



BITS, PILANI – K. K. BIRLA GOA CAMPUS

# Database Systems and Applications (IS F243)

by

**Mrs. Shubhangi Gawali**

Dept. of CS and IS



- A General Hospital consists of a number of specialized departments (such as Maternity, Paediatric, Oncology, etc).
- Each department hosts a number of patients, who are admitted on the recommendation of doctors employed by hospital.
- There are 10 doctors, 3 nurses and 5 ward boys. Each staff has unique staff id.
- Doctors are specialists in some branch of medicine. Permanent doctors get fixed salary. Personal information like name, address, date of birth, etc. required.
- Consulting doctors visits at fixed time every day. Information like name, contact number, specialization, charges etc are required.
- On admission, the personal details of every patient like name, address, relative's name and address, patient's blood group, reason of admission, etc are recorded.
- Patients are admitted to rooms of different types, per day charges depend on room type.
- Various labs in hospital, where several tests are conducted on patient. Each test has fixed charges.
- A separate register is to be held to store the information of the tests undertaken and the results of a prescribed treatment. A number of tests may be conducted for each patient.
- Draw an extended E-R diagram for the system.

# **Relational Database Design by Mapping ER- and EER- model to Relational Model**

# Mapping ER-to-Relational Model

Step 1: Mapping of Strong Entity sets

Step 2: Mapping of Weak Entity sets

Step 3: Mapping Unary Relationship set.

Step 4: Mapping of Binary 1:1 Relationship set

Step 5: Mapping of Binary 1:N Relationship set.

Step 6: Mapping of Binary M:N Relationship set.

Step 7: Mapping of Multivalued attributes.

Step 8: Mapping of N-ary Relationship set.

# Mapping EER-to-Relational Model

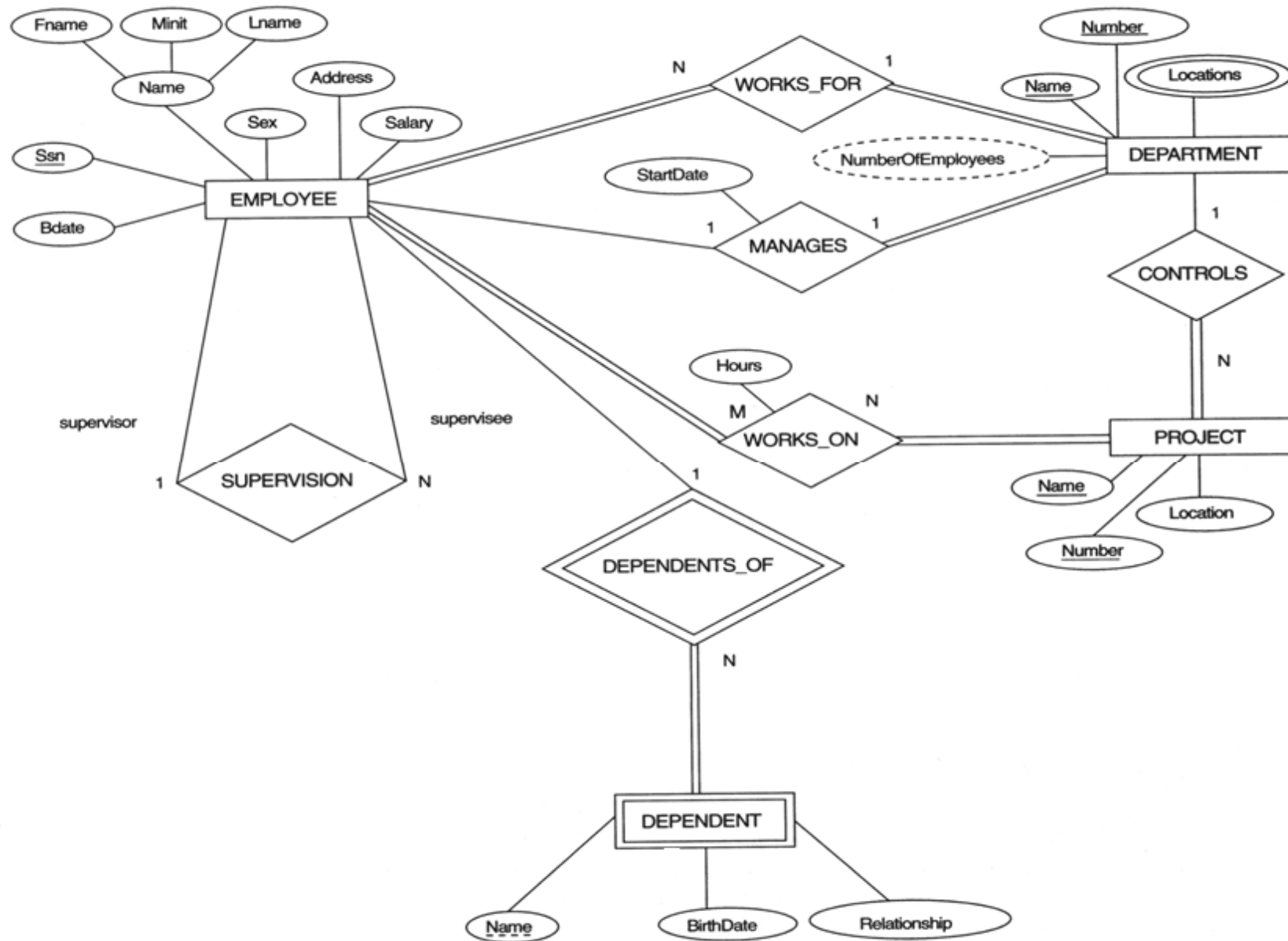
Step 9: Options for Mapping Specialization or Generalization.

Step 10: Mapping of Union Types  
(Categories).

## Step 1: Mapping of Strong Entity sets

- For each regular (strong) entity set 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.

## The ER diagram for the COMPANY database.



## Step1 : Example

EMPLOYEE (ssn, fname, mint, lname,  
bdate,salary, gender, address)

DEPARTMENT (dnumber, name)

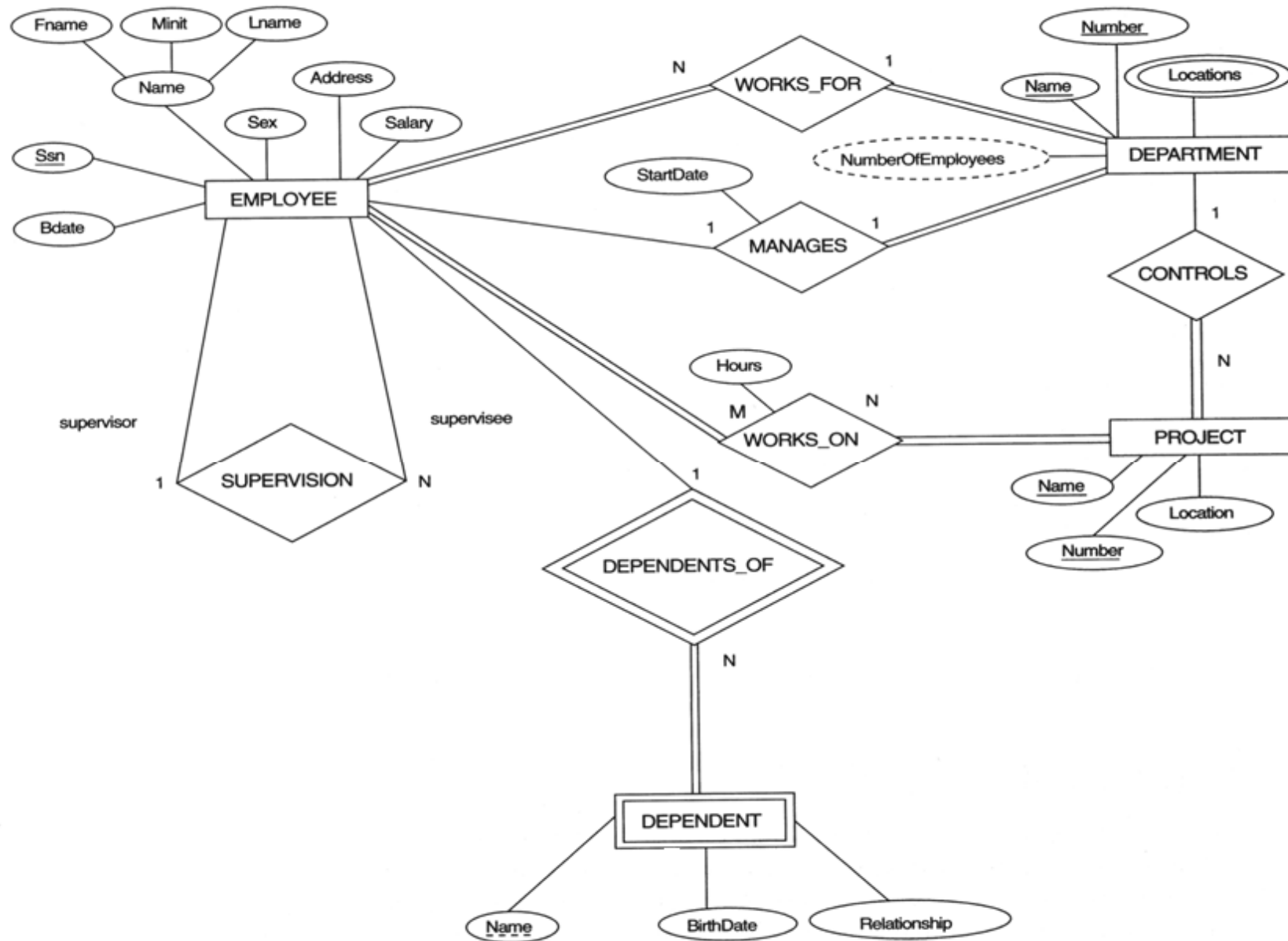
PROJECT(pnumber, pname, plocation)



## Step 2: Mapping of Weak Entity sets

- For each weak entity type  $W$  in the ER schema with owner entity type  $E$ , create a relation  $R$  and include all **simple attributes** (or 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).
- 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.

## The ER diagram for the COMPANY database.



## Step2 : Example

DEPENDENT (ESSN, depname, bdate,  
relationship)

## Step 3: Mapping unary relationship

Example: Employee is unary relationship

From step 1:

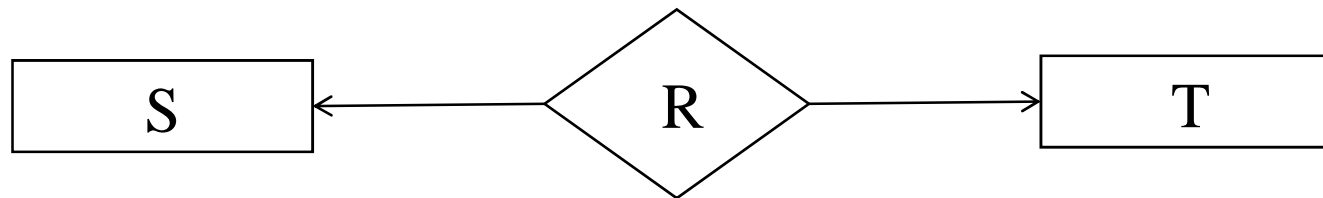
EMPLOYEE (ssn, fname, mint, lname,  
bdate,salary, gender, address)

From step 3:

EMPLOYEE (ssn, fname, mint, lname,  
bdate,salary, gender, address, superssn)

where superrssn refers to ssn

## Step 4: Mapping Binary 1:1 Relationship



Three approaches:

### (1) Foreign Key approach **Example**

From step 3:

EMPLOYEE (ssn, fname, mint, lname, bdate, salary, gender, address, superssn)

DEPARTMENT (dnumber, name, mgrstartdate)

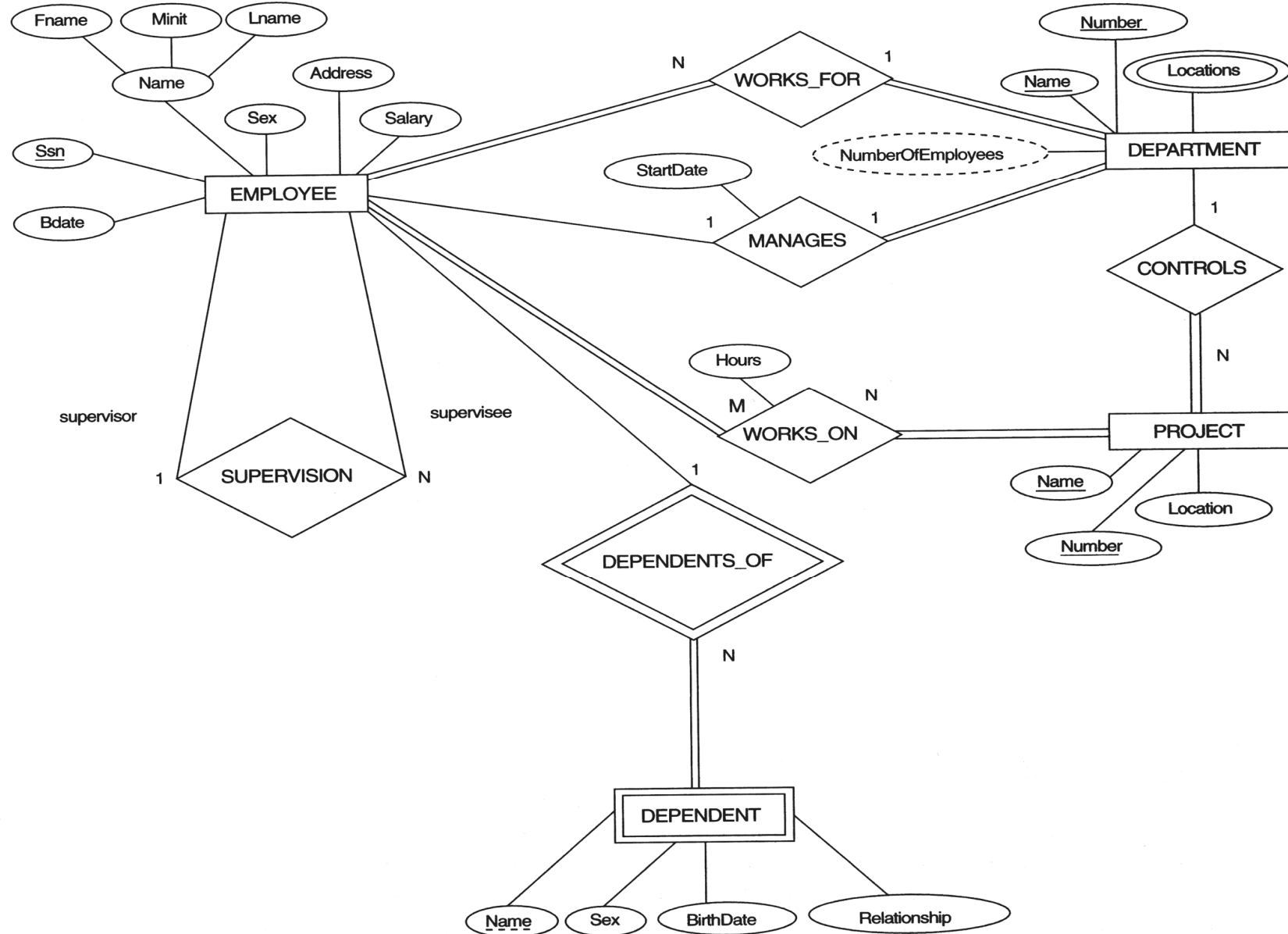


From step 4:

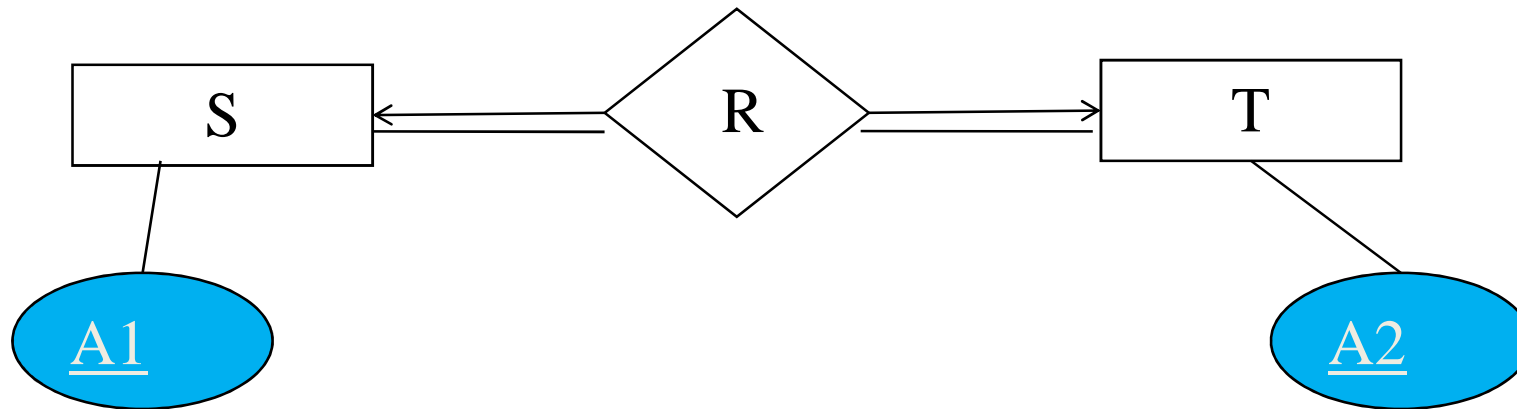
EMPLOYEE (ssn, fname, mint, lname, bdate, salary, gender, address, superssn)

DEPARTMENT (dnumber, name, mgrssn, mgrstartdate)

## The ER diagram for the COMPANY database.



## Step 4: Mapping Binary 1:1 Relationship



Second approach: Merged relation option

S( A1, A2, other attributes of S) and

T( A2, other attributes of T)

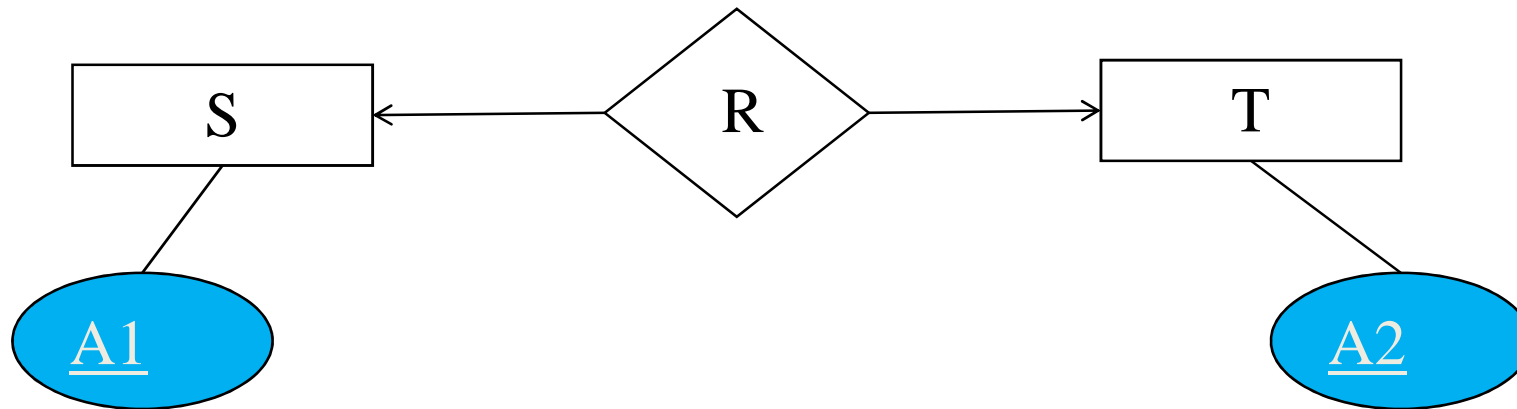
OR

S( A1, other attributes of S) and

T( A2, A1 other attributes of T)

No need to create relation R

## Step 4: Mapping Binary 1:1 Relationship



Third approach

Cross-reference or relationship relation option

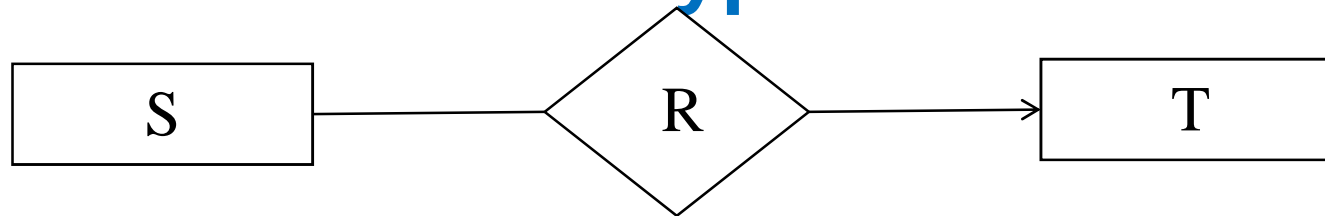
S (A1, other attributes of S) and

T (A2, other attributes of T) and

R (A1, A2 other attributes of R)



## Step 5: Mapping of Binary 1:N Relationship Types.



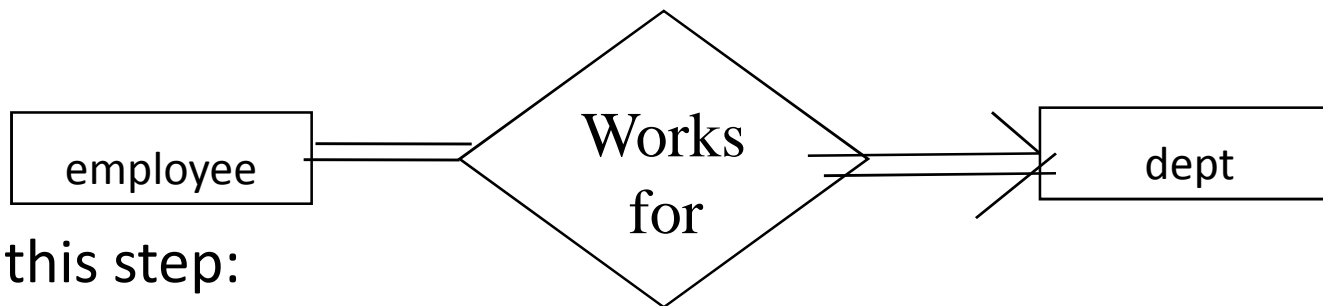
- For each regular binary 1:N relationship type  $R$ , identify the relation  $S$  that represent the participating entity type at the N-side of the relationship type.
- Include as foreign key in  $S$  the primary key of the relation  $T$  that represents the other entity type participating in  $R$ .
- Include any simple attributes of the 1:N relation type as attributes of  $S$ .

## Step 5: Mapping of Binary 1:N Relationship Types.

**Example:** From previous step:

EMPLOYEE (ssn, fname, mint, lname, bdate, salary, gender, address)

DEPARTMENT (dnumber, name, mgrssn, mgrstartdate)



From this step:

EMPLOYEE (ssn, fname, mint, lname, bdate, salary, gender, address, **dno**)

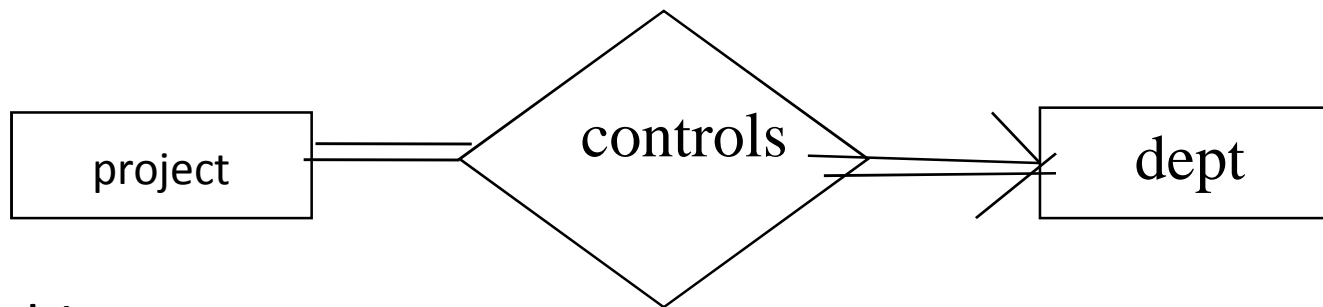
DEPARTMENT (dnumber, name, mgrssn, mgrstartdate)

## Step 5: Mapping of Binary 1:N Relationship Types.

**Example:** From previous step:

DEPARTMENT (dnumber, name, mgrssn, mgrstartdate)

PROJECT(pnumber, pname, plocation)

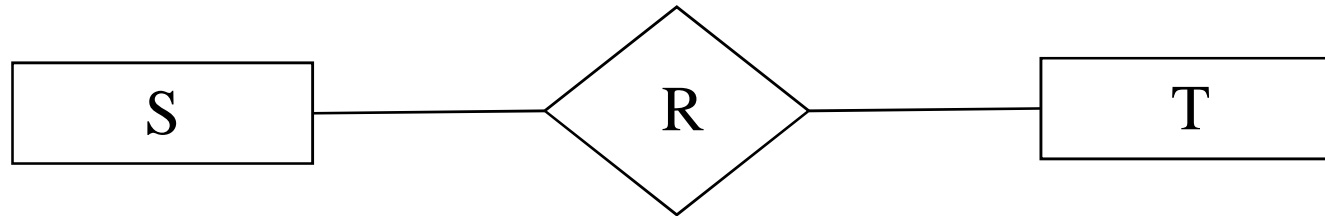


From this step:

DEPARTMENT (dnumber, name, mgrssn, mgrstartdate)

PROJECT(pnumber, pname, plocation, **dno**)

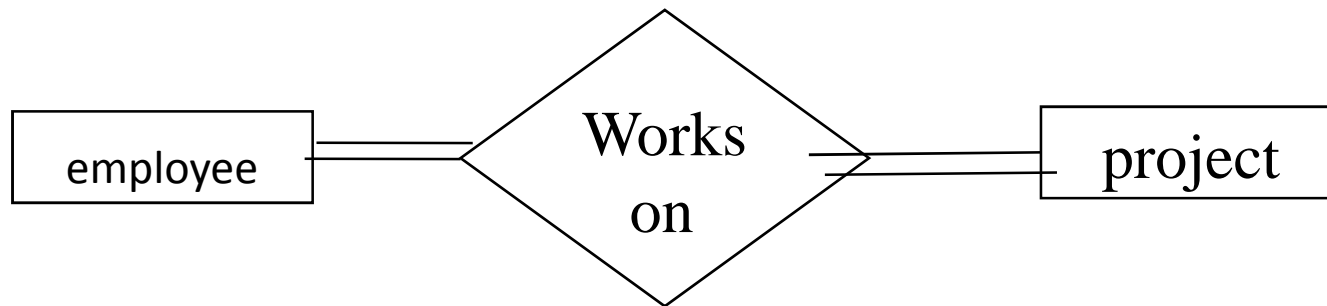
## Step 6: Mapping of Binary M:N Relationship Types.



- 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.

## Step 6: Mapping of Binary M:N Relationship Types.

Example: Works on (essn, pno, hours)



## Step 7: Mapping 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.

## Step 7: Mapping Multivalued attributes.

**Example : DEPT\_LOCATIONS ( Dnumber, Dlocation)**

where Dnumber is the foreign key referred to  
dnumber (primary key) of relation DEPARTMENT  
DEPARTMENT (dnumber, name, mgrssn, mgrstartdate)

# EMPLOYEE

FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
-------	-------	-------	------------	-------	---------	-----	--------	----------	-----

## DEPARTMENT

DNAME	<u>DNUMBER</u>	MGRSSN	MGRSTARTDATE
-------	----------------	--------	--------------

## DEPT\_LOCATIONS

<u>DNUMBER</u>	<u>DLOCATION</u>
----------------	------------------

## PROJECT

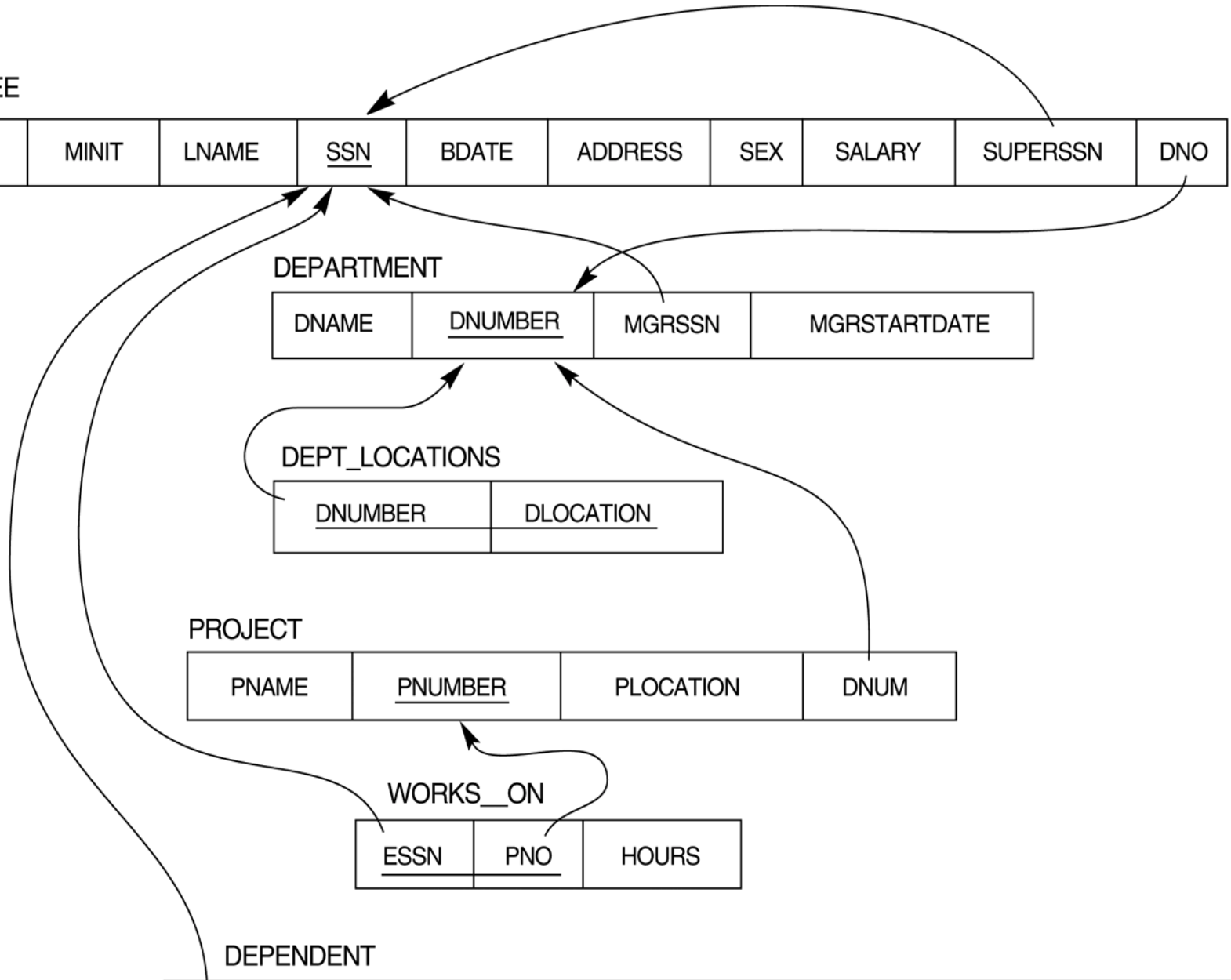
PNAME	<u>PNUMBER</u>	PLOCATION	DNUM
-------	----------------	-----------	------

## WORKS\_ON

<u>ESSN</u>	<u>PNO</u>	HOURS
-------------	------------	-------

## DEPENDENT

<u>ESSN</u>	<u>DEPENDENT_NAME</u>	SEX	BDATE	RELATIONSHIP
-------------	-----------------------	-----	-------	--------------

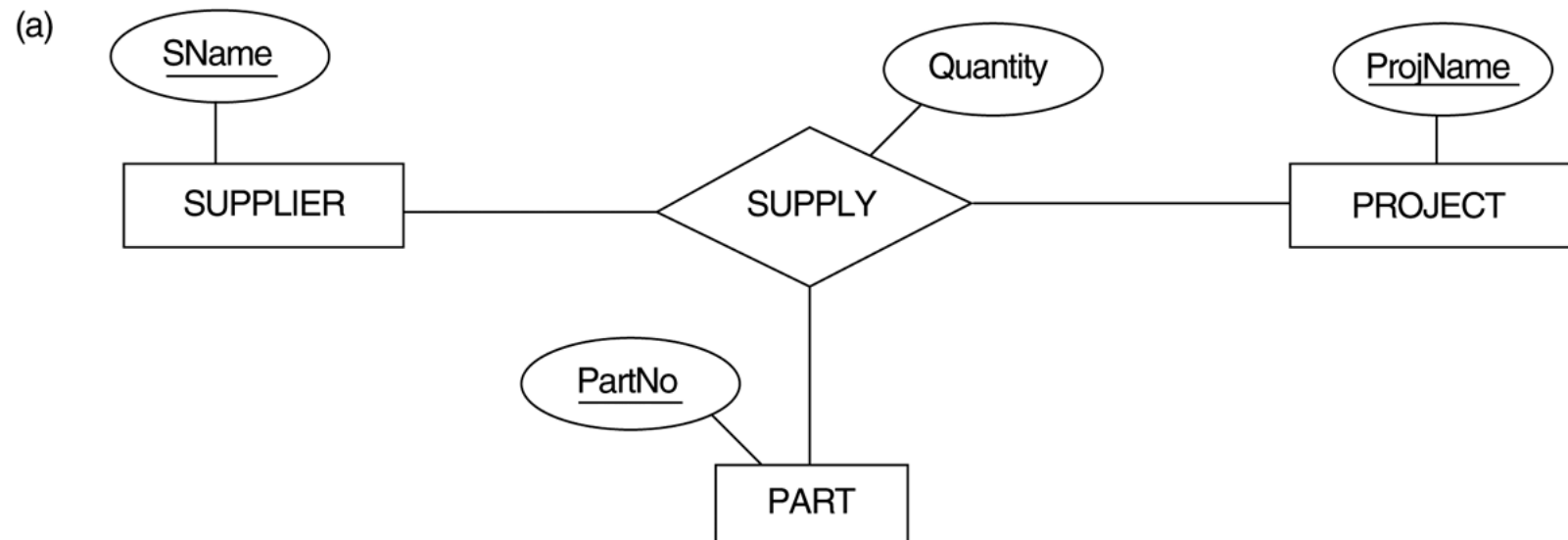




## Step 8: Mapping of N-ary Relationship Types.

- 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.

## Ternary relationship SUPPLY



## Mapping the $n$ -ary relationship type SUPPLY

SUPPLIER

<u>SNAME</u>	...
--------------	-----

PROJECT

<u>PROJNAME</u>	...
-----------------	-----

PART

<u>PARTNO</u>	...
---------------	-----

SUPPLY

<u>SNAME</u>	<u>PROJNAME</u>	<u>PARTNO</u>	QUANTITY
--------------	-----------------	---------------	----------

# **Mapping EER Model to Relational Model**

## Step9: Options for Mapping Specialization or Generalization.

Convert each specialization with  $m$  subclasses  $\{S_1, S_2, \dots, S_m\}$  and generalized superclass  $C$ , where the attributes of  $C$  are  $\{\underline{k}, a_1, \dots, a_n\}$  and  $k$  is the (primary) key, into relational schemas using one of the four following options:

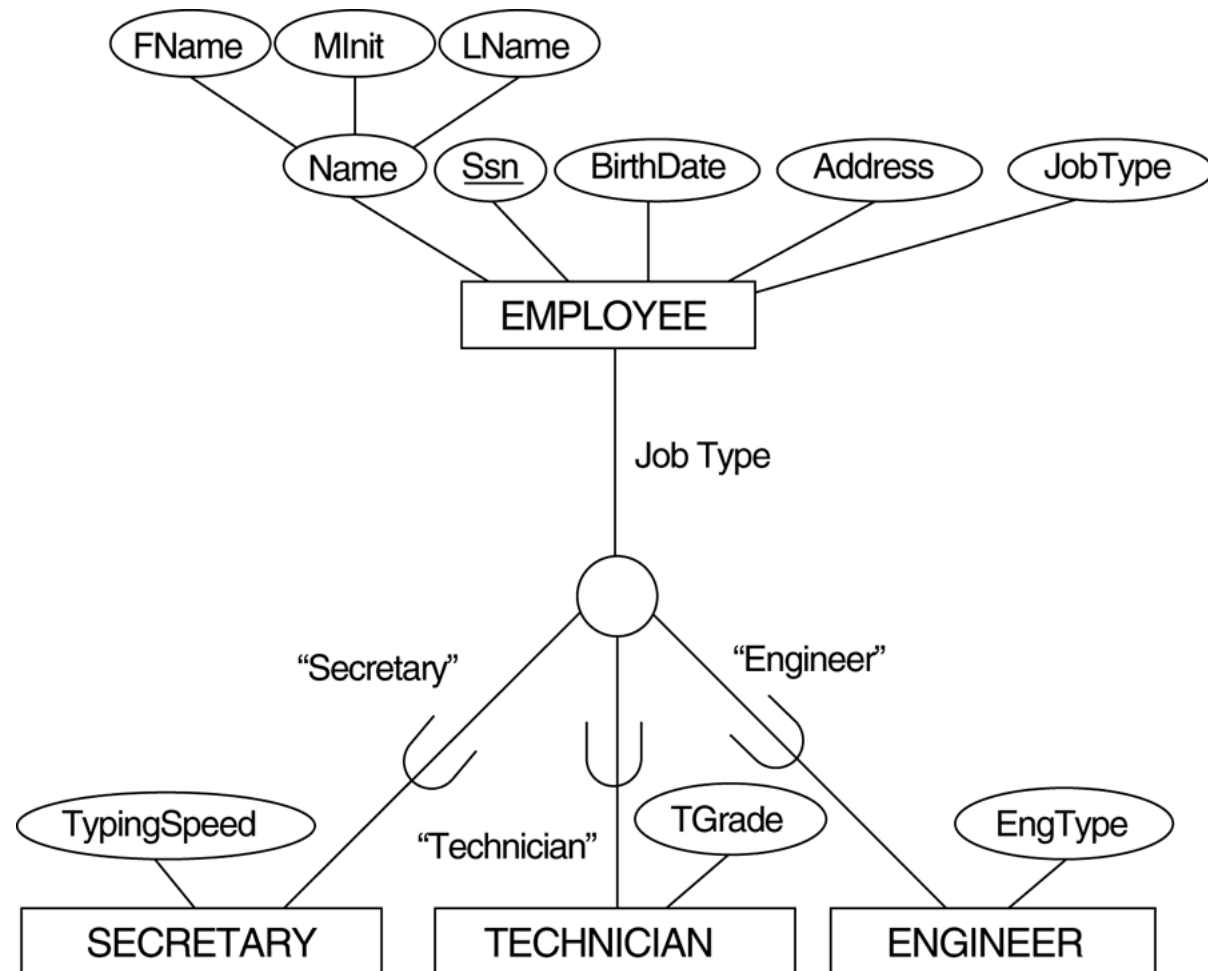
**Option 9A: Multiple relations-Superclass and subclasses.**

**Option 9B: Multiple relations-Subclass relations only**

**Option 9C: Single relation with one type attribute.**

**Option 9D: Single relation with multiple type attributes.**

EER diagram notation for an attribute-defined specialization on JobType.



Mapping the EER schema in Figure using option 8A.

**Multiple relations-Superclass and subclasses.**

(a) EMPLOYEE

<u>SSN</u>	FName	MInit	LName	BirthDate	Address	JobType
------------	-------	-------	-------	-----------	---------	---------

SECRETARY

<u>SSN</u>	TypingSpeed
------------	-------------

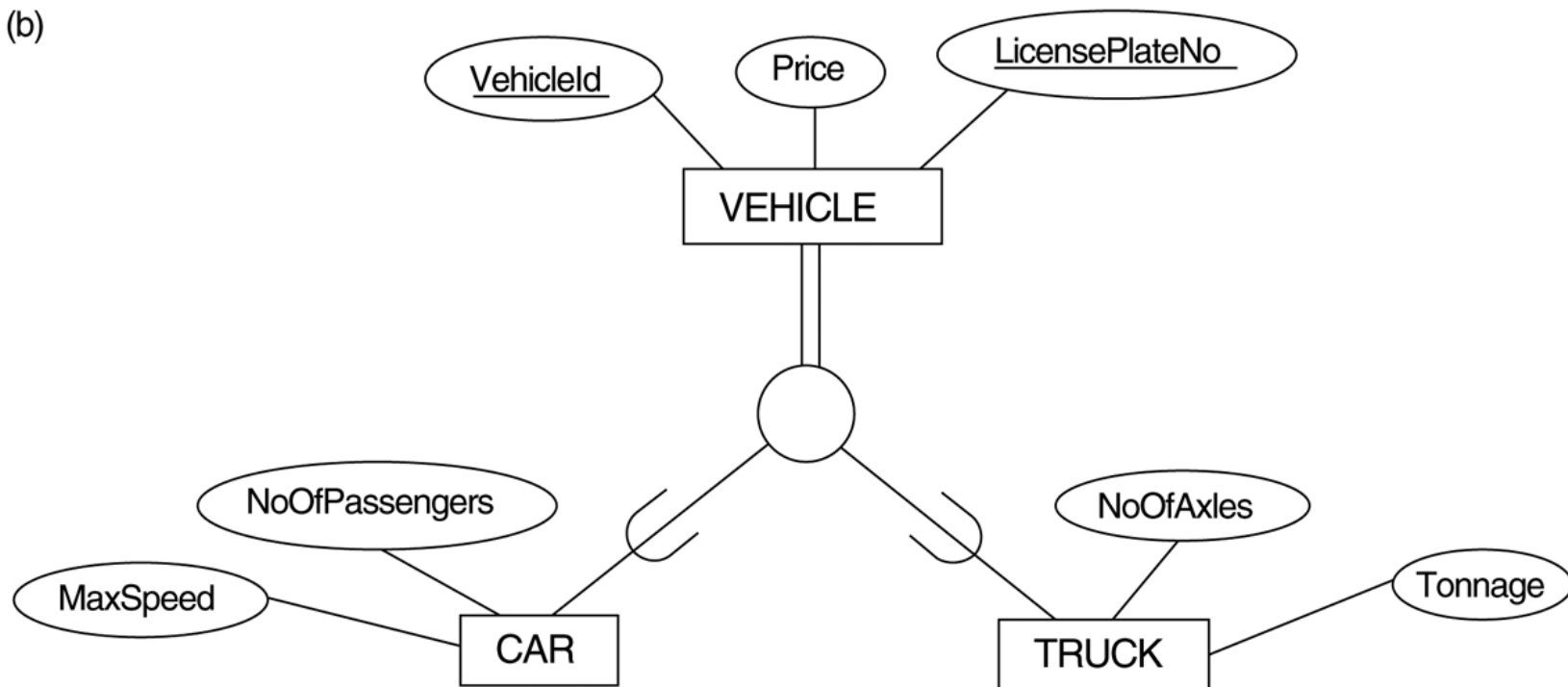
TECHNICIAN

<u>SSN</u>	TGrade
------------	--------

ENGINEER

<u>SSN</u>	EngType
------------	---------

(b)





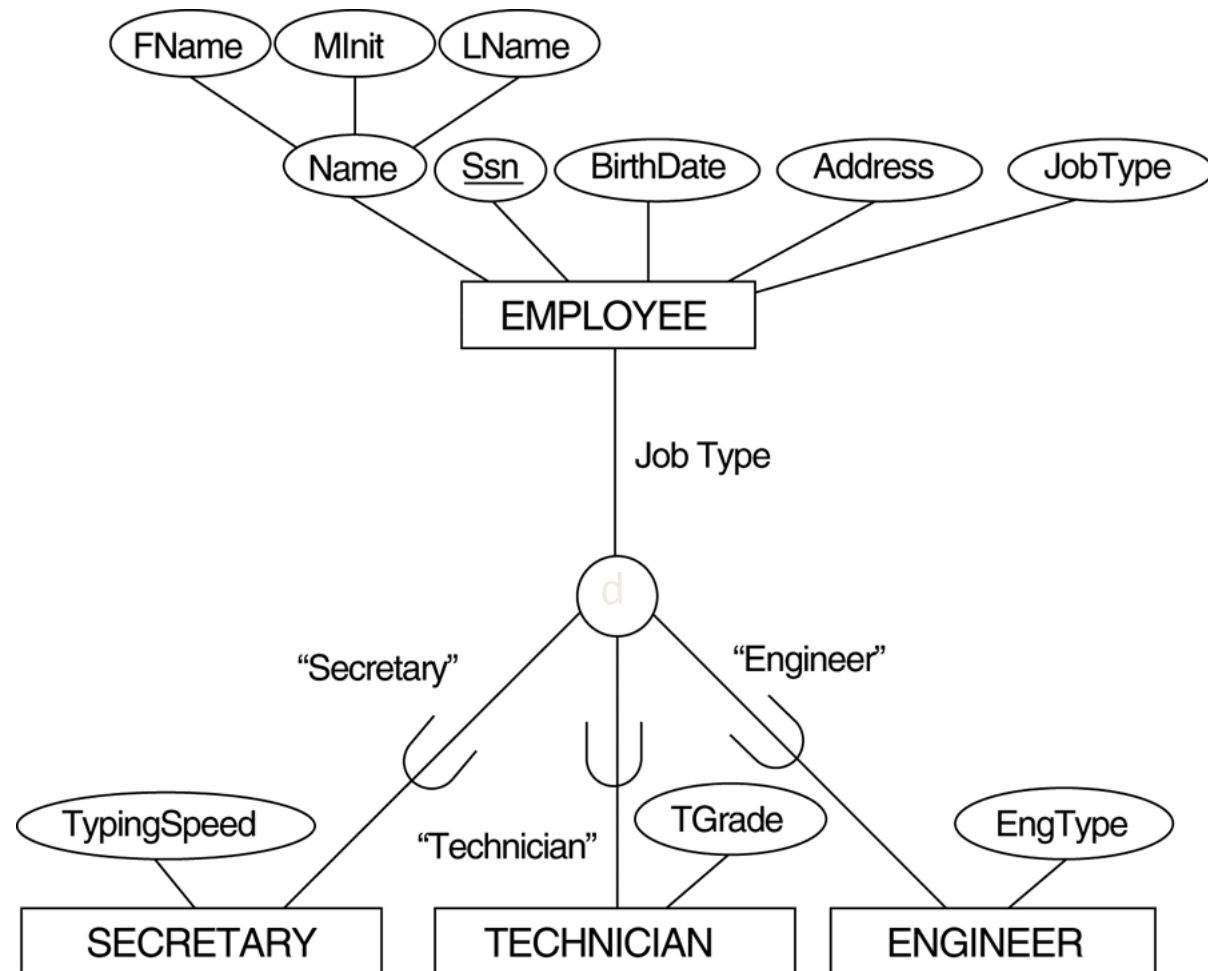
(b) CAR

<u>VehicleId</u>	LicensePlateNo	Price	MaxSpeed	NoOfPassengers
------------------	----------------	-------	----------	----------------

TRUCK

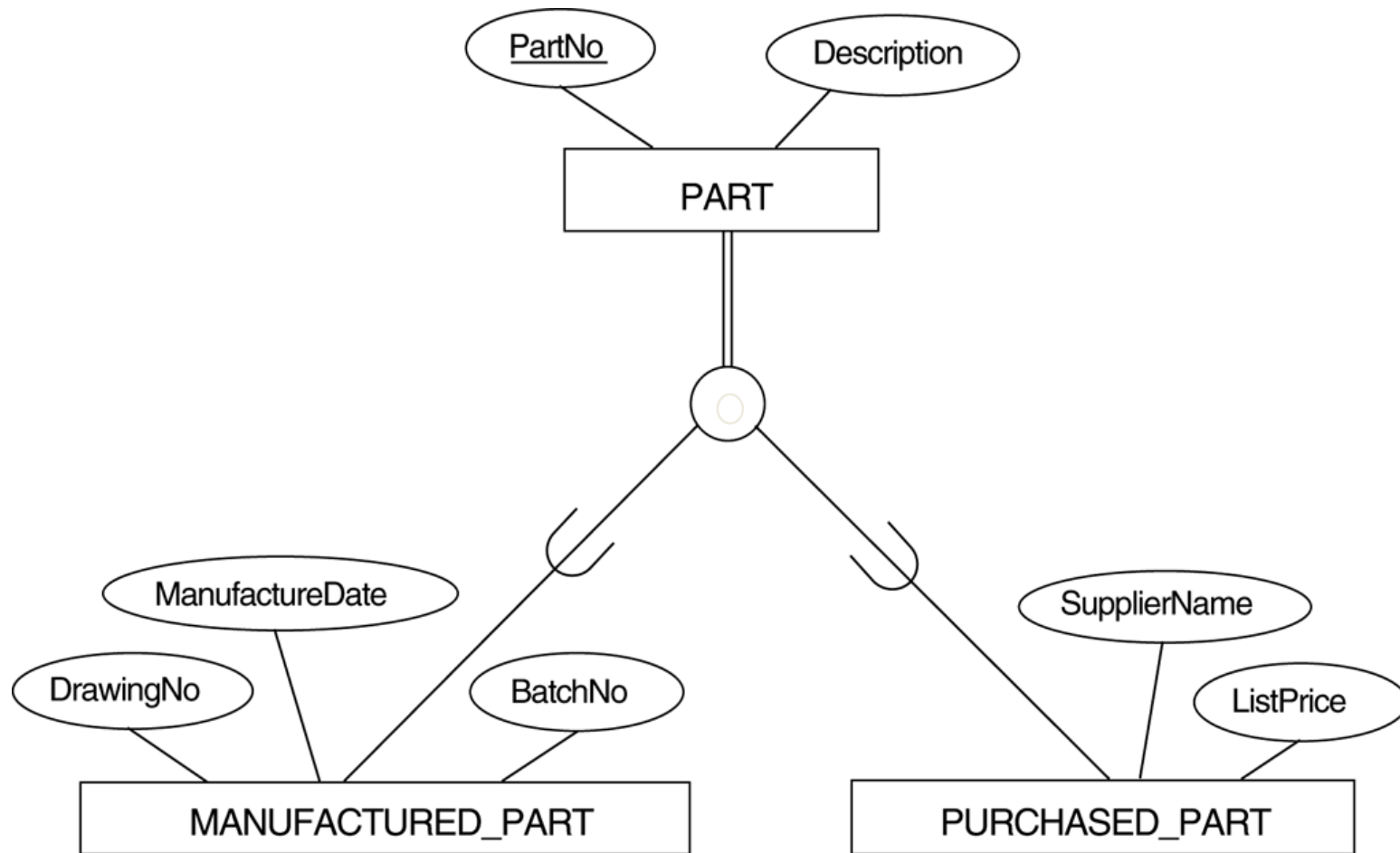
<u>VehicleId</u>	LicensePlateNo	Price	NoOfAxles	
------------------	----------------	-------	-----------	--

EER diagram notation for an attribute-defined specialization on JobType.



(c) EMPLOYEE

<u>SSN</u>	FName	MInit	LName	BirthDate	Address	JobType	TypingSpeed	TGrade	
------------	-------	-------	-------	-----------	---------	---------	-------------	--------	--



Eg. Mapping Figure using option 8D with Boolean type fields Mflag and Pflag.

(d) PART

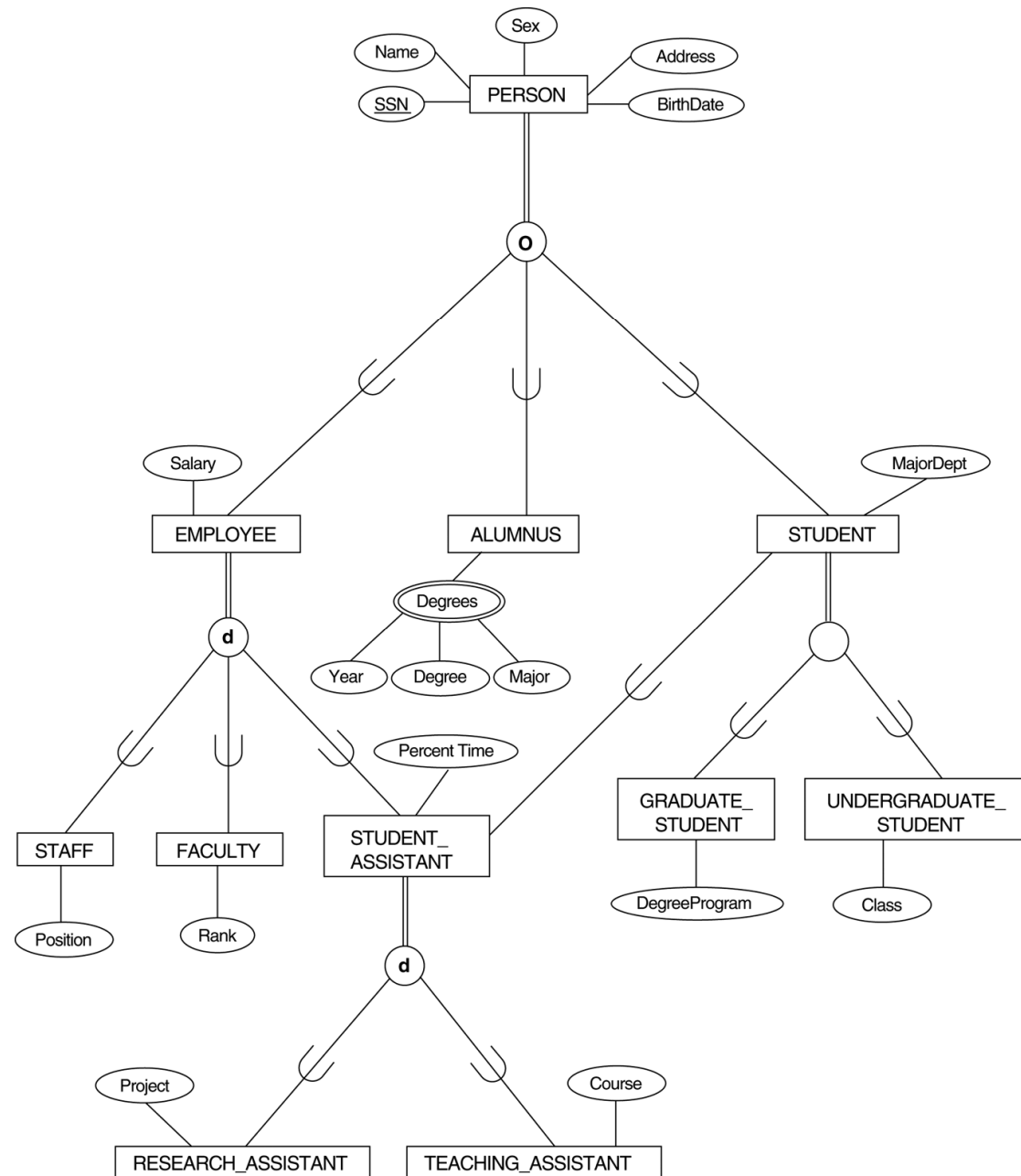
<u>PartNo</u>	Description	MFlag	DrawingNo	ManufactureDate	BatchNo	PFlag	SupplierName	ListPrice
---------------	-------------	-------	-----------	-----------------	---------	-------	--------------	-----------

# Mapping of Shared Subclasses (Multiple Inheritance)

A shared subclass, such as STUDENT\_ASSISTANT, is a subclass of several classes, indicating multiple inheritance. These classes must all have the same key attribute; otherwise, the shared subclass would be modeled as a category.

We can apply any of the options discussed in Step 9 to a shared subclass, subject to the restriction discussed in Step 9 of the mapping algorithm. Below both 9C and 9D are used for the shared class STUDENT\_ASSISTANT.

A specialization lattice with multiple inheritance for a UNIVERSITY database.



## Mapping the EER specialization lattice in Figure using multiple options.

PERSON

<u>SSN</u>	Name	BirthDate	Sex	Address
------------	------	-----------	-----	---------

EMPLOYEE

<u>SSN</u>	Salary	EmployeeType	Position	Rank	PercentTime	RAFlag	TAFlag	Project	
------------	--------	--------------	----------	------	-------------	--------	--------	---------	--

ALUMNUS

<u>SSN</u>
------------

ALUMNUS\_DEGREES

<u>SSN</u>	Year	Degree	
------------	------	--------	--

STUDENT

<u>SSN</u>	MajorDept	GradFlag	UndergradFlag	DegreeProgram	Class	StudAssistFlag
------------	-----------	----------	---------------	---------------	-------	----------------



# Mapping ER-to-Relational Model

Step 1: Mapping of Strong Entity sets

Step 2: Mapping of Weak Entity sets

Step 3: Mapping Unary Relationship set.

Step 4: Mapping of Binary 1:1 Relationship set

Step 5: Mapping of Binary 1:N Relationship set.

Step 6: Mapping of Binary M:N Relationship set.

Step 7: Mapping of Multivalued attributes.

Step 8: Mapping of N-ary Relationship set.

# Mapping EER-to-Relational Model

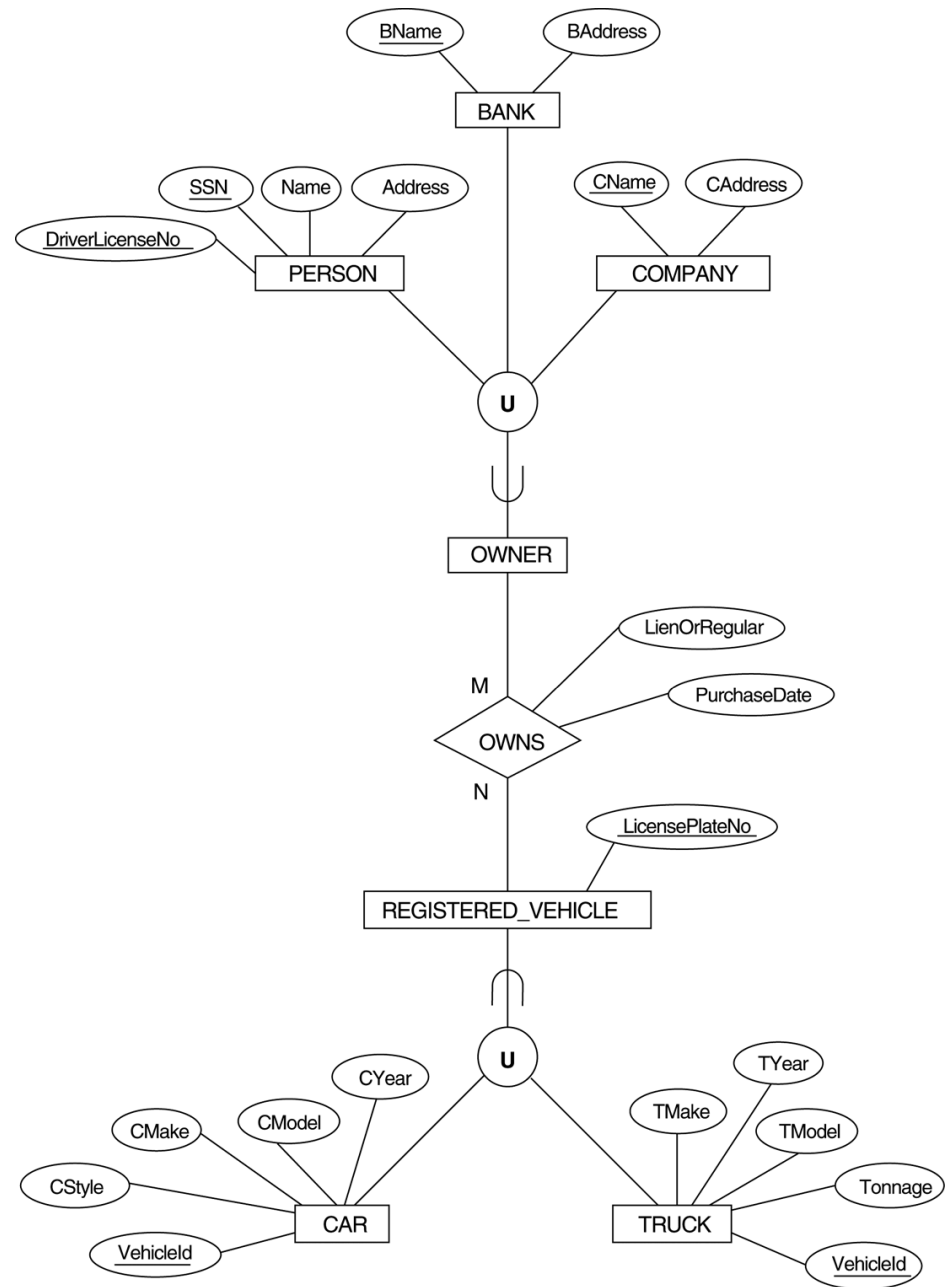
Step 9: Options for Mapping Specialization or Generalization.

Step 10: Mapping of Union Types  
(Categories).

## Step 10: Mapping of Union Types (Categories).

- For mapping a category whose defining superclass 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.
- In the example below we can create a relation OWNER to correspond to the OWNER category and include any attributes of the category in this relation. The primary key of the OWNER relation is the surrogate key, which we called OwnerId.

Two categories (union types):  
OWNER and  
REGISTERED\_VEHICLE.



Mapping the EER categories (union types) to relations.

#### PERSON

<u>SSN</u>	DriverLicenseNo	Name	Address	OwnerId
------------	-----------------	------	---------	---------

#### BANK

<u>BName</u>	BAddress	OwnerId
--------------	----------	---------

#### COMPANY

<u>CName</u>	CAddress	OwnerId
--------------	----------	---------

#### OWNER

<u>OwnerId</u>
----------------

#### REGISTERED\_VEHICLE

<u>VehicleId</u>	LicensePlateNumber
------------------	--------------------

#### CAR

<u>VehicleId</u>	CStyle	CMake	CModel	
------------------	--------	-------	--------	--

#### TRUCK

<u>VehicleId</u>	TMake	TModel	Tonnage	TYear
------------------	-------	--------	---------	-------

#### OWNS

<u>OwnerId</u>	<u>VehicleId</u>	PurchaseDate	LienOrRegular
----------------	------------------	--------------	---------------