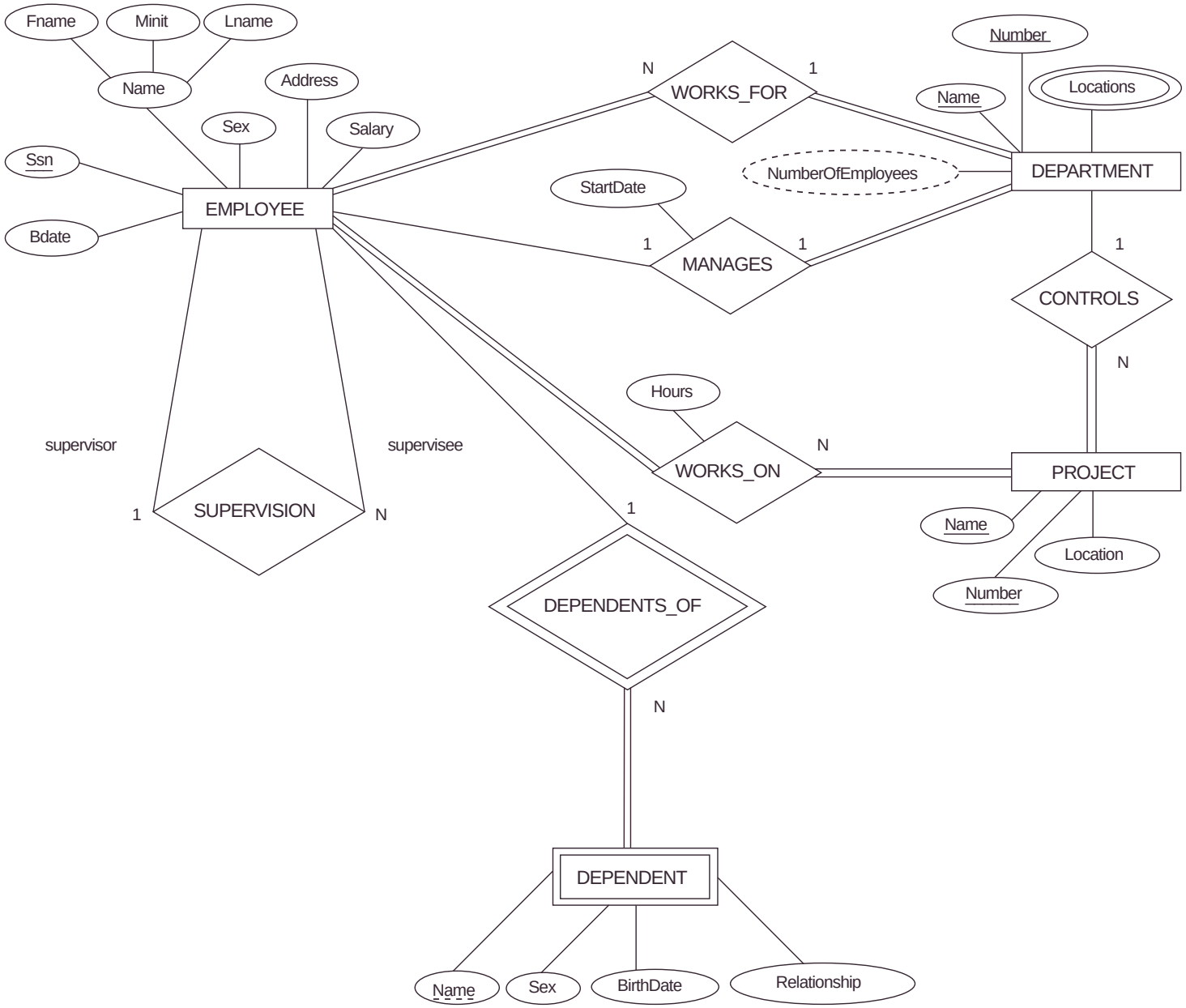
ER & EER to Relational Mapping

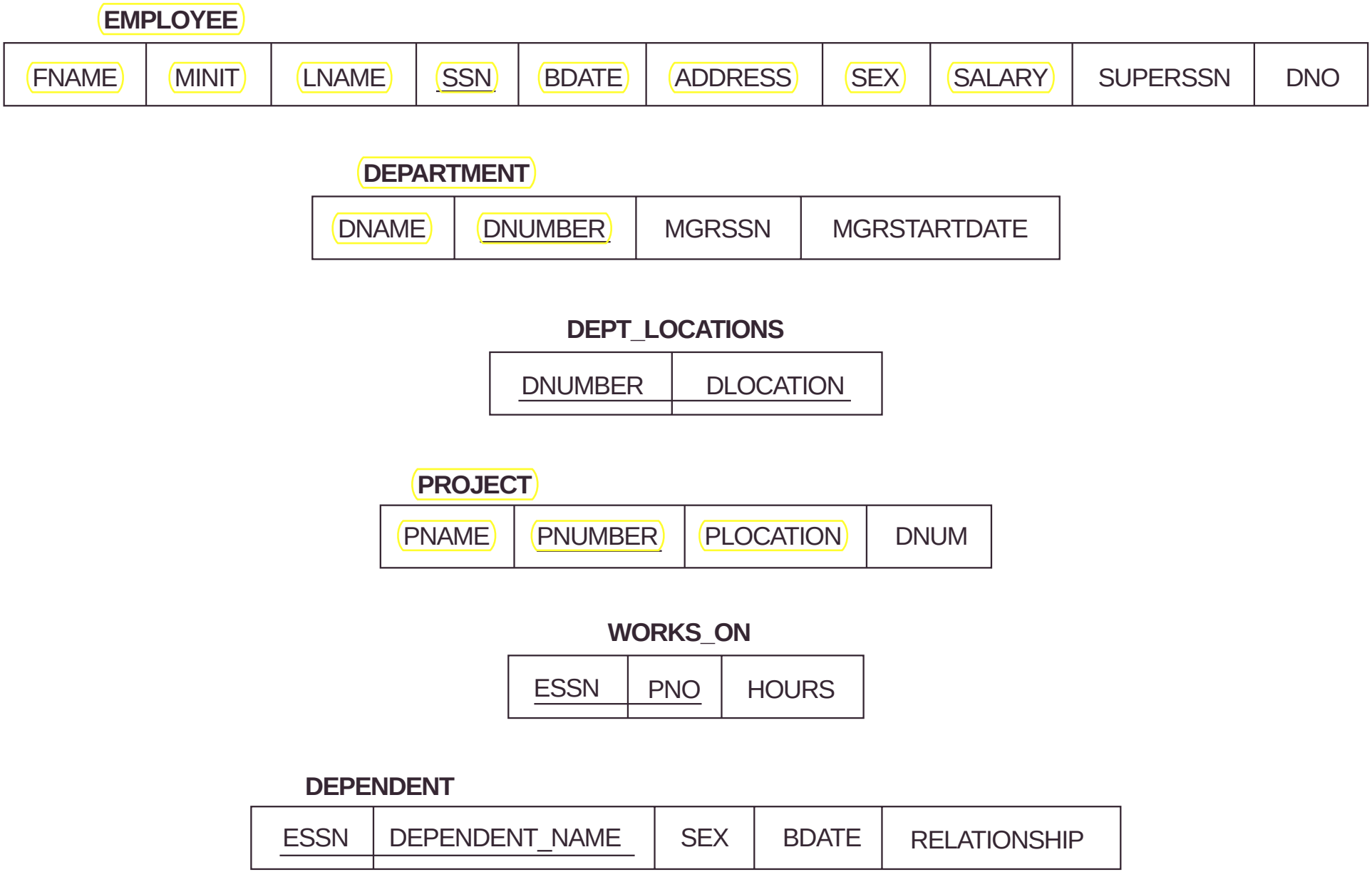
Figure 3.2 ER schema diagram for the company database.



Step 1: For each **regular entity type E**

- Create a relation R that includes all the simple attributes of E.
- Include all the simple component attributes of composite attributes.
- 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.

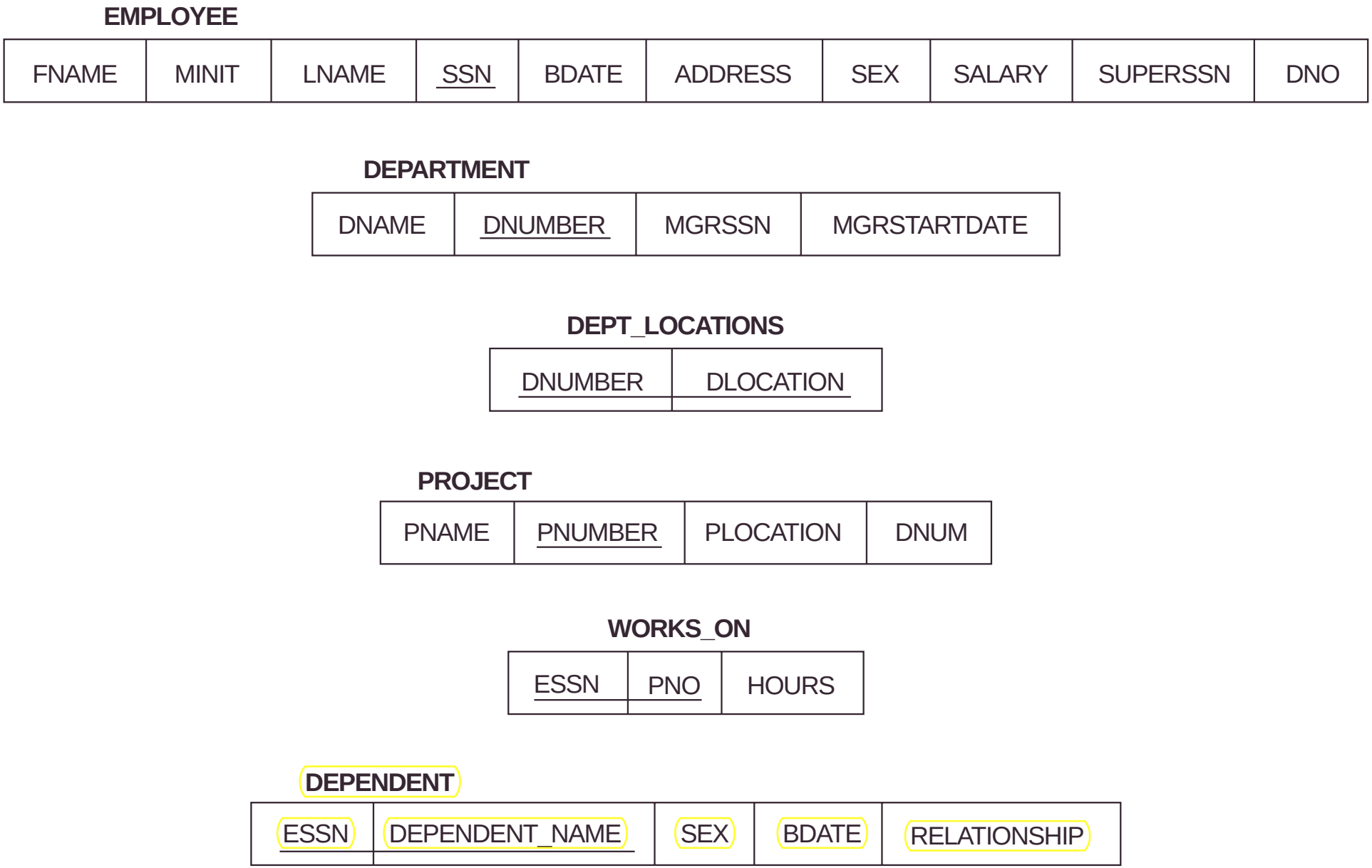
Figure 7.5 Schema diagram for the COMPANY relational database schema; the primary keys are underlined.



Step 2: For each **weak entity type W** with owner entity type E

- Create a relation R, and include all simple attributes and simple components of composite attributes of W as attributes of R.
- In addition, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).

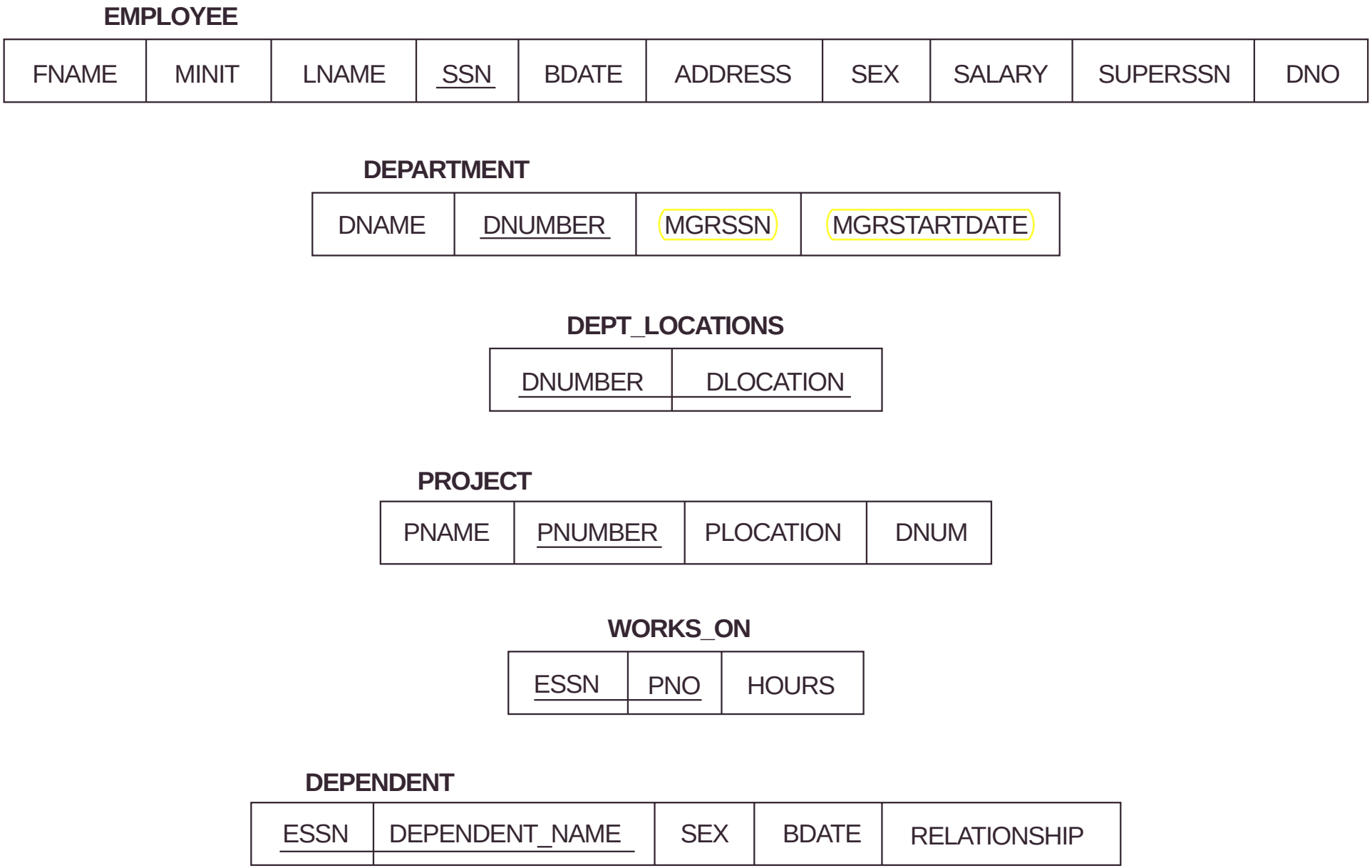
Figure 7.5 Schema diagram for the COMPANY relational database schema; the primary keys are underlined.



Step 3: For each **binary 1:1 relationship type R**

- Identify the relations S and T that correspond to the entity types participating in R. 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.
- Include the simple attributes of the 1:1 relationship type R as attributes of S.
- If both participations are total, we may merge the two entity types and the relationship into a single relation.

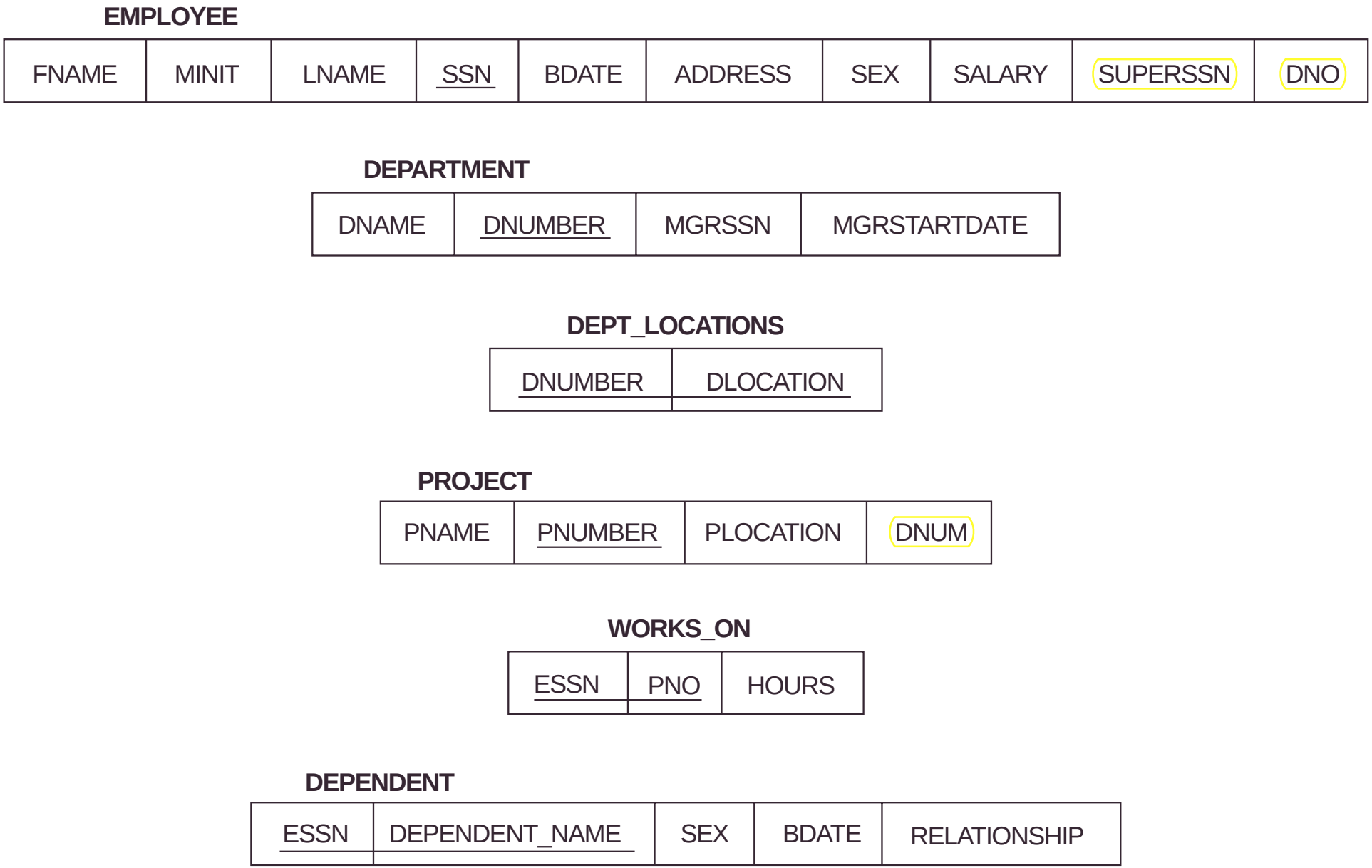
Figure 7.5 Schema diagram for the COMPANY relational database schema; the primary keys are underlined.



Step 4: For each **regular binary 1:N relationship type R**

- Identify the relation S that represents the participating entity type at the N-side of the relationship type.
- Include as foreign key in S the primary key of the relations T that represents the other entity type participating in R.
- Include any simple attributes of the 1:N relationship type as attributes of S.

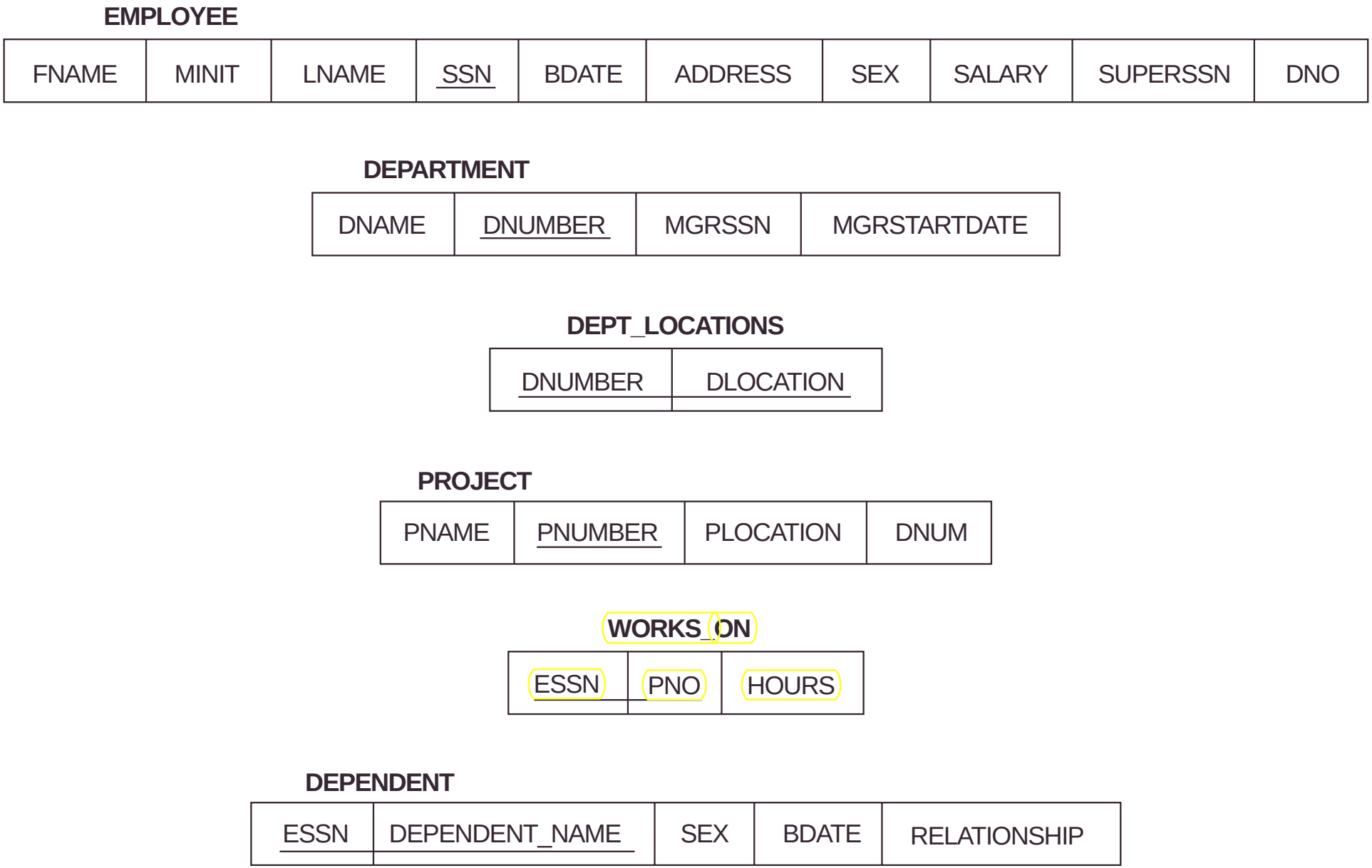
Figure 7.5 Schema diagram for the COMPANY relational database schema; the primary keys are underlined.



Step 5: For each **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 as attributes of S.

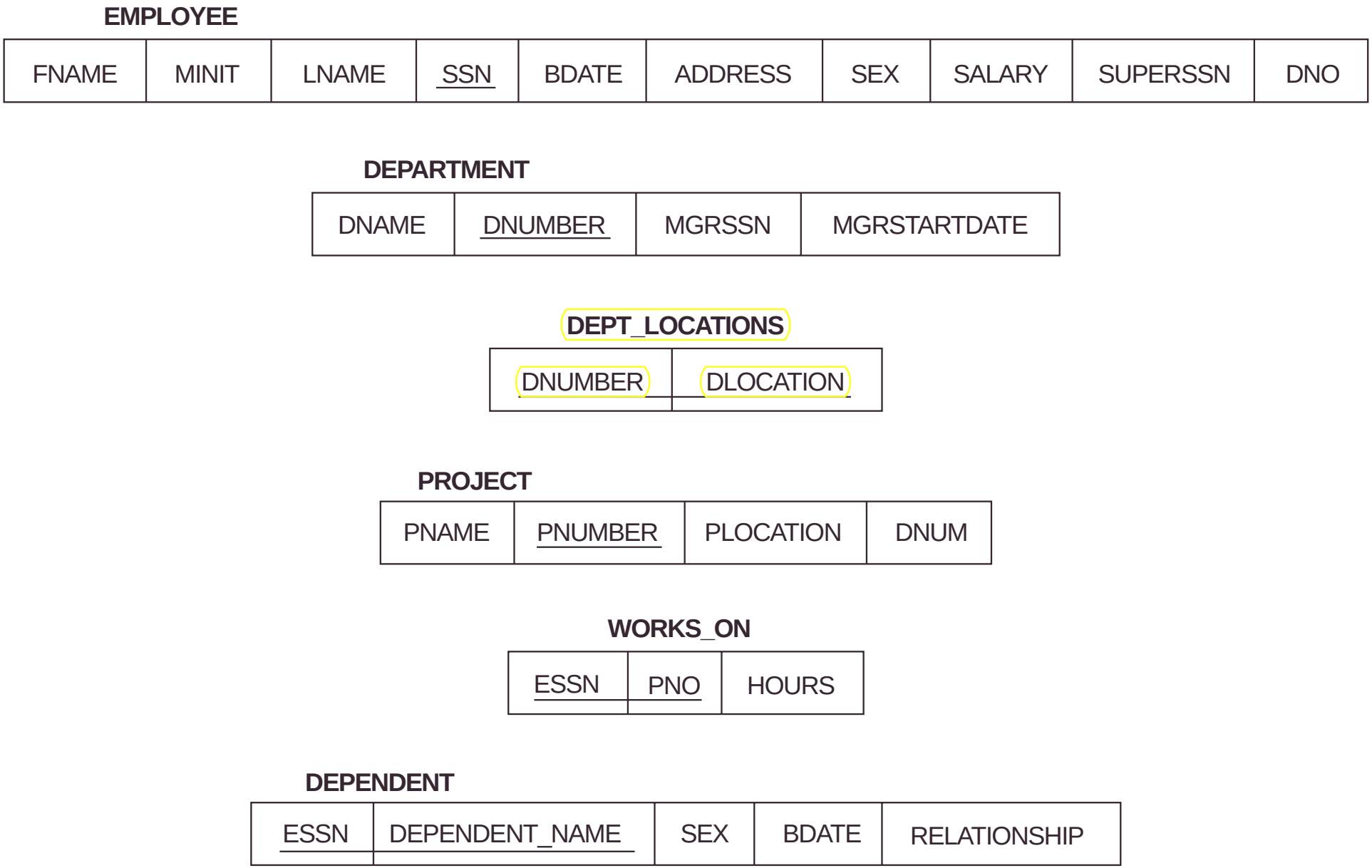
Figure 7.5 Schema diagram for the COMPANY relational database schema; the primary keys are underlined.



Step 6: For each **multi-valued attribute A**

- Create a new relation R that includes 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 or relationship type that has A as an attribute.
- The primary key of R is the combination of A and K. If a multi-valued attribute is composite, we include its components.

Figure 7.5 Schema diagram for the COMPANY relational database schema; the primary keys are underlined.



Step 7: For each **n-ary relationship type R**, n>2

- Create a new relation S to represent R.
- Include as foreign key attributes in the S the primary keys of the relations that represent the participating entity types.
- Also include any simple attributes of the n-ary relationship types as attributes of S.
- The primary key for S is usually a combination of all the foreign keys that reference the relations representing the participating entity types.

TERNARY RELATIONSHIPS

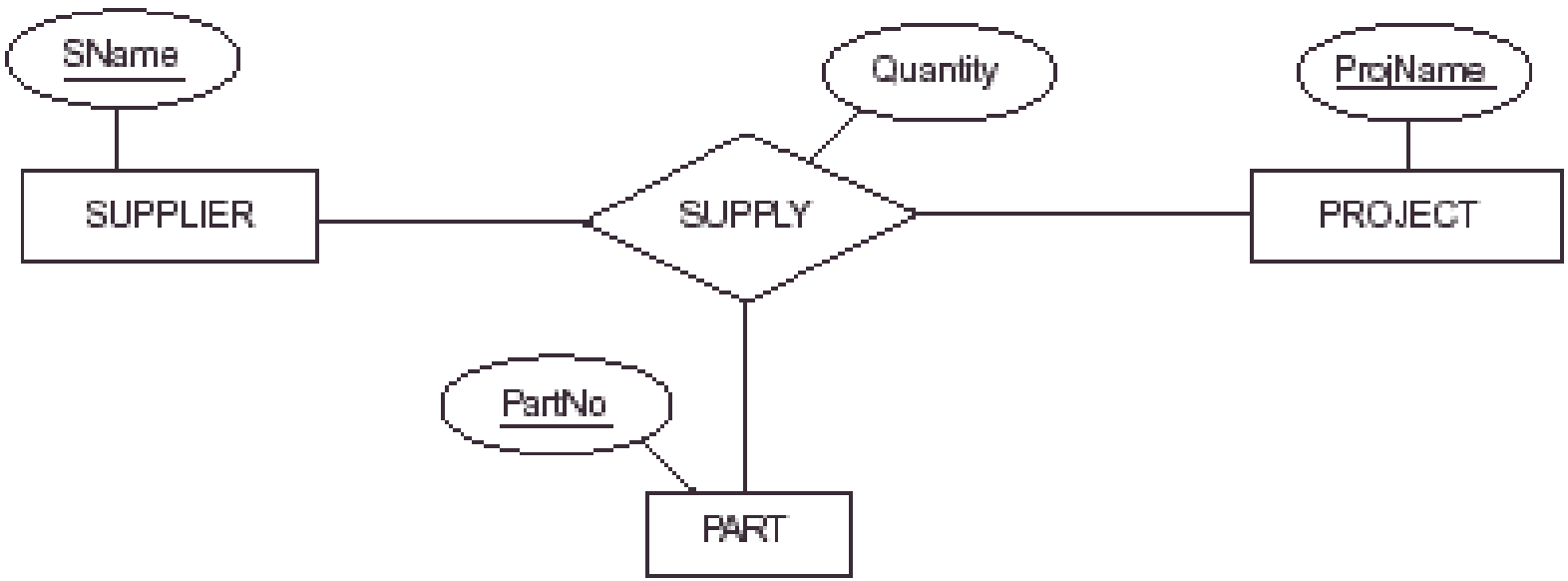
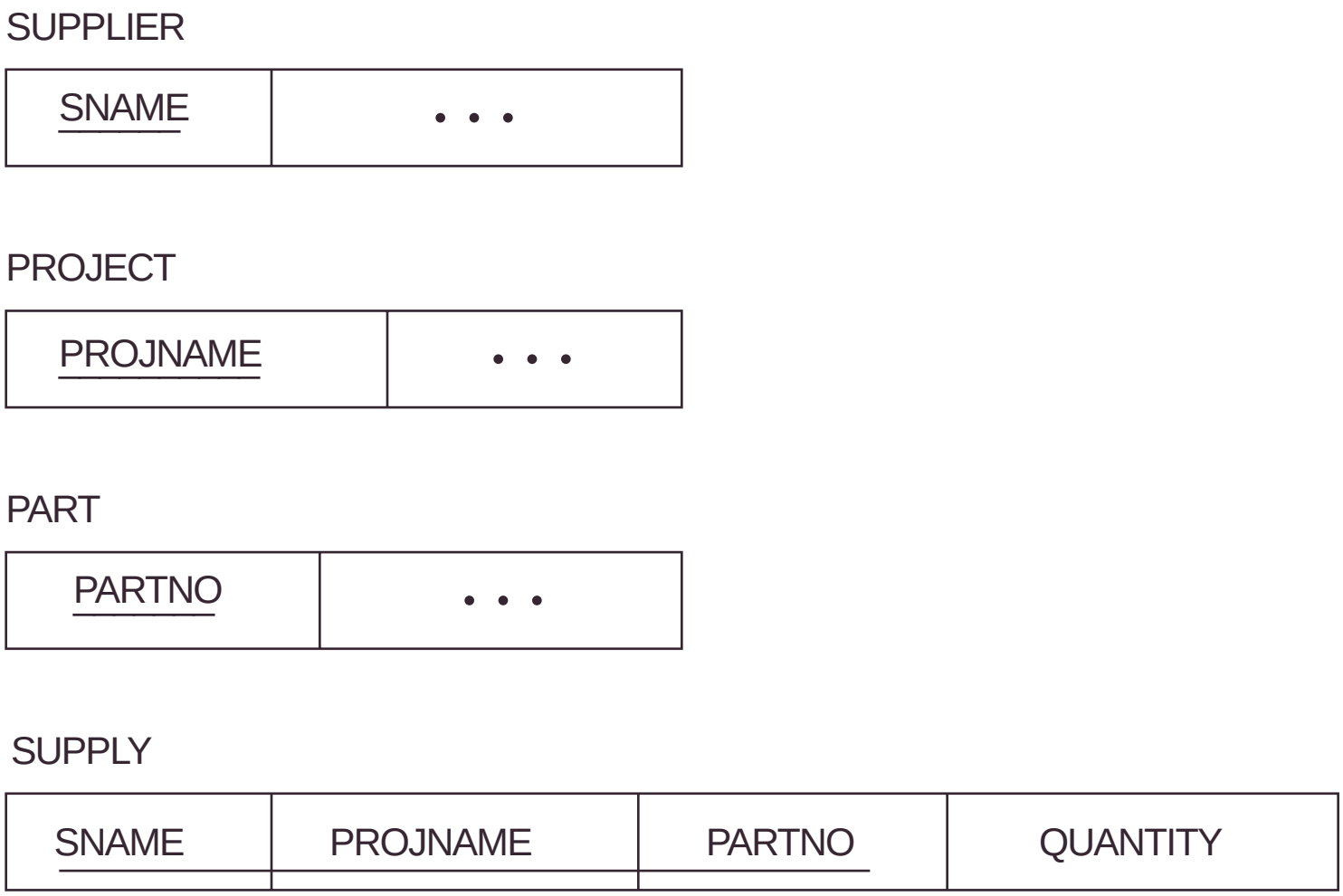


Figure 9.1 Mapping the n-ary relationship type SUPPLY from Figure 4.13(a).



- However, if the participation constraint (min,max) of one of the entity types E participating in the R has max =1, then the primary key of S can be the single foreign key attribute that references the relation E' corresponding to E
- This is because , in this case, each entity e in E will participate in at most one relationship instance of R and hence can uniquely identify that relationship instance.

Step 8: To convert each **super-class/sub-class** relationship into a relational schema you must use one of the four options available.

Let C be the super-class, K its primary key and $A_1, A_2, \ldots, A_n$ its remaining attributes and let $S_1, S_2, \ldots, S_m$ be the sub-classes.

Option 8A (multiple relation option):

- Create a relation L for C with attributes
  Attrs(L) = {K, A_1, A_2, …, A_n} and PK(L) = K.

- Create a relation L_i for each subclass S_i, $1 \leq i \leq m$, with the attributes
  ATTRS(L_i) = {K} U {attributes of S_i} and
  PK(L_i) = K.

- This option works for any constraints: disjoint or overlapping; total or partial.

Option 8B (multiple relation option):

- Create a relation L_i for each subclass S_i, $1 \leq i \leq m$, with
  $ATTRS(L_i) = \{\text{attributes of } S_i\} \cup \{K, A_1, A_2, \ldots, A_n\}$
  $PK(L_i) = K$
- This option works well only for disjoint and total constraints.
- If not disjoint, redundant values for inherited attributes.
- If not total, entity not belonging to any sub-class is lost.

Figure 4.4 An attribute-defined specialization on the JobType attribute of EMPLOYEE.

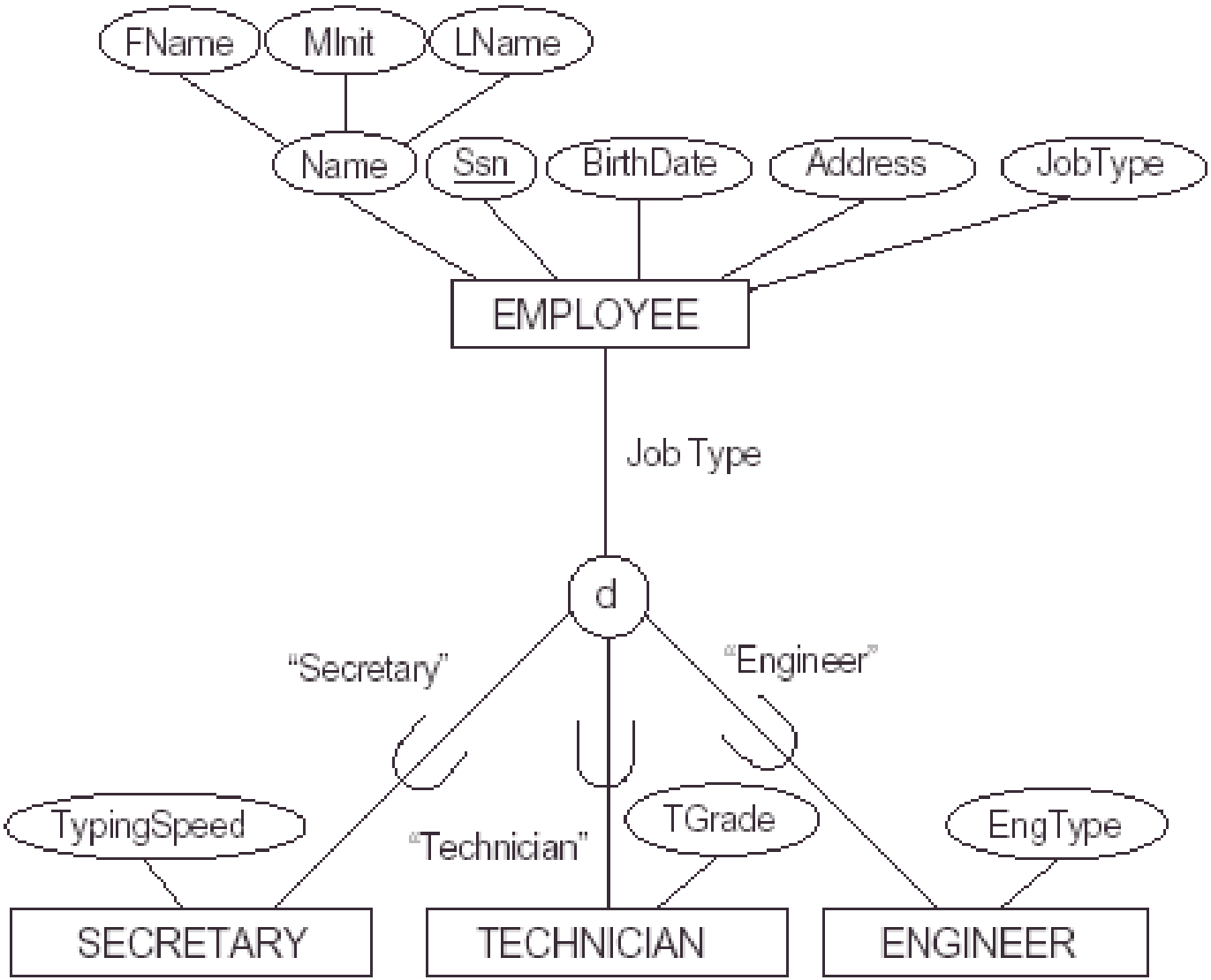


Figure 9.2 Options for mapping specializations (or generalizations) to relations. (a) Mapping the EER schema of Figure 4.4 to relations by using Option A. (b) Mapping the EER schema of Figure 4.3(b) into relations by using Option B. (c) Mapping the EER schema of Figure 4.4 by using Option C, with JobType playing the role of type attribute. (d) Mapping the EER schema of Figure 4.5 by using Option D, with two Boolean type fields Mflag and Pflag.

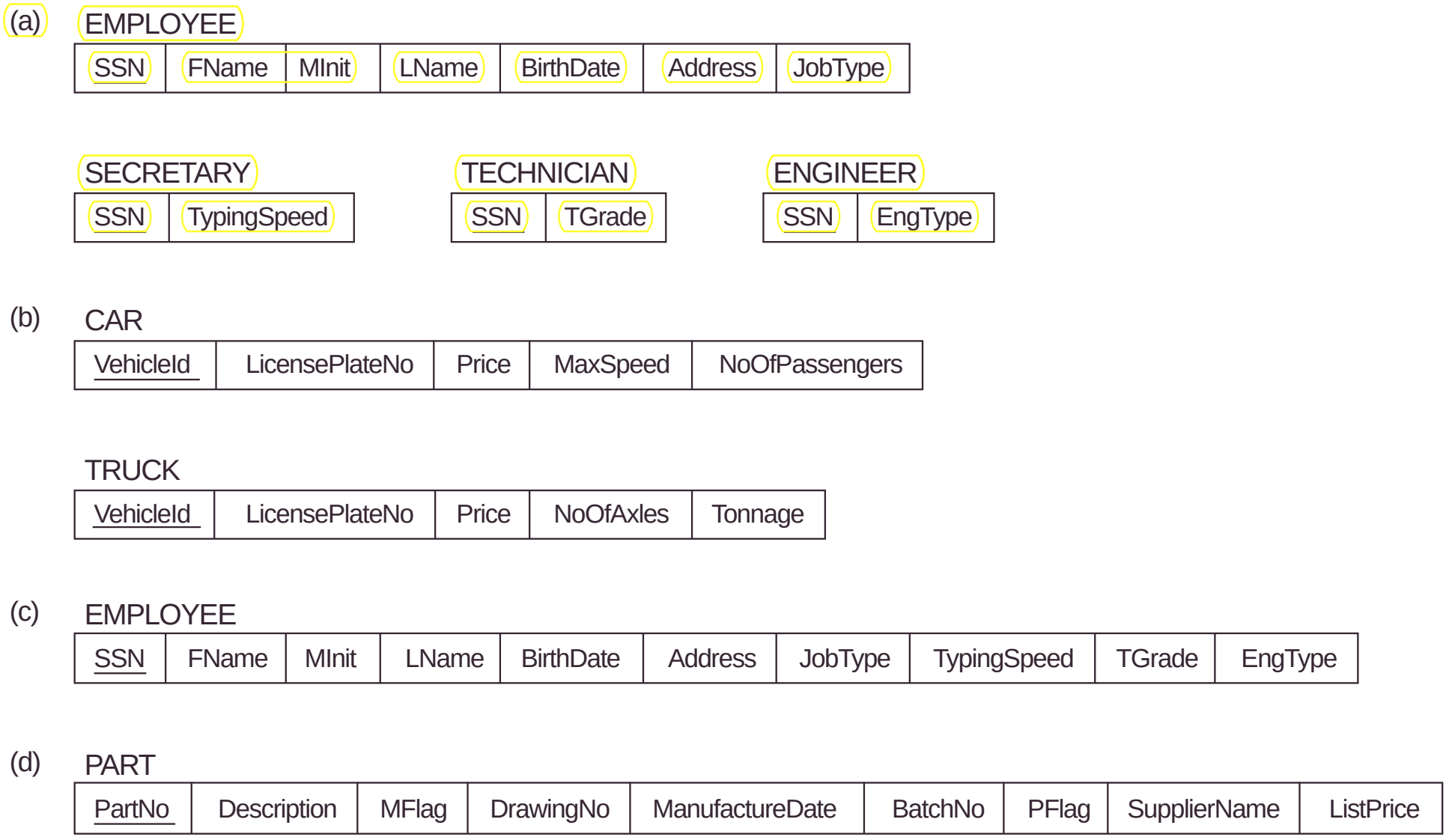


Figure 4.3 Examples of generalization. (a) Two entity types CAR and TRUCK. (b) Generalizing car and TRUCK into VEHICLE.

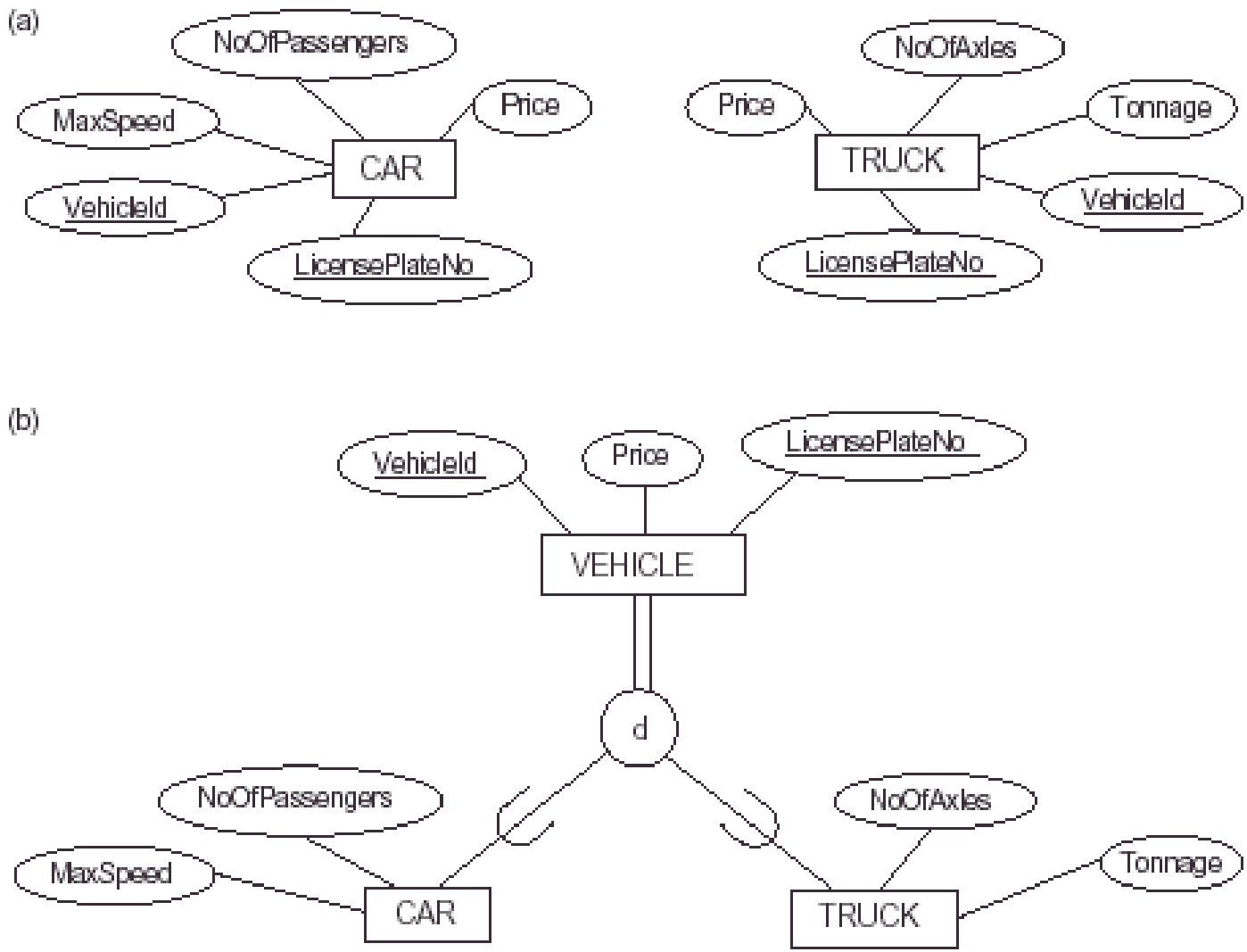


Figure 9.2 Options for mapping specializations (or generalizations) to relations. (a) Mapping the EER schema of Figure 4.4 to relations by using Option A. (b) Mapping the EER schema of Figure 4.3(b) into relations by using Option B. (c) Mapping the EER schema of Figure 4.4 by using Option C, with JobType playing the role of type attribute. (d) Mapping the EER schema of Figure 4.5 by using Option D, with two Boolean type fields Mflag and Pflag.

(a)

EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType

SECRETARY

SSN	TypingSpeed

TECHNICIAN

SSN	TGrade

ENGINEER

SSN	EngType

(b)

CAR

VehicleId	LicensePlateNo	Price	MaxSpeed	NoOfPassengers

TRUCK

VehicleId	LicensePlateNo	Price	NoOfAxles	Tonnage

(c)

EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType	TypingSpeed	TGrade	EngType

(d)

PART

PartNo	Description	MFlag	DrawingNo	ManufactureDate	BatchNo	PFlag	SupplierName	ListPrice

Option 8c (Single Relation Option)

- 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

- This option is for specialization whose subclasses are DISJOINT, and T is a **type** attribute that indicates the subclass to which each tuple belongs, if any. This option may generate a large number of null values.
- Not recommended if many specific attributes are defined in subclasses (will result in many null values!)

Figure 4.4 An attribute-defined specialization on the JobType attribute of EMPLOYEE.

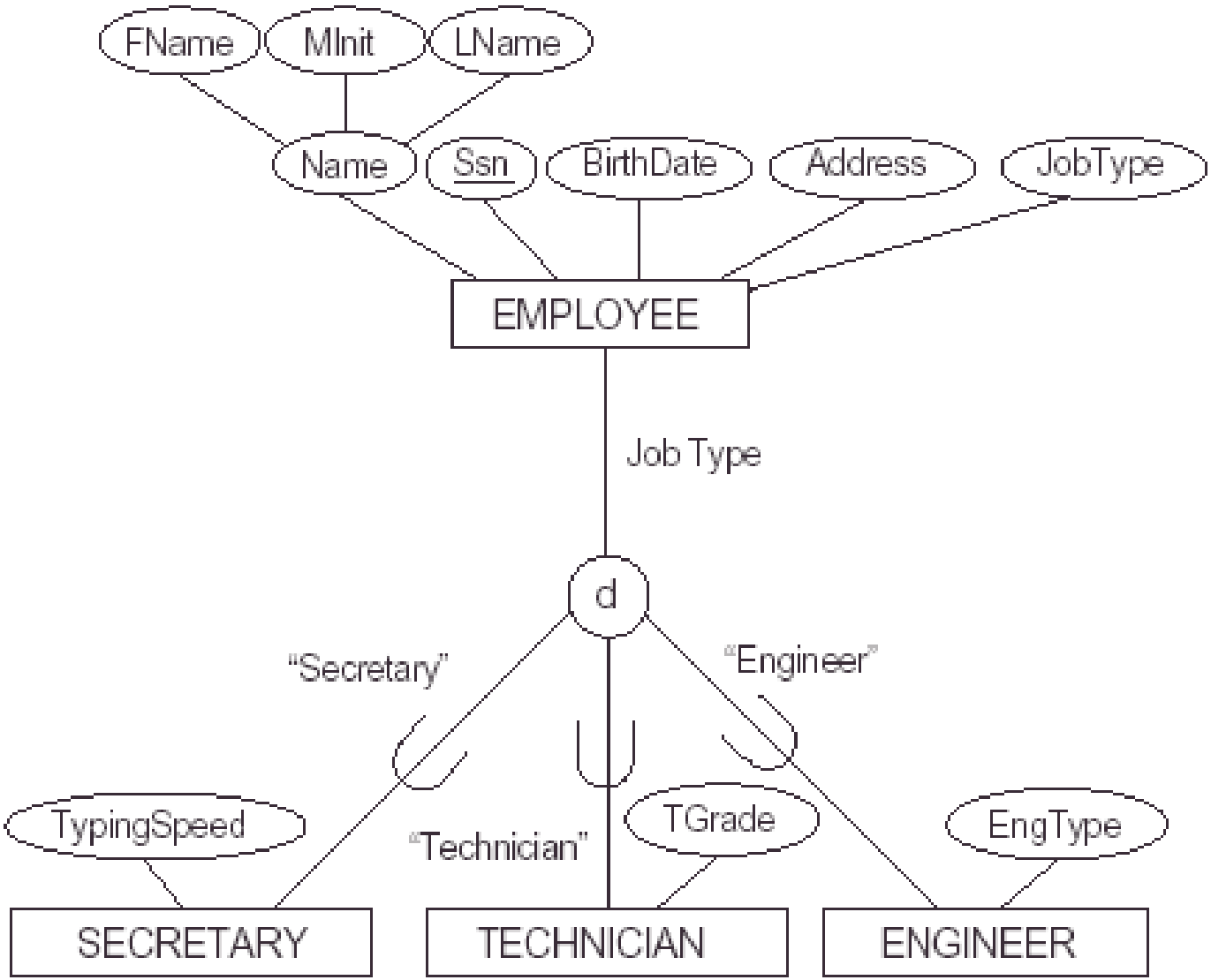


Figure 9.2 Options for mapping specializations (or generalizations) to relations. (a) Mapping the EER schema of Figure 4.4 to relations by using Option A. (b) Mapping the EER schema of Figure 4.3(b) into relations by using Option B. (c) Mapping the EER schema of Figure 4.4 by using Option C, with JobType playing the role of type attribute. (d) Mapping the EER schema of Figure 4.5 by using Option D, with two Boolean type fields Mflag and Pflag.

(a)

EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType

SECRETARY

SSN	TypingSpeed

TECHNICIAN

SSN	TGrade

ENGINEER

SSN	EngType

(b)

CAR

VehicleId	LicensePlateNo	Price	MaxSpeed	NoOfPassengers

TRUCK

VehicleId	LicensePlateNo	Price	NoOfAxles	Tonnage

(c)

EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType	TypingSpeed	TGrade	EngType

(d)

PART

PartNo	Description	MFlag	DrawingNo	ManufactureDate	BatchNo	PFlag	SupplierName	ListPrice

Option 8d (Single Relation Option)

- Create a single relation schema L with attributes

  Attrs(L) = $\{K, A_1, \ldots, A_n\}$ U
  $\{$attributes of $S_1\}$ U… U
  $\{$attributes of $S_m\}$ U $\{T_1, \ldots, T_n\}$

  and PK(L)=K
- This option is for specialization whose subclasses are overlapping, and each T_i, $1 \leq i \leq m$, is a Boolean attribute indicating whether a tuple belongs to subclass S_i.
- This option could be used for disjoint subclasses too.

Figure 4.5 Notation for specialization with overlapping (nondisjoint) subclasses.

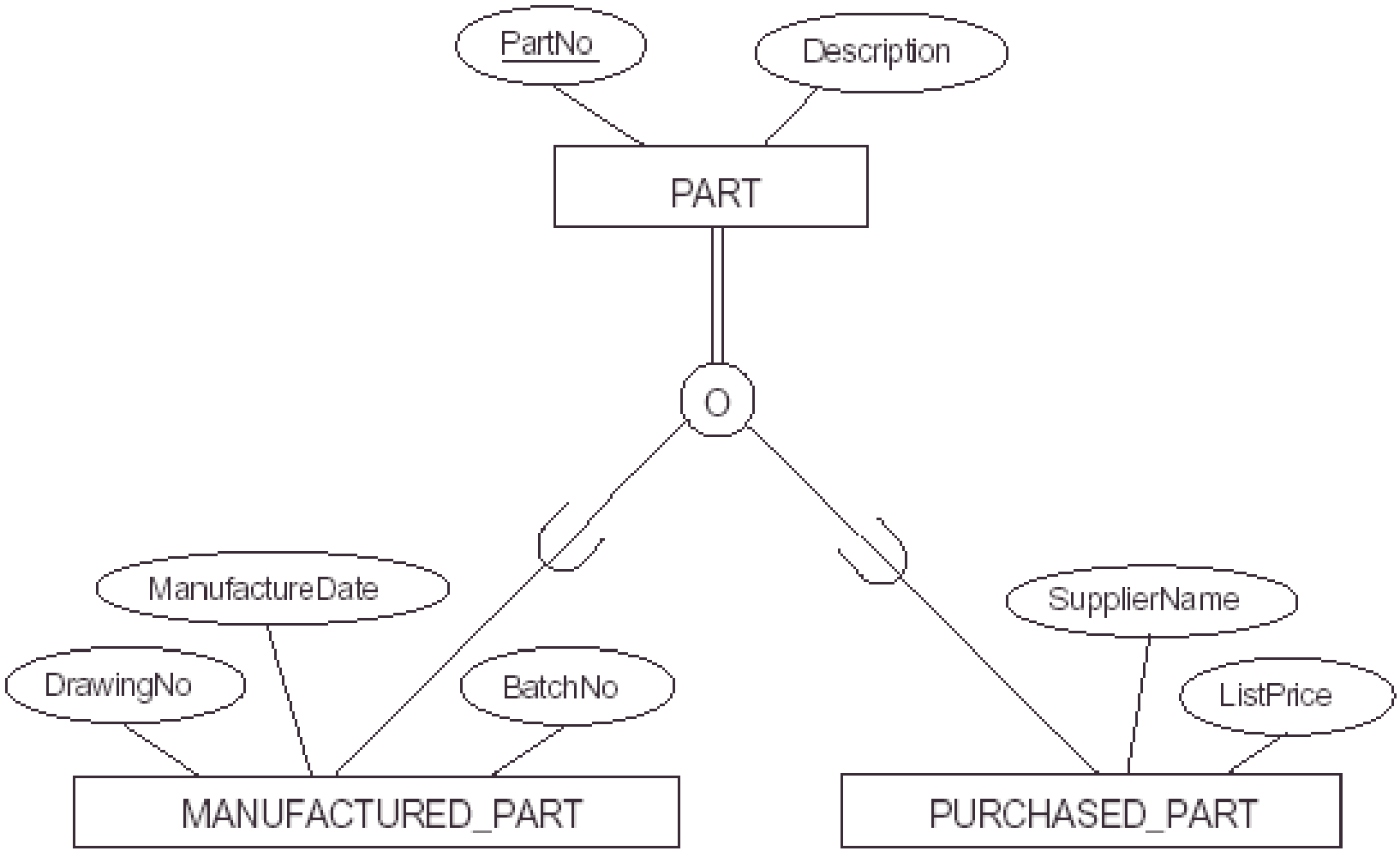
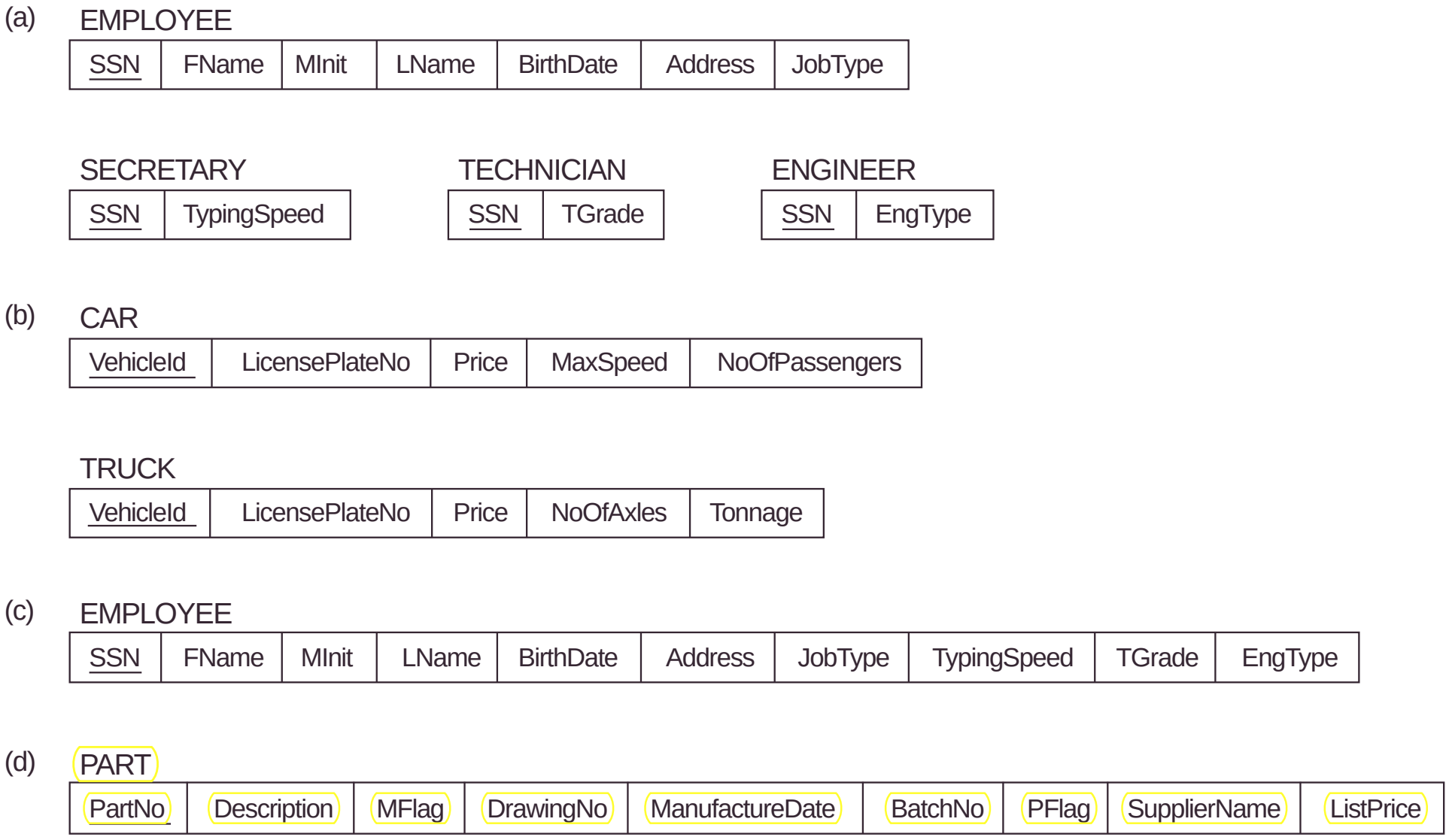


Figure 9.2 Options for mapping specializations (or generalizations) to relations. (a) Mapping the EER schema of Figure 4.4 to relations by using Option A. (b) Mapping the EER schema of Figure 4.3(b) into relations by using Option B. (c) Mapping the EER schema of Figure 4.4 by using Option C, with JobType playing the role of type attribute. (d) Mapping the EER schema of Figure 4.5 by using Option D, with two Boolean type fields Mflag and Pflag.



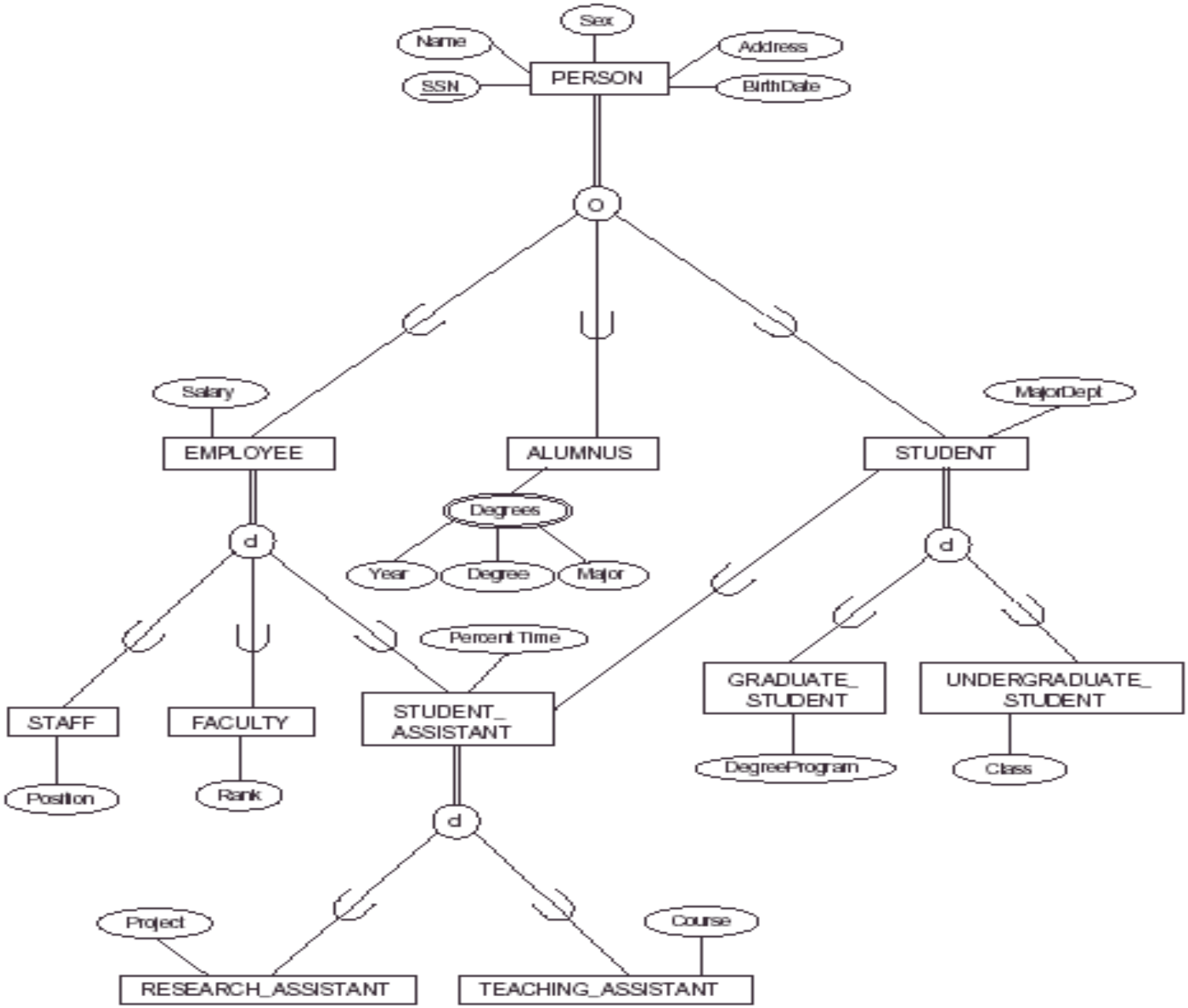
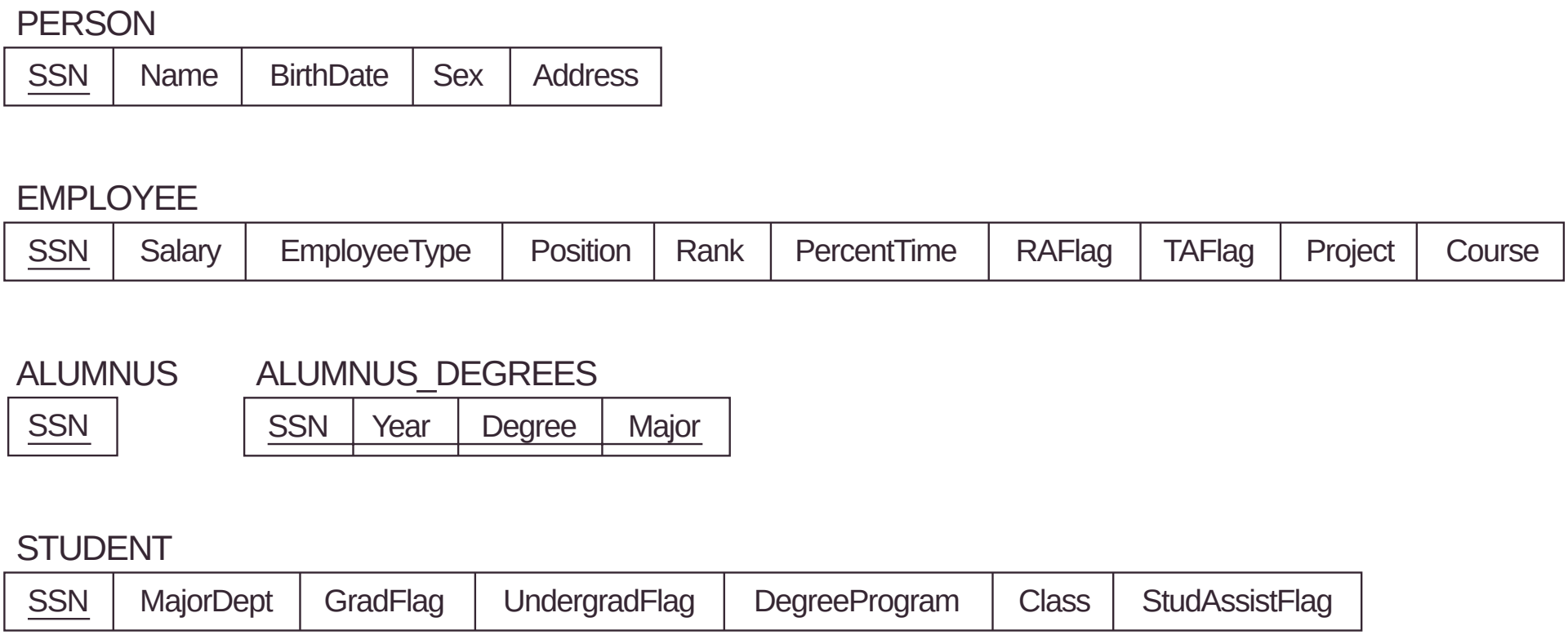

Sex
Name
Address
PERSON
SSN
BirthDate
O
Salary
MajorDept
EMPLOYEE
ALUMNUS
STUDENT
Degrees
d
d
Year
Degree
Major
Percent Time
STAFF
FACULTY
STUDENT_
ASSISTANT
GRADUATE_
STUDENT
UNDERGRADUATE_
STUDENT
Position
Rank
DegreeProgram
Class
d
Project
Course
RESEARCH_ASSISTANT
TEACHING_ASSISTANT

Figure 9.3 Mapping the EER specialization lattice shown in Figure 4.7 using multiple options.



Option 8A for

•PERSON/{EMPLOYEE,ALUMNUS,STUDENT}

Option 8C for

•EMPLOYEE/{STAFF,FACULTY,STUDENT_ASSISTANT}

Option 8D for

•STUDENT_ASSISTANT/{RESEARCH_ASSISTANT,
TEACHING_ASSISTANT}
•STUDENT/{STUDENT_ASSISTANT}
•STUDENT/{GRADUATE_ASSISTANT,
UNDERGRADUATE_STUDENT}