CHAPTER 9

Relational Database Design by ER- and EER-to-Relational Mapping

Chapter Outline

ER-to-Relational Mapping Algorithm

- ✓ Step 1: Mapping of Regular Entity Types
- ✓ Step 2: Mapping of Weak Entity Types
- ✓ Step 3: Mapping of Binary 1:1 Relation Types
- ✓ Step 4: Mapping of Binary 1:N Relationship Types.
- ✓ Step 5: Mapping of Binary M:N Relationship Types.
- ✓ Step 6: Mapping of Multivalued attributes.
- ✓ Step 7: Mapping of N-ary Relationship Types. ■

Mapping EER Model Constructs to Relations

- ✓ Step 8: Options for Mapping Specialization or Generalization.
- ✓ Step 9: Mapping of Union Types (Categories).

GOALS during Mapping

- Preserve all information (that includes all attributes)
- Maintain the constraints to the extent possible
- Minimize null values

ER-to-Relational Mapping Algorithm

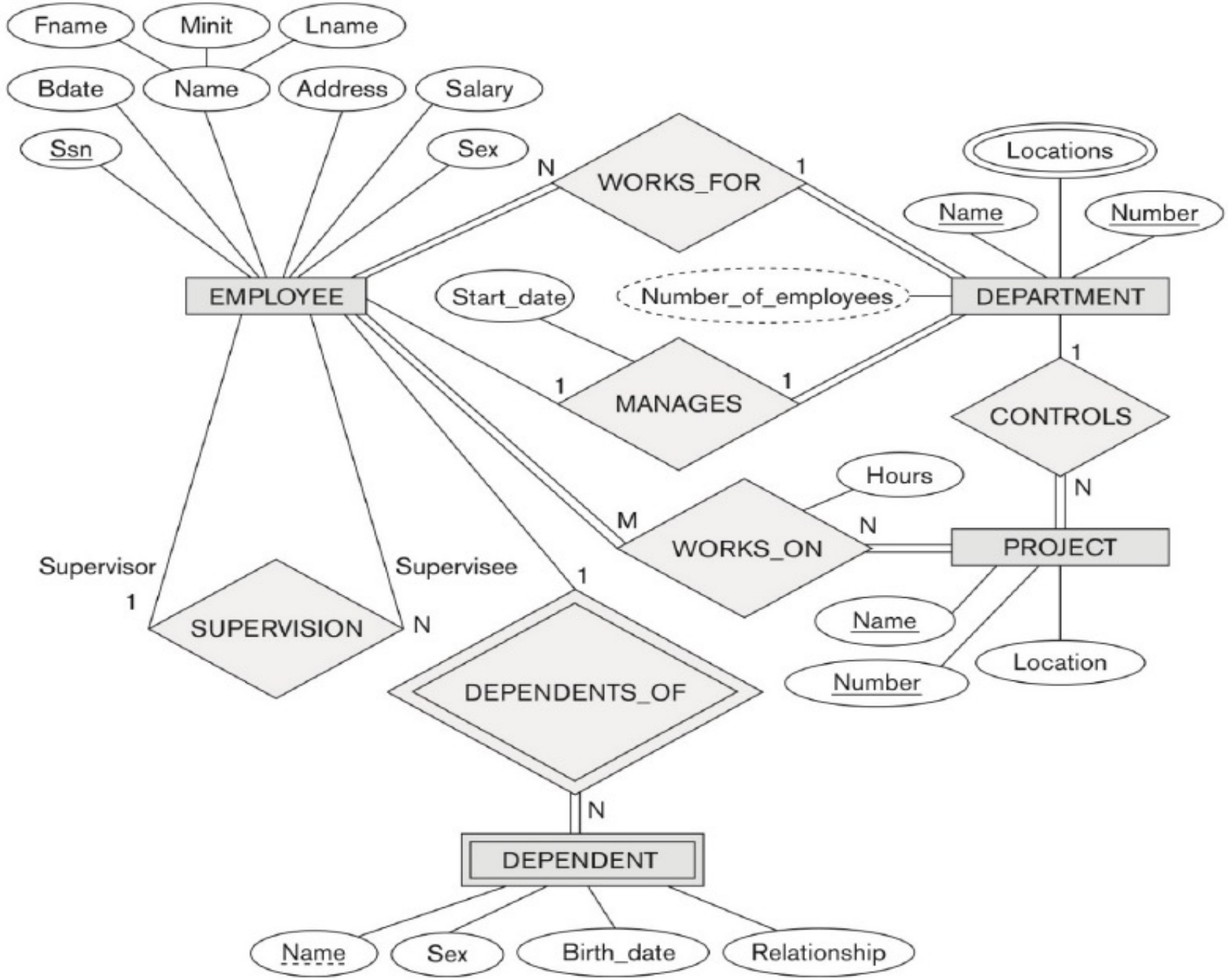
Step 1: Mapping of Regular Entity Types.

- For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.
- Choose one of the key attributes of E as the primary key for R.
- If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

Example: We create the relations EMPLOYEE, DEPARTMENT, and PROJECT in the relational schema corresponding to the regular entities in the ER diagram.

- ✓ SSN, DNUMBER, and PNUMBER are the primary keys for the relations EMPLOYEE, DEPARTMENT, and PROJECT as shown.

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



ER-to-Relational Mapping Algorithm (contd.)

Step 2: Mapping of Weak Entity Types

- ✓For each weak entity type W in the ER schema with owner entity type E, create a relation R & include all simple attributes (or simple components of composite attributes) of W as attributes of R.
- ✓ Also, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).
- ✓ The primary key of R is the *combination of* the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

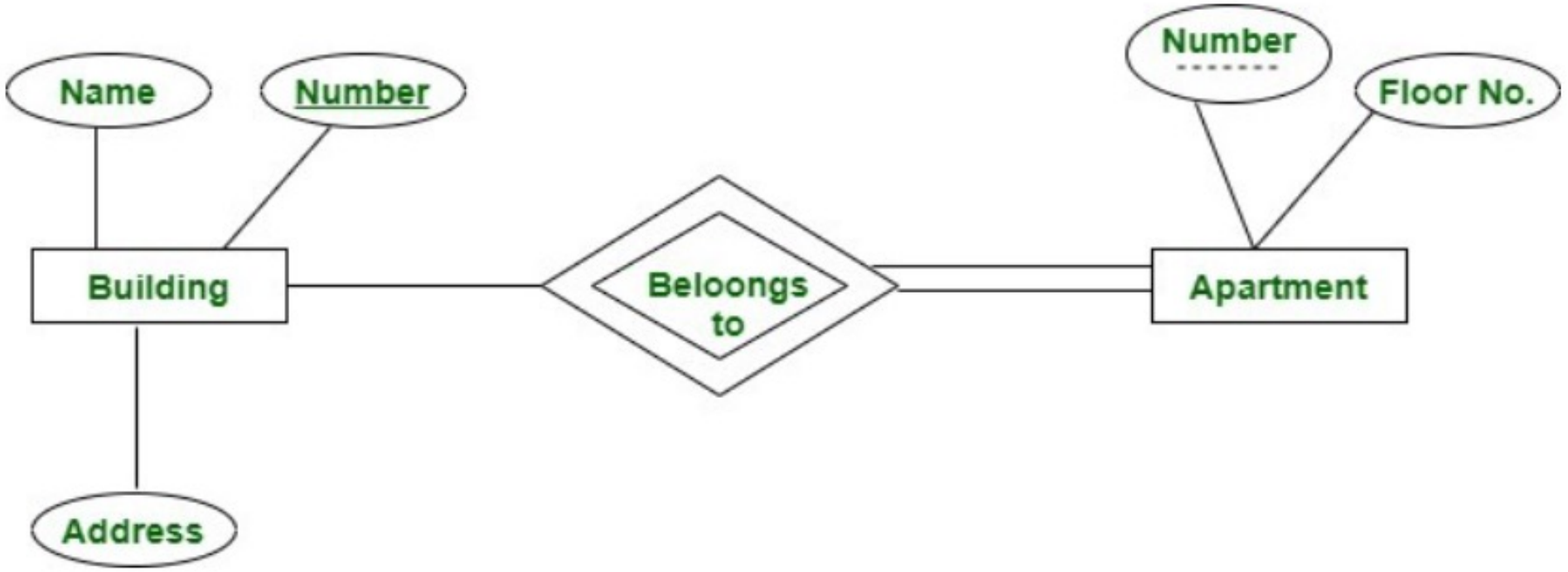
Example: Create the relation DEPENDENT in this step to correspond to the weak entity type DEPENDENT.

- ✓ Include the primary key SSN of the EMPLOYEE relation as a foreign key attribute of DEPENDENT (renamed to ESSN).
- ✓ The primary key of the DEPENDENT relation is the combination {ESSN, DEPENDENT_NAME} because DEPENDENT_NAME is the partial key of DEPENDENT.

Partial Key *(recall …)*

- ✓ Set of attributes that are used to uniquely identify a weak entity set is called the Partial key.
- ✓ The partial Key of the weak entity set is also known as a discriminator
- ✓ It is just a part of the key as only a subset of the attributes can be identified using it.
- ✓ It is partially unique and can be combined with other strong entity set to uniquely identify the tuples.

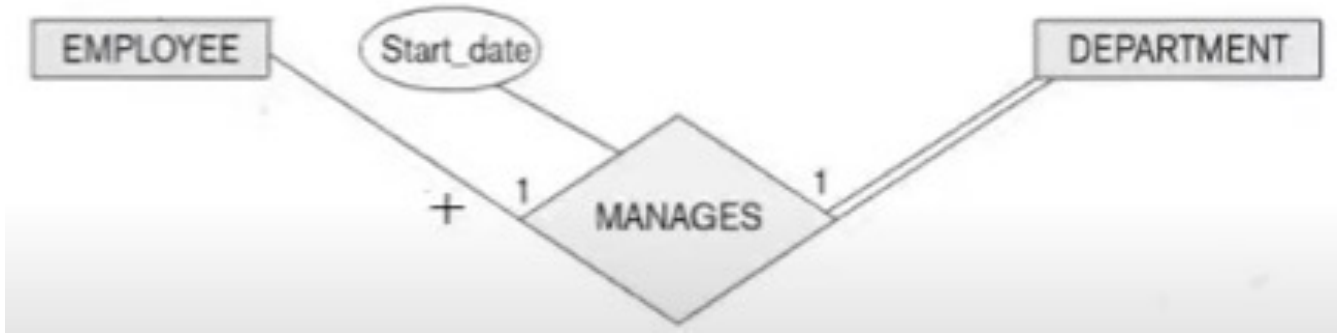
Here we have an **apartment** as a weak entity and **building** as a strong entity type connected via 'belongs to' relationship set. Apartment number is not globally unique i.e. more than one apartment may have same number globally but it is unique for a particular building since a building may not have same apartment number. Thus apartment number cannot be primary key of entity Apartment but it is a partial key shown with a dashed line.



ER-to-Relational Mapping Algorithm (contd.)

Step 3: Mapping of Binary 1:1 Relation Types

- For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.
  Choose one of the relations-say S-and include a foreign key in S the primary key of T.
  It is better to choose an entity type with total participation in R in the role of S.



Example: 1:1 relation MANAGES is mapped by choosing the participating entity type DEPARTMENT to serve in the role of S, because its participation in the MANAGES relationship type is total.

Department (Dname, Dnumber)

After modification –

Department (Dname, Dnumber, mgr_ssn, mgr_start_date)

ER-to-Relational Mapping Algorithm (contd.)

Step 4: Mapping of Binary 1:N Relationship Types.

- ✓ For each regular binary 1:N relationship type R, identify the relation S that represent the participating entity type at the N-side of the relationship type.
- ✓ Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R.
- ✓ Include any simple attributes of the 1:N relation type as attributes of S.

Example: 1:N relationship types WORKS_FOR, CONTROLS, and SUPERVISION in the figure.

- ✓ For WORKS_FOR we include the primary key DNUMBER of the DEPARTMENT relation as foreign key in the EMPLOYEE relation and call it DNO.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

Step 5: Mapping of Binary M:N Relationship Types

- ✓ For each regular binary M:N relationship type R, *create a new relation* S to represent R. This is a *relationship relation.*
- ✓ Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; *their combination will form the primary key* of S.
- ✓ Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

Example: The M:N relationship type WORKS_ON from the ER diagram is mapped by creating a relation WORKS_ON in the relational database schema. ■ The primary keys of the PROJECT and EMPLOYEE relations are included as foreign keys in WORKS_ON and renamed PNO and ESSN, respectively. ■ Attribute HOURS in WORKS_ON represents the HOURS attribute of the relation type. The primary key of the WORKS_ON relation is the combination of the foreign key attributes {ESSN, PNO}.

Works_on (ESSN, PNO, hours)

Step 6: Mapping of Multivalued attributes.

- ✓ For each multivalued attribute A, create a new relation R.
- ✓ This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute.
- ✓ The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

Example: The relation DEPT_LOCATIONS is created.

- ✓ The attribute DLOCATION represents the multivalued attribute LOCATIONS of DEPARTMENT, while DNUMBER-as foreign key-represents the primary key of the DEPARTMENT relation.
- ✓ The primary key of R is the combination of {DNUMBER, DLOCATION}.

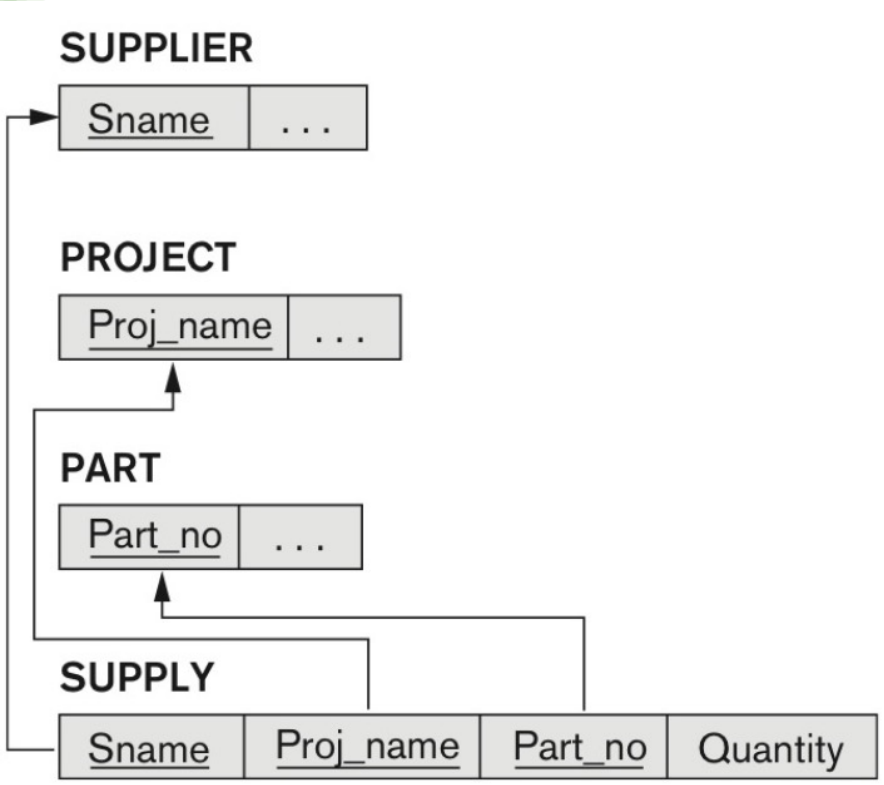
Step 7: Mapping of N-ary Relationship Types.

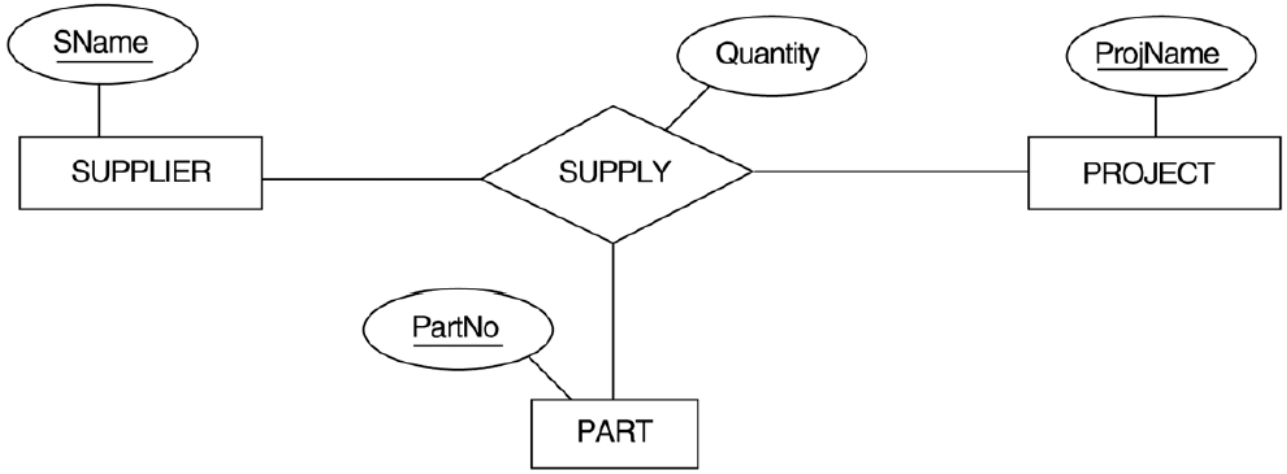
- ✓ For each n-ary relationship type R, where n>2, create a new relationship S to represent R.
- ✓ Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.
- ✓ Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.

Example: The relationship type SUPPY in the ER on the next slide.

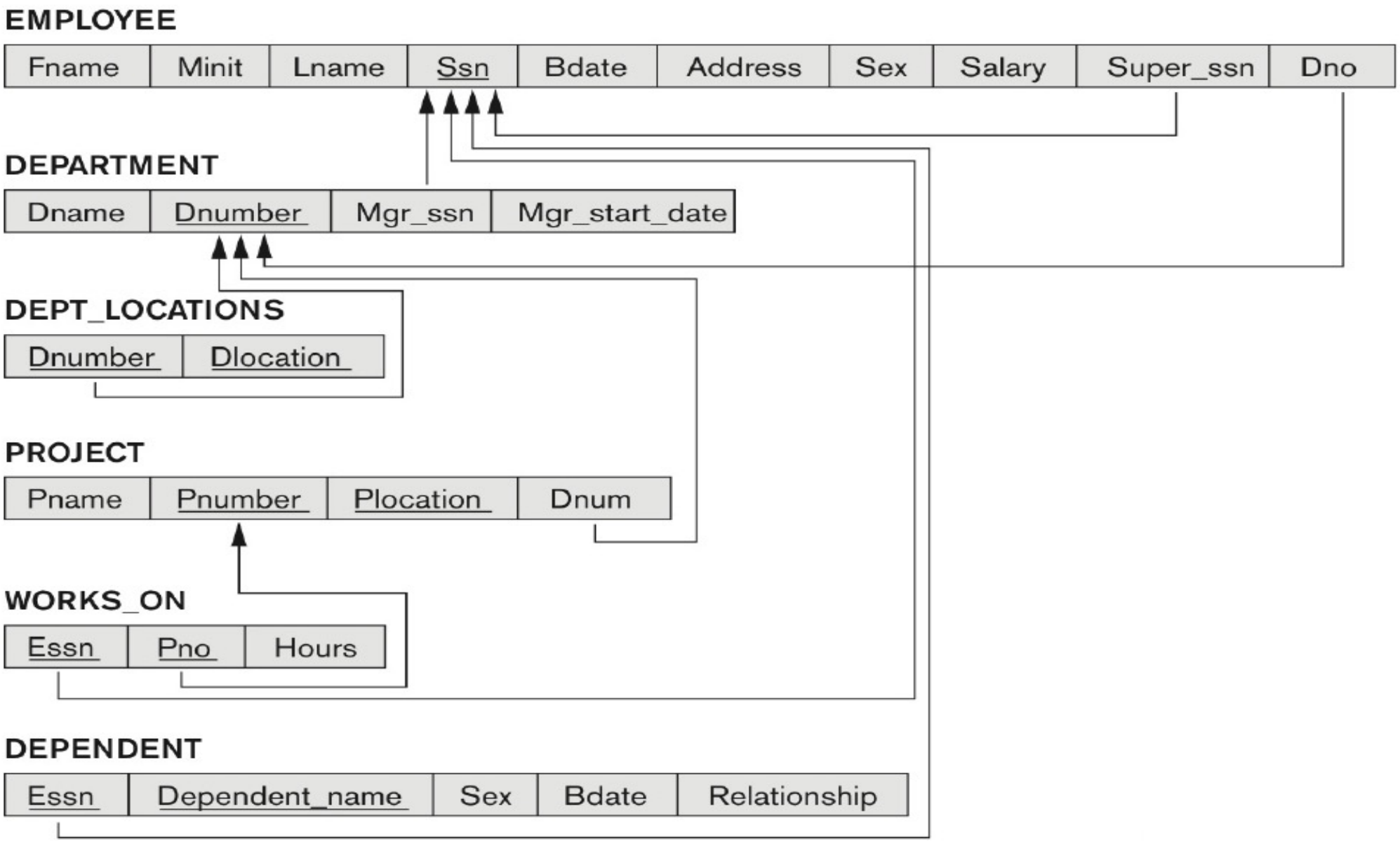
This can be mapped to the relation SUPPLY shown in the relational schema, whose primary key is the combination of the three foreign keys {SNAME, PARTNO, PROJNAME}

Ternary Relationship : Supply





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



Summary of Mapping constructs and constraints

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
Value set	Domain
Key attribute	Primary (or secondary) key

Mapping of Generalization and Specialization Hierarchies to a Relational Schema

Mapping EER Model Constructs to Relations

Step8: Options for Mapping Specialization or Generalization.

- Convert each specialization with m subclasses {S1, S2,….,Sm} and generalized superclass C, where the attributes of C are {k,a1,…an} and k is the (primary) key, into relational schemas using one of the four following options:

✓ Option 8A: Multiple relations-Superclass and subclasses

✓ Option 8B: Multiple relations-Subclass relations only

✓ Option 8C: Single relation with one type attribute

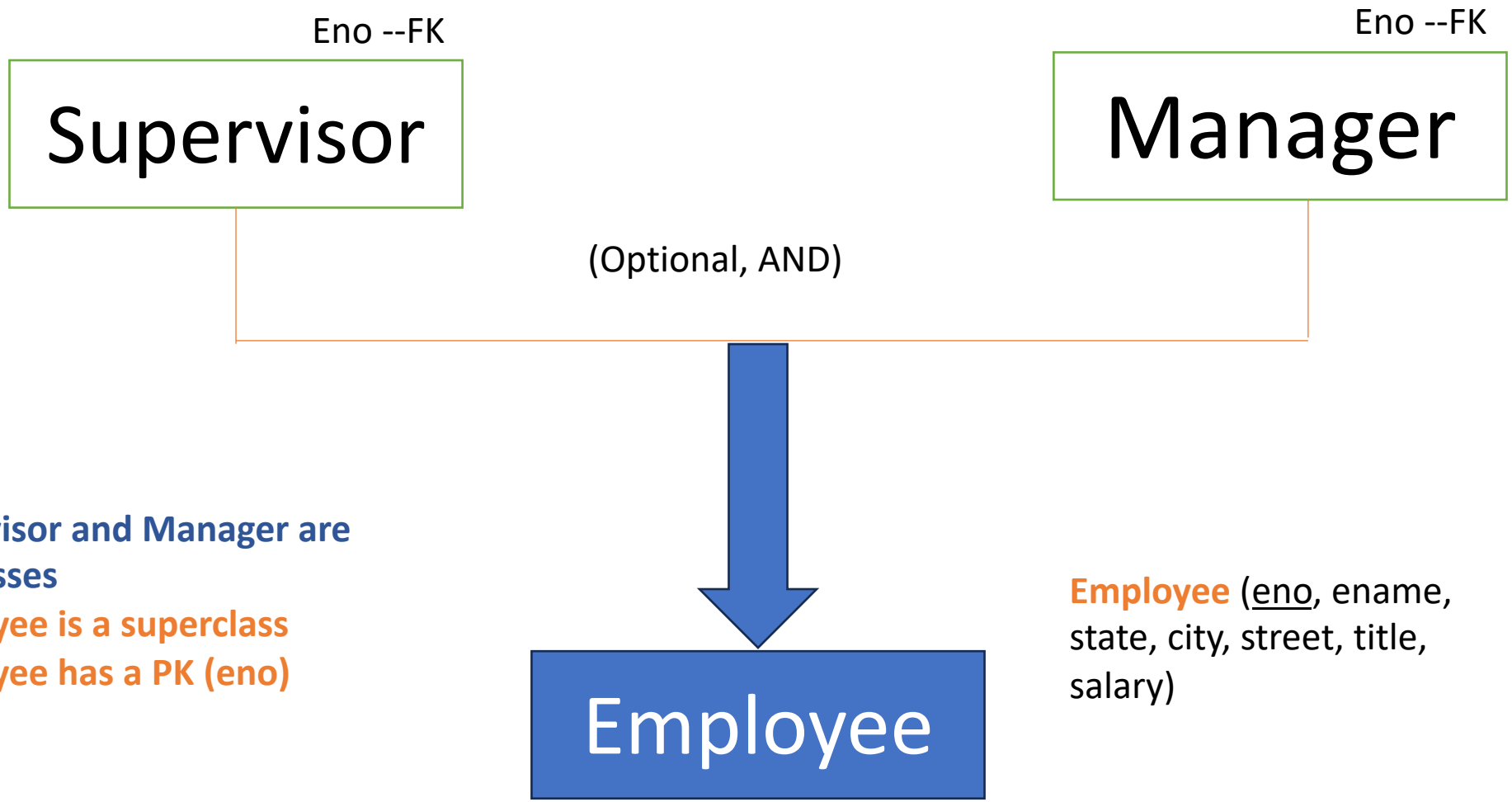
✓ Option 8D: Single relation with multiple type attributes

Mapping EER Model Constructs to Relations

Option 8A: Multiple relations-Superclass and subclasses

Create a relation L for C with attributes Attrs(L) = {k,a1,…an} and PK(L) = k. Create a relation Li for each subclass Si, 1 < i < m, with the attributesAttrs(Li) = {k} U {attributes of Si} and PK(Li)=k. This option works for any specialization (total or partial, disjoint of over-lapping).

Example -



Supervisor and Manager are subclasses
Employee is a superclass
Employee has a PK (eno)

Employee (eno, ename, state, city, street, title, salary)

Option 8B: Multiple relations-Subclass relations only

Create a relation Li for each subclass Si, 1 < i < m, with the attributes Attr(Li) = {attributes of Si} U {k,a1…,an} and PK(Li) = k. This option only works for a specialization whose subclasses are **total(Mandatory Participtionn)**(every entity in the superclass must belong to (at least) one of the subclasses).

Option 8C: Single relation with one type attribute

Create a single relation L **with attributes** Attrs(L) = {k,a1,…an} U {attributes of S1} U…U {attributes of Sm} U {t} and PK(L) = k. The attribute t is called a **type (or discriminating)** attribute **that indicates the subclass to which each tuple belongs**

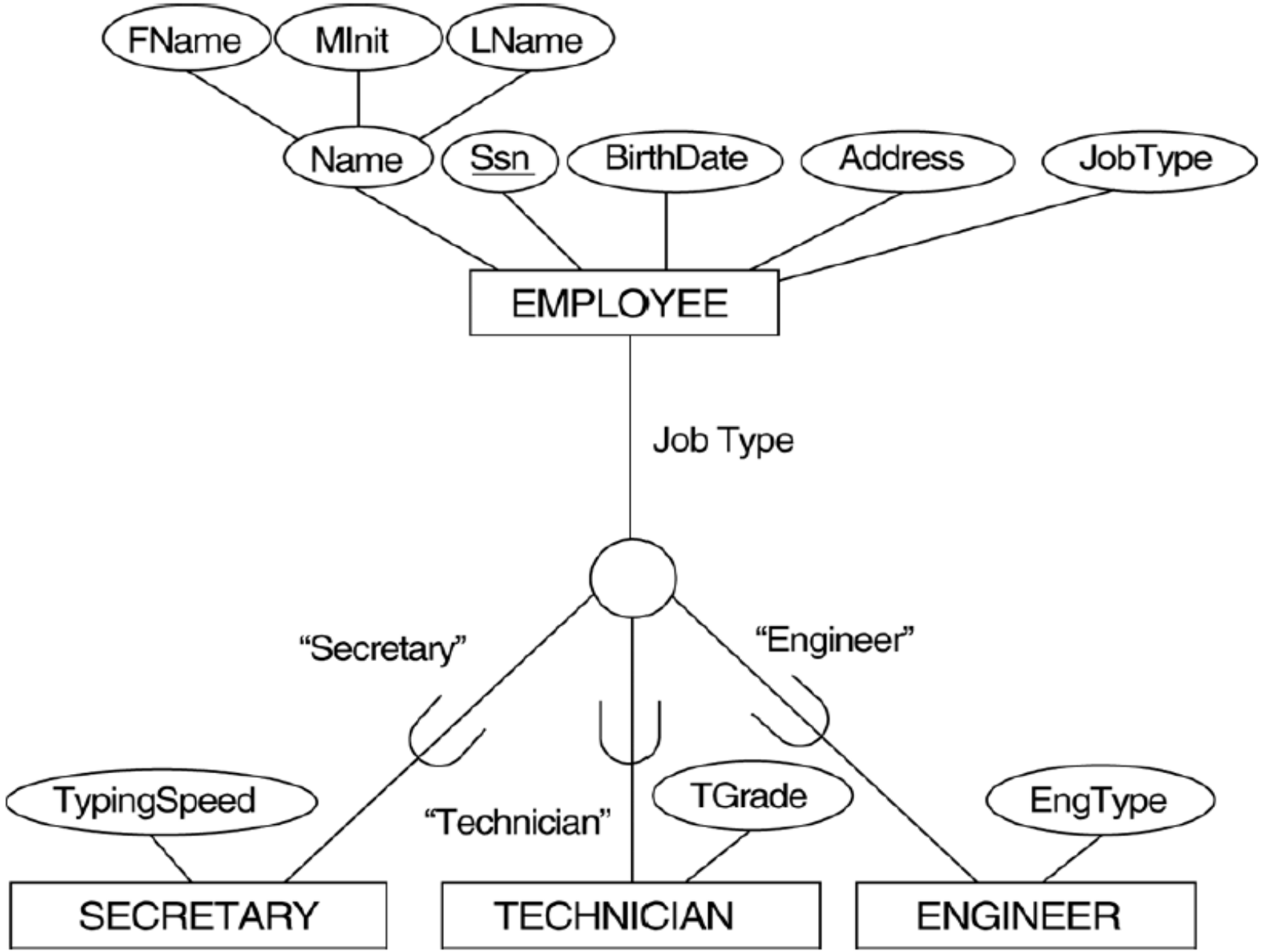
- **works only if subclasses are disjoint**

Option 8D: Single relation with multiple type attributes

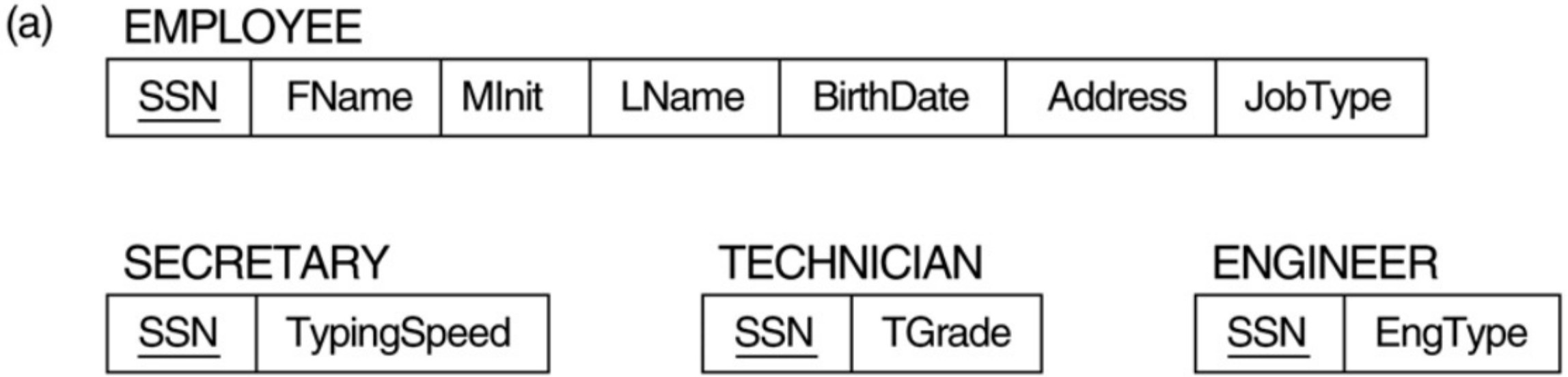
Create a single relation schema L with attributes Attrs(L) = {k,a1,…an} U {attributes of S1} U…U {attributes of Sm} U {t1, t2,…,tm} and PK(L) = k. Each ti, 1 < I < m, is a **Boolean type attribute** indicating whether a tuple belongs to the subclass Si.

- ✓ have a Boolean valued attribute for each subclass – true if in a class otherwise False
- ✓ works if subclasses may be overlapping

EER diagram notation for an attribute-defined specialization on JobType.



Mapping the EER schema in Figure using option 8A

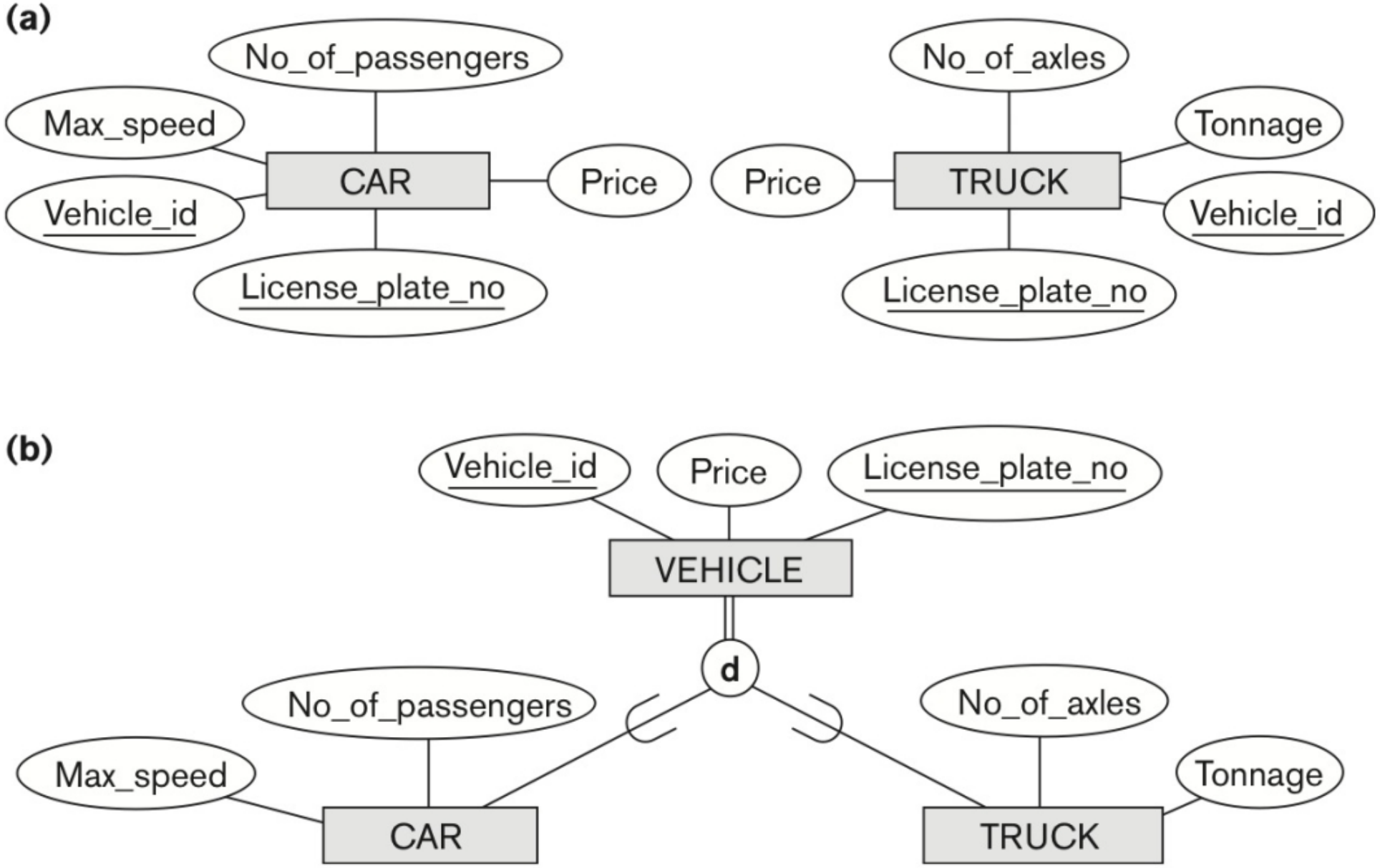


Mapping the EER schema in Figure 4.4 using option 8C

(c) EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType	TypingSpeed	TGrade	

Generalizing CAR and TRUCK into the superclass VEHICLE



Mapping the EER schema in Figure using option 8B.

(b)

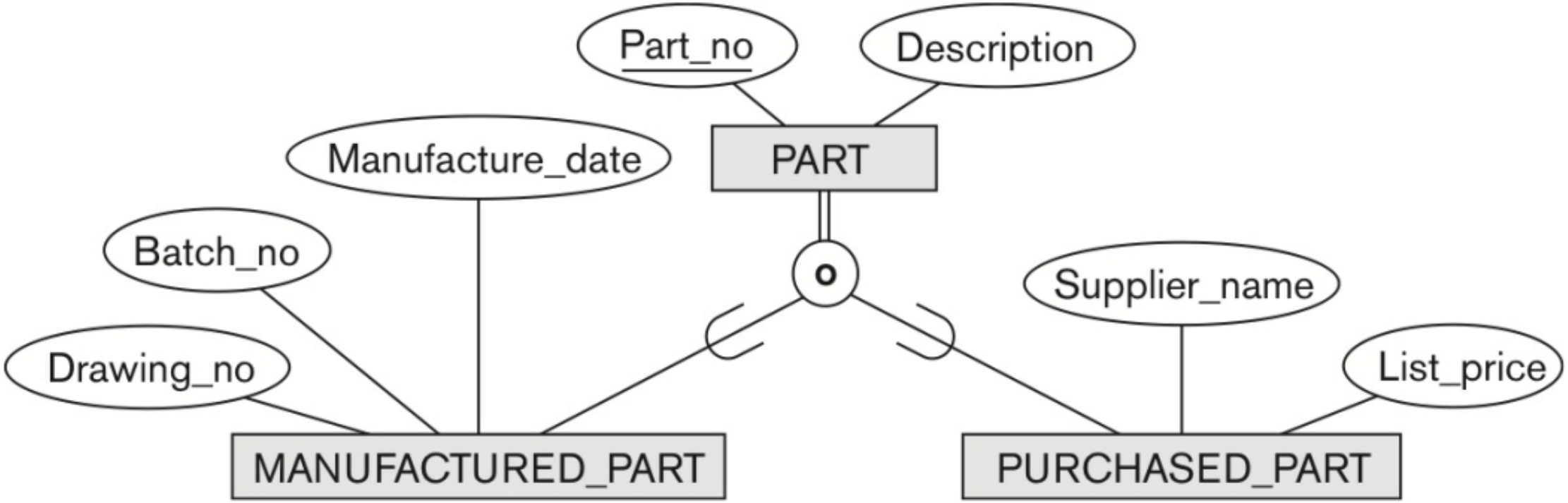
CAR

VehicleId	LicensePlateNo	Price	MaxSpeed	NoOfPassengers

TRUCK

VehicleId	LicensePlateNo	Price	NoOfAxles	

An overlapping (non-disjoint) specialization



(d) PART

PartNo	Description	MFlag	DrawingNo	ManufactureDate	BatchNo	PFlag	SupplierName	ListPrice

Mapping Figure using option 8D with Boolean type fields Mflag and Pflag.

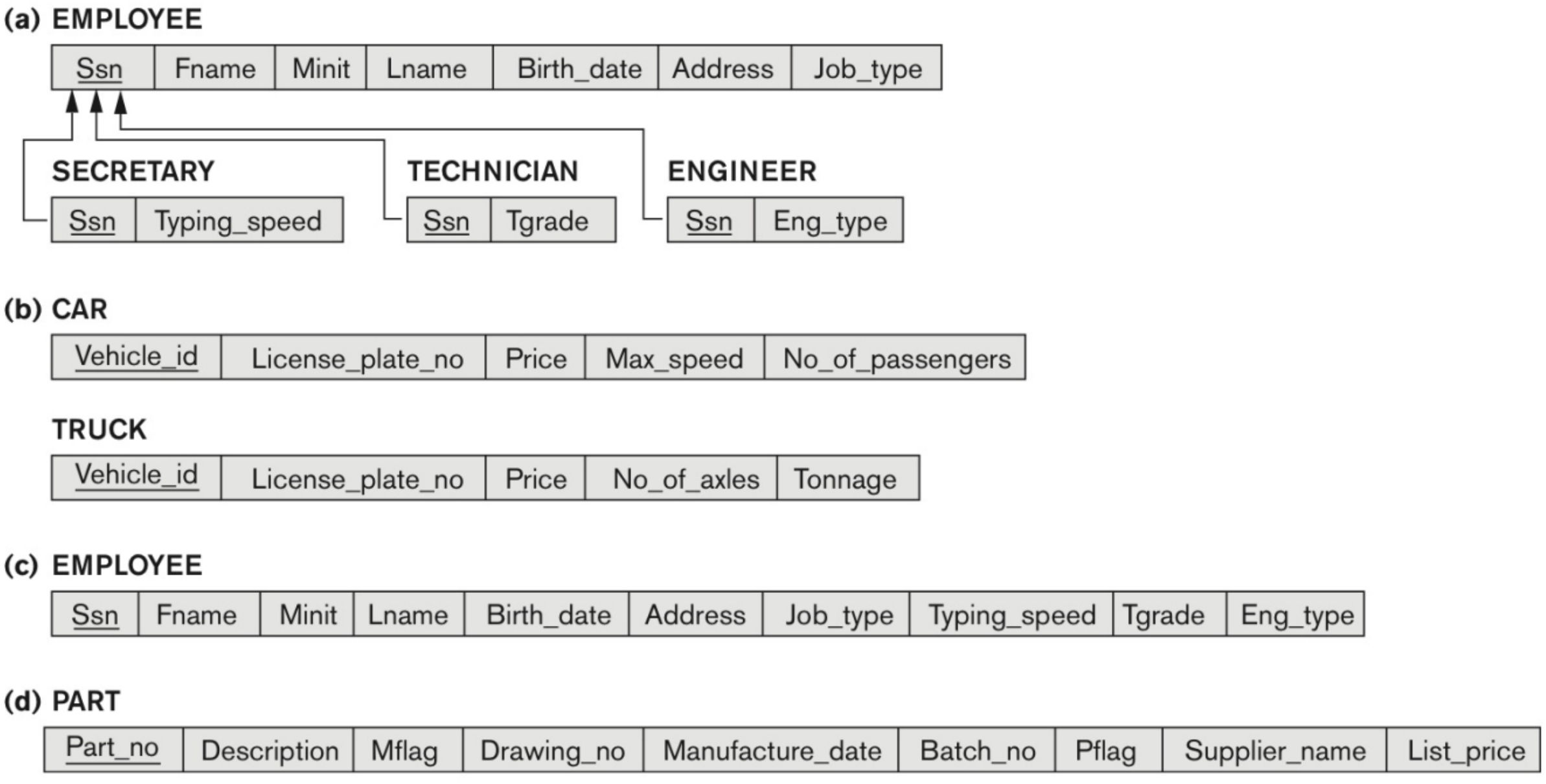
(d) PART

PartNo	Description	MFlag	DrawingNo	ManufactureDate	BatchNo	PFlag	SupplierName	ListPrice

Different Options for Mapping Generalization Hierarchies

- **Next Slide :Figure 9.5** Options for mapping specialization or generalization. (a) Mapping the EER schema in Figure 4.4 using option 8A.
- (b) Mapping the EER schema in Figure 4.3(b) using option 8B.
- (c) Mapping the EER schema in Figure 4.4 using option 8C.
- (d) Mapping Figure 4.5 using option 8D with Boolean type fields Mflag and Pflag.

Fig. 9.5: Different Options for Mapping Generalization Hierarchies - summary



Mapping EER Model Constructs to Relations (contd.)

Mapping of Shared Subclasses (Multiple Inheritance)

- ✓A shared subclass, such as STUDENT_ASSISTANT, is a subclass of several classes, indicating multiple inheritance. These classes **must all have the same key attribute; otherwise, the shared subclass would be modeled as a category**.
- ✓We can apply any of the options discussed in Step 8 to a shared subclass, subject to the restriction discussed in Step 8 of the mapping algorithm. Below both 8C and 8D are used for the shared class STUDENT_ASSISTANT

A specialization lattice with multiple inheritance for a UNIVERSITY database.

Mapping of shared subclasses – Multiple Inheritance.

A shared subclass, STUDENT_ASSISTANT is a subclass of several classes, indicating multiple inheritance

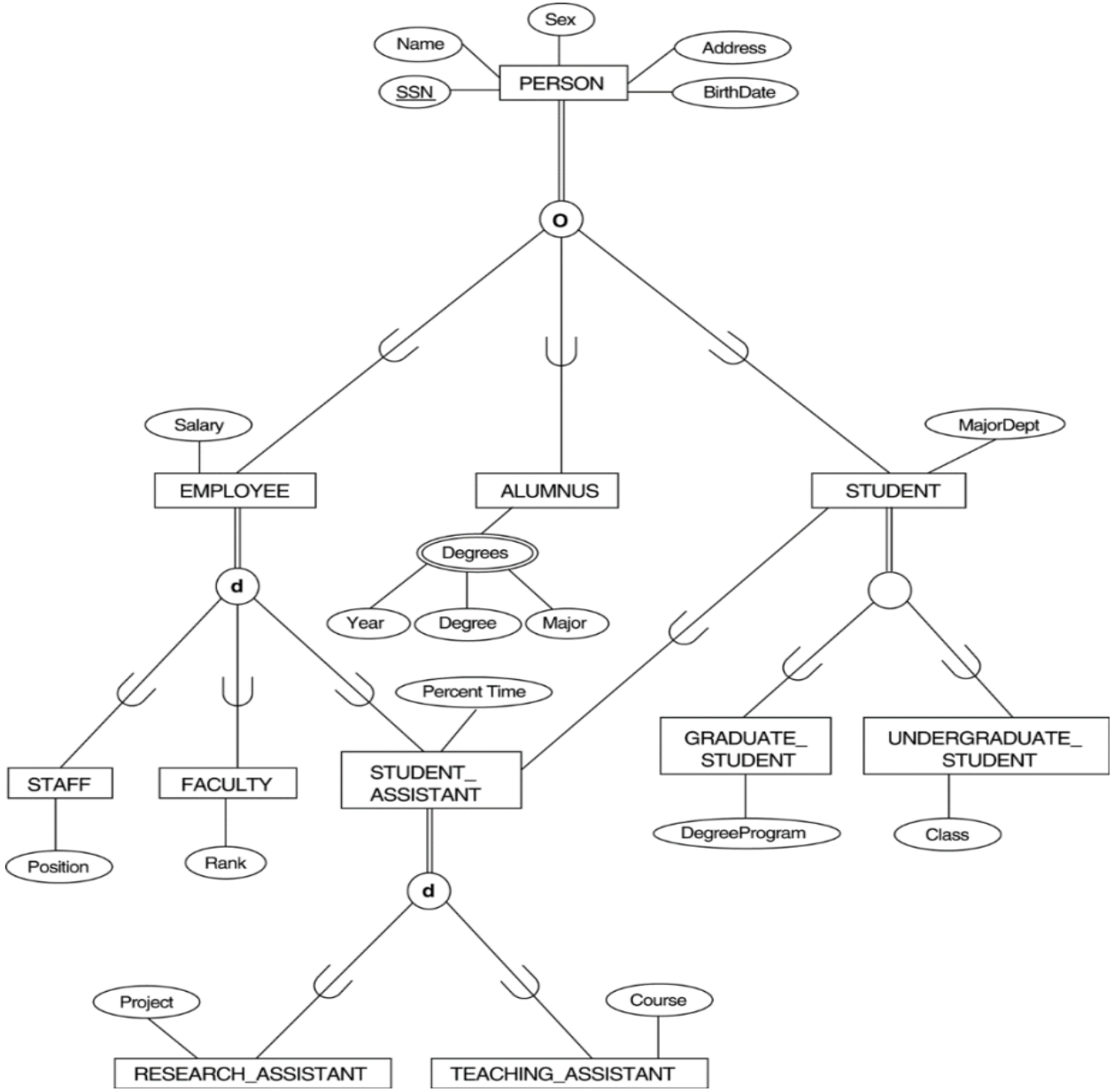
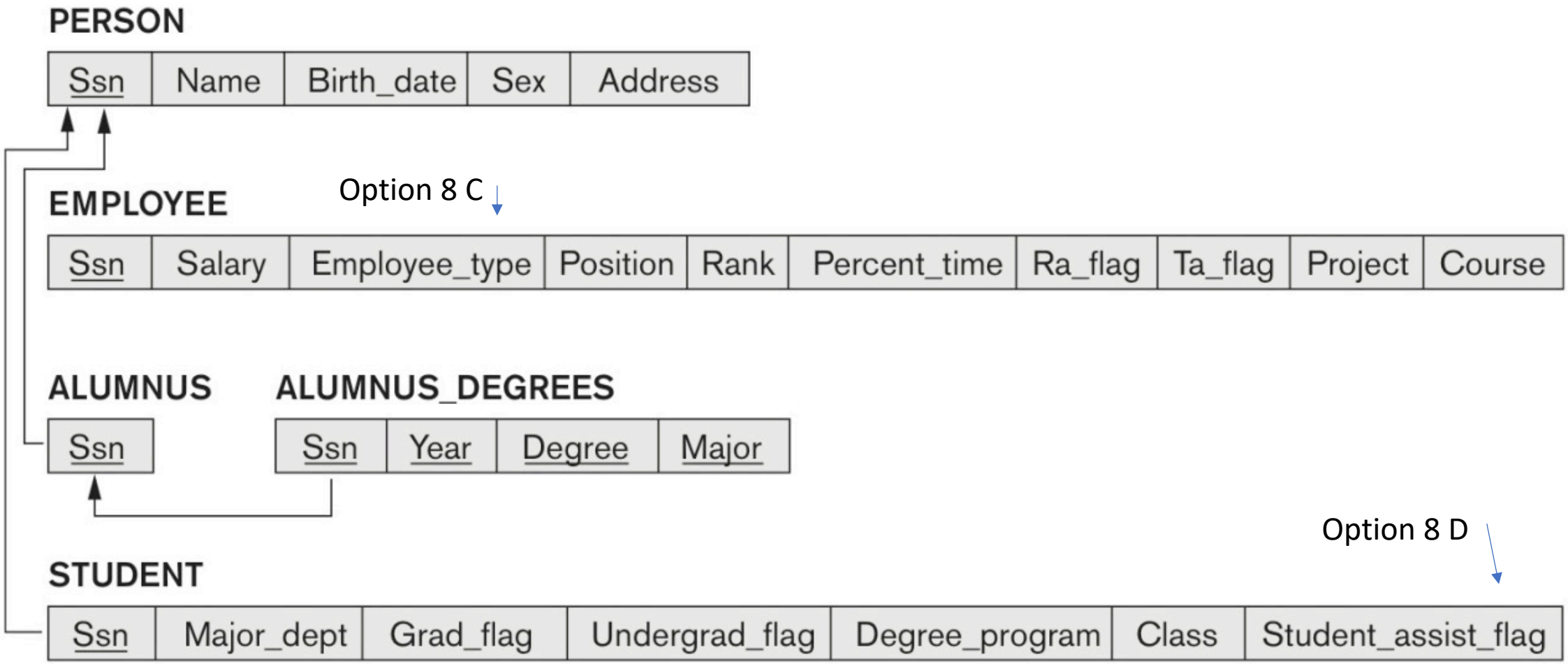


FIGURE 9.6 Mapping the EER specialization lattice in Figure 4.7 using multiple options.

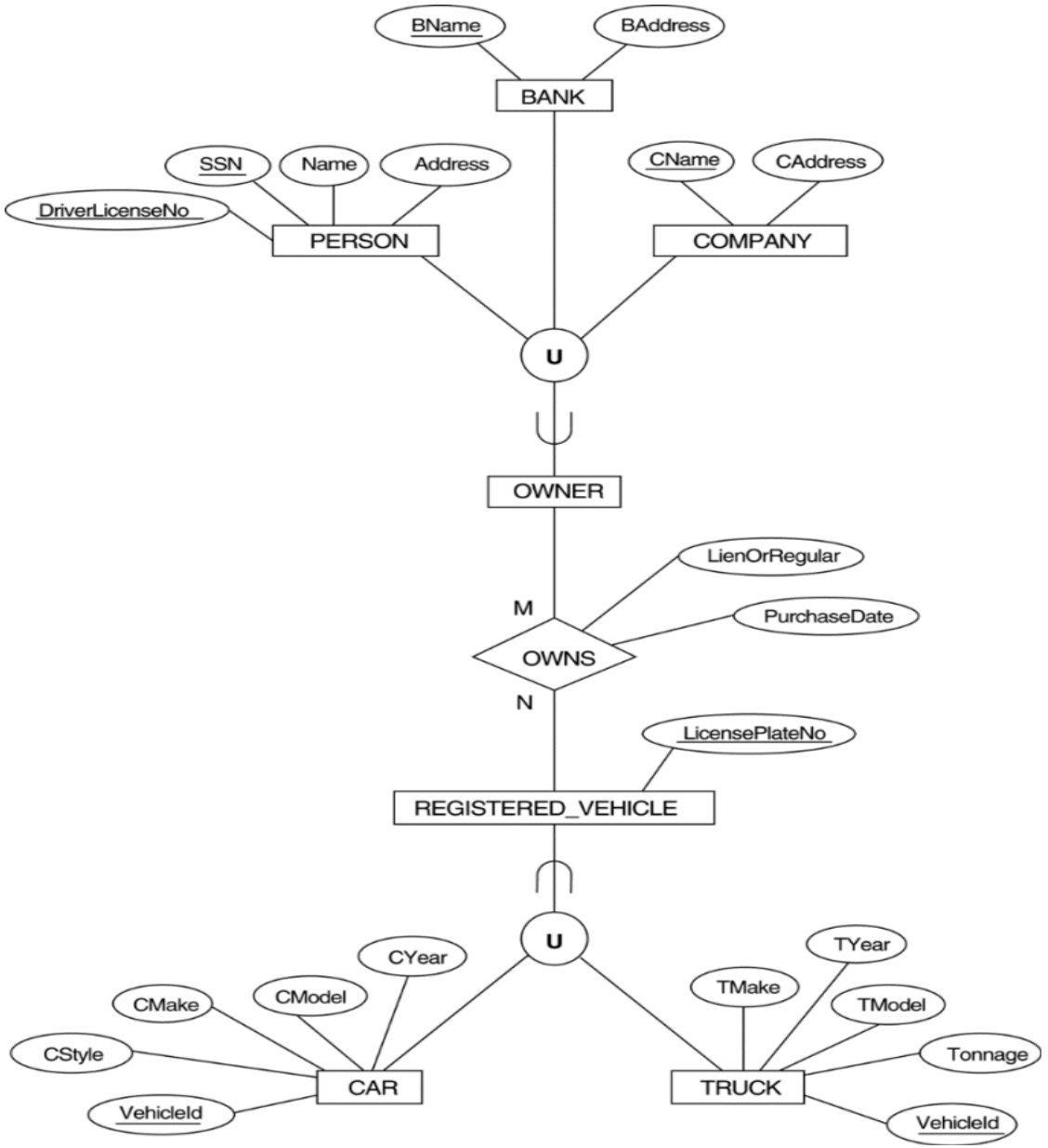


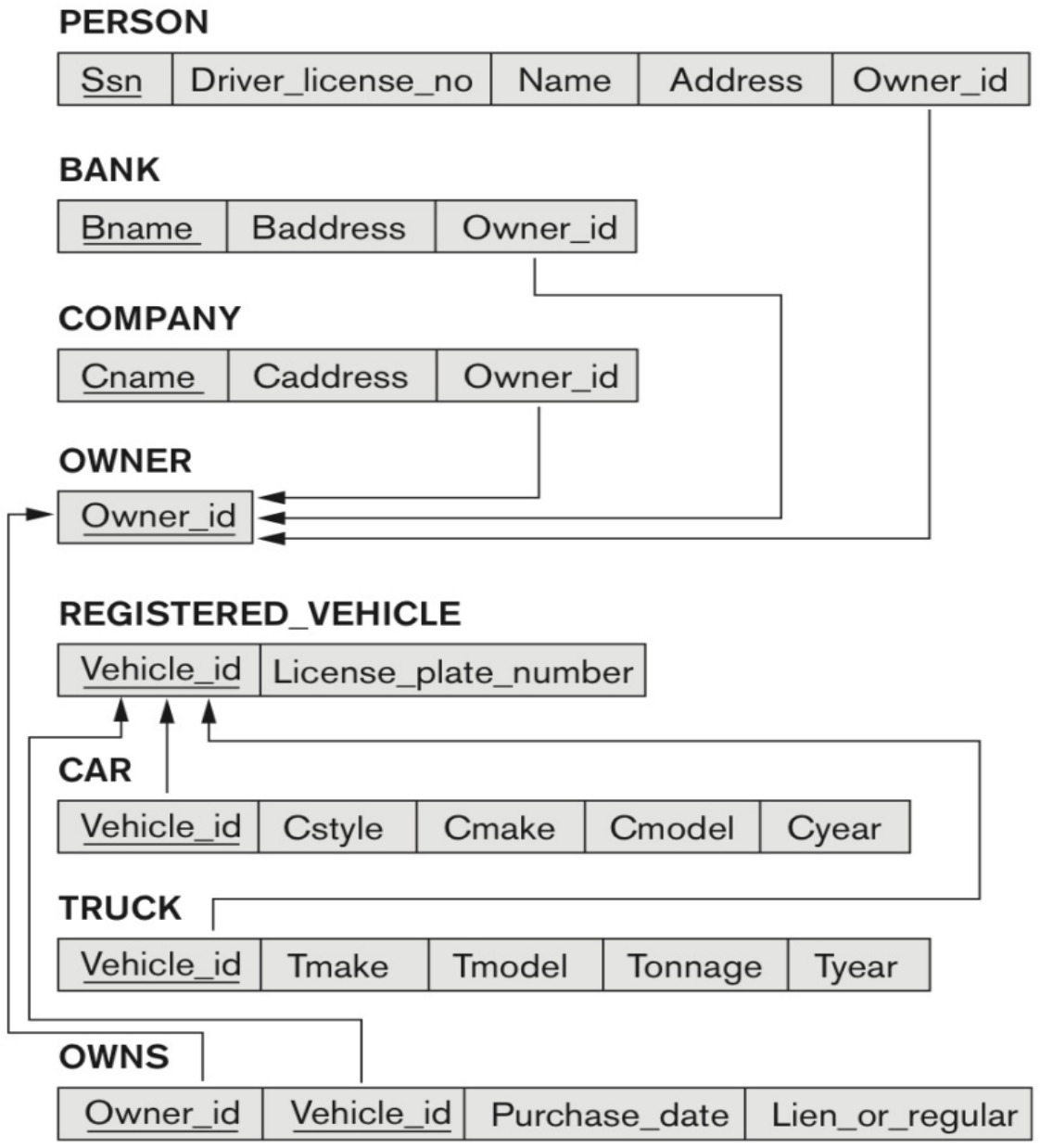
Mapping EER Model Constructs to Relations (contd.)

Step 9: Mapping of Union Types (Categories).

- ✓ For mapping a category whose defining superclass have different keys, it is customary to specify a new key attribute, called a **surrogate key**, when creating a relation to correspond to the category.
- ✓ In the example next, we can create a relation OWNER to correspond to the OWNER category and include any attributes of the category in this relation. The primary key of the OWNER relation is the surrogate key, which we called OwnerId.

FIGURE 4.8 Two categories (union types): OWNER and REGISTERED_VEHICLE.





Mapping Exercise-1 Map this schema into a set of relations

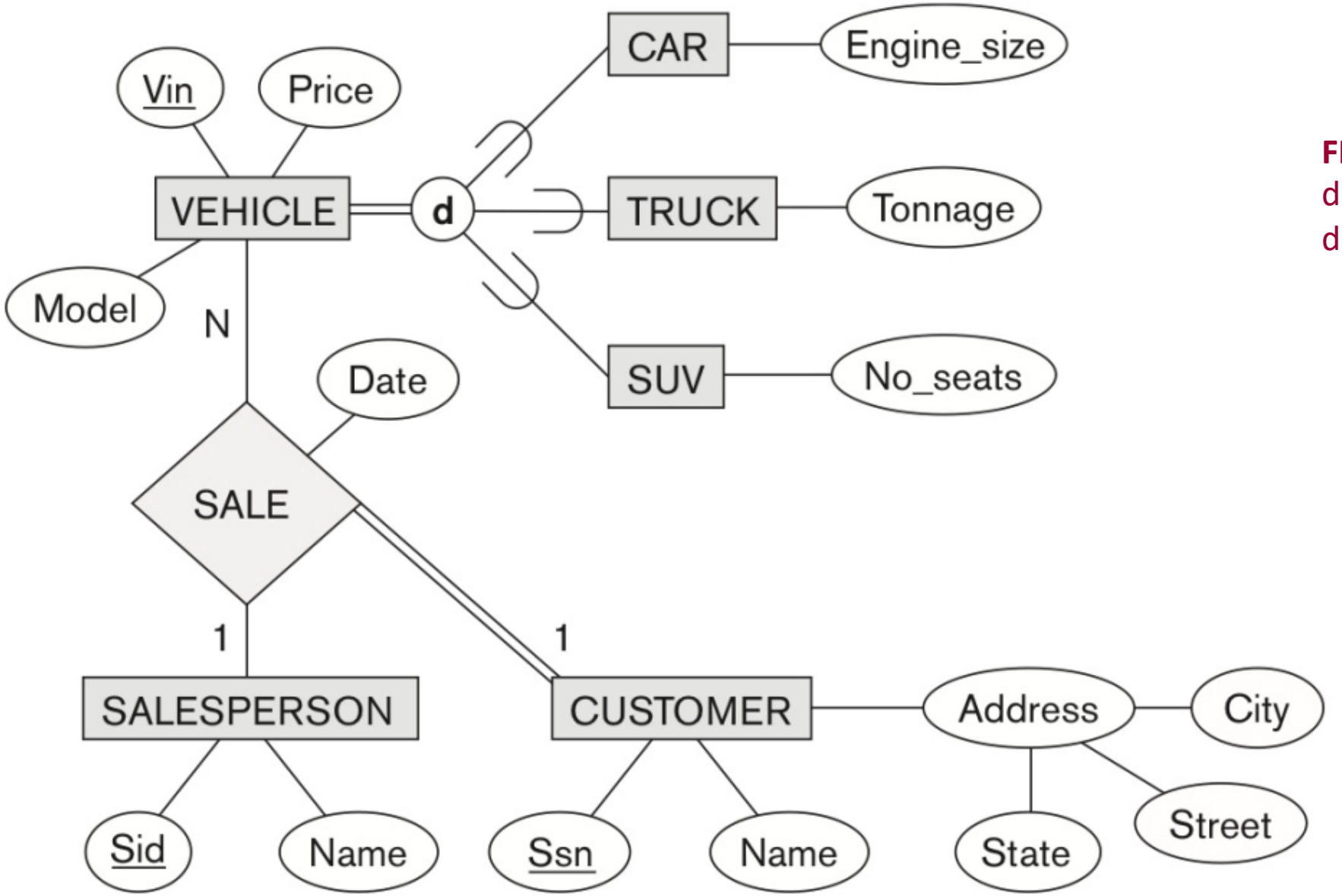


FIGURE 9.9 EER diagram for a car dealer

Chapter Summary

ER-to-Relational Mapping Algorithm

- ✓Step 1: Mapping of Regular Entity Types
- ✓Step 2: Mapping of Weak Entity Types
- ✓ Step 3: Mapping of Binary 1:1 Relation Types
- ✓ Step 4: Mapping of Binary 1:N Relationship Types.
- ✓Step 5: Mapping of Binary M:N Relationship Types.
- ✓ Step 6: Mapping of Multivalued attributes.
- ✓ Step 7: Mapping of N-ary Relationship Types.

Mapping EER Model Constructs to Relations

- ✓ Step 8: Options for Mapping Specialization or Generalization.
- ✓ Step 9: Mapping of Union Types (Categories).