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

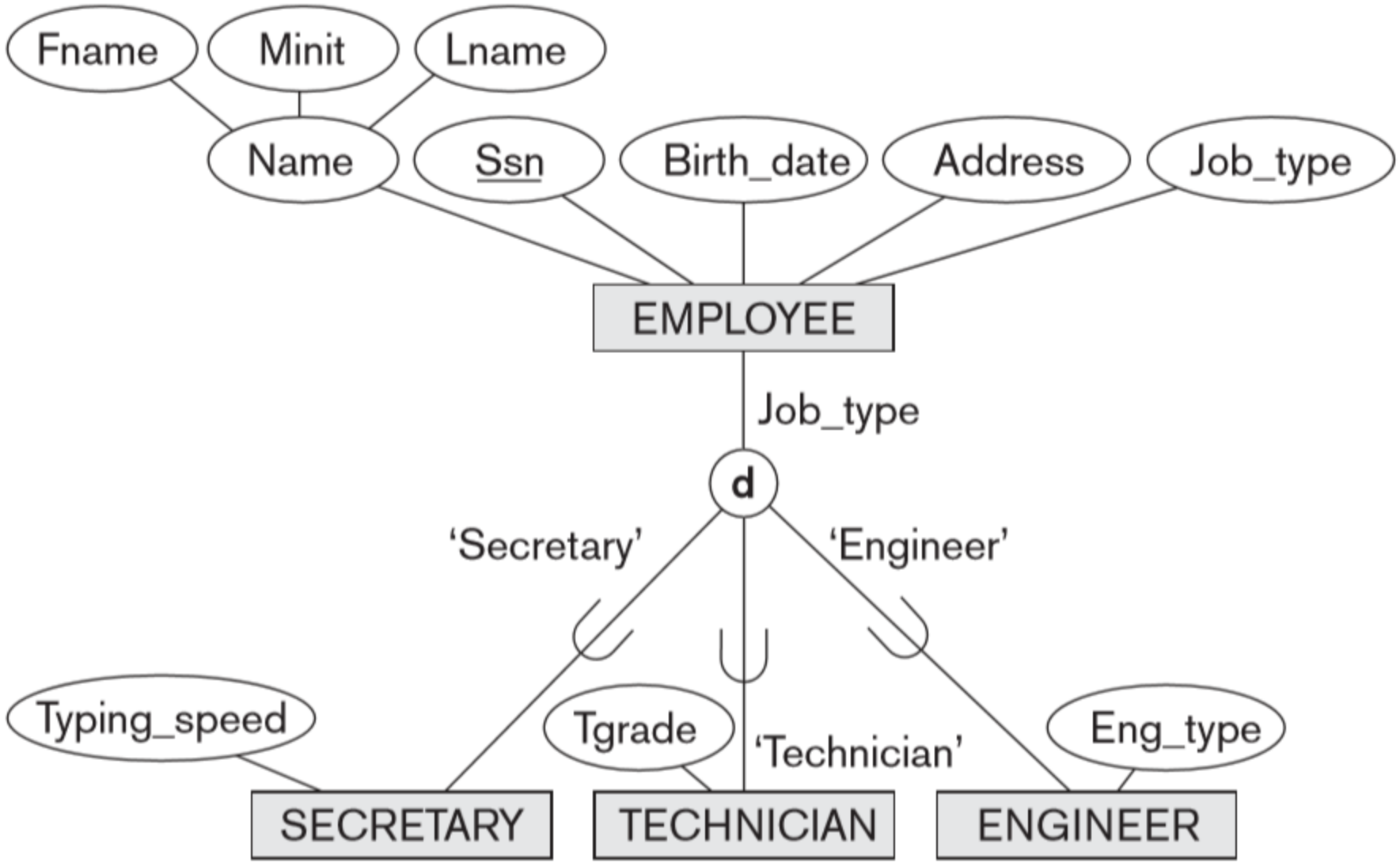
Part 2

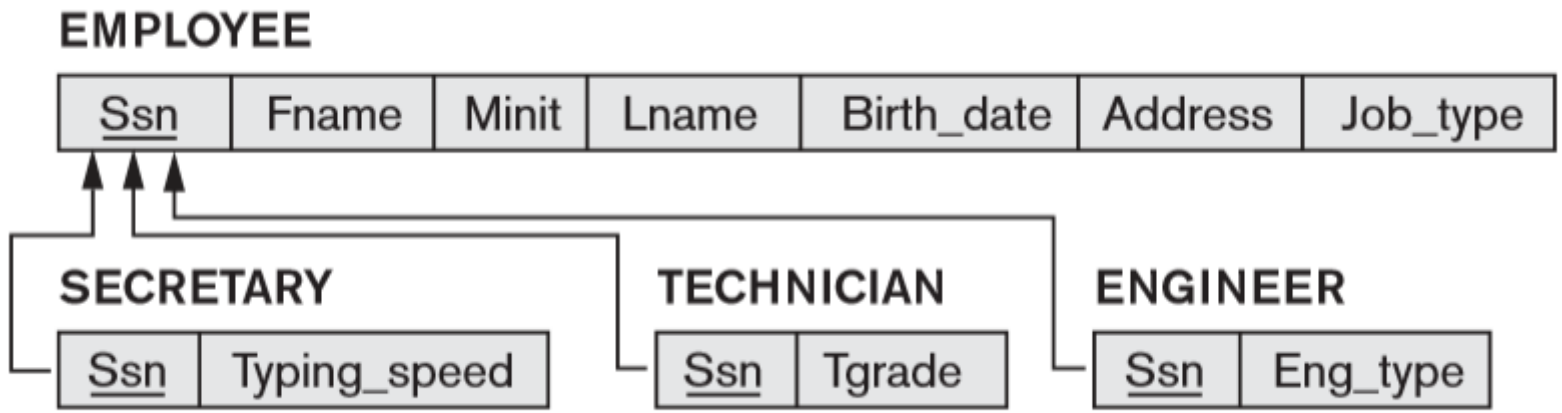
Mapping EER Model Constructs to Relations

Mapping of Specialization or Generalization

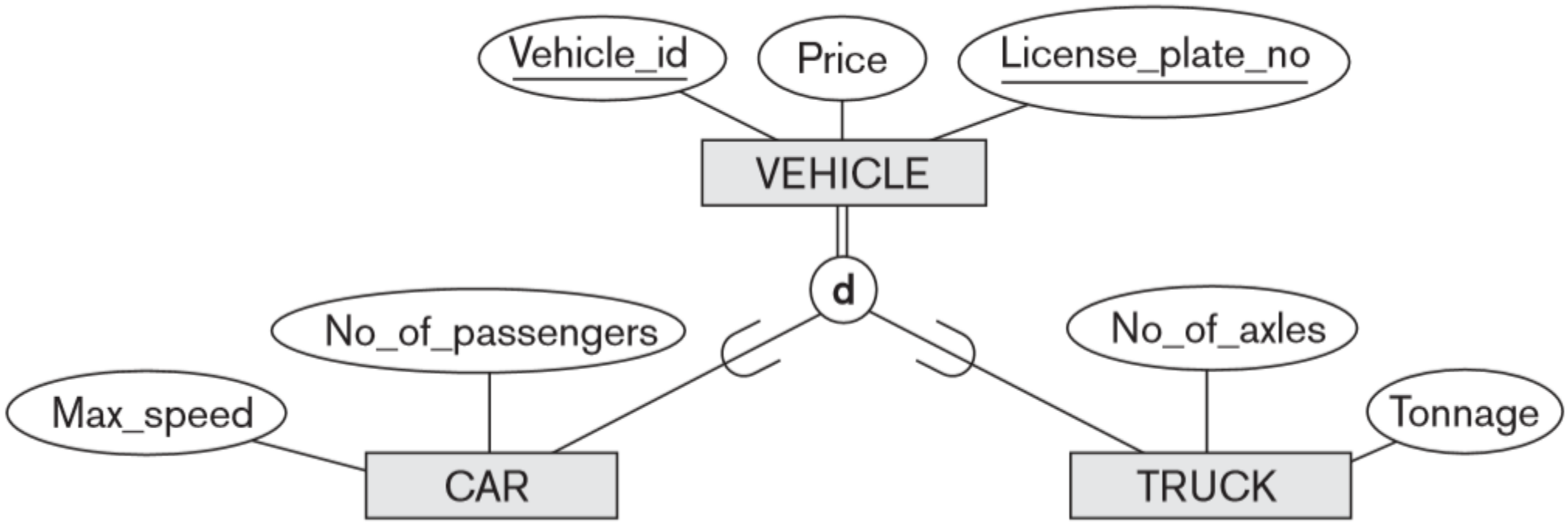
- **Step 8: Options for Mapping Specialization or Generalization.** Convert each specialization with m subclasses $\{S_1, S_2, ..., S_m\}$ and (generalized) superclass C, where the attributes of C are $\{k, a_1, ..., a_n\}$ and k is the (primary) key, into relation schemas.

- **Option 8A: Multiple relations—superclass and subclasses.** Create a relation L for C with attributes Attrs(L) = {k, a_1, ..., 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} ∪ {attributes of S_i} and PK(L_i) = k. This option works for any specialization (total or partial, disjoint or overlapping).


Fname
Minit
Lname
Name
Ssn
Birth_date
Address
Job_type
EMPLOYEE
Job_type
d
'Secretary'
'Engineer'
'Technician'
Typing_speed
Tgrade
Eng_type
SECRETARY
TECHNICIAN
ENGINEER


EMPLOYEE
Ssn
Fname
Minit
Lname
Birth_date
Address
Job_type
SECRETARY
Ssn
Typing_speed
TECHNICIAN
Ssn
Tgrade
ENGINEER
Ssn
Eng_type

- **Option 8B: Multiple relations—subclass relations only.** Create a relation L_i for each subclass S_i, $1 \leq i \leq m$, with the attributes Attrs(L_i) = {attributes of S_i} ∪ {k, a_1, ..., a_n} and PK(L_i) = 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.


Vehicle_id
Price
License_plate_no
VEHICLE
d
No_of_passengers
Max_speed
CAR
No_of_axles
Tonnage
TRUCK

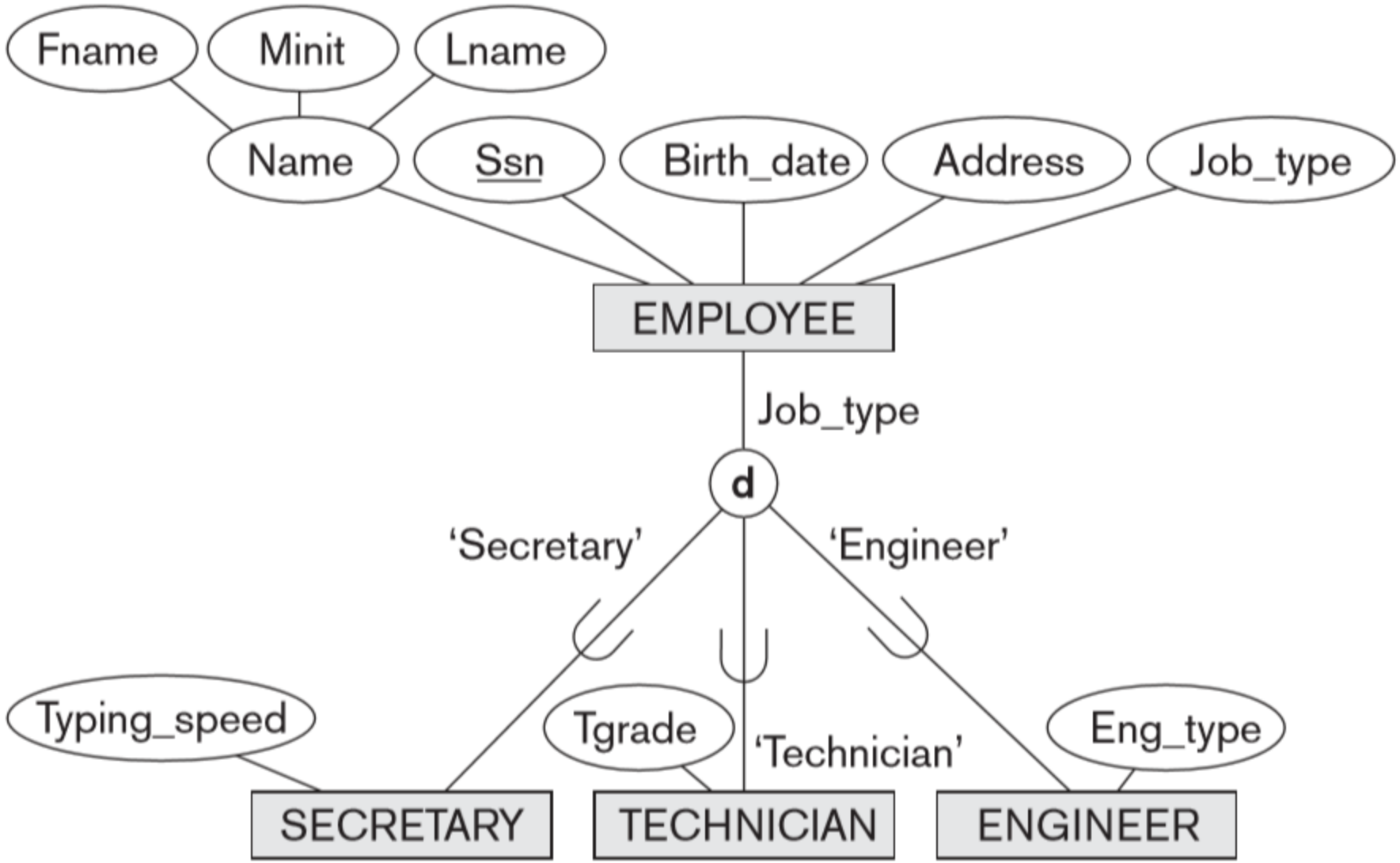
CAR

Vehicle_id	License_plate_no	Price	Max_speed	No_of_passengers

TRUCK

Vehicle_id	License_plate_no	Price	No_of_axles	Tonnage

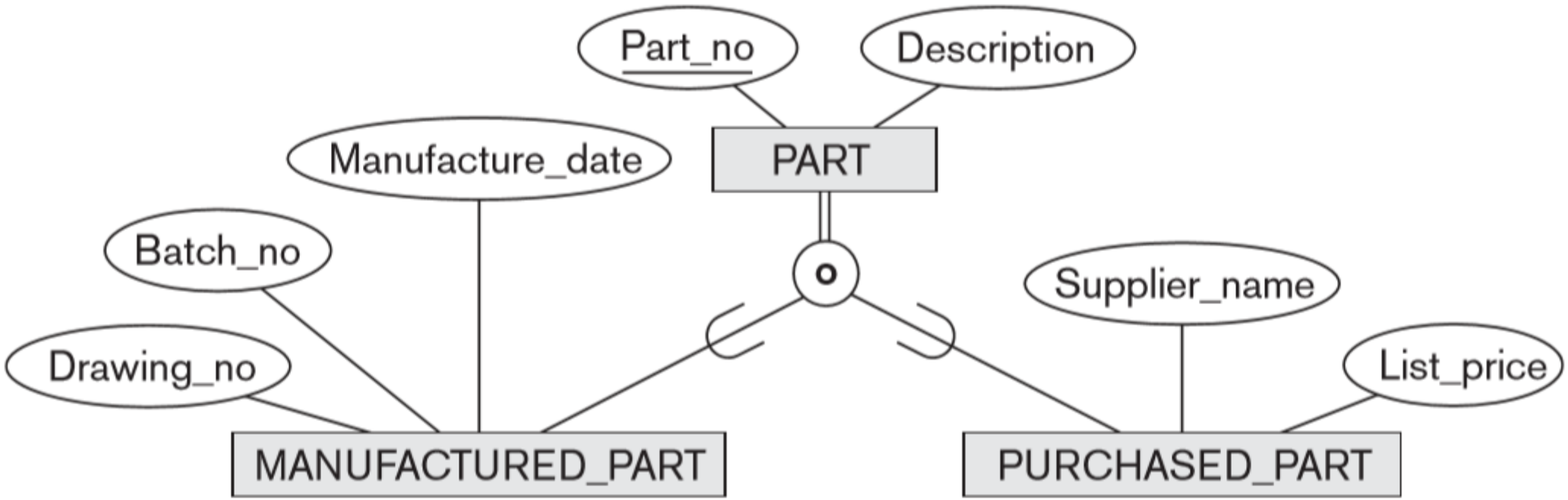
- **Option 8C:Single relation with one type attribute.** Create a single relation L with attributes Attrs(L) = {k, a_1, ..., a_n} ∪ {attributes of S_1} ∪ ... ∪ {attributes of S_m} ∪ {t} and PK(L) = k. The attribute t is called a **type** (or **discriminating**) attribute whose value indicates the subclass 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.


Fname
Minit
Lname
Name
Ssn
Birth_date
Address
Job_type
EMPLOYEE
Job_type
d
'Secretary'
'Engineer'
'Technician'
Typing_speed
Tgrade
Eng_type
SECRETARY
TECHNICIAN
ENGINEER

EMPLOYEE

Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type

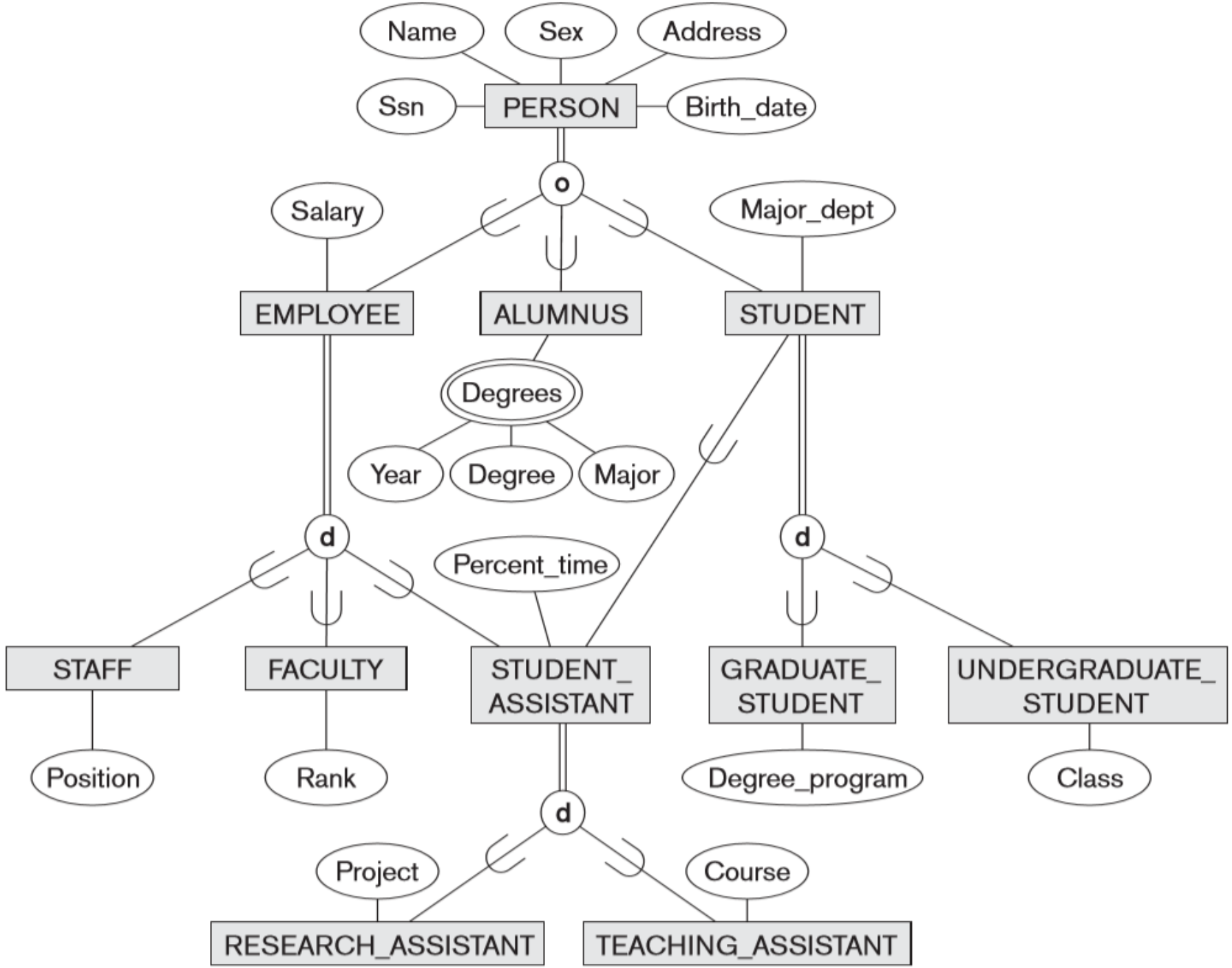
- **Option 8D: Single relation with multiple type attributes.** Create a single relation schema L with attributes Attrs(L) = {k, a_1, ..., a_n} ∪ {attributes of S_1} ∪ ...∪ {attributes of S_m} ∪ {t_1, t_2, ..., t_m} and PK(L) = k. Each t_i, $1 \leq i \leq m$, is a **Boolean type attribute** indicating whether a tuple belongs to subclass S_i. This option is used for a specialization whose subclasses are *overlapping* (but will also work for a disjoint specialization).

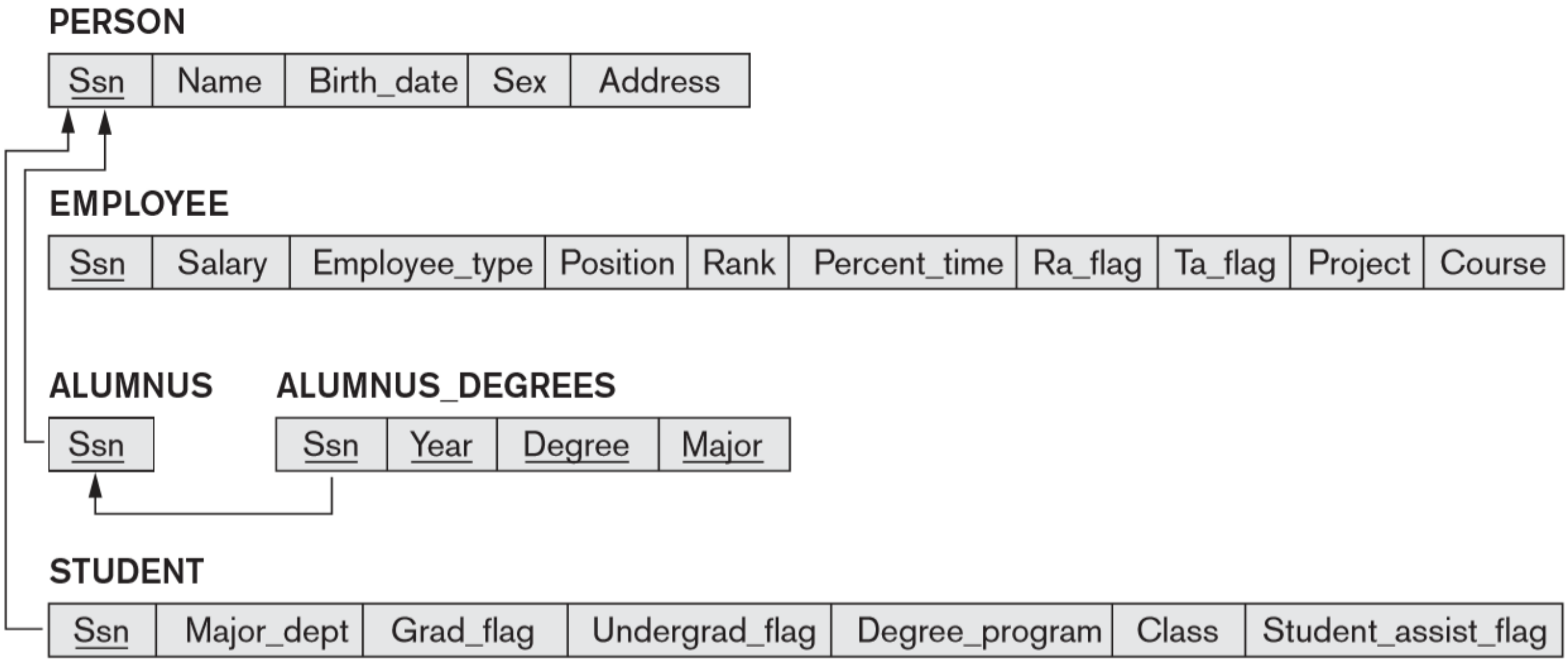

Part_no
Description
Manufacture_date
PART
Batch_no
o
Supplier_name
Drawing_no
List_price
MANUFACTURED_PART
PURCHASED_PART

PART

Part_no	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price

- When we have a multilevel specialization (or generalization) hierarchy or lattice, we do not have to follow the same mapping option for all the specializations. Instead, we can use one mapping option for part of the hierarchy or lattice and other options for other parts.


Name
Sex
Address
Ssn
PERSON
Birth_date
o
Salary
Major_dept
EMPLOYEE
ALUMNUS
STUDENT
Degrees
Year
Degree
Major
d
Percent_time
d
STAFF
FACULTY
STUDENT_
ASSISTANT
GRADUATE_
STUDENT
UNDERGRADUATE_
STUDENT
Position
Rank
Degree_program
Class
d
Project
Course
RESEARCH_ASSISTANT
TEACHING_ASSISTANT

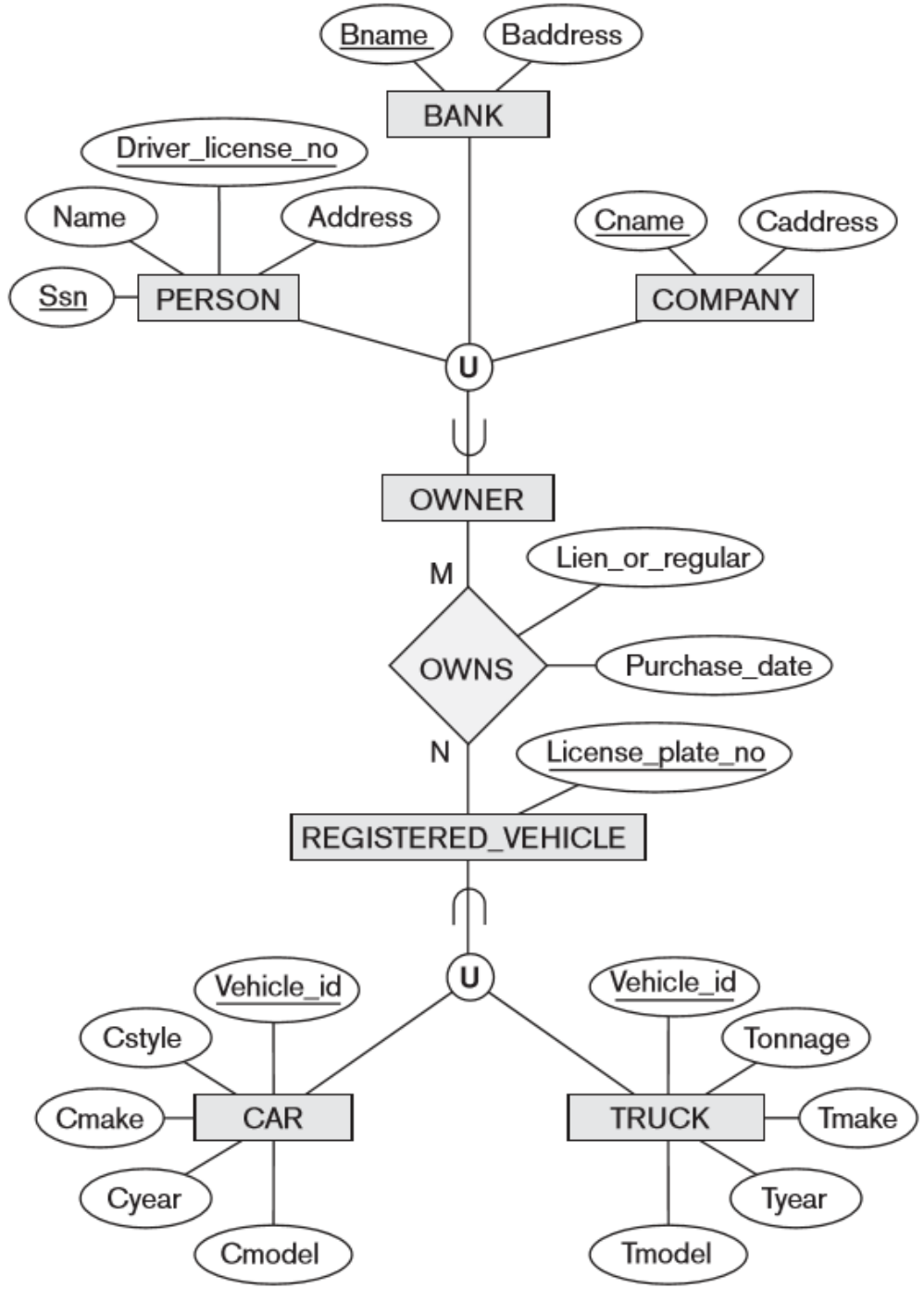

PERSON
Ssn
Name
Birth_date
Sex
Address
EMPLOYEE
Ssn
Salary
Employee_type
Position
Rank
Percent_time
Ra_flag
Ta_flag
Project
Course
ALUMNUS
Ssn
ALUMNUS_DEGREES
Ssn
Year
Degree
Major
STUDENT
Ssn
Major_dept
Grad_flag
Undergrad_flag
Degree_program
Class
Student_assist_flag

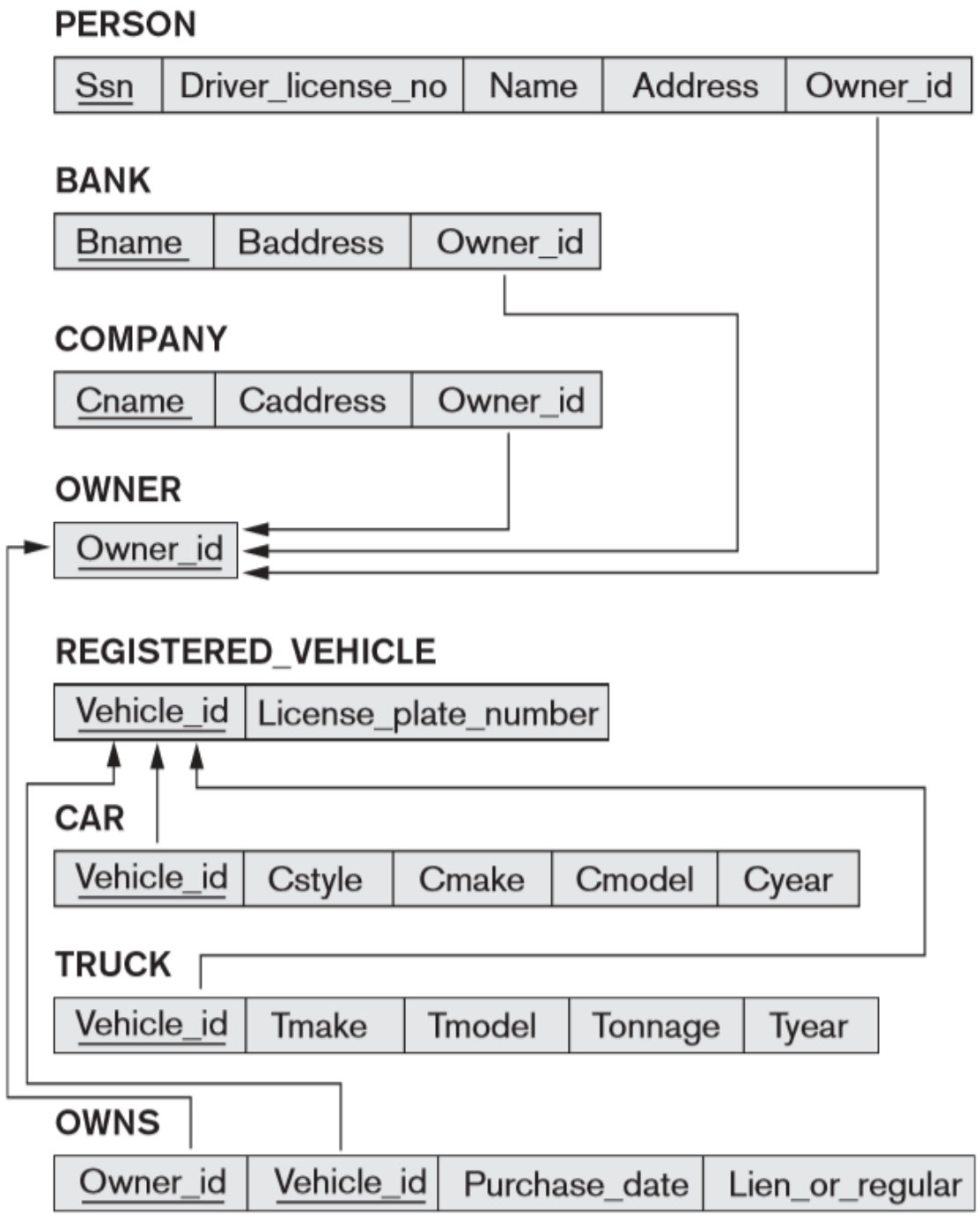
Mapping of Shared Subclasses (Multiple Inheritance)

- We can apply any of the options discussed in step 8 to a shared subclass, subject to the restrictions discussed in step 8 of the mapping algorithm.

Mapping of Categories (Union Types)

- **Step 9: Mapping of Union Types (Categories).** For mapping a category whose defining superclasses have different keys, it is customary to specify a new key attribute, called a **surrogate key**, when creating a relation to correspond to the category.


Bname
Baddress
BANK
Driver_license_no
Name
Address
Ssn
PERSON
Cname
Caddress
COMPANY
U
OWNER
M
Lien_or_regular
OWNS
Purchase_date
N
License_plate_no
REGISTERED_VEHICLE
U
Vehicle_id
Cstyle
Cmake
CAR
Cyear
Cmodel
Vehicle_id
Tonnage
TRUCK
Tmake
Tyear
Tmodel


PERSON
Ssn
Driver_license_no
Name
Address
Owner_id
BANK
Bname
Baddress
Owner_id
COMPANY
Cname
Caddress
Owner_id
OWNER
Owner_id
REGISTERED_VEHICLE
Vehicle_id
License_plate_number
CAR
Vehicle_id
Cstyle
Cmake
Cmodel
Cyear
TRUCK
Vehicle_id
Tmake
Tmodel
Tonnage
Tyear
OWNS
Owner_id
Vehicle_id
Purchase_date
Lien_or_regular