Logical database design

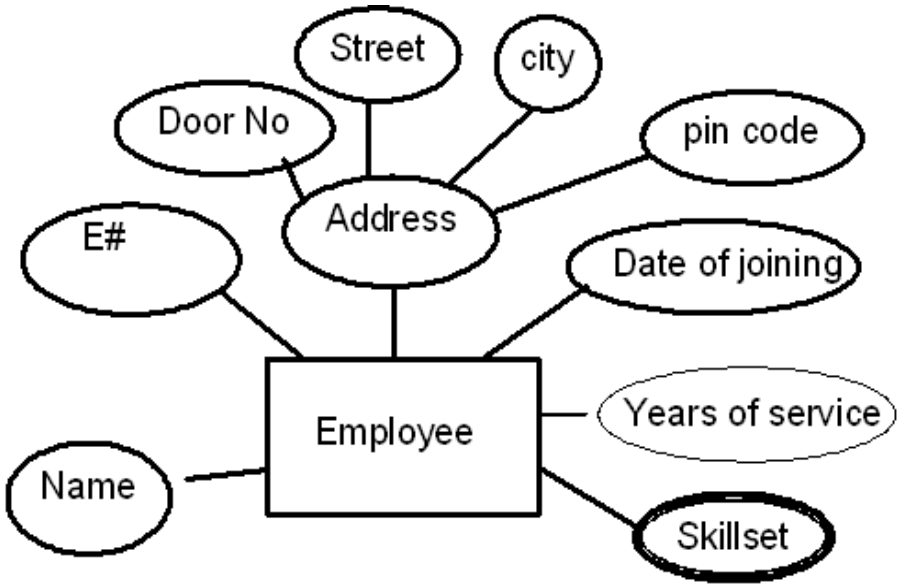
Converting ER diagrams to relational schema

Converting Strong entity types

- Each **entity type** becomes a **table**
- Each **single-valued attribute** becomes a **column**
- **Derived attributes** are **ignored**
- **Composite attributes** are represented by **components**
- **Multi-valued attributes** are represented by a **separate table**
- The **key attribute** of the entiry type becomes the **primary key** of the table

Entity example

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

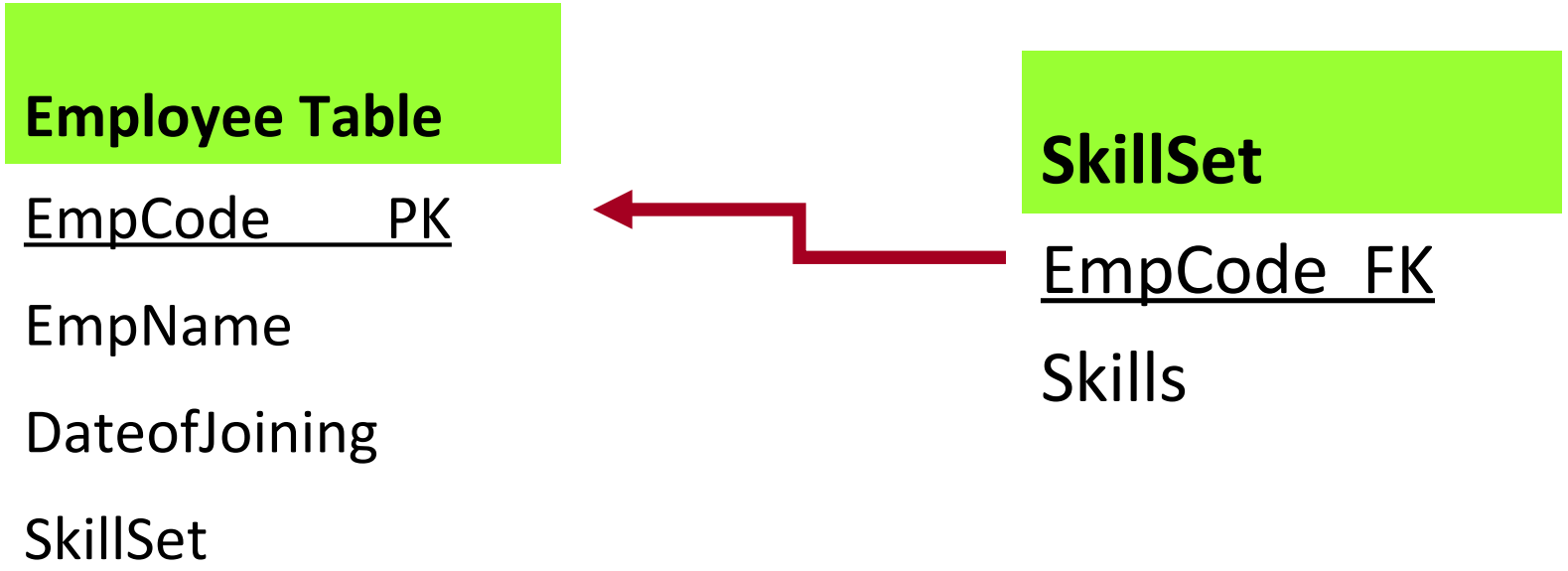


- **The relational Schema**

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

Emp_Skillset(E#, Skillset)

Entity Example (Contd…)

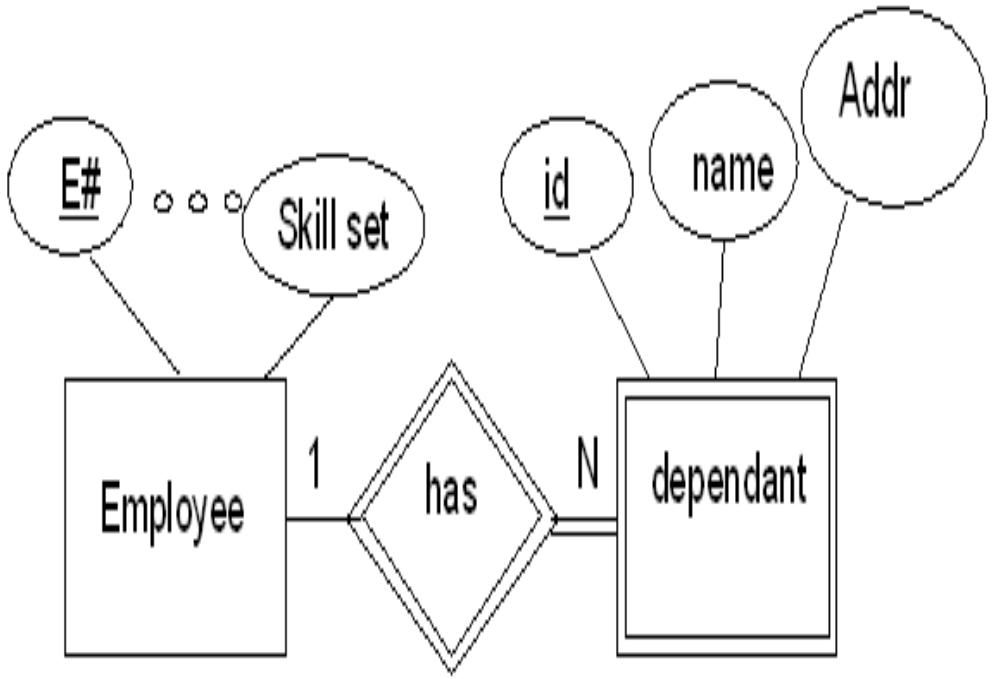


Converting weak entity types

- Weak entity types are converted into a table of their own, with 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

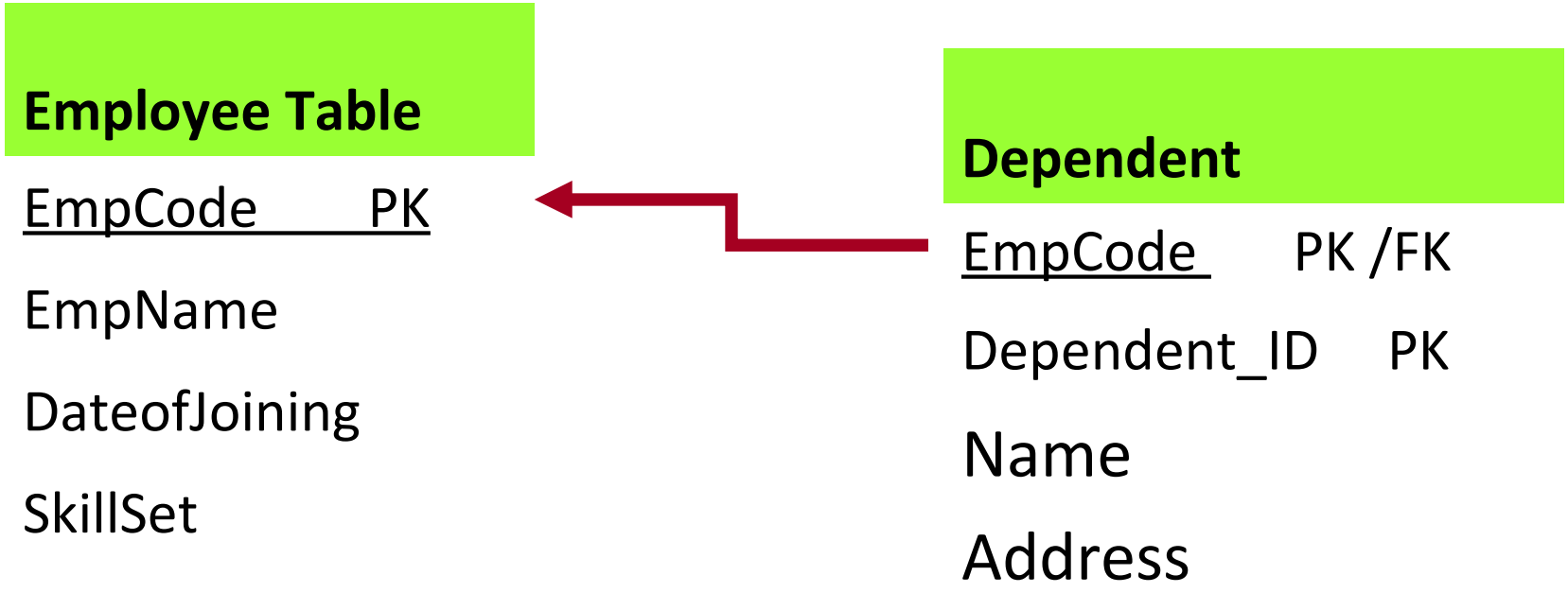
- **The Relational Schema**



Employee (E# ,…….)

Dependant (Employee, Dependant_ID, Name, Address)

Converting weak entity types (Contd…)



Converting relationships

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

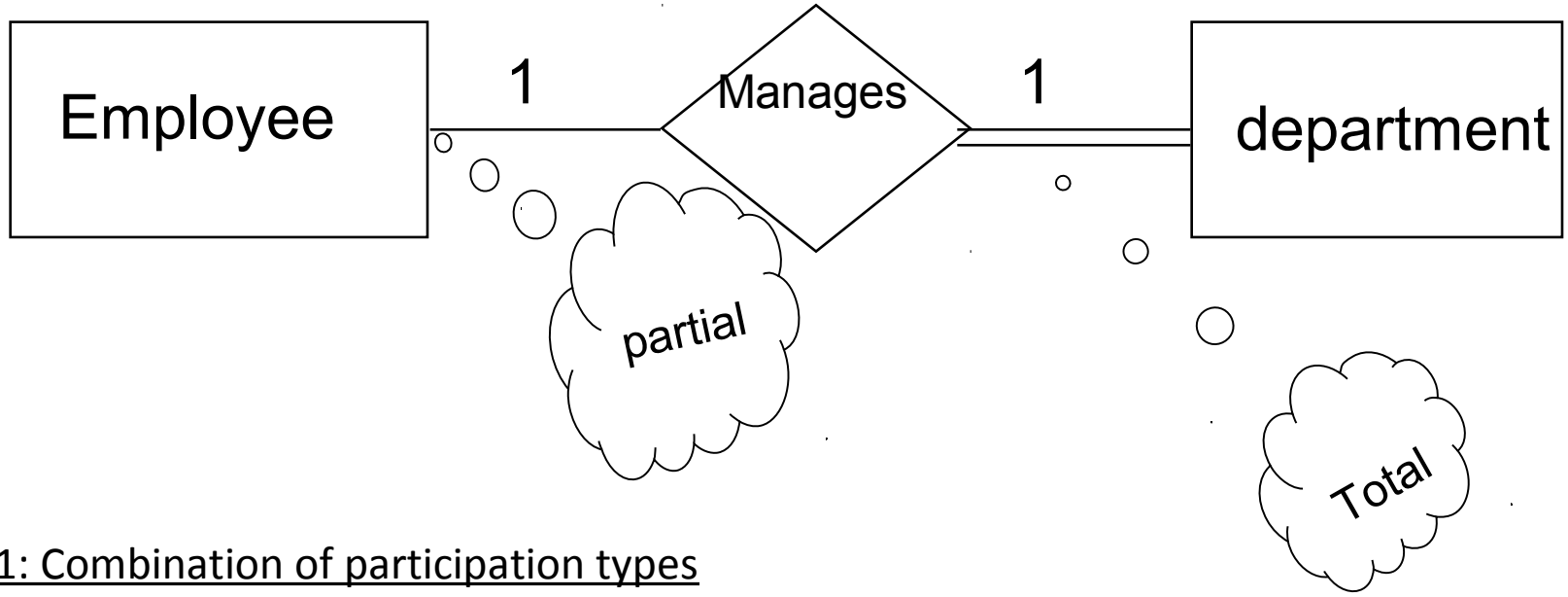
- The possible cardinalities are:

  1:1, 1:M, N:M

- The degrees are:

  Unary
  Binary
  Ternary …

Binary 1:1



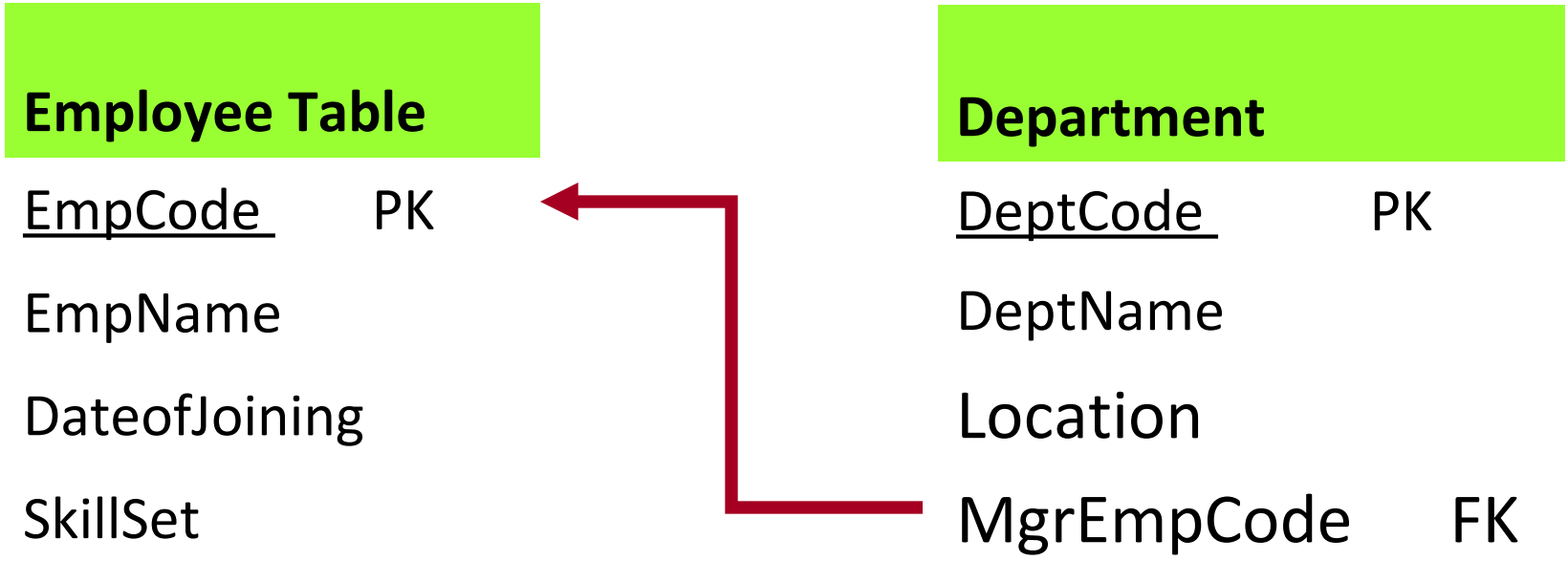
- Case 1: Combination of participation types

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

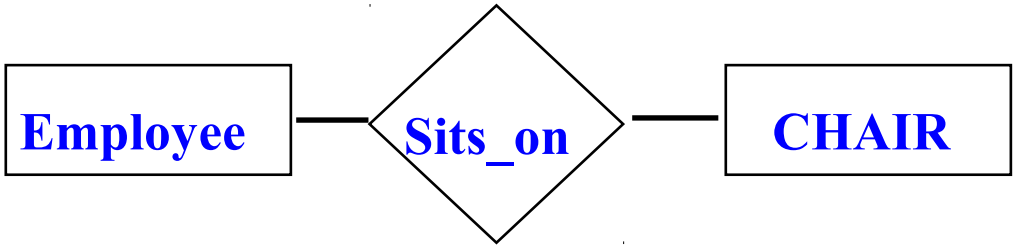
  Employee(E#, Name,…)

  Department (Dept#, Name…,MgrE#)

Binary 1 : 1


Employee Table
EmpCode PK
EmpName
DateofJoining
SkillSet
Department
DeptCode PK
DeptName
Location
MgrEmpCode FK

Binary 1:1



- Case 2: Uniform participation types

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

Employee (E#,name…)

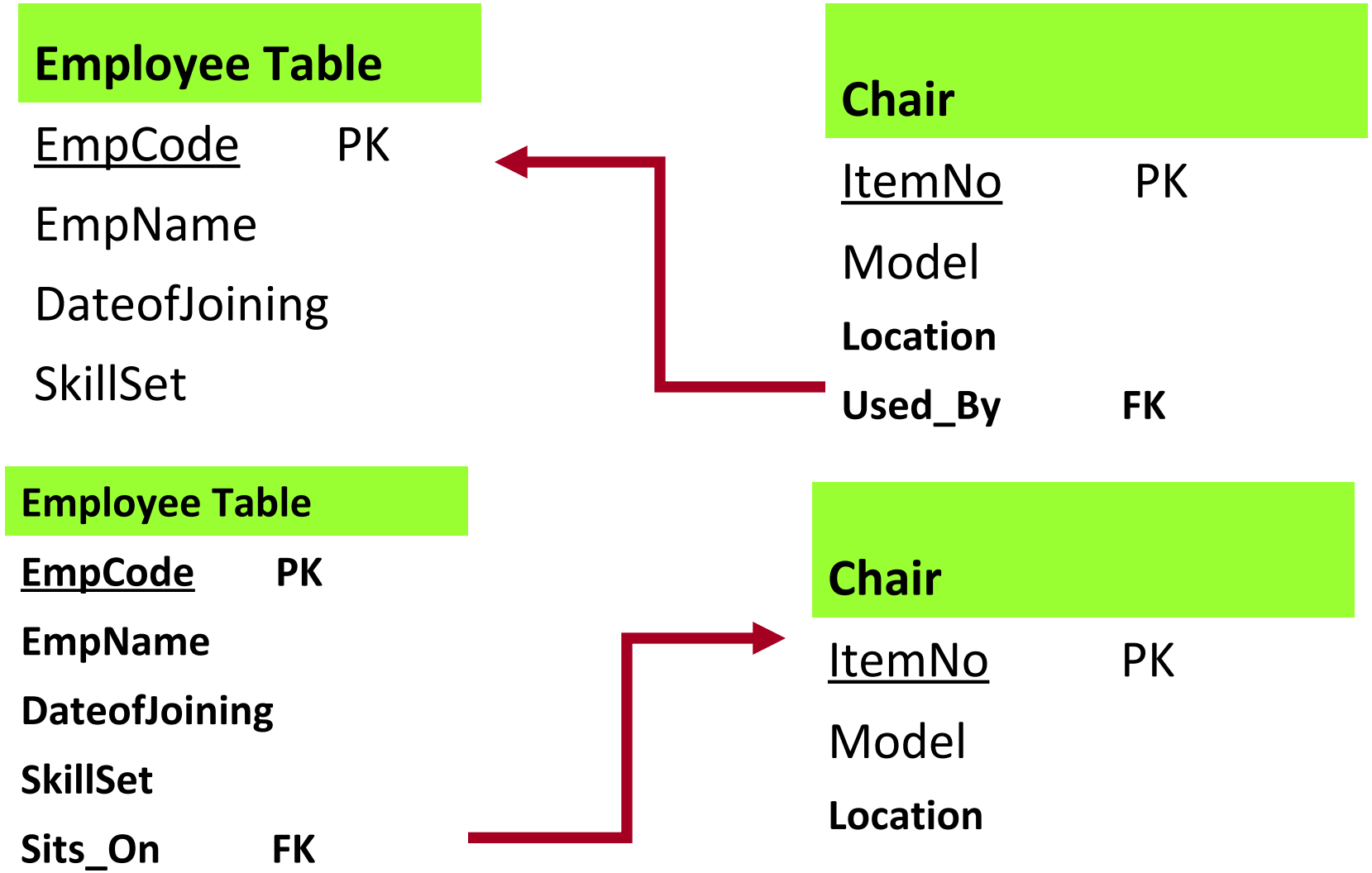
Chair(item#, model, location, used_by)

(or)

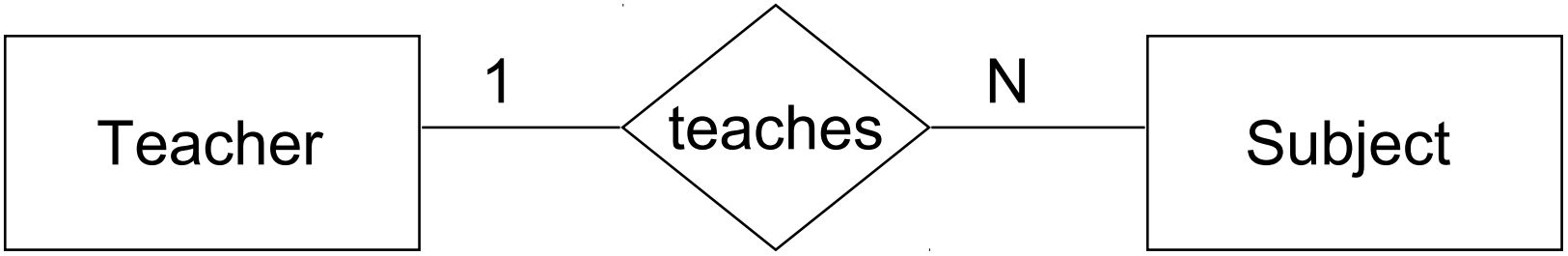
Employee (E#, Name….Sits_on)

Chair (item#,….)

Binary 1 : 1


Employee Table
EmpCode PK
EmpName
DateofJoining
SkillSet
Chair
ItemNo PK
Model
Location
Used_By FK
Employee Table
EmpCode PK
EmpName
DateofJoining
SkillSet
Sits_On FK
Chair
ItemNo PK
Model
Location

Binary 1:N

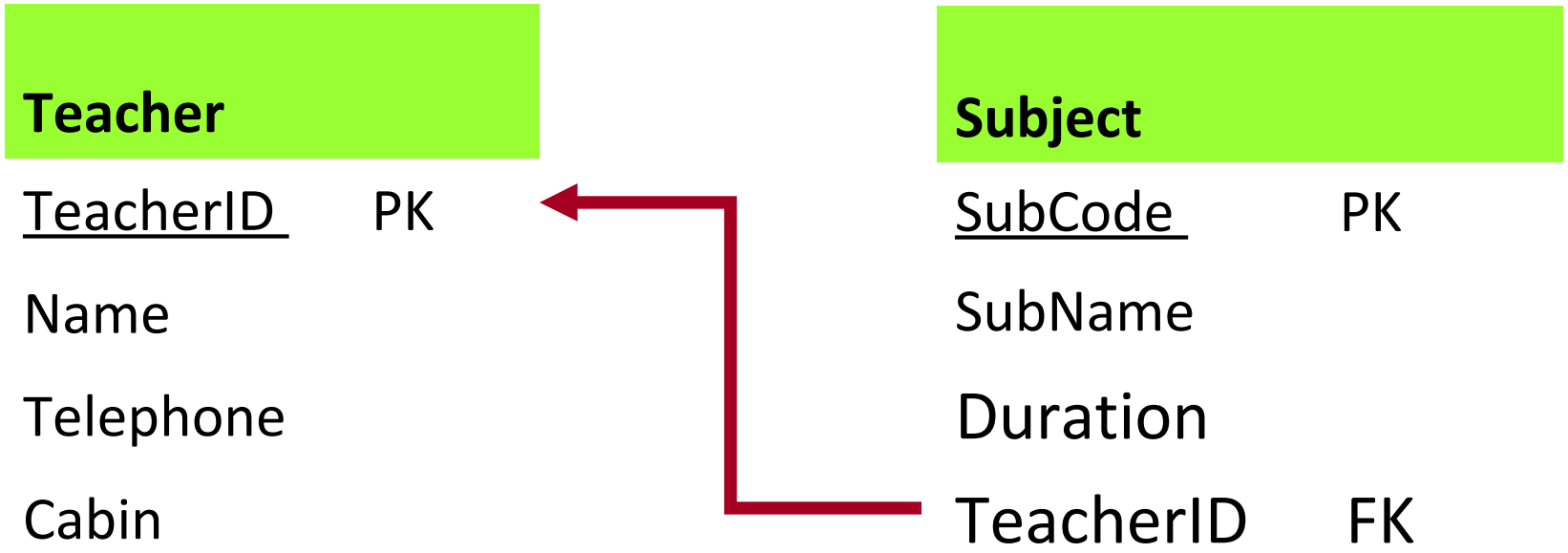


The primary key of the relation on the “1” side of the relationship becomes a foreign key in the relation on the “N” side

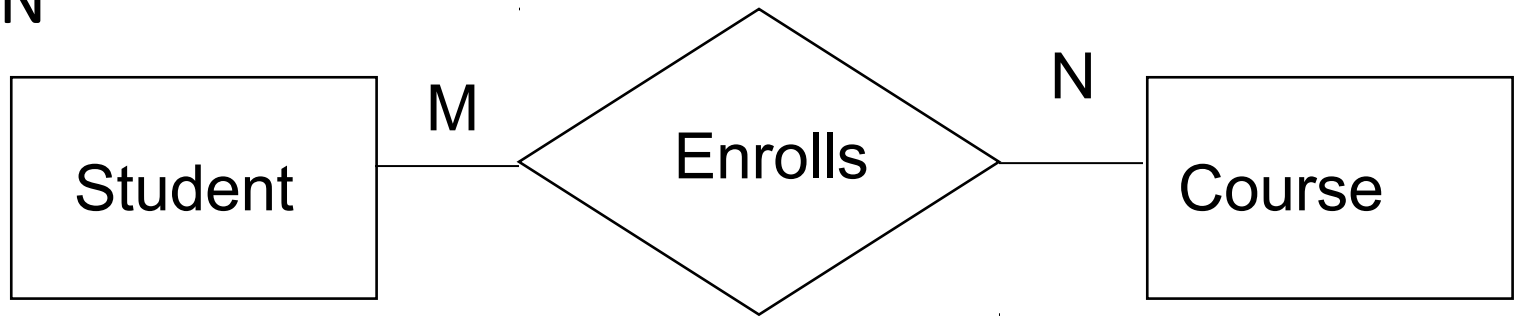
Teacher (<u>ID</u>, Name, Telephone, ...)

Subject (<u>Code</u>, Name, ..., Teacher)

Binary 1 : N


Teacher
TeacherID PK
Name
Telephone
Cabin
Subject
SubCode PK
SubName
Duration
TeacherID FK

Binary M:N

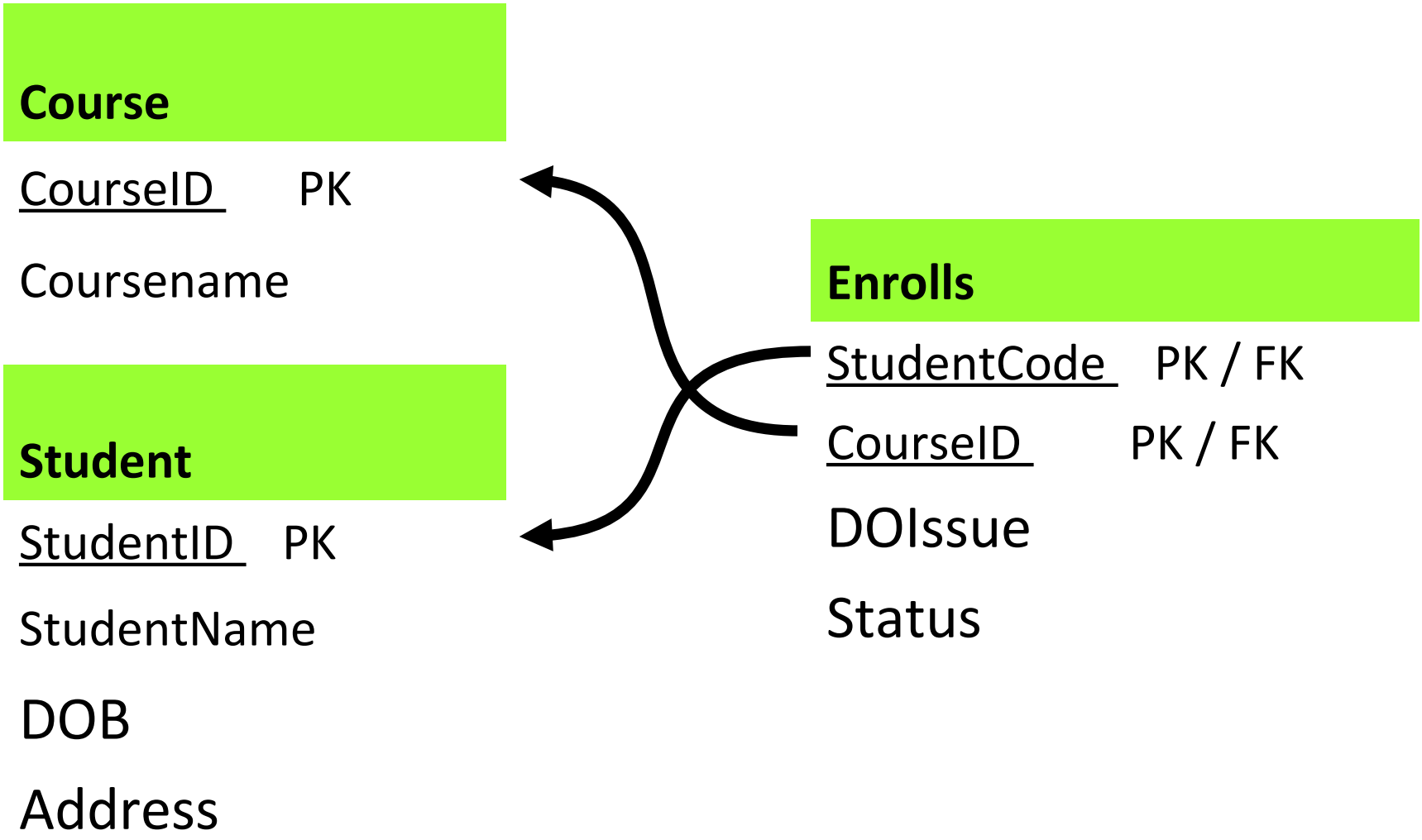


- A new table is created to represent the relationship
- Contains two foreign keys - one from each of the participants in the relationship
- The primary key of the new table is the combination of the two foreign keys

Student (Sid#, Title…) Course(C#, CName,…)

Enrolls (Sid#, C#)

Binary M : N

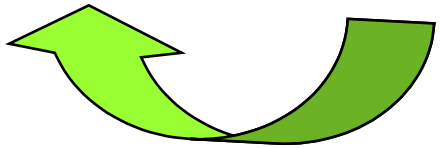

Course
CourseID PK
Coursename
Student
StudentID PK
StudentName
DOB
Address
Enrolls
StudentCode PK / FK
CourseID PK / FK
DOIssue
Status

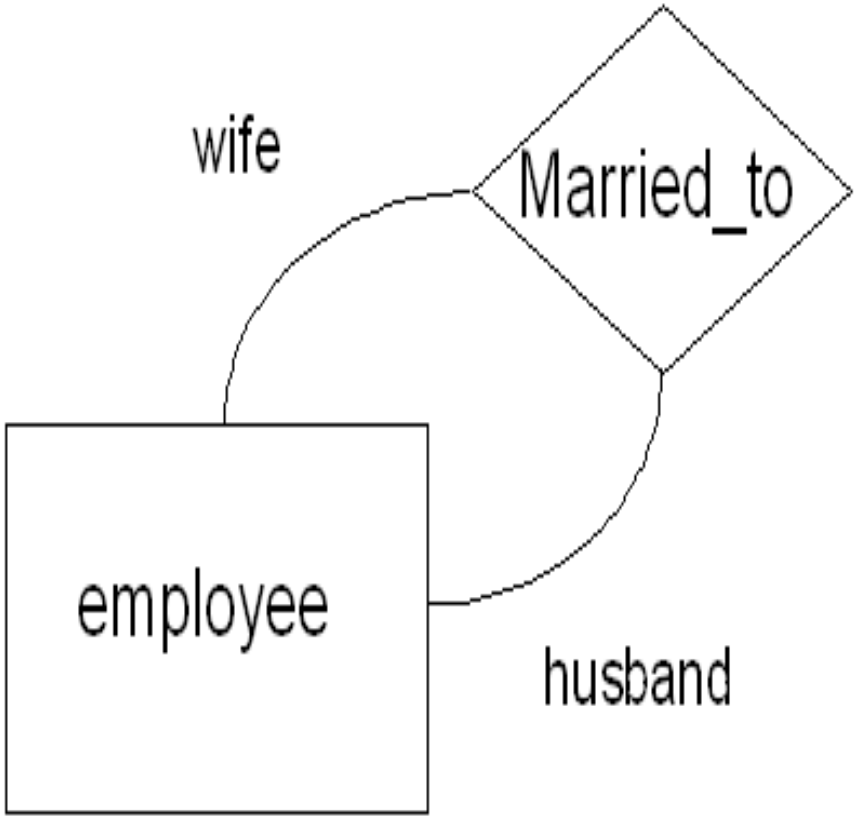
Self referencing 1:1

- Consider employees who are also a couple

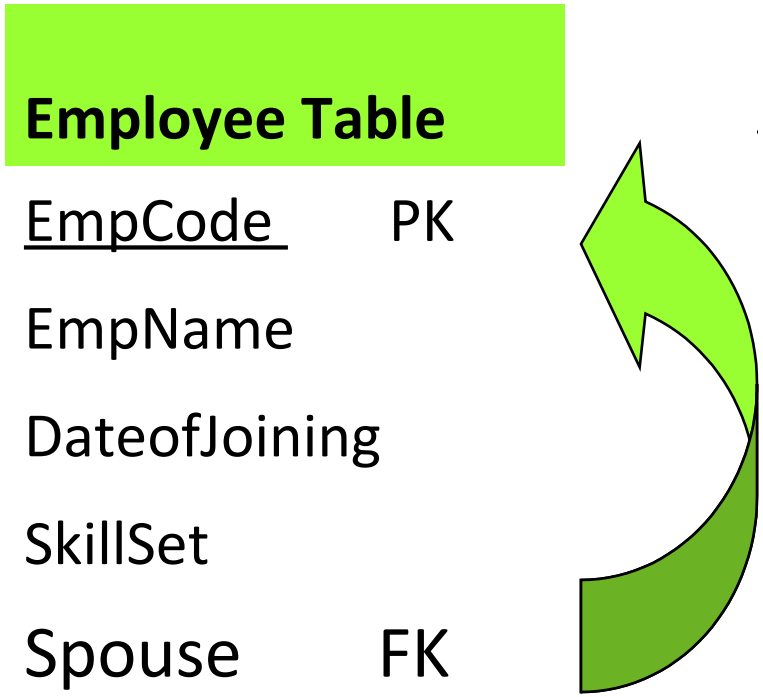
- The primary key field itself will become foreign key in the same table

Employee(E#, Name,... **Spouse**)



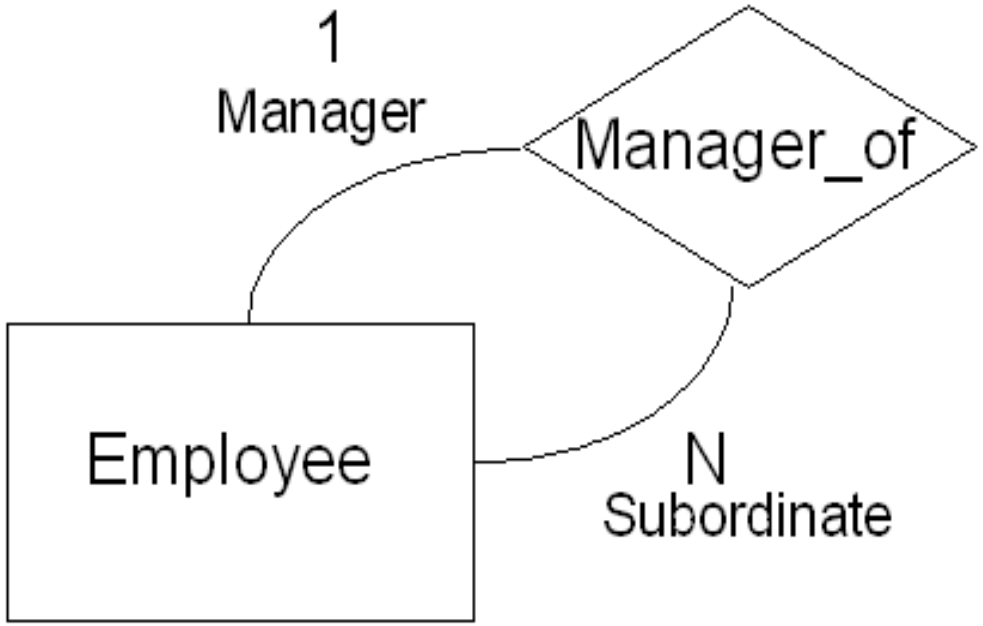


Self referencing 1 : 1



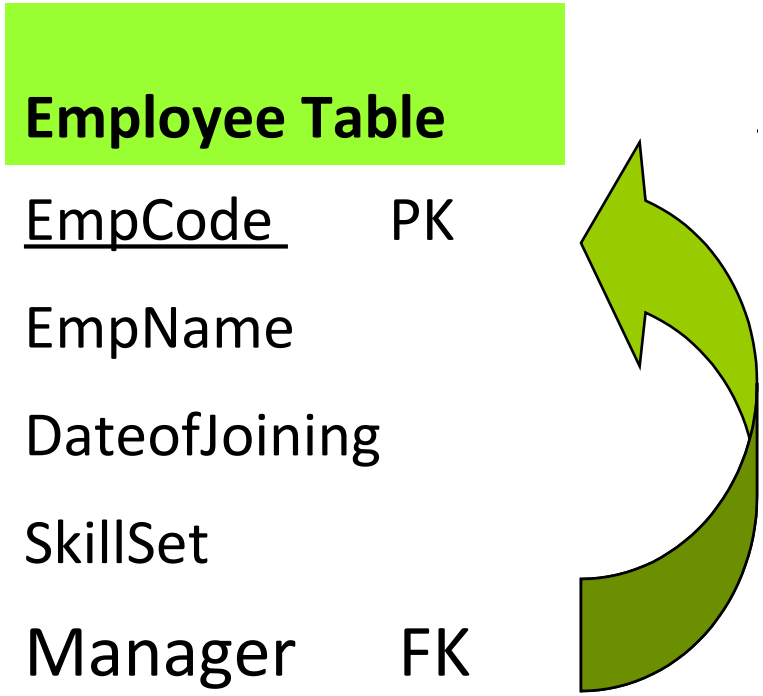
Self referencing 1:N

- The primary key field itself will become foreign key in the same table
- Same as unary 1:1

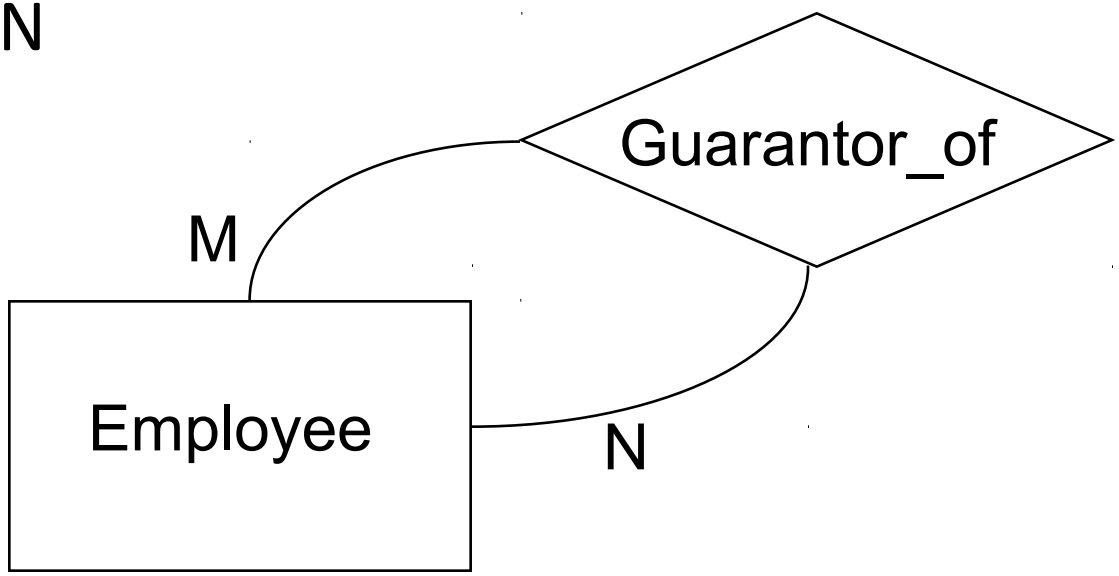


Employee(E#, Name,…,**Manager**)

Self referencing 1 : N



Self referencing M:N

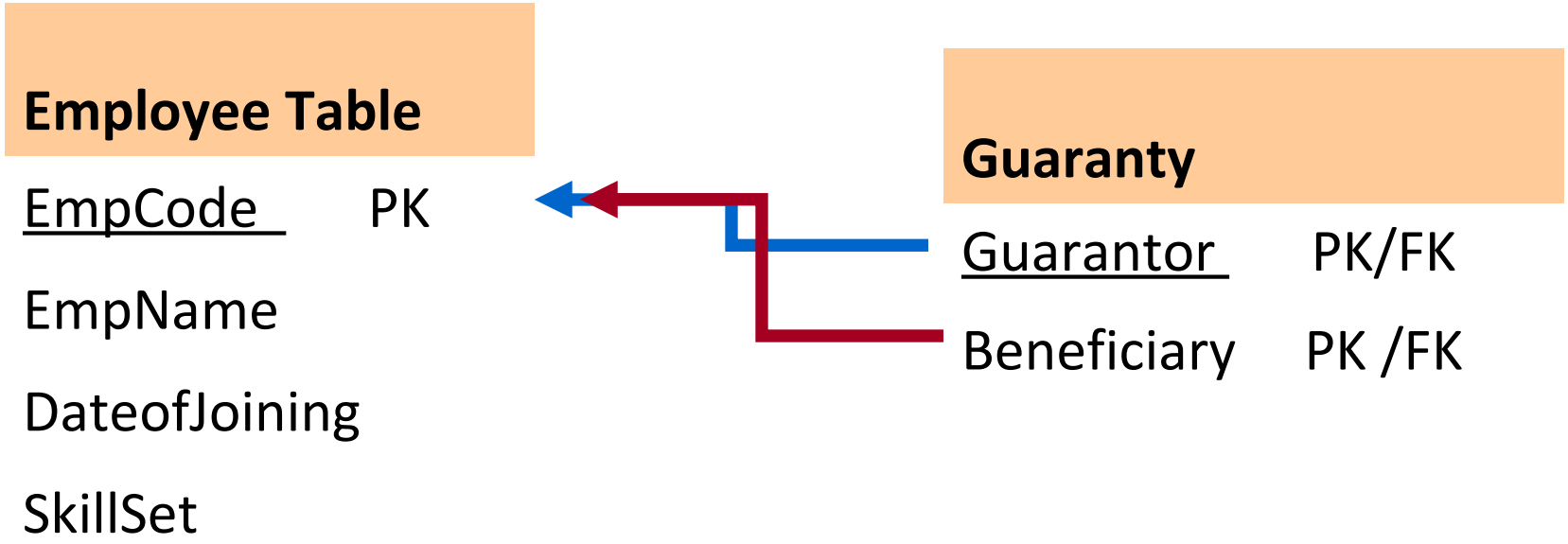


- There will be two resulting tables. One to represent the entity and another to represent the M:N relationship as follows

Employee(E#, Name,…)

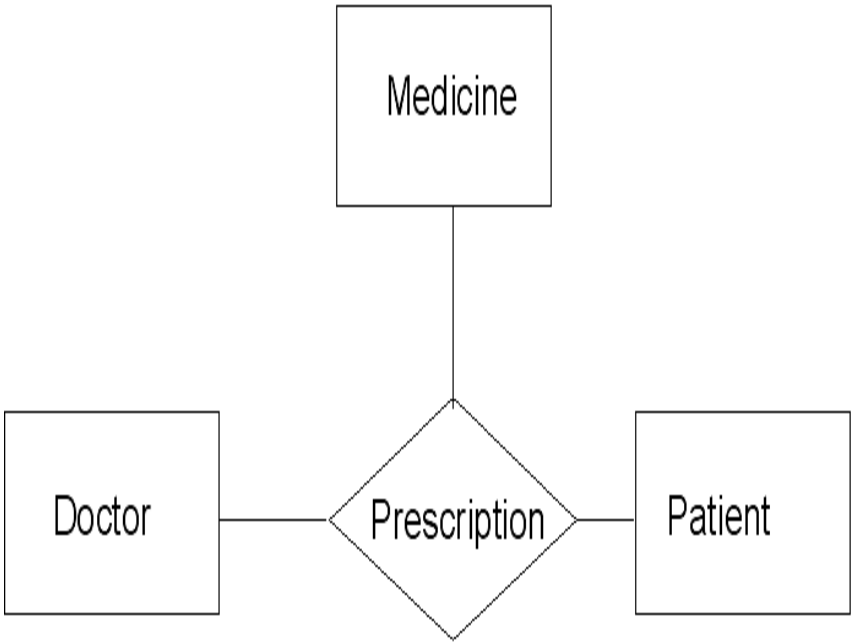
Guaranty(Guarantor, beneficiary)

Self referncing M : N

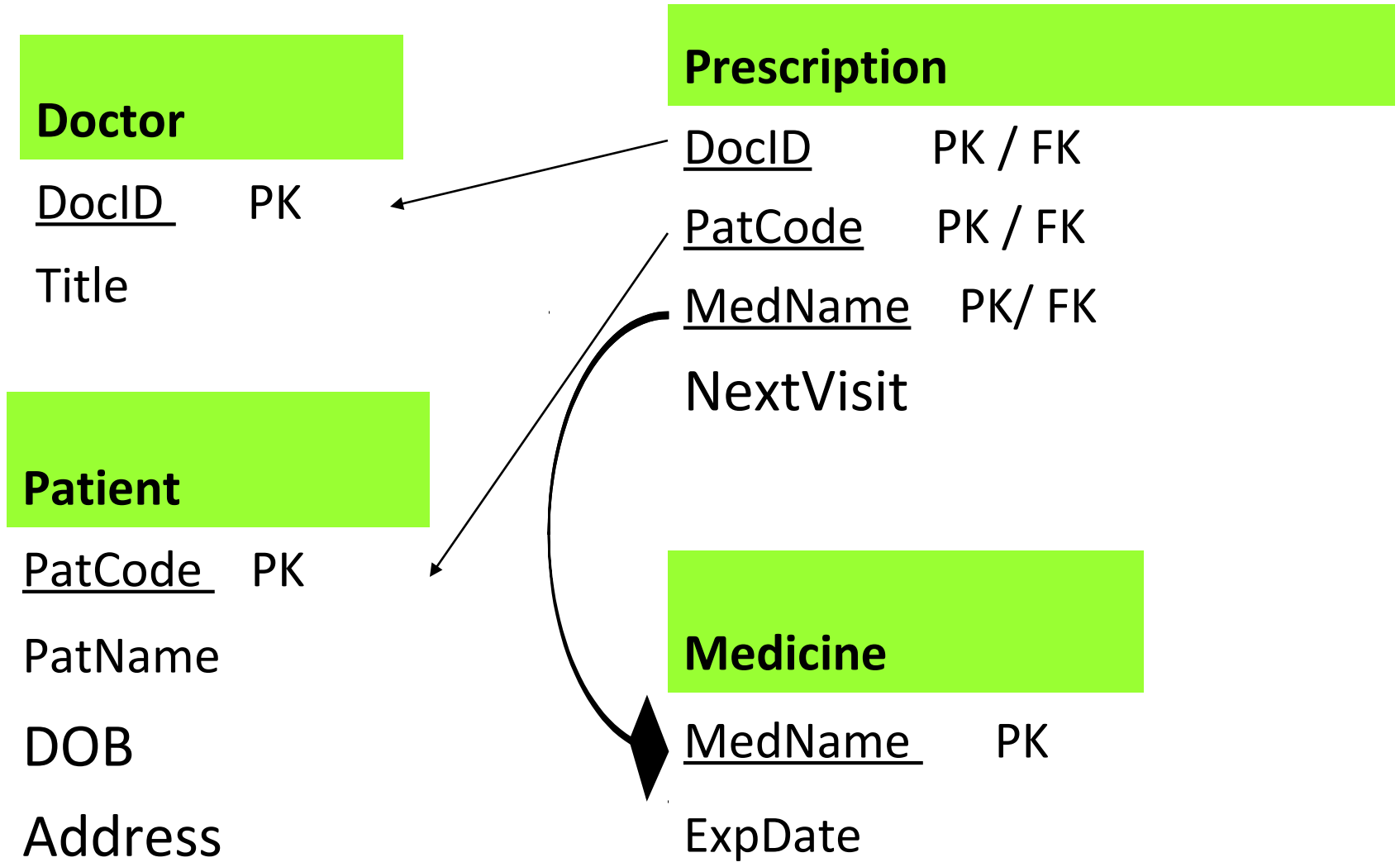


Ternary relationship

- Represented by a new table
- The new table contains three foreign keys - one from each of the participating Entities
- The primary key of the new table is the combination of all three foreign keys
- Prescription (Doctor#, Patient #, Medicine_Name)



Ternary


Doctor
DocID PK
Title
Prescription
DocID PK / FK
PatCode PK / FK
MedName PK/ FK
NextVisit
Patient
PatCode PK
PatName
DOB
Address
Medicine
MedName PK
ExpDate