Database Design: The Enhanced ER Model and Relational Transformations

University of California, Berkeley

School of Information

INFO 257: Database Management

Agenda

- Assignment 1 and 2a due
- Enhanced ER Model
- Relational Advantages and Disadvantages
- Normalization Workshop

Where are we?

	Operational	Informational	
	Transactional	Analytical– Data Warehousing	Analytical– Big Data
Technology	Relational	Relational	Non-relational
Modeling	Conceptual data modeling with (E)ER (~~Chapters 2 and 3~~)		
Design	Logical data modeling with the relational model; Normalization (~~Chapter 4~~)	Data warehousing and data integration (Chapter 9)	Big data technologies, including Hadoop & NoSQL (Chapter 10)
Infrastructure	Physical design of relational databases; Security; Cloud computing (Chapter 8)		
Access	SQL (~~Chapters 5 and 6~~) Applications with SQL (Chapter 7)		
Data analysis	Analytics and its implications (Chapter 11)		
Governance and data management	Lifecycle (~~Chapter 1~~) Governance, data quality, and master data management (Chapter 12)		

Enhanced ER Learning Objectives

3.1 Define terms

3.2 Understand use of supertype/subtype relationships

3.3 Use specialization and generalization techniques

3.4 Specify completeness and disjointness constraints

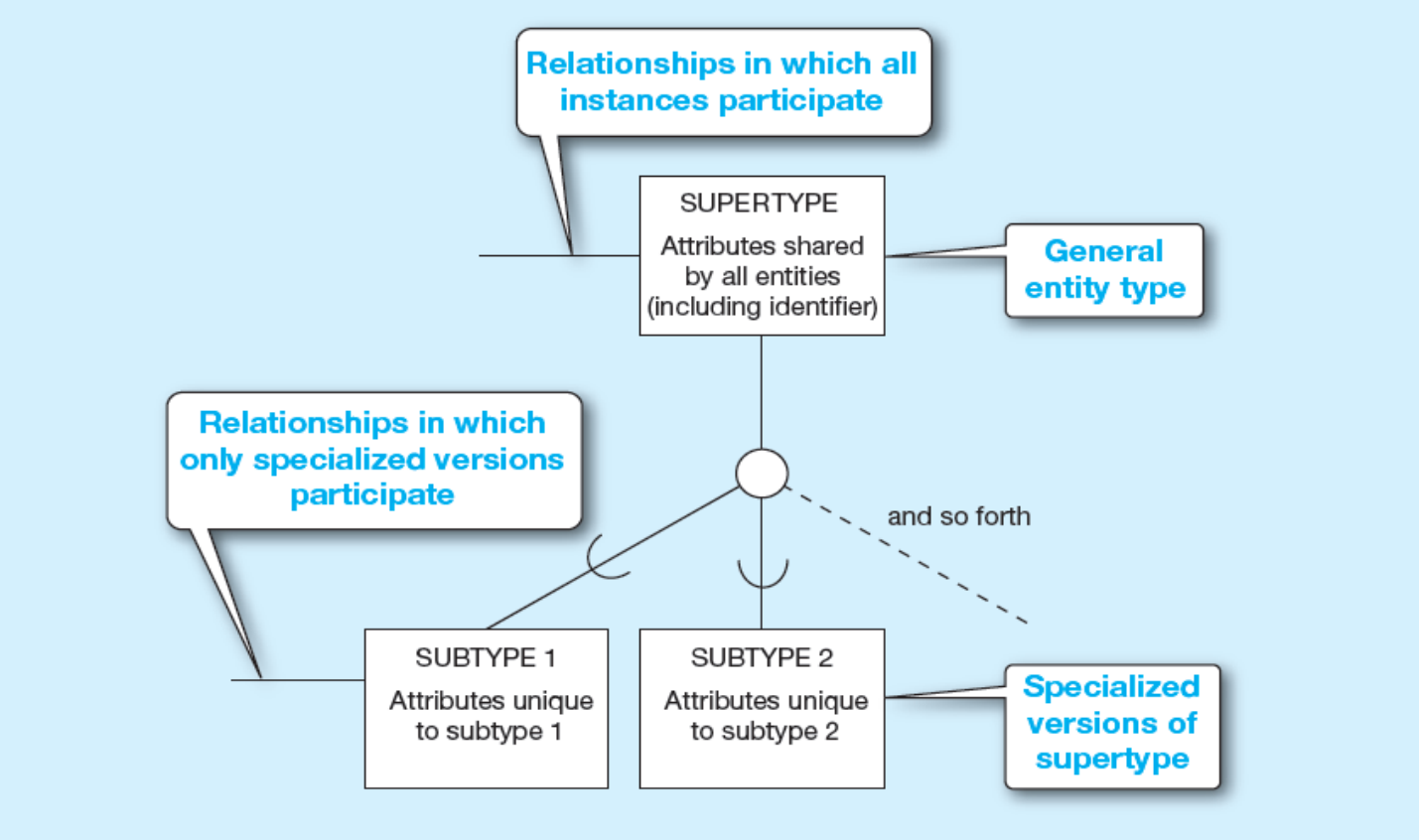
3.5 Develop supertype/subtype hierarchies for business situations

Supertypes and Subtypes

- **Enhanced E-R (EER) model:** extends original E-R model with new modeling constructs
- **Subtype:** A subgrouping of the entities in an entity type that has attributes distinct from those in other subgroupings
- **Supertype:** A generic entity type that has a relationship with one or more subtypes
- **Attribute Inheritance:**
  - Subtype entities inherit values of all attributes and relationships of the supertype
  - An instance of a subtype is also an instance of the supertype

Basic Notation for Supertype/Subtype Notation

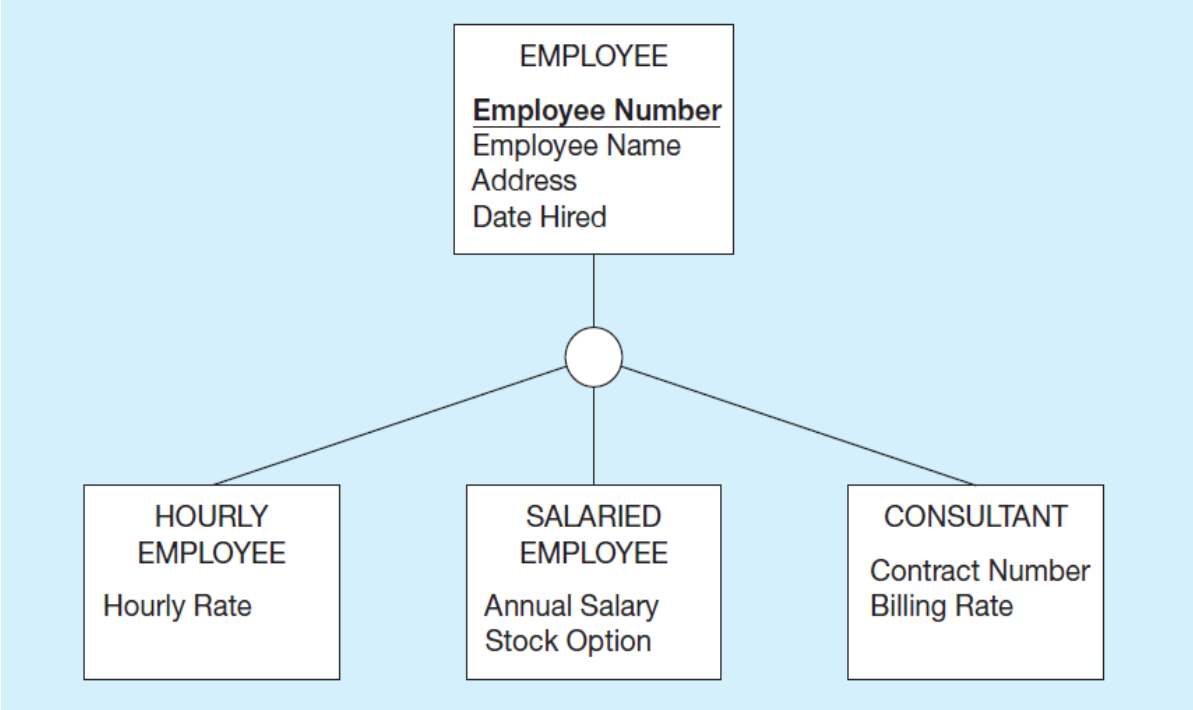
a) EER notation



Employee Supertype with Three Subtypes

All employee subtypes will have employee number, name, address, and date hired

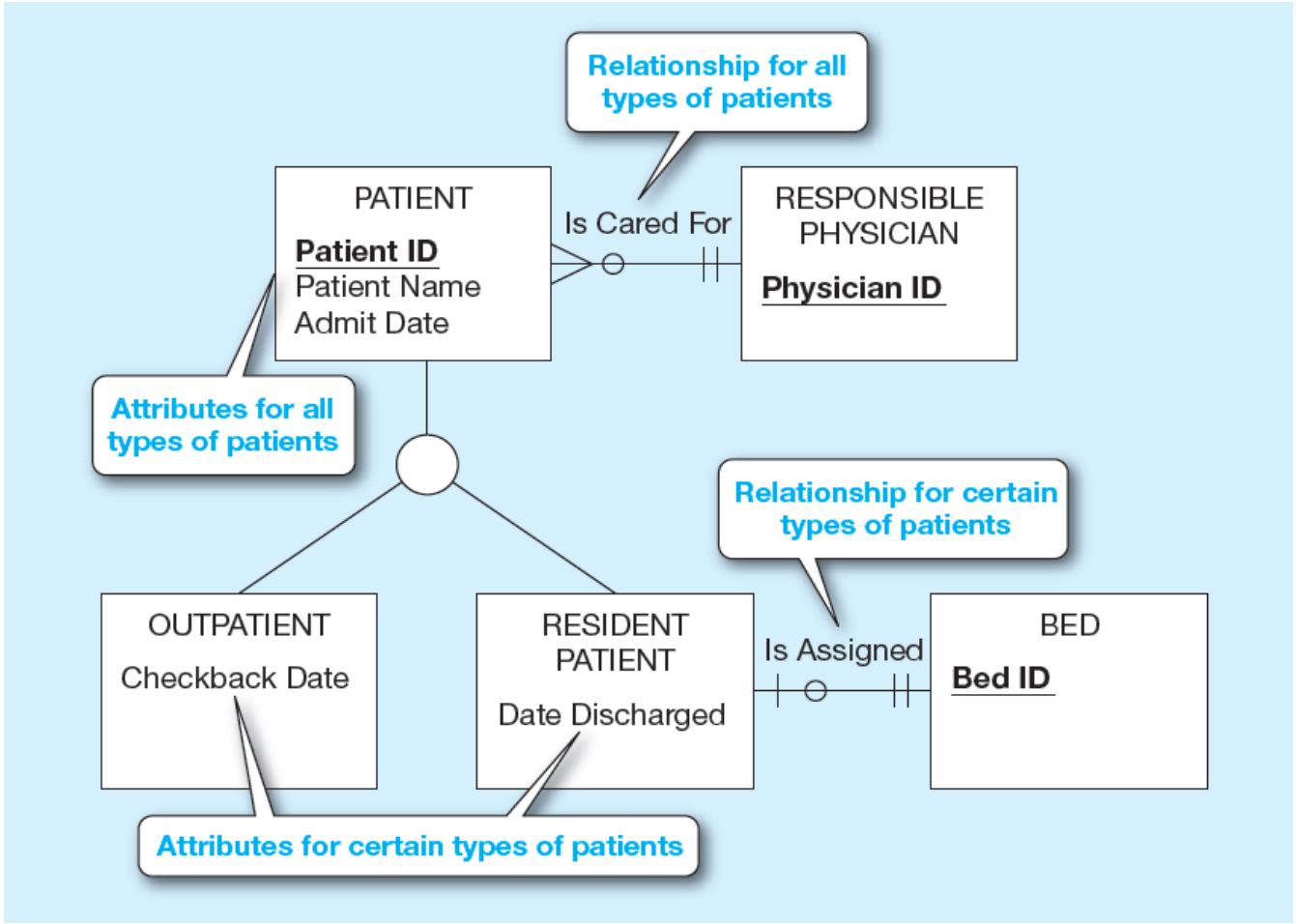
Each employee subtype will also have its own attributes



Relationships and Subtypes

- Relationships at the **supertype** level indicate that all subtypes will participate in the relationship
- The instances of a **subtype** may participate in a relationship unique to that subtype. In this situation, the relationship is shown at the subtype level

Supertype/Subtype Relationships in a Hospital



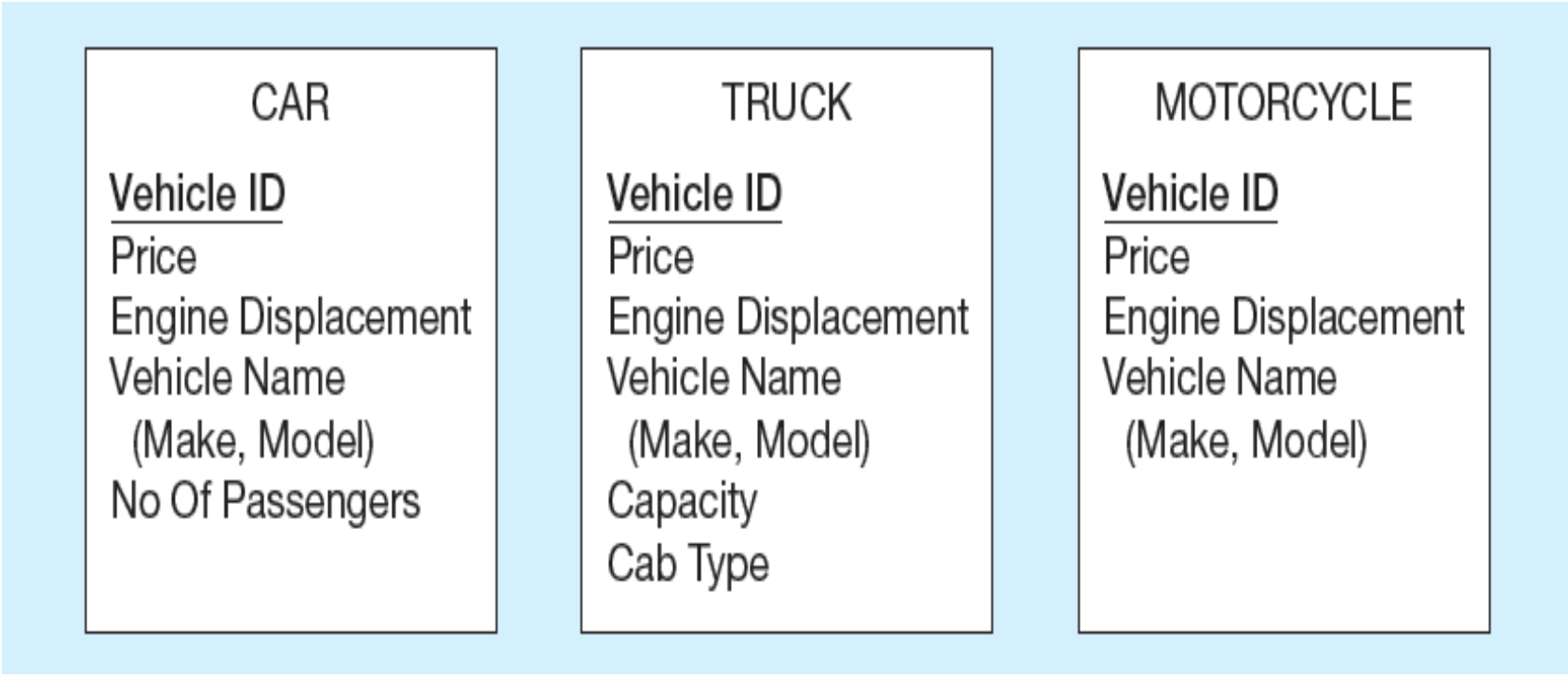
Generalization and Specialization

- **Generalization:** The process of defining a more general entity type from a set of more specialized entity types. BOTTOM-UP
- **Specialization:** The process of defining one or more subtypes of the supertype and forming supertype/subtype relationships. TOP-DOWN

Example of Generalization (1 of 2)

a) Three entity types: CAR, TRUCK, and MOTORCYCLE

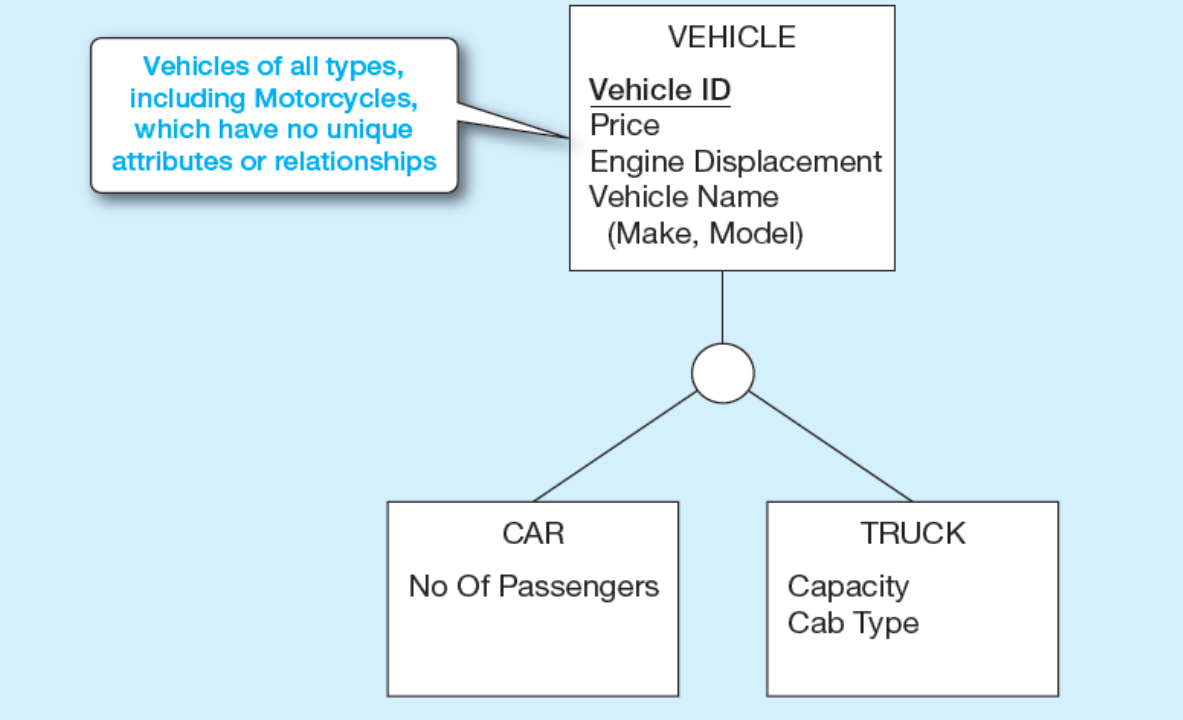
All these types of vehicles have common attributes



Example of Generalization (2 of 2)

b) Generalization to VEHICLE supertype

We put the shared attributes in a supertype. Note: no subtype for motorcycle, since it has no unique attributes

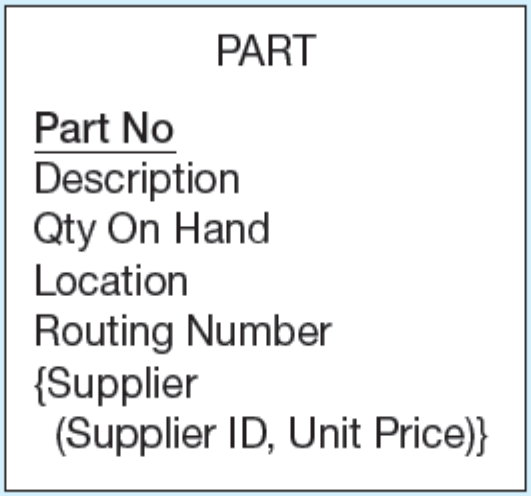


Example of Specialization (1 of 2)

a) Entity type PART

Note: Routing Number only applies if part is manufactured in house.

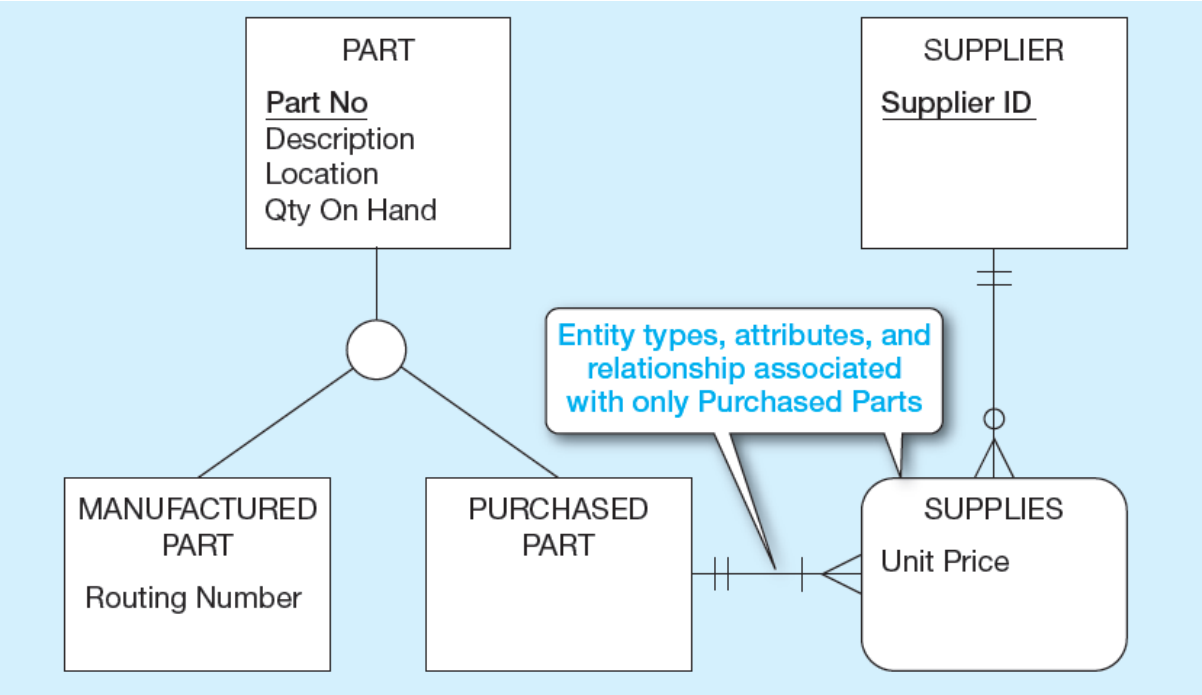
Supplier only applies if part is purchased from a supplier.



Example of Specialization (2 of 2)

b) Specialization to MANUFACTURED PART and PURCHASED PART

Multivalued composite attribute replaced by associative entity relationship to another entity

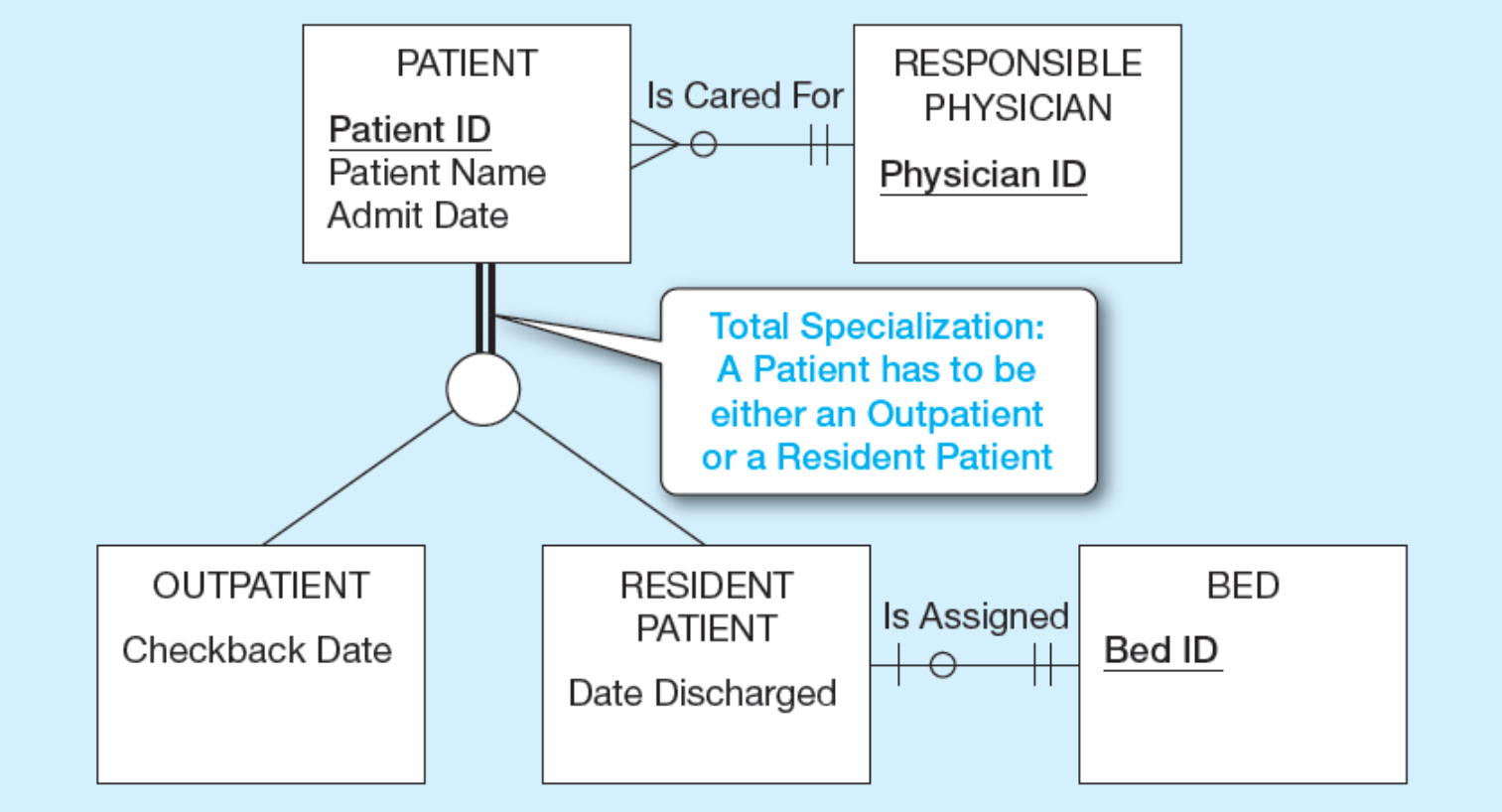


Constraints in Supertype/Subtype Relationships

- **Completeness Constraints**: Whether an instance of a supertype **must** also be a member of at least one subtype
  - Total Specialization Rule: Yes (double line)
  - Partial Specialization Rule: No (single line)

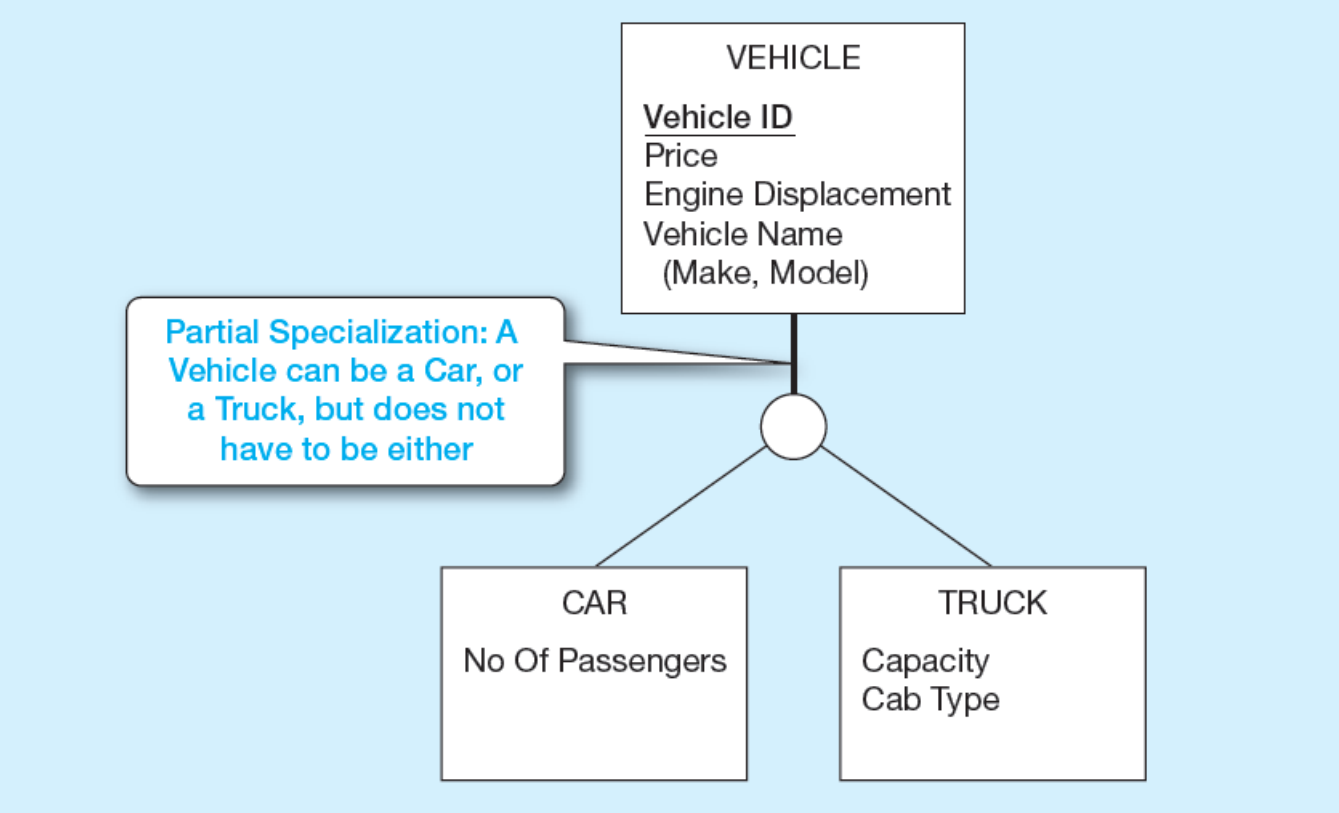
Examples of Completeness Constraints (1 of 2)

a) Total specialization rule



Examples of Completeness Constraints (2 of 2)

b) Partial specialization rule

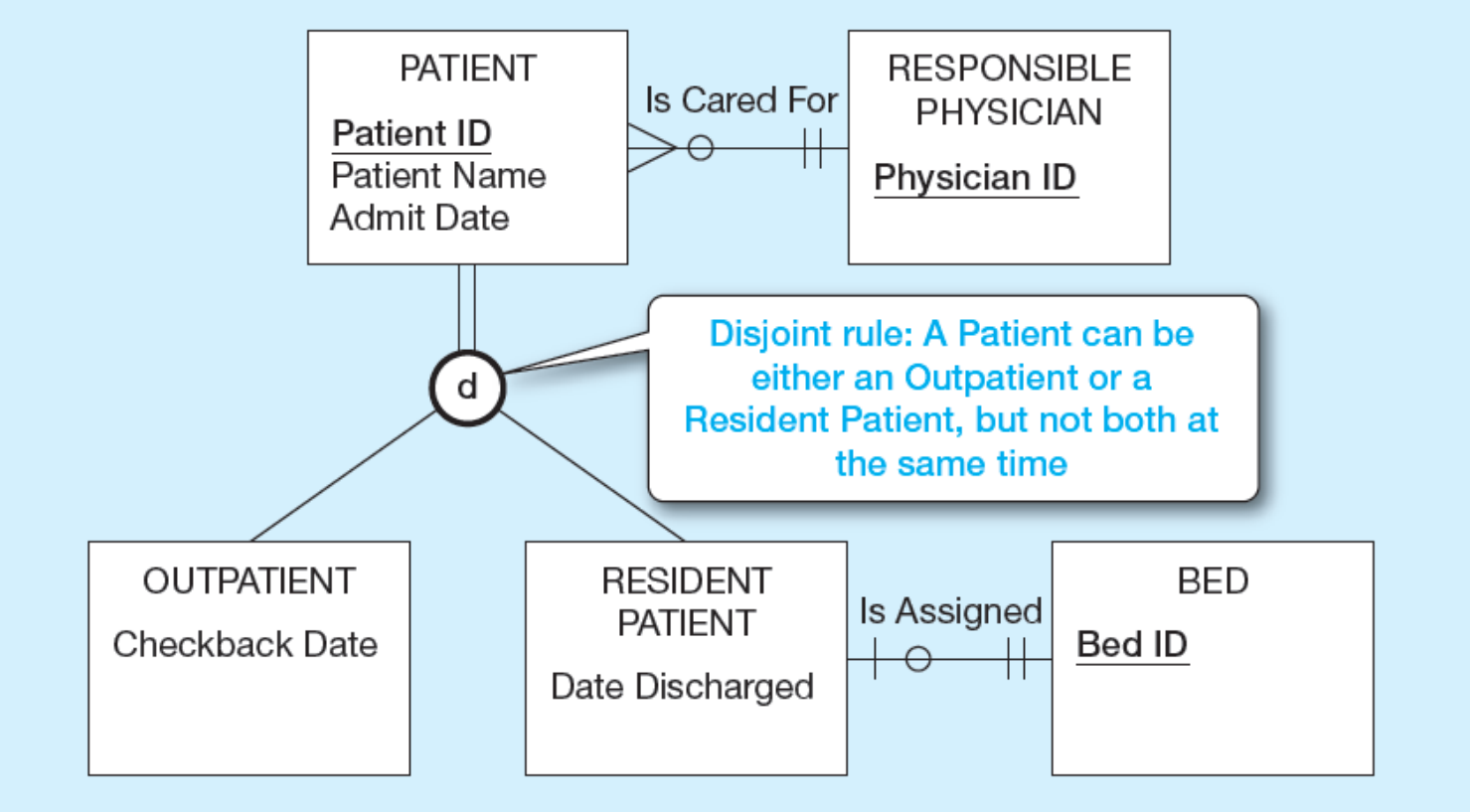


Constraints in Supertype/Subtype Relationships (1 of 2)

- **Disjointness Constraints**: Whether an instance of a supertype may **simultaneously** be a member of two (or more) subtypes
  - Disjoint Rule: An instance of the supertype can be only ONE of the subtypes
  - Overlap Rule: An instance of the supertype could be more than one of the subtypes

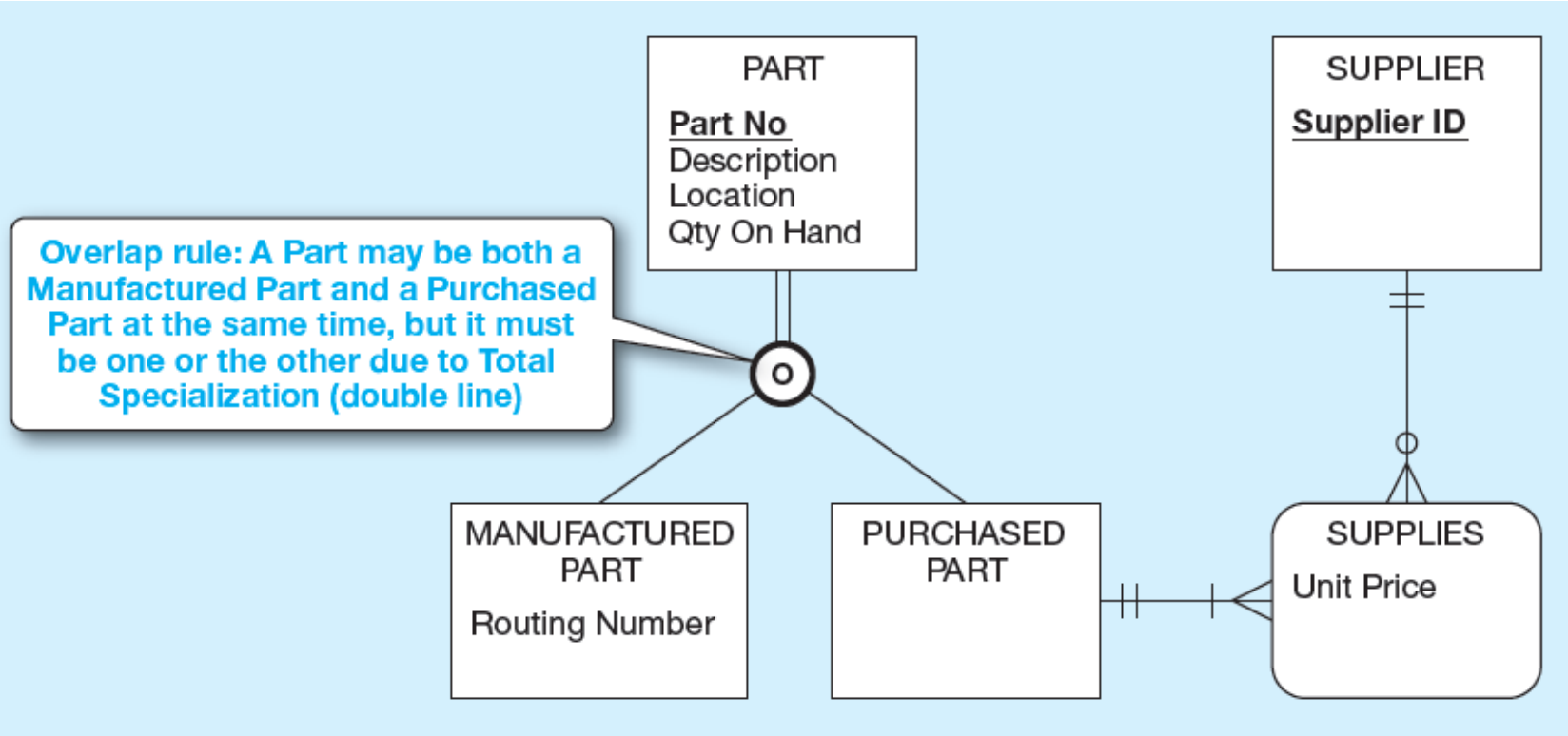
Examples of Disjointness Constraints (1 of 2)

a) Disjoint rule



Examples of Disjointness Constraints (2 of 2)

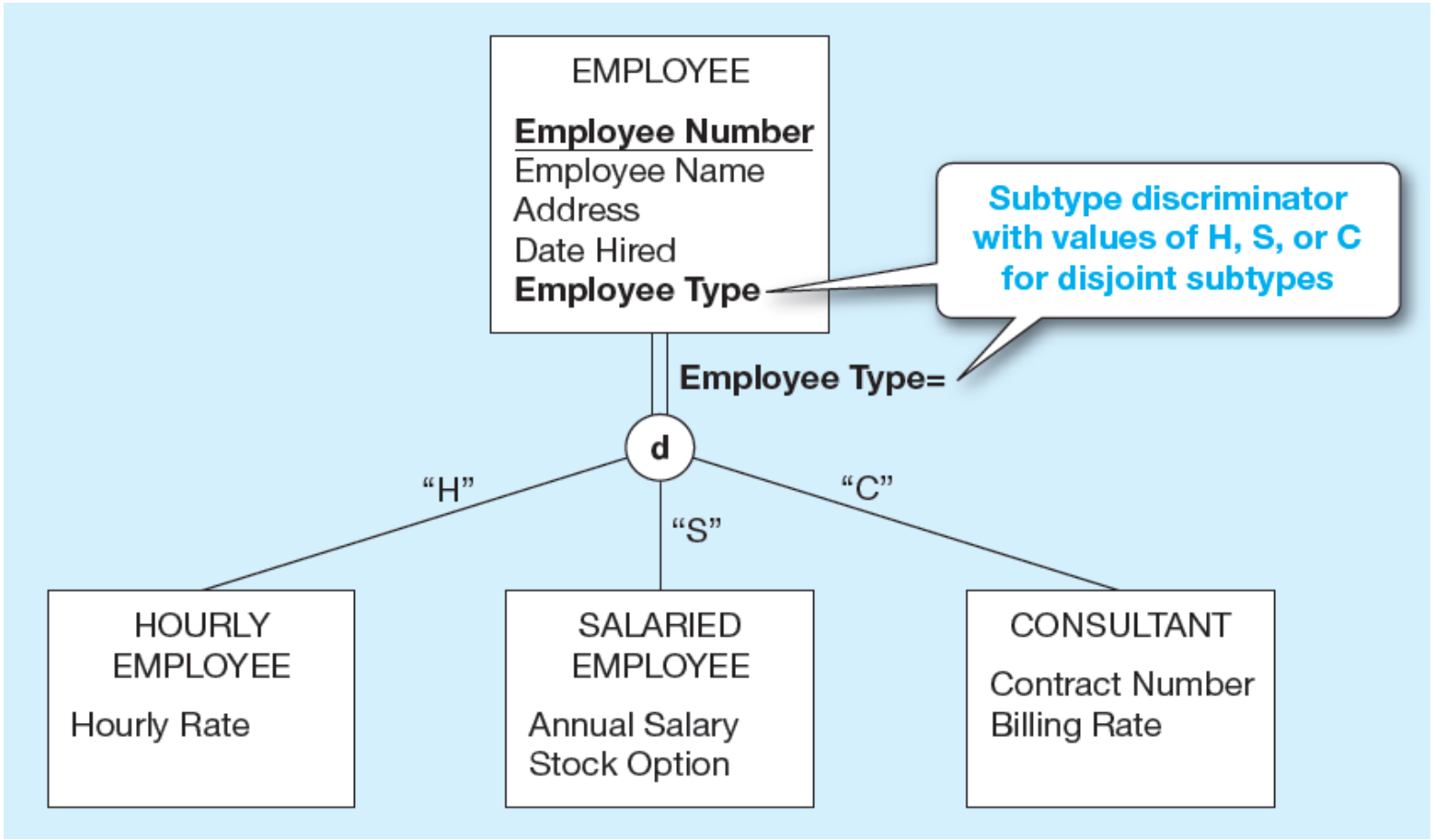
b) Overlap rule



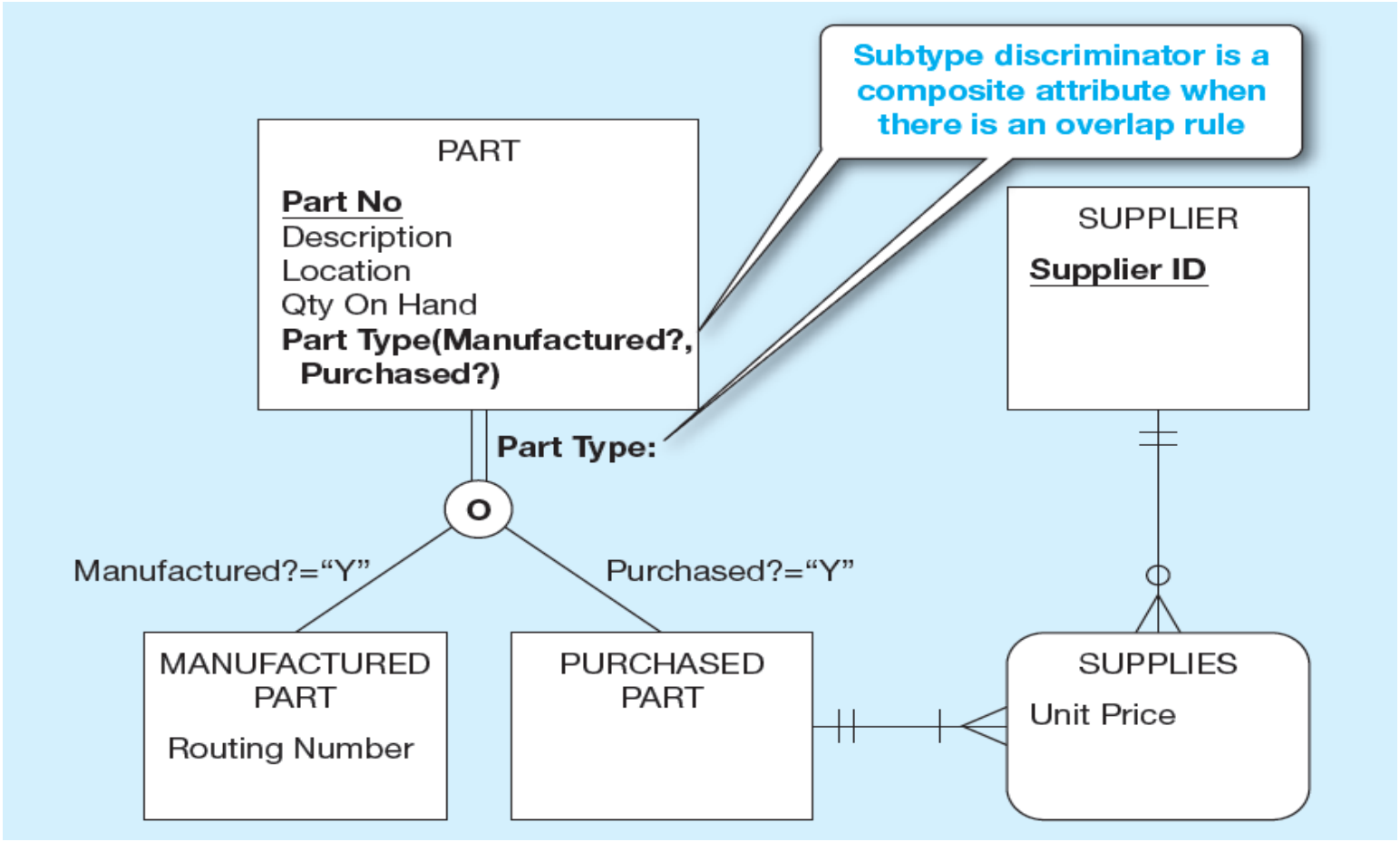
Constraints in Supertype/Subtype Relationships

- **Subtype Discriminator**: An attribute of the supertype whose values determine the target subtype(s)
  - **Disjoint** – a **simple** attribute with alternative values to indicate the possible subtypes
  - **Overlapping** – a **composite** attribute whose subparts pertain to different subtypes. Each subpart contains a Boolean value to indicate whether or not the instance belongs to the associated subtype

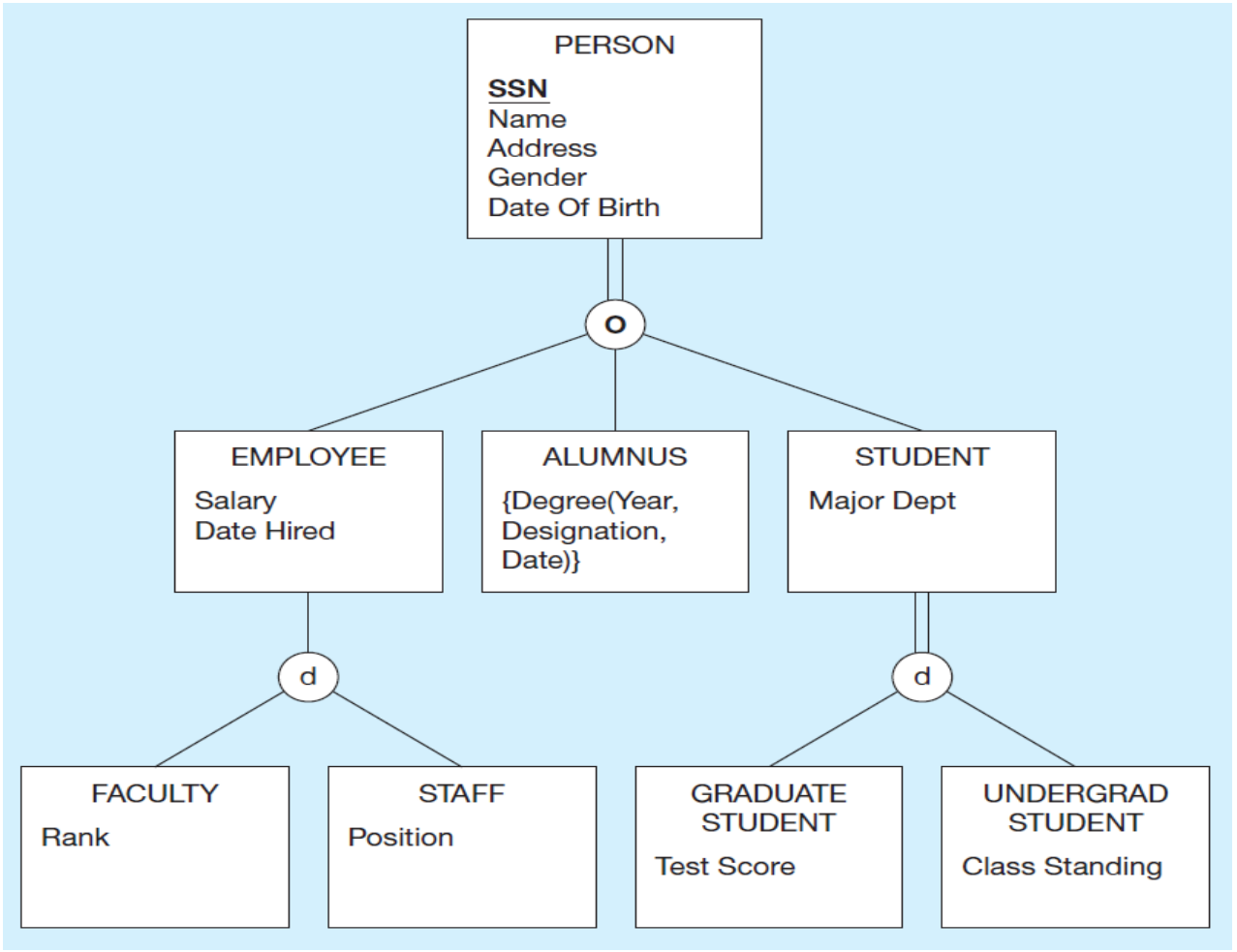
Introducing a Subtype Discriminator (Disjoint Rule)



Subtype Discriminator (Overlap Rule)



Example of Supertype/Subtype Hierarchy



Transforming EER Diagrams into Relations (1 of 7)

- Mapping Regular Entities to Relations
  - Simple attributes: E-R attributes map directly onto the relation
  - Composite attributes: Use only their simple, component attributes
  - Multivalued attributes: Become a separate relation with a foreign key taken from the superior entity

Example of Mapping a Regular Entity

a) CUSTOMER entity type

b) CUSTOMER relation

CUSTOMER

CustomerID	CustomerName	CustomerAddress	CustomerPostalCode

Mapping a Composite Attribute

a) CUSTOMER entity type with composite attribute

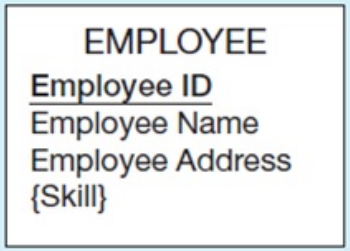
b) CUSTOMER relation with address detail

CUSTOMER

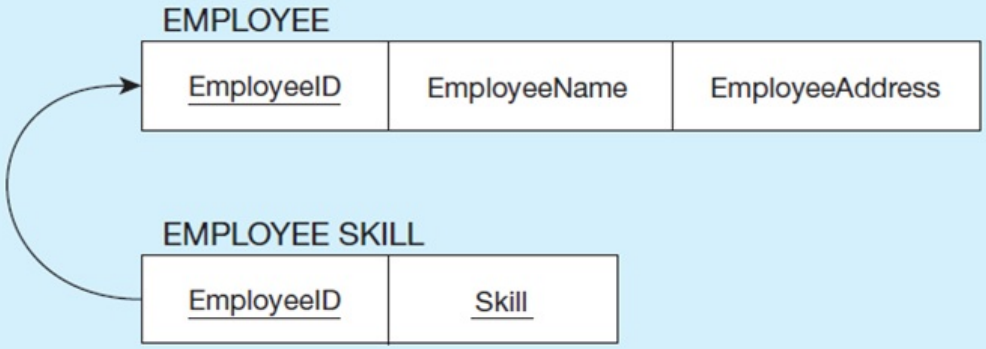
CustomerID	CustomerName	CustomerStreet	CustomerCity	CustomerState	CustomerPostalCode

Mapping an Entity with a Multivalued Attribute

a) EMPLOYEE entity type with multivalued attribute



b) EMPLOYEE and EMPLOYEE SKILL relations

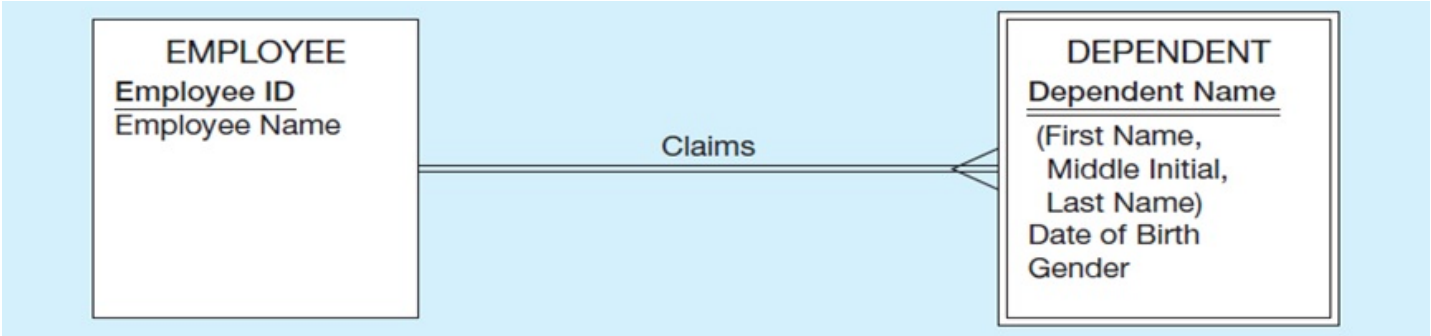


Transforming EER Diagrams into Relations (2 of 7)

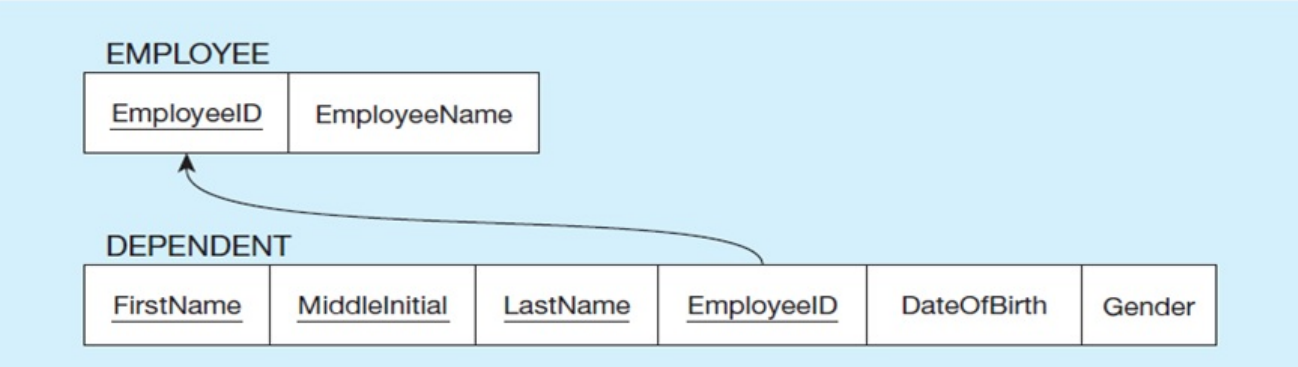
- Mapping Weak Entities
  - Becomes a separate relation with a foreign key taken from the superior entity
  - Primary key composed of:
    - Partial identifier of weak entity
    - Primary key of identifying relation (strong entity)

Example of Mapping a Weak Entity

a) Weak entity DEPENDENT



b) Relations resulting from weak entity

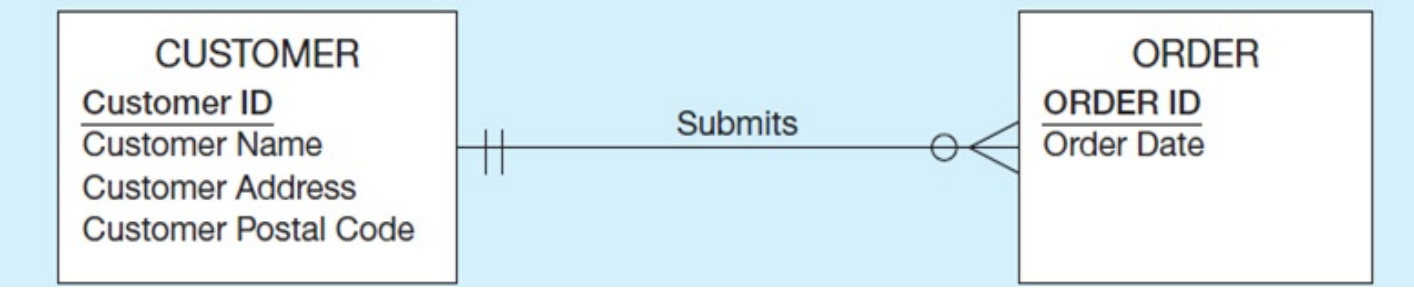


Transforming EER Diagrams into Relations (3 of 7)

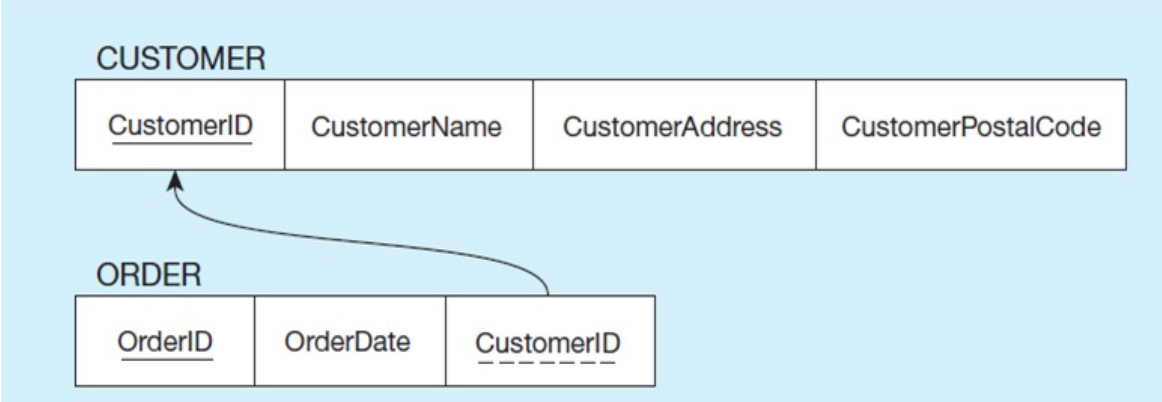
- Mapping Binary Relationships
  - **One-to-Many** – Primary key on the one side becomes a foreign key on the many side
  - **Many-to-Many** – Create a **new relation** with the primary keys of the two entities as its primary key
  - **One-to-One** – Primary key on mandatory side becomes a foreign key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

Example of Mapping a 1:N Relationship

a) Relationship between customers and orders

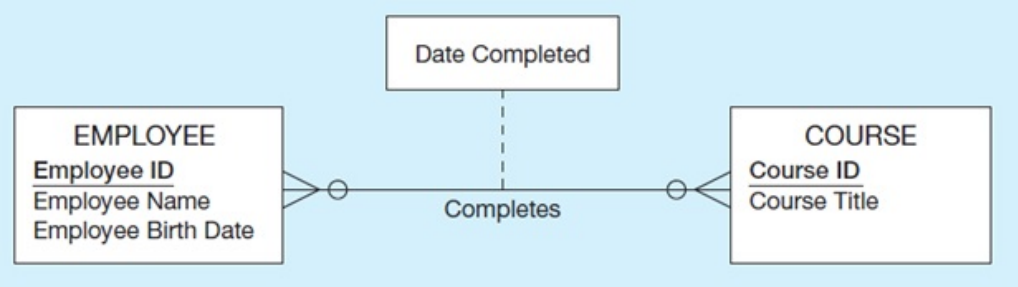


b) CUSTOMER and ORDER relations with a foreign key in ORDER

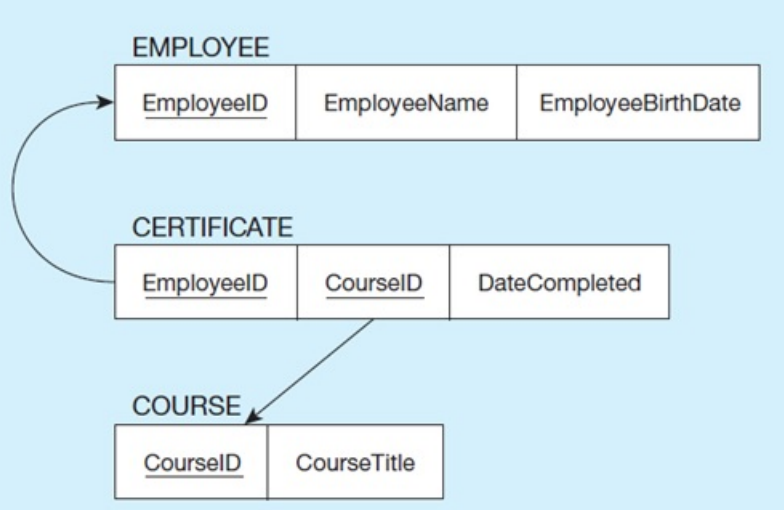


Example of Mapping an M:N Relationship

a) Completes relationship (M:N)

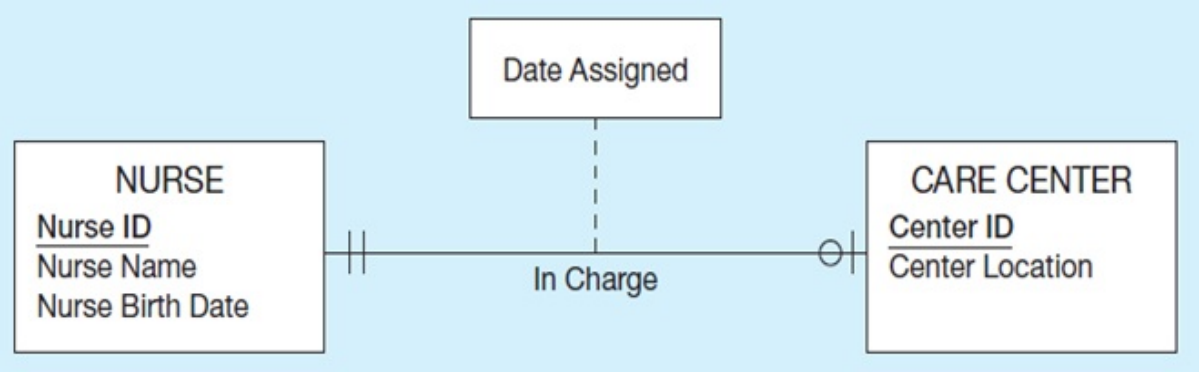


b) Three resulting relations

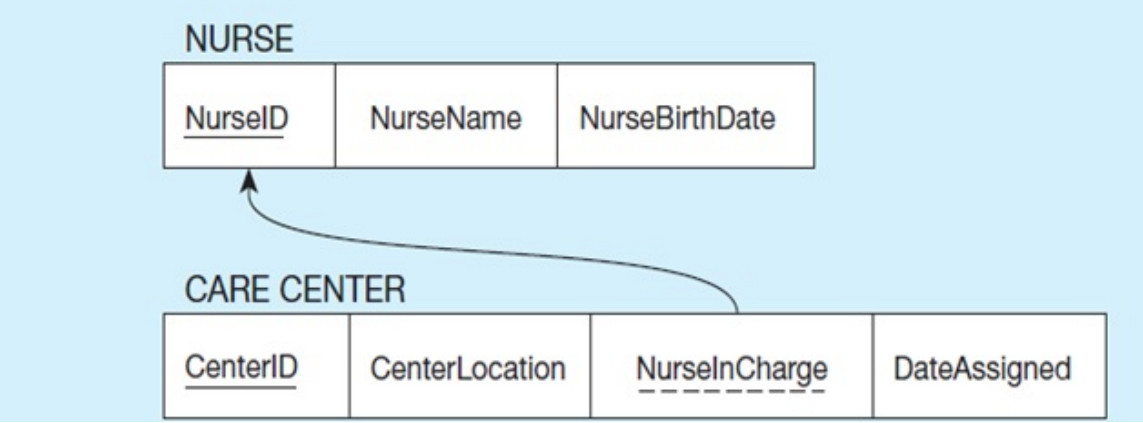


Example of Mapping a Binary 1:1 Relationship

a) In charge relationship (binary 1:1)



b) Resulting relations

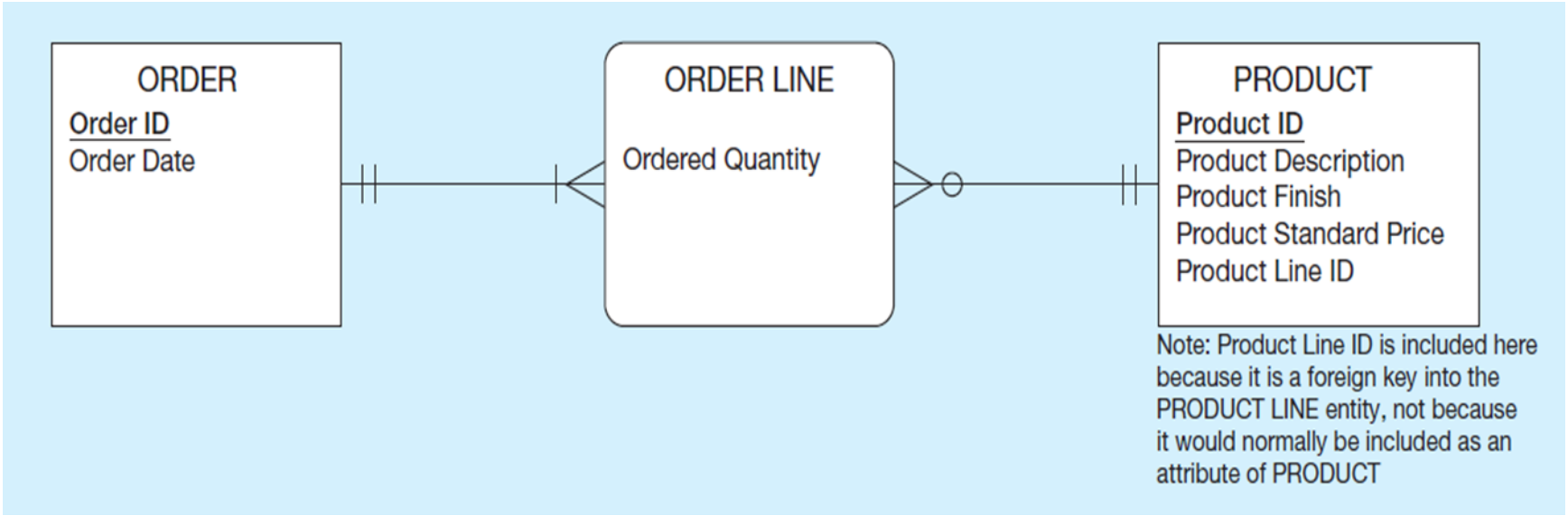


Transforming EER Diagrams into Relations (4 of 7)

- 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

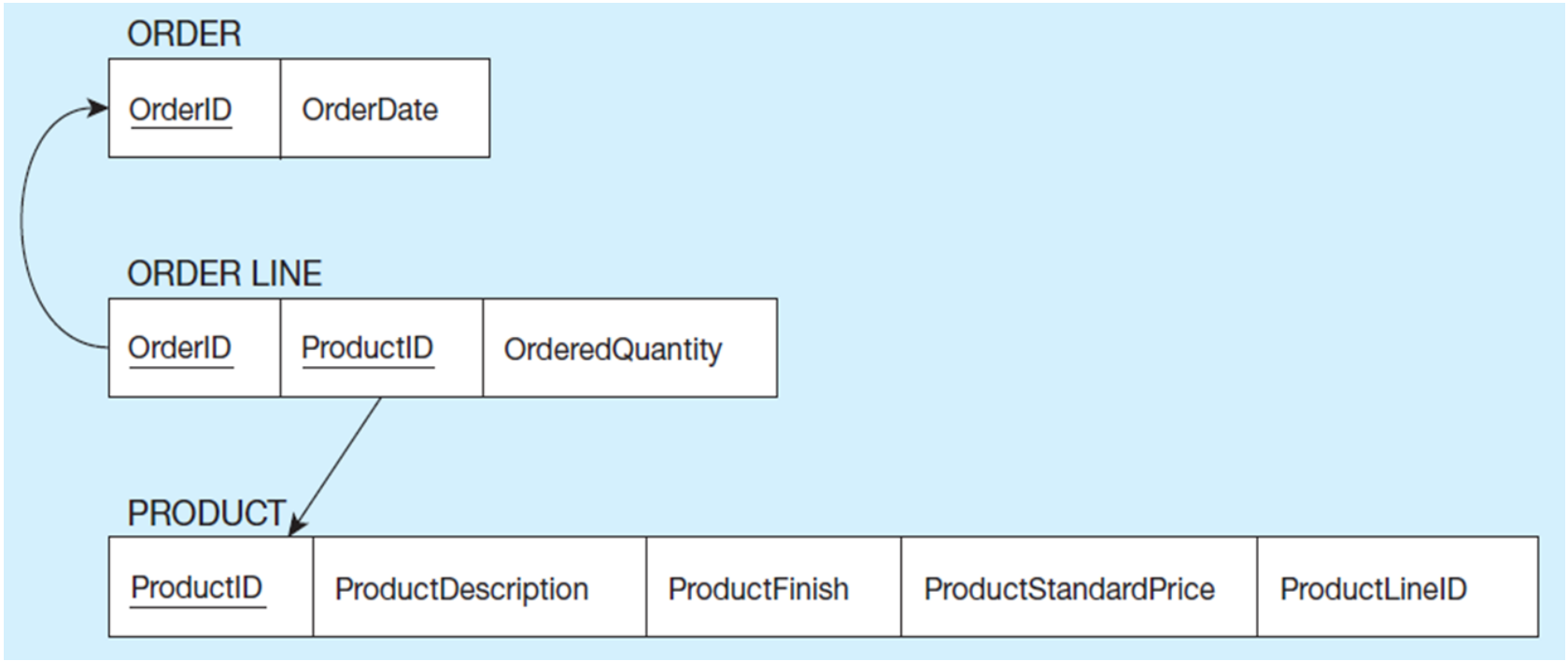
Example of Mapping an Associative Entity (1 of 2)

a) An associative entity



Example of Mapping an Associative Entity (2 of 2)

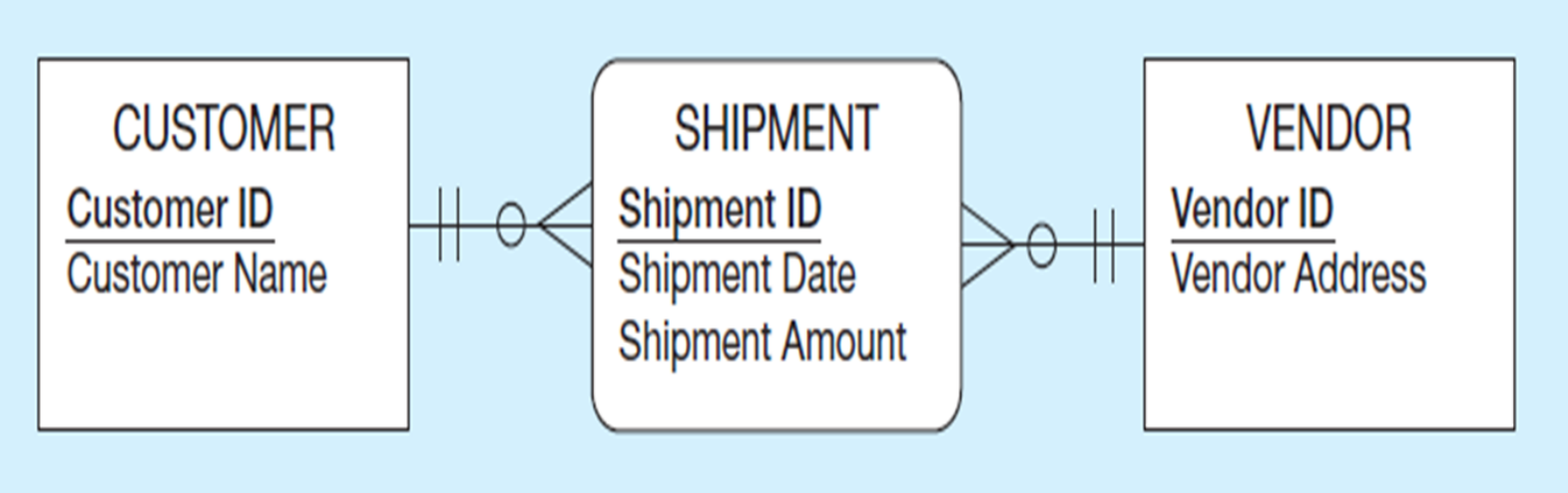
b) Three resulting relations



Composite primary key formed from the two foreign keys

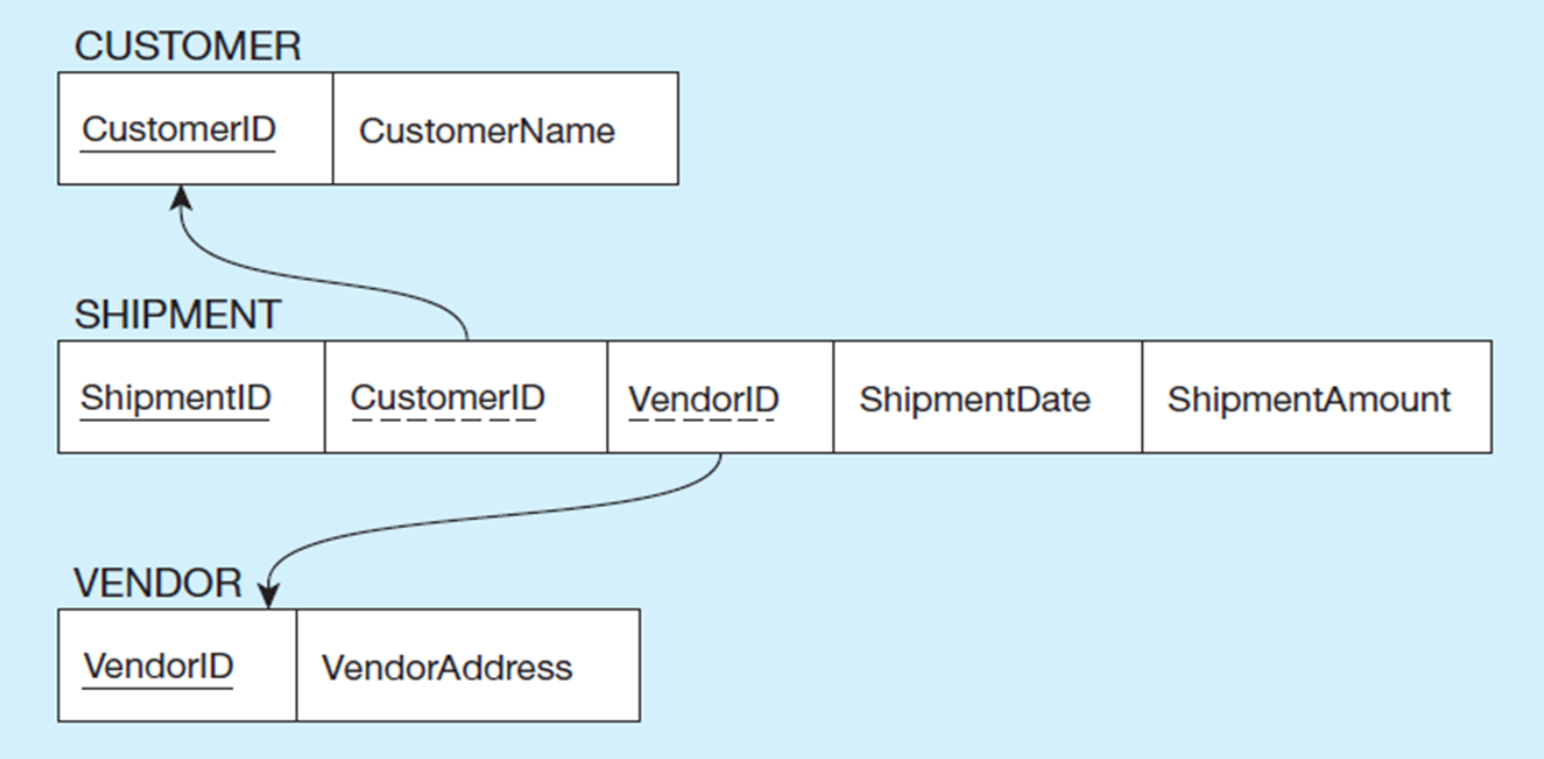
Example of Mapping an Associative Entity with an Identifier (1 of 2)

a) SHIPMENT associative entity



Example of Mapping an Associative Entity with an Identifier (2 of 2)

b) Three resulting relations



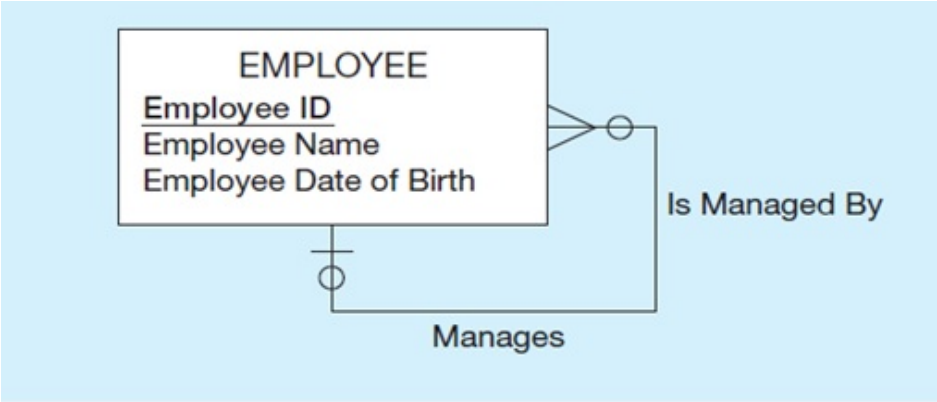
Primary key differs from foreign keys

Transforming EER Diagrams into Relations (5 of 7)

- 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

Example of Mapping a Unary 1:N Relationship

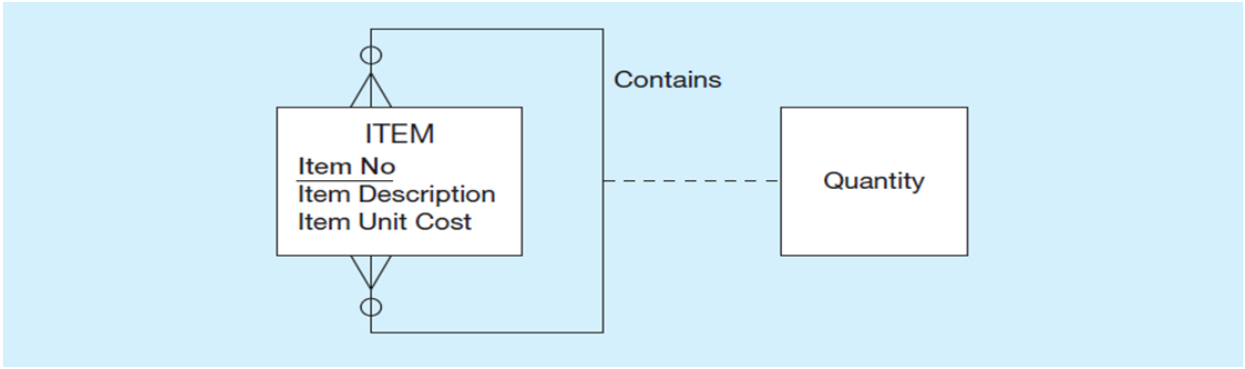
a) EMPLOYEE entity with unary relationship



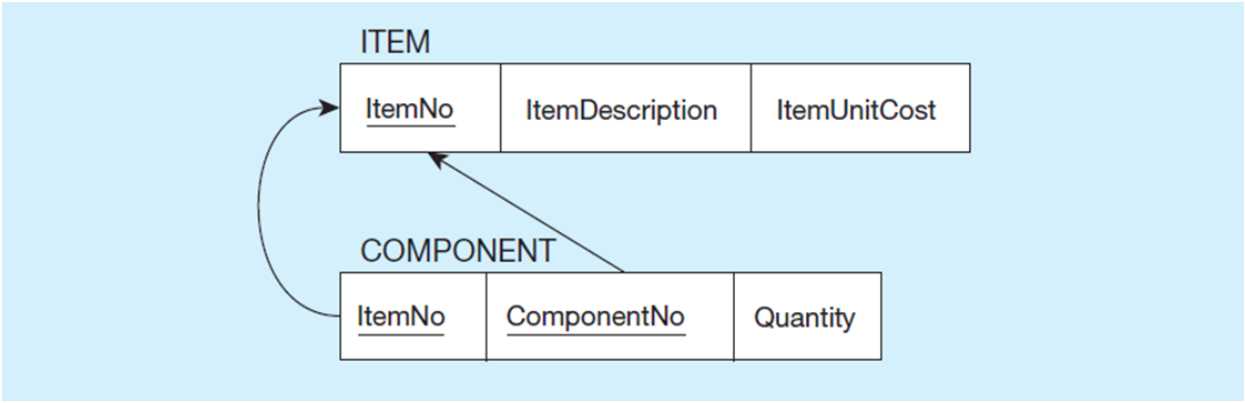
b) EMPLOYEE relation with recursive foreign key

Example of Mapping a Unary M:N Relationship

a) ITEM entity with unary relationship



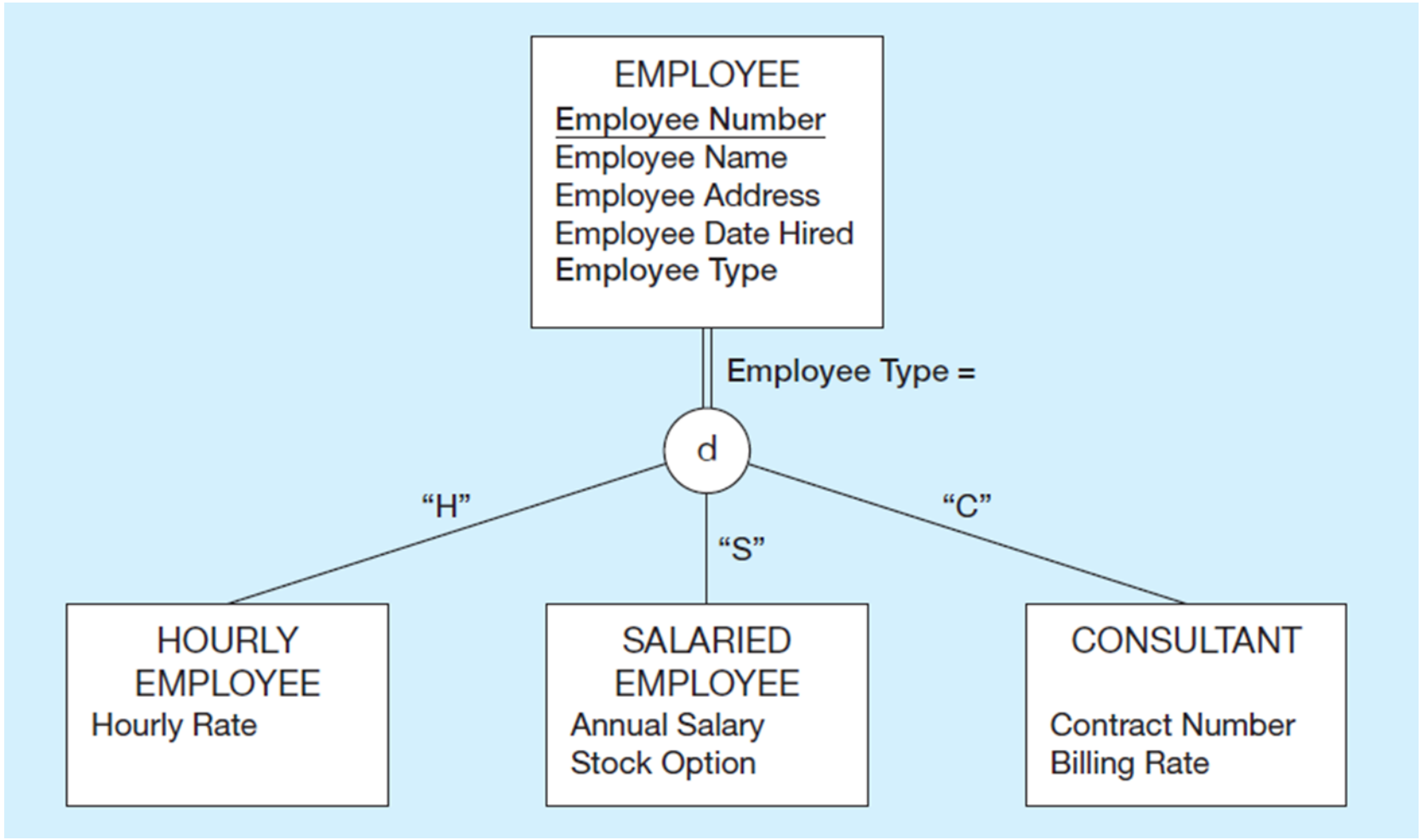
b) ITEM relation with recursive foreign key



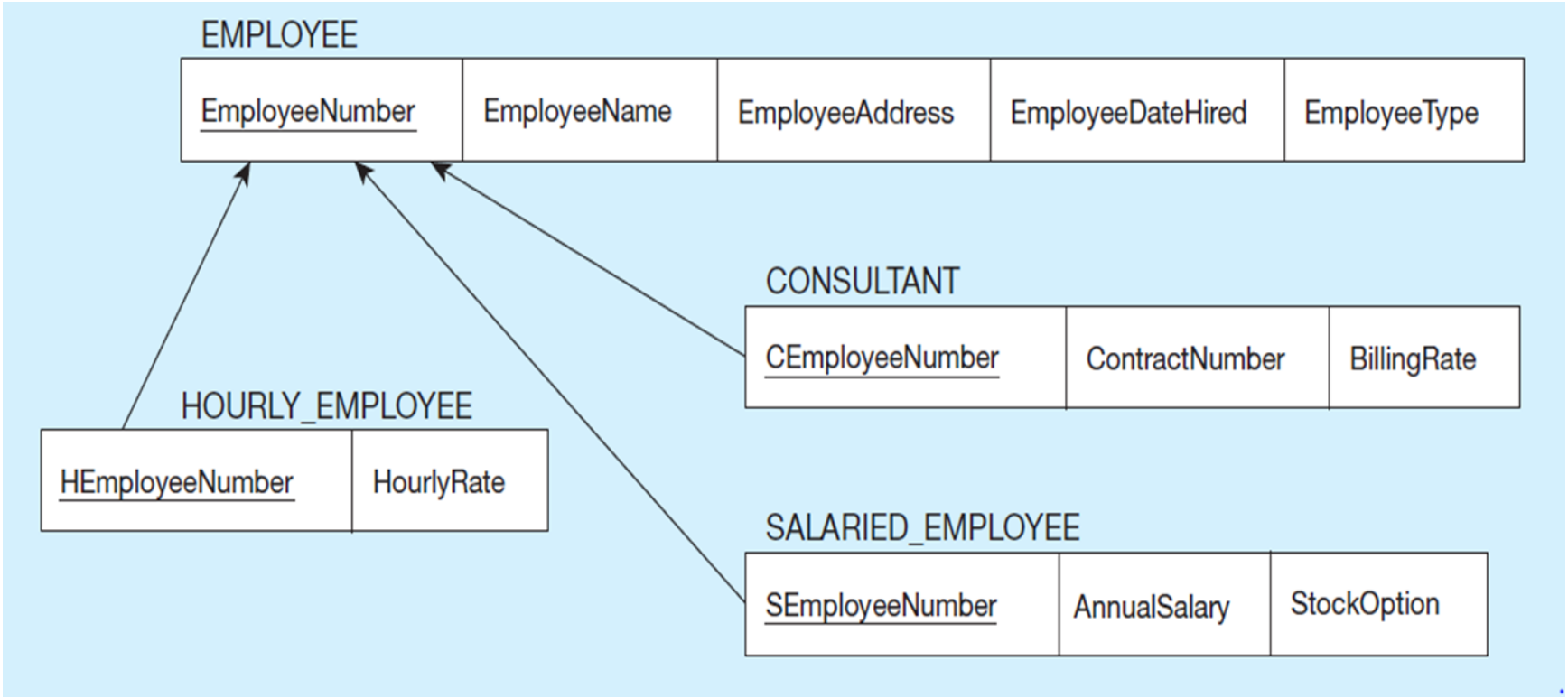
Transforming EER Diagrams into Relations (7 of 7)

- 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



Example of Mapping Supertype/Subtype Relationships to Relations



These are implemented as one-to-one relationships.

Merging Relations

- View Integration – Combining entities from multiple E-R models into common relations
- Issues to watch out for when merging entities from different E-R models:
  - Synonyms – two or more attributes with different names but same meaning
  - Homonyms – attributes with same name but different meanings
  - Transitive dependencies – even if relations are in 3NF prior to merging, they may not be after merging
  - Supertype/subtype relationships – may be hidden prior to merging