Chapter 4: The Enhanced Entity-Relationship (EER) Model

CS-6360 Database Design

Chris Irwin Davis, Ph.D.

Email: cid021000@utdallas.edu
Phone: (972) 883-3574
Office: ECSS 4.705

Chapter 4 Outline

- 4.1 – Subtypes, Supertypes, and Inheritance
- 4.2 – Specialization and Generalization
- 4.3 – Constraints and Characteristics of Specialization and Generalization Hierarchies
- 4.4 – Modeling of UNION Types Using Categories

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 than the ER Model alone
  - 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**

8.1 Subclasses, Superclasses, and Inheritance

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

- **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
  - **Supertype/subtype**
  - **Superclass/subclass**
  - **Class/subclass** relationship
- **Type inheritance**
  - Subclass entity inherits all attributes and relationships of superclass

Subclass and Specialization

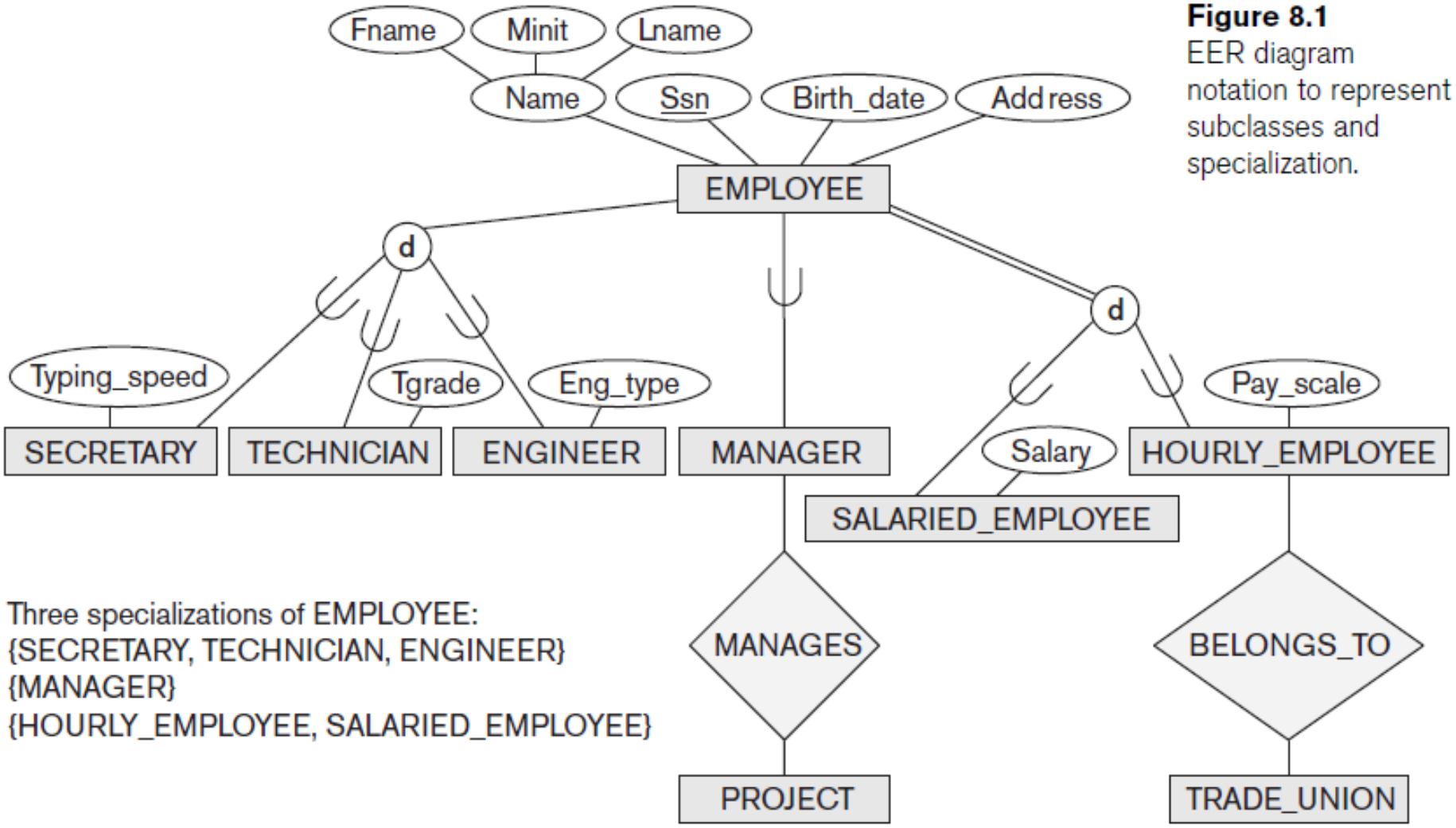


Figure 8.1
EER diagram notation to represent subclasses and specialization.

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

8.2 Specialization and Generalization

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**

Entity Subsets

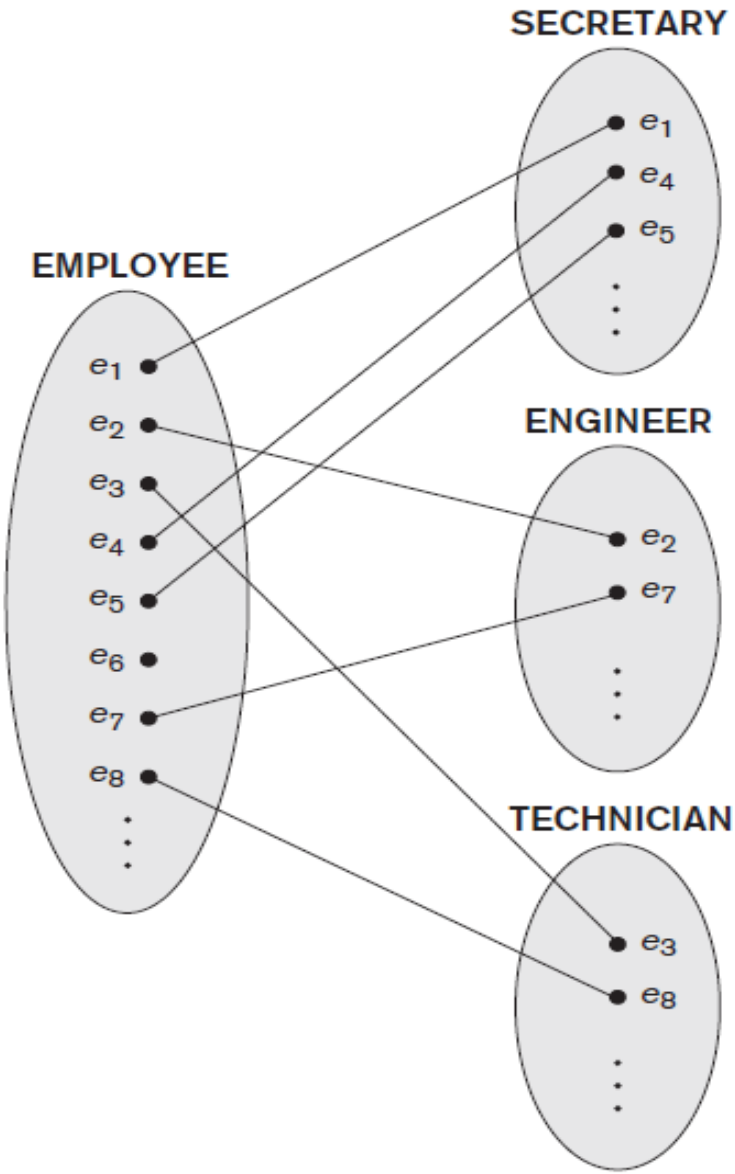
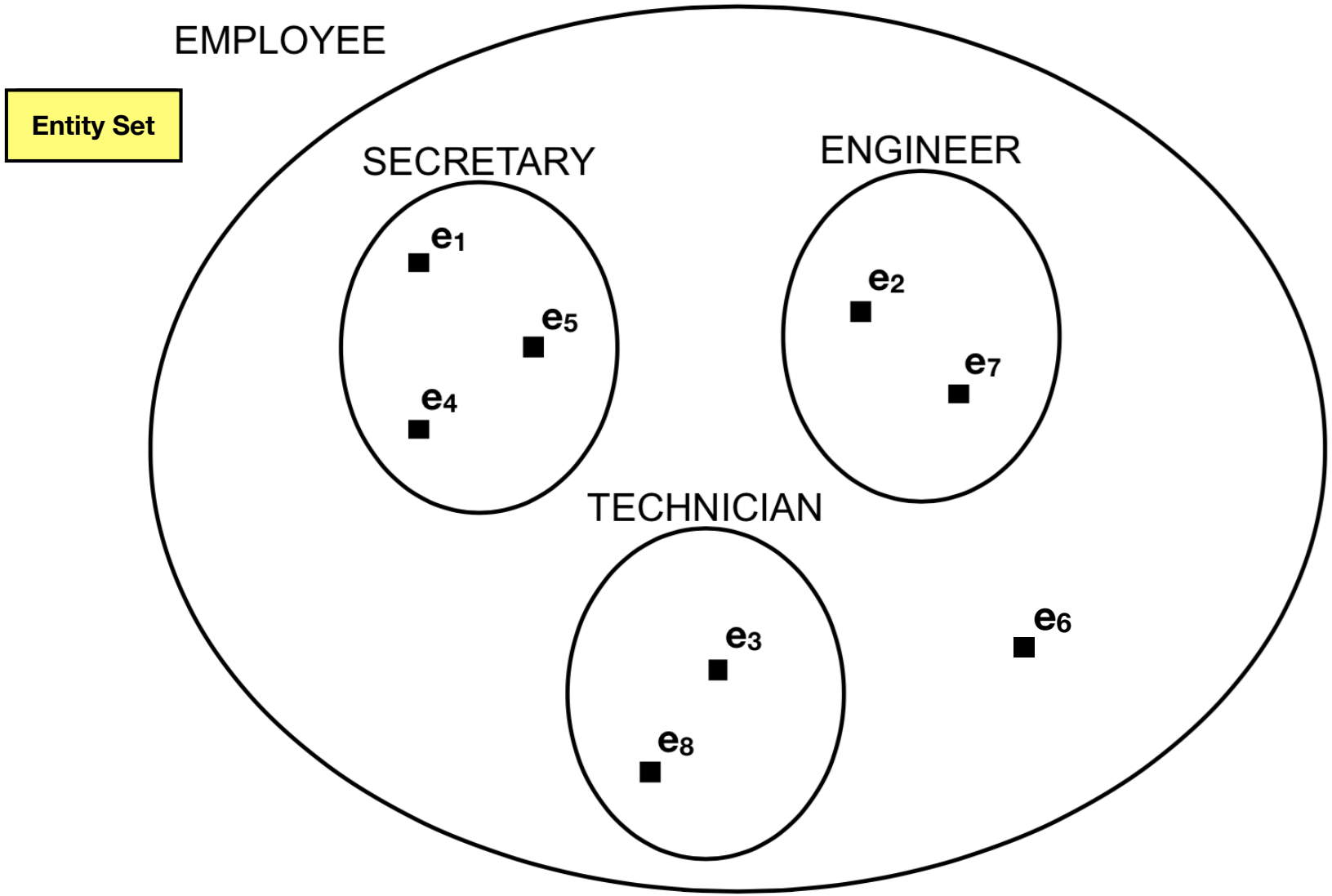


Figure 8.2
Instances of a specialization.

Entity Subsets



Specialization and Generalization (cont'd.)

- Two main 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 members of the subclass

Generalization

- Reverse process of specialization, which is 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

Generalization Example

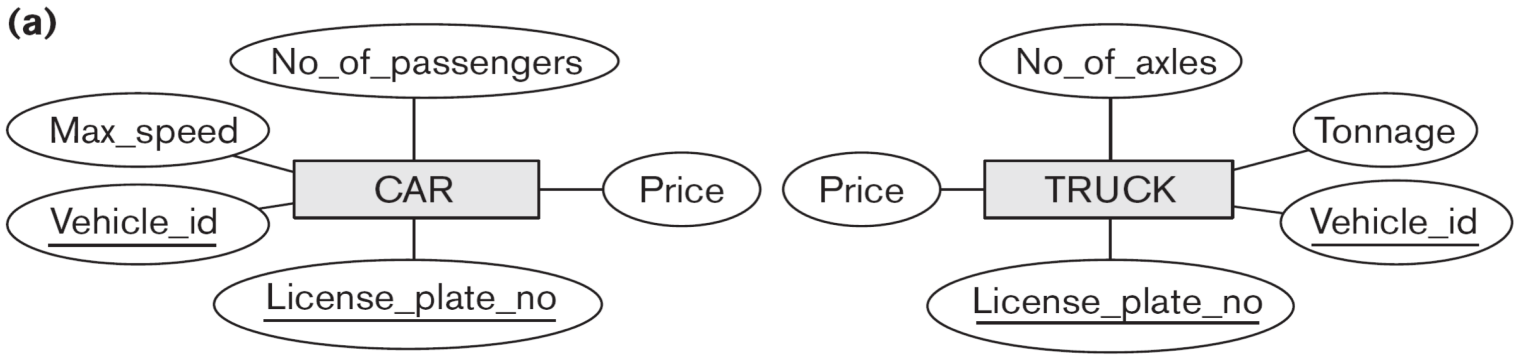


Figure 8.3
Generalization. (a) Two entity types, CAR and TRUCK. (b) Generalizing CAR and TRUCK into the superclass VEHICLE.

Generalization Example

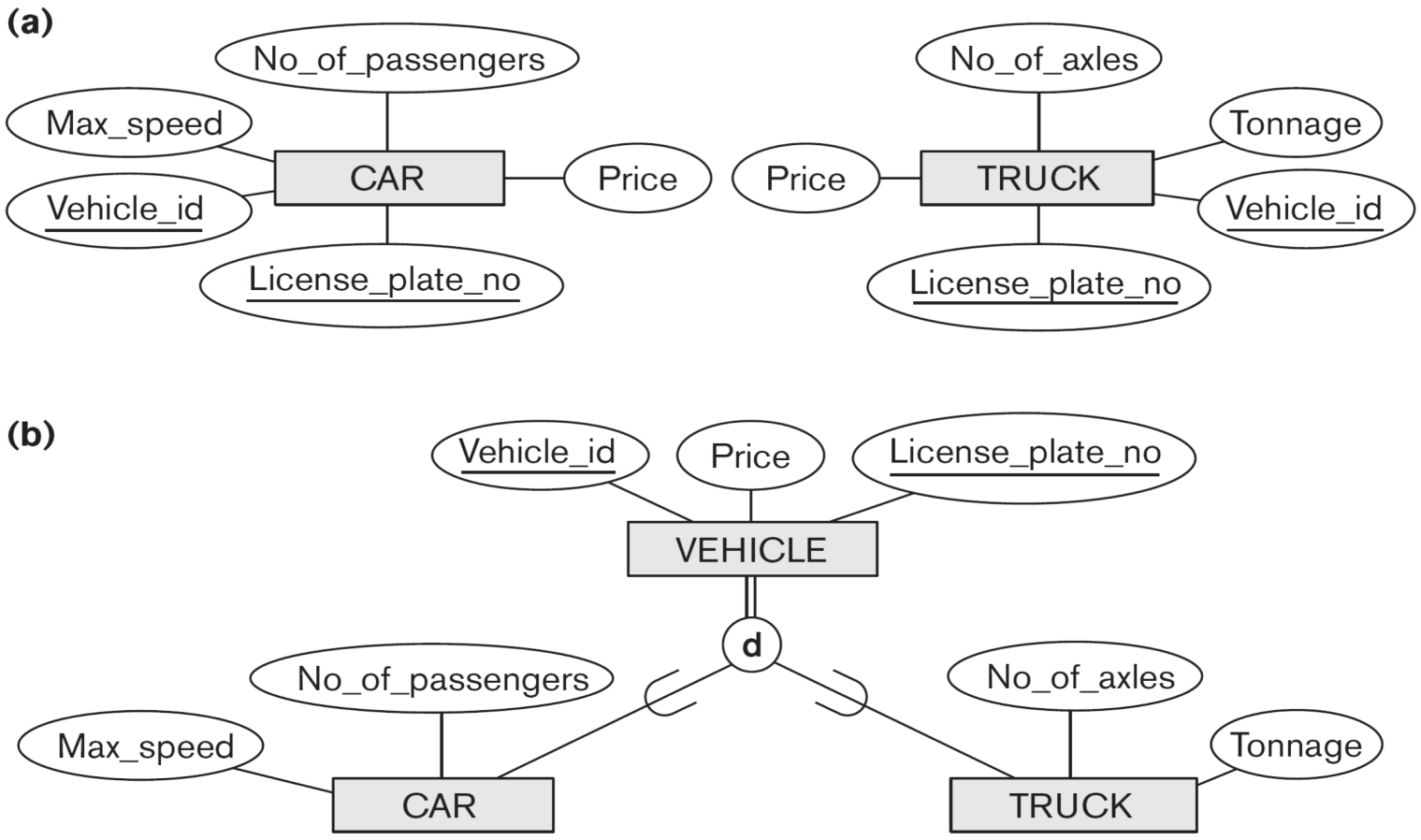


Figure 8.3
Generalization. (a) Two entity types, CAR and TRUCK. (b) Generalizing CAR and TRUCK into the superclass VEHICLE.

4.3 Constraints and Characteristics of Specialization and Generalization Hierarchies

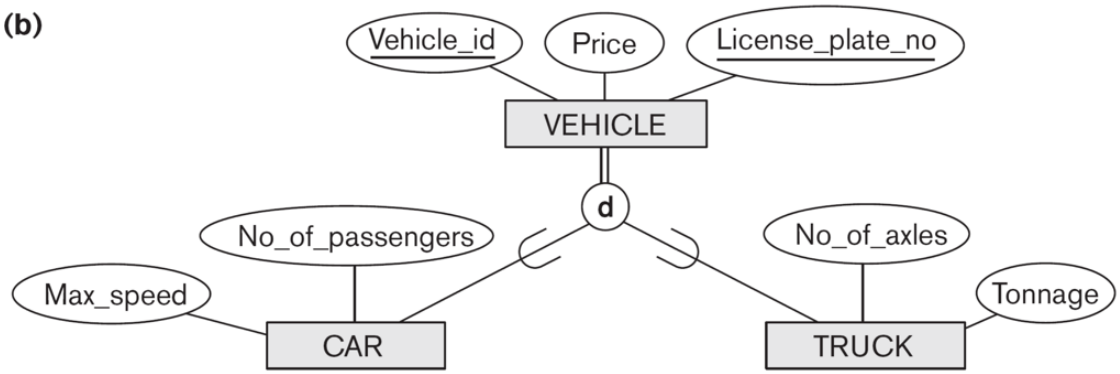
Constraints and Characteristics of Specialization and Generalization Hierarchies

- Constraints that apply to a single **specialization** or a single **generalization**
  - How to deal with?
- Differences between specialization/generalization **lattices** and **hierarchies**
- **Continued…**

Constraints on Specialization and Generalization (cont'd.)

- **Disjointness constraint**
  - Specifies that the subclasses of the specialization must be disjoint

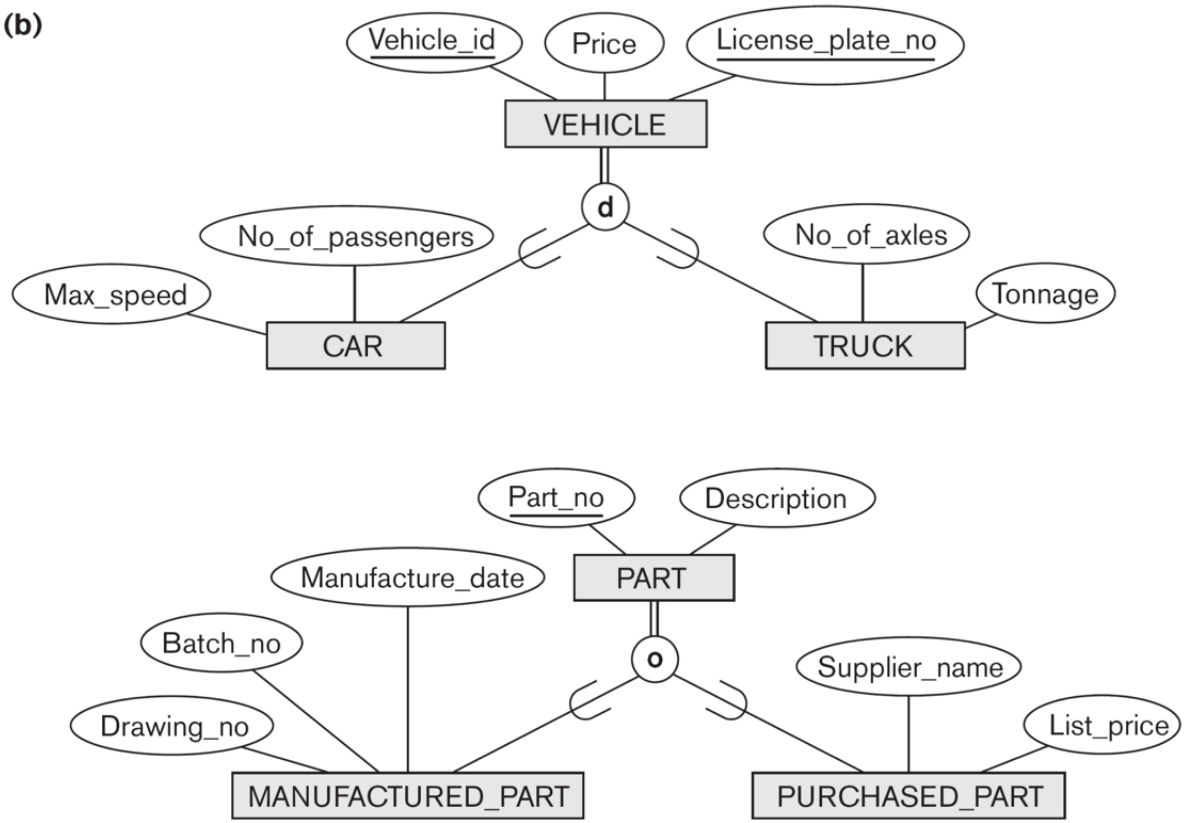
(b)



Constraints on Specialization and Generalization (cont'd.)

- **Disjointness constraint**
  - Specifies that the subclasses of the specialization must be disjoint

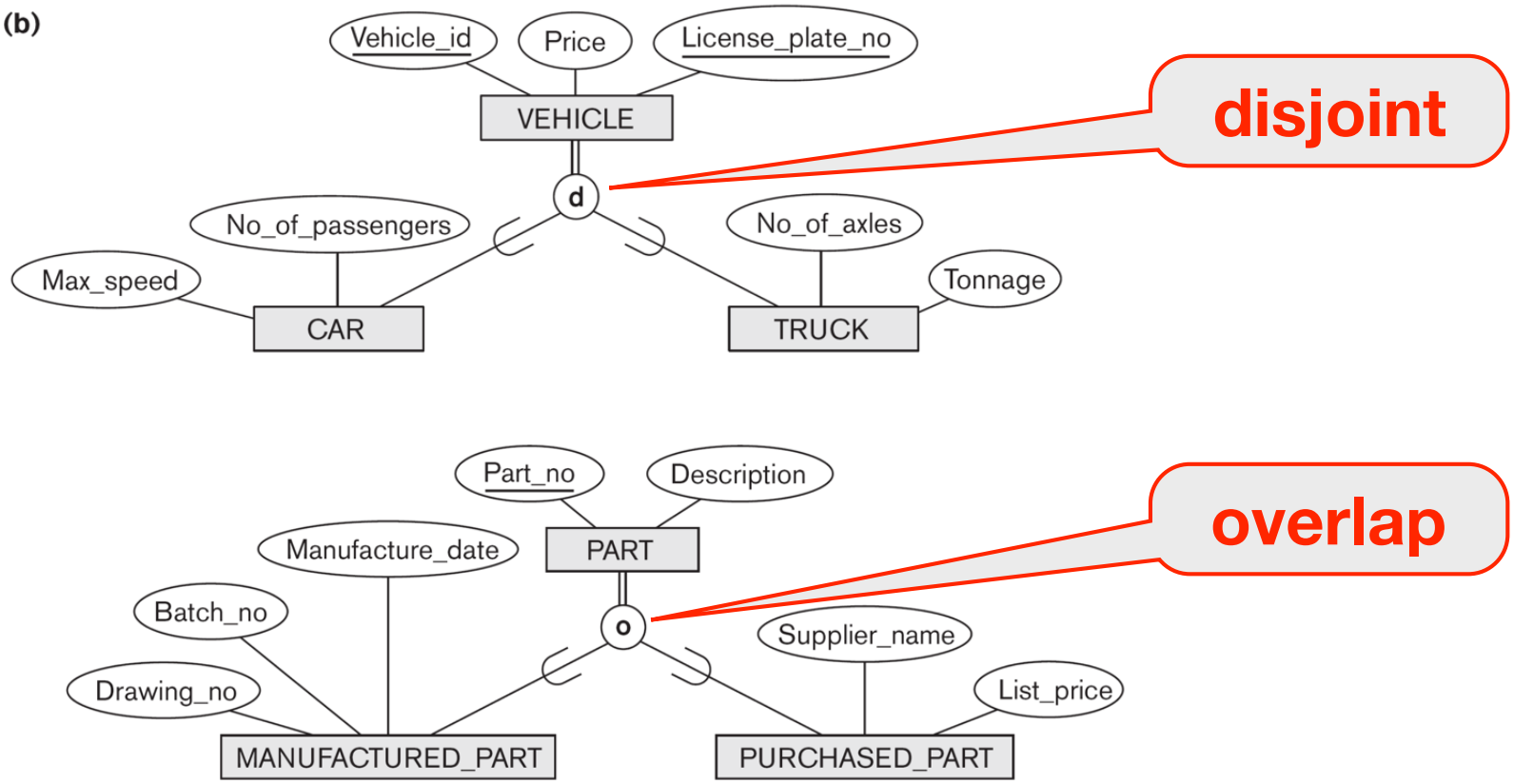
(b)



Constraints on Specialization and Generalization (cont'd.)

- **Disjointness constraint**
  - Specifies that the subclasses of the specialization must be disjoint

(b)



Constraints on Specialization and Generalization (cont'd.)

- **Completeness**, or **totalness**, **constraint** (for subtype)
  - May be **total** or **partial**
- Disjointness and completeness constraints are independent

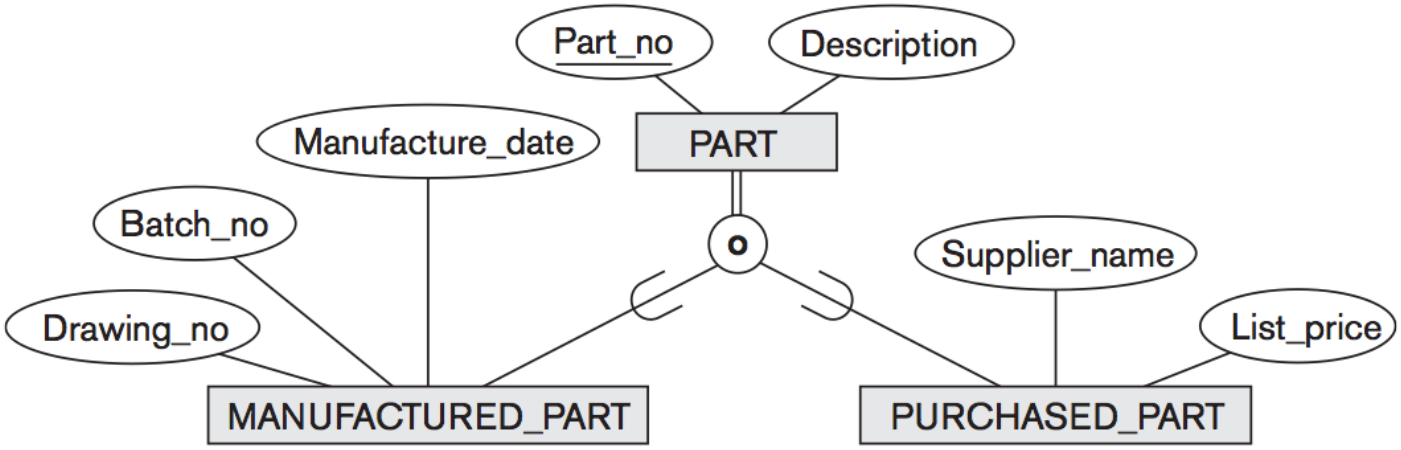


Figure 8.5
EER diagram notation for an overlapping (nondisjoint) specialization.

[7]The notation of using single or double lines is similar to that for partial or total participation of an entity type in a relationship type, as described in Chapter 7.

Constraints on Specialization and Generalization (cont'd.)

- **Completeness** (or **totalness**) **constraint**
  - May be **total** or **partial**
- Disjointness and completeness constraints are independent

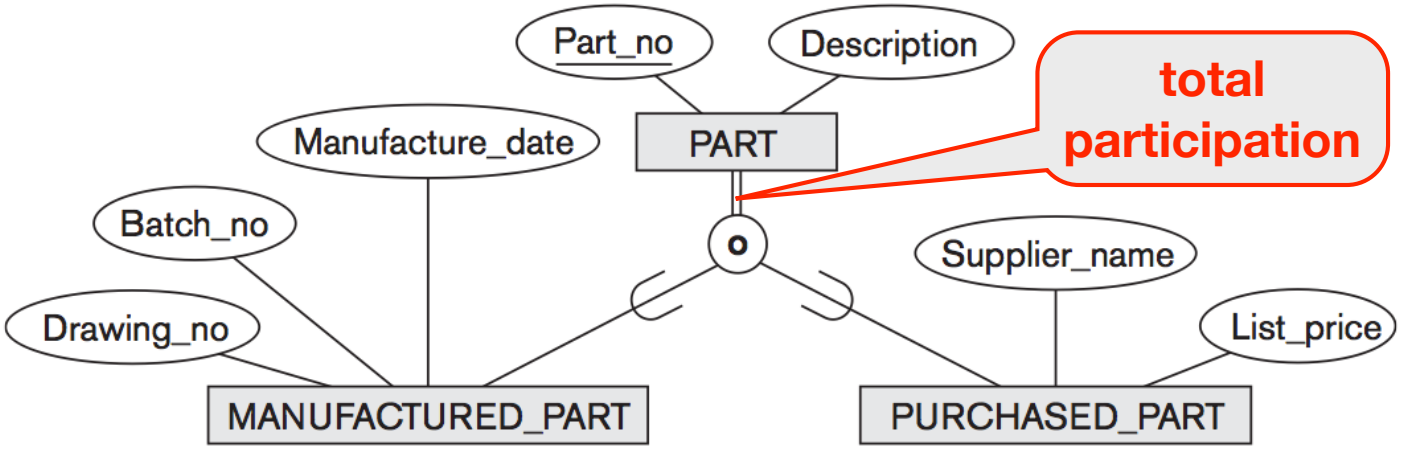


Figure 8.5
EER diagram notation for an overlapping (nondisjoint) specialization.

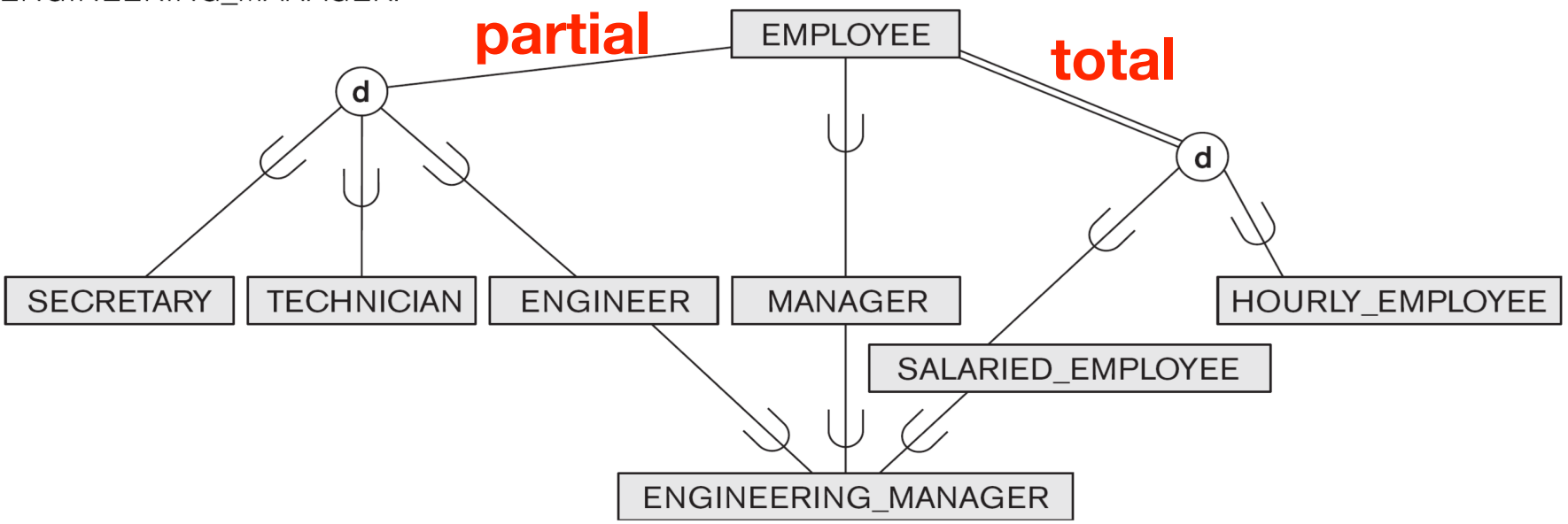
[7]The notation of using single or double lines is similar to that for partial or total participation of an entity type in a relationship type, as described in Chapter 7.

Specialization and Generalization Hierarchies and Lattices

- **Specialization hierarchy**
  - Every subclass participates as a subclass in only one class/subclass relationship
  - Results in a **tree structure** or **strict hierarchy**
- **Specialization lattice**
  - Subclass can be a subclass in more than one class/subclass relationship (i.e. **multiple inheritance**)

Specialization and Generalization Hierarchies and Lattices

Figure 8.6
A specialization lattice with shared subclass ENGINEERING_MANAGER.



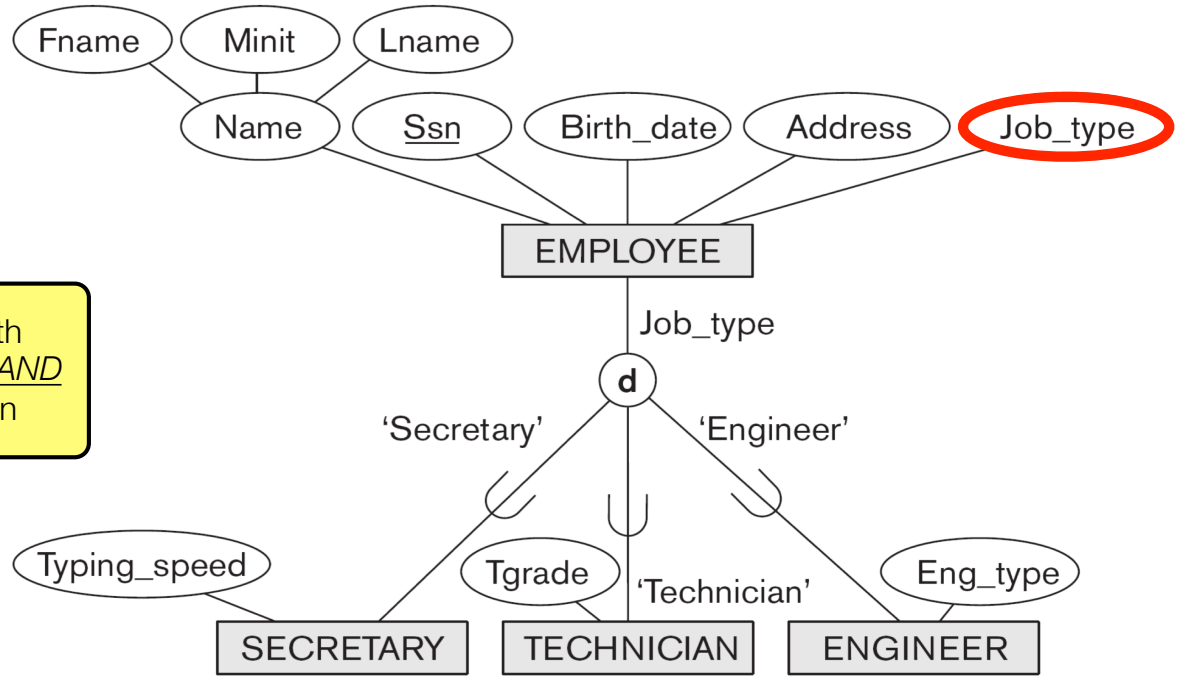
Constraints on Specialization and Generalization

- May be several or one subclass
- Determine entity subtype:
  - **Predicate-defined** (or **condition-defined**) **subclasses**
  - **Attribute-defined specialization**
  - **User-defined**

Constraints on Specialization and Generalization

- May be several or one subclass
- How to determine entity subtype
  - **Predicate-defined** (or **condition-defined**) **subclasses**

Figure 8.4
EER diagram notation for an attribute-defined specialization on Job_type.



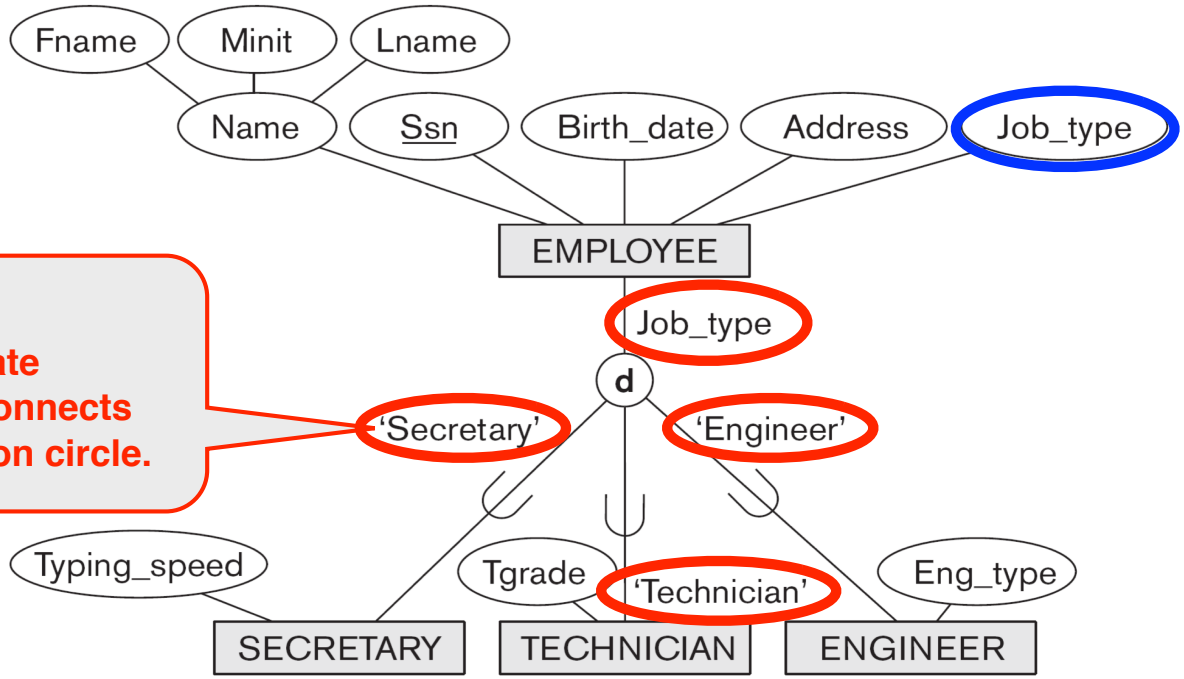
This diagram actually shows both **predicate-defined** specialization *AND* **attribute-defined** specialization

[6]Such an attribute is called a *discriminator* in UML terminology.

Constraints on Specialization and Generalization

- May be several or one subclass
- Determine entity subtype:
  - **Predicate-defined** (or **condition-defined**) **subclasses**

Figure 8.4
EER diagram notation for an attribute-defined specialization on Job_type.



Predicate-defined subclass are displayed by writing the predicate condition next to the line that connects the subclass to the specialization circle.

[6]Such an attribute is called a *discriminator* in UML terminology.

Constraints on Specialization and Generalization

- May be several or one subclass
- Determine entity subtype:
  - **Predicate-defined** (or **condition-defined**) **subclasses**
  - **Attribute-defined specialization**

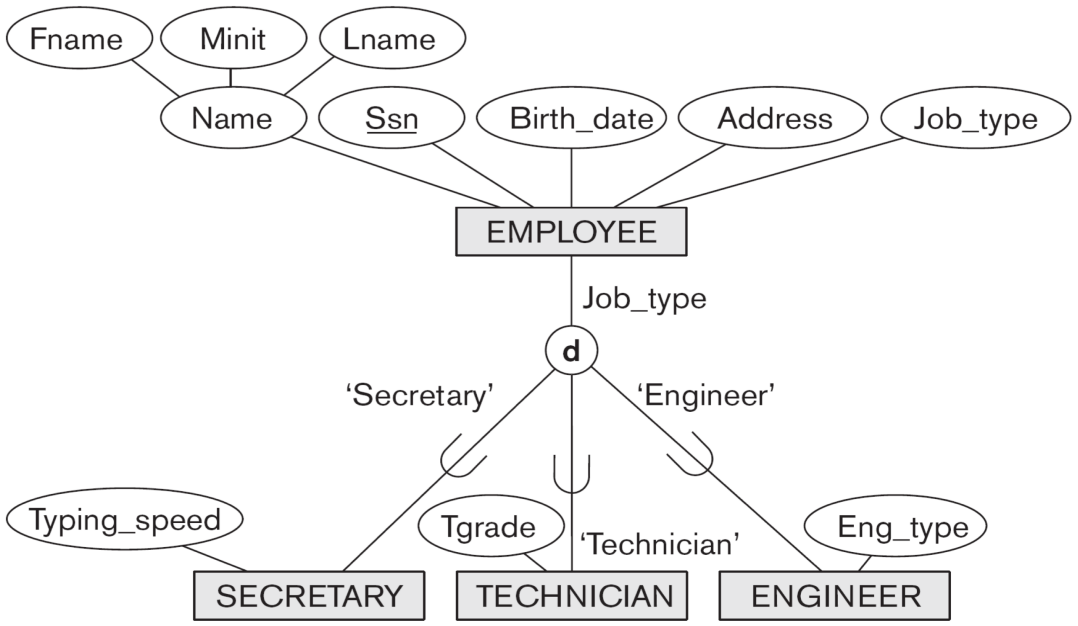


Figure 8.4
EER diagram notation for an attribute-defined specialization on Job_type.

[6]Such an attribute is called a *discriminator* in UML terminology.

Constraints on Specialization and Generalization

- May be several or one subclass
- Determine entity subtype:
  - **Predicate-defined** (or **condition-defined**) **subclasses**
  - **Attribute-defined specialization**

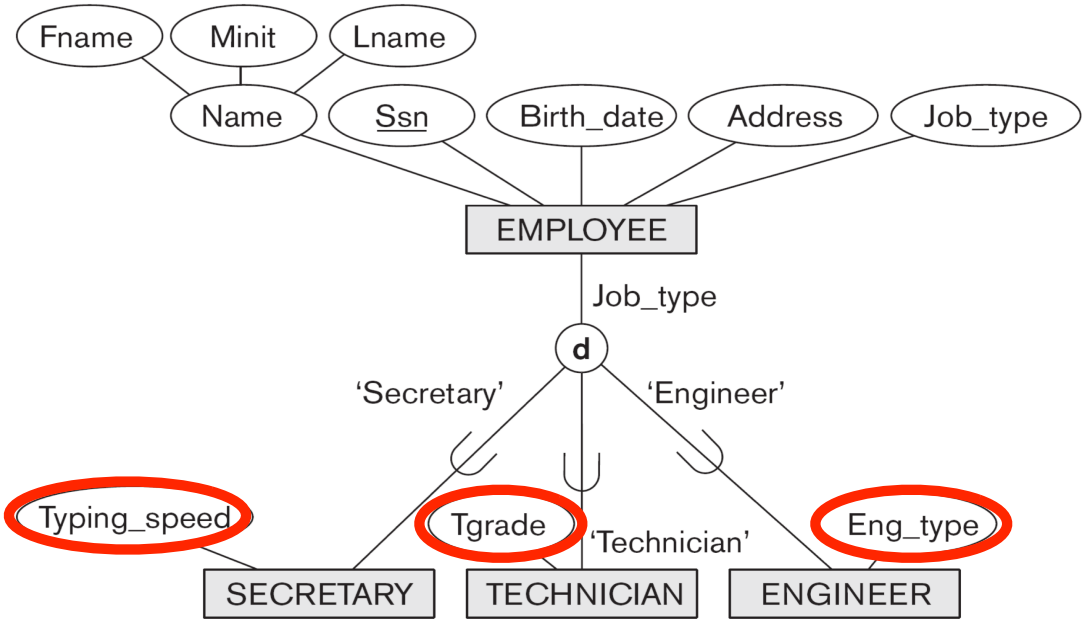


Figure 8.4
EER diagram notation for an attribute-defined specialization on Job_type.

[6]Such an attribute is called a *discriminator* in UML terminology.

Specialization Lattice w/ Multiple Inheritance

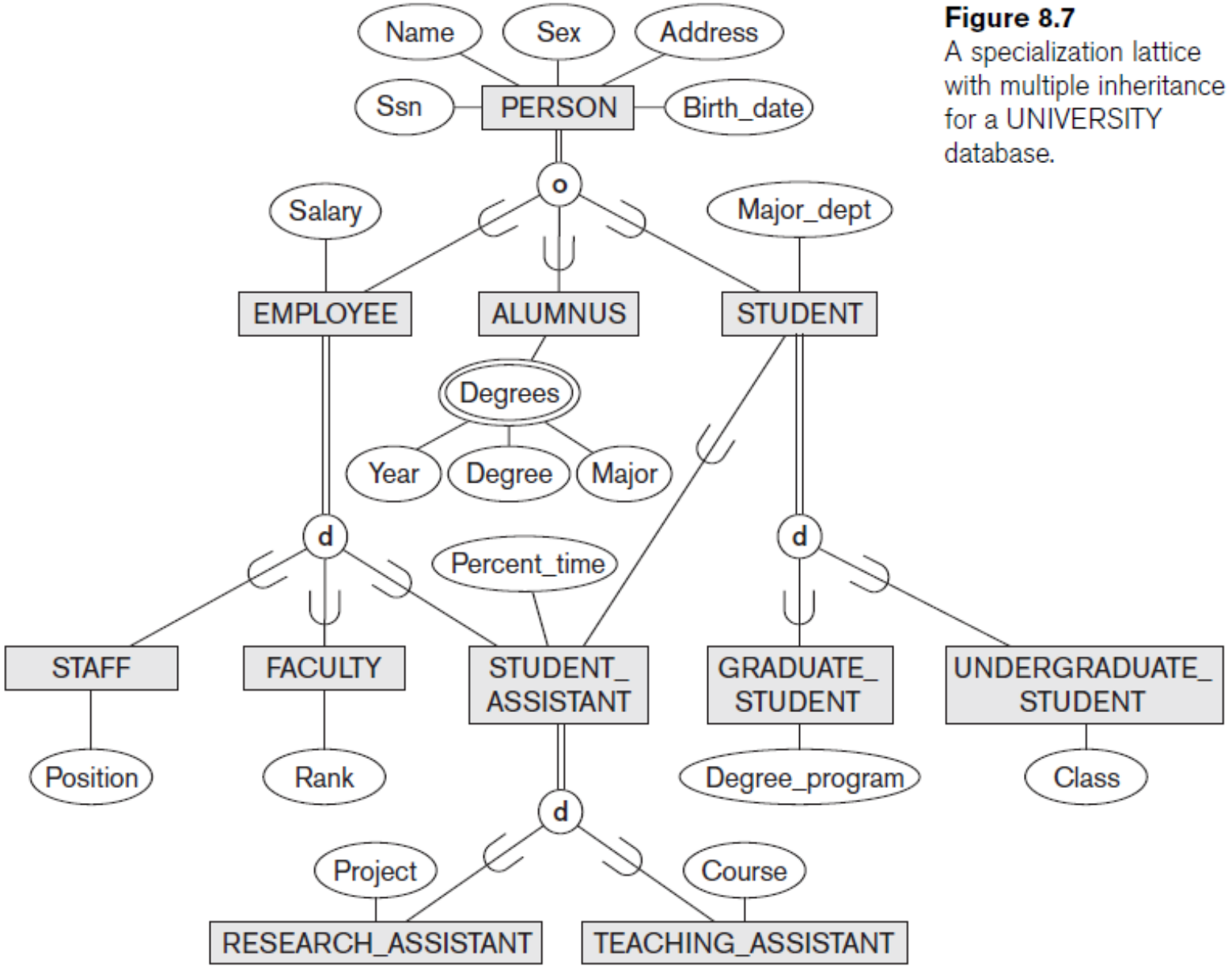


Figure 8.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.

Specialization and Generalization Hierarchies and Lattices (cont'd.)

- **Multiple inheritance**
  - Subclass with more than one superclass
  - If attribute (or relationship) originating in the same superclass inherited more than once via different paths in lattice
    - Included only once in shared subclass
- **Single inheritance**
  - Some models and languages limited to single inheritance

Specialization Lattice w/ Multiple Inheritance

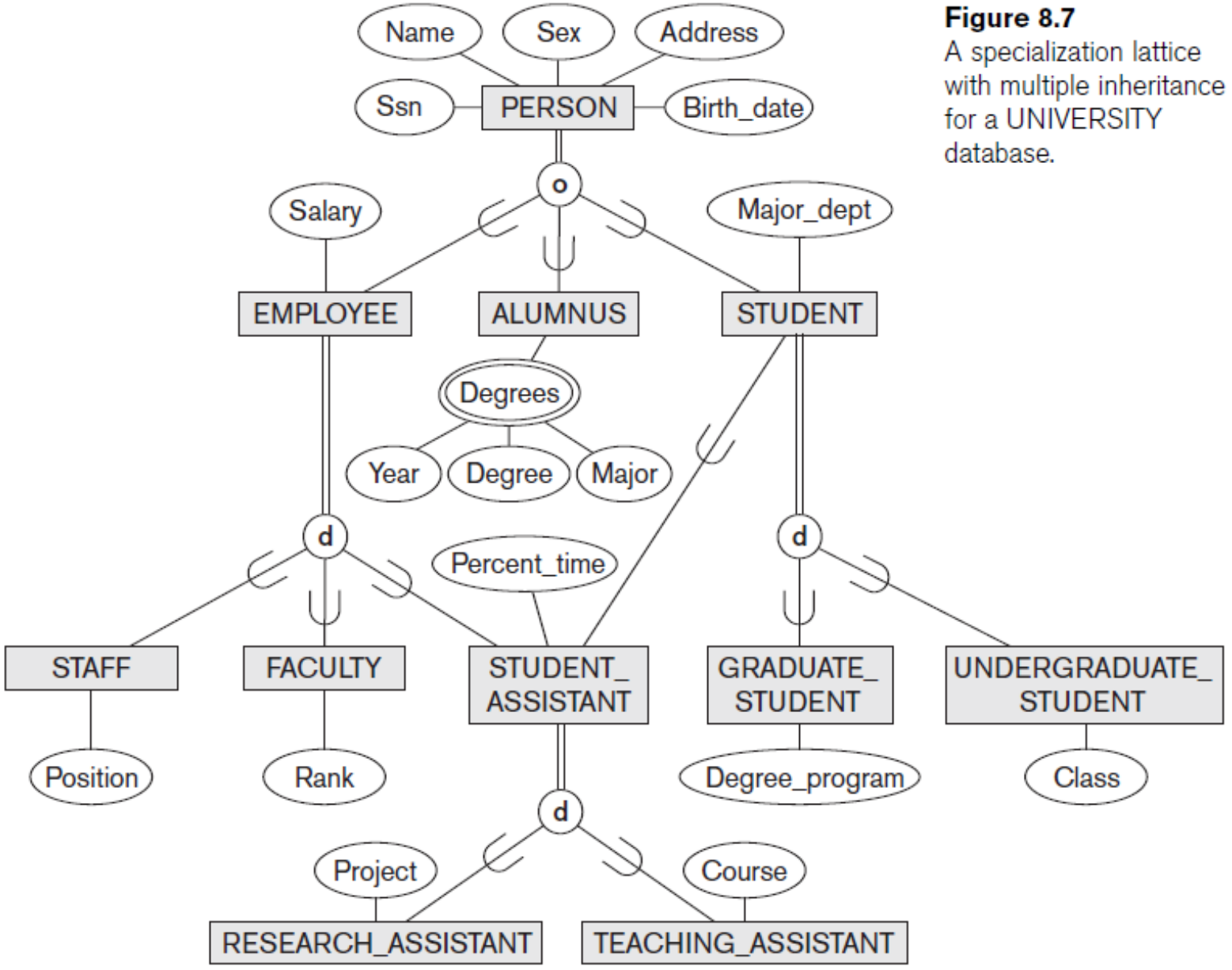


Figure 8.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.

Utilizing Specialization and Generalization in Refining Conceptual Schemas

- **Specialization process**
  - **Top-down** conceptual refinement process
  - Start with entity type then define subclasses by successive specialization
- **Generalization process**
  - **Bottom-up** conceptual synthesis
  - Involves generalization rather than specialization

Modeling of UNION Types Using Categories

- **Union type** or a **category**
  - Represents a single superclass/subclass relationship with more than one superclass (disjoint, not overlap)
  - Subclass represents a collection of objects that is a subset of the UNION of distinct entity types
  - Attribute inheritance works more selectively
  - Category can be **total** or **partial**
- Some modeling methodologies do not have union types

Modeling of UNION Types Using Categories

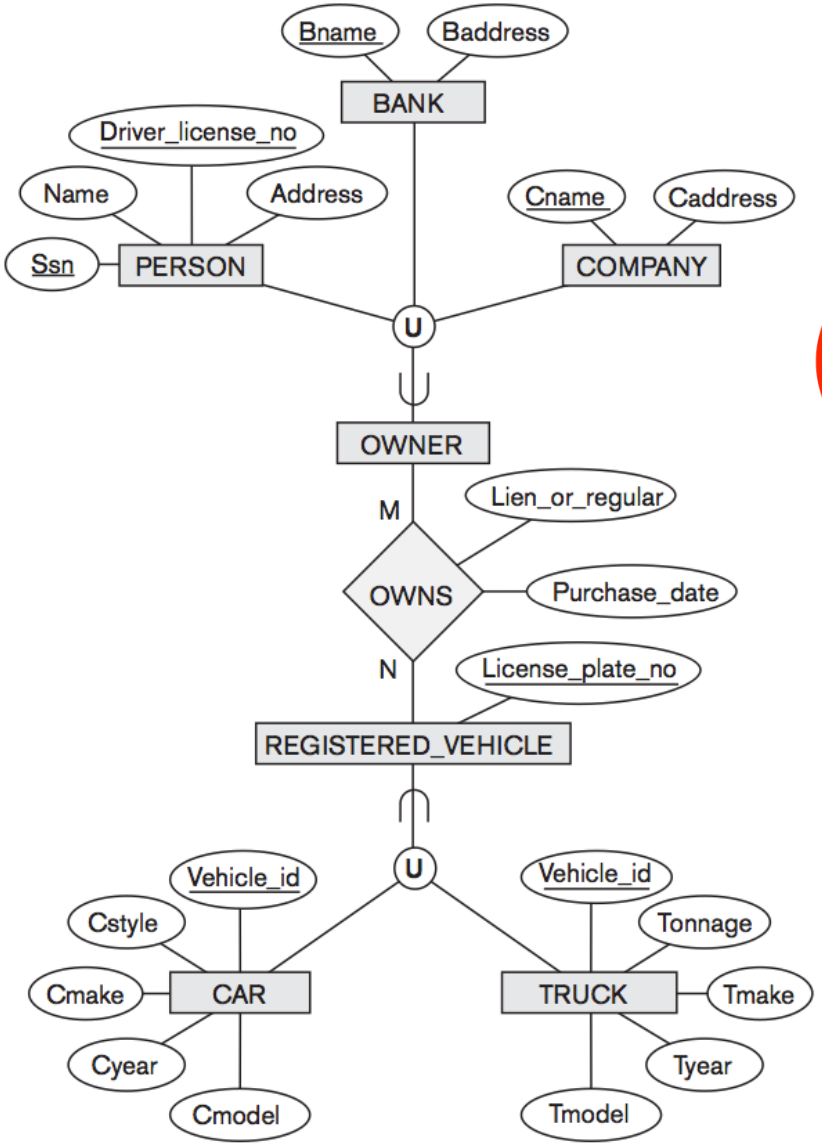
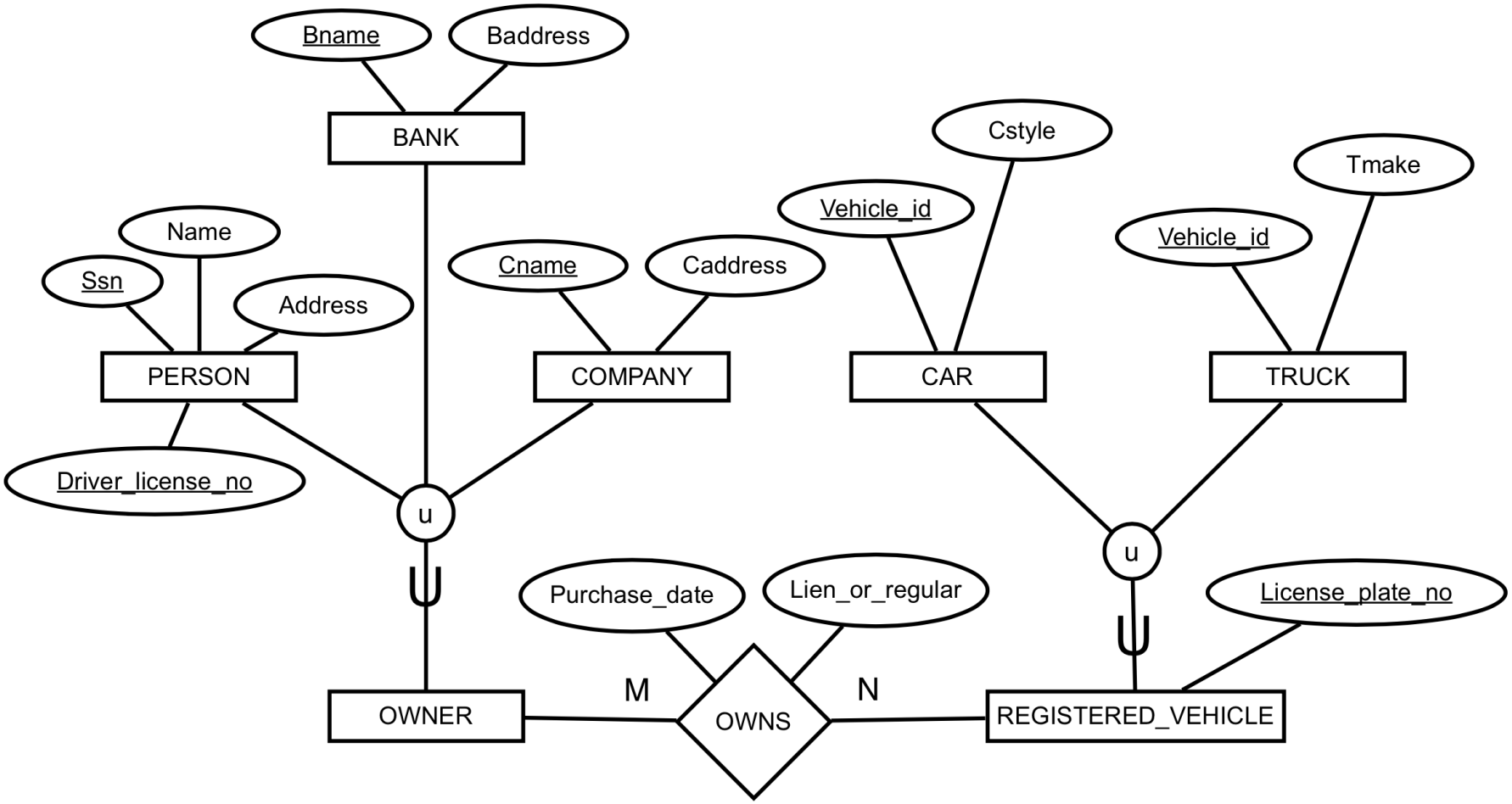


Figure 8.8
Two categories (union types): OWNER and REGISTERED_VEHICLE.

The textbook layout for this diagram may be confusing. Note that OWNER subtype inheritance pitchfork points *up*, but REGISTERED_VEHICLE subtype inheritance pitchfork points *down*.

Modeling of UNION Types Using Categories



Contrast this layout with the previous slide

A Sample UNIVERSITY EER Schema, Design Choices, and Formal Definitions

- The UNIVERSITY Database Example
  - Students and their majors
  - Transcripts, and registration
  - University's course offerings

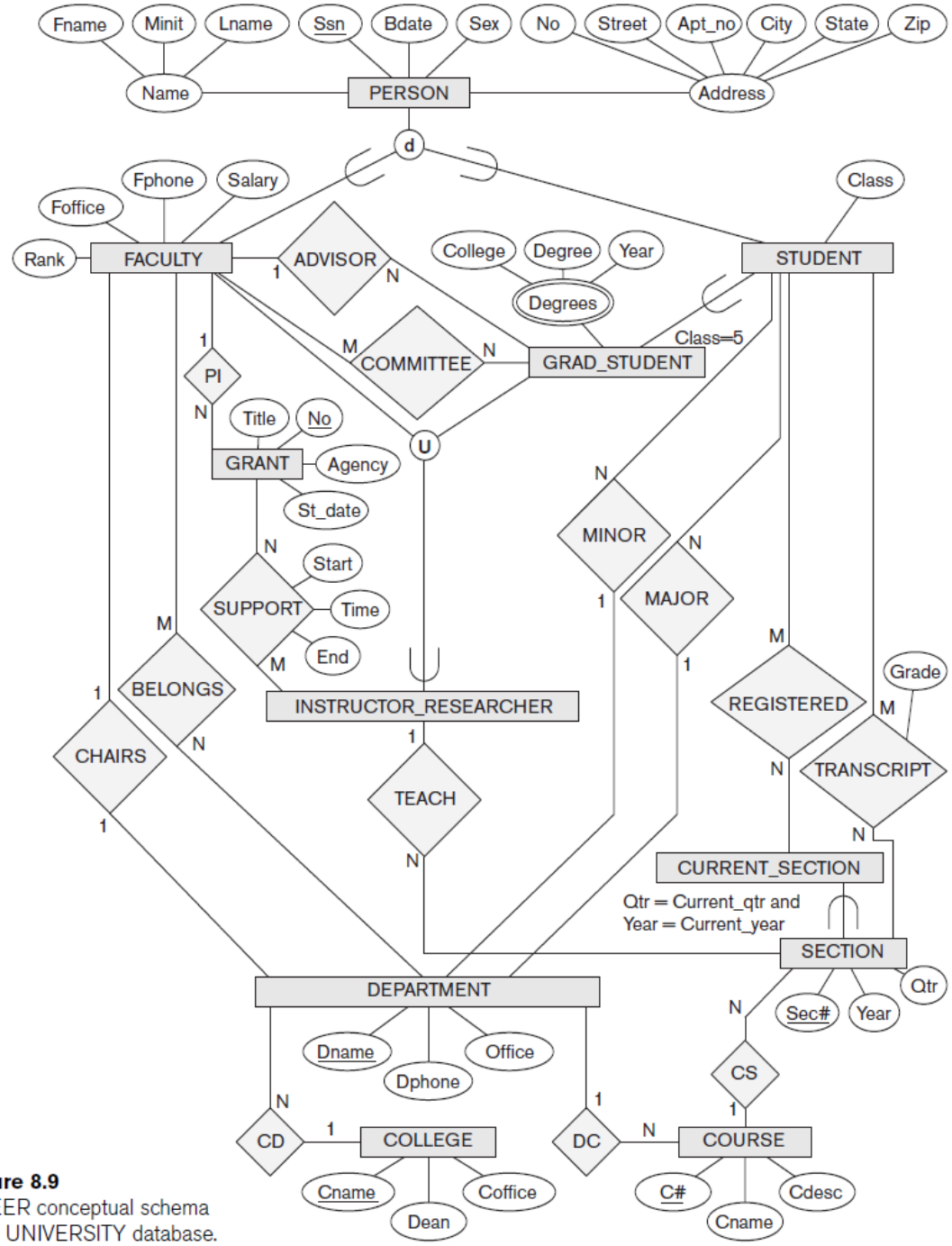


Figure 8.9
An EER conceptual schema for a UNIVERSITY database.

Design Choices for Specialization/ Generalization (cont'd.)

- If all the subclasses of a specialization/generalization have few specific attributes and no specific relationships, then…
  - Can be merged into the superclass (not enough specificity)
  - Replace with one or more type attributes that specify the subclass or subclasses to which each entity belongs

Design Choices for Specialization/ Generalization (cont'd.)

- Union types and categories should generally be avoided
  - Opinion
- Choice of:
  - Disjoint/overlapping
  - Total/partial constraints on spec. vs. gen are…
  - Driven by rules in mini-world being modeled

Formal Definitions for the EER Model Concepts

- **Class**
  - Set or collection of entities
  - Includes any of the EER schema constructs of group entities
- **Subclass**
  - Class whose entities must always be a subset of the entities in another class

Formal Definitions for the EER Model Concepts

- **Specialization**
  - Set of subclasses that have same superclass
- **Generalization**
  - Generalized entity type or superclass

Formal Definitions for the EER Model Concepts (cont'd.)

- **Predicate-defined**
  - Predicate on the attributes of is used to specify which entities in *C* are members of *S*
- **Attribute-defined**
- **User-defined**
  - Subclass that is not defined by a predicate

Formal Definitions for the EER Model Concepts (cont'd.)

- **Category**
  - Class that is a subset of the union of *n* defining superclasses
- **Relationship type**
  - Any class can participate in a relationship

Examples of Other Notations

- **UML**
  - OOP
- **Ontologies and the Semantic Web**
  - RDF / OWL

Example of Other Notation

- Representing specialization and generalization in UML class diagrams
  - Basic notation
    - See Figure 8.10
  - Base class
    - Root superclass
  - Leaf classes
    - Subclasses (leaf nodes)

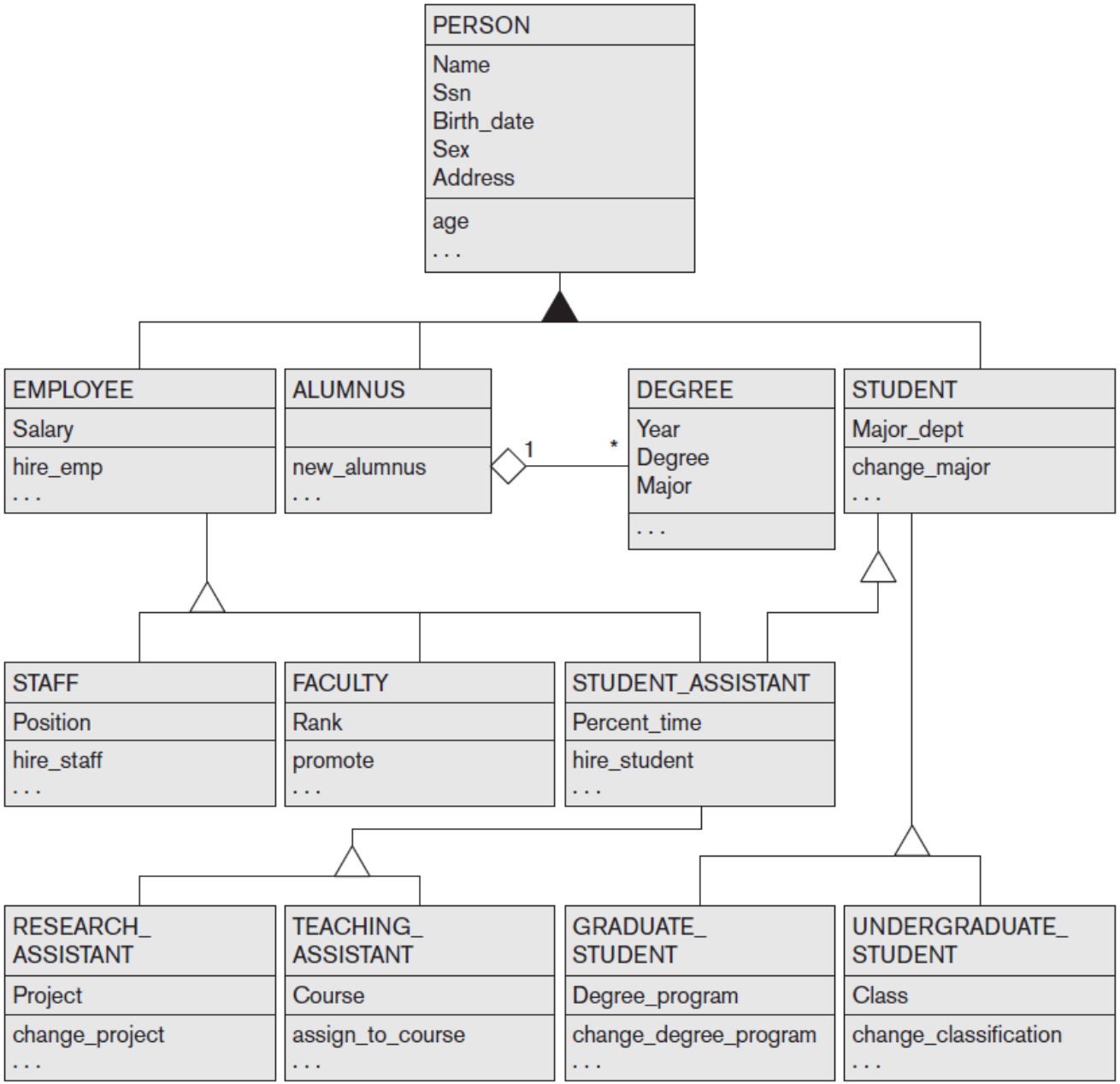


Figure 8.10
A UML class diagram corresponding to the EER diagram in Figure 8.7, illustrating UML notation for specialization/generalization.

Data Abstraction, Knowledge Representation, and Ontology Concepts

- Goal of **knowledge representation (KR)** techniques
  - Accurately model some **domain of knowledge**
  - Create an **ontology** that describes the concepts of the domain and how these concepts are interrelated
- Goals of KR are similar to those of semantic data models
  - Important similarities and differences

Classification and Instantiation

- **Classification**
  - Systematically assigning similar objects/entities to object classes/entity types
- **Instantiation**
  - Inverse of classification
  - Generation and specific examination of distinct objects of a class

Classification and Instantiation (cont'd.)

- **Exception objects**
  - Differ in some respects from other objects of class
  - KR schemes allow such **class properties**
- One class can be an instance of another class (called a meta-class)
  - Cannot be represented directly in EER model

Identification

- Abstraction process
- Classes and objects are made uniquely identifiable by means of some **identifier**
- Needed at two levels
  - To distinguish among database objects and classes
  - To identify database objects and to relate them to their real-world counterparts

Specialization and Generalization

- **Specialization**
  - Classify a class of objects into more specialized subclasses
- **Generalization**
  - Generalize several classes into a higher-level abstract class
  - Includes the objects in all these classes

Aggregation and Association

- **Aggregation**
  - Abstraction concept for building composite objects from their component objects
- **Association**
  - Associate objects from several independent classes
- Main structural distinction
  - When an association instance is deleted
    - Participating objects may continue to exist

Aggregation

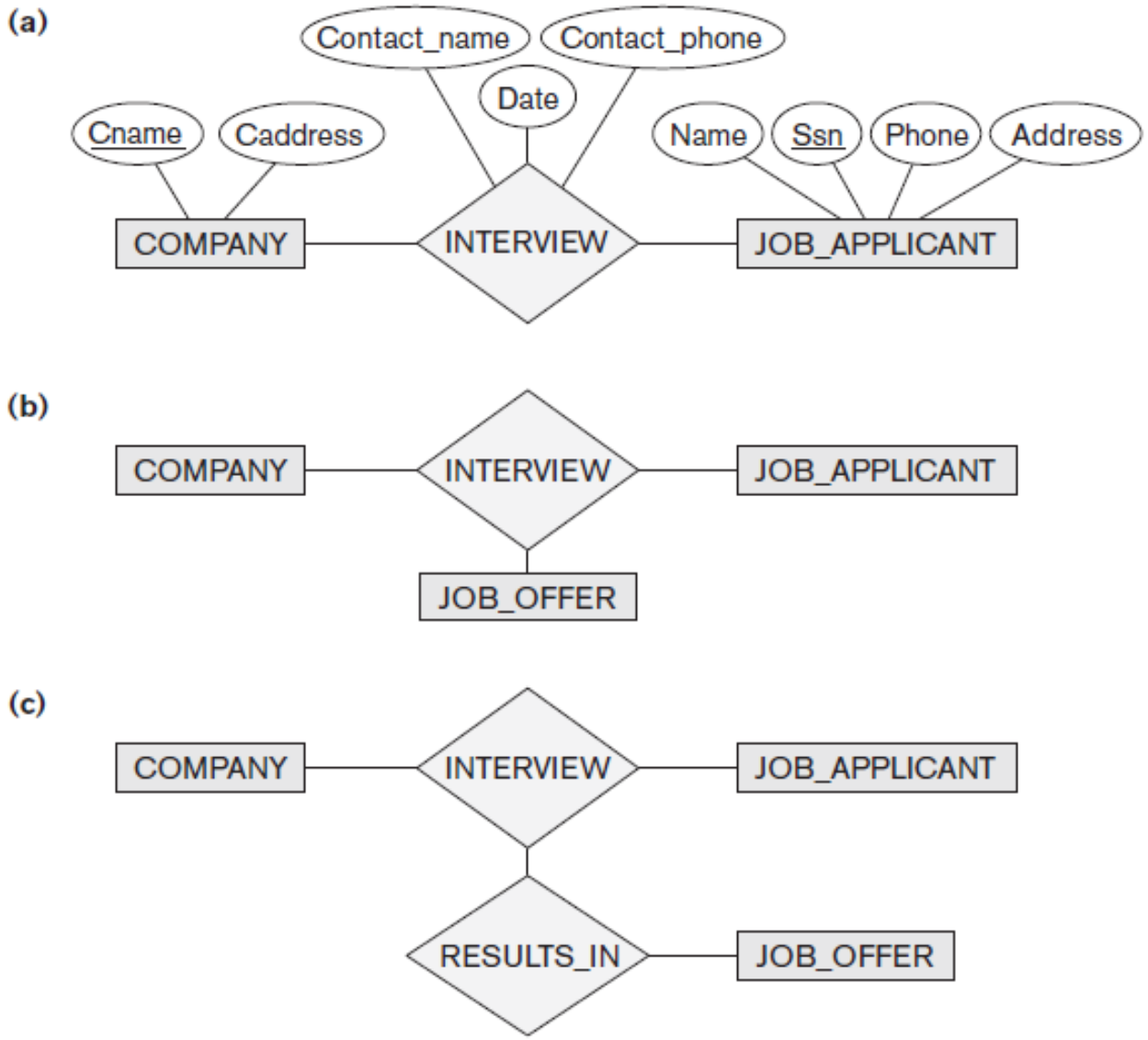


Figure 8.11
Aggregation. (a) The relationship type INTERVIEW. (b) Including JOB_OFFER in a ternary relationship type (incorrect). (c) Having the RESULTS_IN relationship participate in other relationships (not allowed in ER). (d) Using aggregation and a composite (molecular) object (generally not allowed in ER but allowed by some modeling tools). (e) Correct representation in ER.

Aggregation

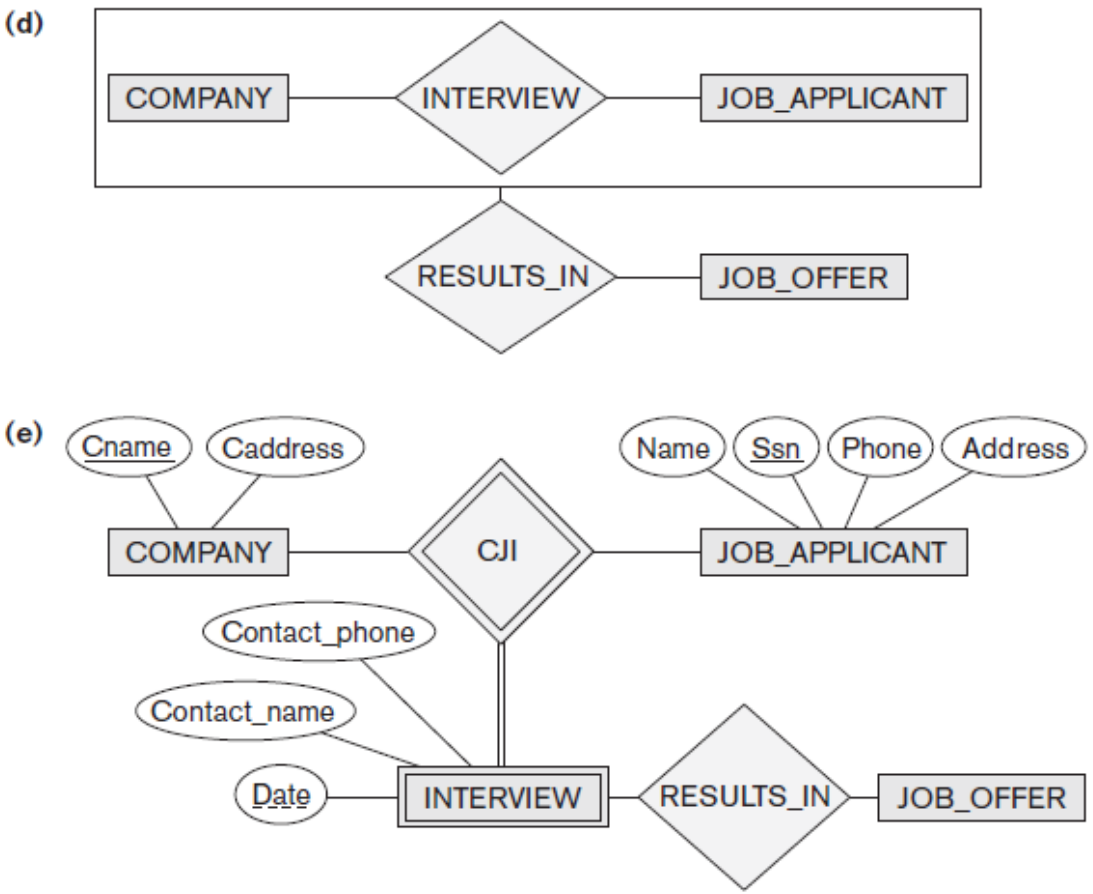


Figure 8.11
Aggregation. (a) The relationship type INTERVIEW. (b) Including JOB_OFFER in a ternary relationship type (incorrect). (c) Having the RESULTS_IN relationship participate in other relationships (not allowed in ER). (d) Using aggregation and a composite (molecular) object (generally not allowed in ER but allowed by some modeling tools). (e) Correct representation in ER.

Ontologies and the Semantic Web

- Documents contain less structure than database information does
- **Semantic Web**
  - Allow meaningful information exchange and search among machines
- **Ontology**
  - Specification of a **conceptualization**
- **Specification**
  - Language and vocabulary terms used to specify conceptualization

Ontologies and the Semantic Web

- Documents contain less structure than database information does
- **Semantic Web**
  - Allow meaningful information exchange and search among machines
- **Ontology**
  - Specification of a **conceptualization**
- **Specification**
  - Language and vocabulary terms used to specify conceptualization

Summary

- Enhanced ER or EER model
  - Extensions to ER model that improve its representational capabilities
  - Subclass and its superclass
  - Category or union type
- Notation and terminology of UML for representing specialization and generalization