

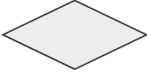







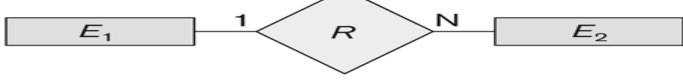
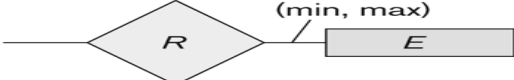


Mapping



Summary of notation for ER diagrams

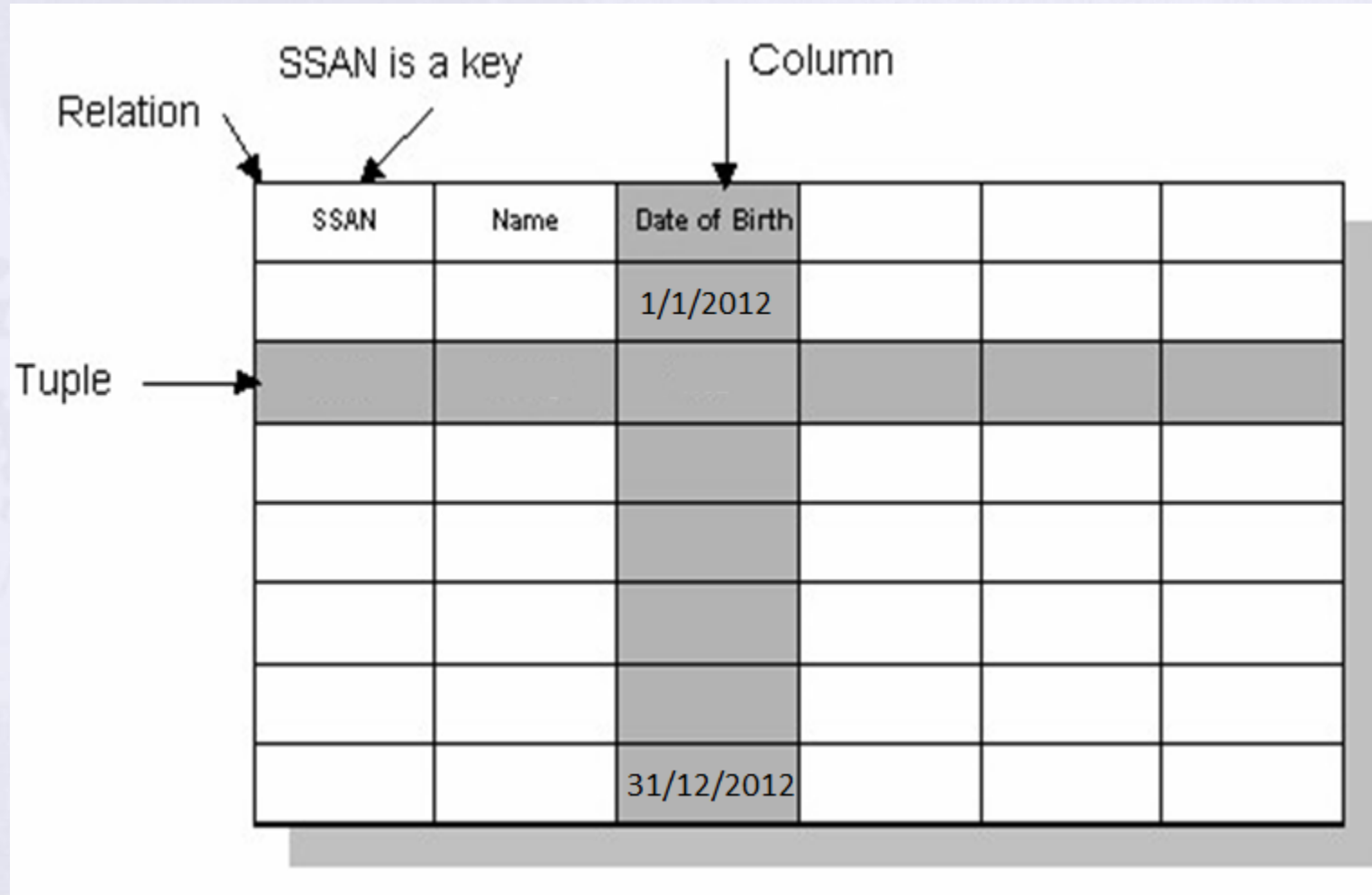
Figure 3.14
Summary of the
notation for ER
diagrams.

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



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.

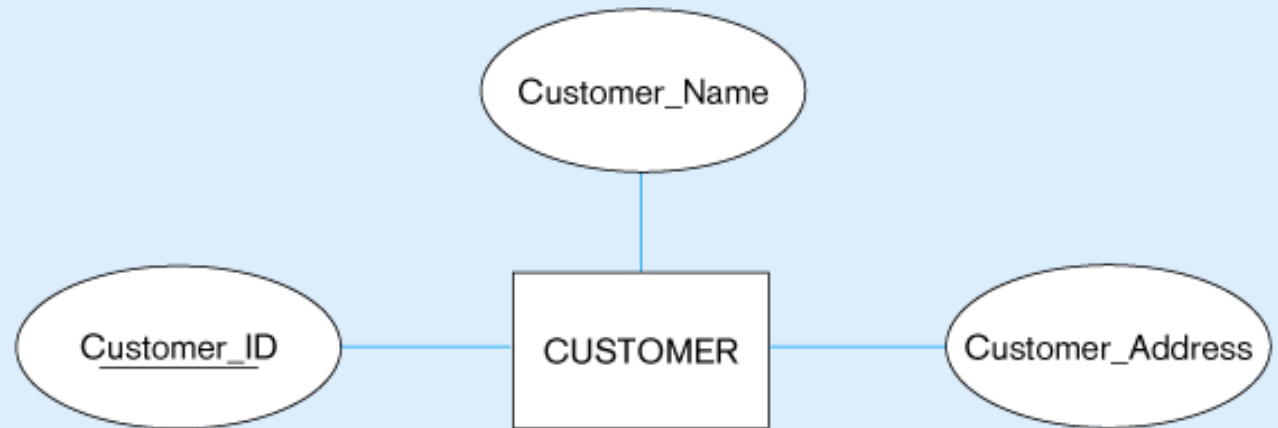
Step 7: Mapping of Unary Relationship.

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

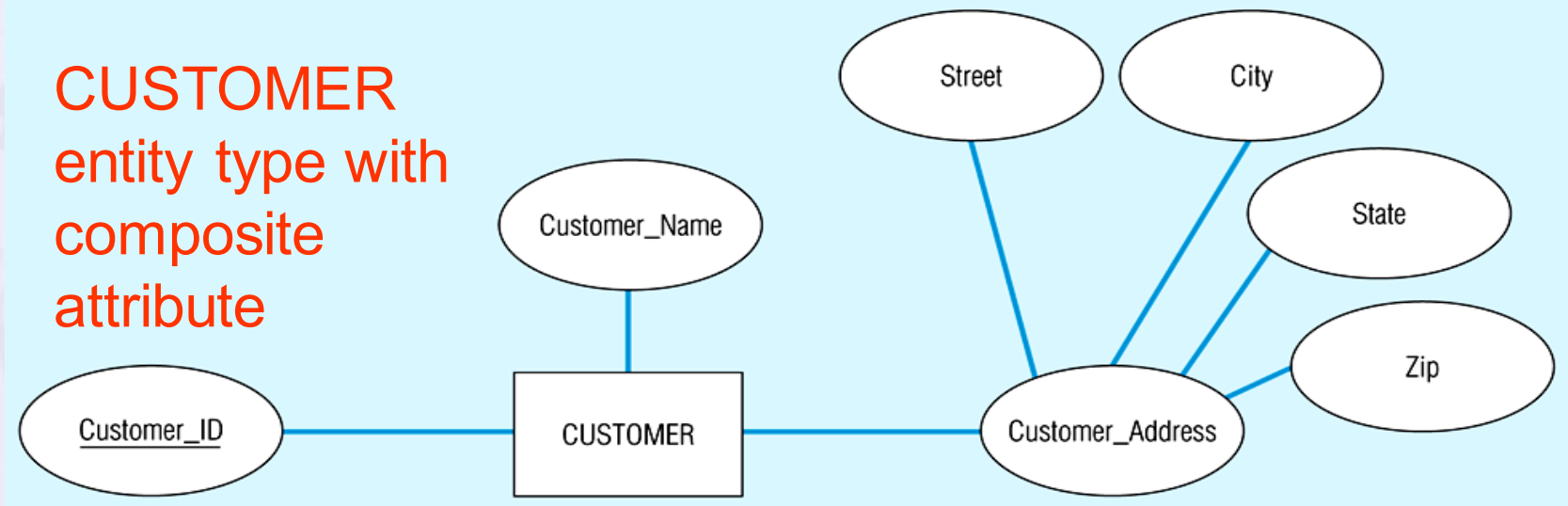
(a) CUSTOMER entity type with simple attributes



(b) CUSTOMER relation

CUSTOMER		
<u>Customer_ID</u>	Customer_Name	Customer_Address

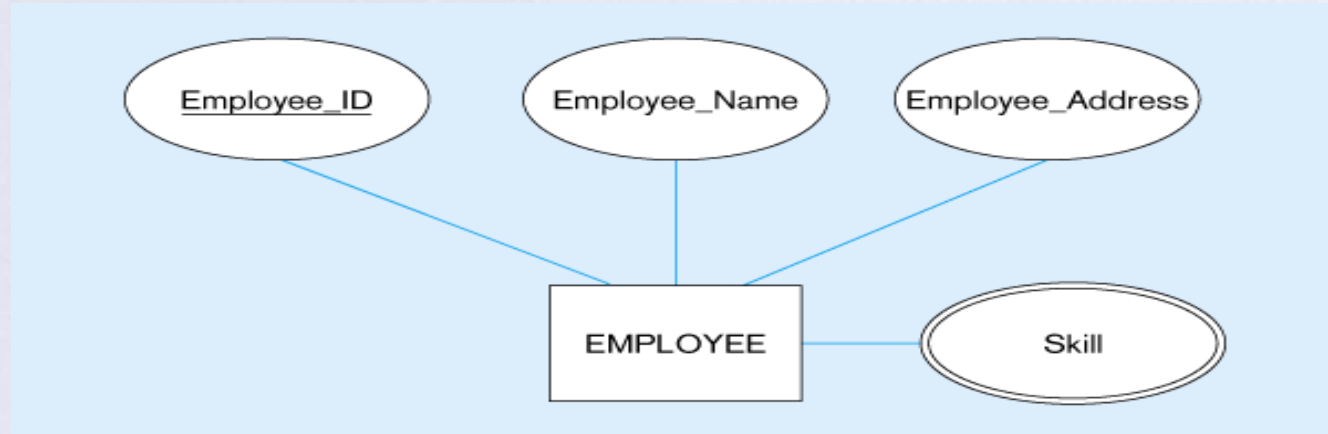
Mapping Composite attribute



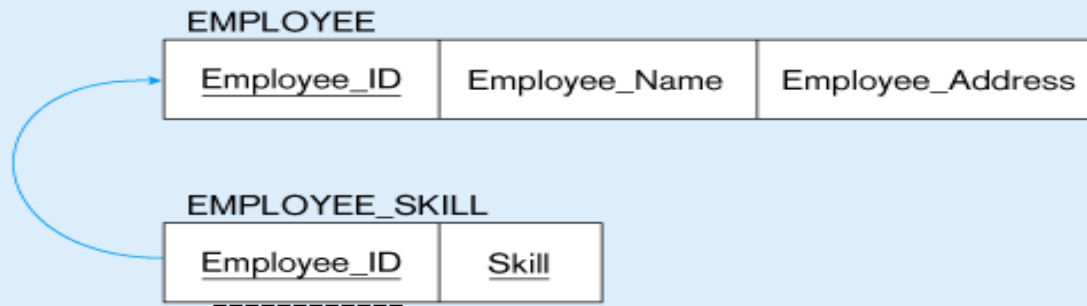
CUSTOMER relation with address detail

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

Mapping Multivalued Attribute



Multivalued attribute becomes a separate relation with foreign key

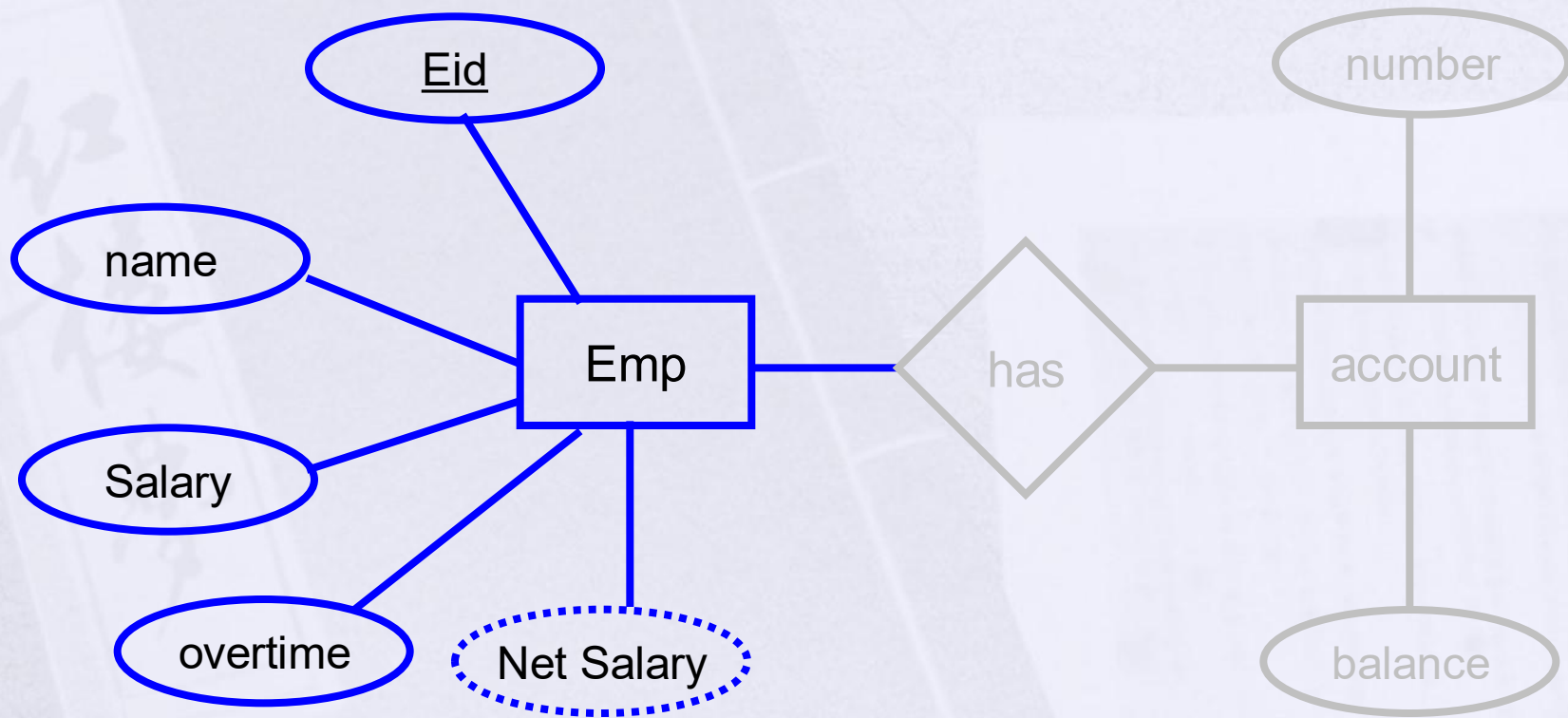


1 – to – many relationship between original entity and new relation

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

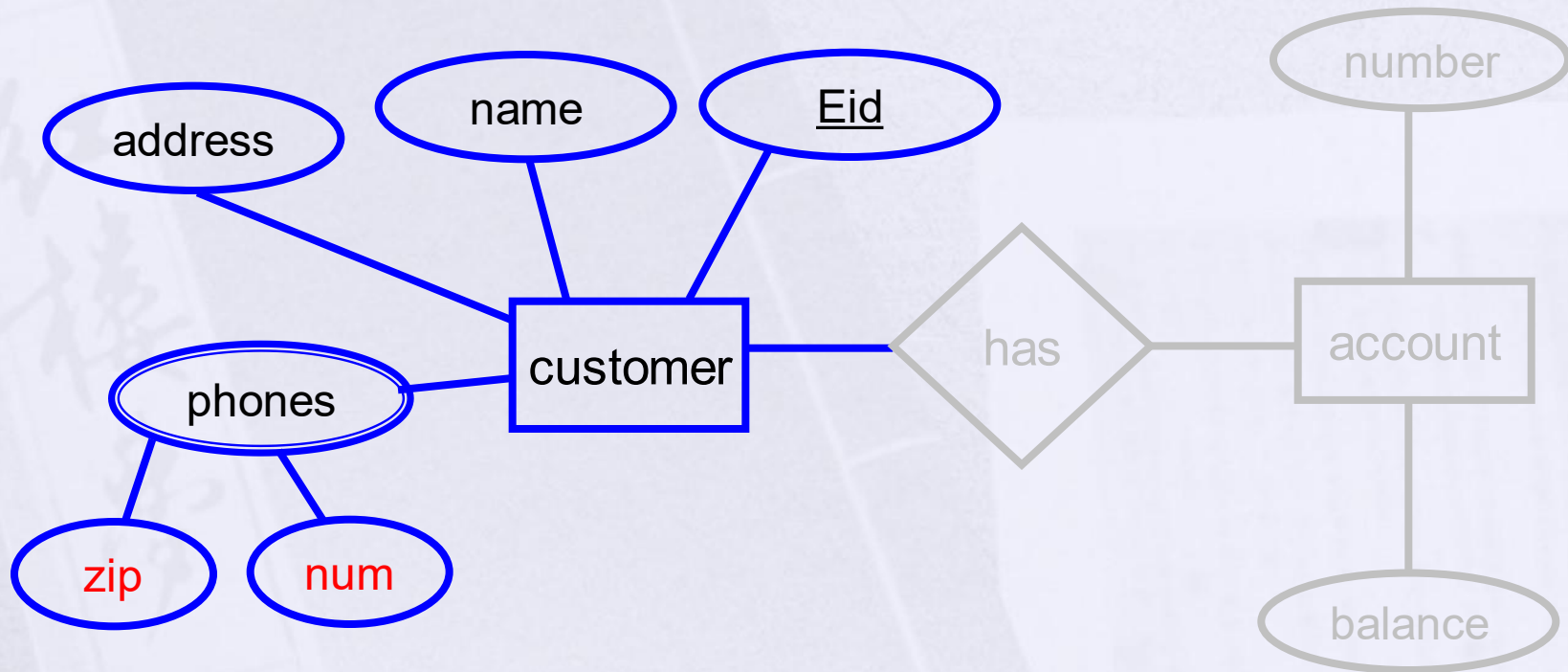


➤ Employee(Eid, Name, salary, overtime)

Note

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

Complex Attribute



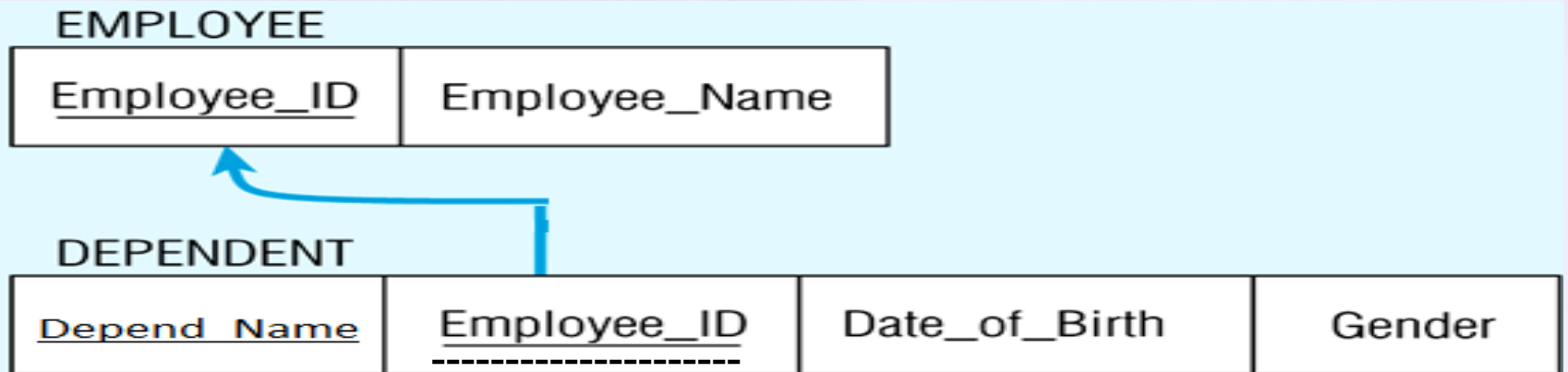
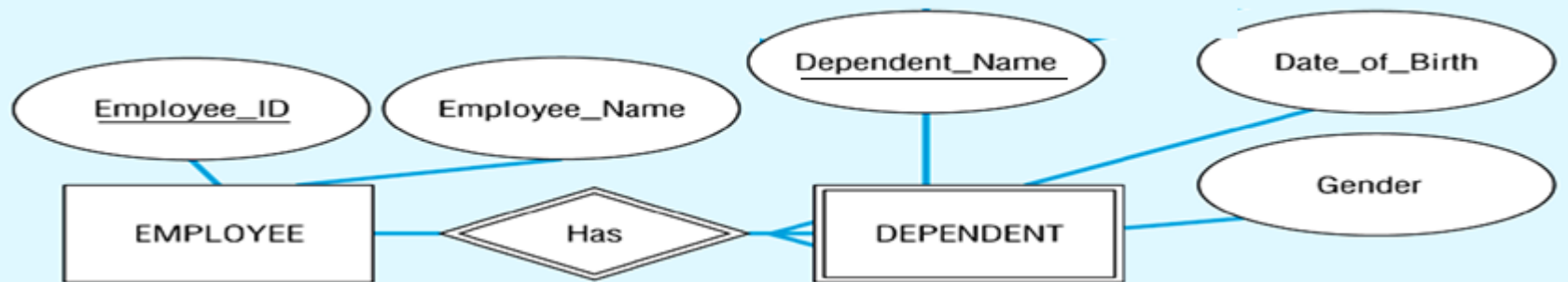
Customer(Eid, Name, address)

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



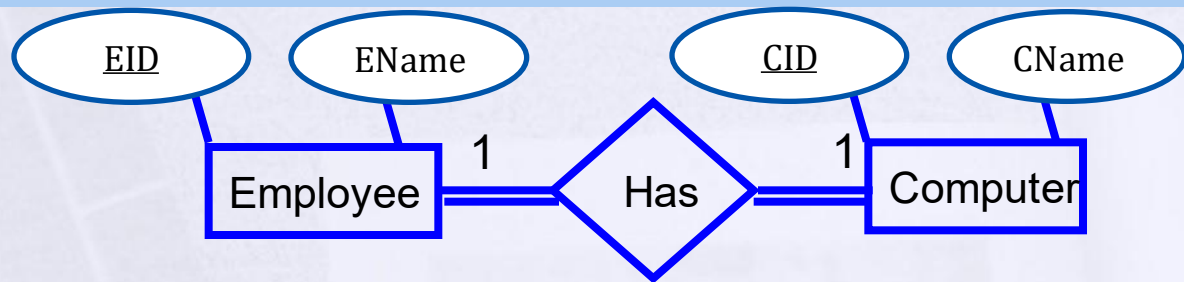
Composite primary key

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 = PKx or PKy

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

Optional-Mandatory

One-to-One

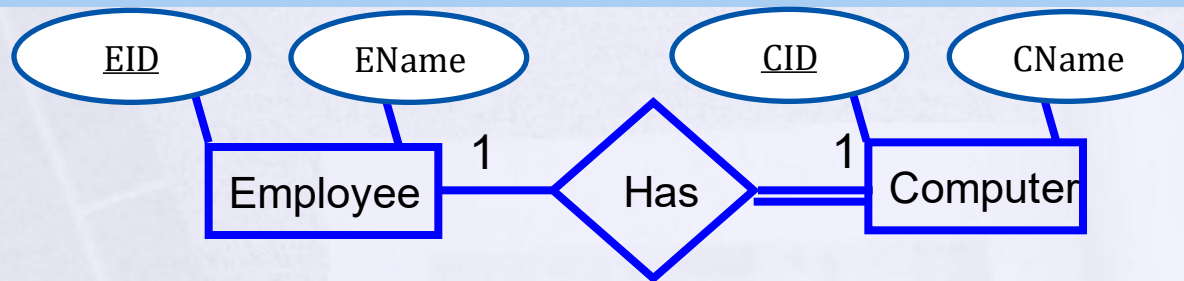
X optional – Y mandatory



2 tables

tbl_x (PK_x, ..., ..)

tbl_y (PK_y, ..., .., PK_x....)



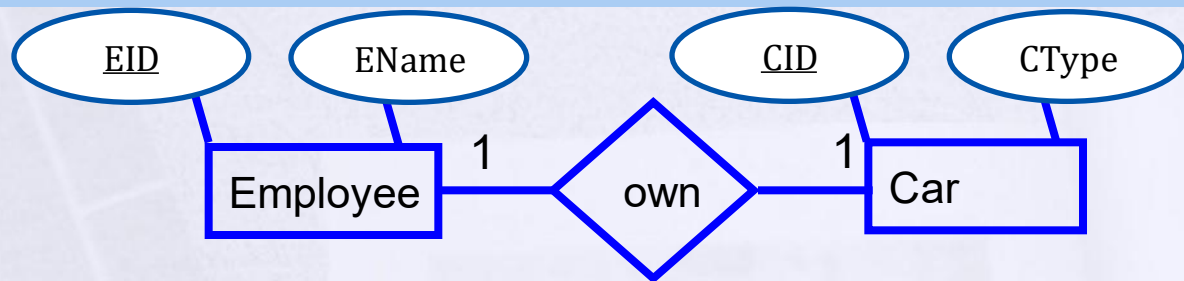
Employee(EID, Ename)

Computer(CID, Cname, **EID_FK**)

2 Optional

One-to-One

2 Optional



3 tables

tbl_x (PK_x,...,.....)

tbl_y (PK_y,...,.....)

tbl_xy (PK_{xy},...,...,FK_{xy},....)

PK_{xy} = PK_x or PK_y

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

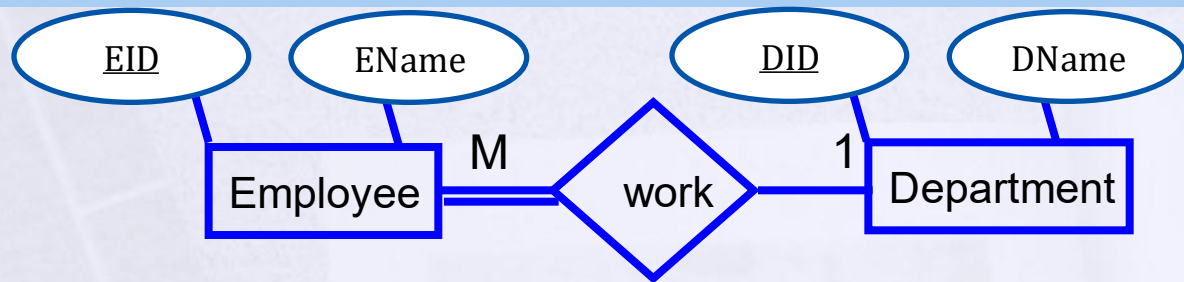
One-to-Many

X whatever – Y mandatory



2 tables

tbl_x (PK_x, ..., ..)
tbl_y (PK_y, ..., .., FK_y....)
FK_y = PK_x



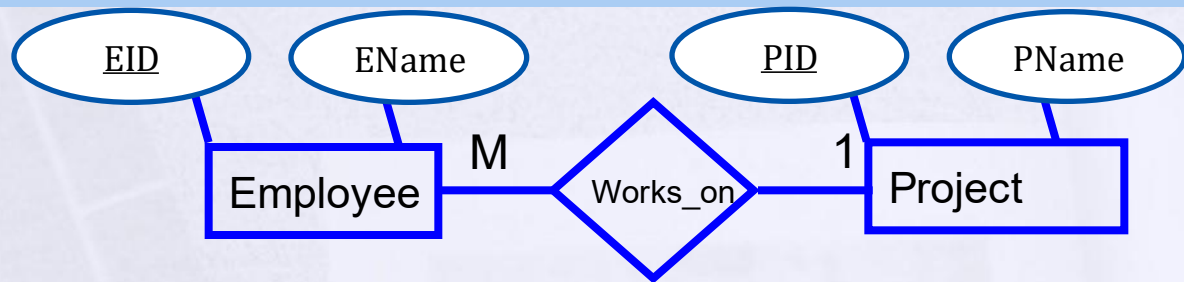
Department(DID, Dname)

Employee(EID, Ename, **DID**)

Many is Optional

One-to-Many

X whatever- Y Optional



3 tables

tbl_x (PK_x,...,.....)tbl_y (PKy,...,.....)tbl_xy (PKxy,...,.....)
$$PK_{xy} = PK_y$$
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



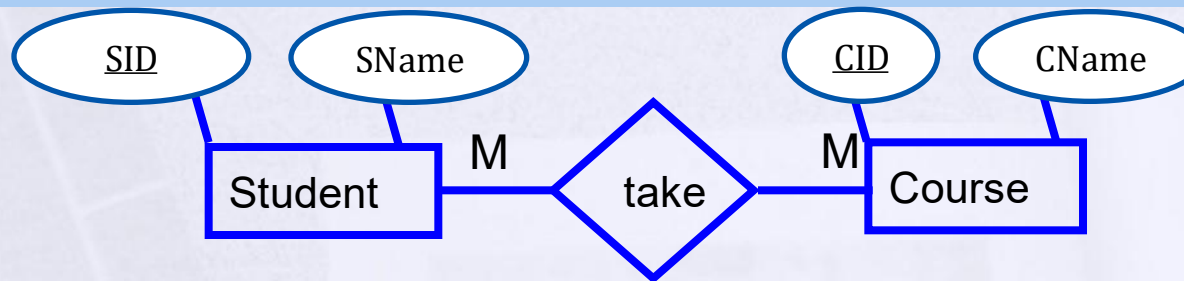
3 tables

tbl_x (PK_x, ...,

tbl_y (PK_y, ...,

tbl_xy (PK_x, PK_y, ...,

PK_{xy} = PK_x + PK_y



Student(SID, Sname)

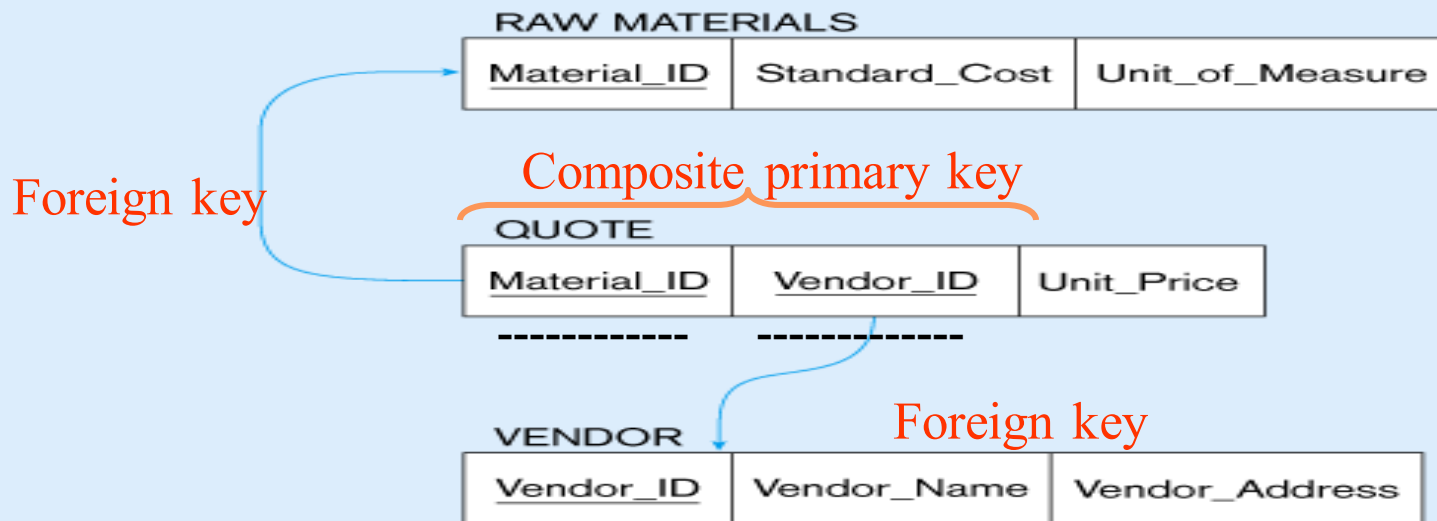
Course(CID, Cname)

Stud_Course(SID, CID)

M:N with attribute



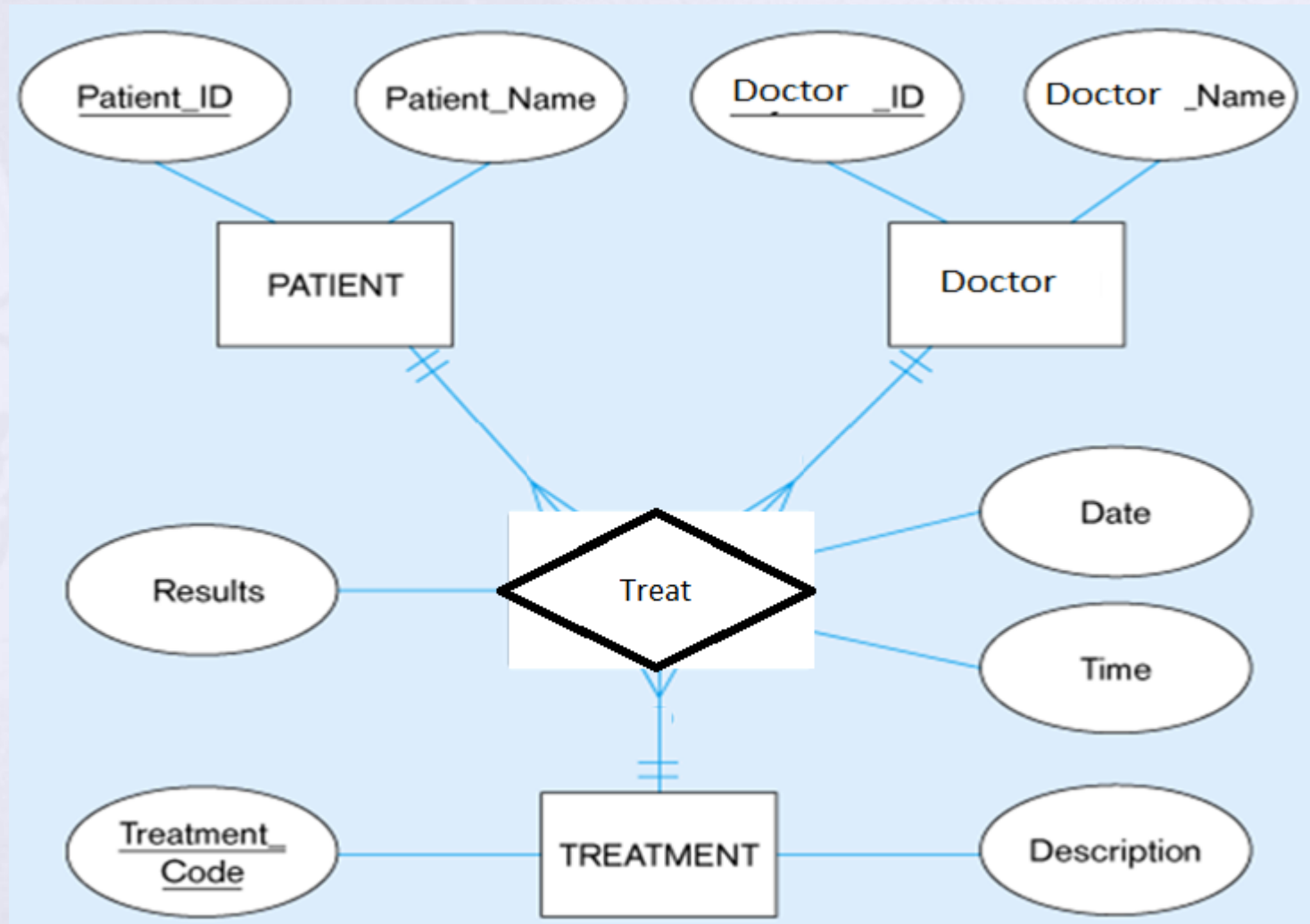
The *Supplies* relationship will need to become a separate relation



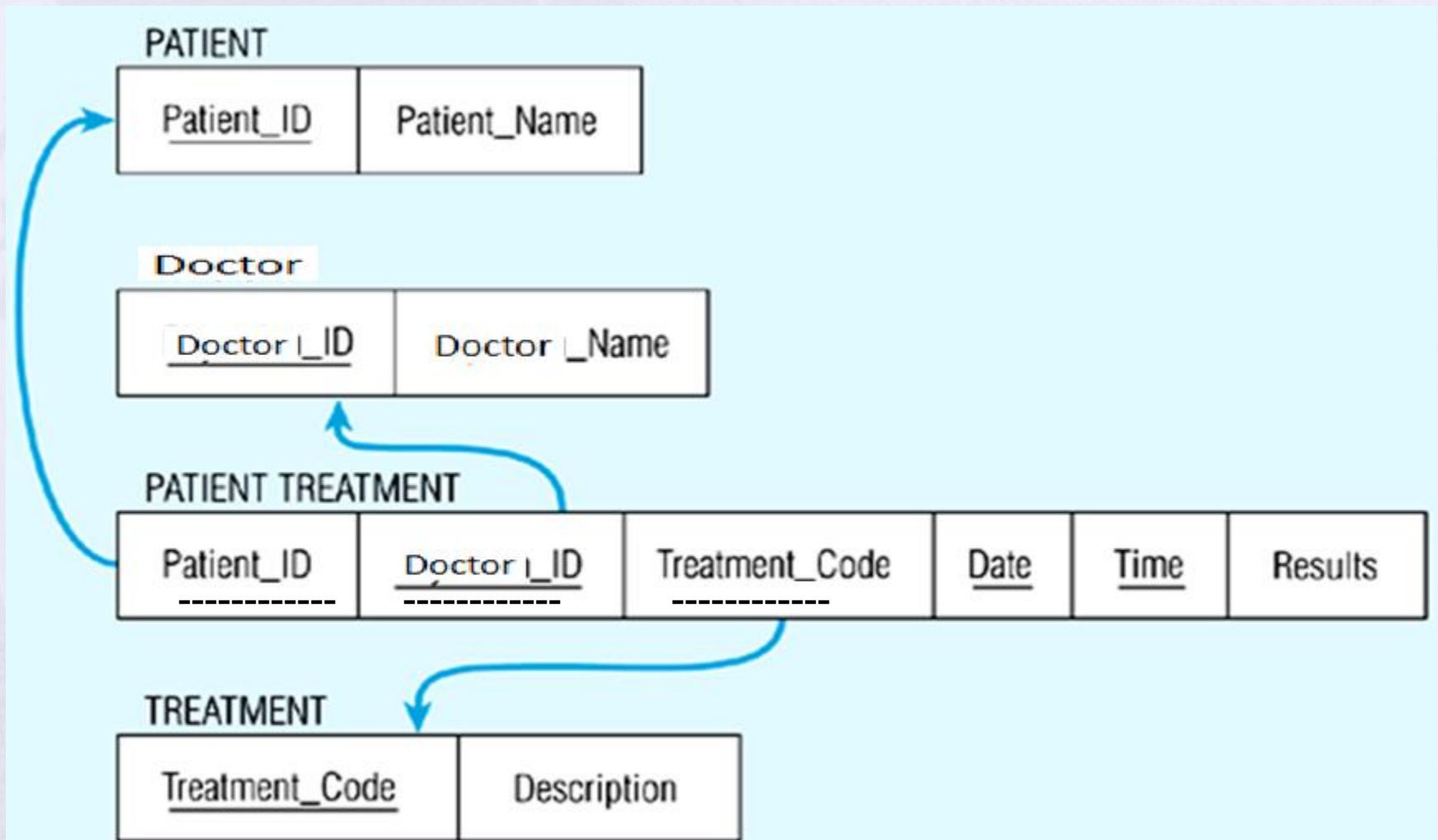
Step 6: Mapping of N-ary Relationship Types.

- 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.

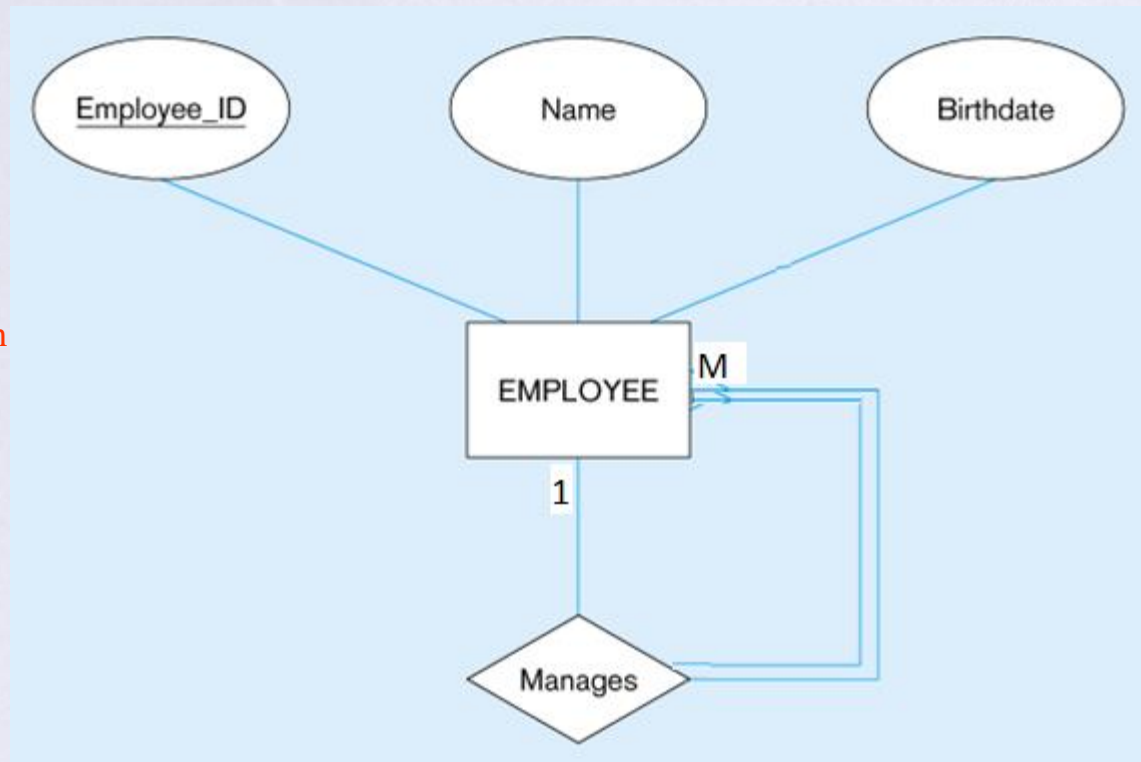


Step 6: Mapping of N-ary Relationship Types.

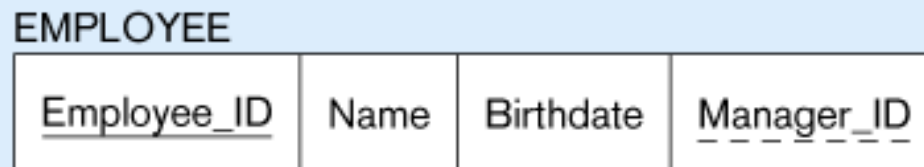


Step 7: Mapping Unary Relationship

(a) EMPLOYEE entity with
Manages relationship

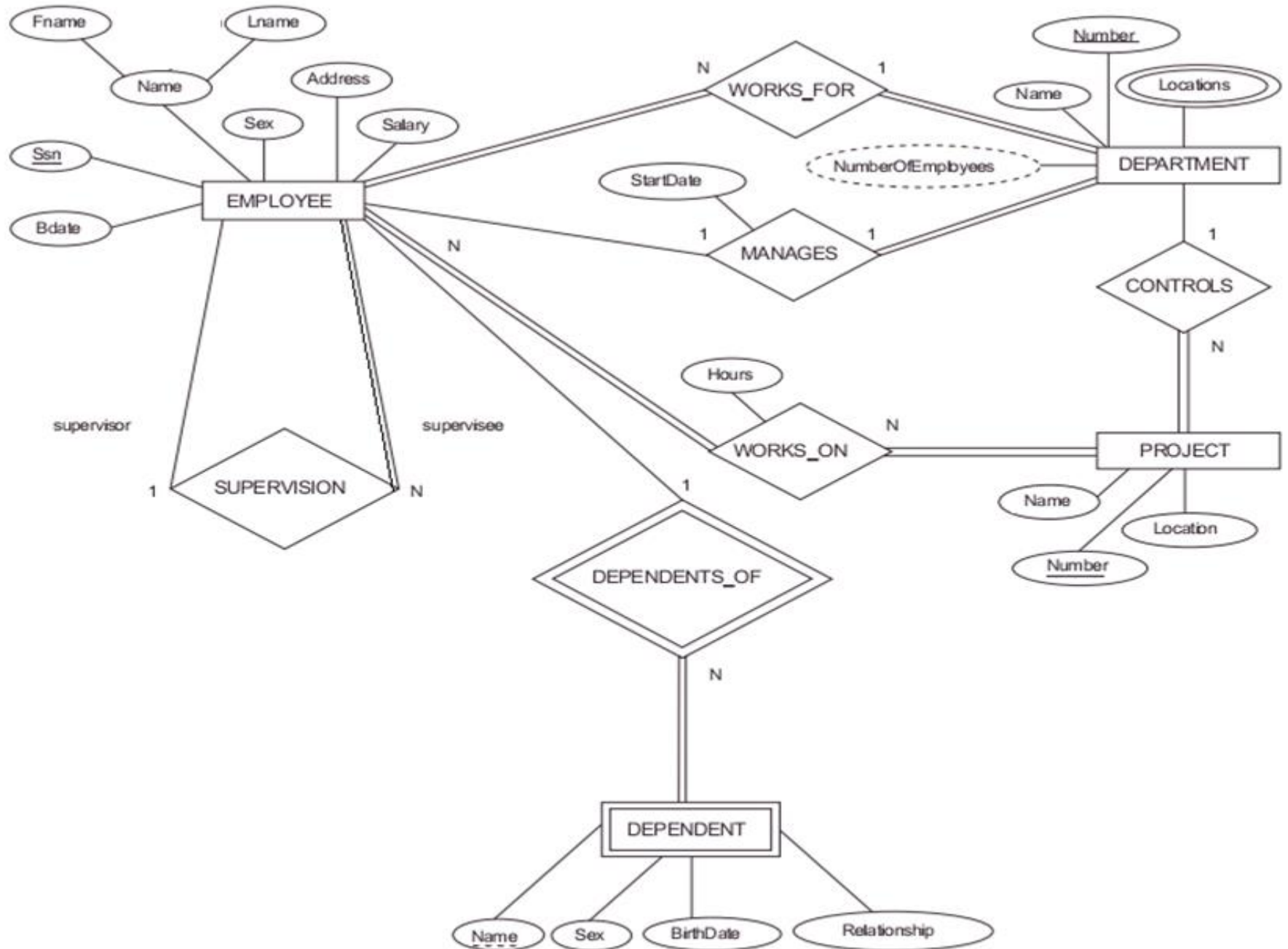


(b) EMPLOYEE
relation with
recursive foreign
key

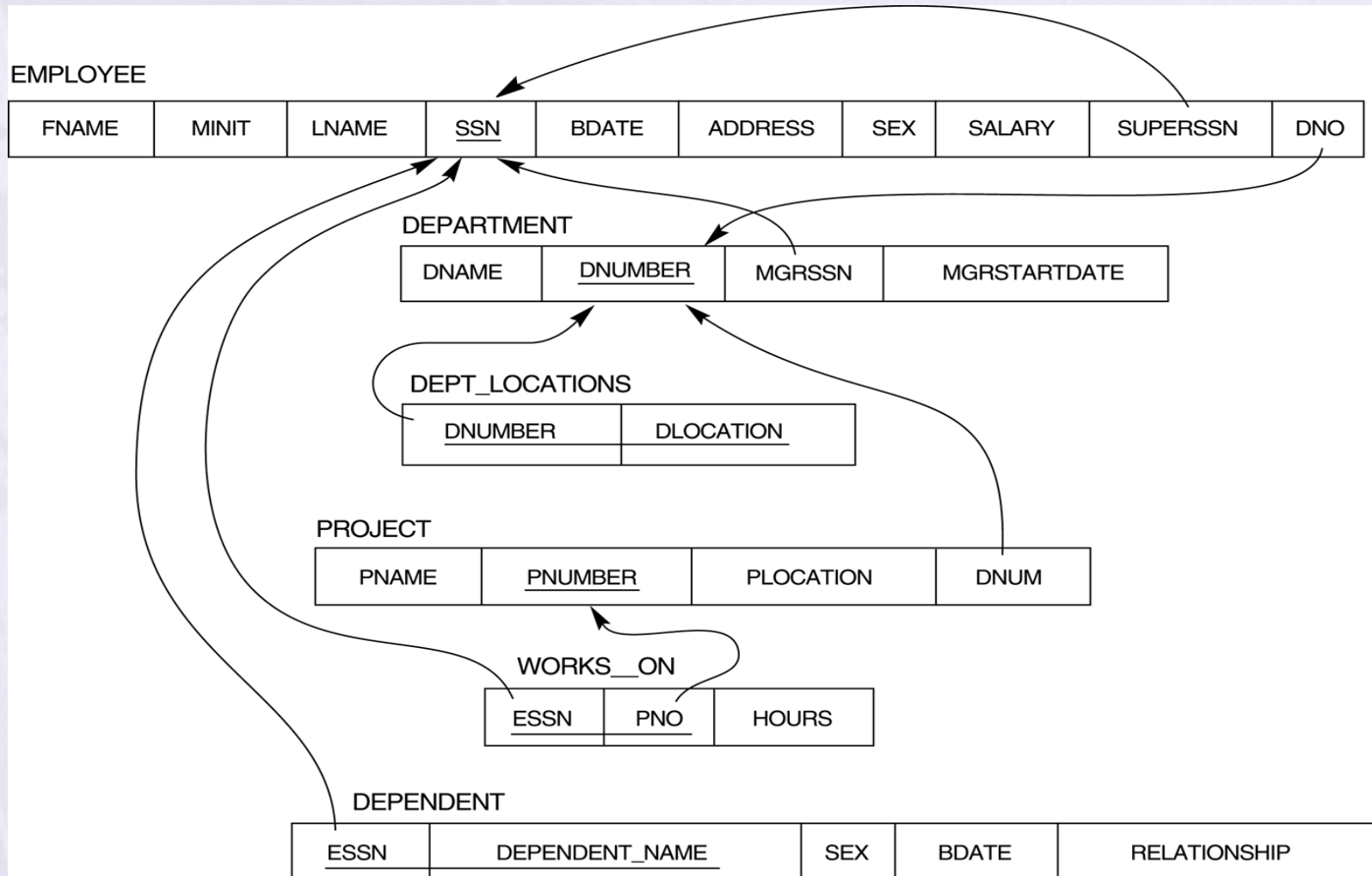


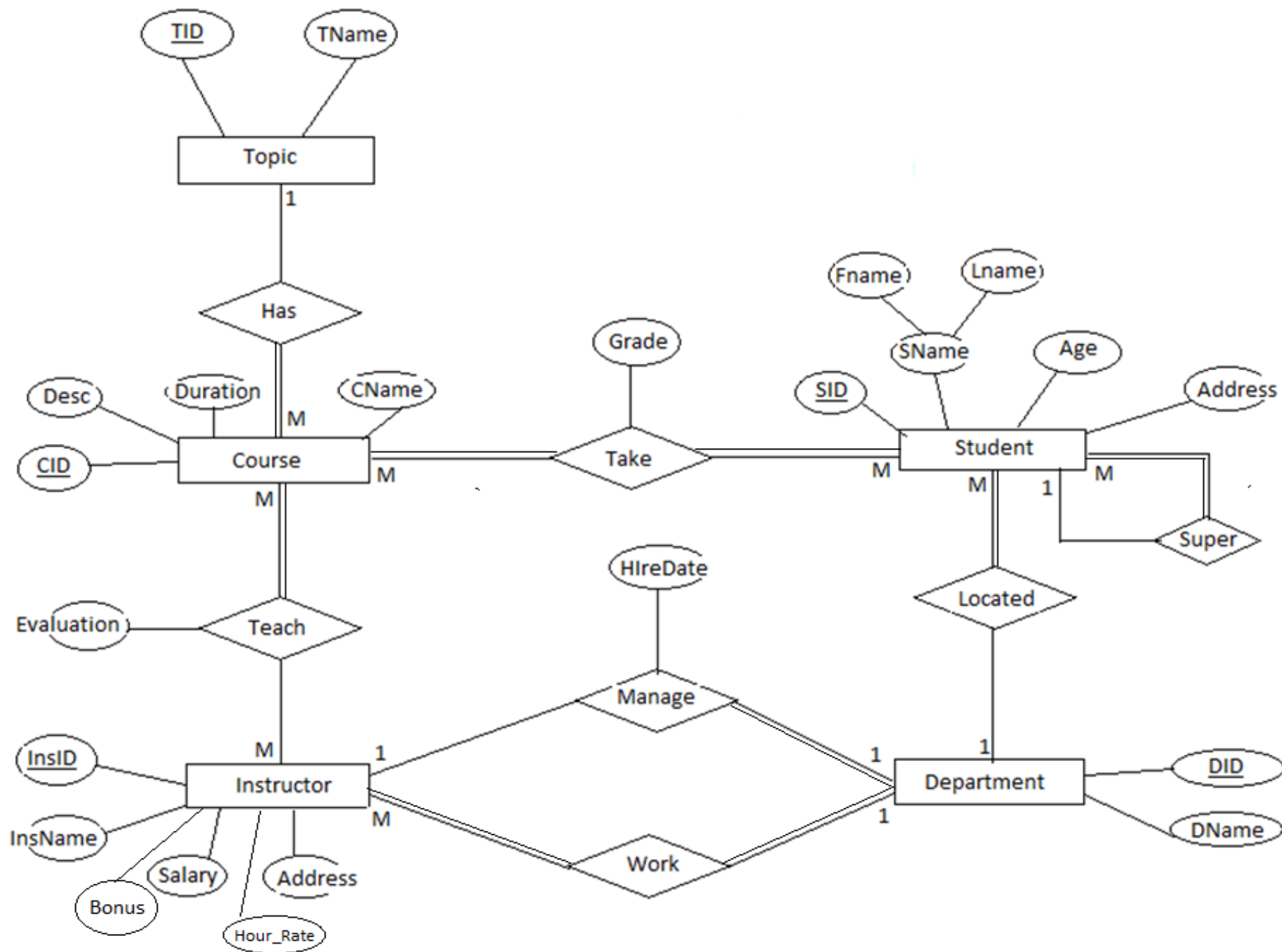


Case Study



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,Evaluation)
- Department(Dept_ID,Dept_Name,Manager_ID,HireDate)

Thank You !!!