Database Analysis

(Dealing with conceptual data modeling)

- Chapter 2: Modeling data in an organization (Basic E-R model)
- Chapter 3: Enhanced E-R model

MDBM-Chapter 2

Modeling Data in the Organization

- Modeling the rules of the organization
- The E-R model
- Modeling entities and attributes
- Modeling the relationships
- E-R modeling example: PVFC

Objectives

- Define terms
- Understand importance of data modeling
- Write good names and definitions for entities, relationships, and attributes
- Distinguish unary, binary, and ternary relationships
- Model different types of attributes, entities, relationships, and cardinalities
- Draw E-R diagrams for common business situations
- Convert many-to-many relationships to associative entities
- Model time-dependent data using time stamps

Modeling the Rules of the Organization

Business Rules

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

Example: "Every student must have a faculty advisor."

The Purpose of data modeling:

To represent some business rules, which in turn govern data, and then to enforce these rules through DB technologies

Example: modeling student data with the rule that a student is any person who is officially registered with the university with an active student status. In this case, an alumnus is not a student and his/her data do not belong to the student data.

A Good Business Rule Is:

- Declarative: what, not how
- Precise: clear, agreed-upon meaning
- Atomic: one statement (indivisible yet sufficient)
- Consistent: not contradict internally and externally
- Expressible: structured, natural language
- Distinct: non-redundant
- Business-oriented: understood and owned by business people

General Guidelines of Naming a Data Object

- Related to business (not technical) characteristics
- Meaningful and self-documenting
- Unique
- Readable (e.g., grade point average, or GPA)
- Composed of words from an approved list
- Repeatable (e.g., Student Birth Date, Employee Birth Date)
- Written in standard syntax (of the organization)

Data Definitions (Structural Assertion)

- Term and fact
  - **Term**: word or phrase with specific meaning (e.g., course, flight)
  - **Fact**: association between two or more terms (e.g., a customer rents a car)
- General guidelines for good data definition (a type of business rule explaining a term or fact)
  - A concise description of essential data meaning
  - Gathered in conjunction with systems requirements
  - Accompanied by diagrams (e.g., E-R diagram)
  - Stated in the **singular** form
  - Achieved by consensus, and iteratively refined

Entity-Relationship (E-R) Model

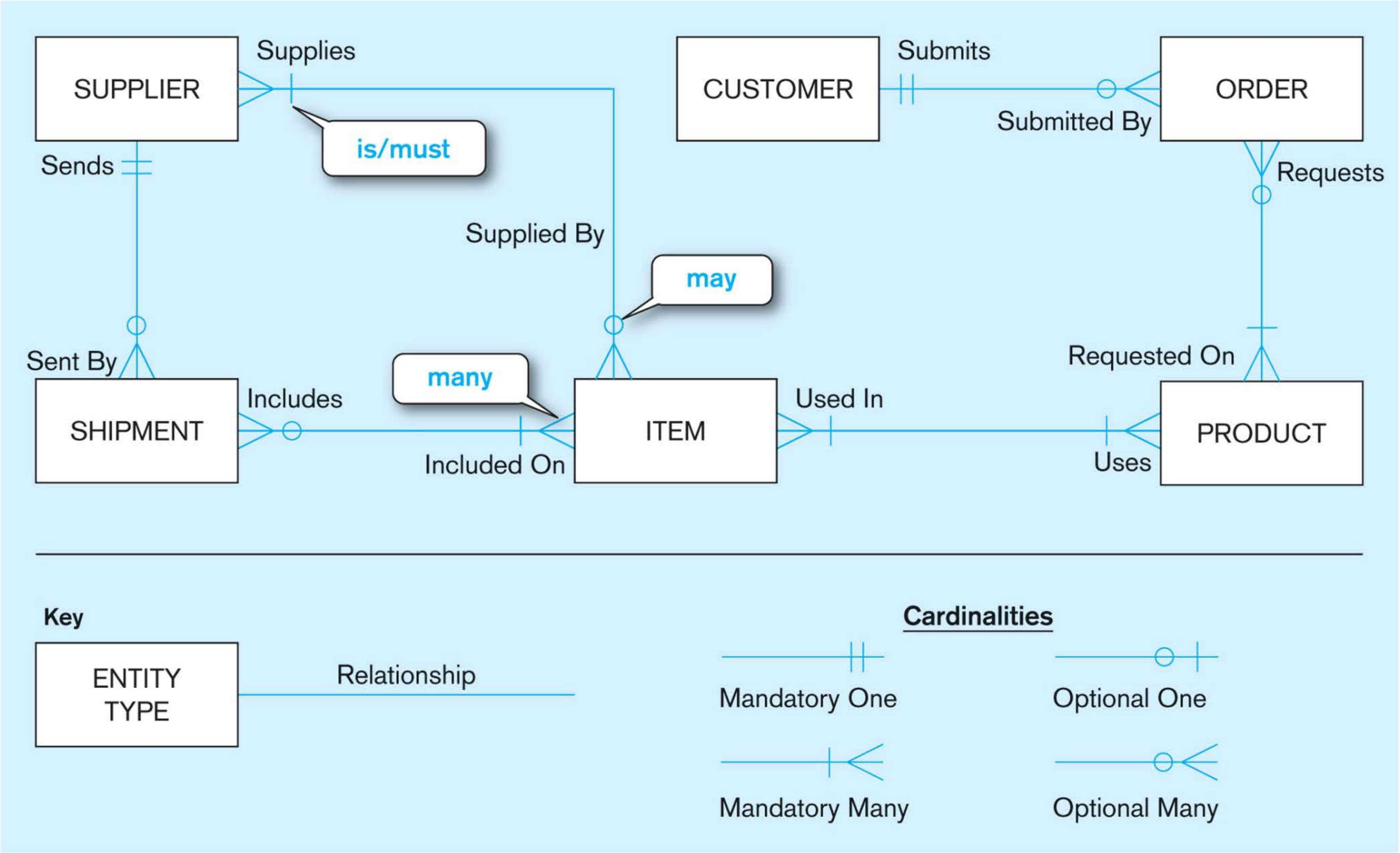
A **logical representation** of the data for an organization or for a business area, using entities for categories of data and relationships for associations between entities.

Entity-relationship diagram (E-R diagram, or ERD): a graphical representation of an E-R model.

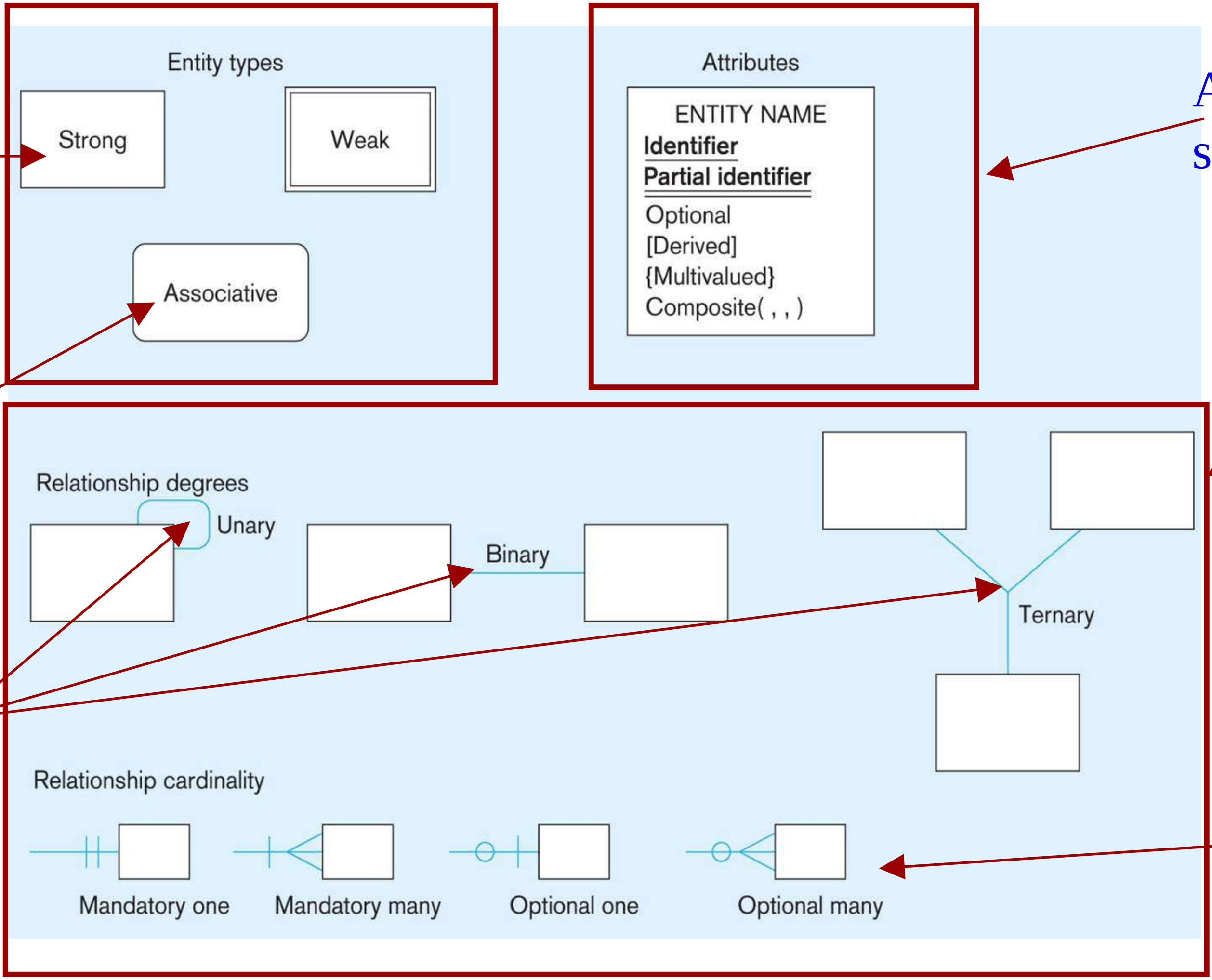
E-R Model Constructs

- **Entity**, which often corresponds to a **table**.
- **Relationship**, which often corresponds to **primary/foreign key** equivalencies in the related tables.
- **Attribute**, which is a property or characteristic of an entity or relationship type and often corresponds to a **field** in a table.

A Sample E-R Diagram



Basic E-R Notation (HRT Notation) – Combinations of Most Desirable Features



Entity symbols

A special entity that is also a relationship

Attribute symbols

Relationship symbols

Relationship degrees specify number of entities involved

Relationship cardinalities specify the number of instances of each entity type allowed

(There is no industry-standard notation.)

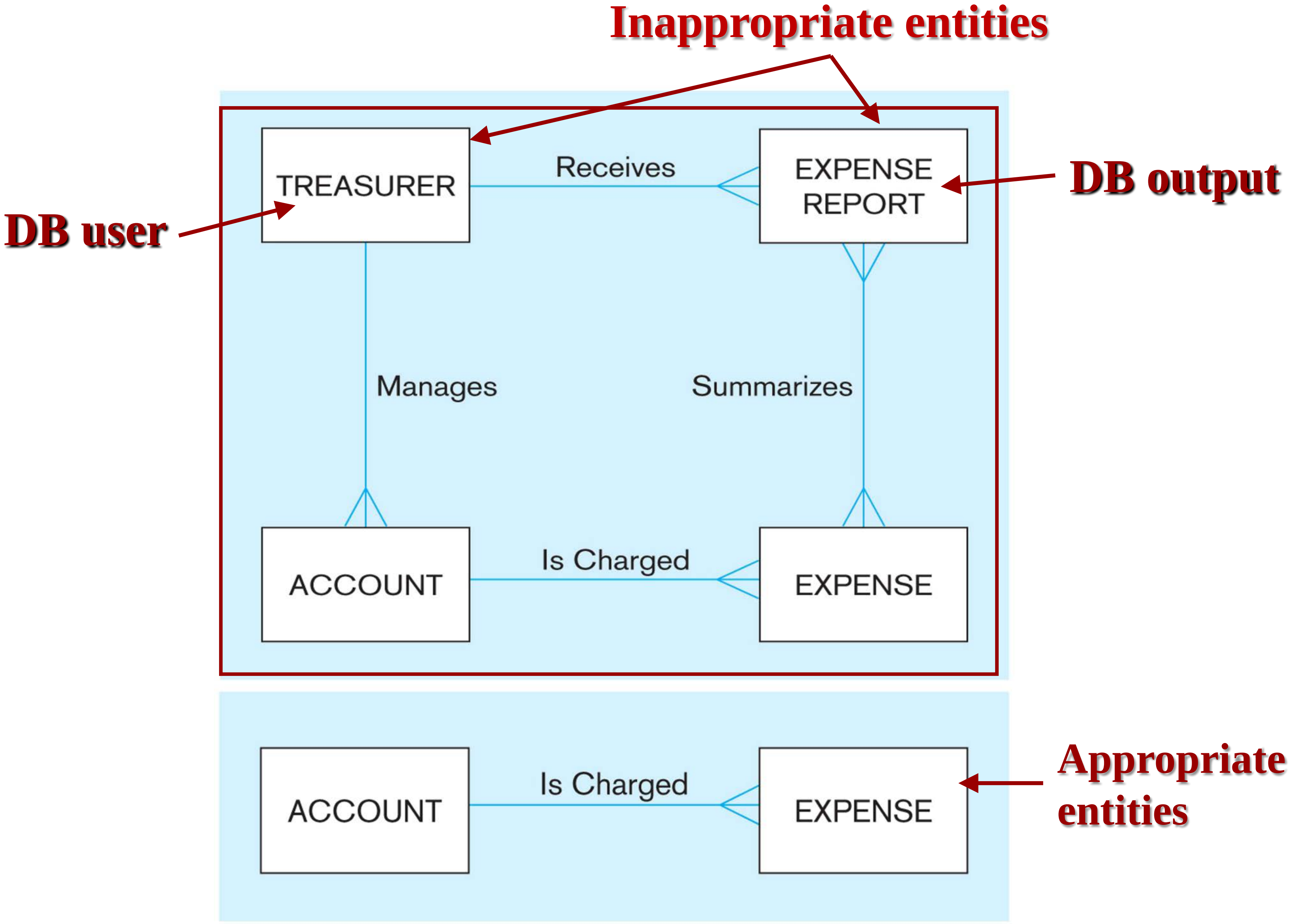
Common Data Modeling (ERD Drawing) Tools/Notations

- ❑ Microsoft Visio
- ❑ Oracle Designer
- ❑ CA ERwin Data Modeler
- ❑ PowerDesigner (Sybase)
- ❑ HRT notation from the textbook (Hoffer-Ramesh-Topi)

Modeling Entities and Attributes

An Entity

- SHOULD BE:
  - An object that will have many instances in the database (Student: Student A, Student B, ...)
  - An object that will be composed of multiple attributes (Name, Gender, Student #, ...)
- SHOULD NOT BE:
  - A user of the database system
  - An output of the database system (e.g., a report)



Inappropriate entities

DB user

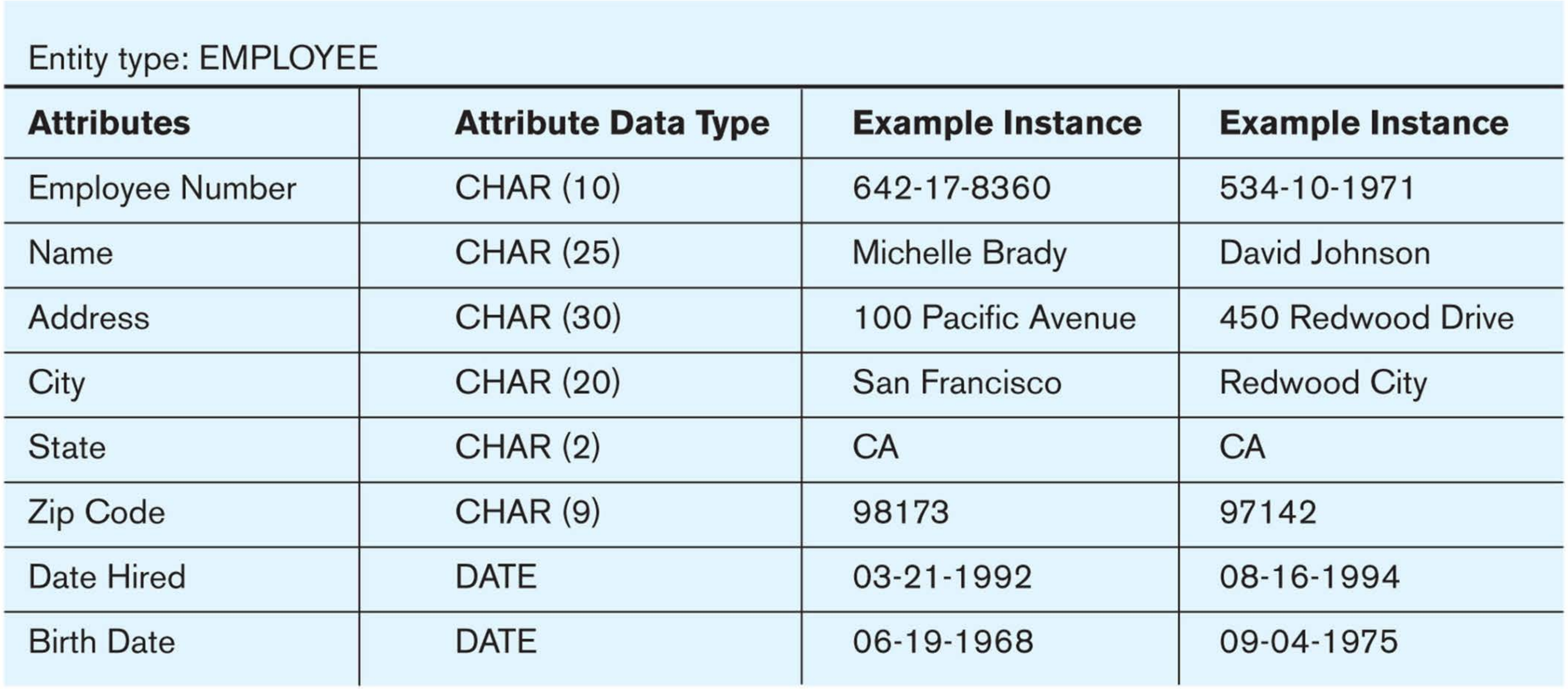
DB output

Appropriate entities

Entity Type vs. Entity Instance

- Entity type
  - is a collection of entities that share common properties or characteristics
  - has a unique singular name
  - is expressed with capital letters
  - is described just once using metadata
  - often corresponds to a DB table
- Entity instance
  - is a single occurrence of an entity
  - often corresponds to a row in a table
  - there may exist many instances in a database

Entity Type EMPLOYEE With Two Instances



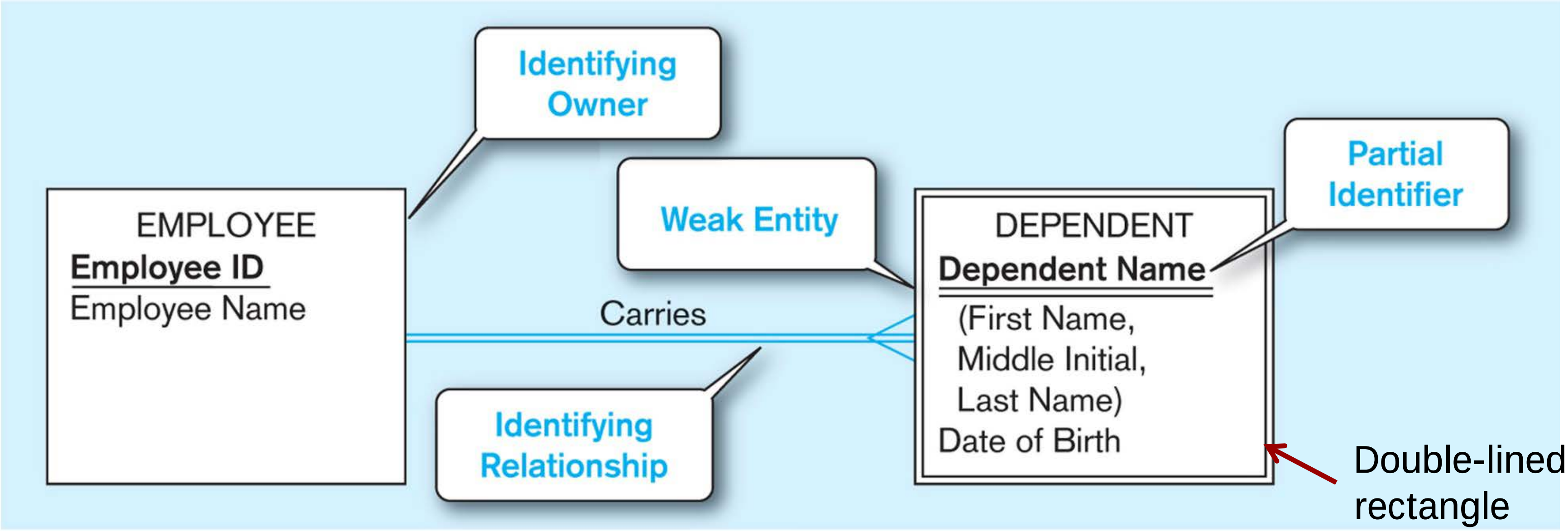
Entity type: EMPLOYEE

Attributes	Attribute Data Type	Example Instance	Example Instance
Employee Number	CHAR (10)	642-17-8360	534-10-1971
Name	CHAR (25)	Michelle Brady	David Johnson
Address	CHAR (30)	100 Pacific Avenue	450 Redwood Drive
City	CHAR (20)	San Francisco	Redwood City
State	CHAR (2)	CA	CA
Zip Code	CHAR (9)	98173	97142
Date Hired	DATE	03-21-1992	08-16-1994
Birth Date	DATE	06-19-1968	09-04-1975

Strong vs. Weak Entities, and Identifying Relationships

- Strong entity
  - exists independently of other types of entities
  - has its own unique identifier
    - identifier underlined with a single line
- Weak entity
  - is 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 a double line
  - entity box has a double line
- Identifying relationship
  - links a strong entity to weak entity

Example of a Weak Entity and its Identifying Relationship



Strong entity

Weak entity

Classifications of attributes

(Attribute: a property or characteristic of an entity or a relationship)

- Identifier attributes
- Required versus optional attributes
- Simple versus composite attributes
- Single-valued versus multivalued attributes
- Stored versus derived attributes

Identifiers (Keys)

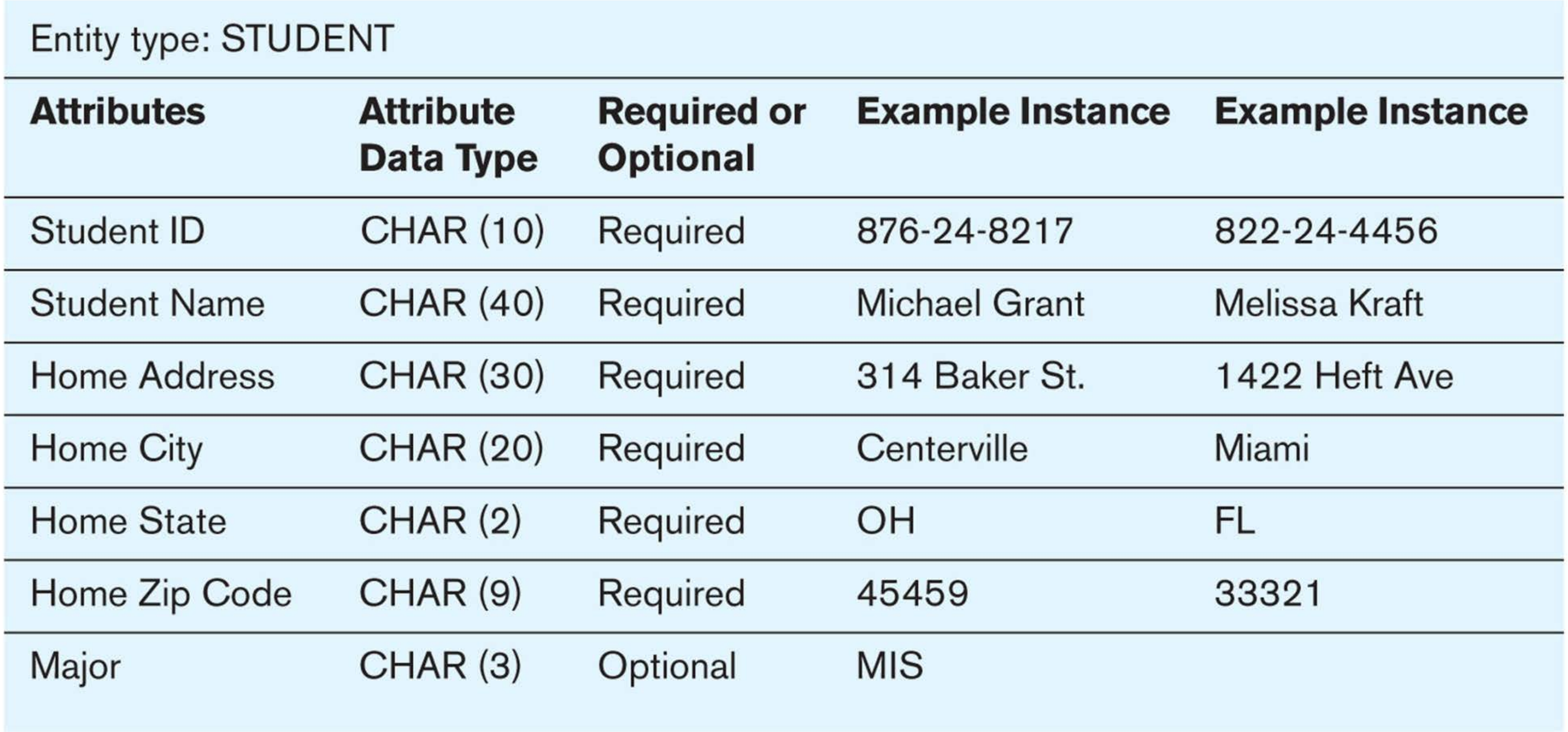
- **Identifier** (Primary Key)–an attribute (or combination of attributes) that uniquely identifies individual instances of an entity
  - Underlined
  - **Boldfaced**
- **Candidate identifier**–an attribute that could be a key...satisfies the requirements for being an identifier
- They can be simple or composite

Criteria for Identifiers

- Choose attributes that
  - are unique
  - will not be null (required)
  - will not change in value
- Avoid intelligent identifiers, whose structure indicates something (e.g., the first two digits indicate a location)
- Substitute new, simple keys (**s**urrogate identifiers) for long, composite keys

Required vs. Optional Attributes

- Required attribute
  - must have a value for every entity instance
- Optional attribute
  - may not have a value for every entity instance

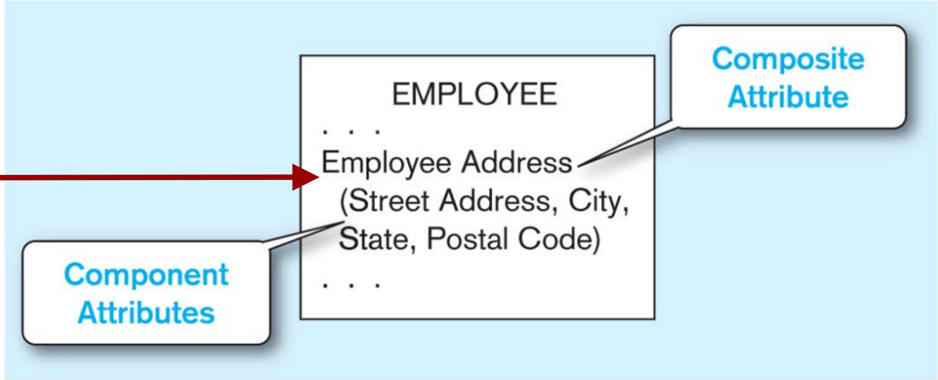


Entity type: STUDENT

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	

A composite attribute

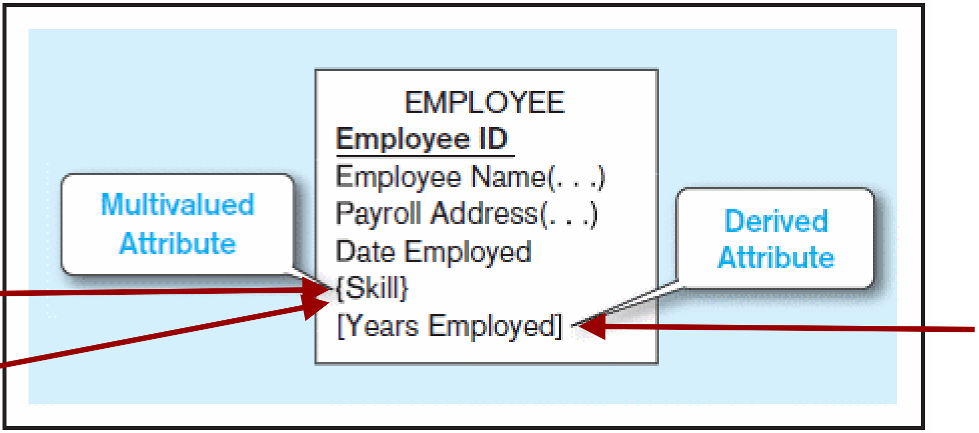
An attribute broken into component parts, which are enclosed in parentheses



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

Multivalued
an employee can have more than one skill

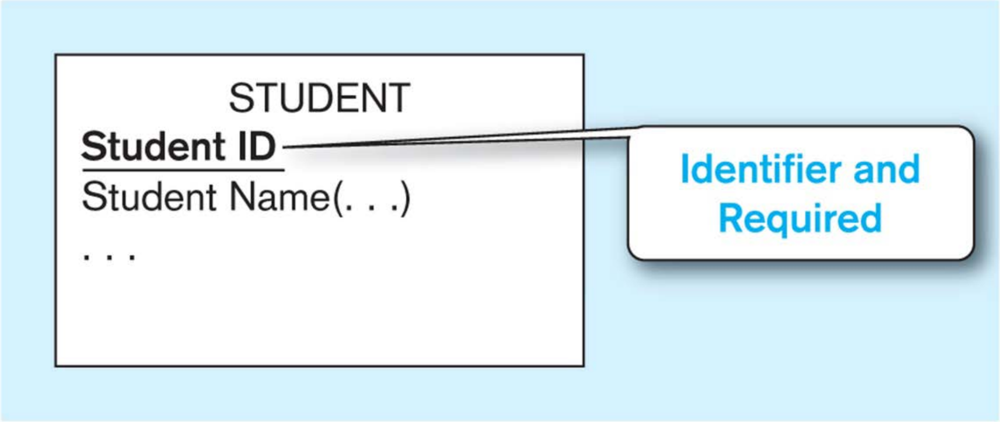
Enclosed with curly brackets { }



Derived
from date employed and current date, enclosed with square brackets [ ]

Simple and Composite Identifier Attributes

(a) Simple identifier attribute



STUDENT

Student ID

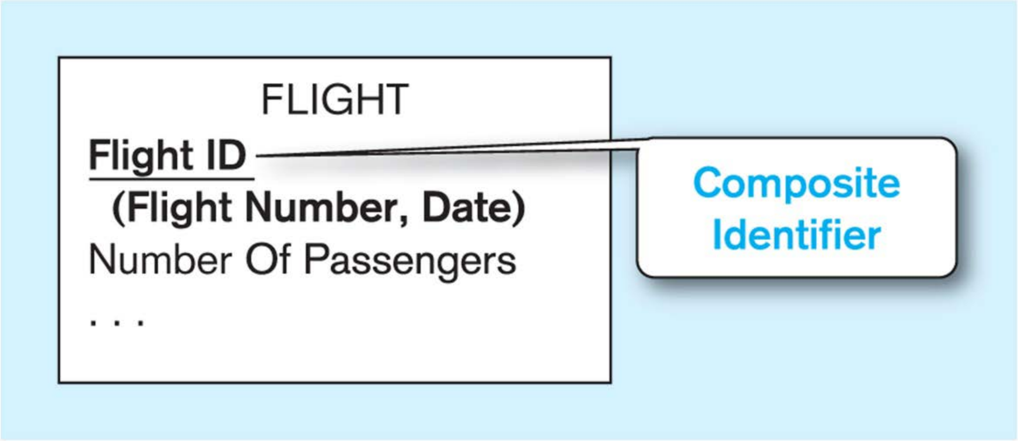
Student Name(. . .)

. . .

Identifier and Required

The identifier is boldfaced and underlined

(b) Composite identifier attribute



FLIGHT

Flight ID

(Flight Number, Date)

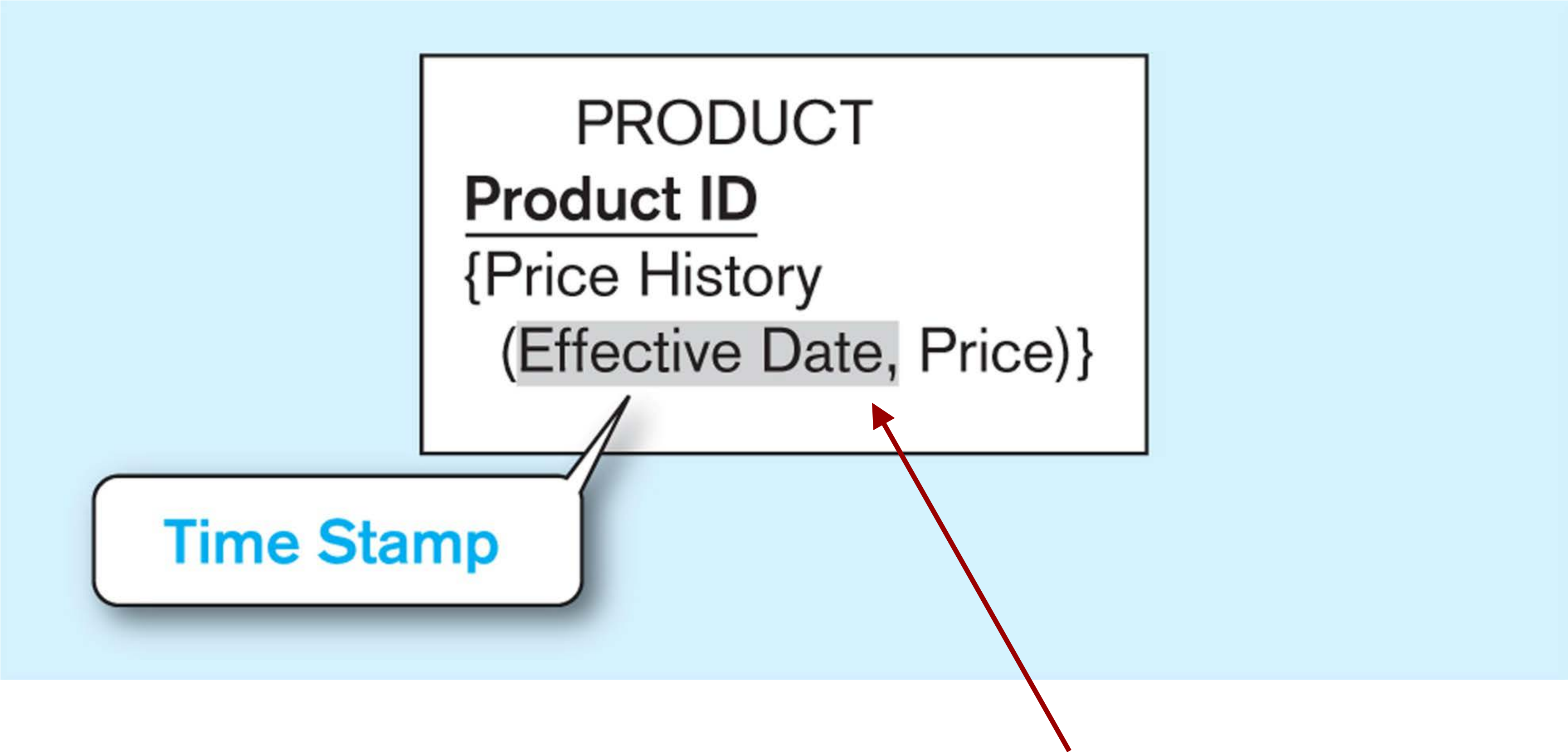
Number Of Passengers

. . .

Composite Identifier

The component attributes are boldfaced but not underlined

Simple Example of Time-stamping



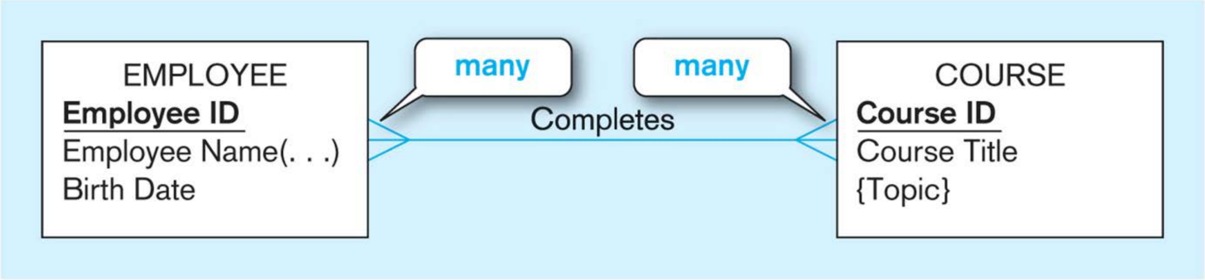
This attribute is both multivalued *and* composite

Modeling the Relationships

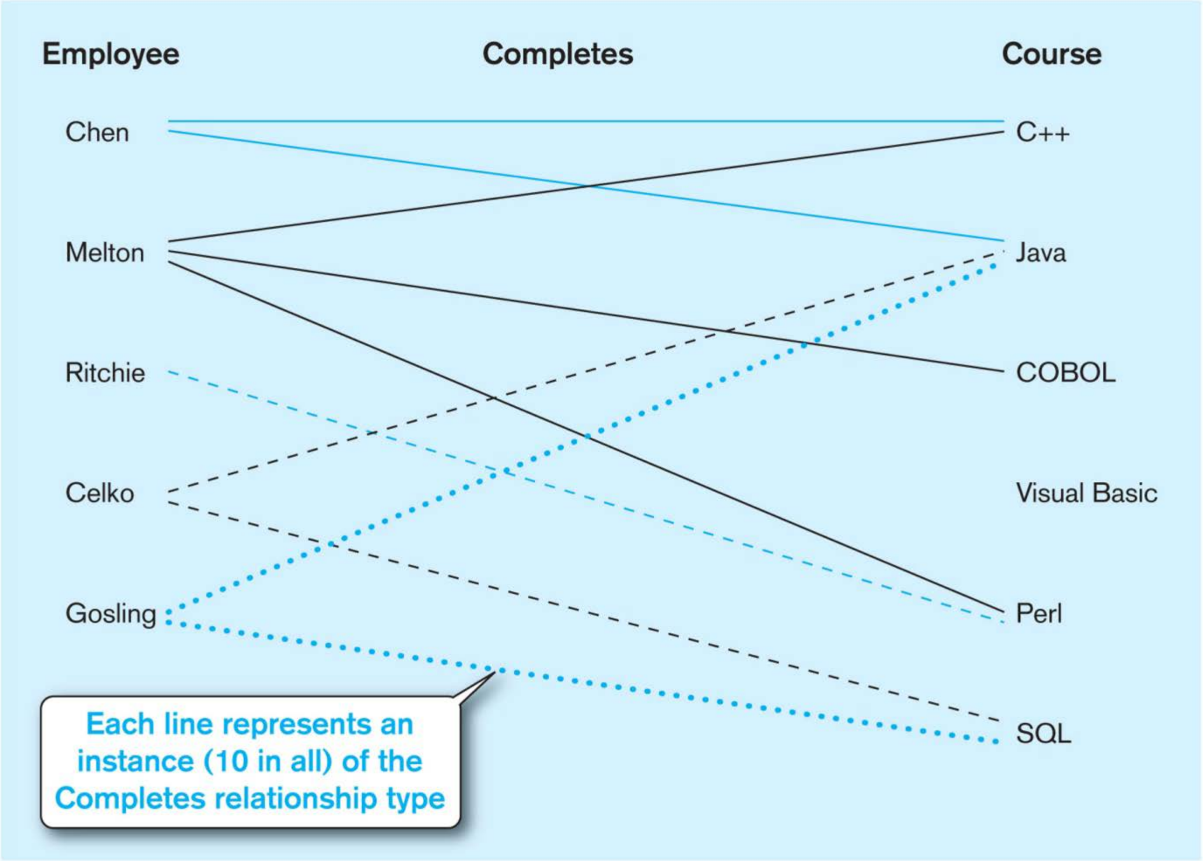
- Relationship types vs. relationship instances
  - The relationship type is modeled as **lines** between entity types…the relationship instance is between specific entity instances
- Relationships can have attributes
  - These describe features pertaining to the association between the entities in the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

Relationship Types and Instances

a) Relationship type ("Completes")



b) Relationship instances

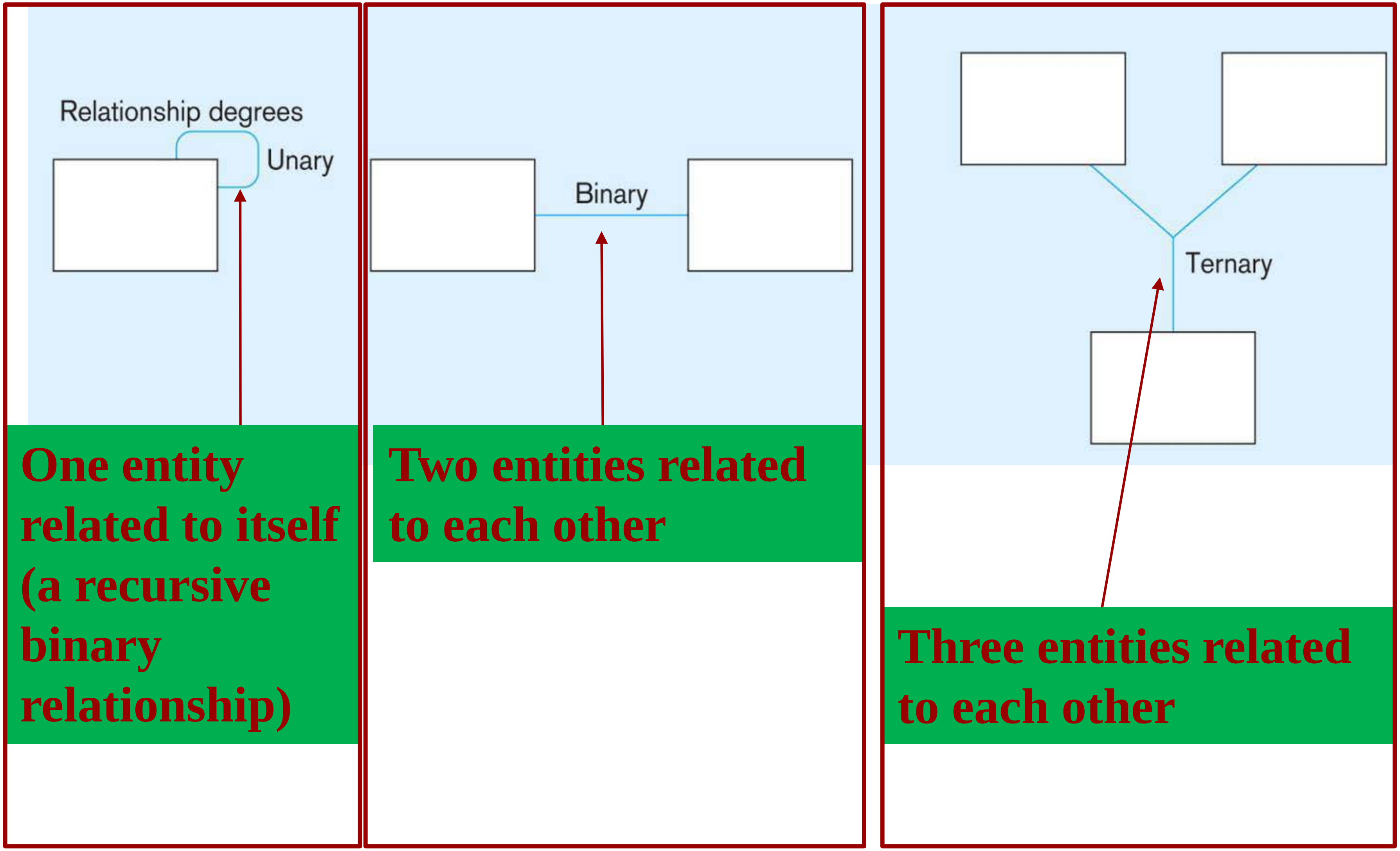


Degree of Relationships

Degree of a relationship is the number of entities that participate in it

- Unary Relationship
- Binary Relationship
- Ternary Relationship

Degree of Relationships



One entity related to itself (a recursive binary relationship)

Two entities related to each other

Three entities related to each other

Cardinality of Relationships

(The number of instances of each entity type allowed)

- One-to-One

  Each entity in the relationship will have exactly one related entity instance

- One-to-Many

  An instance of an entity on one side of the relationship can have many related instances of the entity on the other side, but the entity on the other side will have a maximum of one related instance of the other entity

- Many-to-Many

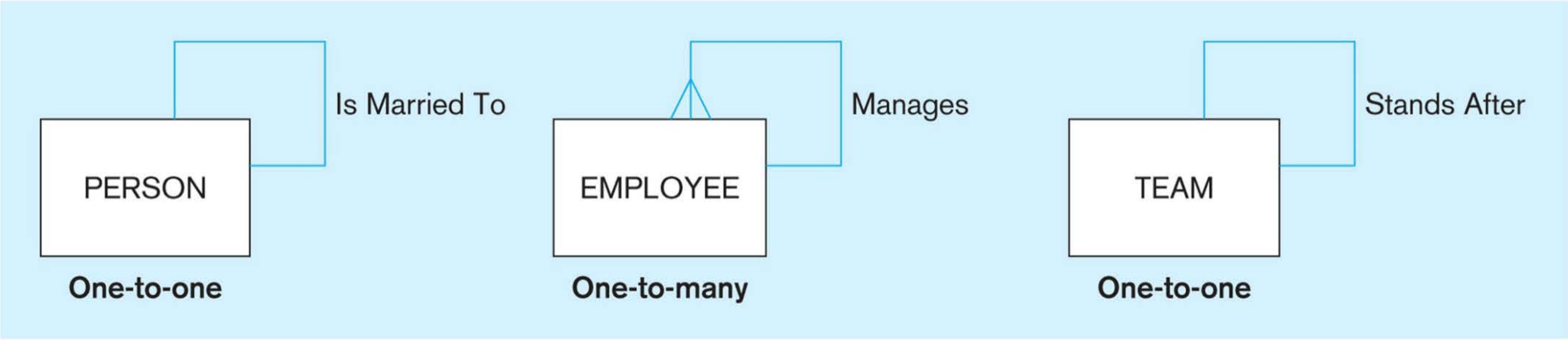
  Entities on both sides of the relationship can have many related instances of the entity on the other side

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 of instances

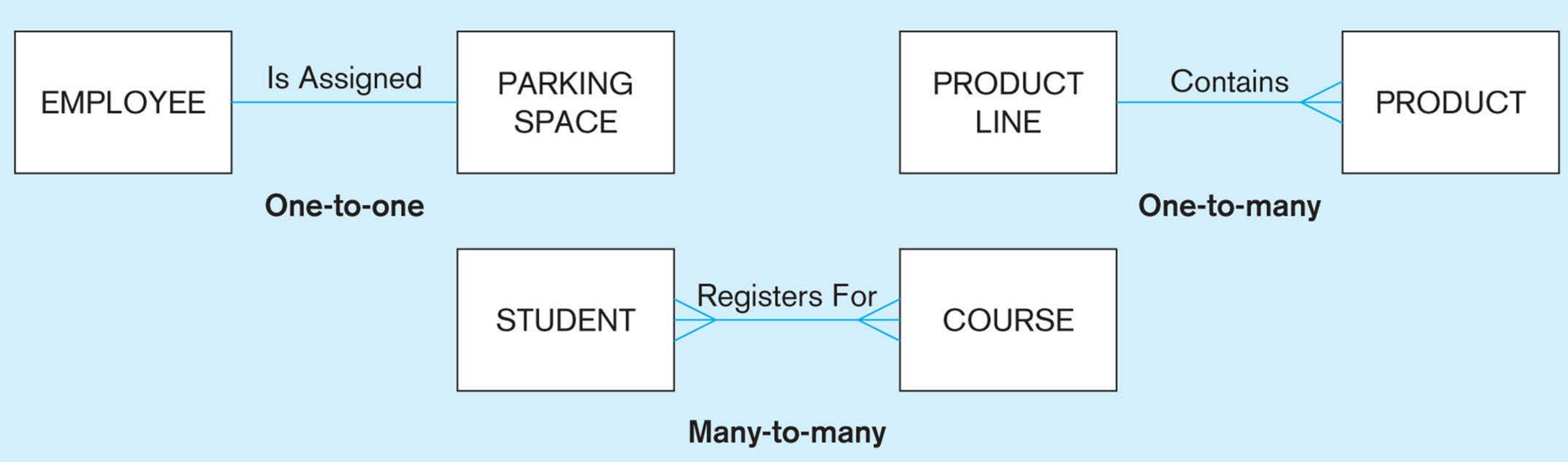
Examples of Relationships of Different Degrees

a) Unary relationships

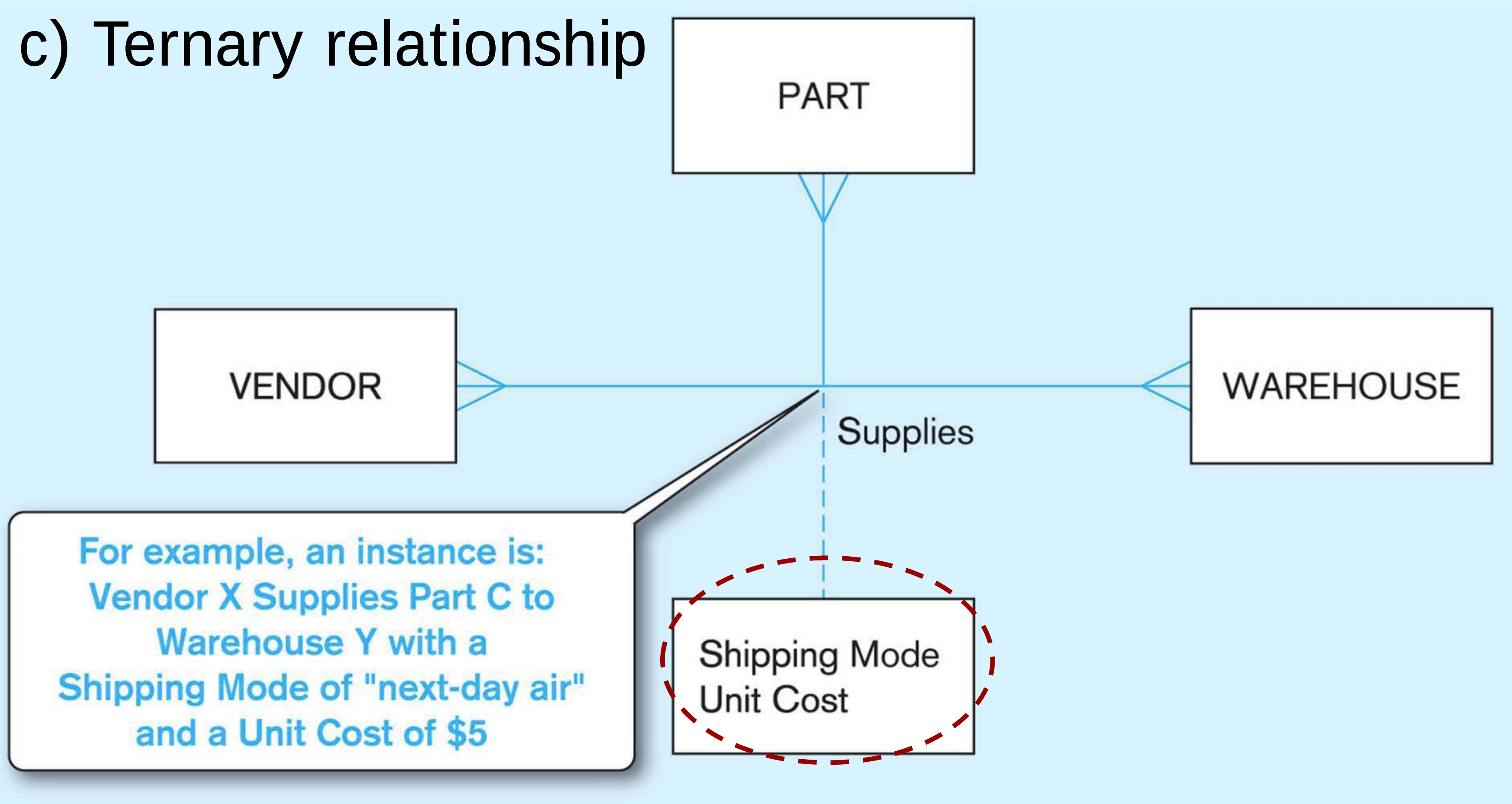


Examples of Relationships of Different Degrees (cont.)

b) Binary relationships



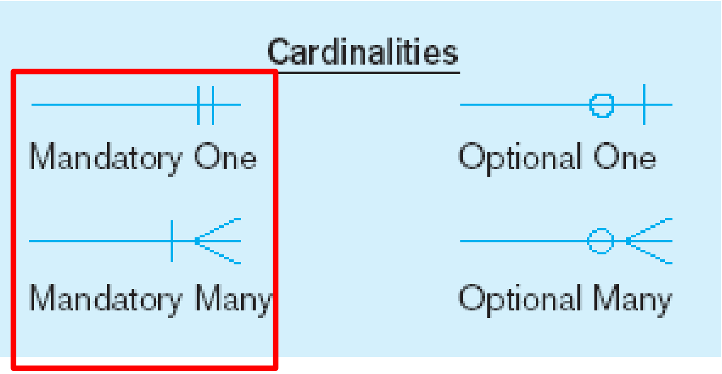
Examples of Relationships of Different Degrees (cont.)



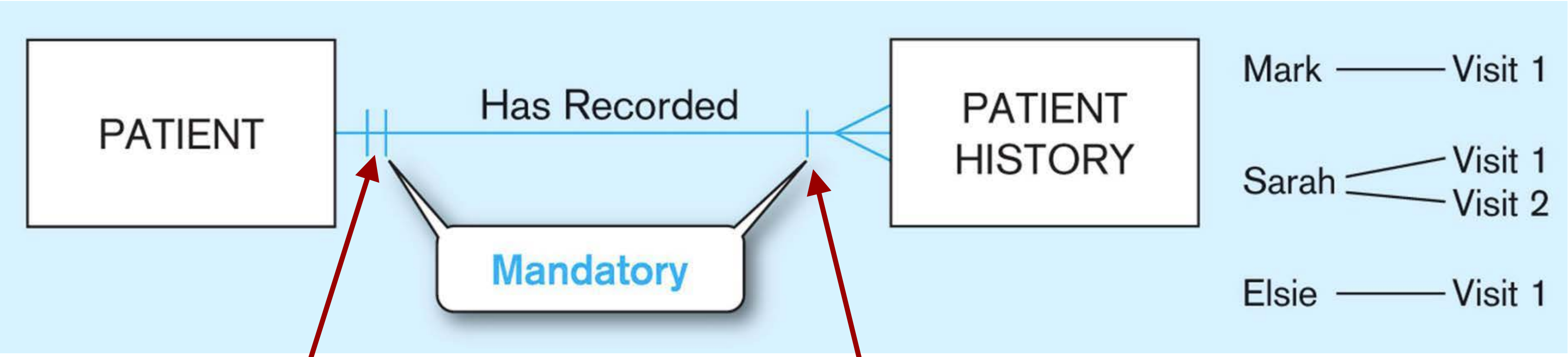
For example, an instance is: Vendor X Supplies Part C to Warehouse Y with a Shipping Mode of "next-day air" and a Unit Cost of $5

Note: a relationship can have attributes of its own

Examples of Cardinality Constraints



a) Mandatory cardinalities

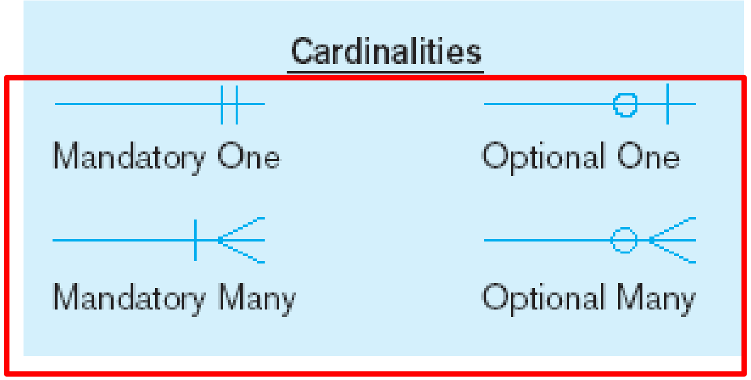


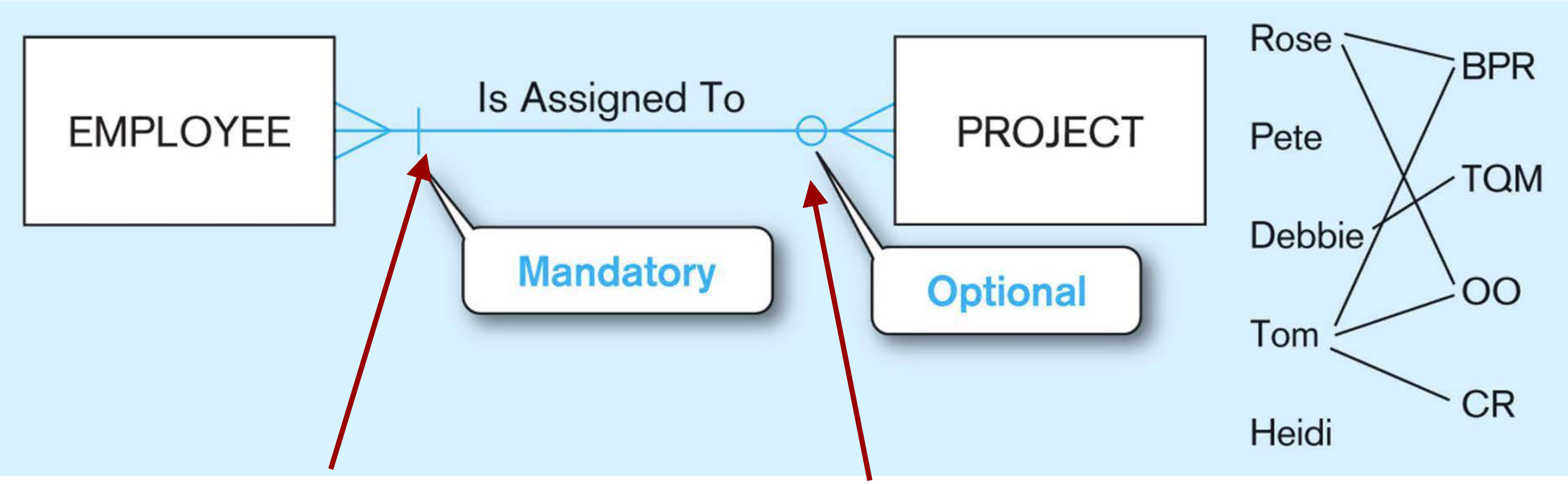
A patient history is recorded for one and only one patient

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

Examples of Cardinality Constraints (cont.)

b) One optional, one mandatory



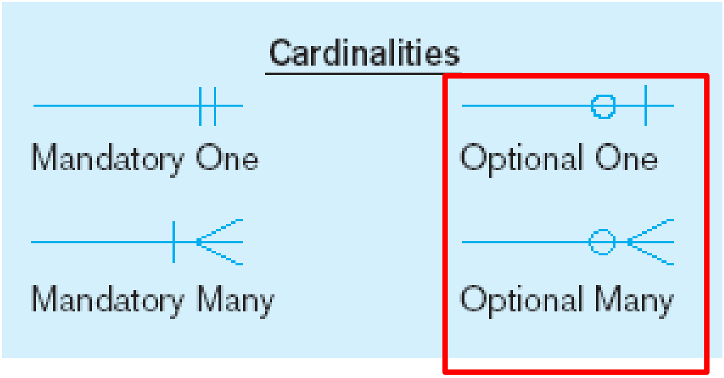


A project must be assigned to at least one employee, and may be assigned to many

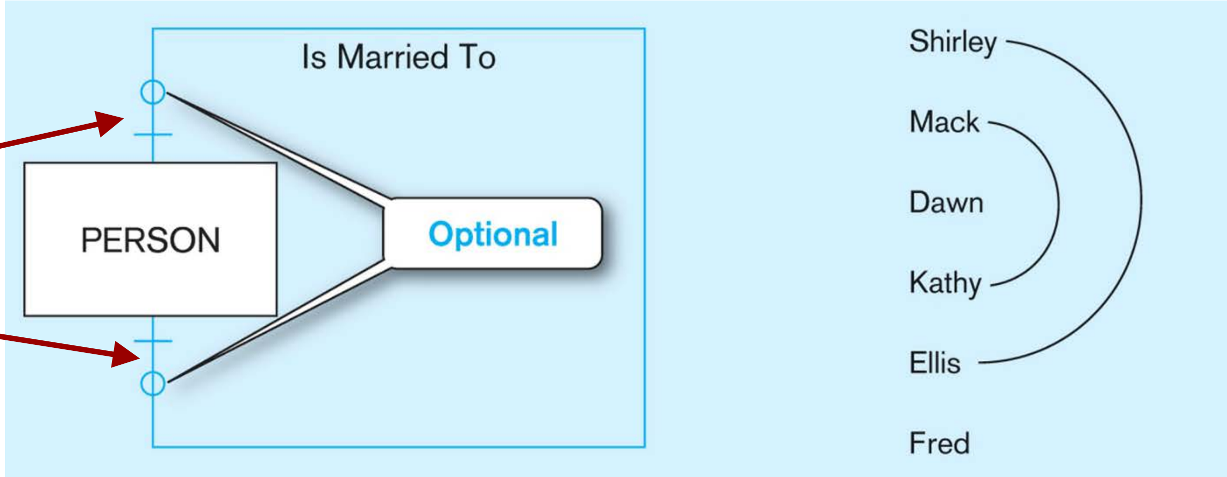
An employee can be assigned to any number of projects, or may not be assigned to any at all

Examples of Cardinality Constraints (cont.)

c) Optional cardinalities

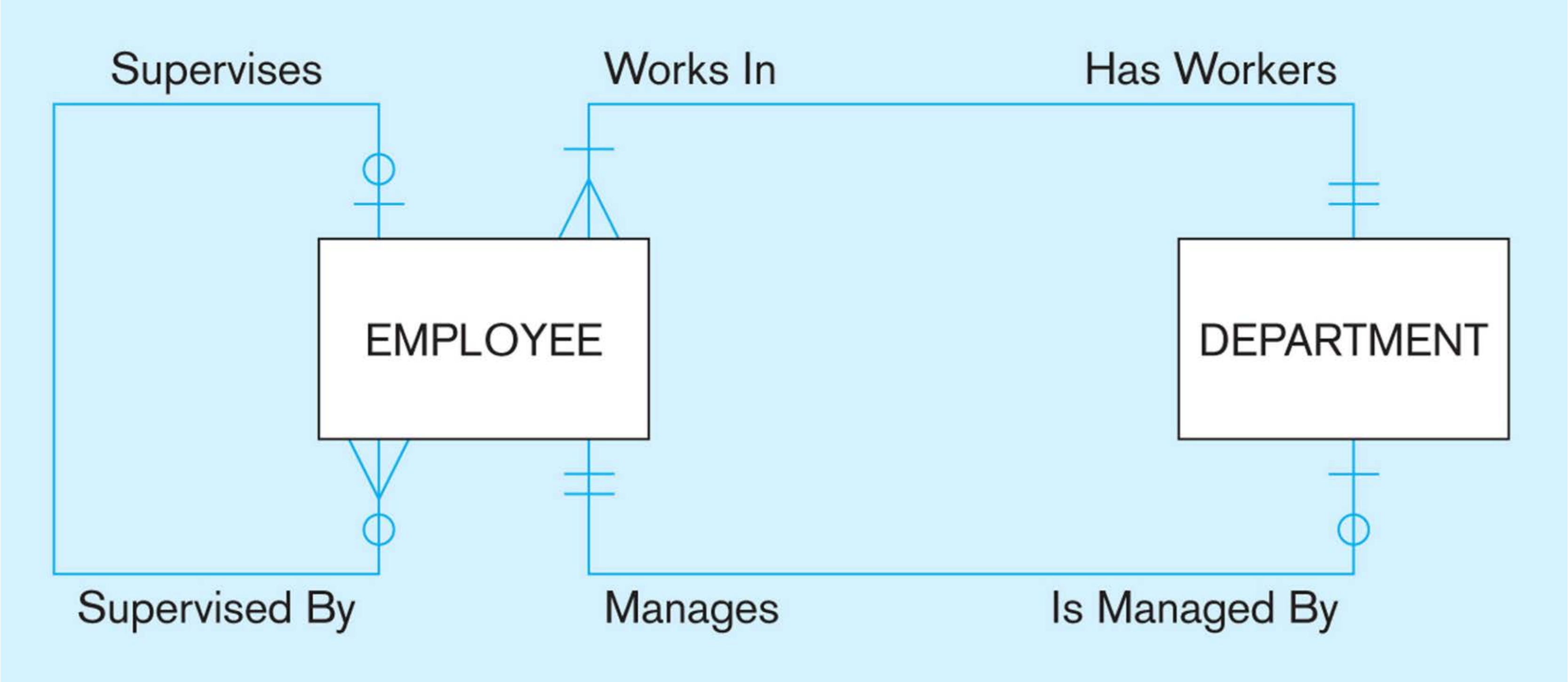


A person is married to at most one other person, or may not be married at all



Examples of Multiple Relationships

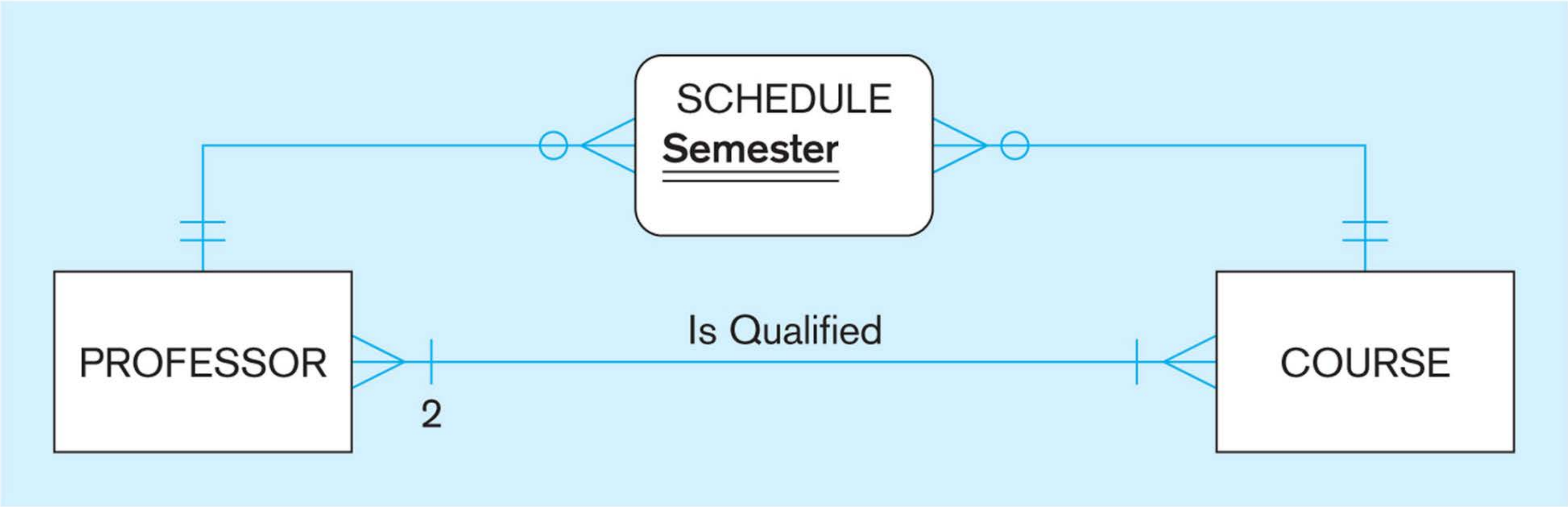
a) Employees and departments



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)



Here, the min cardinality constraint is 2. At least two professors must be qualified to teach each course. Each professor must be qualified to teach at least one course.

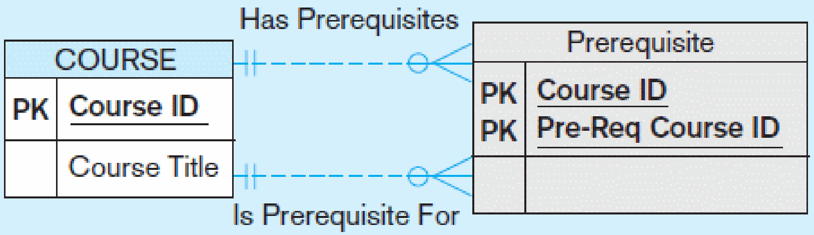
Multivalued Attributes can be Represented as Entities and Relationships

ATTRIBUTE

RELATIONSHIP & ENTITY

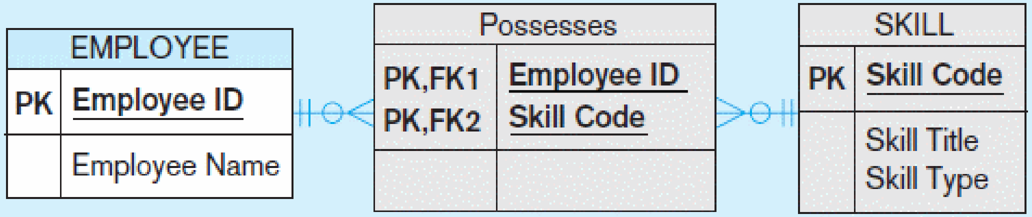
Simple multivalued

COURSE
Course ID
Course Title
{Prerequisite}



Composite multivalued

EMPLOYEE
Employee ID
Employee Name
{Skill (Skill Code,
Skill Title, Skill Type)}



Associative Entity

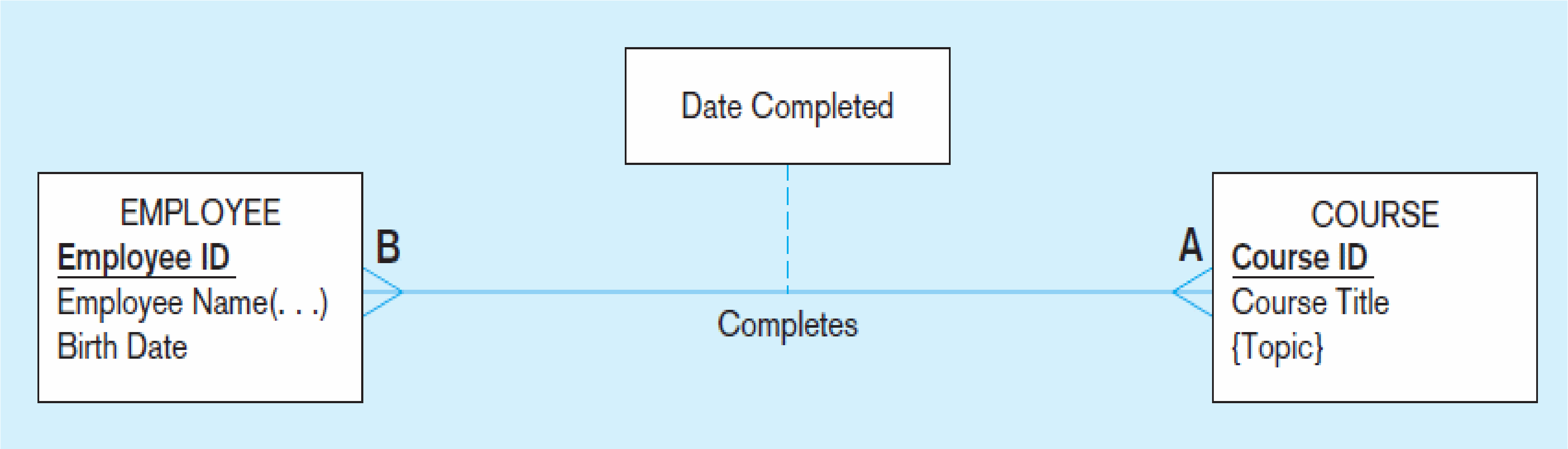
An entity type that associates one or more entity types and contains attributes that are particular to the relationship between those entity instances.

- An entity–has attributes
- A relationship–links entities together

When should a *relationship with attributes* instead be an *associative entity*?

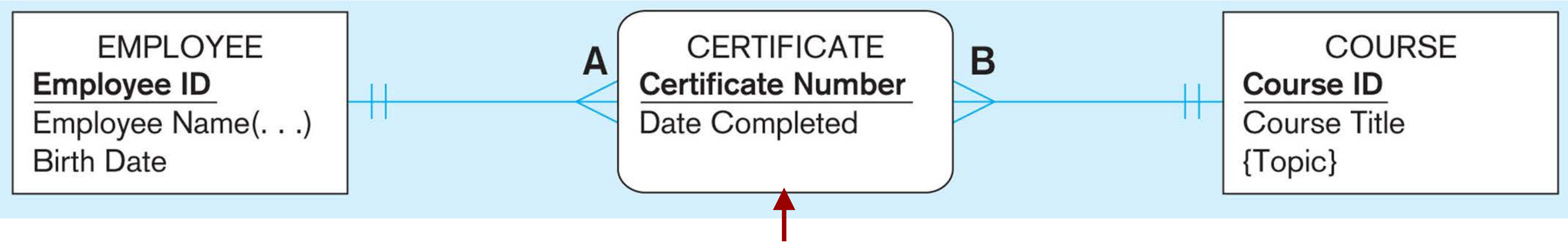
- All relationships for the associative entity should be many.
- The associative entity could have meaning independent of the other entities.
- The associative entity *preferably* has a unique identifier, and should also have other attributes.
- The associative entity may participate in other relationships 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)

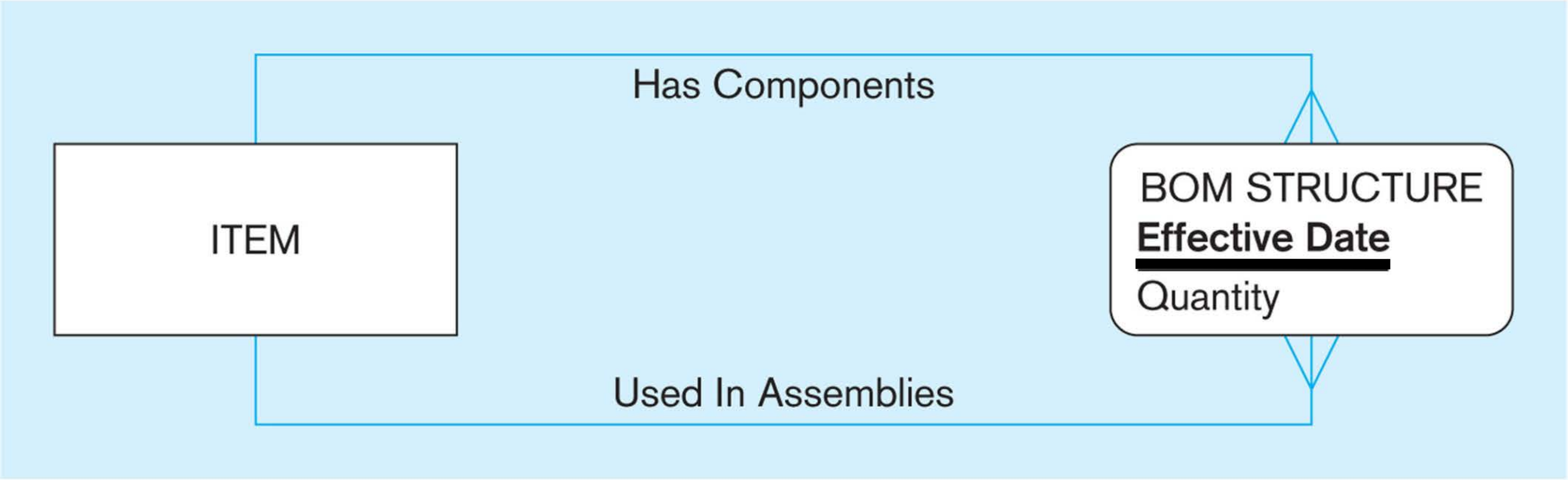


Rectangle w/ rounded corners

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

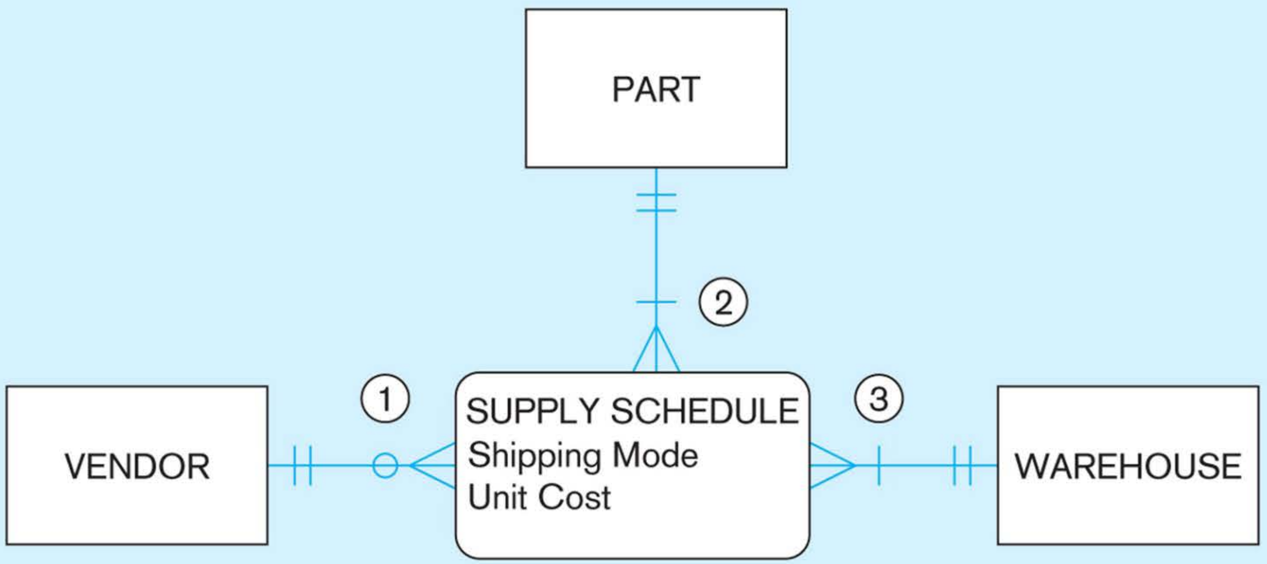
Note that the many-to-many cardinality between entities in previous slide has been replaced by two one-to-many relationships with the associative entity

An Associative Entity – Bill of Materials Structure



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

Cardinality Constraints in a Ternary Relationship

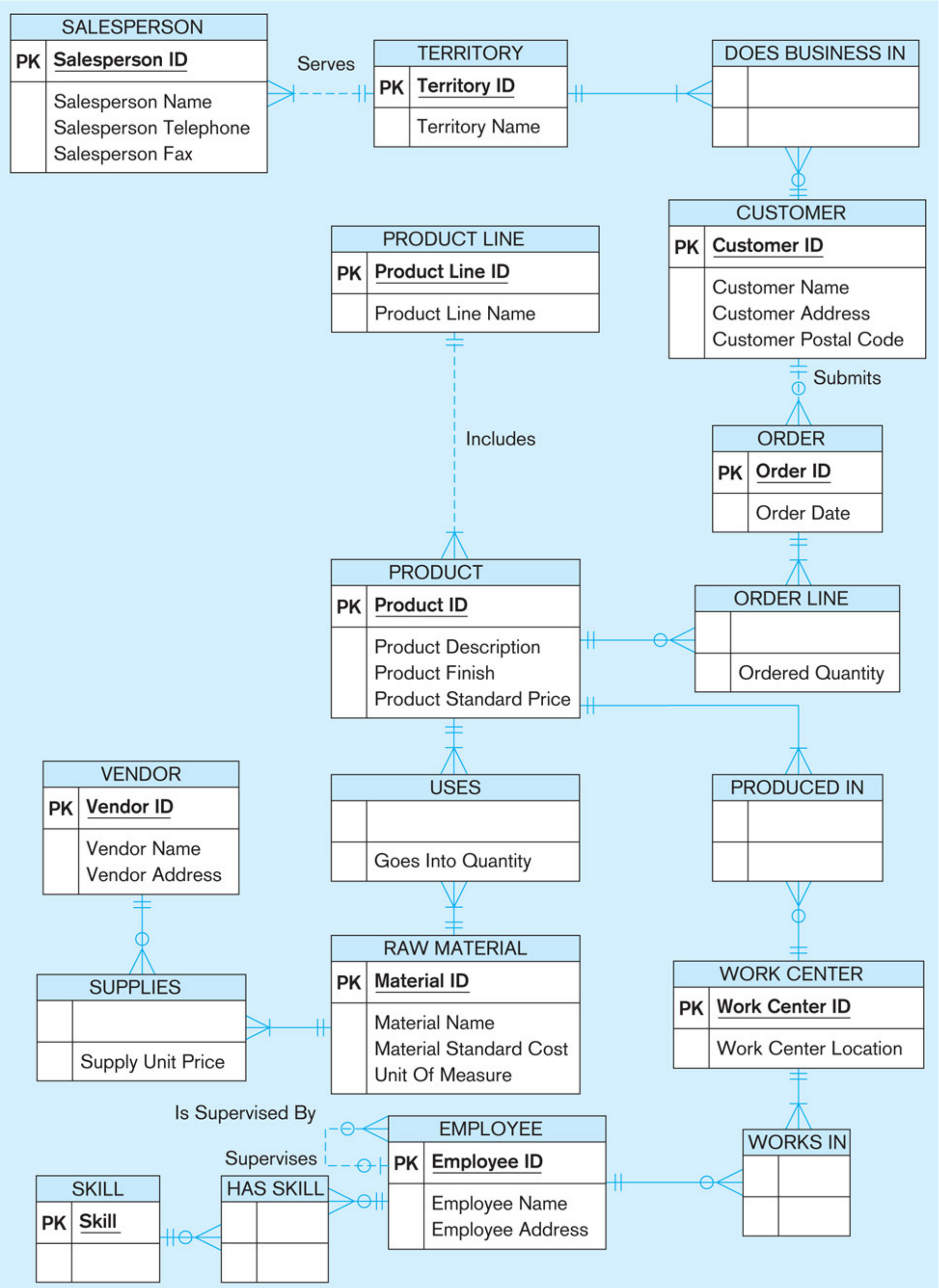


Business Rules

(1) Each vendor can supply many parts to any number of warehouses but need not supply any parts.

(2) Each part can be supplied by any number of vendors to more than one warehouse, but each part must be supplied by at least one vendor to a warehouse.

(3) Each warehouse can be supplied with any number of parts from more than one vendor, but each warehouse must be supplied with at least one part.



SALESPERSON
PK **Salesperson ID**
Salesperson Name
Salesperson Telephone
Salesperson Fax

Serves

TERRITORY
PK **Territory ID**
Territory Name

DOES BUSINESS IN

CUSTOMER
PK **Customer ID**
Customer Name
Customer Address
Customer Postal Code

Submits

ORDER
PK **Order ID**
Order Date

ORDER LINE
Ordered Quantity

PRODUCT LINE
PK **Product Line ID**
Product Line Name

Includes

PRODUCT
PK **Product ID**
Product Description
Product Finish
Product Standard Price

VENDOR
PK **Vendor ID**
Vendor Name
Vendor Address

USES
Goes Into Quantity

PRODUCED IN

SUPPLIES
Supply Unit Price

RAW MATERIAL
PK **Material ID**
Material Name
Material Standard Cost
Unit Of Measure

WORK CENTER
PK **Work Center ID**
Work Center Location

Is Supervised By

Supervises

EMPLOYEE
PK **Employee ID**
Employee Name
Employee Address

WORKS IN

SKILL
PK **Skill**

HAS SKILL

Data model for Pine Valley Furniture Company in Microsoft Visio notation

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