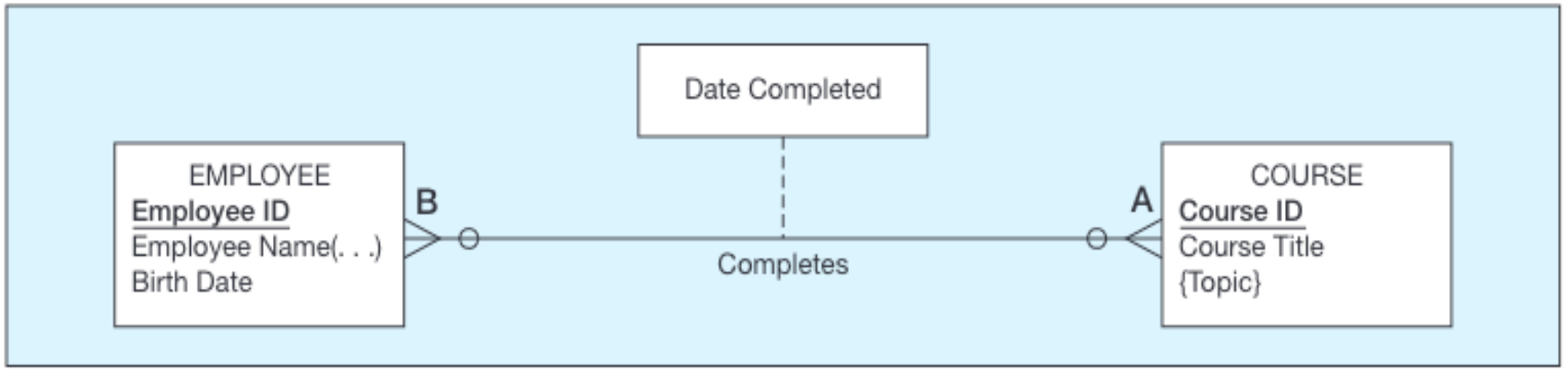
- A definition statement for an attribute should:
  - State what the attribute is and possibly why it is important
  - Indicate what is and is not included in the attribute's value
  - Include aliases in documentation
  - State source of values
  - Specify required vs. optional attributes
  - Indicate the min and max values for multivalued attributes
  - Indicate relationships with other attributes

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Relationships

- Meaningful association among entity types.
- Indicate a relationship type/set, not a single instance
- Characterized by degree, cardinality, and attributes



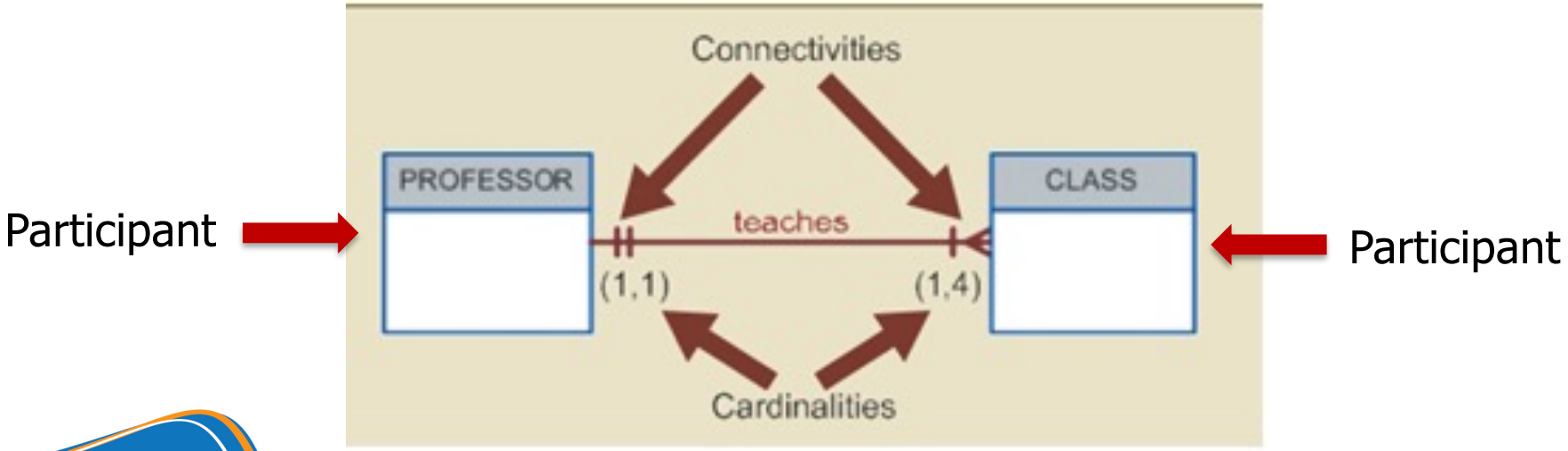


(Coronel and Morris, 2015) (Hoffer et al., 2019)

Relationships

Components

- **Participants:** Entities that participate in a relationship
- **Connectivity**: Describes relationship classifications
  - 1:1, 1:M, and M:N
- **Cardinality**: Expresses the minimum and maximum instances of an entity (type) associated with one instance of related entity. ***Notation***: Placing the appropriate numbers beside the entities, using the format (x, y).

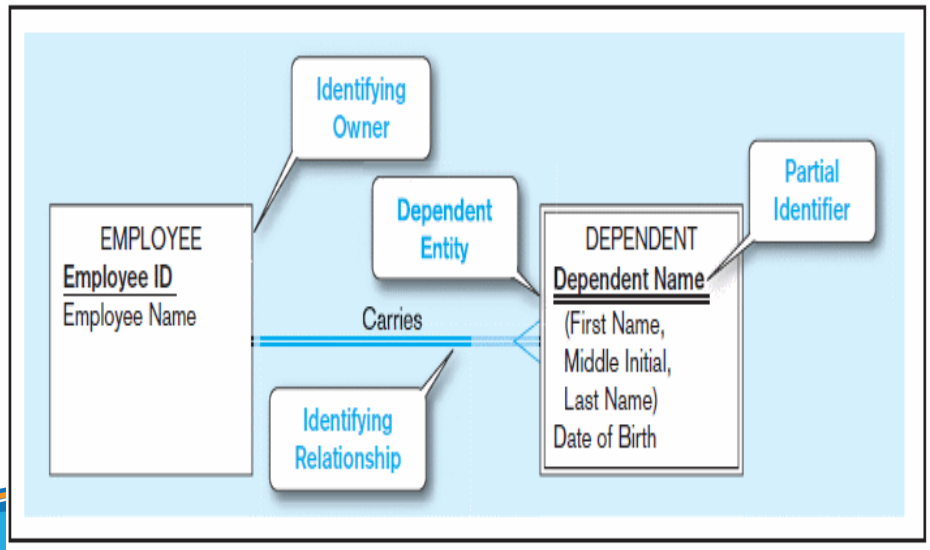


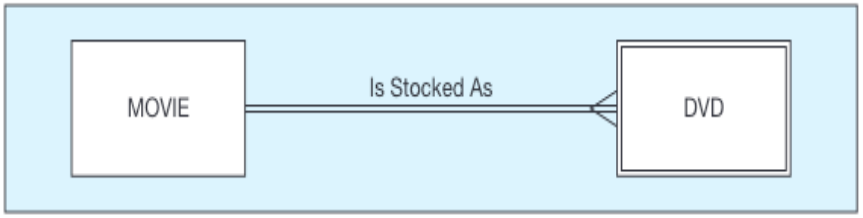
(Coronel and Morris, 2015) (Hoffer et al., 2019)

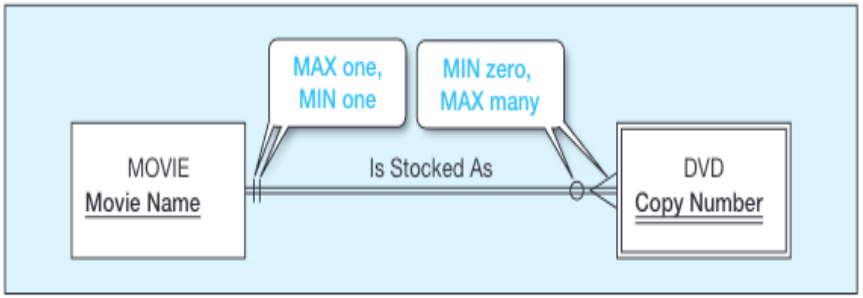
Relationship Strength

- **Weak** (non-identifying) **Relationship**
  - Primary key of the related entity does not contain a primary key component of the parent entity
- **Strong** (identifying) **Relationship**
  - Primary key of the related entity contains a primary key component of the parent entity

Example of identifying relationships



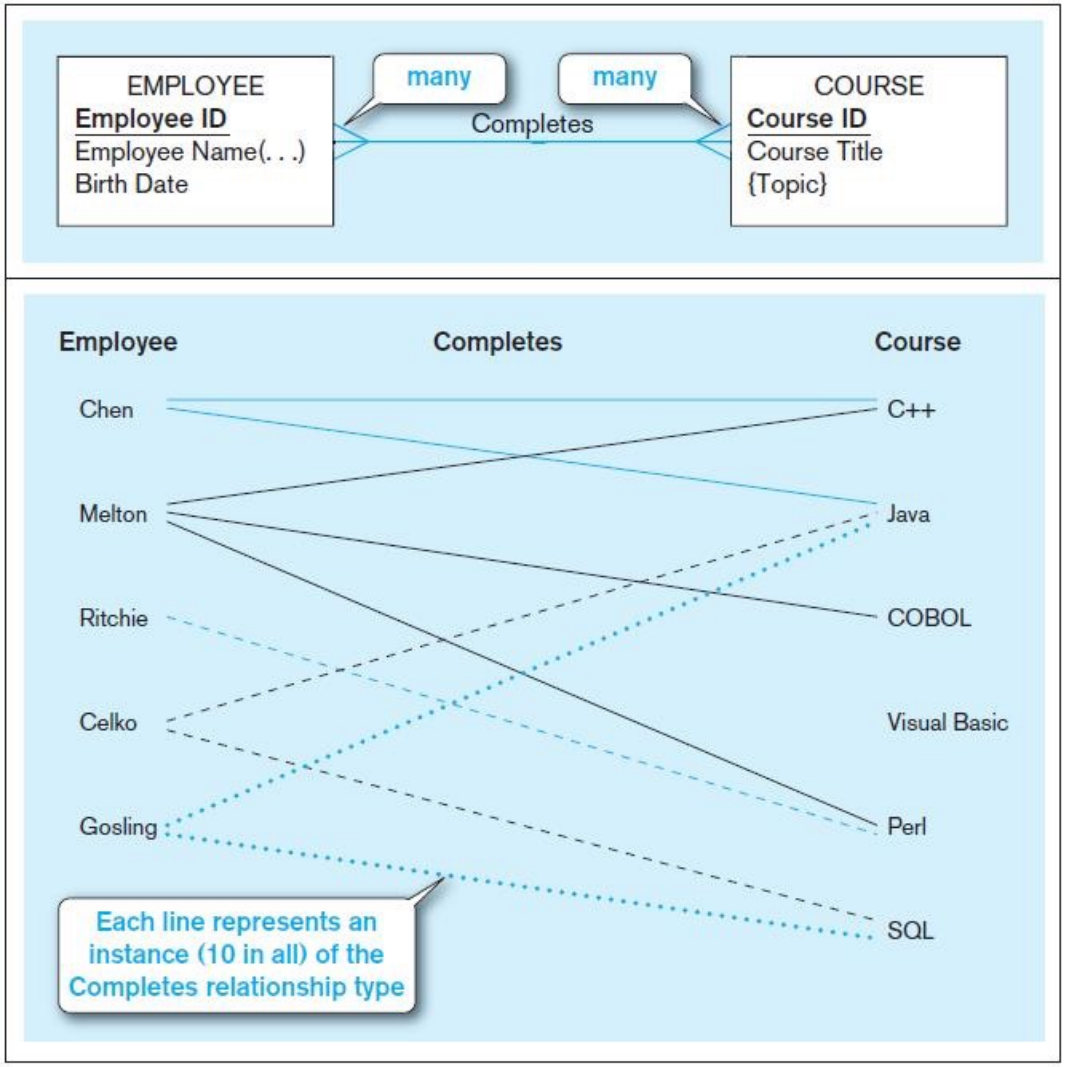




(Coronel and Morris, 2015) (Hoffer et al., 2019)

Modeling Relationships

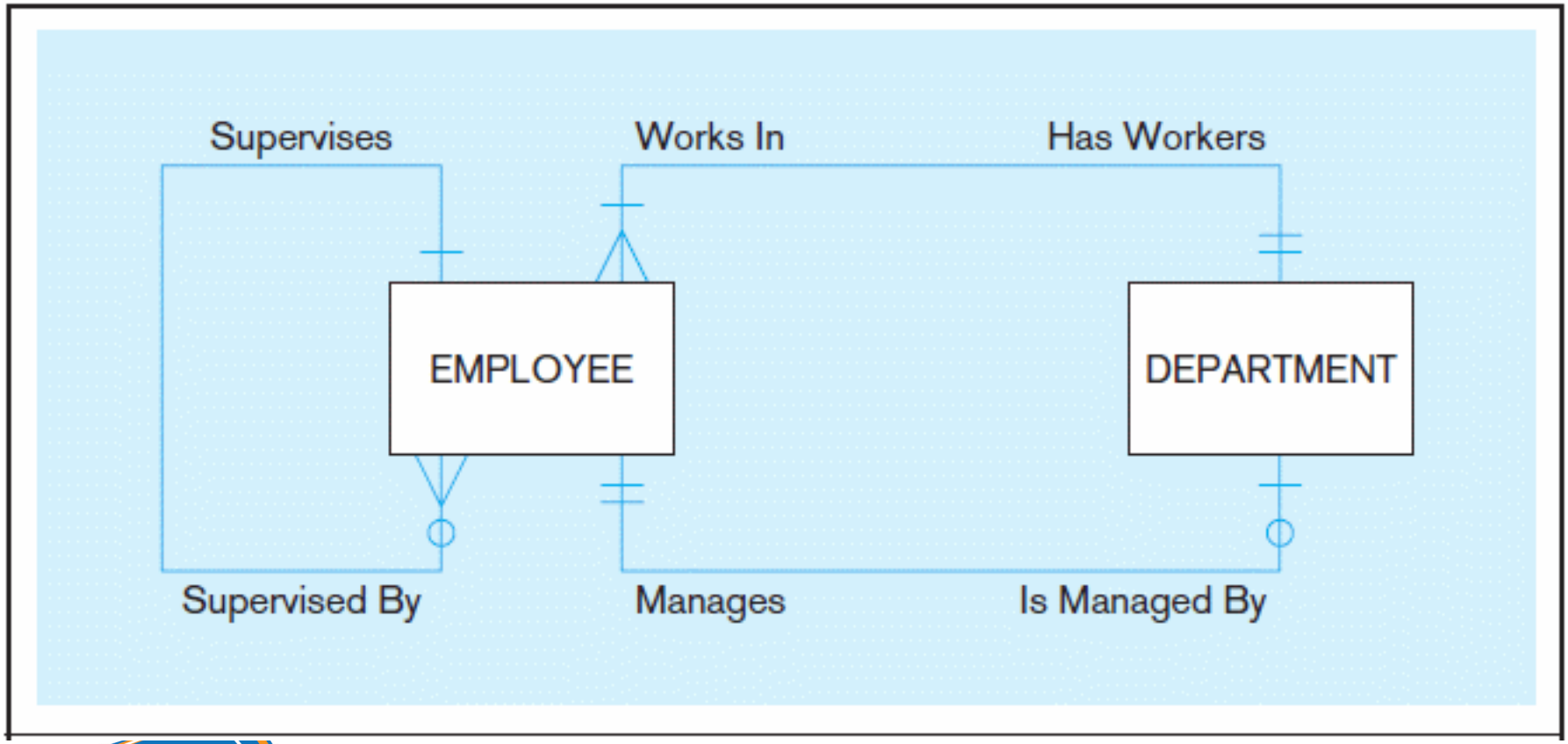
- A relationship type is modeled as a line between entity types (also named entities for short).
- A relationship instance is between entity instances



(Coronel and Morris, 2015) (Hoffer et al., 2019)

Modeling Relationships

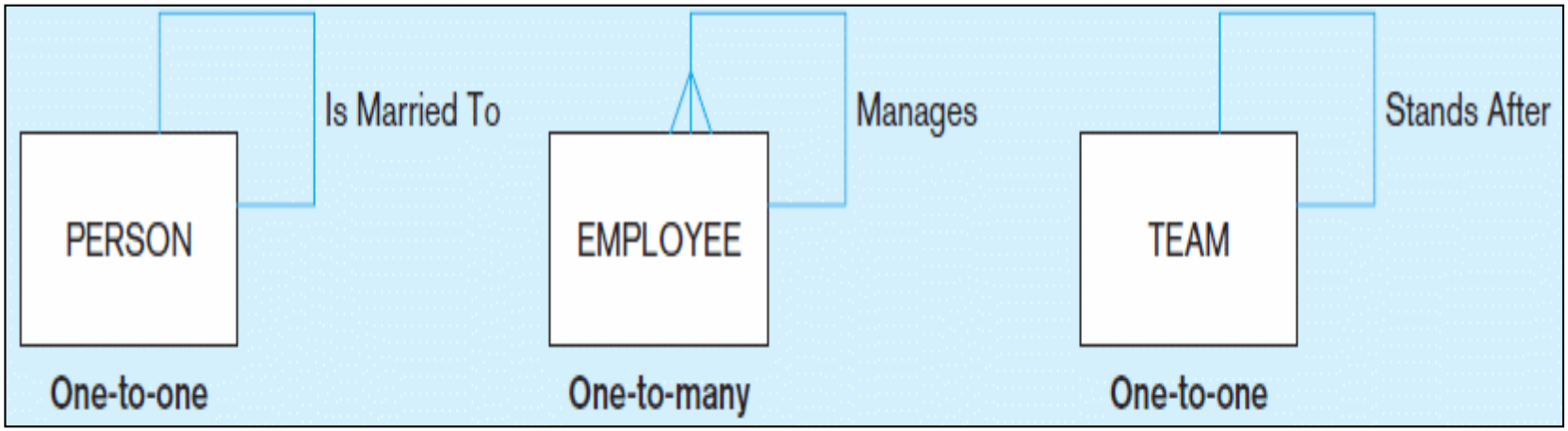
- Two entities can have more than one type of relationship between them (multiple relationships).



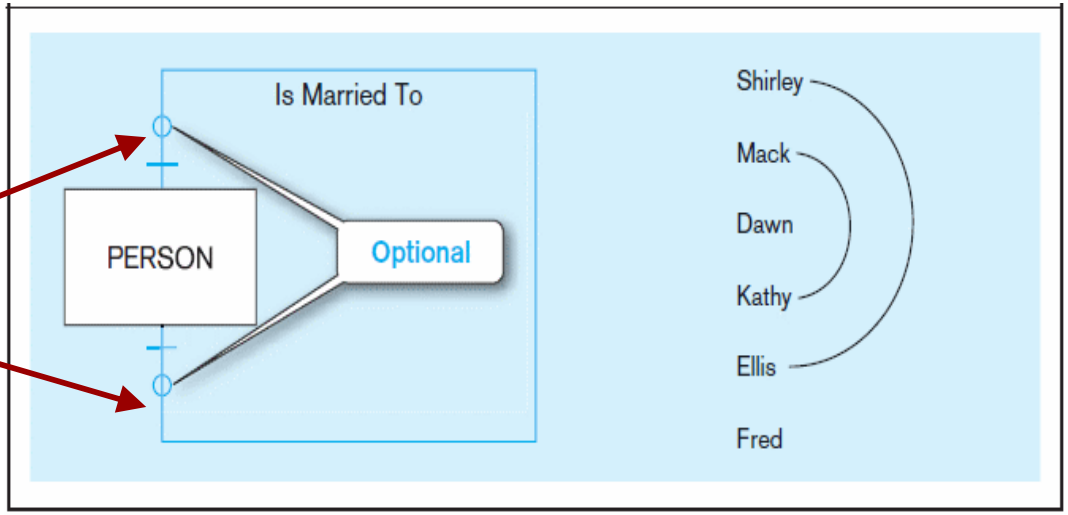
(Coronel and Morris, 2015) (Hoffer et al., 2019)

Modeling Relationships

- A relationship can exist between instances of the same entity set (recursive relationships).



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



(Coronel and Morris, 2015) (Hoffer et al., 2019)

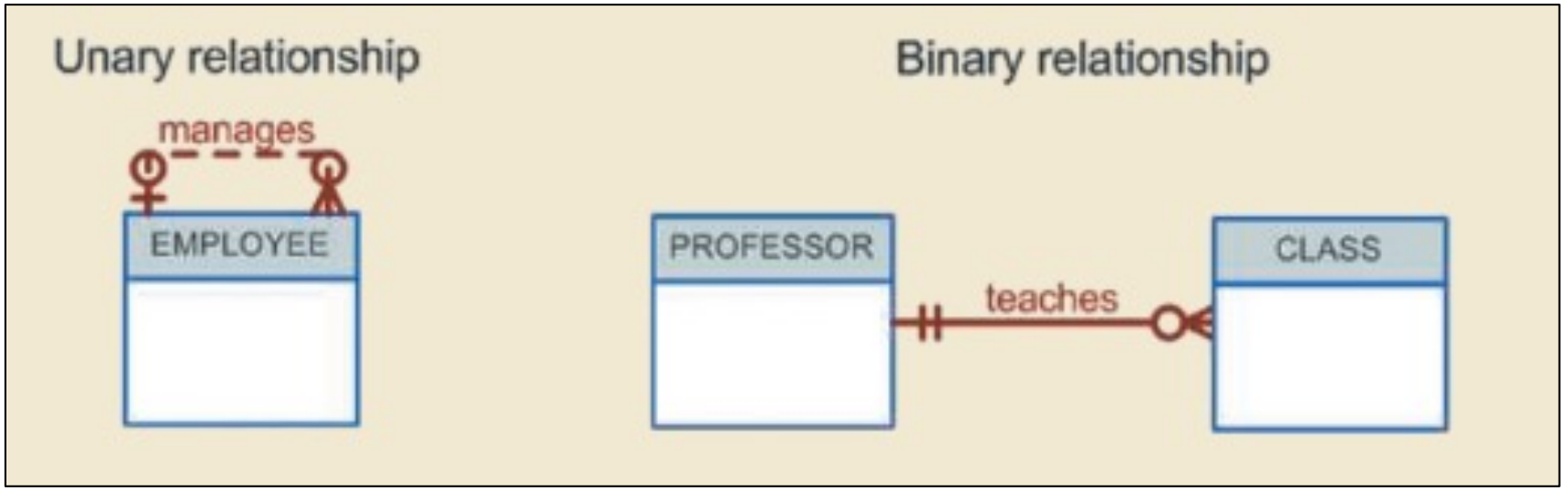
Modeling Relationships

- A relationship is characterized by:
  - Degree
  - Cardinality
  - Attributes (occasionally), which sometimes lead to the formation of an **association entity** (a combination of relationship and entity)

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Degree of Relationships

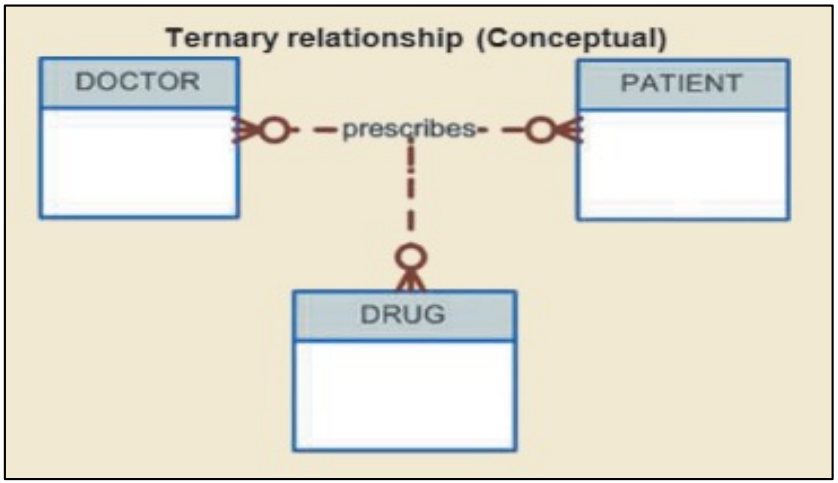
- Number of entities associated with a relationship.
  - Degree 1: Unary relationship (recursive relationship)
  - Degree 2: Binary relationship

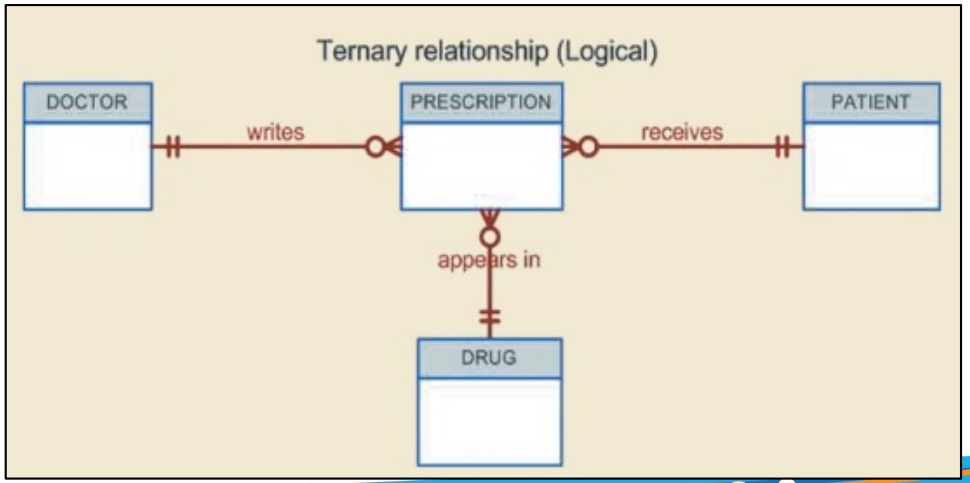


(Coronel and Morris, 2015) (Hoffer et al., 2019)

Degree of Relationships

- Number of entities associated with a relationship.
  - Degree 1: Unary relationship (recursive relationship)
  - Degree 2: Binary relationship
  - Degree 3: Ternary relationship





(Coronel and Morris, 2015) (Hoffer et al., 2019)

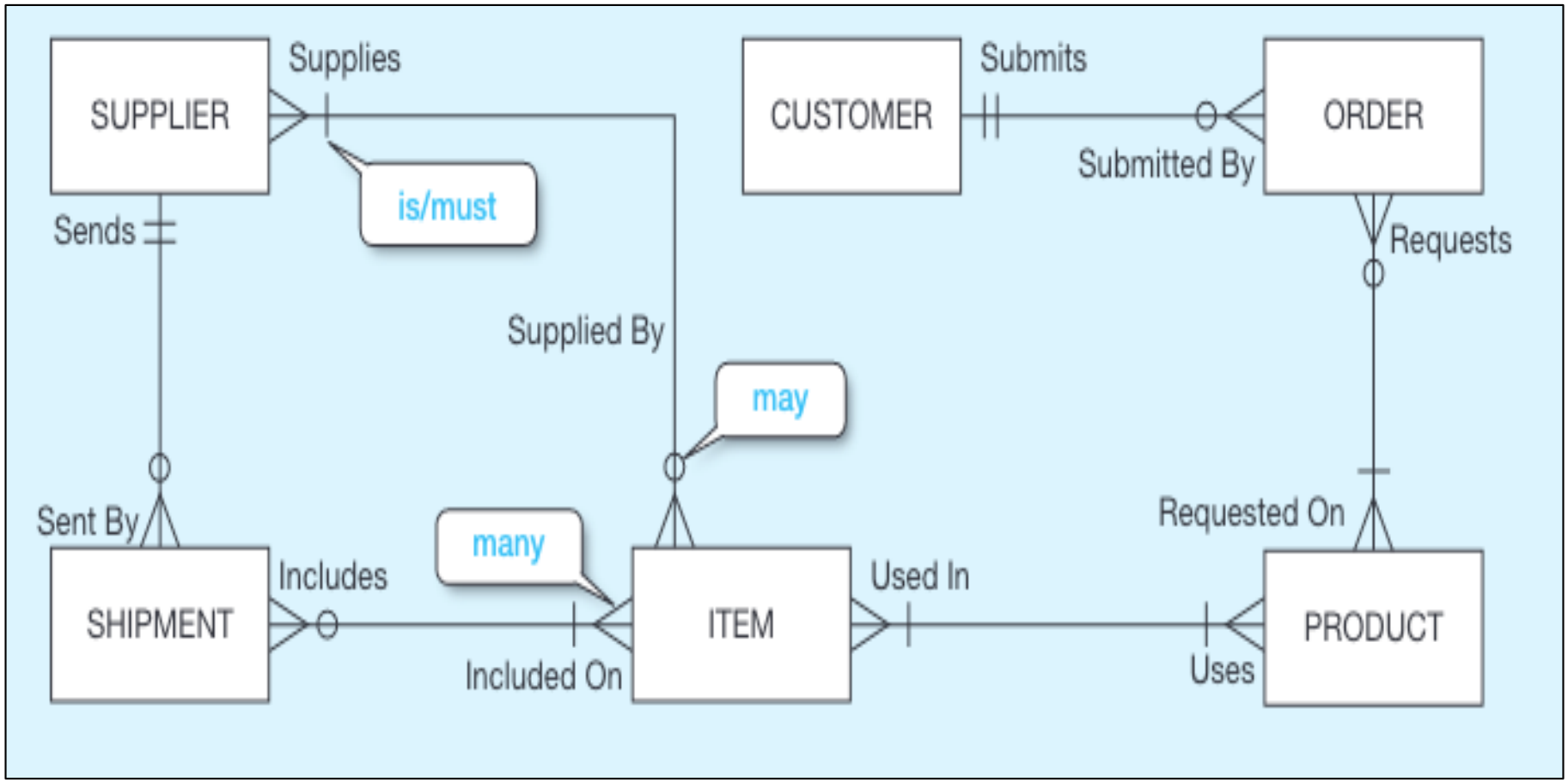
Cardinality Constraints

- The number of instances of one entity (type) that can or must be associated with each instance of another entity
  - **Minimum** Cardinality: The minimum number of instances participating in the relationship
  - **Maximum** Cardinality: The maximum number of instances participating in the relationship

CROW'S FOOT SYMBOLS		
SYMBOL	CARDINALITY	COMMENT
	(0,N)	Zero or many; the "many" side is optional.
	(1,N)	One or many; the "many" side is mandatory.
	(1,1)	One and only one; the "1" side is mandatory.
	(0,1)	Zero or one; the "1" side is optional.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Cardinality Constraints



(Coronel and Morris, 2015) (Hoffer et al., 2019)

Cardinality Constraints

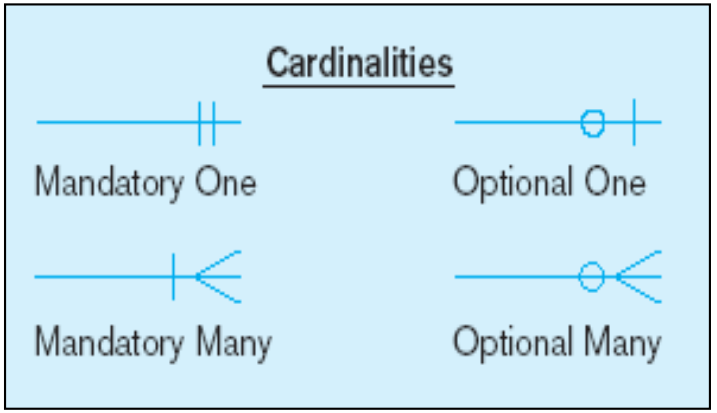
- Based on maximum cardinalities, three types of relationships are determined:
  - One-to-One (1-1): Each entity instance in the relationship will have exactly one related entity instance
  - One-to-Many (1-N): An entity instance on one side of the relationship can have many related entity instances, but the other will have a maximum of one related entity instance.
  - Many-to-Many (N-N): Entity instances on both sides of the relationship can have many related entity instances on the other side.

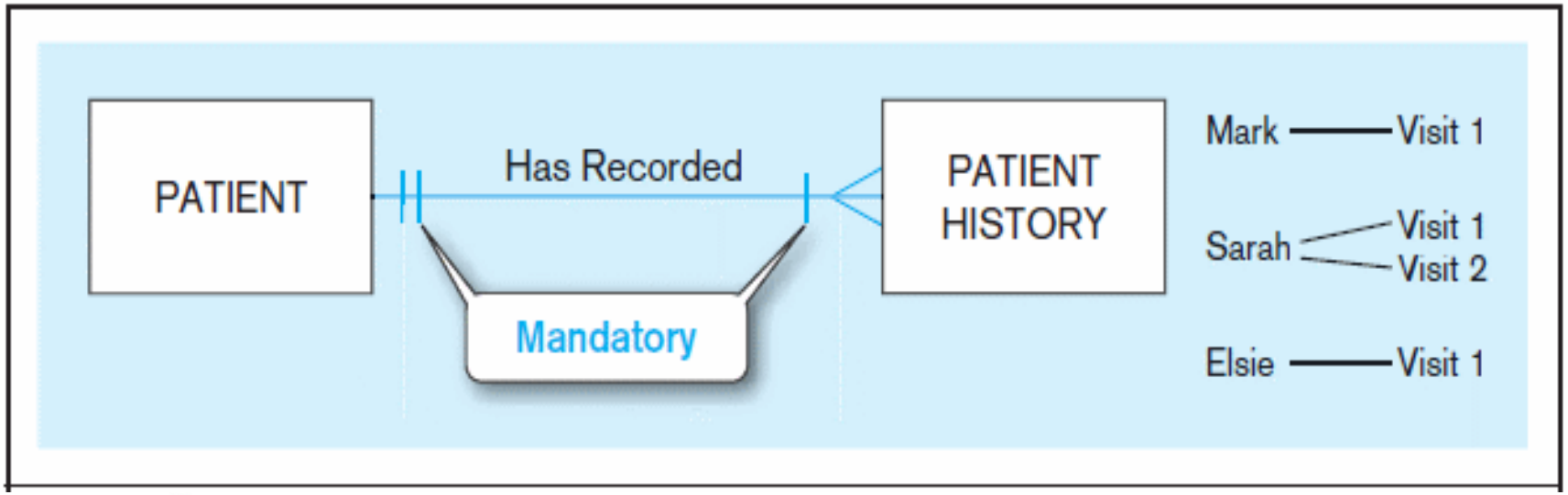
(Coronel and Morris, 2015) (Hoffer et al., 2019)

Cardinality Constraints

a) **Mandatory cardinalities**

- A patient must have recorded at least one history and can have many.
- A patient history is recorded for one and only one patient.



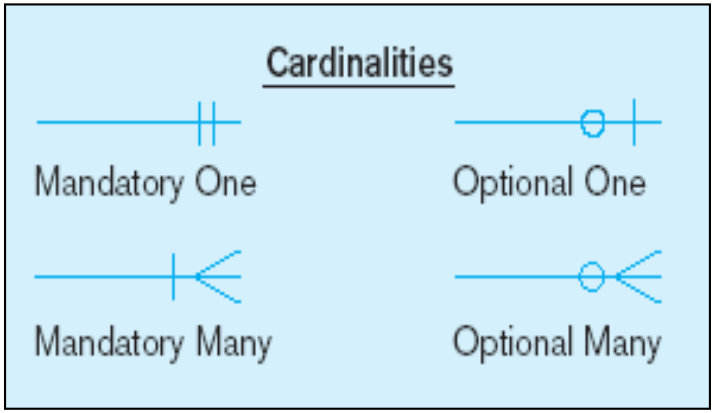


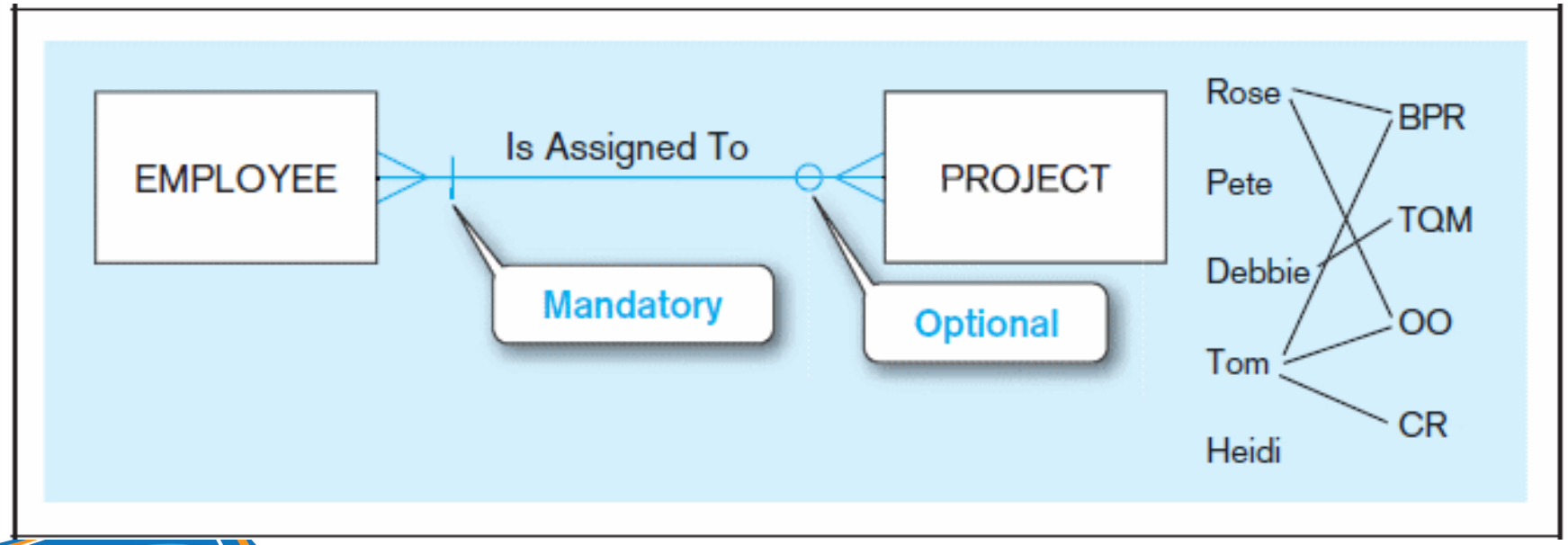
(Coronel and Morris, 2015) (Hoffer et al., 2019)

Cardinality Constraints

b) **One optional, one mandatory**

- An employee can be assigned to any number of projects or may not be assigned to any at all.
- A project must be assigned to at least one employee and may be assigned to many.



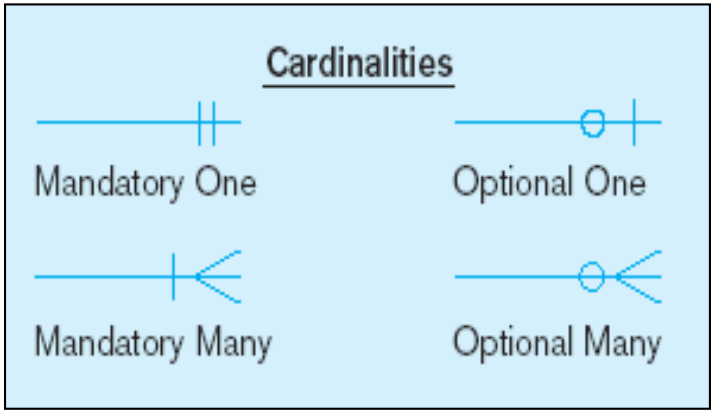


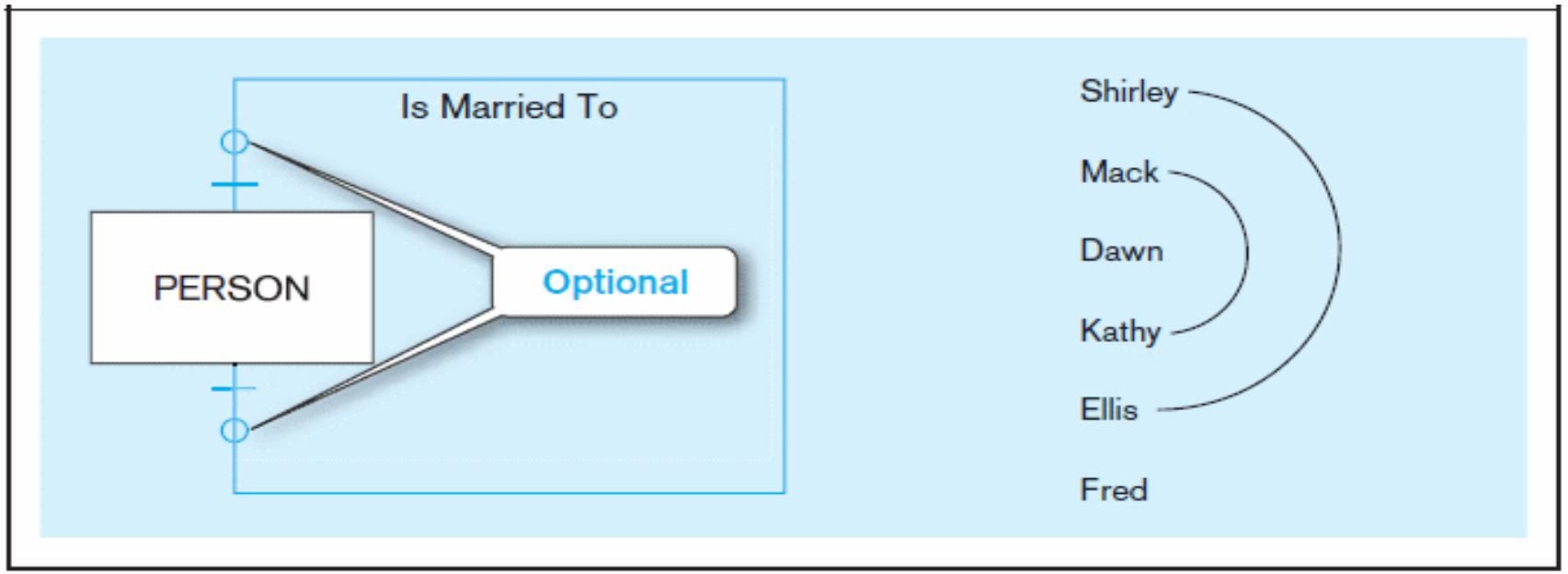
(Coronel and Morris, 2015) (Hoffer et al., 2019)

Cardinality Constraints

b) **Optional cardinalities**

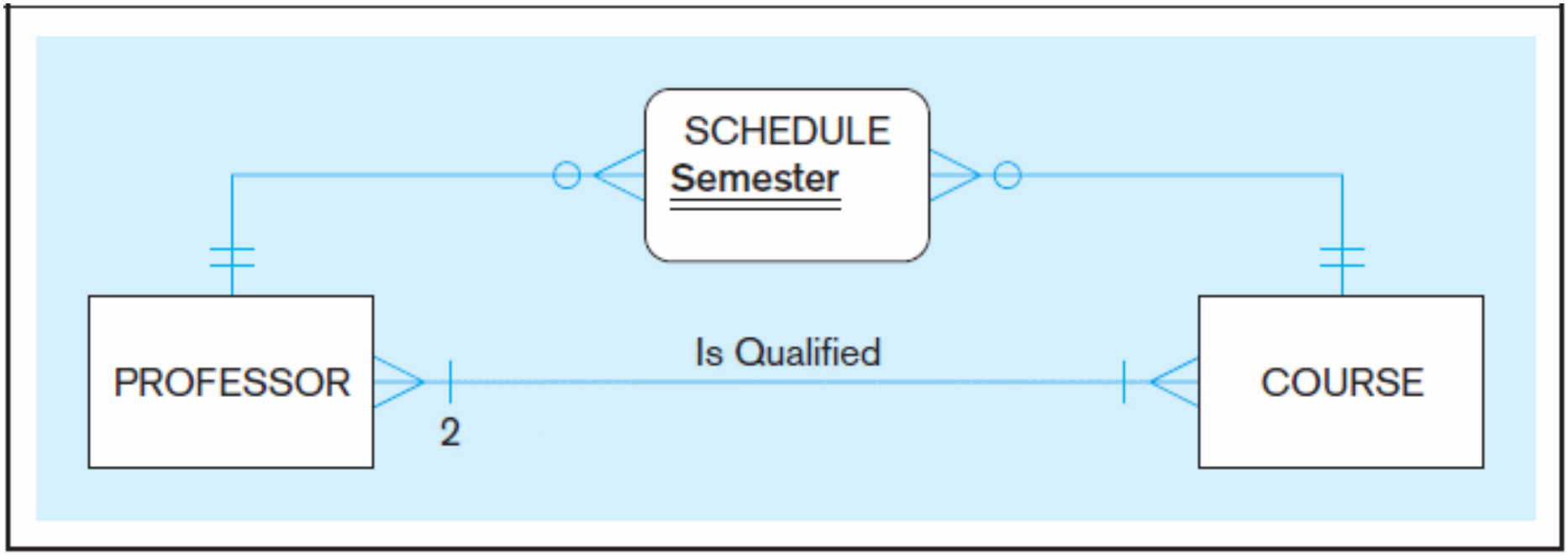
- A person is married to at most one other person or may not be married at all.





(Coronel and Morris, 2015) (Hoffer et al., 2019)

Examples of Fixed Limit Constraint



Here, min cardinality constraint is 2 (fixed lower limit constraint). At least two professors must be qualified to teach each course. Each professor must be qualified to teach at least one course.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

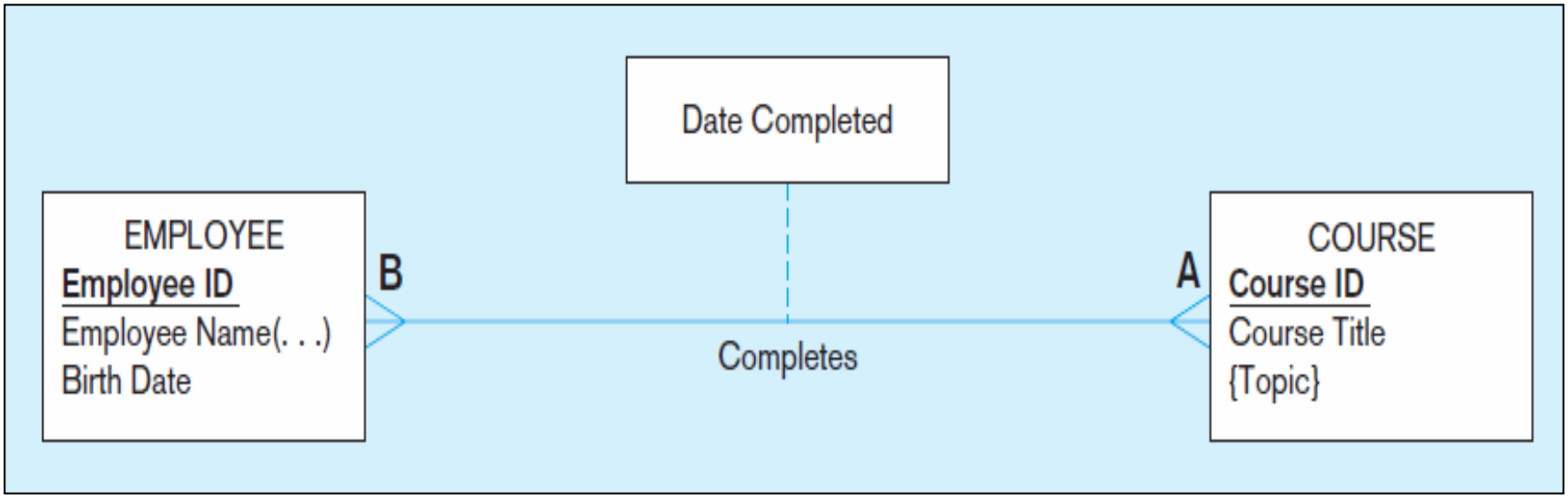
Associative Entities

- Combination of relationships and entities
  - Be an entity → Has attributes and identifiers (occasionally)
  - Be a relationship → Links entities together

- When associative entity?
  - All relationships for the associative entity should be **many**
  - The associative entity could have business 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

(Coronel and Morris, 2015) (Hoffer et al., 2019)

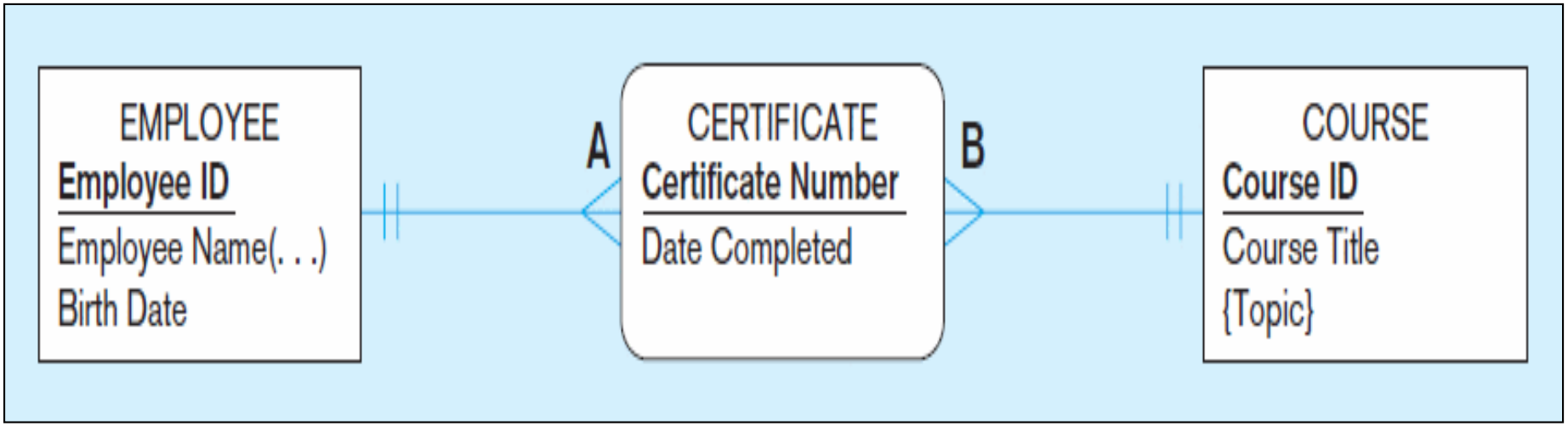
Associative Entities



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

(Coronel and Morris, 2015) (Hoffer et al., 2019)

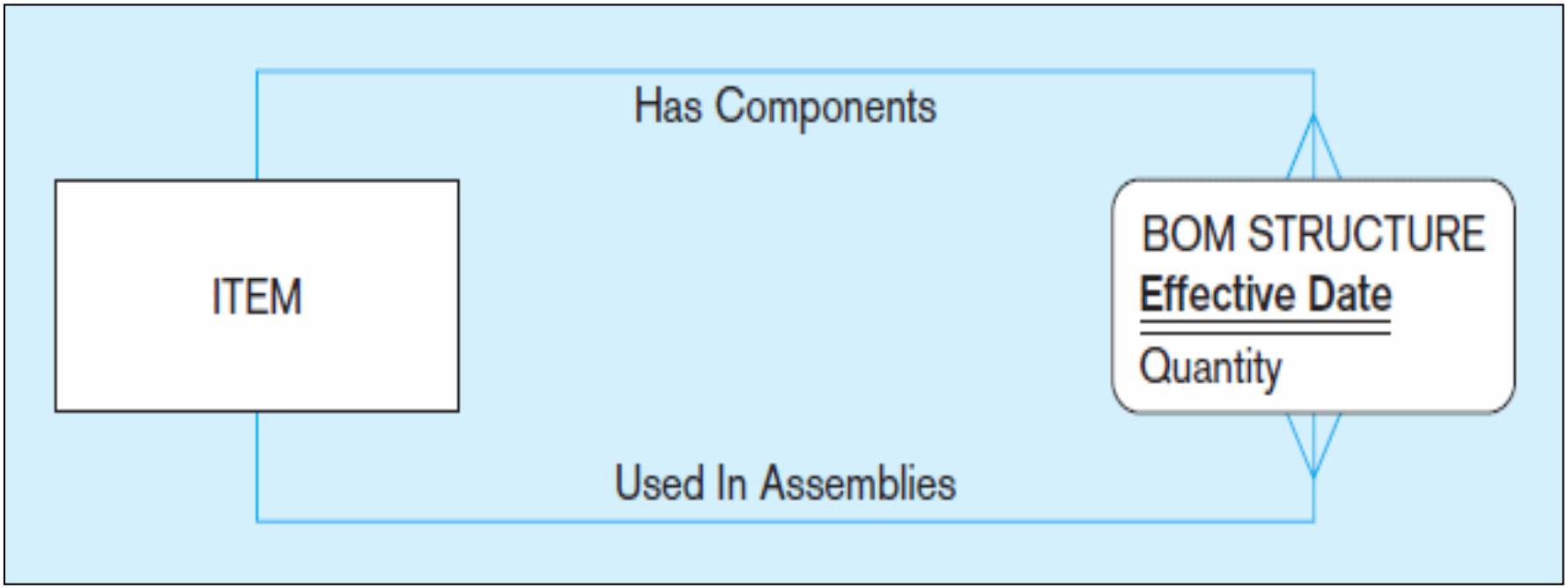
Associative Entities



- 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 has been replaced by two one-to-many relationships with the associative entity.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

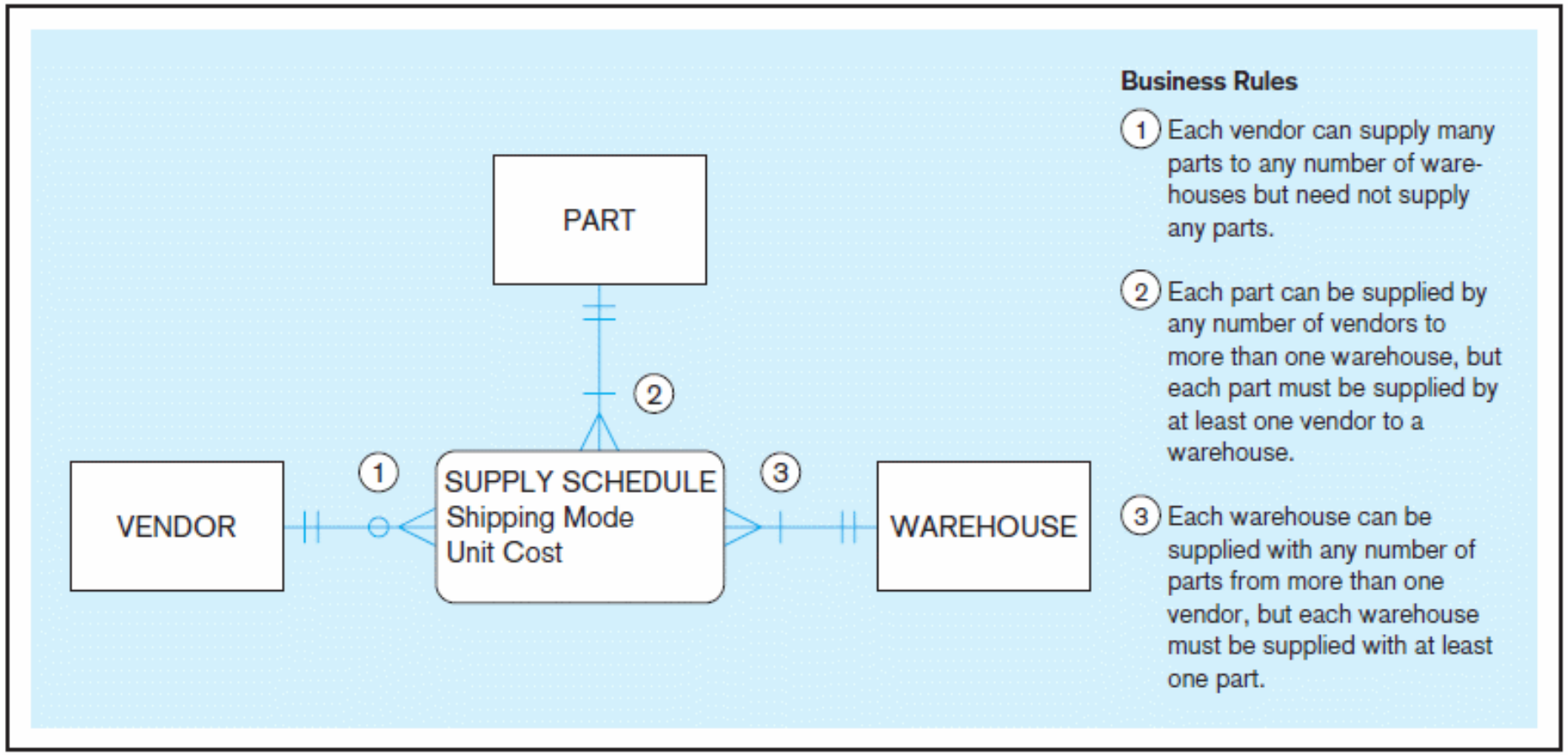
Associative Entities



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

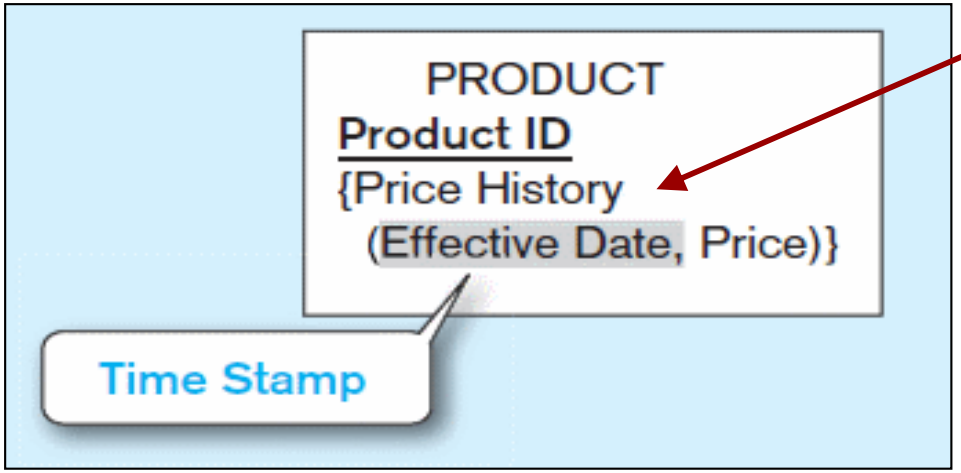
(Coronel and Morris, 2015) (Hoffer et al., 2019)

Ternary Relationship as Associative Entity



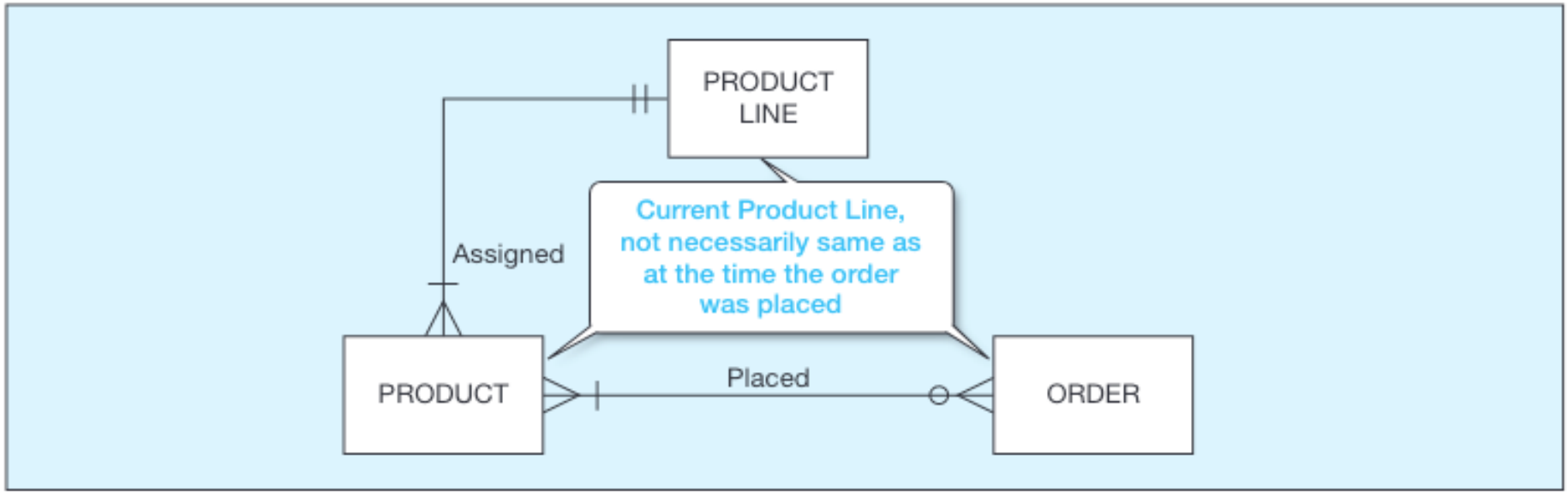
(Coronel and Morris, 2015) (Hoffer et al., 2019)

Time-stamping Value



The Price History attribute is both multivalued *and* composite.

Time-stamp value: Associated with a data value, often indicating when some event occurred that affected the data value.



(Coronel and Morris, 2015) (Hoffer et al., 2019)

Relationship Naming Conventions

- A relationship name should:
  - Be a verb or verb phrase
  - Represent business rules in a **meaningful** manner
  - Indicate the action taken (or interaction) rather than the action results or process
    - (e.g., use “AssignedTo”, not “Assignment”)
  - Avoid vague names (e.g., Has, Is related to, …)

(Coronel and Morris, 2015) (Hoffer et al., 2019)

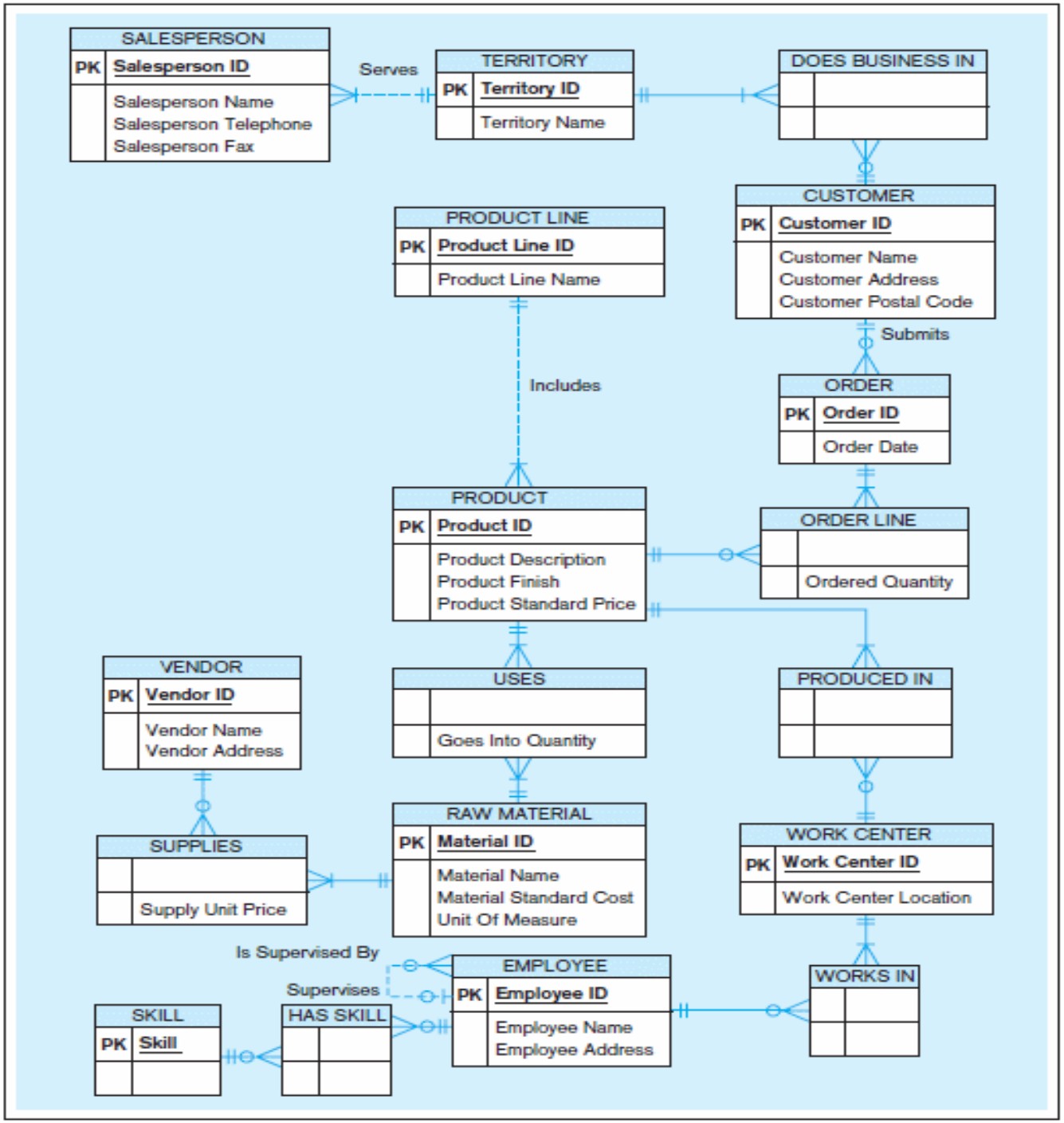
cdio 4.0

fit@hcmus

Guidelines for Defining Relationship

- A definition statement for a relationship should:
  - State what action is and possibly why it is important
  - Clarify actions using examples
  - Explain constraints/restrictions in the relationship:
    - Optional: What conditions lead to zero associated entities?
    - Maximum: What conditions lead to limit constraints?
    - Mutually exclusive relationship (e.g., An employee cannot both be Supervised By and be Married To the same employee.)
  - Explain historical data kept in the relationship
  - Explain participation transfers, if they exists.
    - Can an entity instance ‘E1' involved in relationship instance ‘R1' transfer to another instance ‘R2' based on certain conditions?

(Coronel and Morris, 2015) (Hoffer et al., 2019)



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

Example of ERD in Visio

(Coronel and Morris, 2015) (Hoffer et al., 2019)

References

(Coronel and Morris, 2015, **chapter 2, 4**)

Database System: Design, Implementation, and Management, 12th Edition, Carlos Coronel & Stevem, Morris, 2015.

(Hoffer et al., 2019, **Part II, chapter 2**)

Modern Database Management, 13th Edition, Jeffrey A. Hoffer, V. Ramesh, Heikki Topi, 2019.

THE END