Ch 4

Database Management System

Sumayyea Salahuddin (Lecturer)
Dept. of Computer Systems Eng.
UET Peshawar

Overview

- Logical Database Design & Relational Model
  - Relations
  - Primary key, Composite key, Foreign key, Recursive Foreign key
  - Entity Integrity Rule
  - Referential Integrity Constraint
  - Transforming Conceptual Schema to Relational Schema

Correspondence with ER Model

- Definition: A relation is a named, two-dimensional table of data
  - Table is made up of rows (records), and columns (attribute or field)

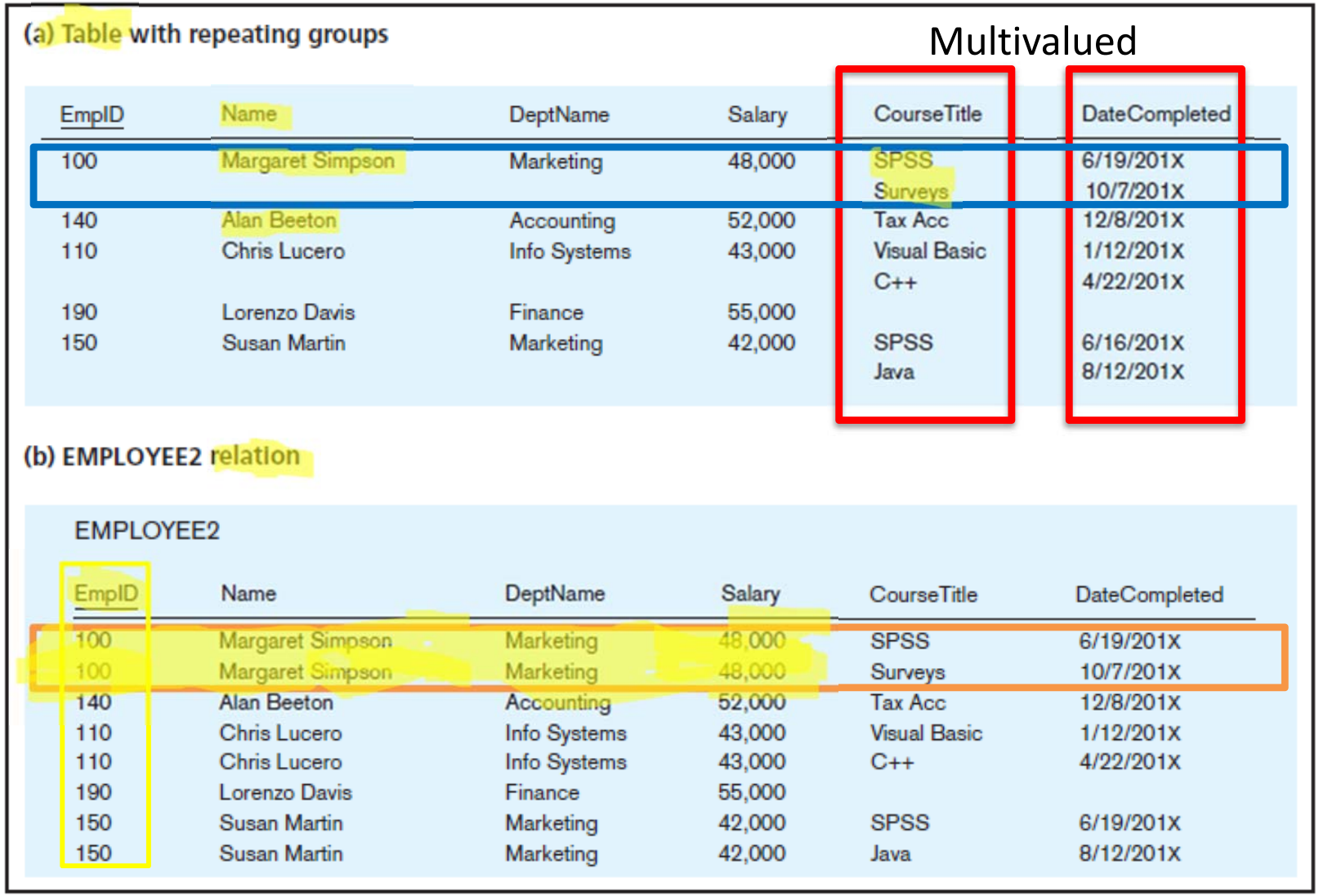
- Not all tables qualify as relations (See Figure 4.2)

- **Requirements**
  - Every relation has a unique name.
  - Every attribute value is atomic (not multi-valued, not composite)
  - Every row is unique (can't have two rows with exactly the same values for all their fields)
  - Attributes (columns) in tables have unique names
  - The order of the columns is irrelevant
  - The order of the rows is irrelevant

Note: All relations are in 1st Normal Form

Difference between Table & Relation

FIGURE 4-2 Eliminating multivalued attributes



(a) Table with repeating groups

Multivalued

EmpID	Name	DeptName	Salary	CourseTitle	DateCompleted
100	Margaret Simpson	Marketing	48,000	SPSS	6/19/201X
				Surveys	10/7/201X
140	Alan Beeton	Accounting	52,000	Tax Acc	12/8/201X
110	Chris Lucero	Info Systems	43,000	Visual Basic	1/12/201X
				C++	4/22/201X
190	Lorenzo Davis	Finance	55,000		
150	Susan Martin	Marketing	42,000	SPSS	6/16/201X
				Java	8/12/201X

(b) EMPLOYEE2 relation

EMPLOYEE2

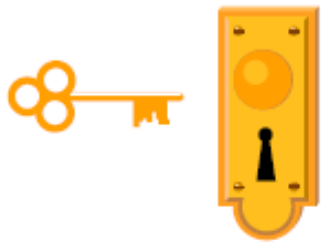
EmpID	Name	DeptName	Salary	CourseTitle	DateCompleted
100	Margaret Simpson	Marketing	48,000	SPSS	6/19/201X
100	Margaret Simpson	Marketing	48,000	Surveys	10/7/201X
140	Alan Beeton	Accounting	52,000	Tax Acc	12/8/201X
110	Chris Lucero	Info Systems	43,000	Visual Basic	1/12/201X
110	Chris Lucero	Info Systems	43,000	C++	4/22/201X
190	Lorenzo Davis	Finance	55,000		
150	Susan Martin	Marketing	42,000	SPSS	6/19/201X
150	Susan Martin	Marketing	42,000	Java	8/12/201X

Relation

- Relations (tables) correspond with entity types and with many-to-many relationship types
- Rows correspond with entity instances and with many-to-many relationship instances
- Columns correspond with attributes

- NOTE: The word relation (in relational database) is NOT the same word relationship (in ER model)

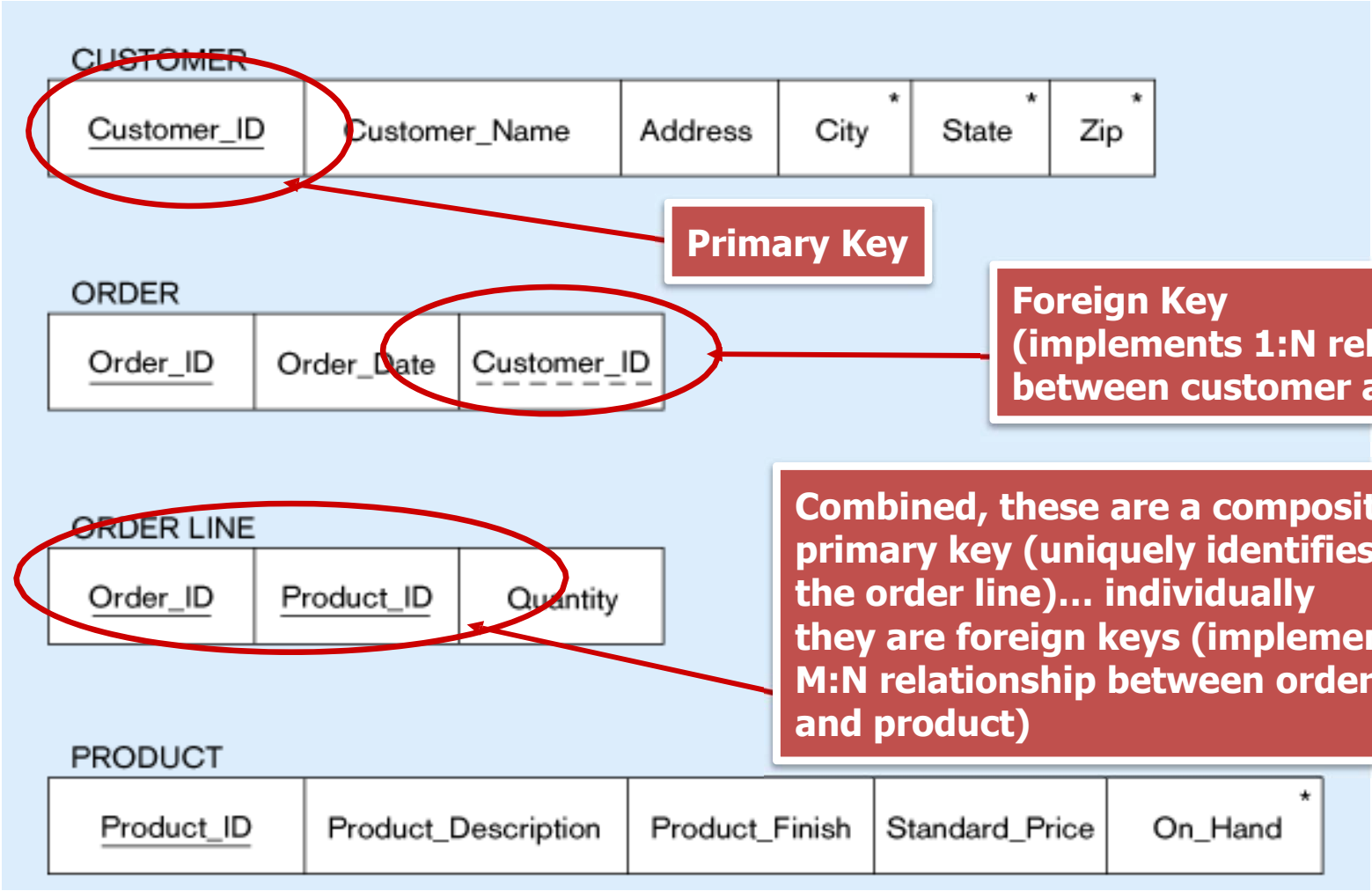
Key Fields



- Keys are special fields that serve two main purposes:
  - Primary keys are unique identifiers of the relation in question. Examples include employee numbers, social security numbers, etc. This is how we can guarantee that all rows are unique
  - Foreign keys are identifiers that enable a dependent relation (on the many side of a relationship) to refer to its parent relation (on the one side of the relationship)

- Keys can be simple (a single field) or composite (more than one field)
- Keys usually are used as indexes to speed up the response to user queries

Schema for Four Relations (Pine Valley Furniture)



Integrity Constraints

- Domain Constraints
  - Allowable values for an attribute (See Table 4-1 on next slide)
- Entity Integrity
  - No primary key attribute may be null. All primary key fields MUST have data

Domain Constraints

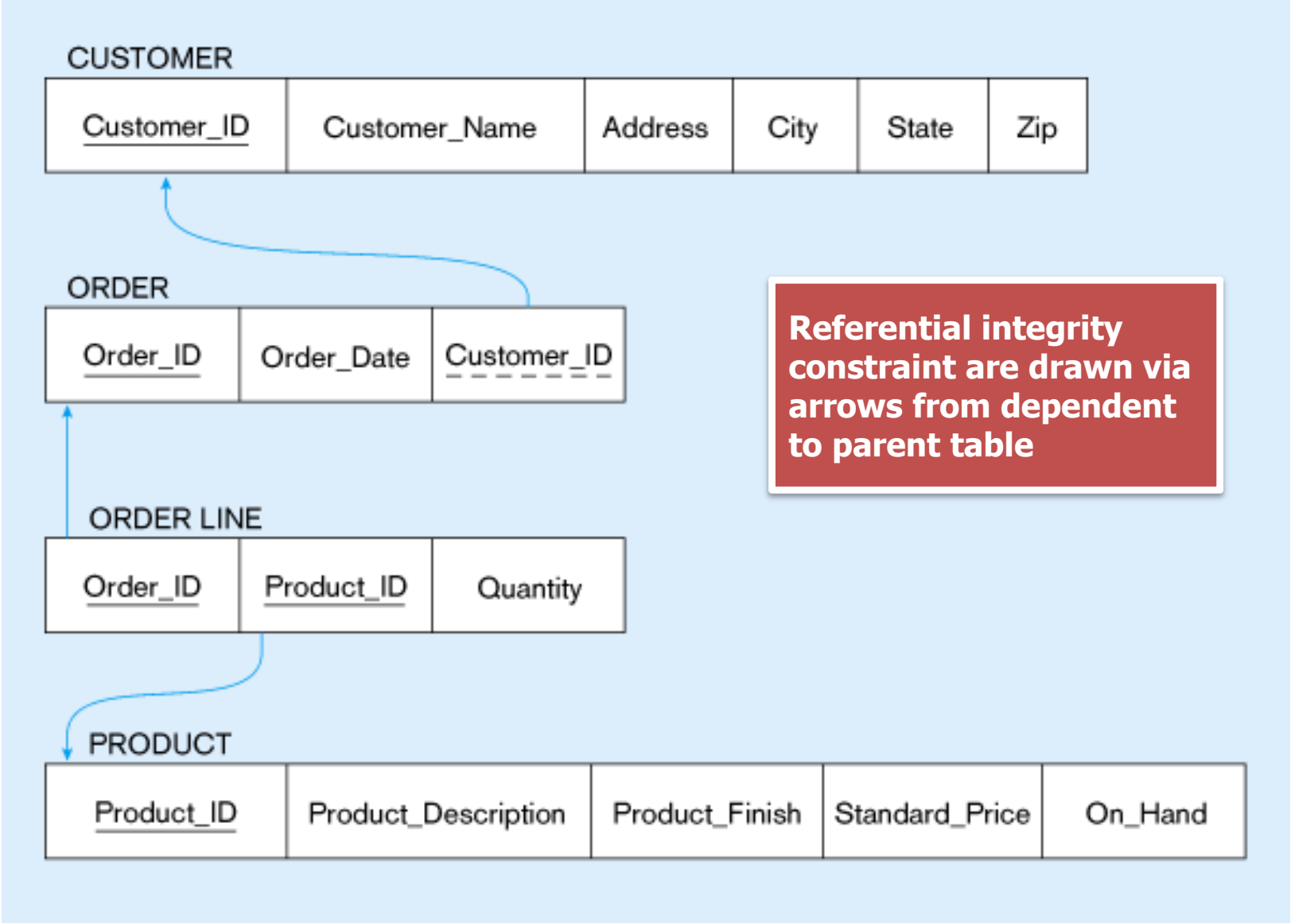
TABLE 4-1 Domain Definitions for INVOICE Attributes

Attribute	Domain Name	Description	Domain
CustomerID	Customer IDs	Set of all possible customer IDs	character: size 5
CustomerName	Customer Names	Set of all possible customer names	character: size 25
CustomerAddress	Customer Addresses	Set of all possible customer addresses	character: size 30
CustomerCity	Cities	Set of all possible cities	character: size 20
CustomerState	States	Set of all possible states	character: size 2
CustomerPostalCode	Postal Codes	Set of all possible postal zip codes	character: size 10
OrderID	Order IDs	Set of all possible order IDs	character: size 5
OrderDate	Order Dates	Set of all possible order dates	date: format mm/dd/yy
ProductID	Product IDs	Set of all possible product IDs	character: size 5
ProductDescription	Product Descriptions	Set of all possible product descriptions	character: size 25
ProductFinish	Product Finishes	Set of all possible product finishes	character: size 15
ProductStandardPrice	Unit Prices	Set of all possible unit prices	monetary: 6 digits
ProductLineID	Product Line IDs	Set of all possible product line IDs	integer: 3 digits
OrderedQuantity	Quantities	Set of all possible ordered quantities	integer: 3 digits

Integrity Constraints (Cont…)

- Referential Integrity – rule that states that any foreign key value (on the relation of the many side) MUST a primary key value in the relation of the one side. (Or the foreign key can be null)
  - For example: Delete Rules
  - Restrict: don't allow delete of parent side of related rows exist in dependent side
  - Cascade: automatically delete dependent side rows that correspond with the parent side row to be deleted
  - Set-to-Null: set the foreign key in the dependent side to null if deleting from the parent side; not allowed for weak entities

Referential Integrity Constraints (PVF)

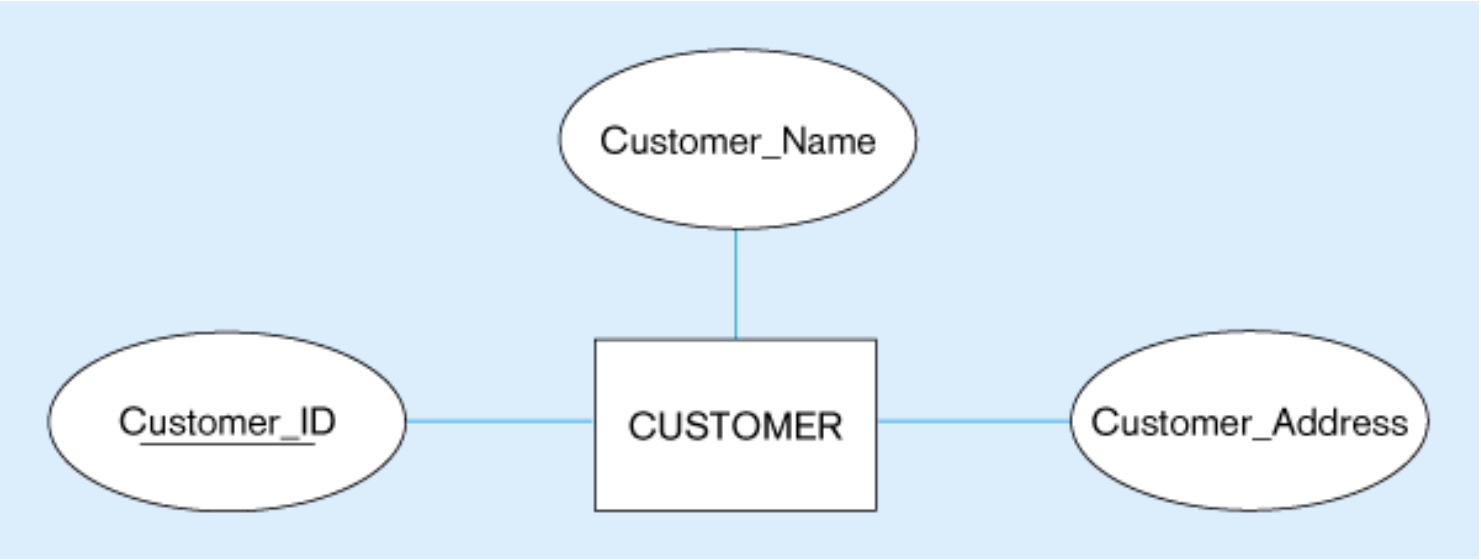


Transforming EER Diagrams into Relations

- Mapping Regular Entities to Relations
  1. Simple Attributes: E-R attributes map directly onto the relation
  2. Composite Attributes: Use only their simple, component attributes
  3. Multi-valued Attribute: Becomes a separate relation with a foreign key take from the superior entity

Mapping a Regular Entity

(a)
CUSTOMER entity type with simple attributes

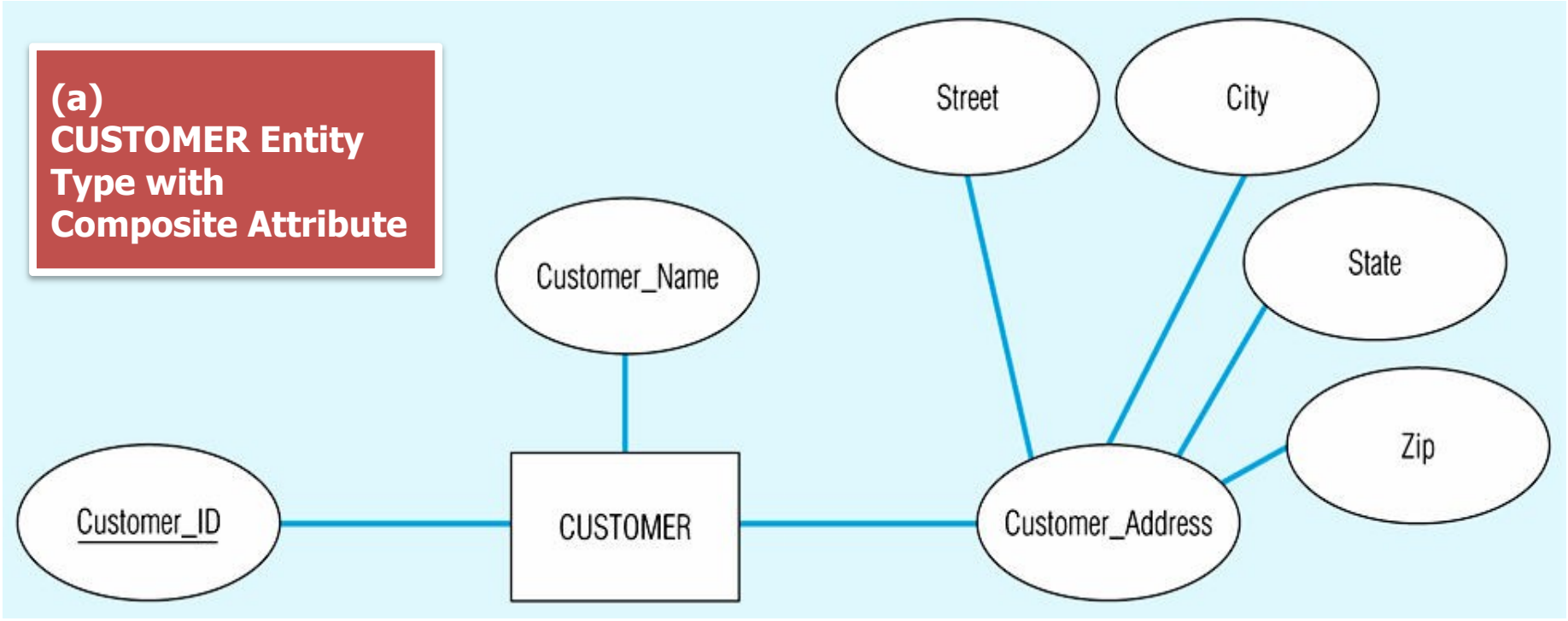


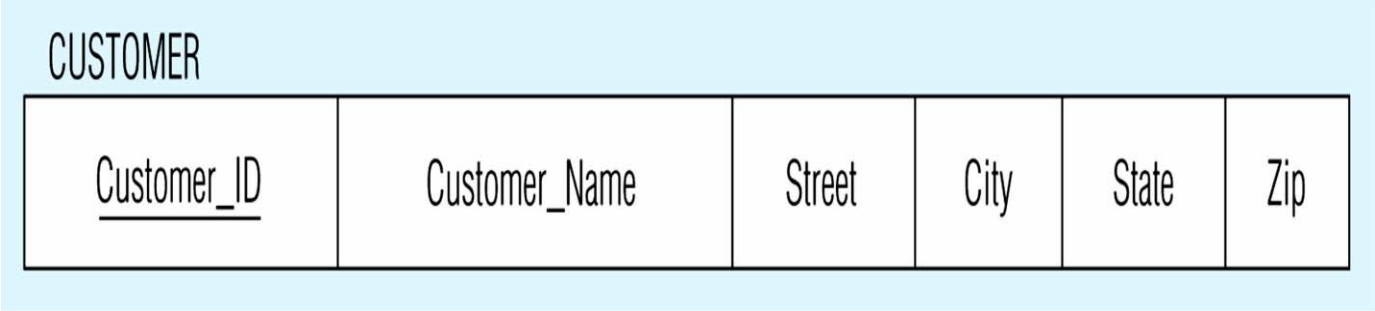
CUSTOMER

Customer_ID	Customer_Name	Customer_Address

(b)
CUSTOMER Relation

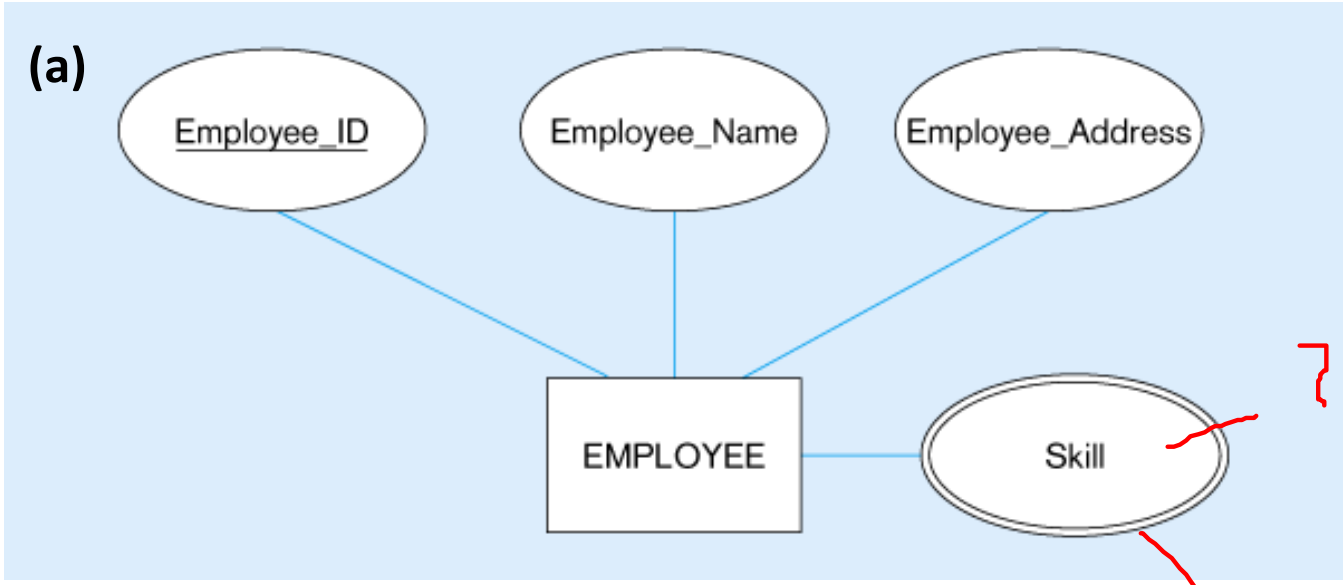
Mapping a Composite Attribute



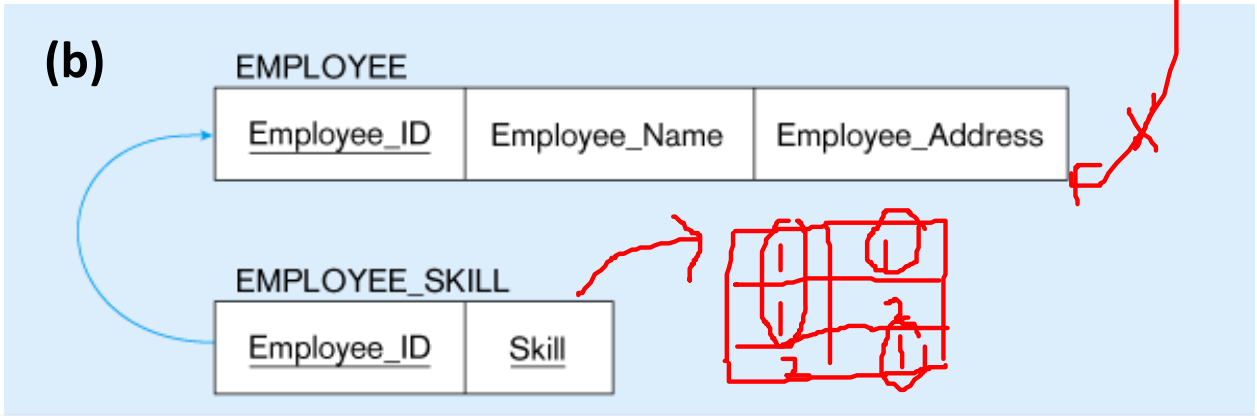


(b)
CUSTOMER relation with Address Detail

Mapping a Multi-Valued Attribute



Multi-valued attribute becomes a separate relation with foreign key

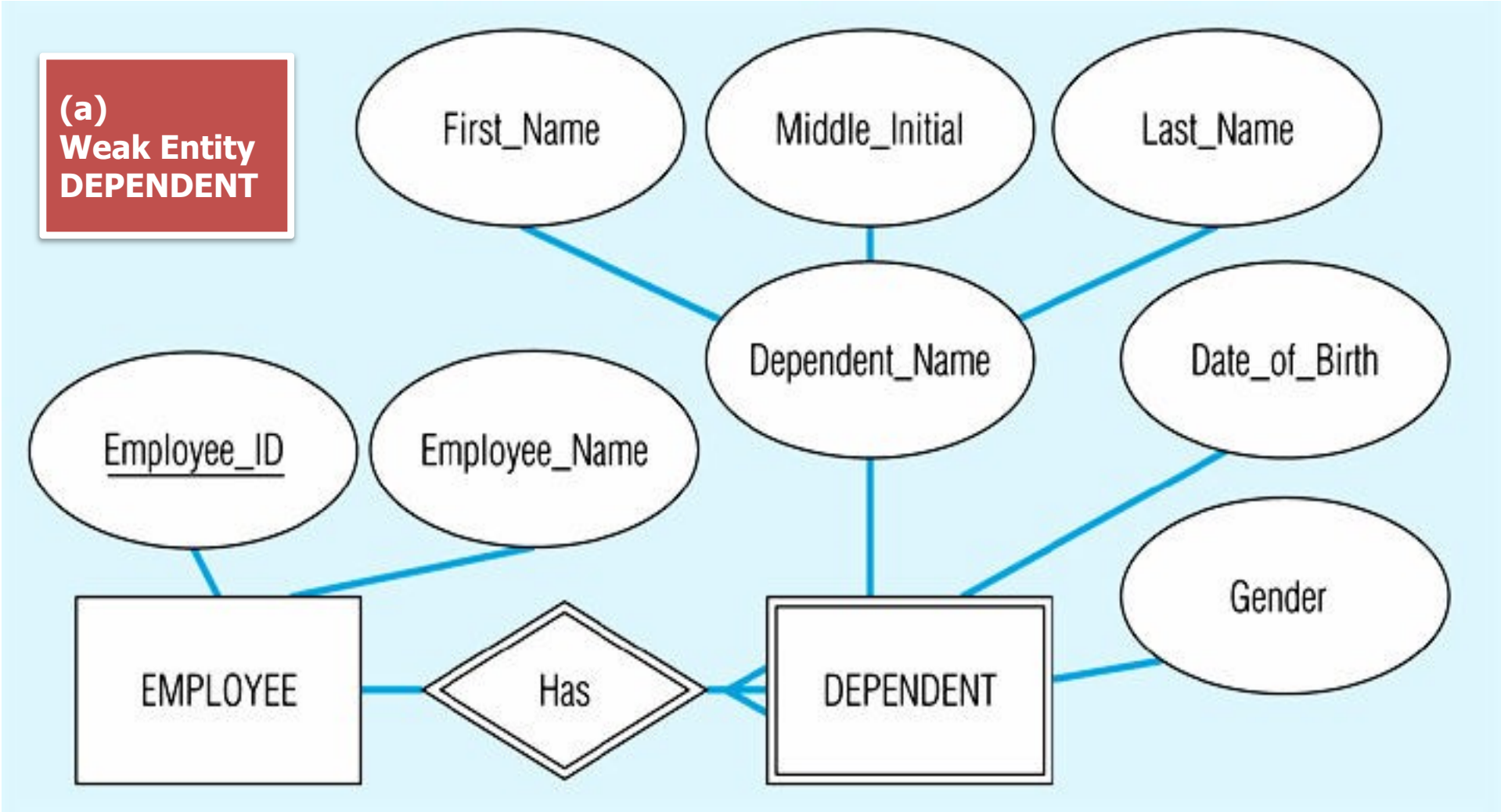


1-to-many relationship between original entity and new relation

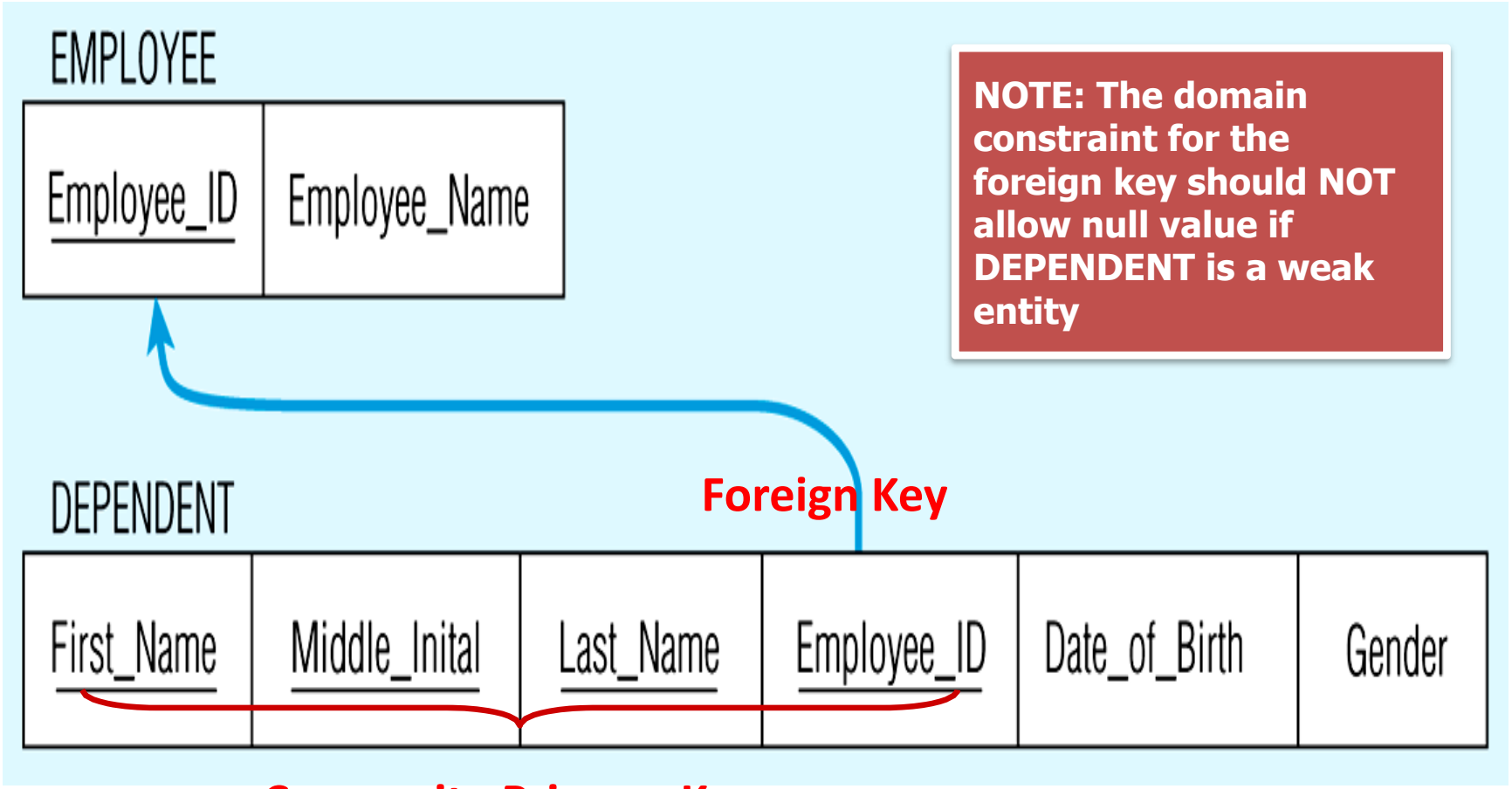
Transforming EER Diagrams into Relations

- Mapping Weak Entities
  - Becomes a separate relation with a foreign key taken from the superior entity
  - Primary key composed of:
    - Partial identifier of weak entity
    - Primary key of identifying relation (strong entity)

Example of Mapping a Weak Entity



Relations Resulting from Weak Entity



Alternate Approach: Surrogate Primary Key

Surrogate primary key
A serial number or other system-assigned primary key for a relation.

DEPENDENT(Dependent#, EmployeeID, FirstName, MiddleInitial, LastName, DateOfBirth, Gender)

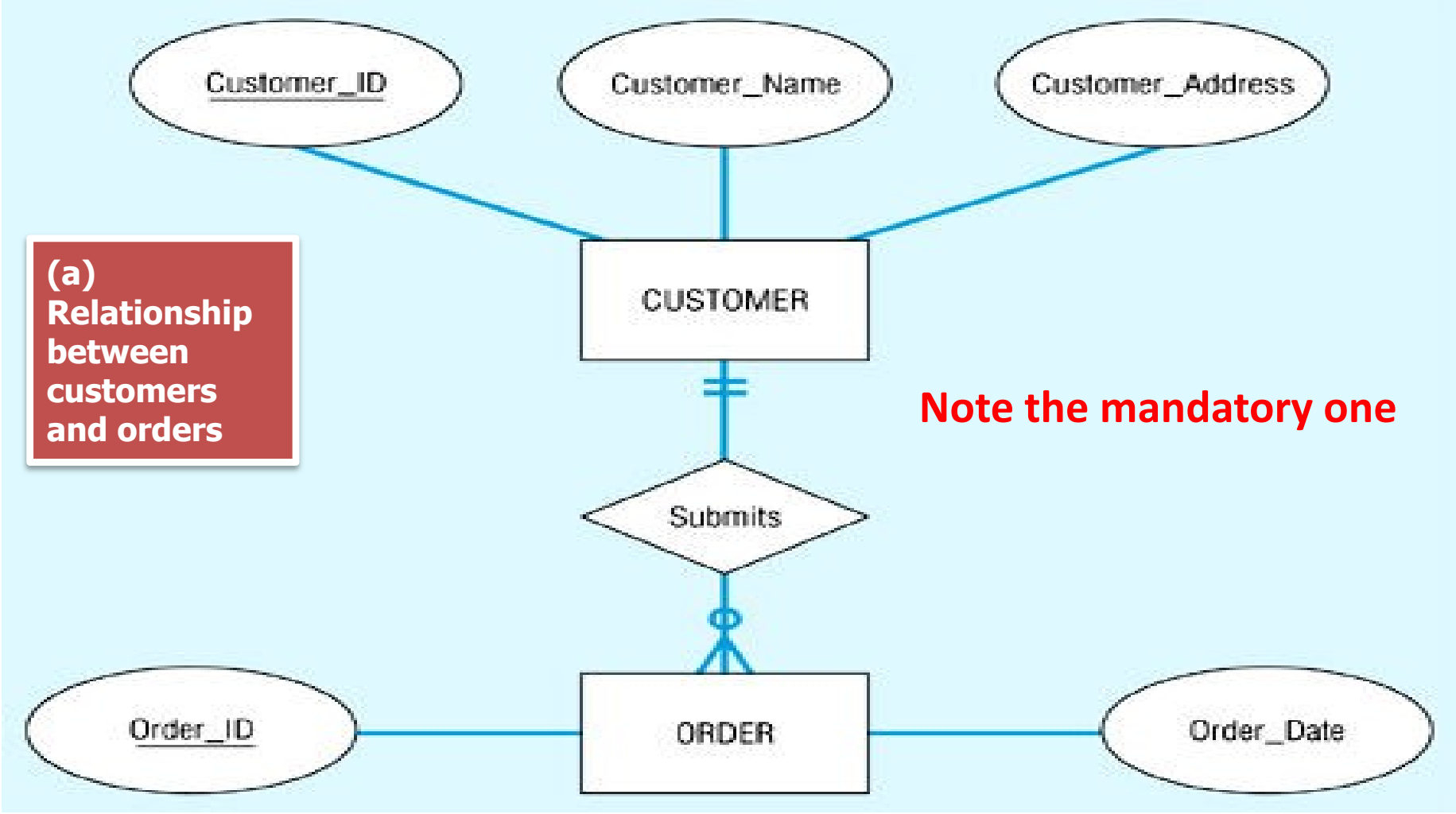
When to create Surrogate Key?

- There is a composite primary key, as in the case of the DEPENDENT relation shown previously with the four component primary key.
- The natural primary key (i.e., the key used in the organization and identified in conceptual data modeling as the identifier) is inefficient (e.g., it may be very long and hence costly for database software to handle if it is used as a foreign key that references other tables).
- The natural primary key is recycled (i.e., the key is reused or repeated periodically, so it may not actually be unique over time); a more general statement of this condition is when the natural primary key cannot, in fact, be guaranteed to be unique over time (e.g., there could be duplicates, such as with names or titles).

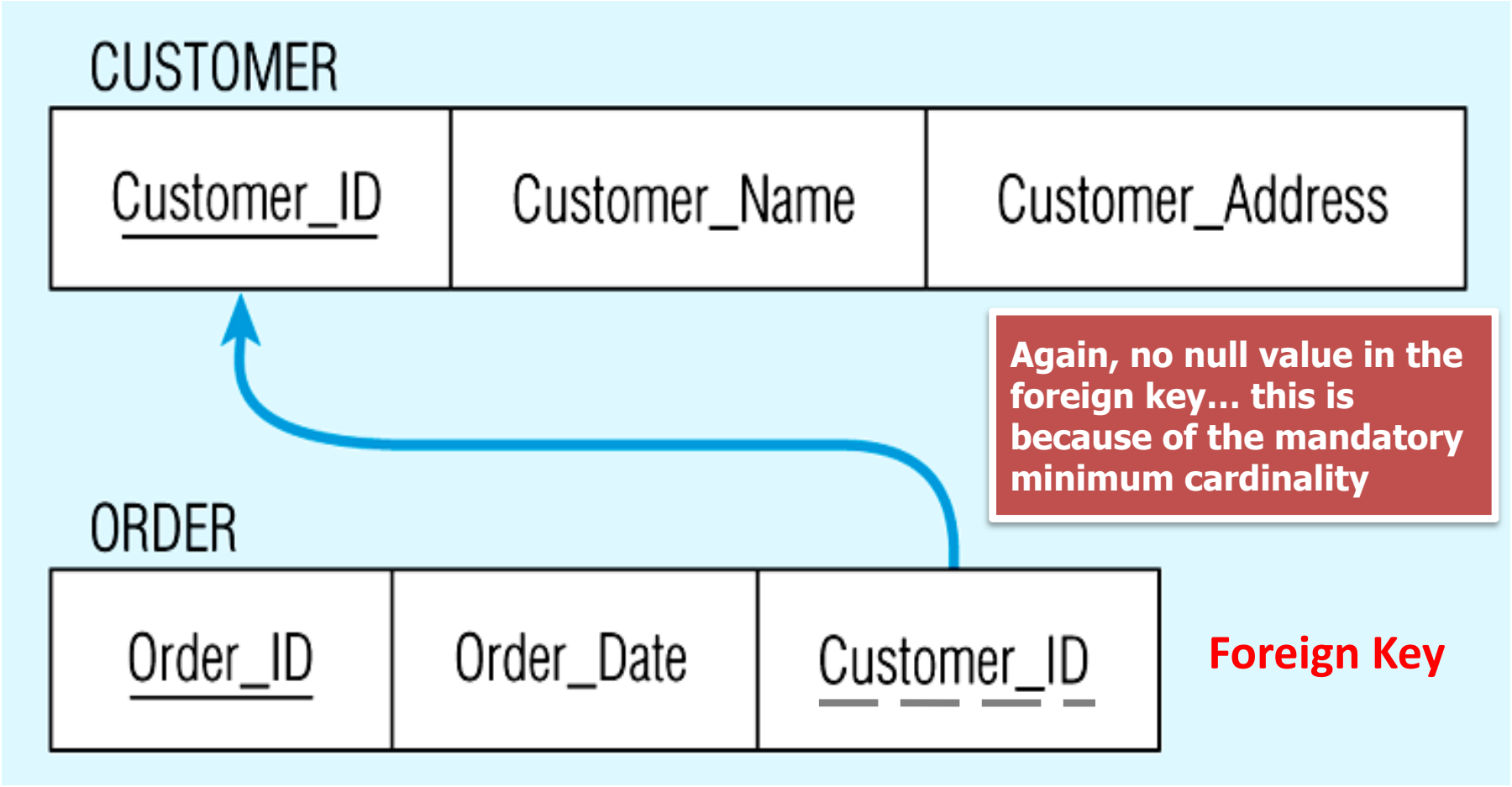
Transforming EER Diagrams into Relations

- Mapping Binary Relationships
  - One-to-Many: Primary key on the one side becomes a foreign key on the many side
  - Many-to-Many: Create a new relation with the primary keys of the two entities as its primary key
  - One-to-One: Primary key on the mandatory side becomes a foreign key on the optional side

Example of Mapping a 1:M Relationship

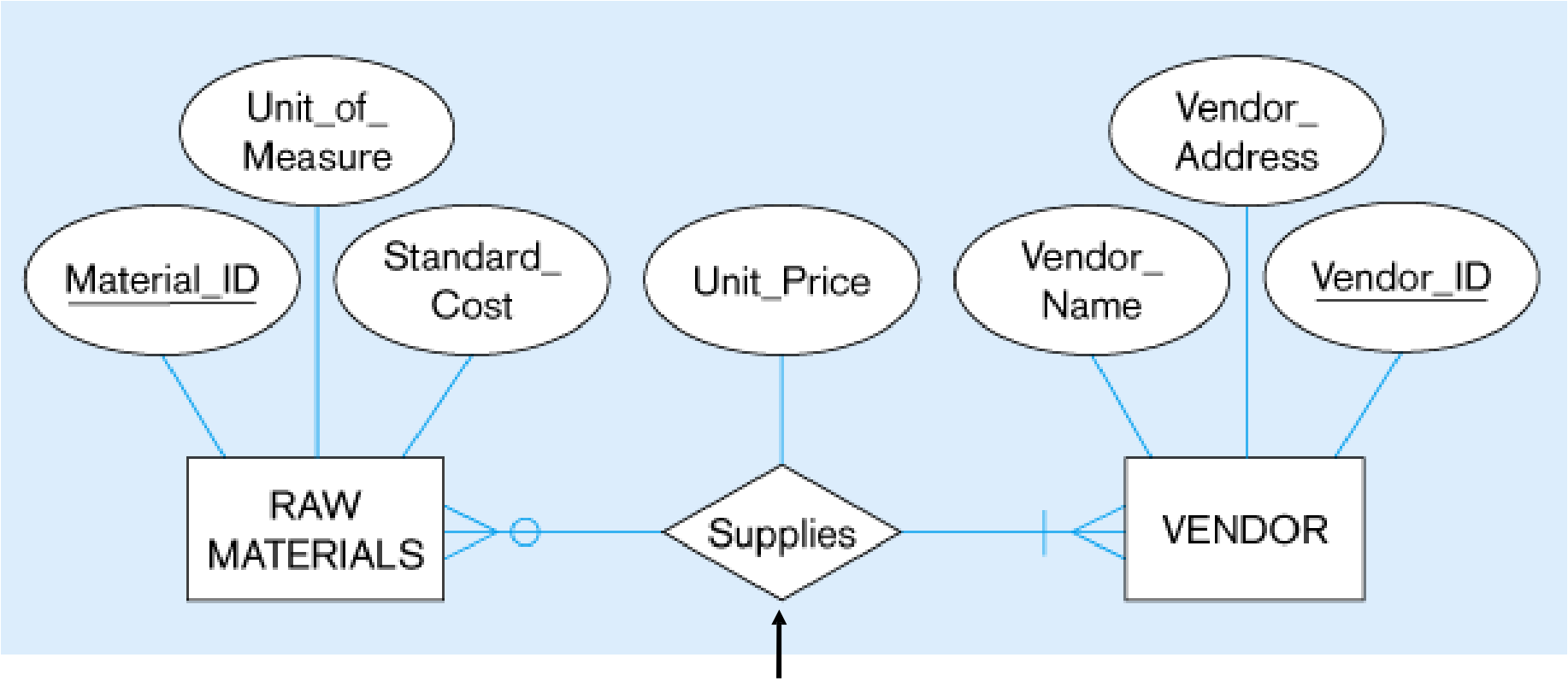


Example of Mapping a 1:M Relationship (cont…)



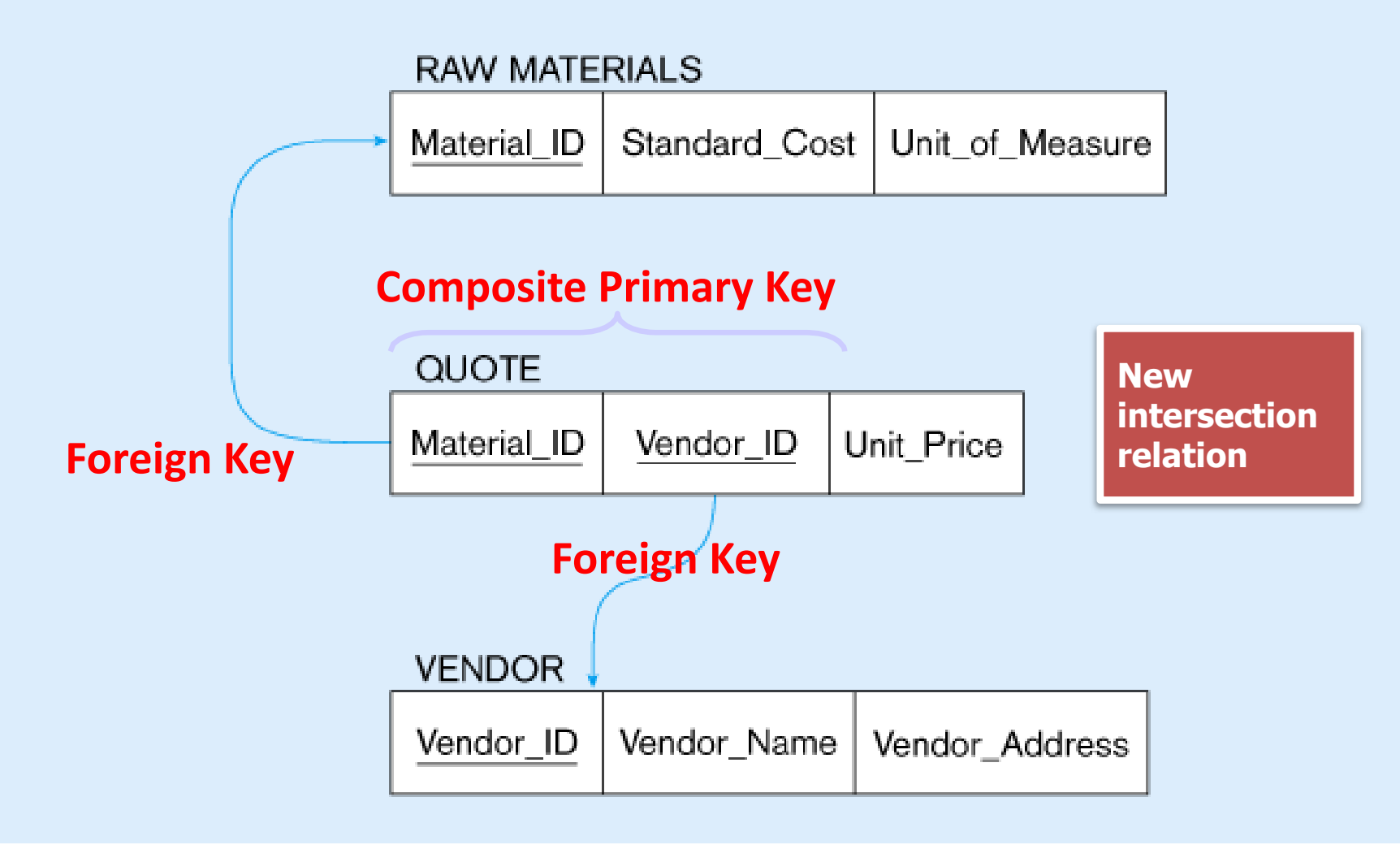
Example of Mapping a M:N Relationship

(a) ER Diagram (M:N)



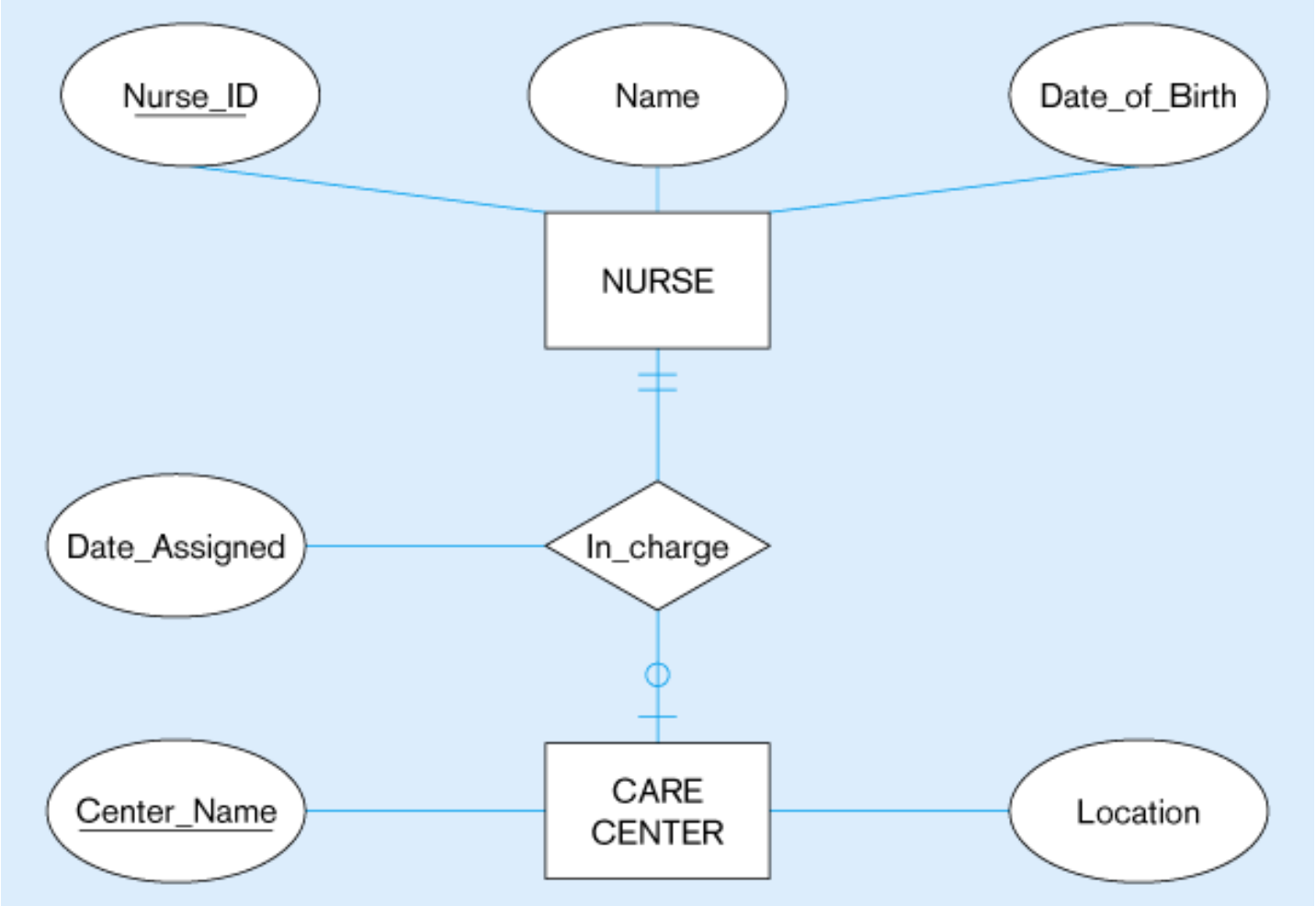
The Supplies relationship will need to become a separate relation

Three Resulting Relations



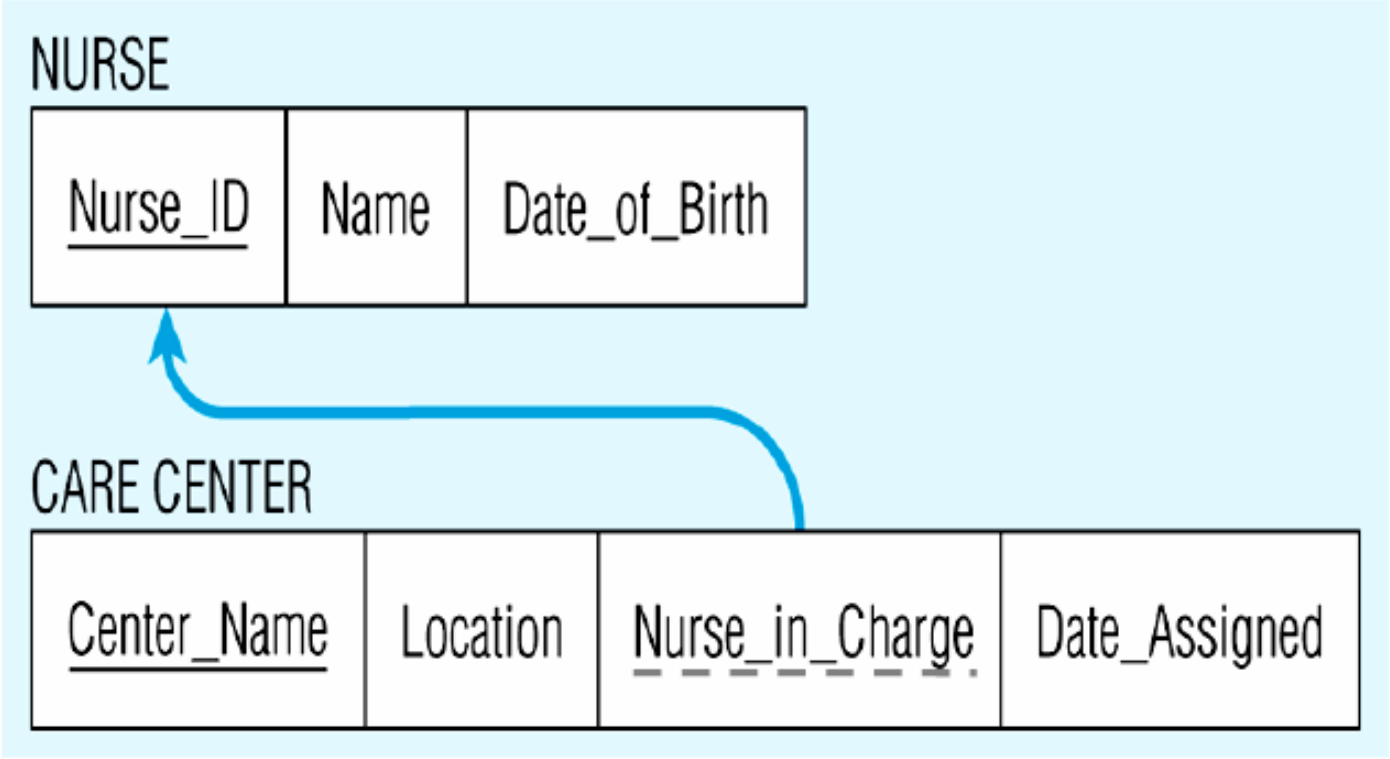
Mapping a 1:1 Relationship

(a) Binary 1:1 Relationship



Mapping a 1:1 Relationship (cont…)

(b) Resulting Relation

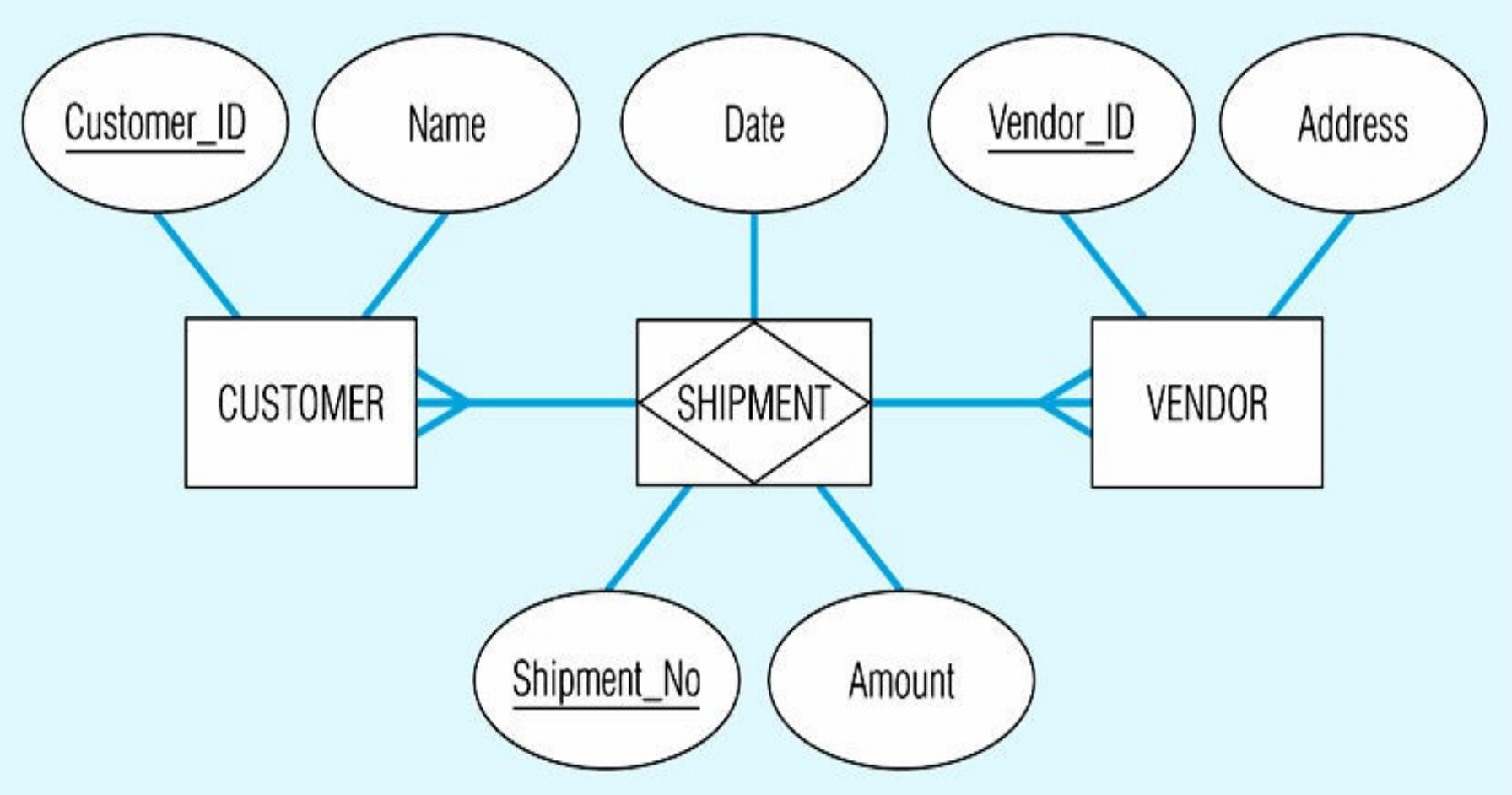


Transforming EER Diagrams into Relations

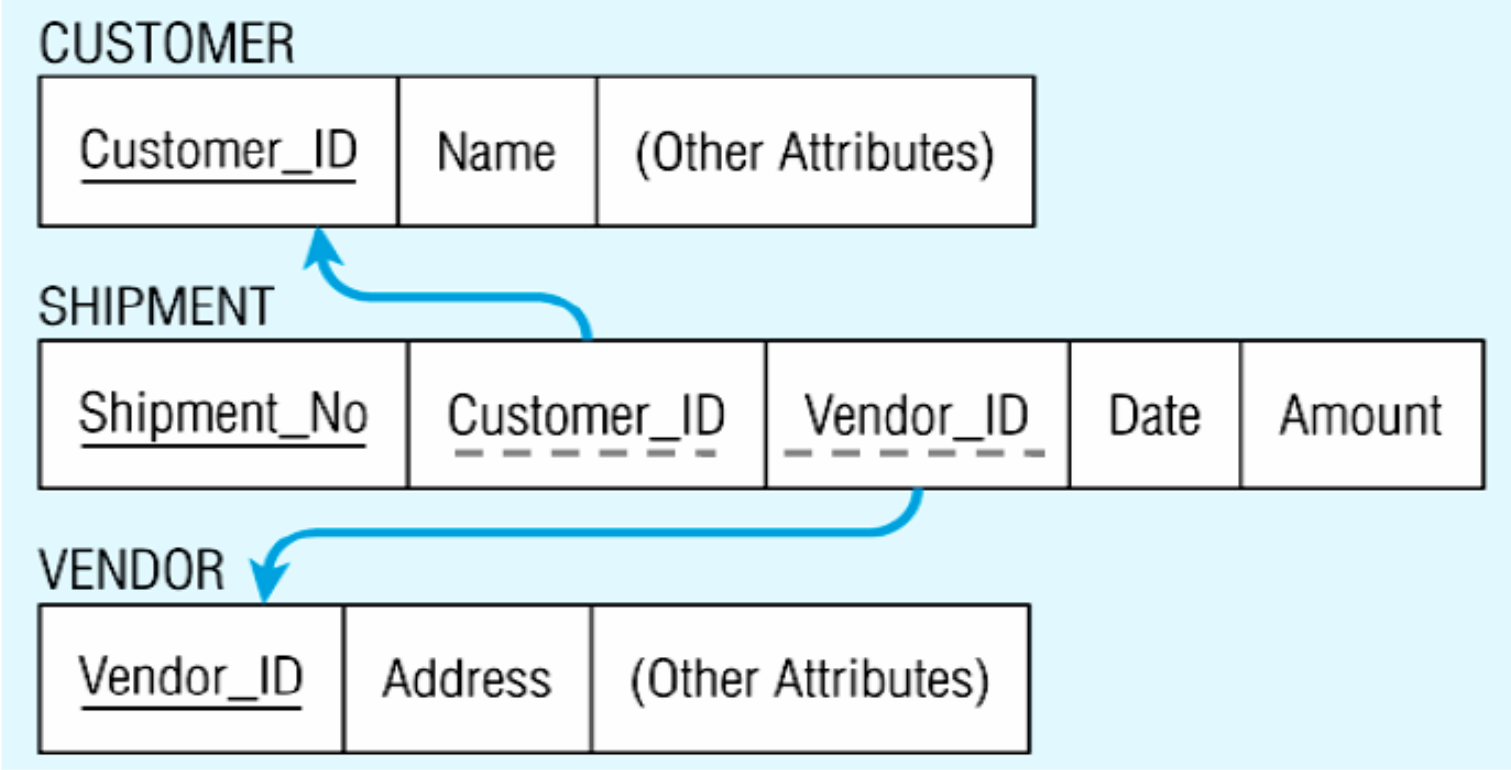
- Mapping Associative Entities
  - Identifier Not Assigned
    - Default primary key for the association relation is composed of the primary keys of the two entities (as in M:N relationship)
  - Identifier Assigned
    - It is natural and familiar to end-users
    - Default identifier may not be unique

Mapping an Associative Entity

(a) Associative Entity



Three Resulting Relations



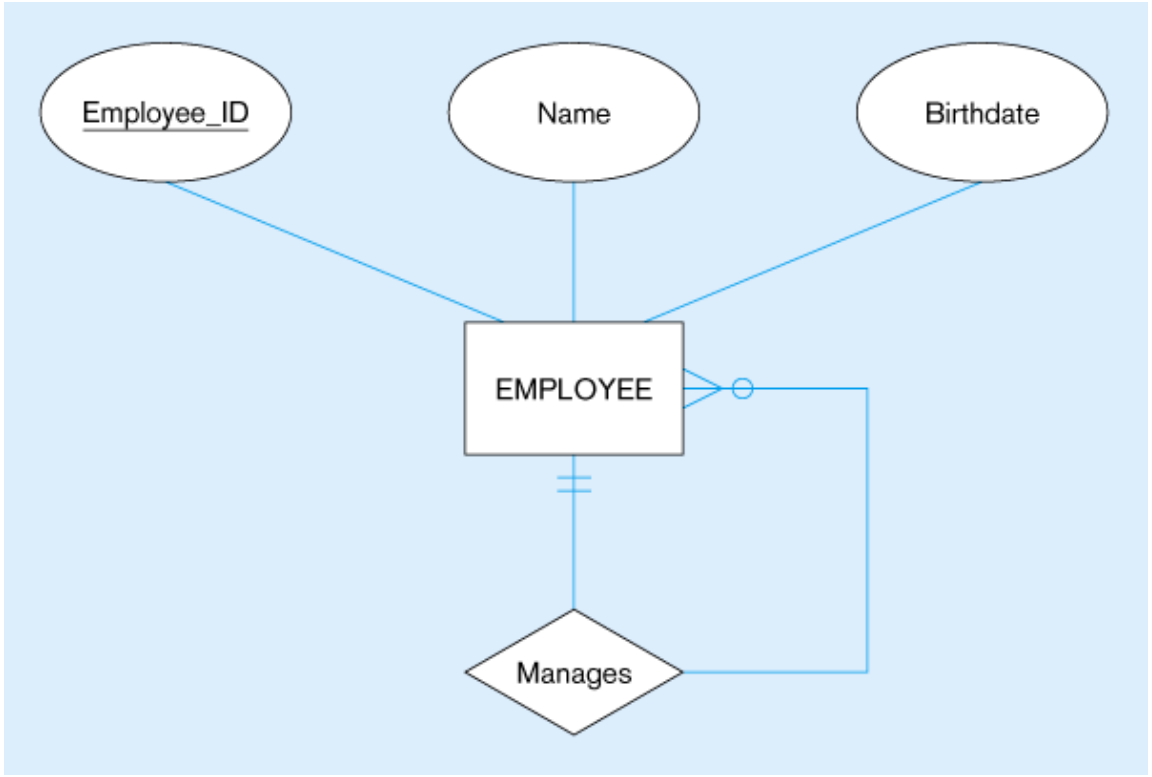
Transforming EER Diagrams into Relations

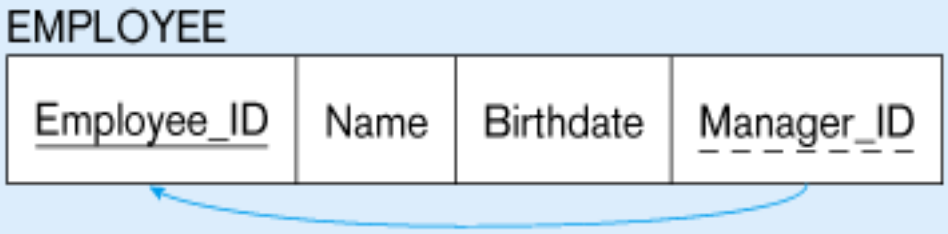
- Mapping Unary Relationships
  - One-to-Many: Recursive foreign key in the same relation
  - Many-to-Many: Two relations
    - One for the entity type
    - One for an associative relation in which the primary key has two attributes, both taken from the primary key of the entity

Recursive foreign key
A foreign key in a relation that references the primary key values of the same relation.

Mapping a Unary 1:N Relationship

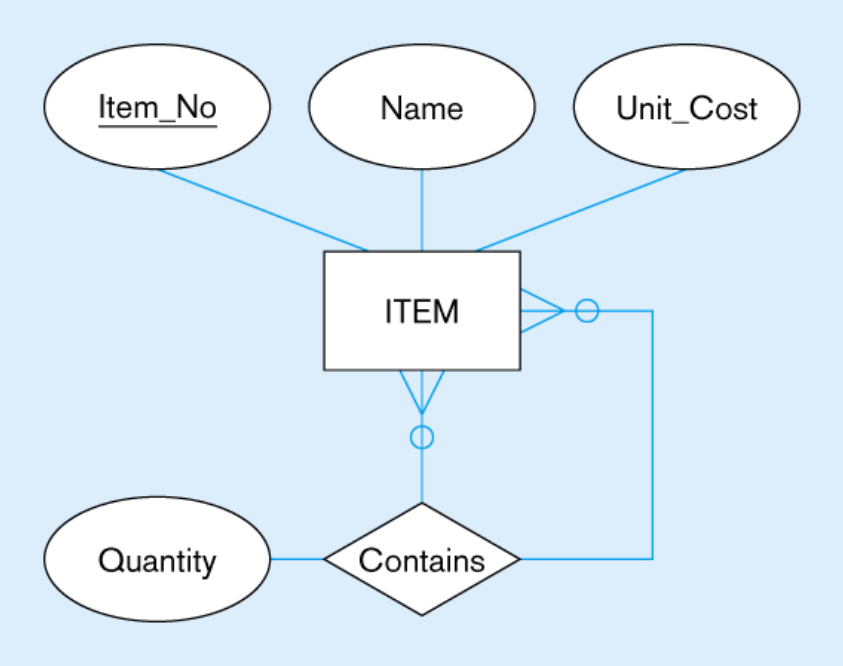
(a)
EMPLOYEE entity with Manages relationship





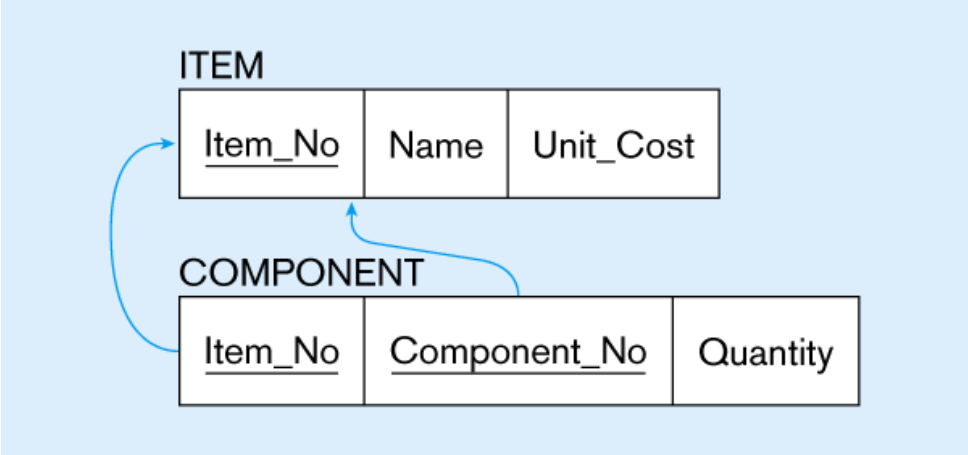
(b)
EMPLOYEE relation with foreign key

Mapping a Unary M:N Relationship



(a)
Bill-of-Materials Relationship (M:N)

(b)
ITEM and COMPONENT Relations

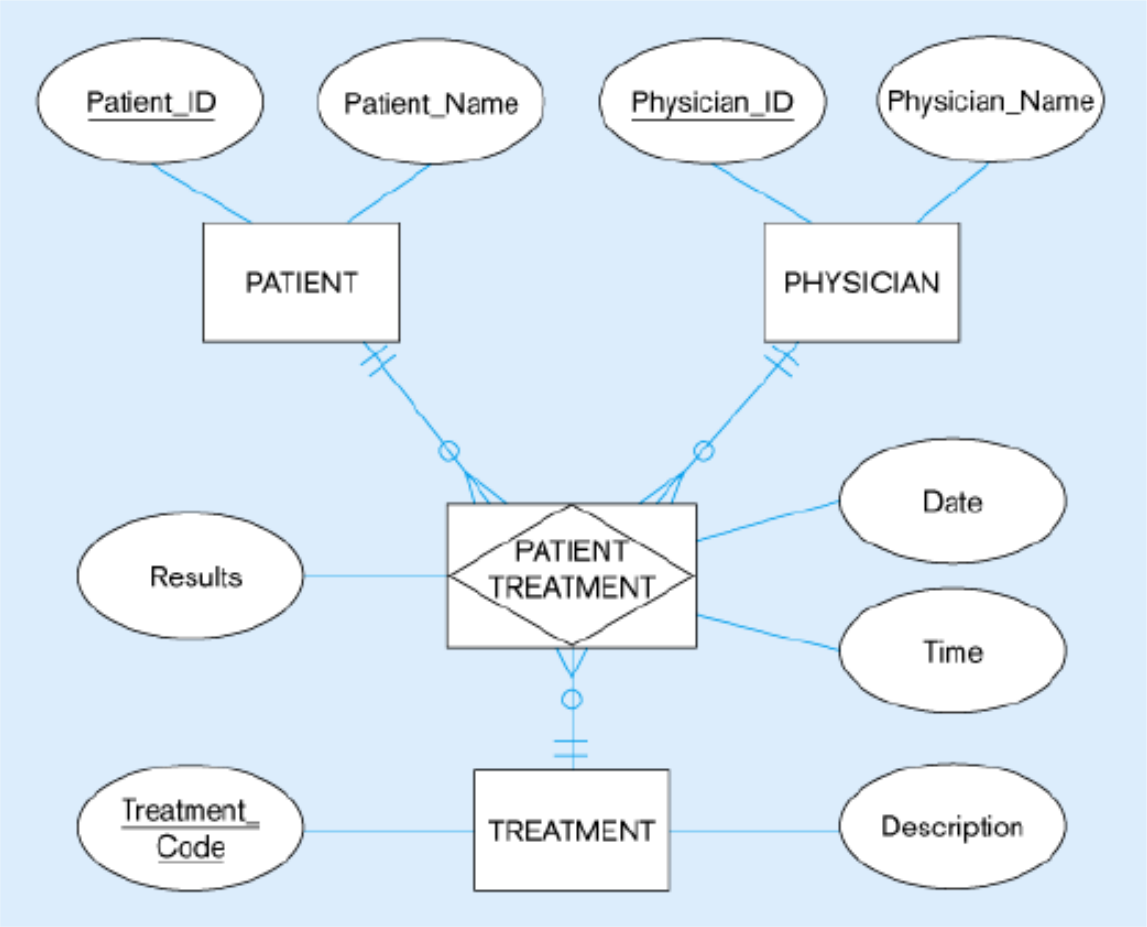


Transforming EER Diagrams into Relations

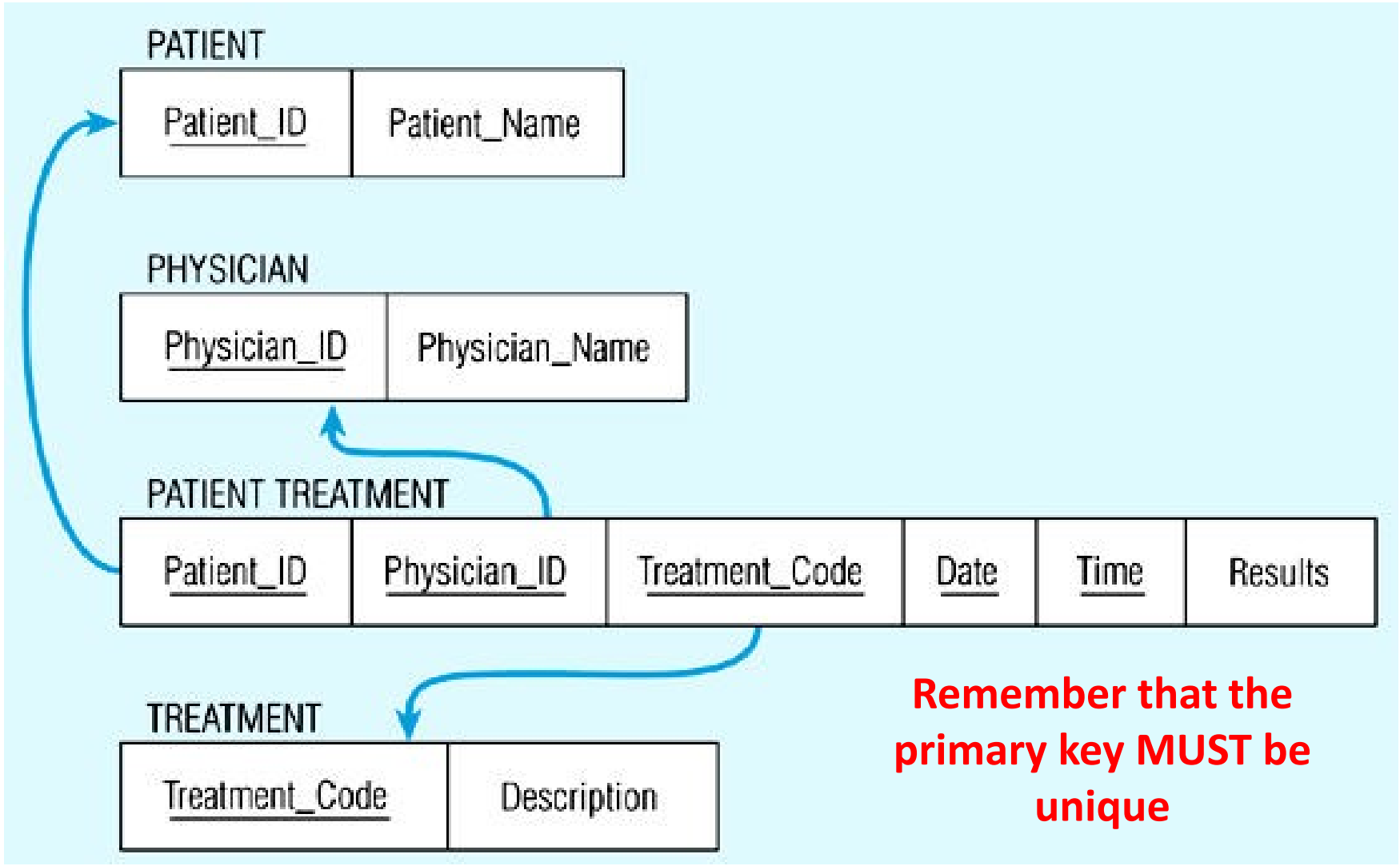
- Mapping Ternary (and n-ary) Relationships
  - One relation for each entity and one for the associative entity
  - Associative entity has foreign keys to each entity in the relationship

Mapping a Ternary Relationship

(a) Ternary relationship with associative entity



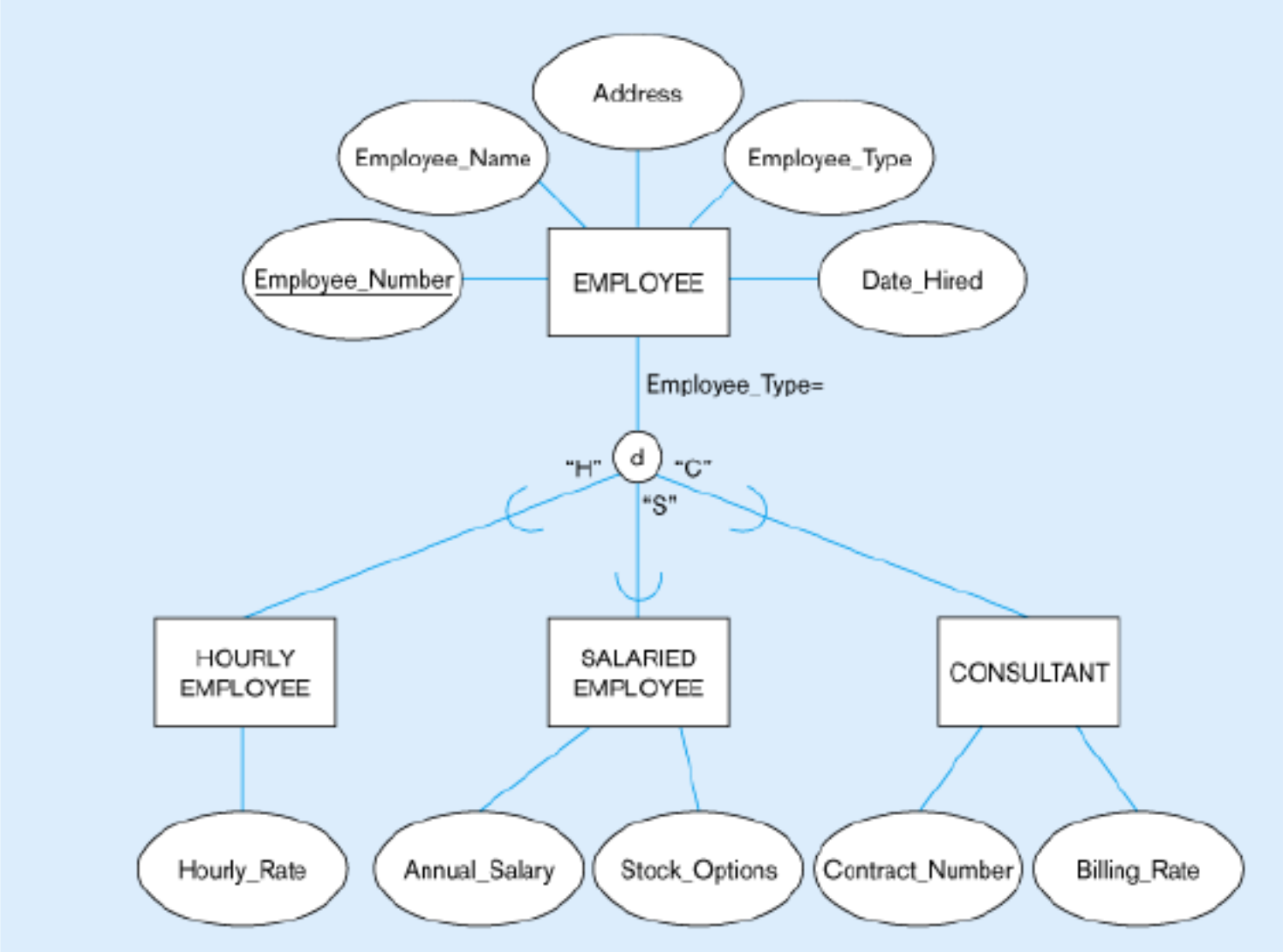
Mapping a Ternary Relationship (cont…)



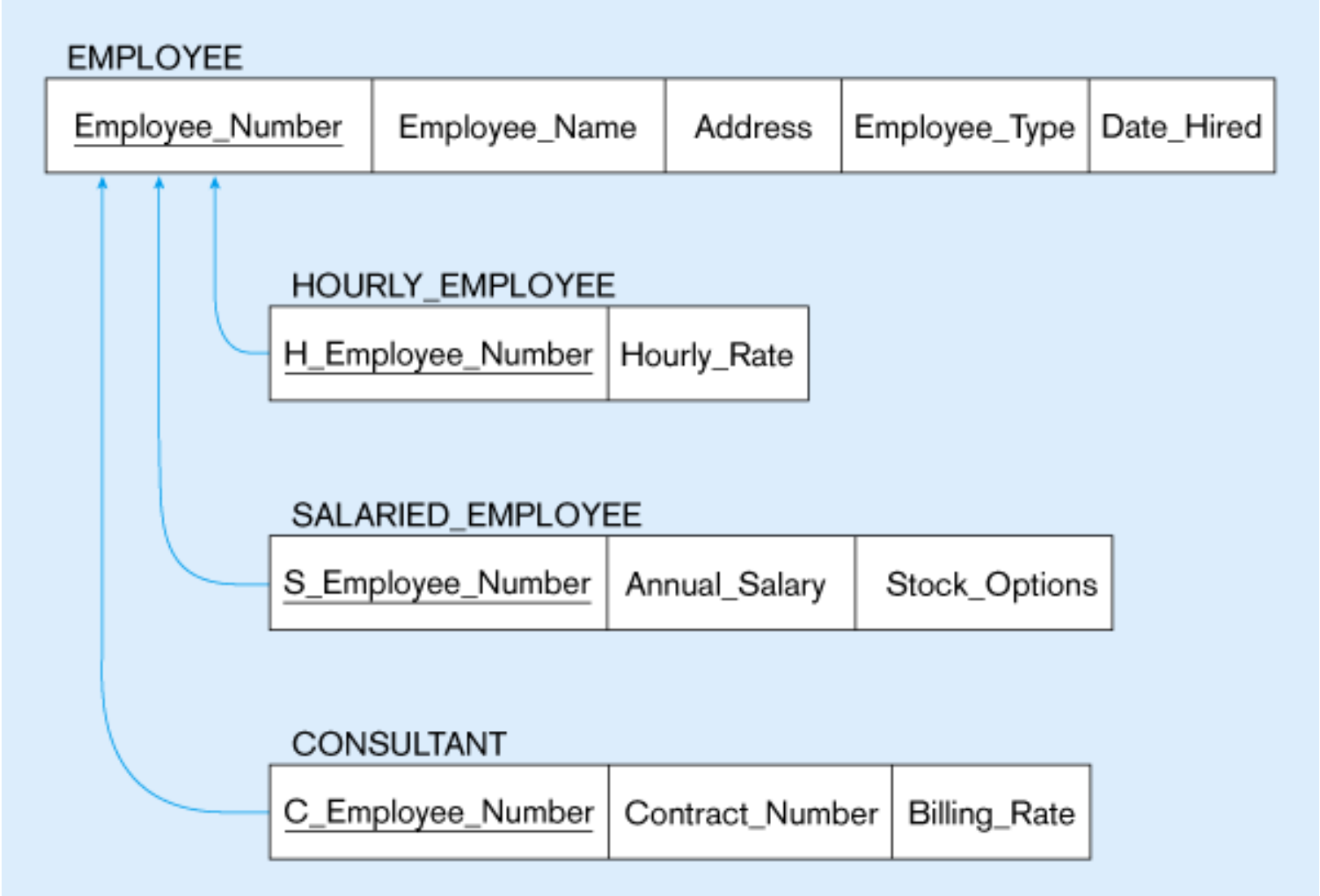
Transforming EER Diagrams into Relations

- Mapping Supertype/Subtype Relationships
  - One relation for supertype and for each subtype
  - Supertype attributes (including identifier and subtype discriminator) go into supertype relation
  - Subtype attributes go into each subtype; primary key of supertype relation also becomes primary key of subtype relation
  - 1:1 relationship established between supertype and each subtype, with supertype as primary table

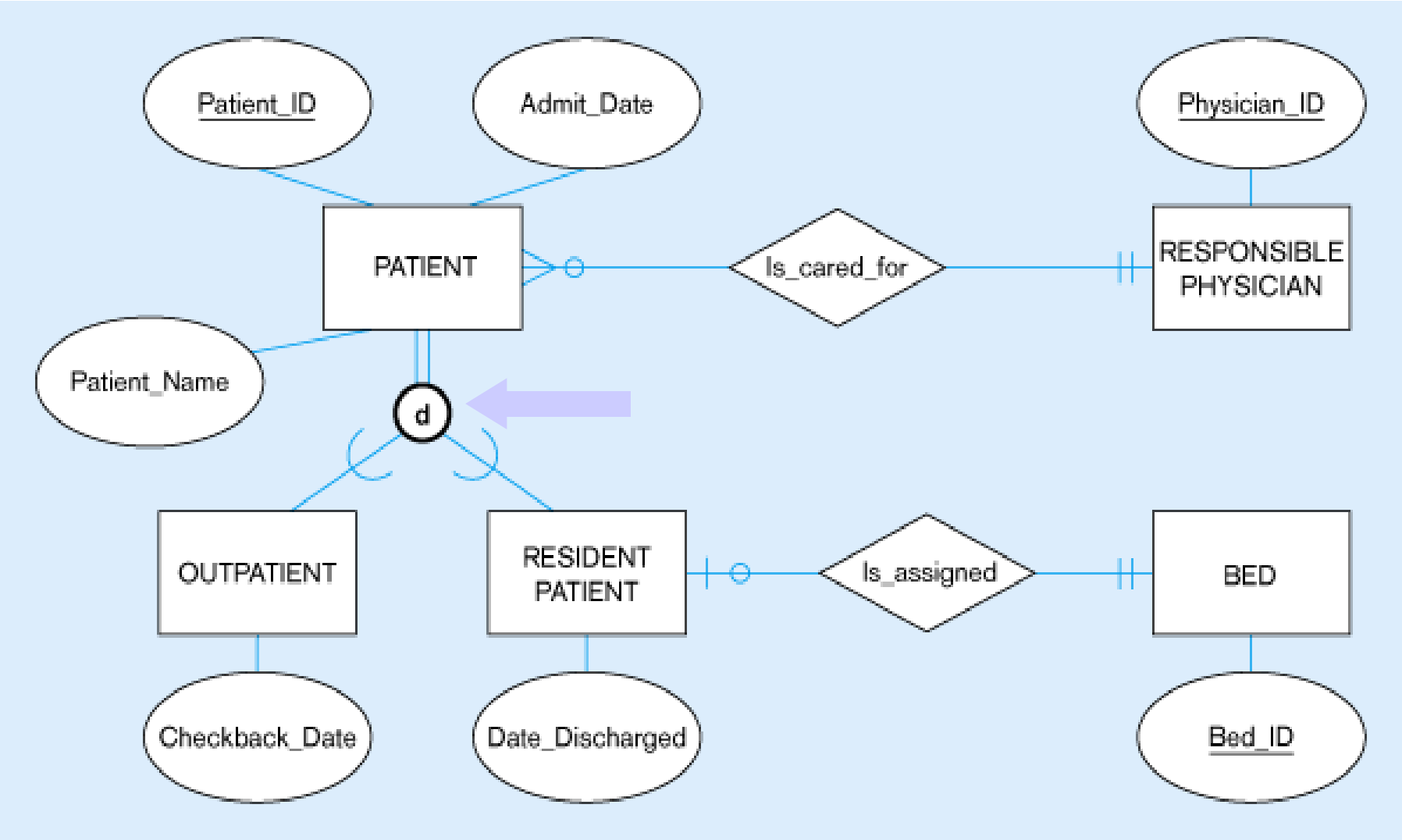
Supertype/Subtype Relationships



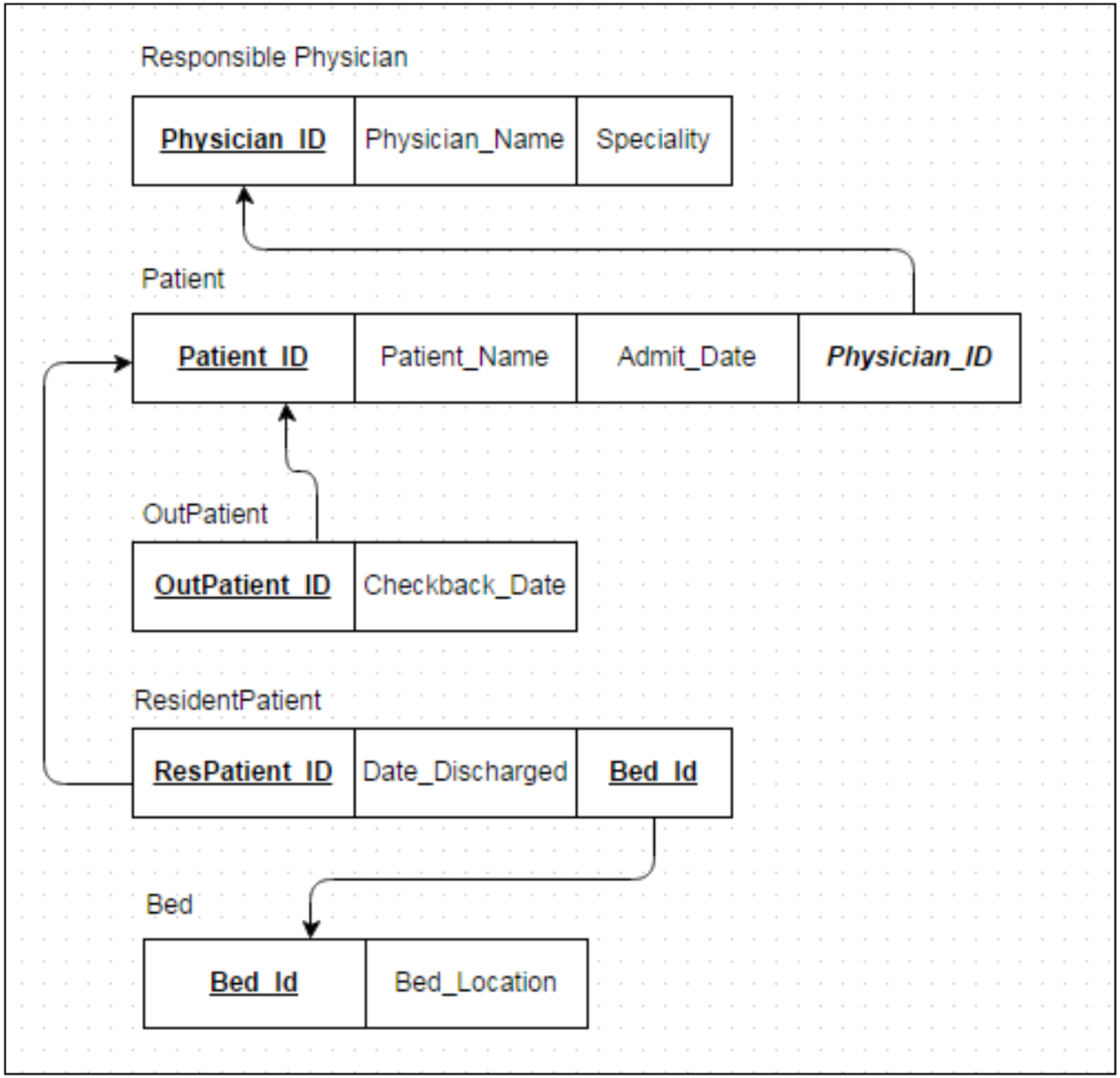
Mapping Supertype/Subtype Relationships to Relations



Example


Patient_ID
Admit_Date
Physician_ID
PATIENT
Is_cared_for
RESPONSIBLE PHYSICIAN
Patient_Name
d
OUTPATIENT
RESIDENT PATIENT
Is_assigned
BED
Checkback_Date
Date_Discharged
Bed_ID

Example – Solution



Summary

- Discussed relation, conversion of conceptual schema (ERD & EERD) into logical schema