Chapter 7.1

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

Addison-Wesley
is an imprint of
PEARSON

Data Model Mapping Phase of Relational DB Design

- DB designers use ER/EER data model to produce a conceptual schema design (*independent* from any specific DBMS) during the *Conceptual Database Design* phase
- In *Logical Database Design* Phase, this conceptual schema design is converted (Mapped) to the data model of the DBMS, typically relational model.

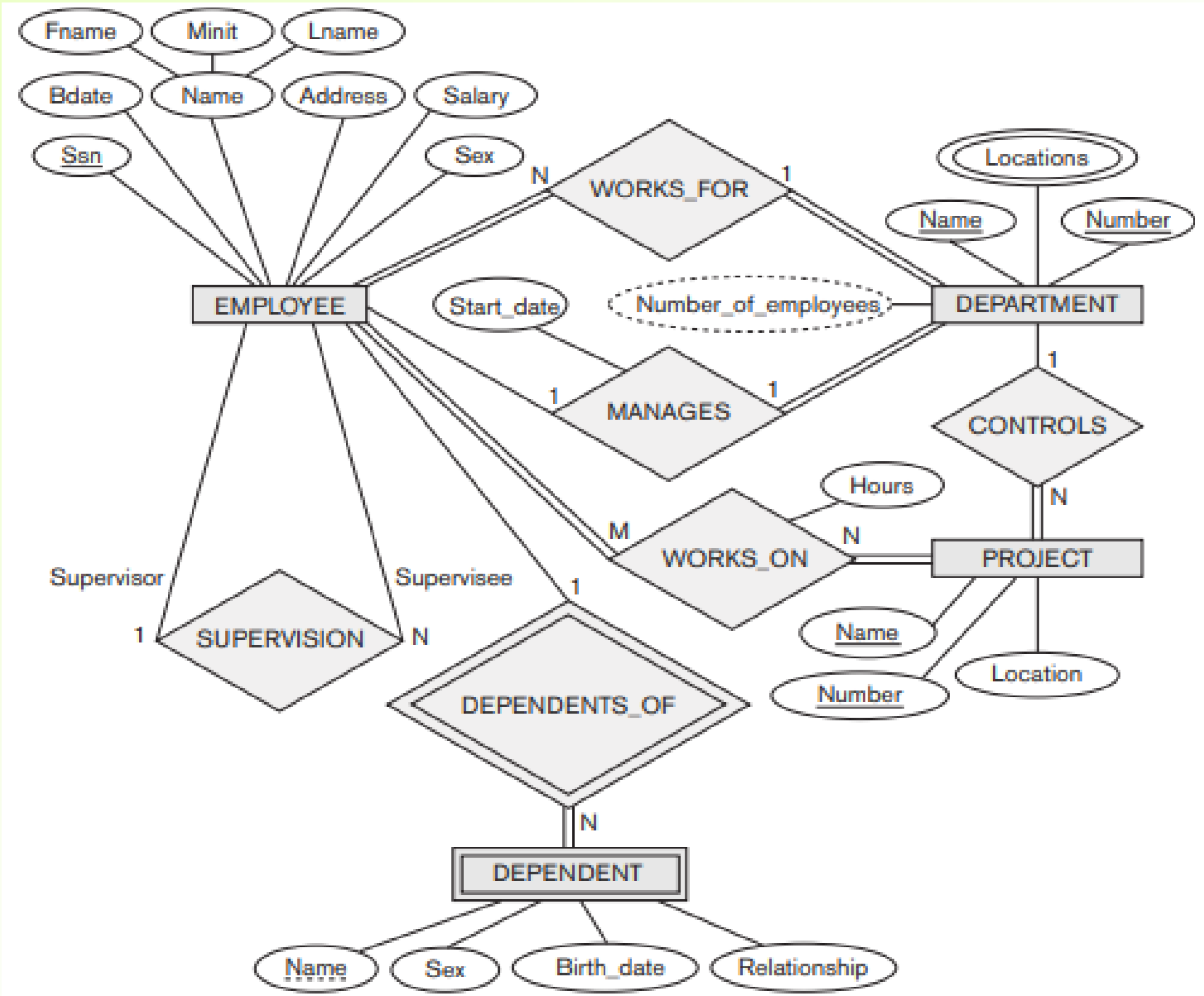


Fig: ER schema diagram for COMPANY database

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 (or simple components of composite attributes) of E.
  - Choose one of the key attributes of E as 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 entity types. SSN, DNUMBER, and PNUMBER are chosen as primary keys for the relations EMPLOYEE, DEPARTMENT, and PROJECT. Additional attributes will be added to these tables in later mapping steps

ER-to-Relational Mapping Algorithm (cont.)

- **Step 2: Mapping of Weak Entity Types**
  - For each weak entity type W with owner entity type E, create a relation R that includes all simple attributes (or simple components of composite attributes) of W as attributes of R.
  - Include as foreign key attribute(s) in R the primary key attribute(s) of the relation(s) that corresponds 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 DEPENDENT is the combination {ESSN, DEPENDENT_NAME} because DEPENDENT_NAME is the partial key of DEPENDENT.

ER-to-Relational Mapping Algorithm (cont.)

- **Step 3: Mapping of Binary 1:1 Relationship 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. Three possible approaches:
    1. **Foreign Key approach:** Choose one of the relations (say S) and include as *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). For example, 1:1 relationship MANAGES is mapped by choosing DEPARTMENT to serve in the role of S (because its participation in the MANAGES relationship type is total). Mgr_SSN of DEPARTMENT is foreign key referencing EMPLOYEE. Attributes of MANAGES become attributes of DEPARTMENT.
    2. **Merged relation option:** Merge the two entity types and the relationship into a single relation (possible when *both participations are total*).

ER-to-Relational Mapping Algorithm (cont.)

3. **Cross-reference or *relationship relation* option:** Set up a third relation R with primary keys of the two relations S and T representing the entity types. The relation R will include the primary key attributes of S and T as foreign keys.. The primary key of R will be one of the two foreign keys, and the other foreign key will be a unique key of R. The drawback is having an extra relation, and requiring extra join operations when combining related tuples from the tables.

ER-to-Relational Mapping Algorithm (cont.)

- **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.
  - Examples: 1:N relationship types are WORKS_FOR, CONTROLS, and SUPERVISION.

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

ER-to-Relational Mapping Algorithm (cont.)

For CONTROLS, we include the primary key DNUMBER of DEPARTMENT as foreign key in PROJECT and call it DNUM.

For SUPERVISION, we include the primary key SSN of EMPLOYEE as foreign key in EMPLOYEE itself and call it Super_ssn (this is a recursive relationship).

- Can also use the **cross-reference** option (create a separate relation that has the primary keys of both relations as foreign keys).
- An **alternative approach** is to use the relationship relation (cross-reference) option as in the third option for binary 1:1 relationships. We create a separate relation R whose attributes are the primary keys of S and T, which will also be foreign keys. The primary key of R is the same as the primary key of S.

ER-to-Relational Mapping Algorithm (cont.)

- **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.
  - 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.
  - The M:N relationship type WORKS_ON is mapped by creating a relation WORKS_ON in the relational database schema. The primary keys of PROJECT and EMPLOYEE are 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 WORKS_ON is the combination {ESSN, PNO}.

ER-to-Relational Mapping Algorithm (cont.)

- **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 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 is foreign key to the DEPARTMENT relation. The primary key of DEPT_LOCATIONS is the combination of {DNUMBER, DLOCATION}.

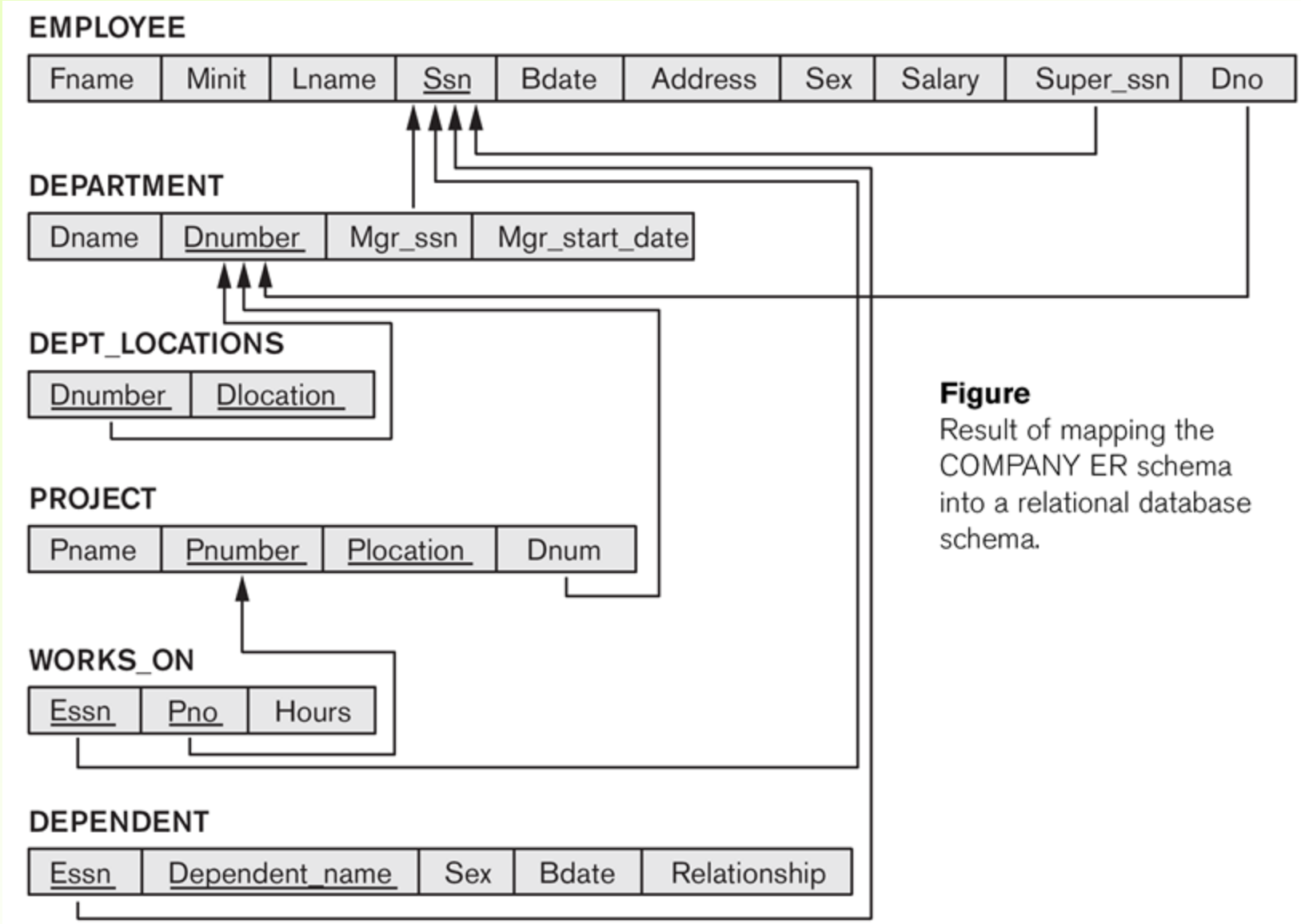


Figure
Result of mapping the COMPANY ER schema into a relational database schema.

ER-to-Relational Mapping Algorithm (cont.)

- **Step 7: Mapping of N-ary Relationship Types.**
  - For each n-ary relationship type R, where n>2, create a new *relationship relation* 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. The relationship type SUPPLY (Figure 7.17(a), next slide)
  - This can be mapped to the relation SUPPLY (Figure 9.4, following slide), whose primary key is the combination of the three foreign keys {SNAME, PARTNO, PROJNAME}

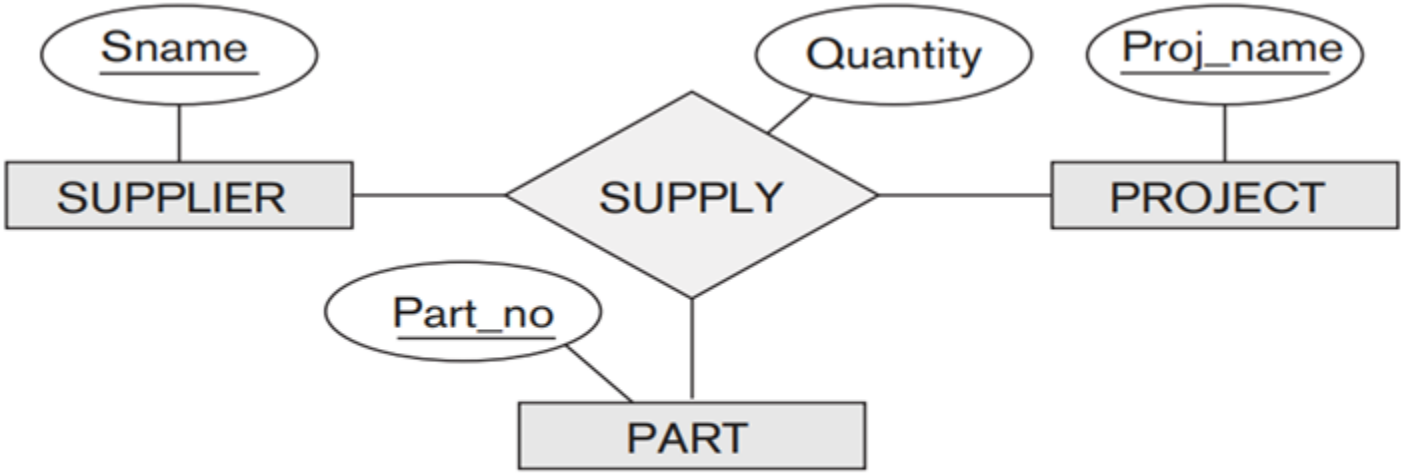


Fig: Ternary relationship

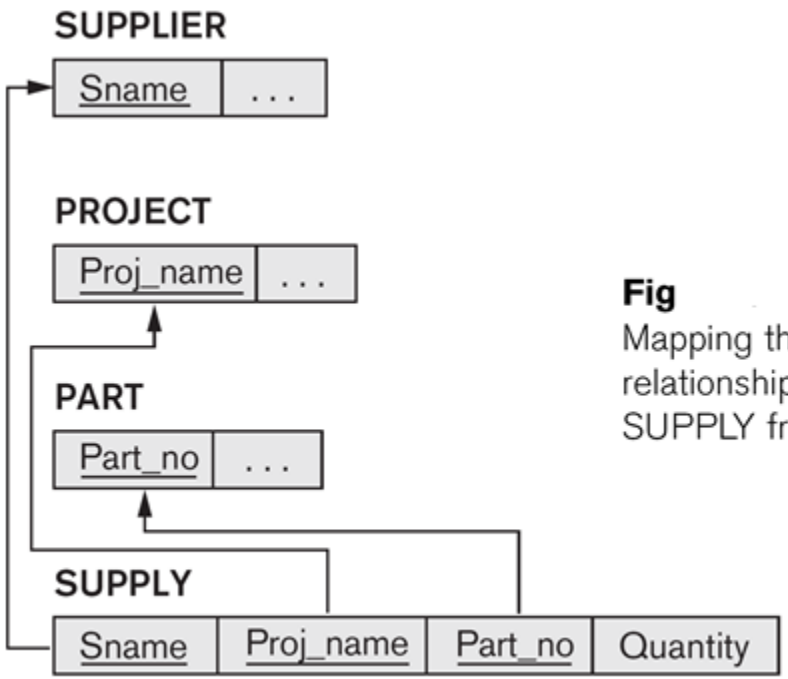


Fig
Mapping the *n*-ary relationship type SUPPLY from Figure

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 (or mapping) attributes

Mapping EER Model Constructs to Relations (cont.)

- **Option 8A: Multiple relations-Superclass and subclasses**
  - Create a relation L for superclass C with attributes Attrs(L) = {k,a1,…an} and PK(L) = k. Create a relation Li for each subclass Si, $1 \leq i \leq m$, with the attributes Attrs(Li) = {k} U {attributes of Si} and PK(Li)=k. This option works for any specialization (total or partial, disjoint of over-lapping).
- **Option 8B: Multiple relations-Subclass relations only**
  - Create a relation Li for each subclass Si, $1 \leq i \leq m$, 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* (every entity in the superclass must belong to (at least) one of the subclasses)
  - Works best if subclasses are also *disjoint*

Mapping EER Model Constructs to Relations (cont.)

- **Option 8C: Single relation with one type attribute**
  - Create a single relation L with attributes Attrs(L) = {k,a_1,…a_n} U {attributes of S_1} U…U {attributes of S_m} 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 for *disjoint* subclasses (see Figure 9.5(c))
- **Option 8D: Single relation with multiple type attributes**
  - Create a single relation schema L with attributes Attrs(L) = {k,a_1,…a_n} U {attributes of S_1} U…U {attributes of S_m} U {t_1, t_2,…,t_m} and PK(L) = k. Each t_i, $1 \leq I \leq m$, is a Boolean type attribute indicating whether or not a tuple belongs to the subclass S_i.
  - Works for *overlapping* subclasses (see Figure 9.5(d))

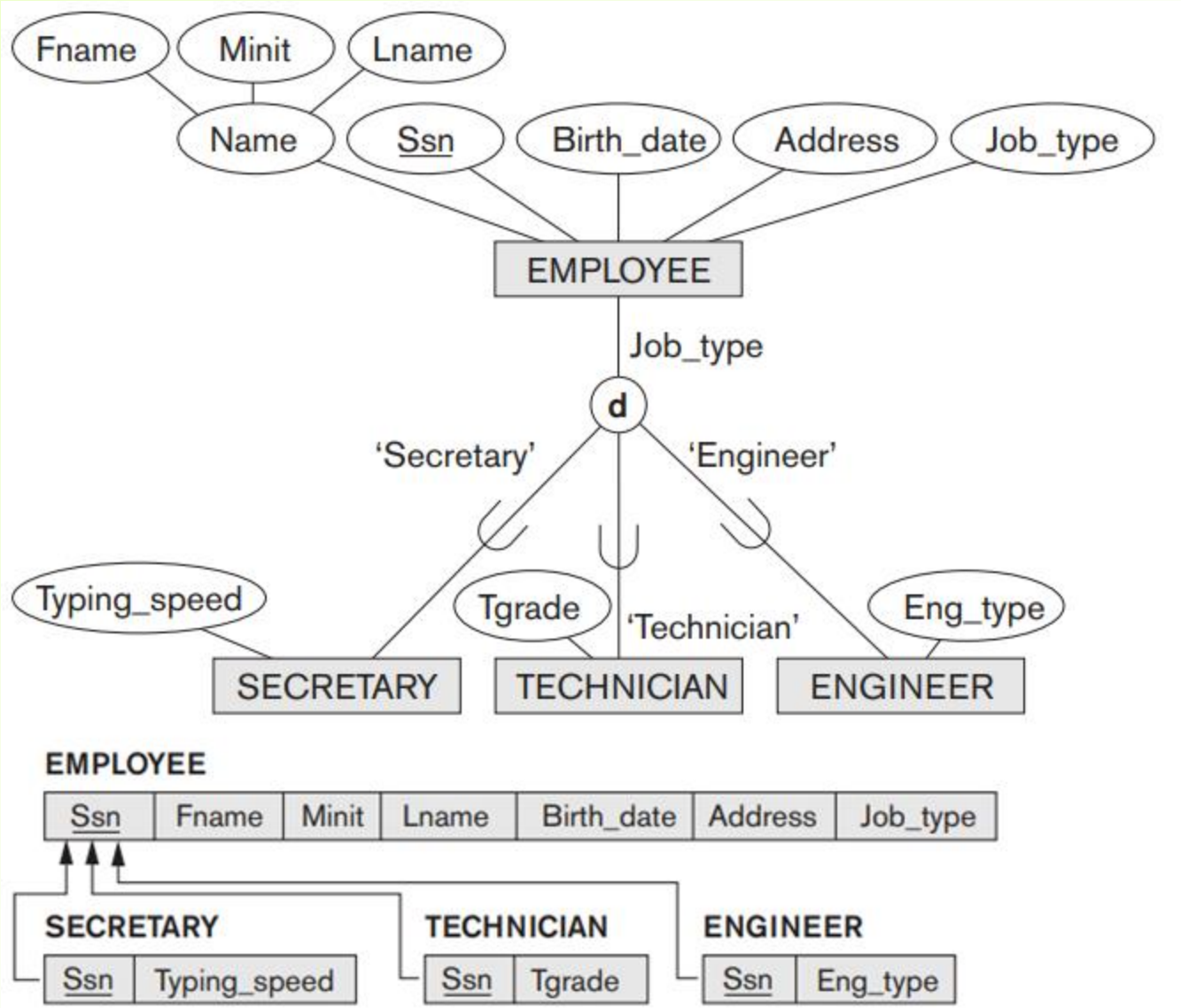
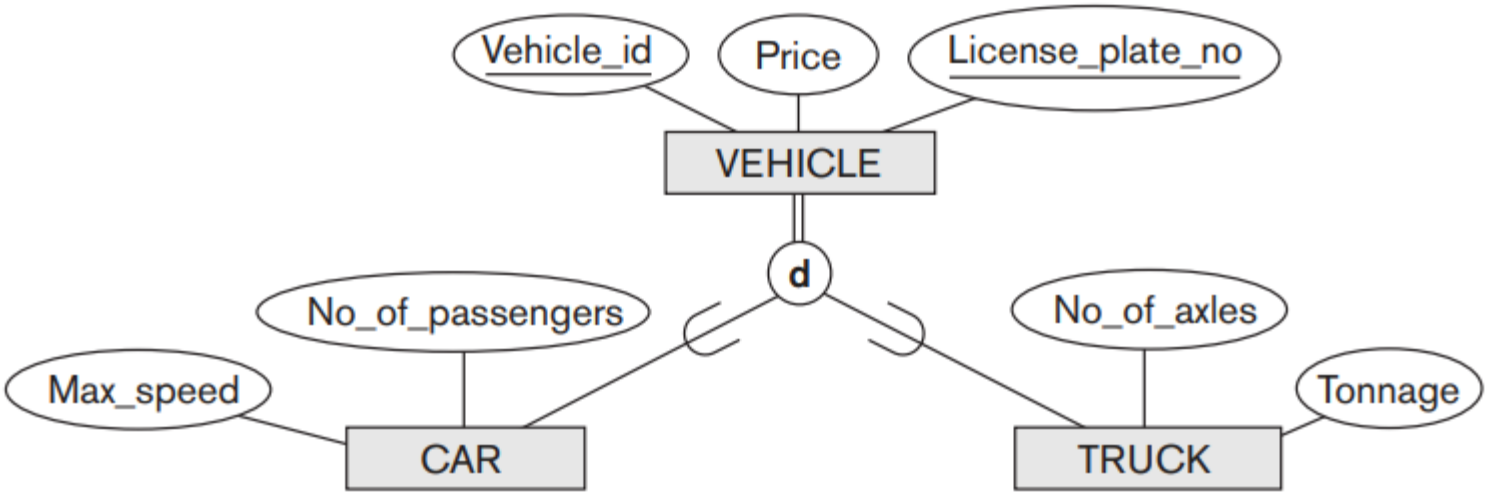


Fig: Option 8A



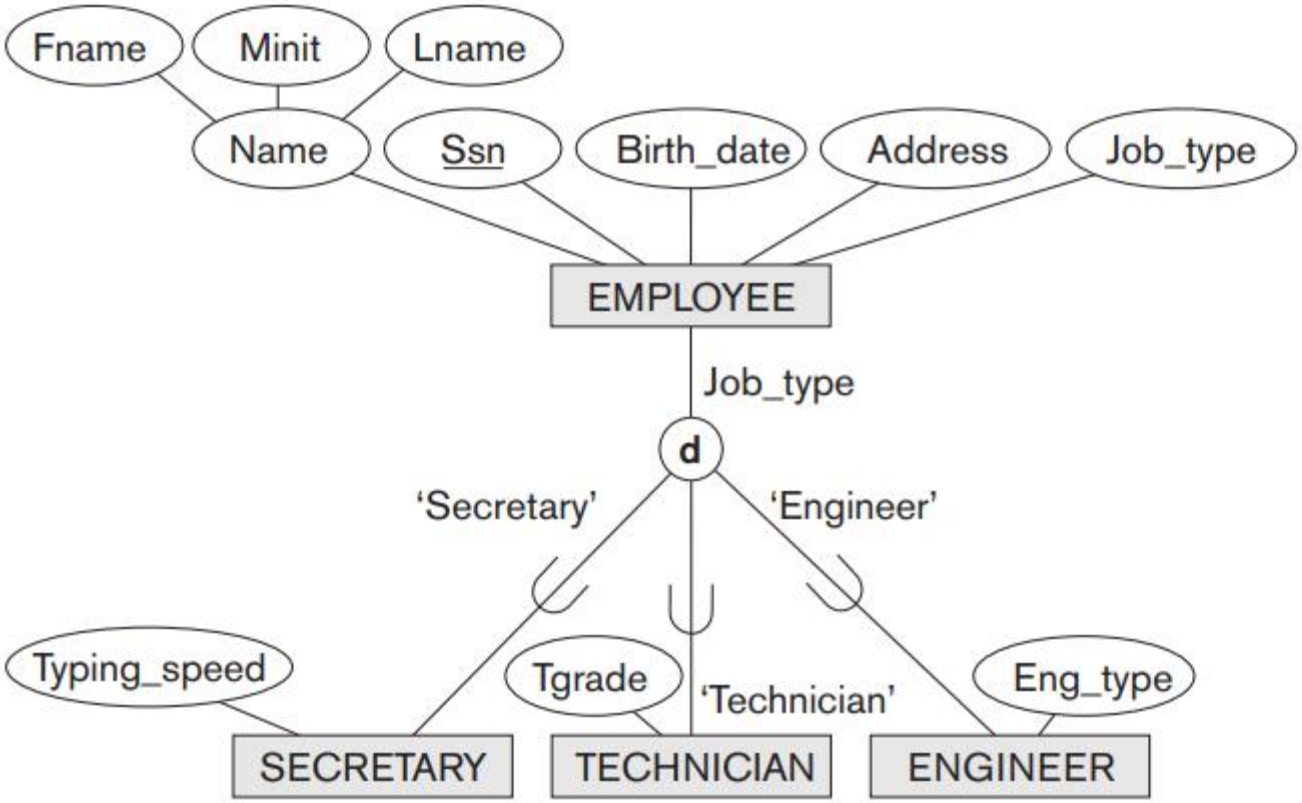
CAR

Vehicle_id	License_plate_no	Price	Max_speed	No_of_passengers

TRUCK

Vehicle_id	License_plate_no	Price	No_of_axles	Tonnage

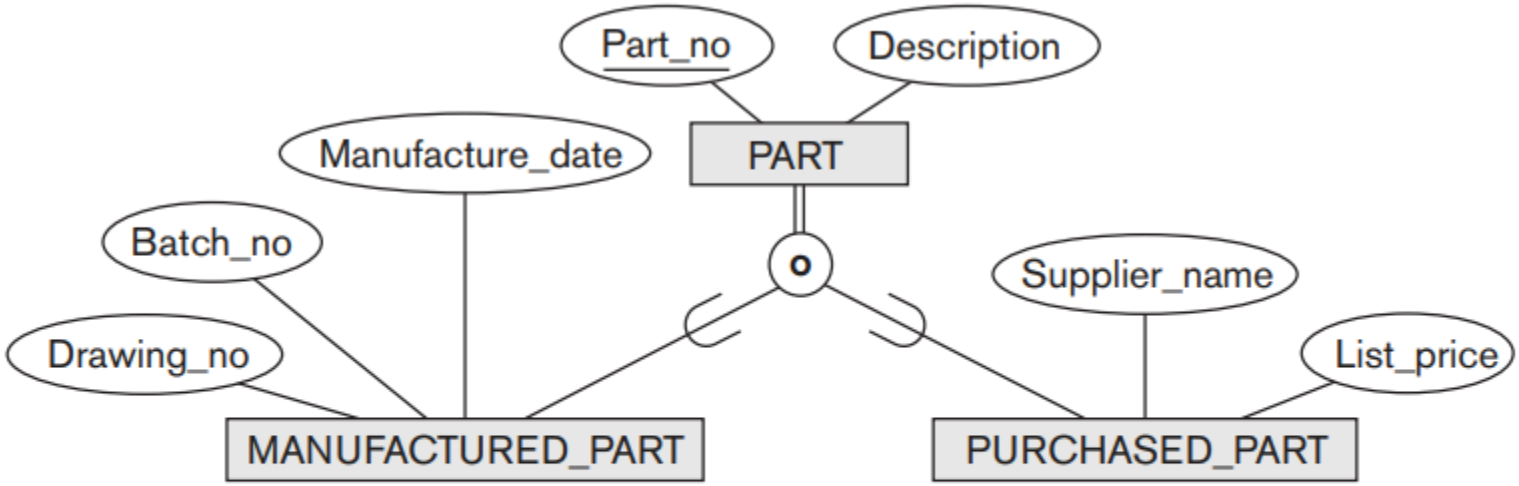
Fig: Option 8B



EMPLOYEE

Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type

Fig: Option 8C



PART

Part_no	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price

Figure: Option 8D