IE301
Analysis and Design of Data Systems
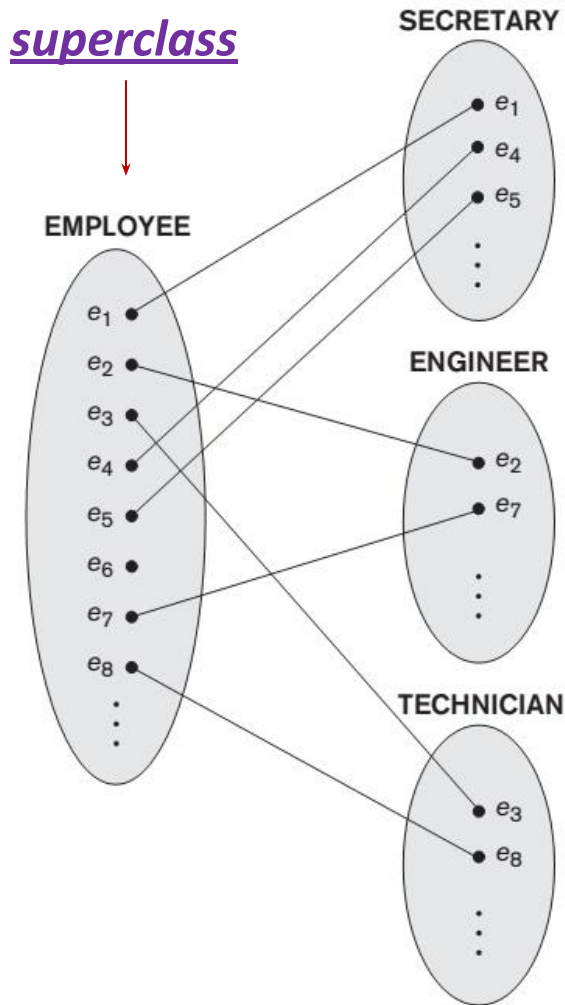
Lecture 11

Enhanced ER (EER) Model

Aram Keryan

Enhanced ER (EER) Model

The EER model includes *all the modeling concepts of the ER model*

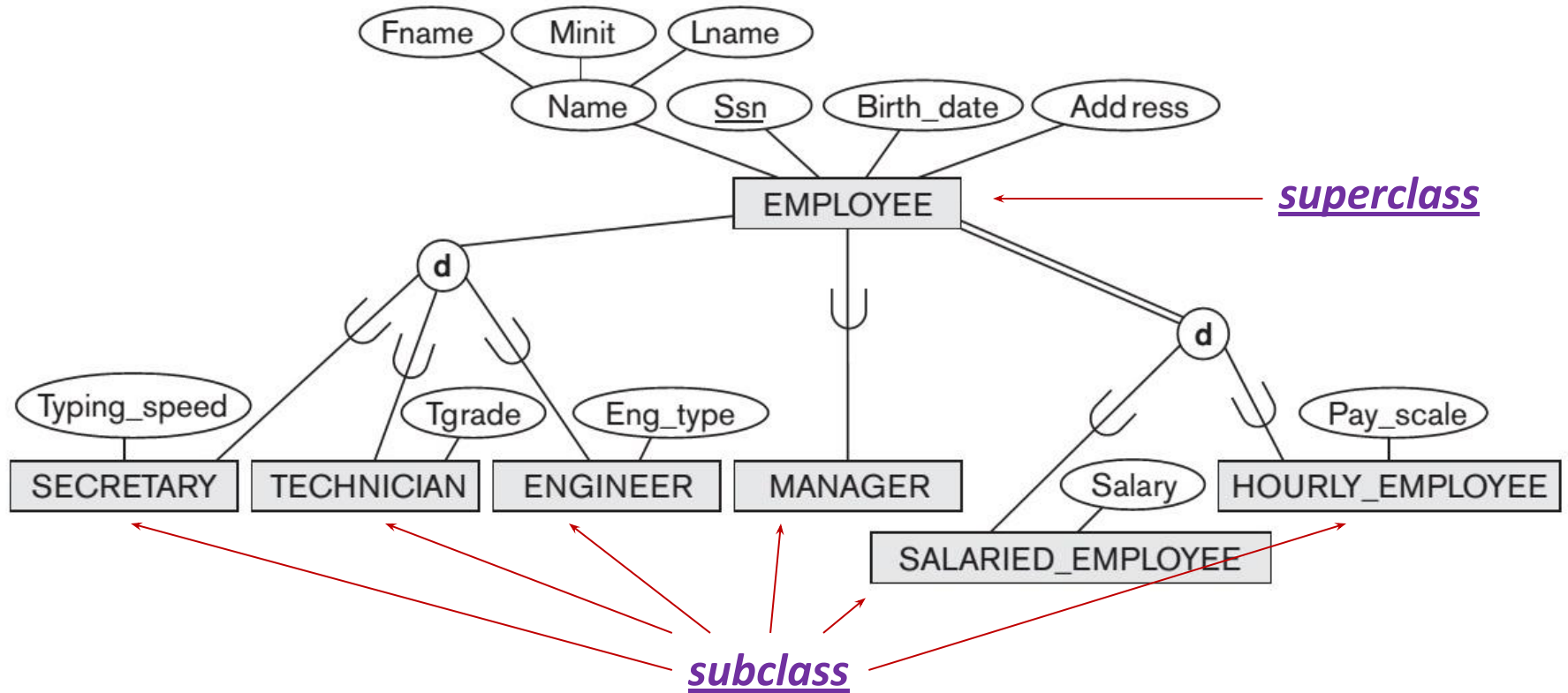# Subclasses and Superclasses

superclass

subclass

- An entity type can have numerous subgrupings that need to be represented explicitly because of their significance:

  - Each of these subgrupings is called a *subclass*

- Each entity that is a member of a subclass is also an entity of a corresponding superclass (*secretary is also an employee)*

- It is not necessary that every entity in a superclass is a member of some subclass
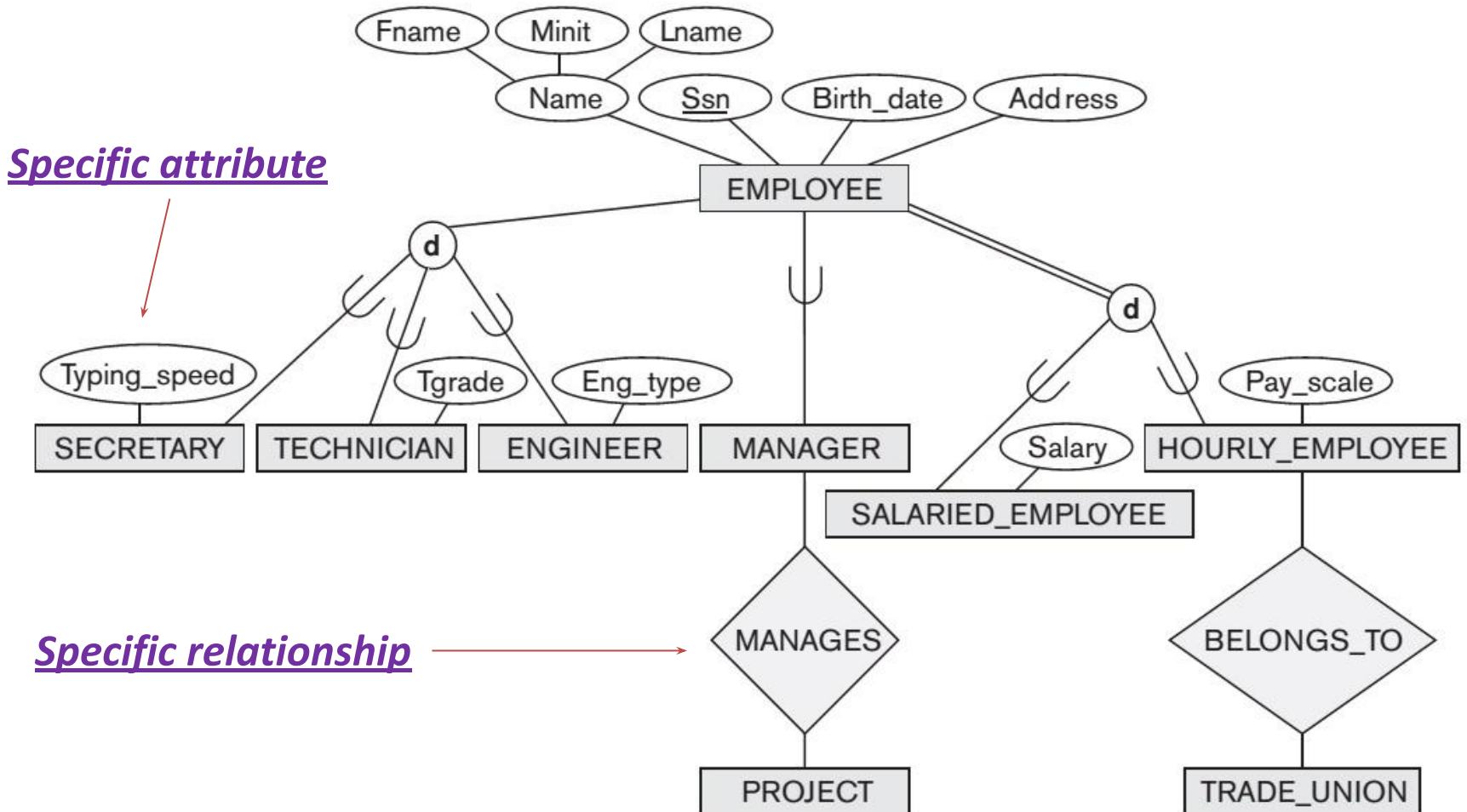
Subclasses and Superclasses

Specialization

- Specialization is the process of defining a set of subclasses of an entity type
- The set of subclasses that forms a specialization is defined on the basis of some distinguishing characteristic of the entities in the superclass

Based on the job type

Based on the method of pay
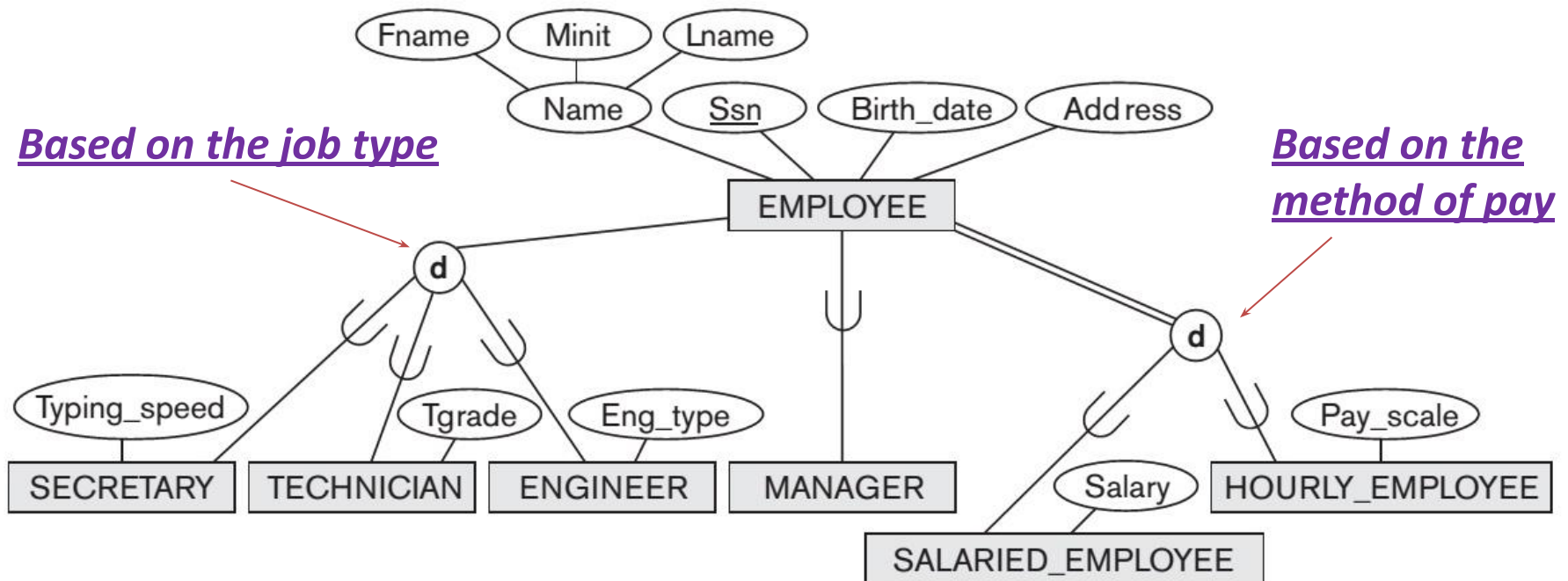
More on Subclasses and Superclasses

- Each entity that is a member of a subclass can also be a member of another subclass (salaried employee who is also an engineer belongs to the two subclasses ENGINEER and SALARIED_EMPLOYEE of the EMPLOYEE entity type)

- Entity that is a member of a subclass *inherits* all the attributes of the entity as a member of the superclass

- Entity that is a member of a subclass also inherits all the relationships in which the superclass participates

Features of Subclasses

Specific attribute

Specific relationship

# Reasons for including class/subclass relationships and specializations in a data model:

- Certain attributes may apply to some but not all entities of the superclass
- Some relationship types may be participated in only by entities that are members of the subclass.

# Specialization process allows us to do the following:

- Define a set of subclasses of an entity type
- Establish additional specific attributes with each subclass
- Establish additional specific relationship types between each subclass and other entity types or other subclasses

Generalization

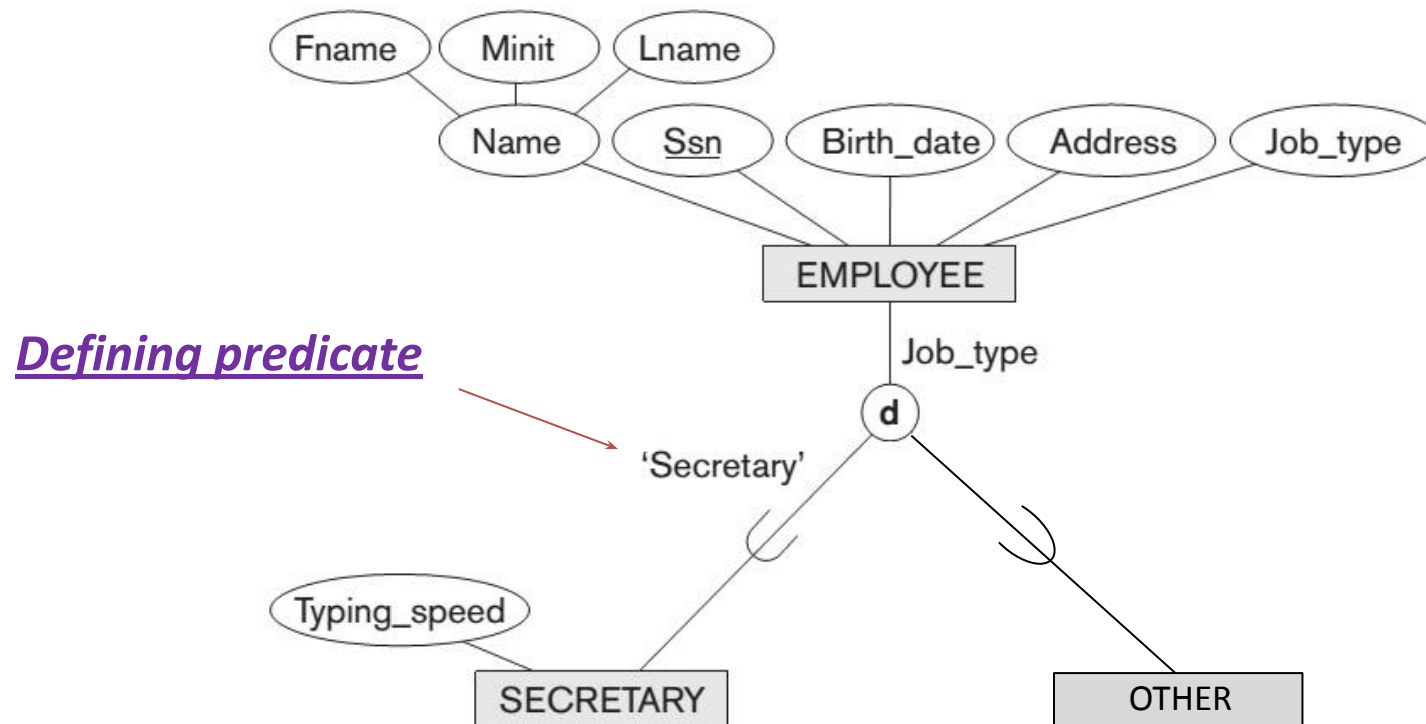Generalization is a reverse process to Specialization
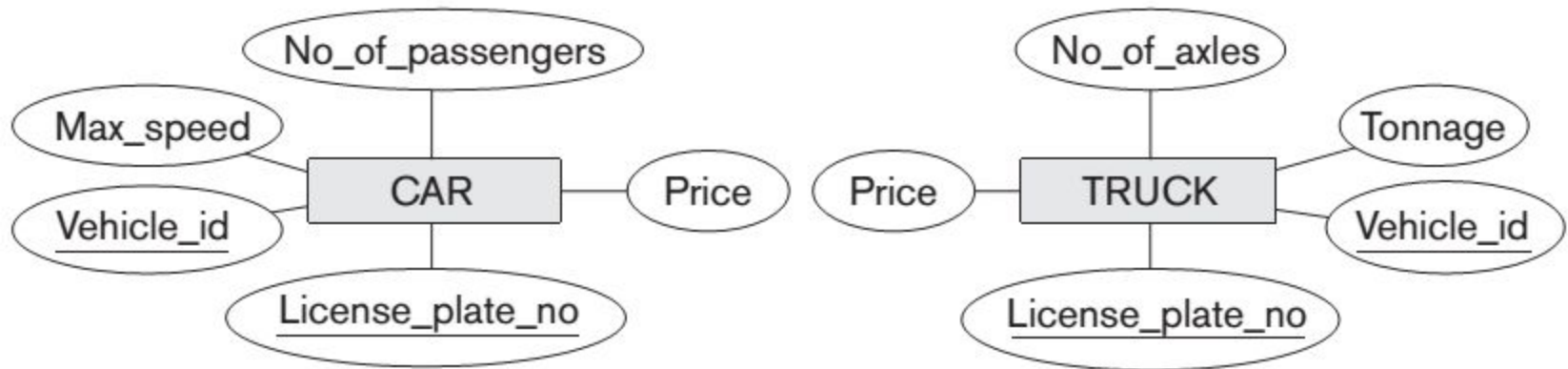
Generalization
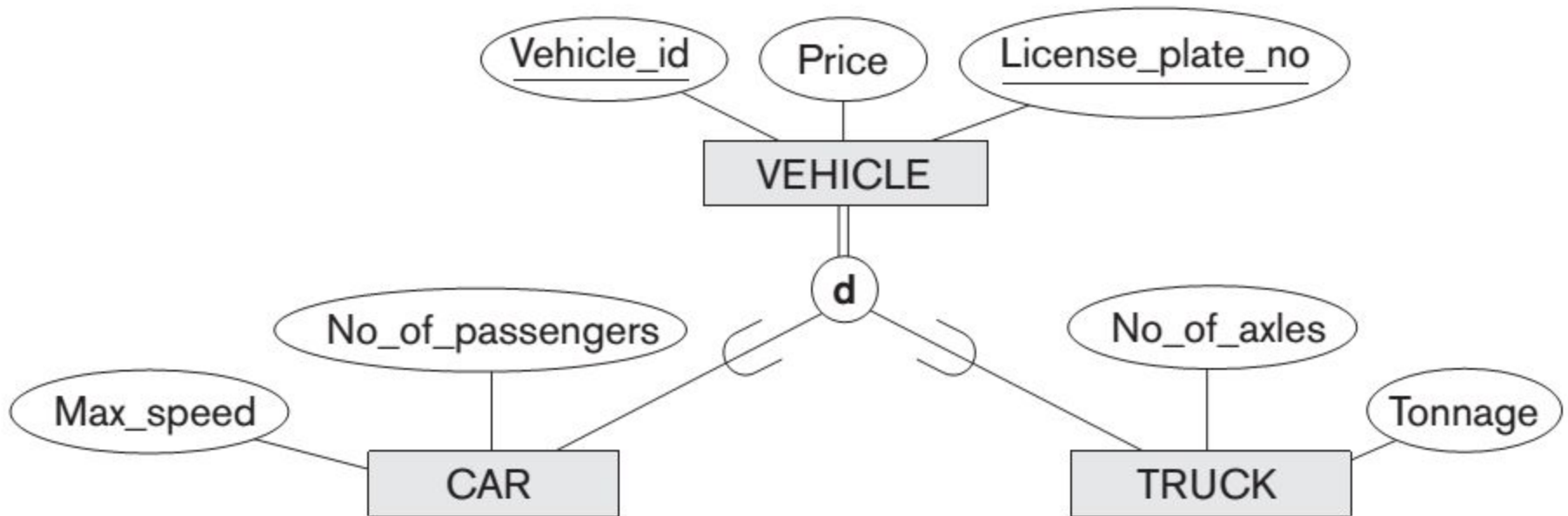
Constraints on Specializations

By placing a condition on the value of some attribute of the superclass we can determine exactly the entities that will become members of each subclass. Such subclasses are called **predicate-defined subclasses.**

Defining predicate

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

Defining attribute
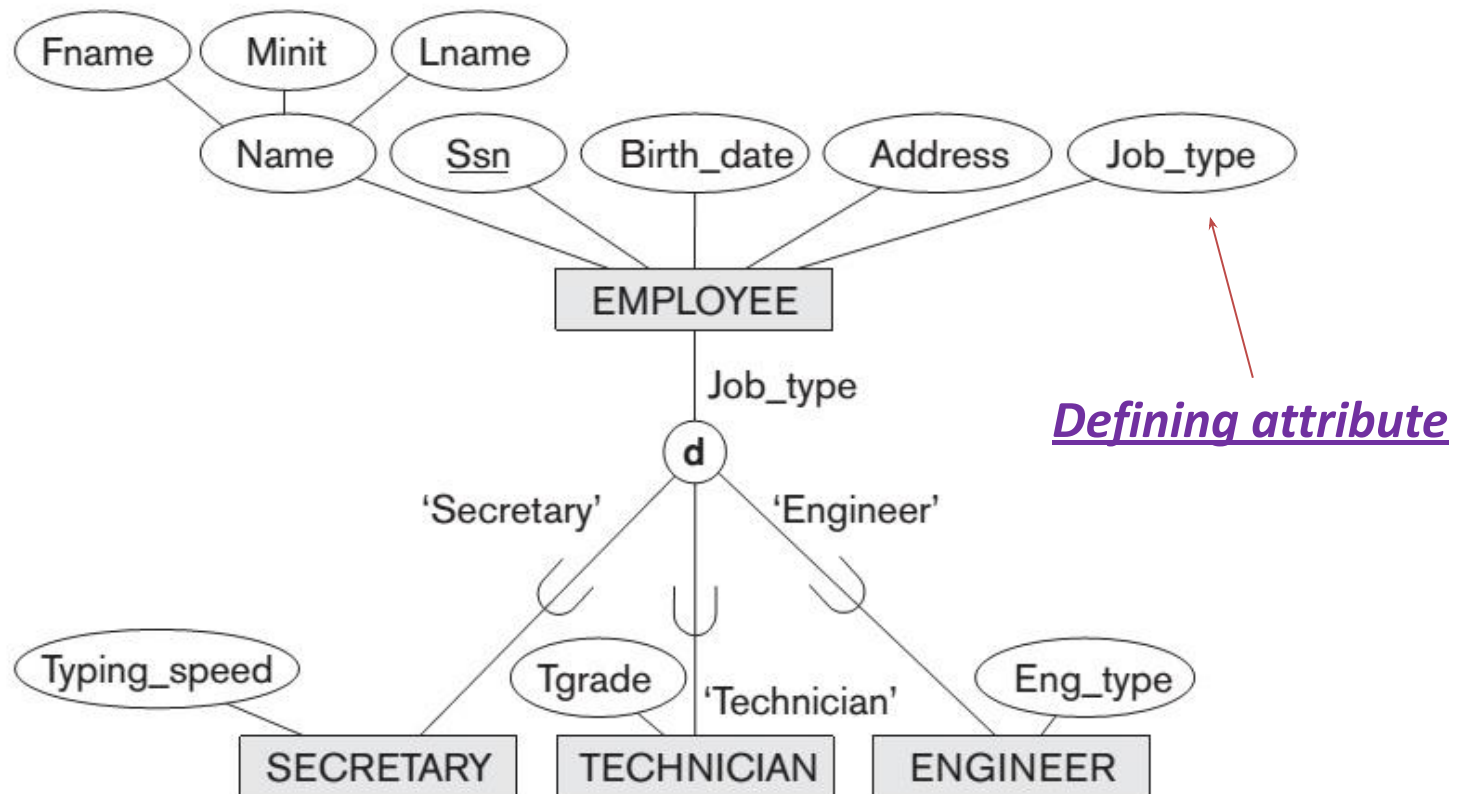
When we do not have a condition for determining membership in a subclass, the subclass is called **user-defined**

Disjointness constraint specifies that an entity can be a member of *at most* one of the subclasses of the specialization.

✔ A specialization that is attribute-defined implies the disjointness constraint
✔ Disjointness also applies to user-defined subclasses of a specialization
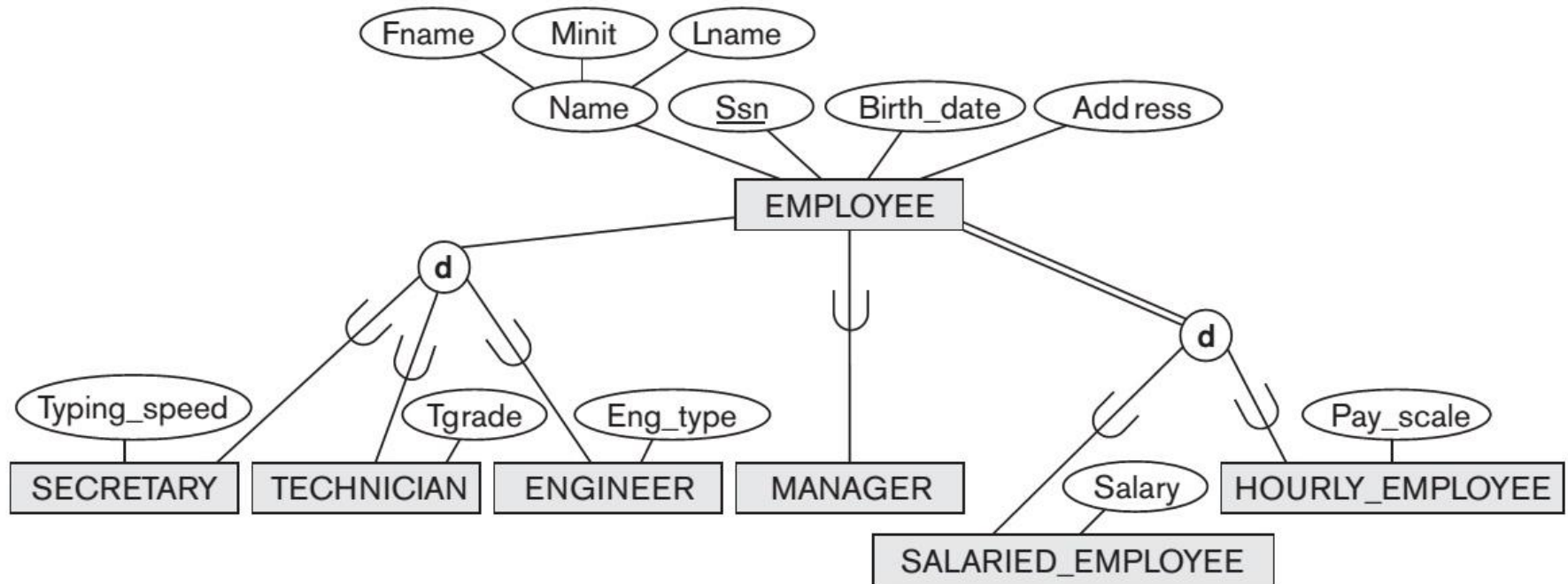
If the subclasses are not constrained to be disjoint, their sets of entities may be overlapping

✔ that is, the same (real-world) entity may be a member of more than one subclass of the specialization
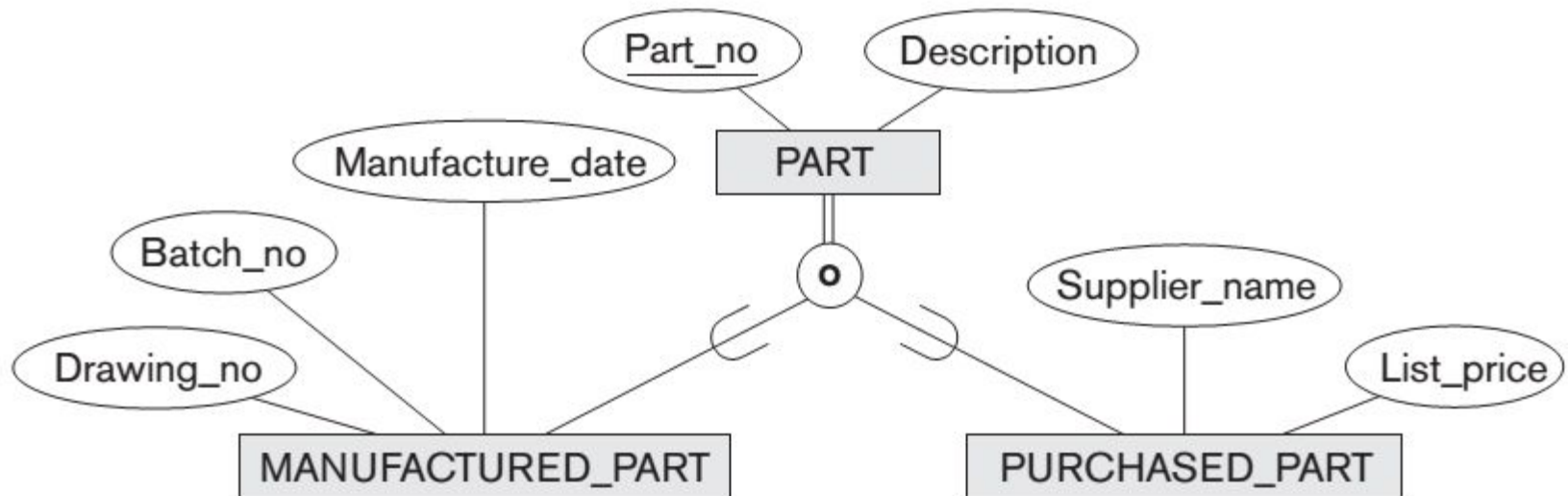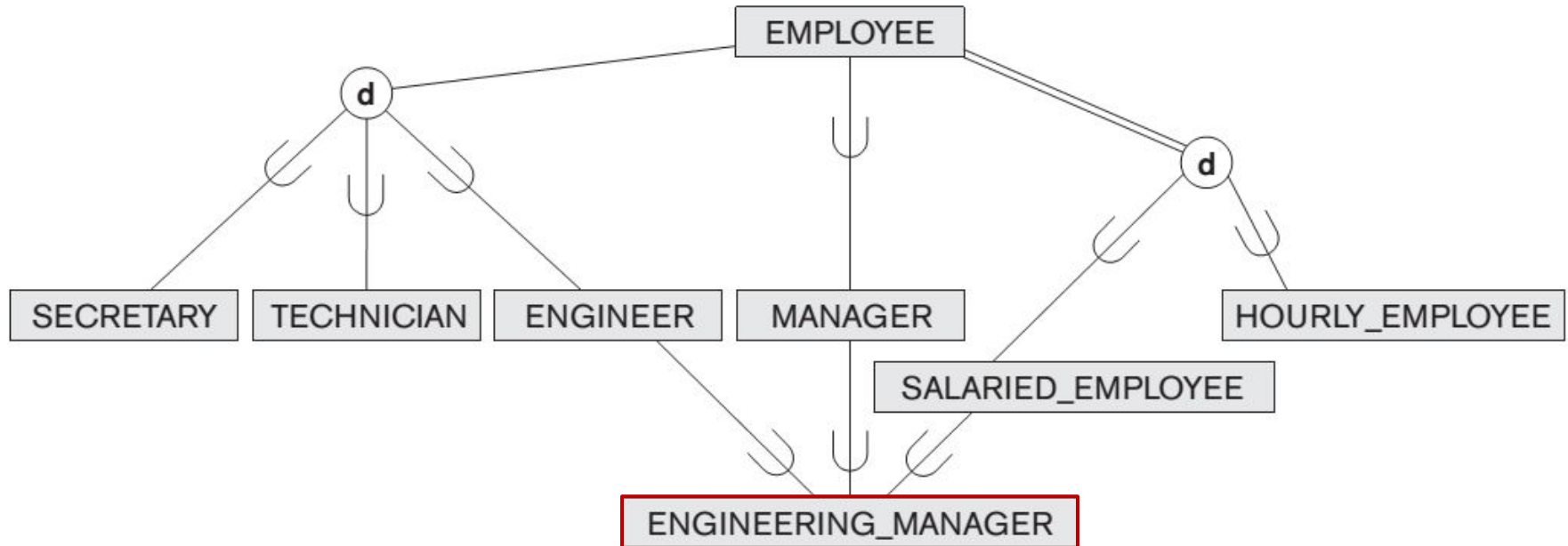
A **total specialization** constraint specifies that *every* entity in the superclass must be a member of at least one subclass in the specialization
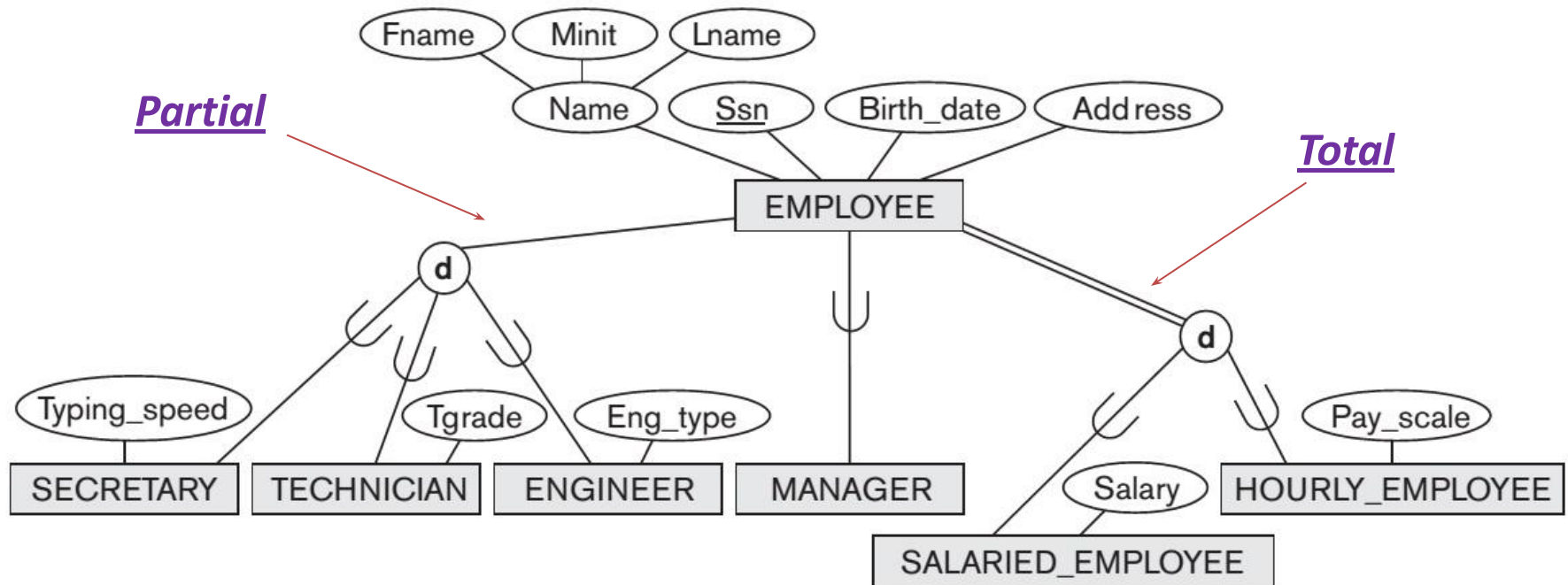
partial specialization allows an entity not to belong to any of the subclasses
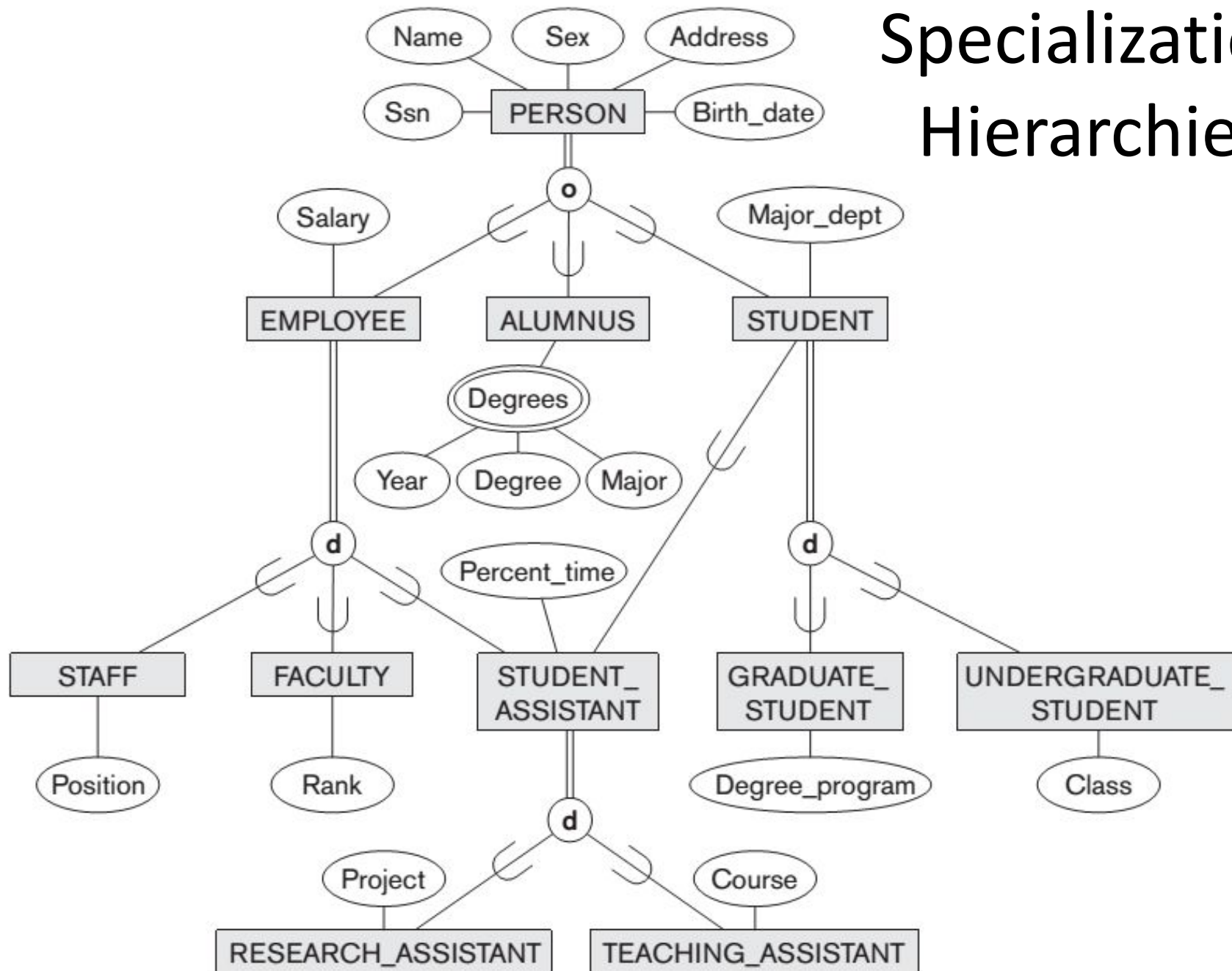
Shared sublasses

Shared sublcasses inherit attributes and relationships from multiple classes. That concept is known as ***multiple inheritance***
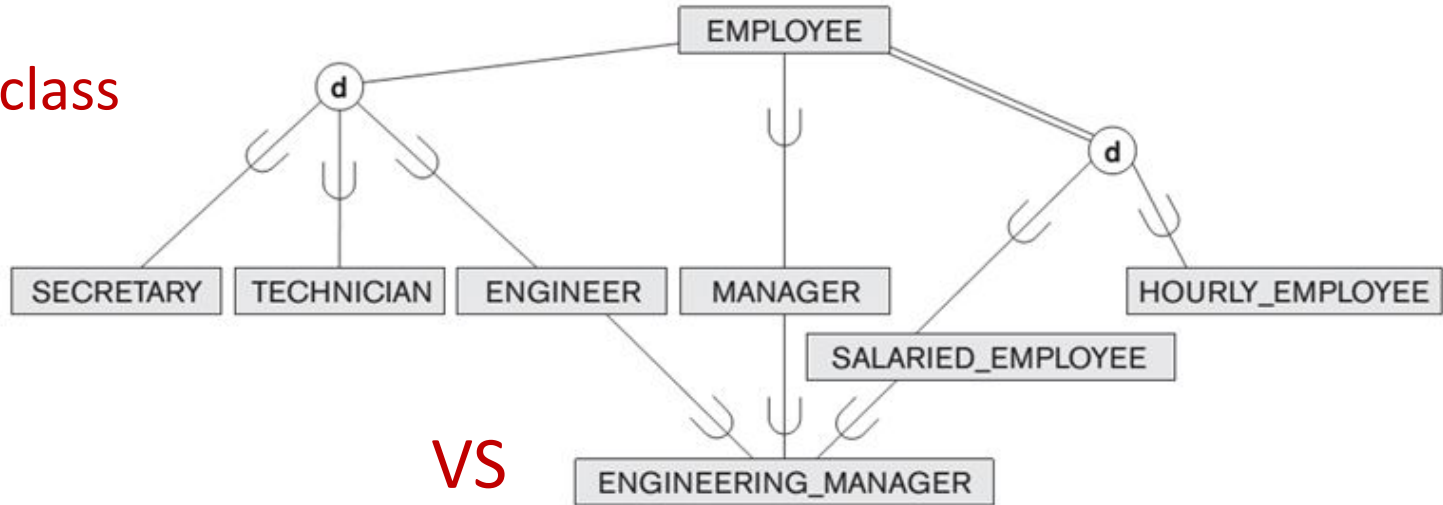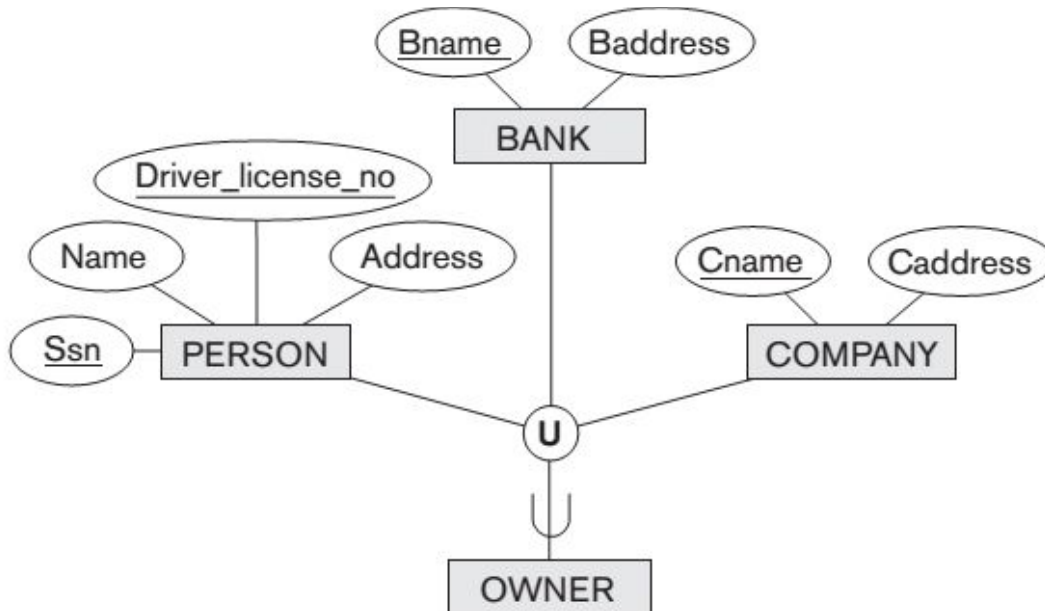
Specialization Hierarchies

UNION Subclasses

Shared Subclass

Superclasses must all have the same key

VS

Union Subclass

Superclasses can have different keys

UNION Subclasses

ENGINEERING_MANAGER is a subclass of *each of* the three superclasses ENGINEER, MANAGER, and SALARIED_EMPLOYEE, so an entity that is a member of ENGINEERING_MANAGER must exist in *all three.*
This represents the constraint that an engineering manager must be an ENGINEER, a MANAGER, *and* a SALARIED_EMPLOYEE simultaneously;
that is, ENGINEERING_MANAGER is a subset of the *intersection* of the three classes (sets of entities).
Shared subclass such as ENGINEERING_MANAGER inherits *all* the attributes of its superclasses SALARIED_EMPLOYEE, ENGINEER, and MANAGER

Union subclass is a subset of the *union* of its superclasses.
Hence, an entity that is a member of OWNER must exist in *only one* of the superclasses. This represents the constraint that an OWNER may be a COMPANY, a BANK, *or* a PERSON.
Union subclass such as OWNER entity inherits the attributes of a COMPANY, a PERSON, or a BANK, depending on the superclass to which the entity belongs.

UNION Subclasses
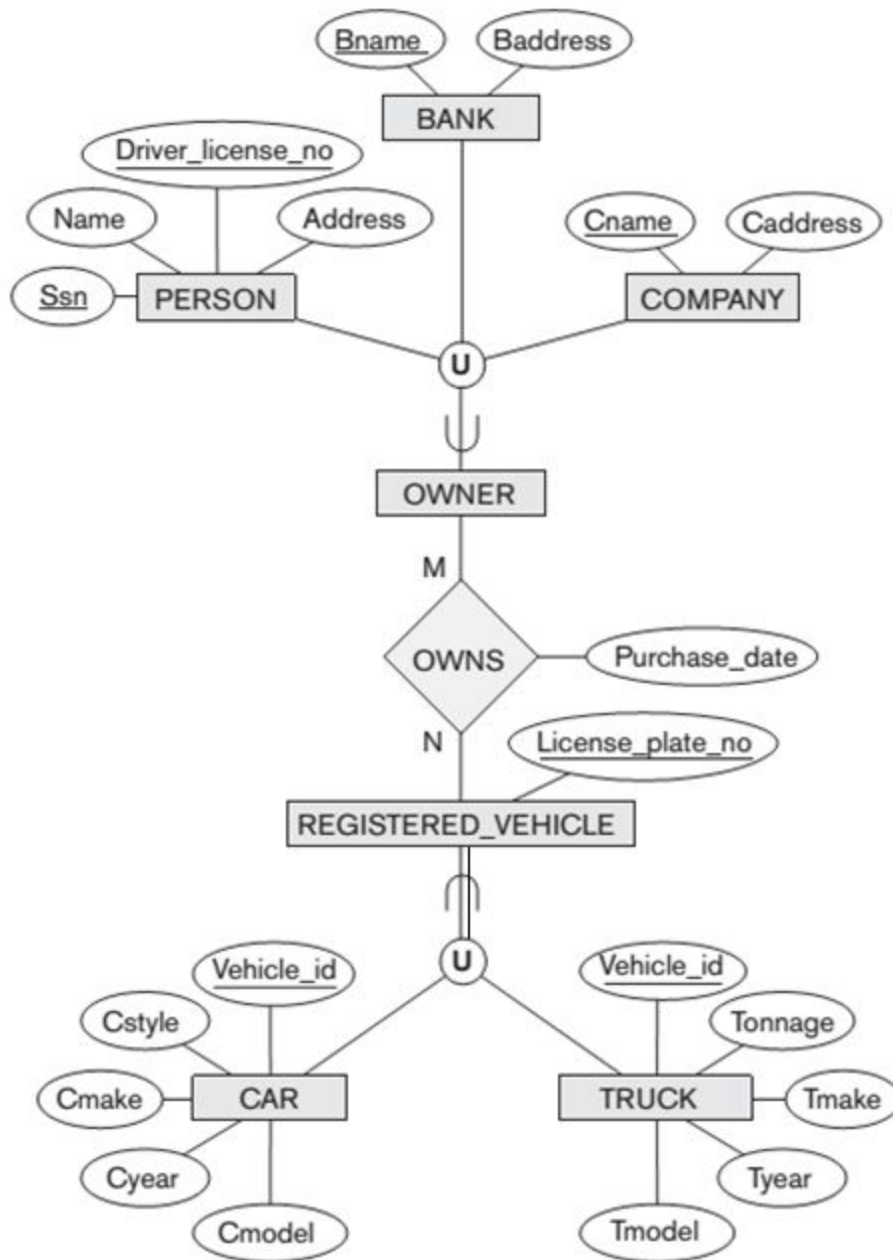
A Union Subclass can be **total** or **partial**. A total category holds the *union* of all entities in its superclasses, whereas a partial category can hold a *subset of the union*. A total category is represented diagrammatically by a double line connecting the category and the circle, whereas a partial category is indicated by a single line.