# RELATIONAL DATABASE DESIGN VIA ER MODELLING

CHAPTER 9 (6/E)

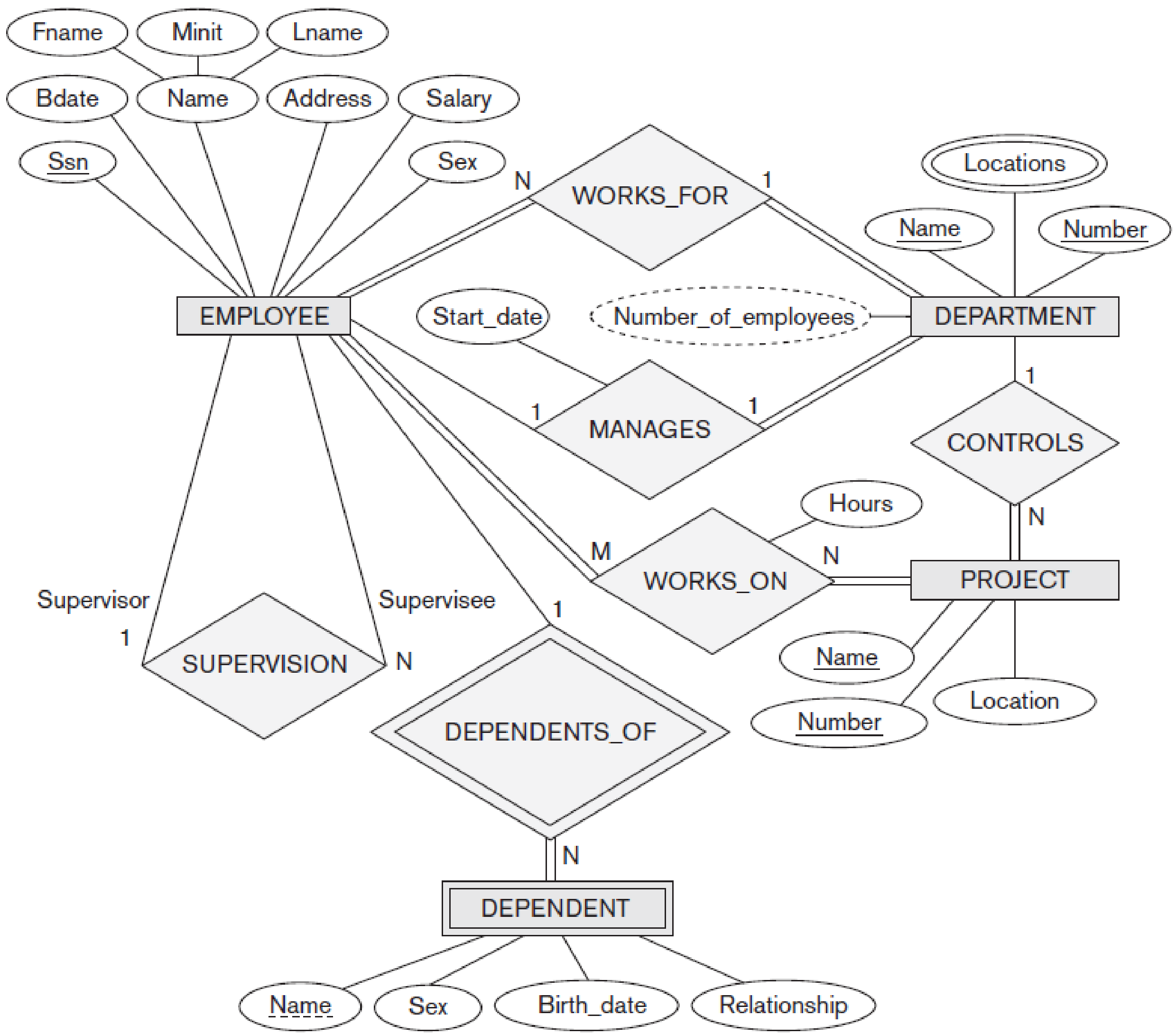
CHAPTER 7 (5/E)

# LECTURE OUTLINE

- Relational Database Design Using ER-to-Relational Mapping
  - Algorithm to convert the basic ER model constructs into relations
- Mapping EER Model Constructs to Relations
  - Additional steps for EER model

# RECALL (BASIC) ER DIAGRAM

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



# END GOAL: RELATIONAL MODEL

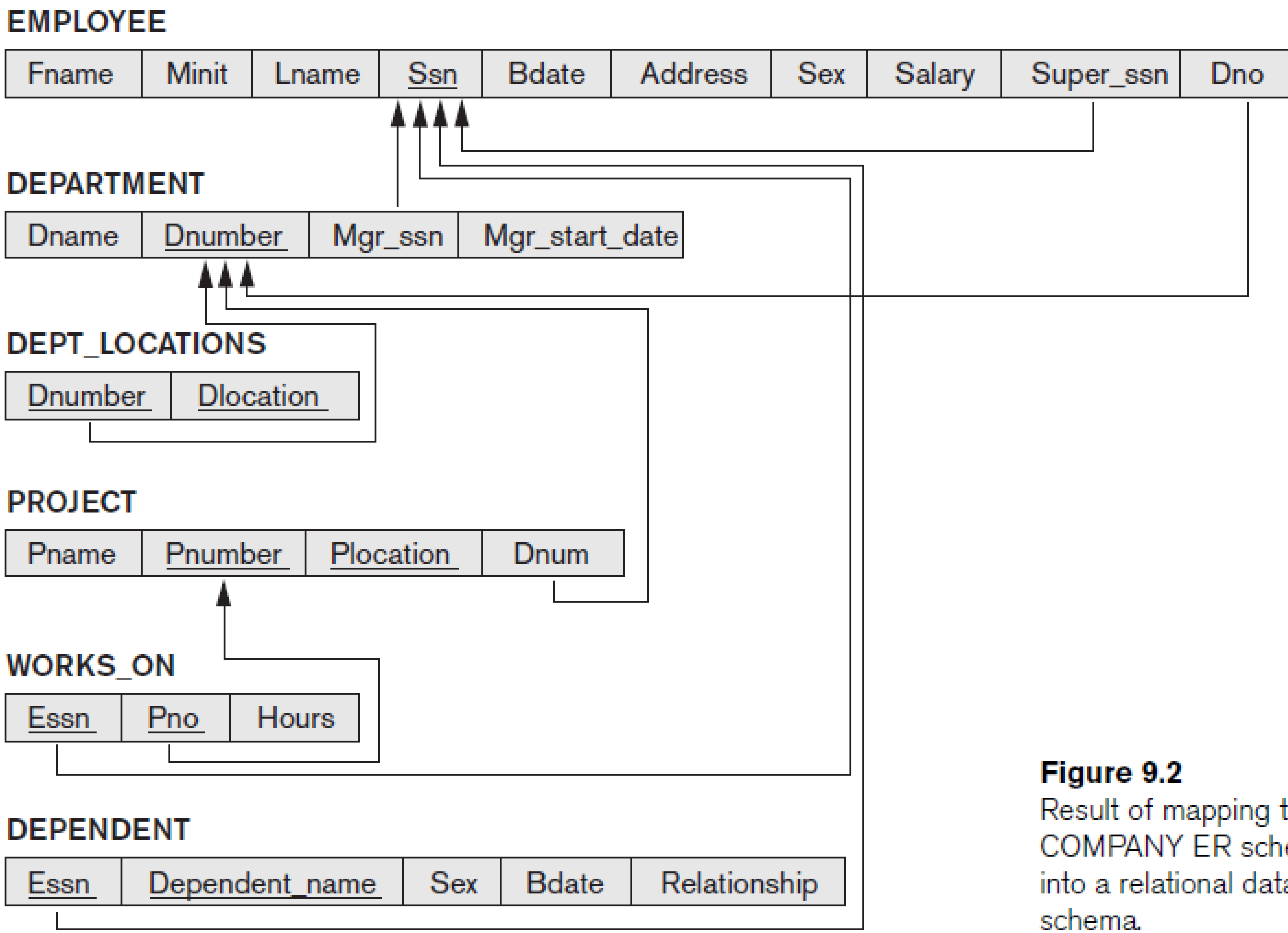



**Figure 9.2**
Result of mapping the COMPANY ER schema into a relational database schema.

# STEP 1: MAP REGULAR ENTITY TYPES

- For each regular entity type, create a relation schema *R* that includes all the single-valued attributes of *E*
  - "Flatten" composite attributes
  - Example renames some attributes (e.g., Dname), but not needed
  - Pick one of the keys as "primary key" and declare the rest to be unique
  - Called **entity relations**
  - Each tuple represents an entity instance

(a) **EMPLOYEE**

| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary |
|---|---|---|---|---|---|---|---|

**DEPARTMENT**

| Dname | <u>Dnumber</u> |
|---|---|

**PROJECT**

| Pname | <u>Pnumber</u> | Plocation |
|---|---|---|

# STEP 2: MAP WEAK ENTITY TYPES

- For each weak entity type, create a relation schema *R* and include all single-valued attributes of the weak entity type *and of the identifying relationship* as attributes of *R*
  - Include primary key attribute of identifying entity as foreign key attribute of *R*
  - Primary key of *R* is primary key of identifying entity together with partial key from *R*
- Omit the identifying relationship when subsequently translating (other) relationship types to relation schemas

(b) **DEPENDENT**

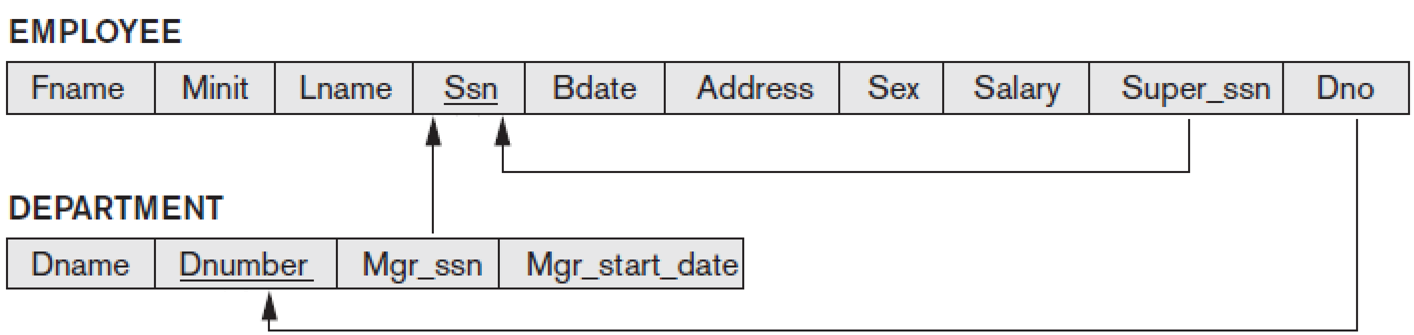
| Essn | Dependent_name | Sex | Bdate | Relationship |
|---|---|---|---|---|

# STEP 3: MAP BINARY 1:1 RELATIONSHIP TYPES

- For each binary 1:1 relationship type *R*, identify relation schemas that correspond to entity types participating in *R*
  - Apply one of three possible approaches:
    - **Foreign key approach**
      - Add primary key of one participating relation as foreign key attribute of the other, which will also represent *R*
        - If only one side is *total*, choose it to represent *R* *(why?)*
      - Declare foreign key attribute as unique
    - **Merged relationship approach**
      - Combine the two relation schemas into one, which will also represent *R*
      - Make one of the primary keys “unique” instead
    - **Cross-reference** or **relationship relation approach**
      - Create new relation schema for *R* with two foreign key attributes being copies of both primary keys
      - Declare one of the attributes as primary key and the other one as unique
  - Add single-valued attributes of relationship type as attributes of *R*

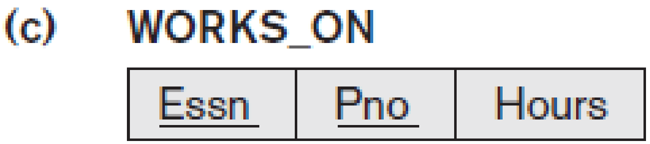
# STEP 4: MAP BINARY 1:*N* RELATIONSHIP TYPES

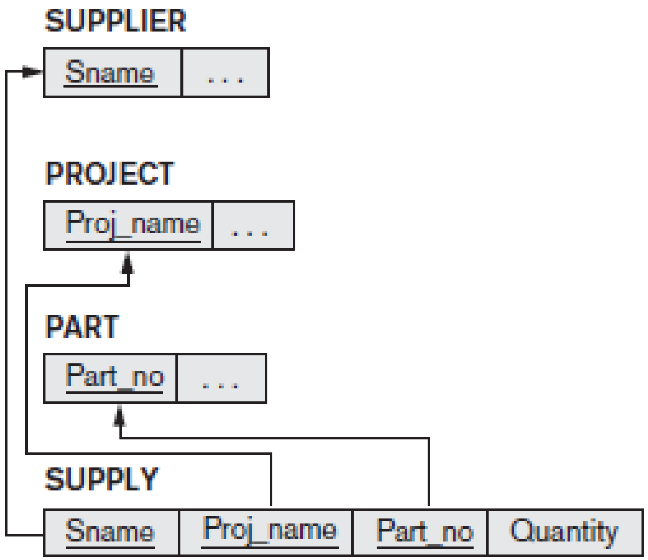
- **Foreign key approach**
  - Identify relation schema *S* that represents participating entity type at *N*-side of 1:*N* relationship type
  - Include primary key of other entity type (1-side) as foreign key in *S*
- **Relationship relation approach**
  - Create new relation schema for *S* with two foreign key attributes being copies of both primary keys
  - Declare the foreign key attribute for the relation schema corresponding to the participating entity type *on the N-side* as primary key
- Include single-valued attributes of relationship type as attributes of *S*



# STEP 5: MAP BINARY *M:N* AND HIGHER ORDER RELATIONSHIP TYPES

- For each binary *M*:*N* relationship type or ternary or higher order relationship type, create a new relation *S*
  - Include primary key of participating entity types as foreign key attributes in *S*
  - Make all these attributes primary key of *S*
  - Include any simple attributes of relationship type in *S*



# STEP 6: MAP MULTIVALUED ATTRIBUTES

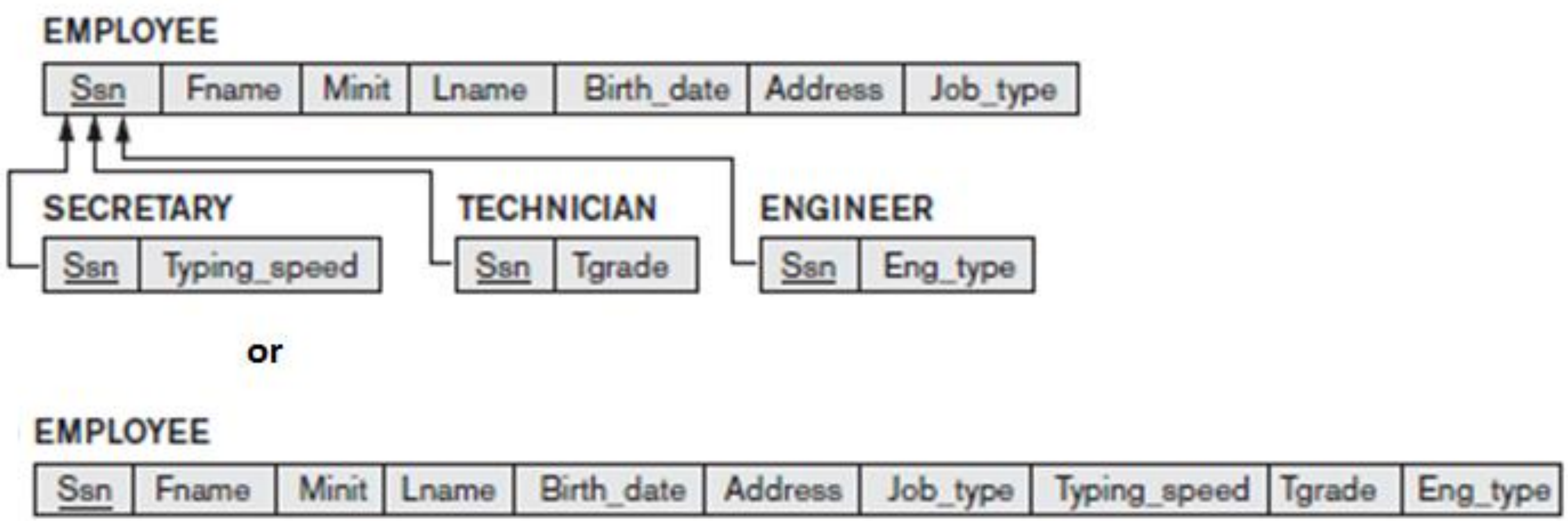
- For each multivalued attribute
  - Create new relation *R* with attribute to hold multivalued attribute values
    - If multivalued attribute is composite, include its simple components
  - Add attribute(s) for primary key of relation schema for entity or relationship type to be foreign key for *R*
  - Primary key of *R* is the combination of all its attributes

(d) DEPT_LOCATIONS

| Dnumber | Dlocation |
|---|---|

# OPTIONS FOR MAPPING SPECIALIZATION OR GENERALIZATION

- For *any* specialization (total or partial, disjoint or overlapping)
  - Separate relation per superclass and subclasses
  - Single relation with at least one attribute per subclass
    - Introduce a Boolean attribute if none specific for subclass

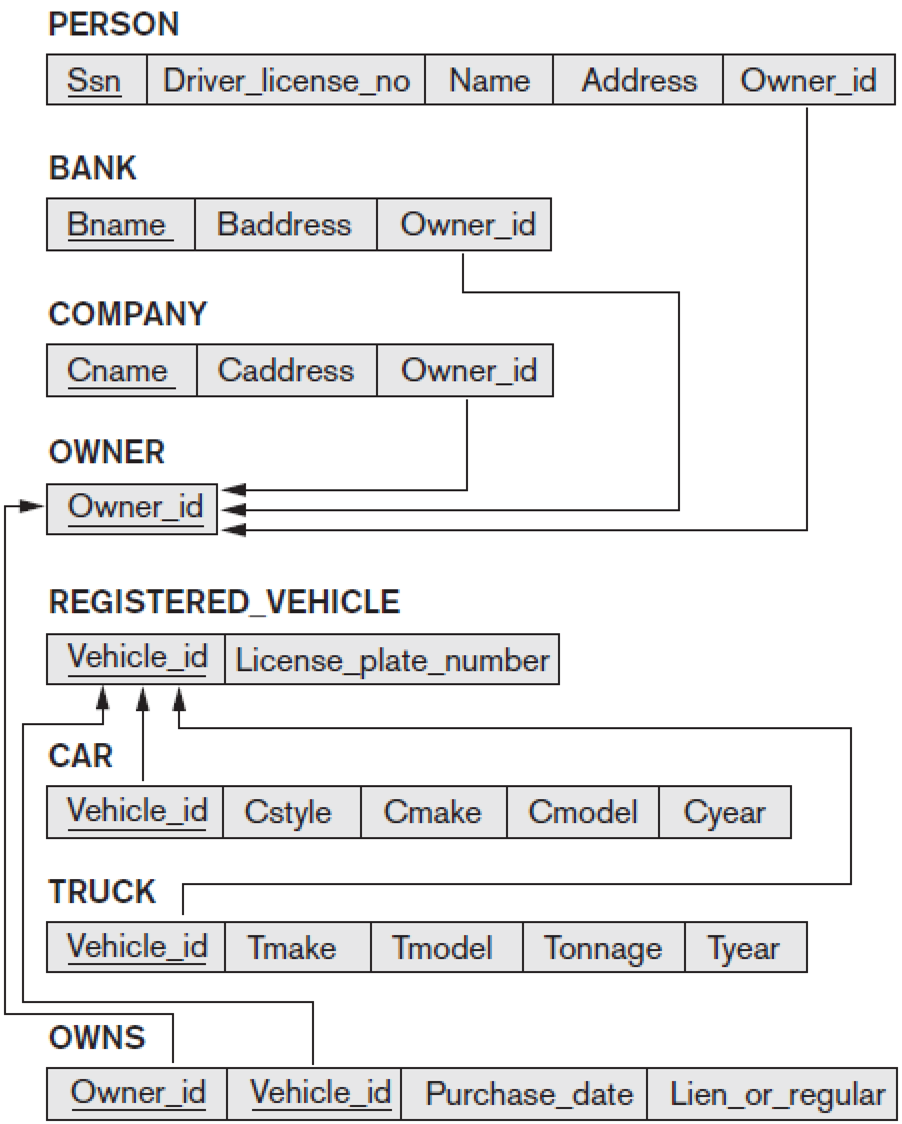


# SPECIALIZATION OPTIONS (CONT'D)

- For *total specializations* (and generalizations) *only*
  - Separate relation per subclass relations only
    - Overlapping subclasses will result in multiple tuples per entity
- For *disjoint specializations only*
  - Single relation with one type attribute
    - **Type** or **discriminating attribute** indicates subclass of tuple
    - Might require many NULL values if several specific attributes exist in subclasses

# MAPPING UNION TYPES

- Create relation schema to represent union type (generalization)
- Specify a new key attribute
  - **Surrogate key**
- Example: *Owner* and *Registered Vehicle*



# LECTURE SUMMARY

- Algorithm for ER-to-relational mapping

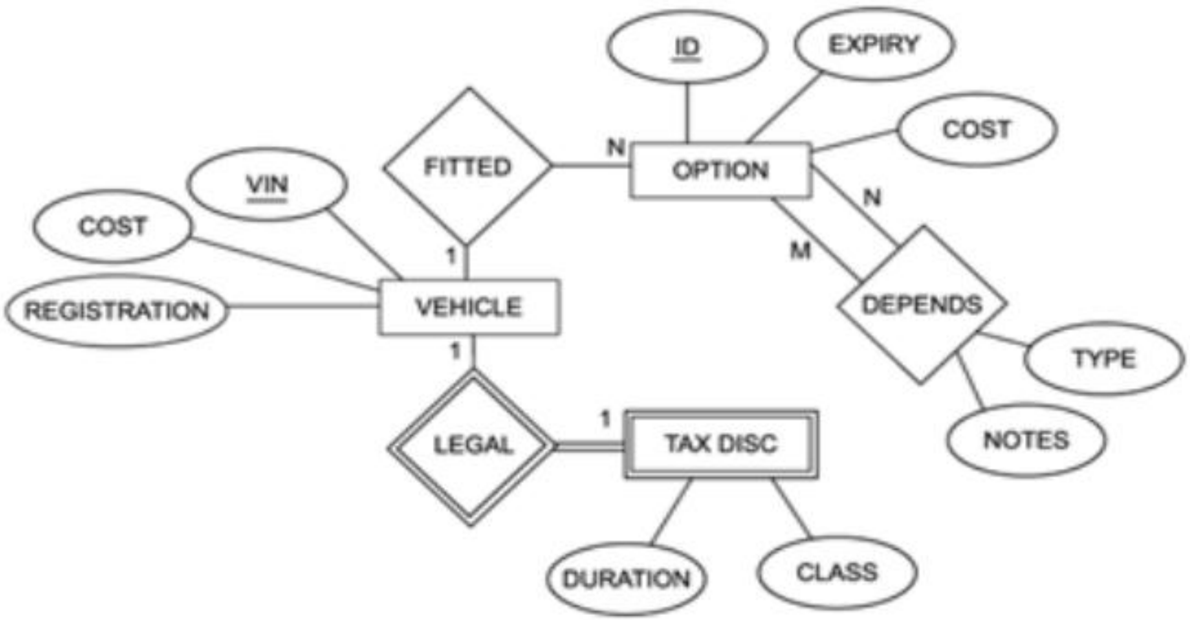
**Table 9.1** Correspondence between ER and Relational Models

| 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 |

- Extensions for mapping constructs from EER model into relational model

# EXERCISE

Translate the following ER Diagram into a relational database schema.



# EXERCISE

What ER Diagram might produce the following relational database schema?

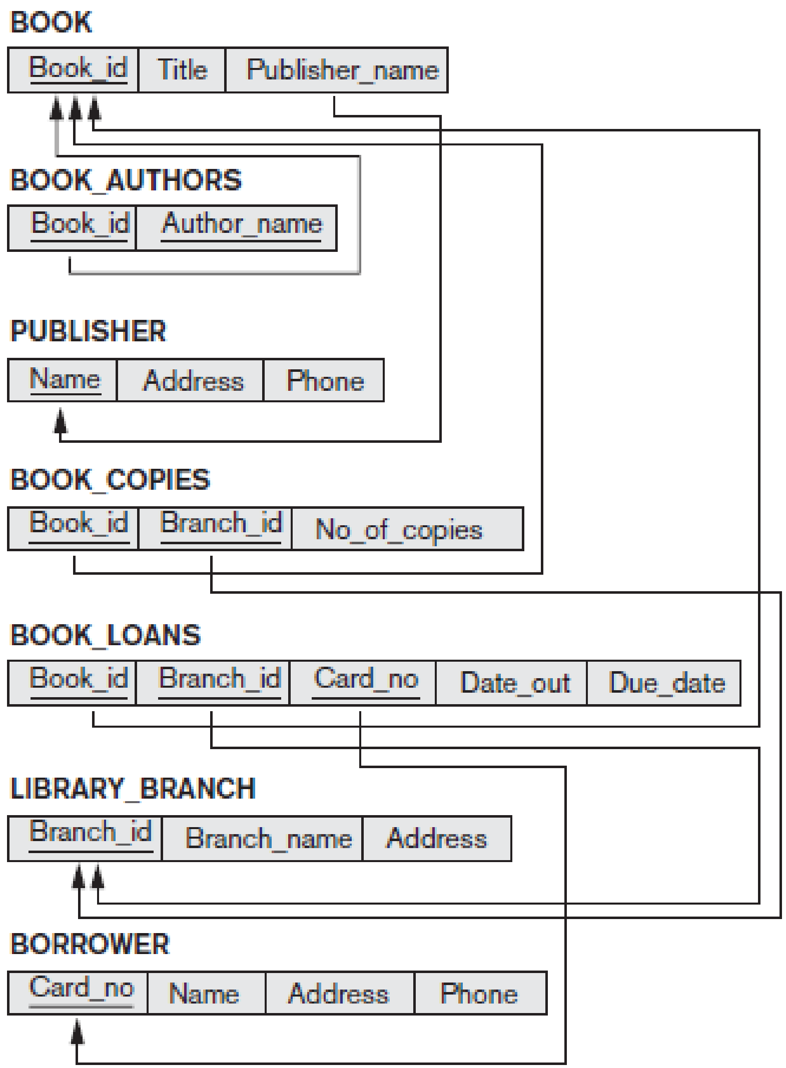



**Figure 6.14**
A relational database schema for a LIBRARY database.