Database Management Systems

ICT1212

Enhanced Entity-Relationship & Object Modeling

(Conceptual Data Modeling)

Department of ICT

Faculty of Technology

University of Ruhuna

Lecture 5

What we discuss Today……..

- Subclasses, Super classes, and Inheritance
- Specialization and Generalization
- Constraints and Characteristics of Specialization and Generalization
- Modeling of UNION Types Using Categories
- An Example UNIVERSITY EER Schema and Formal Definitions for the EER Model
- Conceptual Object Modeling Using UML Class Diagrams
- Relationship Types of a Degree Higher Than Two
- Data Abstraction and Knowledge Representation Concepts

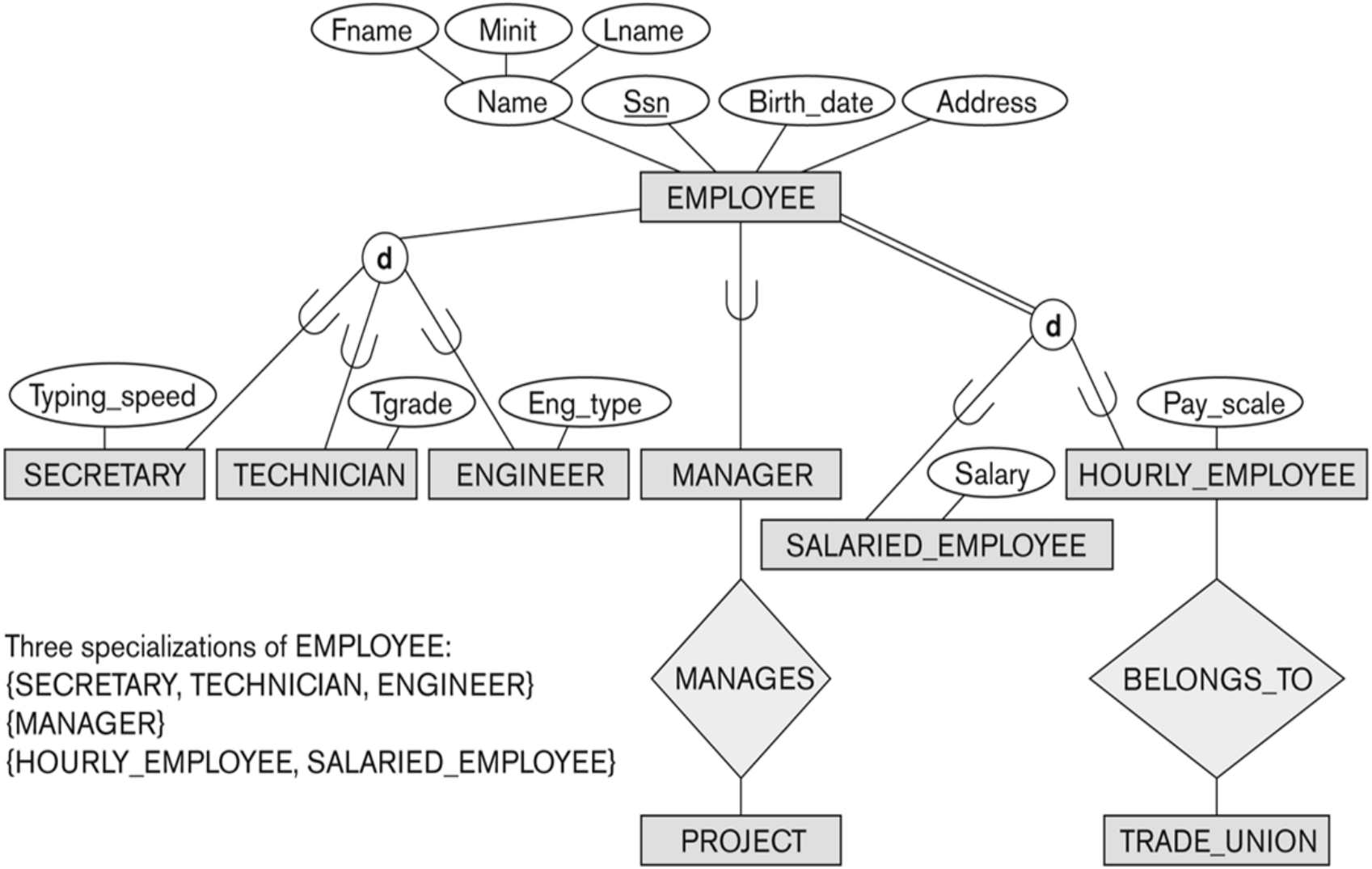
Summary of ER Modeling

- Entity Types, Entity Sets, Attributes, and Keys
- The Conventions for ER Diagrams
- Relationships, Relationship Types, Roles, and Structural Constraints
- Weak Entity Types
- ER Design for the COMPANY Database
- ER Diagrams, Naming Conventions, and Design Issues

Subclasses and Superclasses

- 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

Subclasses and Superclasses



Subclasses and Superclasses

- 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

- 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

- 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

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

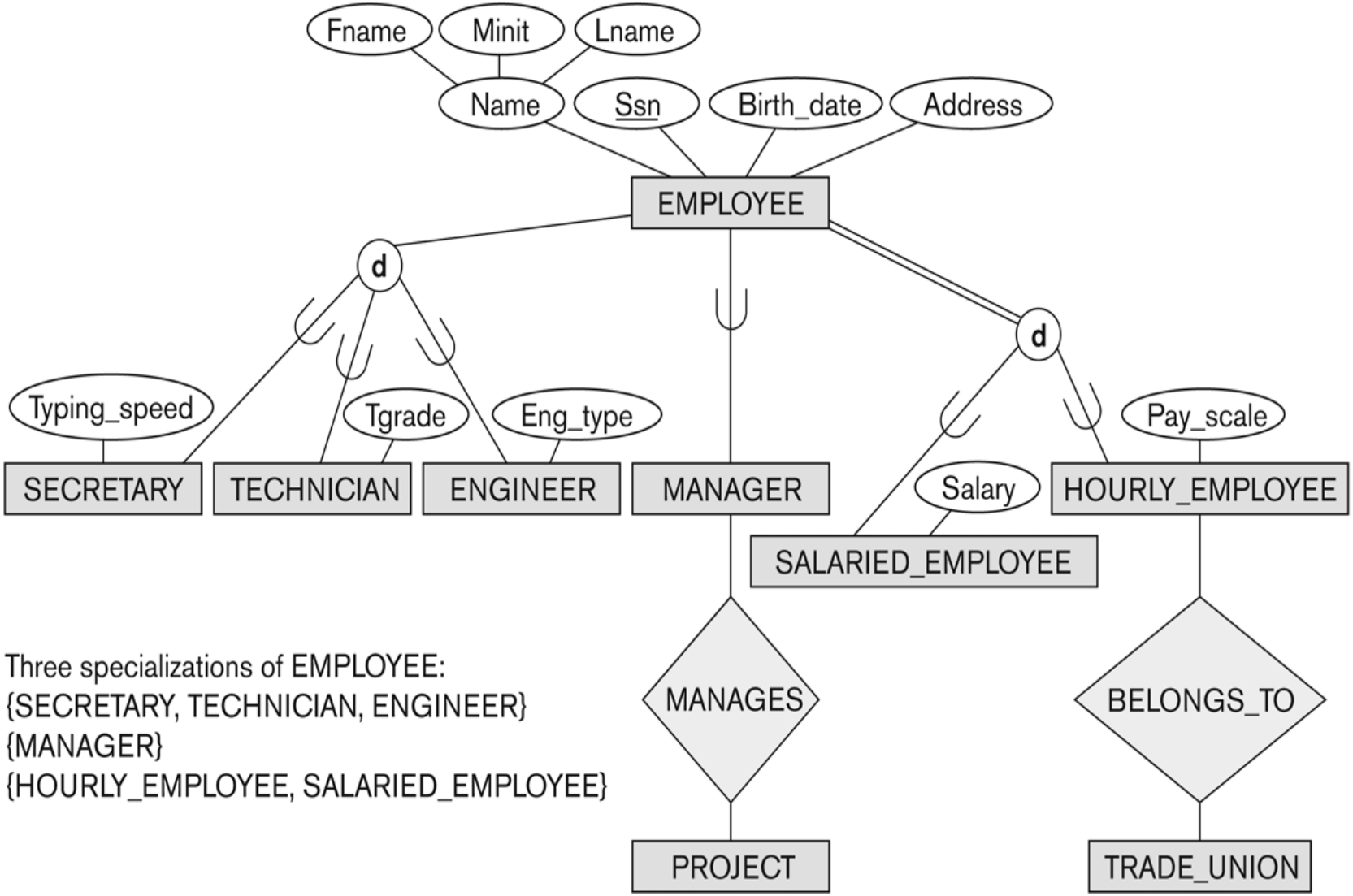
- Specialization is the process of defining a set of subclasses of a superclass

- 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

Specialization



Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}

Two main reasons for Specialization

1. Certain attributes may apply to some but not all entities of the superclass
2. Some relationship types may be participated in only by entities that are members of the subclass.
   - Example :-
     - if only HOURLY_EMPLOYEEs can belong to a trade union, we can represent that fact by creating the subclass HOURLY_EMPLOYEE of EMPLOYEE and relating the subclass to an entity type TRADE_UNION via the BELONGS_TO relationship type

Specialization process

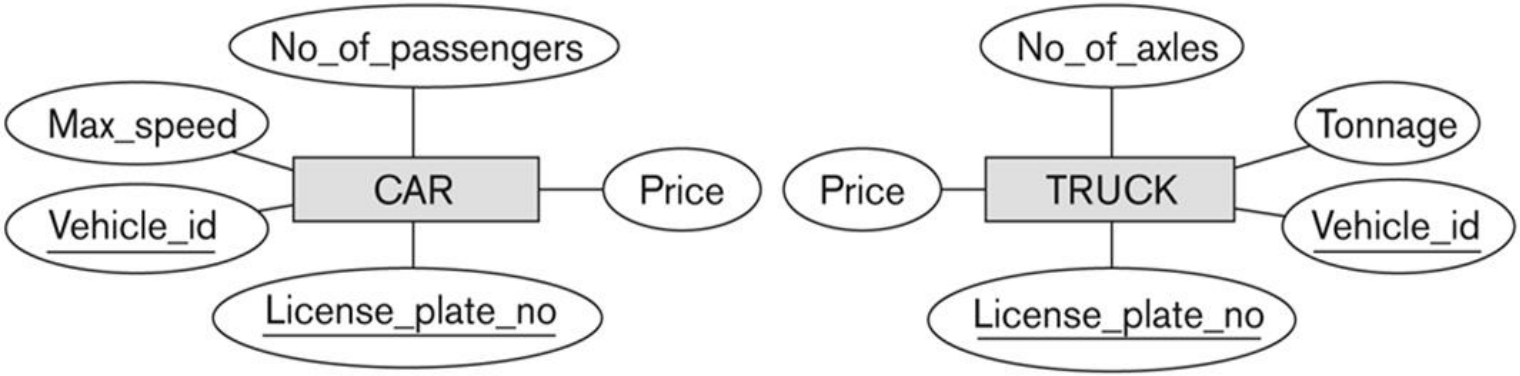
- Define a set of subclasses of an entity type.
- Establish additional specific attributes with each subclass.
- Establish additional specific relationship types between each subclass and other entity types or other subclasses.

Generalization

- Generalization is the reverse of the specialization process
- Several classes with common features are generalized into a superclass;
  - original classes become its subclasses
- Example:

  CAR, TRUCK generalized into VEHICLE;
  - both CAR, TRUCK become subclasses of the superclass VEHICLE.
  - We can view CAR, TRUCK as a specialization of VEHICLE
  - Alternatively, we can view VEHICLE as a generalization of CAR and TRUCK

Generalization



Generalization and Specialization

- Diagrammatic notation are sometimes used to distinguish between generalization and specialization
  - Arrow pointing to the generalized superclass represents a generalization
  - Arrows pointing to the specialized subclasses represent a specialization
  - We do not use this notation because it is often subjective as to which process is more appropriate for a particular situation
  - Advice is not to draw any arrows

Generalization and Specialization

- A superclass or subclass represents a collection (or set or grouping) of entities
- It also represents a particular type of entity
- Shown in rectangles in EER diagrams (as are entity types)
- We can call all entity types (and their corresponding collections) classes, whether they are entity types, superclasses, or subclasses

Constraints on Specialization and Generalization

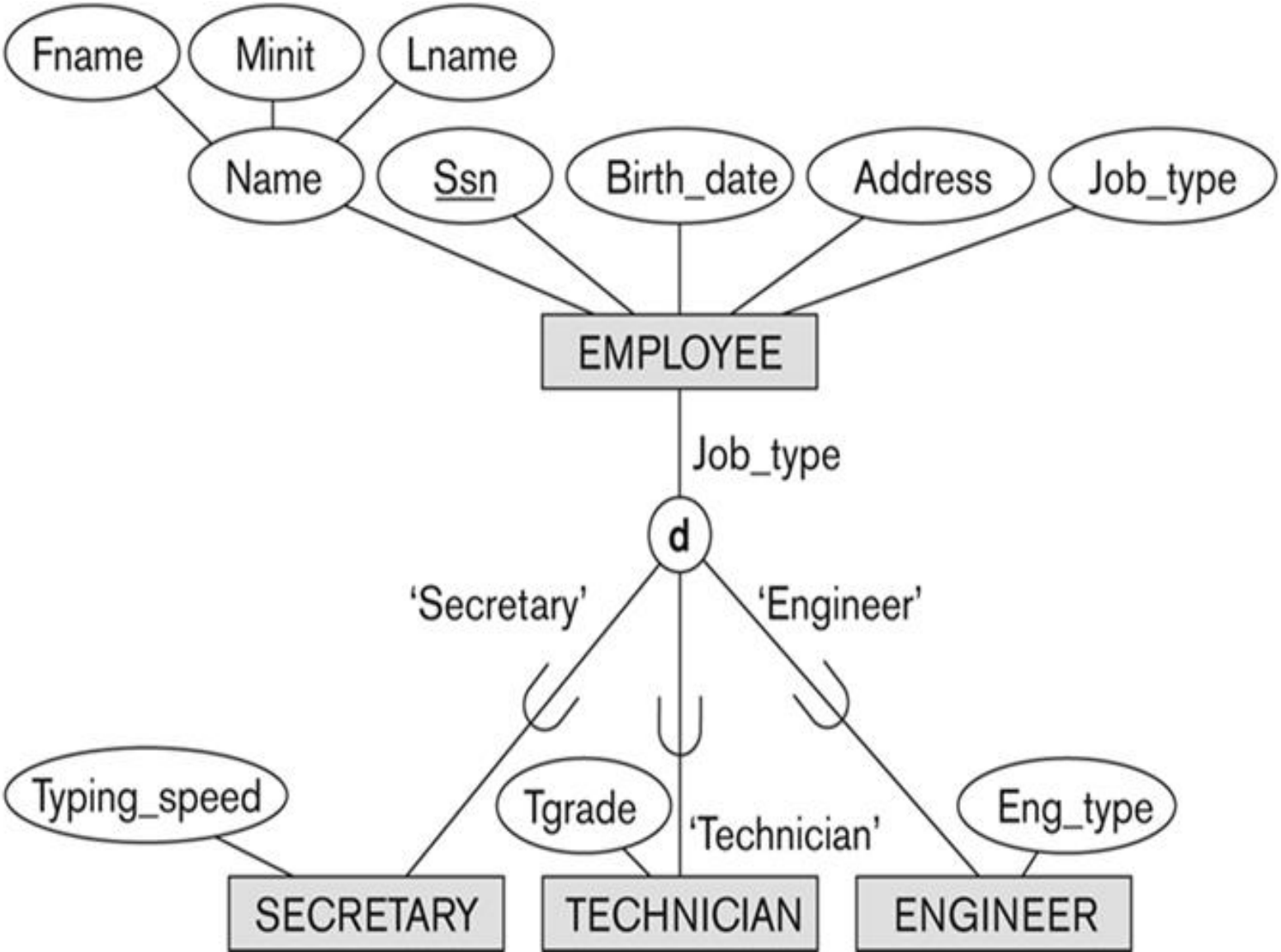
- If we can determine exactly the entities that will become members of each subclass by placing a condition on the value of some attribute of the super class, the subclasses are called **predicate-defined (or condition-defined)** subclasses
  - Condition is a constraint that determines subclass members
  - 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

- 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:

    job_type 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

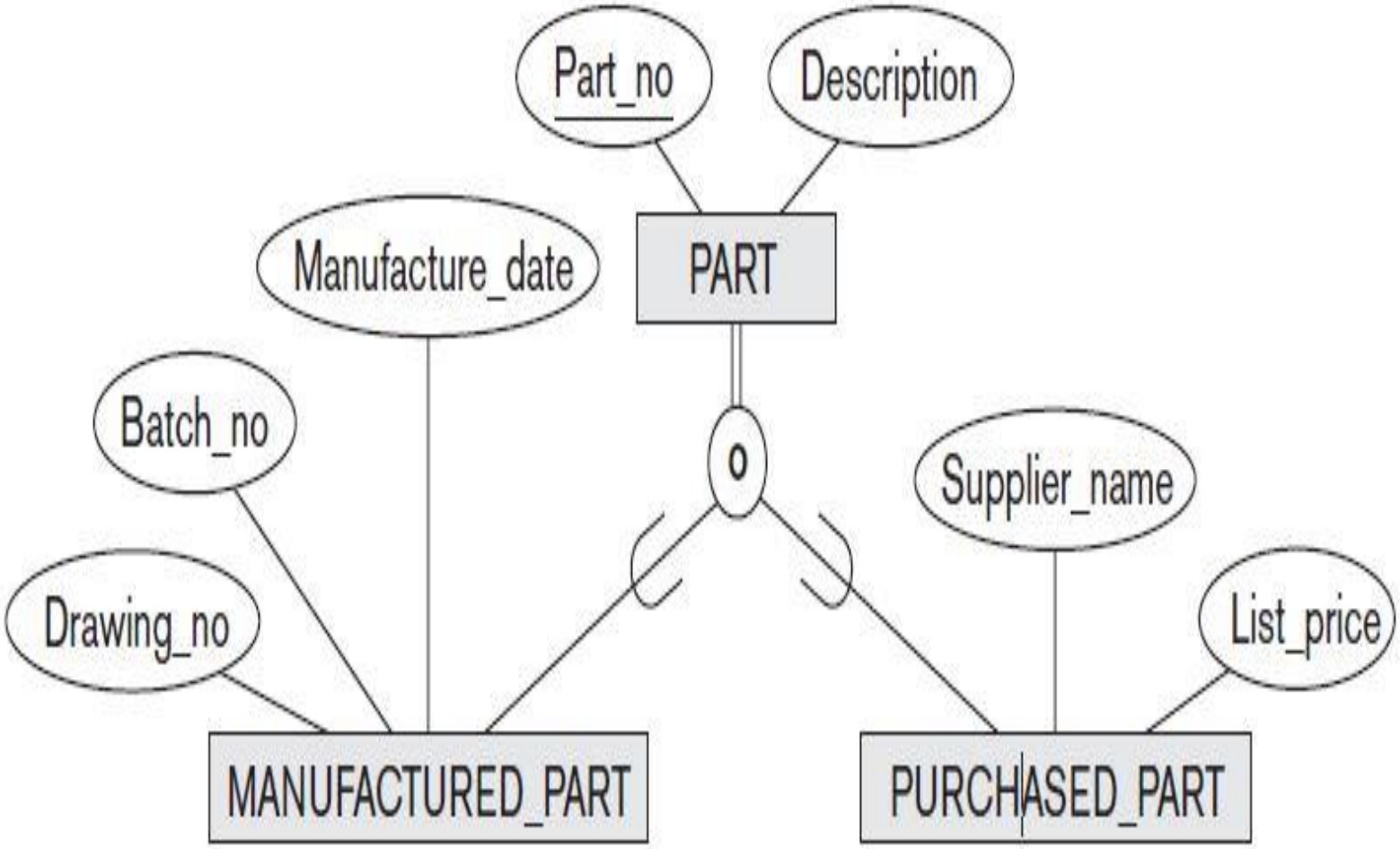
- Two other basic constraints can also apply to a specialization/generalization
  - Disjointness(Disjointedness) Constraint
  - Completeness Constraint

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

Overlapping



Completeness Constraint

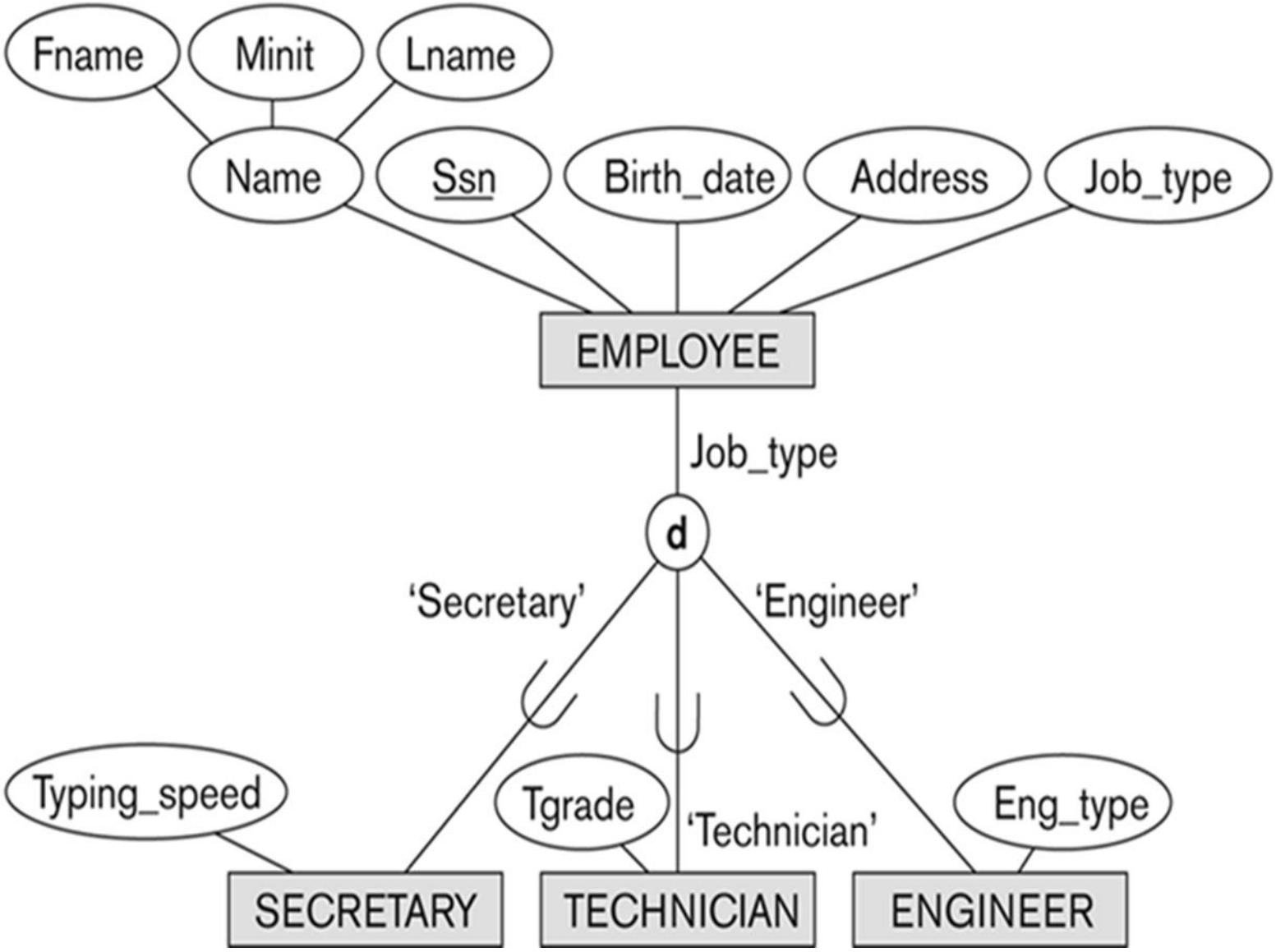
Total

- Total specifies that every entity in the superclass must be a member of at least one subclass in the specialization
- Shown in EER diagrams by a double line

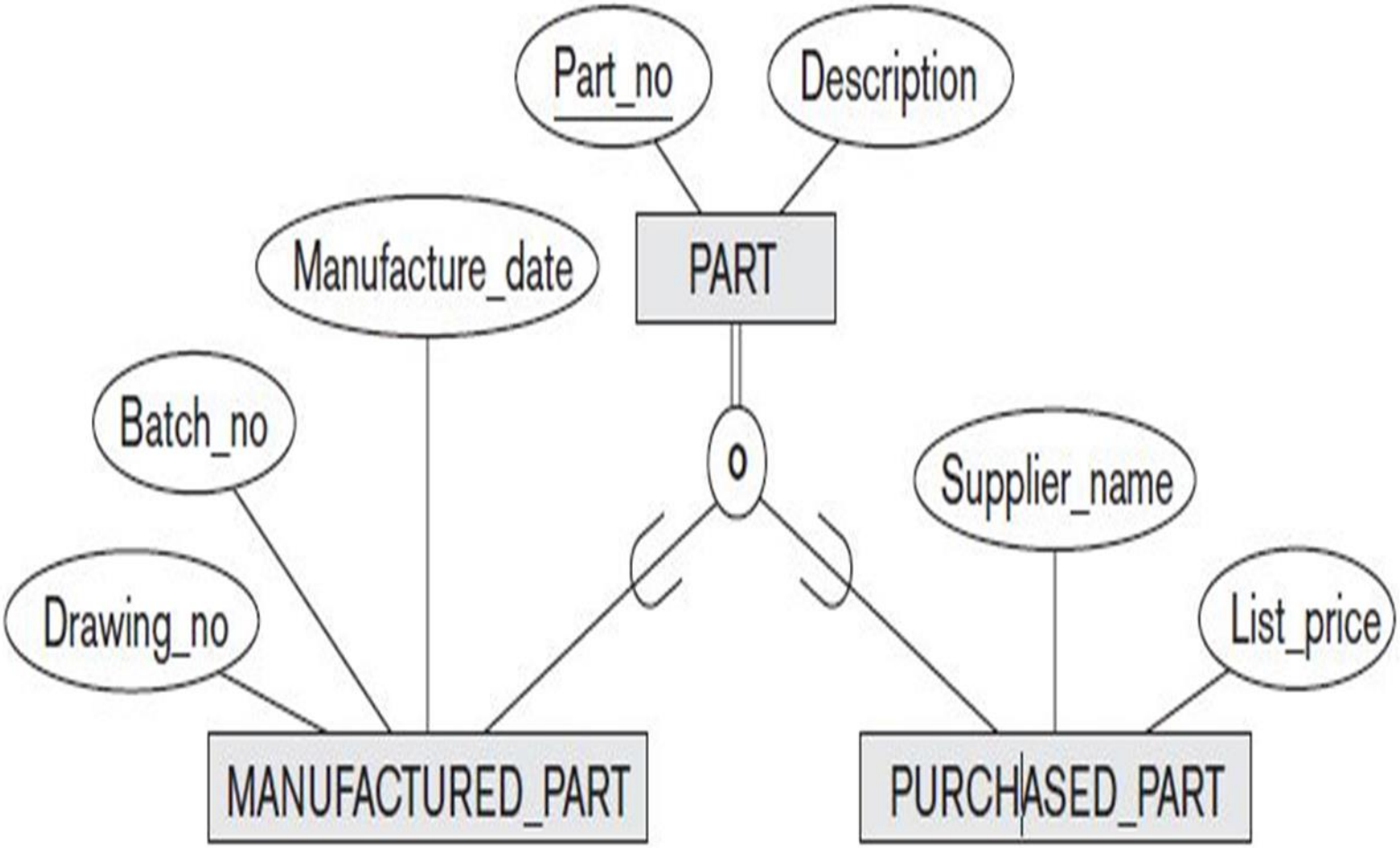
Partial

- Partial allows an entity not to belong to any of the subclasses
- Shown in EER diagrams by a single line

Example of disjoint partial Specialization



Example of overlapping total Specialization



Four possible constraints on specialization

- Disjointness and Completeness constraints are *independent*. Hence, we have the following four possible constraints on specialization
  - Disjoint, total
  - Disjoint, partial
  - Overlapping, total
  - Overlapping, partial

Insertion and Deletion Rules

- Deleting an entity from a superclass implies that it is automatically deleted from all the subclasses to which it belongs.
- Inserting an entity in a superclass implies that the entity is mandatorily inserted in all predicate-defined (or attribute-defined) subclasses for which the entity satisfies the defining predicate.
- Inserting an entity in a superclass of a total specialization implies that the entity is mandatorily inserted in at least one of the subclasses of the specialization.

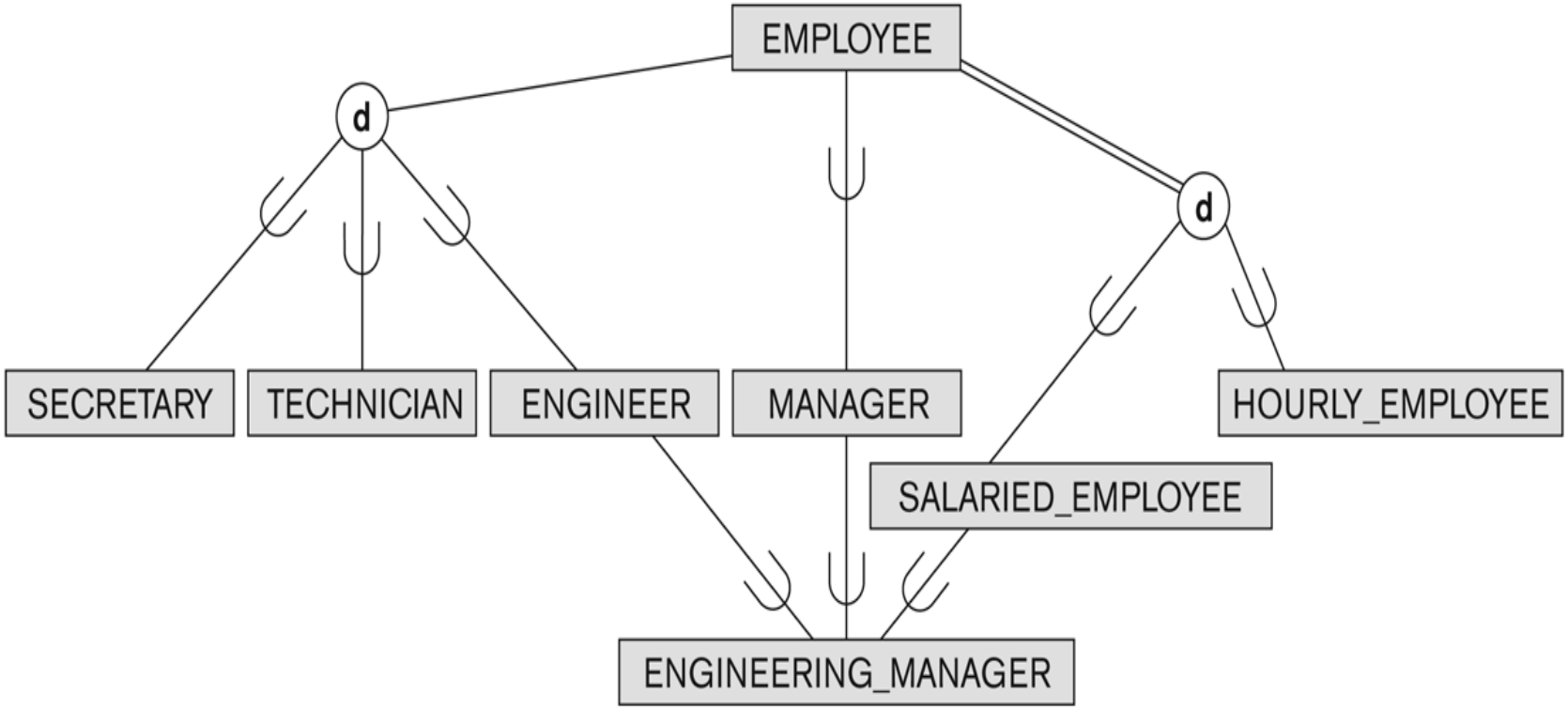
Specialization/Generalization Hierarchies, Lattices & Shared Subclasses

- 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)

Specialization/Generalization Hierarchies, Lattices & Shared Subclasses

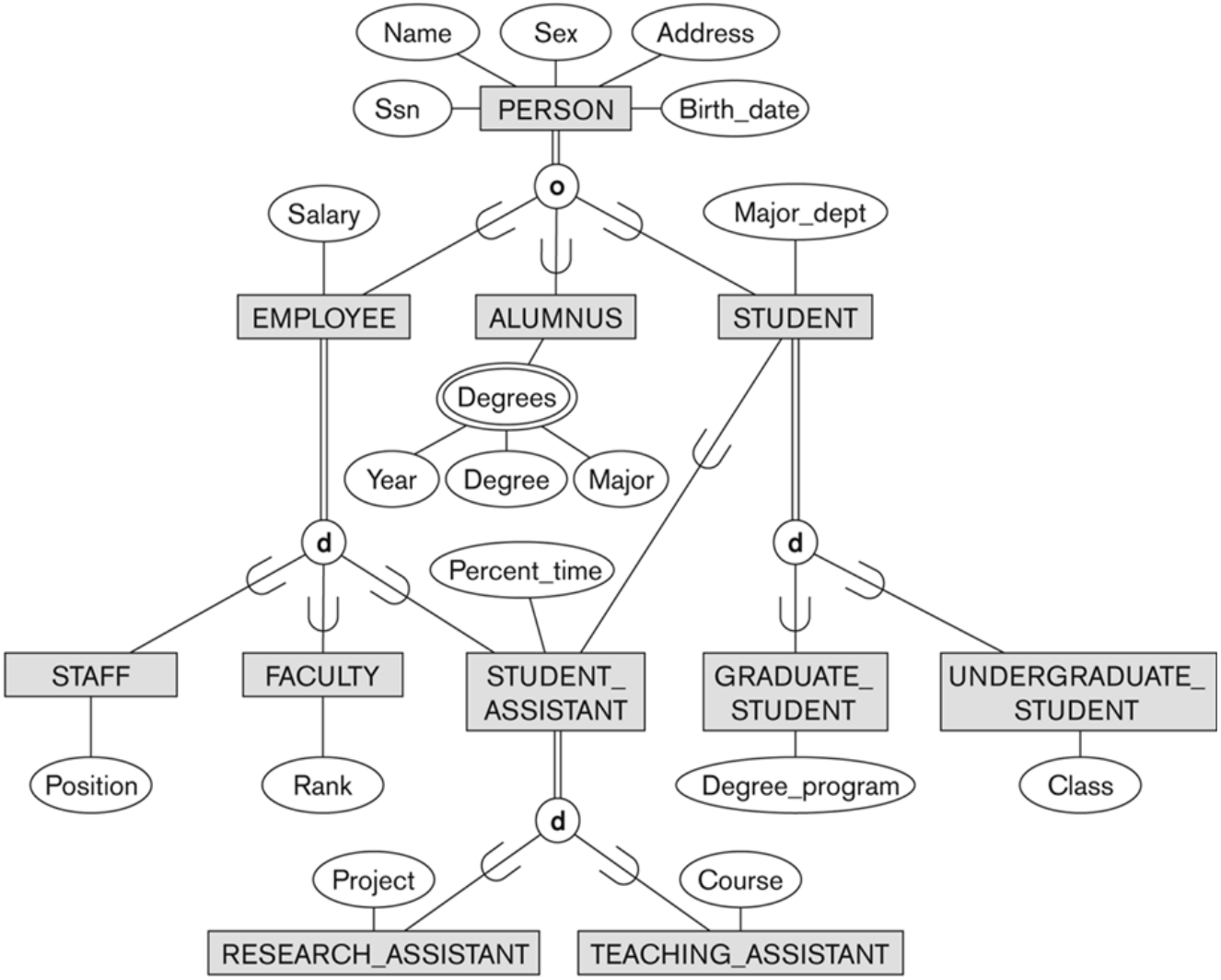
- In a lattice or hierarchy, a subclass inherits attributes not only of its direct superclass, but also of all its predecessor superclasses
- A subclass with more than one superclass is called a shared subclass (multiple inheritance)
- Can have:
  - specialization hierarchies or lattices, or
  - generalization hierarchies or lattices,
  - depending on how they were derived
- For example lets just use specialization (to stand for the end result of either specialization or generalization) as shown below slide.

Shared Subclass "Engineering_Manager"



A specialization lattice with shared subclass ENGINEERING_MANAGER.

Specialization / Generalization Lattice Example (UNIVERSITY)



A specialization lattice with multiple inheritance for a UNIVERSITY database.

Specialization/Generalization Hierarchies, Lattices & Shared Subclasses (cont.)

- 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

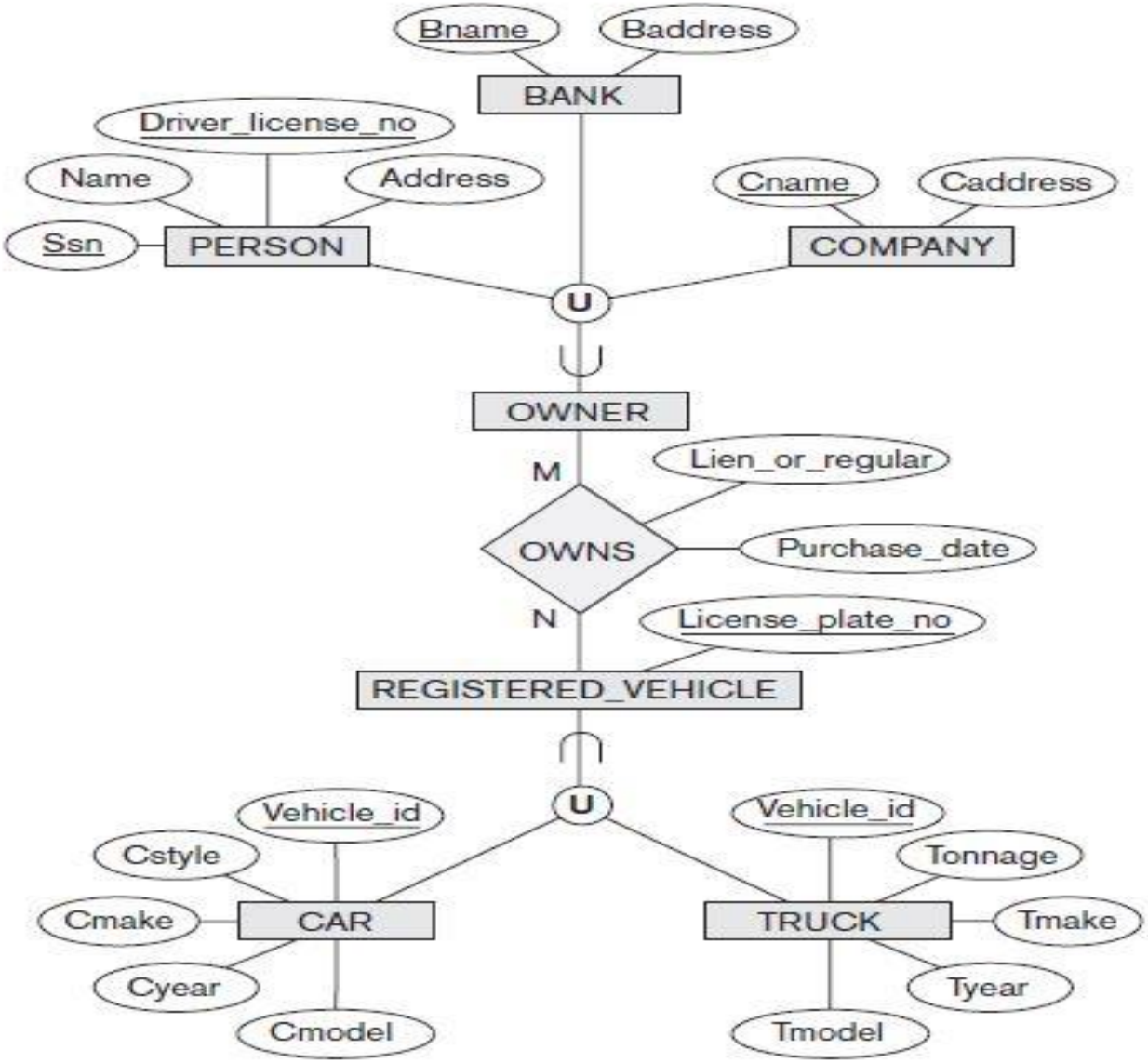
Categories (UNION TYPES)

- All of the superclass/subclass relationships we have seen thus far have a single superclass
- A shared subclass is a subclass in:
  - more than one distinct superclass/subclass relationships
  - each relationships has a single superclass
  - shared subclass leads to multiple inheritance
- In some cases, we need to model a single superclass/subclass relationship with more than one superclass
- Superclasses can represent different entity types
- Such a subclass is called a category or UNION TYPE

Categories (UNION TYPES)

- Example: In a database for vehicle registration, a vehicle owner can be a PERSON, a BANK (holding a leasing on a vehicle) or a COMPANY.
  - A category (UNION type) called OWNER is created to represent a subset of the union of the three superclasses COMPANY, BANK, and PERSON
  - A category member must exist in at least one of its superclasses
- Difference from shared subclass, which is a:
  - subset of the intersection of its superclasses
  - shared subclass member must exist in all of its superclasses

Two categories (UNION types): OWNER, REGISTERED_VEHICLE



Formal Definitions of EER Model

- Class C:
  - A set or collection of entities; this includes any of the EER schema constructs of group entities, such as entity types, subclasses, super classes, and categories.

- Subclass S is a class whose:
  - Type inherits all the attributes and relationship of a class C
  - Set of entities must always be a subset of the set of entities of the other class C
  - $S \subseteq C$
  - C is called the superclass of S
  - A superclass/subclass relationship exists between S and C
  - IS-A relationship between super and sub, denote by C/S

Formal Definitions of EER Model

- Specialization Z: Z = {S1, S2,…, Sn} is a set of subclasses with same superclass G; hence, G/Si is a superclass relationship for i = 1, …., n.
  - G is called a generalization of the subclasses {S1, S2,…, Sn}
  - Z is total if we always have:
    - S1 ∪ S2 ∪ … ∪ Sn = G;
    - Otherwise, Z is partial.
  - Z is disjoint if we always have:
    - Si ∩ Sj empty-set for i ≠ j;
  - Otherwise, Z is overlapping.

Formal Definitions of EER Model

- Subclass S of C is predicate defined if predicate (condition) p on attributes of C is used to specify membership in S;
  - that is, S = C[p], where C[p] is the set of entities in C that satisfy condition p
- A subclass not defined by a predicate is called user-defined
- Attribute-defined specialization: if a predicate A = ci (where A is an attribute of G and ci is a constant value from the domain of A) is used to specify membership in each subclass Si in Z
  - Note: If ci ≠ cj for i ≠ j, and A is single-valued, then the attribute-defined specialization will be disjoint.

Formal Definitions of EER Model

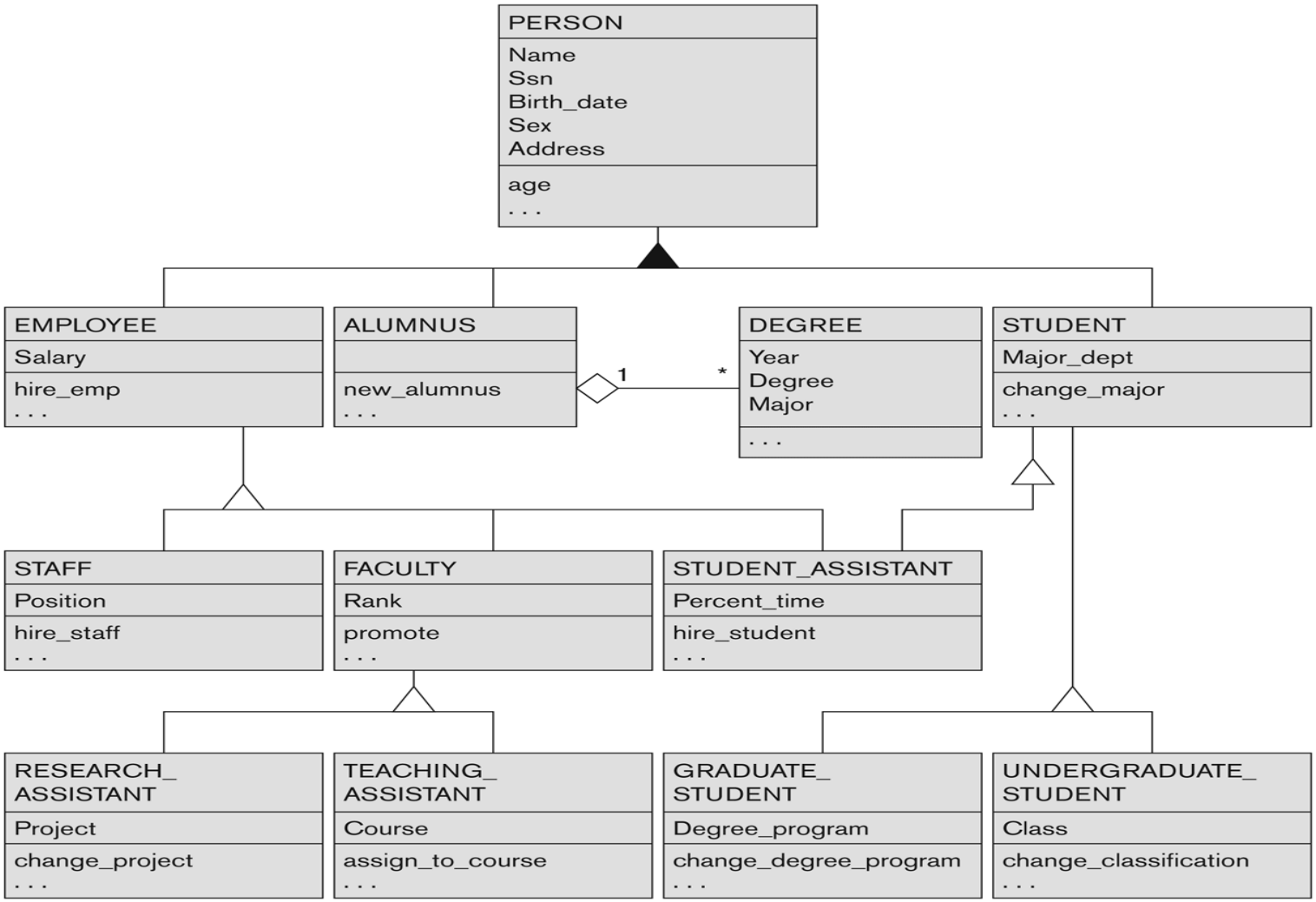
- Category or UNION type T
  - A class that is a subset of the union of n defining superclasses D1, D2,…Dn, n>1:
    - $T \subseteq (D1 \cup D2 \cup \ldots \cup Dn)$
  - Can have a predicate pi on the attributes of Di to specify entities of Di that are members of T.
  - If a predicate is specified on every Di:

    $T = (D1[p1] \cup D2[p2] \cup \ldots \cup Dn[pn])$

Alternative Diagrammatic Notations

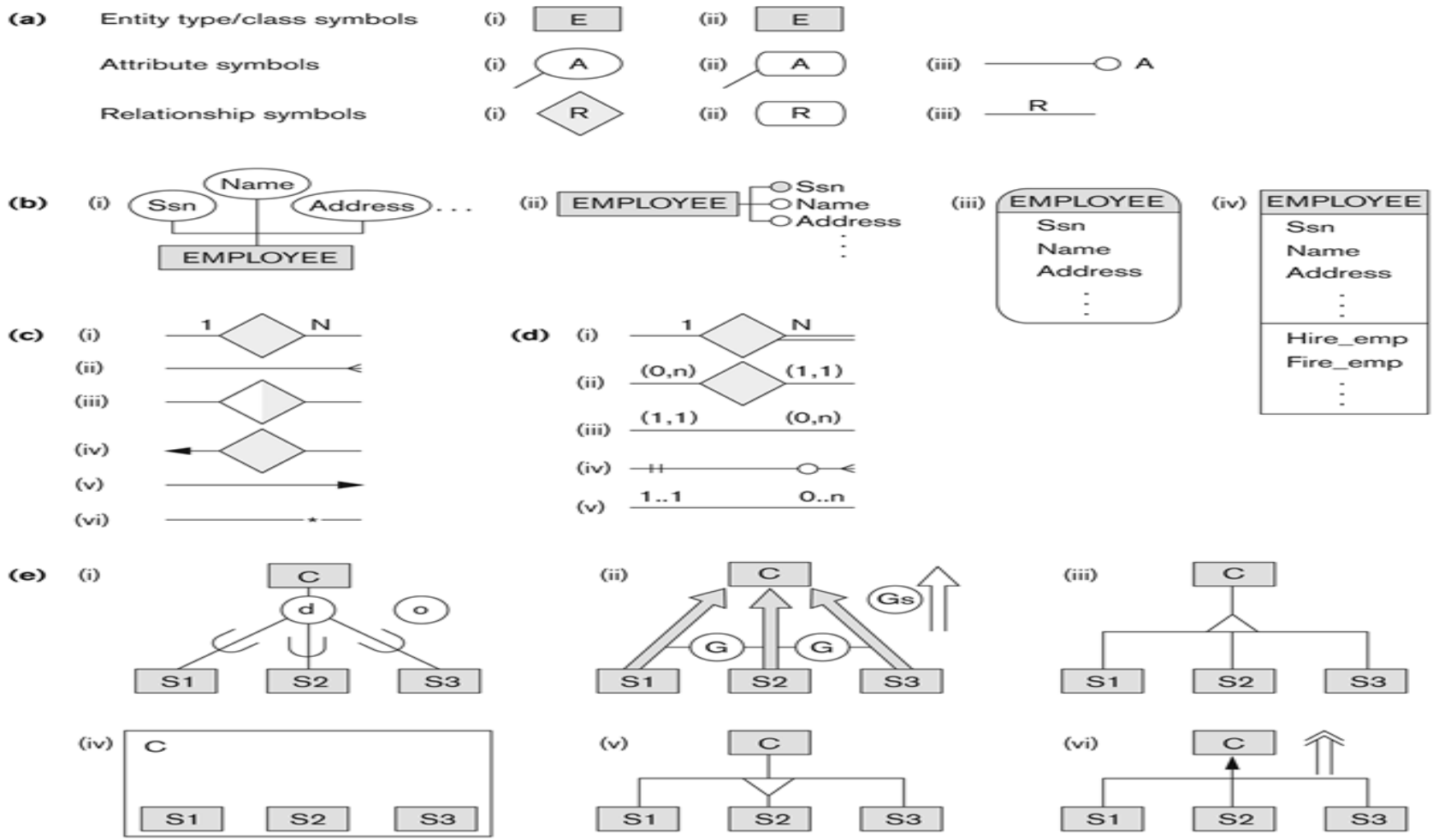
- ER/EER diagrams are a specific notation for displaying the concepts of the model diagrammatically
- DB design tools use many alternative notations for the same or similar concepts
- One popular alternative notation uses UML class diagrams
- see next slides for UML class diagrams and other alternative notations

UML Example for Displaying Specialization / Generalization



A UML class diagram corresponding to the EER diagram in Figure 4.7, illustrating UML notation for specialization/generalization.

Alternative Diagrammatic Notations



Alternative notations. (a) Symbols for entity type/class, attribute, and relationship. (b) Displaying attributes. (c) Displaying cardinality ratios. (d) Various (min, max) notations. (e) Notations for displaying specialization/generalization.

Abstraction Concepts

- Classification and Instantiation,
- Identification
- Specialization and Generalization
- Aggregation and Association

Ontologies and the Semantic Web

- It's time for self studying ☺

- Read section **8.7.5** of the reference book and create a simple note on Ontologies and Semantic Web

Homework

- Find about Keys in a Relation
  - List down and briefly describe them

Reference

- *Chapter 8 - Fundamentals of Database Systems*

(6th Edition) By Remez Elmasri & Shamkant B. Navathe

Questions ???

Thank You