Lecture 5:
Advanced Data Modeling

The Enhanced E-R Model and Business Rules

TRANSFORMING ERD DIAGRAMS INTO RELATION

- Conceptual design: create ERD
- Logical Design: transform the E-R (and EER) diagram into Relational database schemas.
- Many CASE tools can automatically performs many of the conversion process.

ERD into Relations

- Understands the steps in this process for three reasons:
  - CASE tools often cannot model complex data relationships such as ternary relationships and supertype/subtype relationship.
  - Legitimate alternatives where you will need to choose a particular solution.
  - Must be prepare to perform a quality check on the results obtained with a CASE tool.

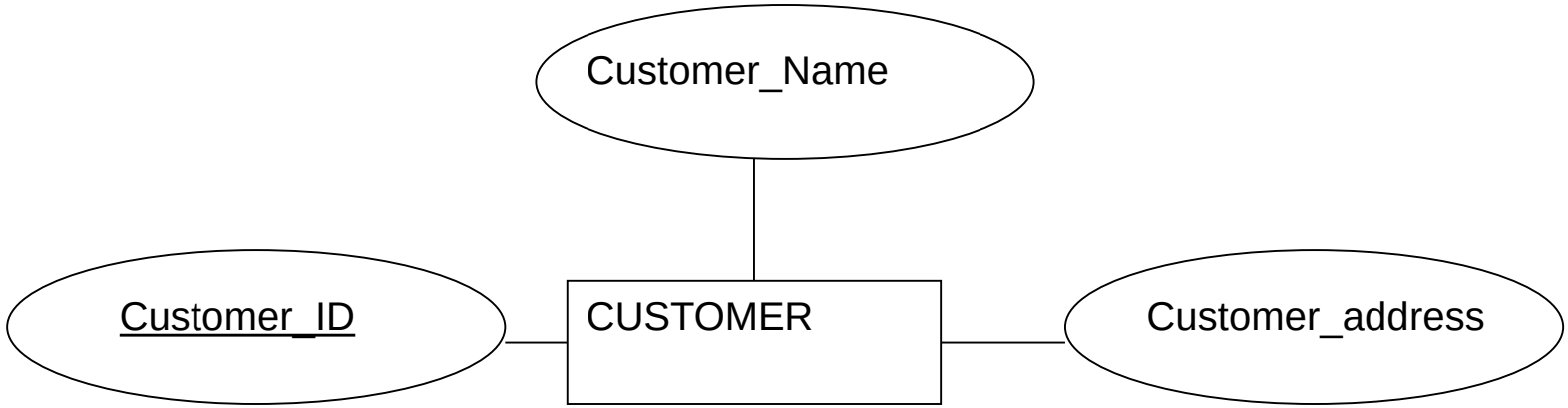
Entity mapping

- Three types of entities:
- Regular entities.
- Weak entities.
- Associative entities.

Step1: Regular Entity mapping

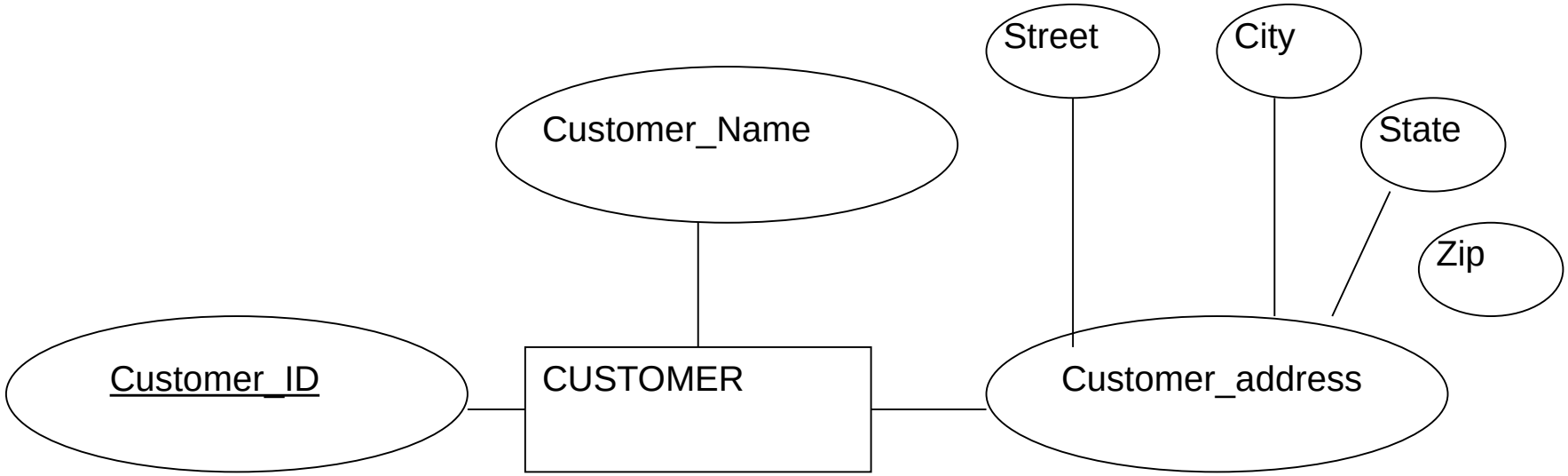
- Each regular entity type in ERD is transformed into a relation.
- The name of relation is same as the entity name.
- Each simple attribute of the entity type becomes an attribute of the relation.
- The identifier of the entity type becomes the primary key of the corresponding relation.

Regular Entity mapping



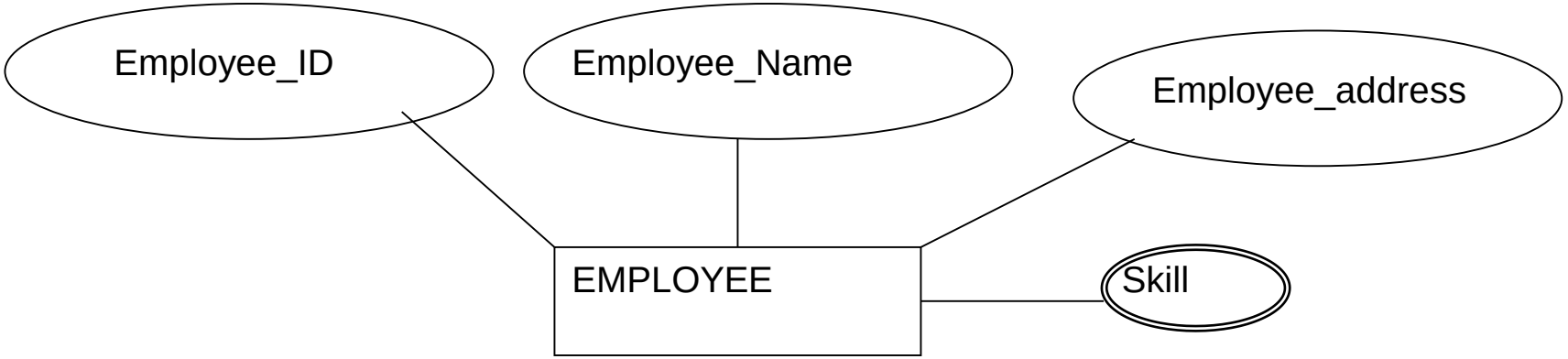
Customer_ID	Customer_Name	Customer_Address

Entity mapping with Composite Attributes



Customer_ID	Customer_Name	Street	City	State	Zip

Entity mapping with Multivalued Attributes



EMPLOYEE

Employee_ID	Employee_Name	Employee_Address

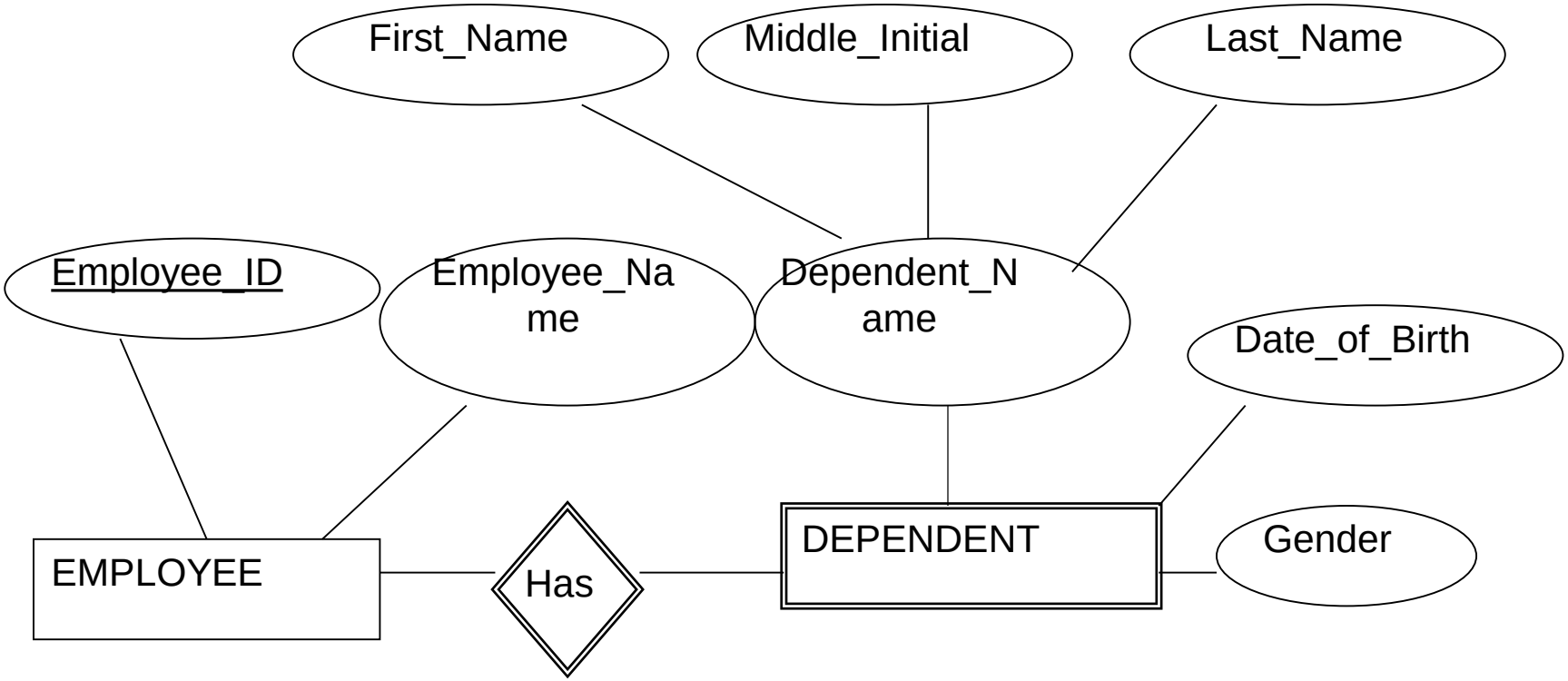
EMPLOYEE_SKILL

Employee_ID	Skill

Regular entity with multivalued attribute

- When the regular entity type contains a multivalued attribute, two new relation are created.
- The first relation contains all of the attributes of the entity type except the mutivalued attribute.
- The second relation contains two attributes that form the primary key of the second relation.
- The first of these attributes is the primary key from the first relation
- The second is the multivalued attribute.

Step 2: Mapping Weak Entities



Mapping Weak Entities

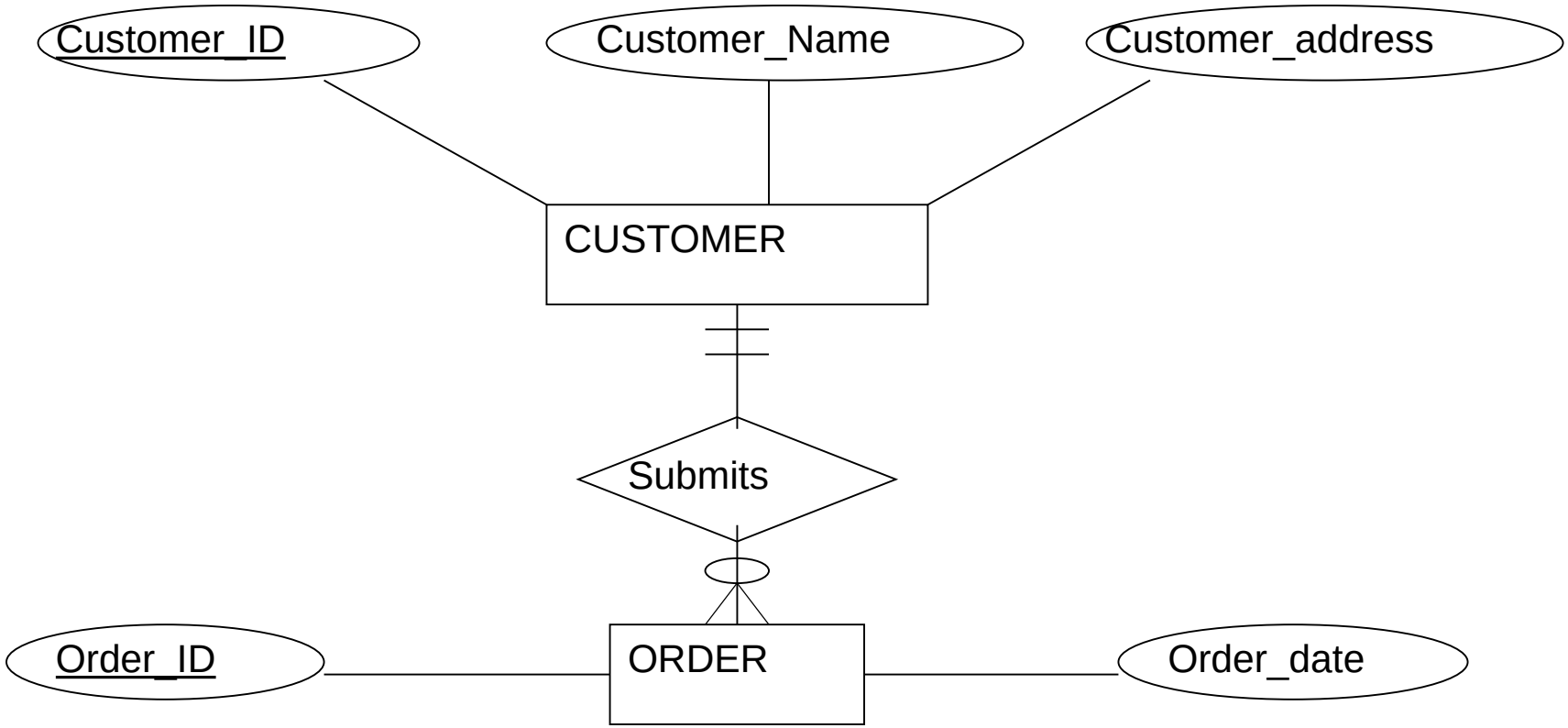
EMPLOYEE

Employee_Id	Employee_Name

DEPENDENT

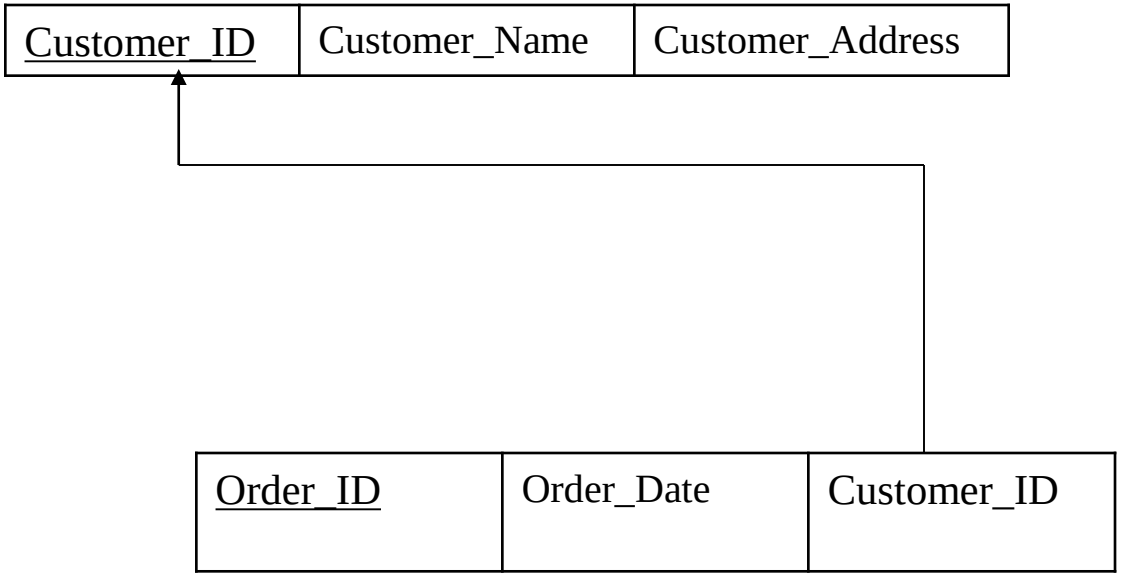
First_name	Middle_initial	Last_Name	Employee_Id	Date_Of_Birth	Gender

STEP3: MAP BINARY RELATIONSHIP

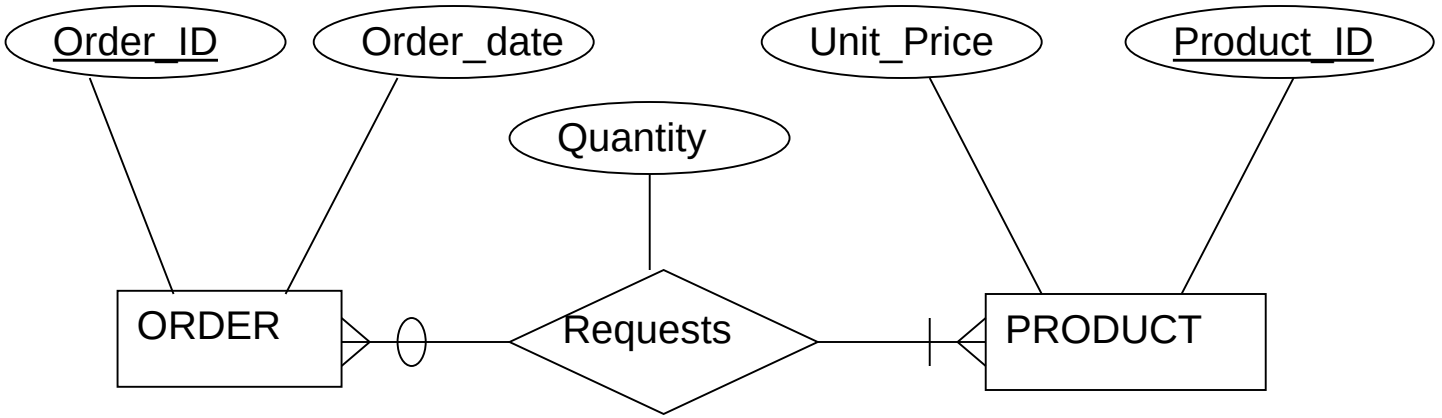


Mapping Binary relationship

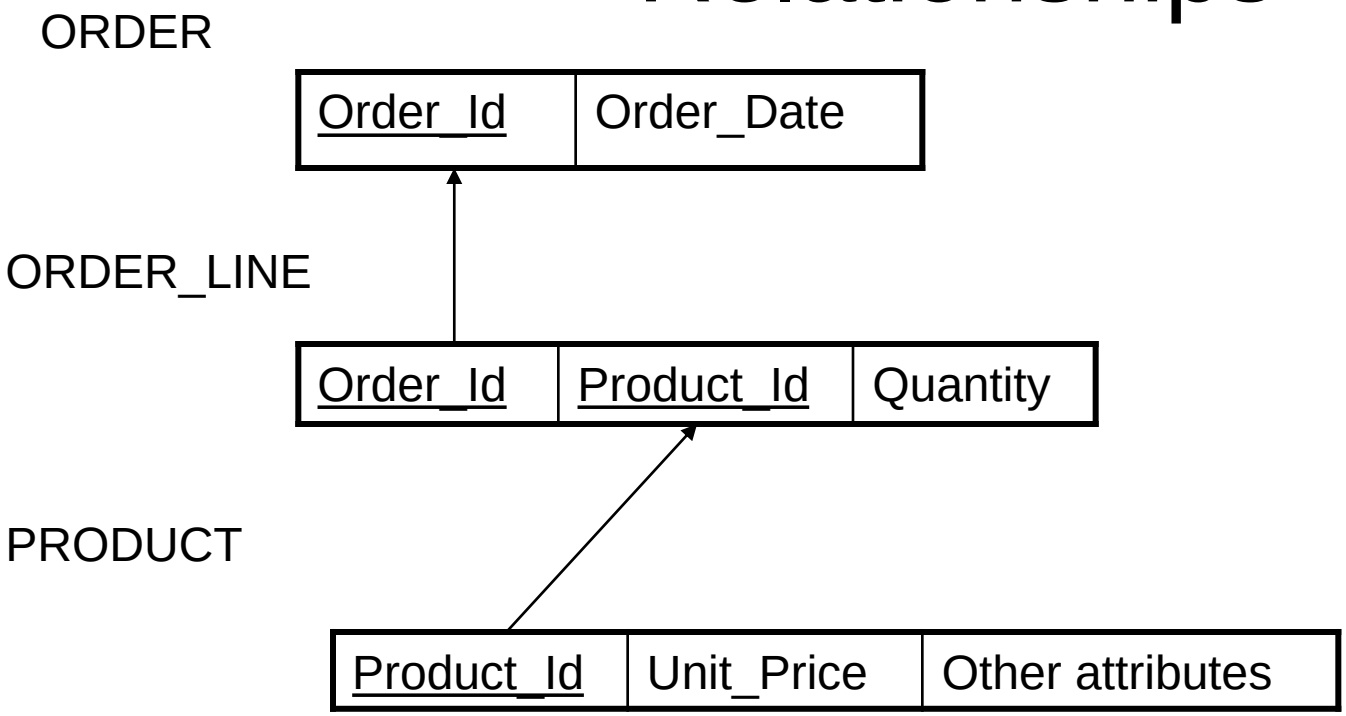
CUSTOMER



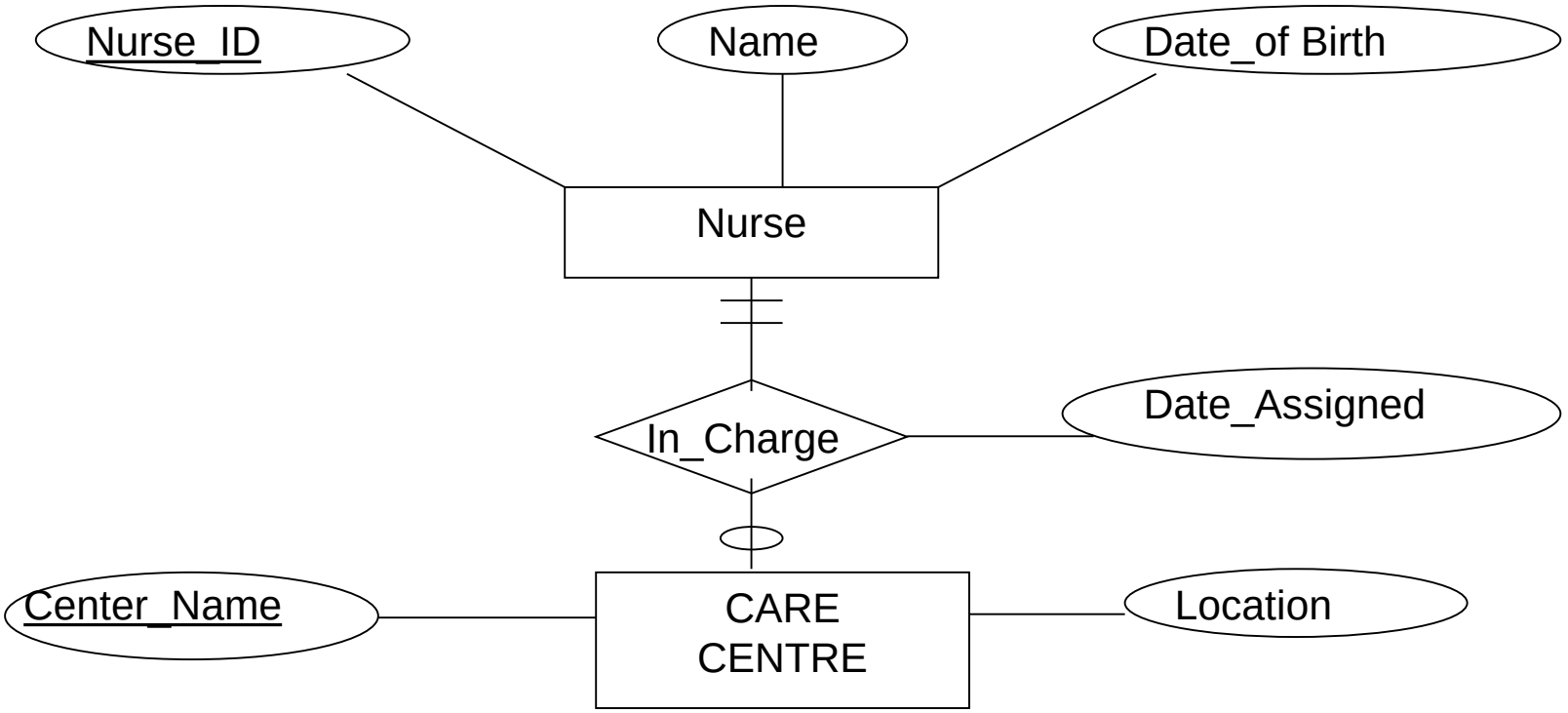
Map Binary Many-To-Many Relationships



Map Binary Many-To-Many Relationships



Map Binary One-To-One Relationships



Map Binary One-To-One Relationships

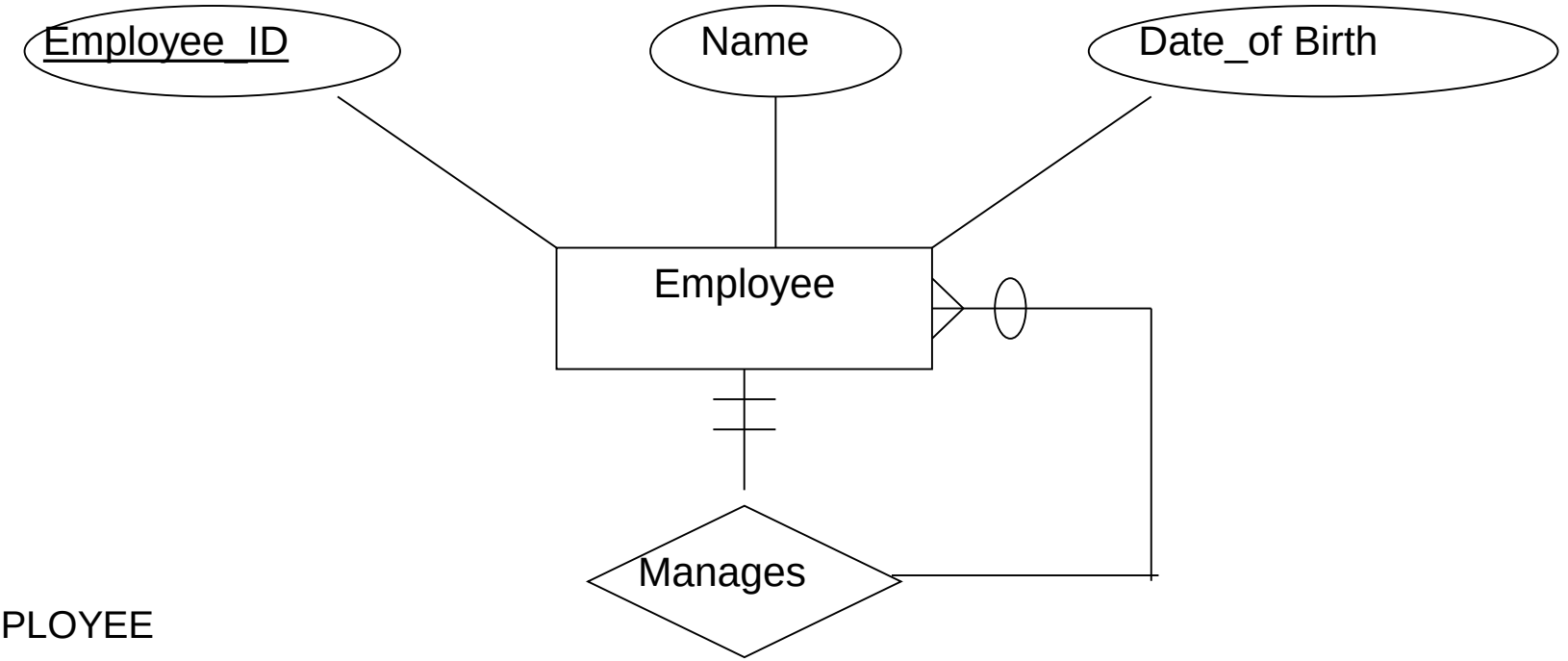
NURSE

Nurse_ID	Name	Date_of_Birth

CARE CENTER

Center_Name	Location	Nurse_in_charge	Date_Assigned

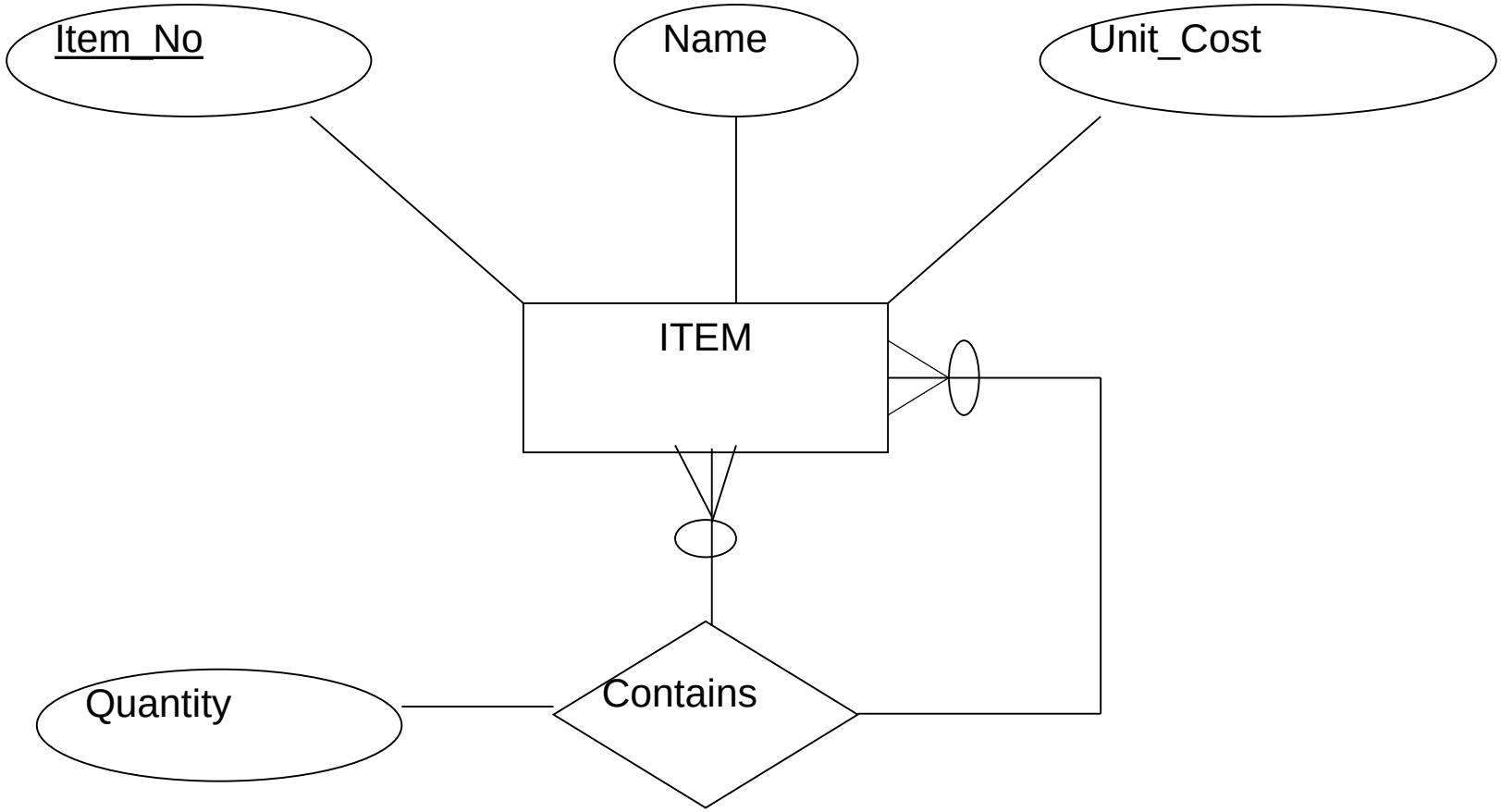
Map Unary One-To-Many Relationships



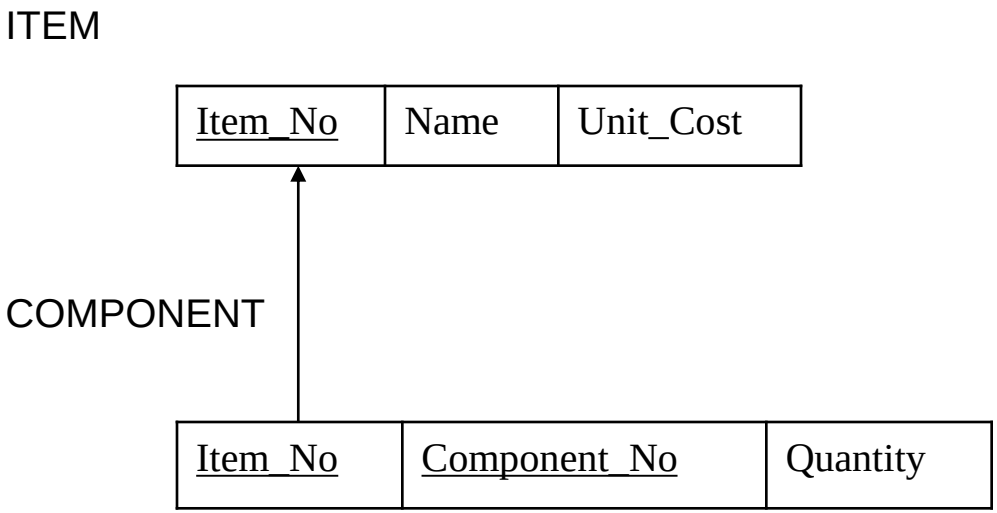
EMPLOYEE

Employee_ID	Name	Date_of_Birth	Manager_ID

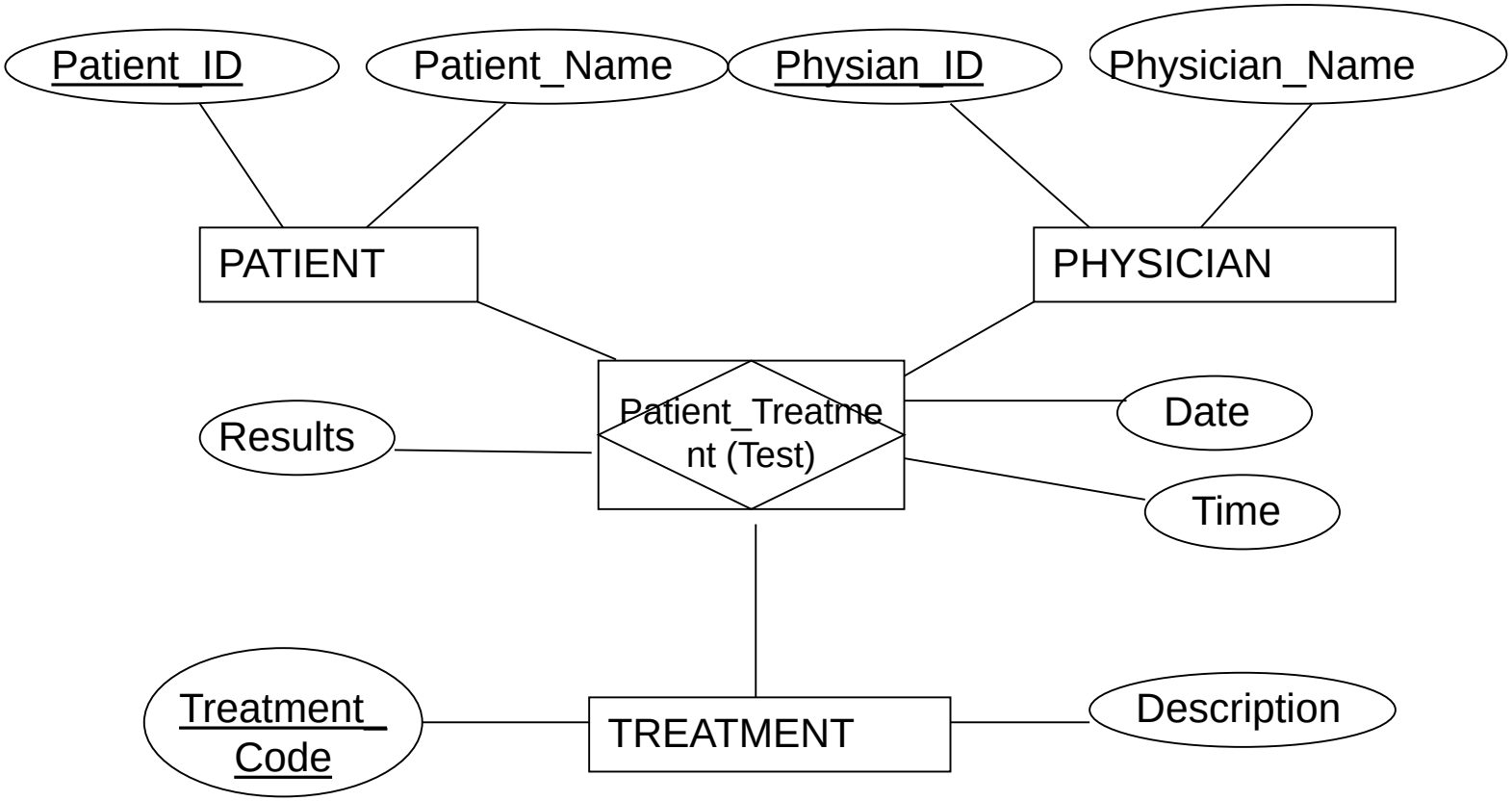
Map Unary Many-To-Many Relationships



Map Unary Many-To-Many Relationships



Map Ternary Relationships



Map Ternary Relationships

PATIENT

Patient_Id	Patient_Name

PHYSICIAN

Physician_ID	Physician_Name

PATIENT TREATMENT

Patient_Id	Physician_ID	Treatment_Code	Date	Time	Result

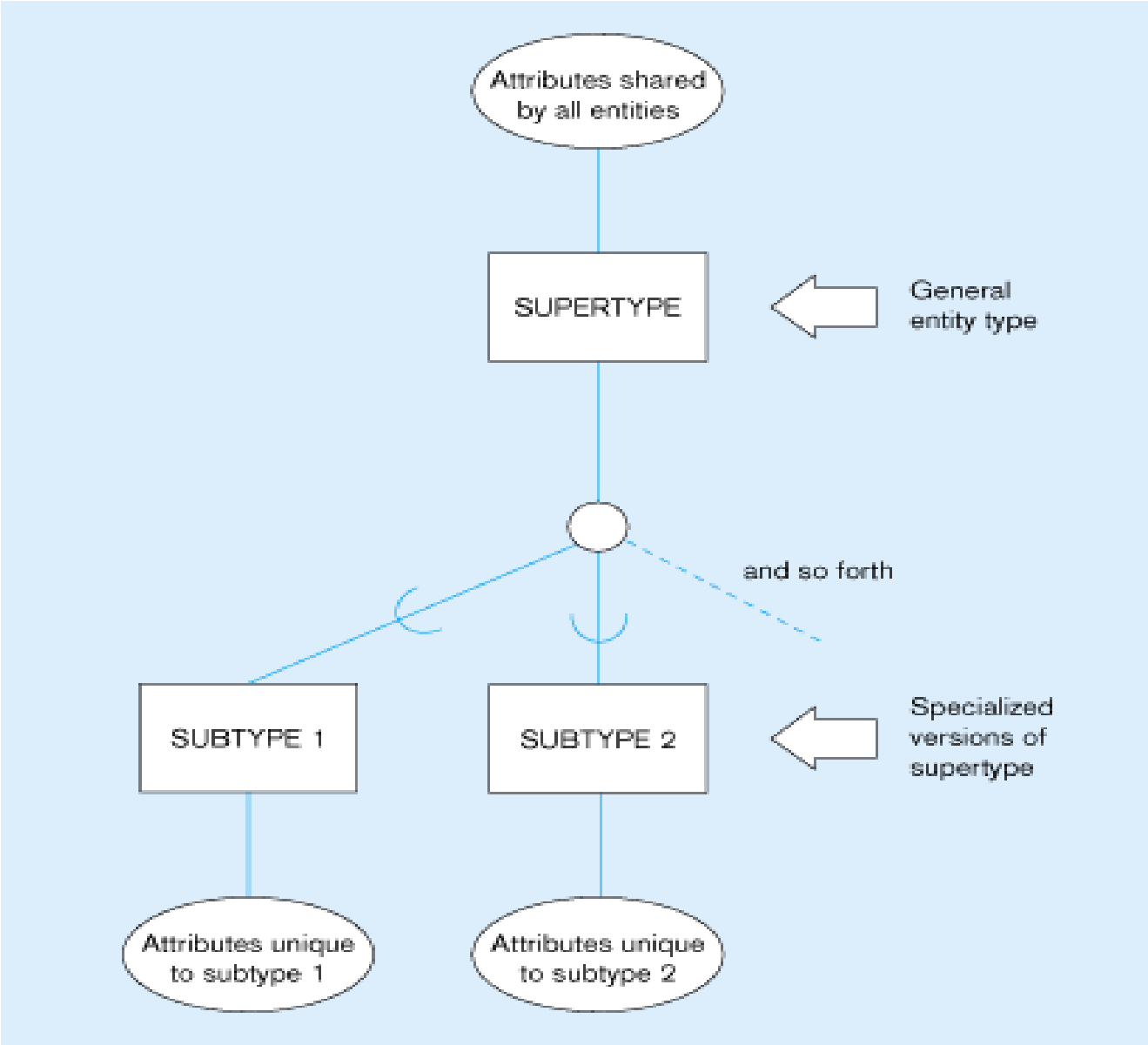
TREATMENT

Treatment_Code	Description

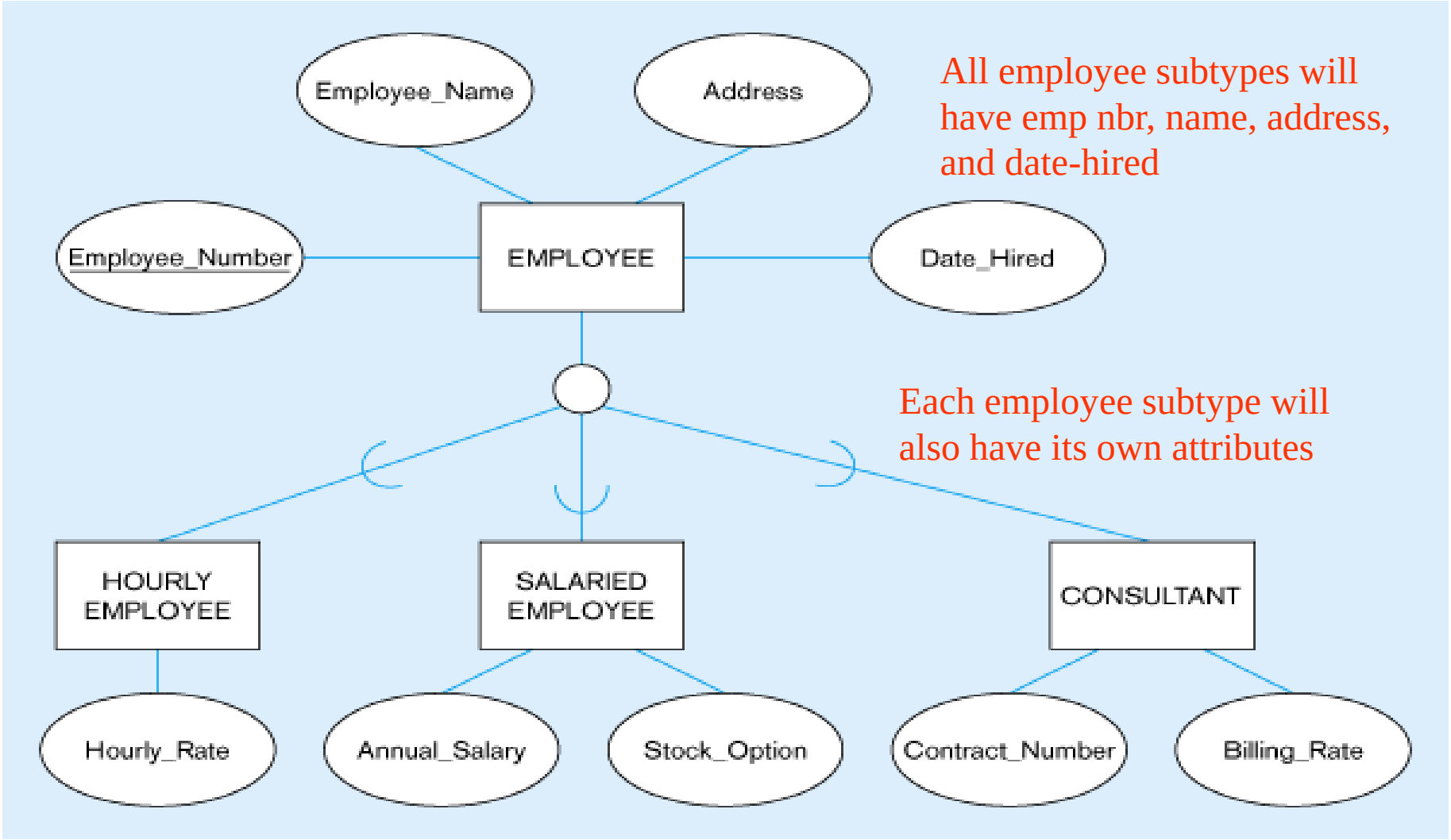
Supertypes and Subtypes

- Subtype: A subgrouping of the entities in an entity type which has attributes that are distinct from those in other subgroupings
- Supertype: An generic entity type that has a relationship with one or more subtypes
- Inheritance:
  - Subtype entities inherit values of all attributes of the supertype
  - An instance of a subtype is also an instance of the supertype

Basic notation for supertype/subtype relationships



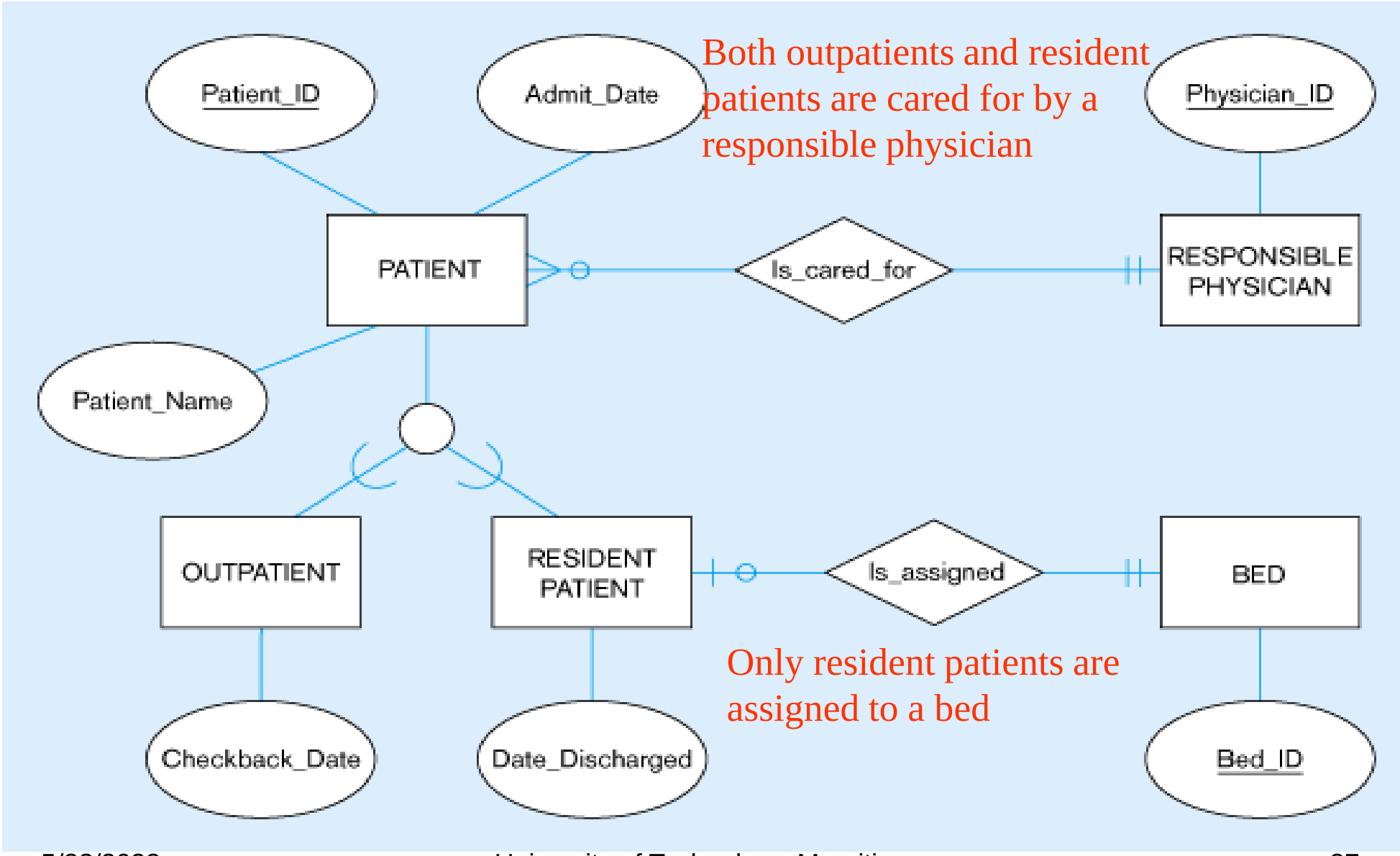
Employee supertype with three subtypes



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

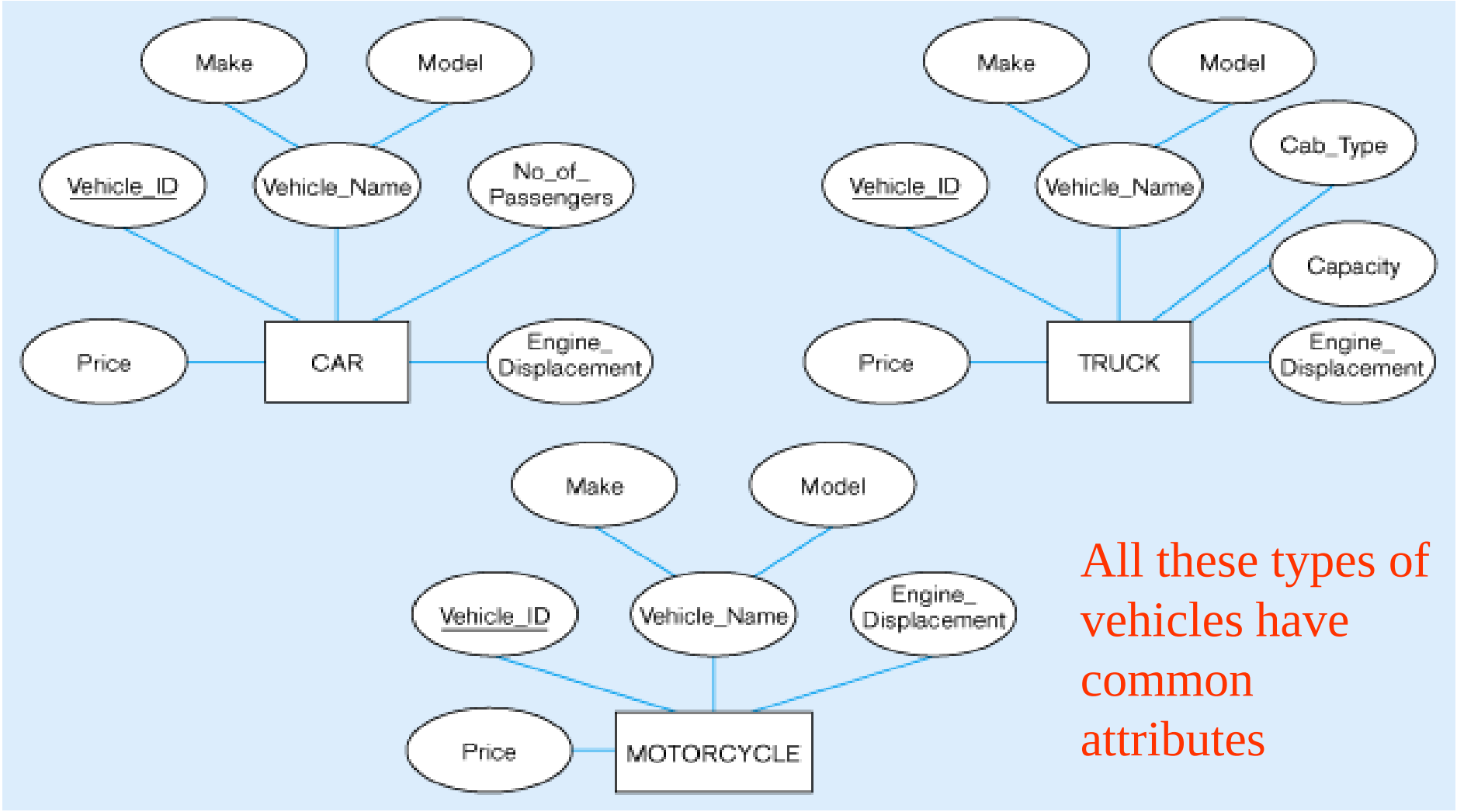
Supertype/subtype relationships in a hospital



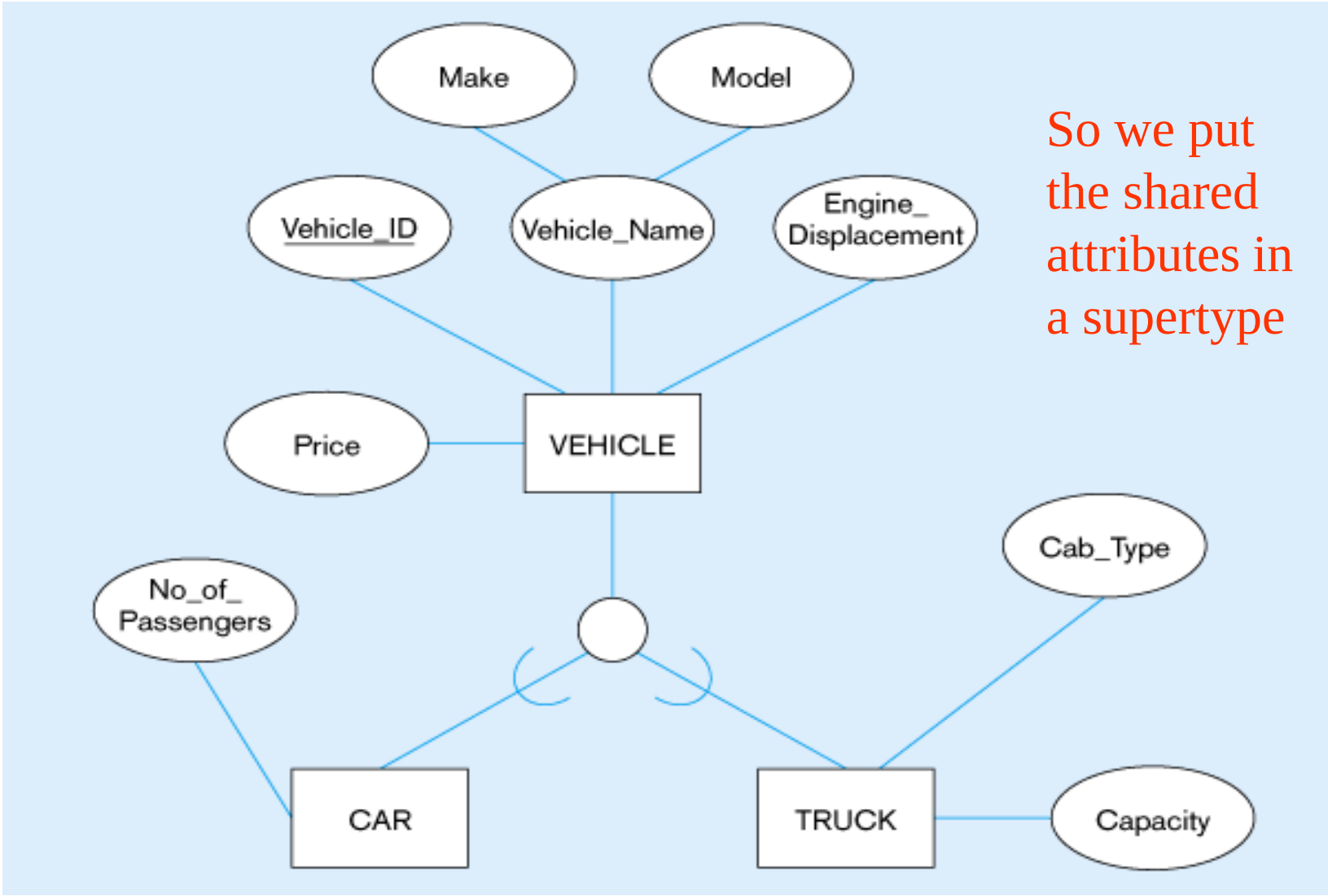
Generalization and Specialization

- ***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
(a) Three entity types: CAR, TRUCK, and MOTORCYCLE



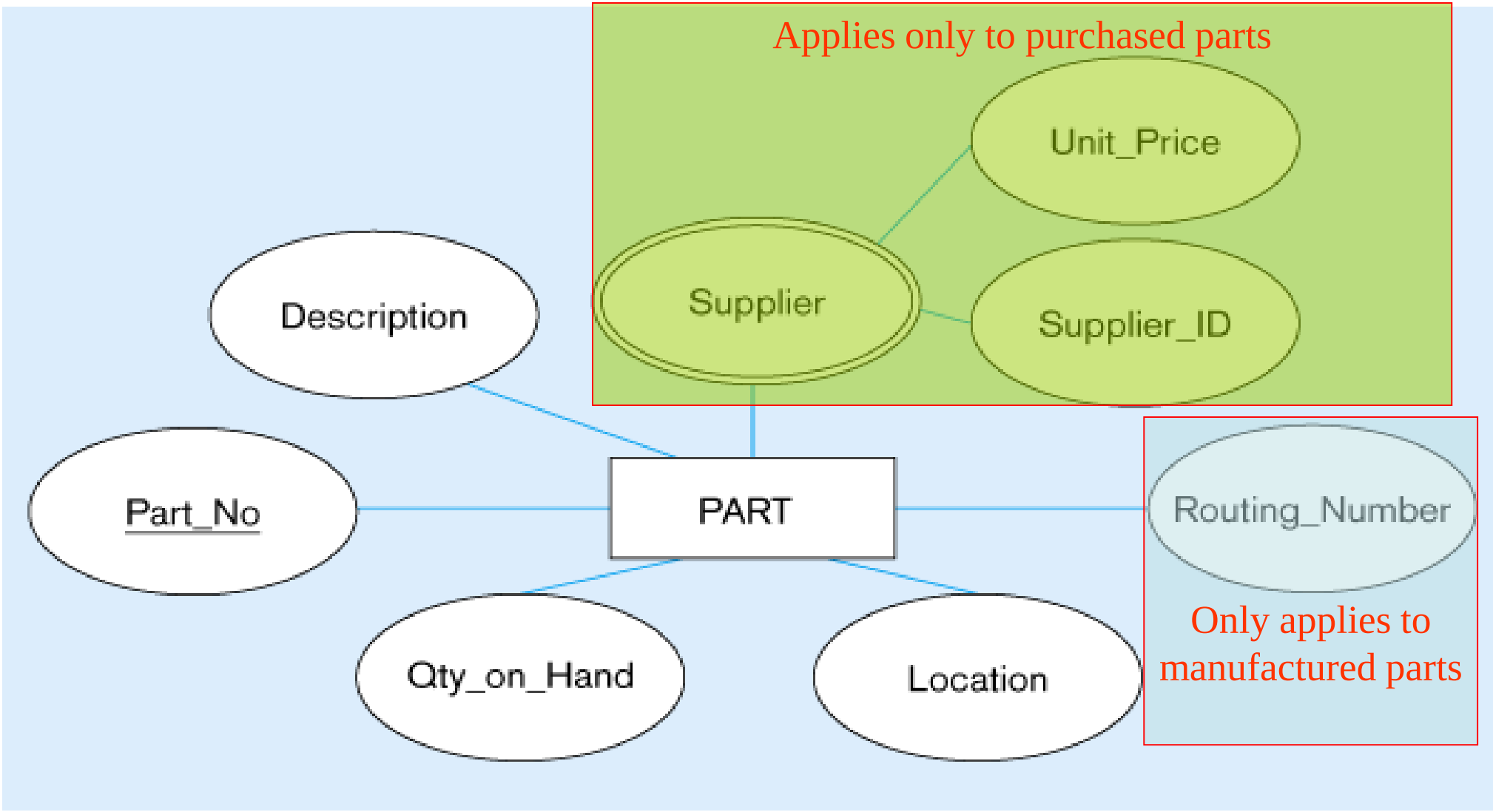
Generalization to VEHICLE supertype



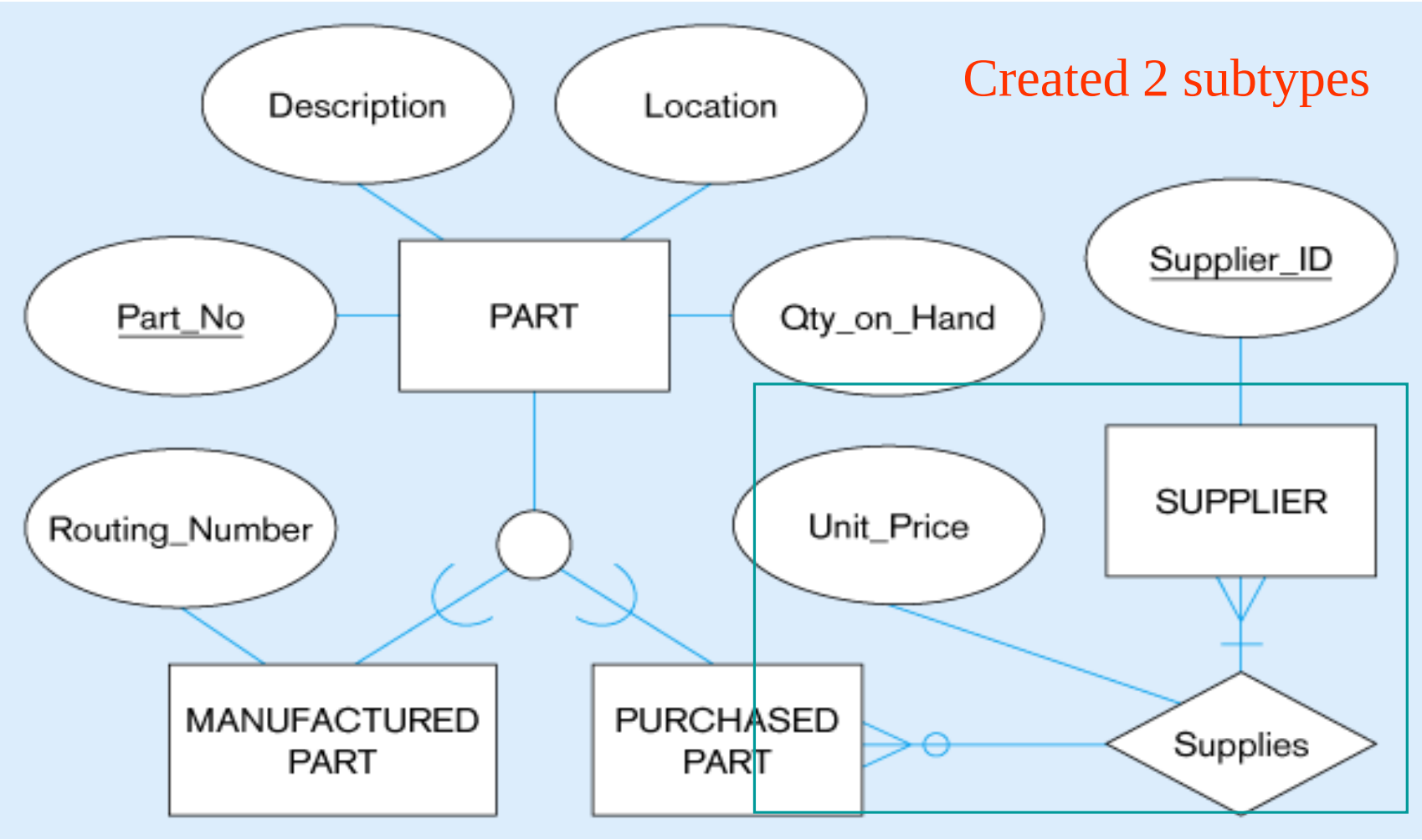
Note: no subtype for motorcycle, since it has no unique attributes

Example of specialization

(a) Entity type PART



Specialization to MANUFACTURED PART and PURCHASED PART

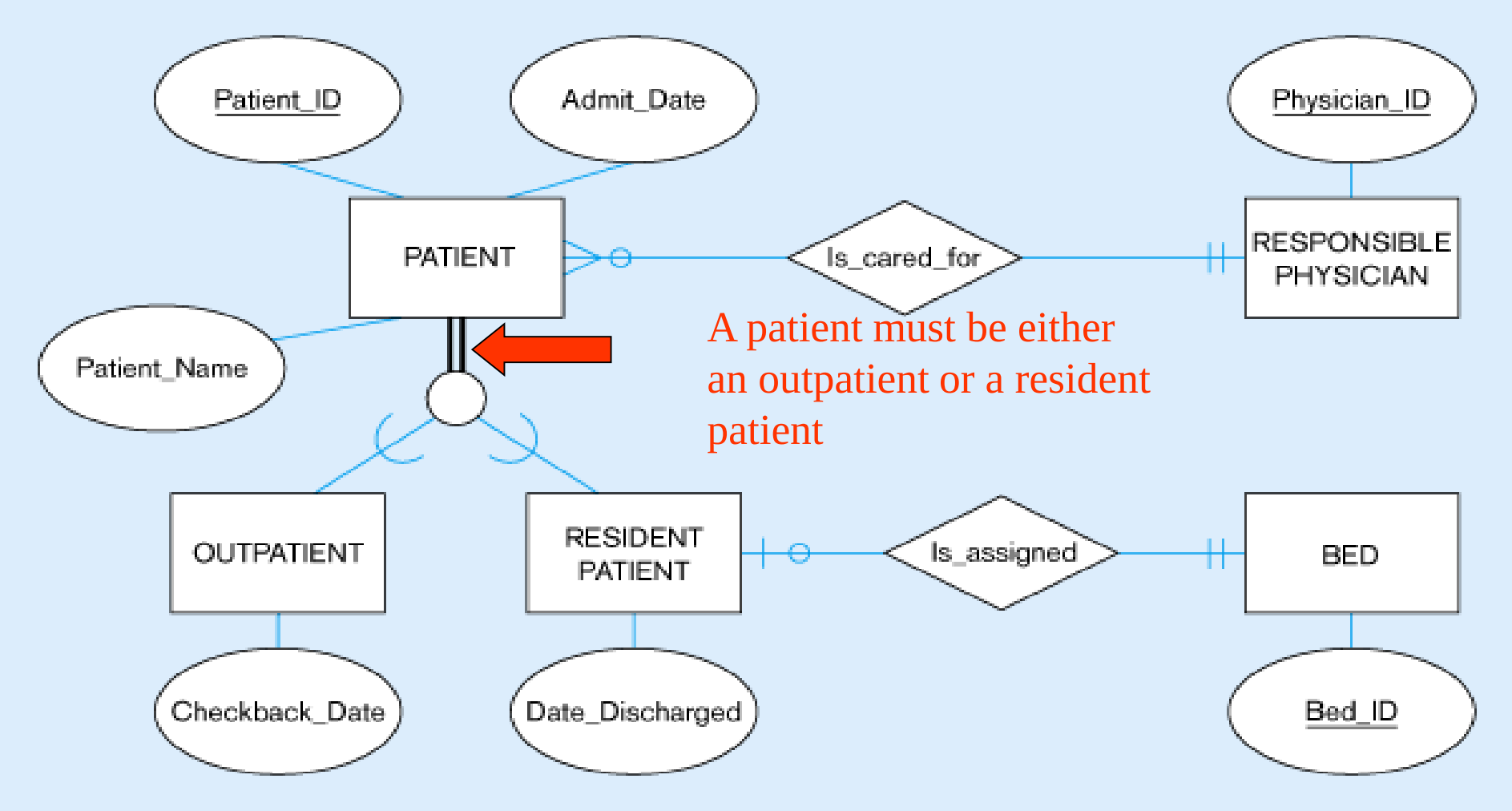


Note: multivalued attribute was replaced by a relationship to another entity

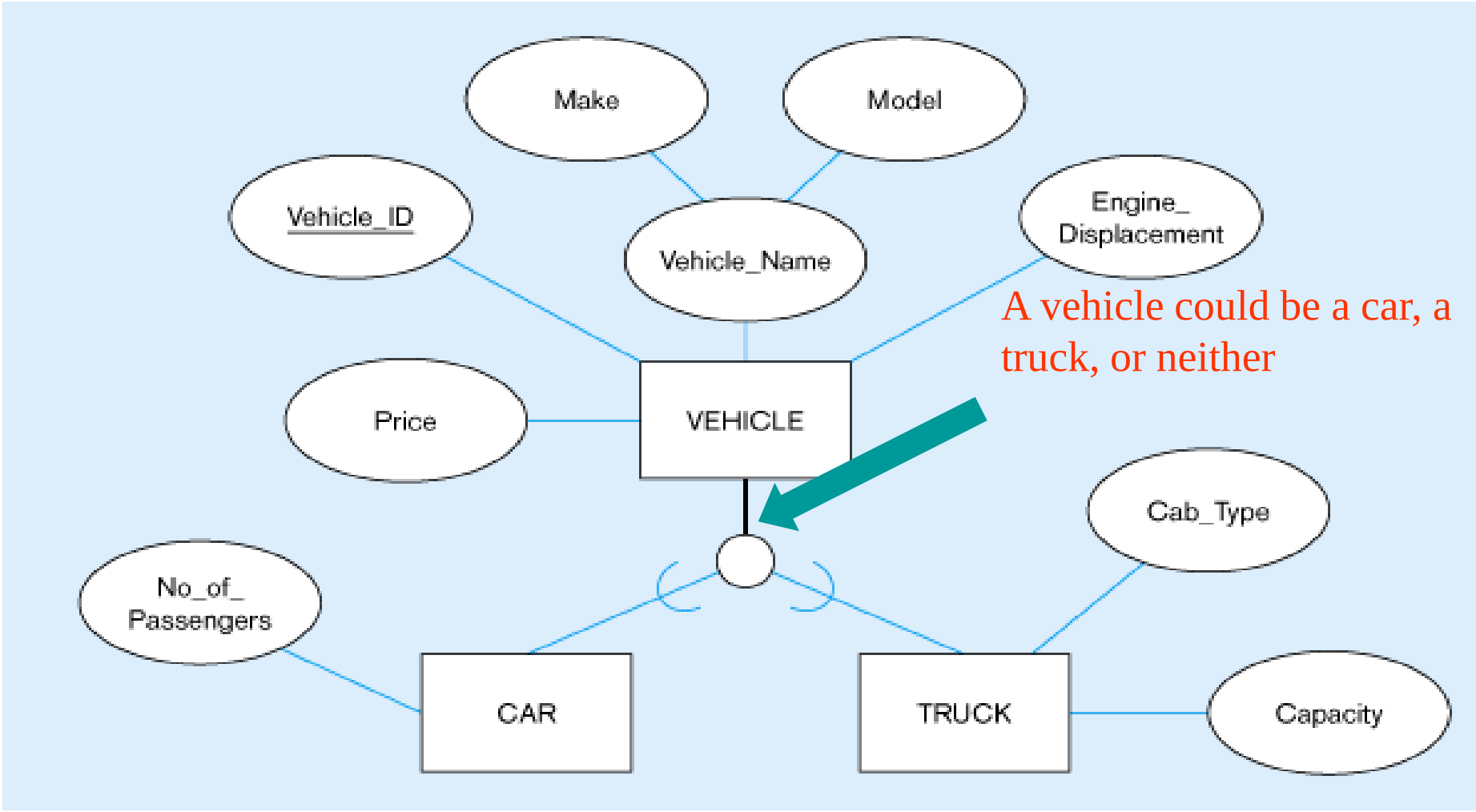
Constraints in Supertype/ Completeness Constraint

- ***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
(a) Total specialization rule



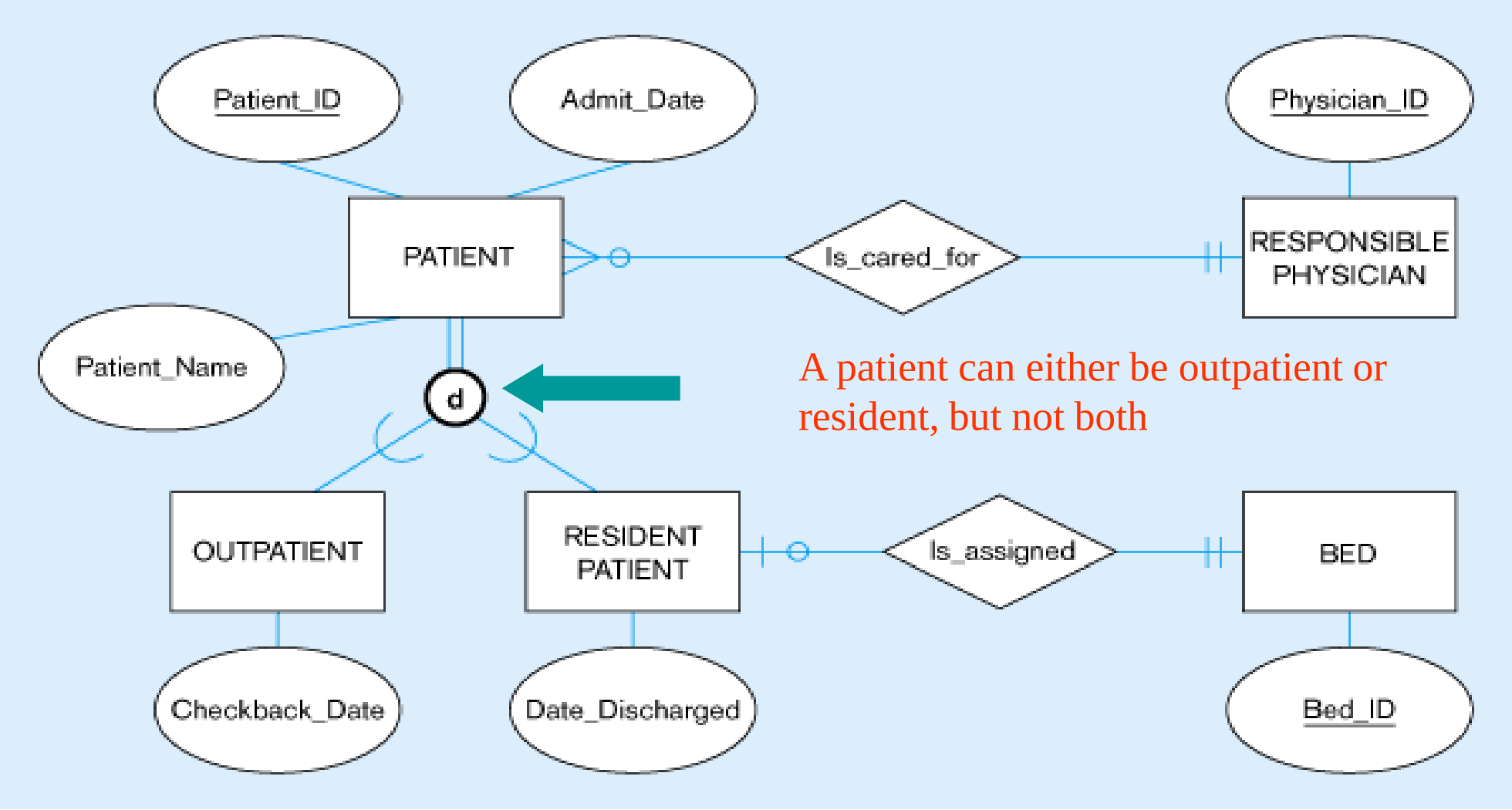
Partial specialization rule



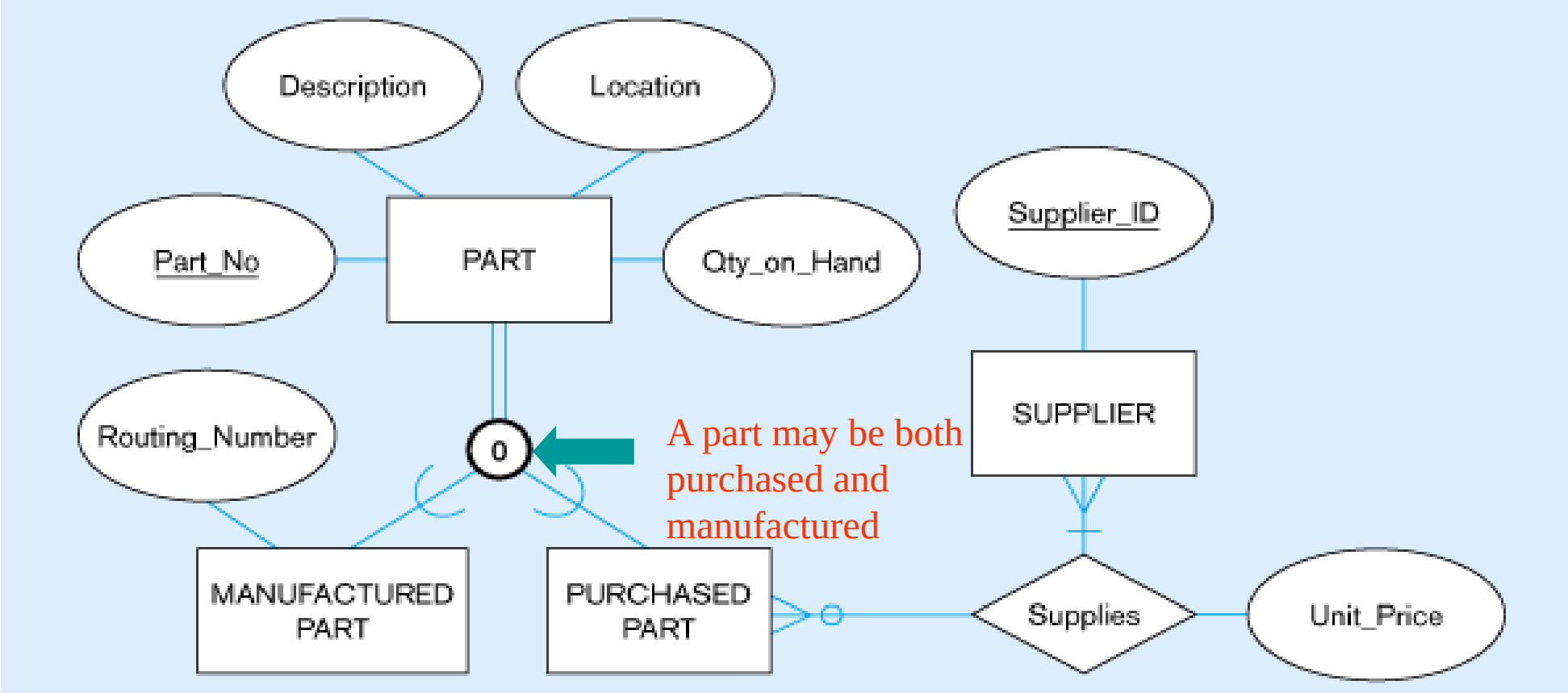
Constraints in Supertype/ Disjointness constraint

- ***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
(a) Disjoint rule



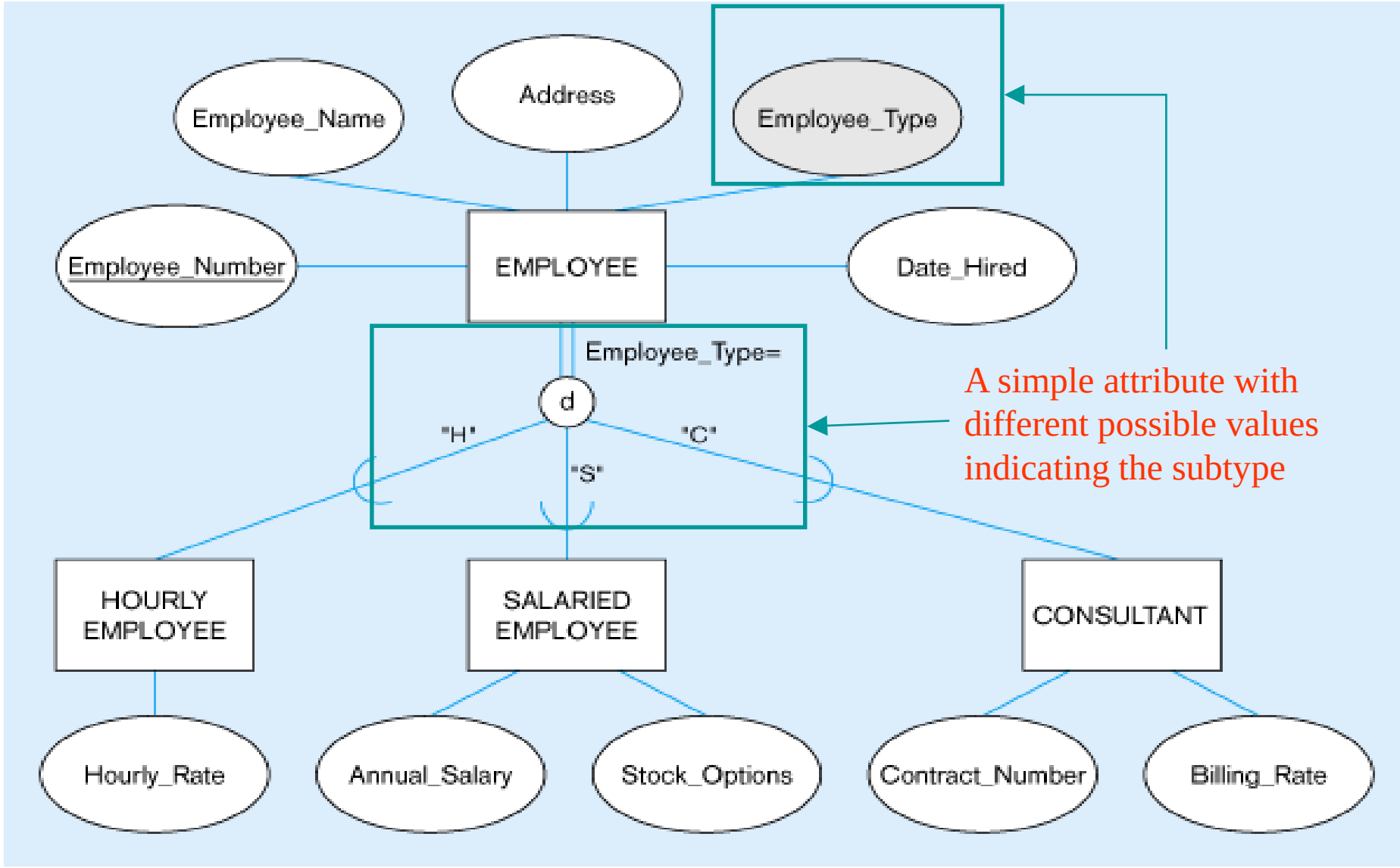
Overlap rule



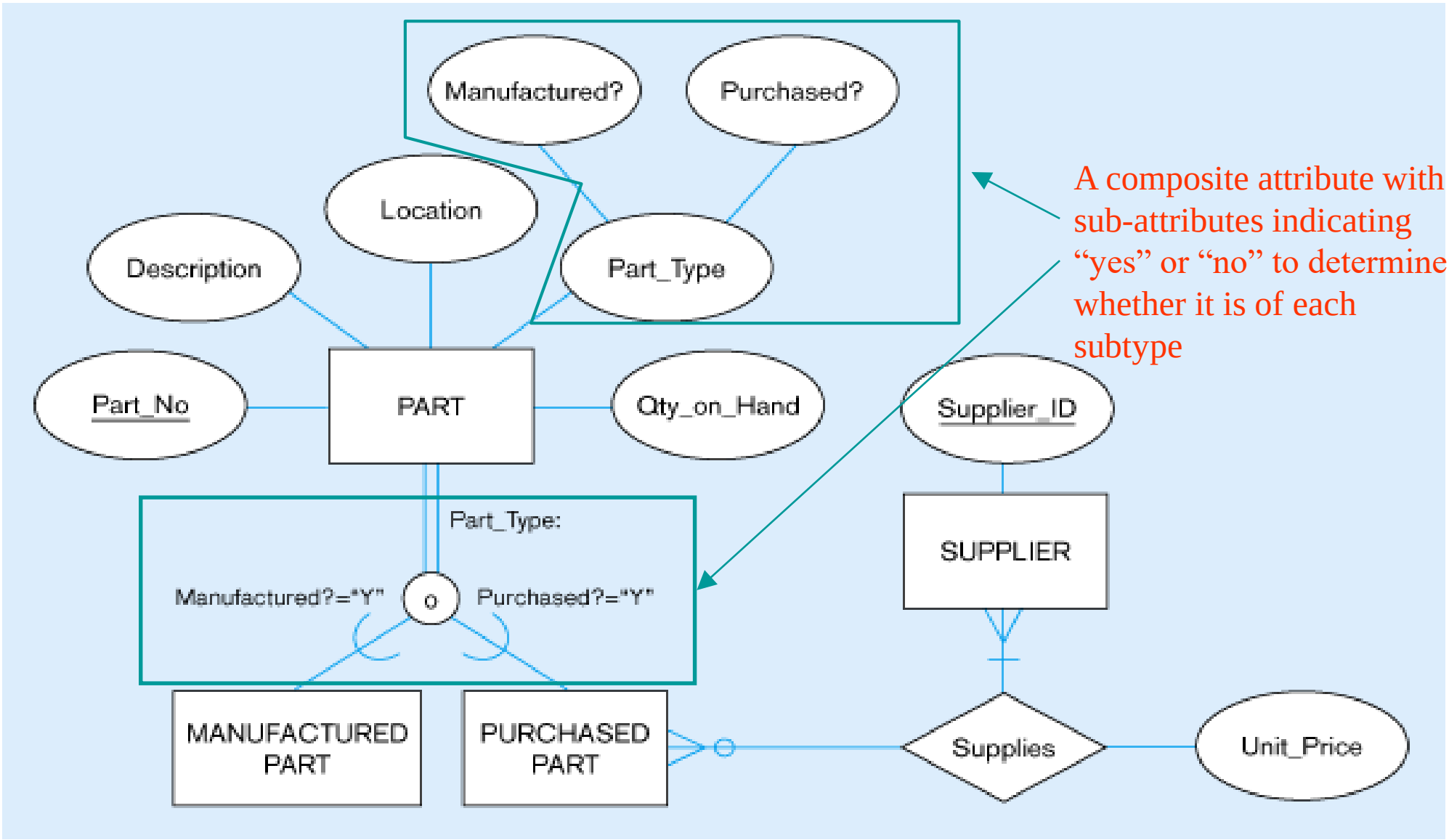
Constraints in Supertype/ Subtype Discriminators

- ***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

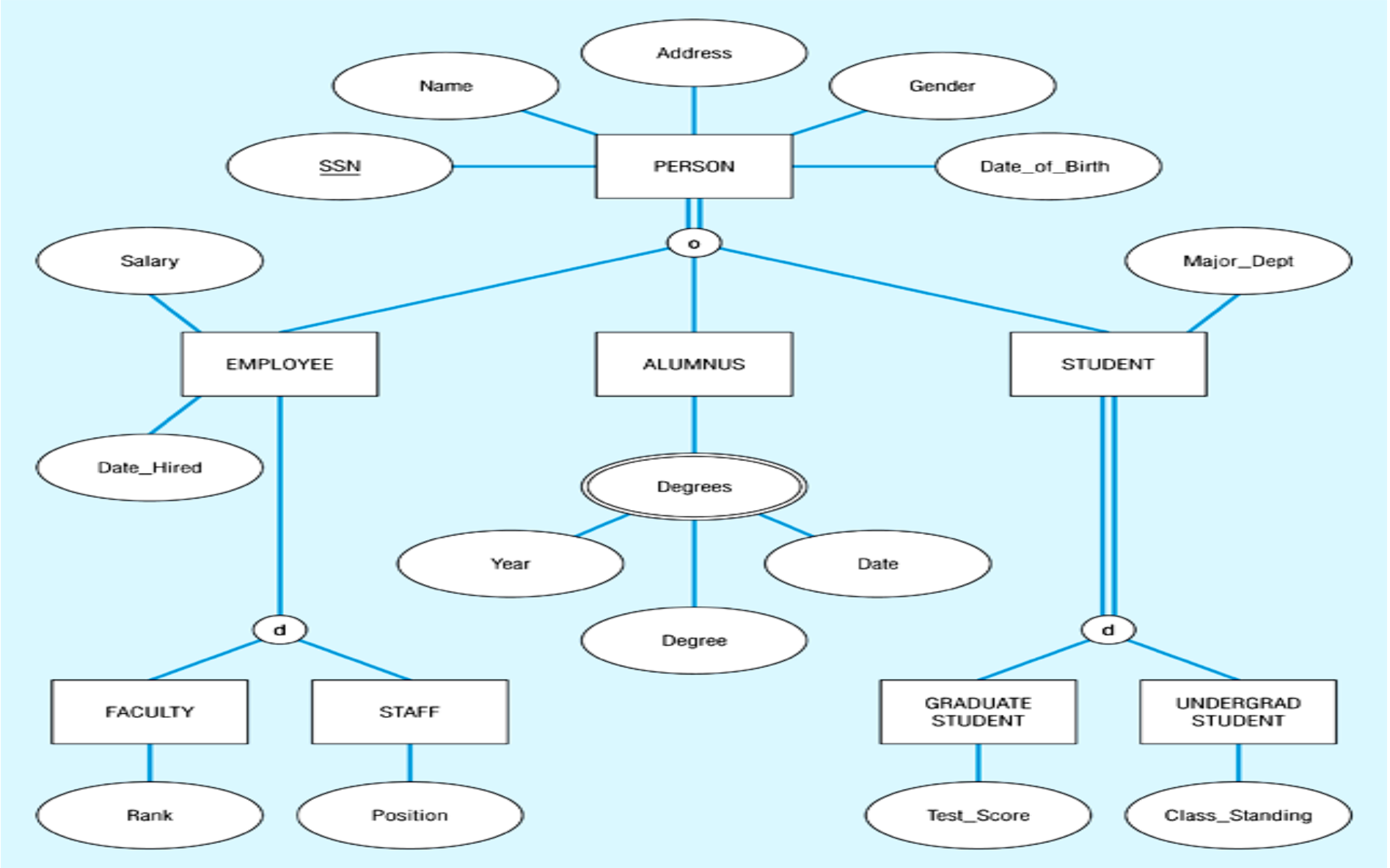
Introducing a subtype discriminator (*disjoint* rule)



Subtype discriminator (*overlap* rule)



Example of supertype/subtype hierarchy



Entity Clusters

- EER diagrams are difficult to read when there are too many entities and relationships
- Solution: group entities and relationships into ***entity clusters***
- **Entity cluster**: set of one or more entity types and associated relationships grouped into a single abstract entity type