

Database Systems and Web (15B11CI312)

Database Systems and Web

Lecture 6: EER

Contents to be covered

- Enhanced ER Modelling

- Subclass

- Superclass

- Generalization

- Specialization

The Enhanced Entity-Relationship (EER) Model

Enhanced ER (EER) model

- Created to design more accurate database schemas
 - Reflect the data properties and constraints more precisely
- More complex requirements than traditional applications

Subclasses, Superclasses, and Inheritance

EER model includes all modeling concepts of the ER model

In addition, EER includes:

- **Subclasses** and **superclasses**
- **Specialization** and **generalization**
- **Category** or **union type**
- **Attribute** and **relationship inheritance**

Subclasses, Superclasses, and Inheritance

Enhanced ER or EER diagrams

- Diagrammatic technique for displaying these concepts in an EER schema

Subtype or subclass of an entity type

- Sub groupings of entities that are meaningful
- Represented explicitly because of their significance to the database application

Subclasses, Superclasses, and Inheritance (cont'd.)

Terms for relationship between a superclass and any one of its subclasses

- **Superclass/subclass**
- **Supertype/subtype**
- **Class/subclass** relationship

Type inheritance

- Subclass entity inherits all attributes and relationships of superclass

Subclasses and Superclasses (1)

An entity type may have additional meaningful subgroupings of its entities

- Example: EMPLOYEE may be further grouped into:
 - SECRETARY, ENGINEER, TECHNICIAN, ...
 - Based on the EMPLOYEE's Job
 - MANAGER
 - EMPLOYEES who are managers
 - SALARIED_EMPLOYEE, HOURLY_EMPLOYEE
 - Based on the EMPLOYEE's method of pay

EER diagrams extend ER diagrams to represent these additional subgroupings, called *subclasses* or *subtypes*

Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Subclasses and Superclasses (2)

Each of these subgroupings is a subset of EMPLOYEE entities

Each is called a subclass of EMPLOYEE

EMPLOYEE is the superclass for each of these subclasses

These are called superclass/subclass relationships:

- EMPLOYEE/SECRETARY
- EMPLOYEE/TECHNICIAN
- EMPLOYEE/MANAGER
- ...

Subclasses and Superclasses (3)

These are also called IS-A relationships

- SECRETARY IS-A EMPLOYEE, TECHNICIAN IS-A EMPLOYEE,

Note: An entity that is member of a subclass represents the same real-world entity as some member of the superclass:

- The subclass member is the same entity in a *distinct specific role*
- An entity cannot exist in the database merely by being a member of a subclass; it must also be a member of the superclass
- A member of the superclass can be optionally included as a member of any number of its subclasses

Subclasses and Superclasses (4)

Examples:

- A salaried employee who is also an engineer belongs to the two subclasses:
 - ENGINEER, and
 - SALARIED_EMPLOYEE
- A salaried employee who is also an engineering manager belongs to the three subclasses:
 - MANAGER,
 - ENGINEER, and
 - SALARIED_EMPLOYEE

It is not necessary that every entity in a superclass be a member of some subclass

Representing Specialization in EER Diagrams

Attribute Inheritance in Superclass / Subclass Relationships

An entity that is member of a subclass *inherits*

- All attributes of the entity as a member of the superclass
- All relationships of the entity as a member of the superclass

Example:

- In the previous slide, SECRETARY (as well as TECHNICIAN and ENGINEER) inherit the attributes Name, SSN, ..., from EMPLOYEE
- Every SECRETARY entity will have values for the inherited attributes

Specialization and Generalization

ER diagrams consider entity types to be primitive objects

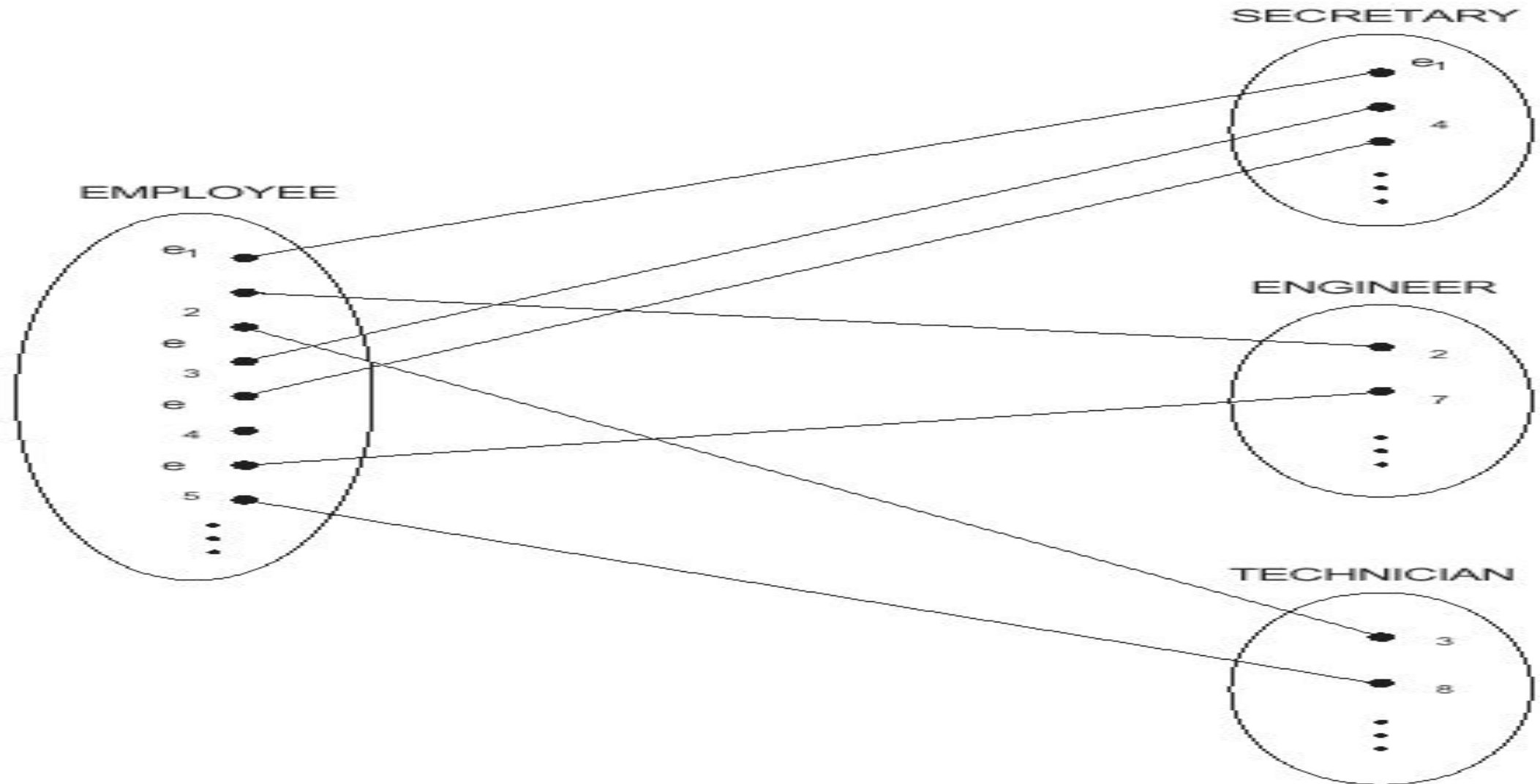
EER diagrams allow refinements within the structures of entity types

Specialization: top-down refinement into **(super)classes** and **subclasses**

Figure 4.1 EER diagram notation for representing specialization and subclasses.

Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Figure 4.2 Some instances of the specialization of EMPLOYEE into the {SECRETARY, ENGINEER, TECHNICIAN} set of subclasses.



Basics

Generalization groups entity types; bottom up synthesis

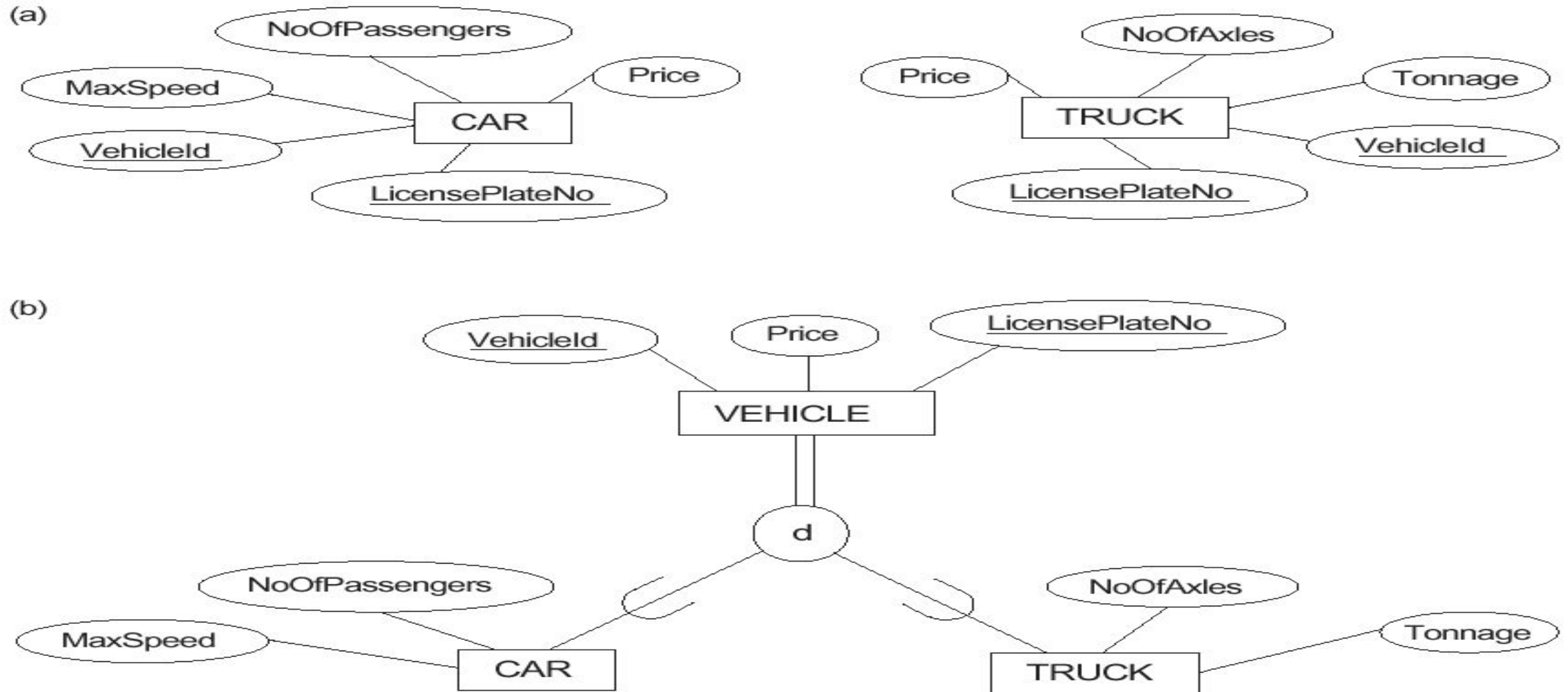
Subclasses **inherit** the attributes and relationships of their super classes

WHY

Relationship types only partially applicable to the superclass

Attributes only partially applying to superclasses

Figure 4.3 Examples of generalization. (a) Two entity types CAR and TRUCK. (b) Generalizing CAR and TRUCK into VEHICLE.



More Specifications: Specialization (1)

Specialization is the process of defining a set of subclasses of a superclass

The set of subclasses is based upon some distinguishing characteristics of the entities in the superclass

- Example: {SECRETARY, ENGINEER, TECHNICIAN} is a specialization of EMPLOYEE based upon *job type*.
- May have several specializations of the same superclass

More Specifications: Specialization (2)

Example: Another specialization of EMPLOYEE based on *method of pay* is {SALARIED_EMPLOYEE, HOURLY_EMPLOYEE}.

- Superclass/subclass relationships and specialization can be diagrammatically represented in EER diagrams
- Attributes of a subclass are called *specific* or *local* attributes.
 - For example, the attribute TypingSpeed of SECRETARY
- The subclass can also participate in specific relationship types.
 - For example, a relationship BELONGS_TO of HOURLY_EMPLOYEE

Review: Specialization and Generalization

Specialization

- Process of defining a set of subclasses of an entity type
- Defined on the basis of some distinguishing characteristic of the entities in the superclass

Subclass can define:

- **Specific attributes**
- **Specific relationship types**

Specialization and Generalization (cont'd.)

Certain attributes may apply to some but not all entities of the superclass

Some relationship types may be participated in only by members of the subclass

Generalization

Reverse process of abstraction

Generalize into a single **superclass**

- Original entity types are special subclasses

Generalization

- Process of defining a generalized entity type from the given entity types

Constraints and Characteristics of Specialization and Generalization Hierarchies

- Constraints that apply to a single specialization or a single generalization
- Differences between specialization/generalization lattices and hierarchies
- If we can determine exactly those entities that will become members of each subclass by a condition, the subclasses are called predicate-defined (or condition-defined) subclasses
 - Condition is a constraint that determines subclass members
 - Display a predicate-defined subclass by writing the predicate condition next to the line attaching the subclass to its superclass

Constraints on Specialization and Generalization (contd..)

If all subclasses in a specialization have membership condition on same attribute of the superclass, specialization is called an attribute-defined specialization

- Attribute is called the defining attribute of the specialization
- Example: JobType is the defining attribute of the specialization {SECRETARY, TECHNICIAN, ENGINEER} of EMPLOYEE

If no condition determines membership, the subclass is called user-defined

- Membership in a subclass is determined by the database users by applying an operation to add an entity to the subclass
- Membership in the subclass is specified individually for each entity in the superclass by the user

Displaying an attribute-defined specialization in EER diagrams

Constraints on Specialization and Generalization (3)

Two basic constraints can apply to a specialization/generalization:

- Disjointness Constraint:
- Completeness Constraint:

Constraints on Specialization and Generalization (4)

Disjointness Constraint:

- Specifies that the subclasses of the specialization must be *disjoint*:
 - an entity can be a member of at most one of the subclasses of the specialization
- Specified by *d* in EER diagram
- If not disjoint, specialization is *overlapping*:
 - that is the same entity may be a member of more than one subclass of the specialization
- Specified by *o* in EER diagram

Constraints on Specialization and Generalization (5)

Completeness Constraint:

- *Total* specifies that every entity in the superclass must be a member of some subclass in the specialization/generalization
- Shown in EER diagrams by a **double line**
- *Partial* allows an entity not to belong to any of the subclasses
- Shown in EER diagrams by a single line

Constraints on Specialization and Generalization (6)

Hence, we have four types of specialization/generalization:

- Disjoint, total
- Disjoint, partial
- Overlapping, total
- Overlapping, partial

Note: Generalization usually is total because the superclass is derived from the subclasses.

Disjointness Constraints

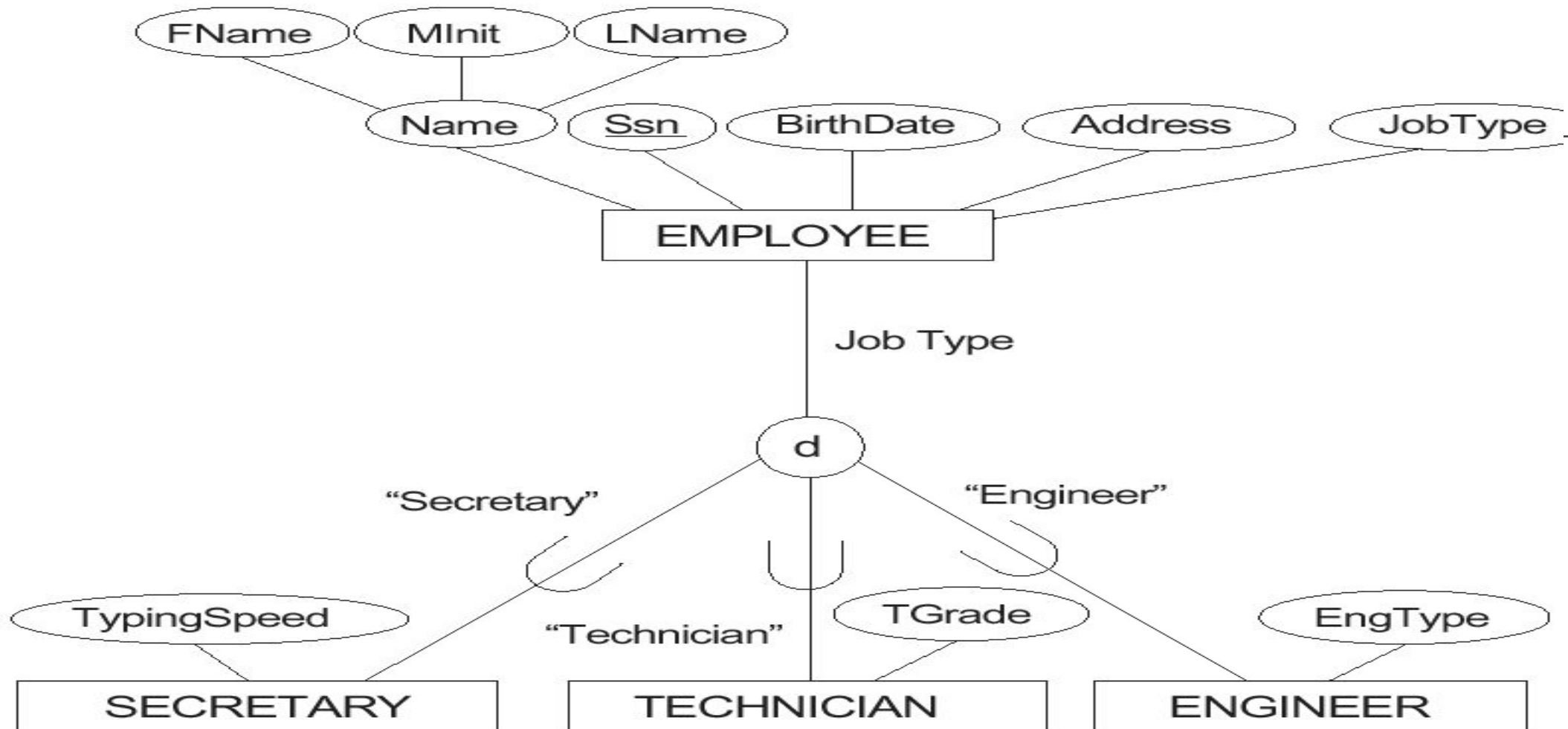
Disjoint (d)

- The subclasses must have disjoint sets of entities

Overlap (o)

- The subclasses may have overlapping sets of entities

Figure 4.4 An attribute-defined specialization on the JobType attribute of EMPLOYEE.



Source: *Fundamentals of database systems* / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Completeness Constraints

Partial

- An entity may not belong to any of the subclasses (single-line)

Total

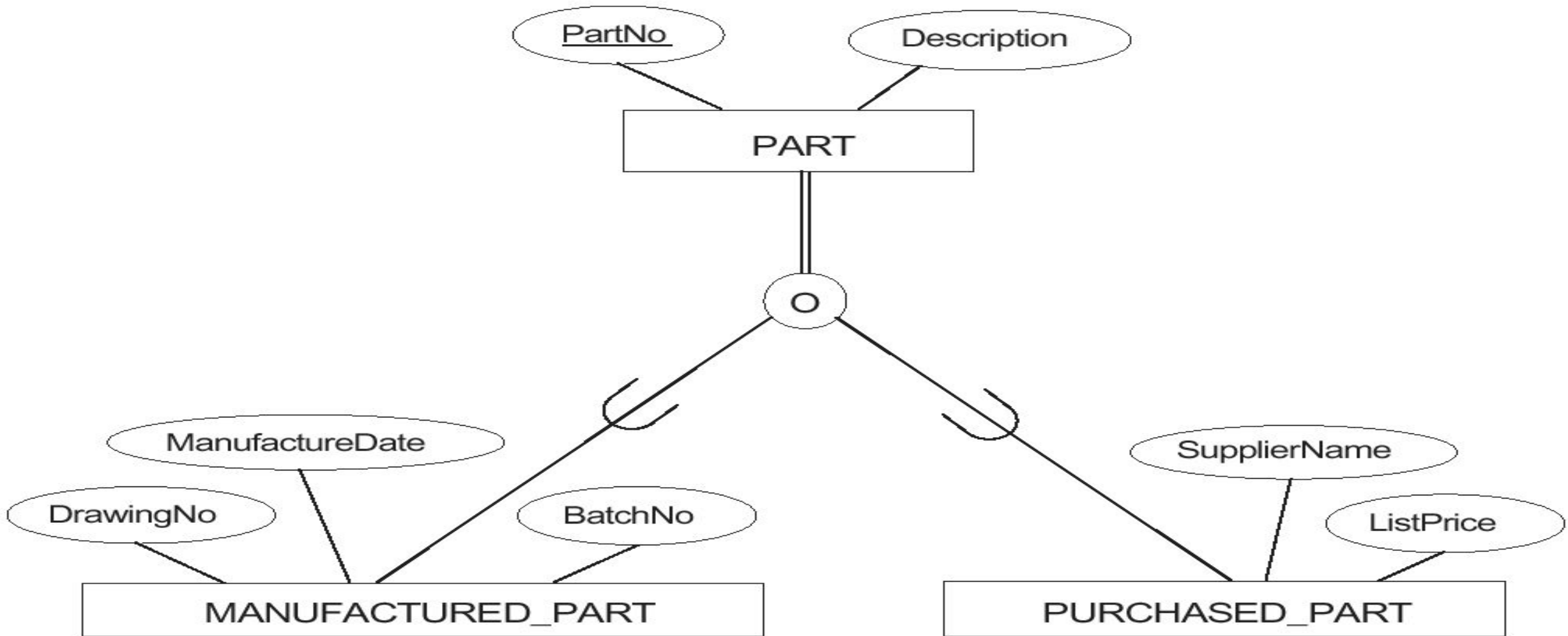
- Every entity in the superclass must be a member of some subclass (double-edge)

Membership Constraints

Predicate defined subclasses

- The subclass is defined through a predicate on the attributes of the superclass
- **Attribute defined subclasses**
- The subclasses in the specialization are all defined by the same attribute of the superclass
- **User defined subclasses**
- Membership in the subclasses is determined at the insertion operation level

Figure 4.5 Notation for specialization with overlapping (nondisjoint) subclasses.



Source: *Fundamentals of database systems* / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Structures in Specialization

Multiple Specializations

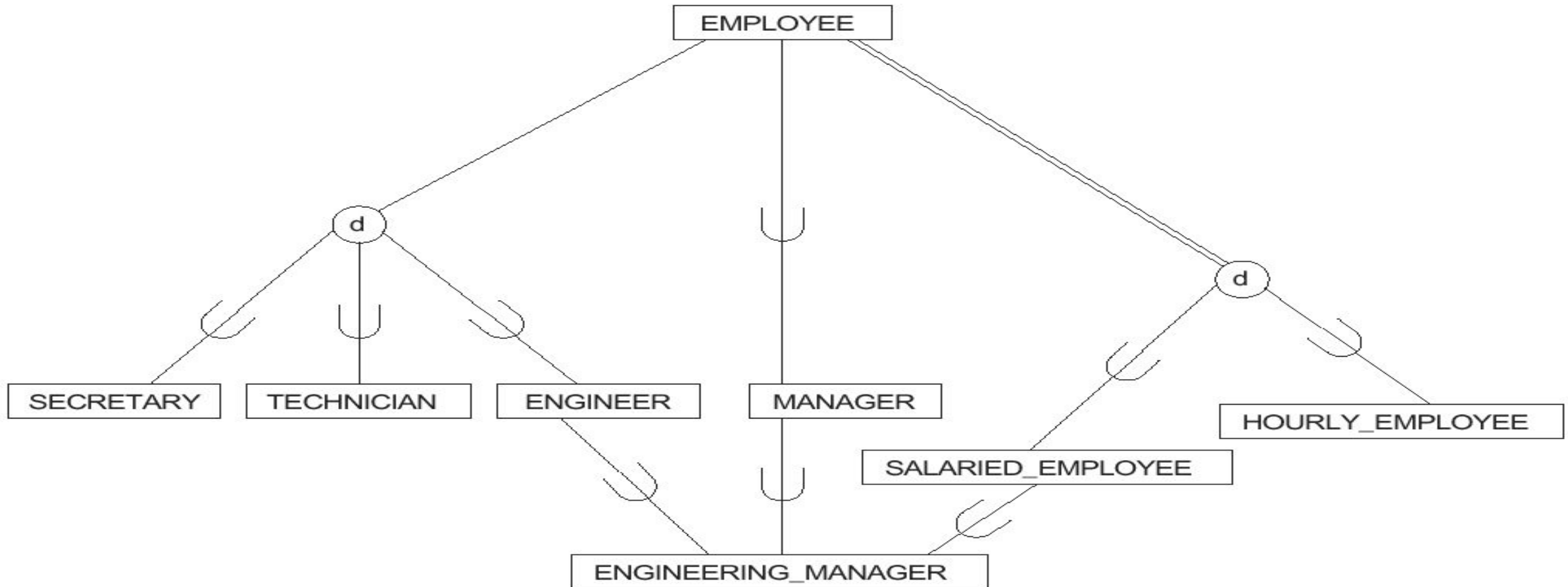
Specialization Hierarchy

- Each subclass belongs to at most one class

Lattice Specializations

- A subclass may belong to more than one class

Figure 4.6 A specialization lattice with the shared subclass ENGINEERING_MANAGER.

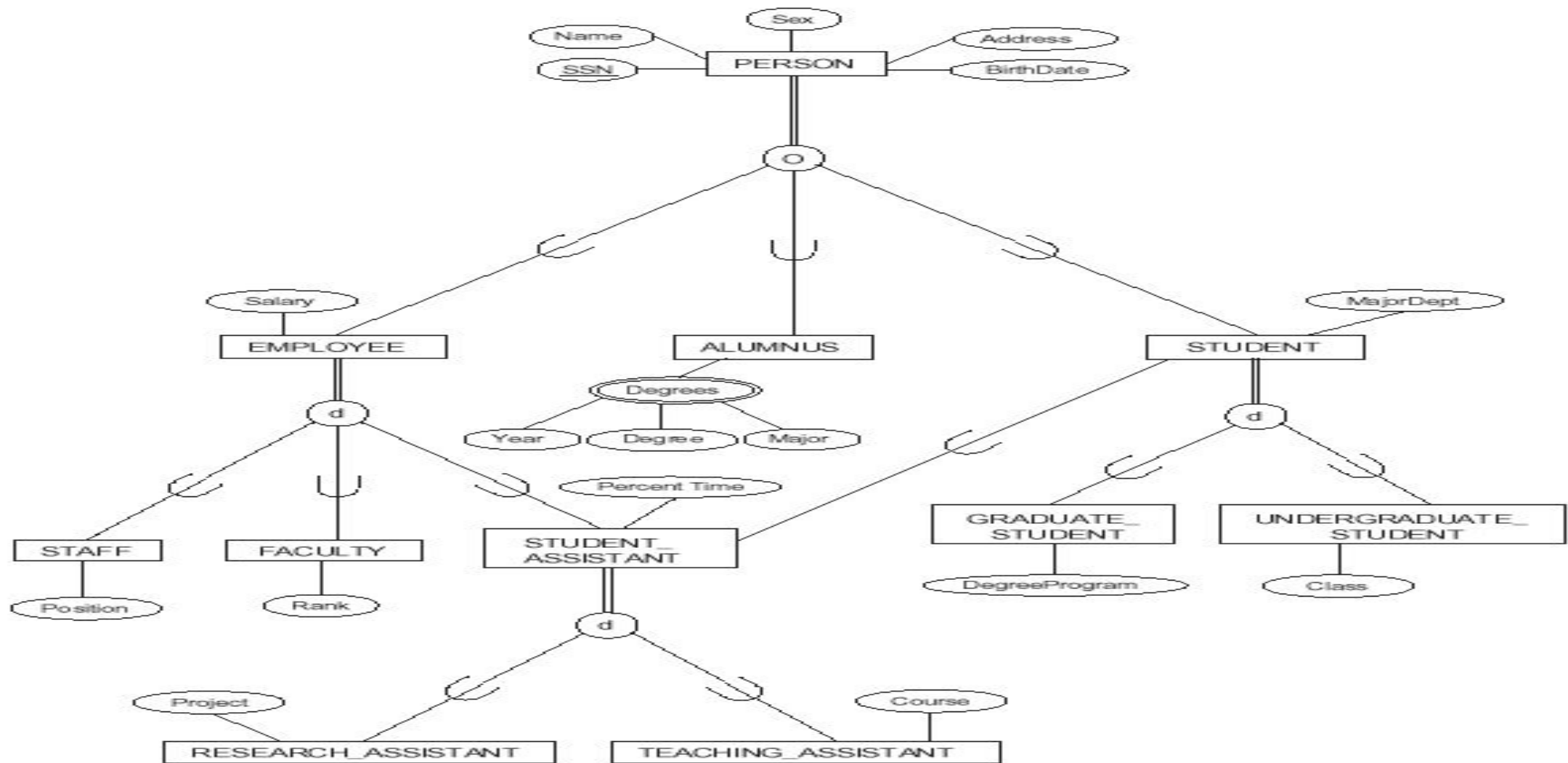


Source: *Fundamentals of database systems* / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Categories

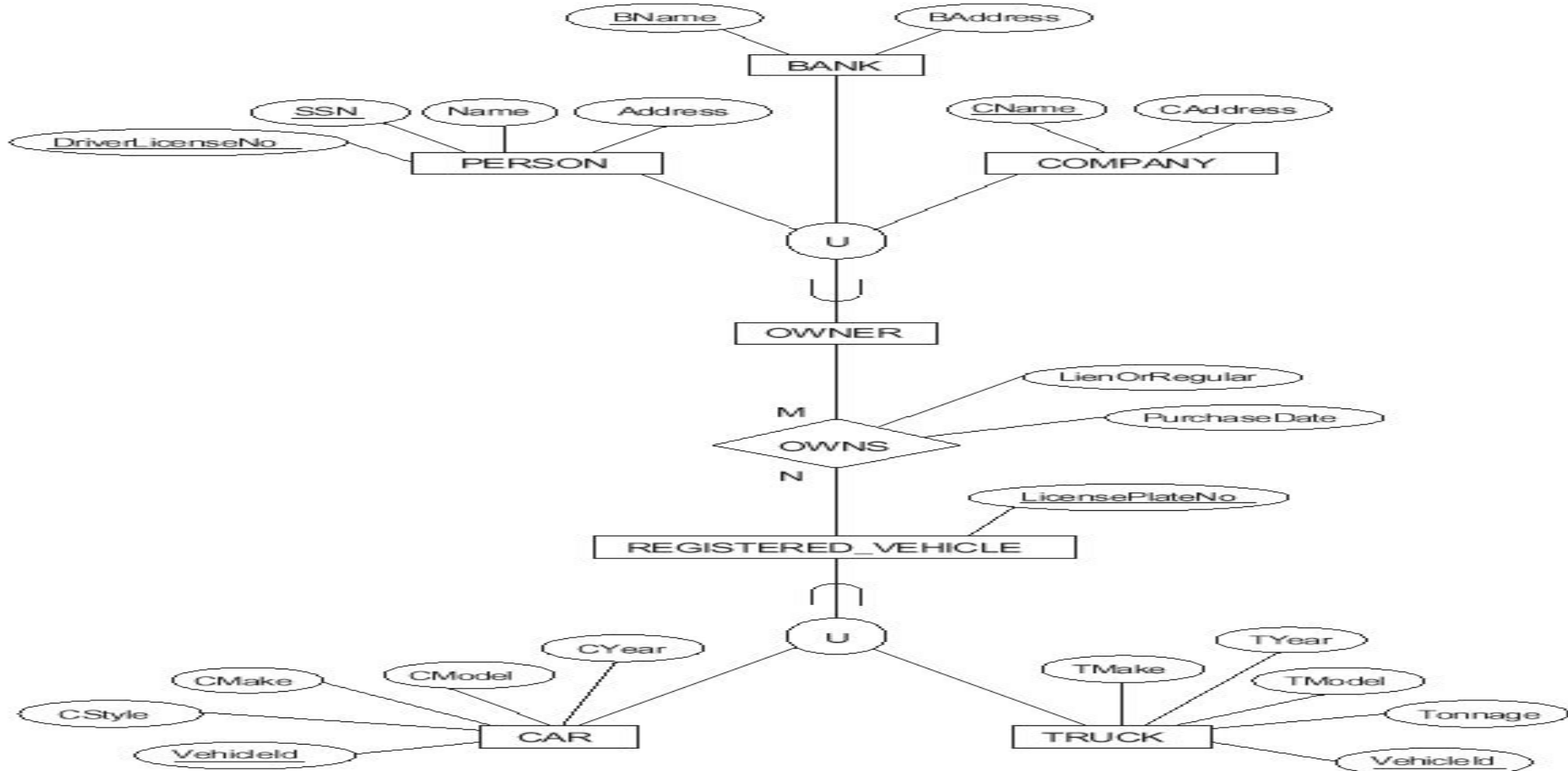
- Associate more than one superclass to a subclass.
- In categories, different entries of the subclass may inherit attributes from different superclasses
- An entity in Category is a member of only one of its superclass
- In specializations, all the subclasses inherit all the attributes of the superclass
- **Category types:** Total or Partial

Figure 4.7 A specialization lattice (with multiple inheritance) for a UNIVERSITY database.



Source: *Fundamentals of database systems* / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Figure 4.8 An illustration of how to represent the UNION of two or more entity types/classes using the category notation. Two categories are shown: OWNER and REGISTERED_VEHICLE.



Example of disjoint partial Specialization

Example of overlapping total Specialization

Source: Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed, Pearson Publications

Specialization/Generalization Hierarchies, Lattices & Shared Subclasses (1)

A subclass may itself have further subclasses specified on it

- forms a hierarchy or a lattice

Hierarchy has a constraint that every subclass has only one superclass (called ***single inheritance***); this is basically a ***tree structure***

In a ***lattice***, a subclass can be subclass of more than one superclass (called ***multiple inheritance***)

Shared Subclass “Engineering_Manager”

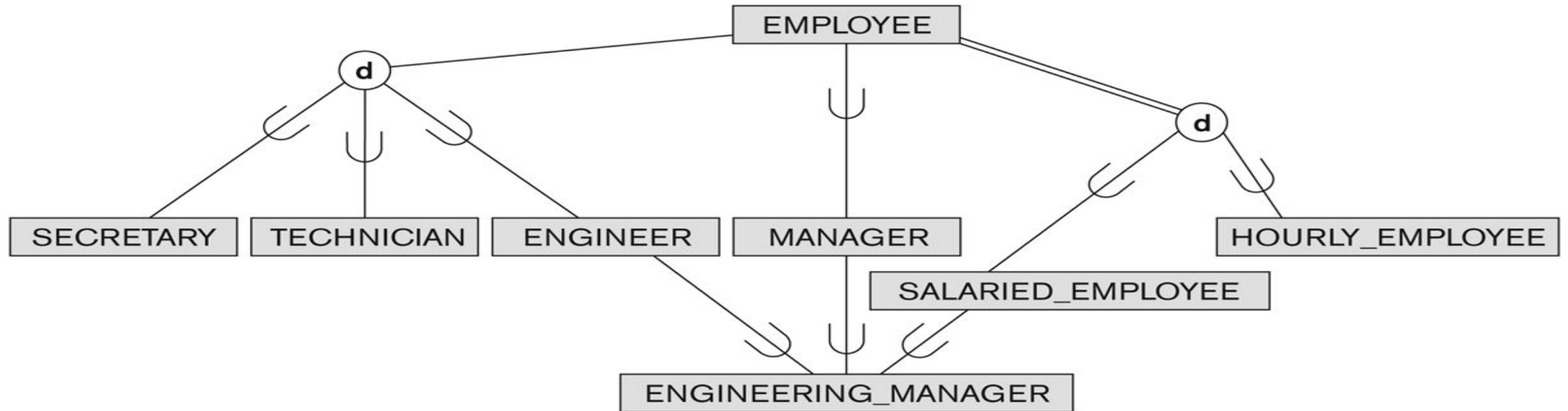


Figure 4.6

A specialization lattice with shared subclass ENGINEERING_MANAGER.

Specialization/Generalization Hierarchies, Lattices & Shared Subclasses (2)

In a lattice or hierarchy, a subclass inherits attributes not only of its direct superclass, but also of all its predecessor superclasses

A subclass with more than one superclass is called a shared subclass (multiple inheritance)

Can have:

- *specialization* hierarchies or lattices, or
- *generalization* hierarchies or lattices,
- depending on how they were *derived*

We just use *specialization* (to stand for the end result of either specialization or generalization)

Specialization/Generalization Hierarchies, Lattices & Shared Subclasses (3)

In *specialization*, start with an entity type and then define subclasses of the entity type by successive specialization

- called a *top down* conceptual refinement process

In *generalization*, start with many entity types and generalize those that have common properties

- Called a *bottom up* conceptual synthesis process

In practice, a *combination of both processes* is usually employed

Specialization / Generalization Lattice Example (UNIVERSITY)

-