Ch 9: Mapping EER to Relational

Follow a seven-step algorithm to convert the basic ER model constructs into relations

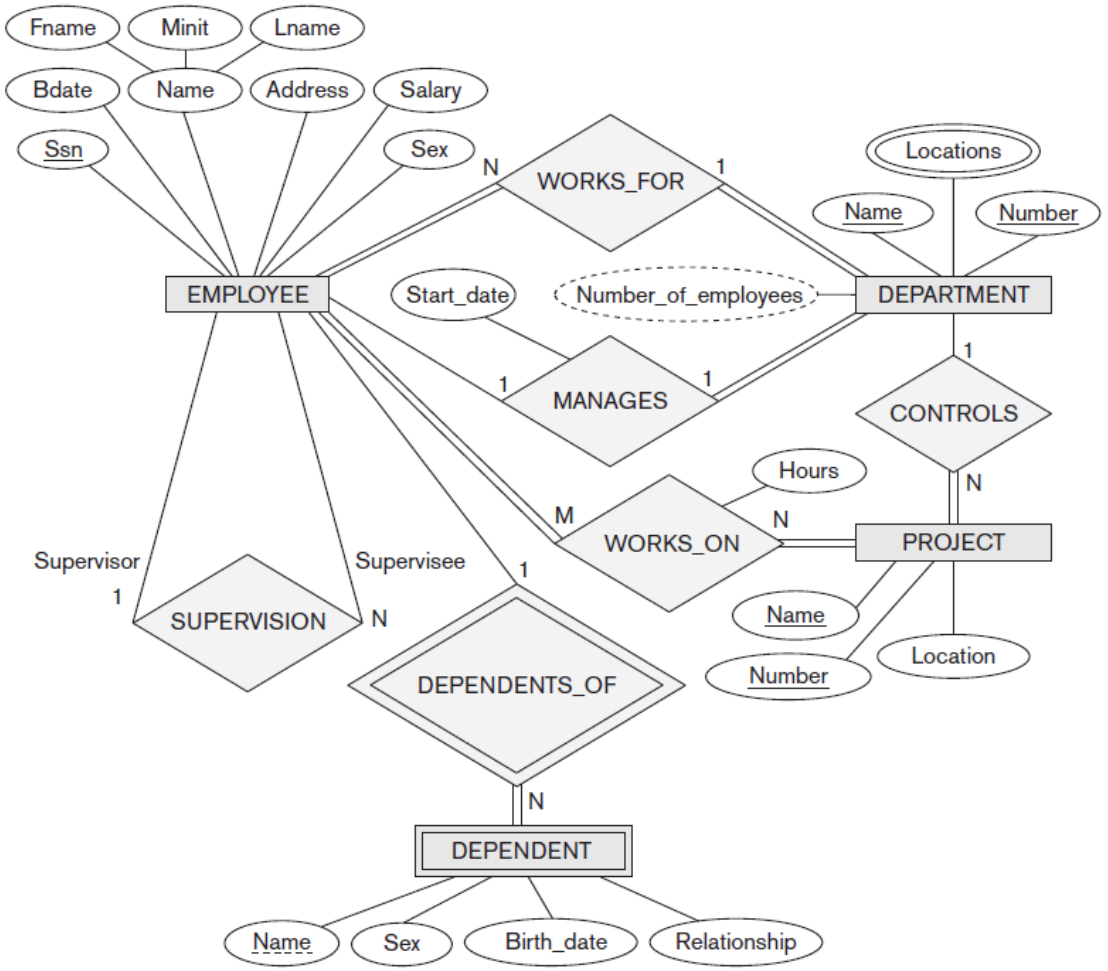
… steps 1-7

Additional steps for EER model

for specialization/generalization … steps 8a thru 8d

ER-to-Relational Mapping

Figure 9.1
The ER conceptual schema diagram for the COMPANY database.



See next slide for result of applying steps 1-7

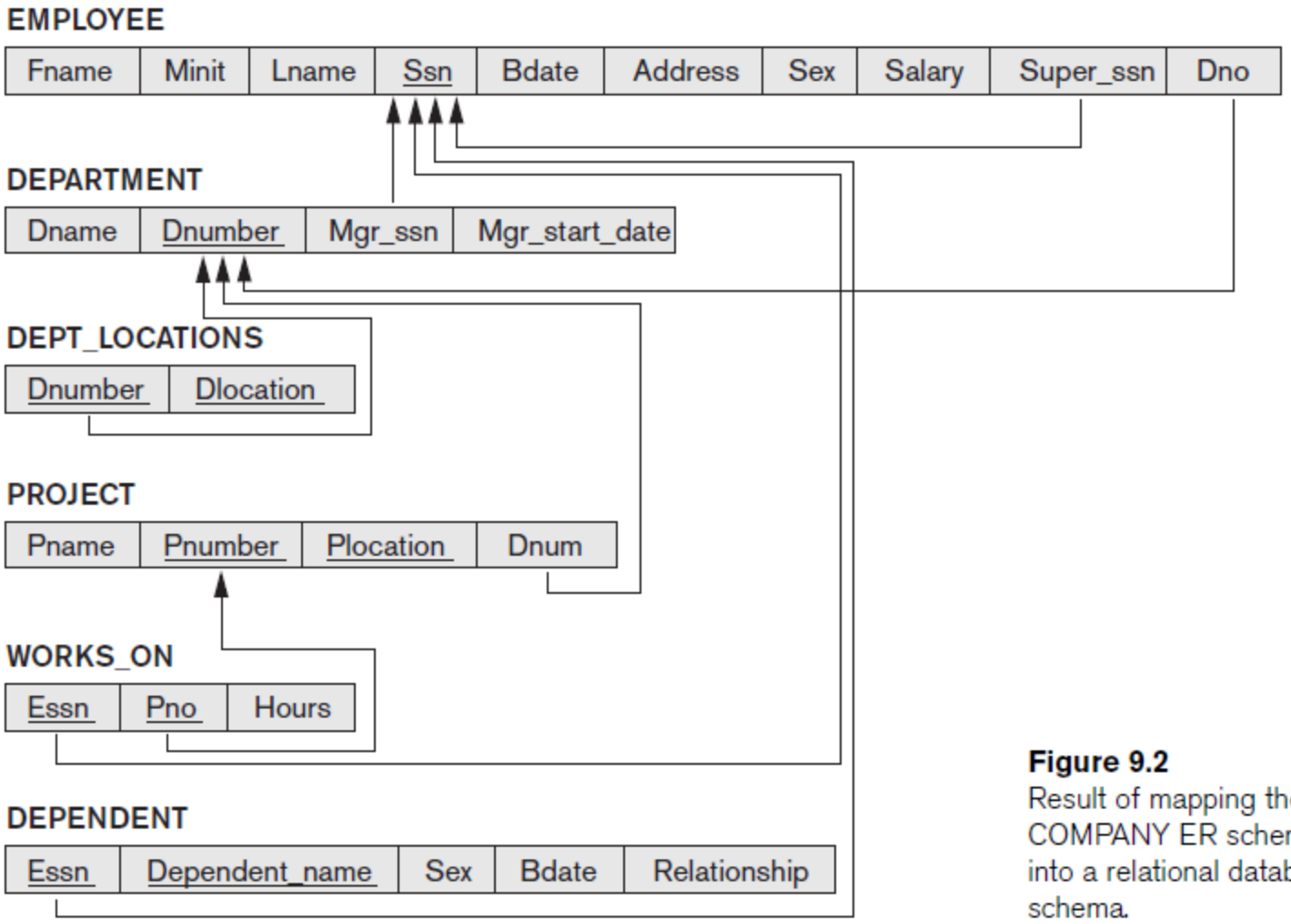


Figure 9.2
Result of mapping the COMPANY ER schema into a relational database schema.

ER-to-Relational Mapping Algorithm

- Step 1: Mapping of Regular Entity Types
  - For each regular entity type, create a relation *R* that includes all the simple attributes of *E*
  - Choose one of the keys as the PK

(simple attributes are attributes that are not composite)

ER-to-Relational Mapping Algorithm (cont'd.)

- Step 2: Mapping of Weak Entity Types
  - For each weak entity type, create a relation *R* and include all simple attributes of the entity type as attributes of *R*
  - Include PK of owner(s)
  - PK of R comprises PK of owner(s) and a discriminating attribute.
  - PKs of *owner(s)* are FKs in R

ER-to-Relational Mapping Algorithm (cont'd.)

- Step 3: Mapping of Binary 1:1 Relationship Types

  For each binary 1:1 relationship type

  - Identify relations that correspond to entity types participating in *R*

  3 approaches:

  - Foreign key approach

    Choose one relation to hold a FK referencing the other

  - Merged relationship approach

    Merge the two relations into one

  - *relationship relation* approach (as with m:n relationship)

    2 FKs in new relation, one FK chosen as PK, other is a key

ER-to-Relational Mapping Algorithm (cont'd.)

- Step 4: Mapping of Binary 1:*N* Relationship Types

  2 approaches:

  – For each binary 1:*N* relationship type
    - Identify relation (S) that represents participating entity type at *N*-side of relationship type
    - Include primary key of other entity type as foreign key in *S*
    - Include simple attributes of 1:*N* relationship type as attributes of *S*

ER-to-Relational Mapping Algorithm

- *Alternative approach*
  - Use the relationship relation option (as with m:n relationship)

ER-to-Relational Mapping Algorithm (cont'd.)

- Step 5: Mapping Binary *M*:*N* Relationship Types
  - For each binary *M*:*N* relationship type

    (The *relationship relation* approach)
    - Create a new relation *S*
    - Include primary key of participating entity types as foreign key attributes in *S*
    - PK is combination of FKs and a discriminator (if exists)
    - Include any simple attributes of *M*:*N* relationship type in S

Elsewhere you may have run across a term such as *intersection* relation or *associative* relation. These represent the same concept as *relationship* relation.

ER-to-Relational Mapping Algorithm (cont'd.)

- Step 6: Mapping of Multivalued Attributes
  - For each multivalued attribute A of an entity type S
    - Create a new relation R
    - Primary key of *R* is the combination of *A* and PK of relation created for S
    - If the multivalued attribute is composite, include its simple components

ER-to-Relational Mapping

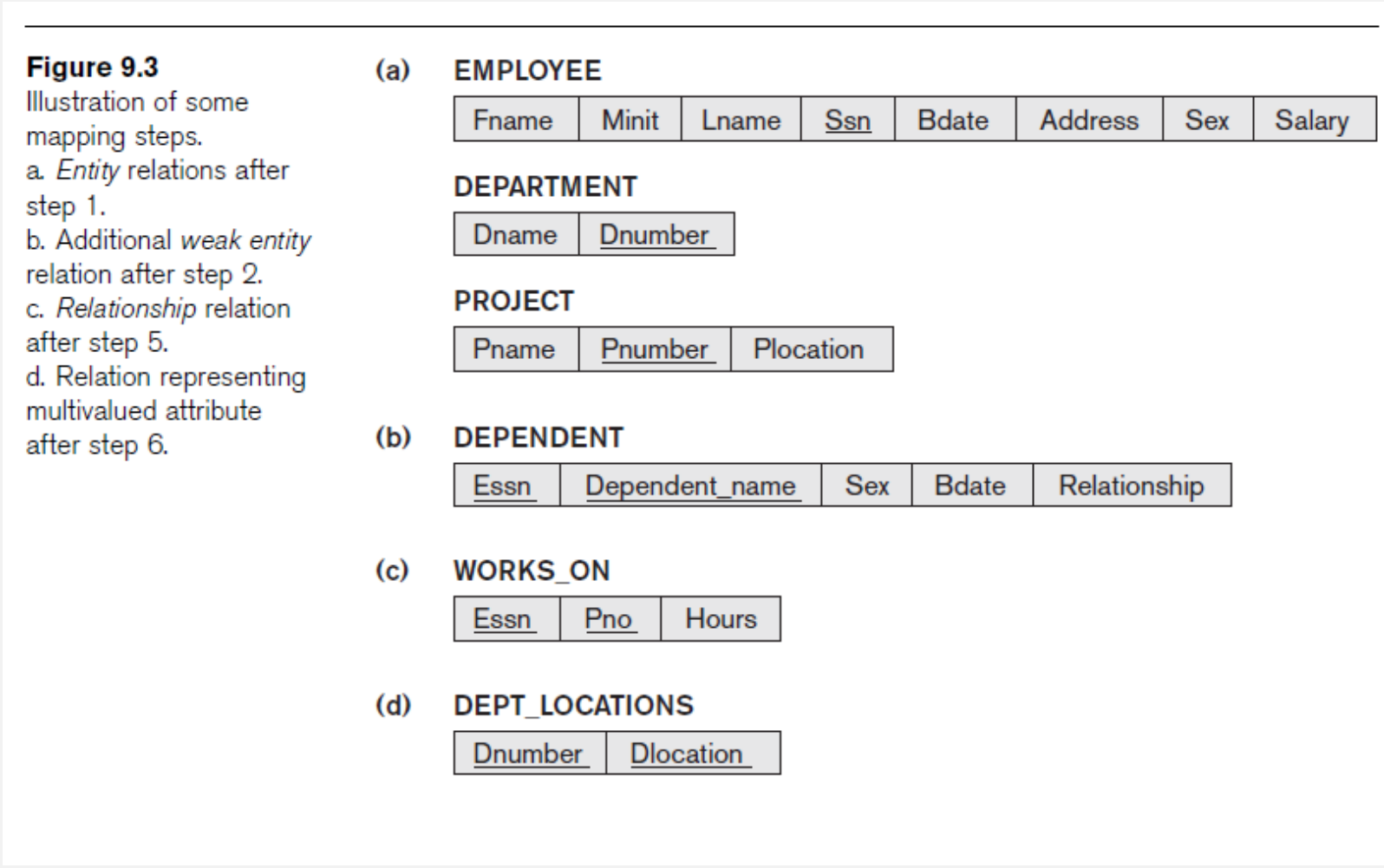


Figure 9.3
Illustration of some mapping steps.
a. *Entity* relations after step 1.
b. Additional *weak entity* relation after step 2.
c. *Relationship* relation after step 5.
d. Relation representing multivalued attribute after step 6.

ER-to-Relational Mapping Algorithm

- Step 7: Mapping of *N*-ary Relationship Types
  - For each *n*-ary relationship type *R*
    - Create a new relation *S* to represent *R*
    - Include primary keys of participating entity types as foreign keys
    - PK is combination of FKs and a discriminator (if exists)
    - Include any simple attributes as attributes in S
    - This is just an extension of m:n relationships

Summary of Mapping for ER Model

Table 9.1 Correspondence between ER and Relational Models

ER MODEL	RELATIONAL MODEL
Entity type	*Entity* relation
1:1 or 1:N relationship type	Foreign key (or *relationship* relation)
M:N relationship type	*Relationship* relation and *two* foreign keys
n-ary relationship type	*Relationship* relation and *n* foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Key attribute	Primary (or secondary) key

Mapping EER Model Constructs to Relations

- Extending ER-to-relational mapping algorithm

Mapping of Specialization or Generalization (9.2.1)

- Step 8: Options for Mapping Specialization or Generalization.

  Can be used for shared subclasses.

  - Option 8A: Multiple relations—one for the superclass and one for each subclass
    - For any specialization

      (total or partial, disjoint or overlapping)
    - PK of subclass relation is FK to superclass relation.
    - An equi-join is needed to get all attributes for an entity that is an instance of a subclass. An entity can be represented many times.
    - Consider Figure 9.5a)

Mapping of Specialization or Generalization (9.2.1)

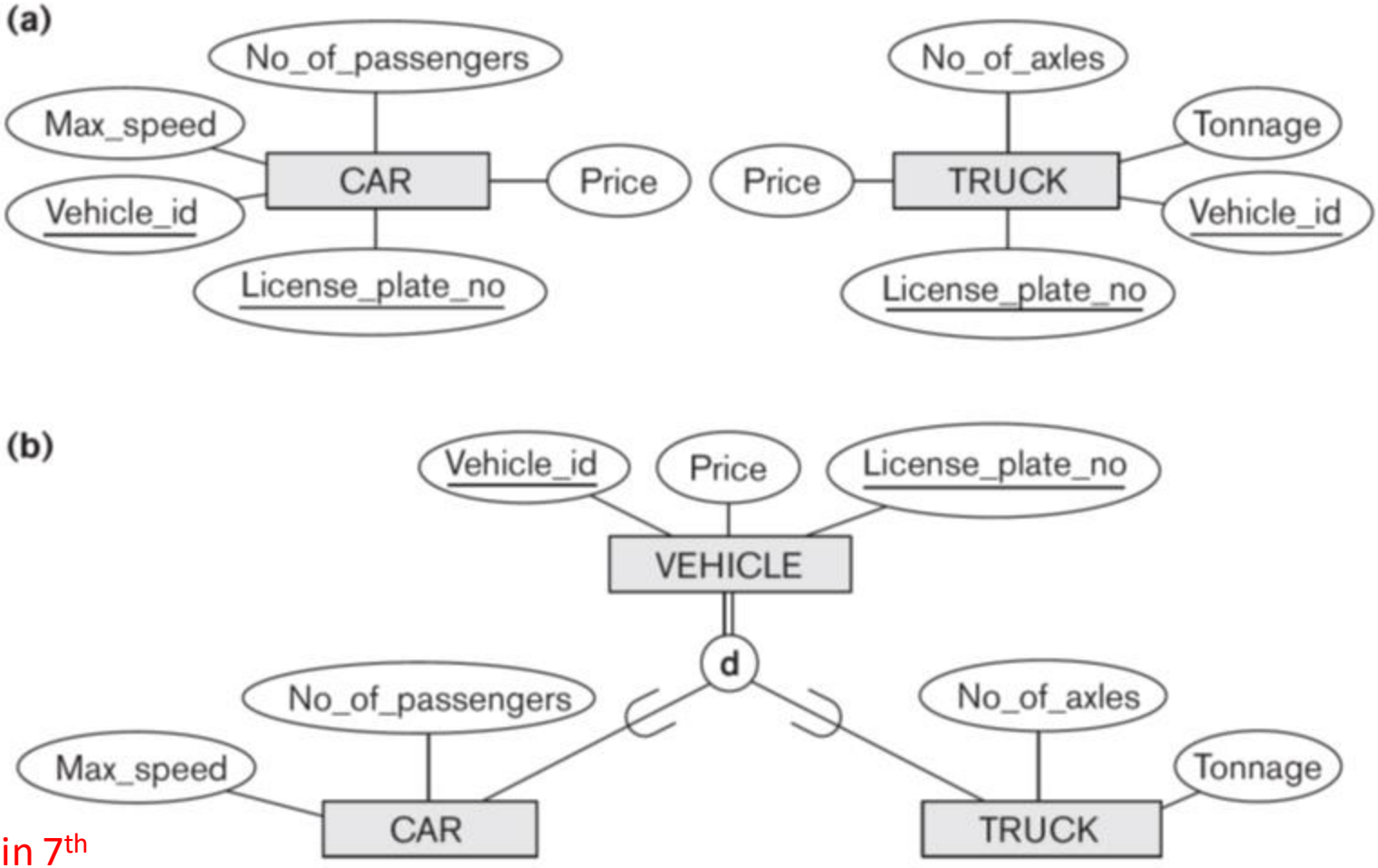
- Option 8B: Multiple relations but only for subclasses
  - Only for subclassing that is *total*
  - If specialization is overlapping there can be entities represented in more than one relation
  - Example, see figure 9.5b)

Mapping of Specialization or Generalization (9.2.1)

- Option 8C: Single relation representing all classes including one type attribute
  - A type (discriminating) attribute indicates subclass
  - Subclasses must be disjoint
  - Potential for generating many NULL values if many specific attributes exist in the subclasses
  - Example 9.5c)

Mapping of Specialization or Generalization (9.2.1)

- Option 8D: Single relation representing all classes including multiple type attributes
  - Useful for overlapping subclasses
  - Potential for generating many NULL values if many specific attributes exist in the subclasses
  - Example 9.5d)

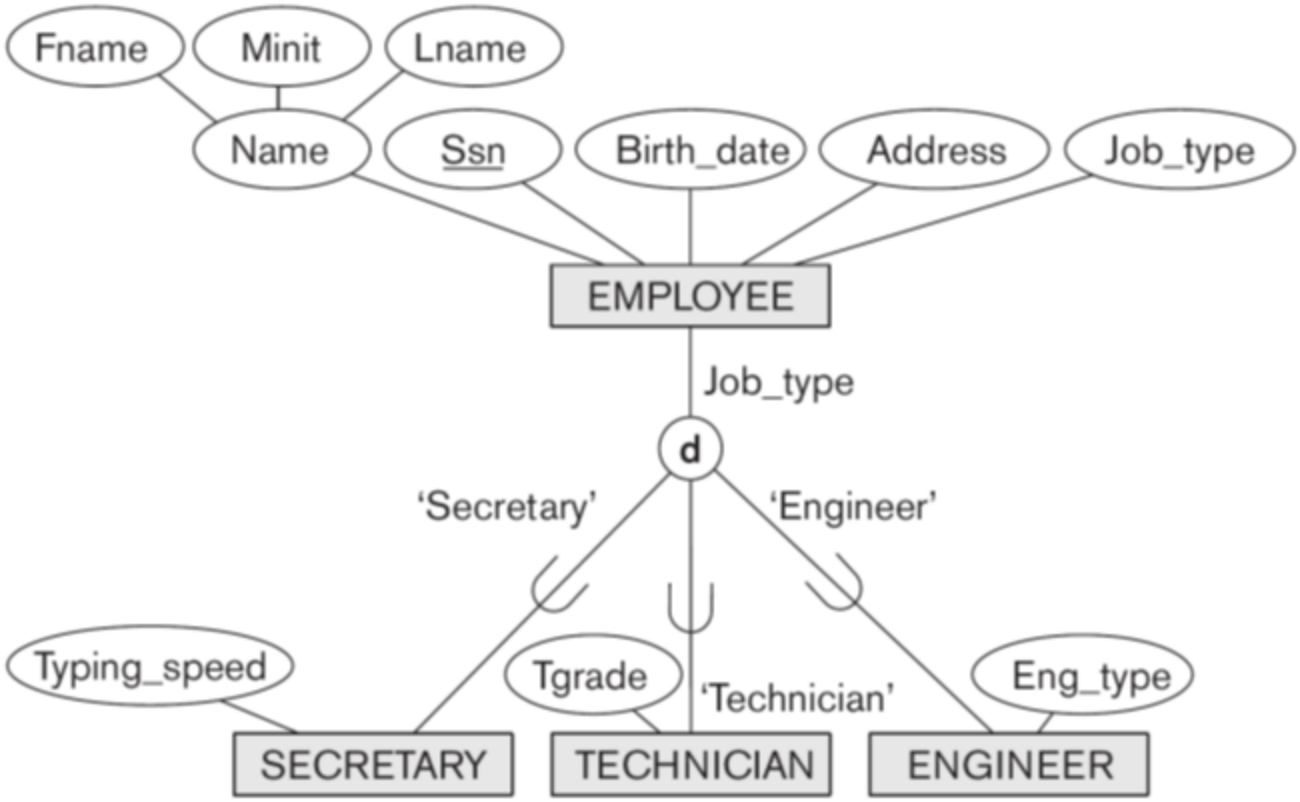


4.3 in 7th

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

4.4 in 7th

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



4.5 in 7th

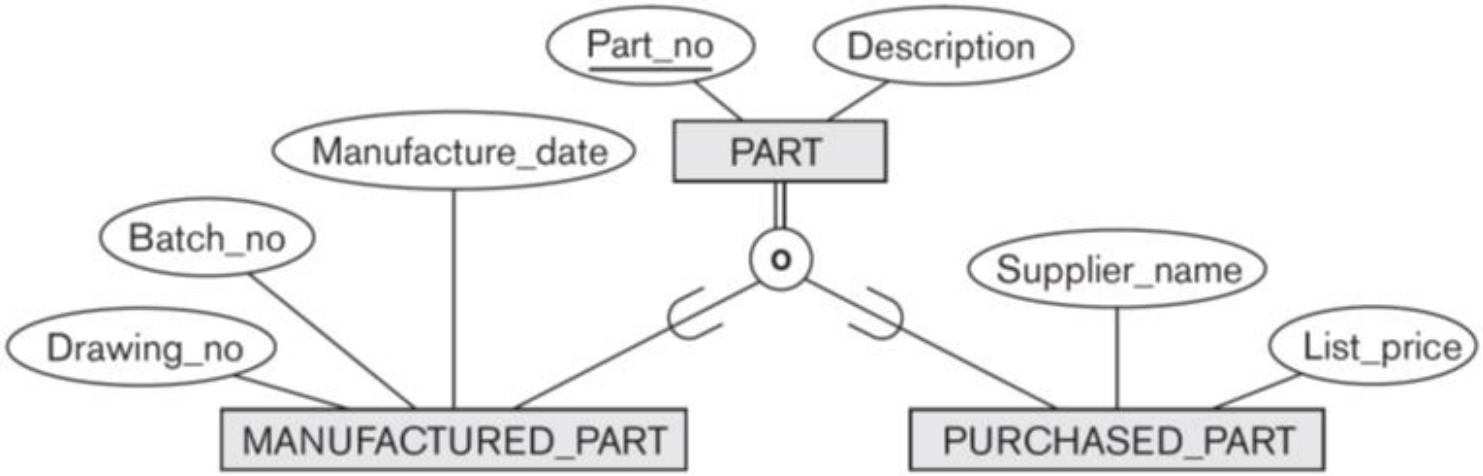


Figure 8.5
EER diagram notation for an overlapping (nondisjoint) specialization.

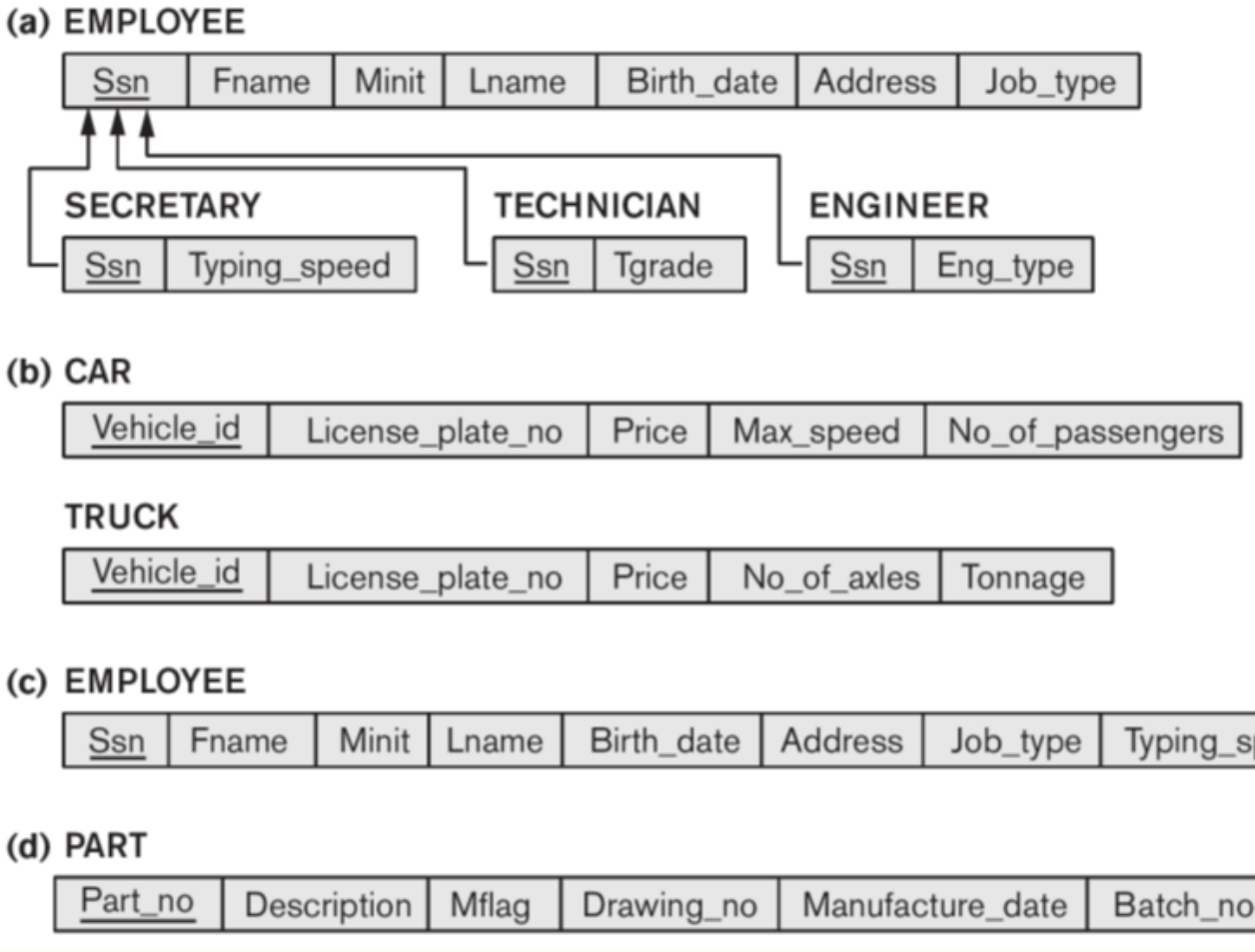


Figure 9.5
Options for mapping specialization or generalization. (a) Mapping the EER schema in Figure 8.4 using option 8A. (b) Mapping the EER schema in Figure 8.3(b) using option 8B. (c) Mapping the EER schema in Figure 8.4 using option 8C. (d) Mapping Figure 8.5 using option 8D with Boolean type fields Mflag and Pflag.

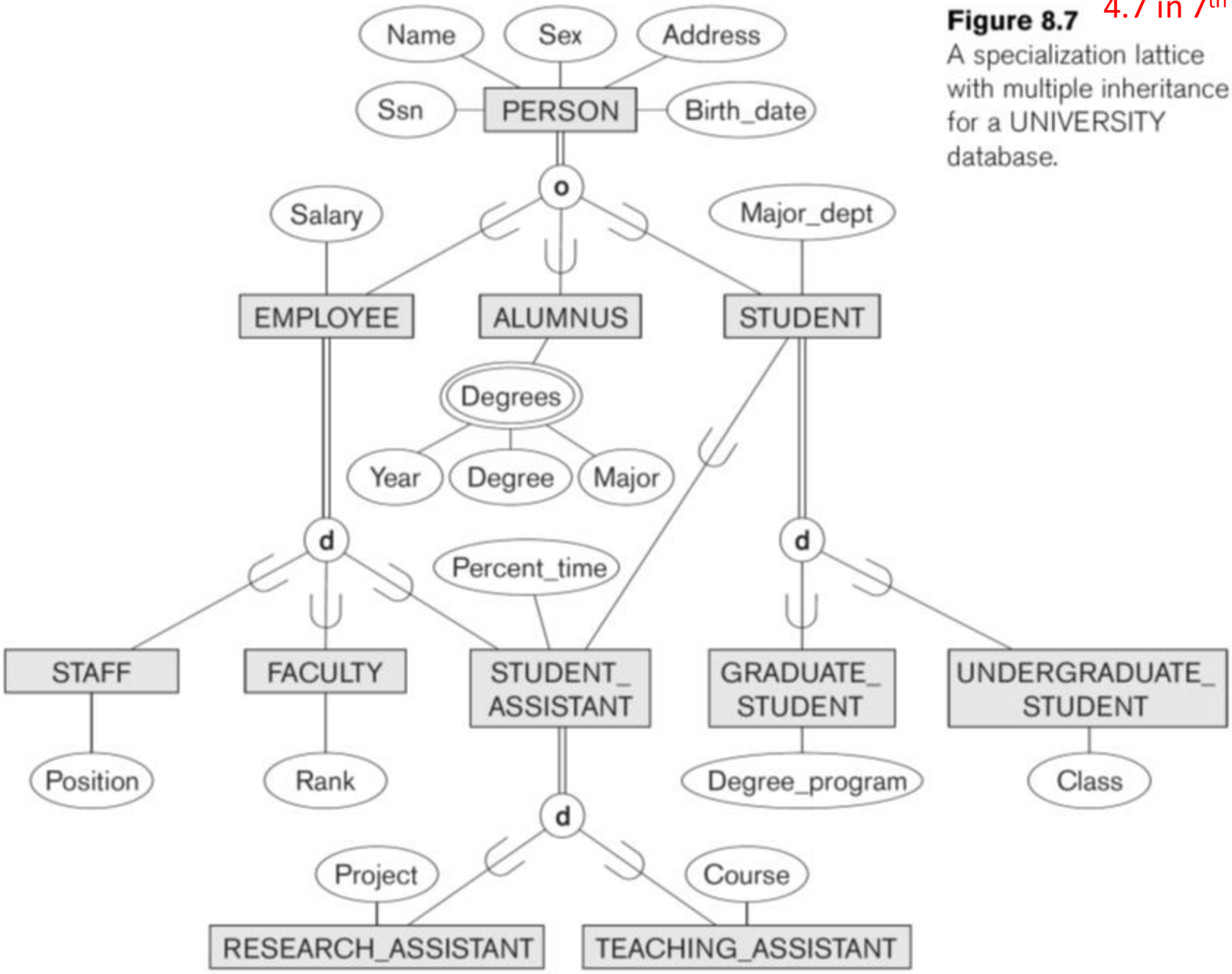


Figure 8.7 4.7 in 7th
A specialization lattice with multiple inheritance for a UNIVERSITY database.

Applied 8A to Person, Employee, Alumnus, Student

Applied 8C to Employee, Staff, Faculty, Student Assistant – *Employee type*

Applied 8D to Student Assistant, Research Assistant, Teaching Assistant – *Ta flag, Ta flag*

Applied 8D to Student, Student Assistant, Graduate Student, Undergraduate Student – *Grad flag, Undergrad flag, Student assist flag*

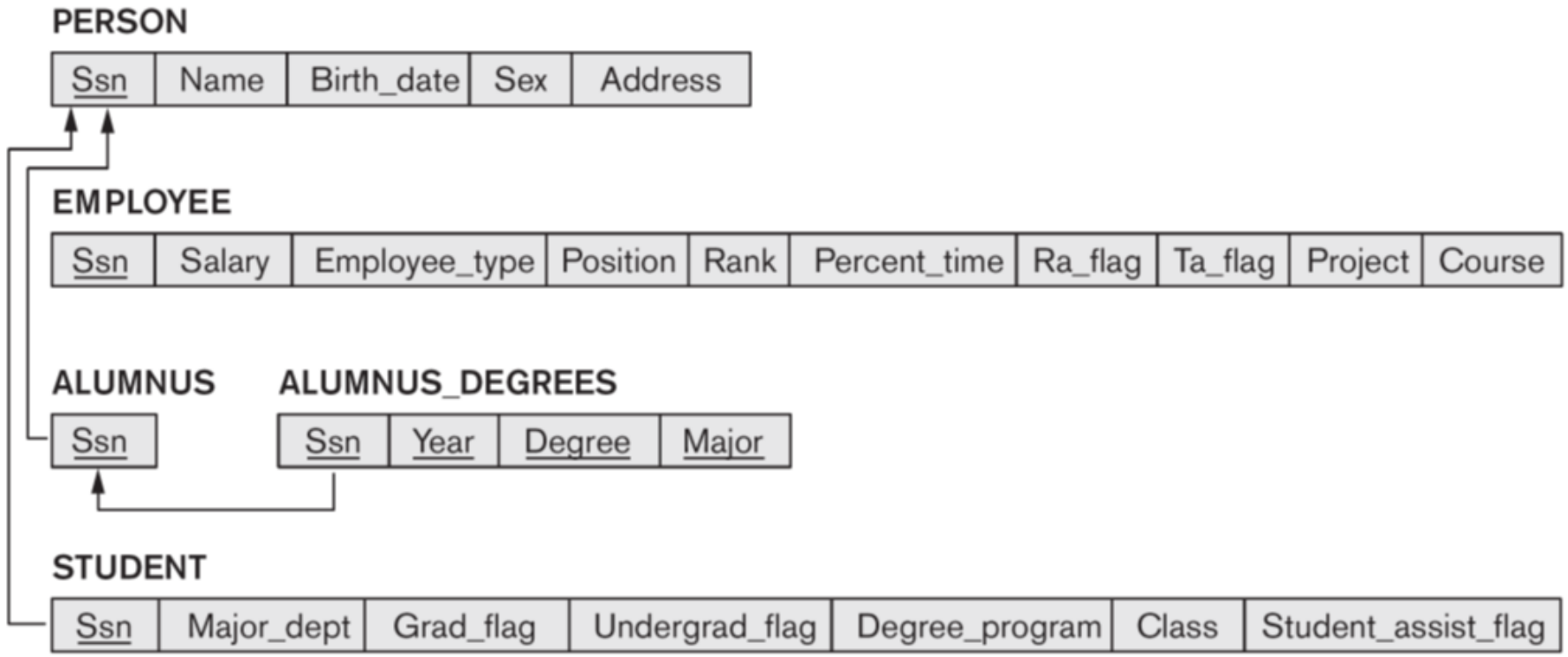


Figure 9.6
Mapping the EER specialization lattice in Figure 8.8 using multiple options.