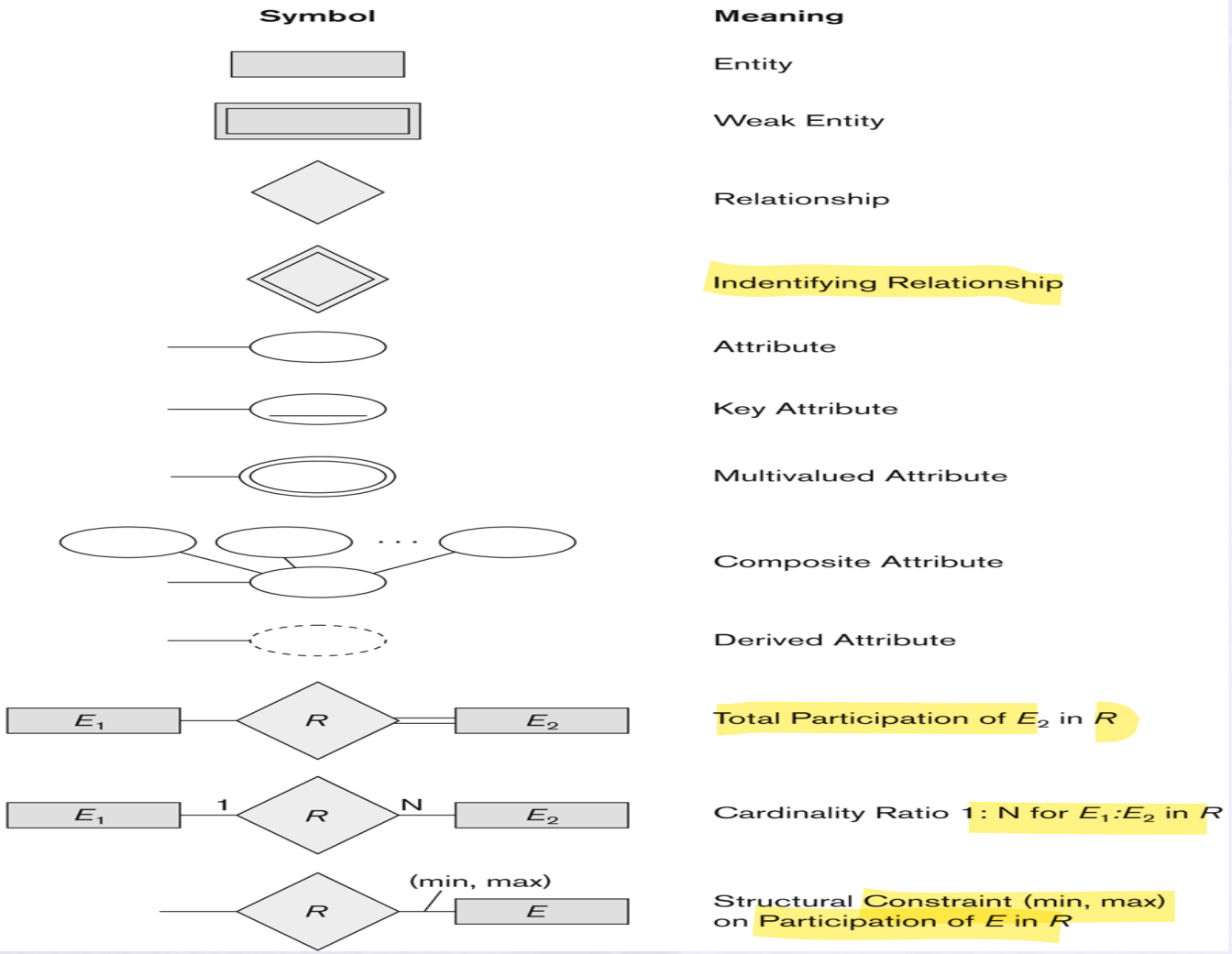
Mapping

Summary of notation for ER diagrams

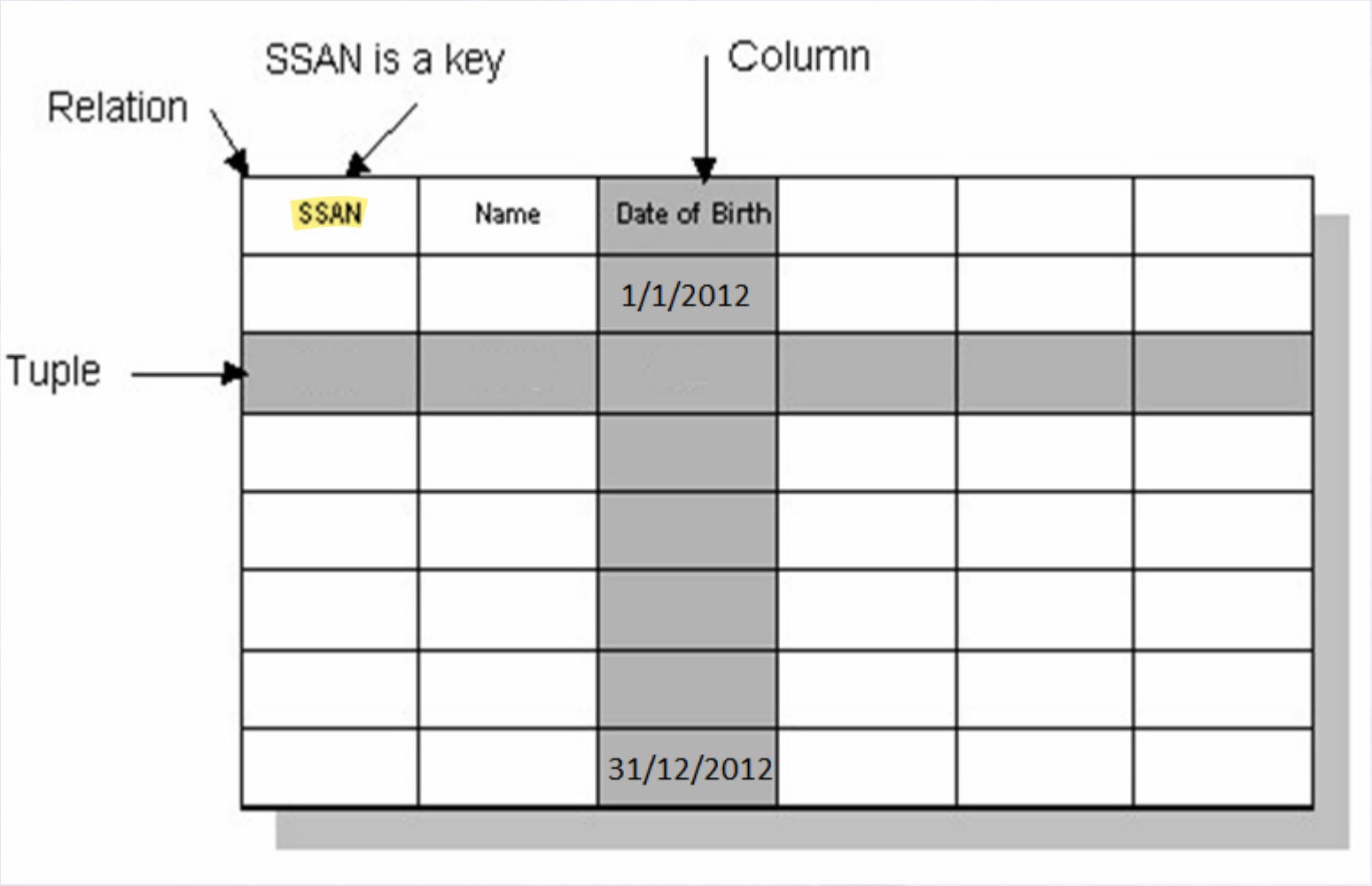
Figure 3.14
Summary of the notation for ER diagrams.



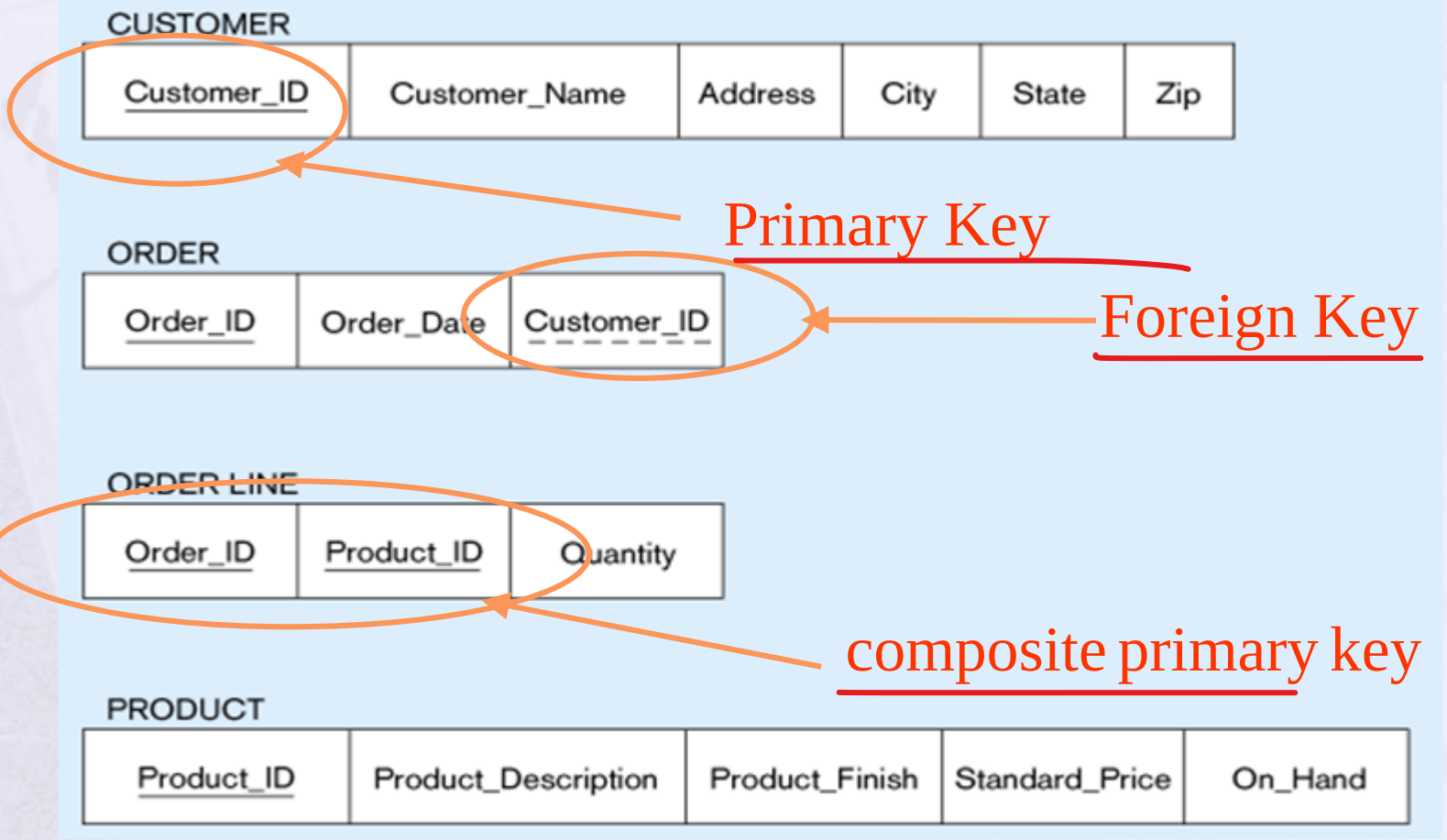
Relational Database Definitions

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

Relational Database



Mapping ->DB Tables



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.

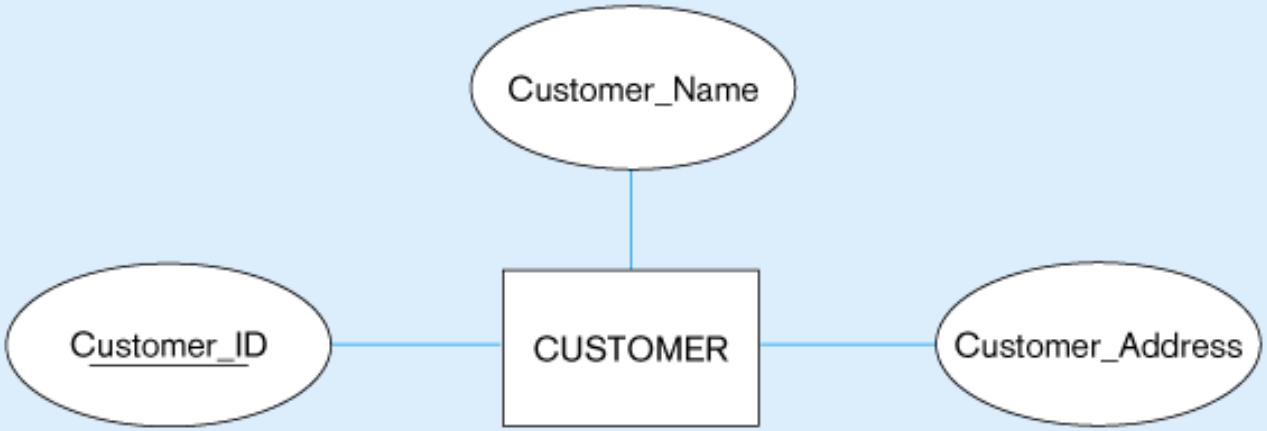
Step 7: Mapping of Unary Relationship.

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



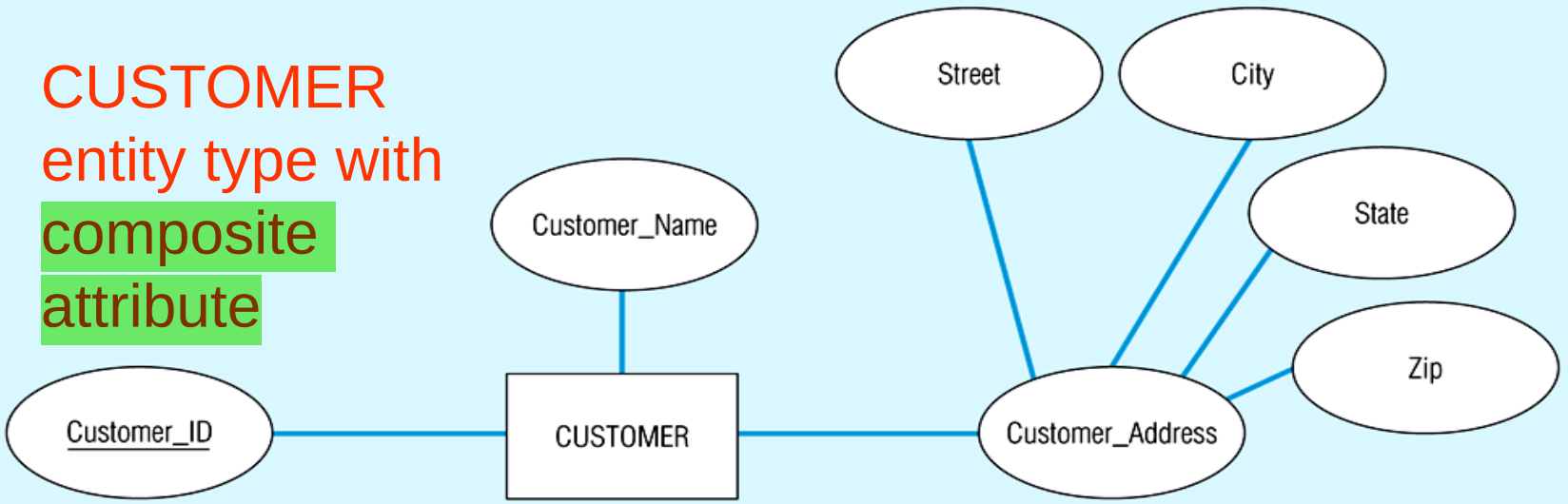
(b) CUSTOMER relation

CUSTOMER

Customer_ID	Customer_Name	Customer_Address

Mapping Composite attribute

CUSTOMER entity type with composite attribute

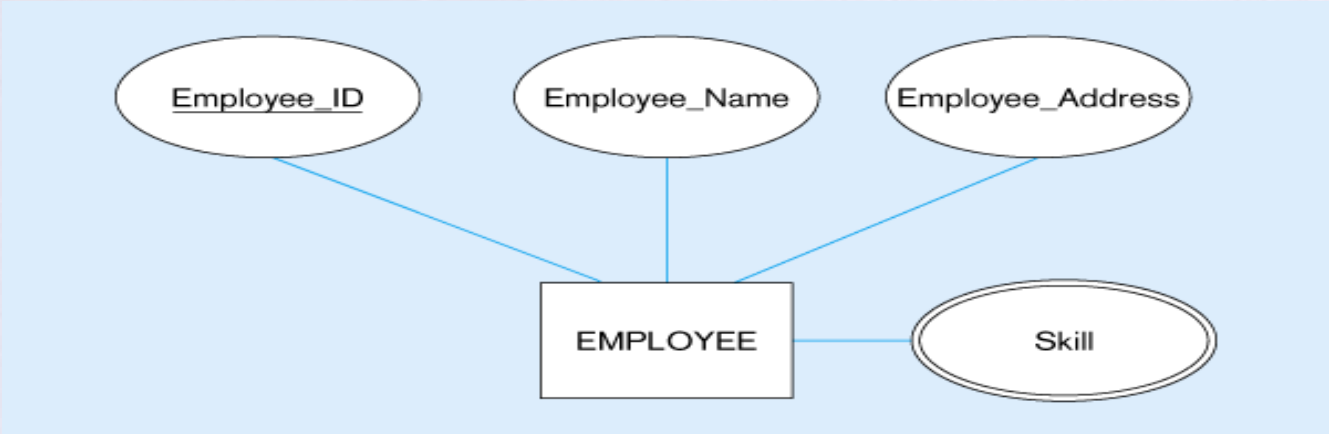


CUSTOMER relation with address detail

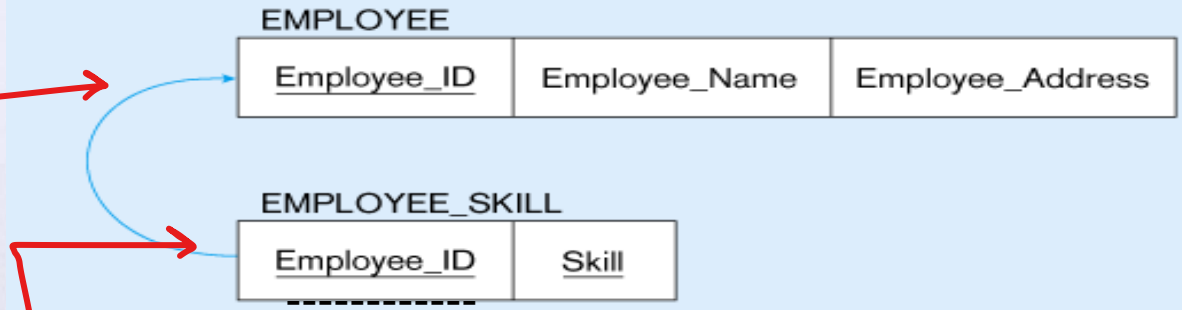
CUSTOMER

Customer_ID	Customer_Name	Street	City	State	Zip

Mapping Multivalued Attribute



Multivalued attribute becomes a separate relation with foreign key

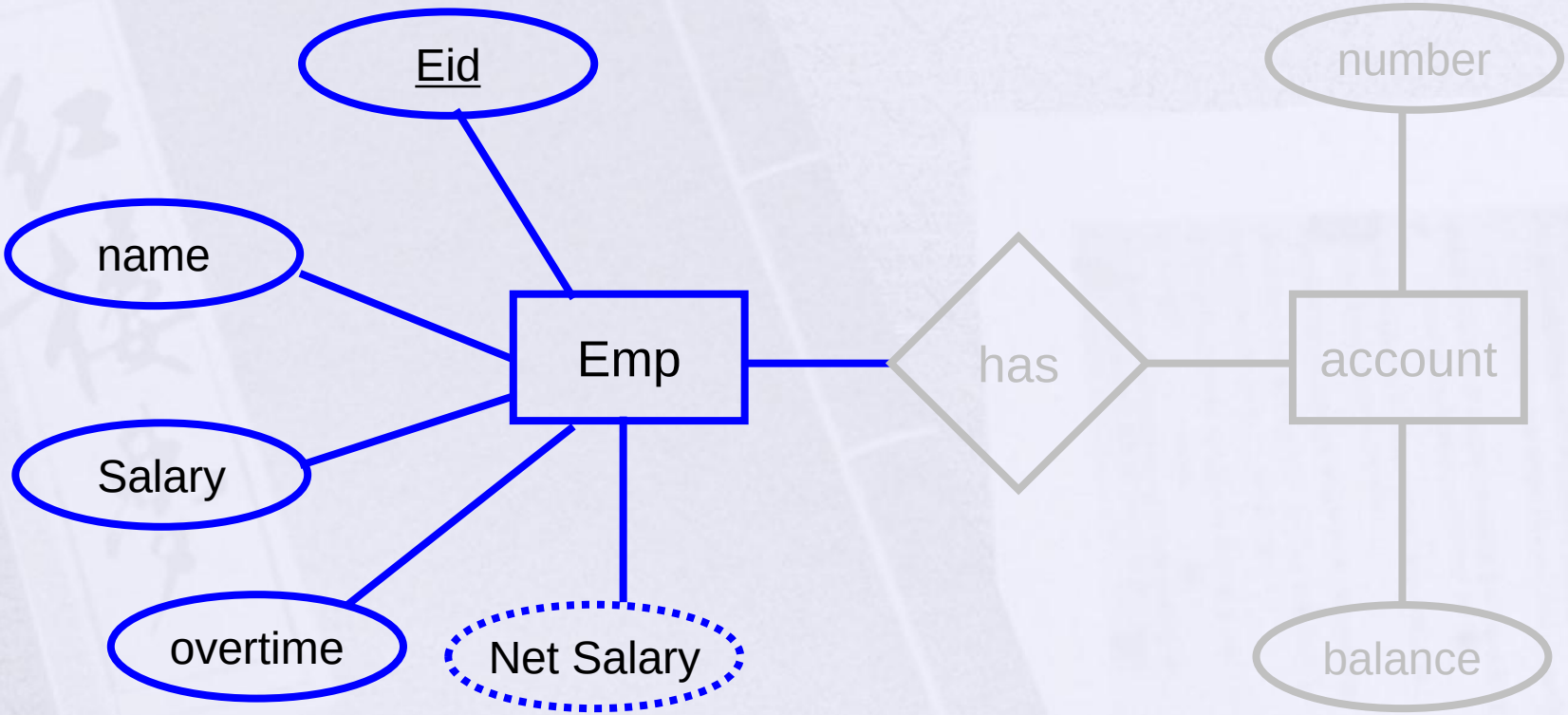


1 – to – many relationship between original entity and new relation

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

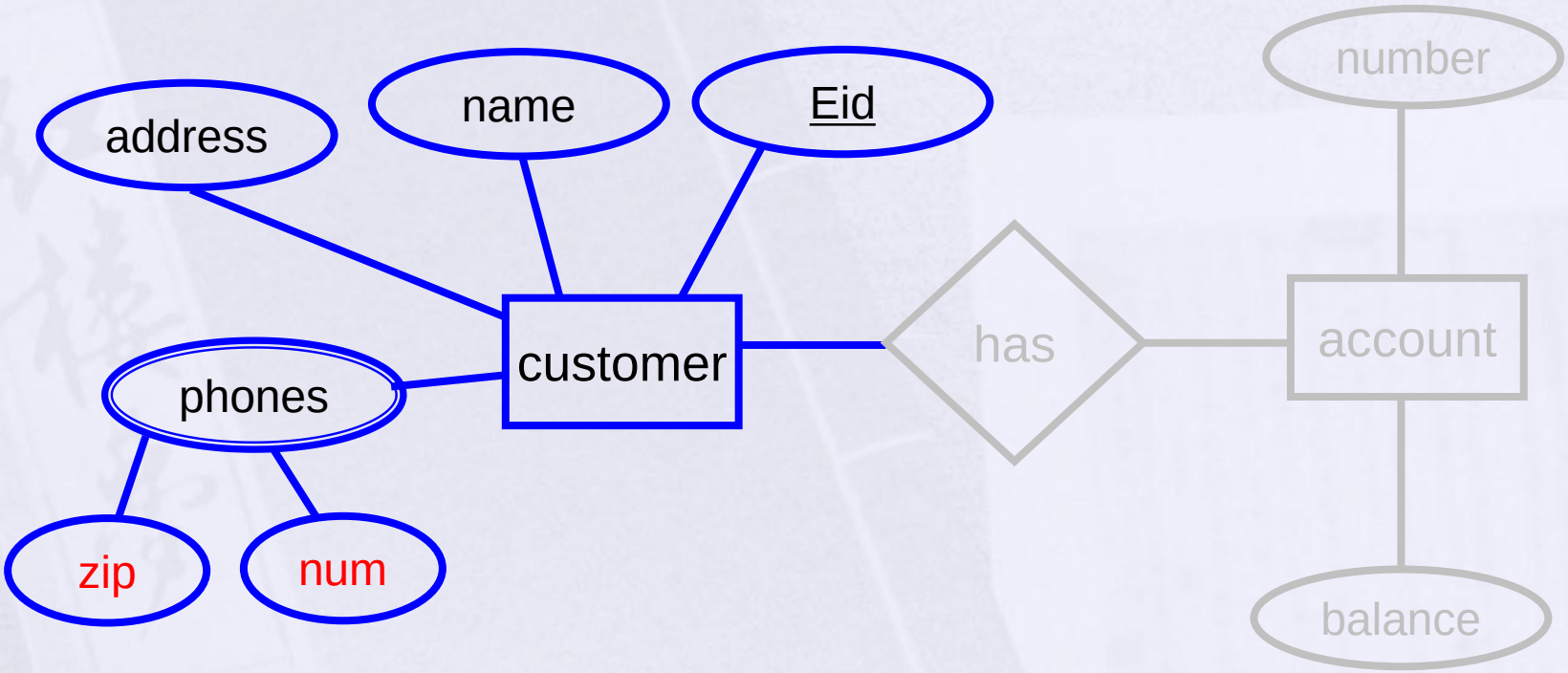


- Employee(Eid, Name, salary, overtime)

Note

Netsal= salary+overtime

Complex Attribute



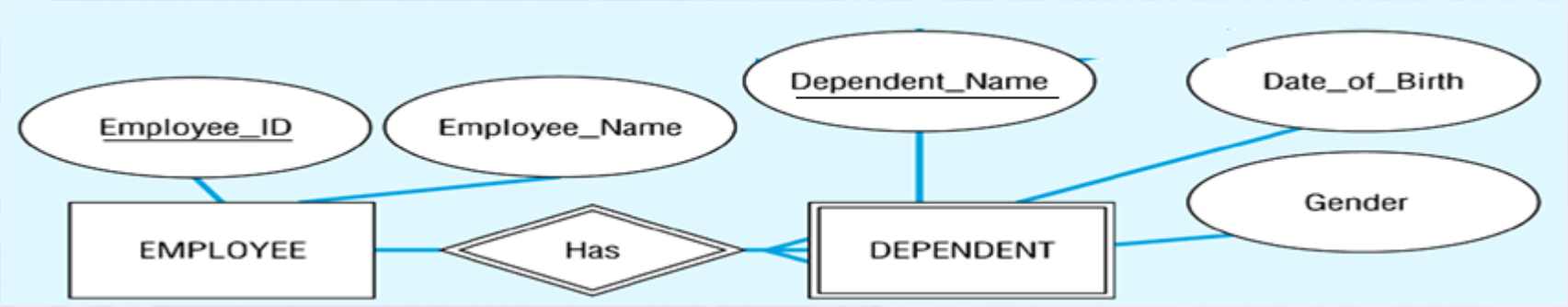
Customer(Eid, Name, address)

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



EMPLOYEE

Employee_ID	Employee_Name

partial key

DEPENDENT

Depend_Name	Employee_ID	Date_of_Birth	Gender

Composite primary key

Step 3: Mapping of Binary 1:1 Relation Types

1. Merged two tables if both sides are Mandatory.

2. Add FK into table with the total participation relationship to represent optional side.

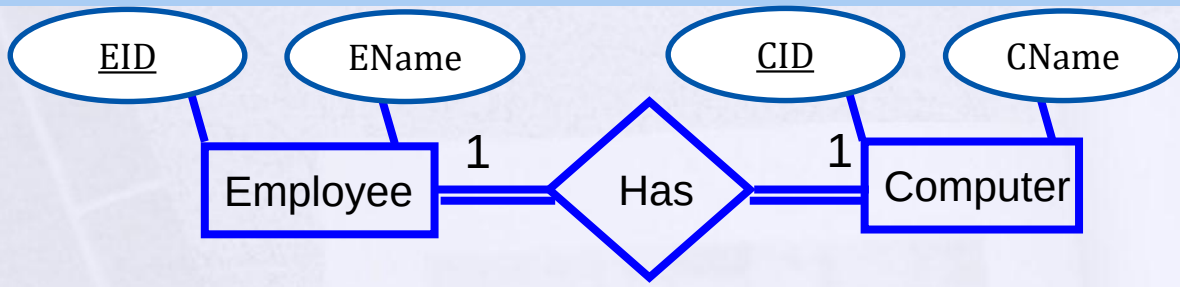
total : partial

3. Create third table if both sides are optional.

2 Mandatory

One-to-One

2 Mandatory



1 table

tbl_xy (PK,….,…,….)

PK = PKx or PKy

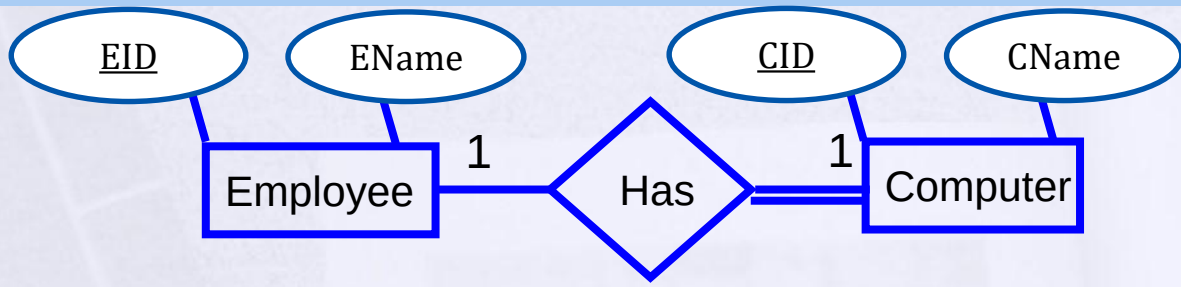
any pk of two entities can be a pk

Emp(EID, Ename, Cname, CID)

Optional-Mandatory

One-to-One

X optional – Y mandatory



2 tables

tbl_x (PKx,….,…….)

tbl_y (PKy,….,….,PKx….)

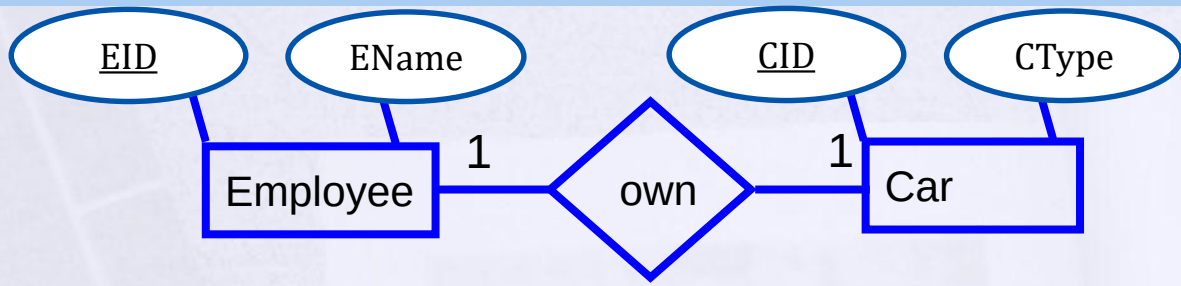
Employee(EID, Ename)

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

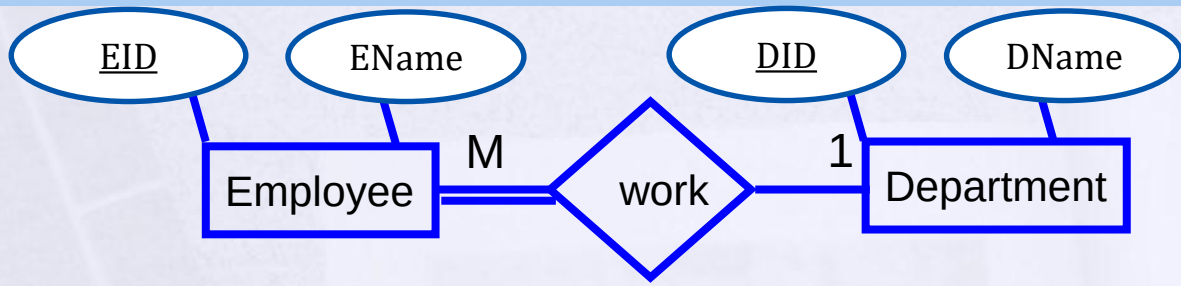
else you can create anther table to represent relation

focus only on m sic

Many is Mandatory

One-to-Many

X whatever– Y mandatory



2 tables

tbl_x (PKx,….,…….)

tbl_y (PKy,….,….,FKy….)

FKy= PKx

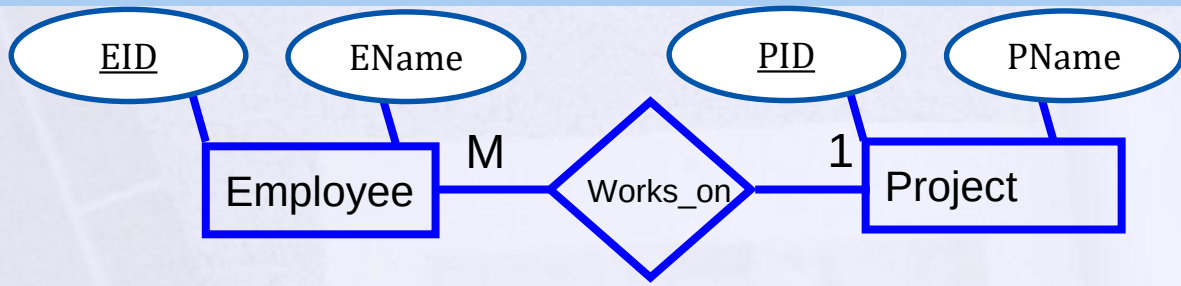
Department(DID, Dname)

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)

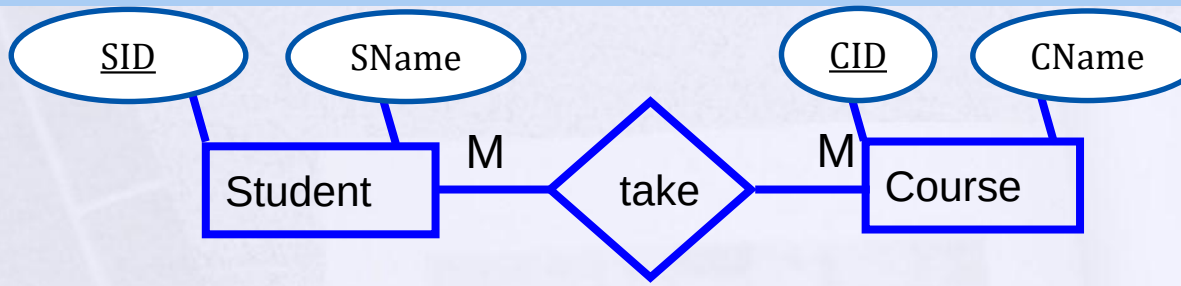
Step 5: Mapping of Binary M:N Relationship Types.

- 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

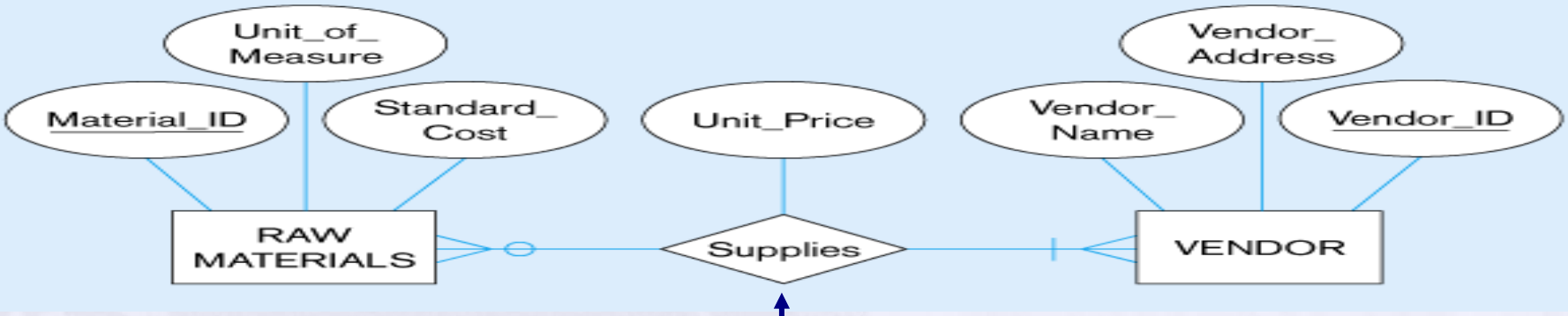
Student(SID, Sname)

Course(CID, Cname)

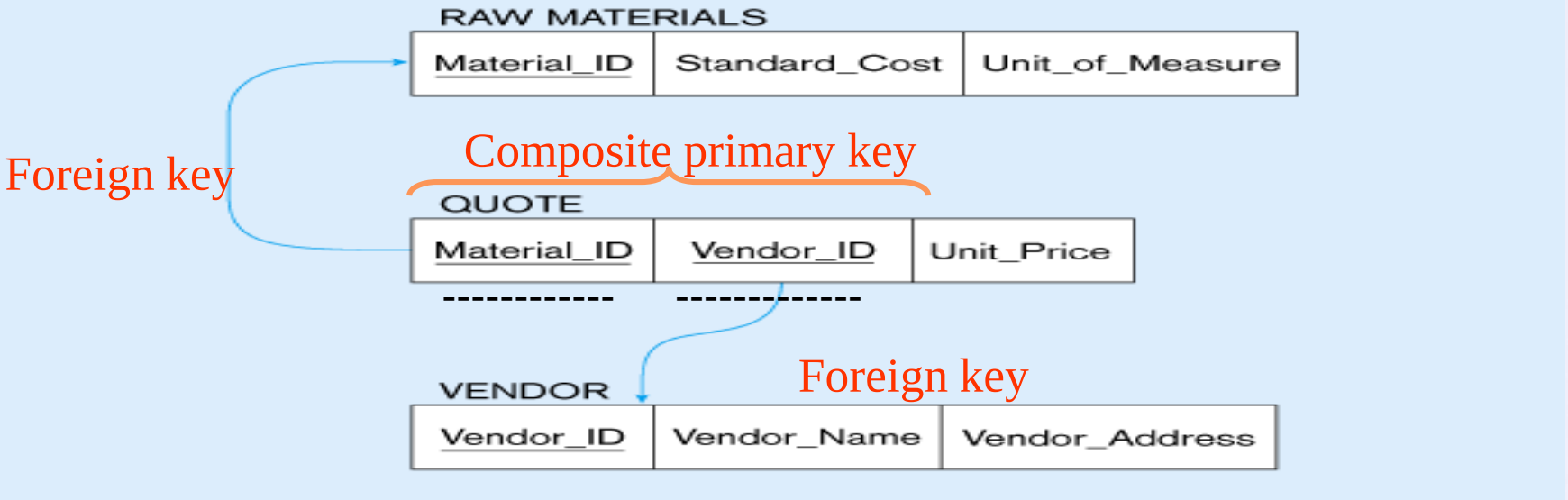
Stud_Course(SID, CID)

----- -----

M:N with attribute



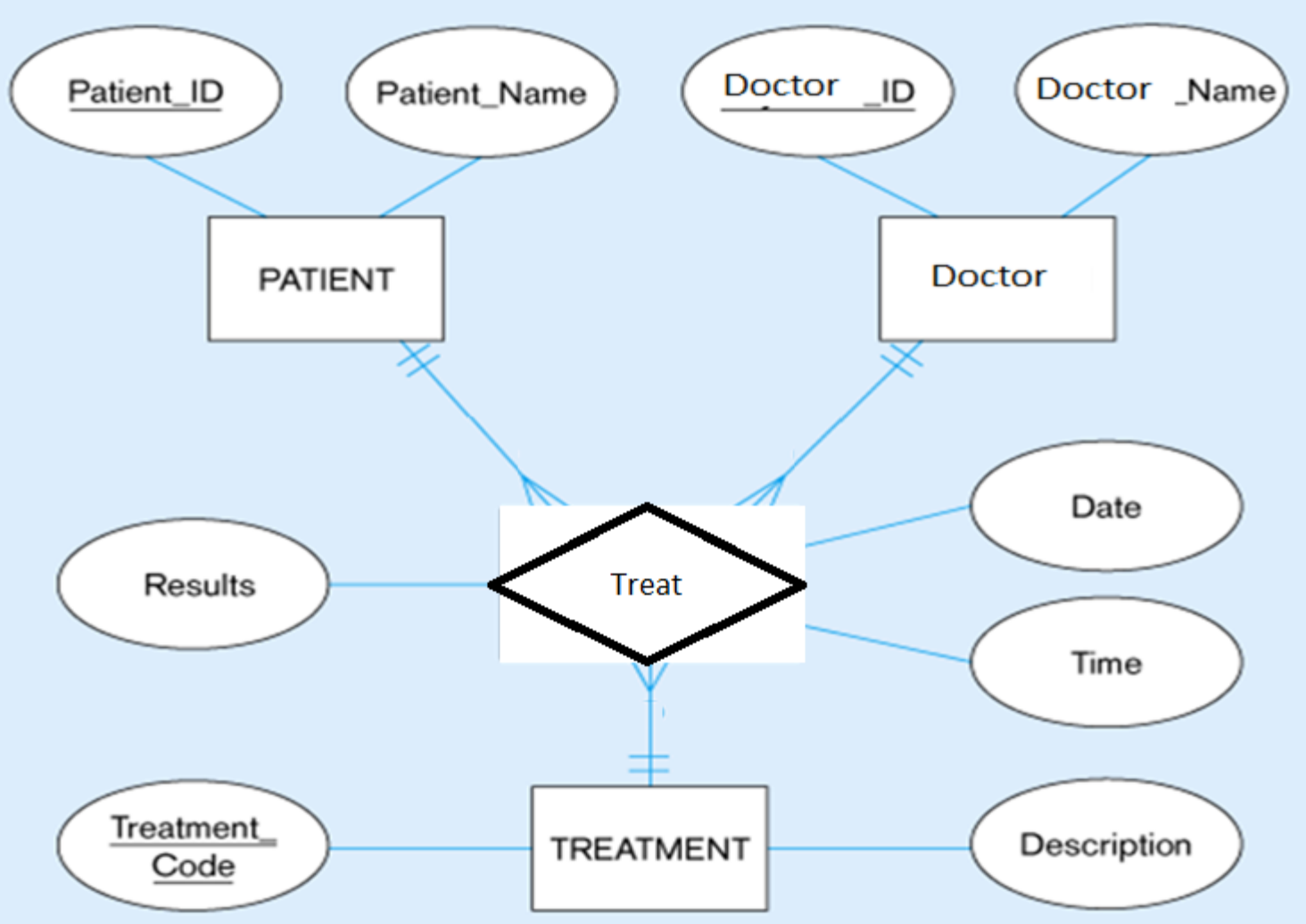
The *Supplies* relationship will need to become a separate relation



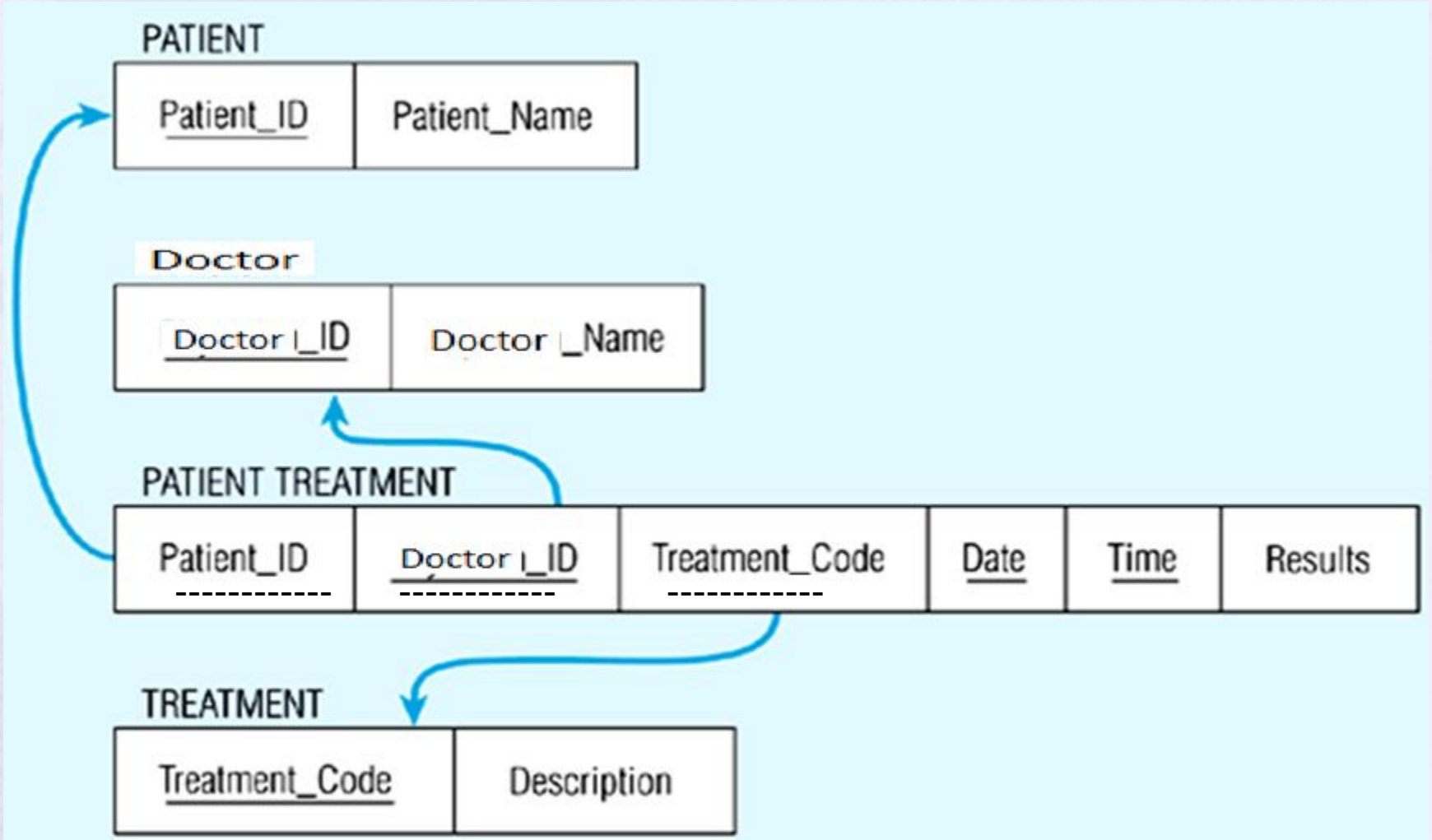
Step 6: Mapping of N-ary Relationship Types.

- 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.

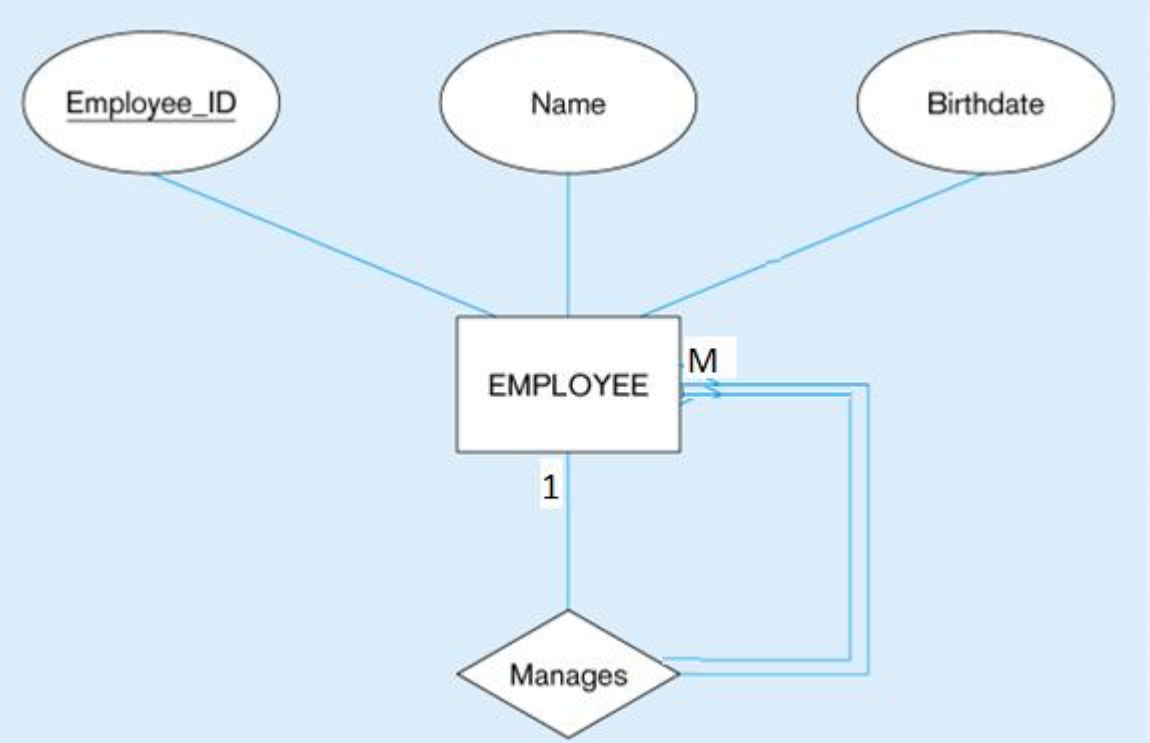


Step 6: Mapping of N-ary Relationship Types.

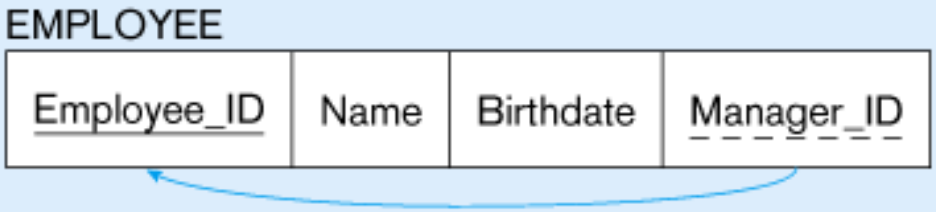


Step 7:Mapping Unary Relationship

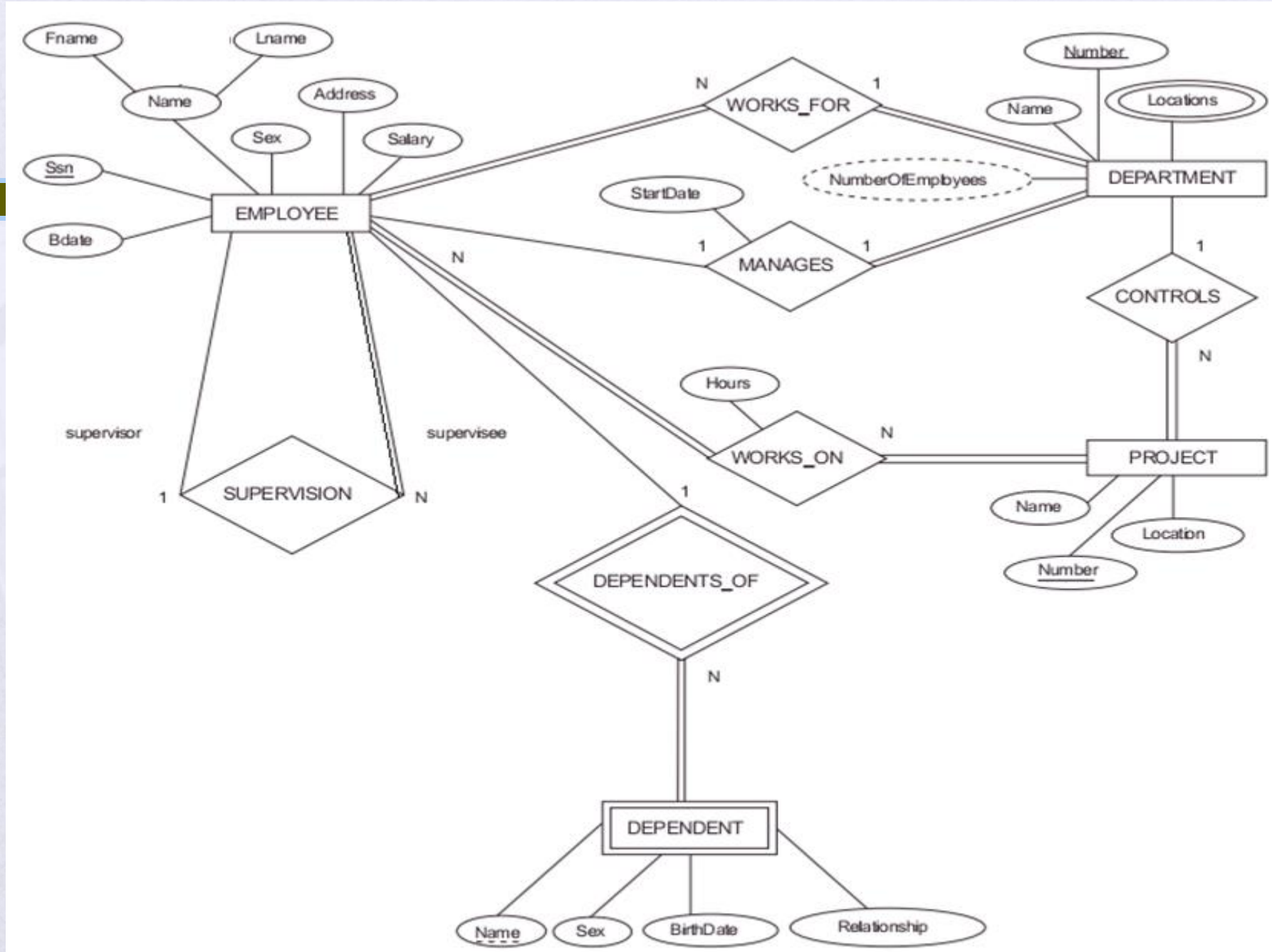
(a) EMPLOYEE entity with Manages relationship



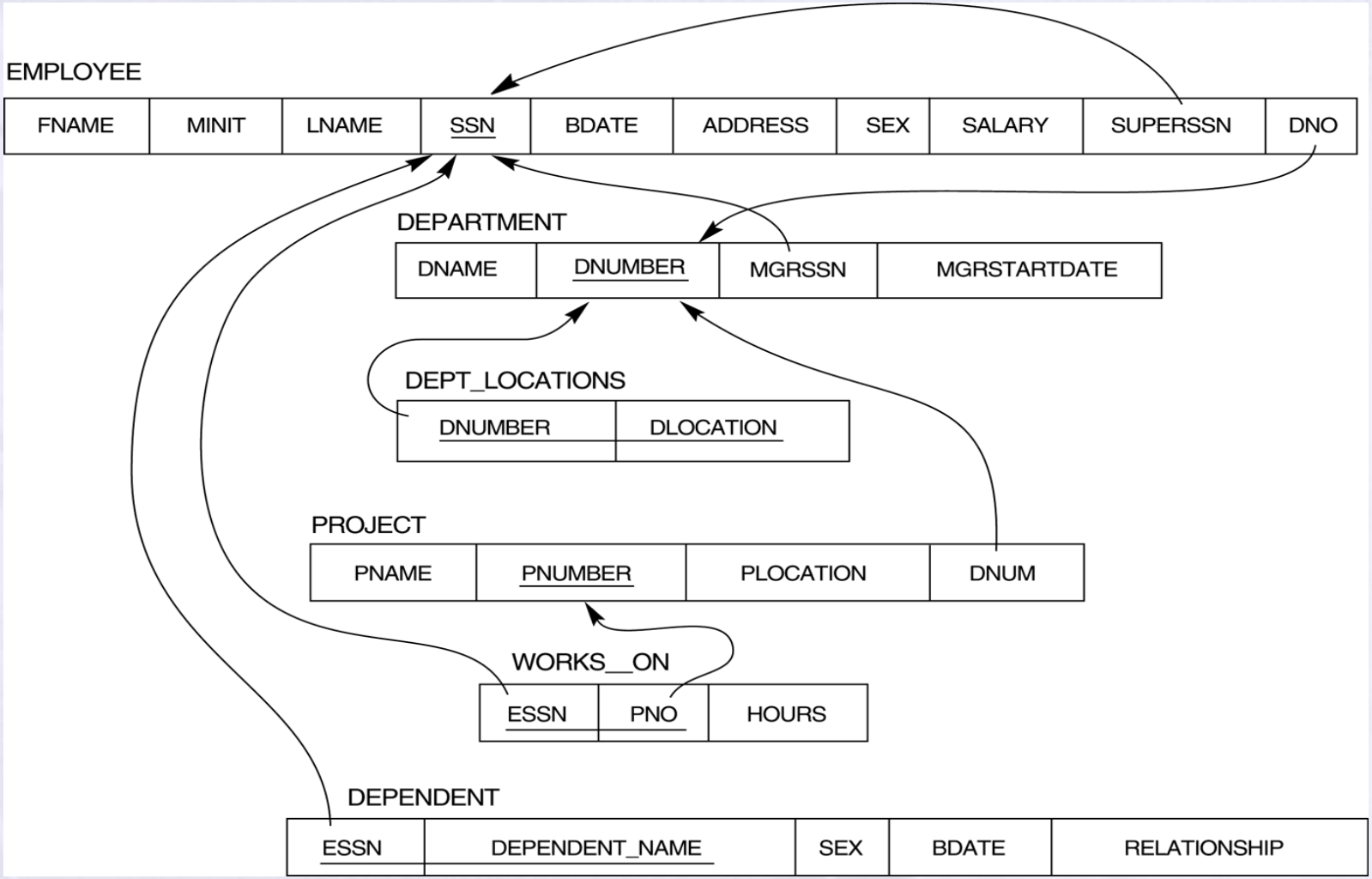
(b) EMPLOYEE relation with recursive foreign key

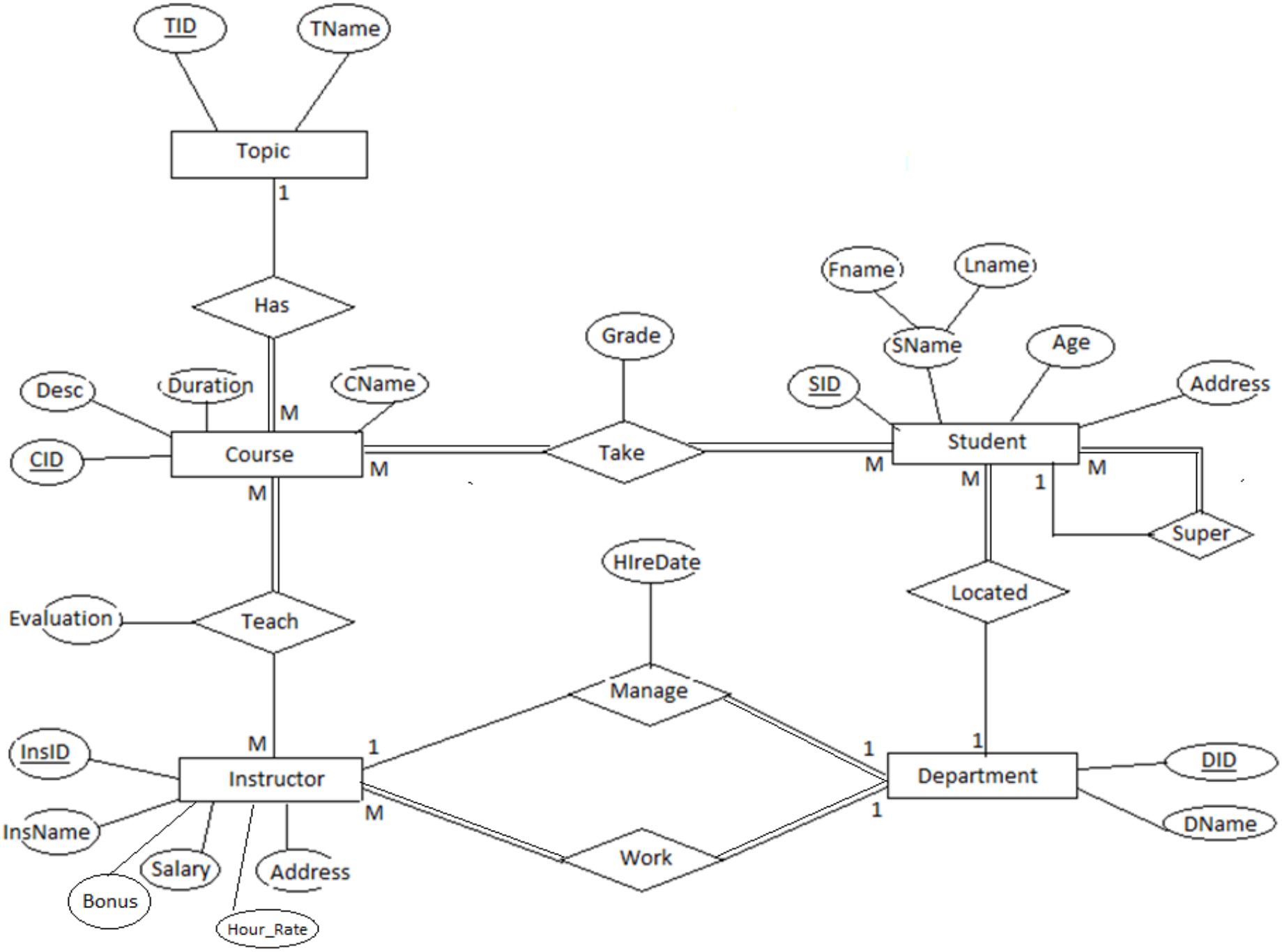


Case Study


Fname
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
N
1
CONTROLS
N
Hours
N
WORKS_ON
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Mapping Result




TID
TName
Topic
1
Has
M
Desc
Duration
CName
CID
Course
M
Grade
Take
M
Fname
Lname
SName
Age
SID
Address
Student
M
1
M
Super
HIreDate
Located
Evaluation
Teach
M
Manage
1
1
InsID
Instructor
M
1
Department
DID
DName
InsName
Salary
Address
Bonus
Hour_Rate
Work

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 !!!