

# UCS310

# Database Management System

## E-R to Relational Mapping

Lecture-15

Date: 16 Feb 2023

Dr. Sumit Sharma  
Assistant Professor

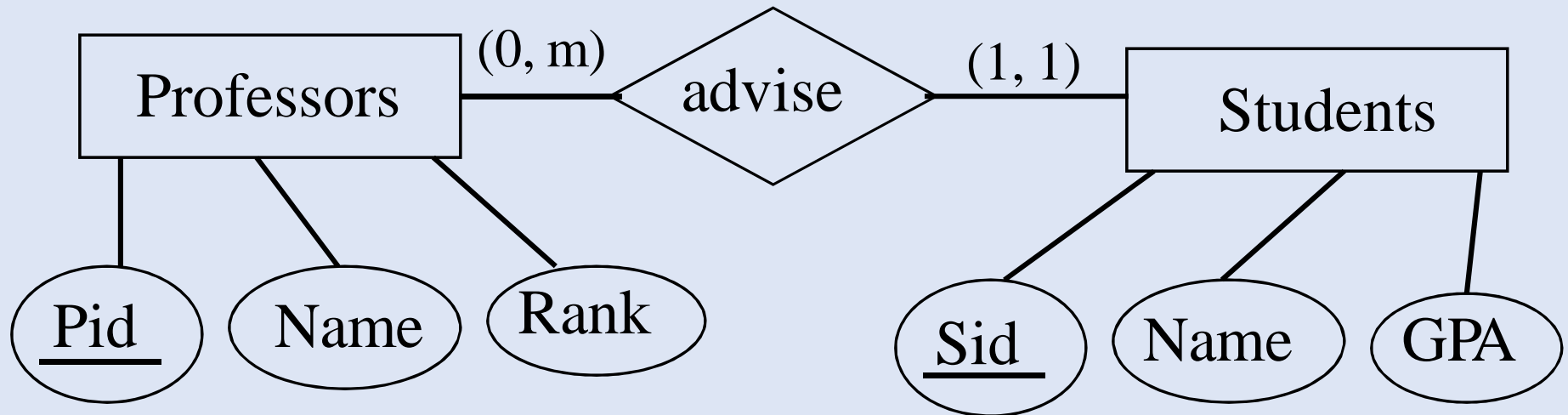
Computer Science and Engineering Department  
Thapar Institute of Engineering and Technology, Patiala

# Recap

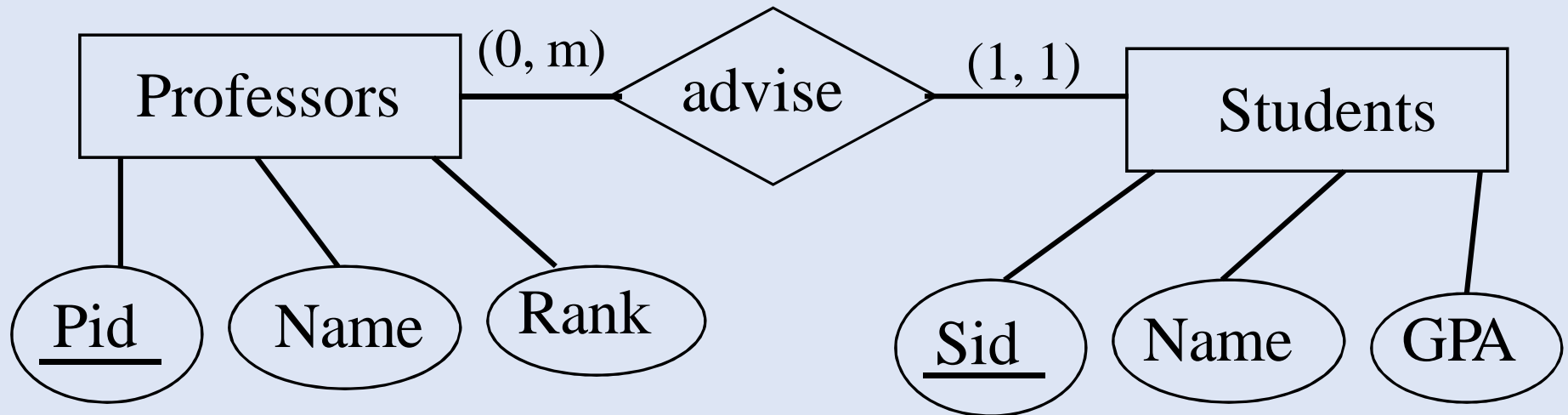
- Aggregation
- Steps for E-R model design
- Relation mapping
  - Simple attribute, composite attribute
  - Strong entity
  - Weak entity
  - 1:1, 1:M binary relations

# ER to Relational Mapping

# ER to Relational Mapping: Example



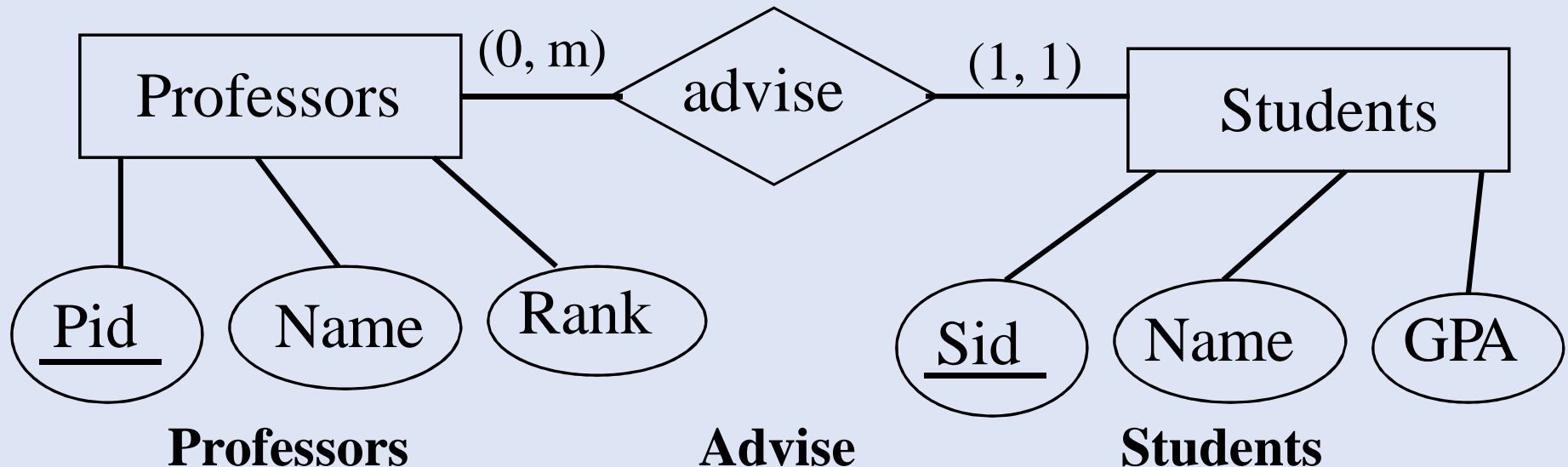
# ER to Relational Mapping: Example



Professors	Advise	Students
p1: 123, Jack, Prof.	p1 advises s1	s1: 456, John, 3.4
p2: 234, Ann, Prof.	p1 advises s2	s2: 567, Carl, 3.2
p3: 345, Bob, Prof.	p3 advises s3	s3: 678, Ken, 3.5

# ER to Relational Mapping: Example

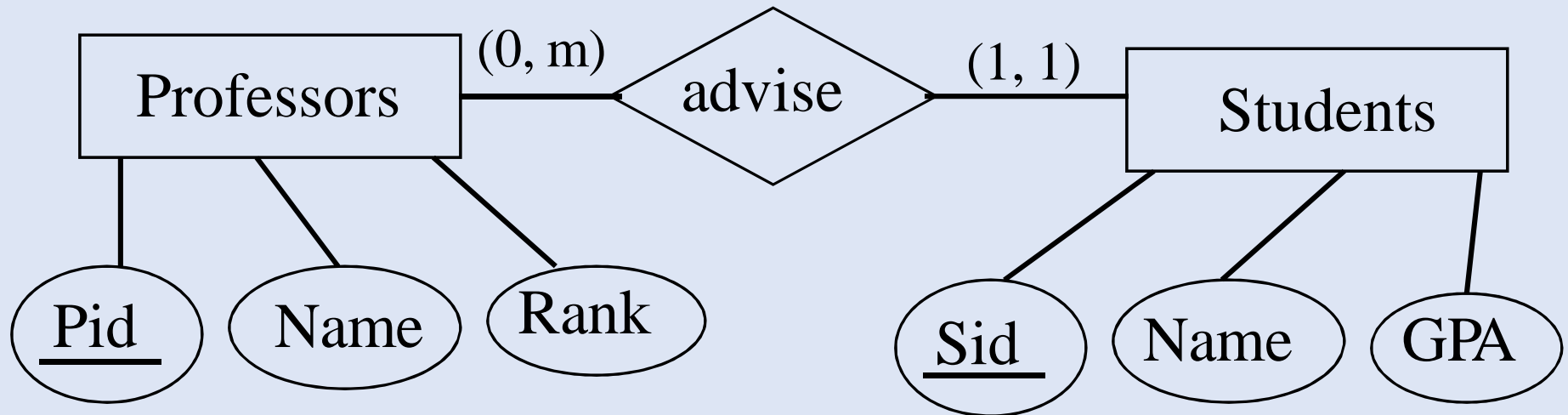
Transform the ER diagram into three relations:



Pid	Name	Rank	Pid	Sid	Sid	Name	GPA
123	Jack	Prof.	123	456	456	John	3.4
234	Ann	Prof.	123	567	567	Carl	3.2
345	Bob	Prof.	345	678	678	Ken	3.5

# ER to Relational Mapping: Example

Two relations are sufficient:



**Professors**

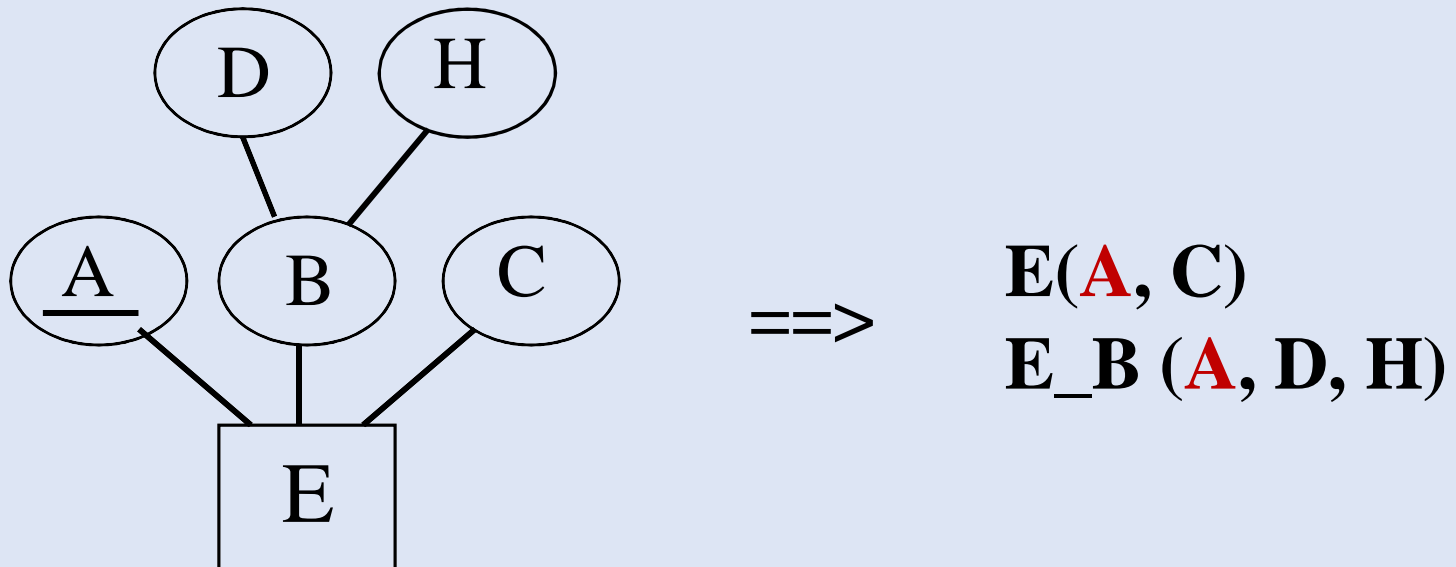
<b>Pid</b>	<b>Name</b>	<b>Rank</b>
123	Jack	Prof.
234	Ann	Prof.
345	Bob	Prof.

**Students**

<b>Sid</b>	<b>Name</b>	<b>GPA</b>	<b>Pid</b>
456	John	3.4	123
567	Carl	3.2	123
678	Ken	3.5	345

# Mapping Composite Attribute

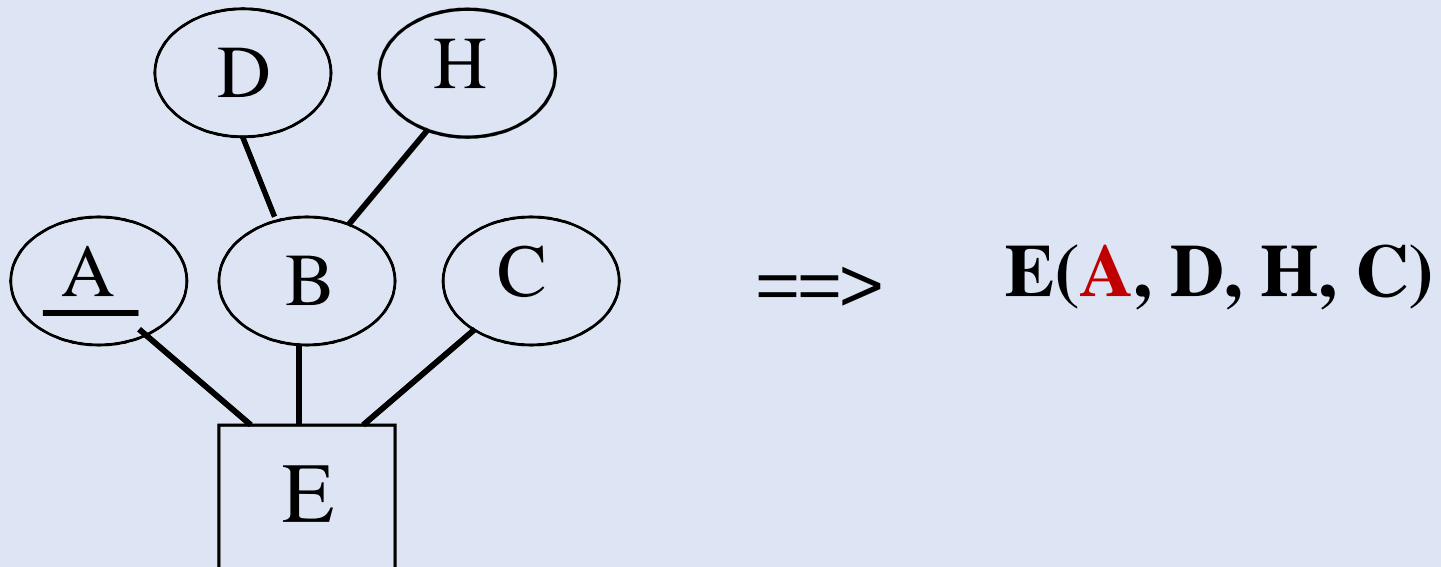
Method 1: Transform the composite attribute to a separate relation





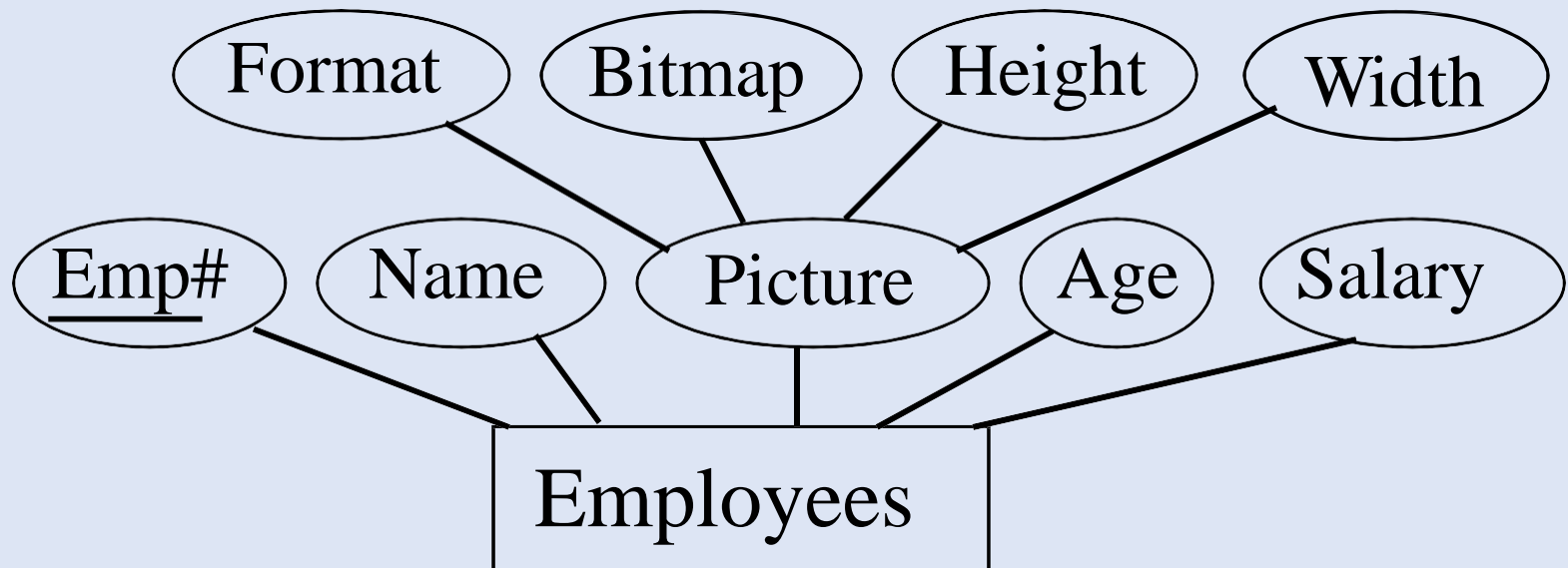
# Mapping Composite Attribute

Method 2: Use only simple attributes and ignore the composite attribute.



# Mapping Composite Attribute: Example

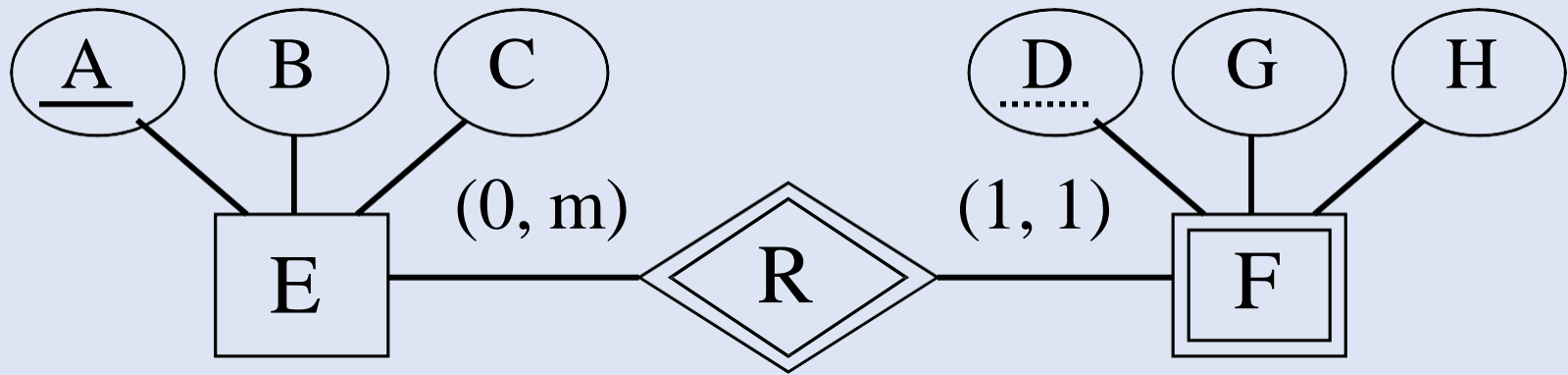
Method 1:



Employees (**Emp#**, Name, Age, Salary)

Emp\_Pic (**Emp#**, Bitmap, Format, Height, Width)

# Weak Entity Mapping



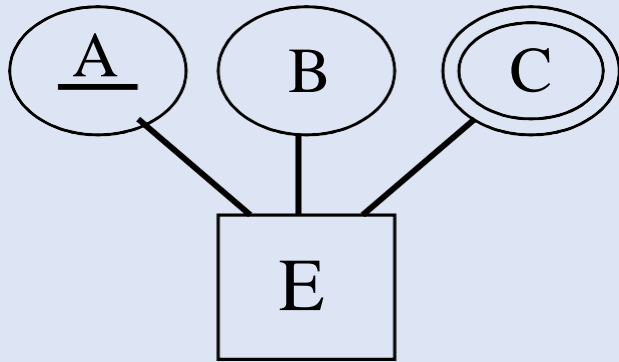
**E (A, B, C),    F(A, D, G, H)**

- The key of **F** consists of the key of **E** and the partial key of **F**.
- **F.A** is a foreign key referencing **E.A**

# Multivalued Attribute

# Mapping Multivalued Attribute

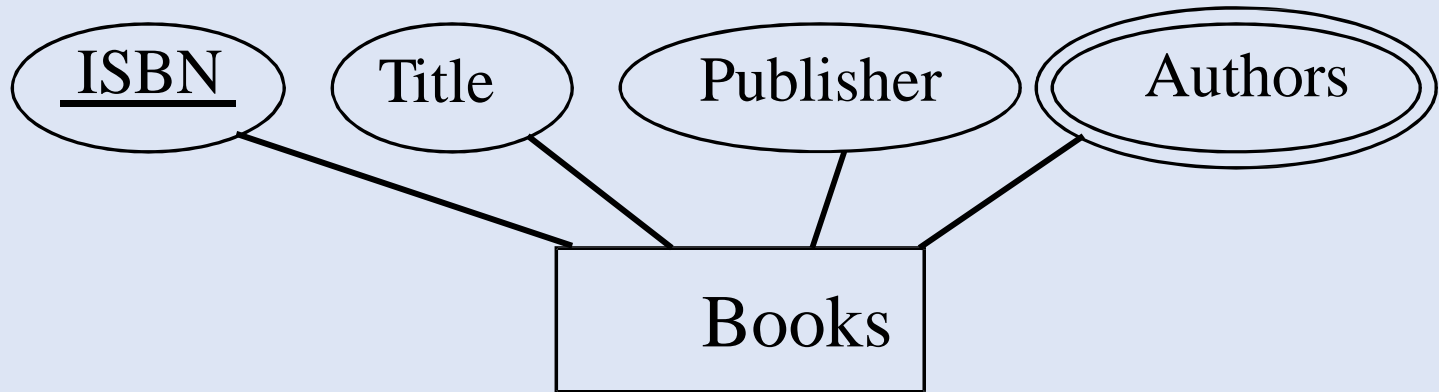
- Create a separate relation for each multi-valued attribute.



$E(\textcolor{red}{A}, B),$   
 $E\_C(\textcolor{brown}{A}, \textcolor{brown}{C})$

- $E\_C.A$  should be defined to be a foreign key referencing  $E.\textcolor{red}{A}$

# Mapping Multivalued Attribute

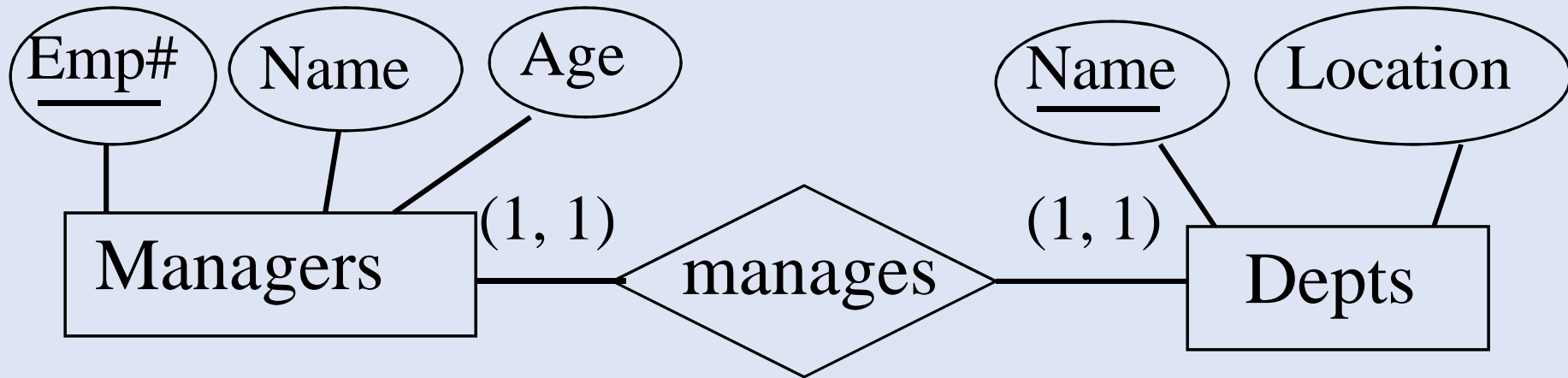


**Books** (**ISBN**, Title, Publisher)

**Book\_Authors** (**ISBN**, **Author**)

- Define **Book\_Authors.ISBN** as a foreign key referencing **Books.ISBN**

**1:1**



**Depts**(**Name**, Location)

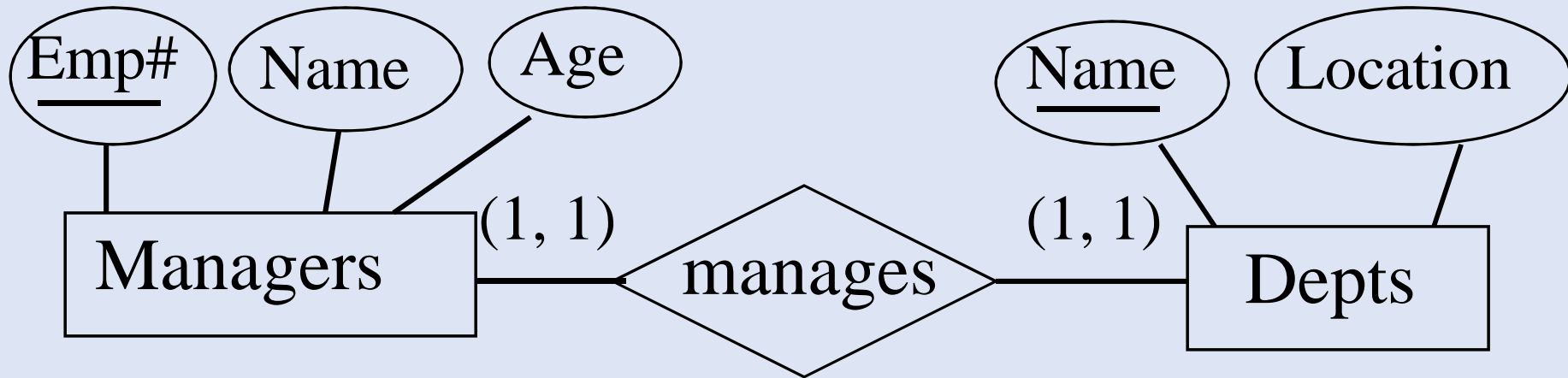
**Managers**(**Emp#**, Name, Age, **Dept\_name**)

**OR**

**Depts**(**Name**, Location, **Manager\_Emp#**)

**Managers**(**Emp#**, Name, Age)

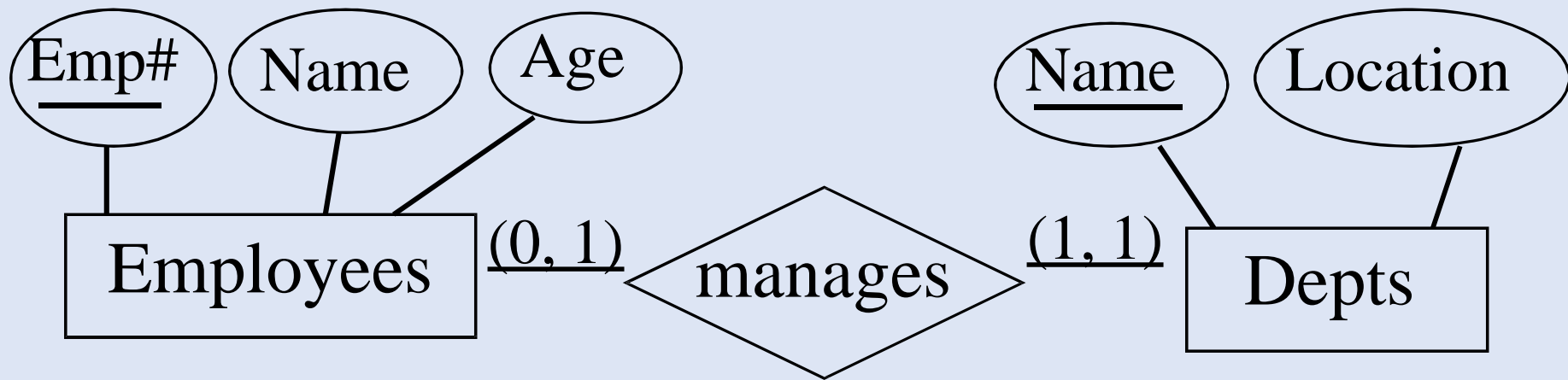
**1:1**



**Manager\_Depts(Emp#, Name, Age, DName, Location)**



**1:1**

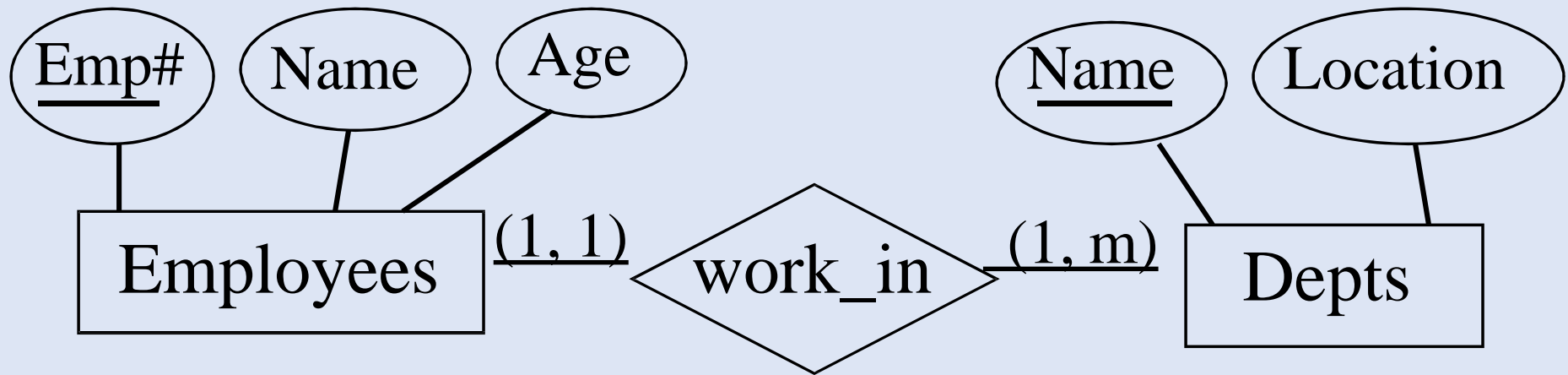


**Depts(**Name**, Location, **Manager\_Emp#**)**

**Employees(**Emp#**, Name, Age)**

- The entity set with the total participation is transformed to a relation with a foreign key.
- Why not let Employees have the foreign key?

1:M

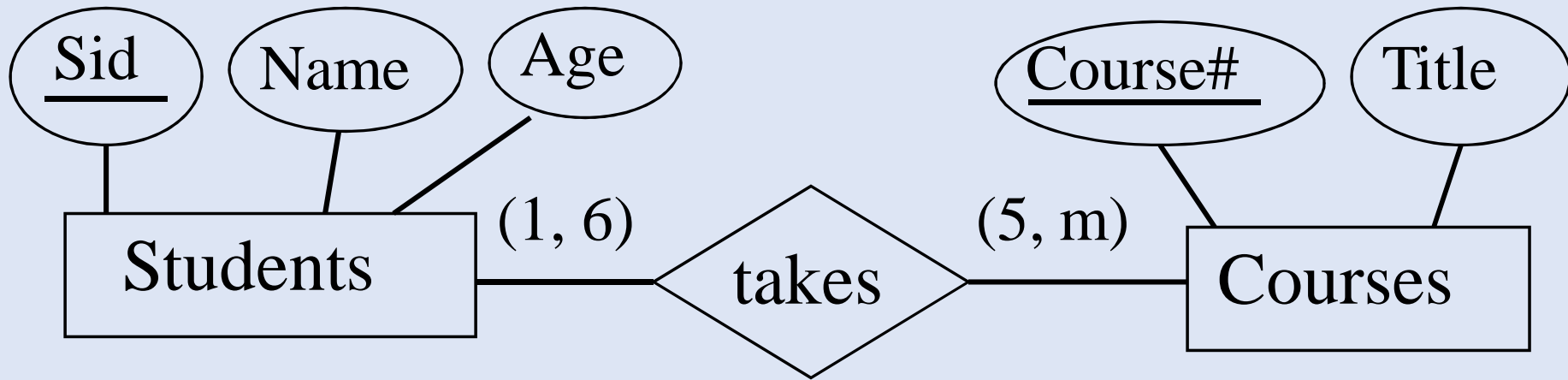


Depts(**Name**, Location)

Employees(**Emp#**, Name, Age, **Dept\_name**)

*Renaming* is useful for improving understandability.

# M:N

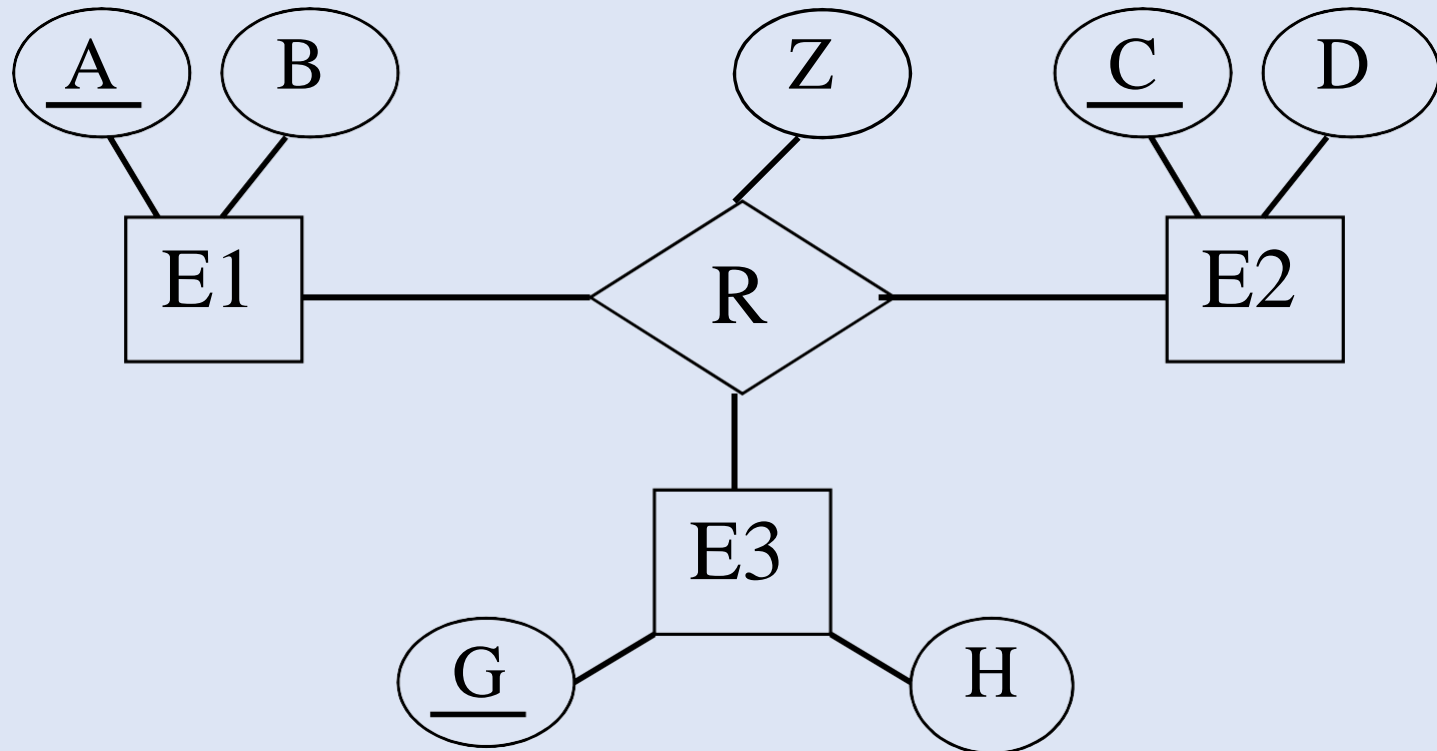


Students(**Sid**, Name, Age)

Courses(**Course#**, Title)

Takes(**Sid**, **Course#**)

# Ternary Relationship

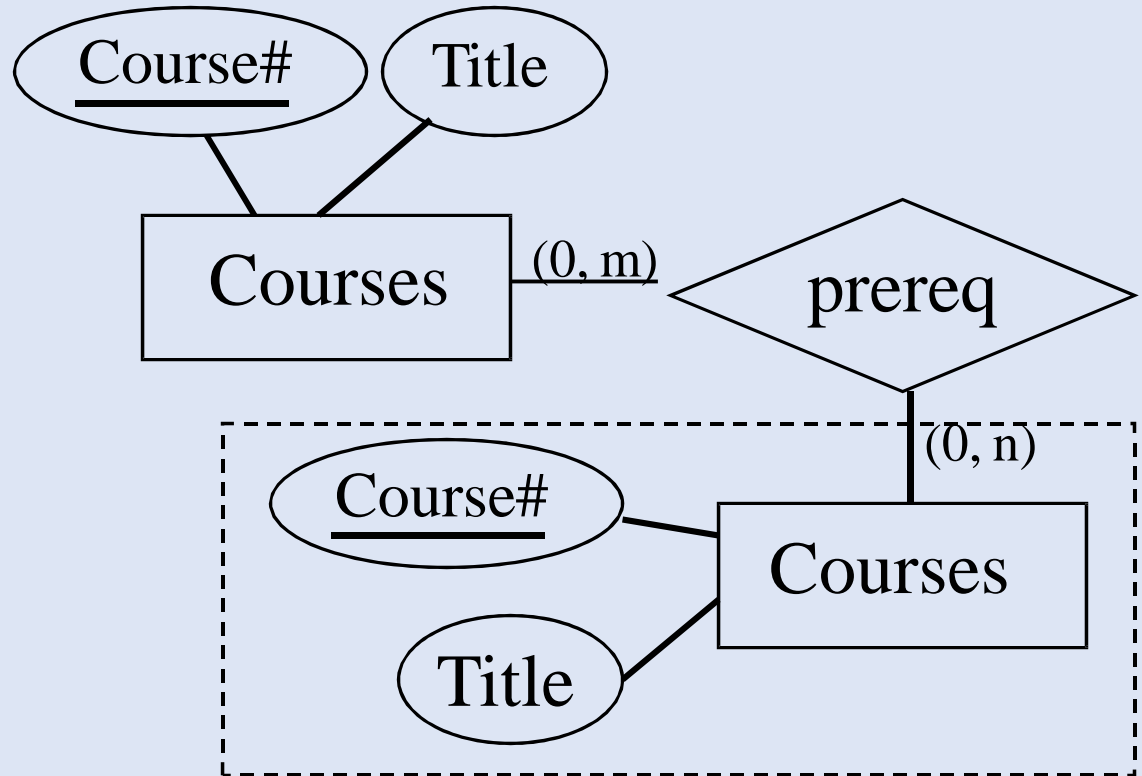
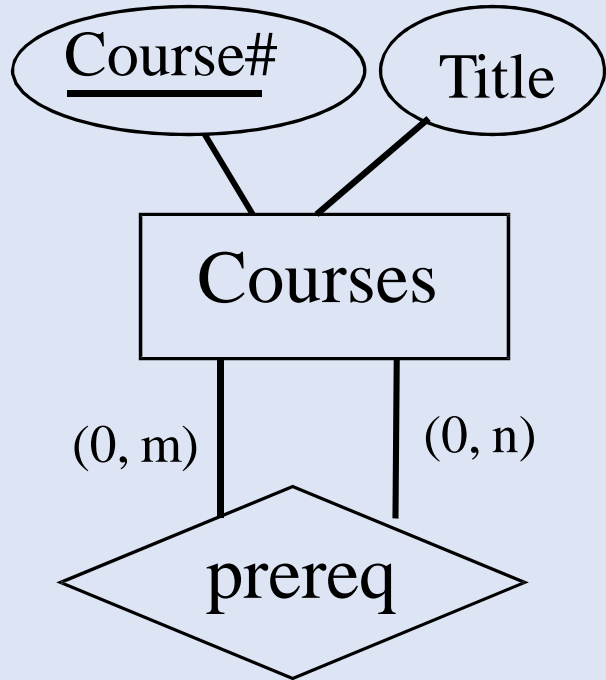


$E1(\textcolor{red}{A}, B), \quad E2(\textcolor{red}{C}, D), \quad E3(\textcolor{red}{G}, H),$   
 $R(\textcolor{brown}{A}, \textcolor{brown}{C}, \textcolor{brown}{G}, Z)$

# Unary Relationship

- Create a **shadow** entity set and transform the unary relationship into a binary relationship
- Apply the rules for transforming binary relationships
- After the transformation, remove one redundant relation, or if there is no redundant relation, remove the relation with fewer attributes

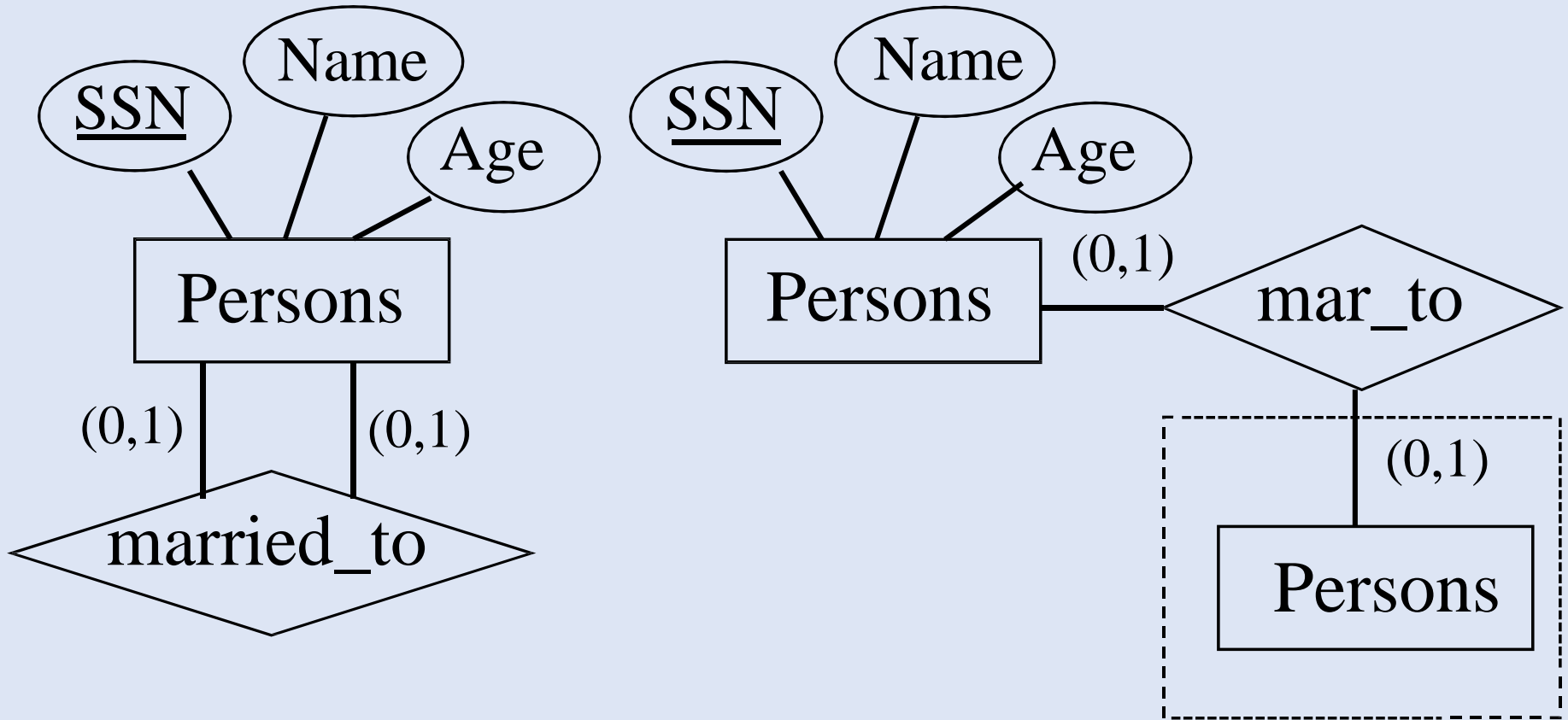
# Unary Relationship Example



**Courses**(**Course#**, Title)

**Prereq**(**Course#**, **Prereq\_Course#**)

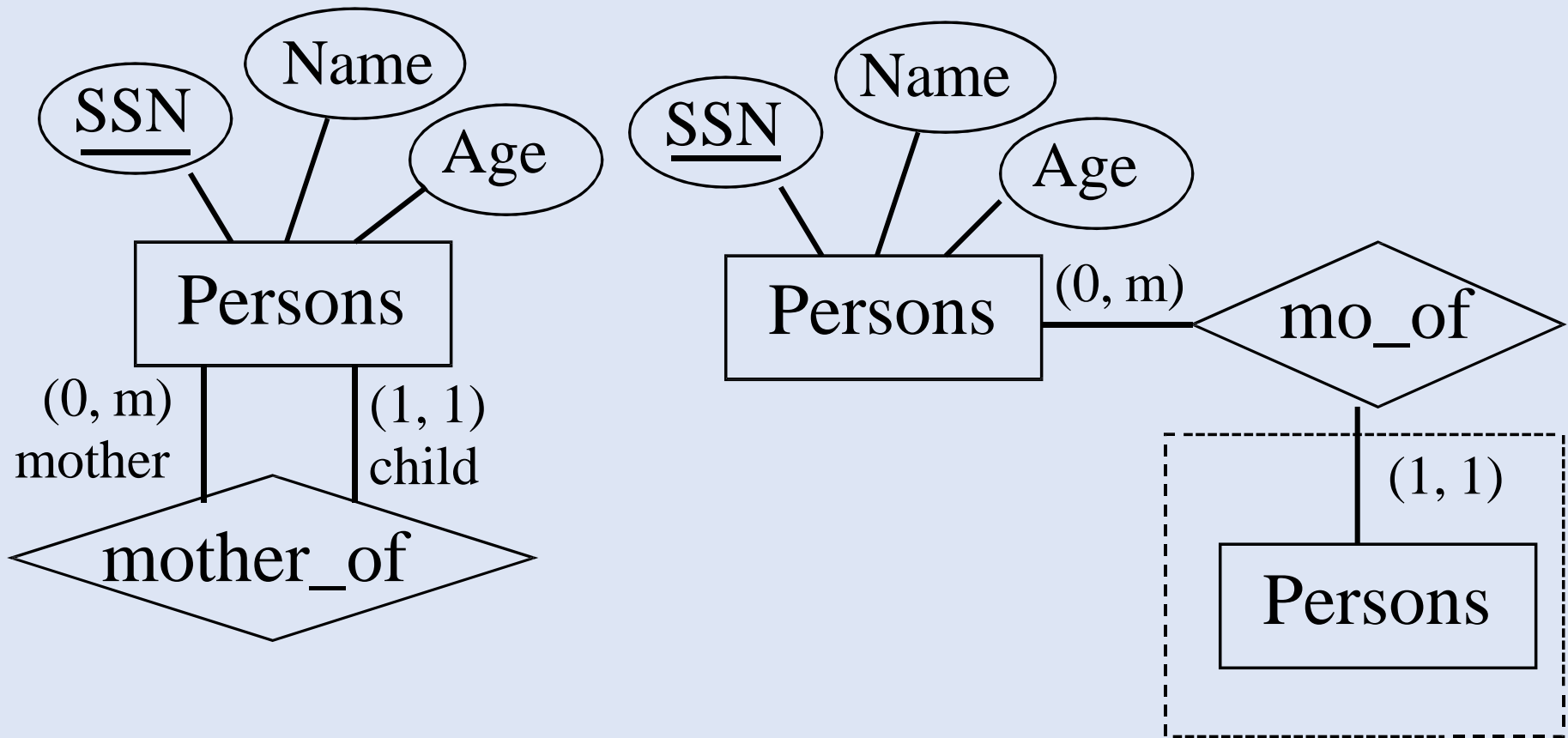
# Unary Relationship Example



If total Participation on both side

Persons(**SSN**, Name, Age, Spouse\_SSN)

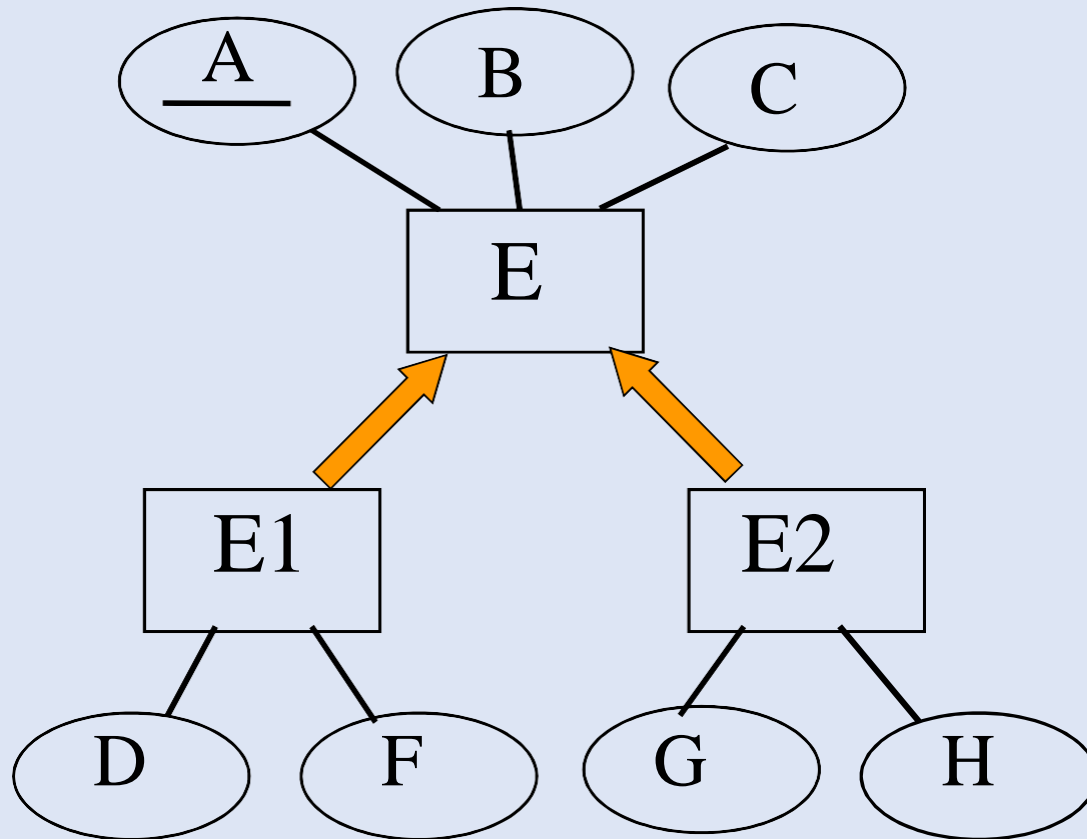
# Unary Relationship Example



**Persons(SSN, Name, Age, Mother\_SSN)**



# Mapping of IS\_A Hierarchy

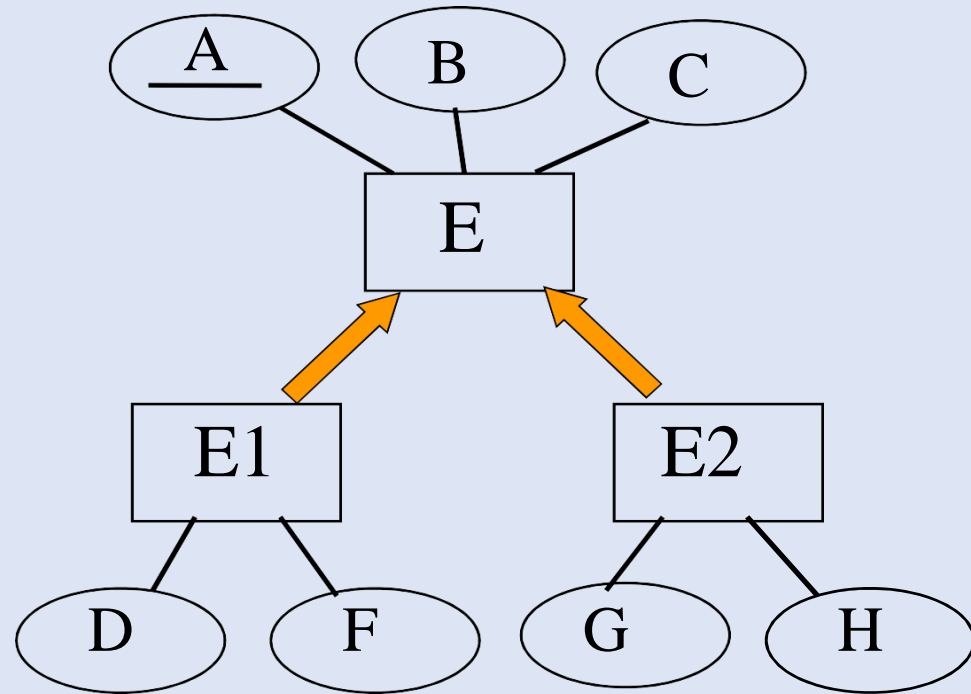


# Mapping of IS\_A Hierarchy

*Method 1:* For general case

$E(\textcolor{red}{A}, B, C),$

$E1(\textcolor{red}{A}, D, F), E2(\textcolor{red}{A}, G, H)$

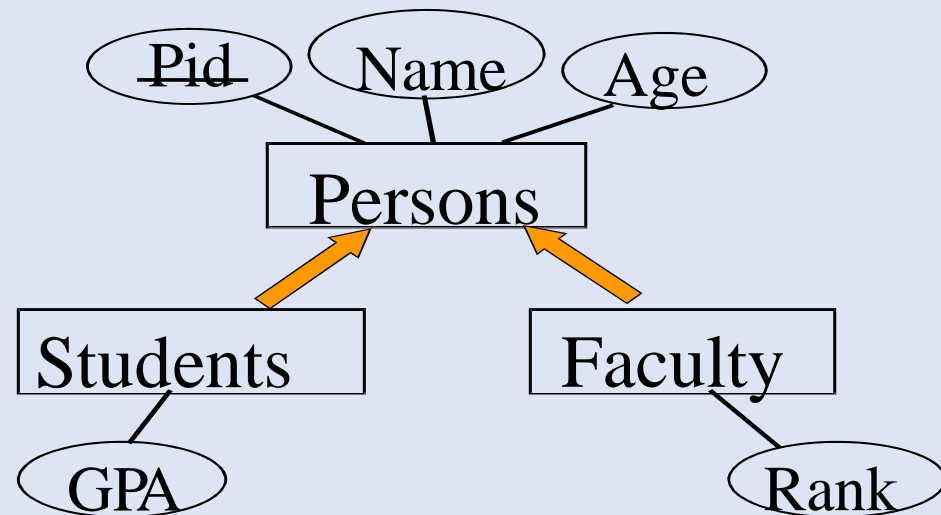


- Only the key is explicitly inherited from the super entity set
- A tuple in  $\textcolor{red}{E}$  either corresponds to an entity in  $E$  or an entity in a sub entity set.
- $\textcolor{brown}{E1.A}$  and  $\textcolor{brown}{E2.A}$  are defined to be foreign keys referencing  $\textcolor{red}{E.A}$

# Mapping of IS\_A Hierarchy Example

Real world information:

	Pid	Name	Age	GPA	Rank
stud:	123456789	John	27	3.5	
facul:	234567891	Bill	43		Prof.
staff:	345678912	Mary	37		



# Mapping of IS\_A Hierarchy Example

*Method 1:*

**Persons**

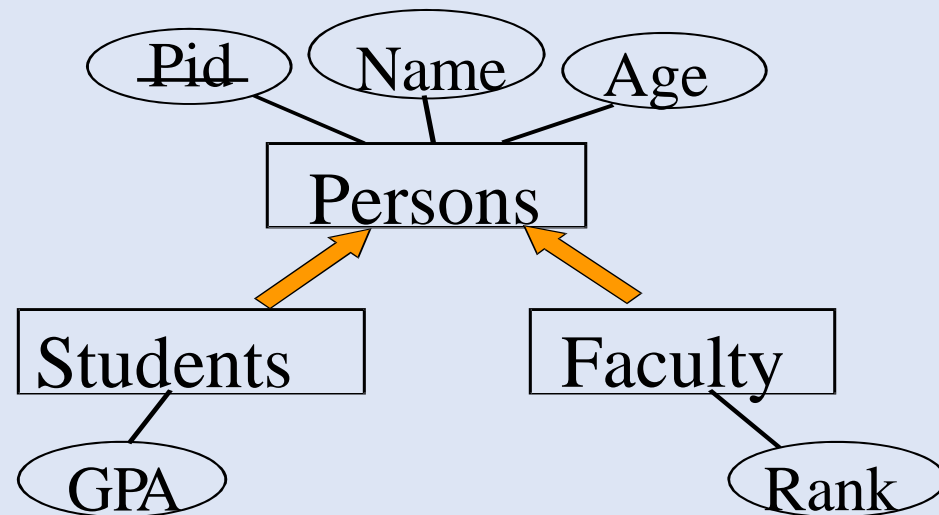
Pid	Name	Age
123456789	John	27
234567891	Bill	43
345678912	Mary	37

**Students**

Pid	GPA
123456789	3.5

**Faculty**

Pid	Rank
234567891	Prof.



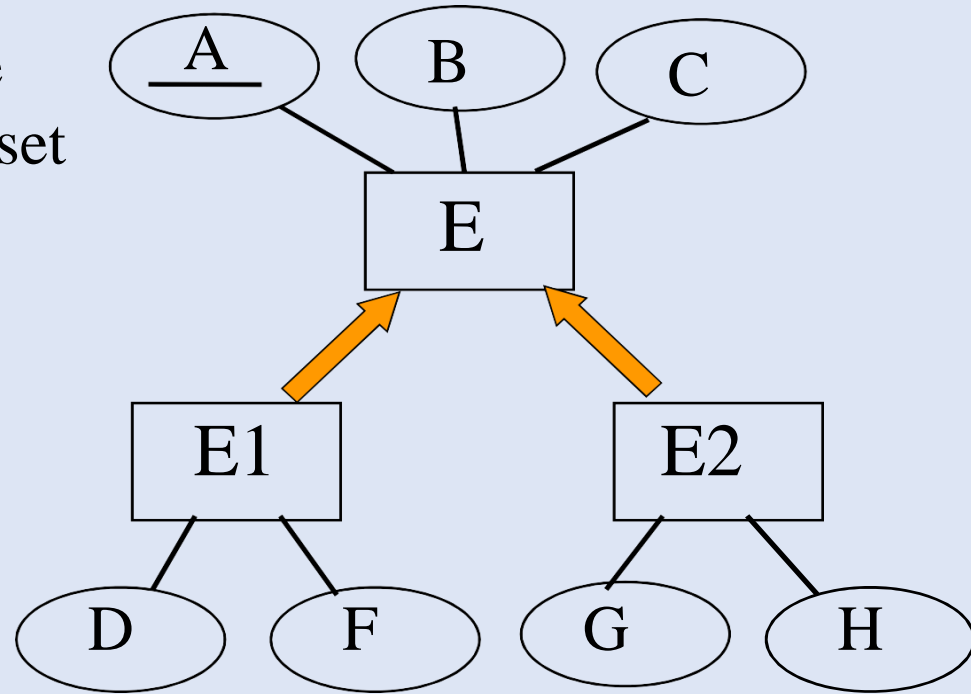
# Mapping of IS\_A Hierarchy

**Method 2:** When the union of the sub entity sets contains the same set of entities as the super entity set

**E(A, B, C),**

**E1(A, D, F, B, C),**

**E2(A, G, H, B, C)**



- All attributes are explicitly inherited from the super entity set

# Mapping of IS\_A Hierarchy Example

## *Method 2:*

### Persons

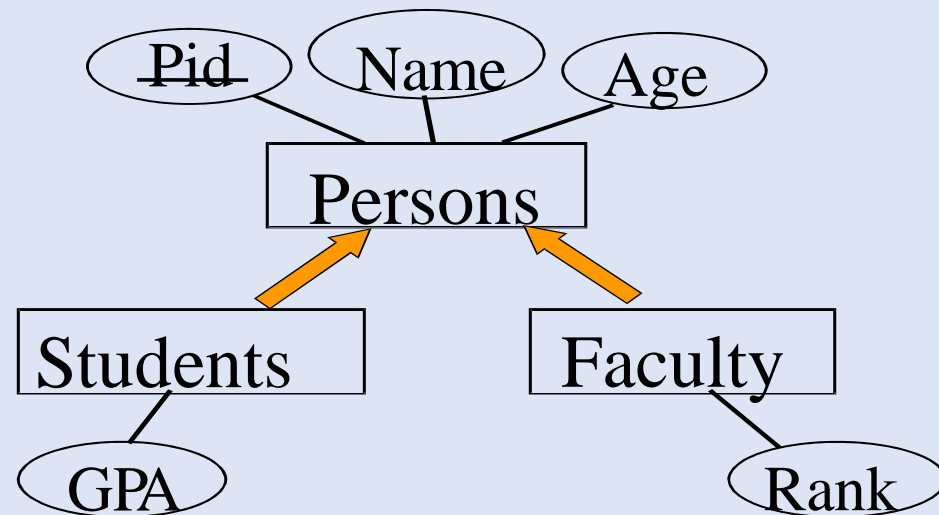
Pid	Name	Age
123456789	John	27
234567891	Bill	43
345678912	Mary	37

### Students

Pid	Name	Age	GPA
123456789	John	27	3.5

### Faculty

Pid	Name	Age	Rank
234567891	Bill	43	Prof.



# Mapping of IS\_A Hierarchy Example

*Method 3:* When the sub entity sets are disjoint based on the values of an (implicit) attribute K

$\Rightarrow E(A, B, C, D, F, G, H, K)$

- K has the same value for entities from the same entity set but different values for entities from different entity sets

E.g.: Super entity set: Employees

Sub entity sets: Engineer,  
Typewriter, ..., Technician

K: Job Type

# Mapping of IS\_A Hierarchy Example

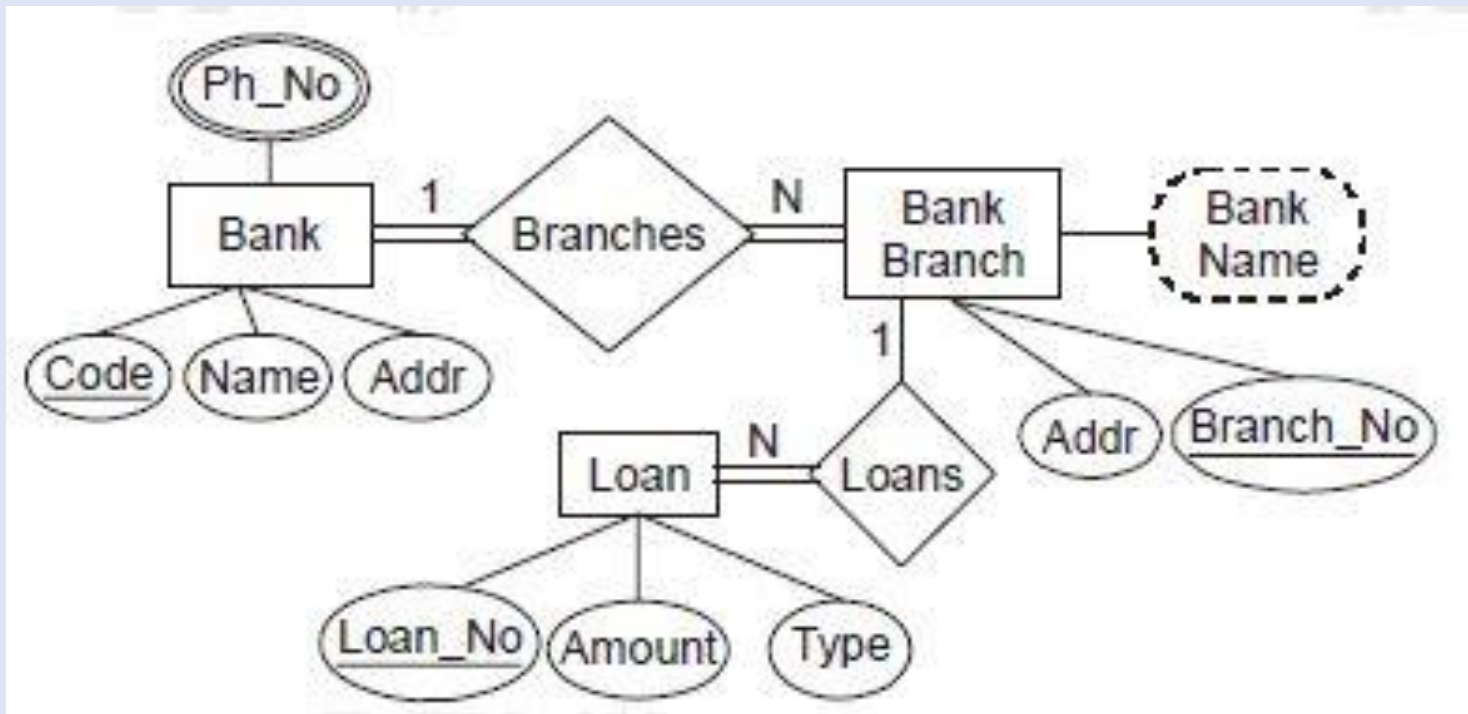
## *Method 3:*

### **Persons**

Pid	Name	Age	GPA	Rank	Type
123456789	John	27	3.5		Student
234567891	Bill	43		Prof.	Faculty
345678912	Mary	37			Staff



# Minimum Tables?



# ER Mapping Procedure

A general procedure:

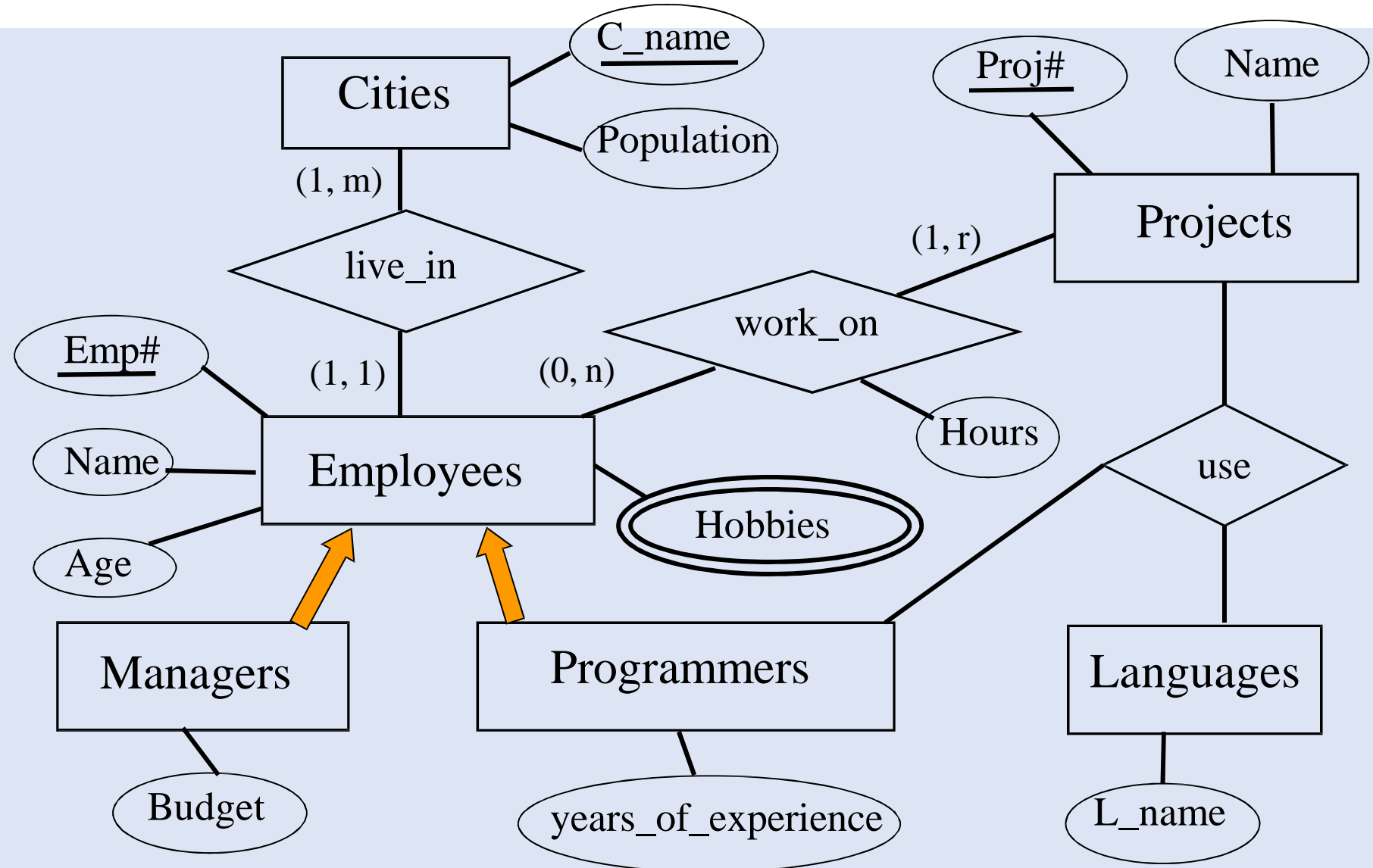
- Transform each entity set into a relation (excluding multi-valued and composite attributes)
  - Transform each IS\_A hierarchy
    - **Consider two adjacent levels (parent-child) at a time.**
    - **For methods 1 and 2, transform entity sets in a top-down manner (keep inheritance in mind)**
  - Transform each multi-valued attribute into a separate relation
  - Transform each composite attribute
  - Specify the key for each relation

# ER Mapping Procedure

Transform each relationship set

- For any unary/binary 1-to-1 or 1-to-m relationship, transform it by adding a foreign key to an appropriate relation
- Transform any m-to-m or high degree (degree > 2) relationship by creating a separate relation. Specify the key
- Re-visit relations involved in IS\_A hierarchies to deal with feature inheritance
- Specify foreign keys

# ER Mapping Example



# ER Mapping Example

Method 1:

Employees(**Emp#**, Name, Age, C\_name)

Employee-Hobby(**Emp#**, **Hobby**)

Managers(**Emp#**, Budget)

Programmers(**Emp#**, Years\_of\_experience)

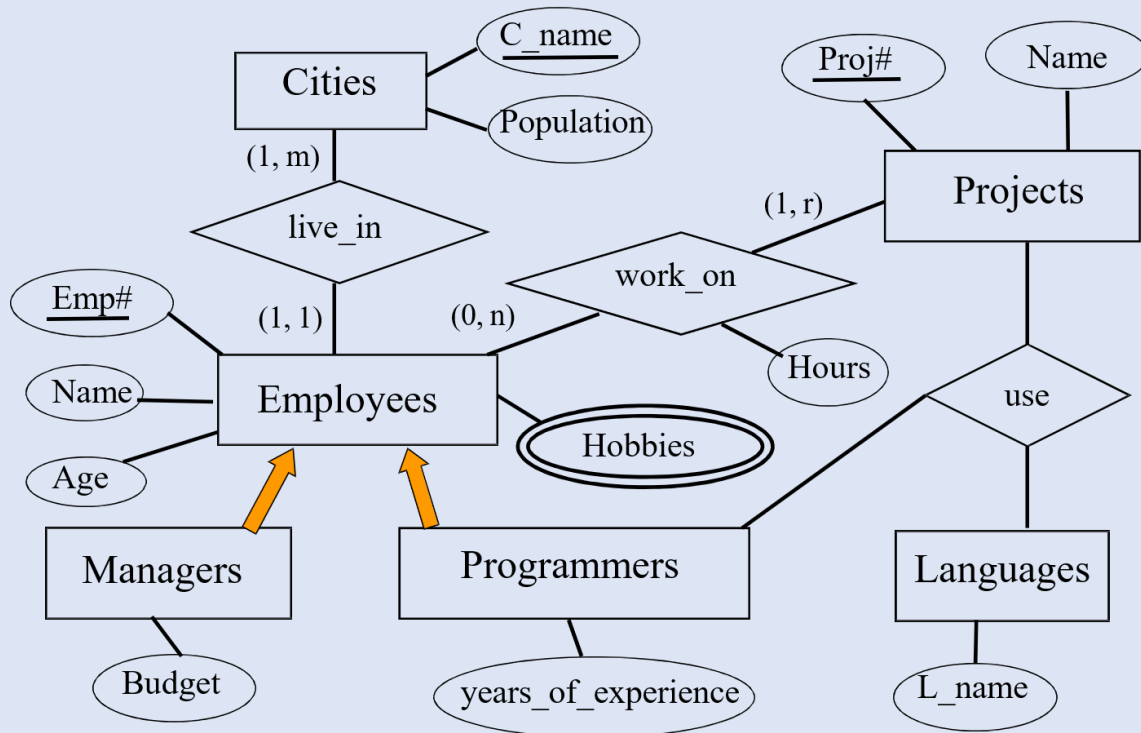
Cities(**C\_name**, Population)

Projects(**Proj#**, Name)

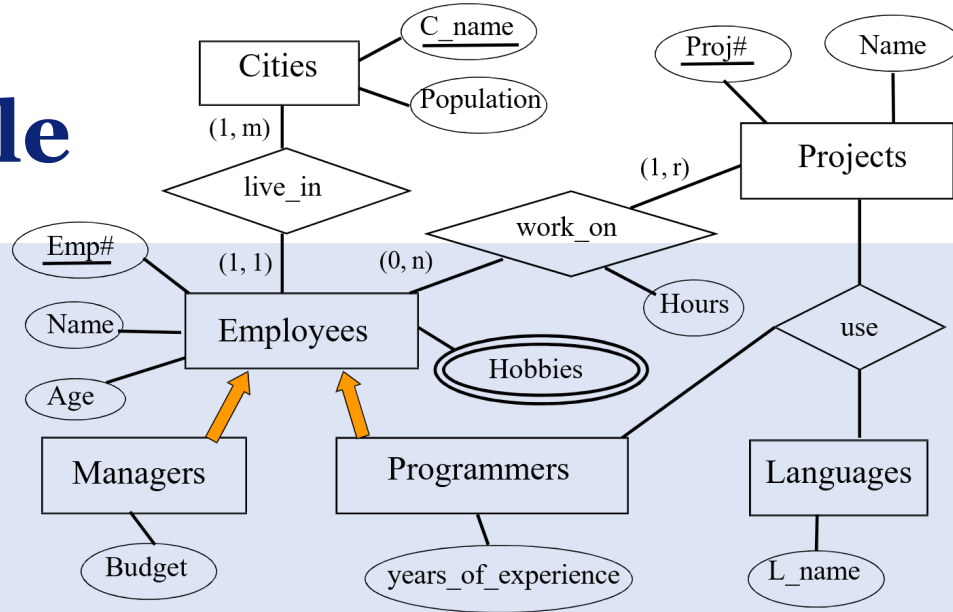
Languages(**L\_name**)

Work\_on(**Emp#**, **Proj#**, Hours)

Use(**Emp#**, **Proj#**, **L\_name**)



# ER Mapping Example



## Method 2:

Employees(**Emp#**, Name, Age, C\_name)

Employee-Hobby(**Emp#**, **Hobby**)

Managers(**Manager-Emp#**, Name, Age, Budget, C\_name)

Manager-Hobby(**Manager-Emp#**, **Hobby**)

Programmers(**Programmer-Emp#**, Name, Age, Years\_of\_experience, C\_name)

Programmer-Hobby(**Programmer-Emp#**, **Hobby**)

# ER Mapping Example

## Method 2:

Cities(**C\_name**, Population)

Projects(**Proj#**, Name)

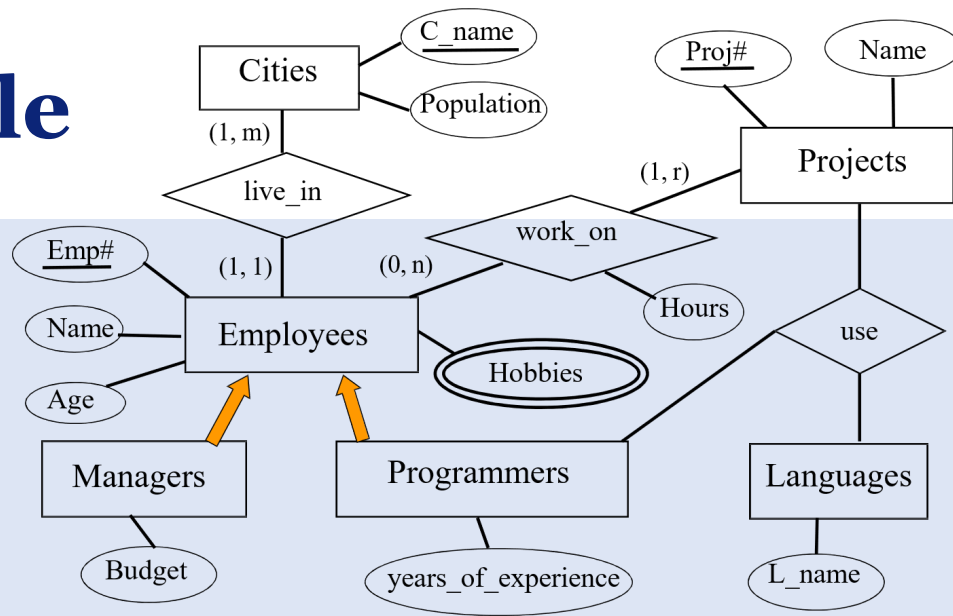
Languages(**L\_name**)

Work\_on(**Emp#**, **Proj#**, Hours)

Manager-Work\_on(**Manager-Emp#**, **Proj#**, Hours)

Programmer-Work\_on(**Programmer-Emp#**, **Proj#**, Hours)

Use(**Programmer-Emp#**, **Proj#**, **L\_name**)



# ER Model Case Study

Suppose you are given the following requirements for a simple database for the National Hockey League (NHL):

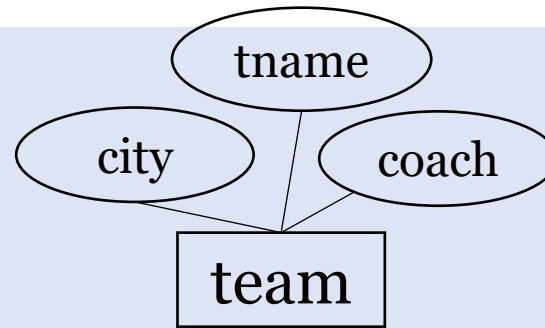
- the NHL has many teams,
- each team has a name, a city, a coach, a captain, and a set of players,
- each player belongs to only one team,
- each player has a name, a position (such as left-wing or goalie), a skill level, and a set of injury records,
- a team captain is also a player,
- a game is played between two teams (referred to as `host_team` and `guest_team`) and has a date (such as Feb 16th, 2023) and a score (such as 4 to 2)

Construct a clean and concise ER diagram for the NHL database



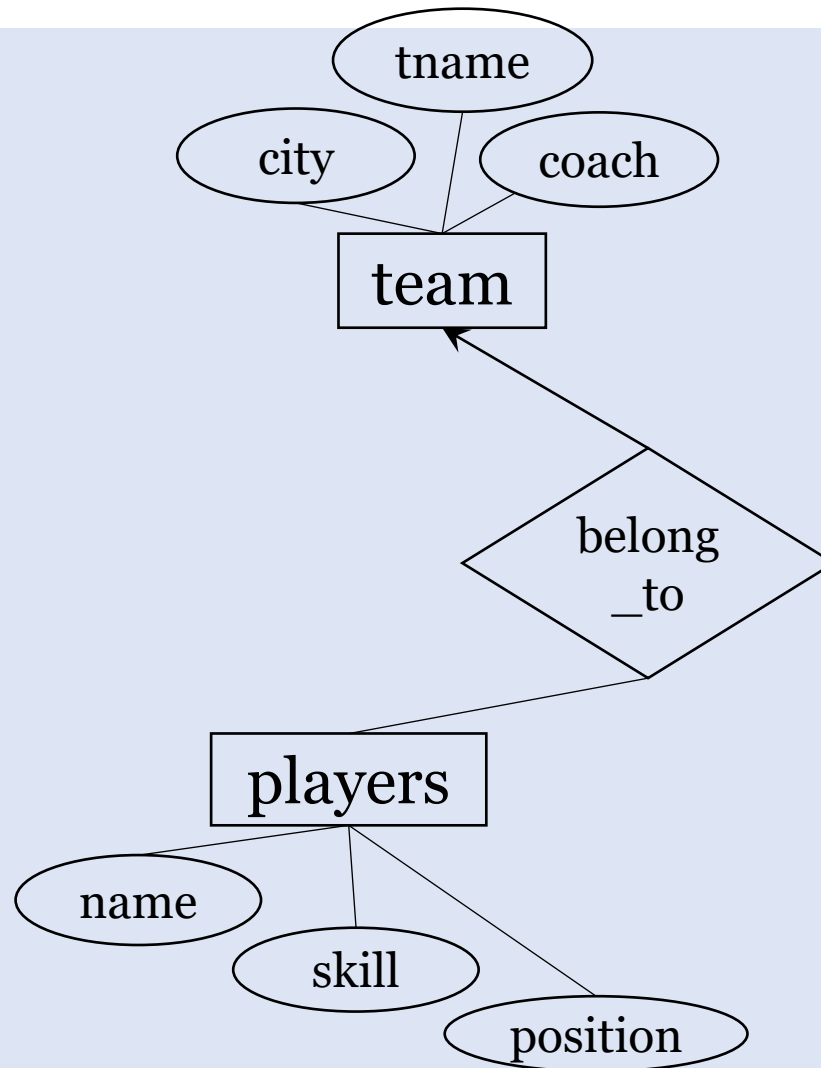
# ER Model Case Study

- the NHL has many **teams**,
- each team has a name, a city, a coach, a captain, and a set of players,
- each player belongs to only one team,
- each player has a name, a position (such as left-wing or goalie), a skill level, and a set of injury records,
- a team captain is also a player,
- a game is played between two teams (referred to as `host_team` and `guest_team`) and has a date (such as Feb 16th, 2023) and a score (such as 4 to 2)



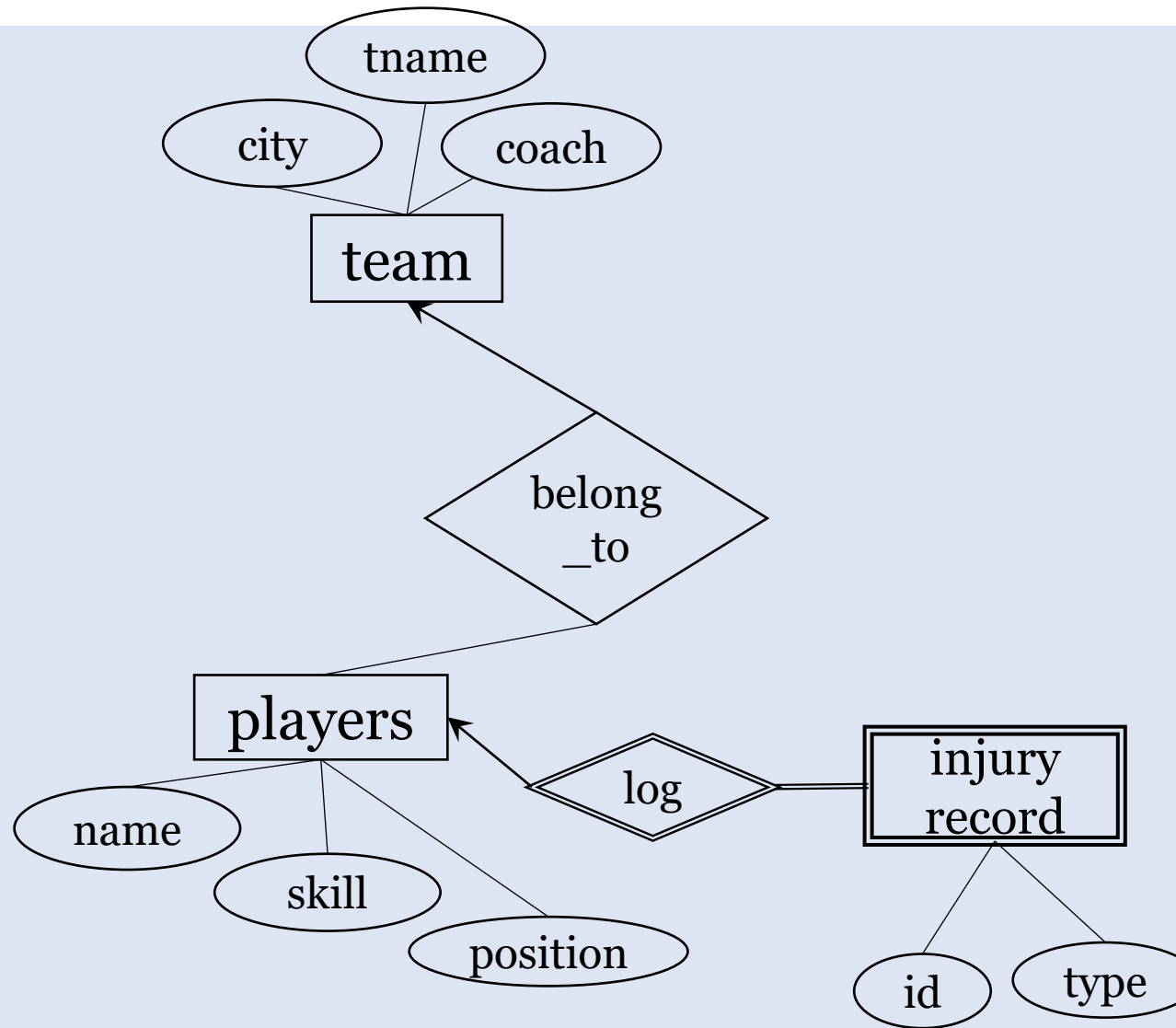
# ER Model Case Study

- the NHL has many **teams**,
- each team has a name, a city, a coach, a captain, and a set of players,
- **each player belongs to only one team**,
- **each player has a name, a position (such as left-wing or goalie), a skill level, and a set of injury records**,
- a team captain is also a player,
- a game is played between two teams (referred to as `host_team` and `guest_team`) and has a date (such as Feb 16th, 2023) and a score (such as 4 to 2)



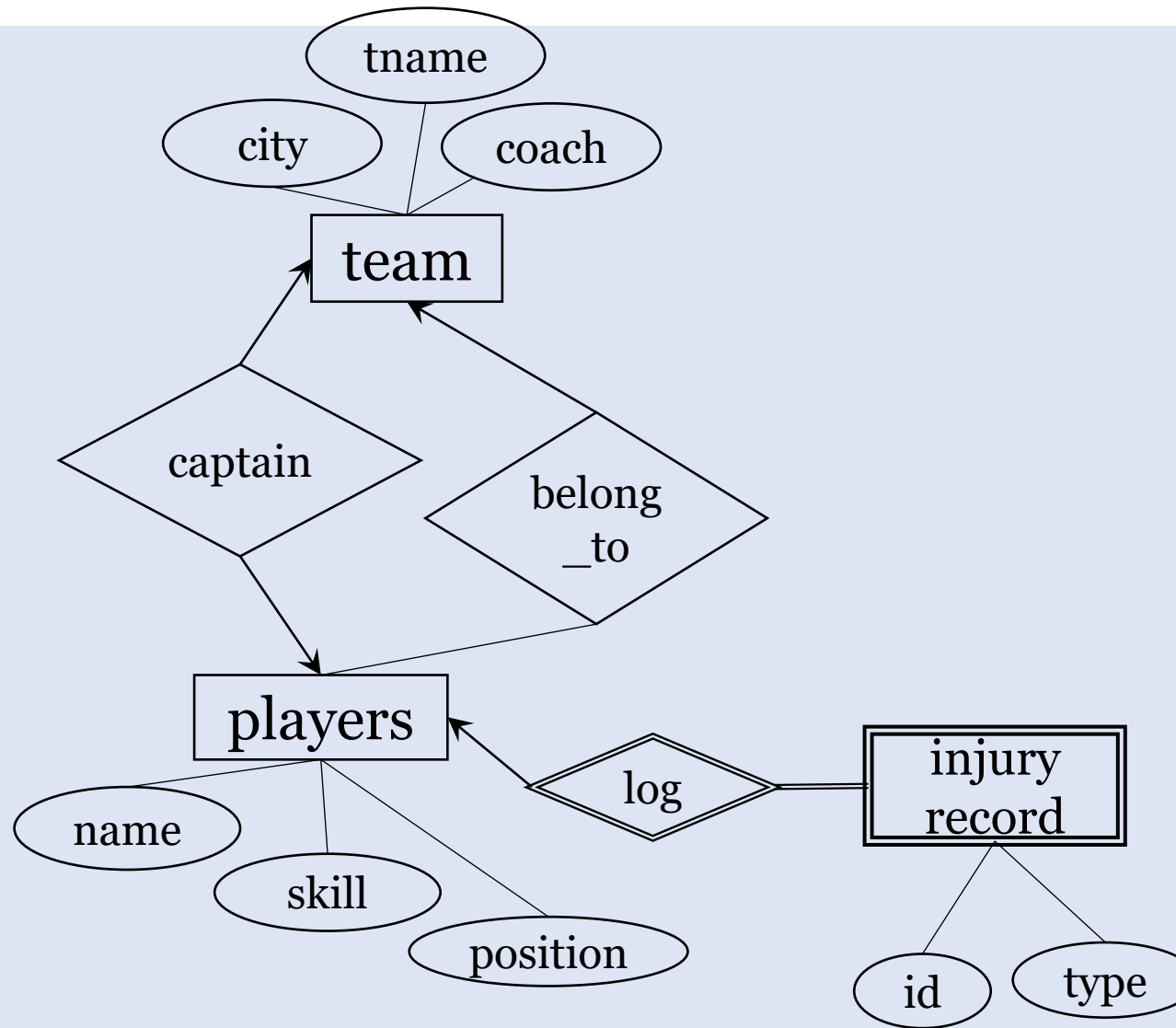
# ER Model Case Study

- the NHL has many **teams**,
- each team has a name, a city, a coach, a captain, and a set of players,
- each player belongs to only one team,
- each player has a name, a position (such as left-wing or goalie), a skill level, and a **set of injury records**,
- a team captain is also a player,
- a game is played between two teams (referred to as `host_team` and `guest_team`) and has a date (such as Feb 16th, 2023) and a score (such as 4 to 2)



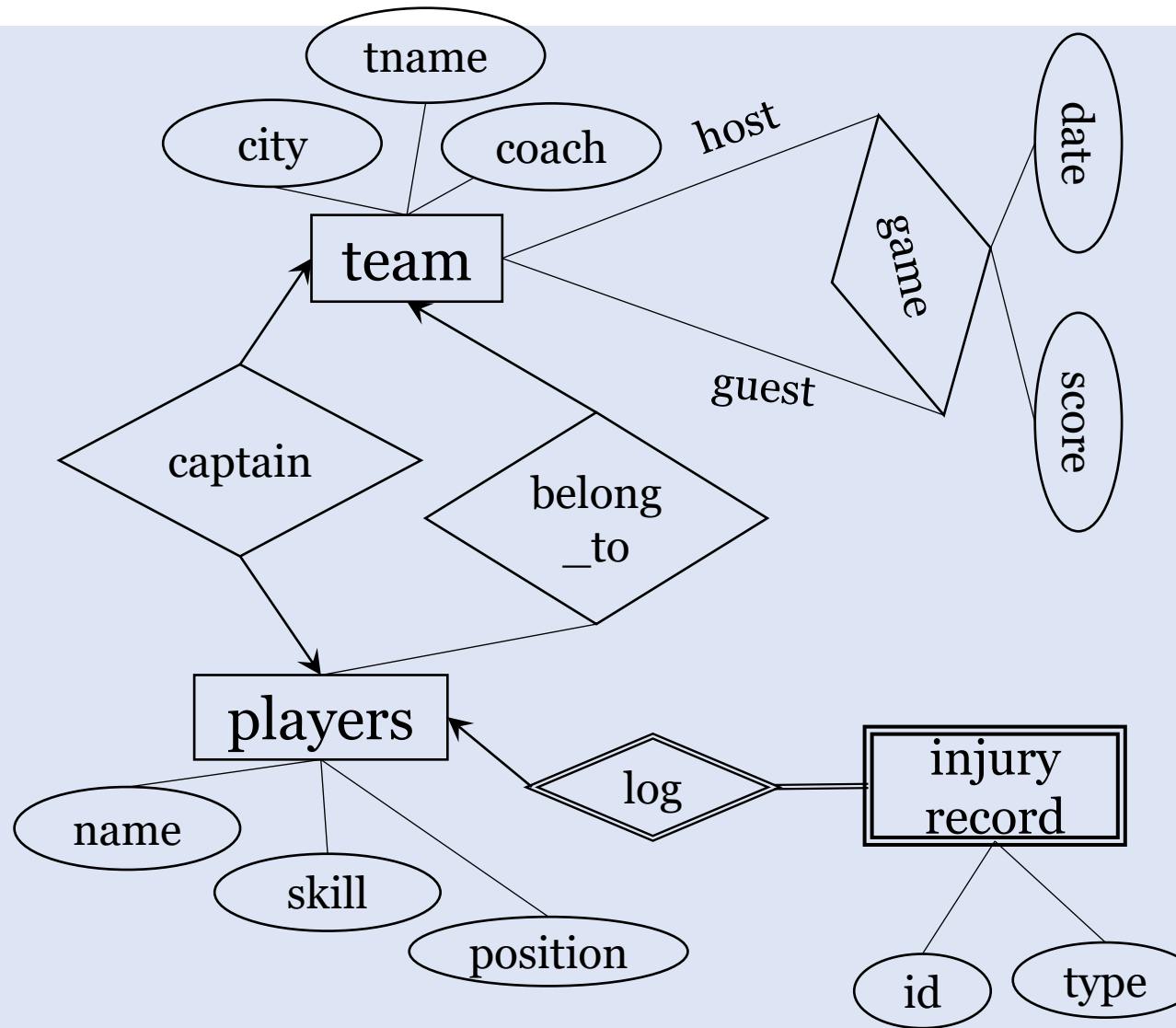
# ER Model Case Study

- the NHL has many **teams**,
- each team has a name, a city, a coach, a captain, and a set of players,
- each player belongs to only one team,
- each player has a name, a position (such as left-wing or goalie), a skill level, and a set of injury records,
- a team captain is also a player,**
- a game is played between two teams (referred to as `host_team` and `guest_team`) and has a date (such as Feb 16th, 2023) and a score (such as 4 to 2)



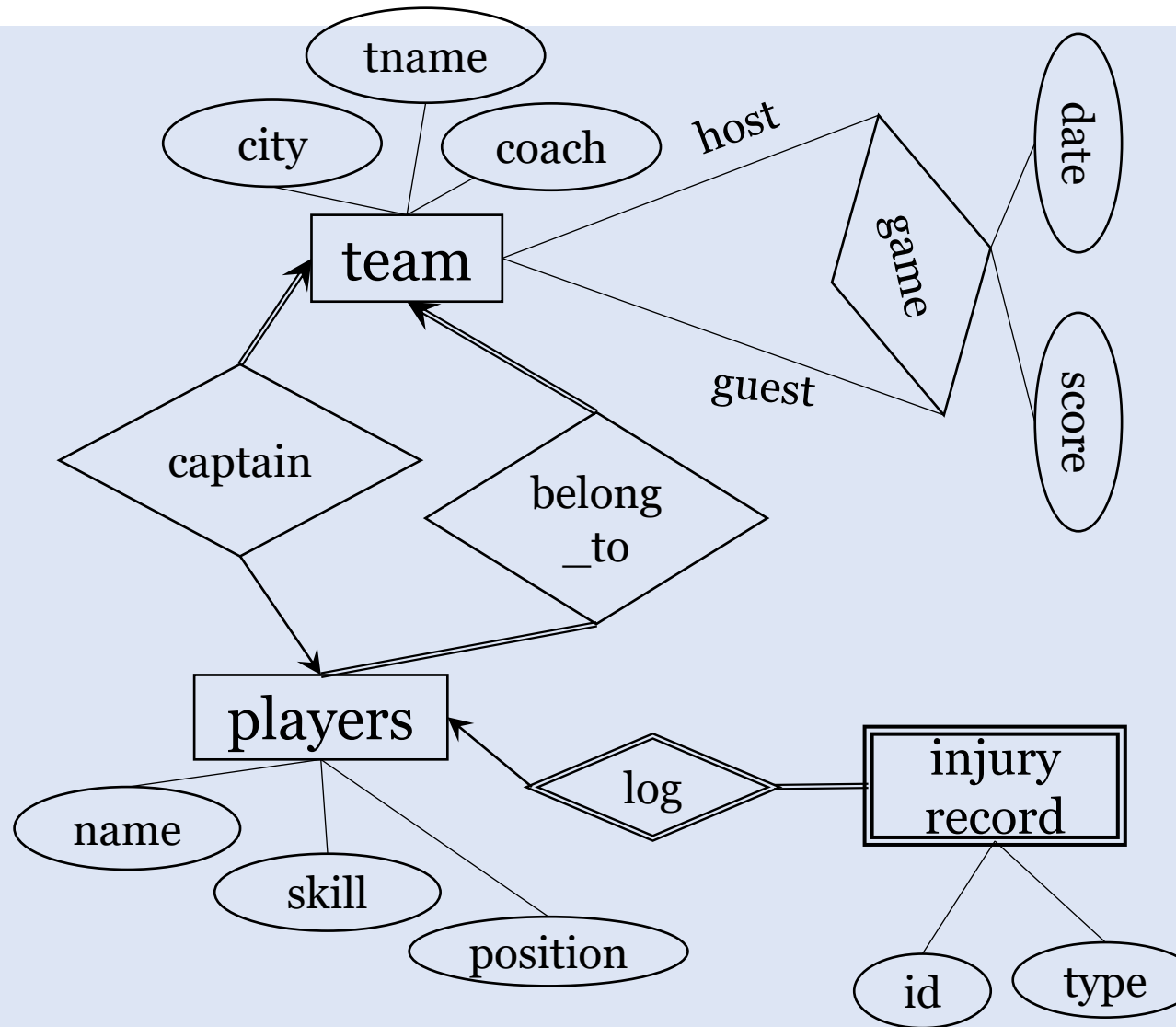
# ER Model Case Study

- the NHL has many **teams**,
- each team has a name, a city, a coach, a captain, and a set of players,
- each player belongs to only one team,
- each player has a name, a position (such as left-wing or goalie), a skill level, and a set of injury records,
- a team captain is also a player,
- a game is played between two teams (referred to as `host_team` and `guest_team`) and has a date (such as Feb 16th, 2023) and a score (such as 4 to 2)



# ER Model Case Study

- the NHL has many **teams**,
- each team has a name, a city, a coach, a captain, and a set of players,
- each player belongs to only one team,
- each player has a name, a position (such as left-wing or goalie), a skill level, and a set of injury records,
- a team captain is also a player,
- a game is played between two teams (referred to as `host_team` and `guest_team`) and has a date (such as Feb 16th, 2023) and a score (such as 4 to 2)



**Thanks!**

