ENSF 608: Understanding and Mapping the Relational Data Model

Dr. Emily Marasco

Fall 2021

Textbook: Fundamentals of Database Systems, 7th Ed., Elmasri & Navathe

Goals during Mapping

- Preserve all information (that includes all attributes)
- Maintain the constraints to the extent possible
  - Relational model cannot preserve all constraints
    - e.g., max cardinality ratio such as 1:10 in ER; exhaustive classification into subtypes, e.g., STUDENTS are specialized into Domestic and Foreign)
- Minimize null values
- Minimize data redundancy

Figure 9.1 The ER Conceptual Schema Diagram for the COMPANY Database

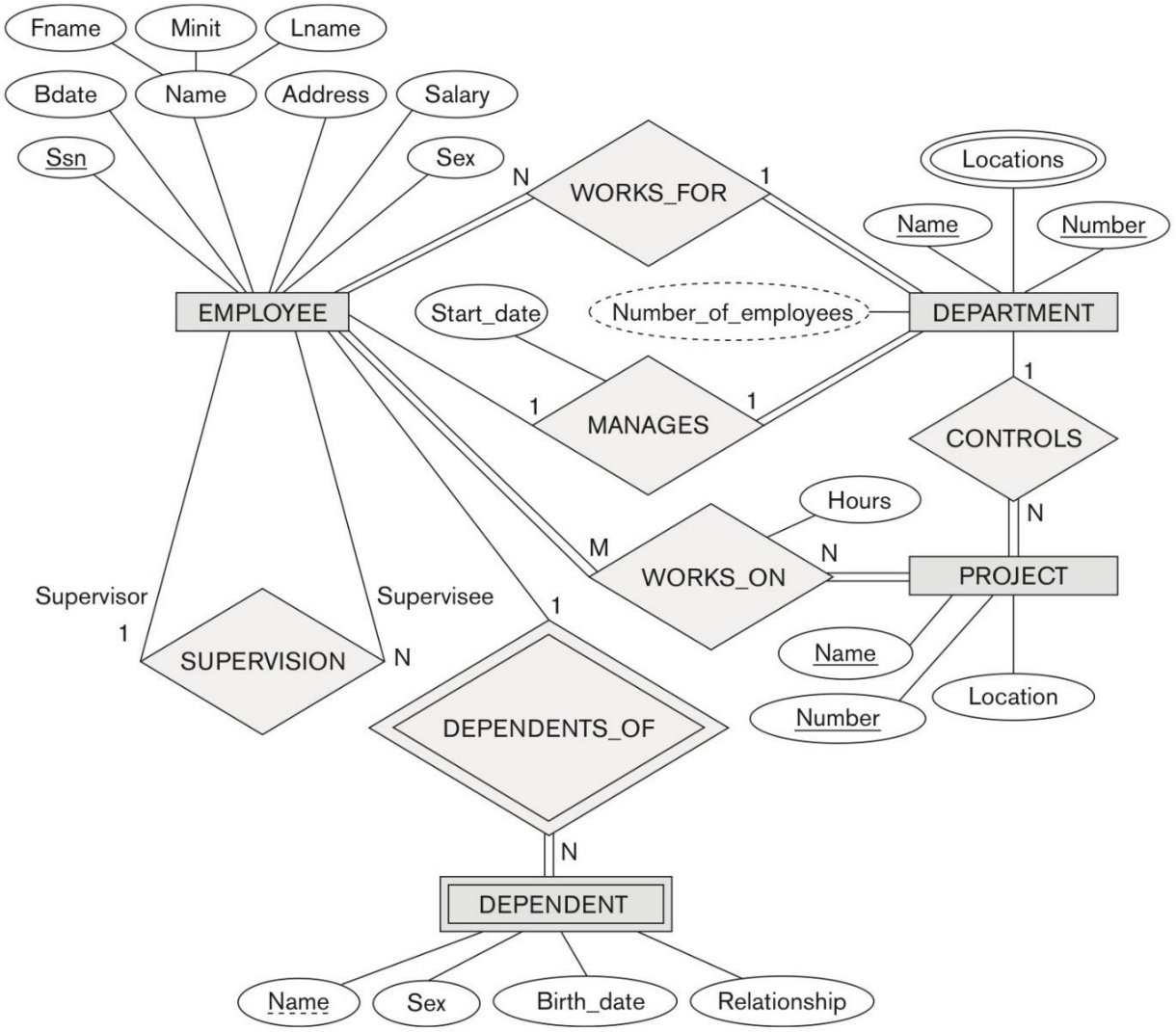
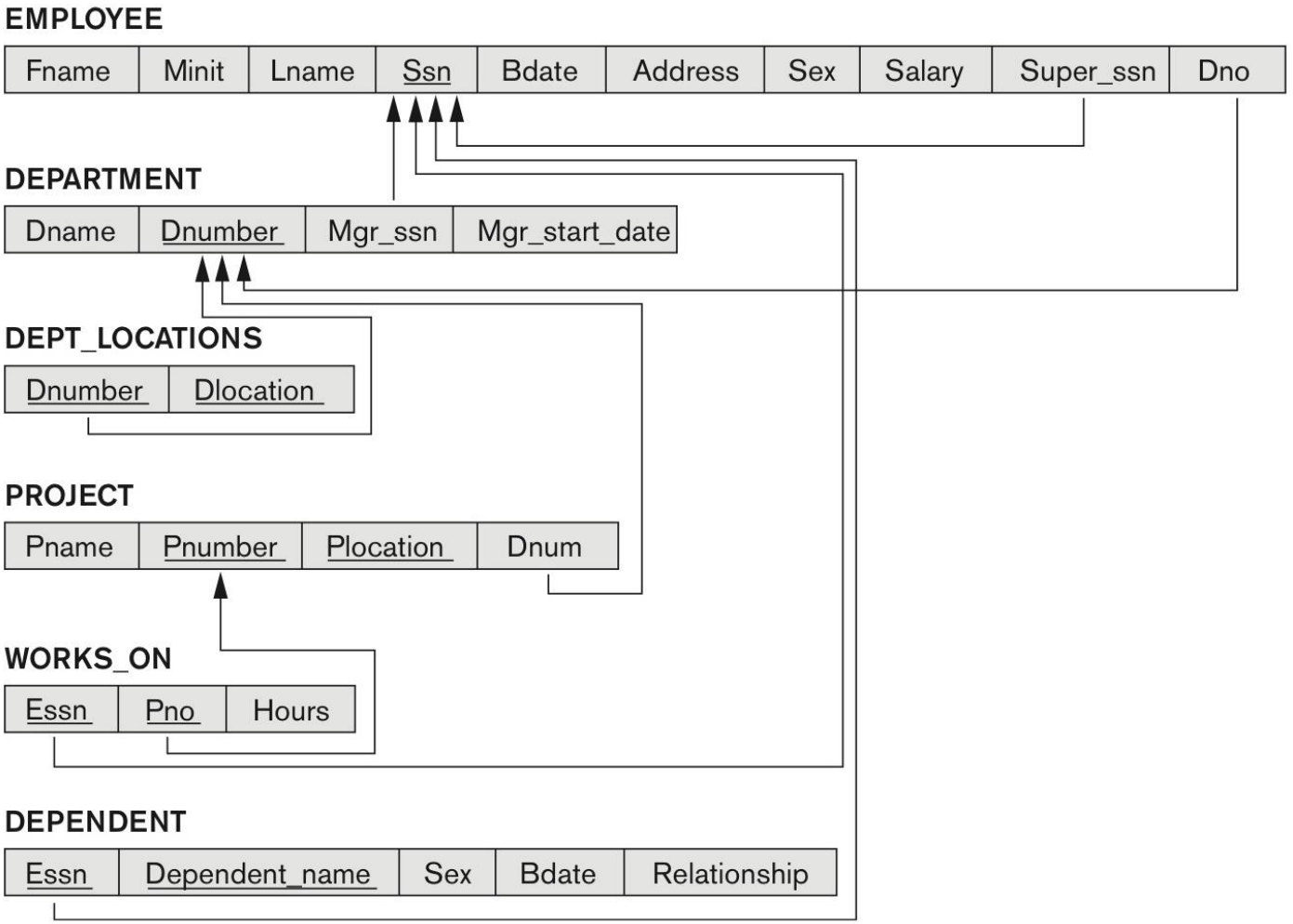


Figure 9.2 Result of Mapping the COMPANY ER Schema into a Relational Database Schema



ER-to-Relational Mapping Algorithm (1 of 14)

- 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*.

ER-to-Relational Mapping Algorithm (2 of 14)

- 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.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

PROJECT

Pname	Pnumber	Plocation	Dnum

ER-to-Relational Mapping Algorithm (3 of 14)

- **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.

ER-to-Relational Mapping Algorithm (4 of 14)

- **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.

DEPENDENT

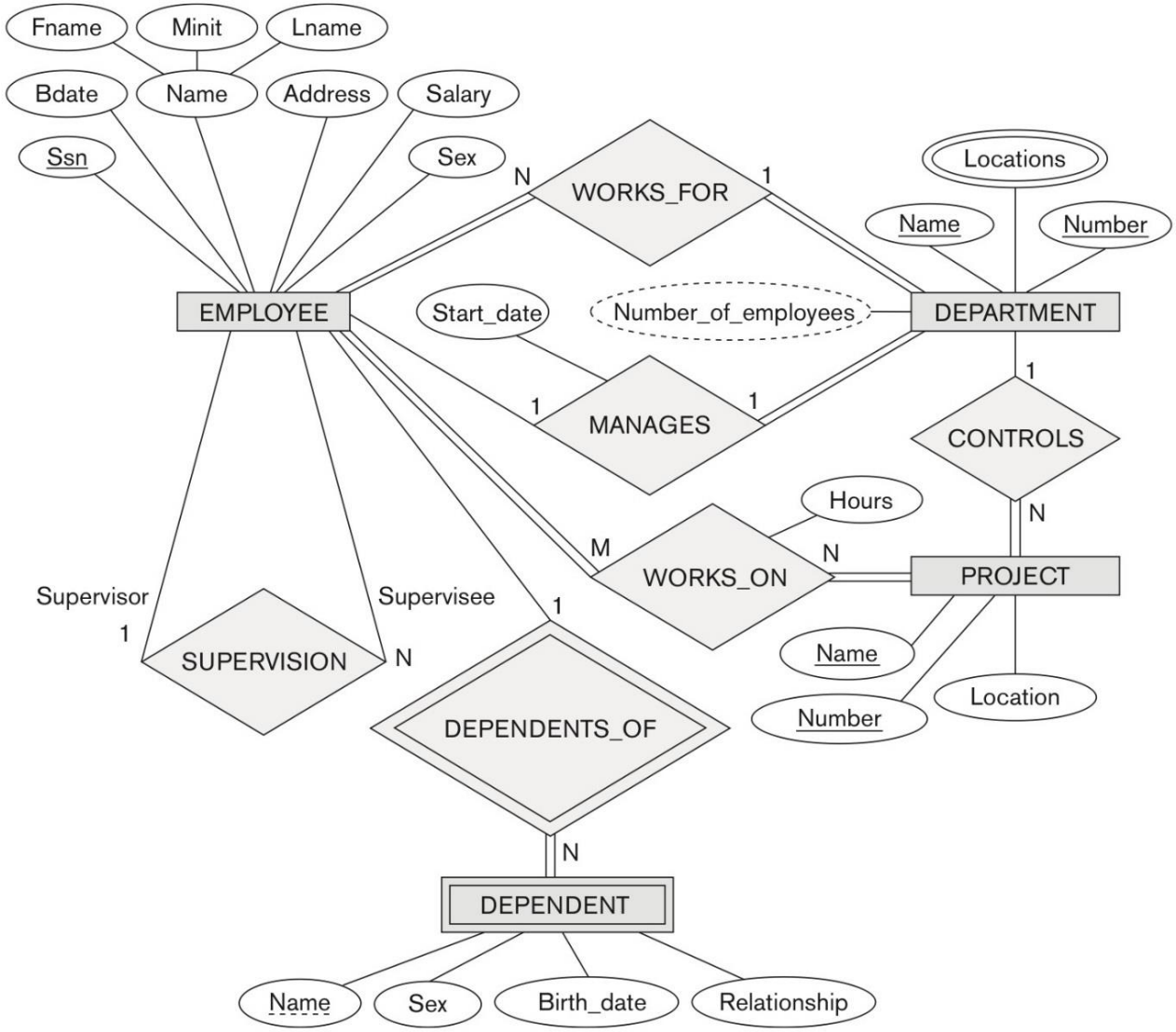
Essn	Dependent_name	Sex	Bdate	Relationship

ER-to-Relational Mapping Algorithm (5 of 14)

- **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*.
- There are three possible approaches:
  1. **Foreign Key (2 relations) approach:** 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.

ER-to-Relational Mapping Algorithm (6 of 14)

2. **Merged relation (1 relation) option:** An alternate mapping of a 1:1 relationship type is possible by merging the two entity types and the relationship into a single relation. This may be appropriate when both participations are total.
3. **Cross-reference or relationship relation (3 relations) option:** The third alternative is to set up a third relation *R* for the purpose of cross-referencing the primary keys of the two relations *S* and *T* representing the entity types.



DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

ER-to-Relational Mapping Algorithm (7 of 14)

- 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*.

ER-to-Relational Mapping Algorithm (8 of 14)

- 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.
- An alternative approach is to use a Relationship relation (cross referencing relation) – this is rarely done.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

ER-to-Relational Mapping Algorithm (9 of 14)

- **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*.

ER-to-Relational Mapping Algorithm (10 of 14)

- 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

ER-to-Relational Mapping Algorithm (11 of 14)

- **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.

ER-to-Relational Mapping Algorithm (12 of 14)

- **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}.

DEPT_LOCATIONS

Dnumber	Dlocation

Figure 9.1 The ER Conceptual Schema Diagram for the COMPANY Database

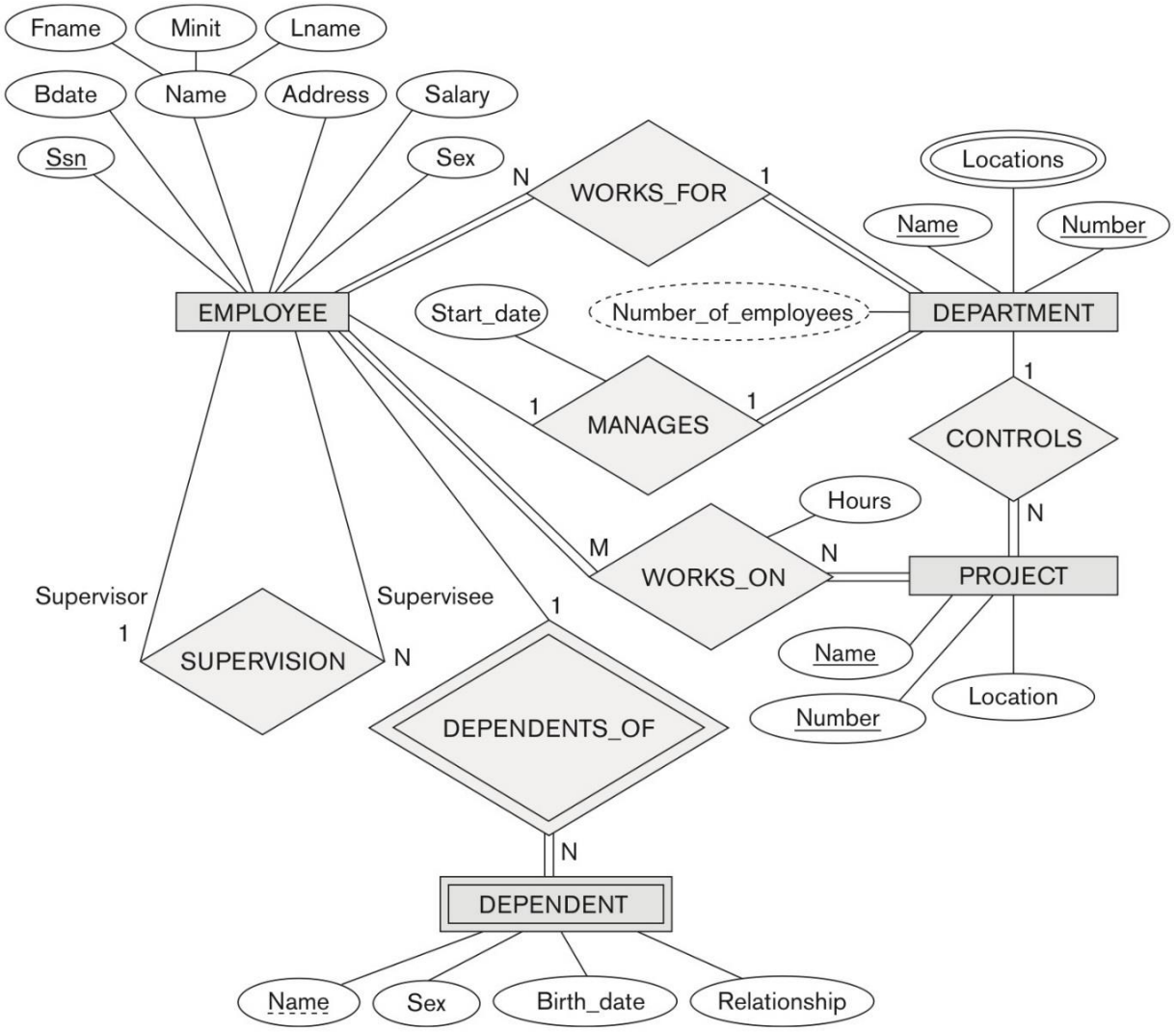
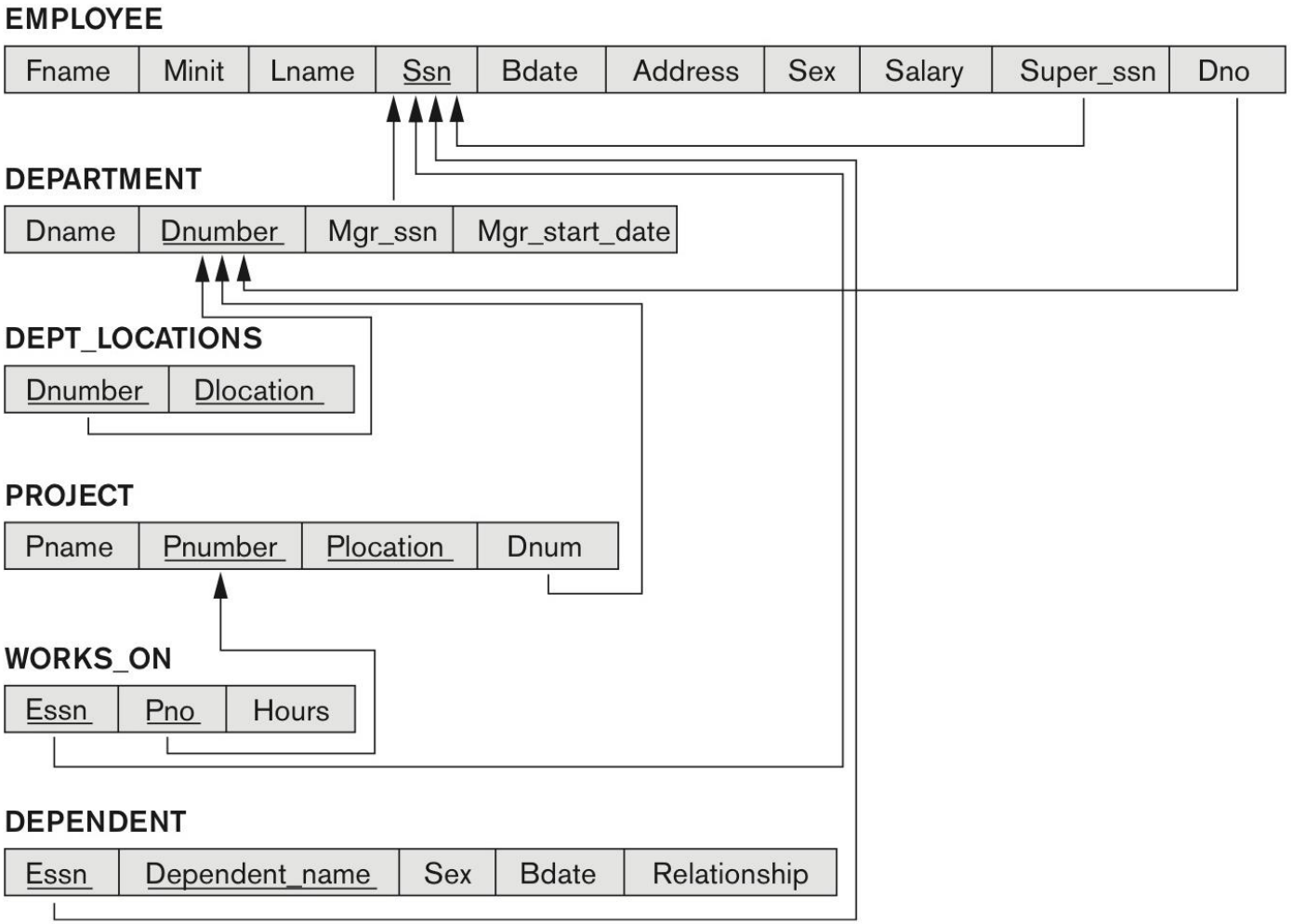


Figure 9.2 Result of Mapping the COMPANY ER Schema into a Relational Database Schema



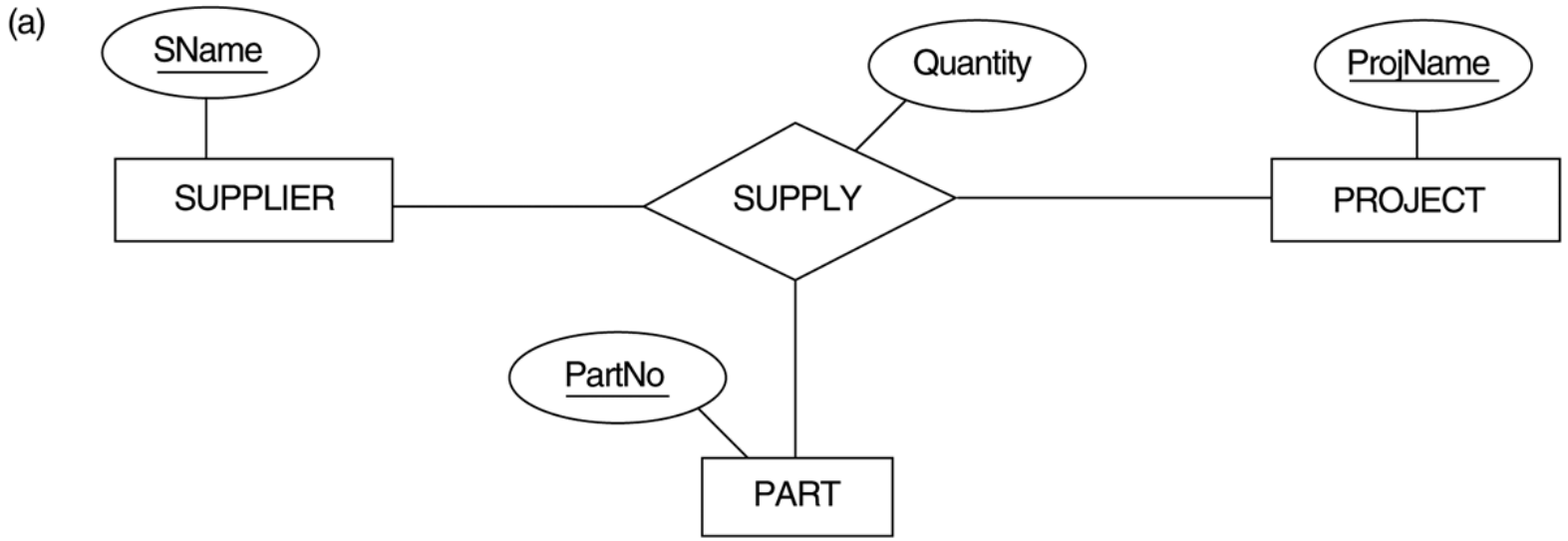
ER-to-Relational Mapping Algorithm (13 of 14)

- **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*.

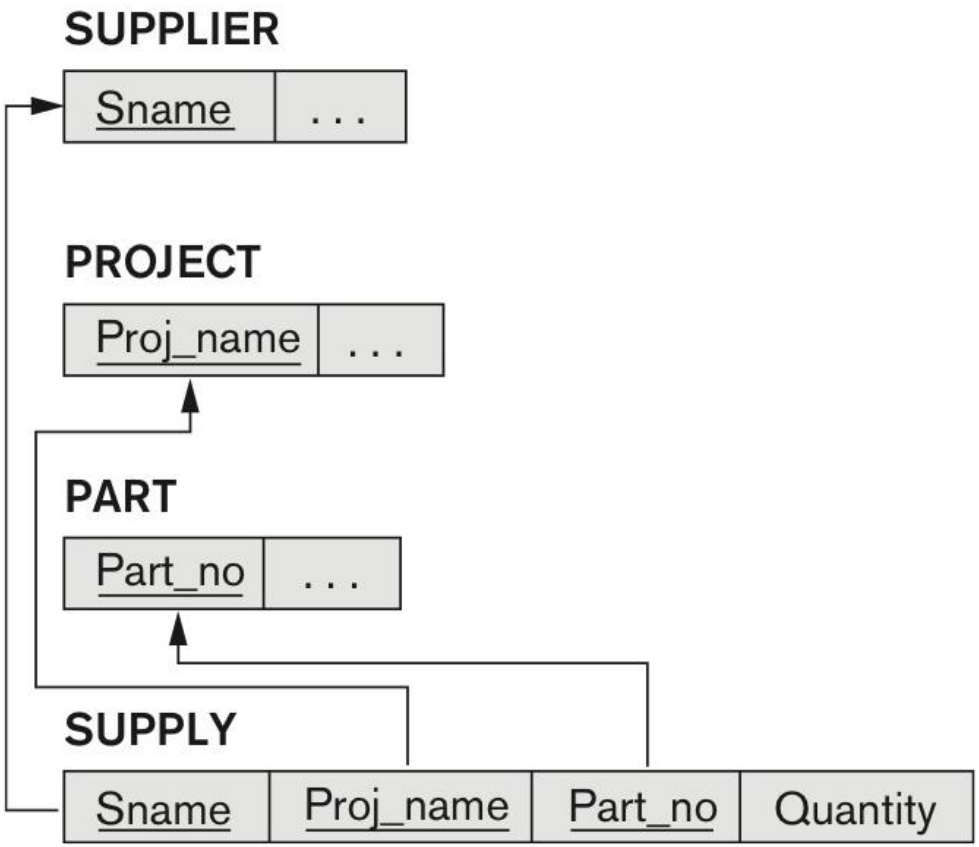
ER-to-Relational Mapping Algorithm (14 of 14)

- **Example:** The relationship type SUPPLY 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, PART NO, PROJNAME}

Figure 3.17 Ternary Relationship: Supply



Mapping the *n*-ary relationship type SUPPLY



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