

Database Design & Applications

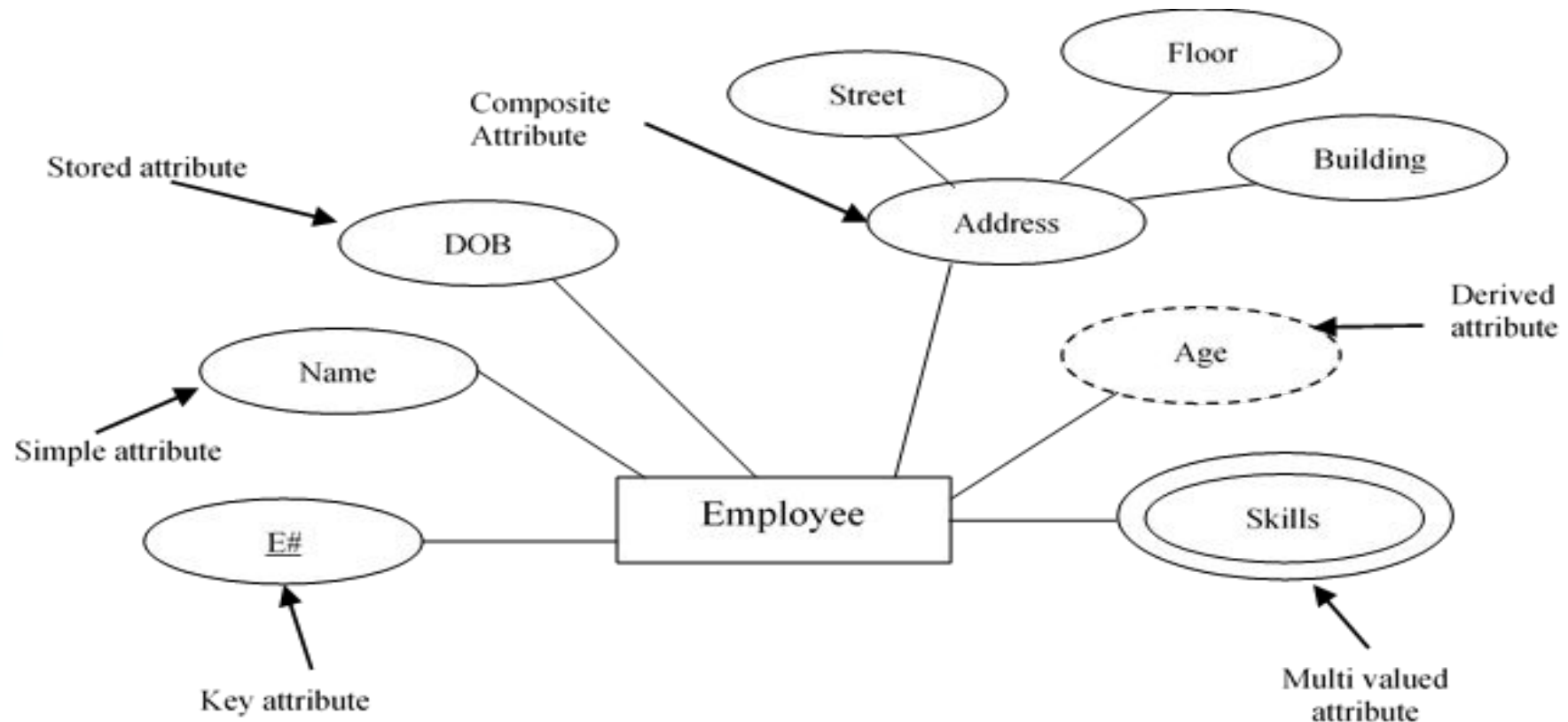
Conversion of an E-R Schema to Tables

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 entity type becomes the primary key of the table



Attributes



Converting Strong Entity Types

- The final Relational Schema:

Employee (E#, Name, Street, Floor, Building, Date_Of_Joining)

Emp_Skillset(E#, Skills)

Employee

E#

PK

Name

Street

Floor

Building

DOJ

Emp_Skillset

E#

PK/FK

Skills

PK



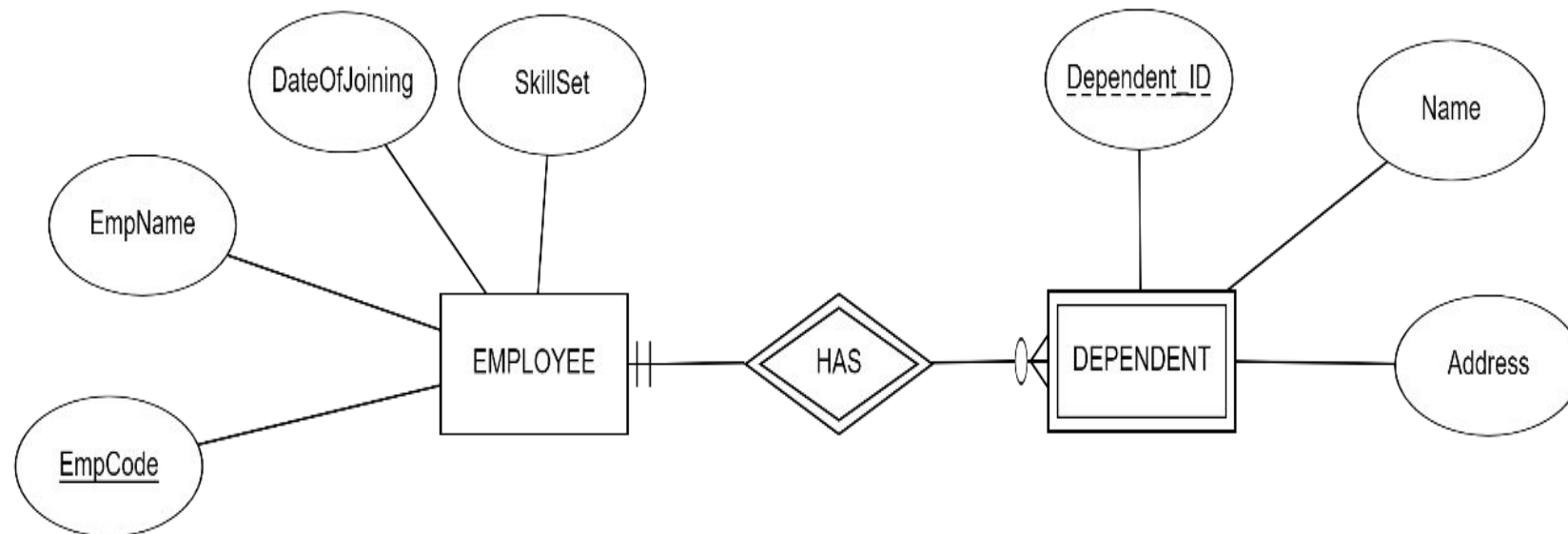
Converting Strong Entity Types

As per the rules:

- Derived attributes are ignored
- Composite attributes are represented by components
- Multi-valued attributes are represented by a separate table
- The primary key of Employee table will become the foreign key in Emp_SkillSet Table

Converting Weak Entity types

- Weak entity types are converted into a table of their own
- The primary key of the strong entity act 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

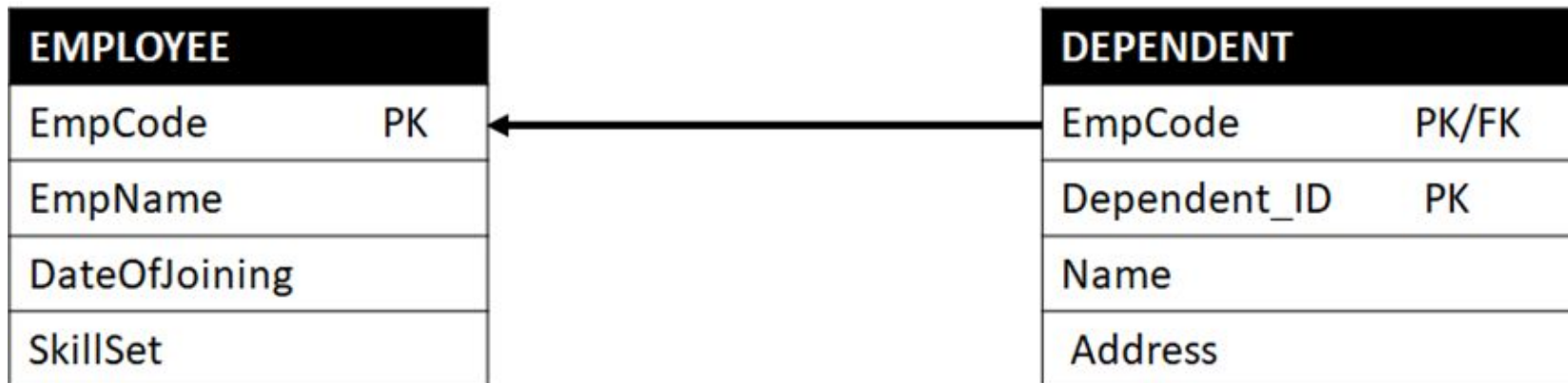


Converting Weak Entity types

- The Relational Schema:

Employee (EmpCode, EmpName, DateOfJoining, SkillSet)

Dependent (EmpCode, Dependant_ID, Name, Address)



Converting Relationships

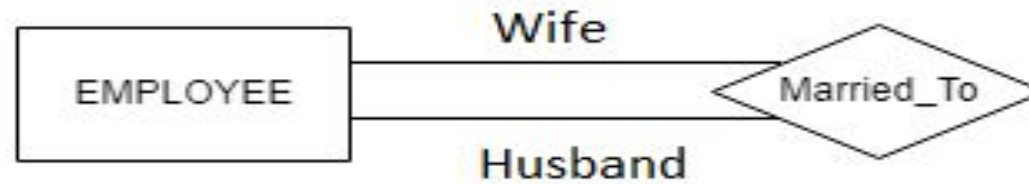
- The representation of relationships depends on the cardinality and the degree of the relationship.
- The possible cardinalities are:
 - 1:1
 - 1:M
 - M:N
- The degrees are:
 - Unary
 - Binary
 - Ternary

Unary (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)



Unary (1:1)



- Conversion of Unary (1:1) Relationship to table

Employee Table	
<u>EmpCode</u>	PK
EmpName	
DateofJoining	
SkillSet	
Spouse	FK



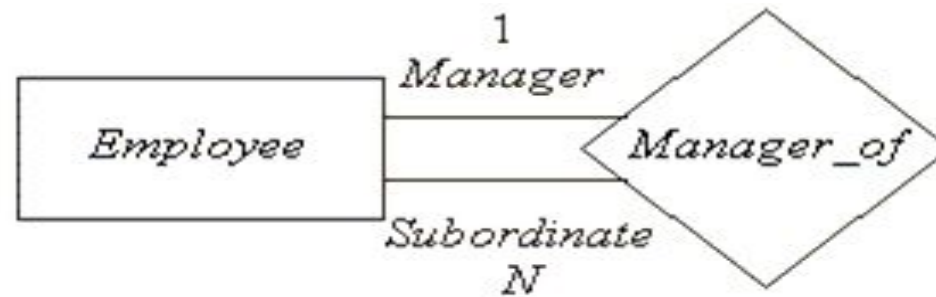
Unary (1:N)

- Consider there are multiple employees, and few employees among them work as a manager.
- The primary key field itself will become foreign key in the same table

Employee (E#, Name, . . . Manager)



Unary (1:N)



- Conversion of Unary (1:N) Relationship to table

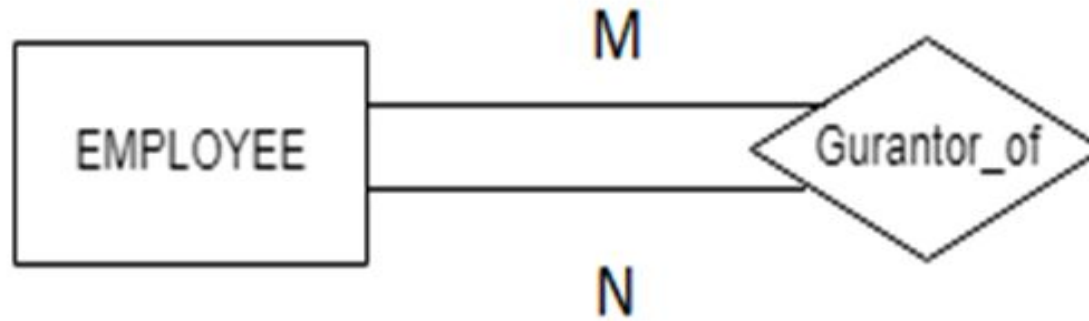
Employee Table	
<u>EmpCode</u>	PK
EmpName	
DateofJoining	
SkillSet	
Manager	FK



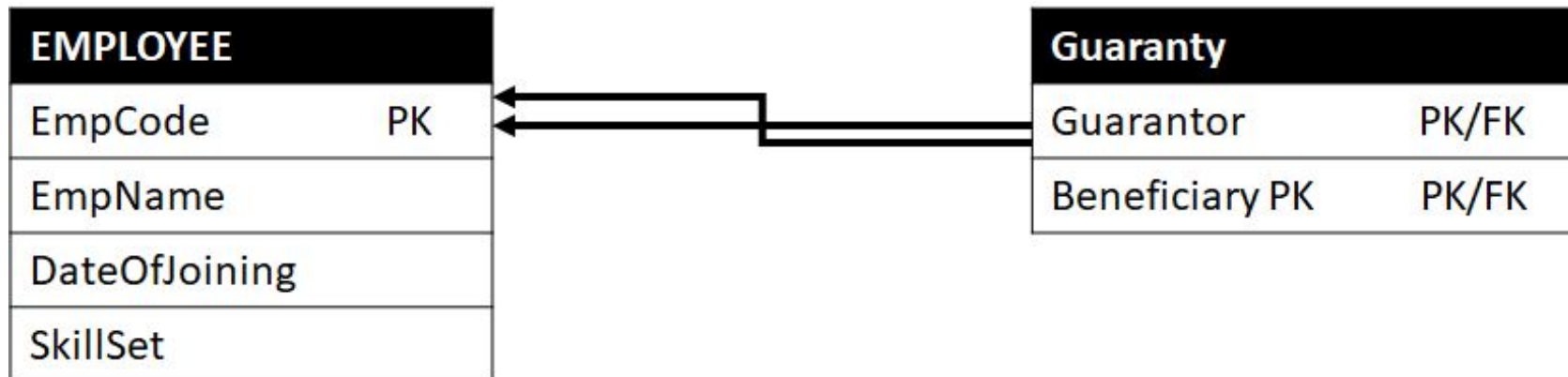
Unary (M:N)

- Consider there are multiple employees, and few employees gives guarantee for other employees.
- One employee can give guarantee for many employees
- A employee can take the guarantee from many employees.
- There will be two resulting tables:
 - One to represent the entity
 - Another to represent the many to many relationship

Unary (M:N)

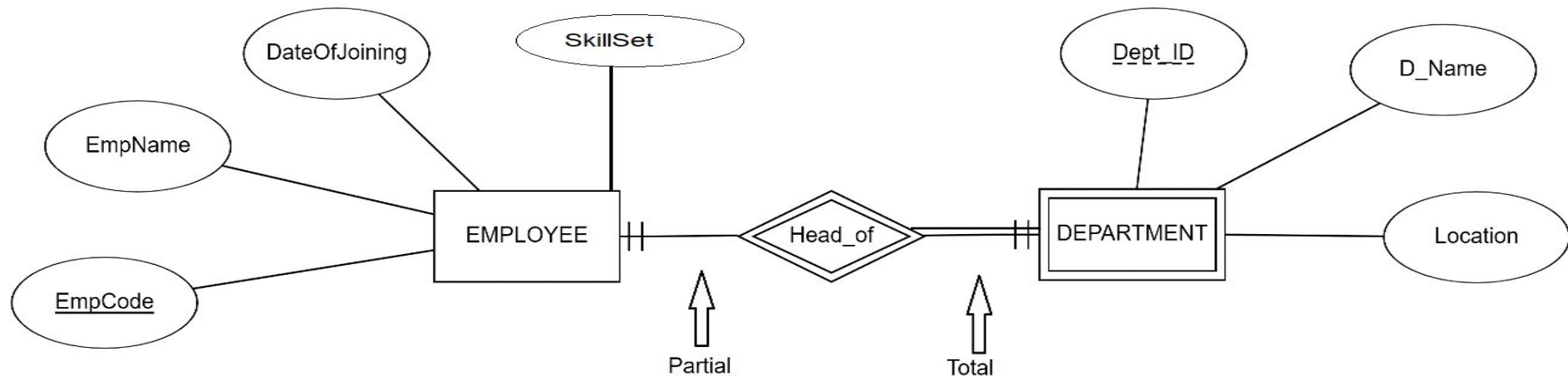


- Conversion of Unary (M:N) Relationship to tables



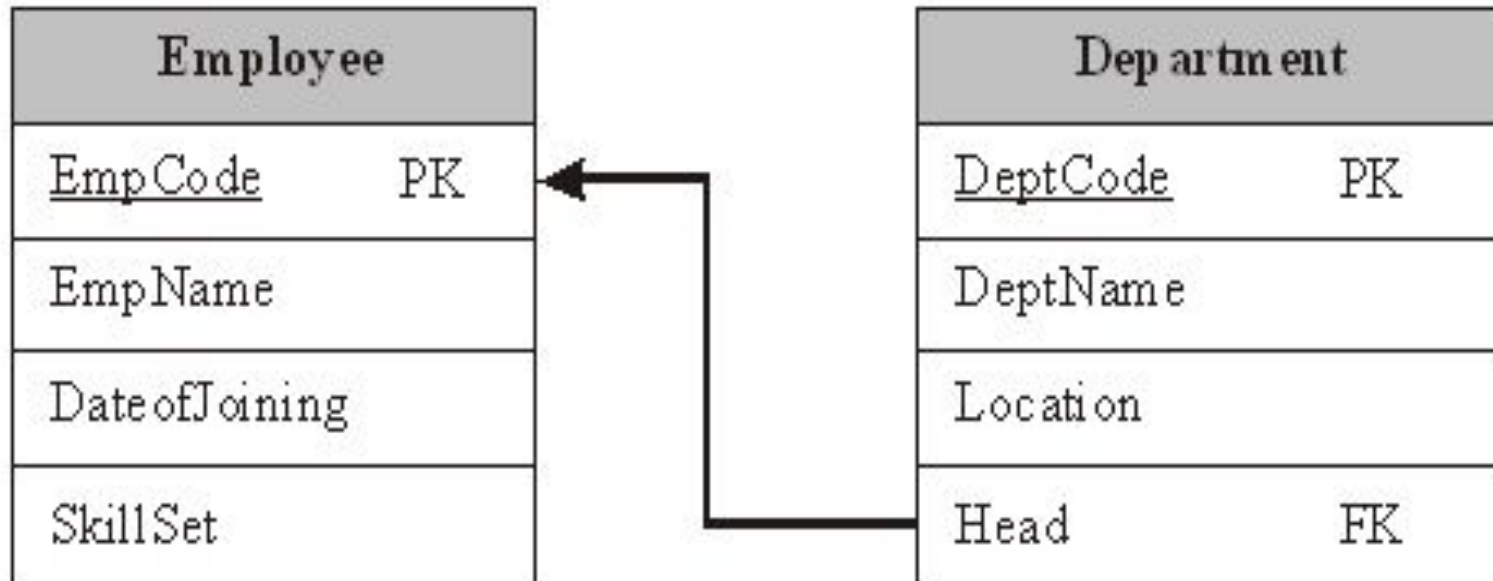
Binary 1:1

- **Combination of Participation types:** In binary (1:1) relationship if combination of participation exists, the primary key of the partial participation will become the foreign key of the total participation.
- **Conversion of total and partial participation to tables**



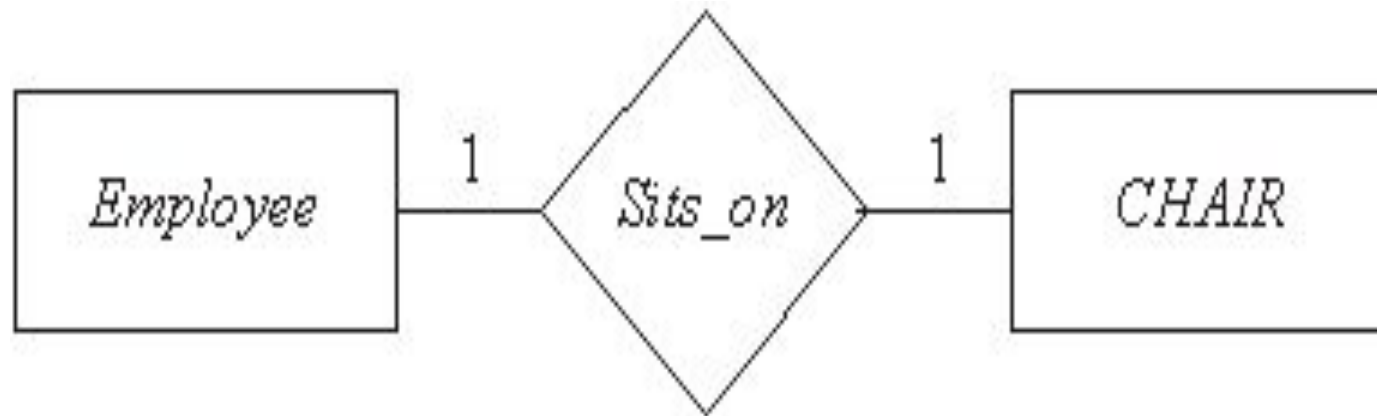
Binary 1:1

- Conversion of total and partial participation to tables



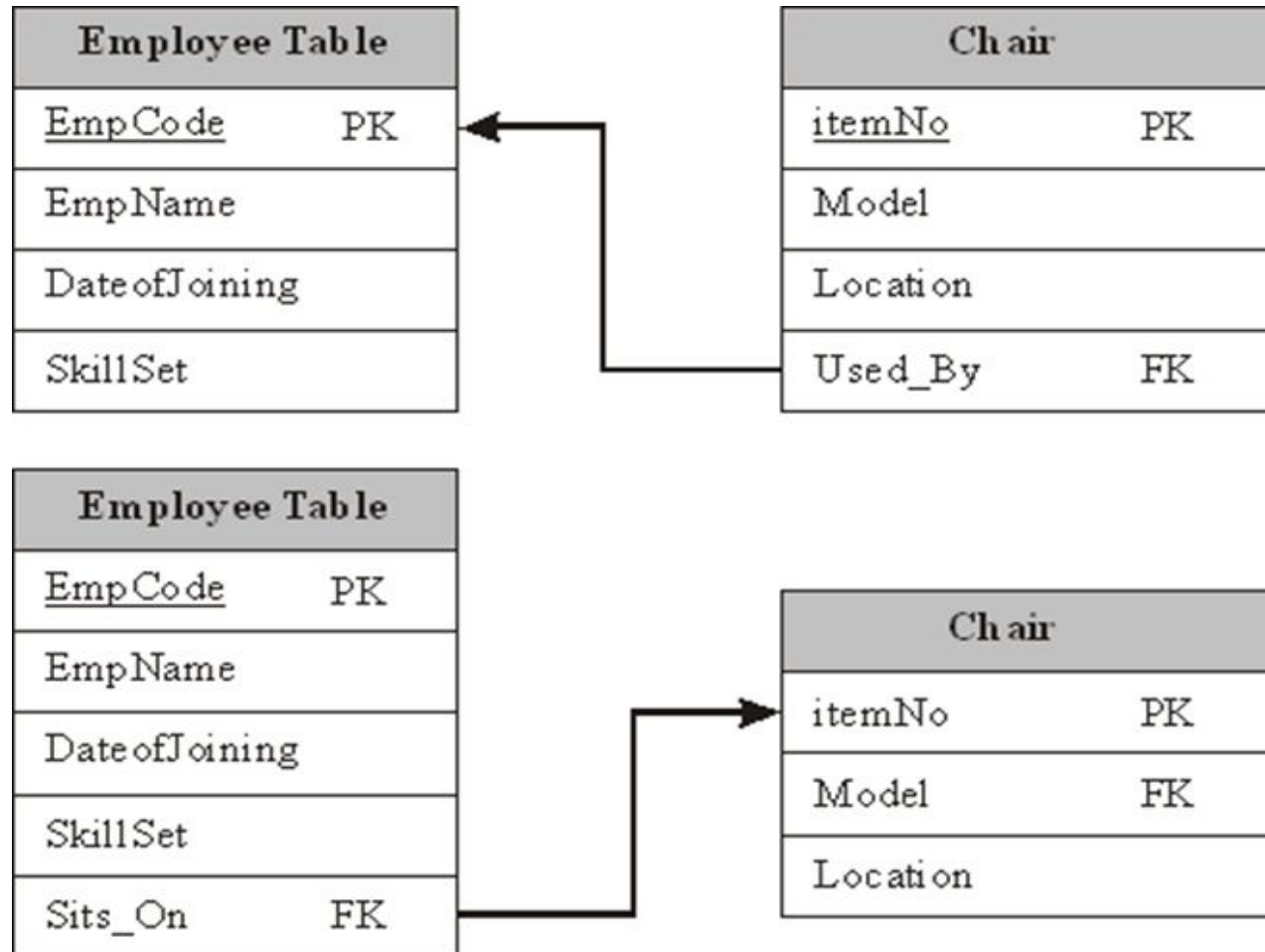
Binary 1:1

- **Uniform Participation type:** If both the entities take uniform participation in the Binary (1:1) relationship the primary key of either of the participants can become a foreign key in the other.



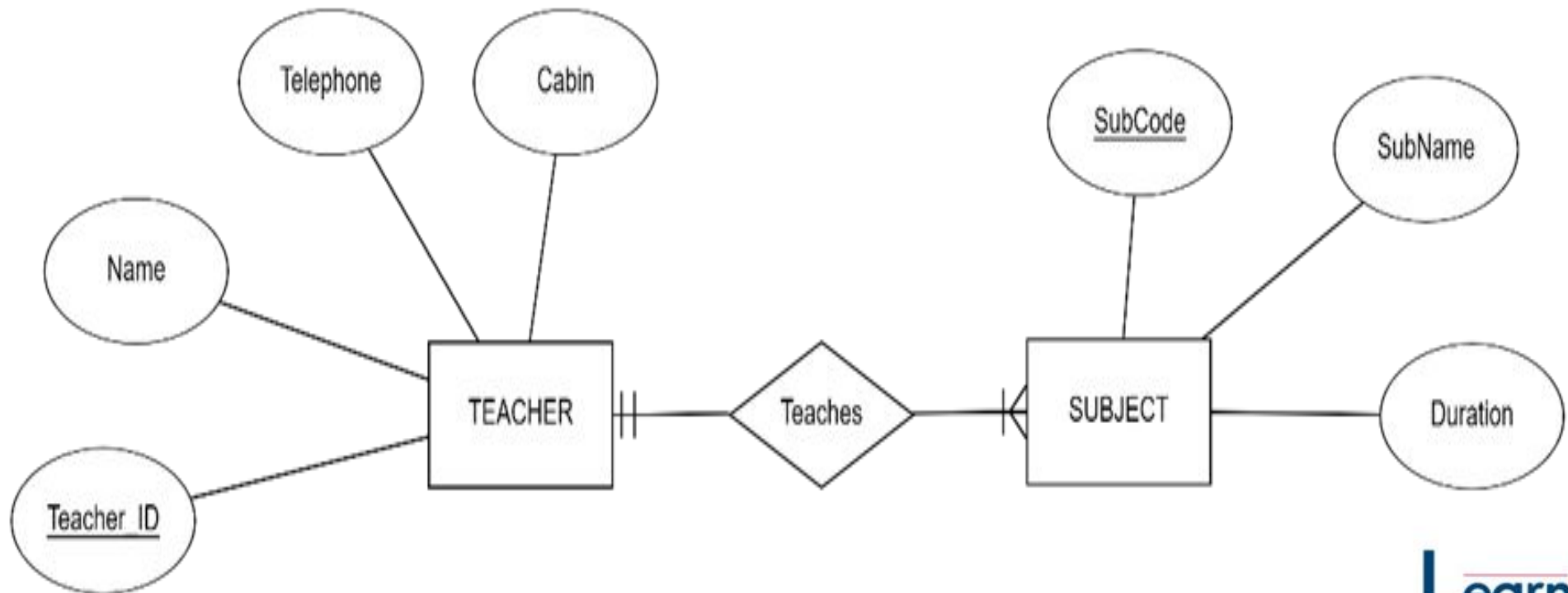
Binary 1:1

- Conversion of Binary (1:1) Uniform participation into tables



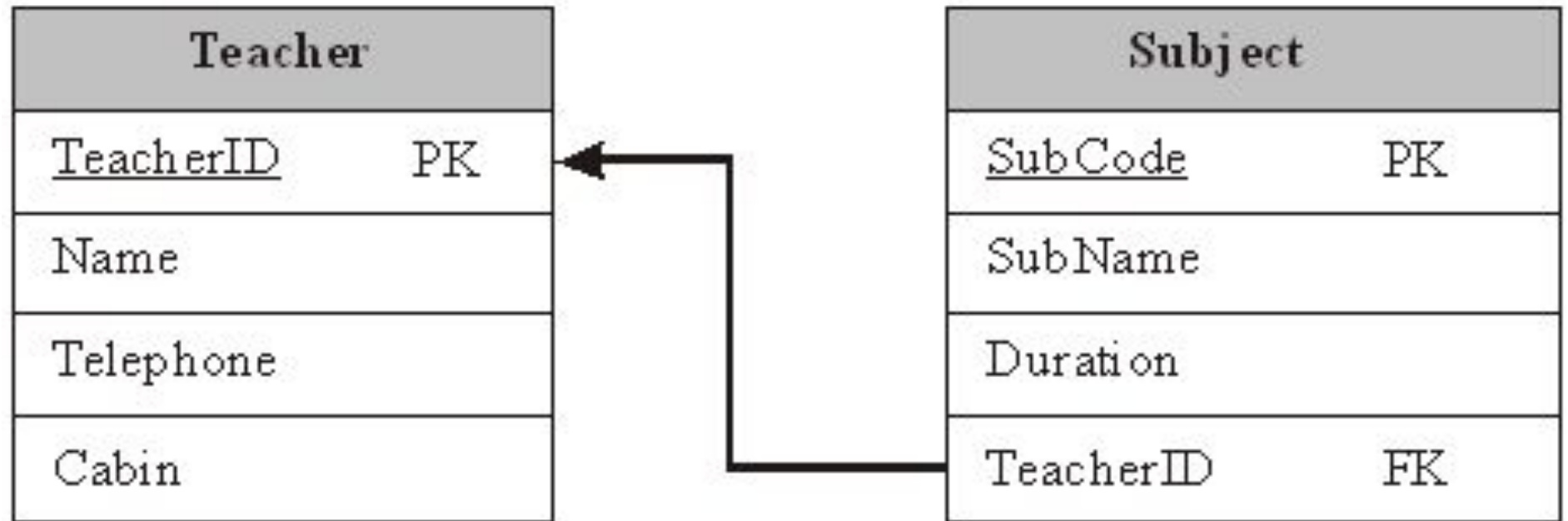
Binary 1: N

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



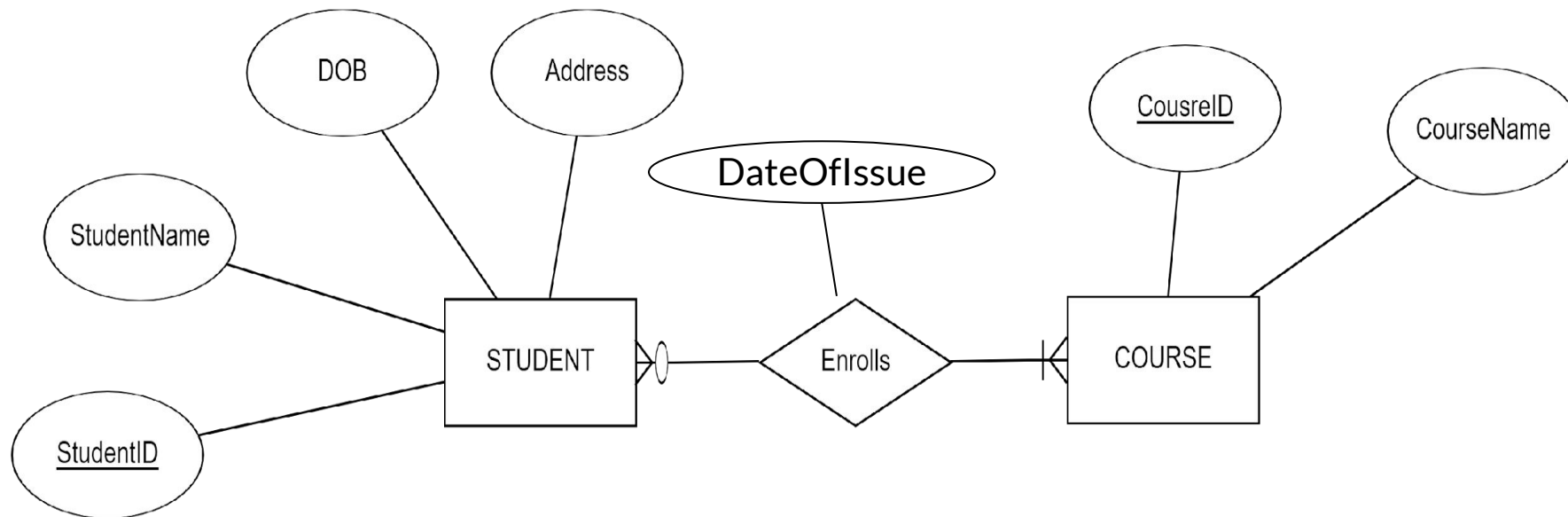
Binary 1: N

- Conversion of binary (1:N) relationship to tables



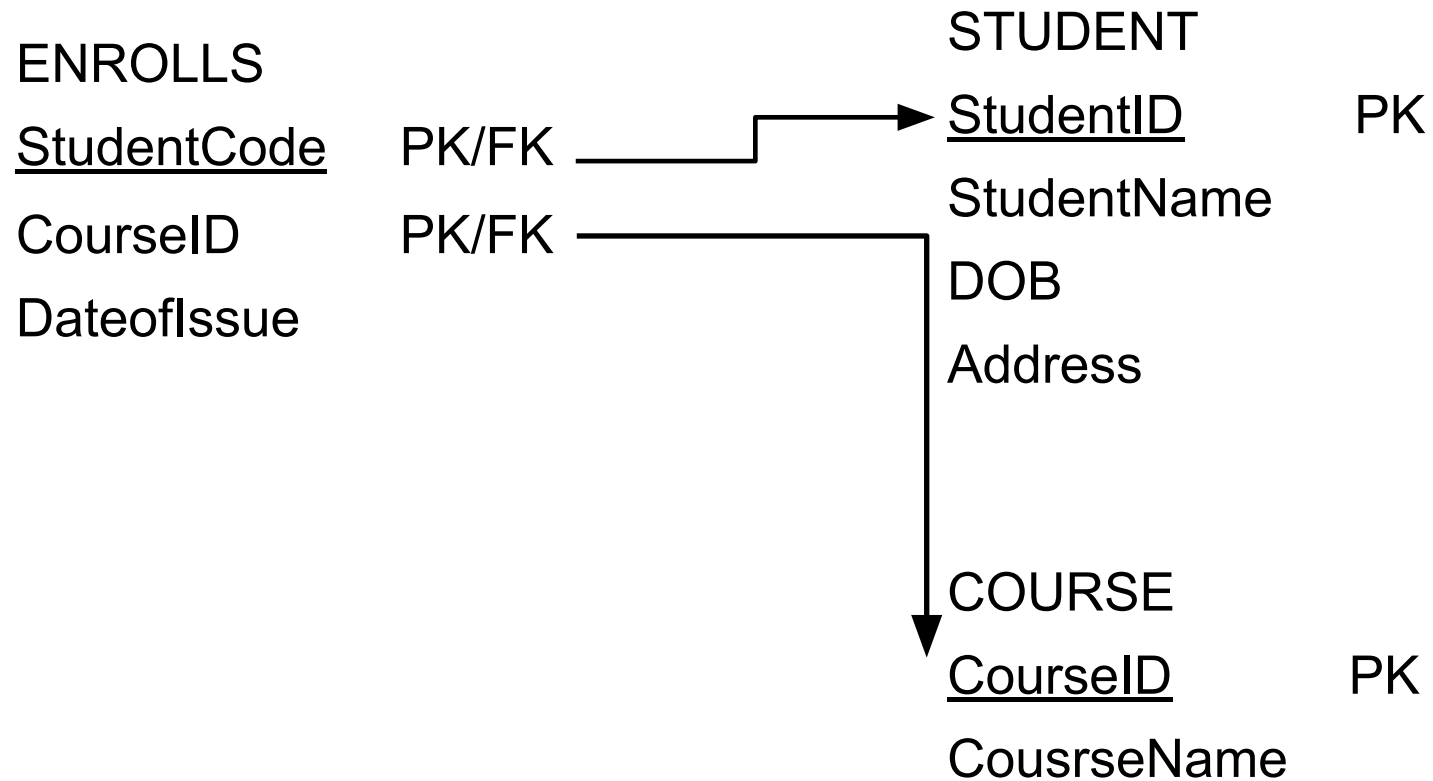
Binary M: N

- A new table is created to represent relationship
- This new table contains two foreign keys – one from each of participating entities
- The primary key of new table is the combination of two foreign keys



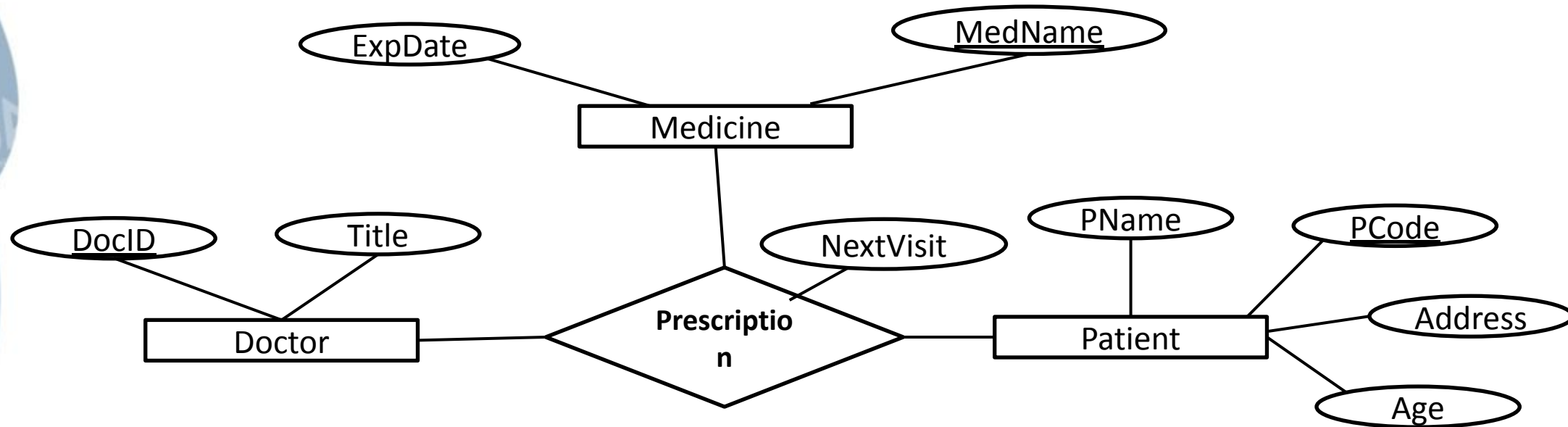
Binary M: N

- Conversion of Binary (M:N) relationship to tables

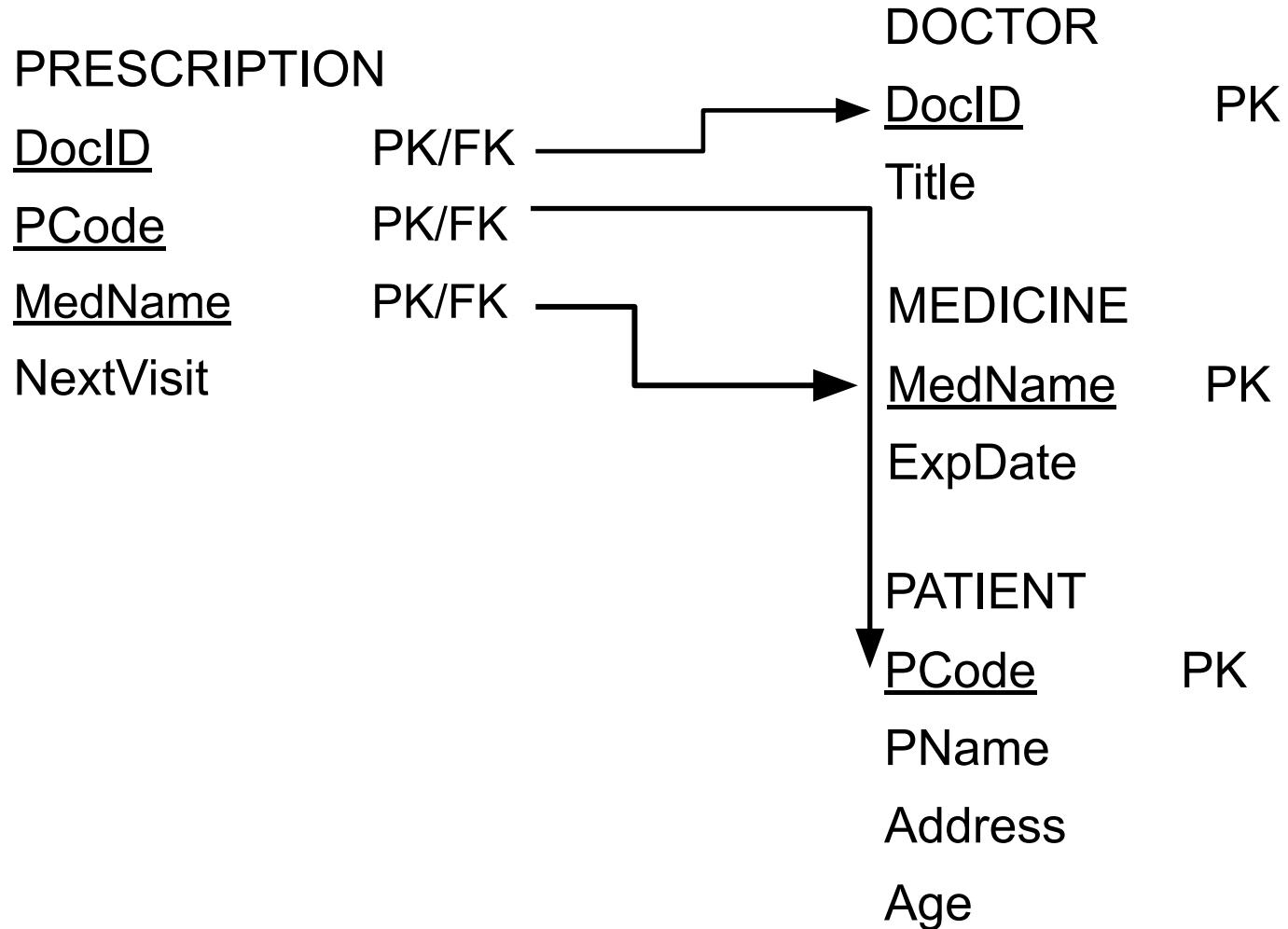


Ternary Relationship

- A new table is created to represent relationship
- The new table contains three foreign keys- one from each of the participating entities
- The primary key of new table is the combination of all three foreign keys



Ternary Relationship



Converting Bank ERD into Tables

Assumptions :

There are multiple banks and each bank has many branches.

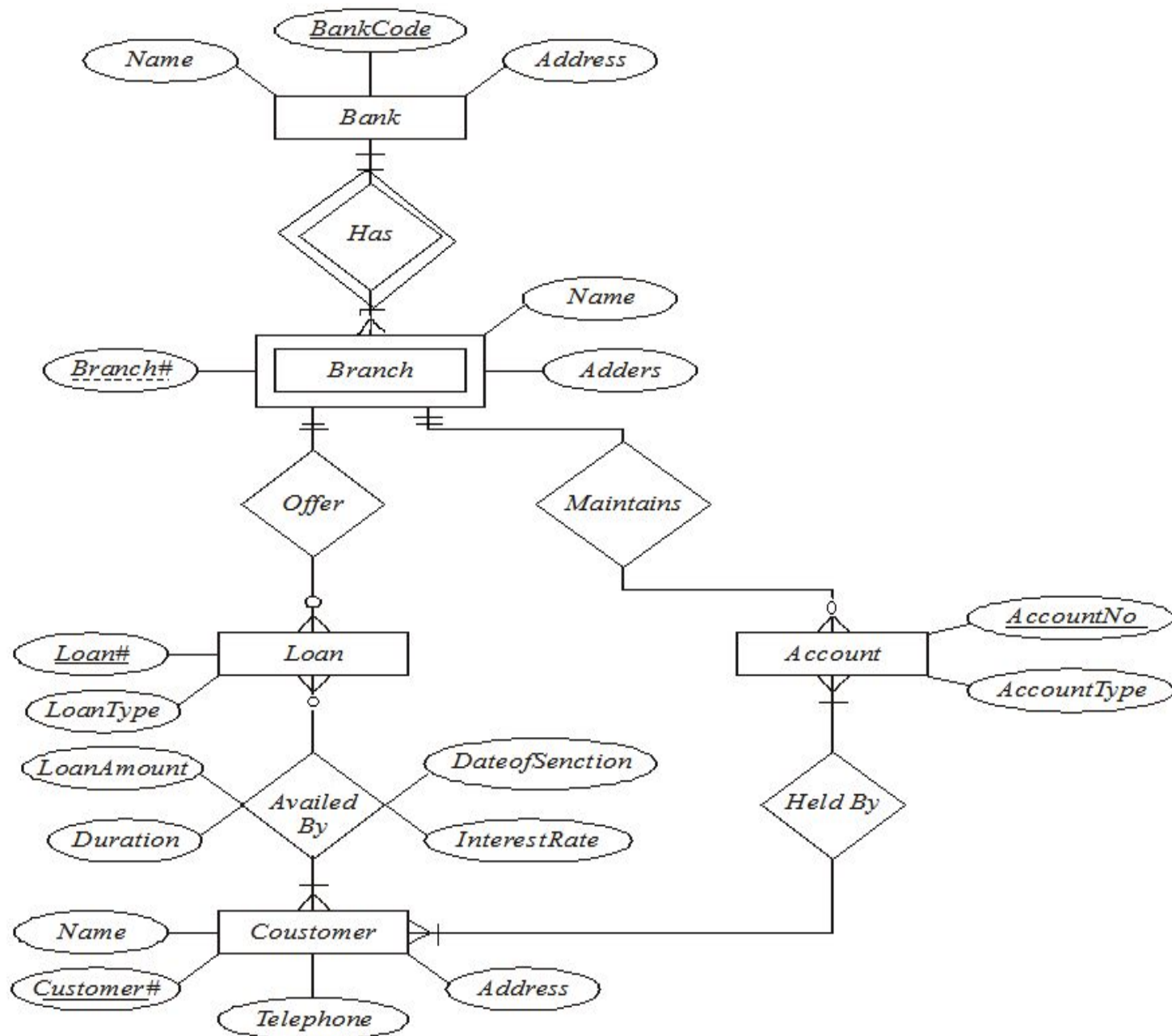
Each branch has multiple customers

Customers have various types of accounts in any branch

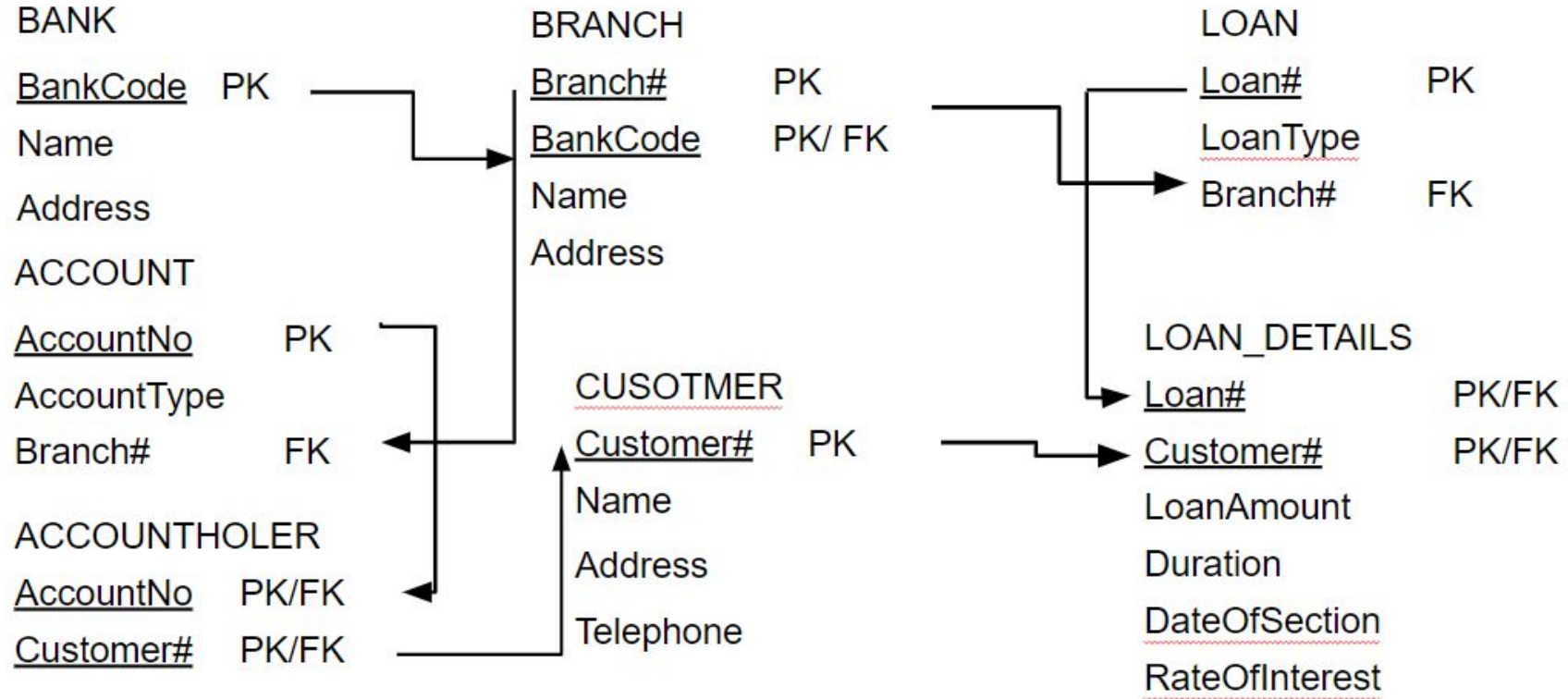
Some Customers also had taken different types of loans from these bank branches

One customer can have multiple accounts and Loans

Bank ERD



Final Bank Schema



Thank You!

