204222 - Fundamentals of Database Systems

Chapter 9

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

Adapted for 204222

by Areerat Trongratsameethong

Addison-Wesley
is an imprint of
PEARSON

Chapter 9 Outline

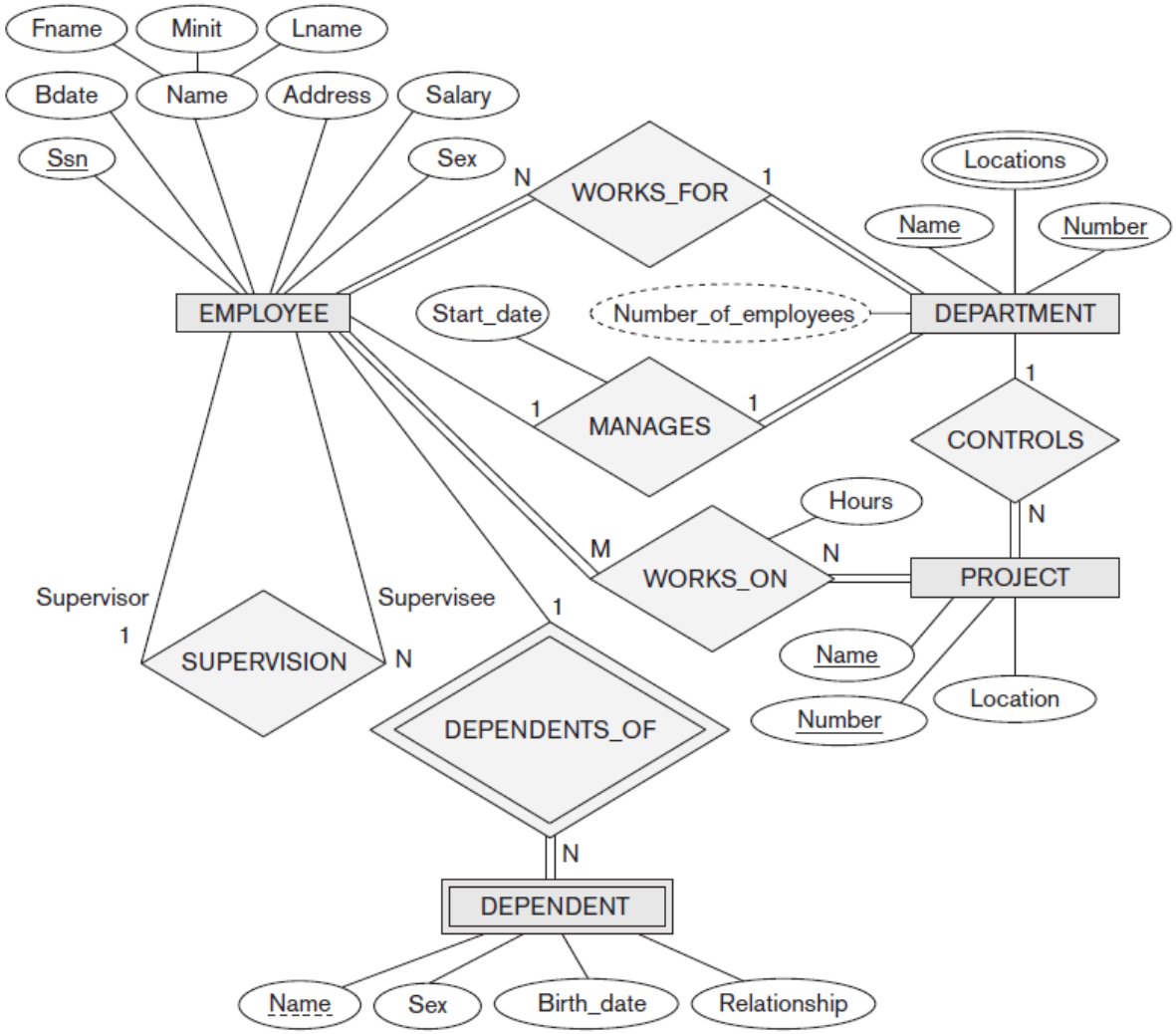
- Relational Database Design Using ER-to-Relational Mapping
- Relational Database Design Using EER-to-Relational Mapping

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

- **Design a relational database schema**
  - Based on a conceptual schema design
- Seven-step algorithm to convert the basic ER model constructs into relations

Relational Database Design Using ER-to-Relational Mapping

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



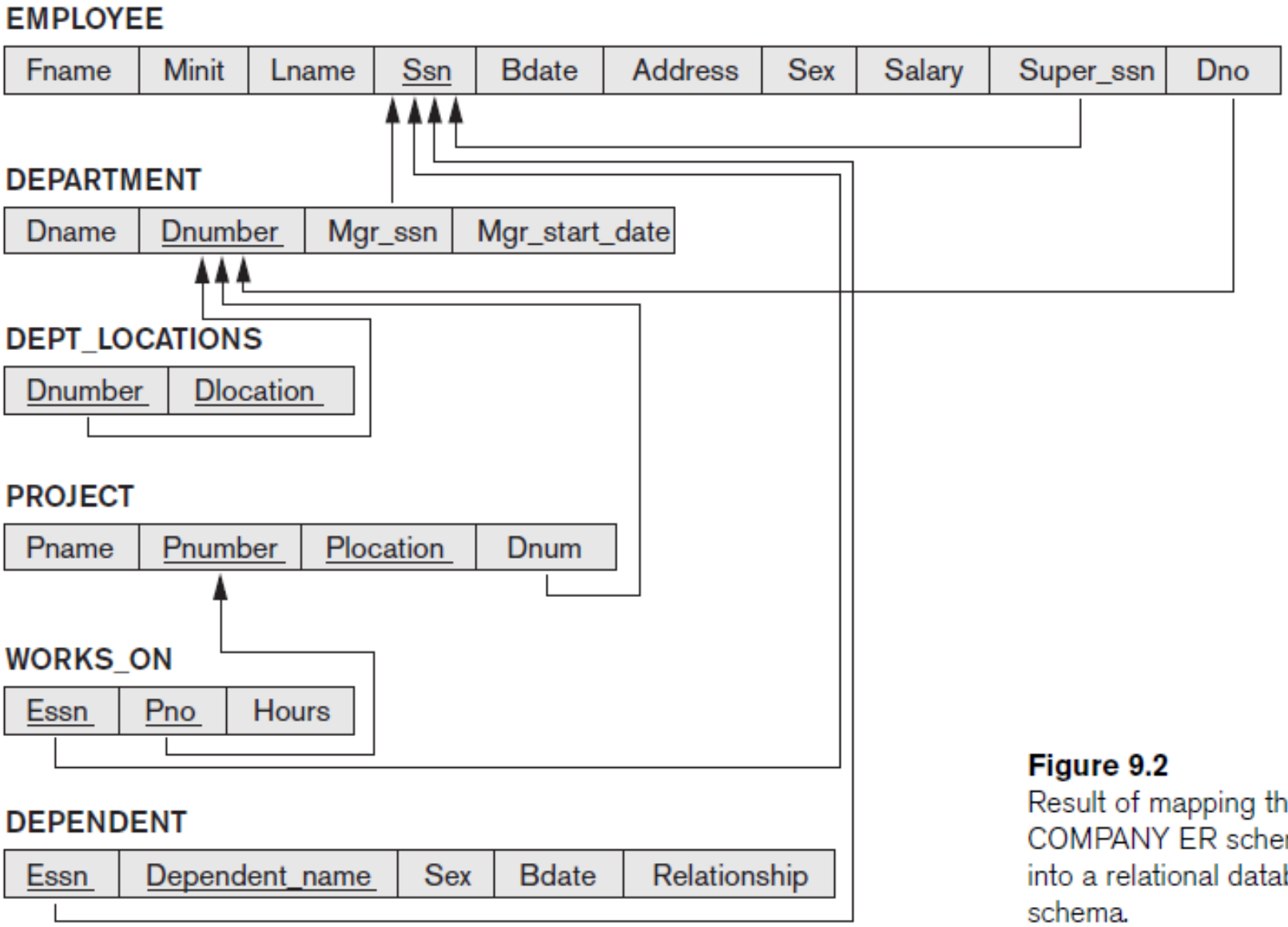


Figure 9.2
Result of mapping the COMPANY ER schema into a relational database schema.

ER-to-Relational Mapping Algorithm

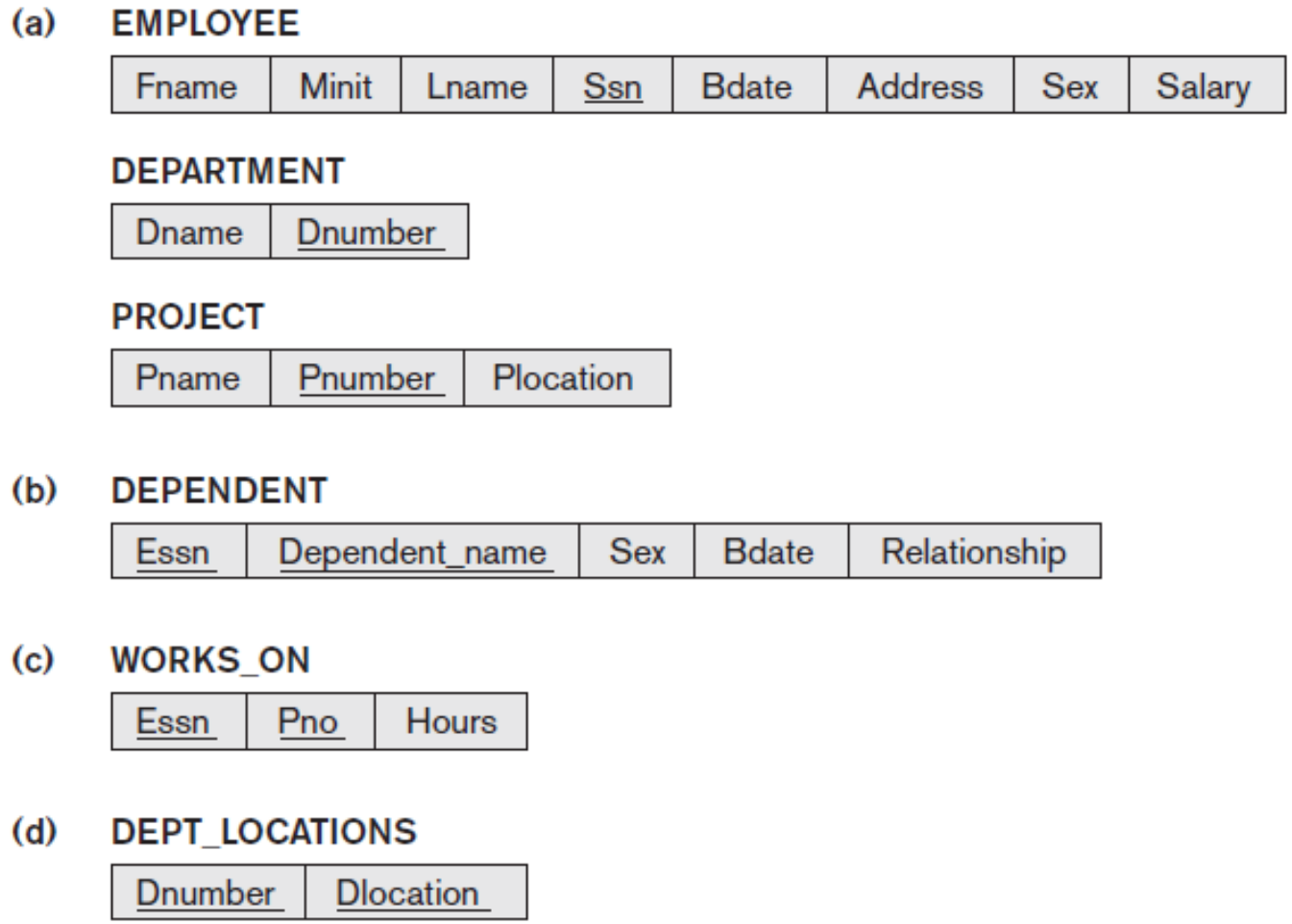
- COMPANY database example
  - Assume that the mapping will create tables with simple single-valued attributes
- Step 1: Mapping of Regular Entity Types
  - For each regular entity type, create a relation *R* that includes all the simple attributes of *E*
  - Called **entity relations**
    - Each tuple represents an entity instance

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 primary key attribute of owner as foreign key attributes of *R*

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

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 (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*
  - Possible approaches:
    - **Foreign key approach:** e.g. Employee manages department (total participation : every department has a manager, PK Employee → FK in Department, otherwise a lot of NULL in Employee)
    - **Merged relationship approach:** when both entities are total participation
    - **Cross reference or relationship relation approach:**

      create third relation and keeps PK of both entities, disadvantage: have one more table and join is needed to join data from both

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

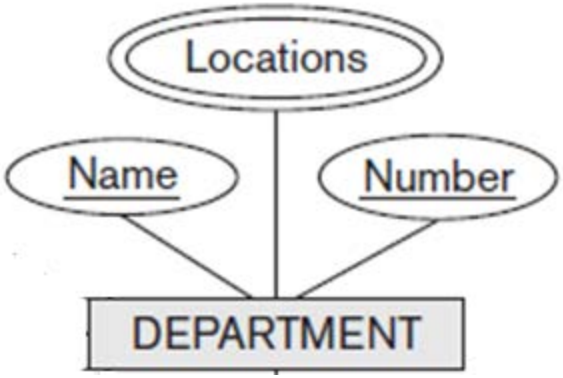
- Step 4: Mapping of Binary 1:*N* Relationship Types
  - For each regular binary **1:*N* relationship type**
    - Identify relation that represents participating entity type at *N*-side of relationship type
    - Include primary key of other entity type as foreign key in *S*
    - Include simple attributes of 1:*N* relationship type as attributes of *S*
  - Alternative approach
    - Use the **relationship relation** (cross-reference) option as in the third option for binary 1:1 relationships

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

- Step 5: Mapping of Binary *M*:*N* Relationship Types
  - For each binary ***M*:*N* relationship type**
    - Create a new relation *S*
    - Include primary key of participating entity types as foreign key attributes in *S*
    - Include any simple attributes of *M*:*N* relationship type

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

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



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

- 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
    - Include any simple attributes as attributes

Discussion and Summary of Mapping for ER Model Constructs

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

Discussion and Summary of Mapping for ER Model Constructs (cont'd.)

- In a relational schema relationship, types are not represented explicitly
  - Represented by having two attributes *A* and *B*: one a primary key and the other a foreign key

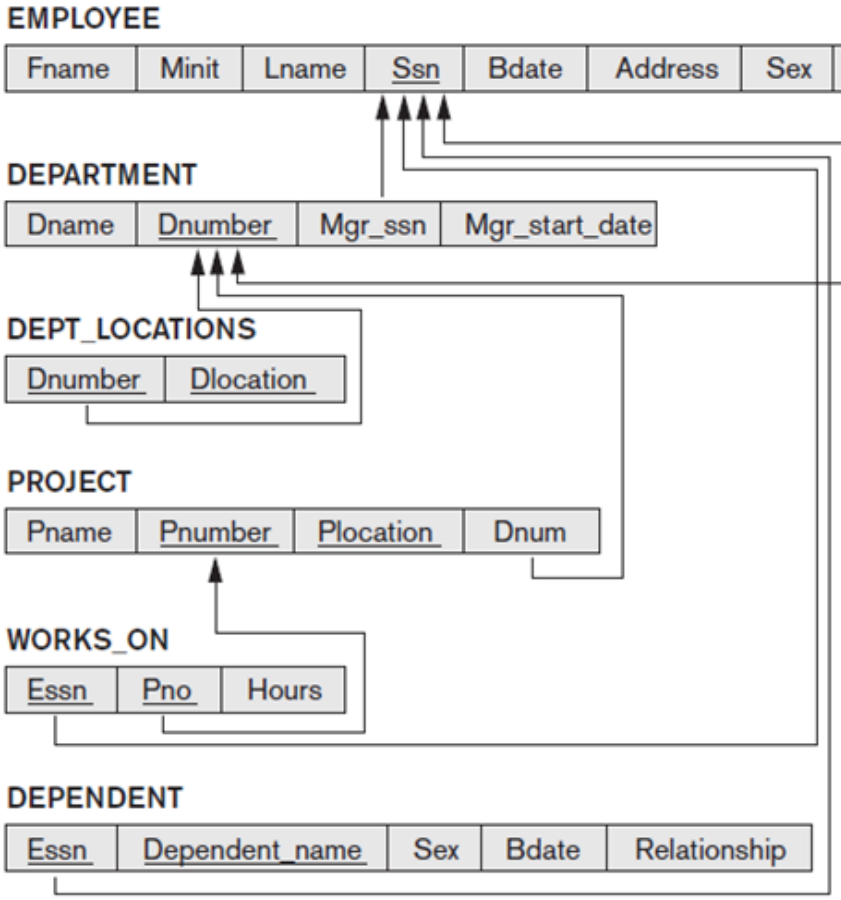
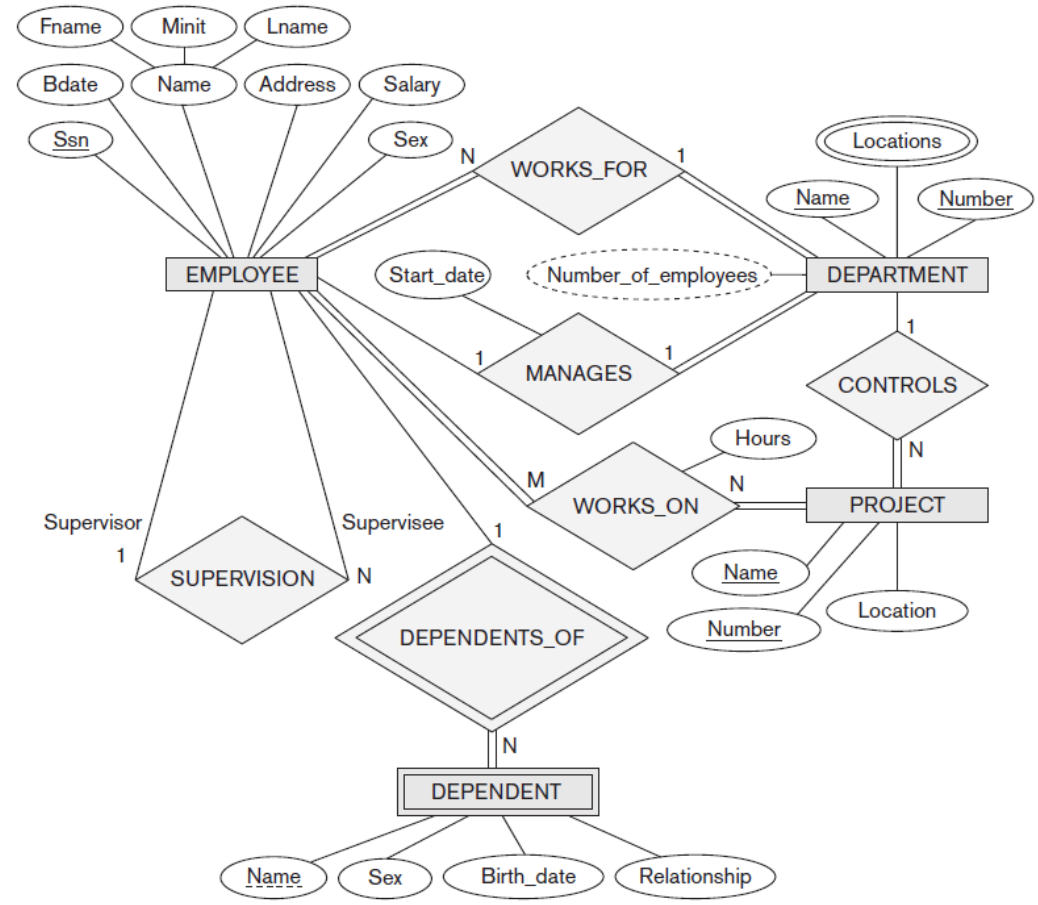


Figure 9.1
The ER conceptual schema diagram for the COMPANY database.



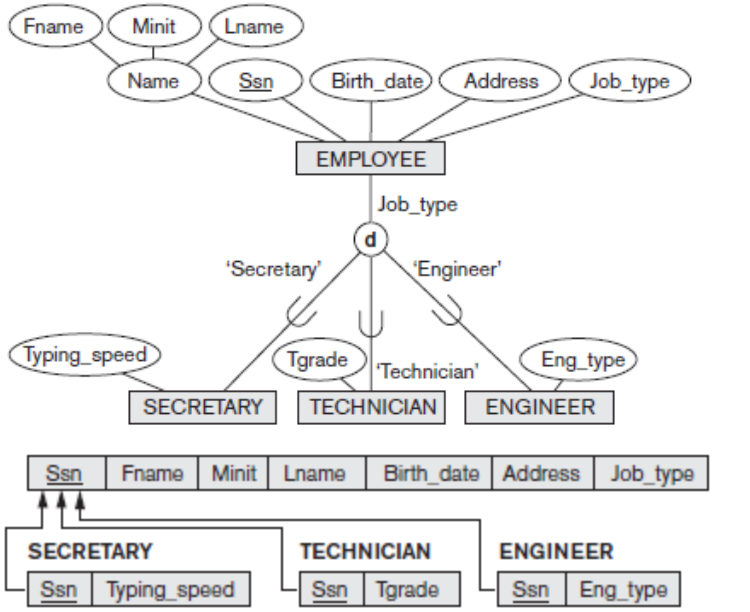
Mapping EER Model Constructs to Relations

- Extending ER-to-relational mapping algorithm
- **Step 8:** 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 relation* schemas using one of the following 4 options: 8A – 8D

Mapping of Specialization or Generalization

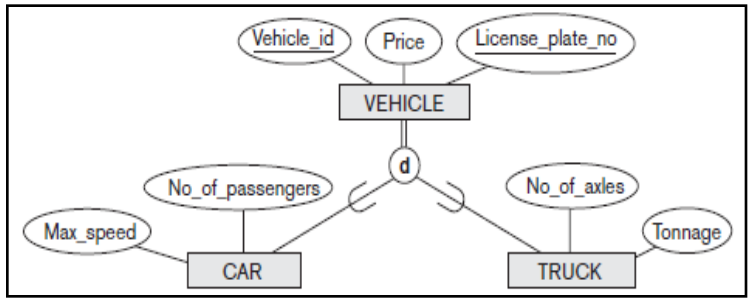
Option 8A: Multiple relations superclass and subclasses.

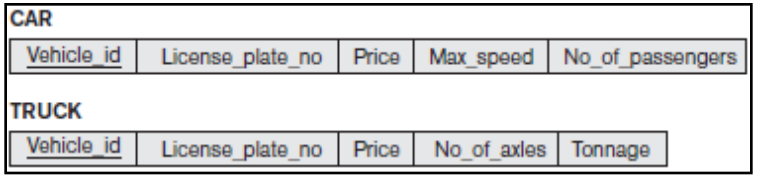
Create a relation *L for C with attributes Attrs(L) = {k, a1, ..., an} and PK(L) = k. Create a* relation *Li for each subclass Si,1 ≤ i ≤ m, with the attributes Attrs(Li) = {k} ∪* {attributes of *Si} and PK(Li) = k. This option works for any specialization* (total or partial, disjoint or overlapping).



Option 8B: Multiple relations—subclass relations only.

Create a relation *Li for each subclass Si, 1 ≤ i ≤ m, with the attributes Attrs(Li) = {attributes of Si} ∪ {k, a1, ..., an} and PK(Li) = k. This option only works for a specialization* whose subclasses are *total (every entity in the superclass must belong to (at* least) one of the subclasses). Additionally, it is only recommended if the specialization has the *disjointedness constraint. If the specialization* is *overlapping, the same entity may be duplicated in several relations.*



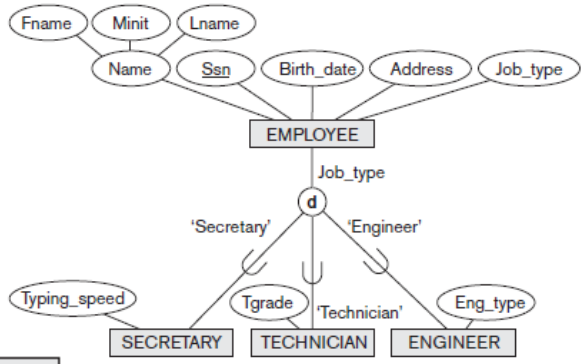


Mapping of Specialization or Generalization

Option 8C: Single relation with one type attribute

Create a single relation *L with attributes*
Attrs(L) = {k, a1, ..., an} ∪ {attributes of S1} ∪ ... ∪ {attributes of *Sm} ∪ {t} and PK(L) = k.*

The attribute t is called a ***type (or* discriminating) attribute whose value indicates the subclass to which each*** tuple belongs, if any. This option works only for a specialization whose subclasses are *disjoint, and has the potential for generating many NULL values if* many specific attributes exist in the subclasses.

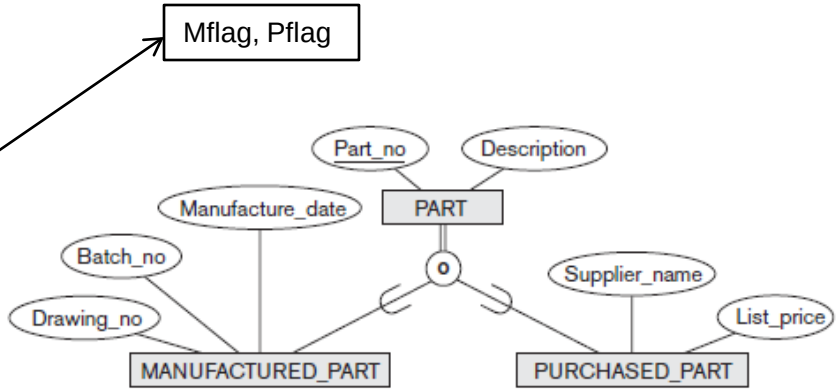


EMPLOYEE

Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type

Option 8D: Single relation with multiple type attributes

Create a single relation schema *L with attributes*
Attrs(L) = {k, a1, ..., an} ∪ {attributes of S1} ∪ ... ∪ {attributes of *Sm} ∪ {t1, t2, ..., tm} and PK(L) = k.*
Each ti, 1 ≤ i ≤ m, is **a Boolean type attribute indicating whether a tuple belongs to subclass *Si*.** This option is used for a specialization whose subclasses are *overlapping (but* will also work for a disjoint specialization).



PART

Part_no	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price

Mapping of Shared Subclasses (Multiple Inheritance)

- Apply any of the options discussed in step 8 to a shared subclass

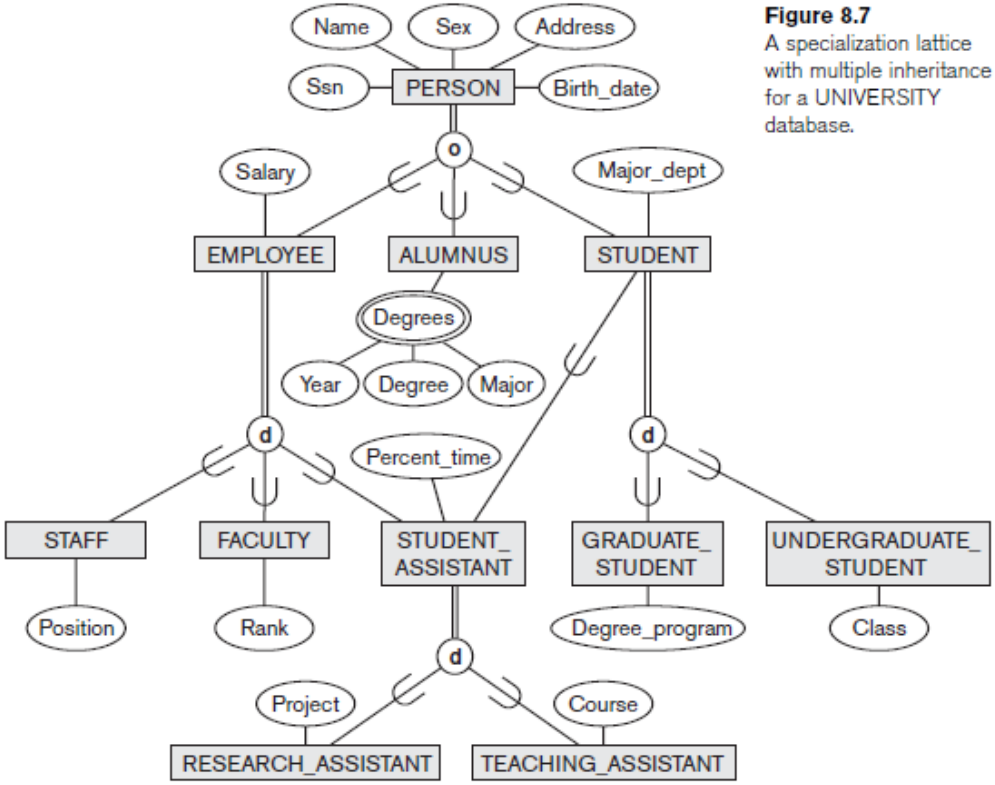


Figure 8.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.

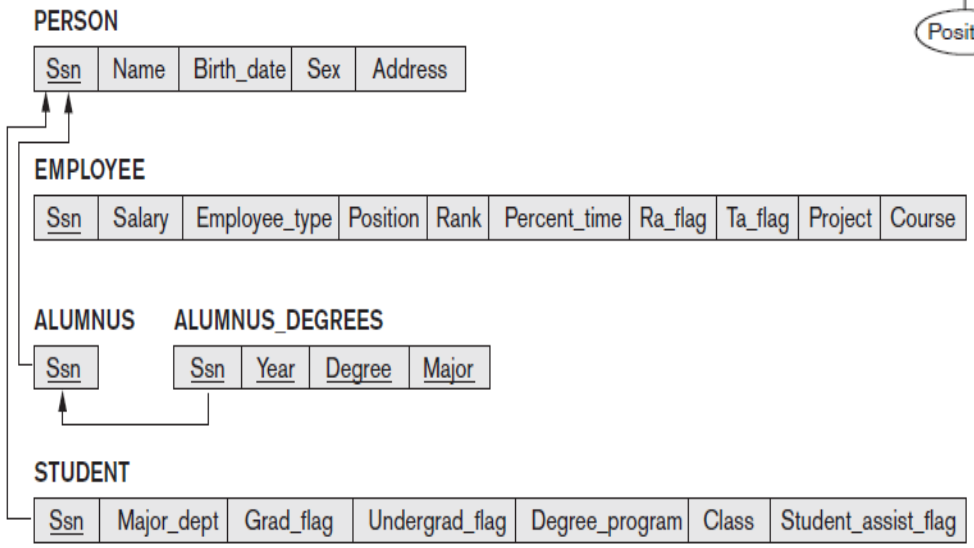


Figure 9.6
Mapping the EER specialization lattice in Figure 8.8 using multiple options.

Mapping of Categories (Union Types)

- Step 9: Mapping of Union Types (Categories)
  - Defining superclasses have different keys
  - Specify a new key attribute
    - **Surrogate key**

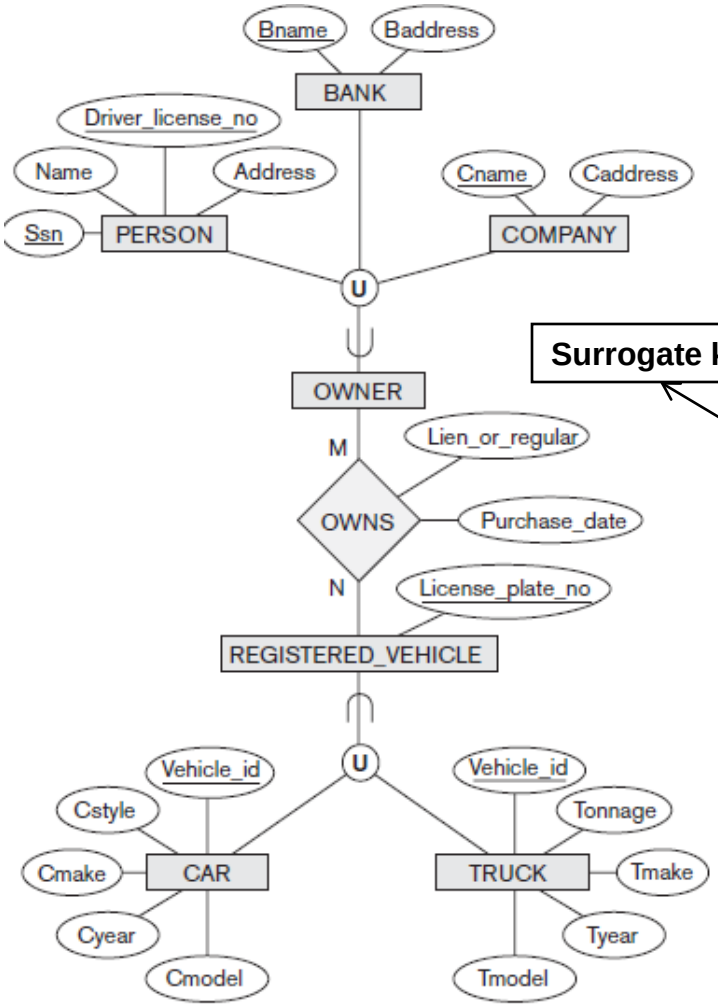


Figure 8.8
Two categories (union types): OWNER and REGISTERED_VEHICLE.

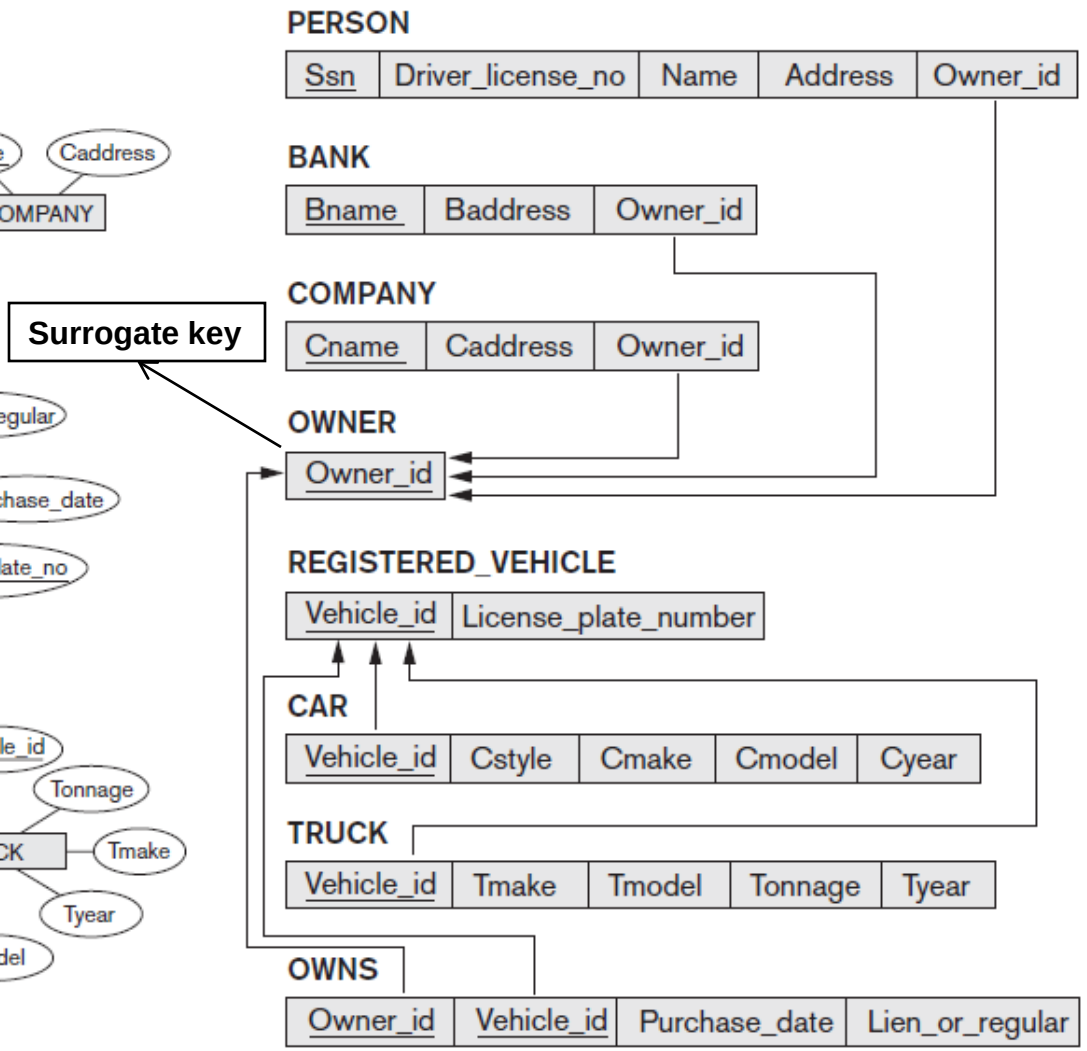


Figure 9.7
Mapping the EER categories (union types) in Figure 8.8 to relations.

Summary

- Map conceptual schema design in the ER model to a relational database schema
  - Algorithm for ER-to-relational mapping
  - Illustrated by examples from the COMPANY database
- Include additional steps in the algorithm for mapping constructs from EER model into relational model