

# Mapping



# Summary of notation for ER diagrams

**Figure 3.14**  
Summary of the  
notation for ER  
diagrams.

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of $E_2$ in $R$
	Cardinality Ratio 1: N for $E_1:E_2$ in $R$
	Structural Constraint (min, max) on Participation of $E$ in $R$

# Relational Database Definitions

- Table or entity: a collection of records
- Attribute or Column or field: a Characteristic of an entity
- Row or Record or tuple: the specific characteristics of one entity
- Database: a collection of tables
- Server: a collection of DBs

# Relational Database

The diagram illustrates a relational database table structure. The table has columns labeled "SSAN", "Name", and "Date of Birth". The "Date of Birth" column contains the values "1/1/2012" and "31/12/2012". The first row, where "SSAN" is a key, is highlighted with grey shading. The second row, which is a tuple, is also highlighted with grey shading. Arrows point from the labels "Relation", "Tuple", "Column", and "SSAN is a key" to their respective components in the table.

SSAN	Name	Date of Birth			
		1/1/2012			
		31/12/2012			

# Mapping ->DB Tables

CUSTOMER

<u>Customer_ID</u>	Customer_Name	Address	City	State	Zip
--------------------	---------------	---------	------	-------	-----

ORDER

<u>Order_ID</u>	Order_Date	<u>Customer_ID</u>
-----------------	------------	--------------------

Primary Key

Foreign Key

ORDER LINE

<u>Order_ID</u>	<u>Product_ID</u>	Quantity
-----------------	-------------------	----------

composite primary key

PRODUCT

<u>Product_ID</u>	Product_Description	Product_Finish	Standard_Price	On_Hand
-------------------	---------------------	----------------	----------------	---------

# ER-to-Relational Mapping

Step 1: Mapping of Regular Entity Types if there is 1 to 1 relationship mandatory

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relation Types

Step 4: Mapping of Binary 1:N Relationship Types.

Step 5: Mapping of Binary M:N Relationship Types.

Step 6: Mapping of N-ary Relationship Types.

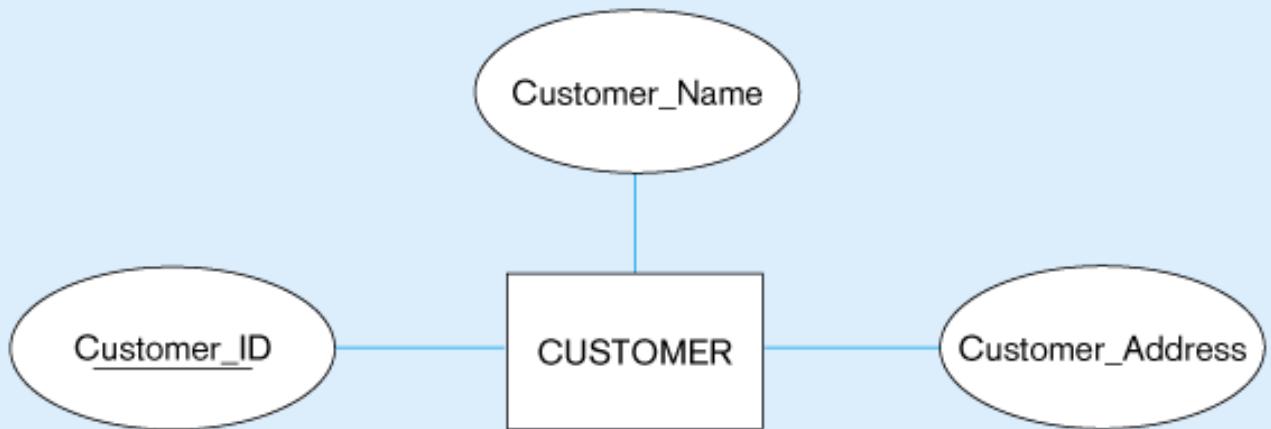
Step 7: Mapping of Unary Relationship.

# Step 1: Mapping of Regular Entity Types

- Create table for each entity type -> if there is no 1-1 relationship mandatory from 2 sides
- Choose one of key attributes to be the primary key

# Mapping Regular entity

(a) CUSTOMER entity type with simple attributes

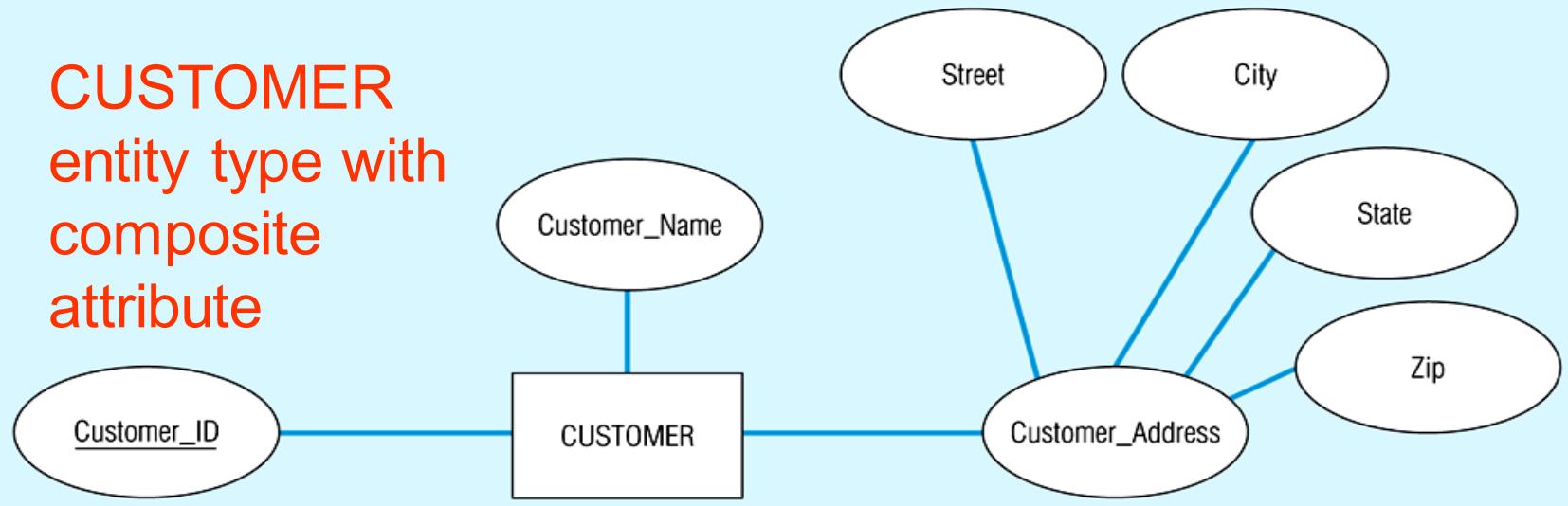


(b) CUSTOMER relation

CUSTOMER		
<u>Customer_ID</u>	Customer_Name	Customer_Address

# Mapping Composite attribute

CUSTOMER  
entity type with  
composite  
attribute

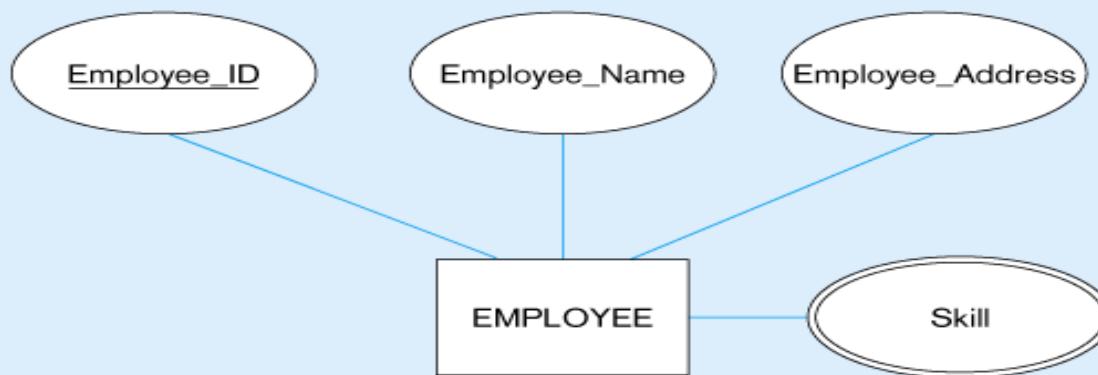


CUSTOMER relation with address detail

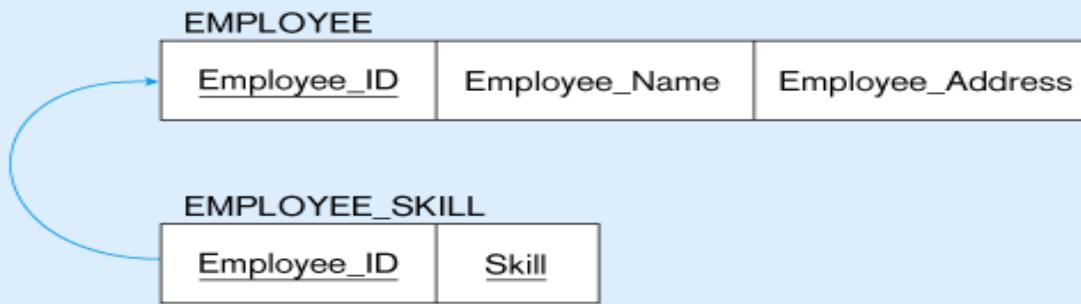
CUSTOMER

<u>Customer_ID</u>	Customer_Name	Street	City	State	Zip

# Mapping Multivalued Attribute



Multivalued attribute becomes a separate relation with foreign key

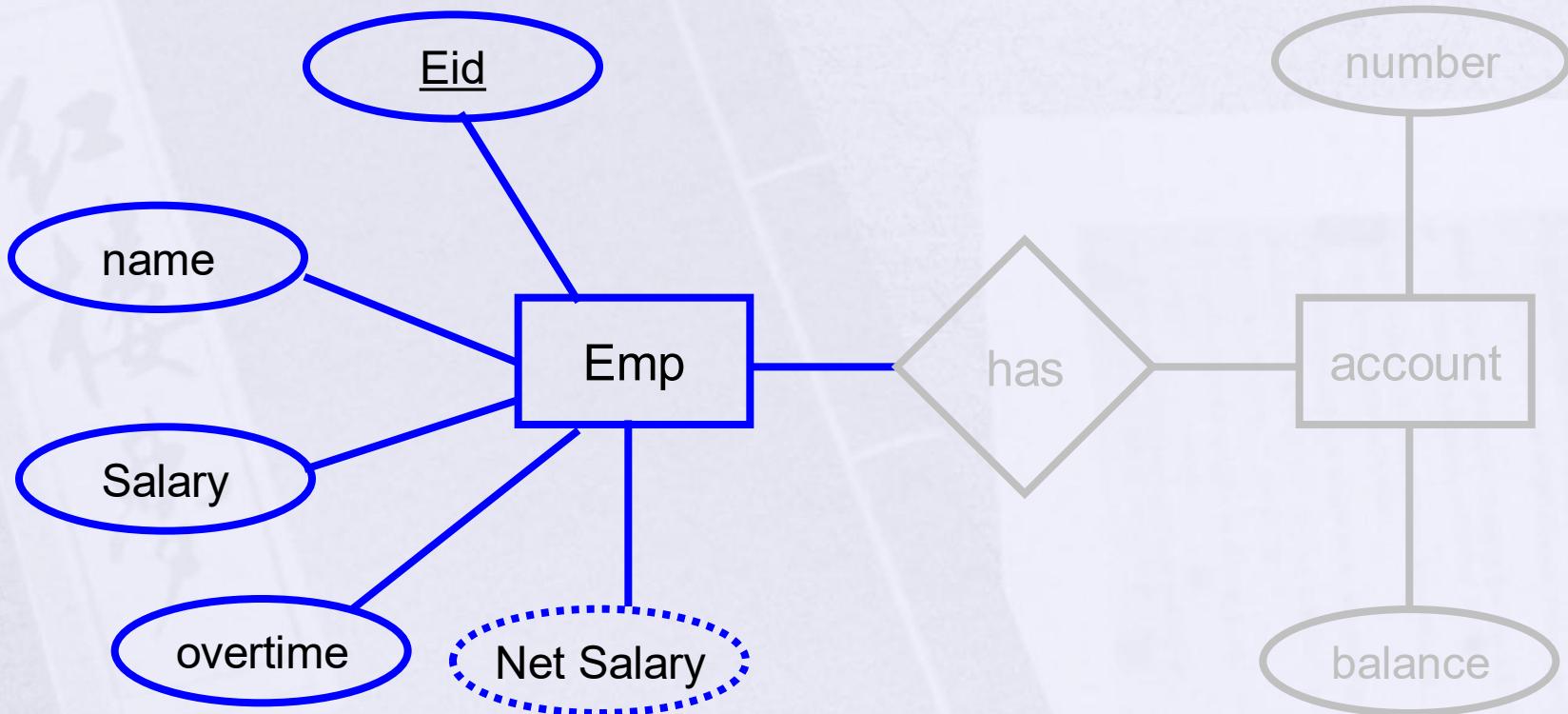


1 – to – many relationship between original entity and new relation

# Mapping Derived & Complex

- In the most cases Derived attribute not be stored in DB
- Mapping Complex Like Mapping Multivalued attribute then including parts of the multivalued attributes as columns in DB

# Derived Attribute

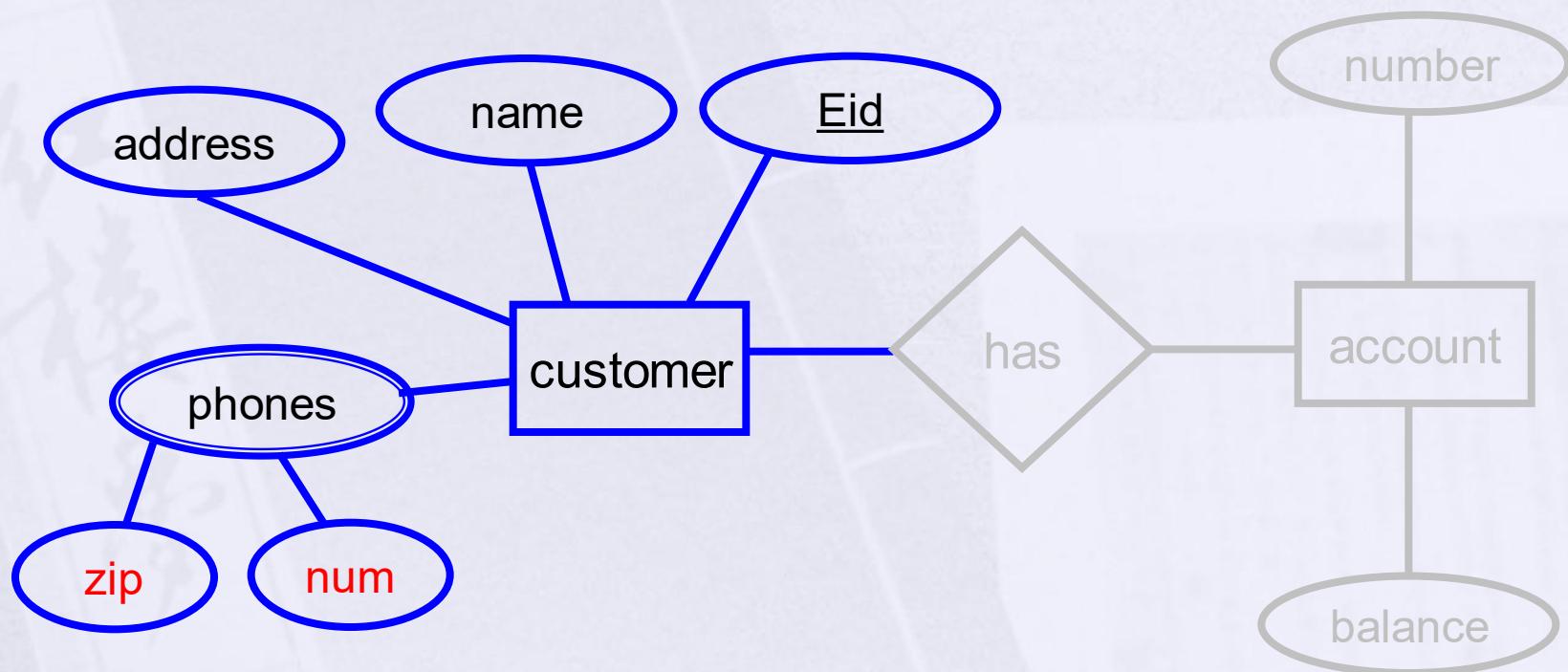


Employee(Eid, Name, salary, overtime)

Note

$\text{Netsal} = \text{salary} + \text{overtime}$

# Complex Attribute



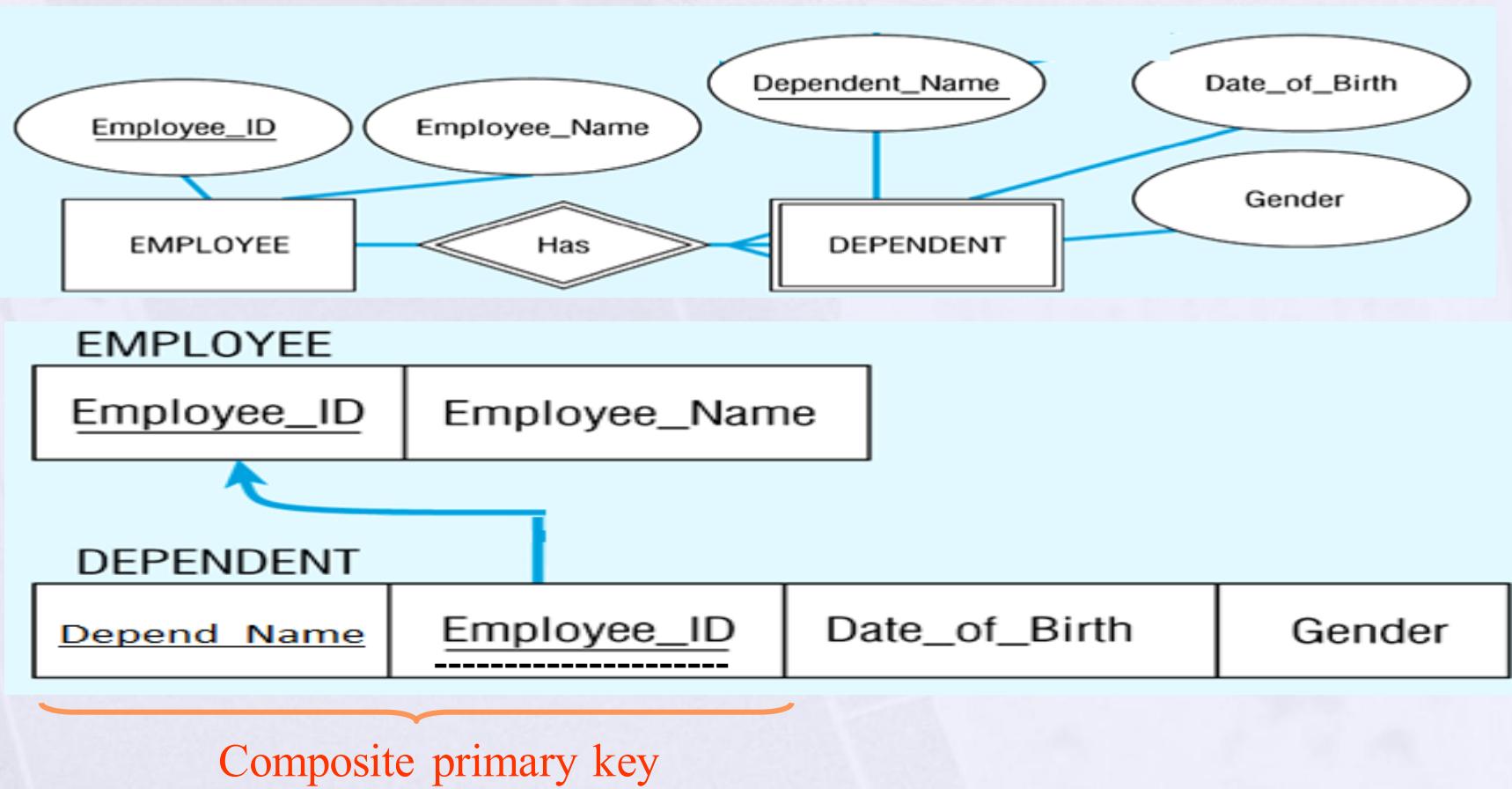
**Customer(Eid, Name, address)**

**Emp\_phones(Eid, Zip, Num)**

## Step 2: Mapping of Weak Entity Types

- Create table for each weak entity.
- Add foreign key that correspond to the owner entity type.
- Primary key composed of:
  - Partial identifier of weak entity
  - Primary key of identifying relation (strong entity)

# Mapping Weak entity



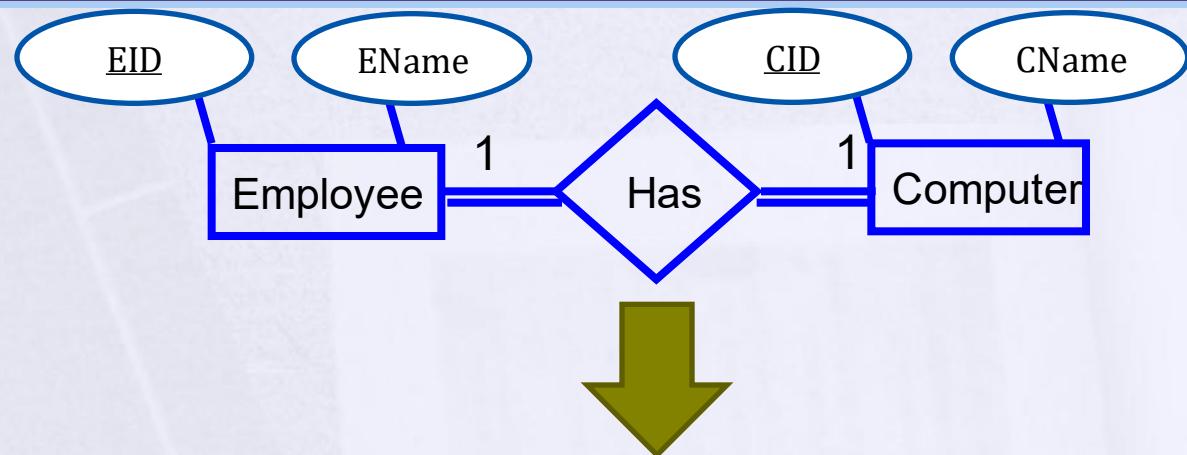
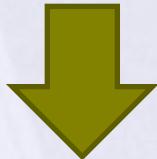
# Step 3: Mapping of Binary 1:1 Relation Types

- Merged two tables if both sides are Mandatory.
- Add FK into table with the total participation relationship to represent optional side.
- Create third table if both sides are optional.

# 2 Mandatory

One-to-One

2 Mandatory



1 table

tbl\_xy (PK,...,...,...)

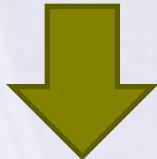
PK = PK<sub>x</sub> or PK<sub>y</sub>

Emp(EID, Ename, Cname, **CID**)

# Optional-Mandatory

**One-to-One**

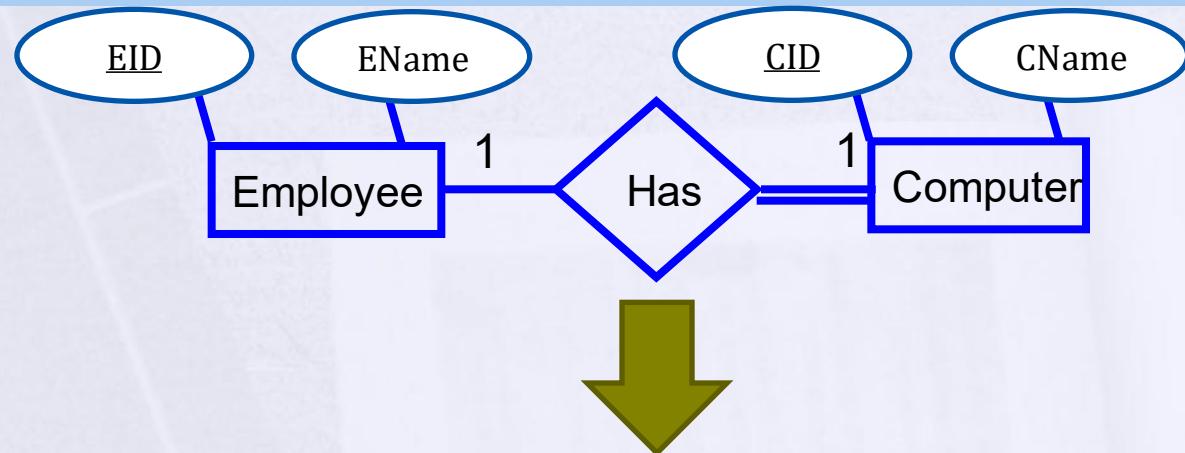
X optional – Y mandatory



**2 tables**

tbl\_x (PKx,.....,.....)

tbl\_y (PKy,.....,.....,PKx....)



Employee(EID, Ename)

Computer(CID, Cname, **EID\_FK**)

-----

# 2 Optional

One-to-One

2 Optional



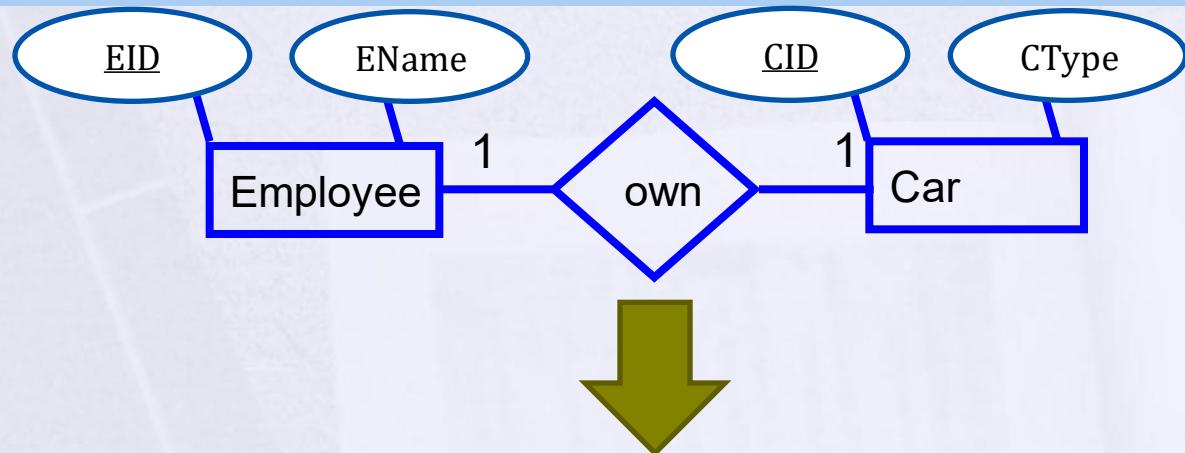
3 tables

tbl\_x (PKx,.....,.....)

tbl\_y (PKy,.....,.....)

tbl\_xy (PKxy,.....,....,FKxy,....)

PKxy = PKx or PKy



Employee(EID, Ename)

Car(CID, CType)

Emp\_Car(EID, CID\_FK)

## Step 4: Mapping of Binary 1:N Relationship Types.

- Add FK to N-side table if N side mandatory  
else you can create another table to represent  
relation

# Many is Mandatory

## One-to-Many

X whatever – Y mandatory

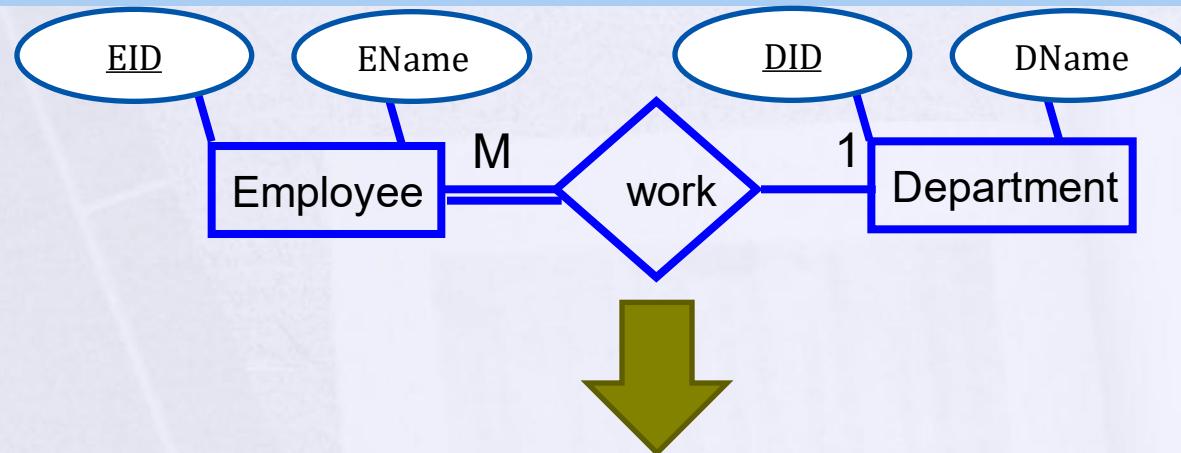


2 tables

tbl\_x (PKx,.....,.....)

tbl\_y (PKy,.....,....,FKy....)

FKy= PKx



Department(DID, Dname)

Employee(EID, Ename,DID)

-----

# Many is Optional

**One-to-Many**

X whatever- Y Optional



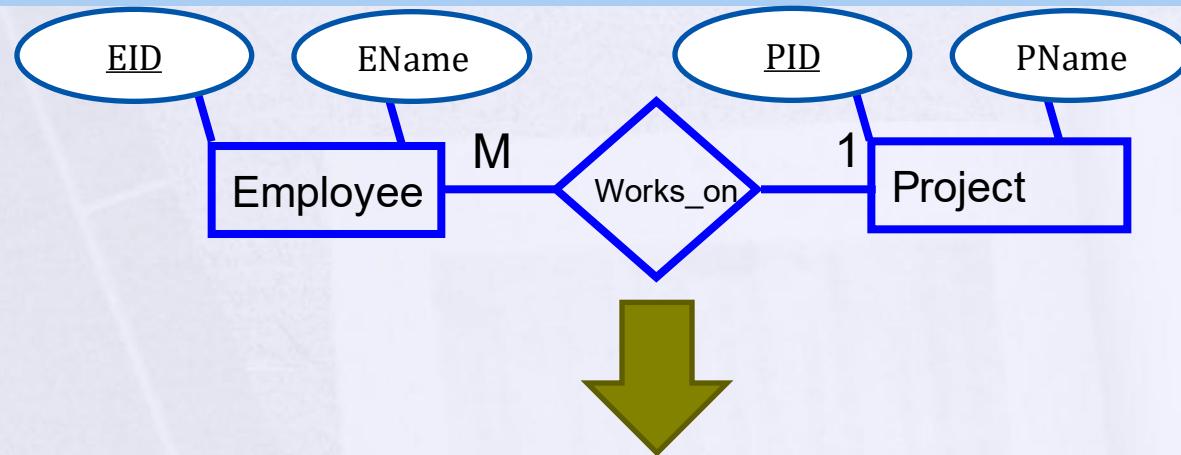
**3 tables**

tbl\_x (PKx,.....,.....)

tbl\_y (PKy,.....,.....)

tbl\_xy (PKxy,.....,.....)

PKxy = PKy



Project(PID, Pname)

Employee(EID, Ename)

Proj\_Emp(EID,PID\_FK)  
----- -----

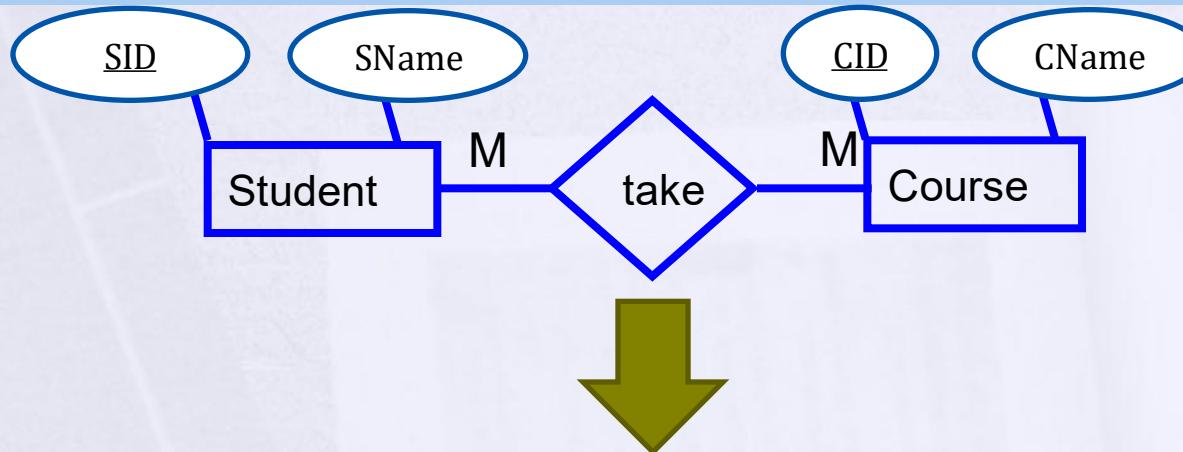
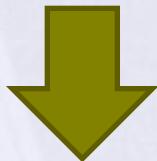
# Step 5: Mapping of Binary M:N Relationship Types.

- Create a new third table
- Add FKs to the new table for both parent tables
- Add simple attributes of relationship to the new table if any .

# M:N

## Many-to-Many

X whatever - Y whatever



3 tables

tbl\_x (PKx,.....,.....)

tbl\_y (PKy,.....,.....)

tbl\_xy (PKx ,PKy, .....,.....)

PKxy=PKx+PKy

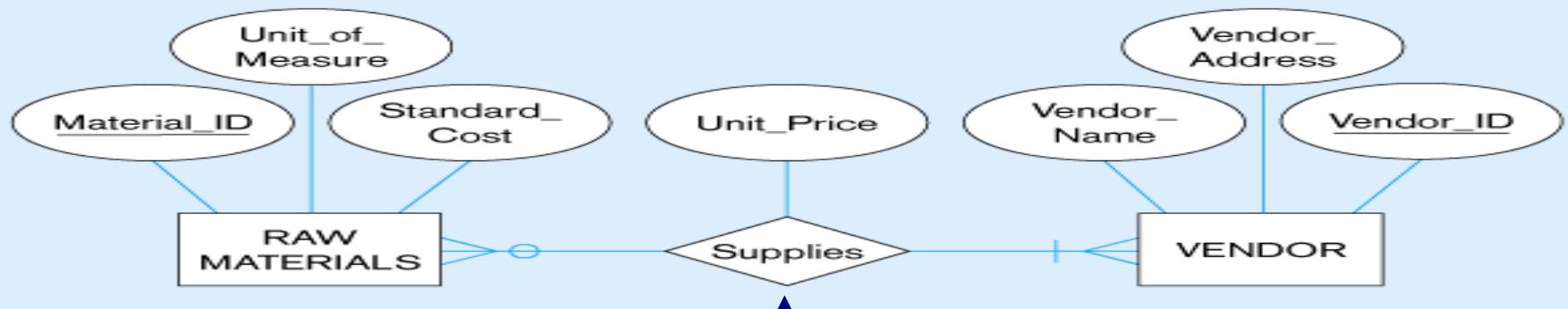
Student(SID, Sname)

Course(CID, Cname)

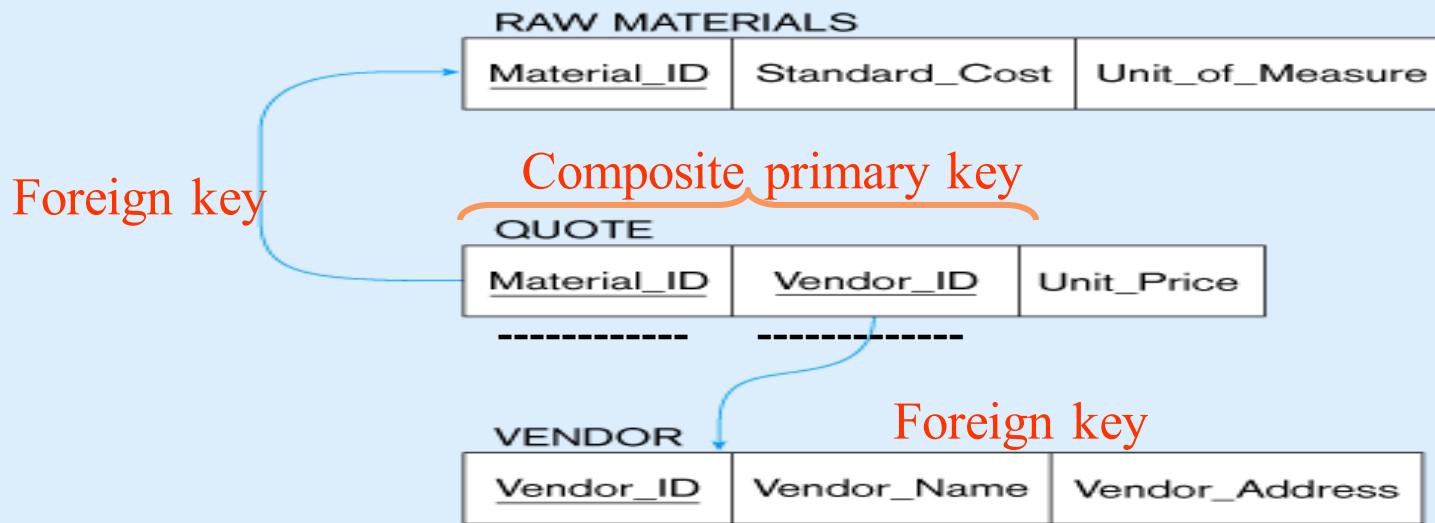
Stud\_Course(SID, CID)

----- -----

# M:N with attribute



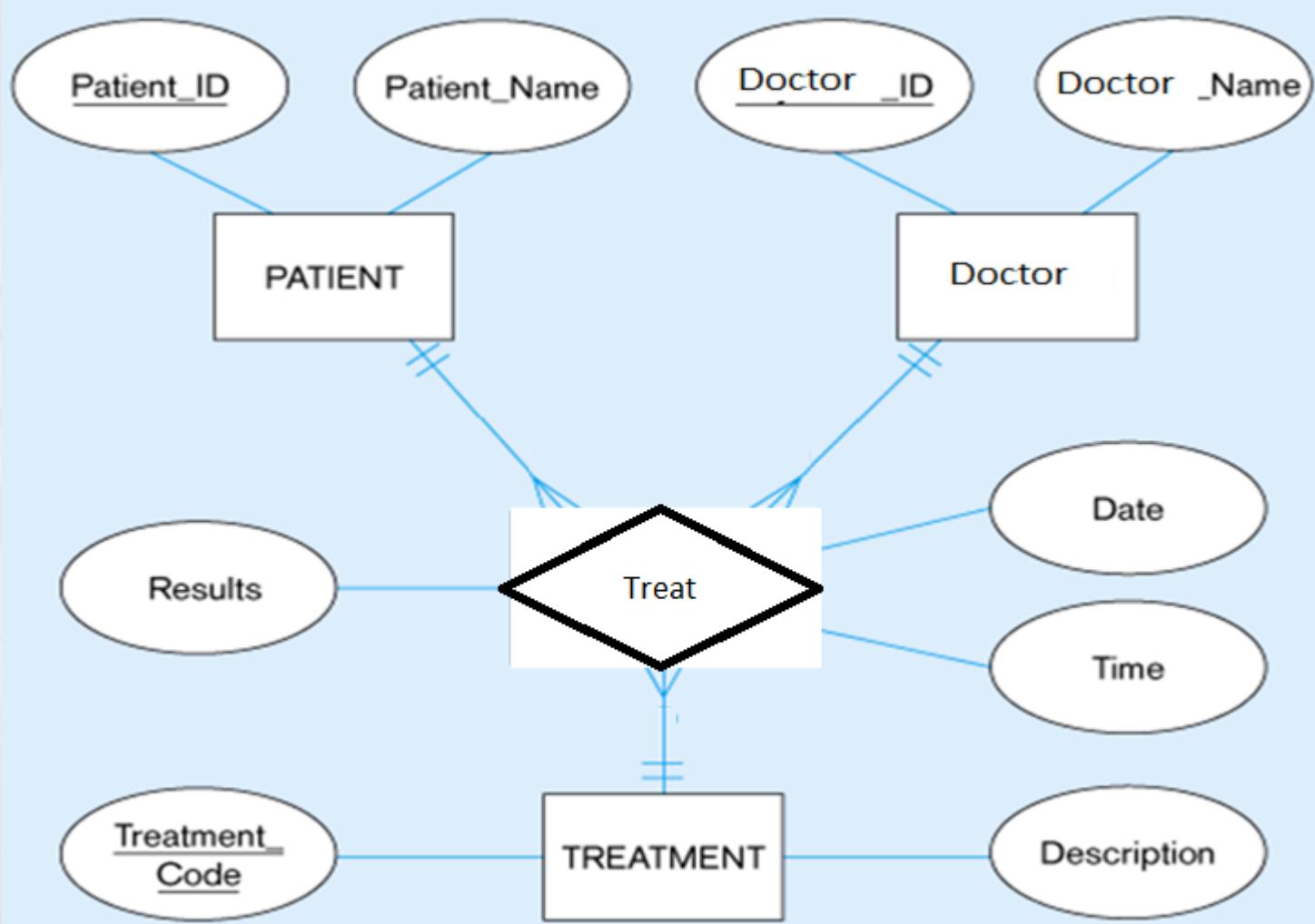
The *Supplies* relationship will need to become a separate relation



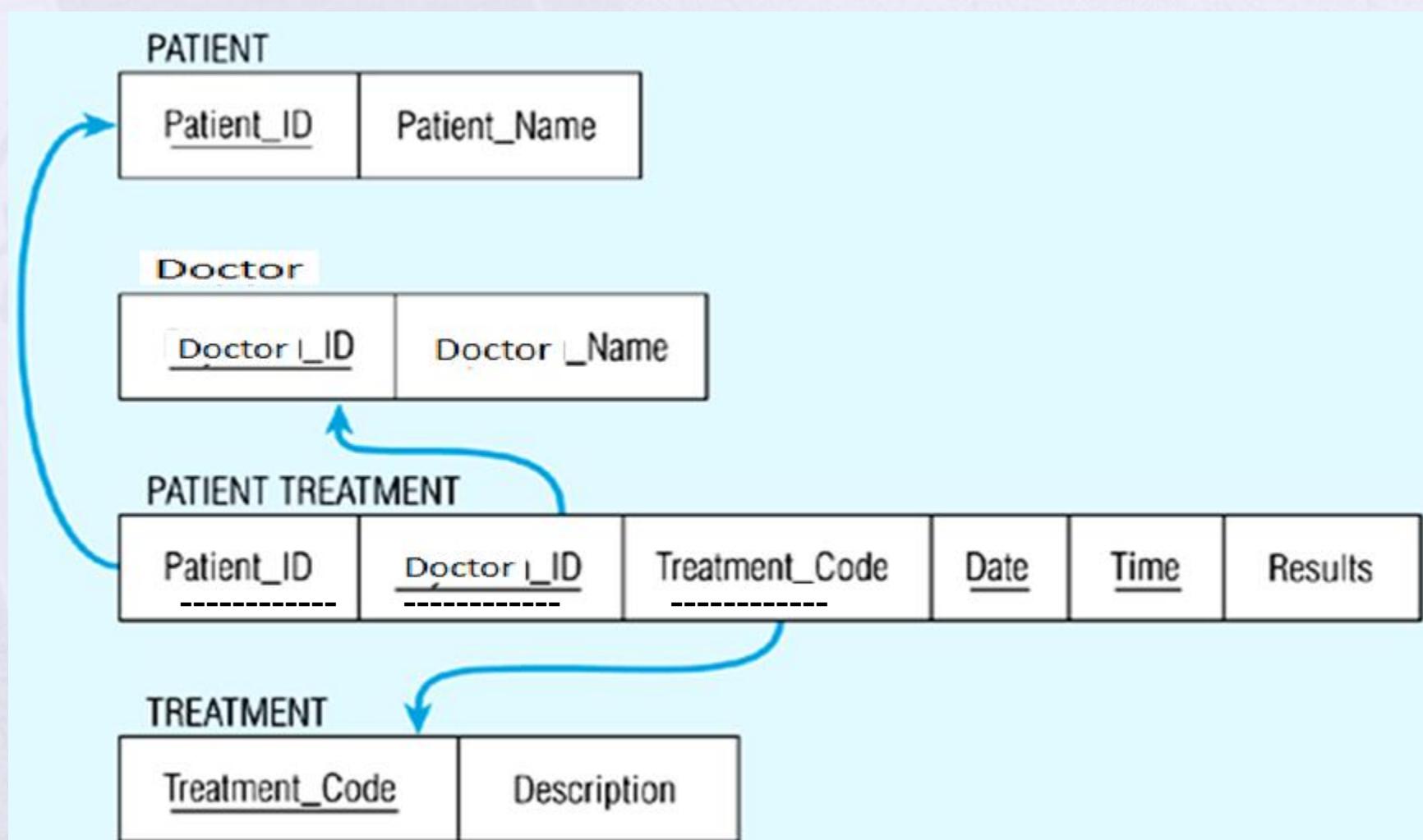
# Step 6: Mapping of N-ary Relationship Types.

- If  $n > 2$  then :
- Create a new table
- Add FKs to the new table for all parent tables

# Step 6: Mapping of N-ary Relationship Types.

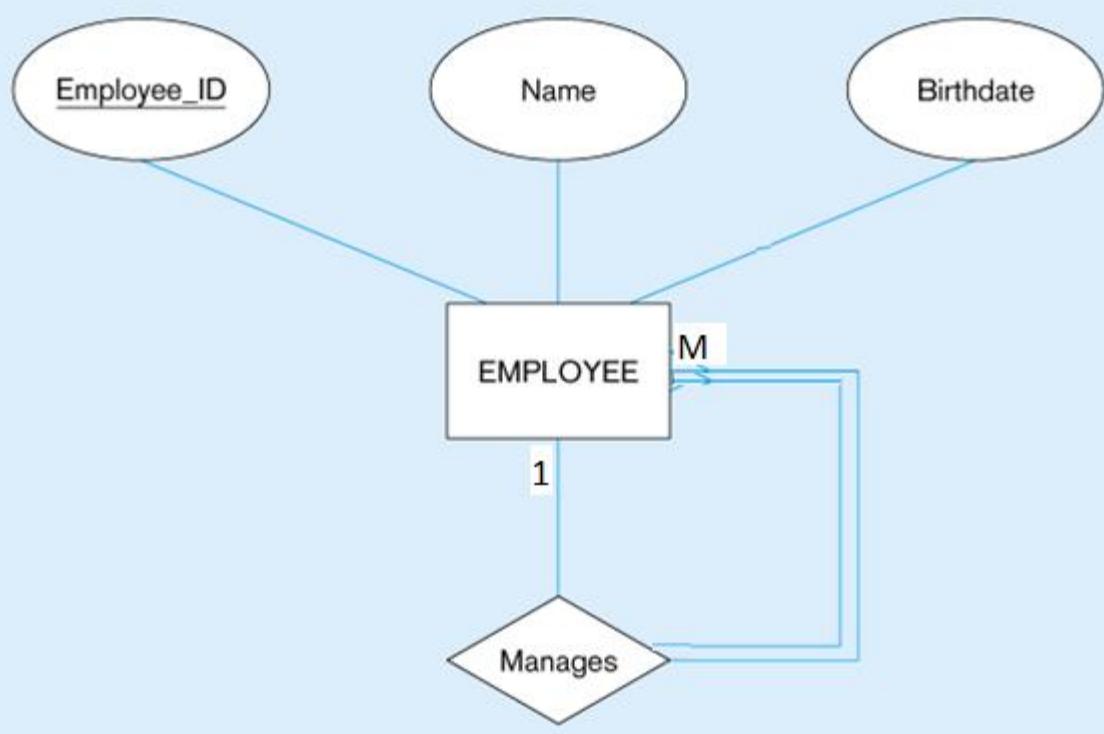


# Step 6: Mapping of N-ary Relationship Types.



# Step 7: Mapping Unary Relationship

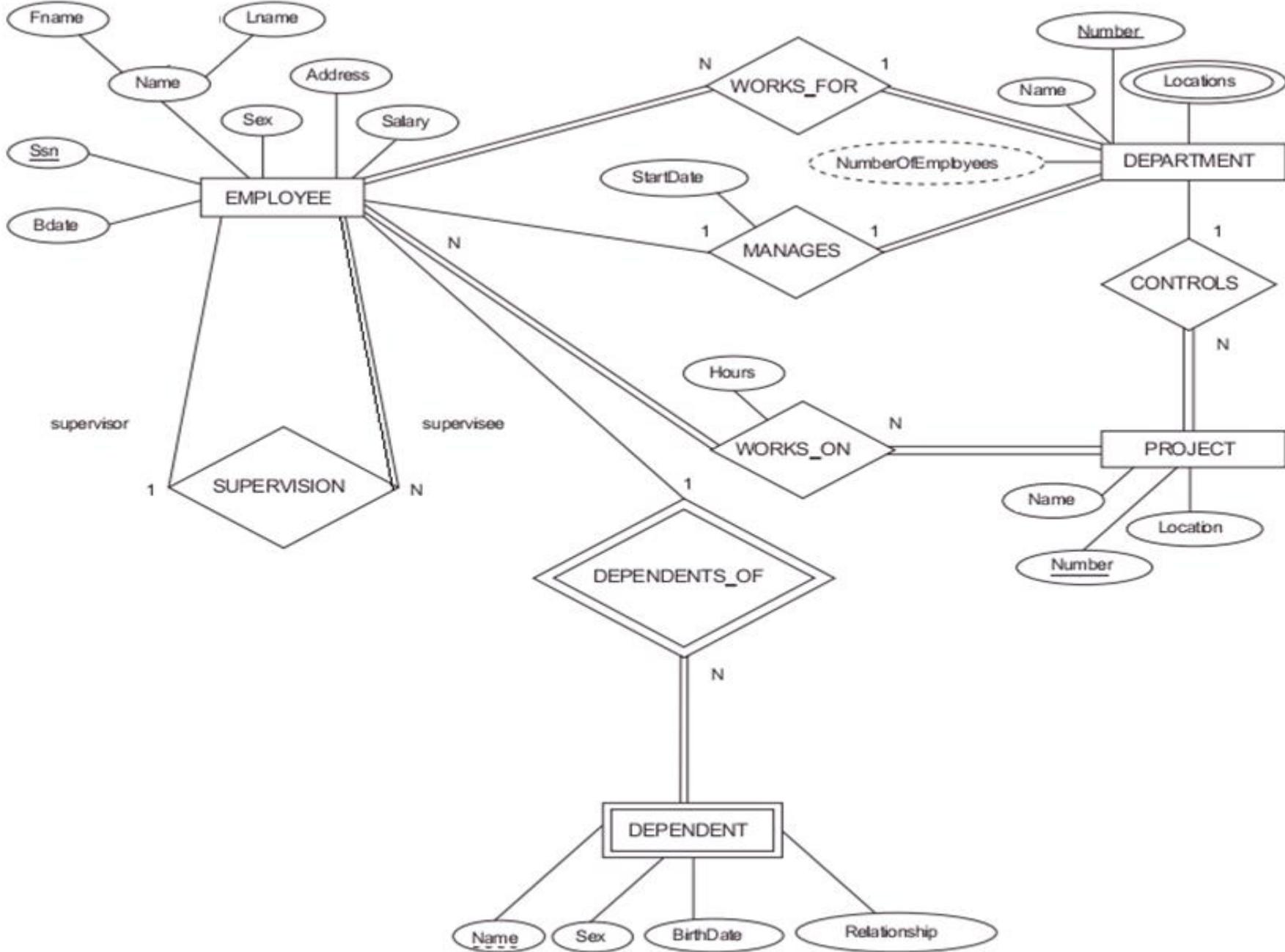
(a) EMPLOYEE entity with  
Manages relationship



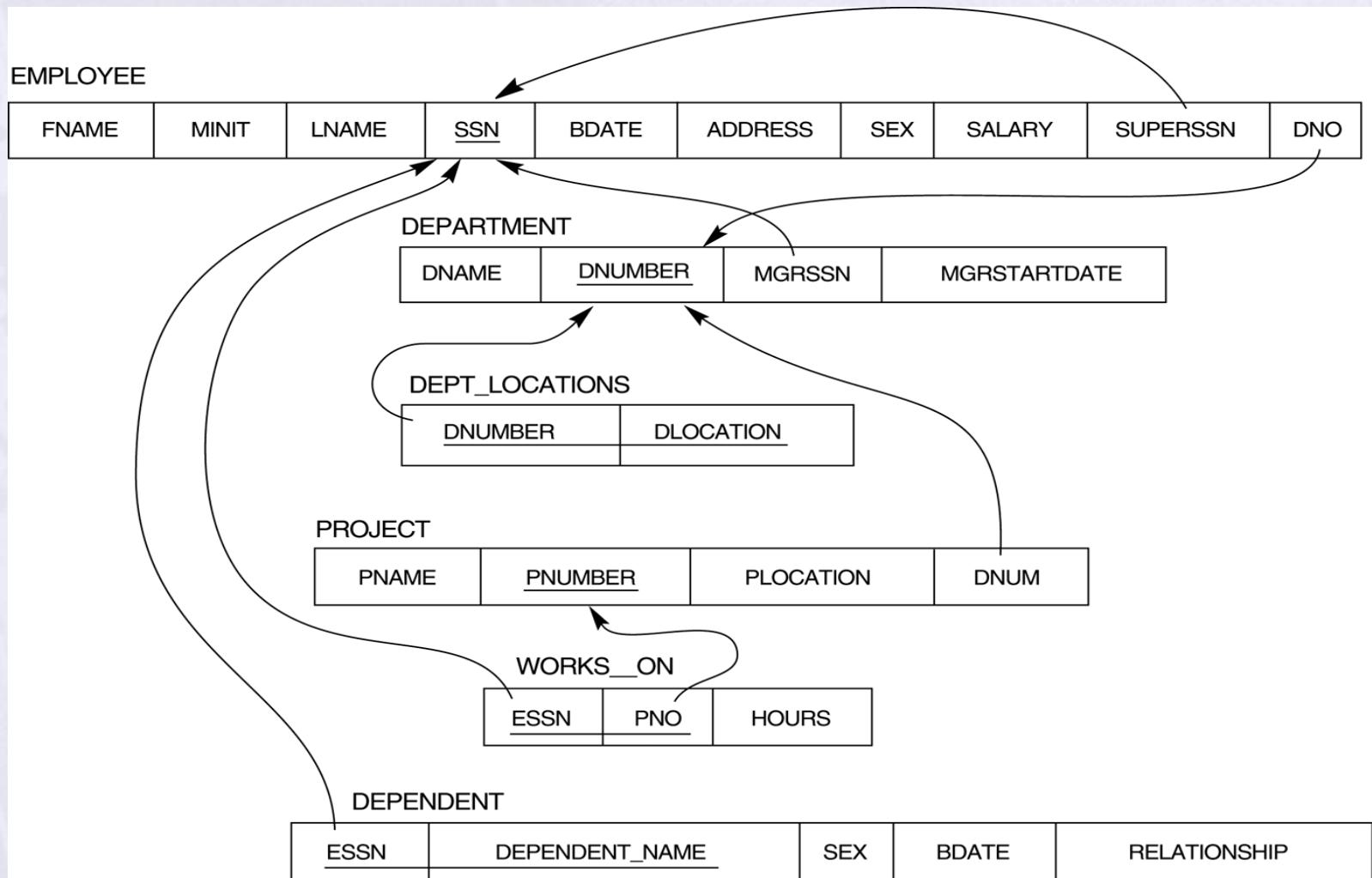
(b) EMPLOYEE  
relation with  
recursive foreign  
key

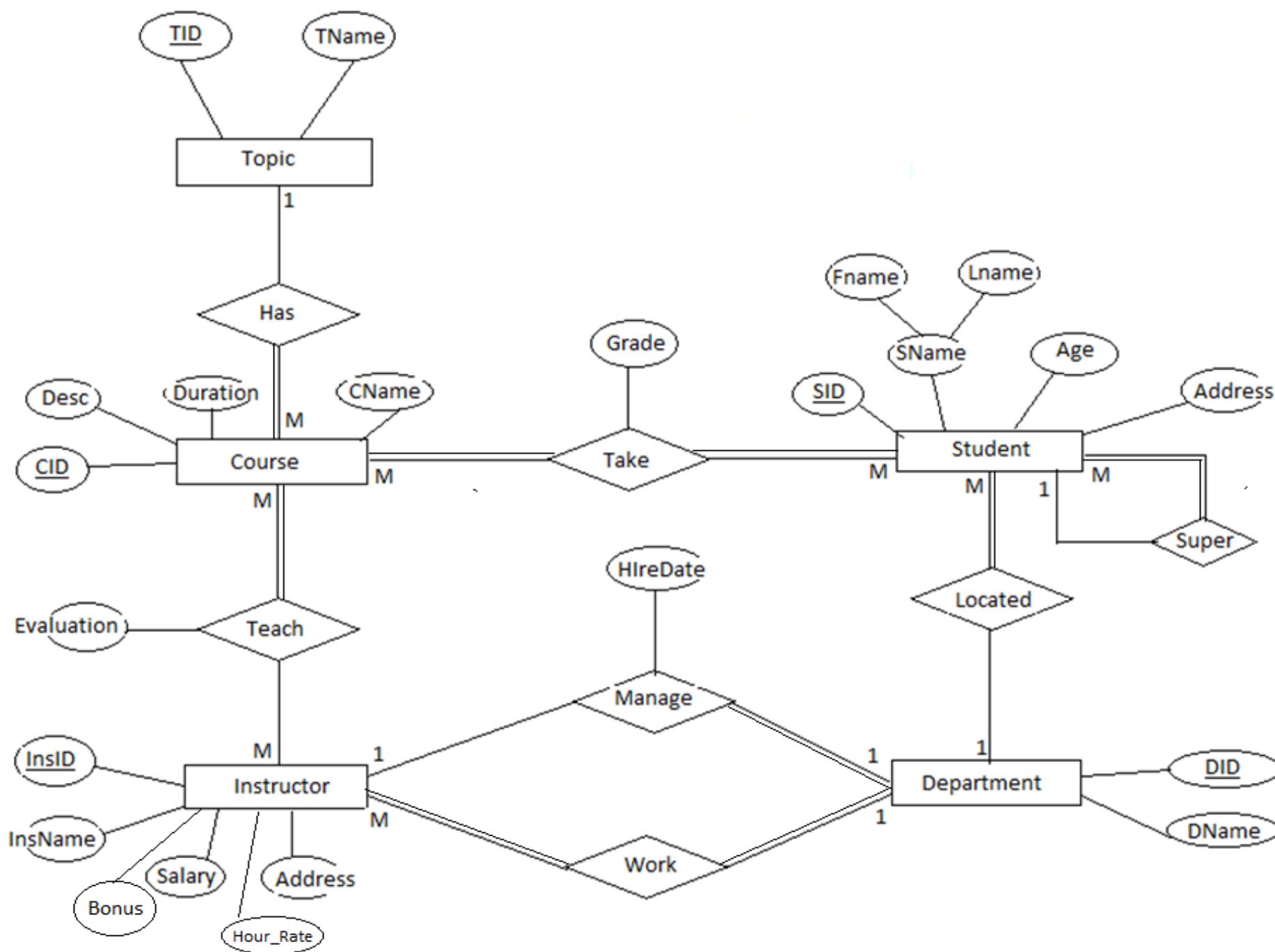
EMPLOYEE			
<u>Employee_ID</u>	Name	Birthdate	<u>Manager_ID</u>

# Case Study



# Mapping Result





# Mapping Result

- Student(St\_id,st\_fname,st\_Lname,st\_age,st\_super,Dept\_ID)
- Course(Crs\_id.Crs\_Name,Crs\_Duration,Top\_id)
- Topic(Top\_ID,Top\_Name)
- Stud\_Course(St\_ID,Crs\_ID,grade)
- Instructor(Ins\_ID,ins\_Name,Address,Salary,Dept\_ID)
- Ins\_Course(Ins\_ID,Crs\_ID,Evalution)
- Department(Dept\_ID,Dept\_Name,Manager\_ID,HireDate)

# Thank You !!!