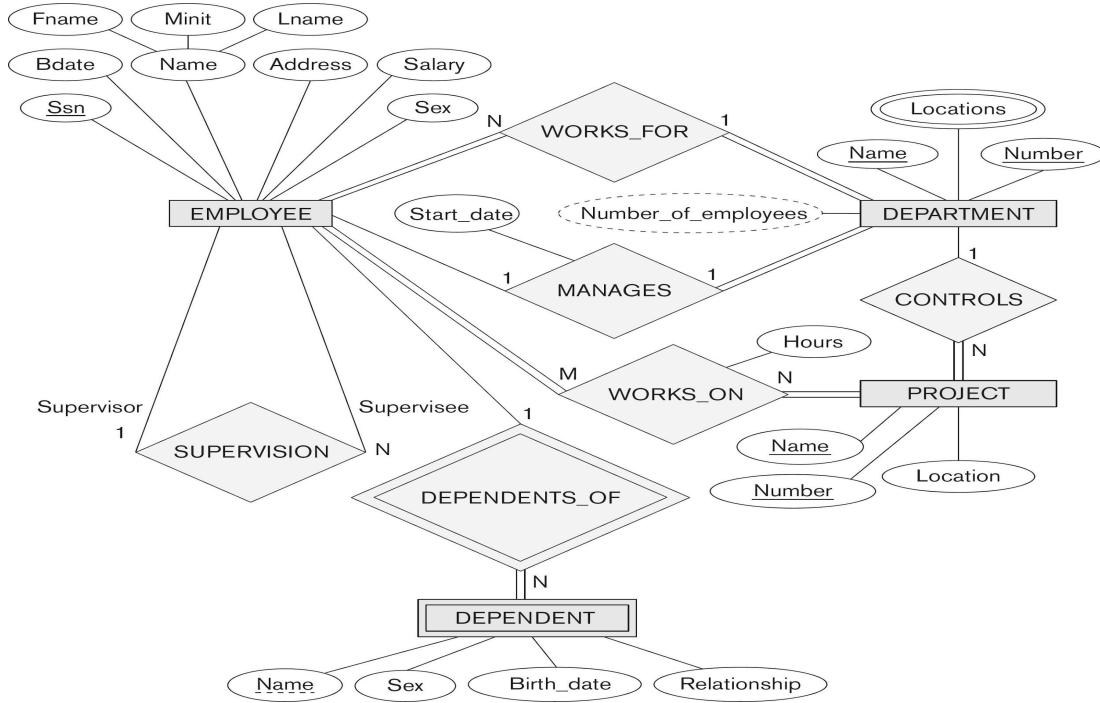
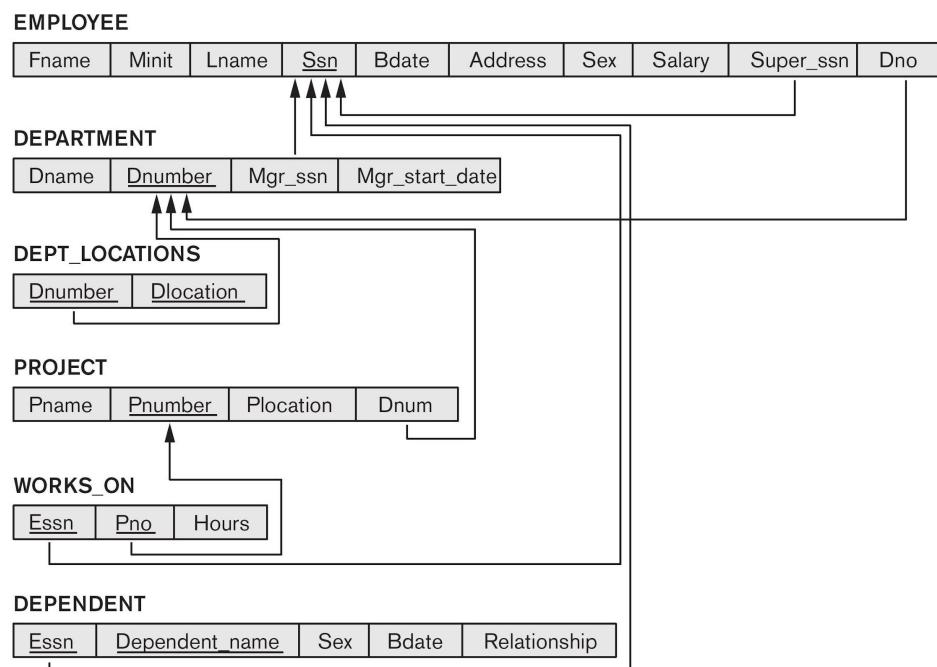


Example



2

Resulting Relational Schema



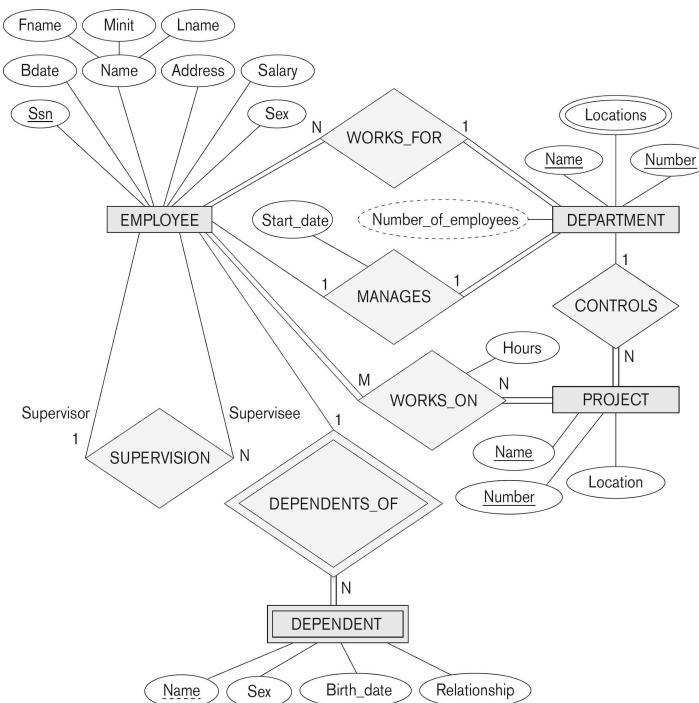
3

Step 1: Regular Entity Types

- i. For regular/strong entity type, create a corresponding relation that includes all the simple attributes (includes simple attributes of composite relations)
- ii. Choose one of the key attributes as primary
 - If composite, the simple attributes together form the primary key
- iii. Any remaining key attributes are kept as secondary unique keys (these will be useful for physical tuning w.r.t. indexing analysis)

4

Example



5

Step 1 Result

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary
-------	-------	-------	-----	-------	---------	-----	--------

DEPARTMENT

Dname	Dnumber
-------	---------

PROJECT

Pname	Pnumber	Plocation
-------	---------	-----------

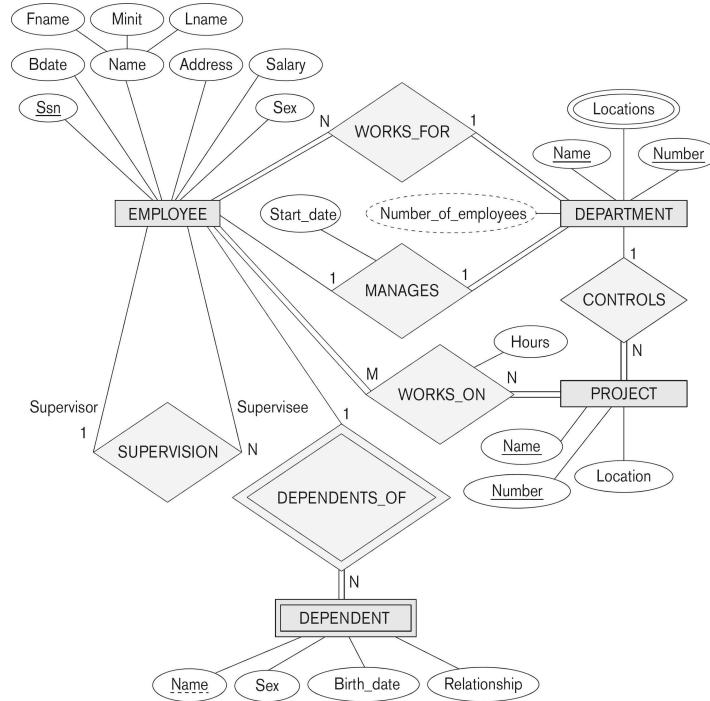
6

Step 2: Weak Entity Types

- i. For weak entity type, create a corresponding relation that includes all the simple attributes
- ii. Add as a foreign key all of the primary key attribute(s) in the entity corresponding to the owner entity type
- iii. The primary key is the combination of all the primary key attributes from the owner and the partial key of the weak entity, if any

7

Example



8

Step 2 Result

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>	Dependent_name	Sex	Bdate	Relationship
-------------	----------------	-----	-------	--------------

9

Step 3: Mapping Binary 1-to-1

Three approaches

- **Foreign Key**

- Usually appropriate

- Merged Relation

- Possible when both participations are total

- Relationship Relation

- Not discussed

10

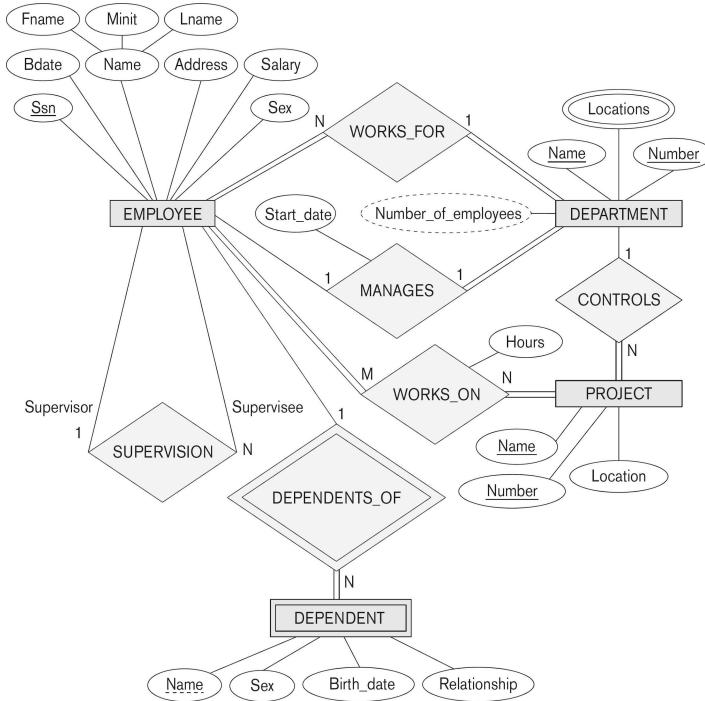
Step 3: Mapping Binary 1-to-1

Foreign Key

- i. Choose one relation as S , the other T
 - Better if S has total participation (reduces number of NULL values)
- ii. Add to S all the simple attributes of the relationship
- iii. Add as a foreign key in S the primary key attributes of T

11

Example



12

Step 2 Result

EMPLOYEE

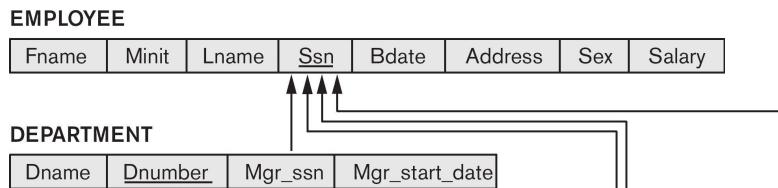
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary
-------	-------	-------	-----	-------	---------	-----	--------

DEPARTMENT

Dname	Dnumber
-------	---------

13

Step 3 Result



14

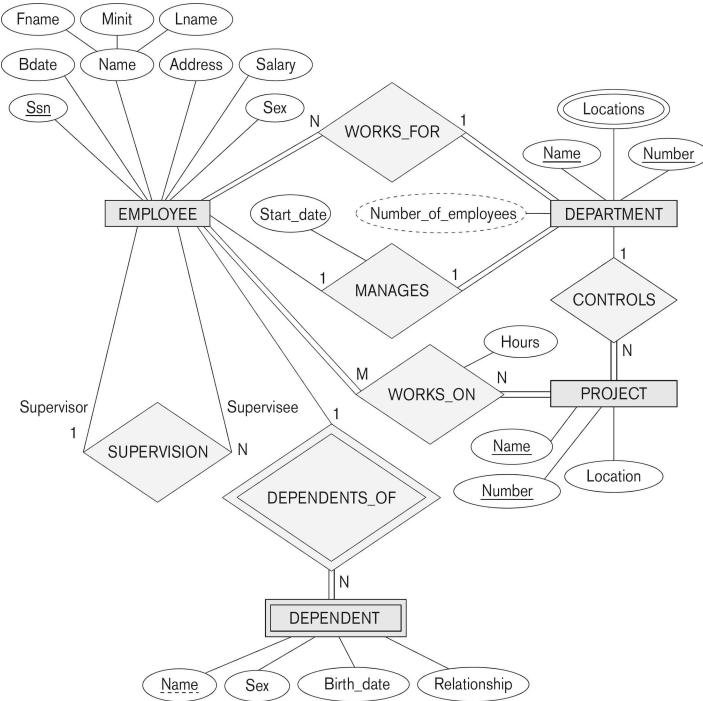
Step 4: Binary 1-to-N

- i. Choose the S relation as the type at the N -side of the relationship, other is T
- ii. Add as a foreign key to S all of the primary key attribute(s) of T

Another approach: create a relationship relation

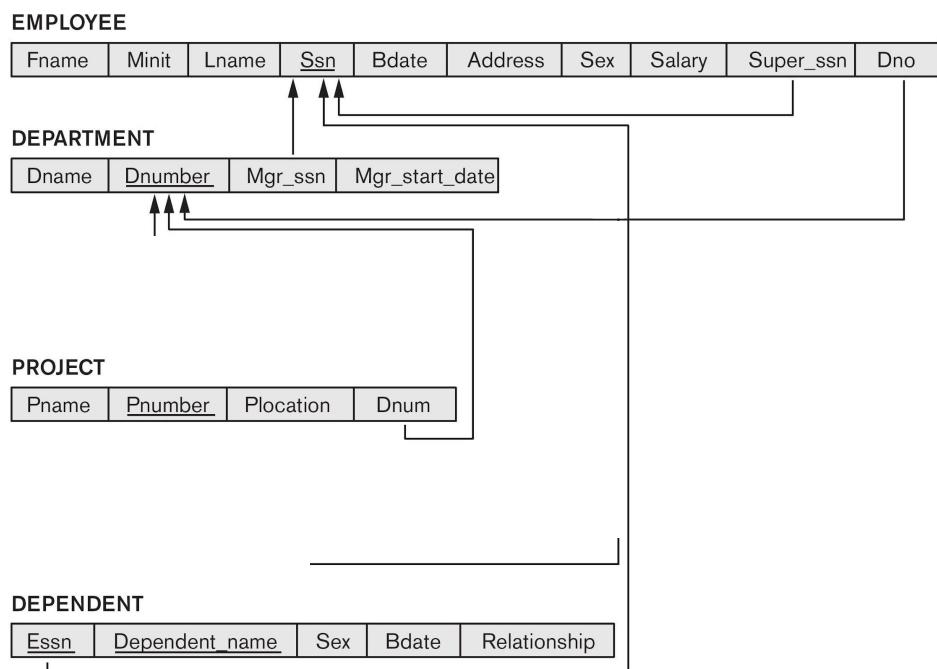
15

Example



16

Step 4 Result



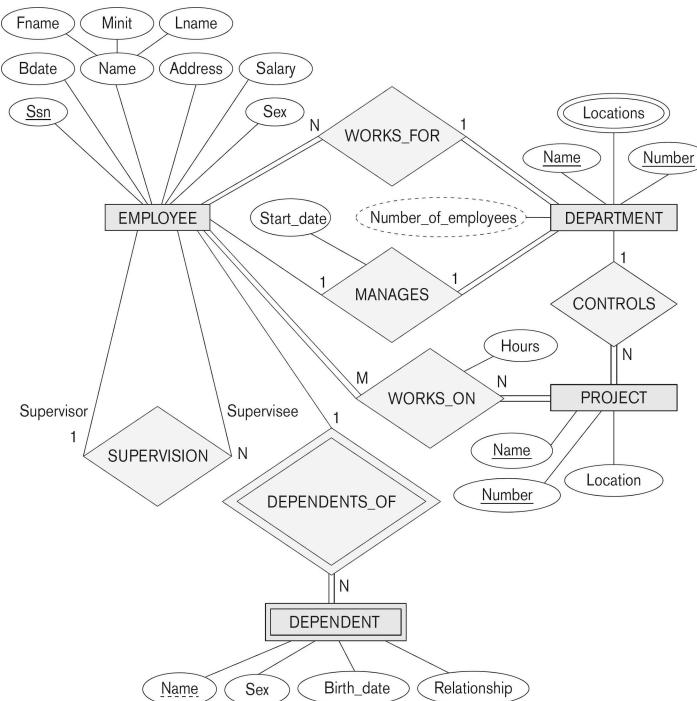
17

Step 5: Binary M-to-N

- i. Create a new relation S (termed: *relationship relation*)
 - In some ERD dialects, actually drawn in
- ii. Add as foreign keys the primary keys of both relations; their combination forms the primary key of S
- iii. Add any simple attributes of the M:N relationship to S

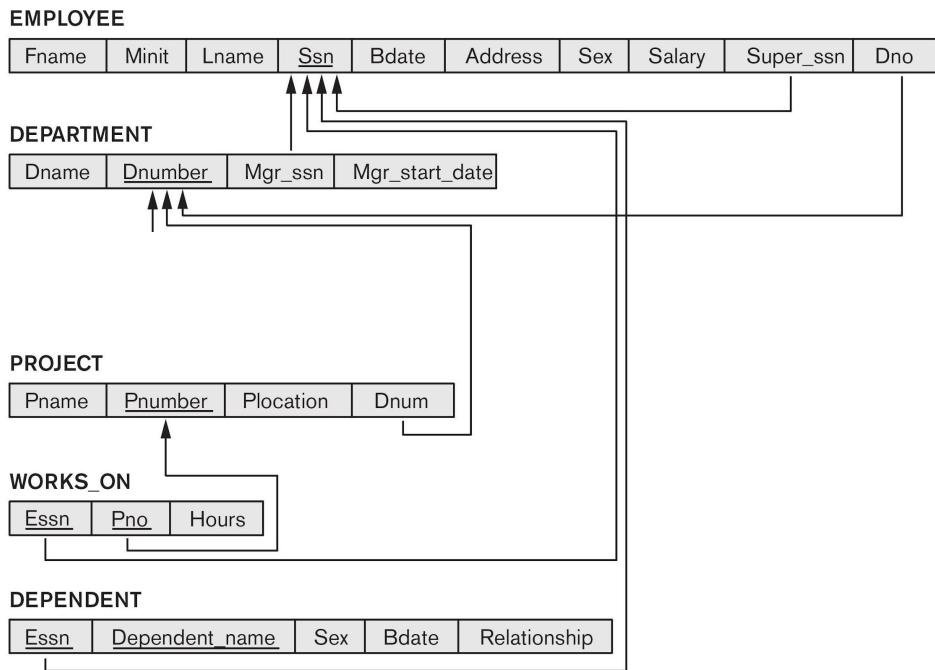
18

Example



19

Step 5 Result



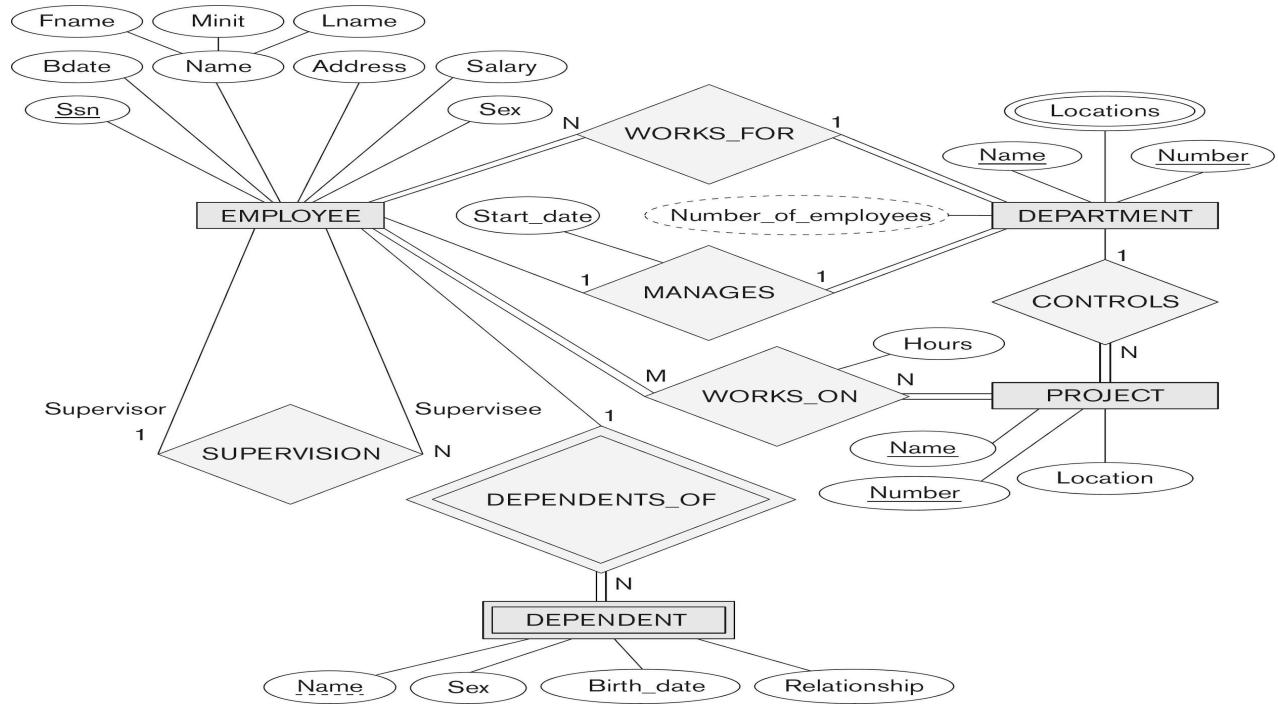
20

Step 6: Multivalued Attributes

- i. Create a new relation S
- ii. Add as foreign keys the primary keys of the corresponding relation
- iii. Add the attribute to S (if composite, the simple attributes); the combination of all attributes in S forms the primary key

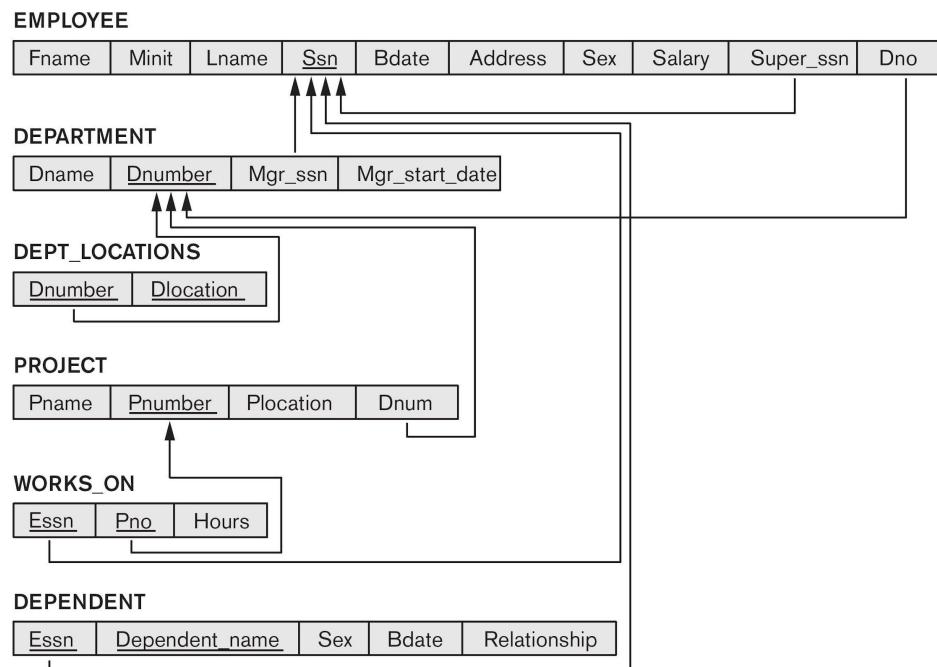
21

Example



22

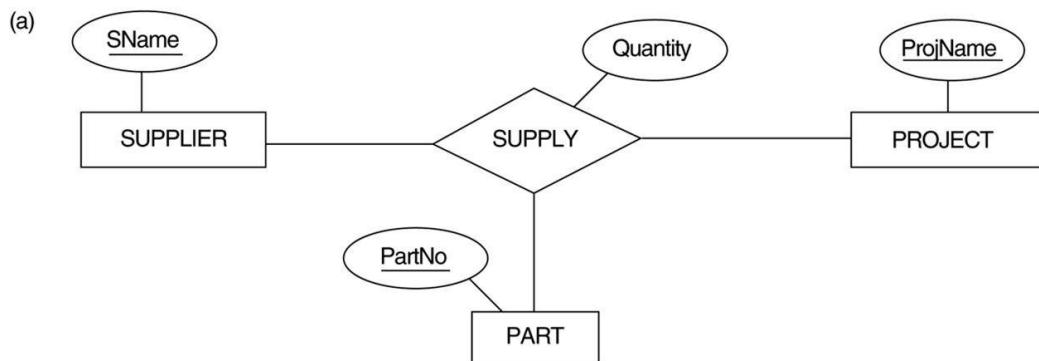
Step 6 Result



23

■ **Step 7: 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.
- **Example:** The relationship type SUPPY in the ER on the next slide.
- This can be mapped to the relation SUPPLY shown in the relational schema, whose primary key is the combination of the three foreign keys {SNAME, PARTNO, PROJNAME}



SUPPLIER	
<u>SNAME</u>	...

PROJECT	
<u>PROJNAME</u>	...

PART	
<u>PARTNO</u>	...

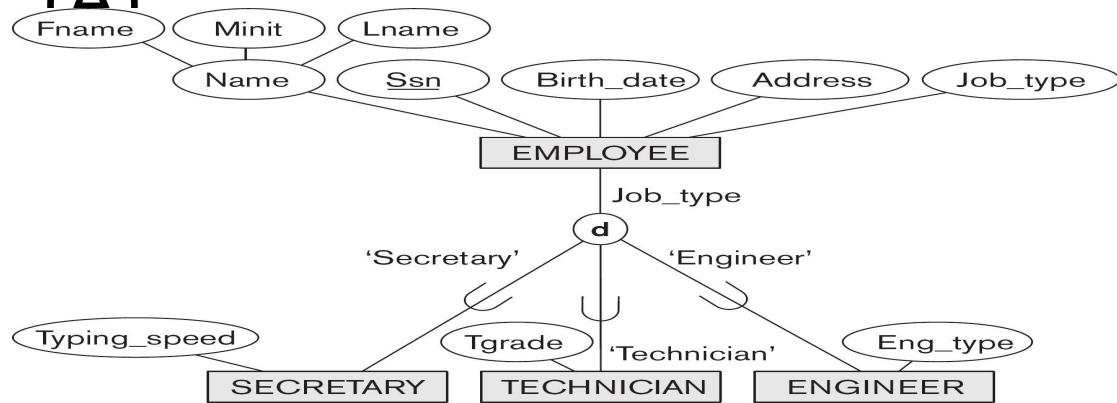
SUPPLY			
<u>SNAME</u>	<u>PROJNAME</u>	<u>PARTNO</u>	QUANTITY

Step 8: Specialization/Generalization

- A. Multiple relations – subclass and superclass
 - Usually works (assumes unique id at parent)
- B. Multiple relations – subclass only
 - Should only be used for disjoint
- C. Single relation with one type attribute
 - Only for disjoint, can result in many NULLs
- D. Single relation with multiple type attributes
 - Better for overlapping, could be disjoint

Specialization/Generalization

(Δ)

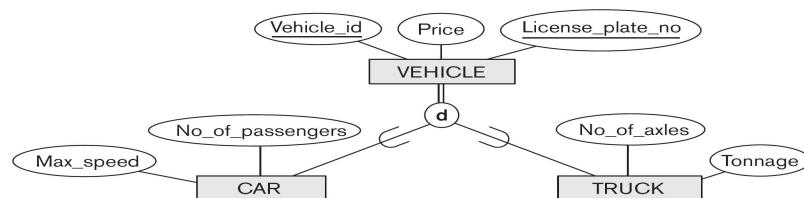


(a) EMPLOYEE



28

Specialization/Generalization (P')



CAR

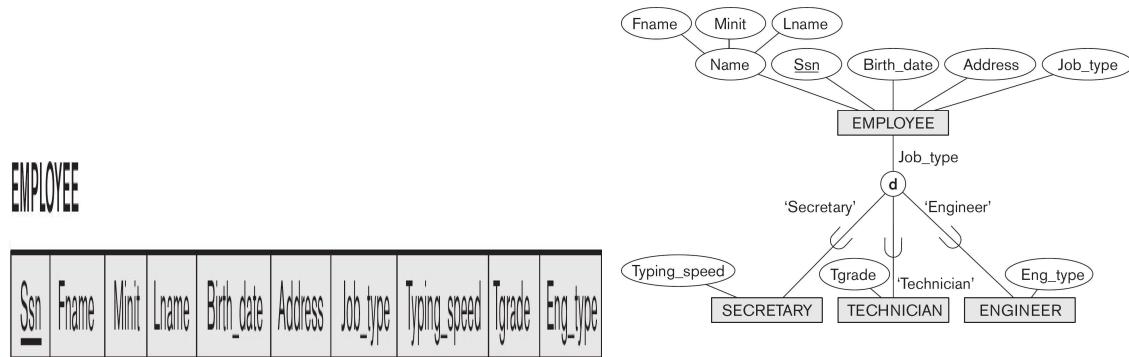
<u>Vehicle_id</u>	License_plate_no	Price	Max_speed	No_of_passengers
-------------------	------------------	-------	-----------	------------------

TRUCK

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

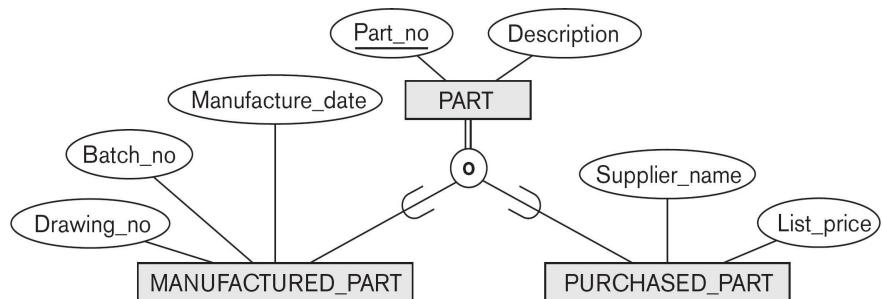
29

Specialization/Generalization (C)



30

Specialization/Generalization (D)



(d) PART

Part_no	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price
---------	-------------	-------	------------	------------------	----------	-------	---------------	------------

31