CZ2007 Introduction to Database Systems (Week 1)

Topic 1: Entity Relationship Diagram (2)

Dr. Ng Wee Keong
Associate Professor

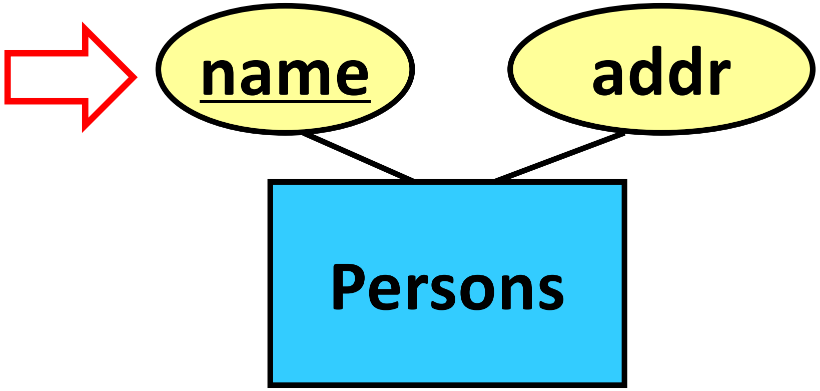
This Lecture

- Constraints ⬅
- Subclasses
- Weak Entity Sets

Constraints

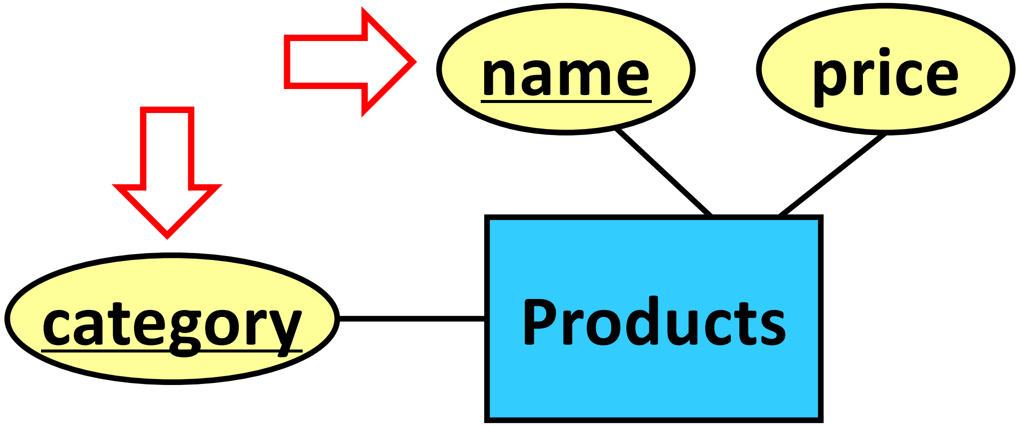
- Some conditions that entity sets and relationships should satisfy
- We will focus on three types of constraints
  - Key constraints ✓
  - Referential integrity constraints ✓
  - Degree constraints ✓

Key



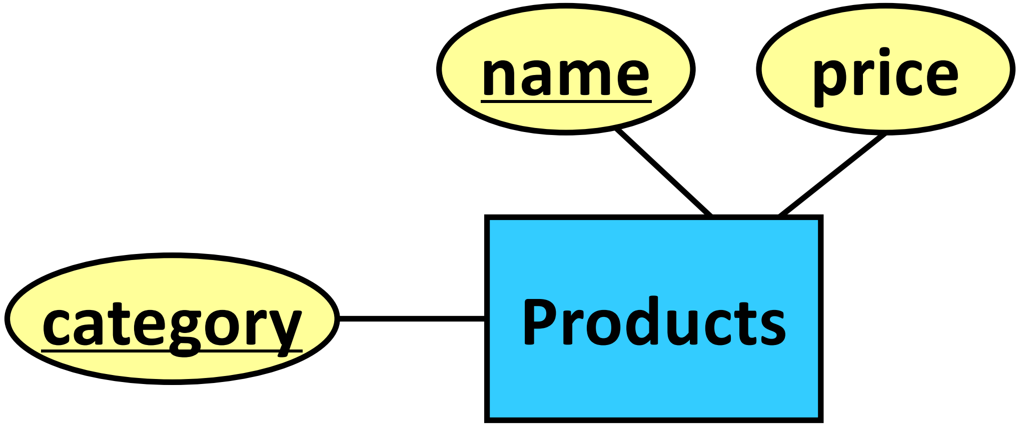
- One or more attributes that are underlined
- Meaning: They uniquely represent each entity in the entity set
- Example: The names uniquely represent the persons
- i.e., each person must have a unique name

Key



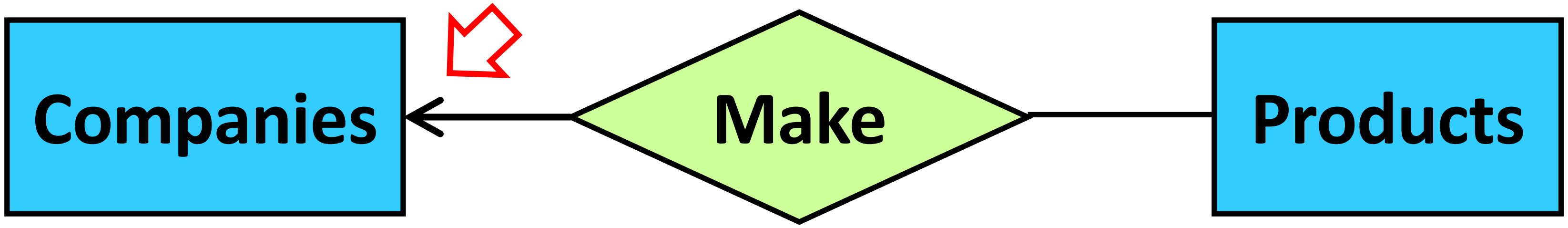
- One or more attributes that are underlined
- What now?
- Each product has a unique <name, category> combination
- But there can be products with the same name, or the same category, but not both
- Example
  - ❑ Name = "Apple", Category = "Fruit", Price = "1"
  - ❑ Name = "Apple", Category = "Phone", Price = "888"

Key



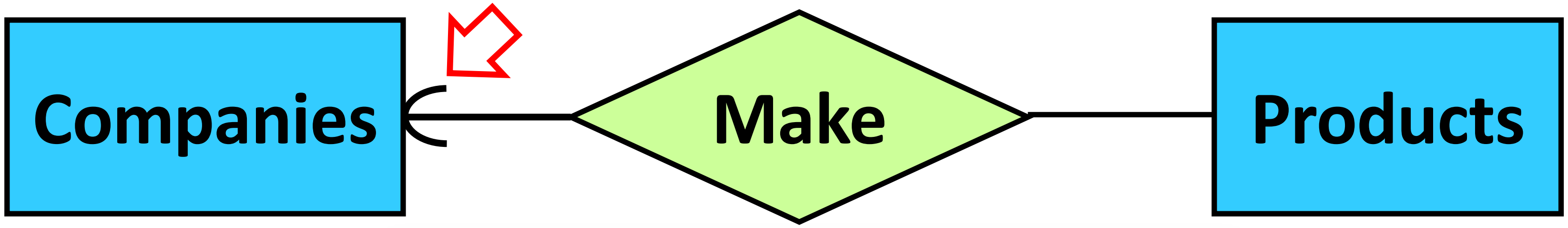
- Rule: Every entity set should have a key
  - So that we can uniquely refer to each entity in the entity set

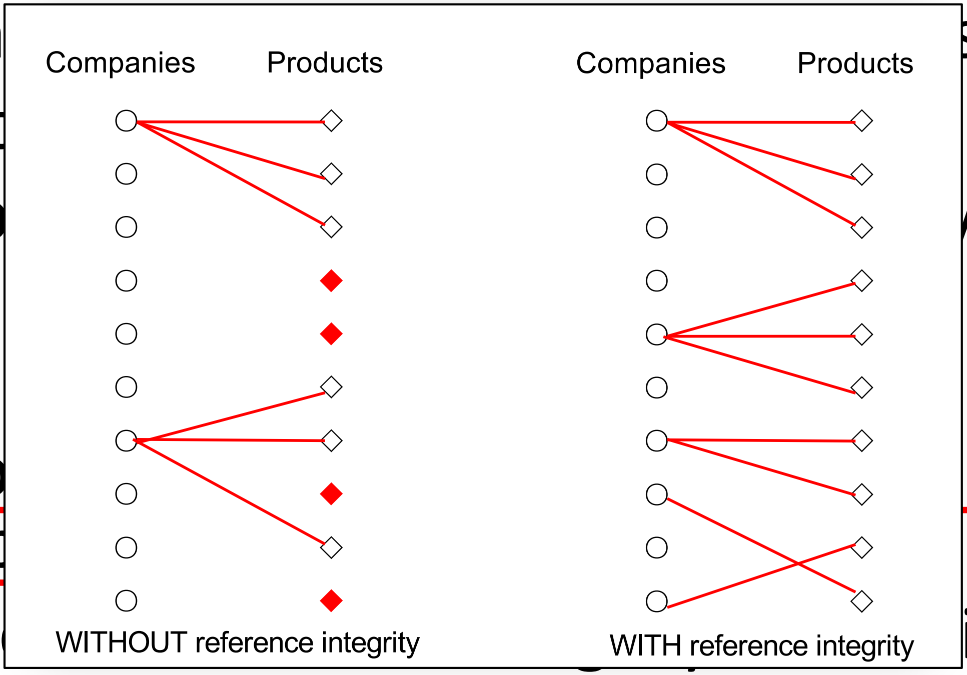
Referential Integrity



- One company may make multiple products
- One product is made by one company
- Can there be a product that is not made by any company?
- No.
- i.e., every product must be involved in the Make relationship
- This is called a referential integrity constraint.
- How do we specify this in an ER diagram?
- Use a rounded arrow instead of a pointed arrow

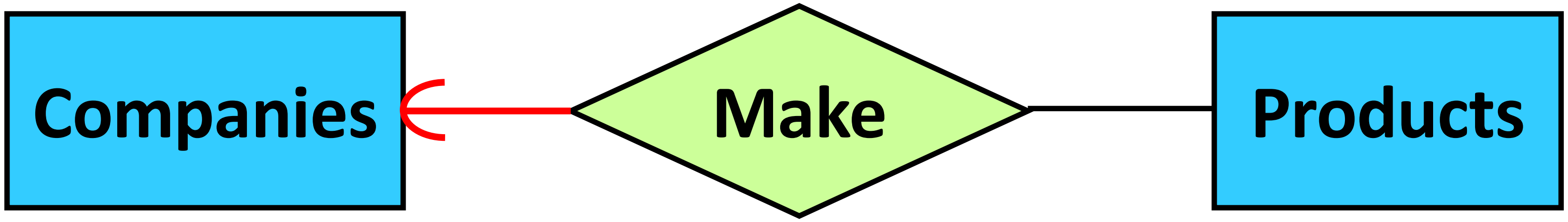
Referential Integrity



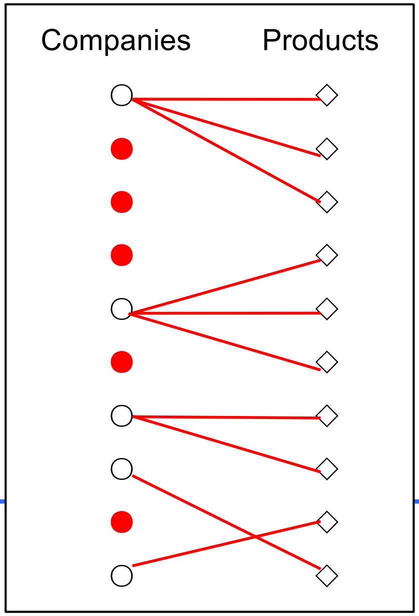


- One compa s
- One produc
- Can there b y any company?
- No.
- i.e., every p Make relationship
- This is calle int.
- How do we specify this in an ER diagram?
- Use a rounded arrow instead of a pointed arrow

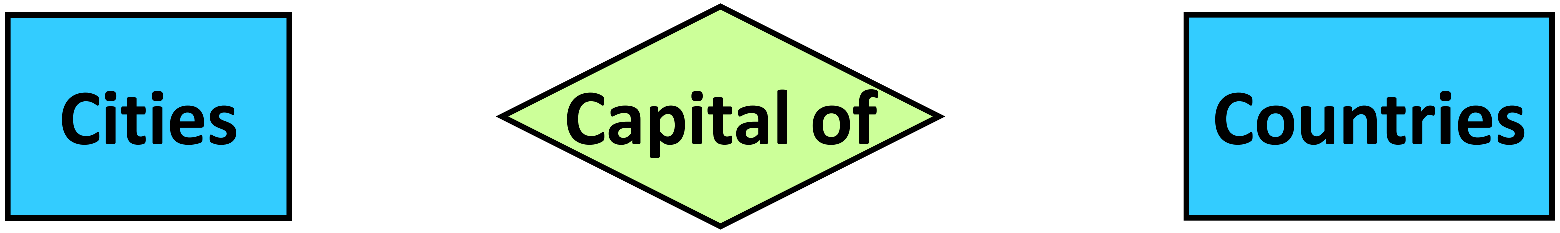
Referential Integrity



- What if every company should make at least one product?
- No arrow there …. but we indicate using degree constraints
- In general, a referential integrity constraint can only apply to the "one" side of
  - A many-to-one relationship, or
  - A one-to-one relationship
-
-



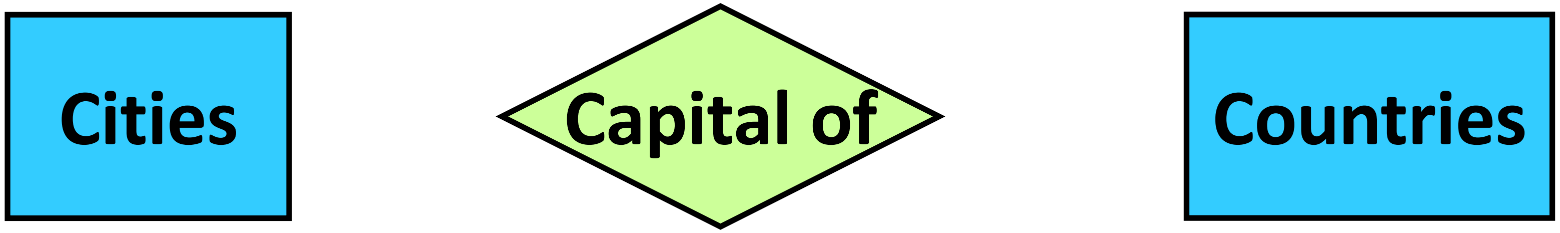
Referential Integrity: Exercise



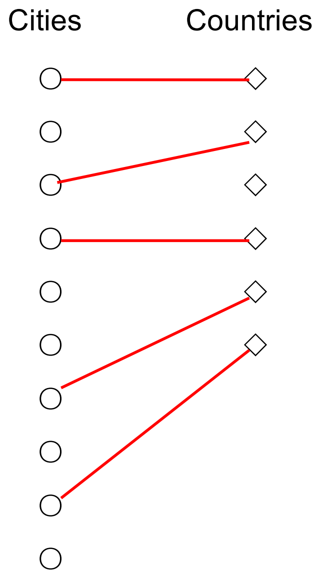
- A city can be the capital of only one country
- A country must have a capital

one

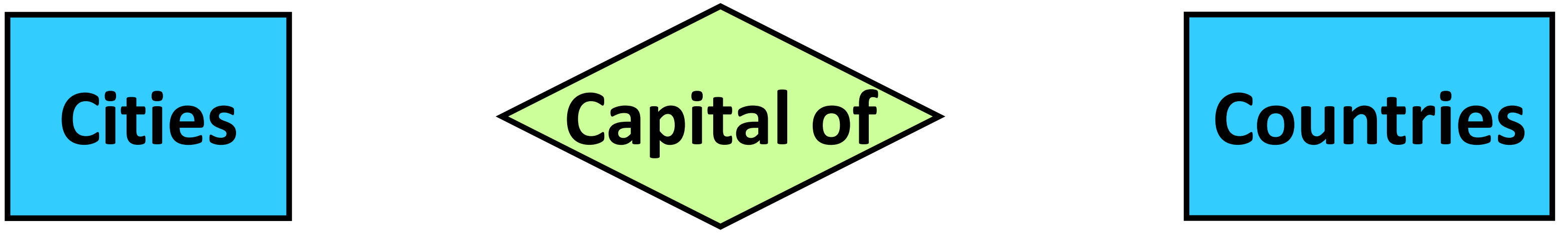
Referential Integrity: Exercise

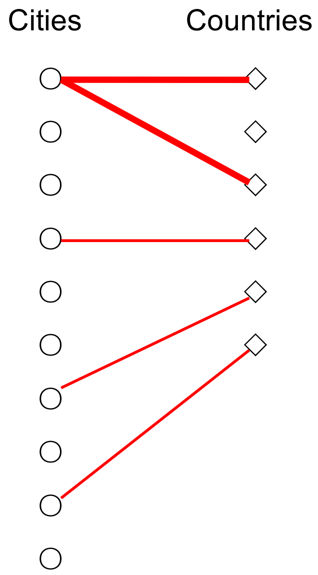


- A city can be the capital of only one country
-



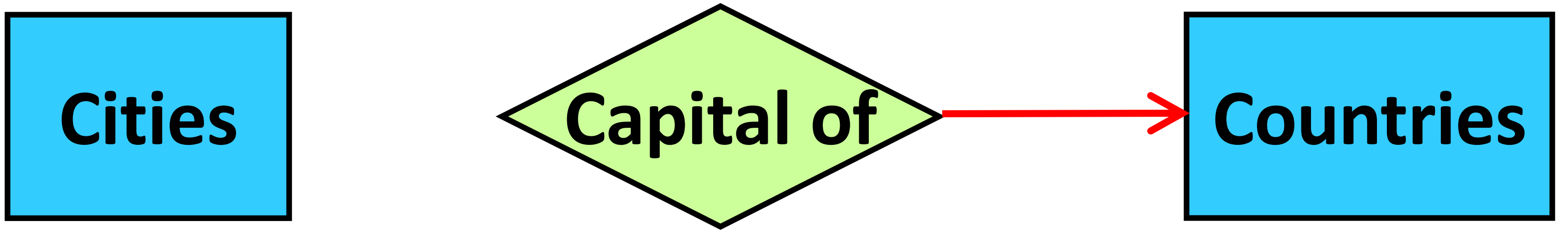
Referential Integrity: Exercise



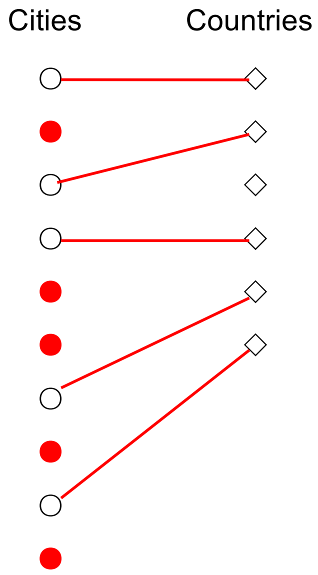


- A city can be the capital of only one country
-

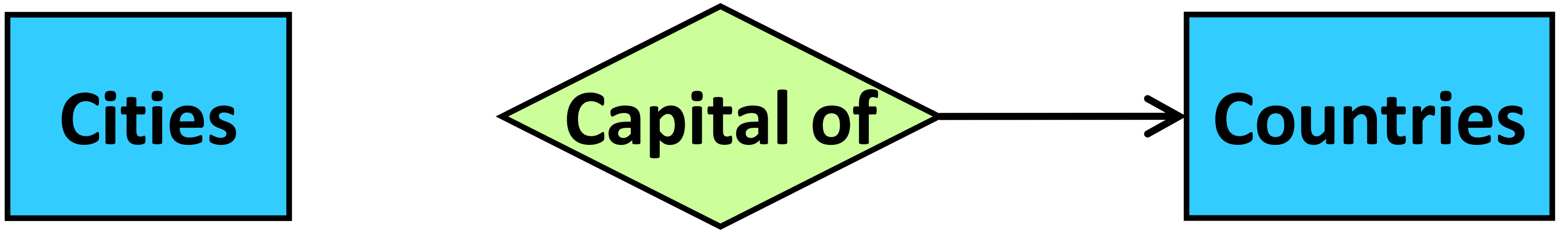
Referential Integrity: Exercise



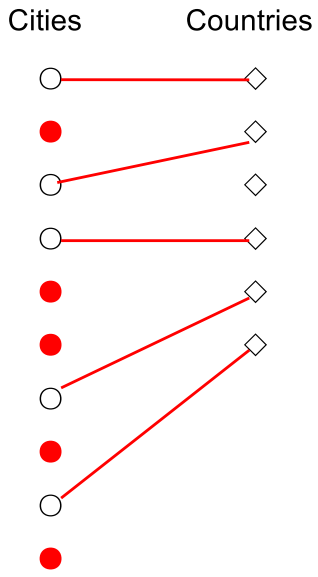
- A city can be the capital of only one country



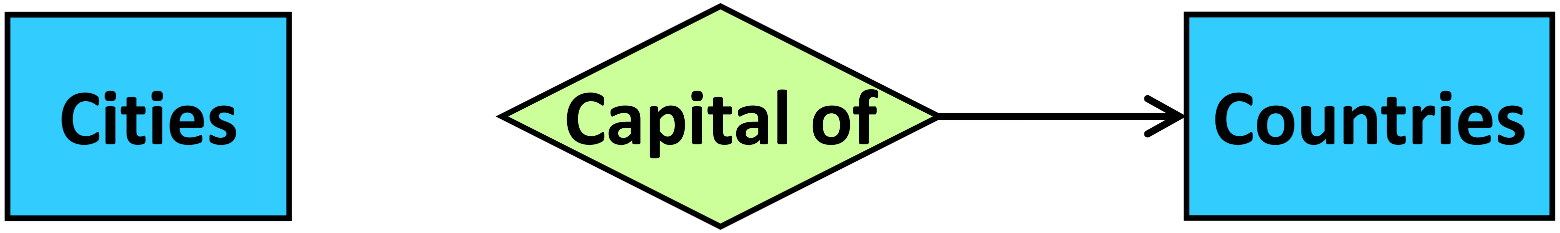
Referential Integrity: Exercise



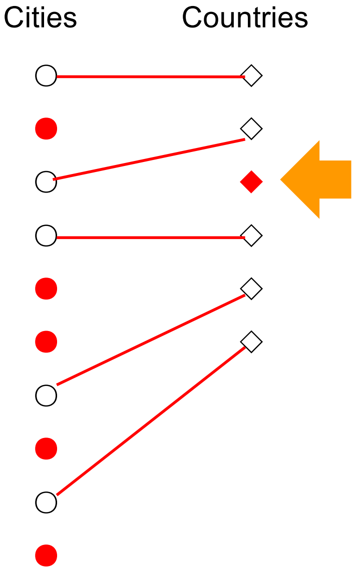
- A city can be the capital of only one country
- A country must have a capital



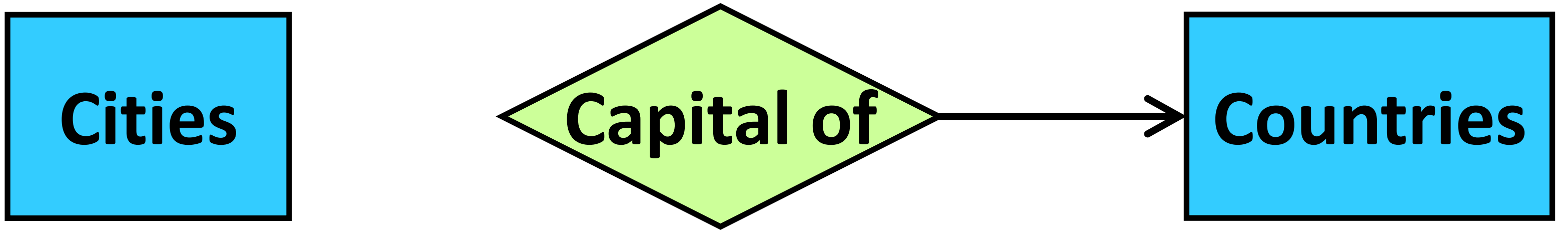
Referential Integrity: Exercise



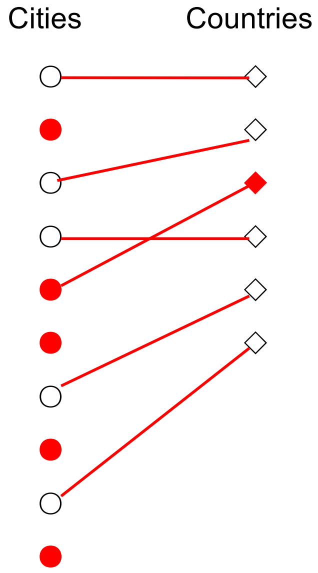
- A city can be the capital of only one country
- A country must have a capital



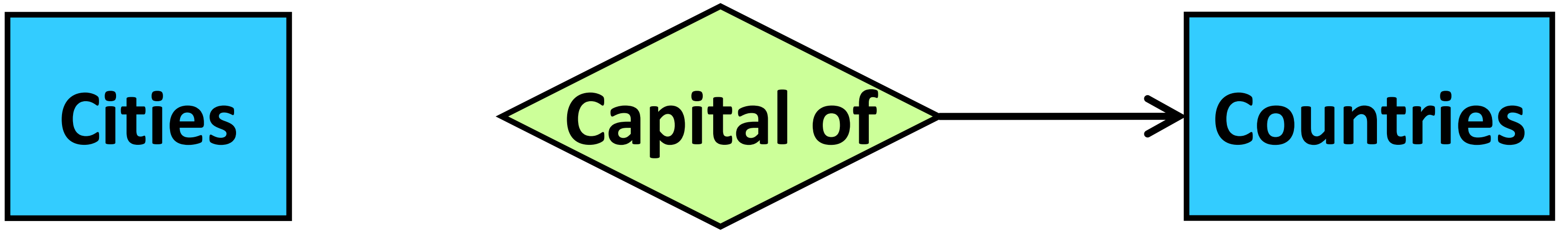
Referential Integrity: Exercise



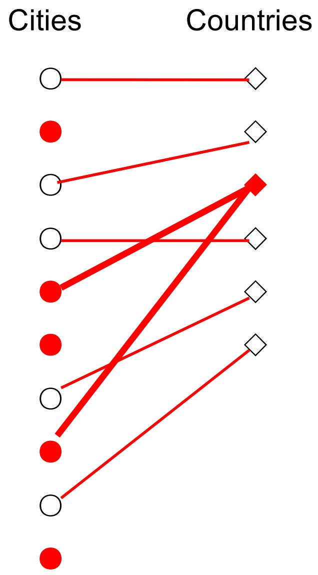
- A city can be the capital of only one country
- A country must have a capital



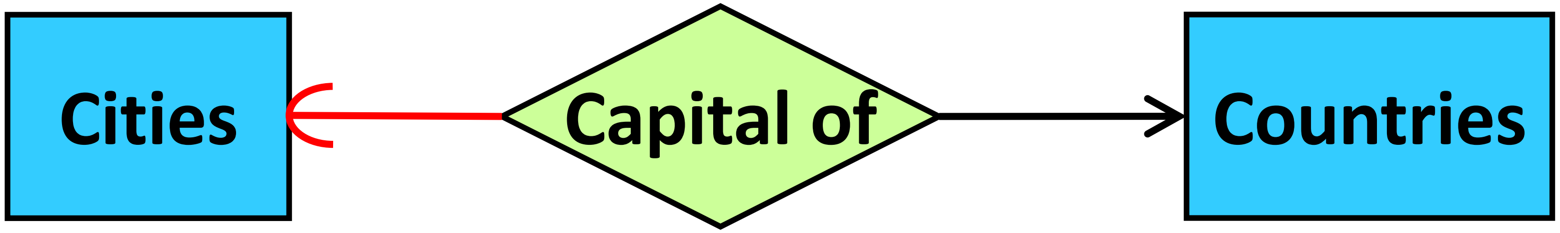
Referential Integrity: Exercise



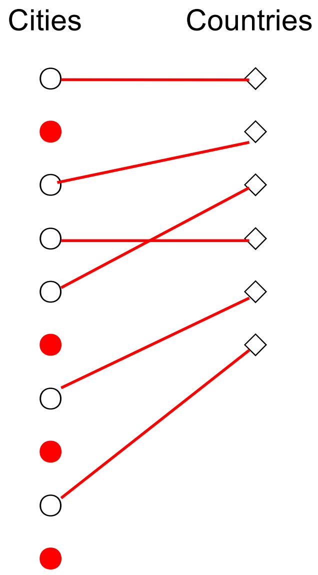
- A city can be the capital of only one country
- A country must have a capital



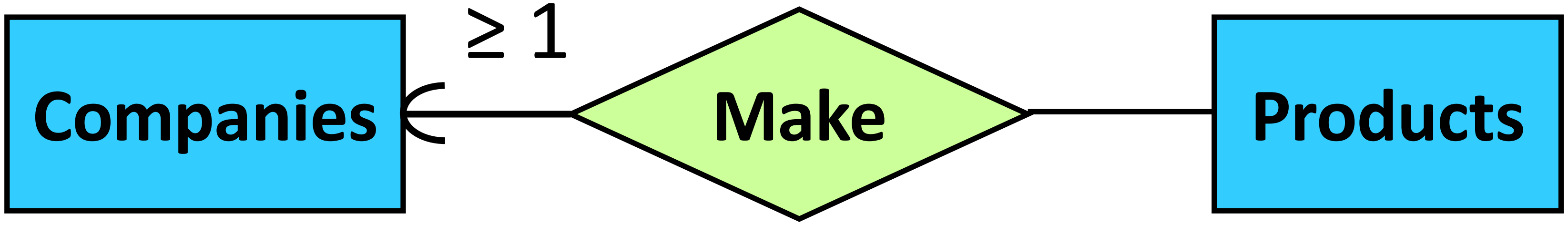
Referential Integrity: Exercise



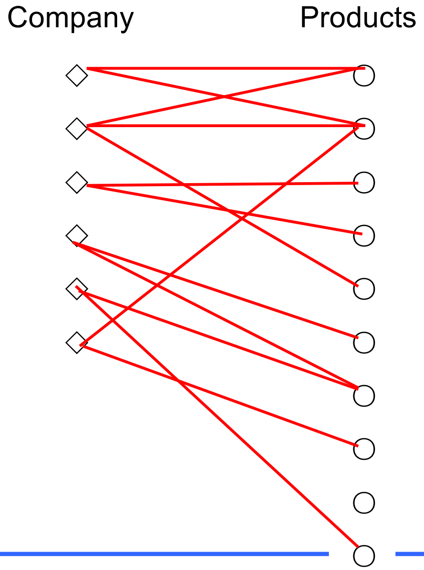
- A city can be the capital of only one country
- A country must have a capital



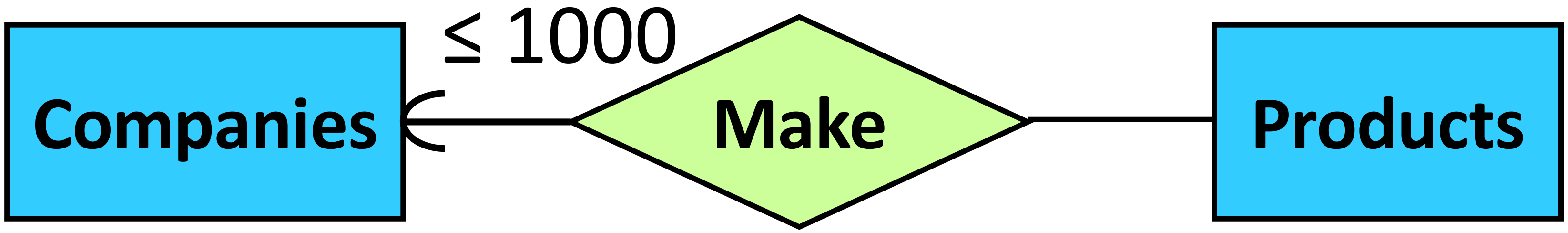
Degree Constraint



- Each company should make at least 1 products

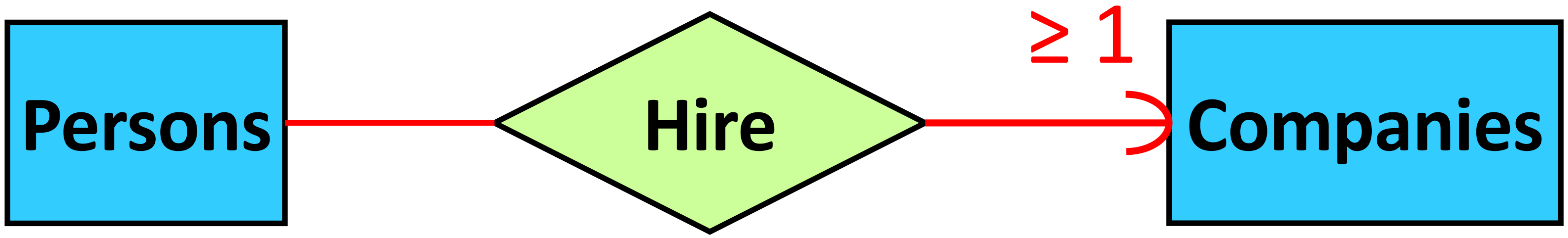


Degree Constraint



- Each company can make at most 1000 products
- Note
  - Not required in the quiz/exam
  - Degree constraints are not easy to enforce in a DBMS
  - Key and referential integrity constraints can be easily enforced

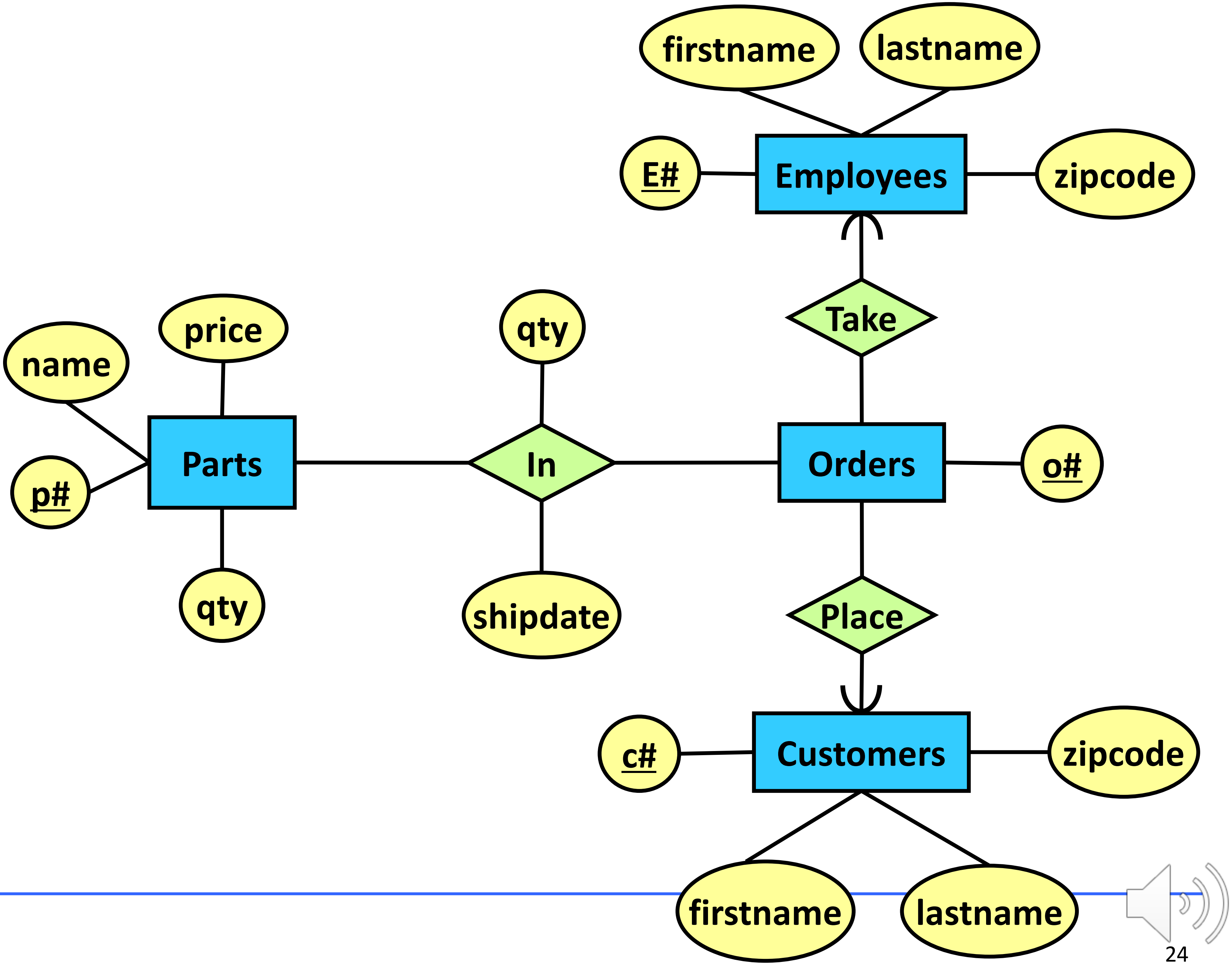
Exercise

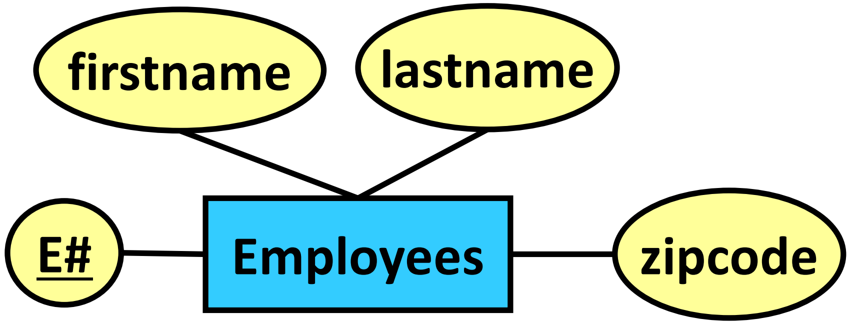


- A company must hire at least one person
- A person must be hired by exactly one company

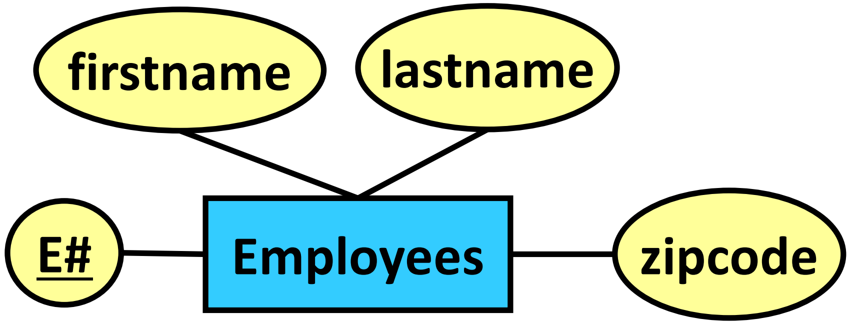
ER Diagram Design: Exercise

- Consider a mail order database in which employees take orders for parts from customers. The requirements are:
- Each employee is identified by a unique employee number, and has a first name, a last name, and a zip code.
- Each customer is identified by a unique customer number, and has a first name, last names, and a zip code.
- Each part being sold is identified by a unique part number. It has a part name, a price, and a quantity in stock.
- Each order placed by a customer is taken by one employee and is given a unique order number. Each order may contain certain quantities of one or more parts. The shipping date of each part is also recorded.

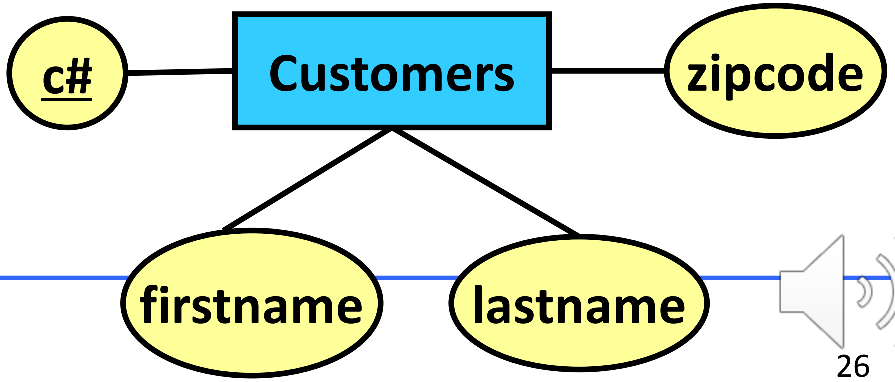

firstname
lastname
E#
Employees
zipcode
Take
price
name
qty
Parts
p#
qty
In
shipdate
Orders
o#
Place
c#
Customers
zipcode
firstname
lastname

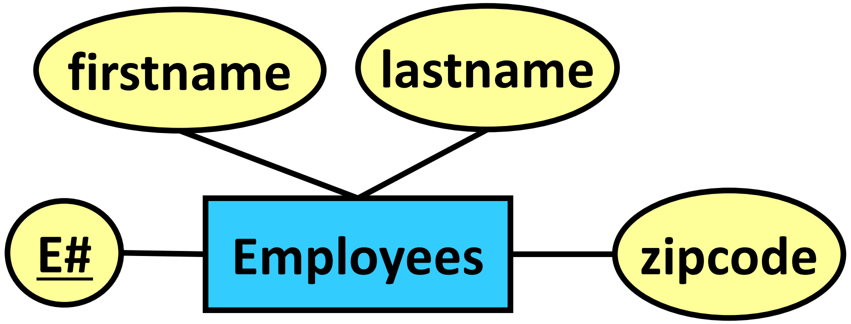


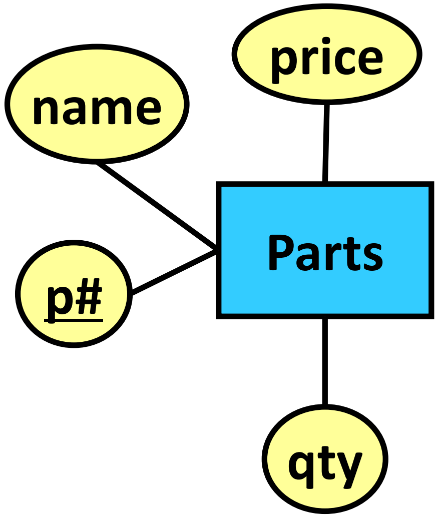
- Each employee is identified by a unique employee number, and has a first name, a last name, and a zip code.



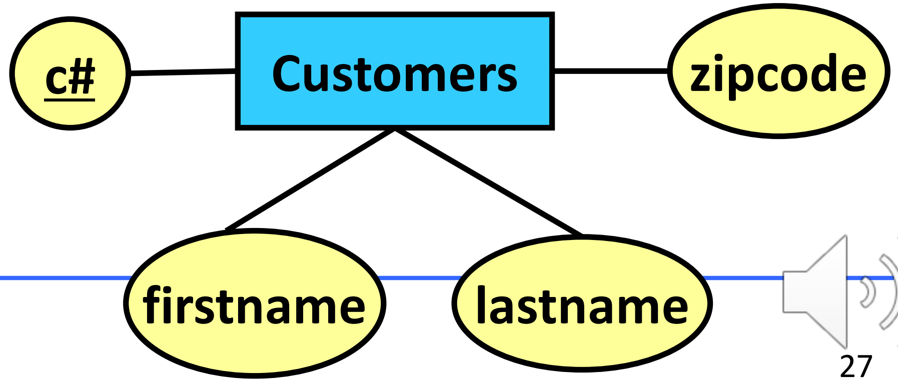
- Each customer is identified by a unique customer number, and has a first name, last names, and a zip code.

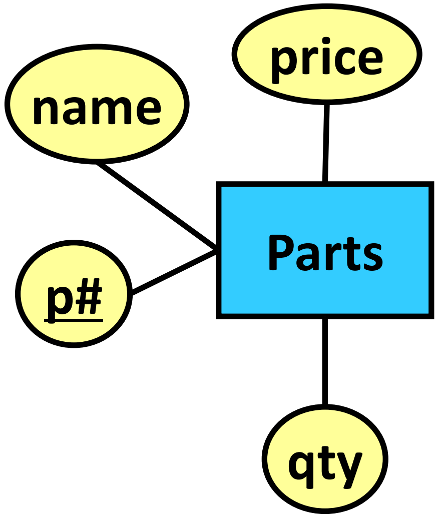




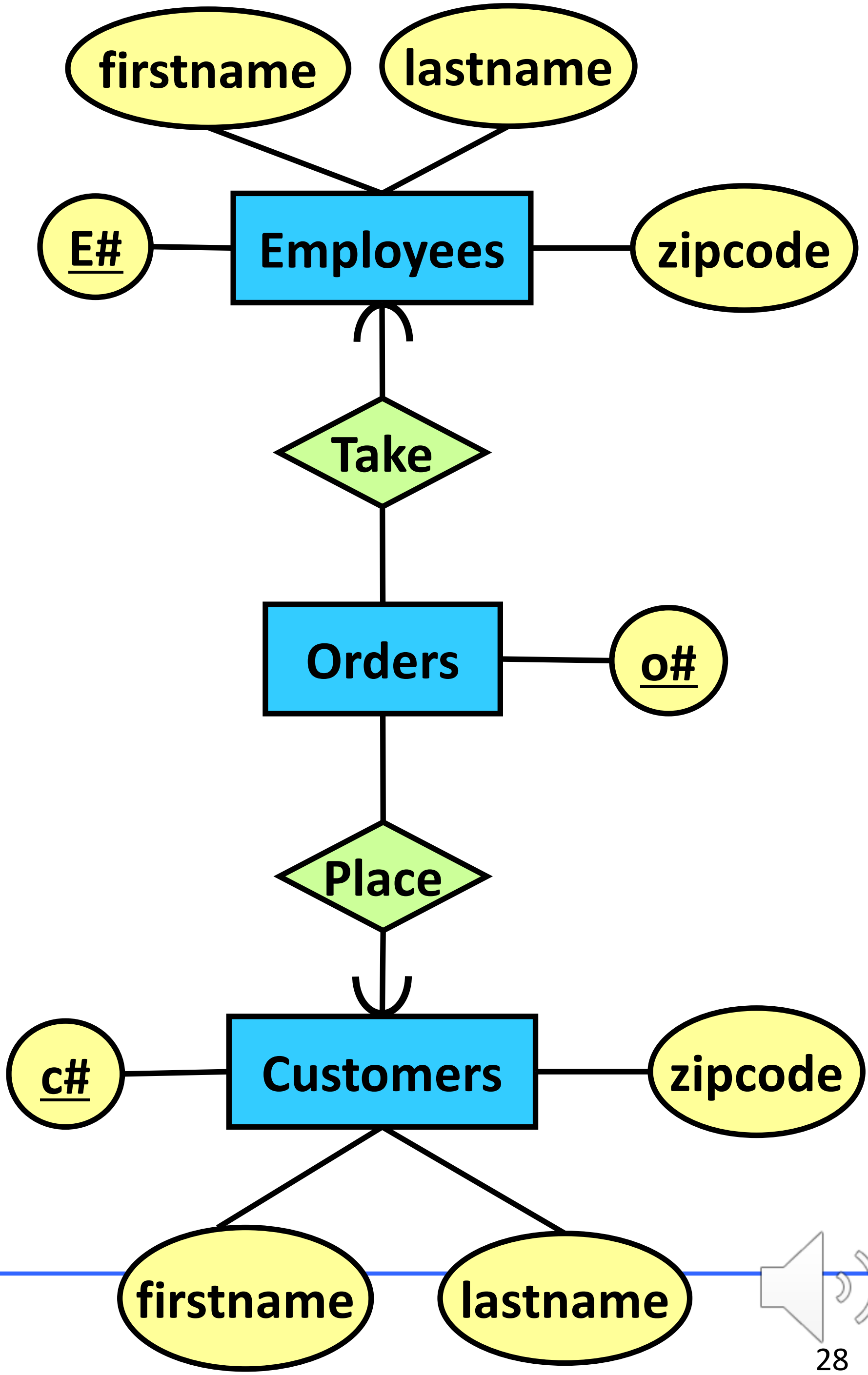


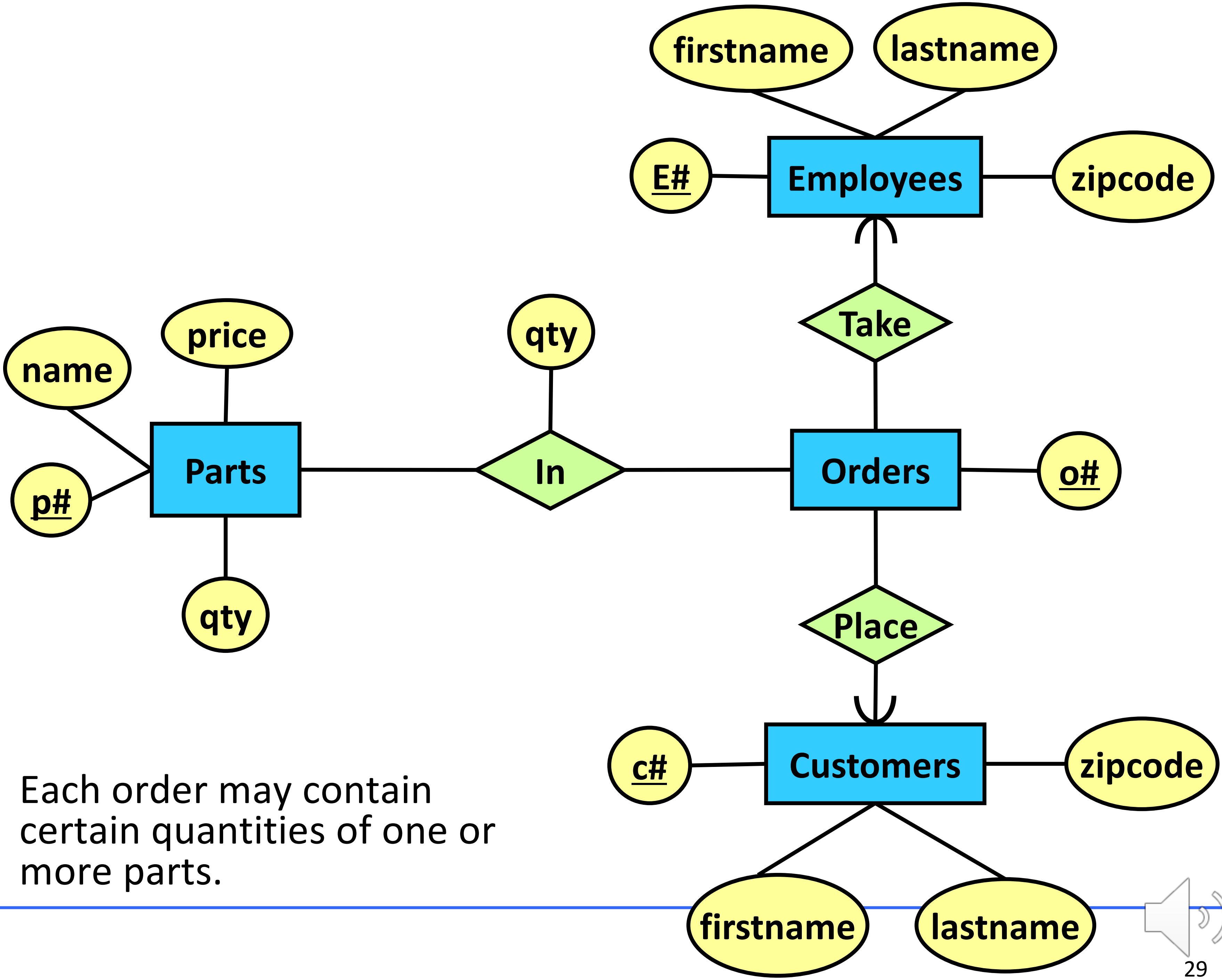
- Each part being sold is identified by a unique part number. It has a part name, a price, and a quantity in stock.



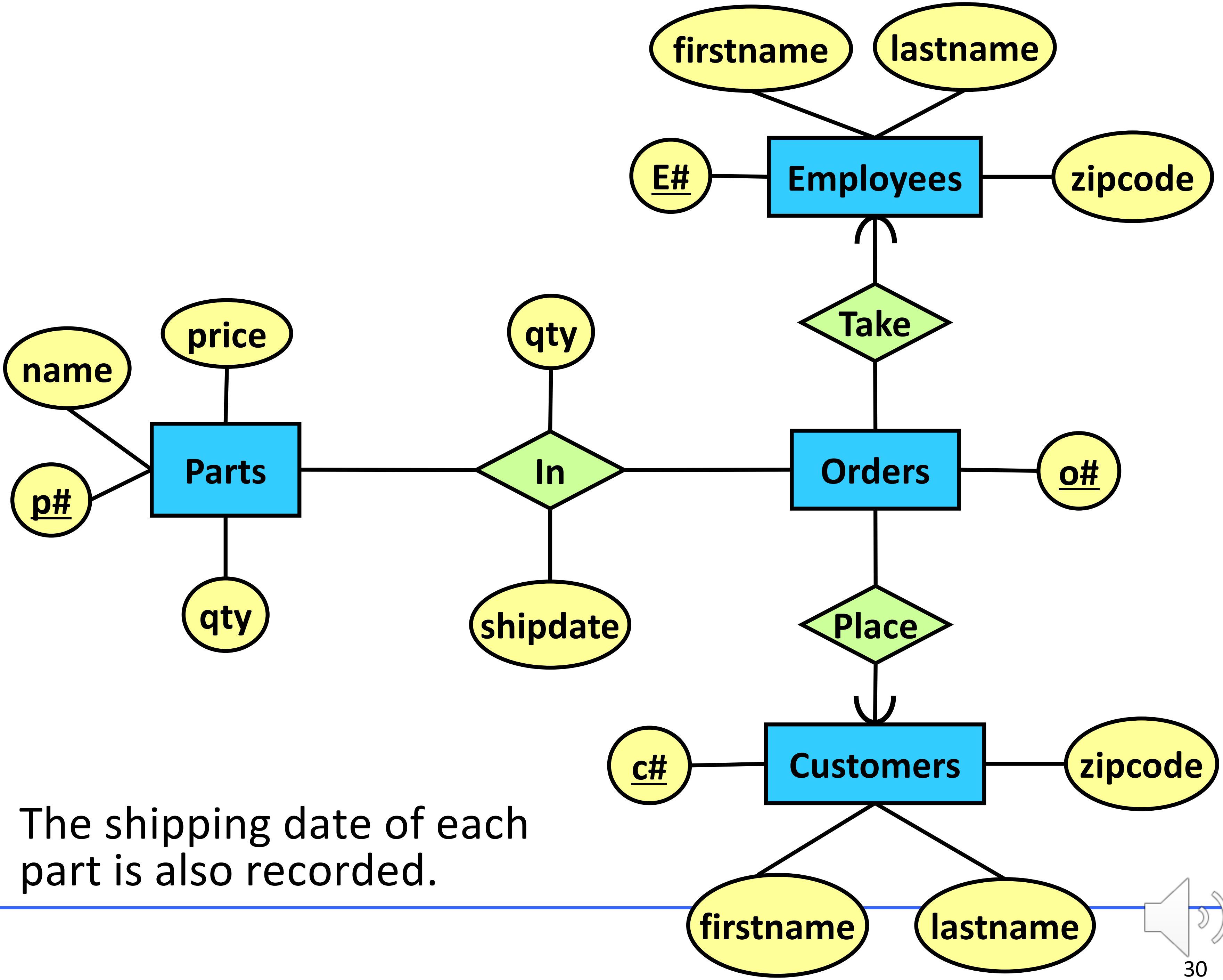


- Each order placed by a customer is taken by one employee and is given a unique order number.





- Each order may contain certain quantities of one or more parts.

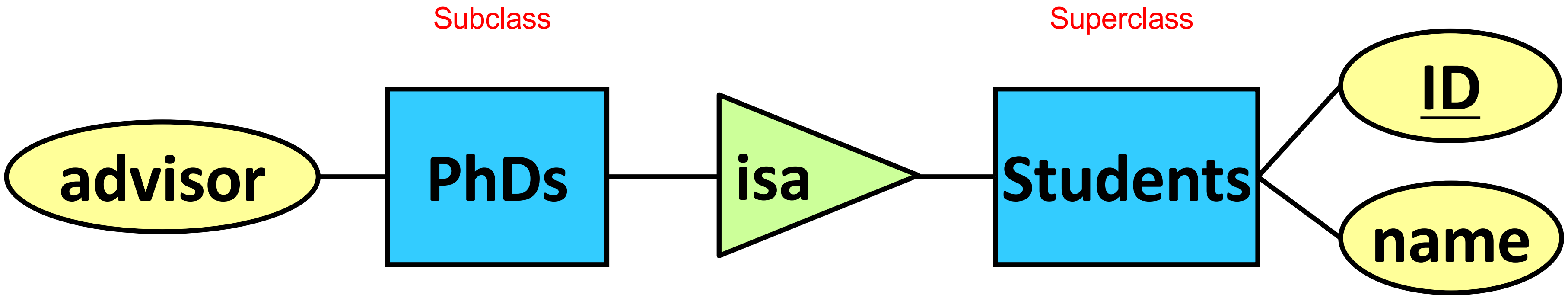


- The shipping date of each part is also recorded.

This Lecture

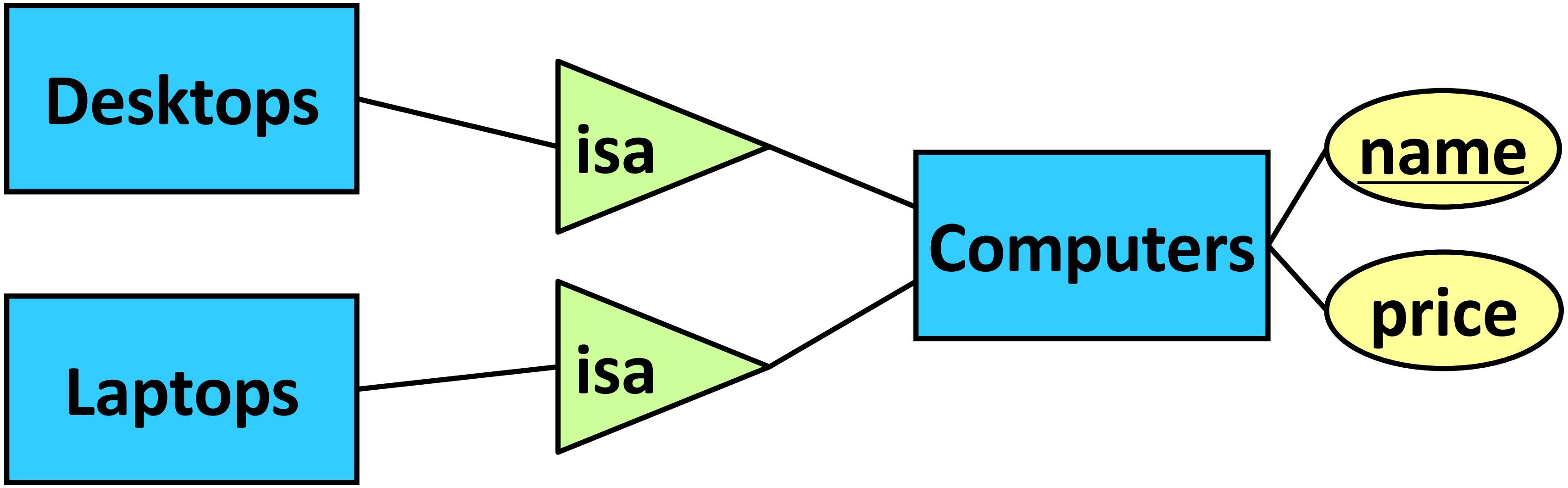
- Constraints
- Subclasses ⬅
- Weak Entity Sets

Subclass

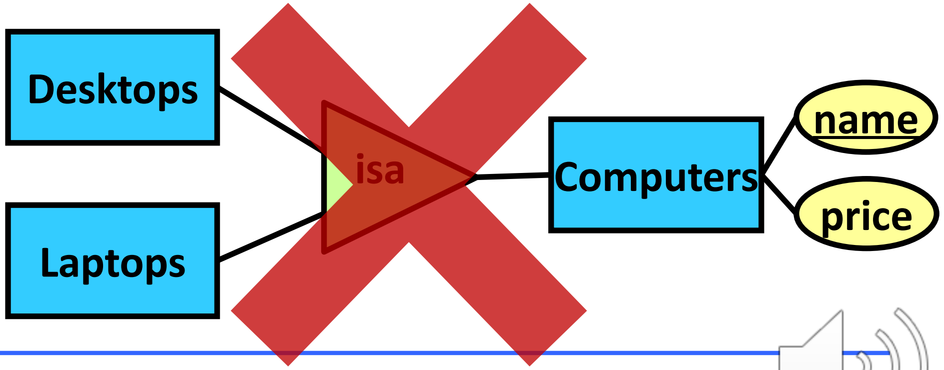


- PhDs are a special type of Students
- Subclass = Special type
- The connection between a subclass and its superclass is captured by the isa relationship, which is represented using a triangle
- Key of a subclass = key of its superclass
- Example: Key of Phds = Students.ID
- Students is referred to as the superclass of PhDs

Subclass



- An entity set can have multiple subclasses
- Example
  - Superclass: Computers
  - Subclass 1: Desktop
  - Subclass 2: Laptop

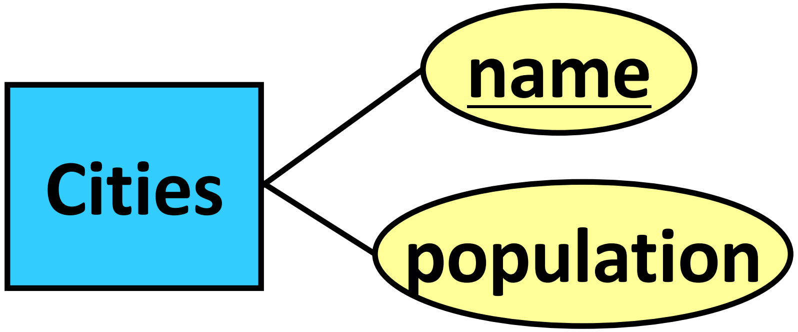


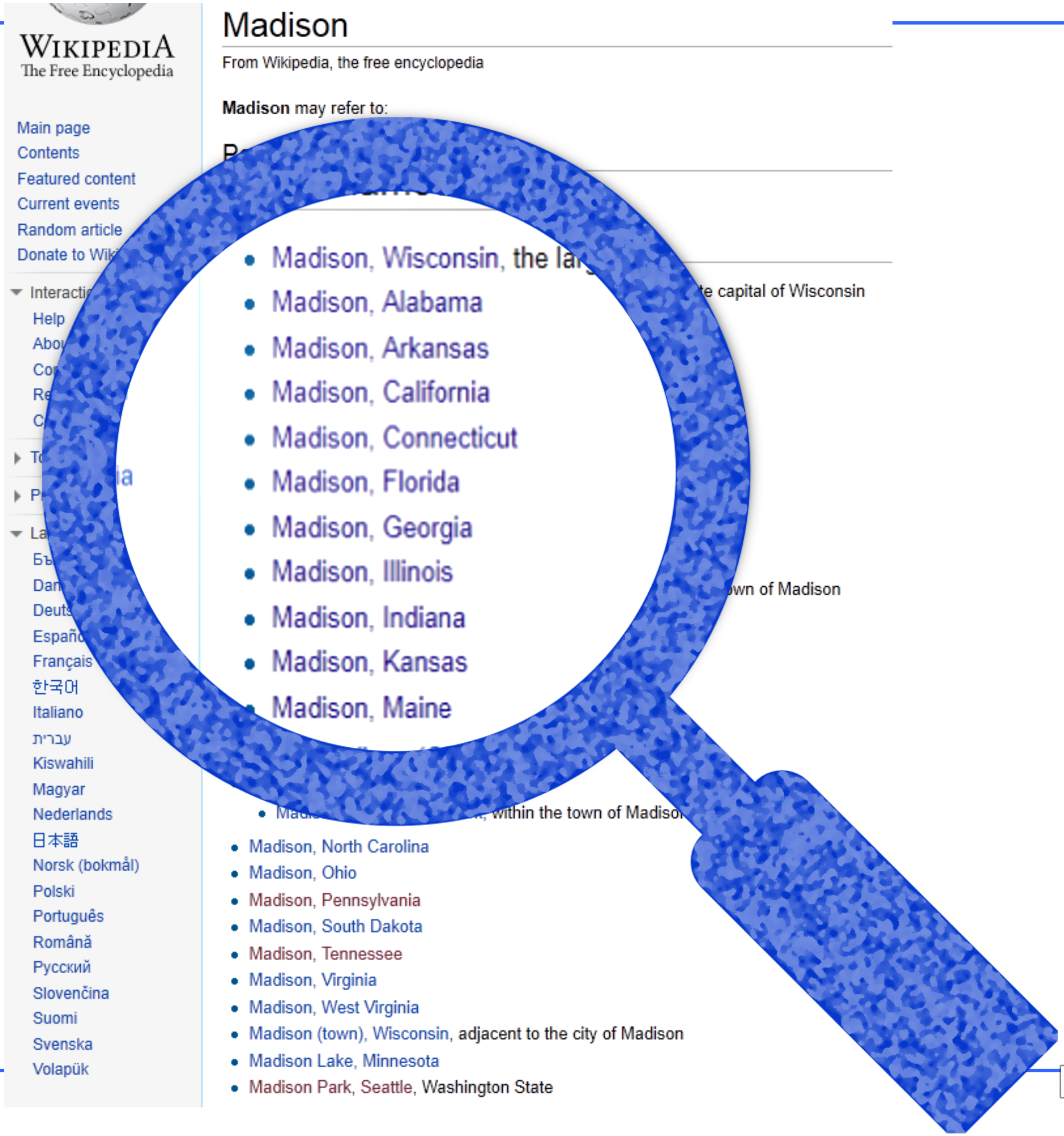
This Lecture

- Constraints
- Subclasses
- Weak Entity Sets ⇦

Weak Entity Sets

- Weak entity sets are a special type of entity sets that
  - cannot be uniquely identified by their own attributes
  - needs attributes from other entities to identify themselves
- Example: Cities in USA
- Problem: there are cities with identical names





WIKIPEDIA
The Free Encyclopedia

- Main page
- Contents
- Featured content
- Current events
- Random article
- Donate to Wiki

Interacti
- Help
- Abo
- Co
- Re
- C

To

P

La
- Бъ
- Dan
- Deuts
- Españ
- Français
- 한국어
- Italiano
- עברית
- Kiswahili
- Magyar
- Nederlands
- 日本語
- Norsk (bokmål)
- Polski
- Português
- Română
- Русский
- Slovenčina
- Suomi
- Svenska
- Volapük

Madison

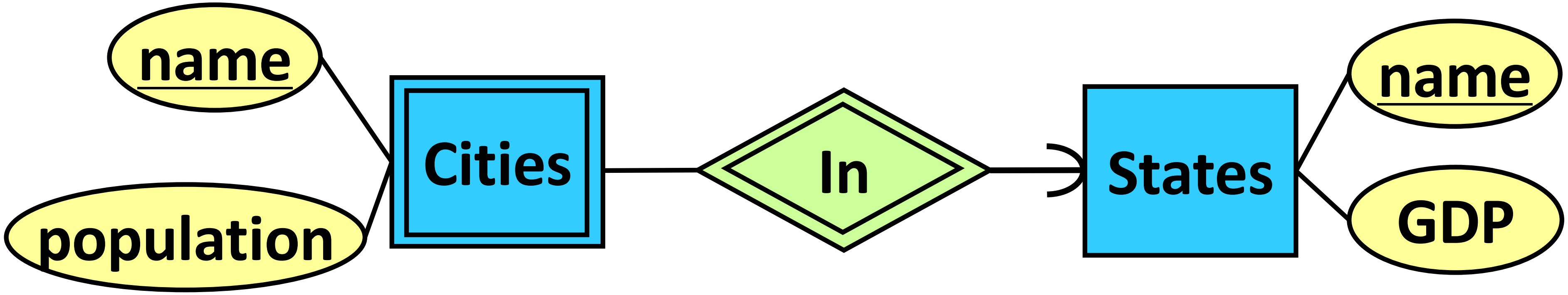
From Wikipedia, the free encyclopedia

Madison may refer to:

P

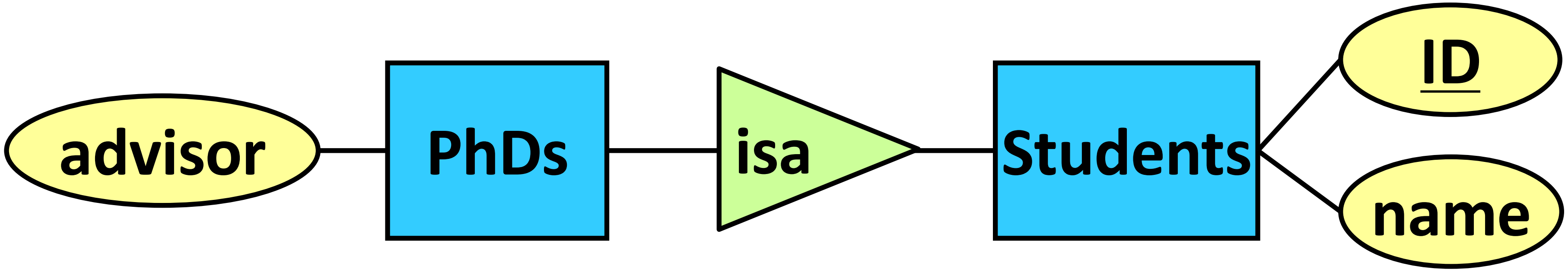
- Madison, Wisconsin, the la … te capital of Wisconsin
- Madison, Alabama
- Madison, Arkansas
- Madison, California
- Madison, Connecticut
- Madison, Florida
- Madison, Georgia
- Madison, Illinois
- Madison, Indiana
- Madison, Kansas
- Madison, Maine
- … wn of Madison
  - Madi … within the town of Madiso
- Madison, North Carolina
- Madison, Ohio
- Madison, Pennsylvania
- Madison, South Dakota
- Madison, Tennessee
- Madison, Virginia
- Madison, West Virginia
- Madison (town), Wisconsin, adjacent to the city of Madison
- Madison Lake, Minnesota
- Madison Park, Seattle, Washington State

Weak Entity Sets

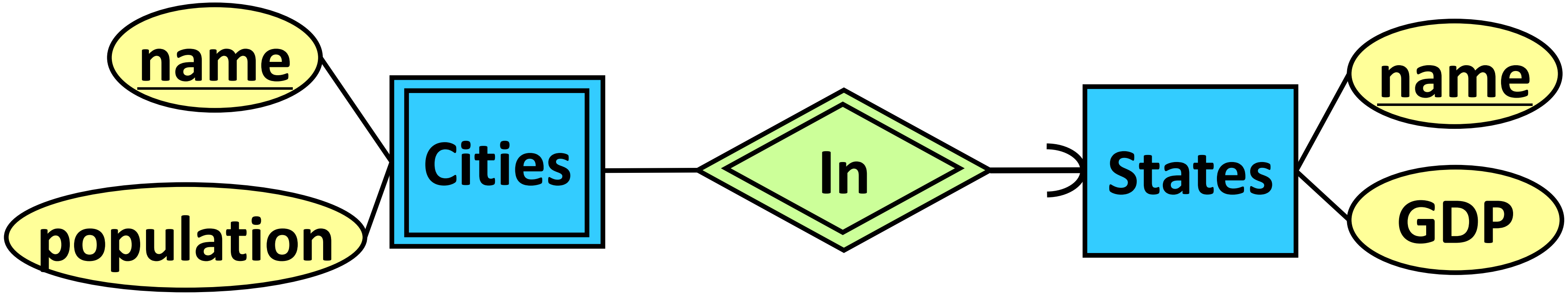


- Problem: there are cities with identical names
- Observation: cities in the same state would have different names
- Solution: make Cities a weak entity set associated with the entity set States
- The relationship In is called the supporting relationship of Cities
- Weak entity set = Double-lined rectangle
- Supporting relationship = Double-lined diamond
- The key of Cities = (State.name, Cities.name)

Subclass vs. Weak Entity Sets



- PhDs are a special type of Students

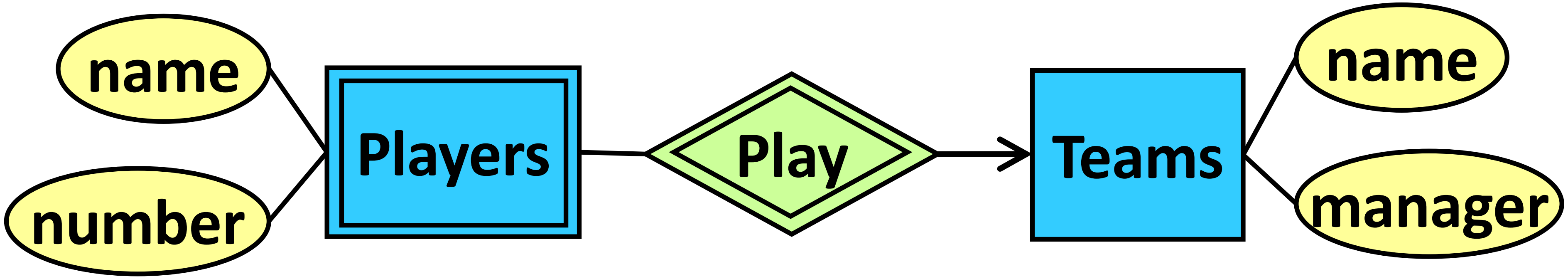


- Cities are NOT a special type of States

Exercise

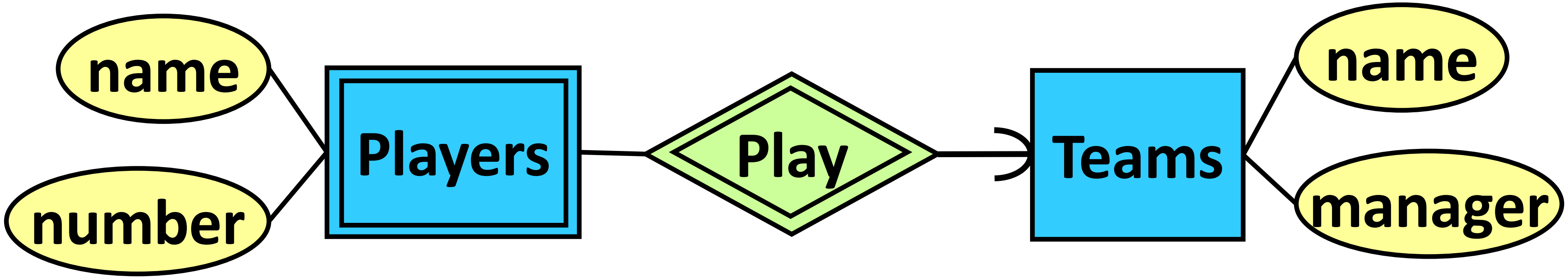
- Consider two entity sets: Players and Teams
- Each player has a name and a number
- Each team has a name and a manager
- Each player plays for exactly one team, and is uniquely identified within the team by his/her number
- Each team is uniquely identified by its name
- Different players may have the same name
- Draw a ER diagram that captures the above statements
- What is the key of Players?

Exercise



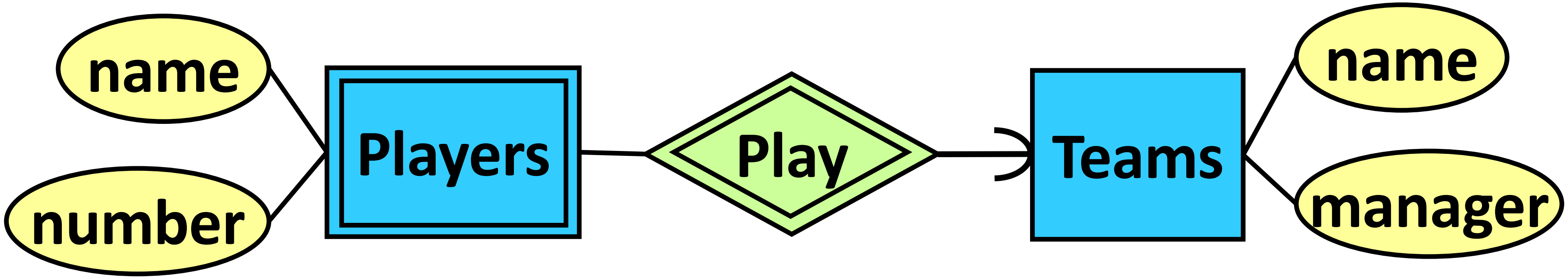
- Consider two entity sets: Players and Teams
- Each player has a name and a number
- Each team has a name and a manager
- Each player plays for exactly one team, and is uniquely identified within the team by his/her number

Exercise



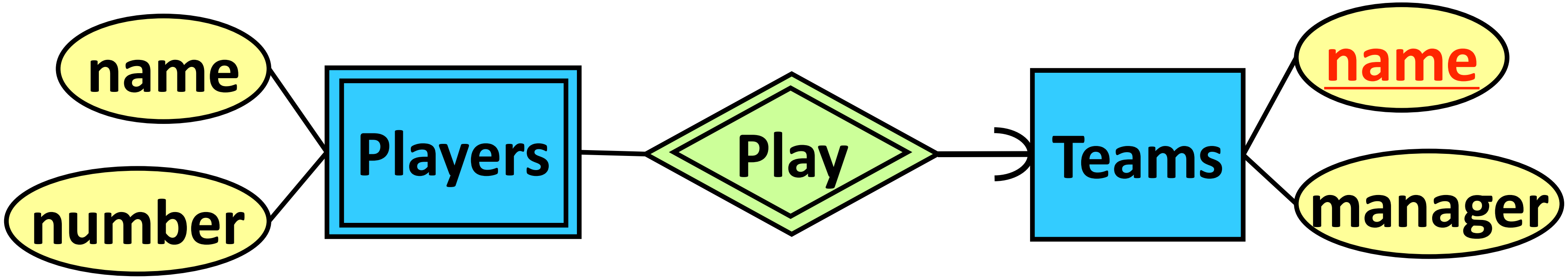
- Consider two entity sets: Players and Teams
- Each player has a name and a number
- Each team has a name and a manager
- Each player plays for exactly one team, and is uniquely identified within the team by his/her number
-
-
-
-

Exercise



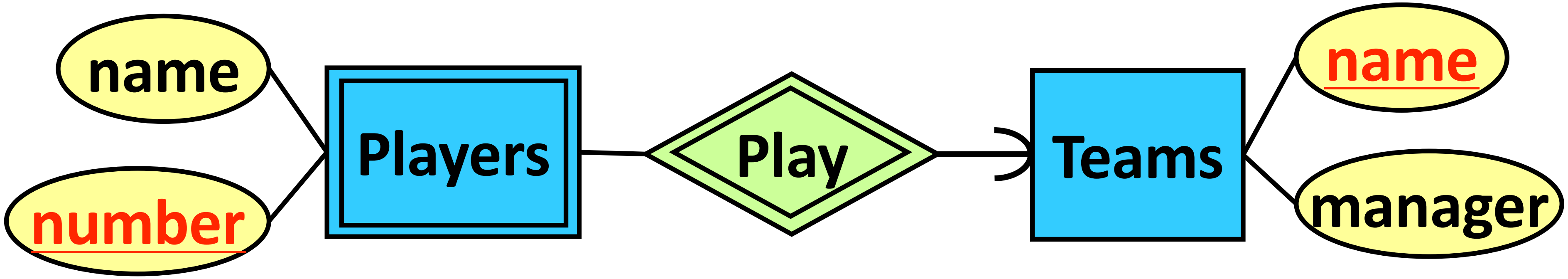
- Consider two entity sets: Players and Teams
- Each player has a name and a number
- Each team has a name and a manager
- Each player plays for exactly one team, and is uniquely identified within the team by his/her number
- Each team is uniquely identified by its name

Exercise



- Consider two entity sets: Players and Teams
- Each player has a name and a number
- Each team has a name and a manager
- Each player plays for exactly one team, and is uniquely identified within the team by his/her number
- Each team is uniquely identified by its name
- Different players may have the same name
- Draw a ER diagram that captures the above statements
- What is the key of Players?

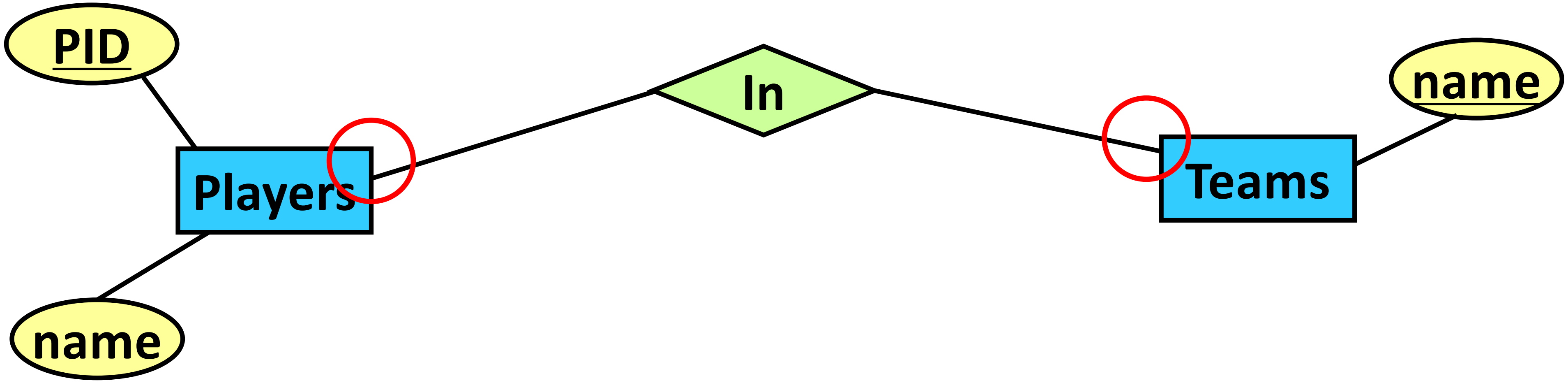
Exercise



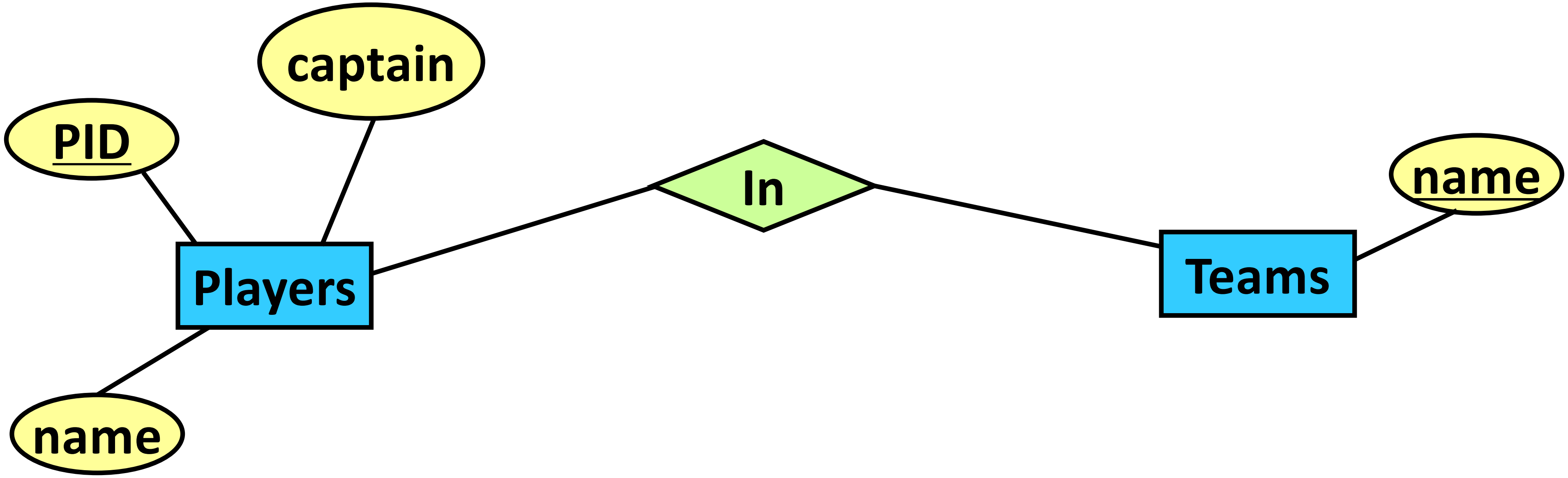
- Consider two entity sets: Players and Teams
- Each player has a name and a number
- Each team has a name and a manager
- Each player plays for exactly one team, and is uniquely identified within the team by his/her number
- Each team is uniquely identified by its name
- Different players may have the same name
- Draw a ER diagram that captures the above statements
- What is the key of Players? (Players.number, Teams.name)

Exercise: ER-Diagram Design

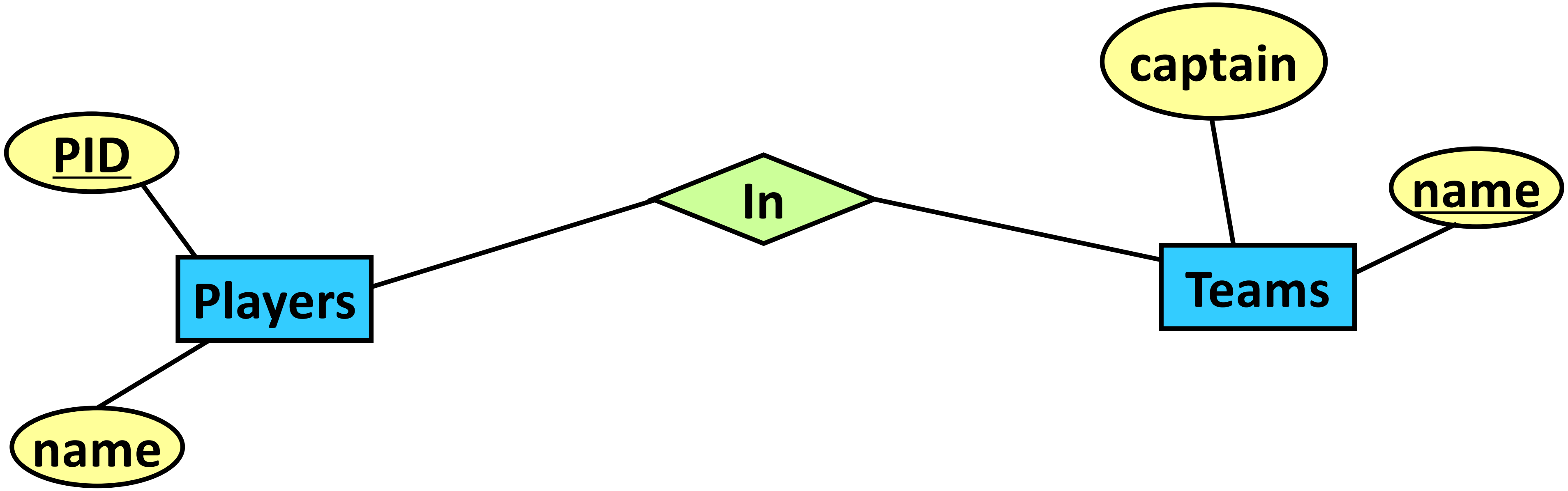
- Record info about teams, players, and their fans, including:
  - For each team, its name, its players, its team captain (who is also a player)
  - For each player, his/her name, and the history of teams on which he/she has played, including the start and ending dates for each team
  - For each fan, his/her name, favorite teams, favorite players
- Additional information:
  - Each team has at least one player, and exactly one captain
  - Each team has a unique name
  - Two players (or two fans) may have the same name
  - Each fan has at least one favorite team and at least one favorite player



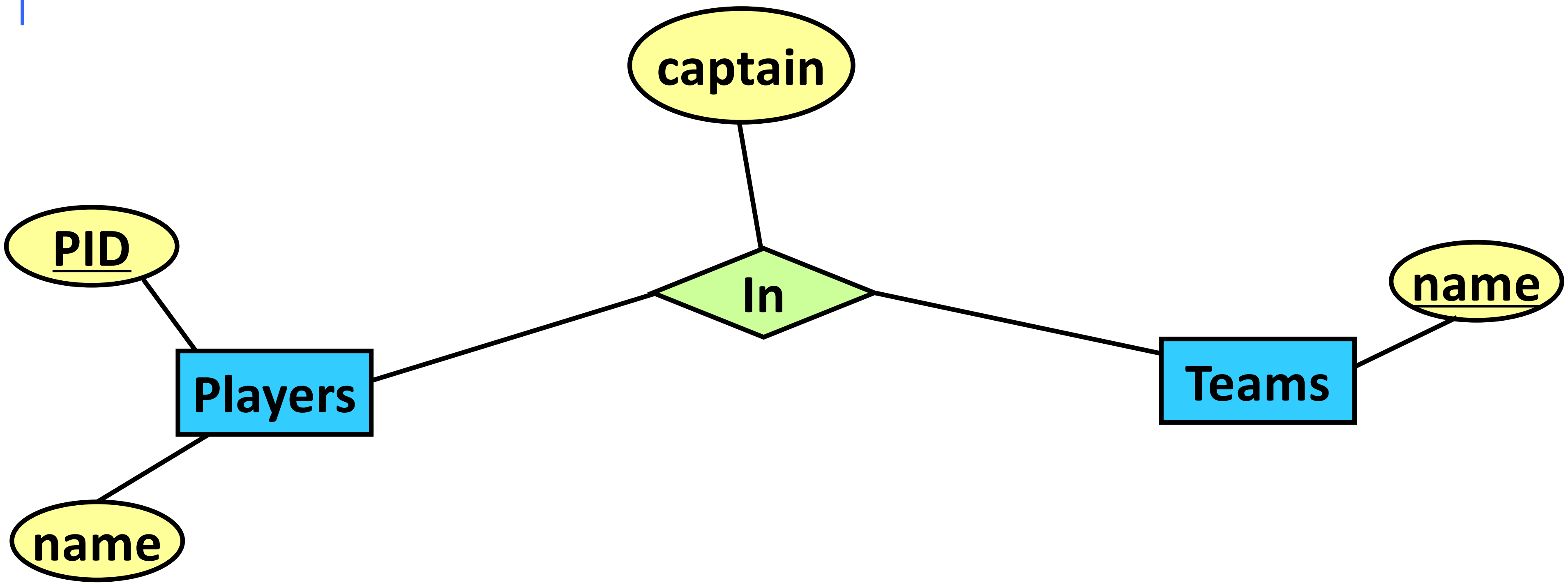
For each team, its name, its players, its team captain (who is also a player)



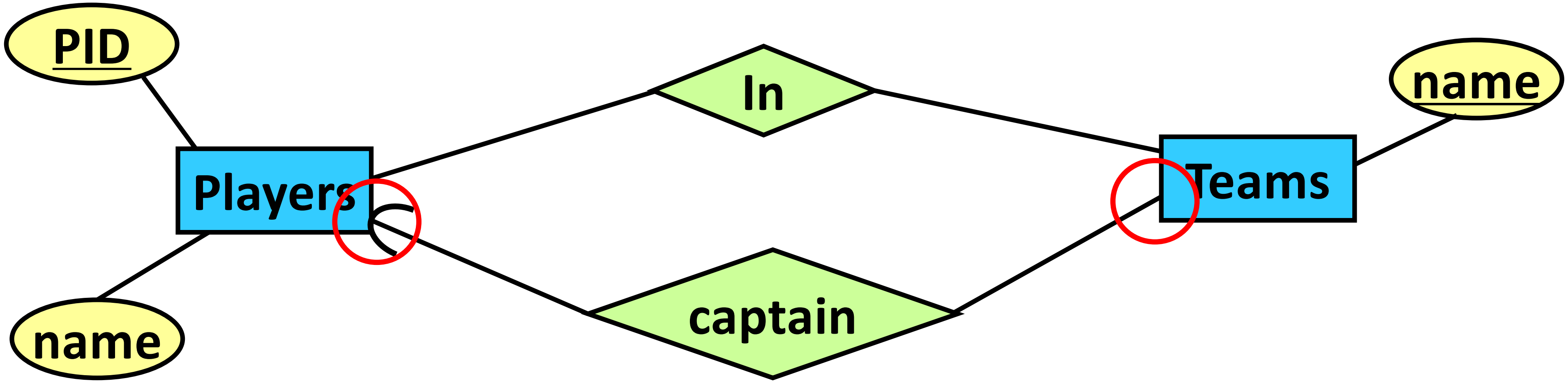
For each team, its name, its players, its team captain (who is also a player)



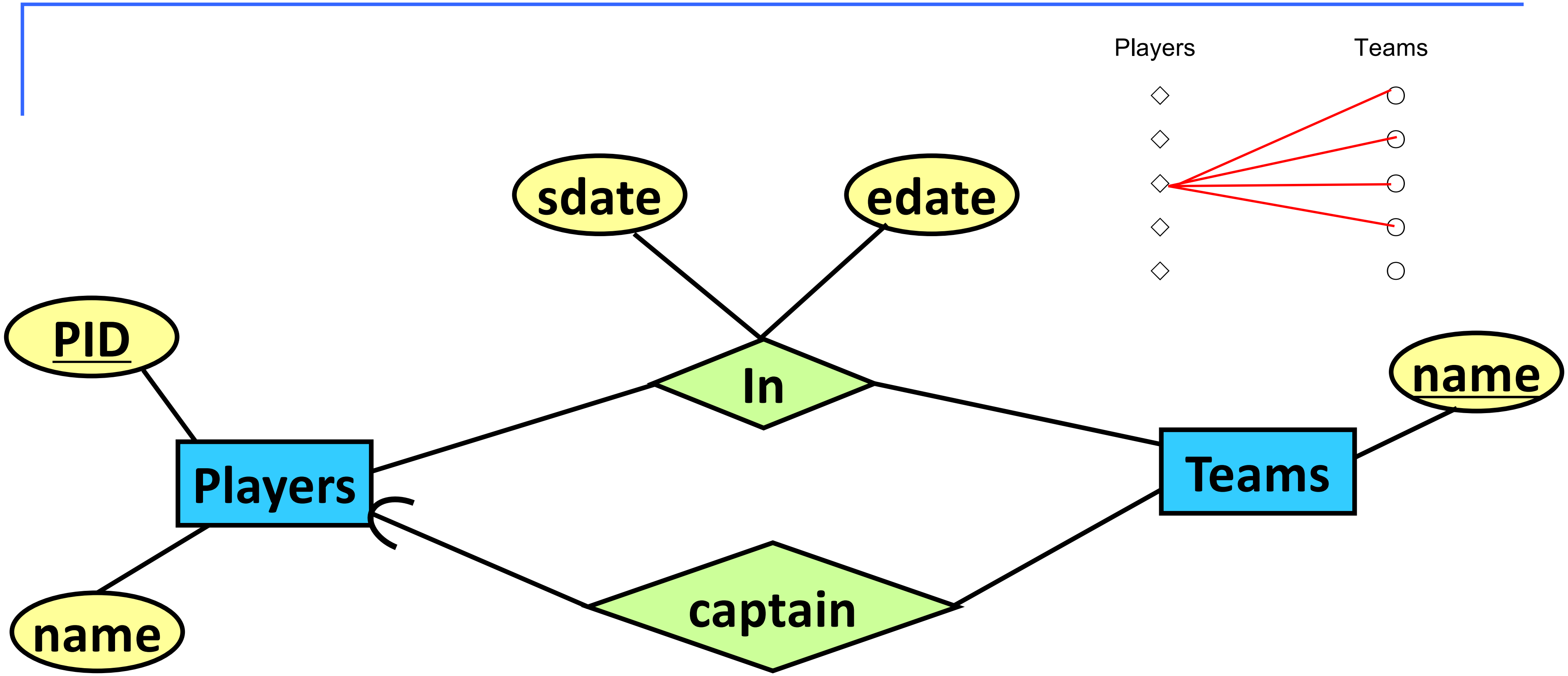
For each team, its name, its players, its team captain (who is also a player)



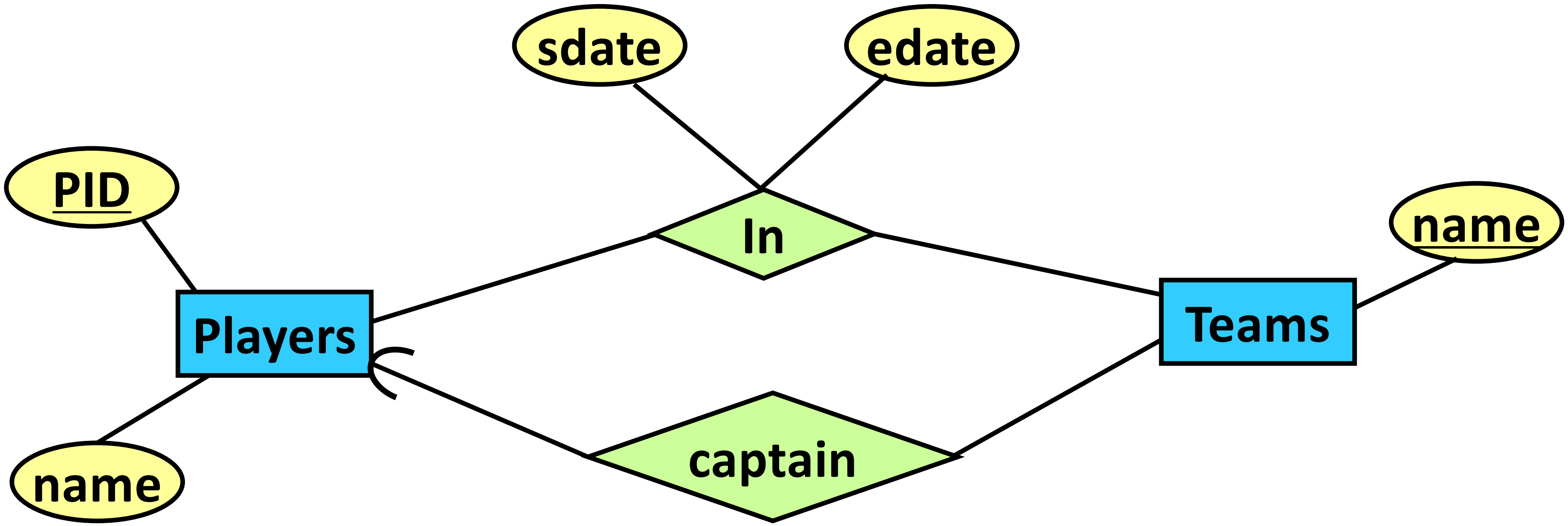
For each team, its name, its players, its team captain (who is also a player)



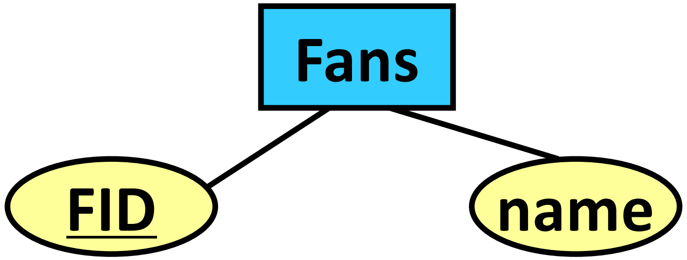
For each team, its name, its players, its team captain (who is also a player)

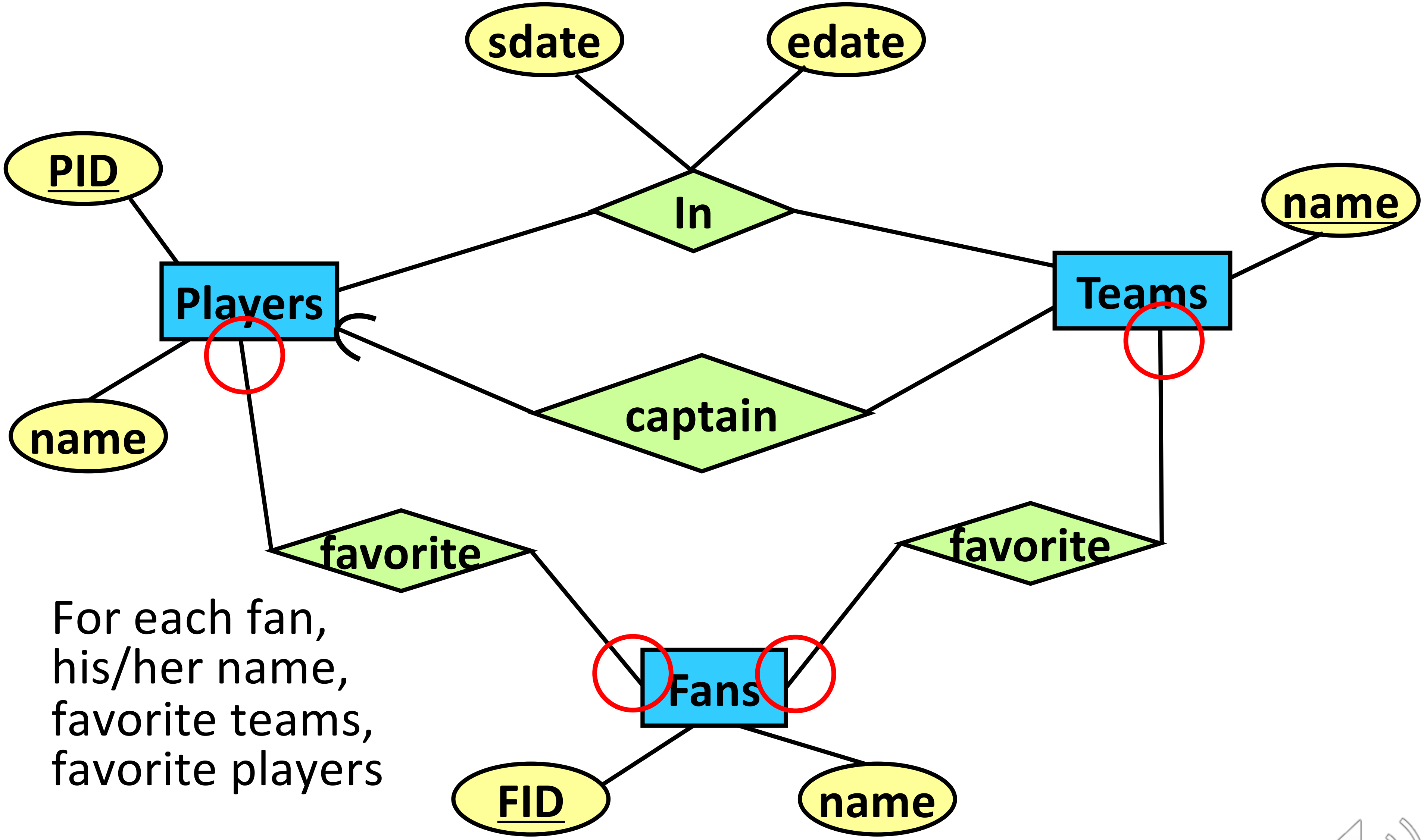


For each player, his/her name, and the history of teams on which he/she has played, including the start and ending dates for each team

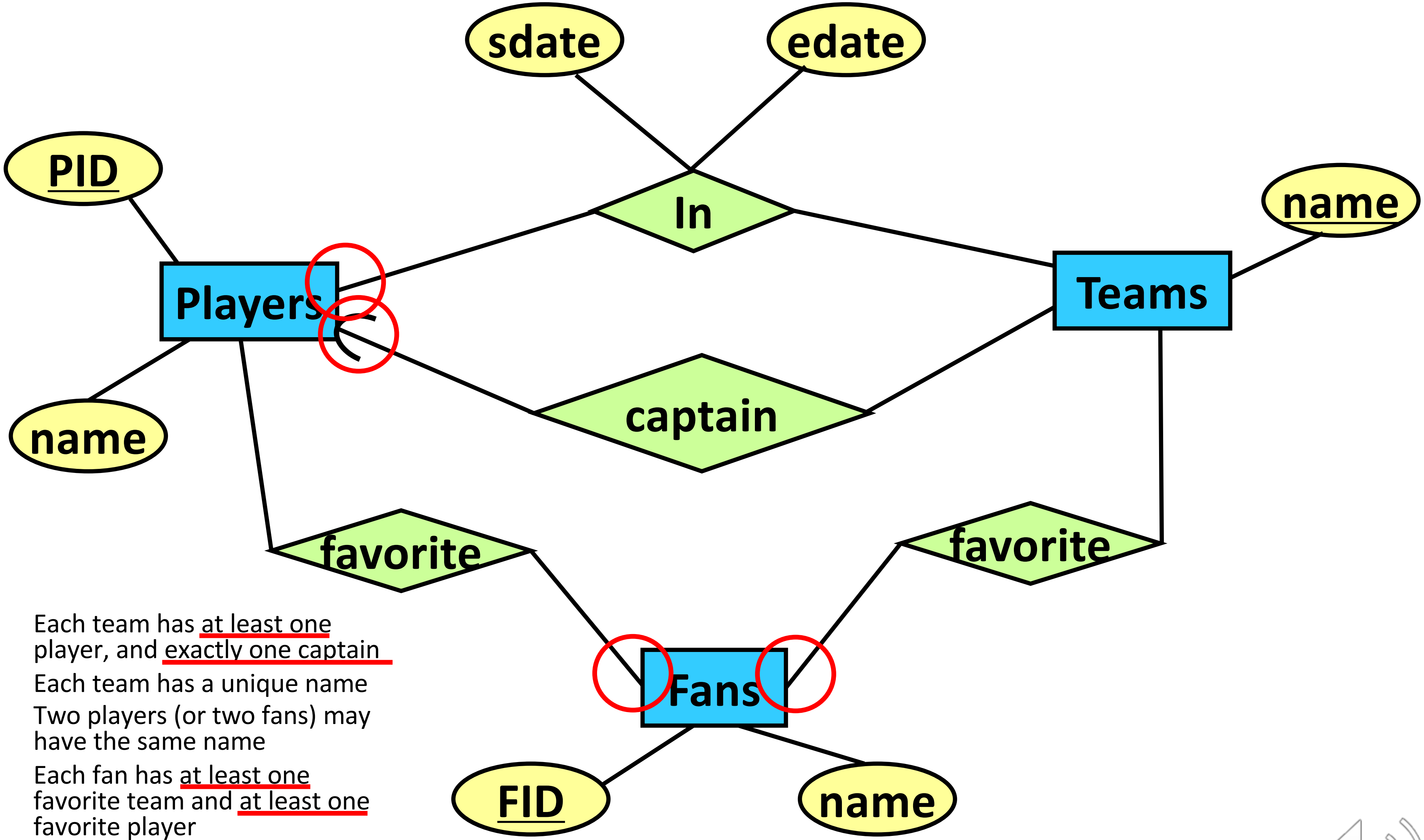


For each fan,
his/her name,
favorite teams,
favorite players





For each fan,
his/her name,
favorite teams,
favorite players



Each team has at least one player, and exactly one captain

Each team has a unique name

Two players (or two fans) may have the same name

Each fan has at least one favorite team and at least one favorite player

To continue in

Topic 1: Entity Relationship Diagram (3)