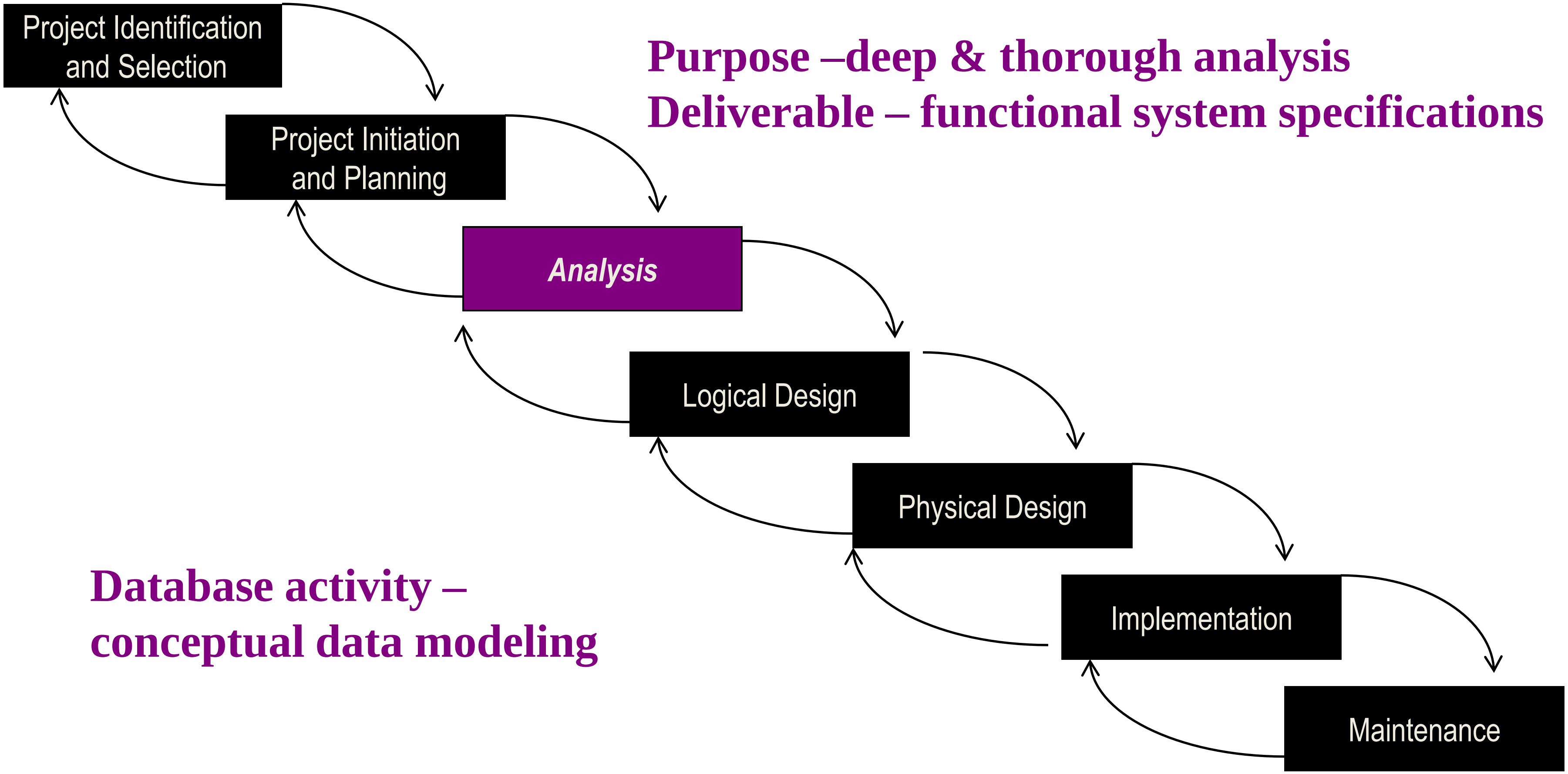
Chapter 3-5:
Modeling Data in the Organization

ENTITY – RELATIONAL MODELING (PART 1)

Introduction (Entity Analysis)

- **Entity**
  - **a thing about which an organisation holds information**
- **Entity Modelling**
  - **shows entities and the relationship between them**
- **Entity Analysis**
  - **the process of creating entity models.**
- **The diagram produced from entity modelling is called the *Entity Relationship Diagram* (ERD).**
- **When is Entity Analysis performed?**
  - **During analysis of the existing system.**
  - **In conjunction with relational data analysis.**

SDLC – Data Modeling is an Analysis Activity

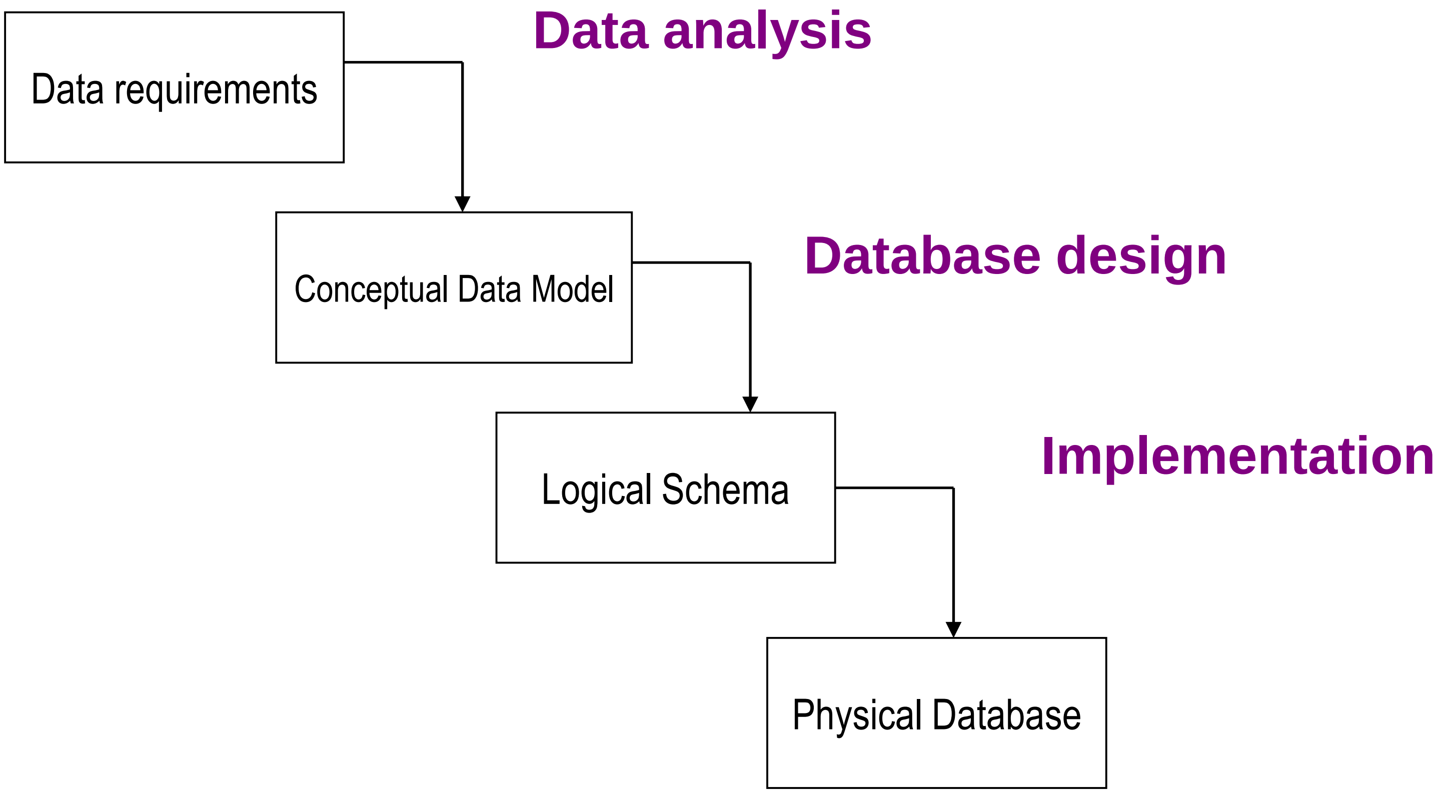

Project Identification and Selection
Project Initiation and Planning
Analysis
Logical Design
Physical Design
Implementation
Maintenance

Purpose –deep & thorough analysis
Deliverable – functional system specifications

Database activity – conceptual data modeling

Database Development Model

Establishing Requirements



Data requirements

Data analysis

Conceptual Data Model

Database design

Logical Schema

Implementation

Physical Database

DATABASE DESIGN

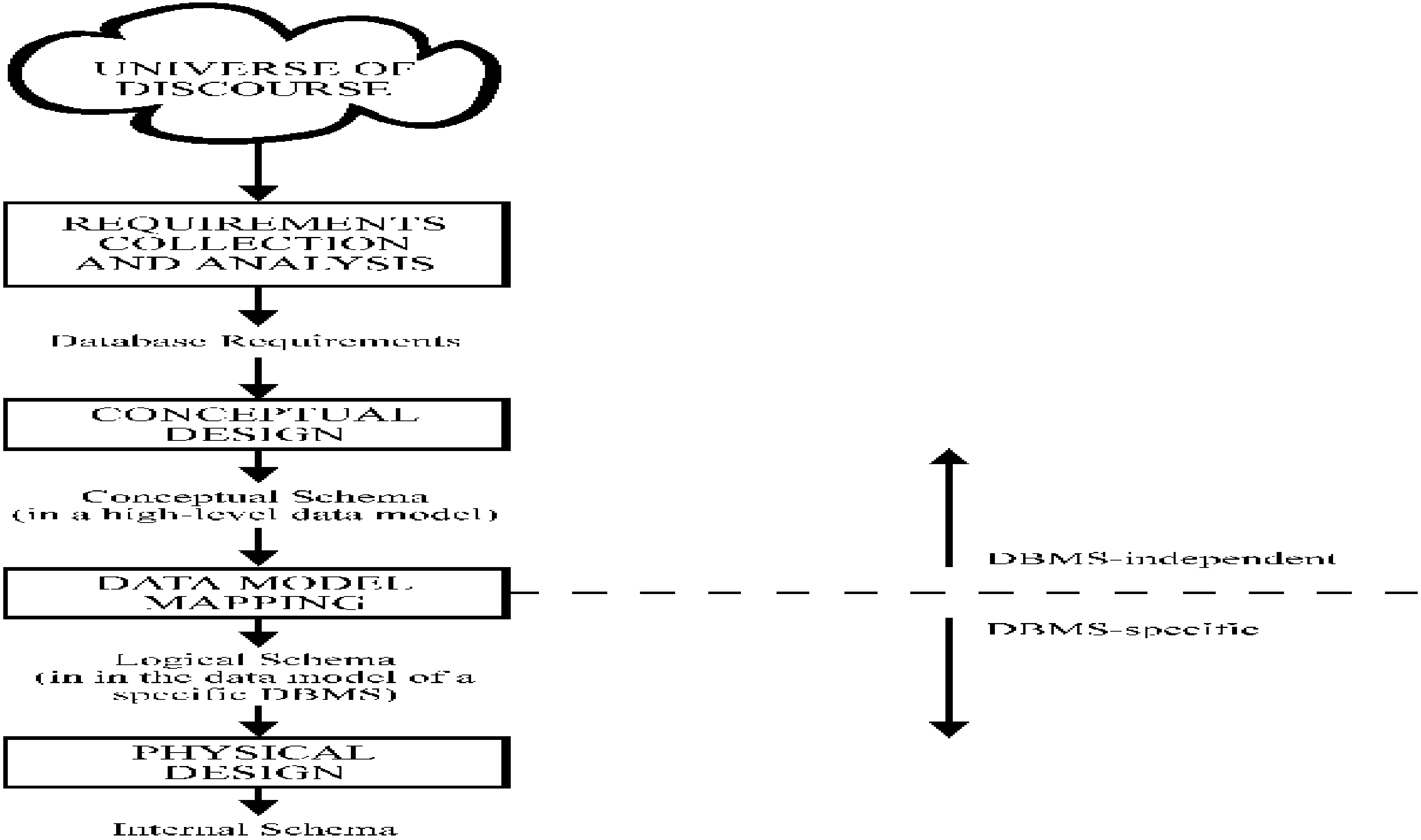
- Four steps:
- 1. Requirements Collection and Analysis
- 2. Conceptual Design
- 3. Logical Design
- 4. Physical Design

The four stages

- **REQUIREMENTS COLLECTION AND ANALYSIS**
  - **Interview users and document data requirements.**
- **CONCEPTUAL DESIGN**
  - **Form a concise description of the data requirements using a high-level data model. This description will be independent of storage requirements.**
- **LOGICAL DESIGN**
  - **Most database systems have an implementational level data model. Any good conceptual design technique will specify the mapping from the conceptual model to a variety of implementational models (e.g. relations, CODASYL, IMS).**
- **PHYSICAL DESIGN**
  - **Some database systems allow the database administrator to make decisions about physical storage. These will be driven by performance considerations.**

- **Conceptual database design** – to build the conceptual representation of the database, which includes identification of the important entities, relationships, and attributes.
- **Logical database design** – to translate the conceptual representation to the logical structure of the database, which includes designing the relations.
- **Physical database design** – to decide how the logical structure is to be physically implemented (as relations) in the target Database Management System (DBMS).

Database Process


UNIVERSE OF DISCOURSE
REQUIREMENTS COLLECTION AND ANALYSIS
Database Requirements
CONCEPTUAL DESIGN
Conceptual Schema (in a high-level data model)
DATA MODEL MAPPING
Logical Schema (in in the data model of a specific DBMS)
PHYSICAL DESIGN
Internal Schema
DBMS-independent
DBMS-specific

The Database Design Process

Data analysis

- Provides a way of structuring data, testing its use before detailed design
- Removing inconsistencies
- Planning confidentiality and validity control
- Relationships between data items (Entities) are established as part of data analysis

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

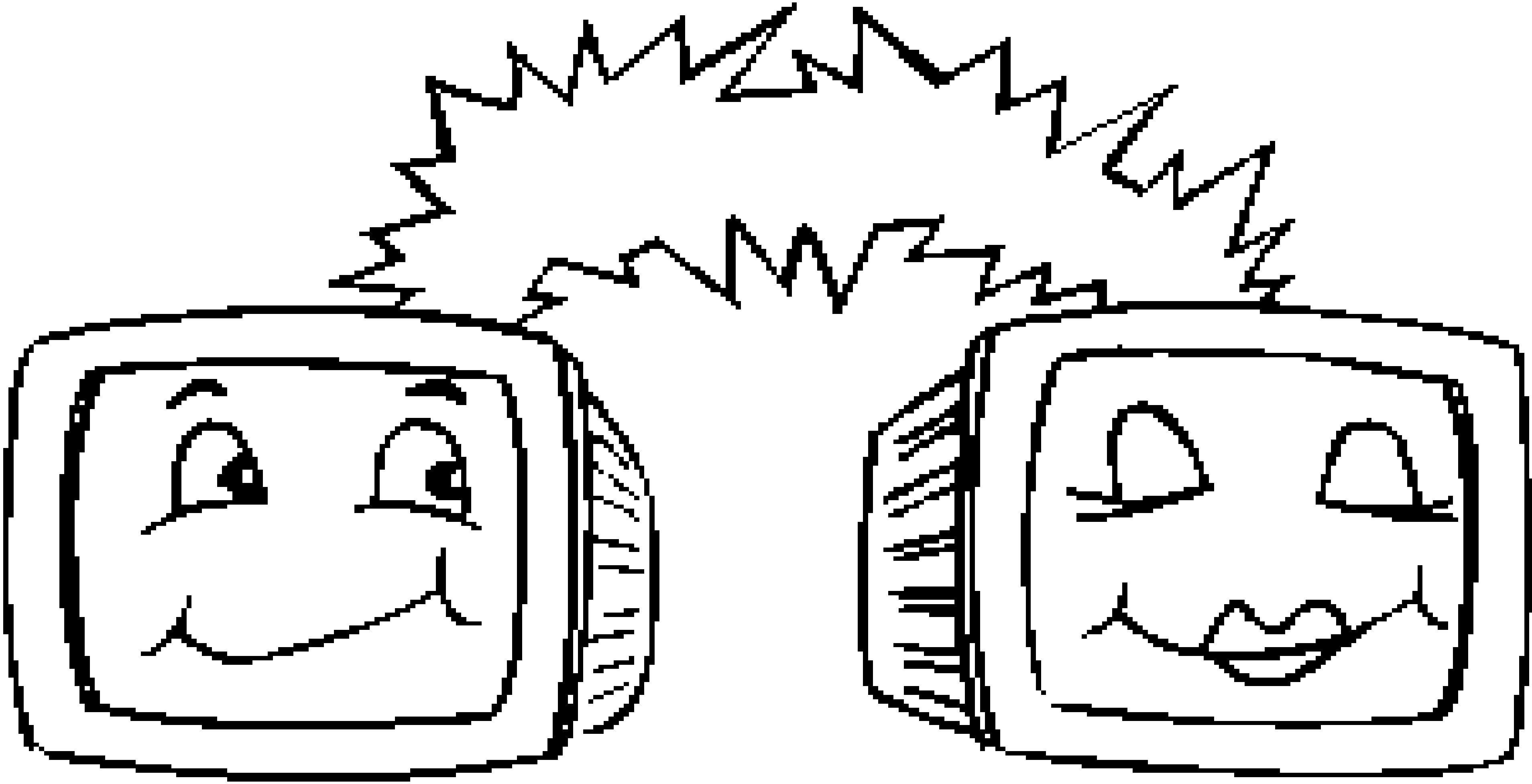
A Good Business Rule is:

- 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

E-R Model Constructs

- **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)
- Attribute - property or characteristic of an entity type (often corresponds to a field in a table)
- Relationship instance – link between entities (corresponds to primary key-foreign key equivalencies in related tables)
  - ***Relationship type*** – category of relationship…link between entity types

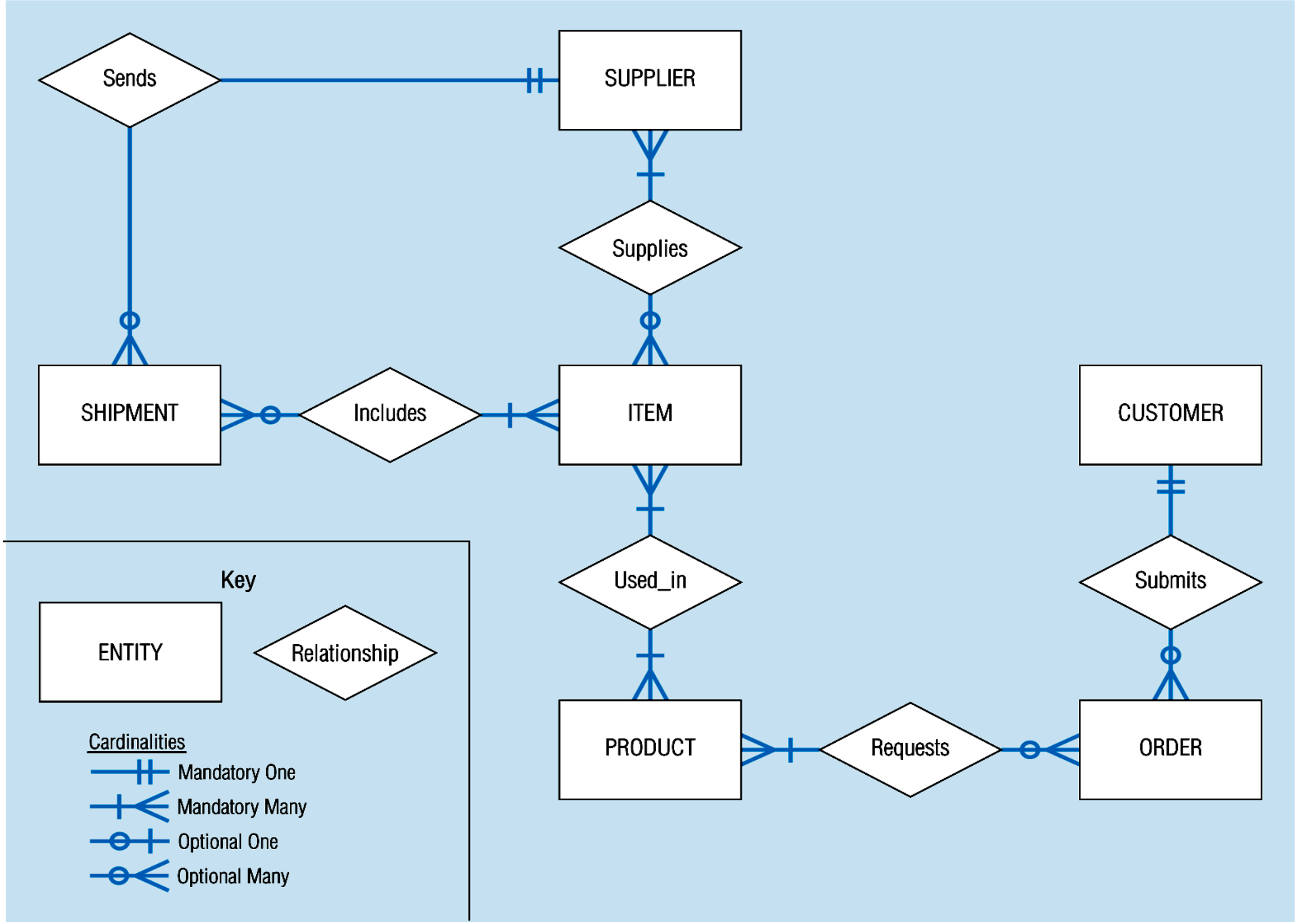
Relationship



E-R Model Constructs

- Degree of relationship
- Participation condition (mandatory or optional)
- Constraints and assumptions

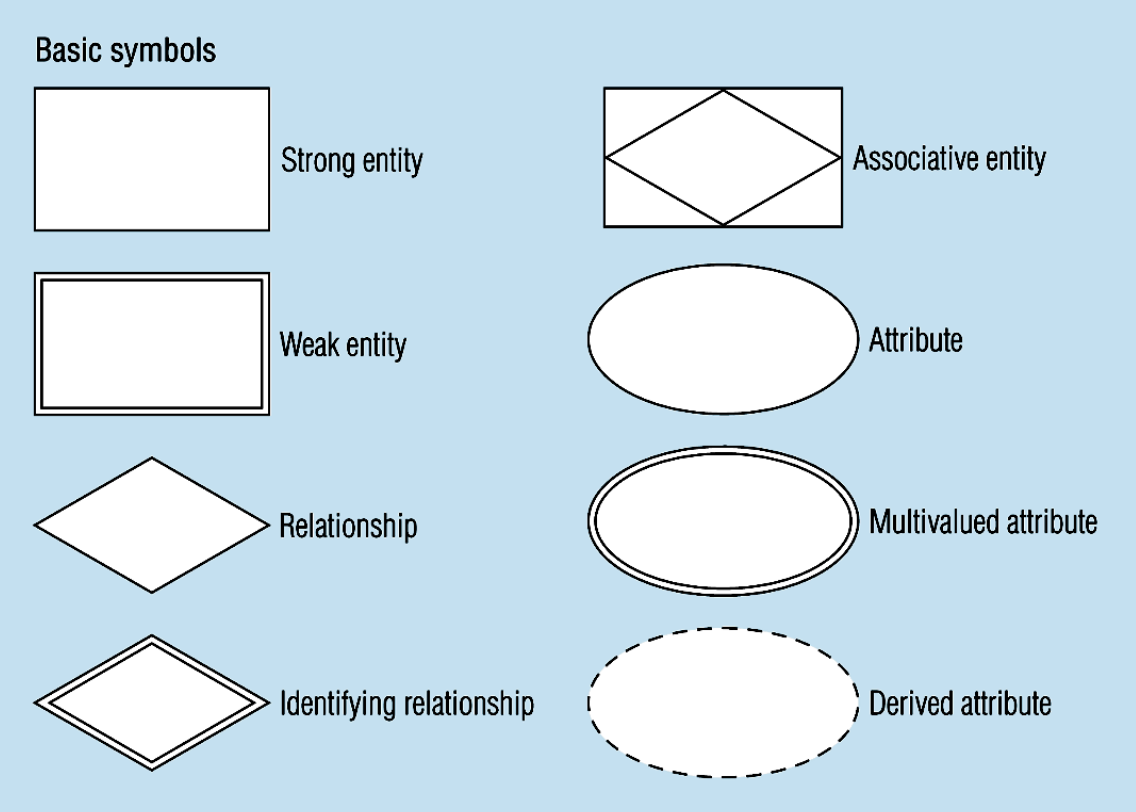
Sample E-R Diagram



Basic E-R Notation

Entity symbols

Relationship symbols



A special entity that is also a relationship

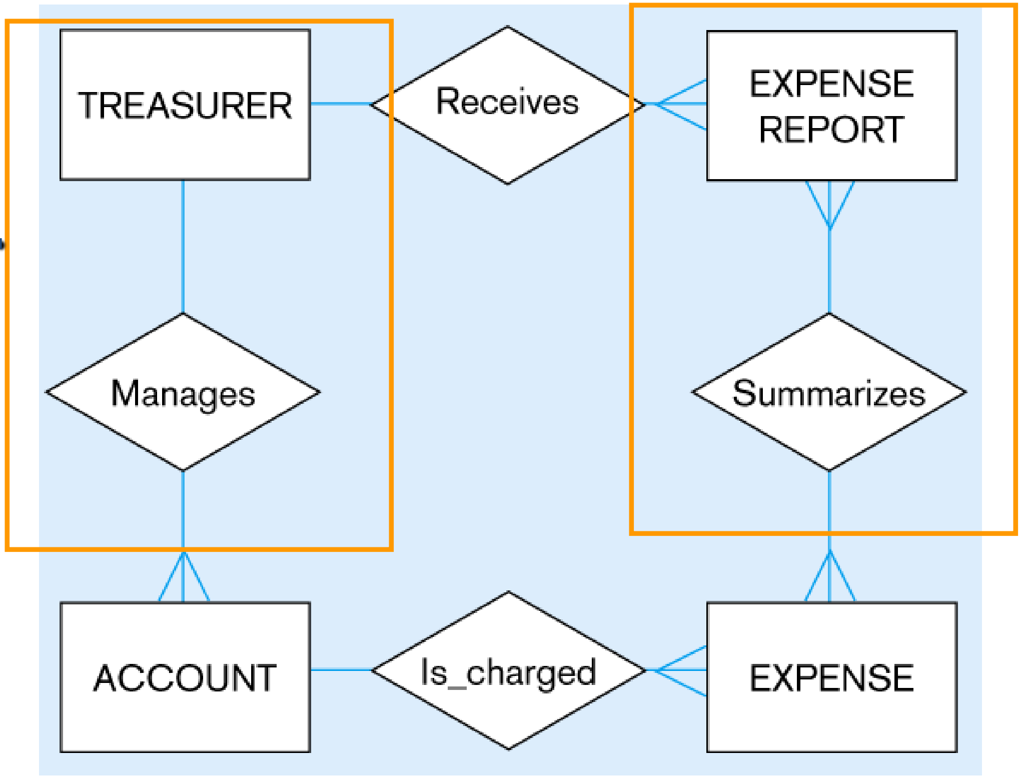
Attribute symbols

What Should an Entity Be?

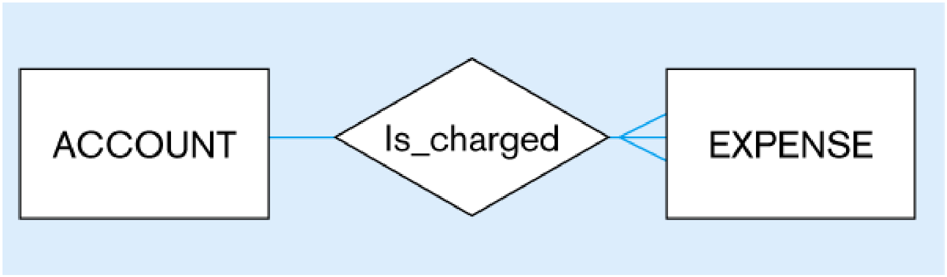
- SHOULD BE:
  - An object that will have many instances in the database
  - An object that will be composed of multiple attributes
  - An object that we are trying to model
- SHOULD NOT BE:
  - A user of the database system
  - An output of the database system (e.g. a report)

Inappropriate entities

System user



System output



Appropriate entities

Attributes

- Attribute - property or characteristic of an entity type
- Classifications of attributes:
  - Simple versus Composite Attribute
  - Single-Valued versus Multivalued Attribute
  - Stored versus Derived Attributes
  - Identifier Attributes

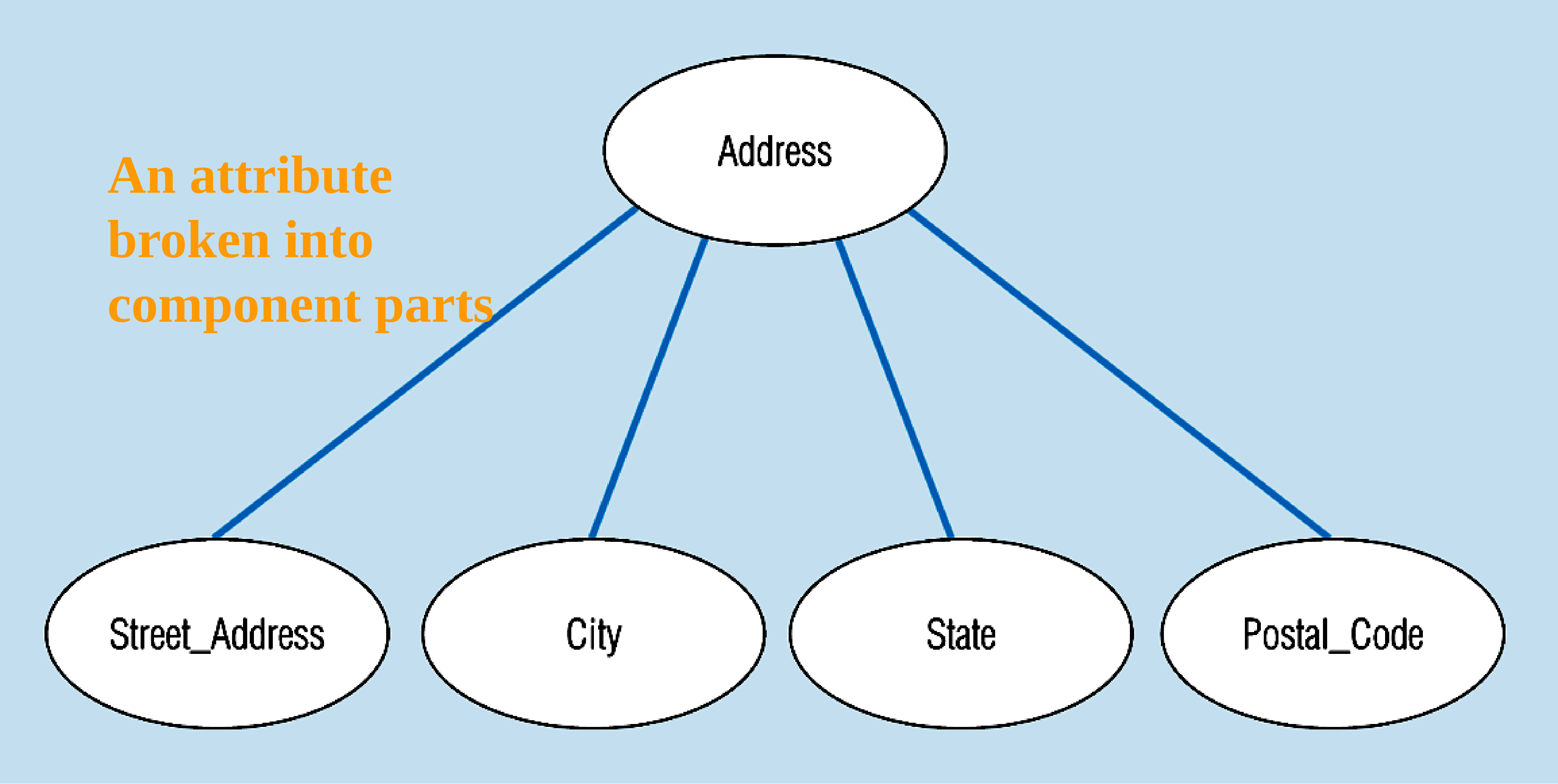
Identifiers (Keys)

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

Characteristics of Identifiers

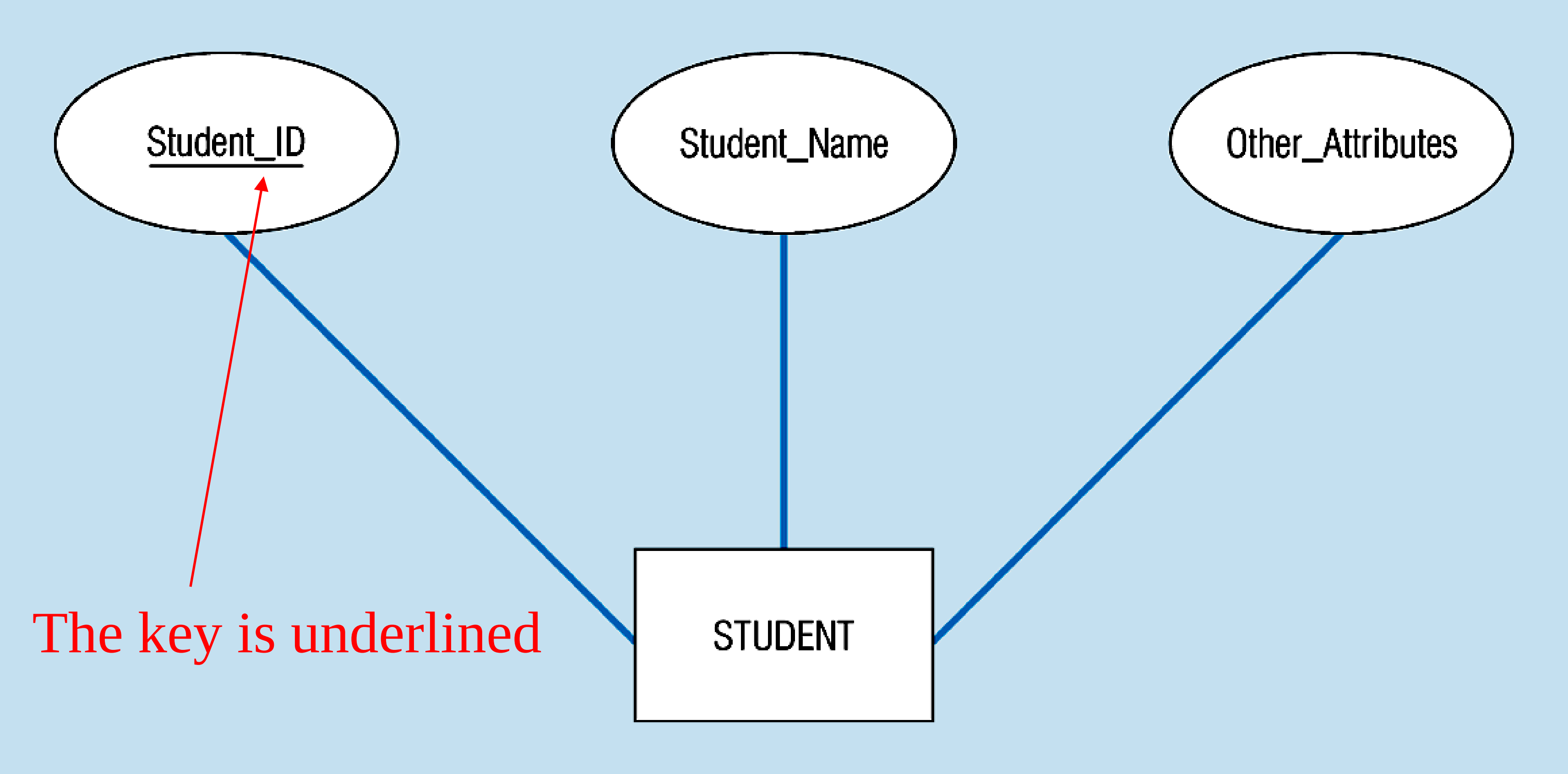
- Will not change in value
- Will not be null
- No intelligent identifiers (e.g. containing locations or people that might change)
- Substitute new, simple keys for long, composite keys

A **composite** attribute



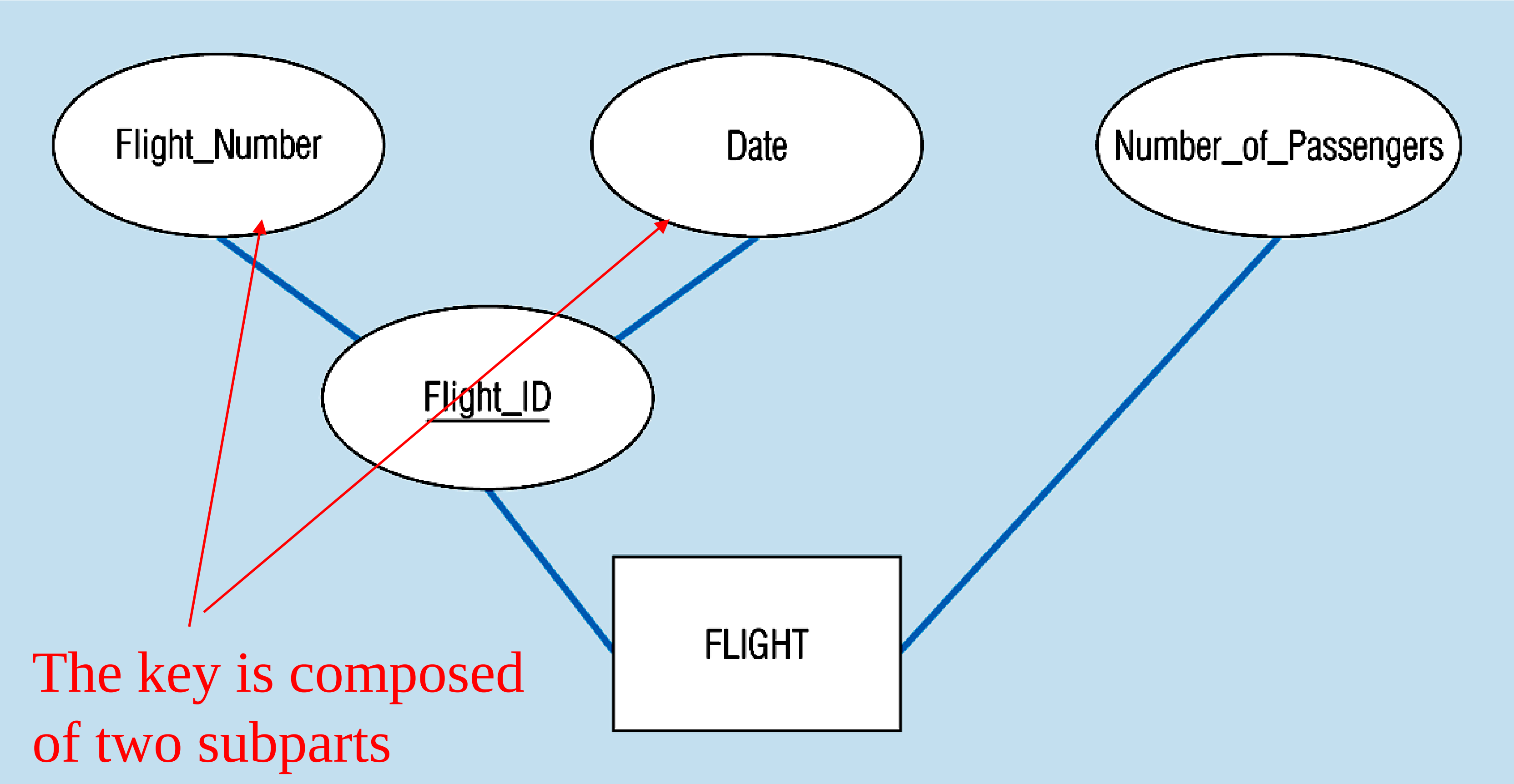
An attribute broken into component parts

Simple key attribute



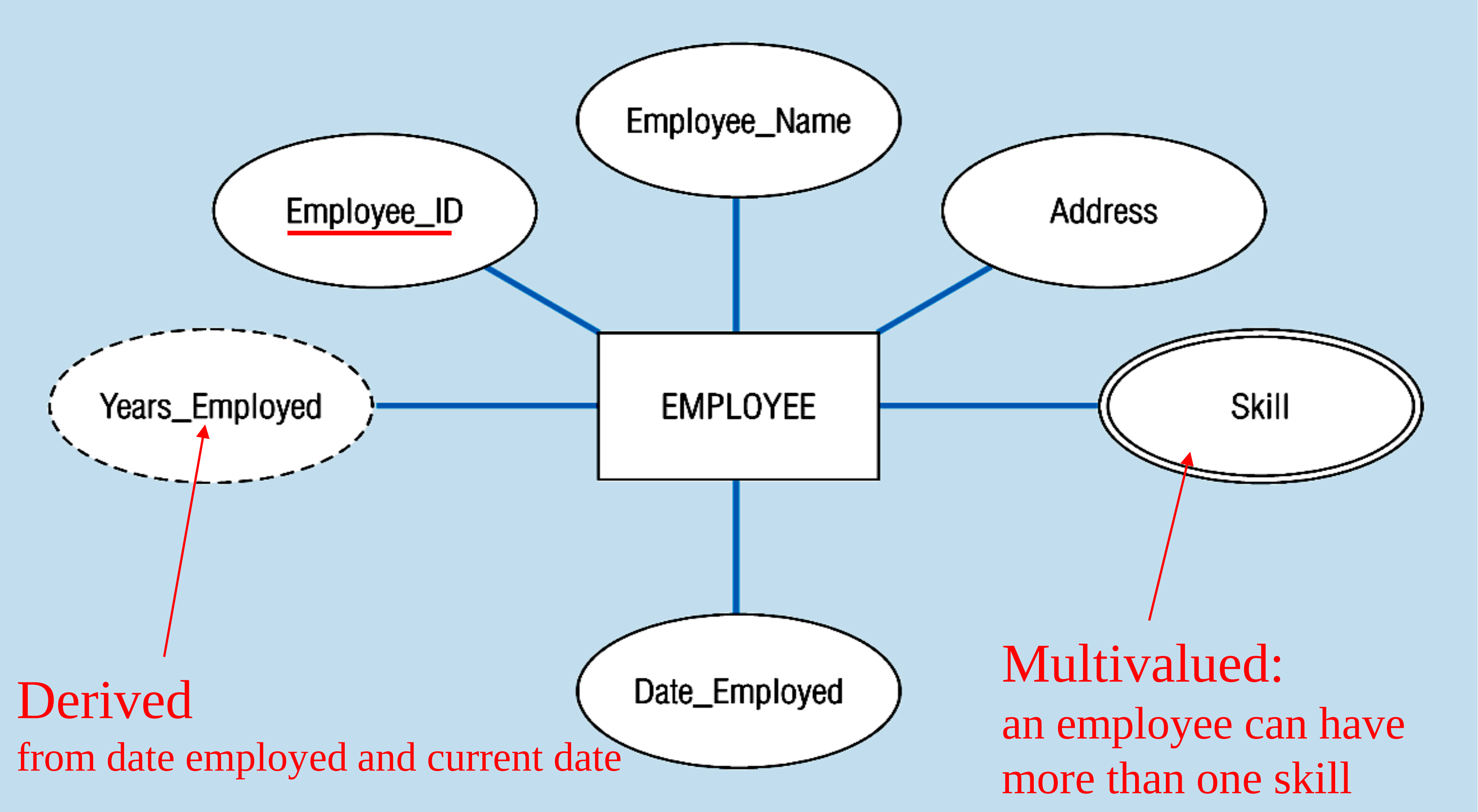
The key is underlined

Composite key attribute



The key is composed of two subparts

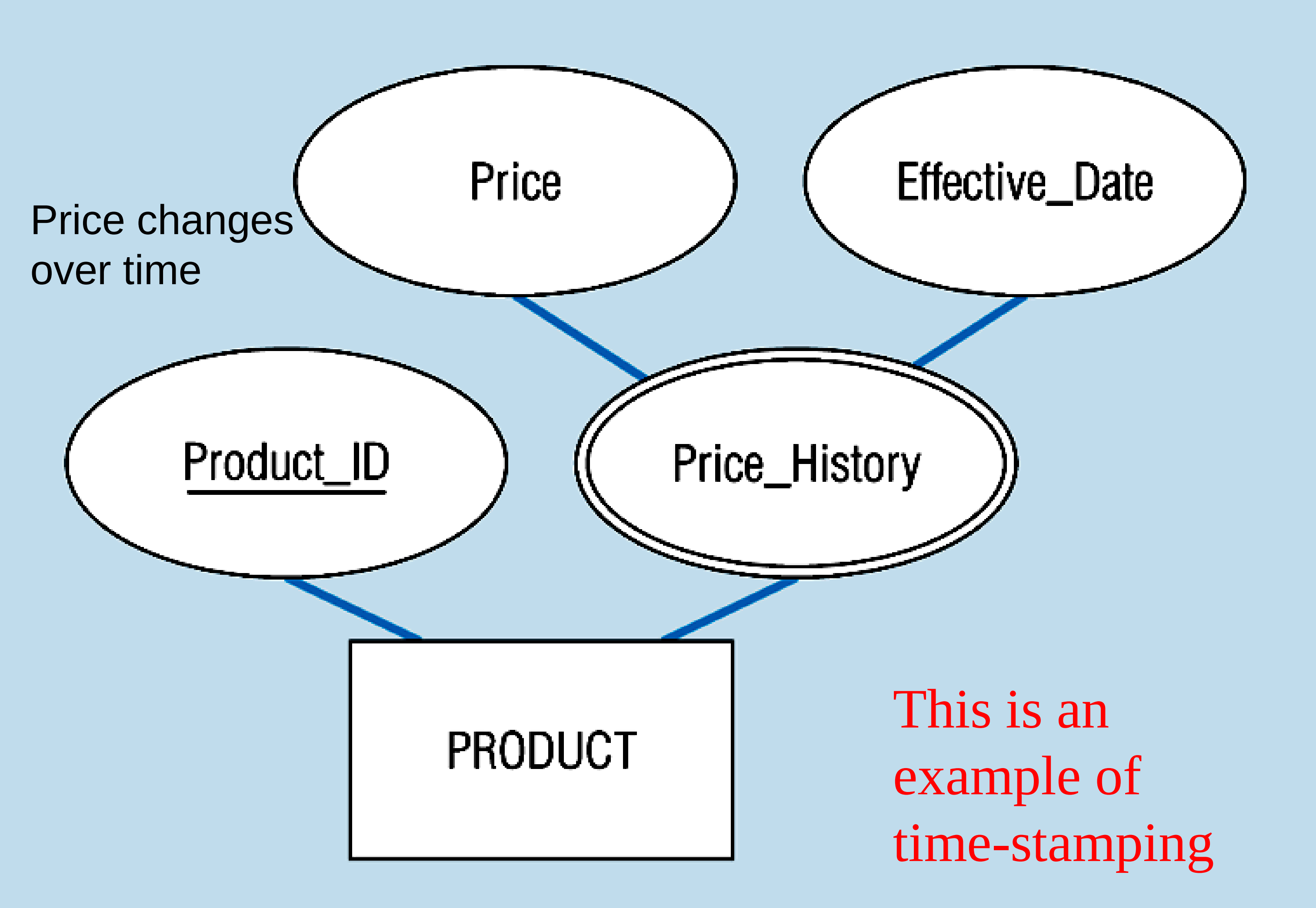
Entity with a multivalued attribute (Skill) and derived attribute (Years_Employed)



Derived
from date employed and current date

Multivalued:
an employee can have more than one skill

An attribute that is both multivalued and composite



Price changes over time

This is an example of time-stamping

More on Relationships

- Relationship Types vs. Relationship Instances
  - The relationship type is modeled as the diamond and 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

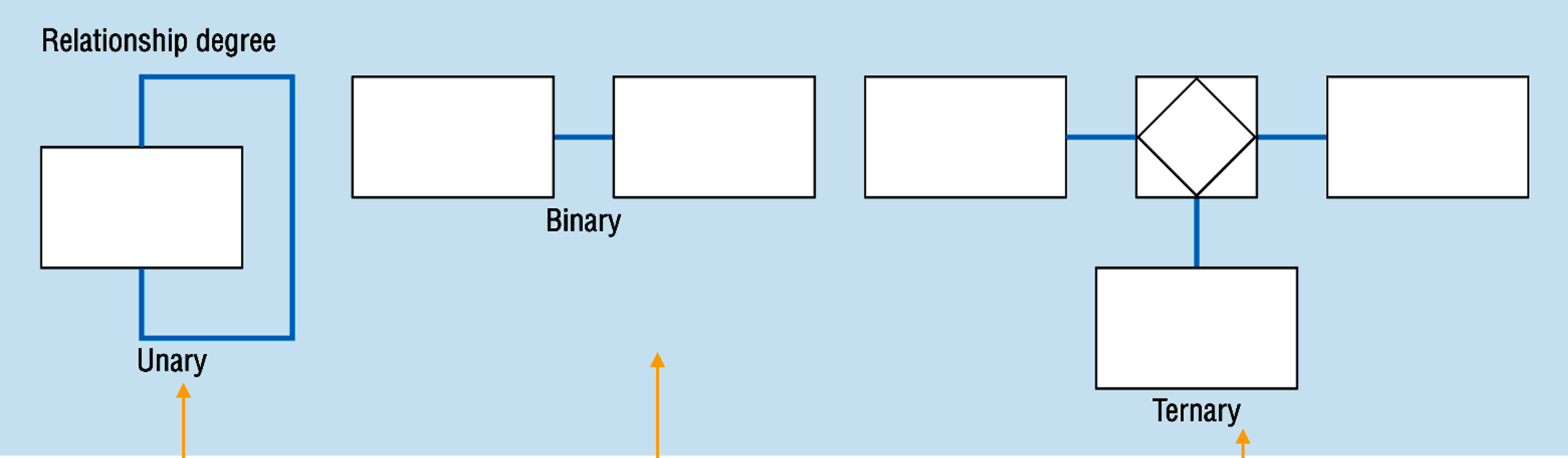
More on Relationships

- Two entities can have more than one type of relationship between them (multiple relationships)
- Associative Entity = combination of relationship and entity
  - More on this later

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

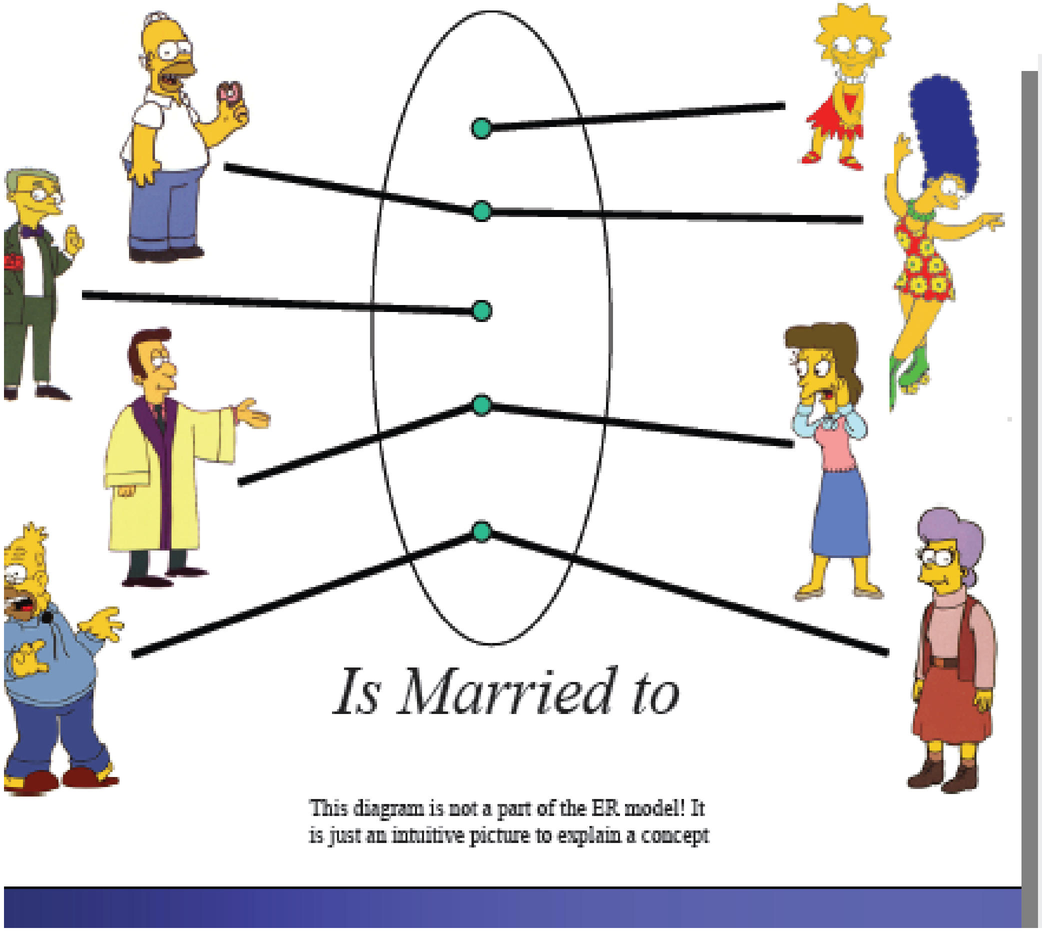
Entities of two different types related to each other

Entities of three different types related to each other

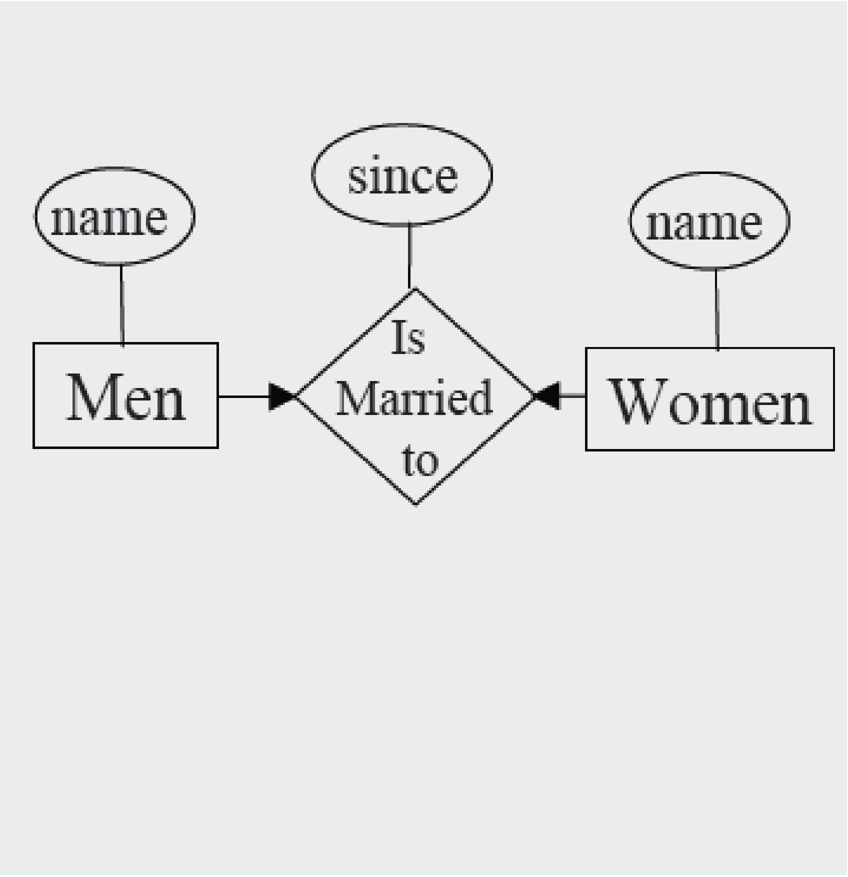
Cardinality of Relationships

- 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 – one
  - An entity in A is associated with at most one entity in B. An entity in B is associated with any number in A
- Many – to – Many
  - Entities on both sides of the relationship can have many related entities on the other side

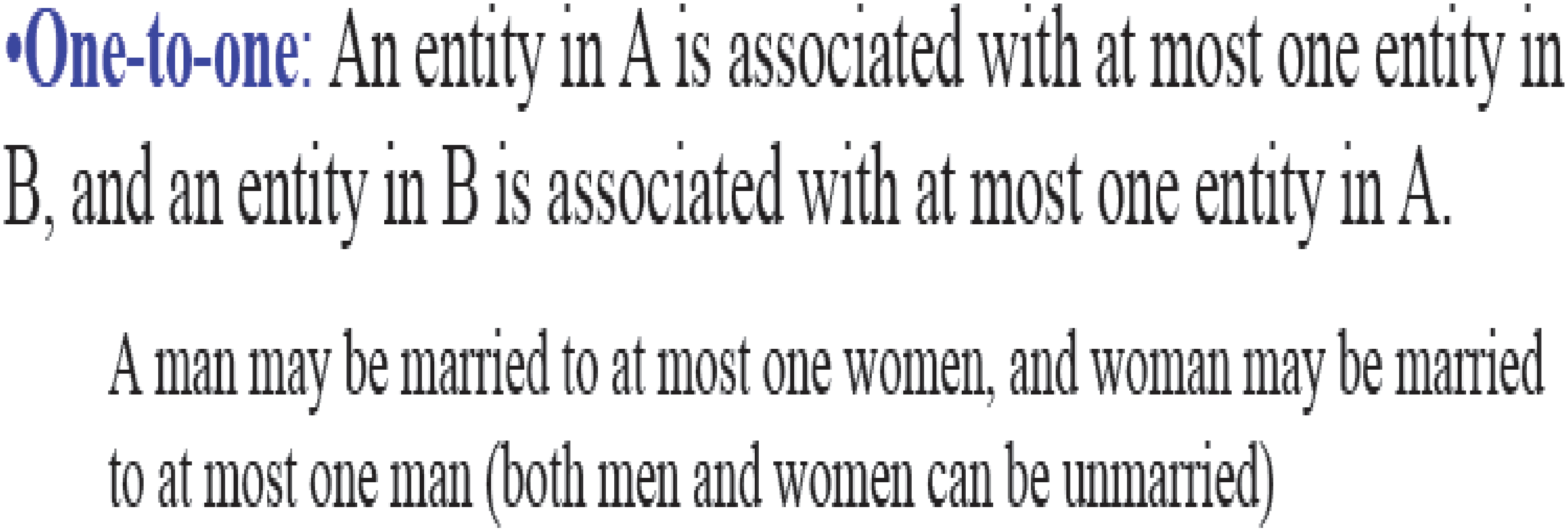
One – to – One
Key Constraint example



This diagram is not a part of the ER model! It is just an intuitive picture to explain a concept

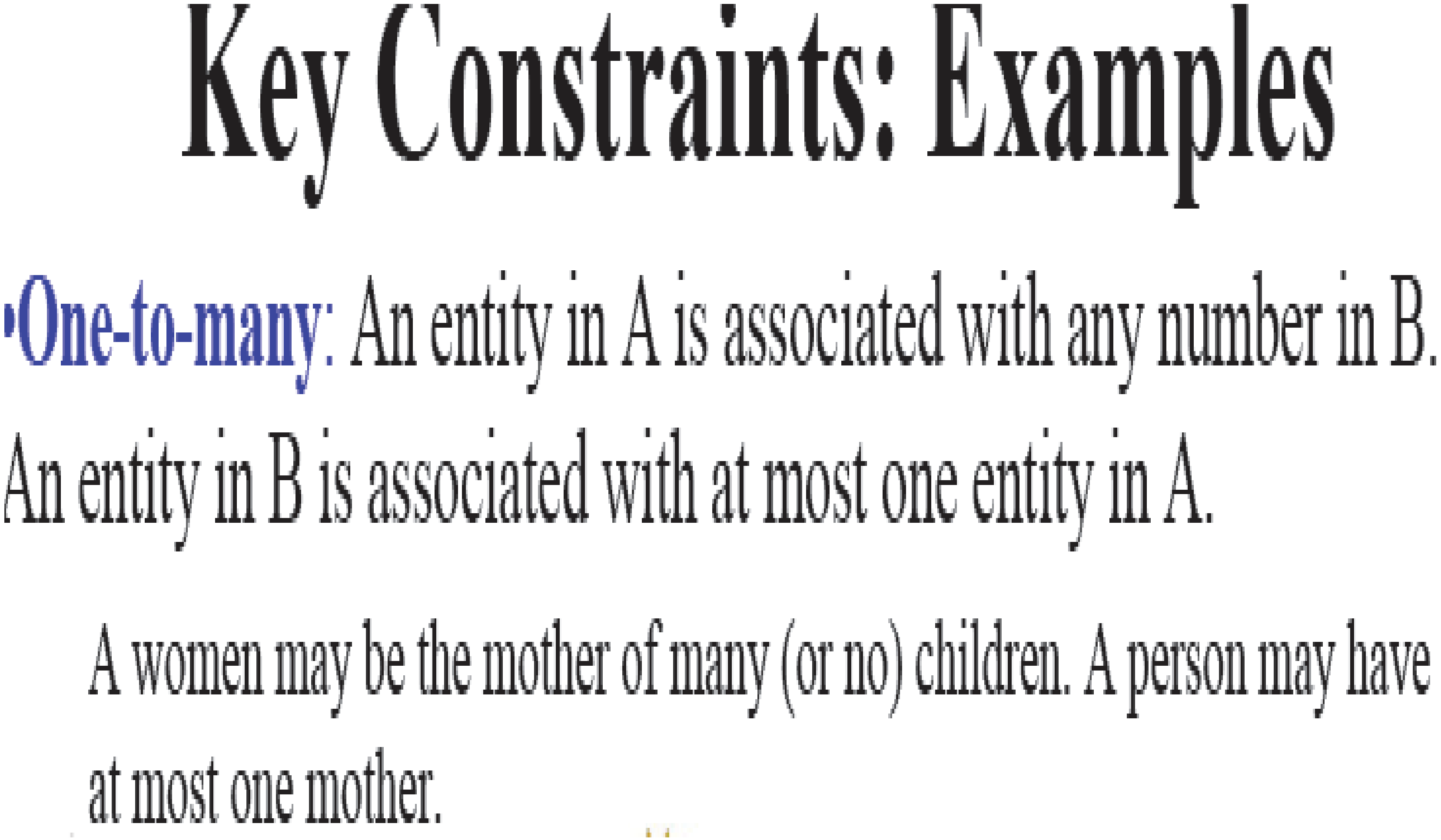


Key Constraint example



- **One-to-one:** An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.

A man may be married to at most one women, and woman may be married to at most one man (both men and women can be unmarried)

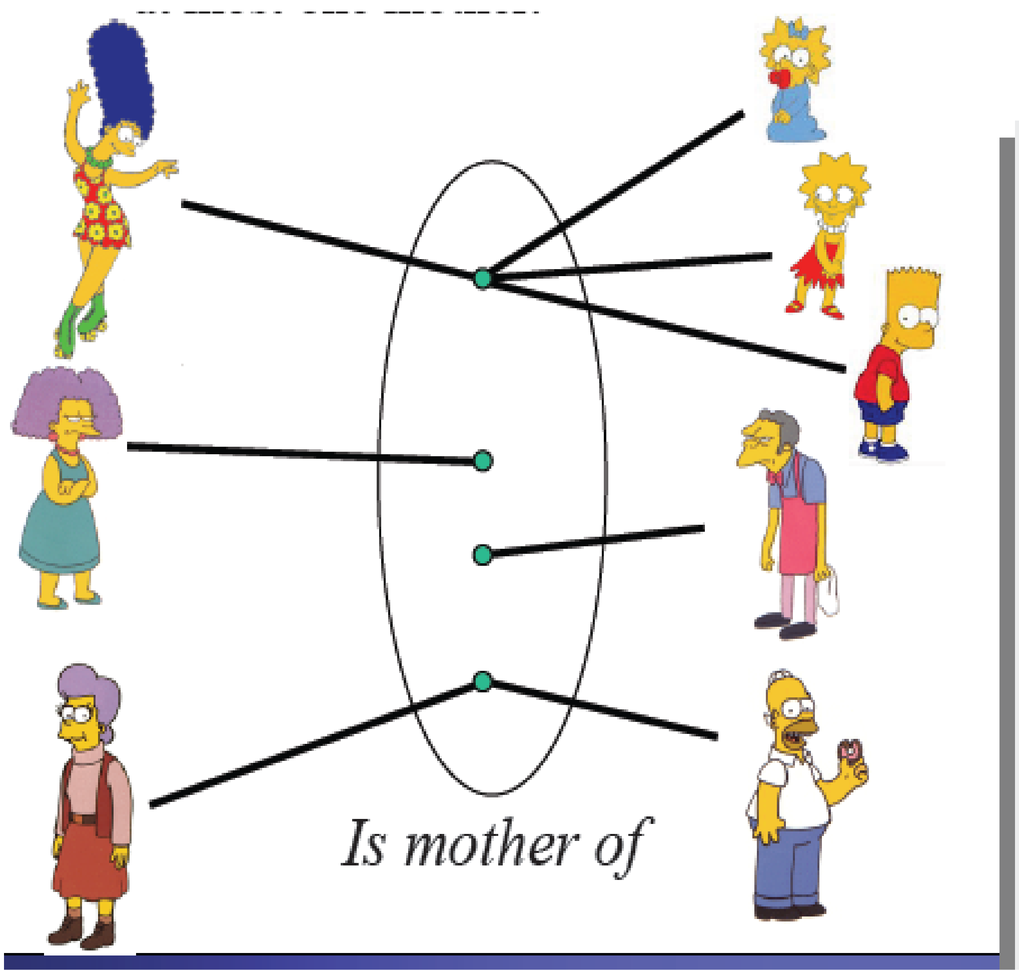


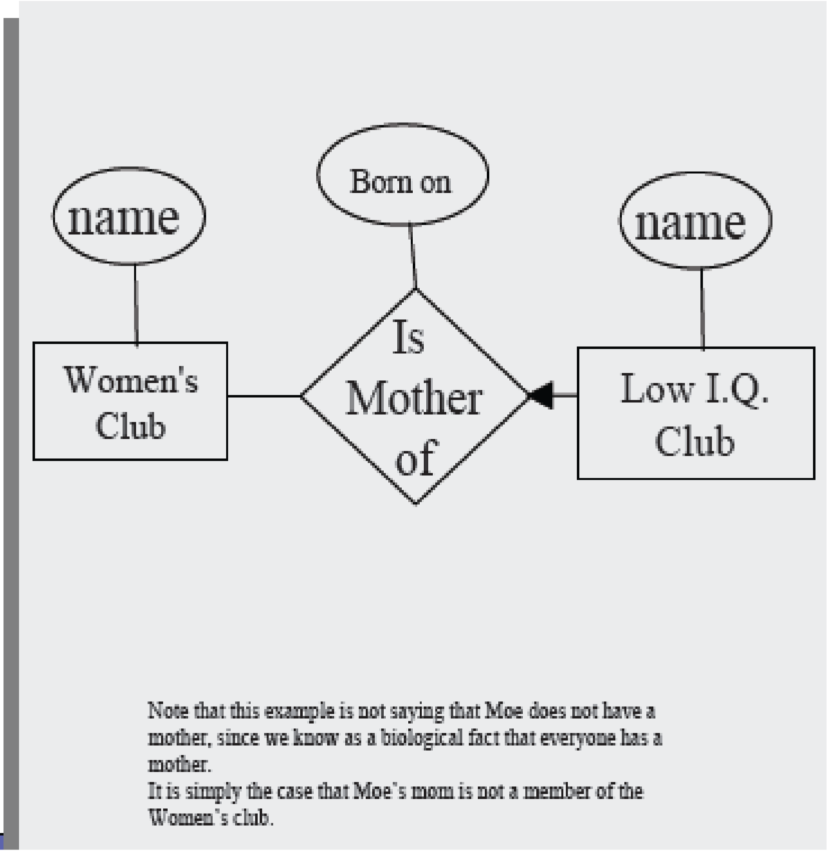
Key Constraints: Examples

- **One-to-many:** An entity in A is associated with any number in B. An entity in B is associated with at most one entity in A.

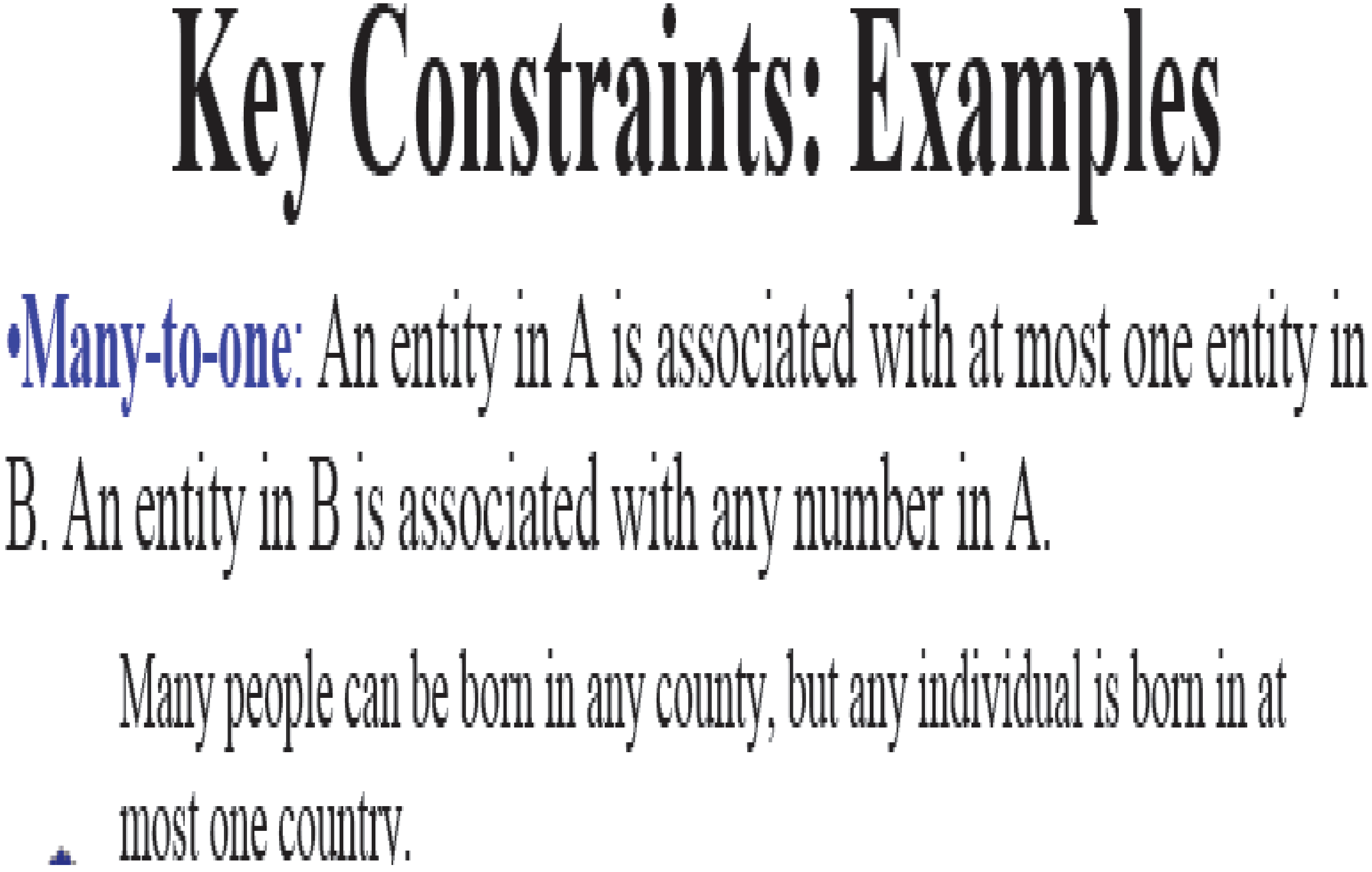
A women may be the mother of many (or no) children. A person may have at most one mother.

Key Constraint example





Note that this example is not saying that Moe does not have a mother, since we know as a biological fact that everyone has a mother.
It is simply the case that Moe's mom is not a member of the Women's club.

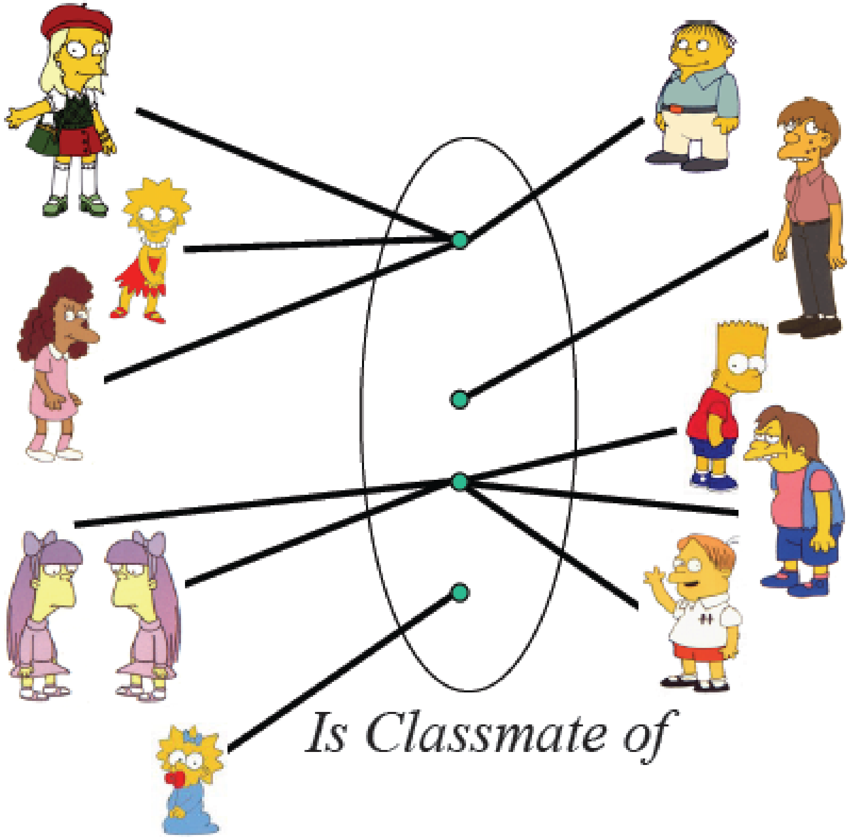


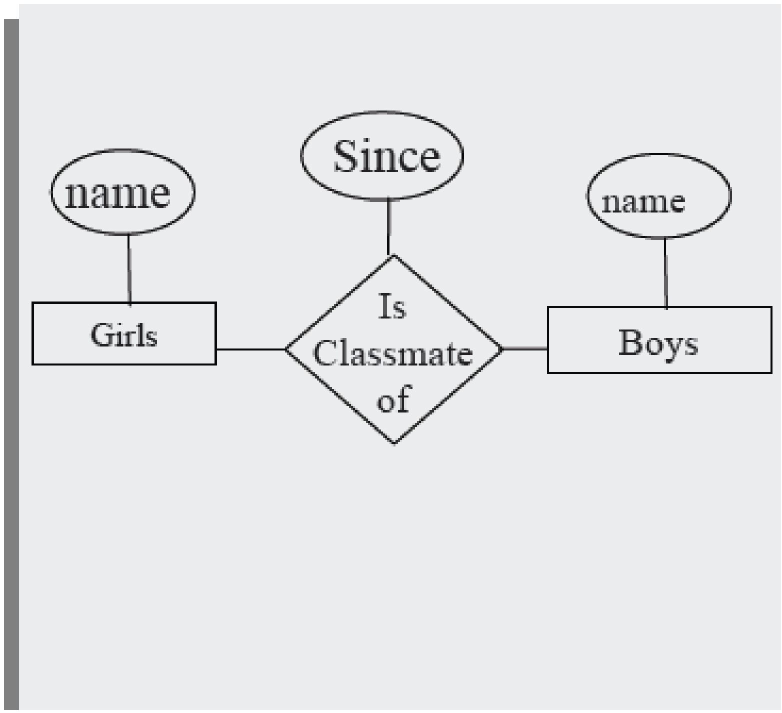
Key Constraints: Examples

- **Many-to-one**: An entity in A is associated with at most one entity in B. An entity in B is associated with any number in A.

   - Many people can be born in any county, but any individual is born in at most one country.

Key Constraint Example
Many – to - Many

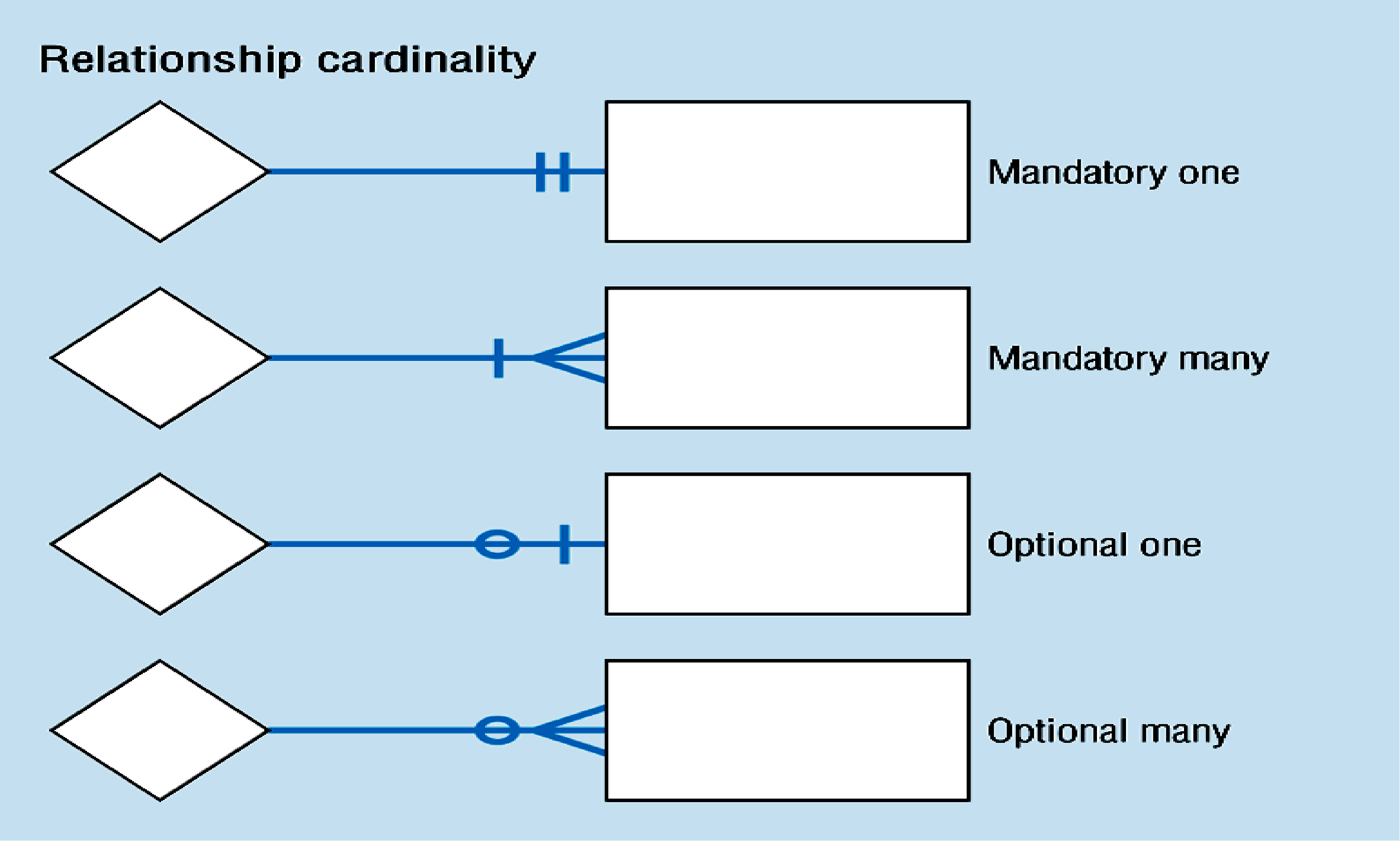




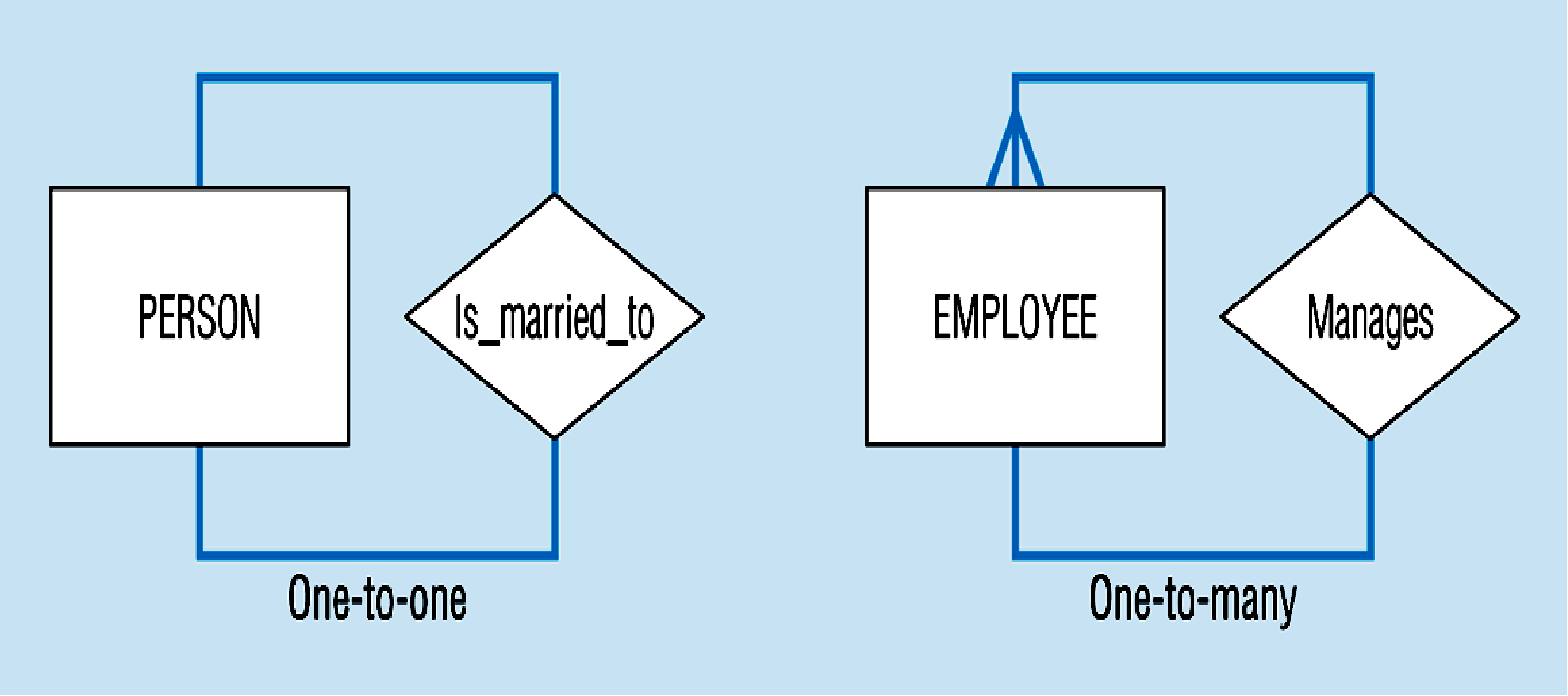
Cardinality Constraints

- 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

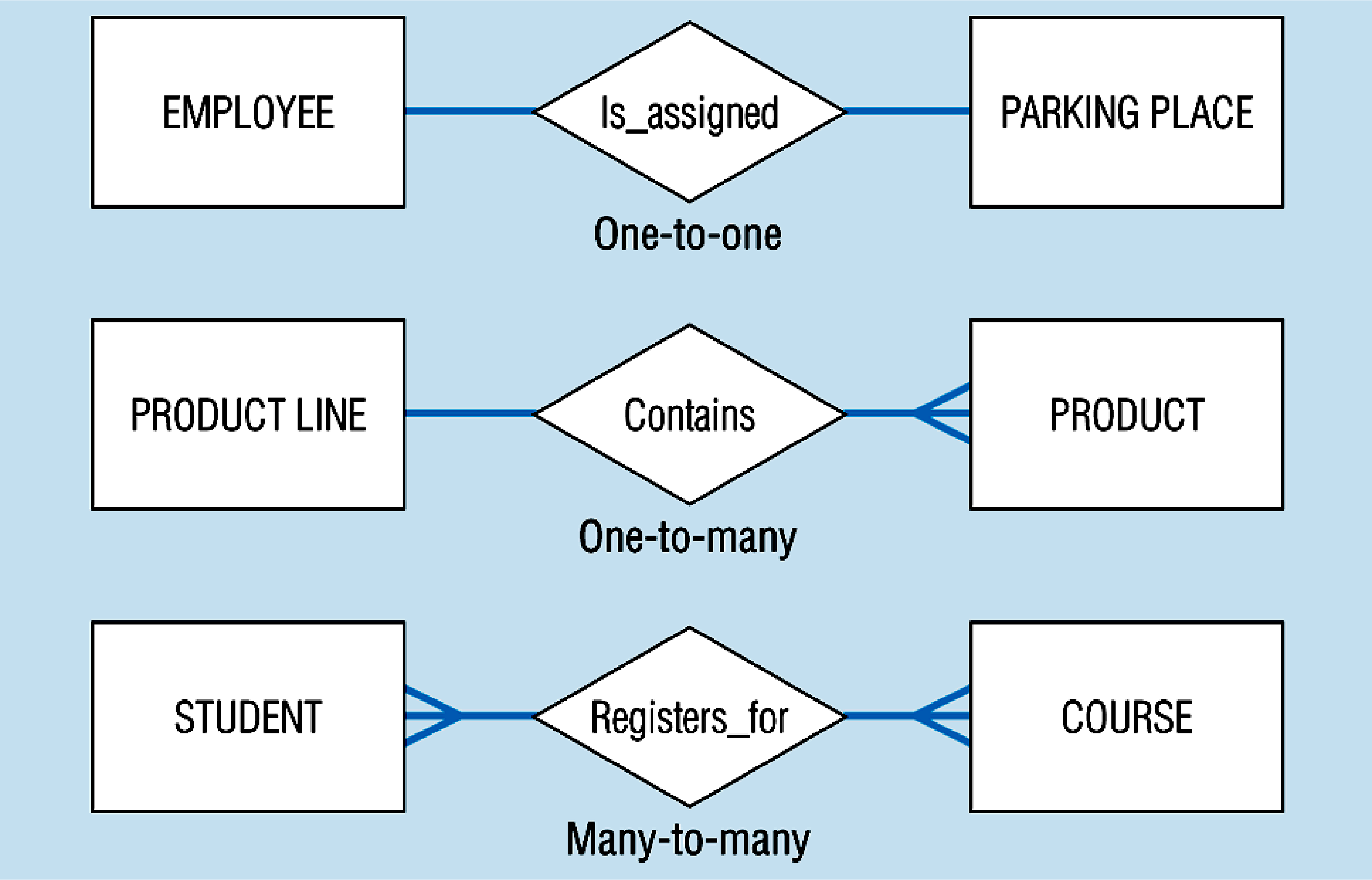
Cardinality



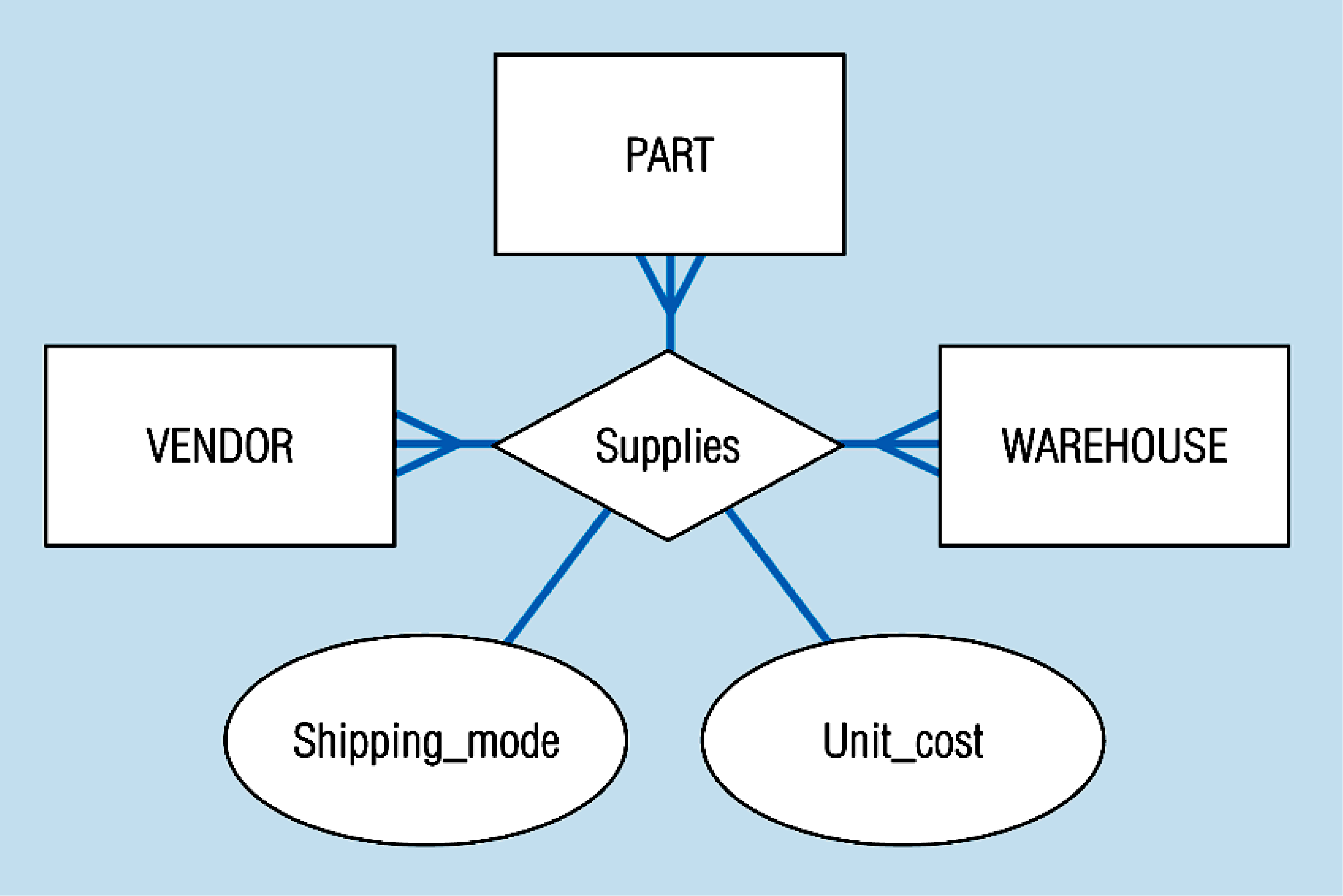
Unary relationships



Binary relationships

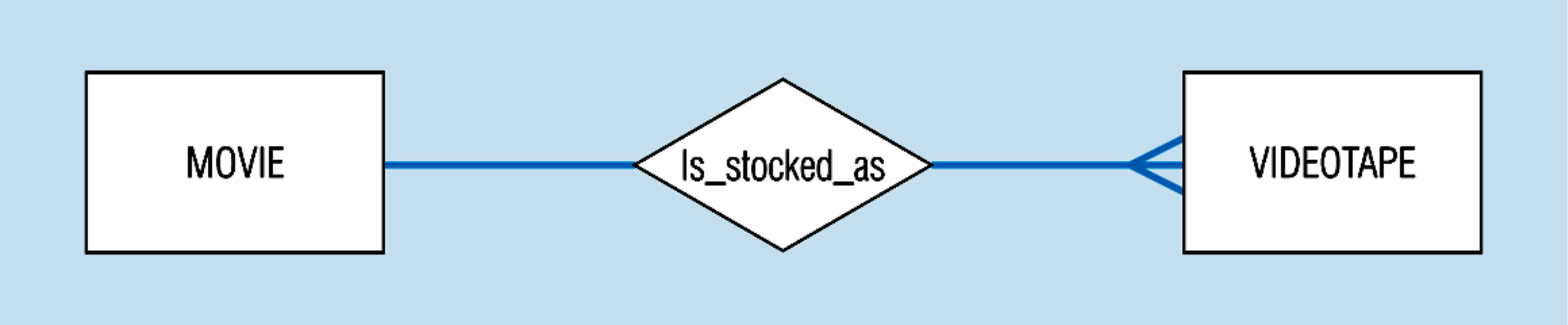


Ternary relationships

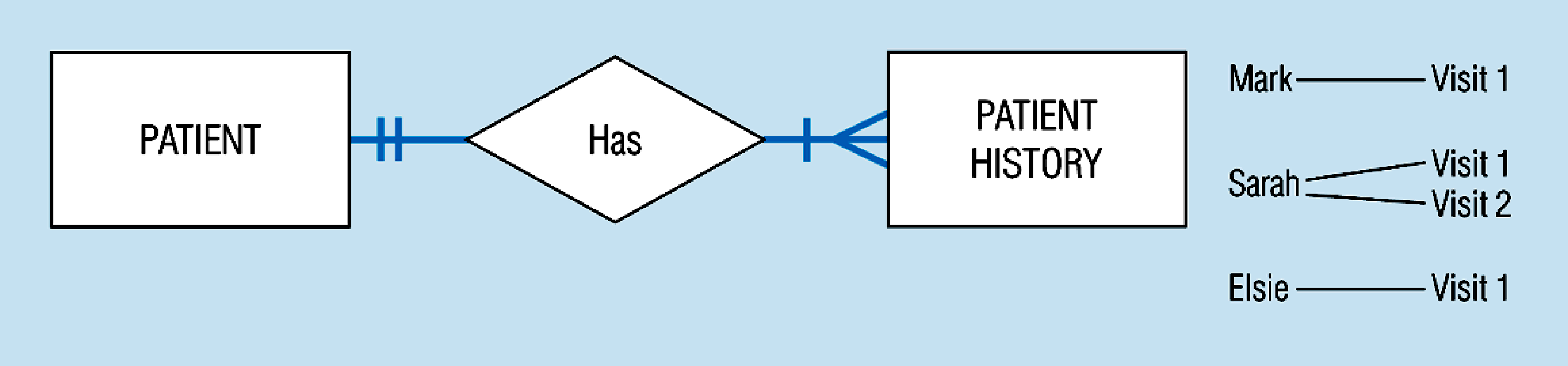


Note: a relationship can have attributes of its own

Basic relationship with only maximum cardinalities



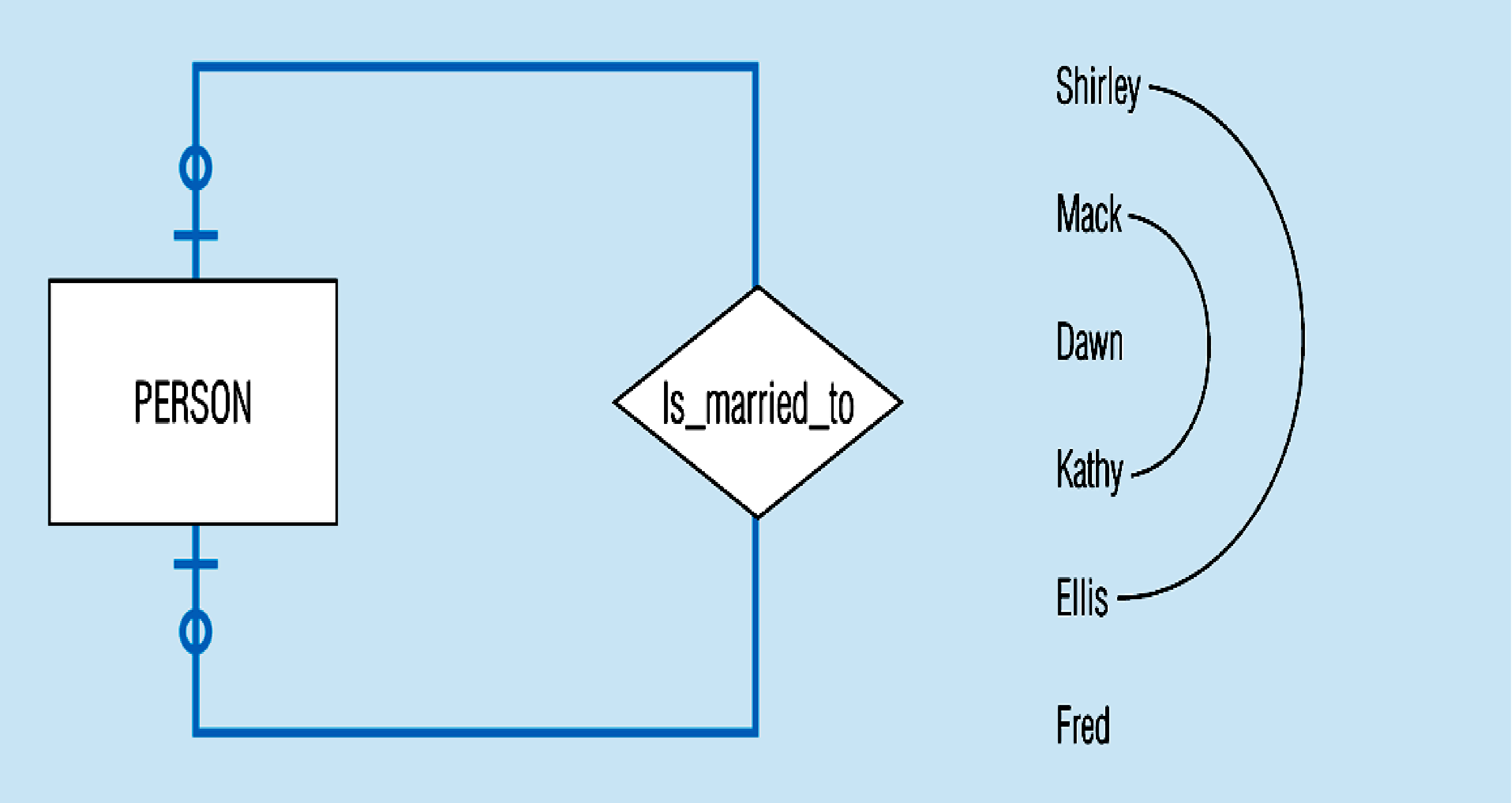
Mandatory minimum cardinalities



Patient must have one or more patient history (Mandatory many:1)

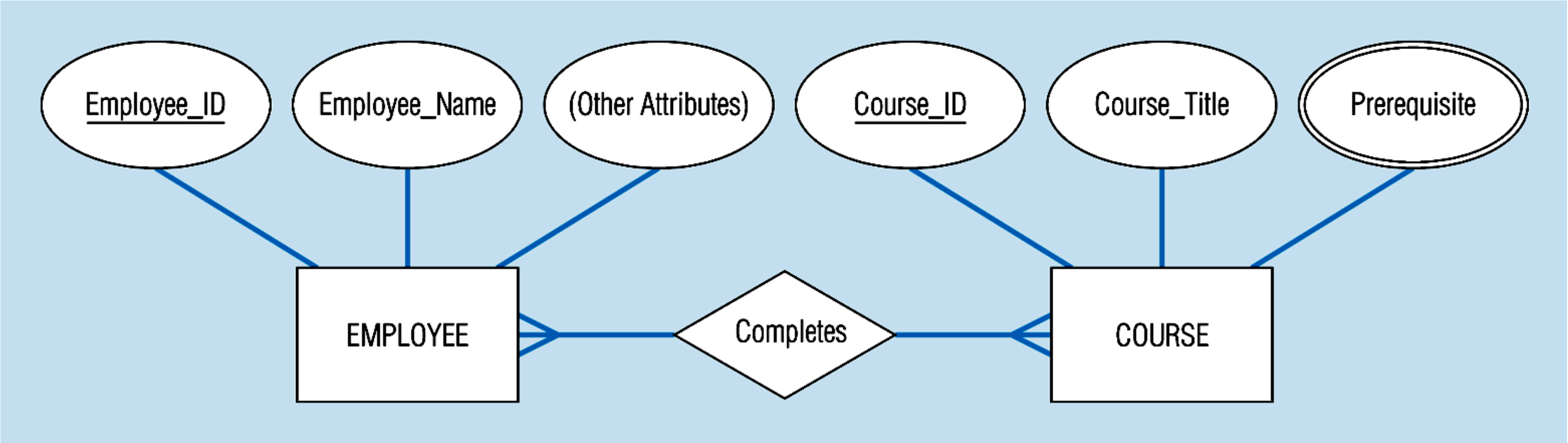
Patient history must be exactly for one patient (mandatory 1:1)

Optional cardinalities with unary degree, one-to-one relationship

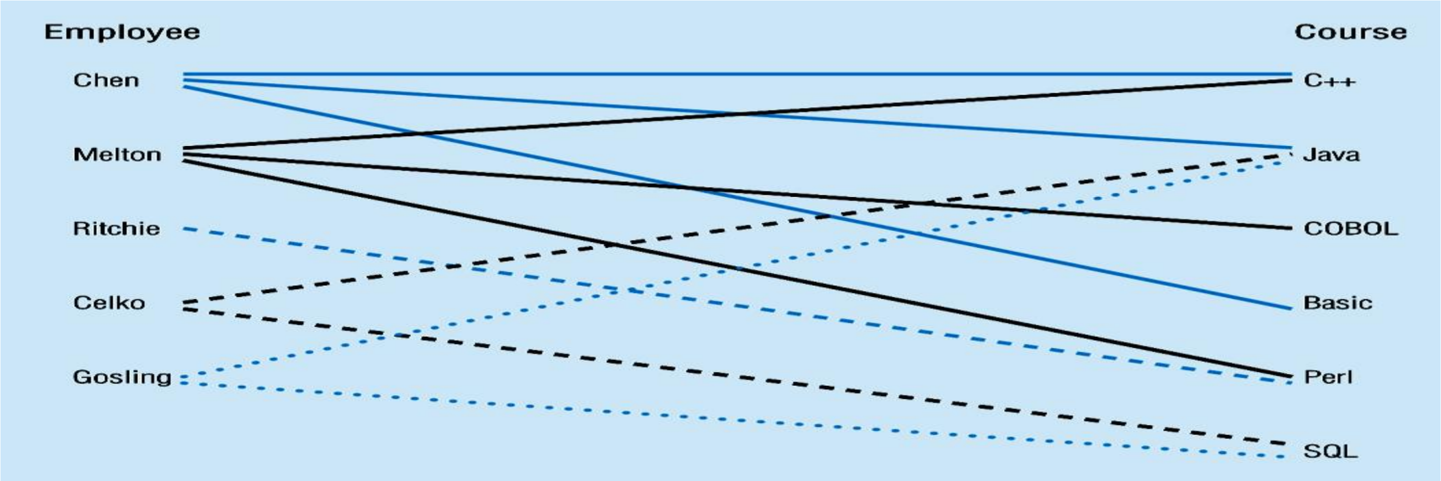


This is an optional zero or one cardinality in both directions since a person may or may not be married

Relationship type

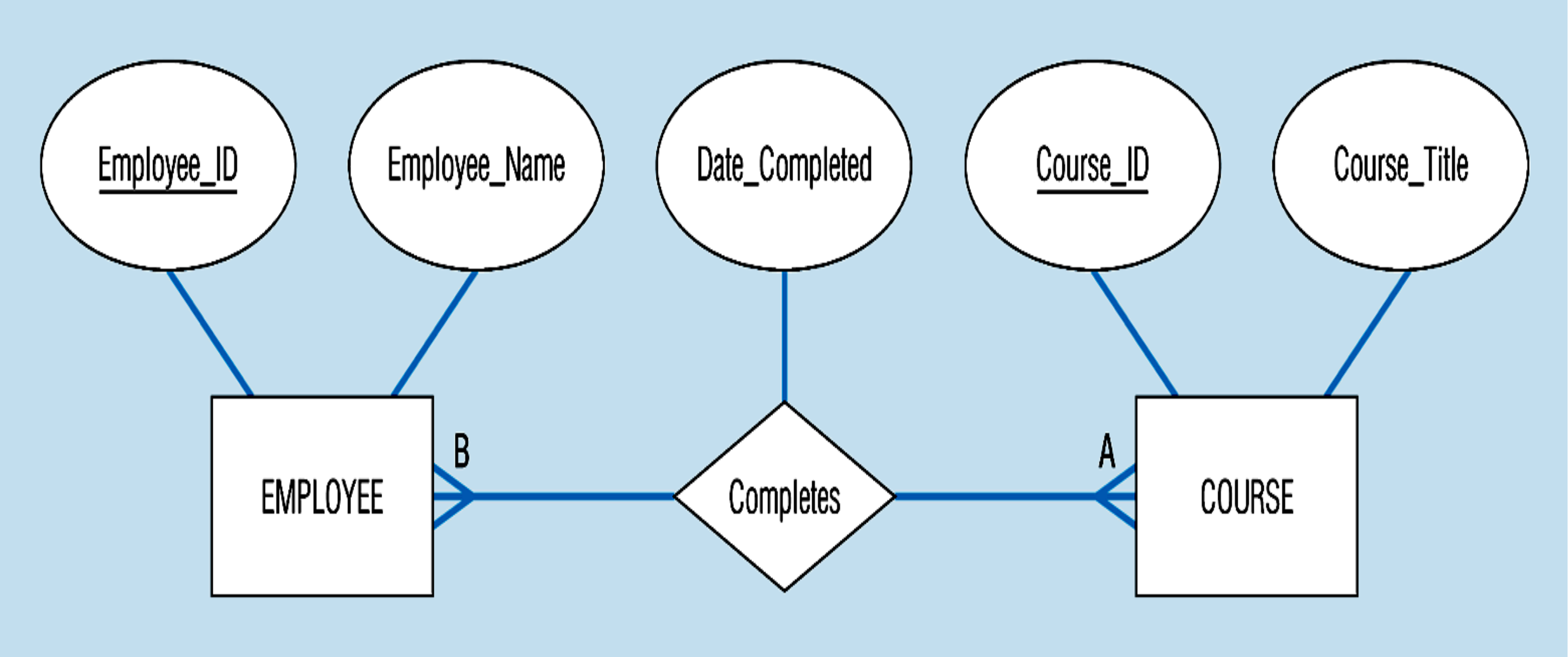


Entity and Relationship instances



A binary relationship with an attribute

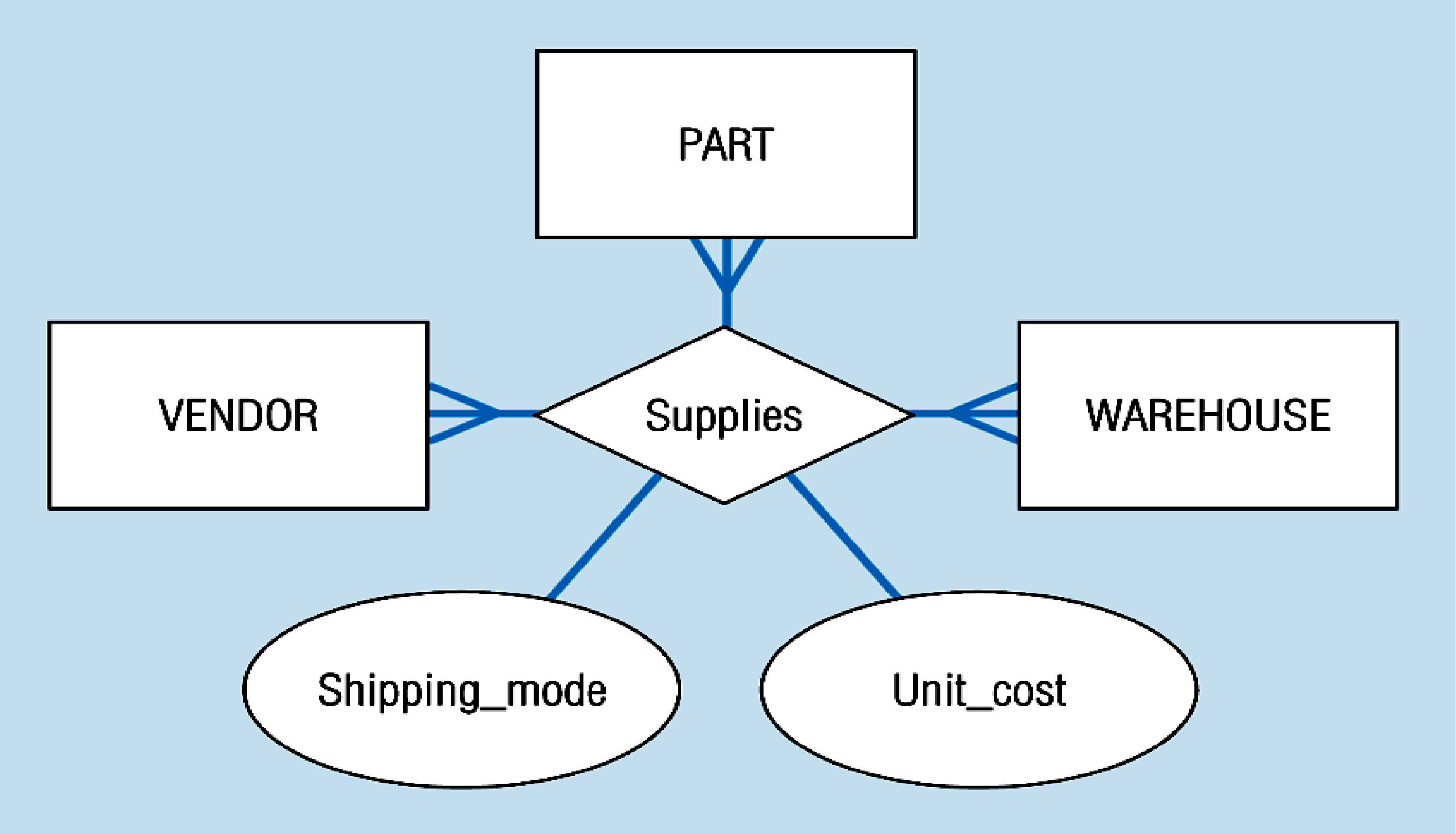
Here, the date completed attribute pertains specifically to the employee’s completion of a course…it is an attribute of the *relationship*



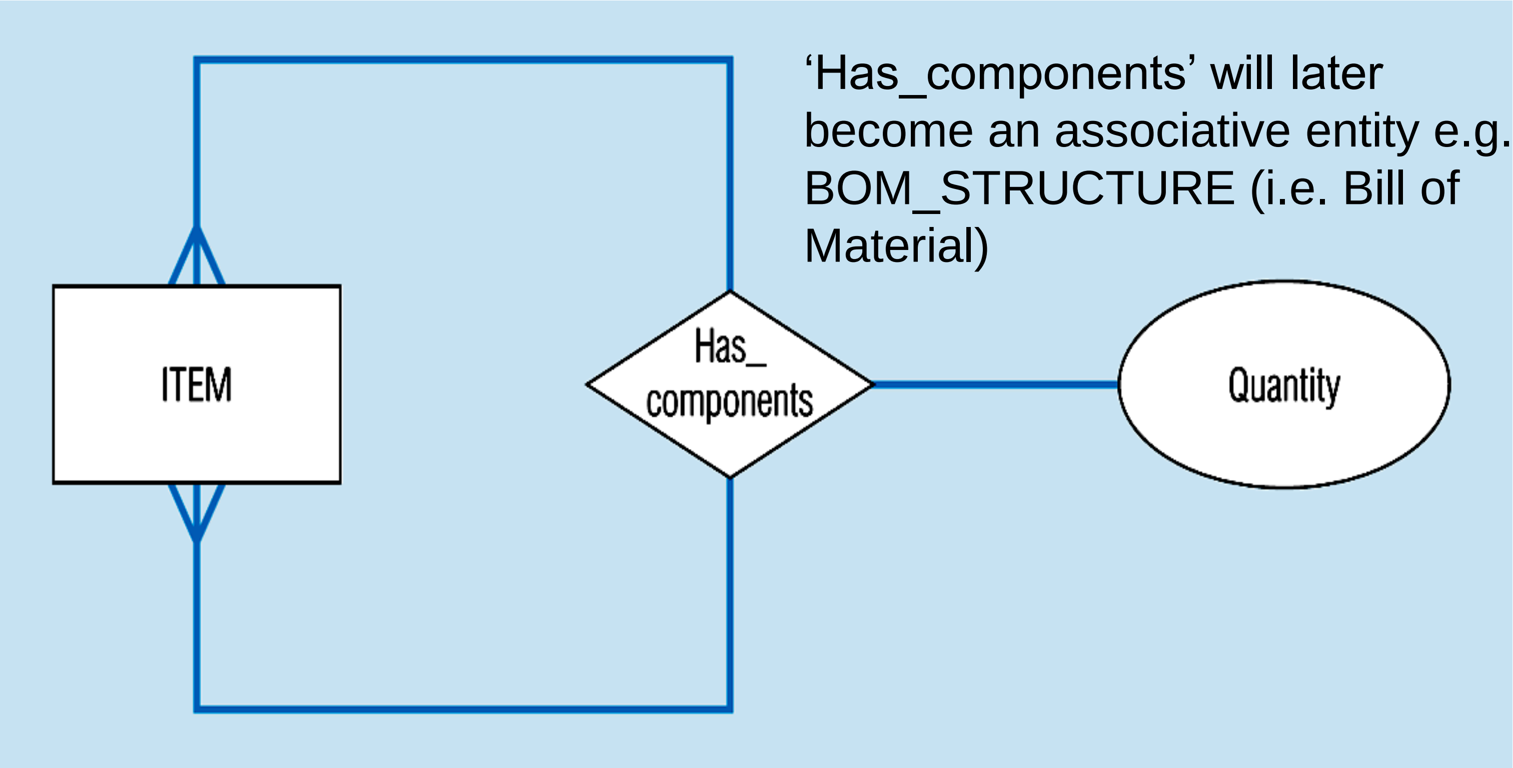
The presence of one or more attributes on a relationship suggests to the designer a many : many relationship. The relationship should instead be represented as an ‘Associative entity’ e.g.. Certificate with Certificate_Number & Date_Completed

A ternary relationship with attributes

Vendors can supply various parts to warehouses. Two attributes on supplies relationship. Thus it will be later converted into an associative entity e.g. Supply_Schedule



A unary relationship with an attribute. This has a many-to-many relationship

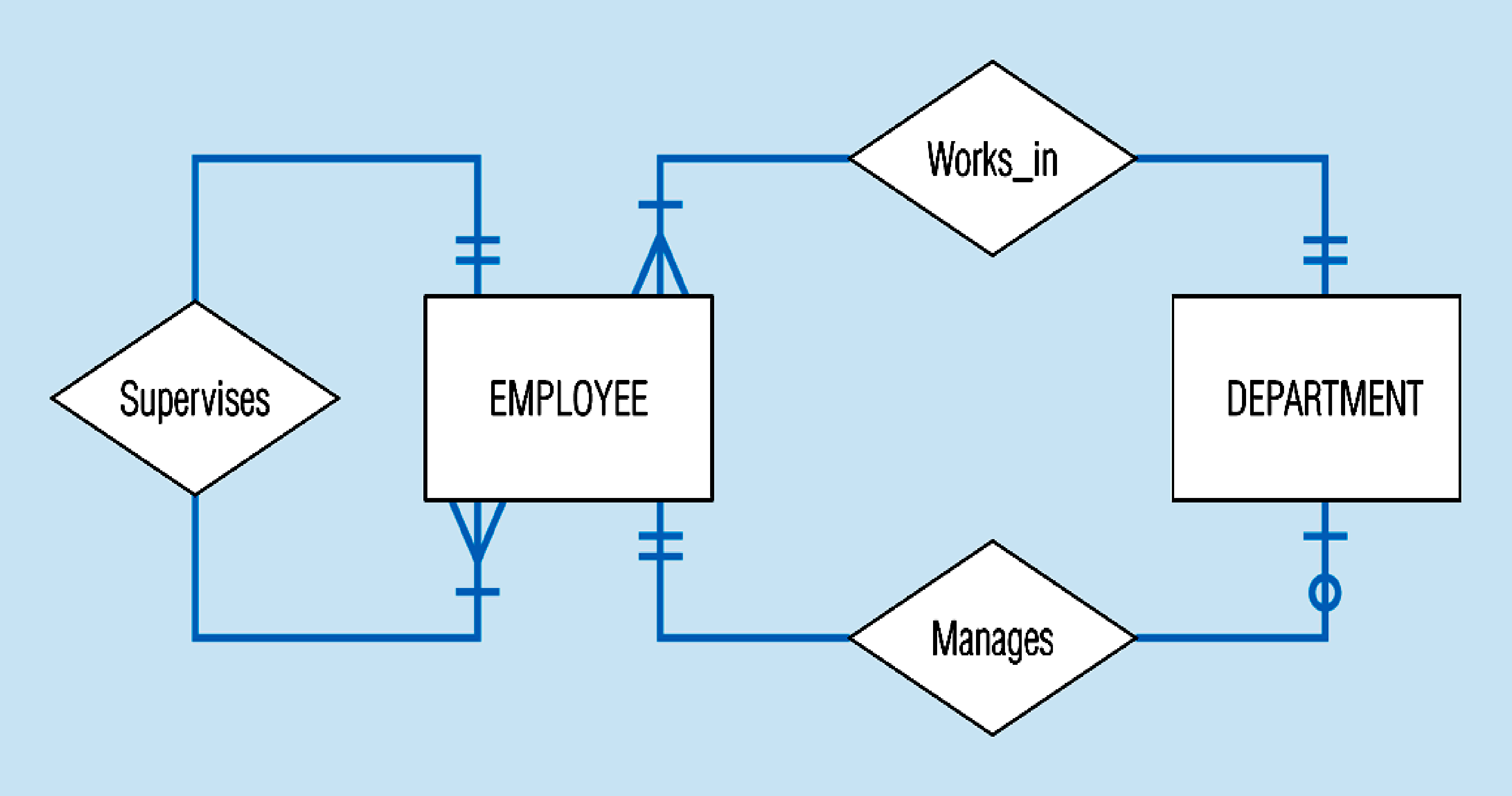


‘Has_components’ will later become an associative entity e.g. BOM_STRUCTURE (i.e. Bill of Material)

Representing a bill-of -materials structure

Examples of multiple relationships – entities can be related to one another in more than one way

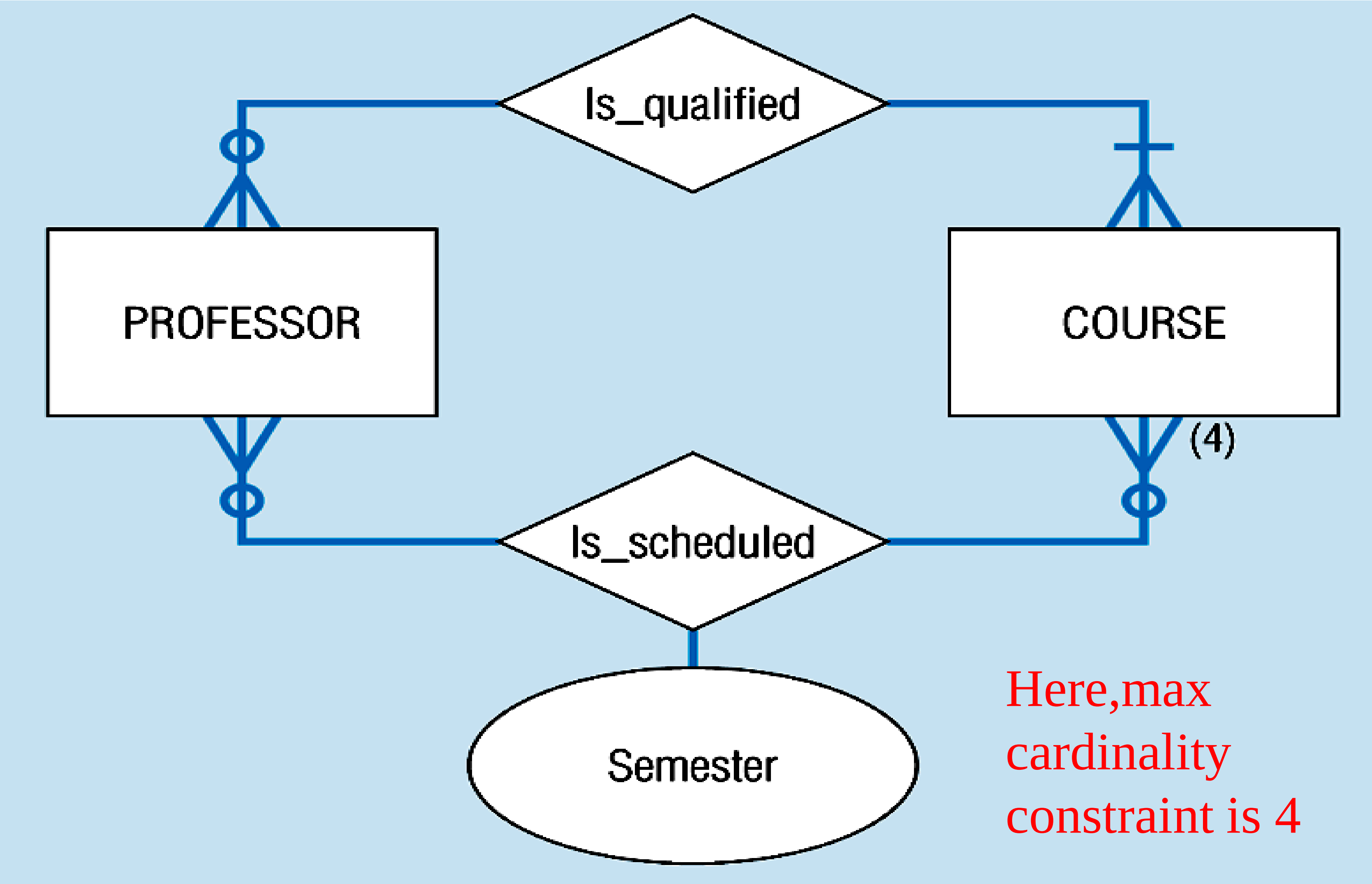
Employees and departments



Multiple relationships

- ‘Works_In’ is 1:many and mandatory in both directions
- Department must have at least one employee
- And each employee must be assigned to exactly one department
- ‘Manages’: From Department to Employee it is mandatory one i.e. a Department must have exactly one manager
- From Employee to Department it is optional one since a given Employee either is or is not a department manager

Professors and courses (fixed upon constraint)



Here,max cardinality constraint is 4

Multiple relationships

- A given course may have more than one qualified instructor or optionally may not have any qualified instructor (In the case the course is new & just entered in the catalogue)
- Each instructor must be qualified to teach at least one course
- Note maximum cardinality for courses scheduled for an instructor during semester is four. This is how you can record a fixed constraint.

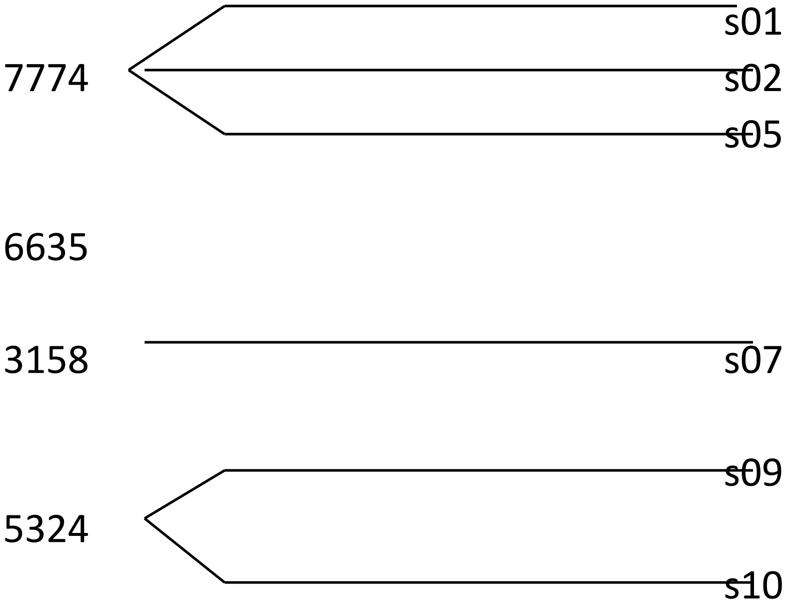
Occurrence diagram

- A relationship occurrence is a specific set of association that exist at a given time. These can be captured in an occurrence diagram

Example of occurrence diagram

- E.g. Staff counseling students

- Staff student



7774 s01 s02 s05

6635

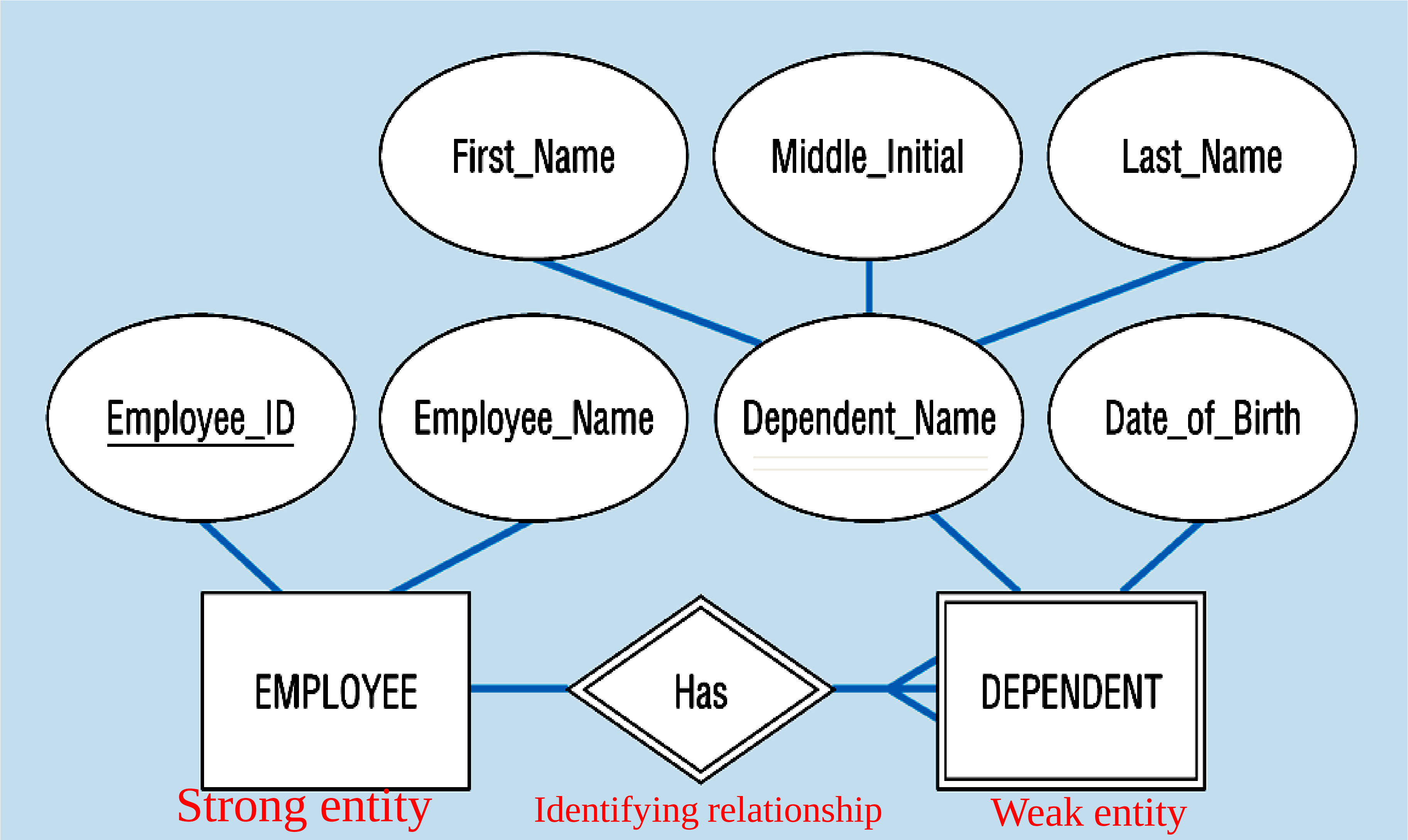
3158 s07

5324 s09 s10

Strong vs. Weak Entities, and Identifying Relationships

- Strong entities
  - exist independently of other types of entities
  - has its own unique identifier
  - represented with single-line rectangle
- Weak entity
  - dependent on a strong entity…cannot exist on its own
  - Does not have a unique identifier
  - represented with double-line rectangle
- Identifying relationship
  - links strong entities to weak entities
  - represented with double line diamond

Strong and weak entities

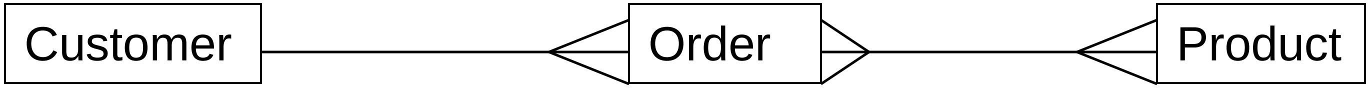


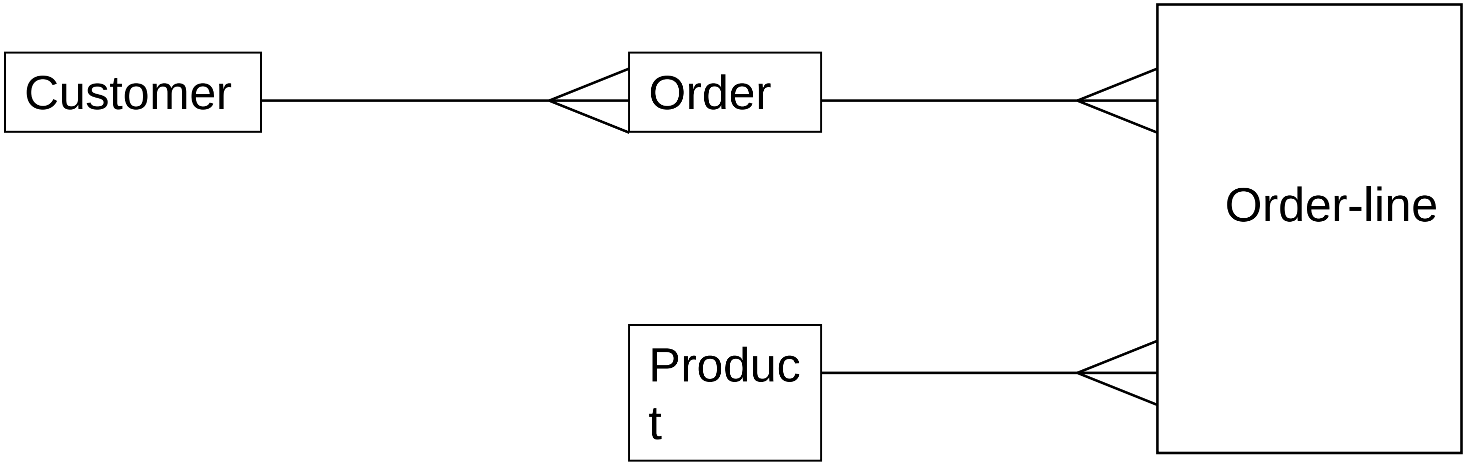
Strong entity

Identifying relationship

Weak entity

Resolving m:n relationships

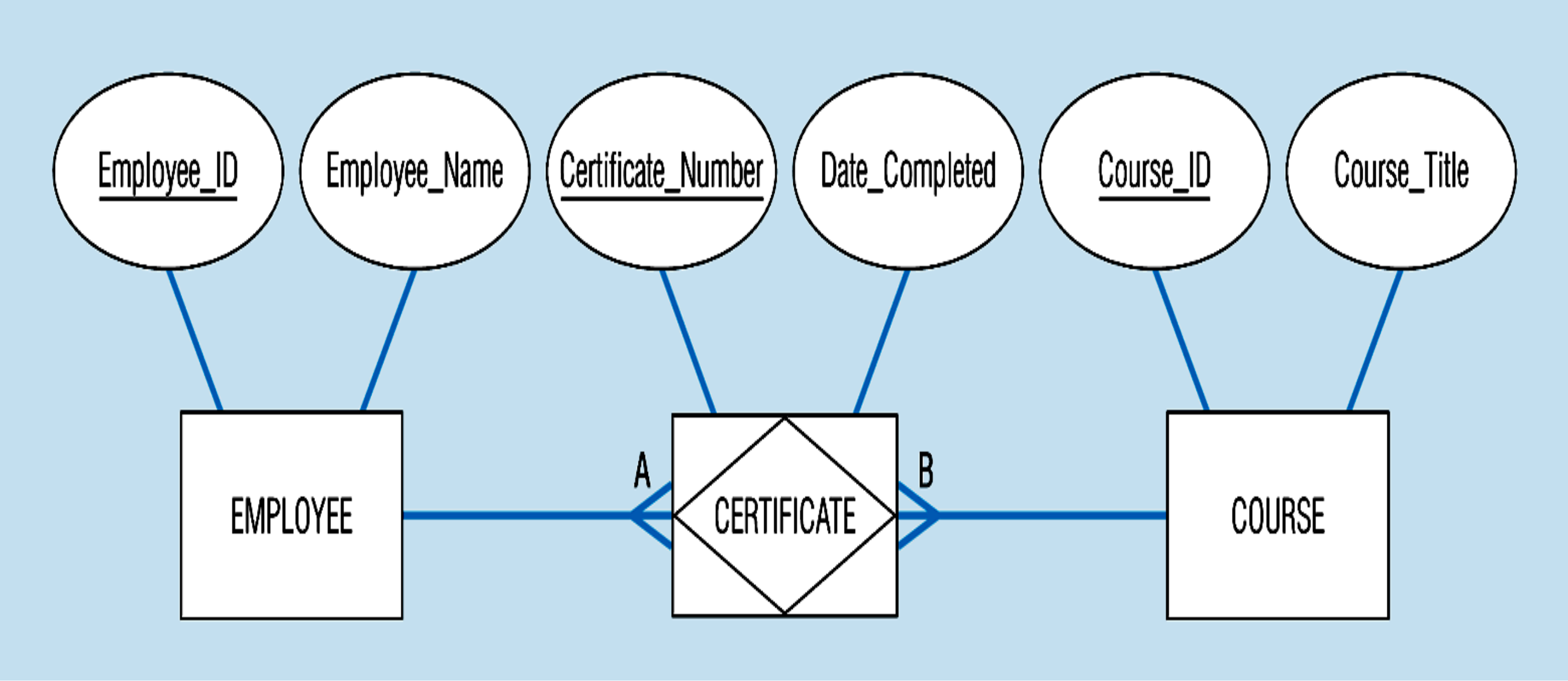




Associative Entities

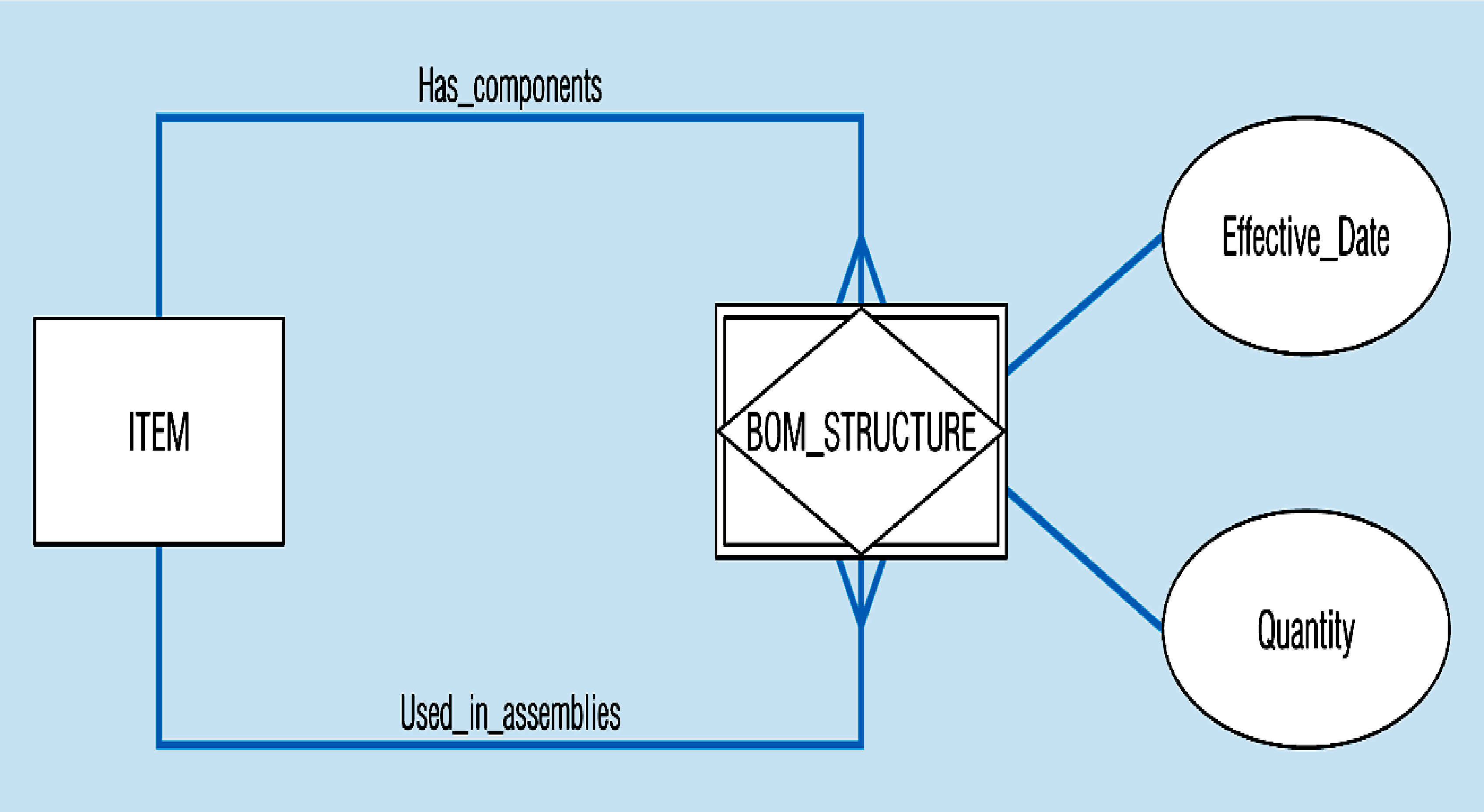
- It's an entity – it has attributes
- AND it's a relationship – it links entities together
- When should a *relationship with attributes* instead be an *associative entity*?
  - All relationships for the associative entity should be many
  - The associative entity could have meaning independent of the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

An associative entity (CERTIFICATE)



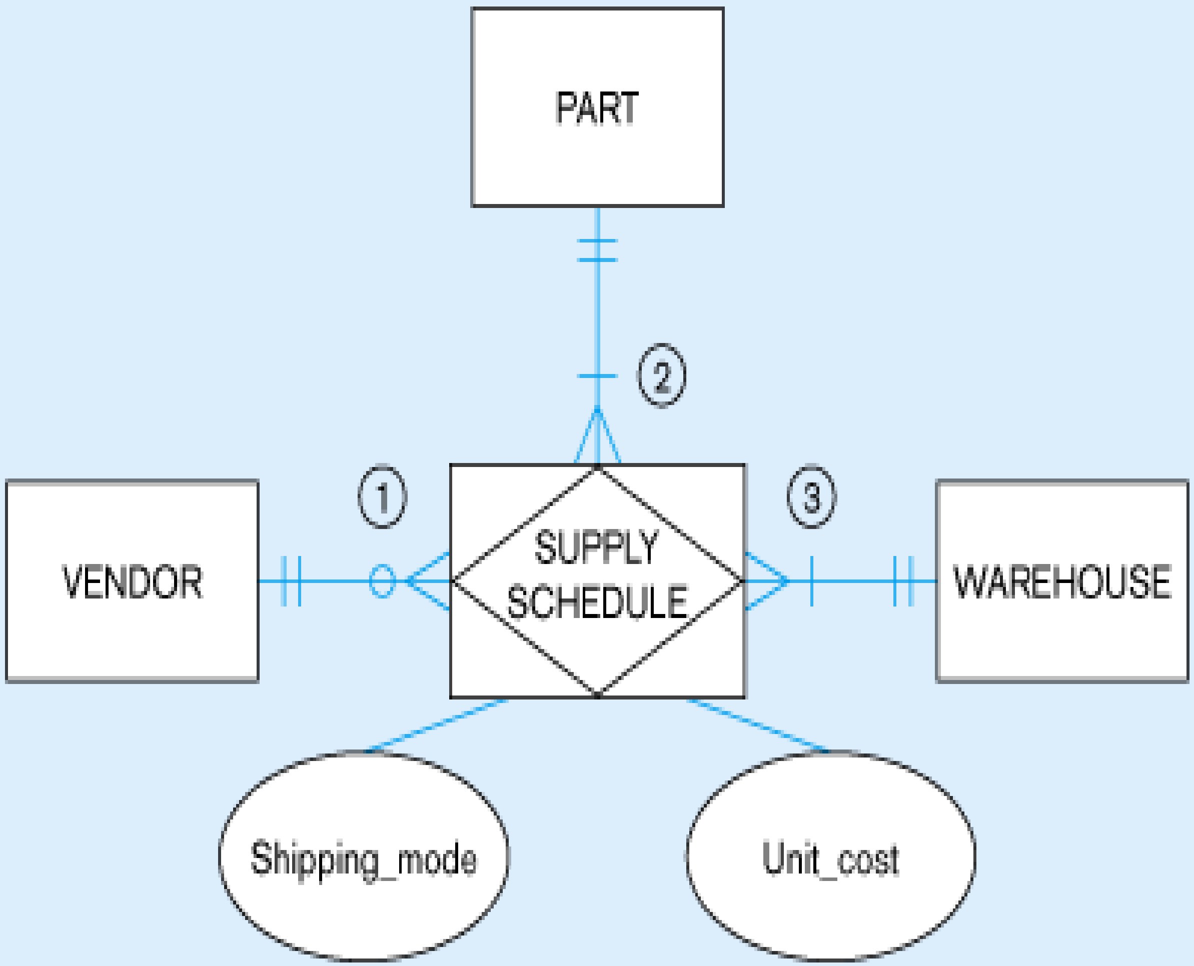
Associative entity involves a rectangle with a diamond inside. Note that the many-to-many cardinality symbols face toward the associative entity and not toward the other entities

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



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.

Constraints & assumptions

- Constraints
  - Not all data requirements can be captured using entities, attributes and relationships.
  - Such items are simply recorded as *constraints*
  - A course cannot accommodate more than say 25 students

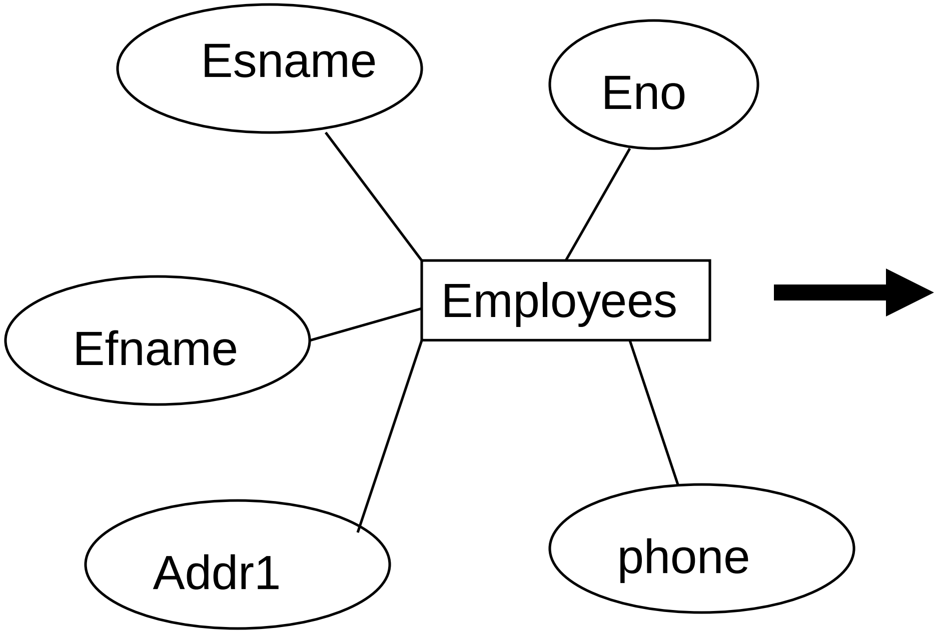
- Assumptions
  - To construct ERD we make assumptions about the data that are not stated in the data requirements
  - E.g. assume that student registers only once

Data models

- A method for organizing databases at the conceptual level
- Main concern is HOW to represent the relationships between database records
- E.g. Hierarchical, Network, relational models

Next step:Logical Database design

- i.e. ERD to relational model
  - Entity sets into tables



LOGICAL DB MODEL

```
CREATE TABLE EMPLOYEES (
Eno CHAR(5),
Esname CHAR(15),
Efname CHAR (15),
Addr1 CHAR (),
Phone INT(7),
PRIMARY KEY (Eno))
```

Constructing an ER model

- Identify entities
- Read the requirements specification carefully, and list all potential entity types.
- These are the objects of interest in the system. (It is better to put too many in and then discard them.)
- Remove duplicate entities

Constructing an ER model

- **Ensure that they really separate entity types or just two names for the same thing. Remember do not include the system as an entity type, e.g., if modelling a library, the entity types might be books, borrowers etc. There should not be a library entity type, because the library is the system.**
- **3. List the attributes of each entity**
- **Ensure that the entity types are really needed, are any of them just attributes of another entity type? For example, is telephone an entity, or an attribute telephone-number of say an Office entity?**
- **If so keep them as attributes and cross them off the entity list.**

Constructing an ER model

- 4. Mark the primary keys
- 5. Define the relationships
- Examine each entity type to see how it is related to all of the others.
- 6. Describe the cardinality and optionality of the relationships
- Examine the constraints which hold between participating entities.

Constructing an ER model

- **7. Remove redundant relationships**
- **Examine the ER model for redundant relationships**
- **ER modelling is an iterative process, so you will draw several versions, refining each one until you are happy with it. NB, there is no one right answer to the problem, but some solutions are better than others.**

Recipe to build ERD

- **Identify Potential Entities**
- **Identify attributes**
- **Choose identifiers**
- **Draw an initial ER Diagram**
- **Add relationship information**
- **Add degree information**
- **Add participation information**
- **Redraw the ER Diagram**

Example to try yourself:

- Identify the entity types, attributes and relationships in the following problem.
- Then draw the ER model.
- Feline Publishing Inc. is a publishing company that produces animal care books on various pets. Each book is written by an author who specialises in that animal. An author can write several books. Each book is edited by a single editor, although an editor can work on several books at the same time.