GLA UNIVERSITY MATHURA
Recognised by UGC Under Section 2(f)
Accredited with A Grade by NAAC

APPLIED DATABASE MANAGEMENT SYSTEM BCSC0014

Dr. Neeraj Gupta
Assistant Professor, Dept. of CEA
neeraj.gupta@gla.ac.in

OUTLINE

- ❑The Enhanced Entity-Relationship (EER) Model
- ❑Specialization
- ❑Generalization

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

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

SUBCLASSES, SUPERCLASSES, AND INHERITANCE

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

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

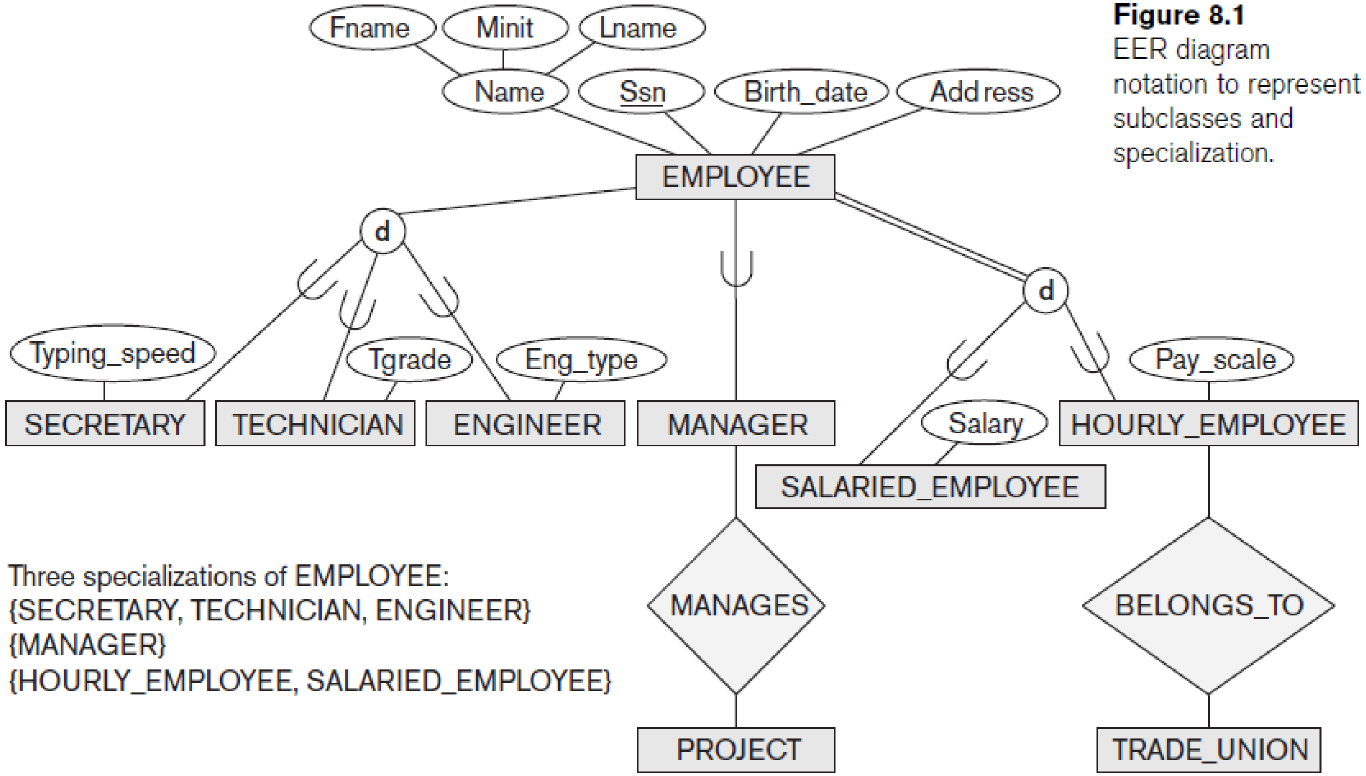


Figure 8.1
EER diagram notation to represent subclasses and specialization.

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

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

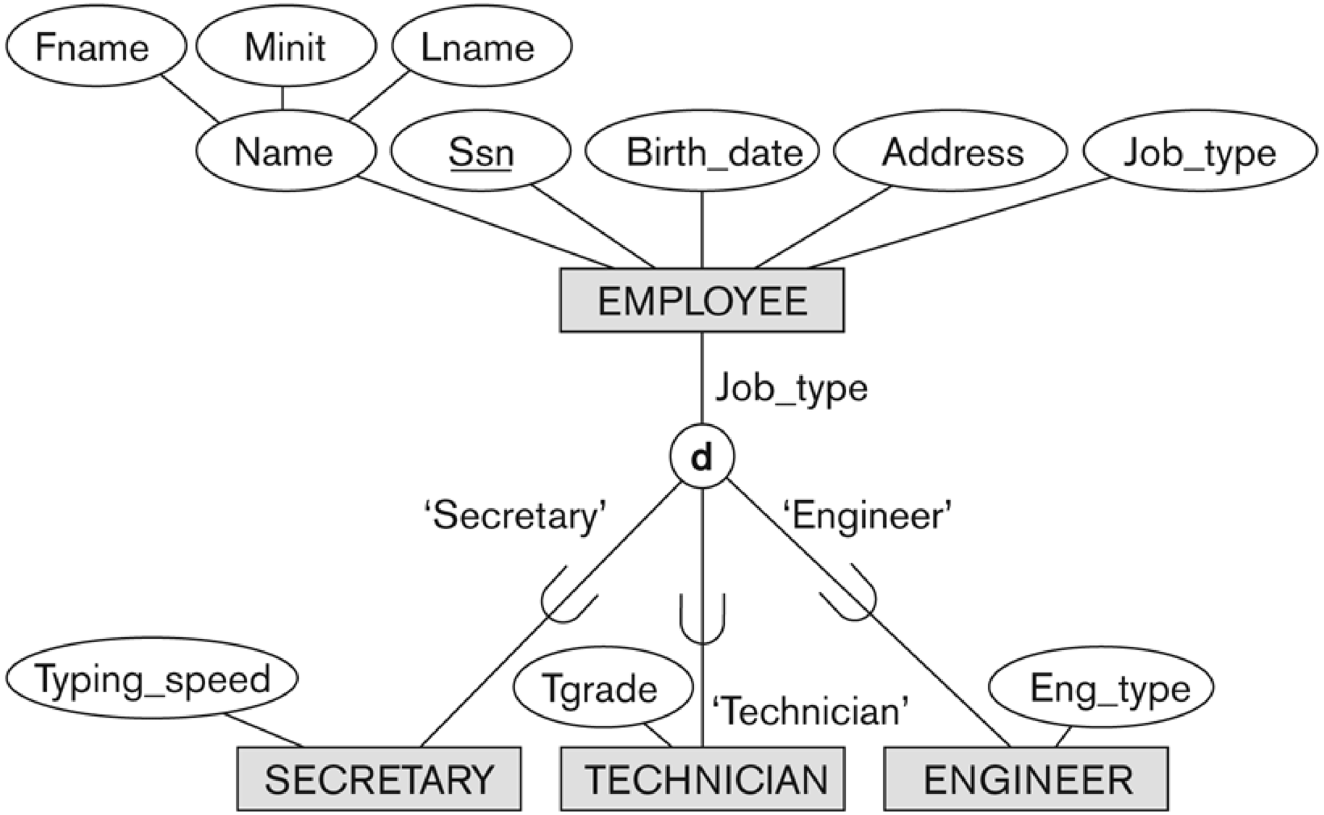
Examples:

- 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

REPRESENTING SPECIALIZATION IN EER DIAGRAMS

Figure 4.4
EER diagram notation for an attribute-defined specialization on Job_type.



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

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

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

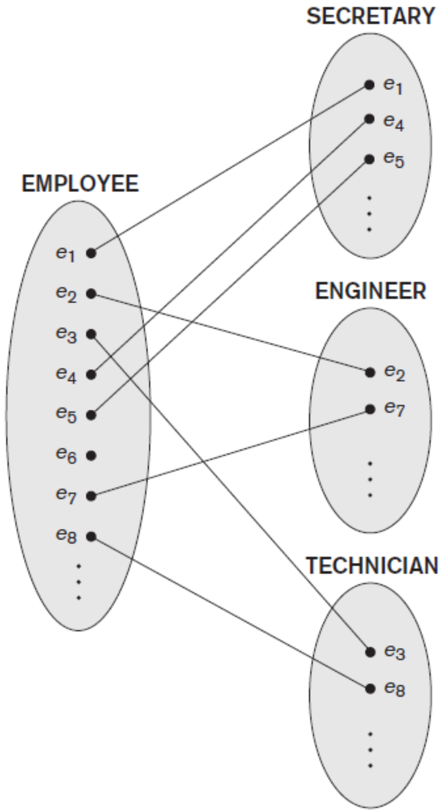


Figure 8.2
Instances of a specialization.

SPECIALIZATION AND GENERALIZATION

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 ON SPECIALIZATION AND GENERALIZATION

Two basic constraints can apply to a specialization/generalization:

- Disjointness Constraint:
- Completeness Constraint:

CONSTRAINTS ON SPECIALIZATION AND GENERALIZATION

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

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 ***<u>double line</u>***
- *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

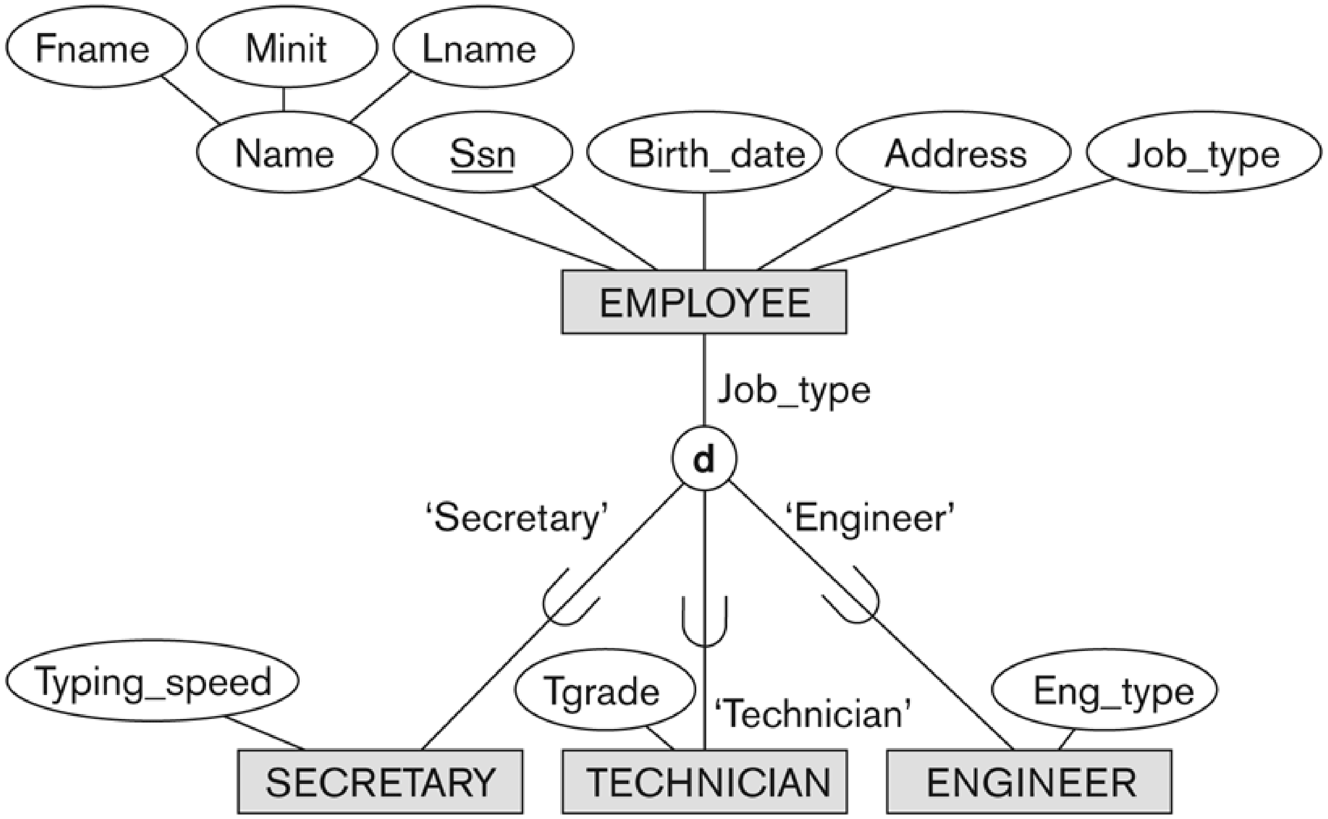
Hence, we have four types of specialization/generalization:

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

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

EXAMPLE OF DISJOINT PARTIAL SPECIALIZATION

Figure 4.4
EER diagram notation for an attribute-defined specialization on Job_type.



EXAMPLE OF OVERLAPPING TOTAL SPECIALIZATION

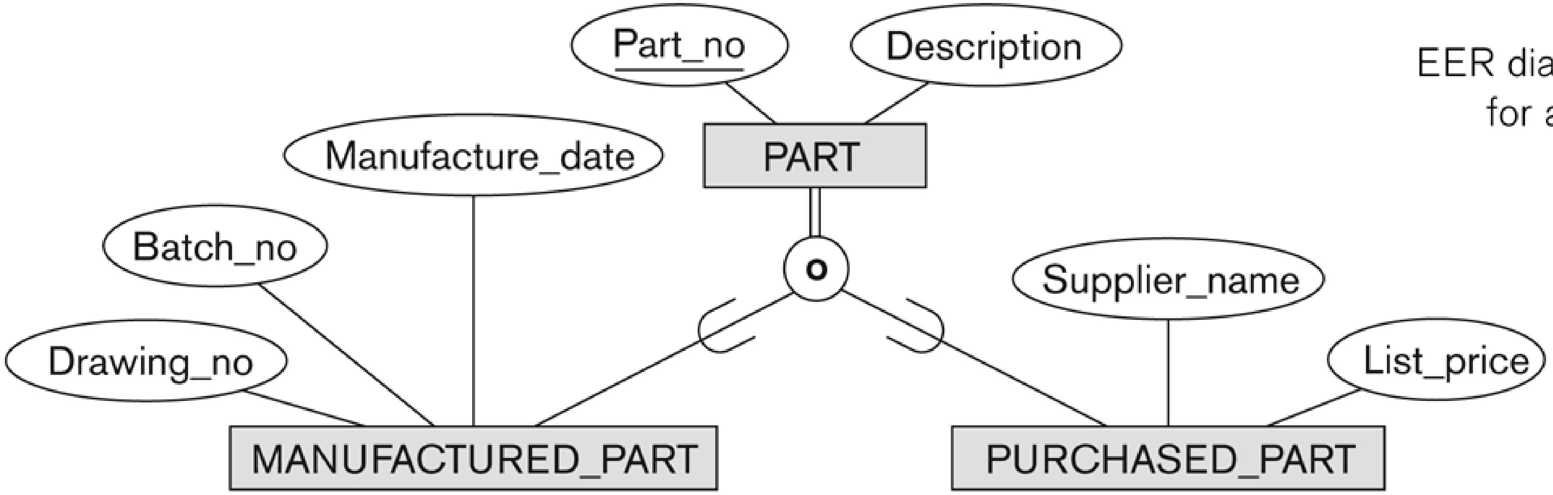


Figure 4.5
EER diagram notation for an overlapping (nondisjoint) specialization.

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***)

Shared Subclass “Engineering_Manager”

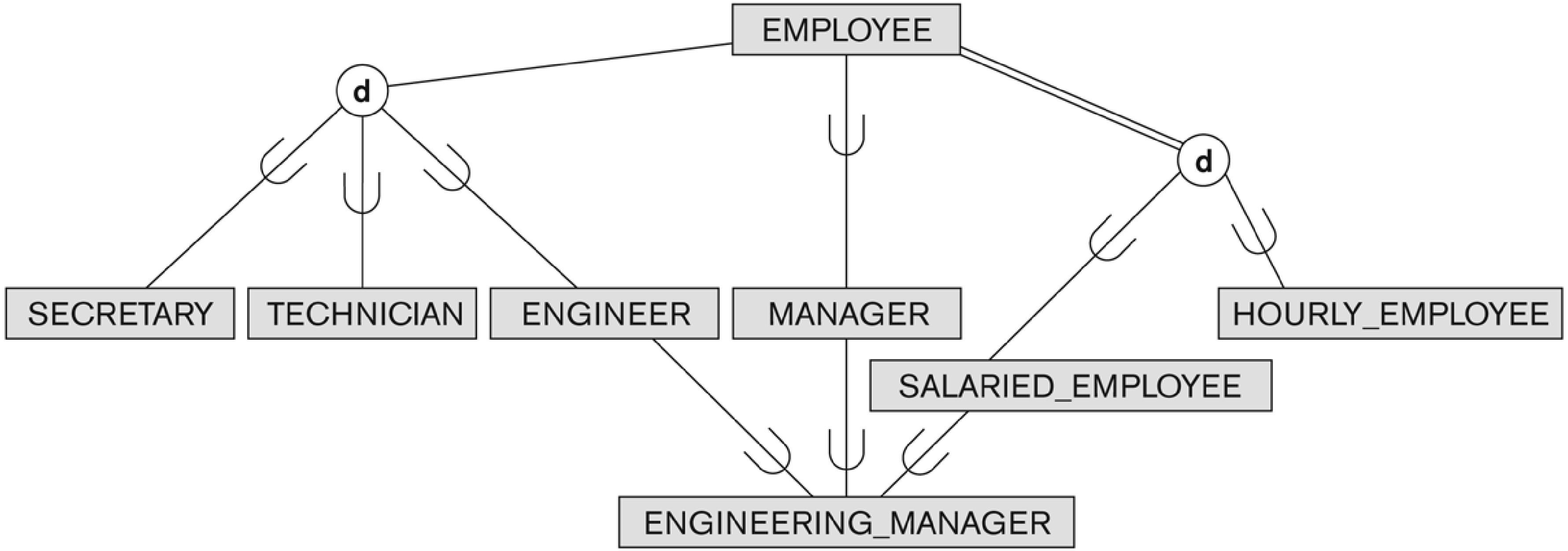


Figure 4.6
A specialization lattice with shared subclass ENGINEERING_MANAGER.

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)

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)

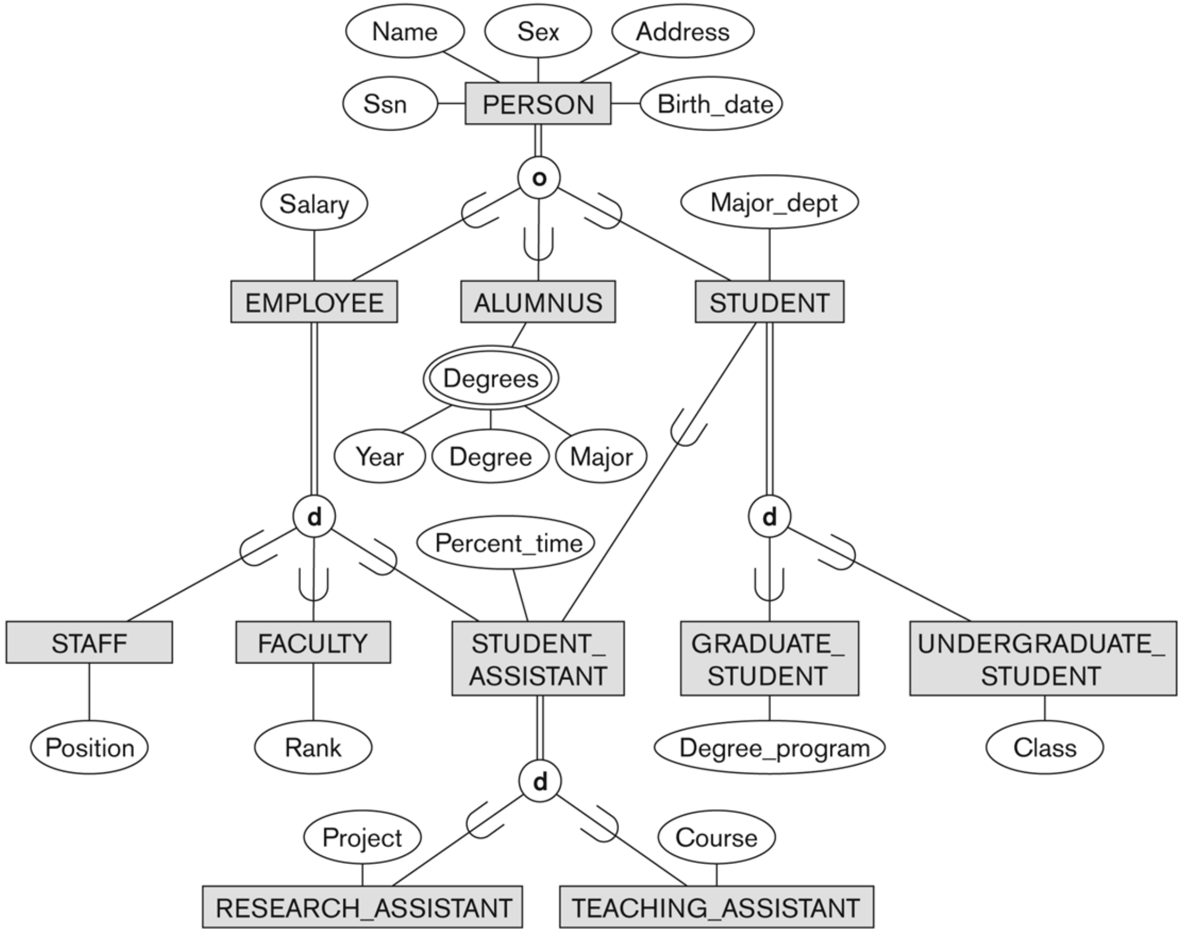
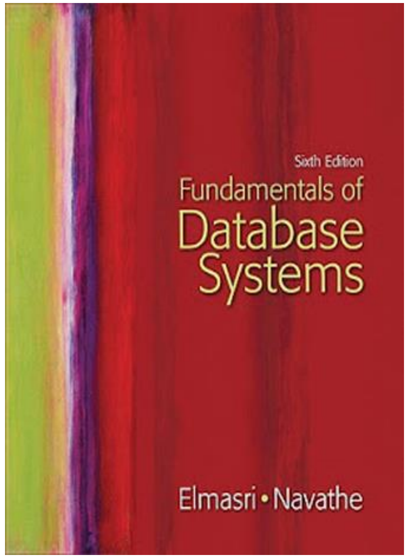
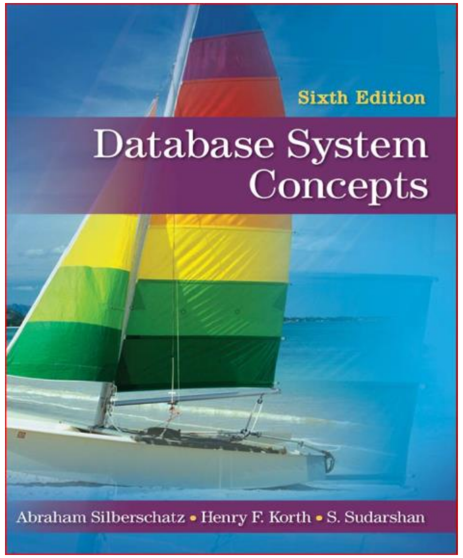


Figure 4.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.

REFERENCE BOOKS



Sixth Edition
Fundamentals of Database Systems
Elmasri • Navathe



Sixth Edition
Database System Concepts
Abraham Silberschatz • Henry F. Korth • S. Sudarshan

Keep Learning
Keep Growing