Fundamentals of Database Systems

Chapter 5: Mapping ER

Converting ER Diagram to Database Schema

eSPIN University

Overview

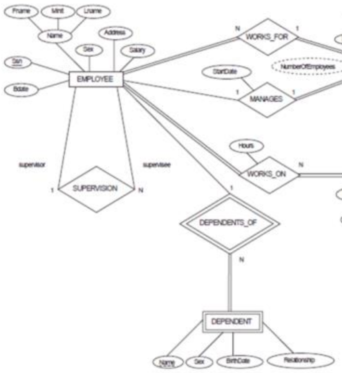
- **We learned that:**

1. A DBMS will allow us to create and manage databases.
2. We can query the database (retrieve data from tables in the database).
3. We can update the database (change, add, or delete data from tables in the database).
4. We can complete a conceptual design of the database using the ER and the EER model.

This chapter

- Learning how to convert the ER diagram to the database
- ERD is a just a conceptual design.
- Now we need to learn how to convert this conceptual design to an actual database schema.

ER model/EER model

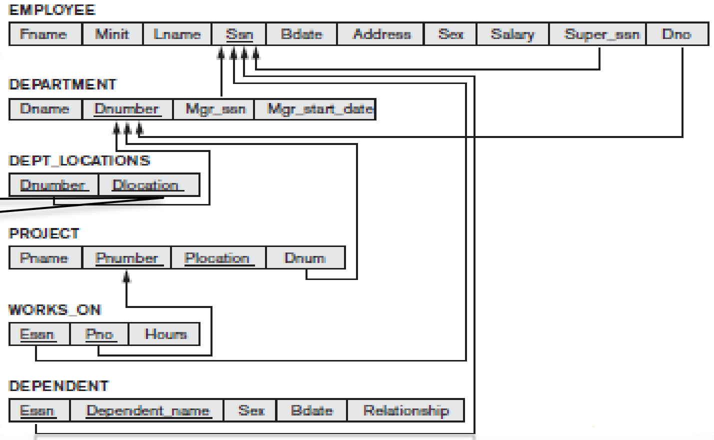


7 steps

Relationnel model

Schema construct

Database schema



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Mapping ER diagrams to relational schema

- **7 Steps (rules):**

1. Mapping of Regular Entity Types
2. Mapping of Weak Entity Types
3. Mapping of Multivalued Attributes
4. Mapping of Binary 1:1 Relationship Types
5. Mapping of Binary 1:N Relationship Types
6. Mapping of Binary N:M Relationship Types
7. Self referencing

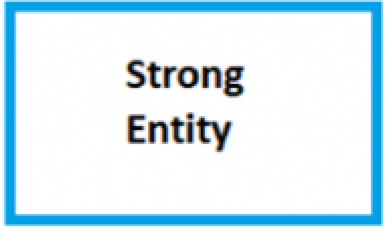
- *Additional rules for mapping the EER*

Strong Entity & Weak Entity

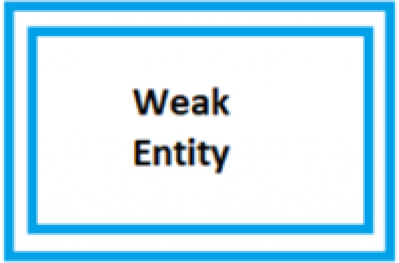
The **strong entity** has a `primary key`. Weak entities are dependent on strong entity. Its existence is not dependent on any other entity.

The **weak entity** in DBMS do not have a primary key and are dependent on the parent entity. It mainly depends on other entities.

Weak Entity is represented by double rectangle:



Strong Entity



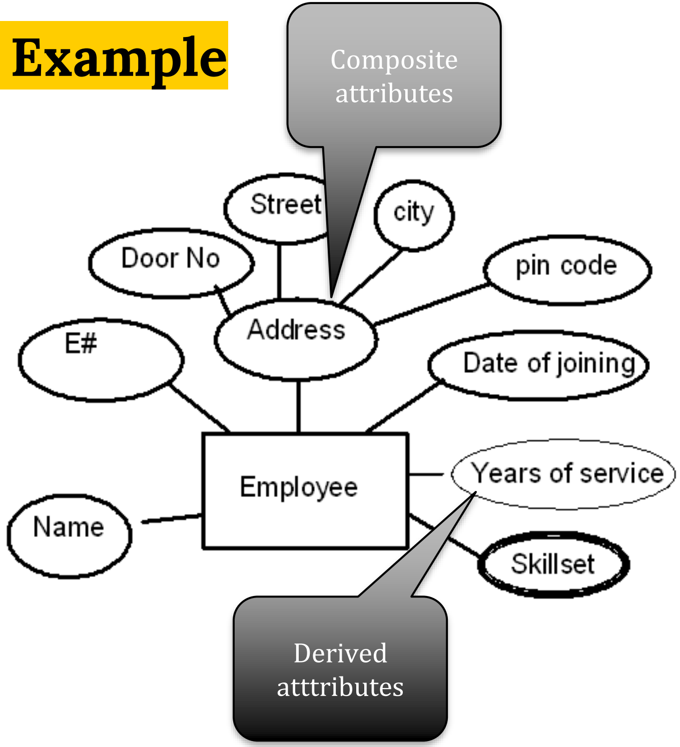
Weak Entity

1. Converting Strong entity types

- **How to convert:**
- Each entity type becomes a table
- Each single-valued attribute becomes a column
- Derived attributes are ignored (why?)
- Composite attributes are represented by components
- The key attribute of the entity type becomes the primary key of the table

1. Converting Strong entity types: Example

- Here address is a composite attribute
- Years of service is a derived attribute (can be calculated from date of joining and current date)



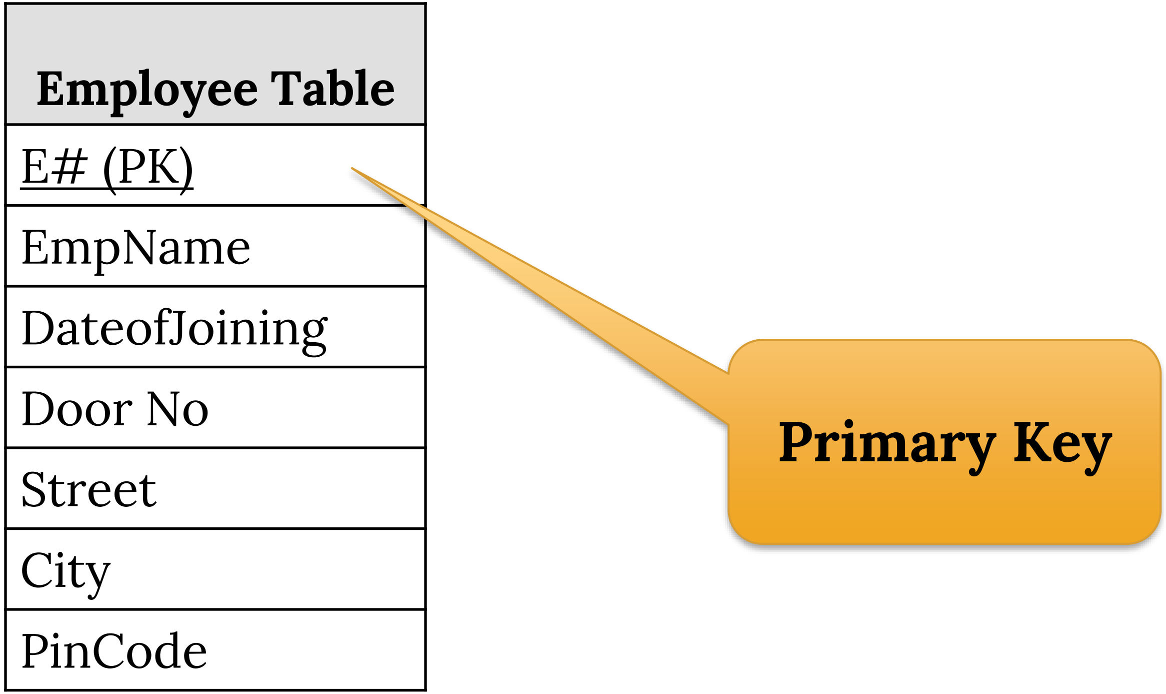
Composite attributes

Derived atttributes

The Database Schema:

Employee (E#, Name, Door_No, Street, City, Pincode, Date_Of_Joining)

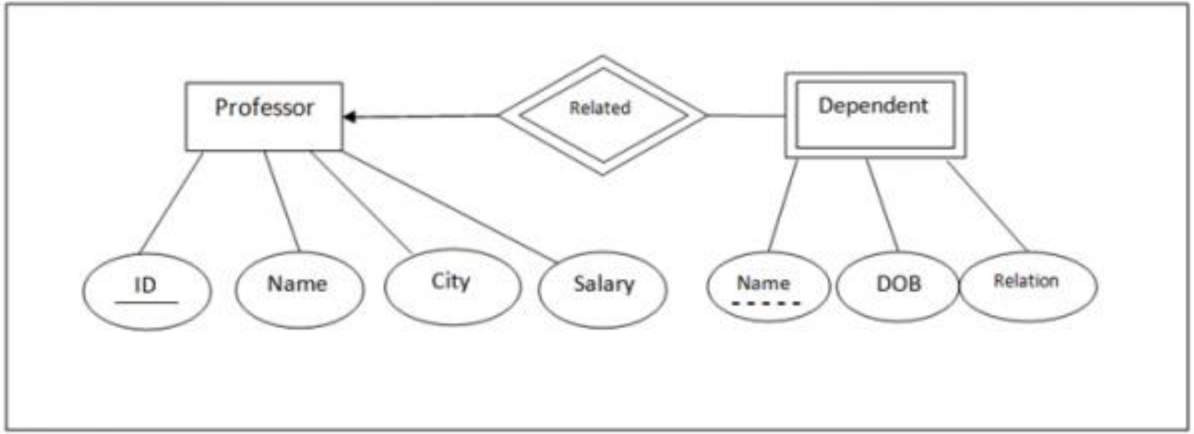
1. Converting Strong entity types: Example



Employee Table
E# (PK)
EmpName
DateofJoining
Door No
Street
City
PinCode

Primary Key

Example of Strong and Weak Entity



The **Strong Entity** is **Professor**, whereas **Dependent** is a **Weak Entity**.

ID is the primary key (represented with a line) and **Name** in Dependent entity is called Partial Key (represented with a dotted line).

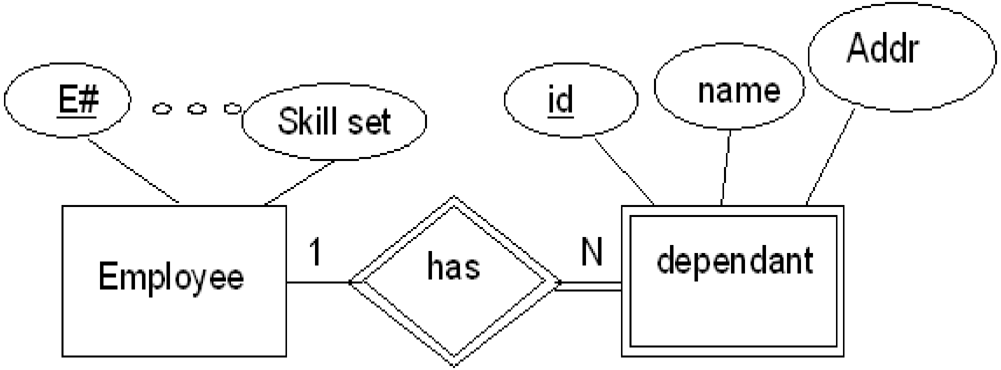
2. Converting Weak Entity Types

- **How to convert:**
- Weak entity types are converted into a table of their own

- The *primary key* of the strong entity acting as a *foreign key* in the table

- This foreign key along with the key of the weak entity form the composite primary key of this table

2. Converting Weak Entity Types: Example



The Relational Schema

Employee (E#.....)

Dependant (Id, E#, Name, Address)

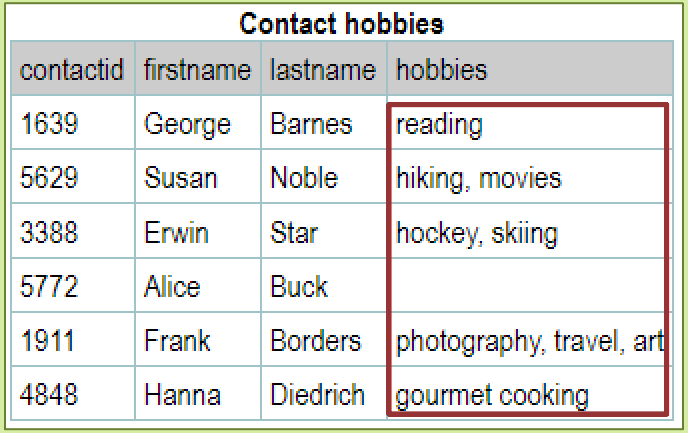
E# in dependent is a FK referencing E# in Employee

3. Converting **Multivalued Attributes**

- **How to convert:**
- For each multivalued attribute A, create a new relation R.

A multivalued attribute can have more than one value at a time for an attribute. For example,

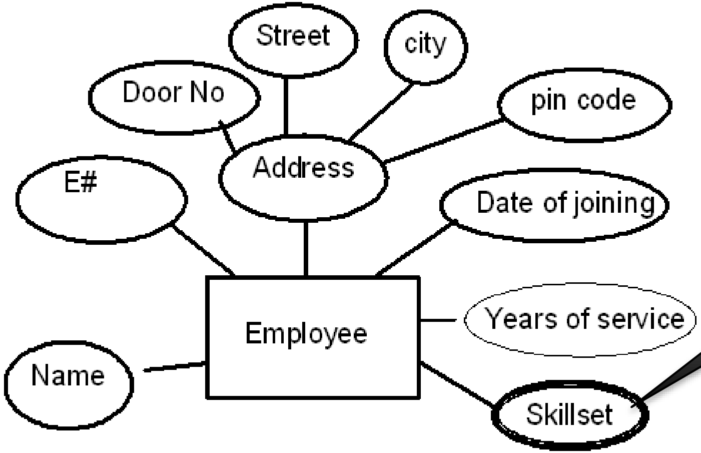
- the ***skills of a surgeon*** is a multivalued attribute since a surgeon can have more than one skill.



Contact hobbies

contactid	firstname	lastname	hobbies
1639	George	Barnes	reading
5629	Susan	Noble	hiking, movies
3388	Erwin	Star	hockey, skiing
5772	Alice	Buck	
1911	Frank	Borders	photography, travel, art
4848	Hanna	Diedrich	gourmet cooking

3. Converting Multivalued Attributes: *Example*



Skill set is a multi-valued attribute

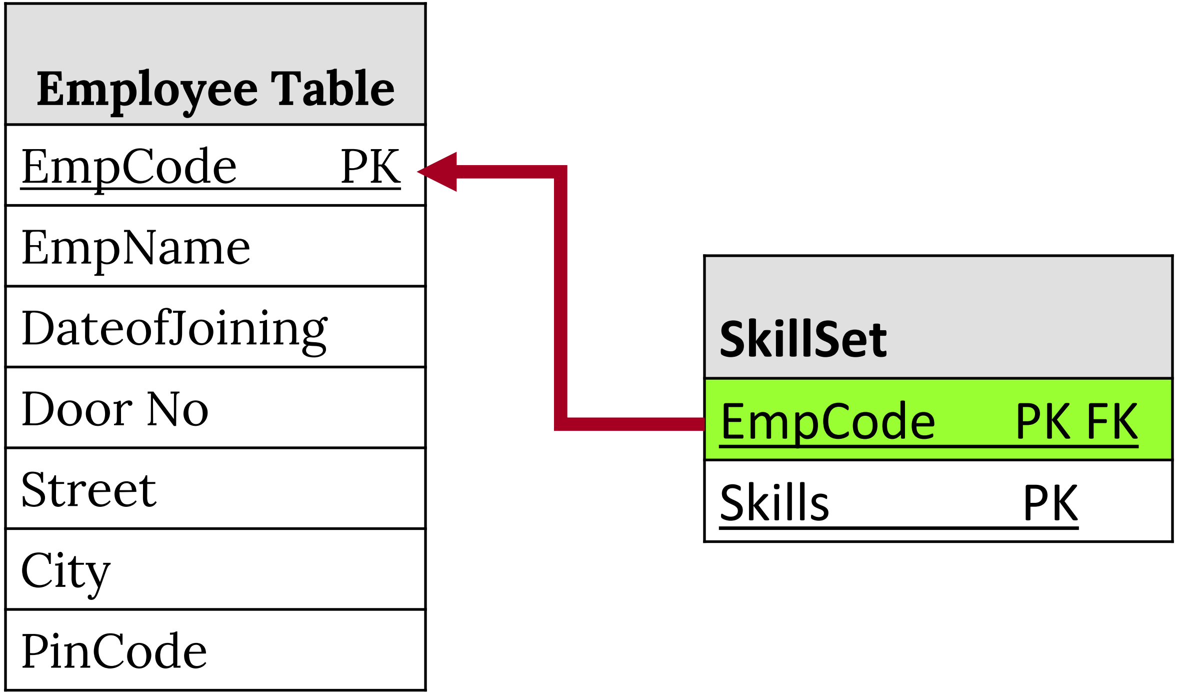
The Relational Schema

Employee (E, Name, Door_No, Street, City, Pincode, Date_Of_Joining)

Emp_Skillset(E#, Skillset)

E# and skillset are both the PK

3. Converting Multivalued Attributes: *Example*



Employee Table	
EmpCode	PK
EmpName	
DateofJoining	
Door No	
Street	
City	
PinCode	

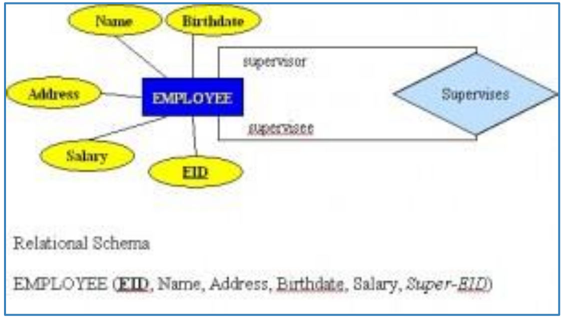
SkillSet	
EmpCode	PK FK
Skills	PK

Converting Relationships

- The way relationships are converted depends on the cardinality and the degree of the relationship

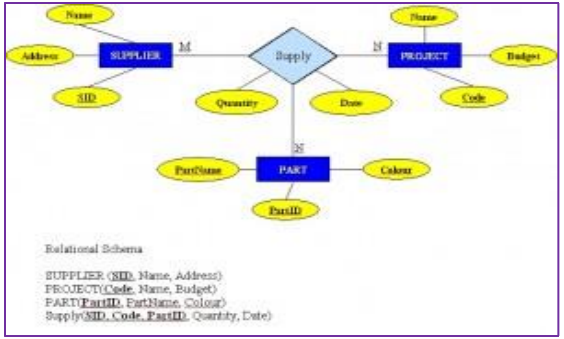
- **The possible cardinalities are:**
- 1:1, 1:M, N:M

- **The degrees are:**
- Unary
- Binary
- Ternary



Relational Schema

EMPLOYEE (EID, Name, Address, Birthdate, Salary, *Super-EID*)



Relational Schema

SUPPLIER (SID, Name, Address)
PROJECT(Code, Name, Budget)
PART(PartID, PartName, Colour)
Supply(SID, Code, PartID, Quantity, Date)

4. Converting Binary 1:1 Relationship Types

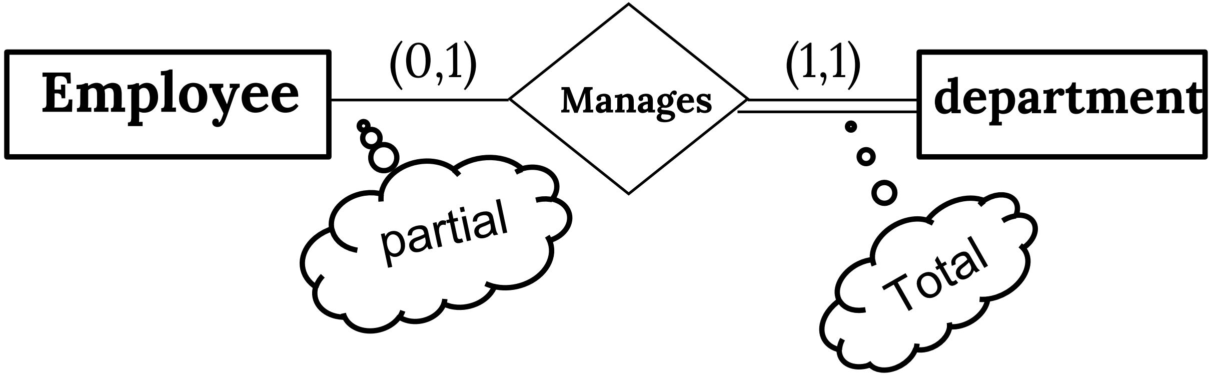
- **How to convert:**
- Two cases:

a. **Combination of participation types (total and partial)**

b. **Uniform participation types**

4. Converting Binary 1:1.. *Example*

a. Combination of participation types



The primary key of the partial participant will become the foreign key of the total participant

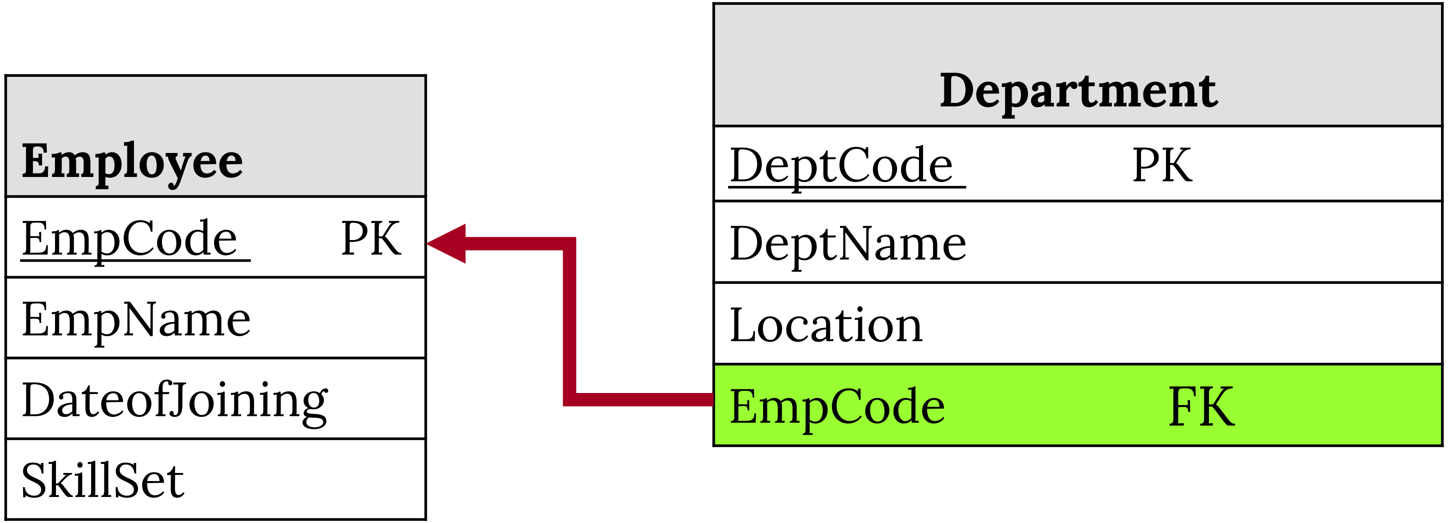
- **Database schema:**

Employee(E, Name,...)

Department (Dept, Name...,E#)

4. Converting Binary 1:1.. *Example*

a. Combination of participation types

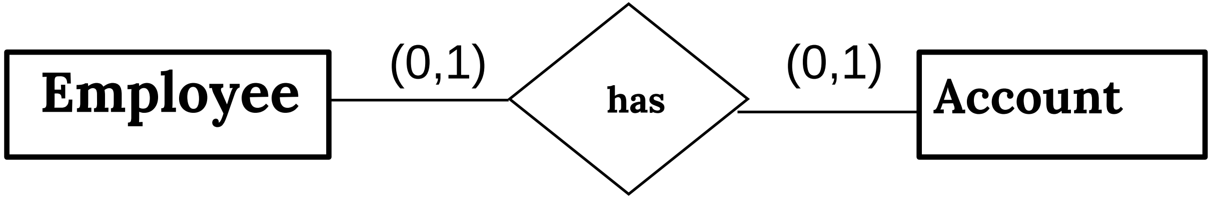


Employee	
EmpCode	PK
EmpName	
DateofJoining	
SkillSet	

Department	
DeptCode	PK
DeptName	
Location	
EmpCode	FK

4. Converting Binary 1:1.. *Example*

b. Uniform participation types



The primary key of either of the participants can become a foreign key in the other

- **Database schema:**

  Employee (E#, name…)

  Account(Account_ID, E#, bank)

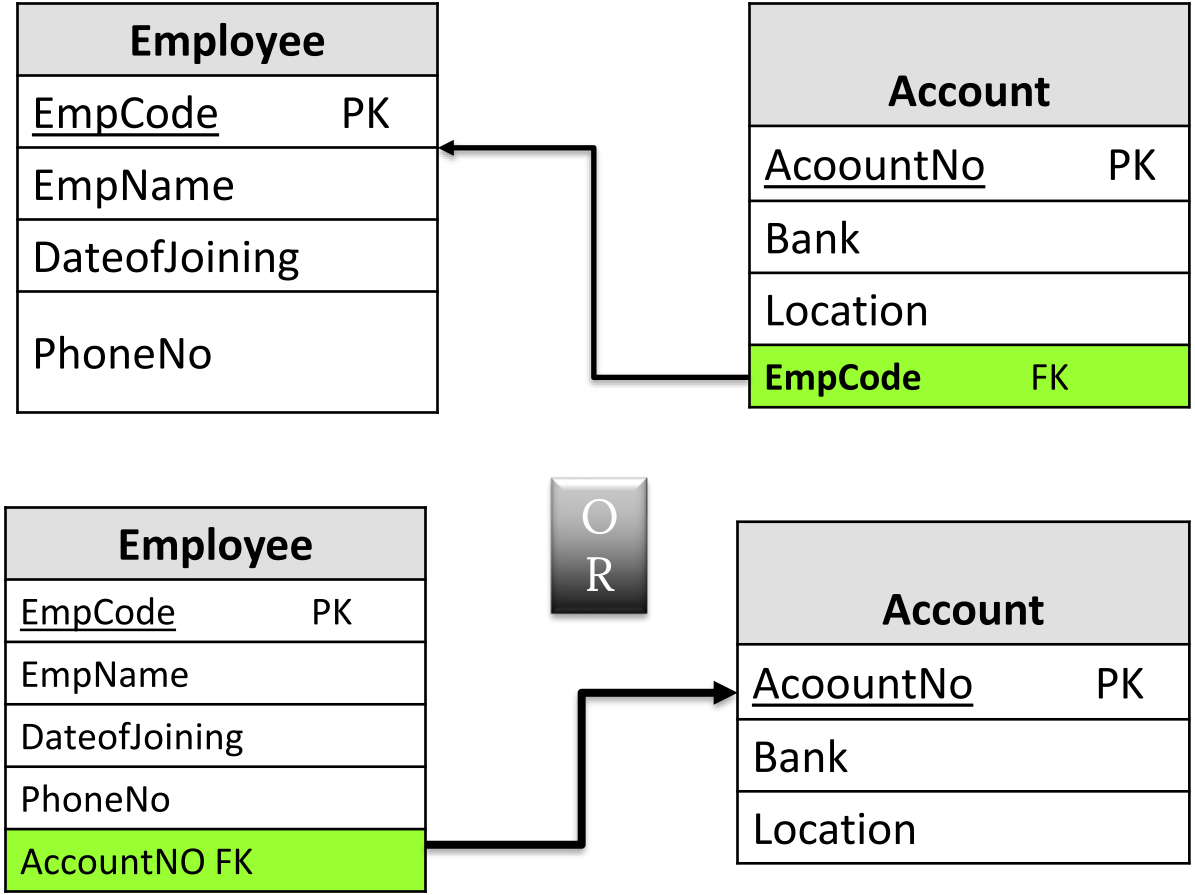
  (or)

  Employee (E#, Account_ID,name…)

  Account(Account_ID, bank)

4. Converting Binary 1:1.. *Example*

b. Uniform participation types



Employee	
EmpCode	PK
EmpName	
DateofJoining	
PhoneNo	

Account	
AcoountNo	PK
Bank	
Location	
EmpCode	FK

O
R

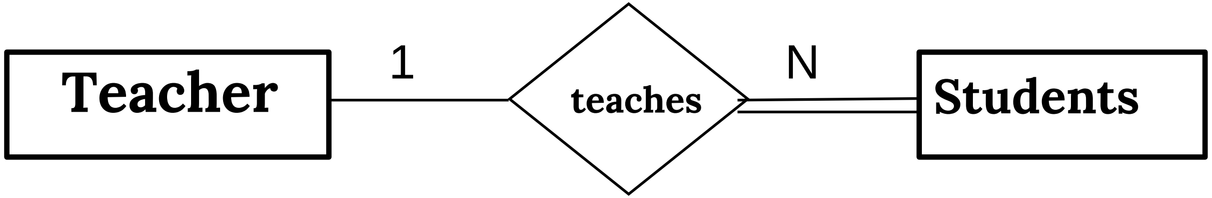
Employee	
EmpCode	PK
EmpName	
DateofJoining	
PhoneNo	
AccountNO FK	

Account	
AcoountNo	PK
Bank	
Location	

5. Converting Binary 1:N Relationship Types

- **How to convert:**
- **The primary key of the relation on the “1” side of the relationship becomes a foreign key in the relation on the “N” side**

5. Converting Binary 1:N.. *Example*



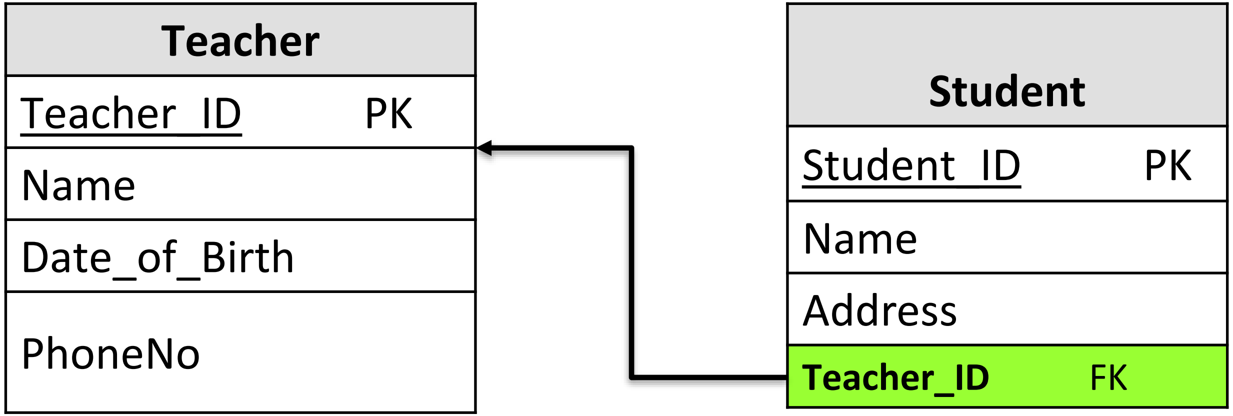
The primary key of the relation on the "1" side of the relationship becomes a foreign key in the relation on the "N" side

- **Database schema:**

Author (Teacher_ID, name…)

Book(Student_ID, Name, Teacher_ID)

5. Converting Binary 1:N.. *Example*



Teacher	
Teacher_ID	PK
Name	
Date_of_Birth	
PhoneNo	

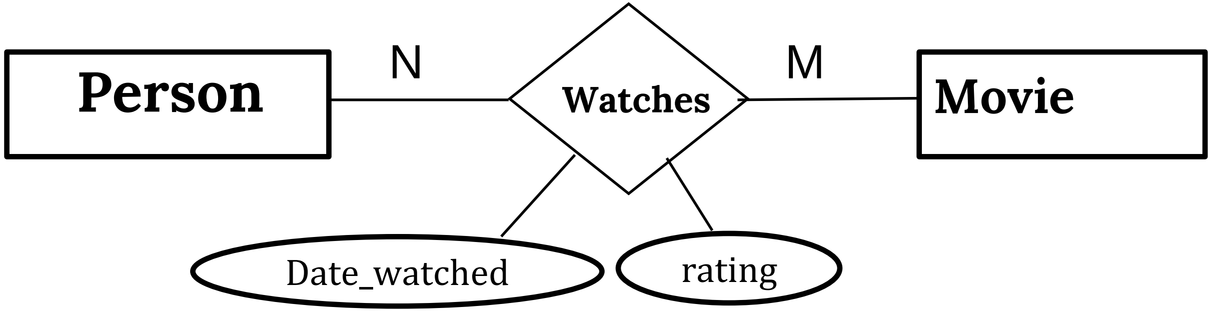
Student	
Student_ID	PK
Name	
Address	
Teacher_ID	FK

6. Converting Binary N:M Relationship Types

◉ How to convert:

1. A **new table** is created to represent the relationship
2. Contains **two** foreign keys - one from each of the participants in the relationship
3. The primary key of the new table is the combination of the two foreign keys

6. Converting Binary 1:N.. *Example*



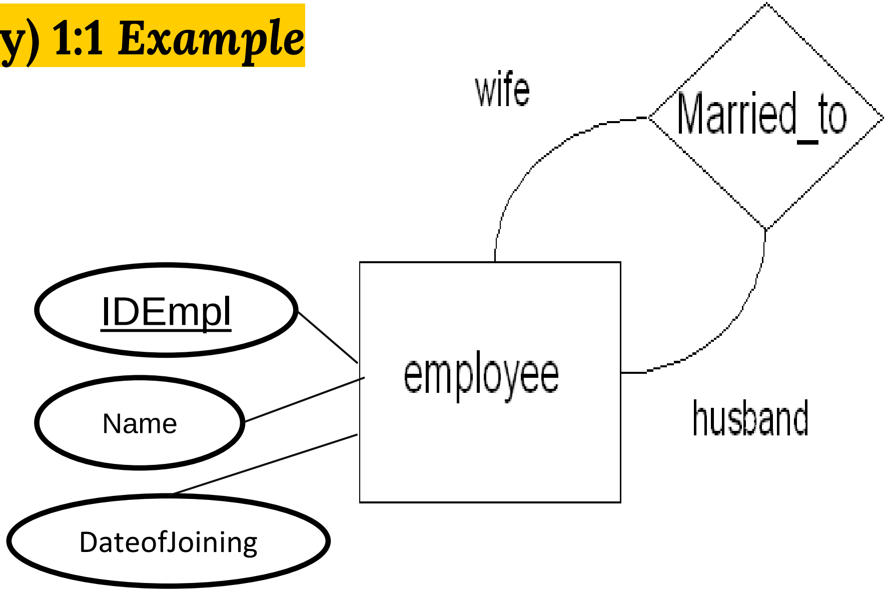
- **Database schema:**

Person (P_ID, name…)

Movie (Movie_ID, title, length)

Watched_Movies(P_ID, Movie_ID, date_watched, rating)

7. Self Referencing (unary) 1:1 *Example*

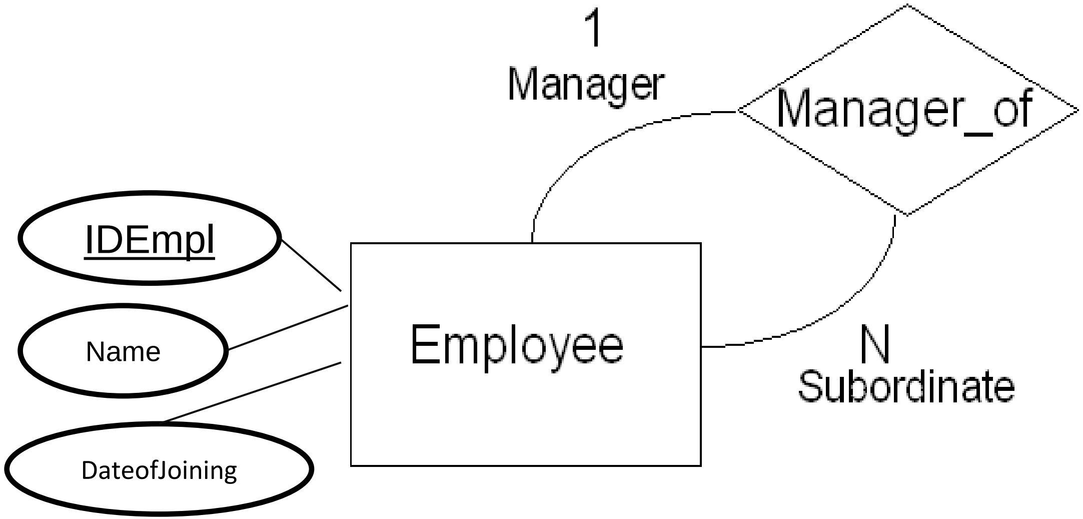


New attribute is added. The new attribute is the primary key field and it becomes a foreign key in the same table

- **Database schema:**

Employee (IDEmpl, name, DateOfJoining, partner_ID)

7. Self Referencing (unary) 1:N *Example*

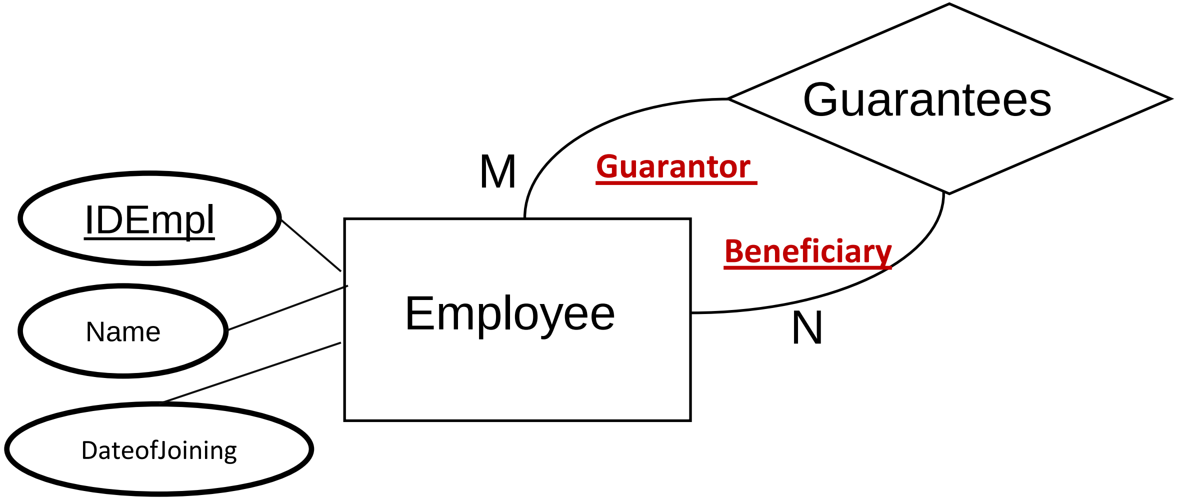


Exactly the same as 1:1

- **Database schema:**

Employee (IDEmpl, name, DateOfJoining, manager_ID)

7. Self Referencing (unary) N:M *Example*



There will be two resulting tables. One to represent the entity and another to represent the M:N relationship

- **Database schema:**

Employee (IDEmpl, name, DateOfJoining)

Guarantors (ID_Guarantors, ID_Beneficiary)