

# Lesson 5.1: E/ER-to-Relational Model Mapping

---

CSC430/530 – DATABASE MANAGEMENT SYSTEMS

DR. ANDREY TIMOFEYEV

A solid blue horizontal bar at the bottom of the slide.

# MAPPING ALGORITHM STEPS

---

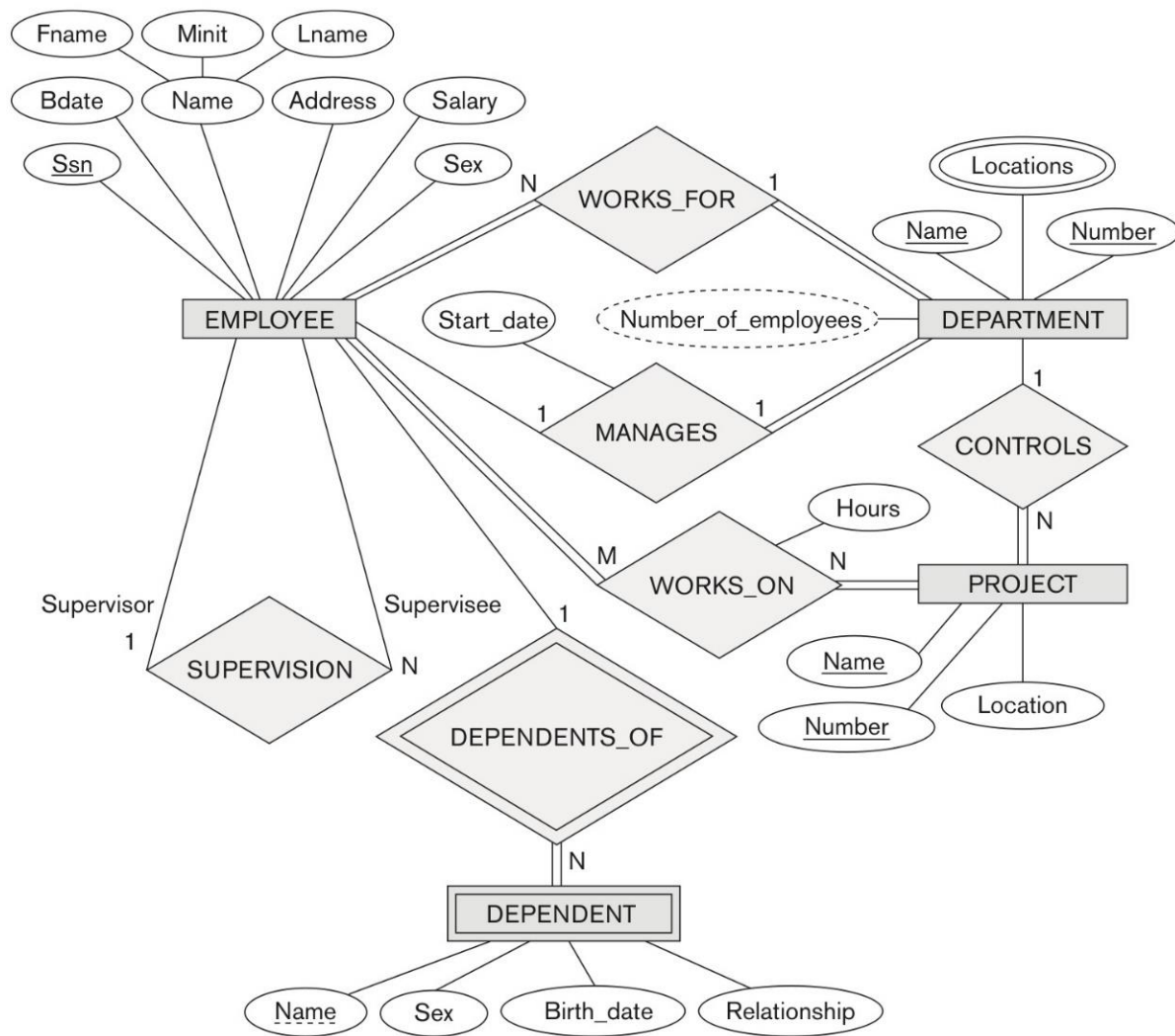
- Step 1: Mapping of **regular entity types**.
- Step 2: Mapping of **weak entity types**.
- Step 3: Mapping of **binary 1:1 relationship types**.
- Step 4: Mapping of **binary 1:N relationship types**.
- Step 5: Mapping of **binary M:N relationship types**.
- Step 6: Mapping of **multivalued attributes**.
- Step 7: Mapping of **n-ary relationship types**.
- Step 8: Mapping of **specializations and generalizations**.
- Step 9: Mapping of **union types (categories)**.

# STEP 1: MAPPING OF REGULAR ENTITY TYPES

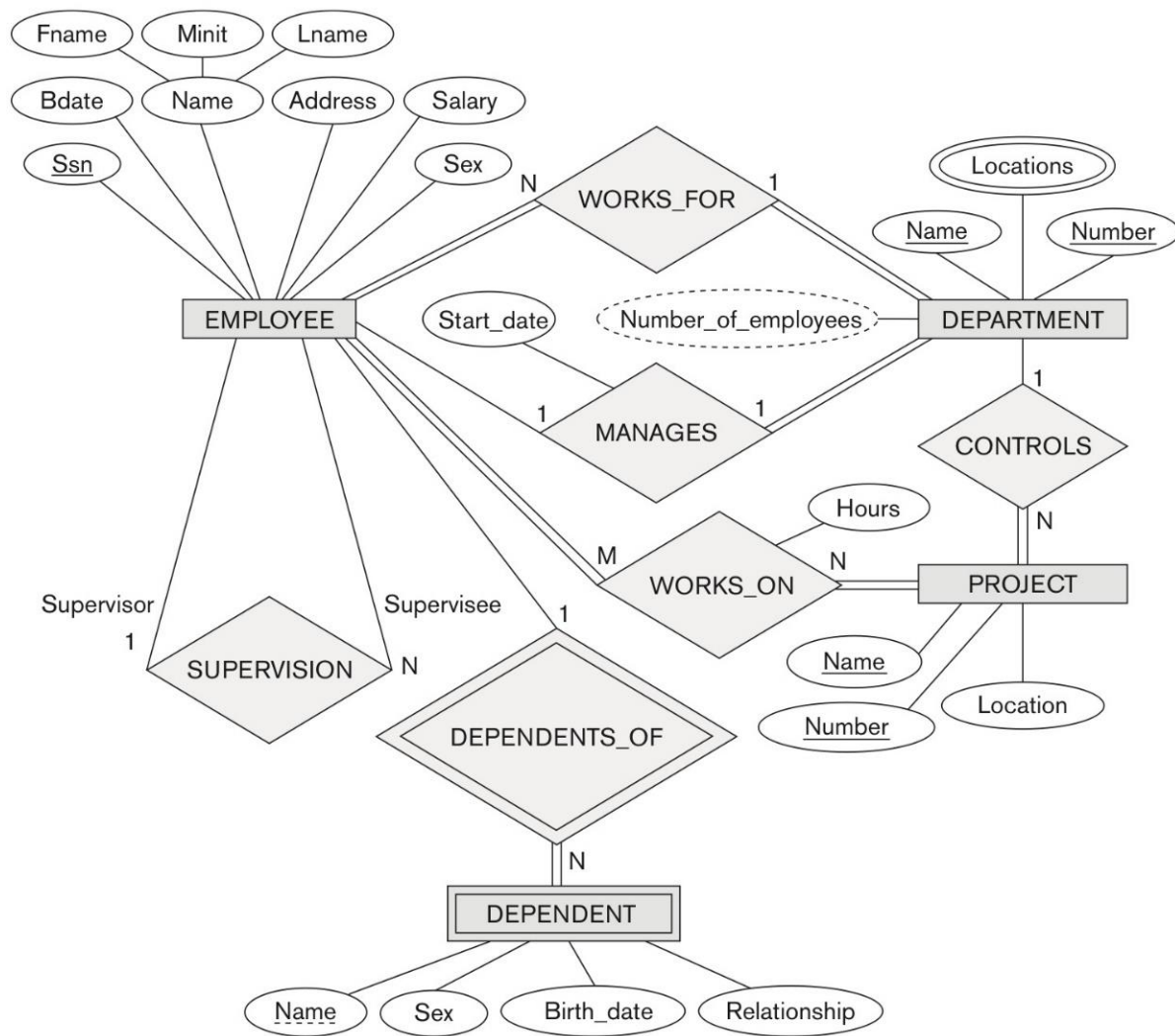
---

- 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  $R$ .

# STEP 1: MAPPING OF REGULAR ENTITY TYPES



# STEP 1: MAPPING OF REGULAR ENTITY TYPES



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary
-------	-------	-------	------------	-------	---------	-----	--------

## DEPARTMENT

Dname	<u>Dnumber</u>
-------	----------------

## PROJECT

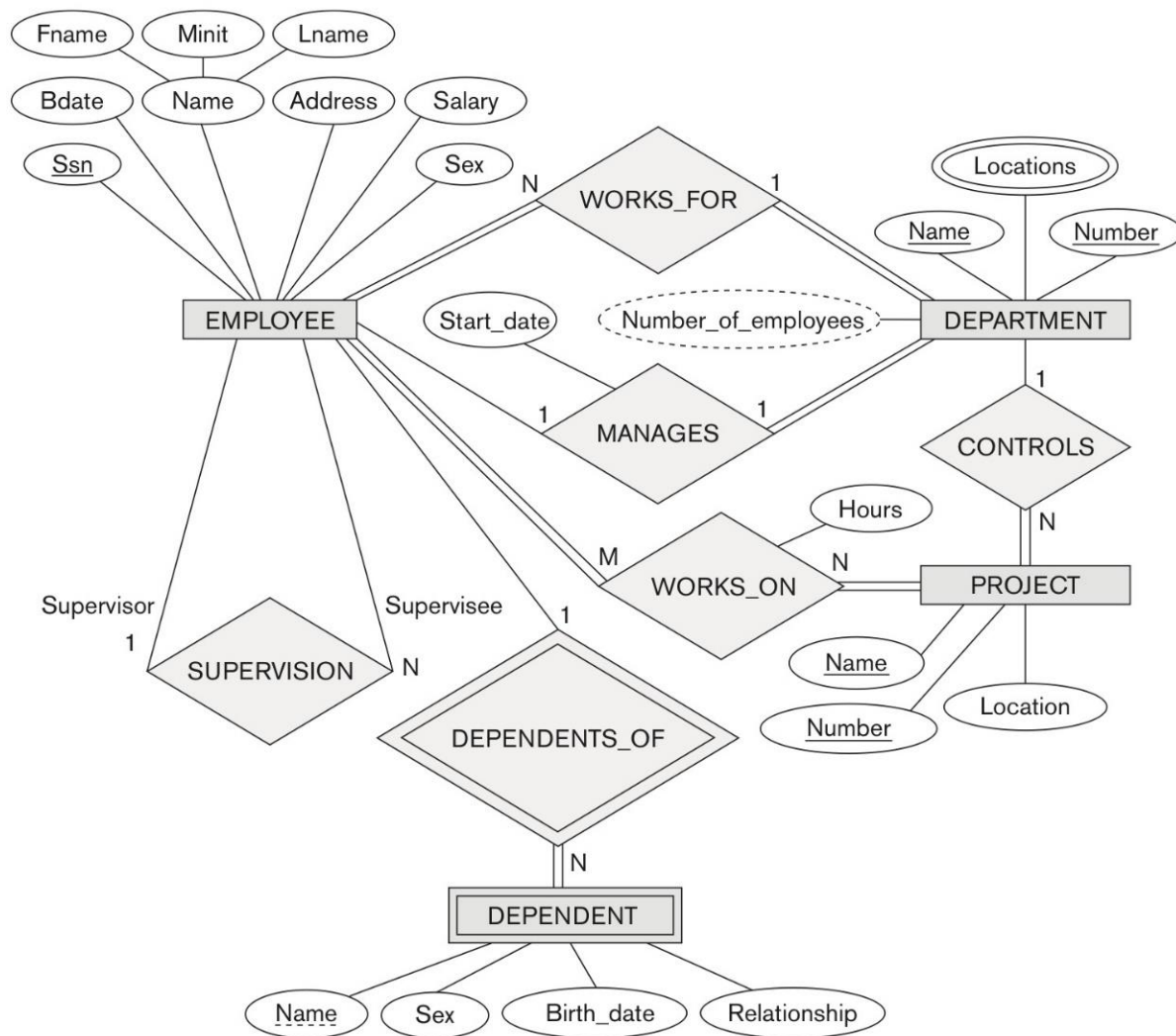
Pname	<u>Pnumber</u>	Plocation
-------	----------------	-----------

# 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**  $R$  and include **all simple attributes** of  $W$  as **attributes** of  $R$ .
- Also, include as **foreign key attributes** of  $R$  the **primary key attribute(s)** of the **relation** that correspond to the **owner entity type**.
- 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.

# STEP 2: MAPPING OF WEAK ENTITY TYPES



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary
-------	-------	-------	------------	-------	---------	-----	--------

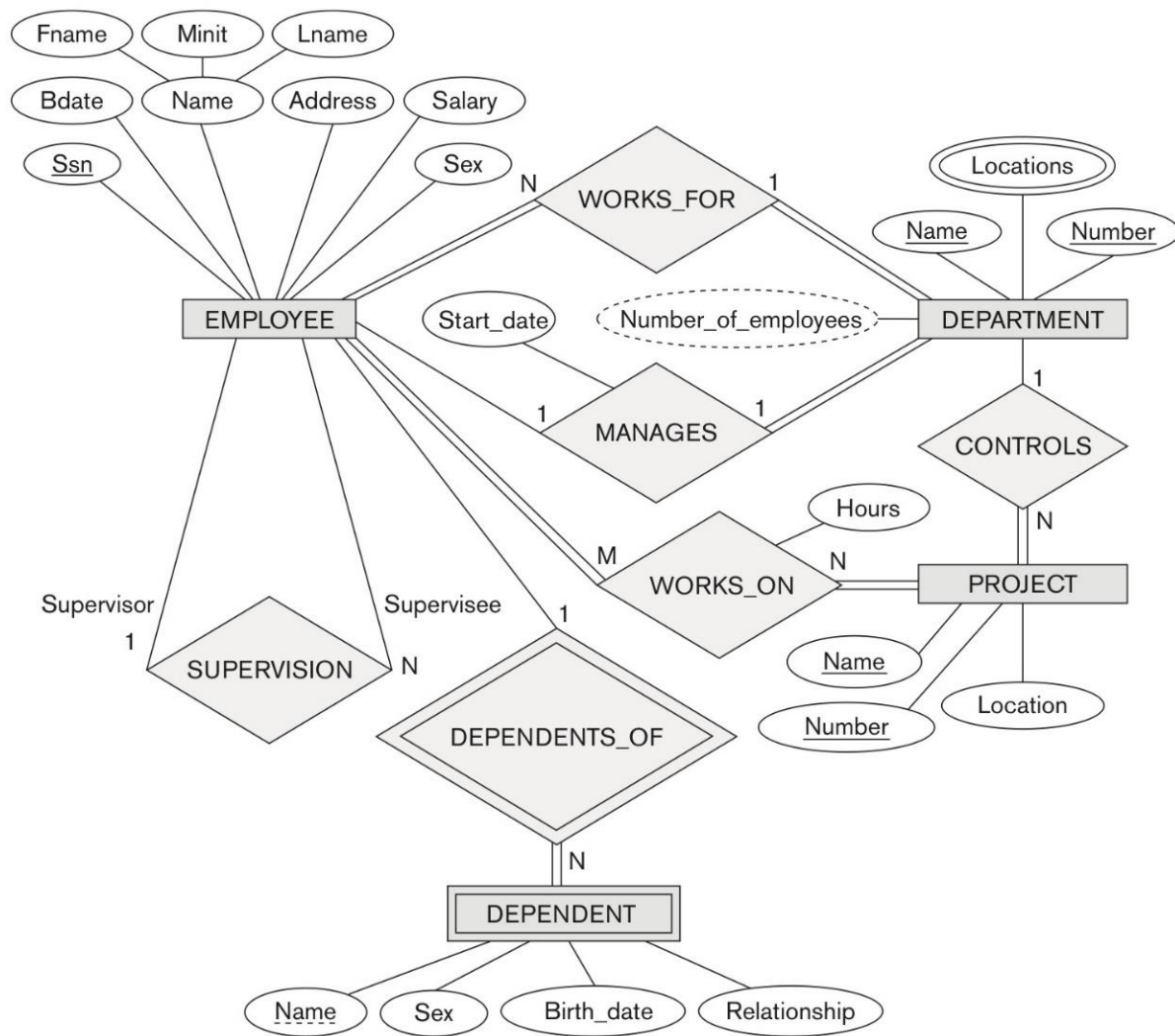
## DEPARTMENT

Dname	<u>Dnumber</u>
-------	----------------

## PROJECT

Pname	<u>Pnumber</u>	Plocation
-------	----------------	-----------

# STEP 2: MAPPING OF WEAK ENTITY TYPES



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary
-------	-------	-------	------------	-------	---------	-----	--------

## DEPARTMENT

Dname	<u>Dnumber</u>
-------	----------------

## PROJECT

Pname	<u>Pnumber</u>	Plocation
-------	----------------	-----------

## DEPENDENT

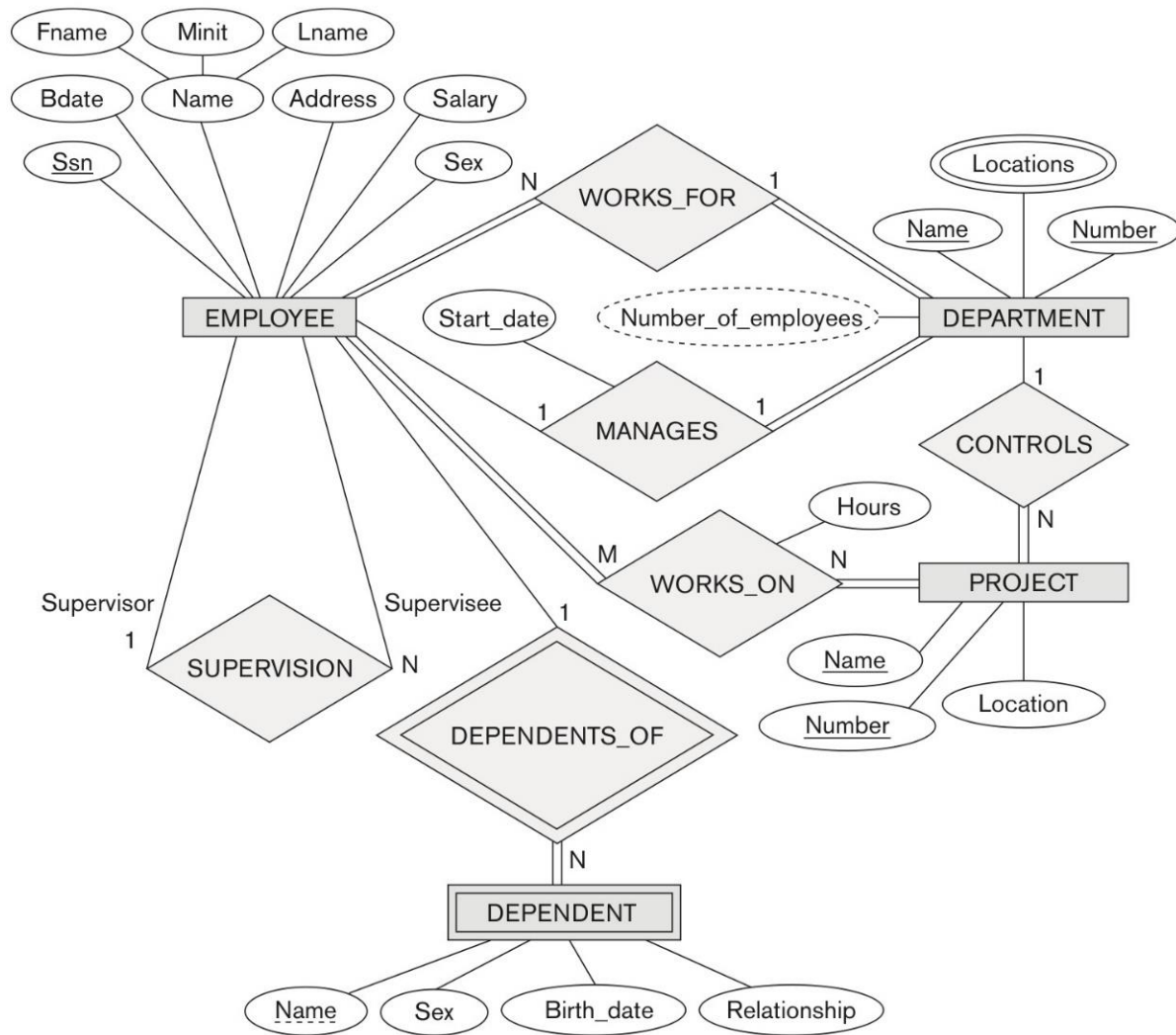
<u>Essn</u>	<u>Dependent name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------



# STEP 3: MAPPING OF BINARY 1:1 RELATIONSHIPS

- **Three possible options:**
- **Foreign Key** (*2 relations –  $S, T$* ).
  - Choose one of the **relations**  $S$  and include as 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  $S$ .
- **Merged relation** (*1 relation*).
  - Merging the **two entity types** and the **relationship** into a **single relation**.
    - Appropriate when both participations are **total**.
- **Cross-reference or relationship relation** (*3 relations –  $R, S, T$* ).
  - 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**.
    - The **relation**  $R$  will include the **primary key** attributes of  $S$  and  $T$  as **foreign keys** to  $S$  and  $T$ .
    - The **primary key** of  $R$  will be one of the two **foreign keys**, and the other **foreign key** will be a **unique key** of  $R$ .

# STEP 3: MAPPING OF BINARY 1:1 RELATIONSHIPS



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary
-------	-------	-------	------------	-------	---------	-----	--------

## DEPARTMENT

Dname	<u>Dnumber</u>
-------	----------------

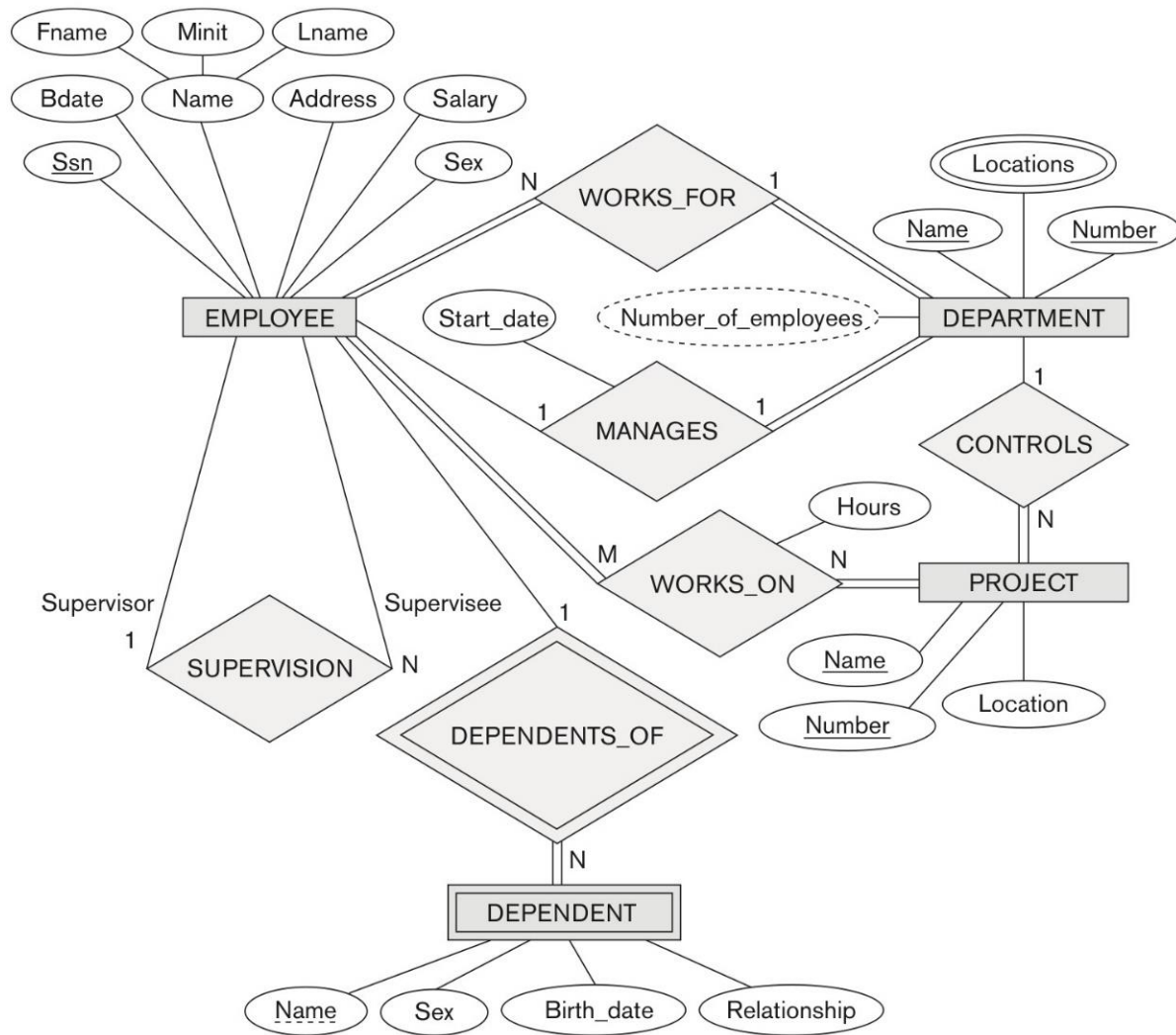
## PROJECT

Pname	<u>Pnumber</u>	Plocation
-------	----------------	-----------

## DEPENDENT

<u>Essn</u>	<u>Dependent name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

# STEP 3: MAPPING OF BINARY 1:1 RELATIONSHIPS



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary
-------	-------	-------	------------	-------	---------	-----	--------

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

## PROJECT

Pname	<u>Pnumber</u>	Plocation
-------	----------------	-----------

## DEPENDENT

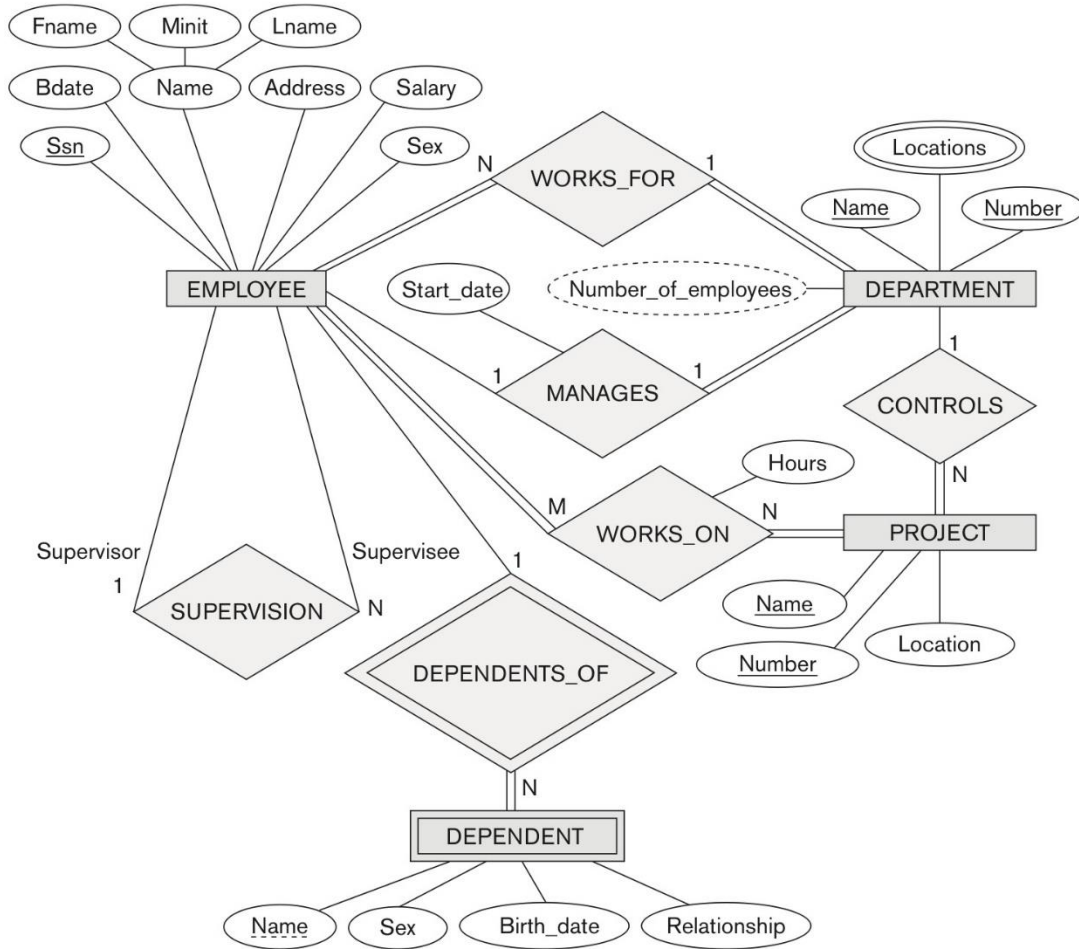
<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

# STEP 4: MAPPING OF BINARY 1:N RELATIONSHIPS

---

- For each **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 4: MAPPING OF BINARY 1:N RELATIONSHIPS



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary
-------	-------	-------	------------	-------	---------	-----	--------

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

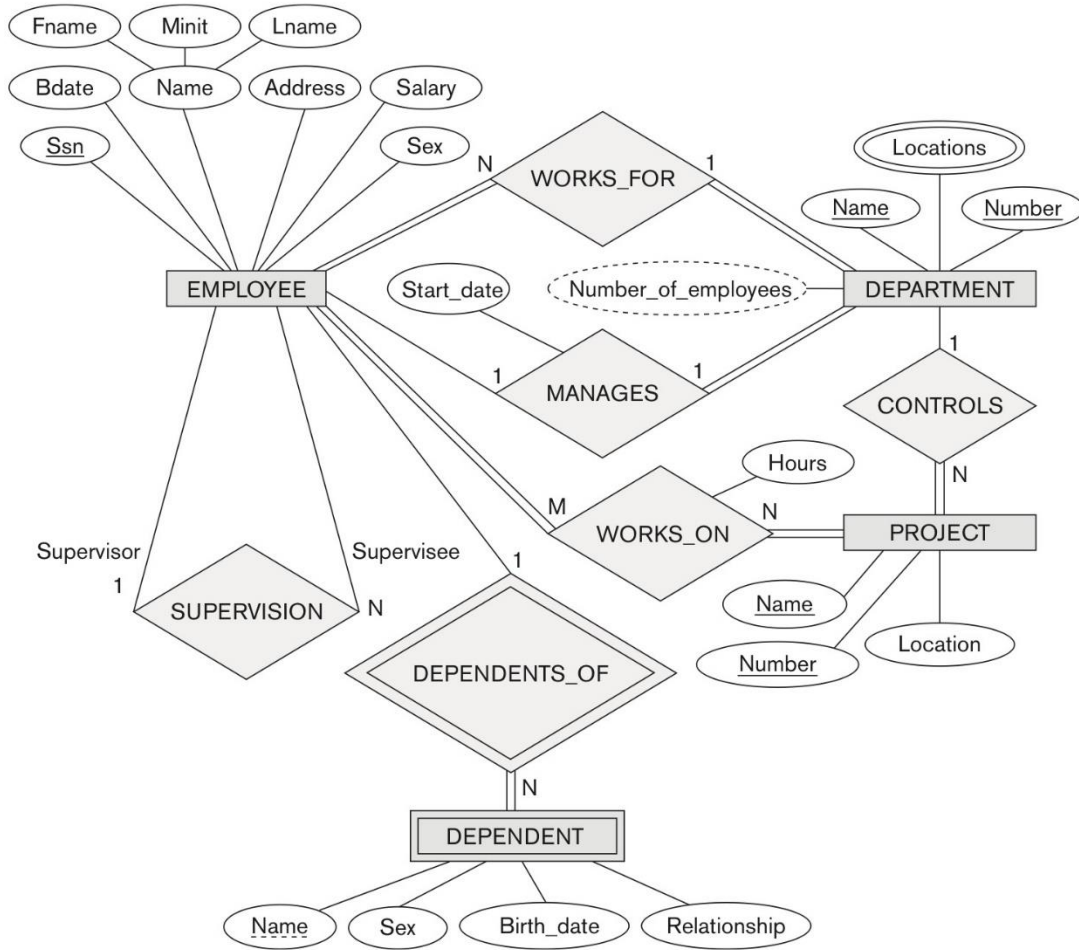
## PROJECT

Pname	<u>Pnumber</u>	Plocation
-------	----------------	-----------

## DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

# STEP 4: MAPPING OF BINARY 1:N RELATIONSHIPS



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

## PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

## DEPENDENT

<u>Essn</u>	<u>Dependent name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

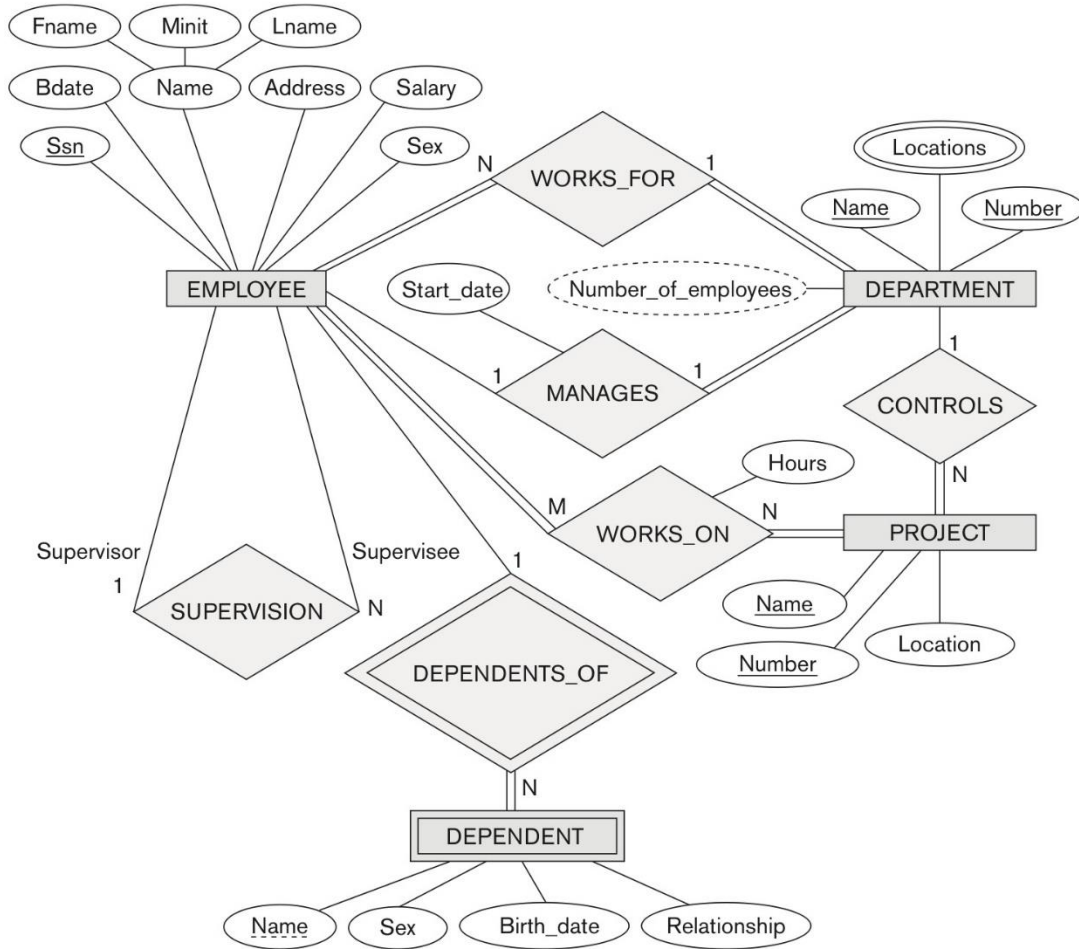
# STEP 5: MAPPING OF BINARY M:N RELATIONSHIPS

---

- For each **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 key attributes** 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  $S$ .



# STEP 5: MAPPING OF BINARY M:N RELATIONSHIPS



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

## PROJECT

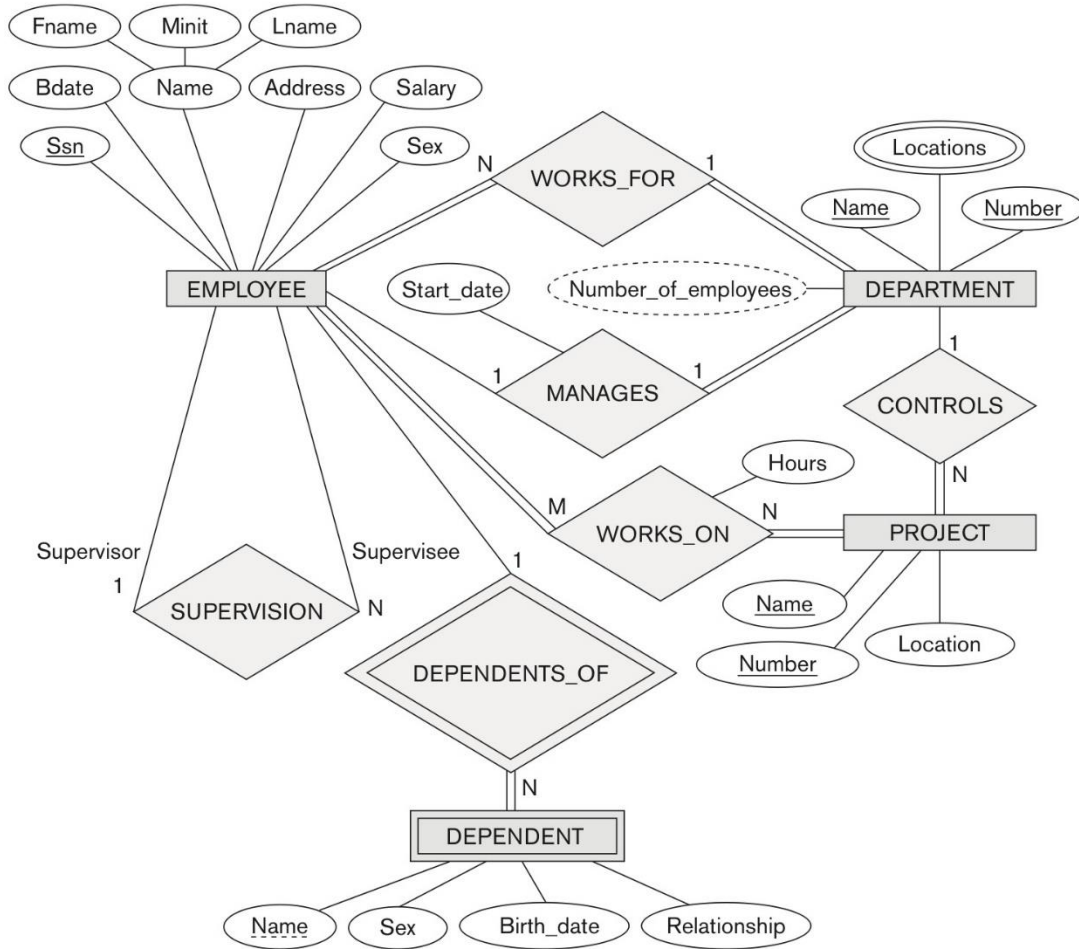
Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

## DEPENDENT

<u>Essn</u>	<u>Dependent name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------



# STEP 5: MAPPING OF BINARY M:N RELATIONSHIPS



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

## PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

## DEPENDENT

<u>Essn</u>	<u>Dependent name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

## WORKS\_ON

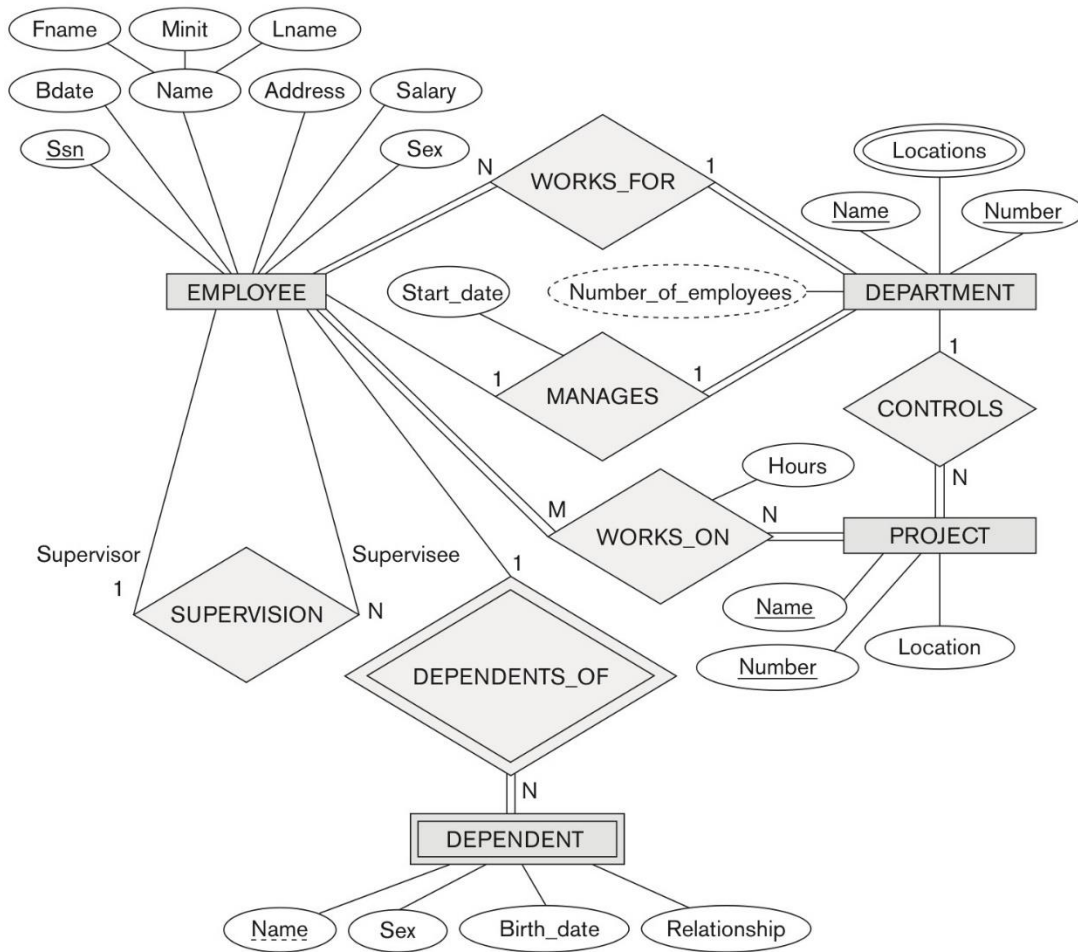
<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

# STEP 6: MAPPING OF MULTIVALUED ATTRIBUTES

---

- For each **multivalued attribute**  $A$ , create a new **relation**  $R$ .
  - **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** or **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**, include **all** its **simple** components.

# STEP 6: MAPPING OF MULTIVALUED ATTRIBUTES



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

## PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

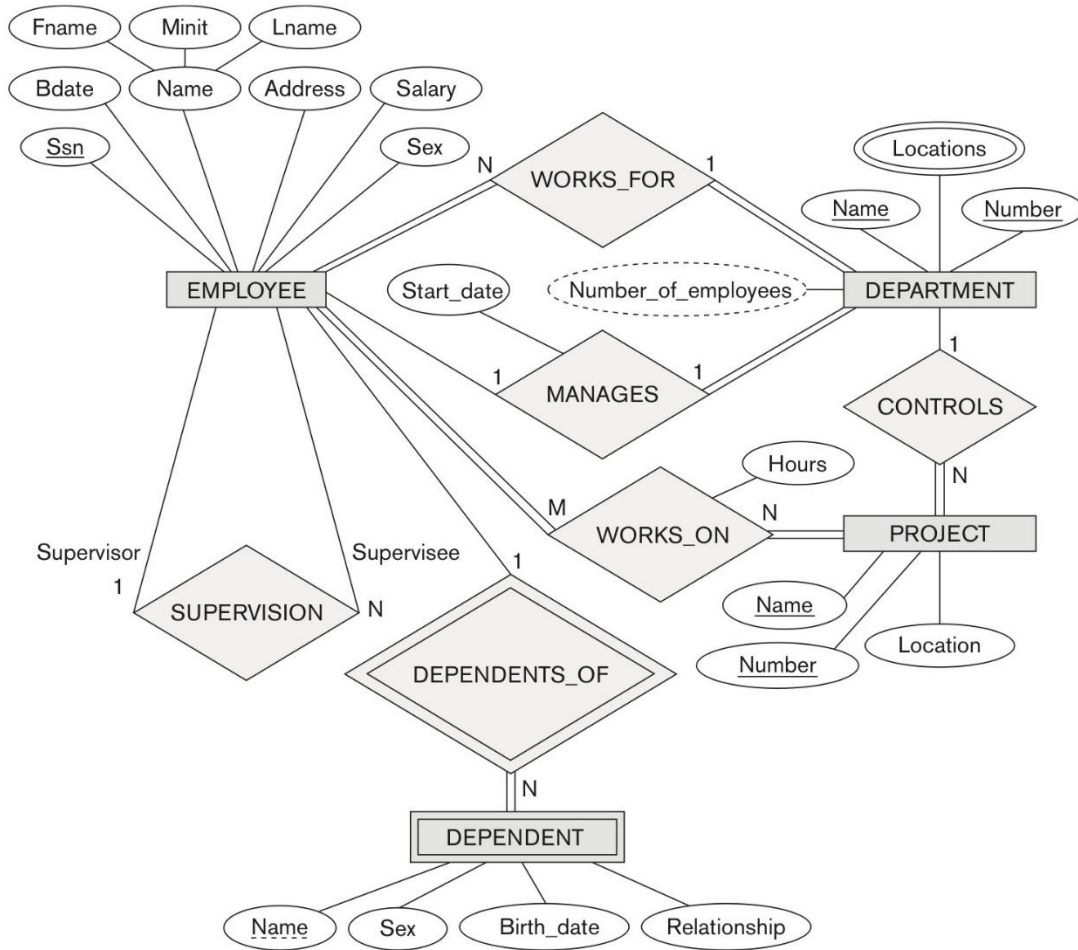
## DEPENDENT

<u>Essn</u>	<u>Dependent name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

## WORKS\_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

# STEP 6: MAPPING OF MULTIVALUED ATTRIBUTES



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

## PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

## DEPENDENT

<u>Essn</u>	<u>Dependent name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

## WORKS\_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

## DEPT\_LOCATIONS

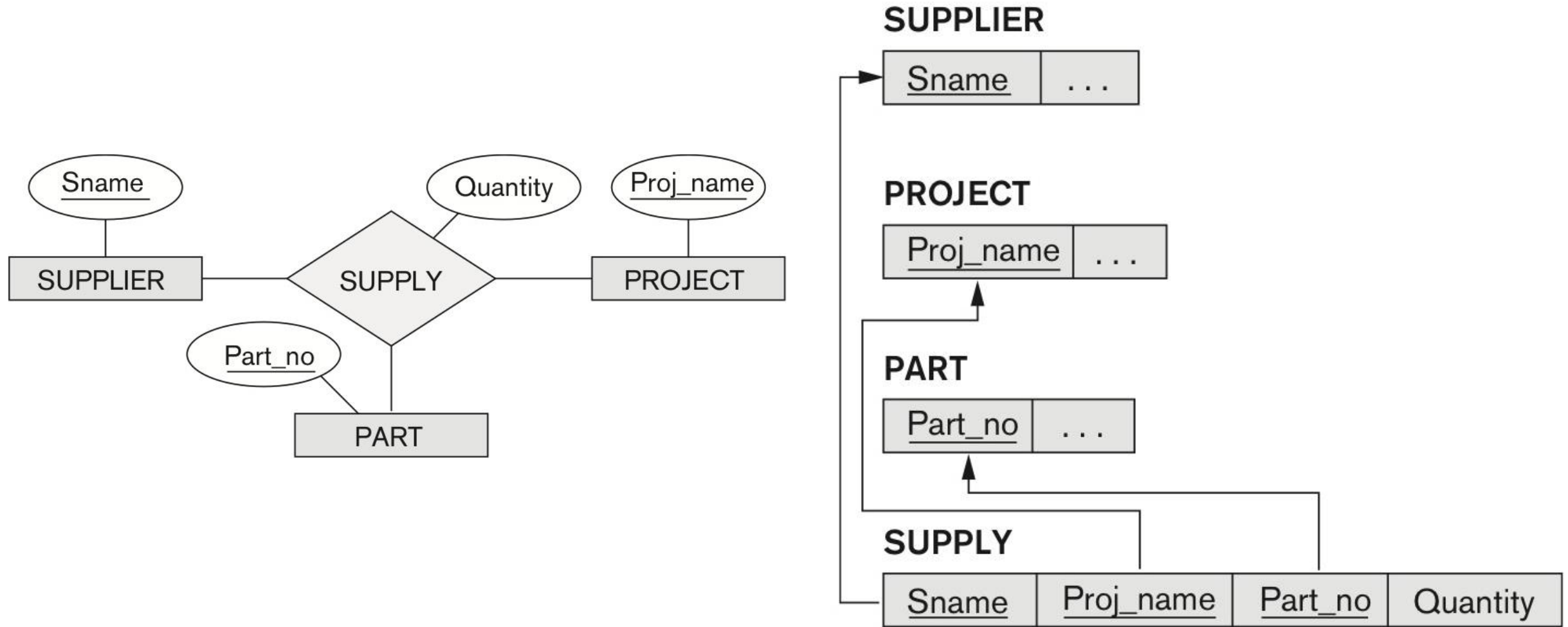
<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

# STEP 7: MAPPING OF N-ARY RELATIONSHIPS

---

- For each **n-ary relationship type**  $R$  ( $n > 2$ ), create a new **relationship**  $S$  to represent  $R$ .
  - Include as **foreign key attributes** in  $S$  the **primary key attributes** of the **relations** that represent the **participating entity types**.
  - Also, include any **simple attributes** of the **n-ary relationship type** as **attributes** of  $S$ .
  - The **primary key** of  $S$  is a combination of all the **foreign keys** that reference the **relations** representing the participating entity types.

# STEP 7: MAPPING OF N-ARY RELATIONSHIPS



# MAPPING BETWEEN ER AND RELATIONAL CONCEPTS

ER Model	Relational Model
Entity type	Entity relation
1:1 or 1:N relationship type	Foreign key (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 attributes	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

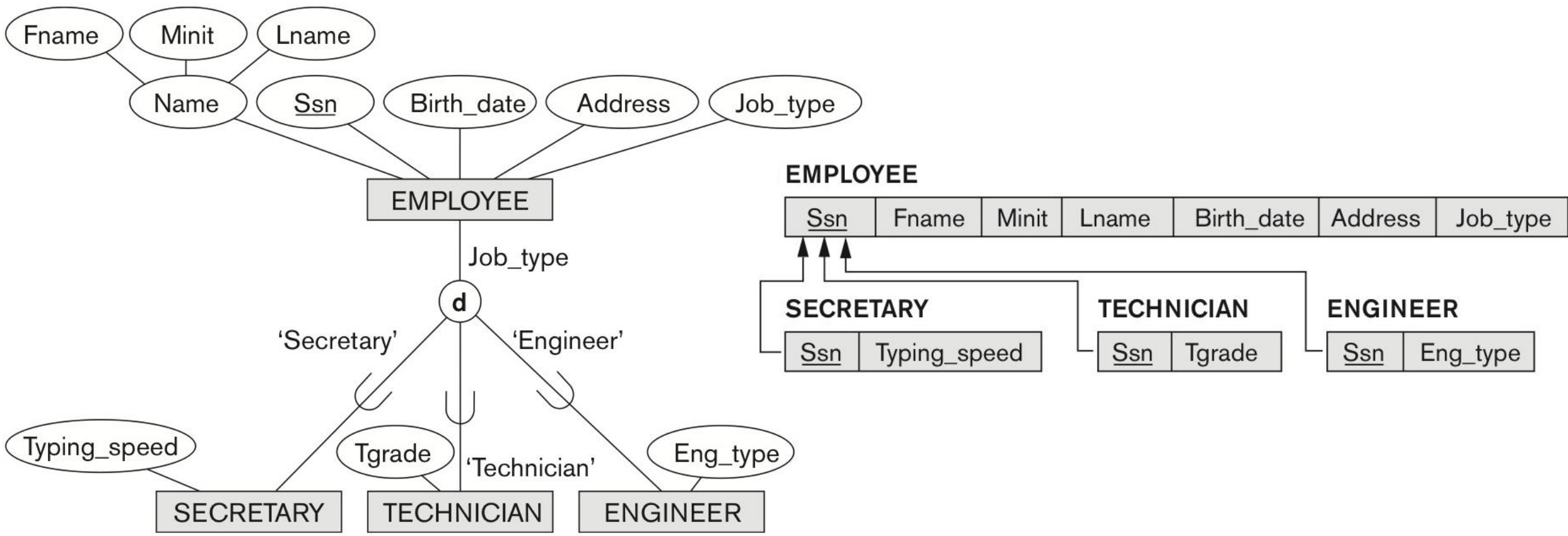
ER and relational models concepts

# STEP 8: MAPPING OF SPECIALIZATIONS & GENERALIZATIONS (1)

- Convert each **specialization** with  $m$  subclasses  $\{S_1, S_2, \dots, S_m\}$  and generalized **superclass**  $C$ , where the **attributes** of  $C$  are  $\{k, a_1, \dots, a_n\}$  and  $k$  is the **primary key**, into **relational schemas** using one of the four following options:
  - **Option 8A: Multiple relations - superclass and subclasses relations.**
    - Create a relation  $L$  for  $C$  with attributes  $Attrs(L) = \{k, a_1, \dots, a_n\}$  and  $L[PK] = k$ .
    - Create a relation  $L_i$  for each subclass  $S_i$  with the attributes  $Attrs(L_i) = \{k\} \cup \{attributes\ of\ S_i\}$  and  $L_i[PK] = k$ .
      - Works for any specialization (total or partial, disjoint or overlapping).



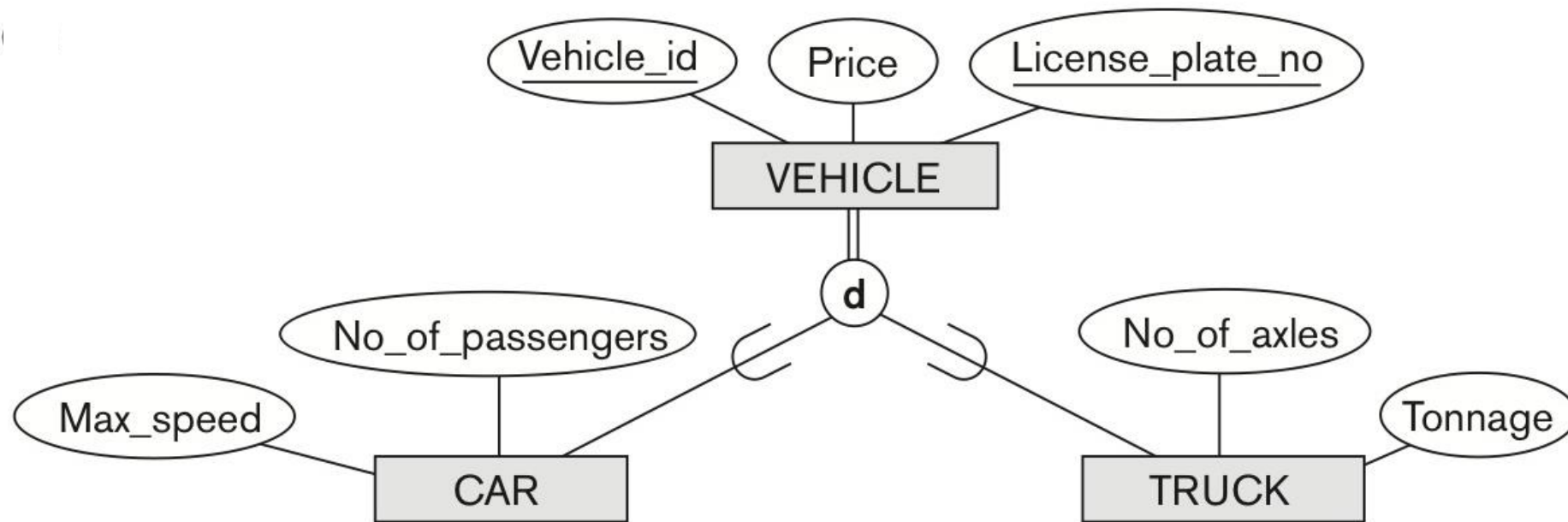
# STEP 8: MAPPING OF SPECIALIZATIONS & GENERALIZATIONS (1)



# STEP 8: MAPPING OF SPECIALIZATIONS & GENERALIZATIONS (2)

- Convert each **specialization** with  $m$  subclasses  $\{S_1, S_2, \dots, S_m\}$  and generalized **superclass**  $C$ , where the **attributes** of  $C$  are  $\{k, a_1, \dots, a_n\}$  and  $k$  is the **primary key**, into **relational schemas** using one of the four following options:
  - **Option 8B: Multiple relations - subclass relations only.**
    - Create a relation  $L_i$  for each subclass  $S_i$  with the attributes  $Attr(L_i) = \{attributes\ of\ S_i\} \cup \{k, a_1, \dots, a_n\}$  and  $L_i[PK] = k$ .
      - Works for a total & disjoint specialization.

# STEP 8: MAPPING OF SPECIALIZATIONS & GENERALIZATIONS (2)



## CAR

<u>Vehicle_id</u>	License_plate_no	Price	Max_speed	No_of_passengers
-------------------	------------------	-------	-----------	------------------

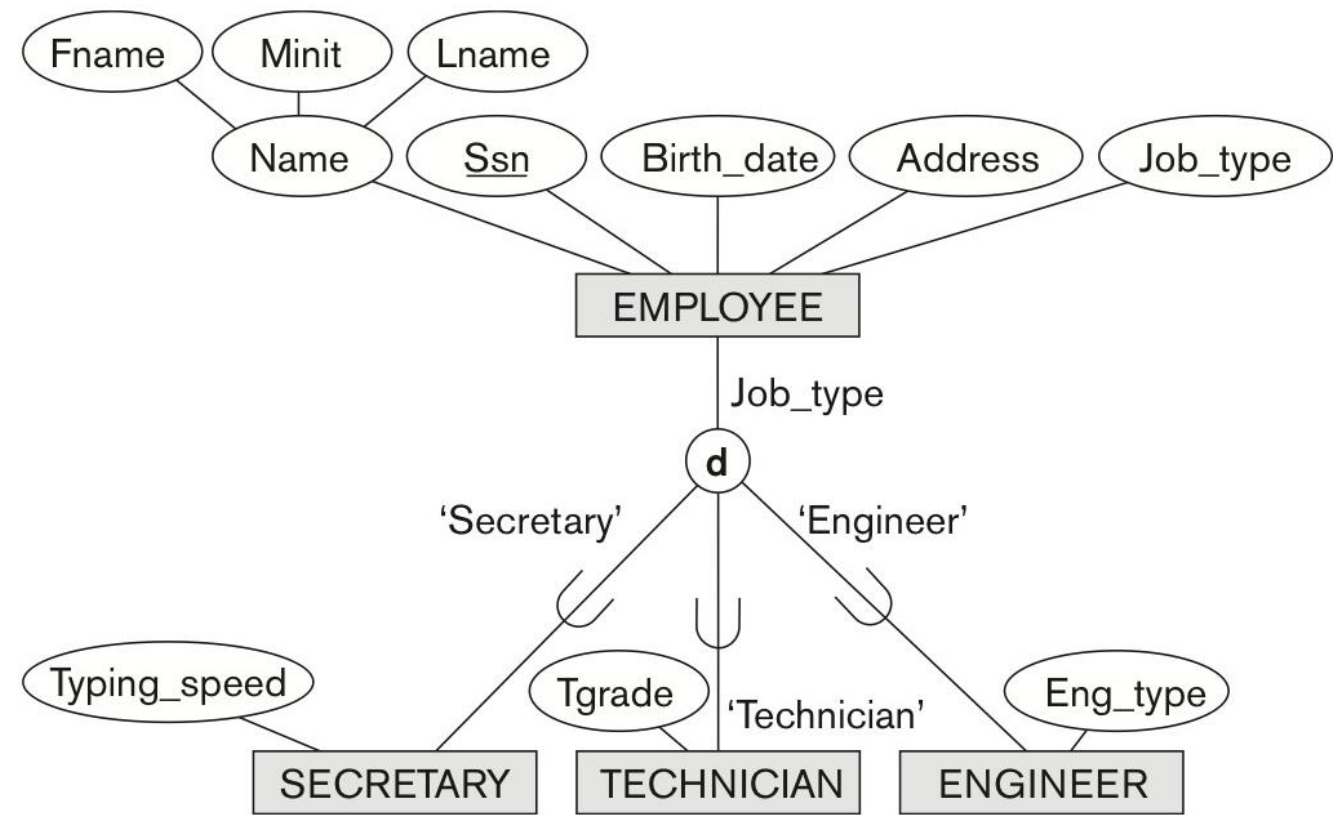
## TRUCK

<u>Vehicle_id</u>	License_plate_no	Price	No_of_axles	Tonnage
-------------------	------------------	-------	-------------	---------

# STEP 8: MAPPING OF SPECIALIZATIONS & GENERALIZATIONS (3)

- Convert each **specialization** with  $m$  subclasses  $\{S_1, S_2, \dots, S_m\}$  and generalized **superclass**  $C$ , where the **attributes** of  $C$  are  $\{k, a_1, \dots, a_n\}$  and  $k$  is the **primary key**, into **relational schemas** using one of the four following options:
  - **Option 8C: Single relation with one type attribute.**
    - Create a single relation  $L$  with attributes  $Attrs(L) = \{k, a_1, \dots, a_n\} \cup \{attributes\ of\ S_1\} \cup \dots \cup \{attributes\ of\ S_m\} \cup \{t\}$  and  $L[PK] = k$ .
    - The attribute  $t$  is called a type (or discriminating) attribute that indicates the subclass to which each tuple belongs.
      - Works for a disjoint specialization.

# STEP 8: MAPPING OF SPECIALIZATIONS & GENERALIZATIONS (3)



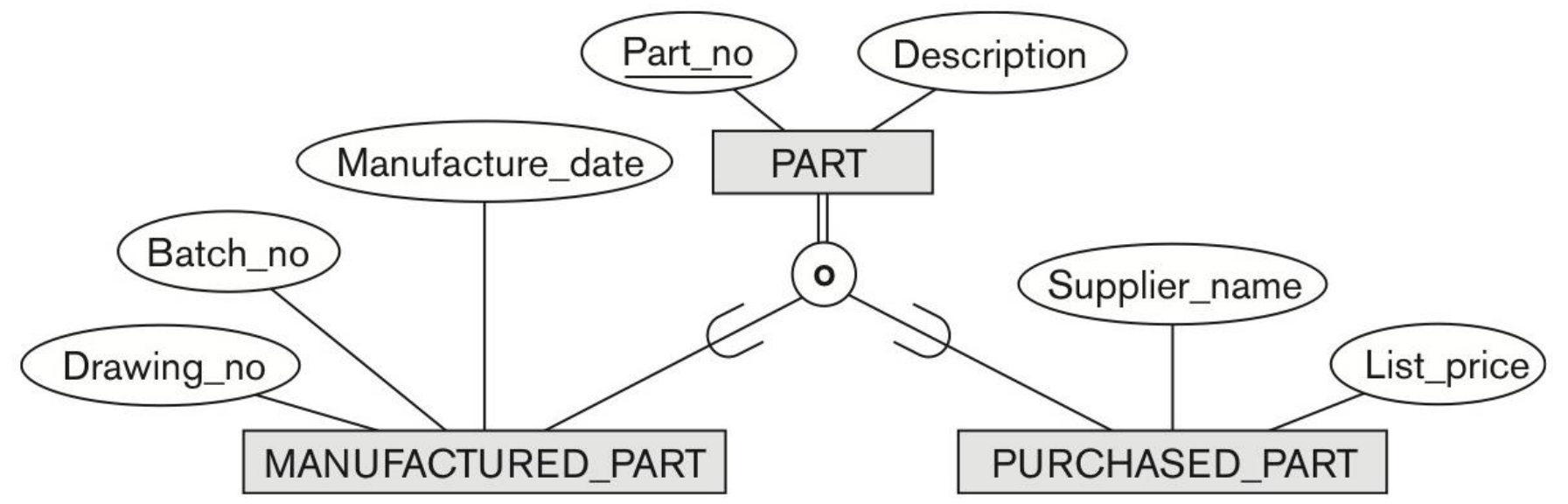
**EMPLOYEE**

<u>Ssn</u>	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type
------------	-------	-------	-------	------------	---------	----------	--------------	--------	----------

# STEP 8: MAPPING OF SPECIALIZATIONS & GENERALIZATIONS (4)

- Convert each **specialization** with  $m$  subclasses  $\{S_1, S_2, \dots, S_m\}$  and generalized **superclass**  $C$ , where the **attributes** of  $C$  are  $\{k, a_1, \dots, a_n\}$  and  $k$  is the **primary key**, into **relational schemas** using one of the four following options:
  - **Option 8D: Single relation with multiple type attributes.**
    - Create a single relation schema  $L$  with attributes  $Attrs(L) = \{k, a_1, \dots, a_n\} \cup \{attributes\ of\ S_1\} \cup \dots \cup \{attributes\ of\ S_m\} \cup \{t_1, t_2, \dots, t_m\}$  and  $L[PK] = k$ .
    - Each  $t_i$  is a Boolean type attribute indicating whether a tuple belongs to the subclass  $S_i$ .
      - Works for an overlapping specialization.

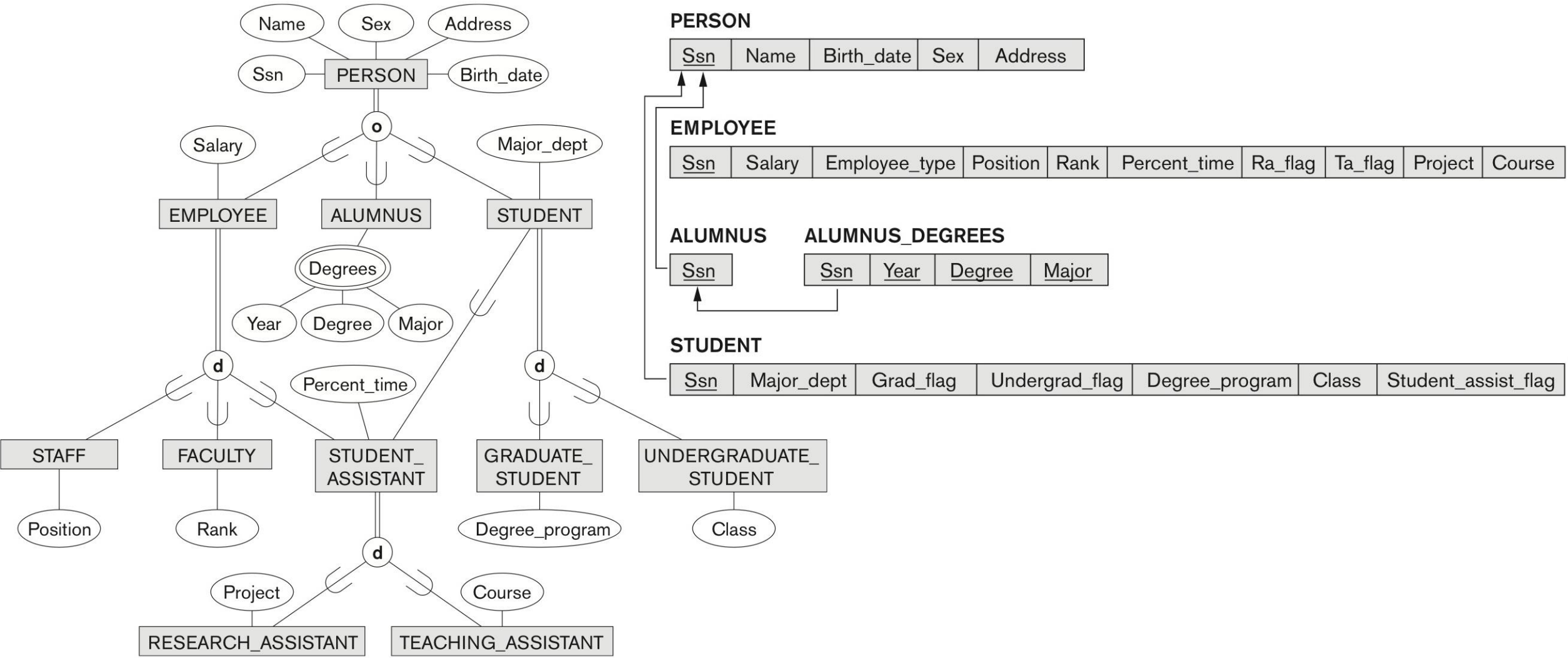
# STEP 8: MAPPING OF SPECIALIZATIONS & GENERALIZATIONS (4)



**PART**

<u>Part_no</u>	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price
----------------	-------------	-------	------------	------------------	----------	-------	---------------	------------

# STEP 8: MAPPING OF SPECIALIZATIONS & GENERALIZATIONS (5)



Mapping of specialization (multiple)

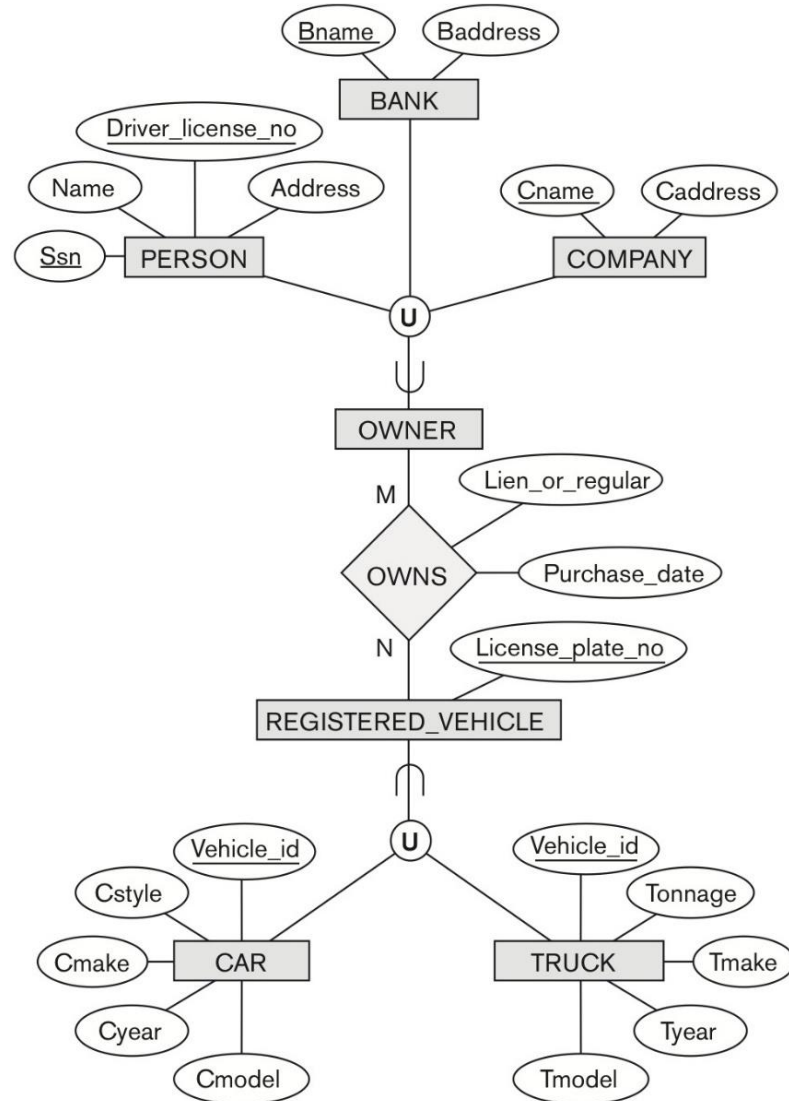


# STEP 9: MAPPING OF UNION TYPES (CATEGORIES)

---

- For mapping a **category** whose **defining superclass** has **different keys**, it is customary to specify a new **key attribute**, called a **surrogate key**, when creating a **relation** to correspond to the **category**.

# STEP 9: MAPPING OF UNION TYPES (CATEGORIES)



## PERSON

<u>Ssn</u>	Driver_license_no	Name	Address	Owner_id
------------	-------------------	------	---------	----------

## BANK

<u>Bname</u>	Baddress	Owner_id
--------------	----------	----------

## COMPANY

<u>Cname</u>	Caddress	Owner_id
--------------	----------	----------

## OWNER

<u>Owner_id</u>
-----------------

## REGISTERED\_VEHICLE

<u>Vehicle_id</u>	License_plate_number
-------------------	----------------------

## CAR

<u>Vehicle_id</u>	Cstyle	Cmake	Cmodel	Cyear
-------------------	--------	-------	--------	-------

## TRUCK

<u>Vehicle_id</u>	Tmake	Tmodel	Tonnage	Tyear
-------------------	-------	--------	---------	-------

## OWNS

<u>Owner_id</u>	<u>Vehicle_id</u>	Purchase_date	Lien_or_regular
-----------------	-------------------	---------------	-----------------

Mapping of union types (categories)