

COL 362/632

Introduction To Database Management Systems

Data Modelling – Entity-Relationship (ER) Model

Kaustubh Beedkar

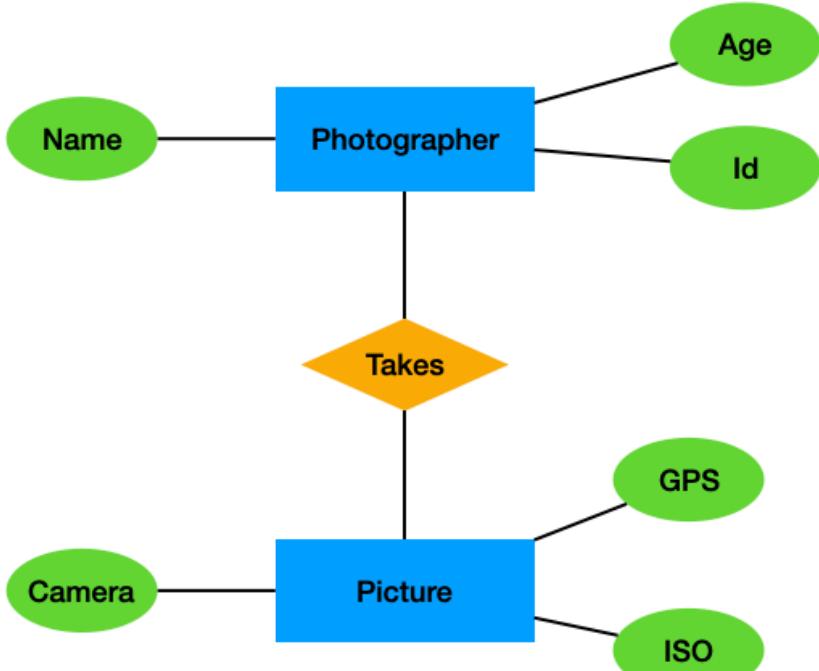
Administrivia

- No class on Friday 10.01.2025!
- Course evaluation has been updated
- NEW: In-class quizzes (no surprises here)
 - Total 8 (best 5 will be considered)
 - More information will soon be on the website

Syllabus

Overview

1. Design and Programming
- ▶ Data Modelling
 - ER Model
 - Relational Model
 - Schema Design
- ▶ Relational Algebra
- ▶ SQL (Basic + Advanced)
2. Implementation and Internals
3. Misc. Topics



From RealWorld to Database Schema



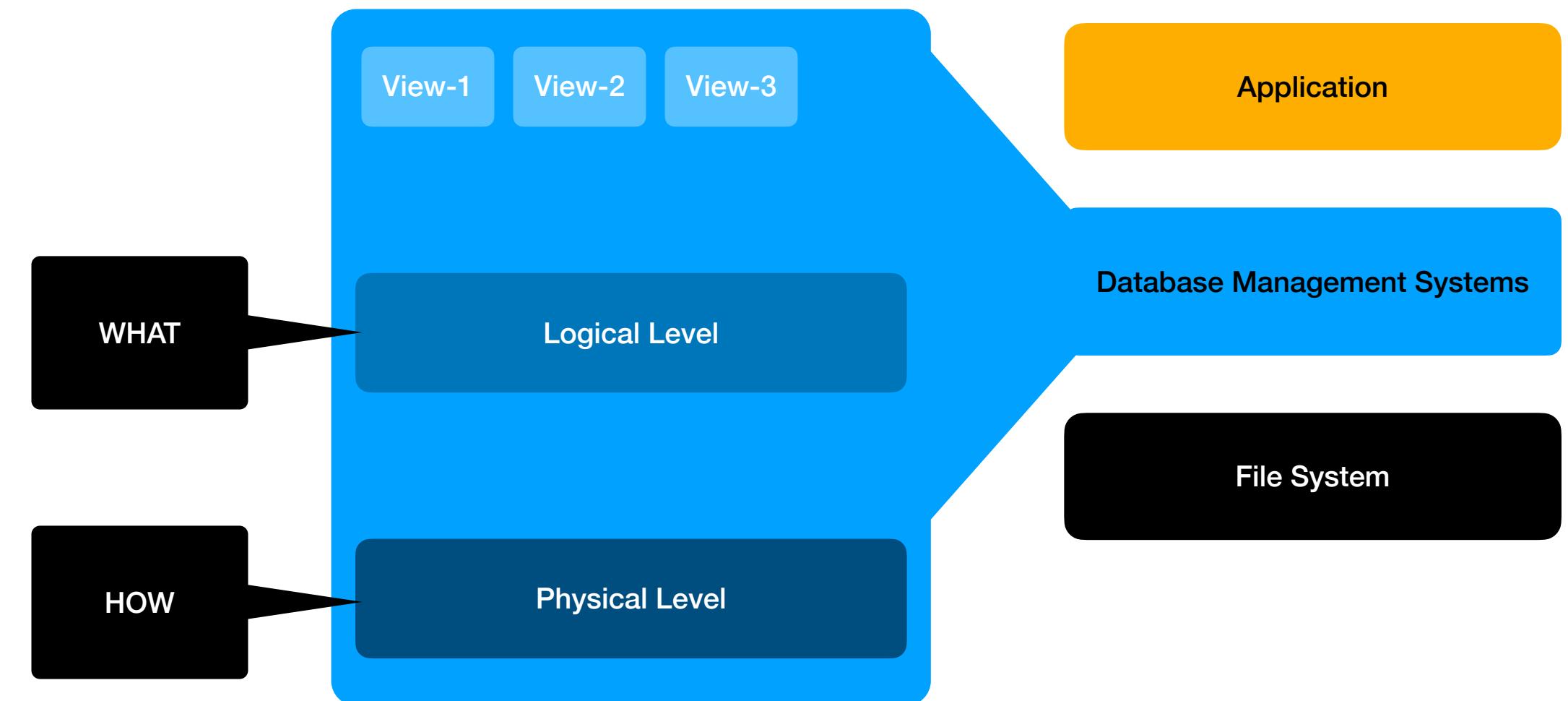
Database Schema
↓

Photographer				Picture			
First Name	Last Name	Age	ID	Camera	GPS	ISO	Aperture

Photographer = {[Name: String, Age: Integer, ID: Integer]}
Picture = {[Camera: String, GPS: Point]}

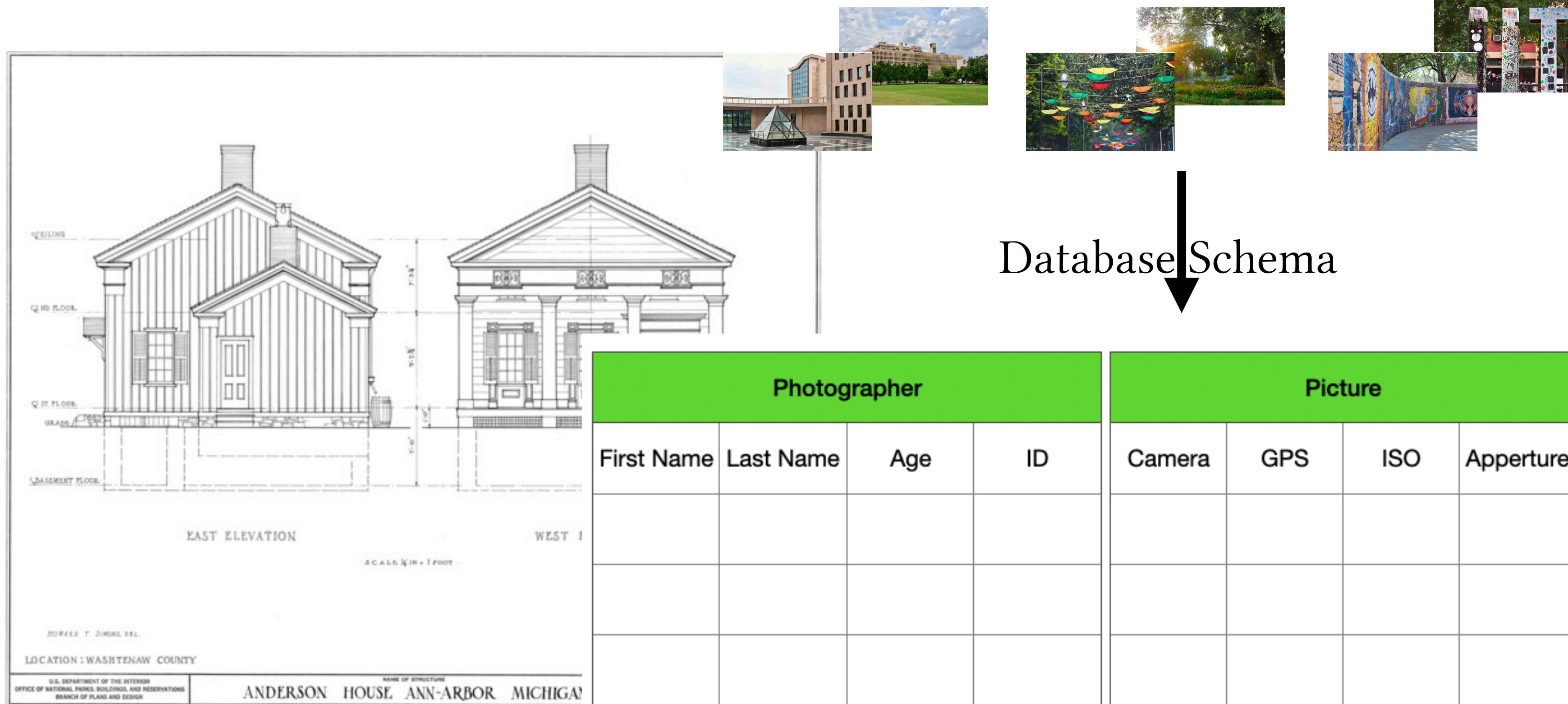
Data Model

- A model is an abstract representation
- Do you know your data?
 - ▶ Is there something missing?
 - ▶ Are there redundancies?
 - ▶ How do various components of your data fit together?
- To design a database, we first need a schema
 - ▶ Recall logical and physical independence
- Different modelling approaches
 - ▶ ER, Relational, Arrays/Matrices, Hierarchical, Network,...

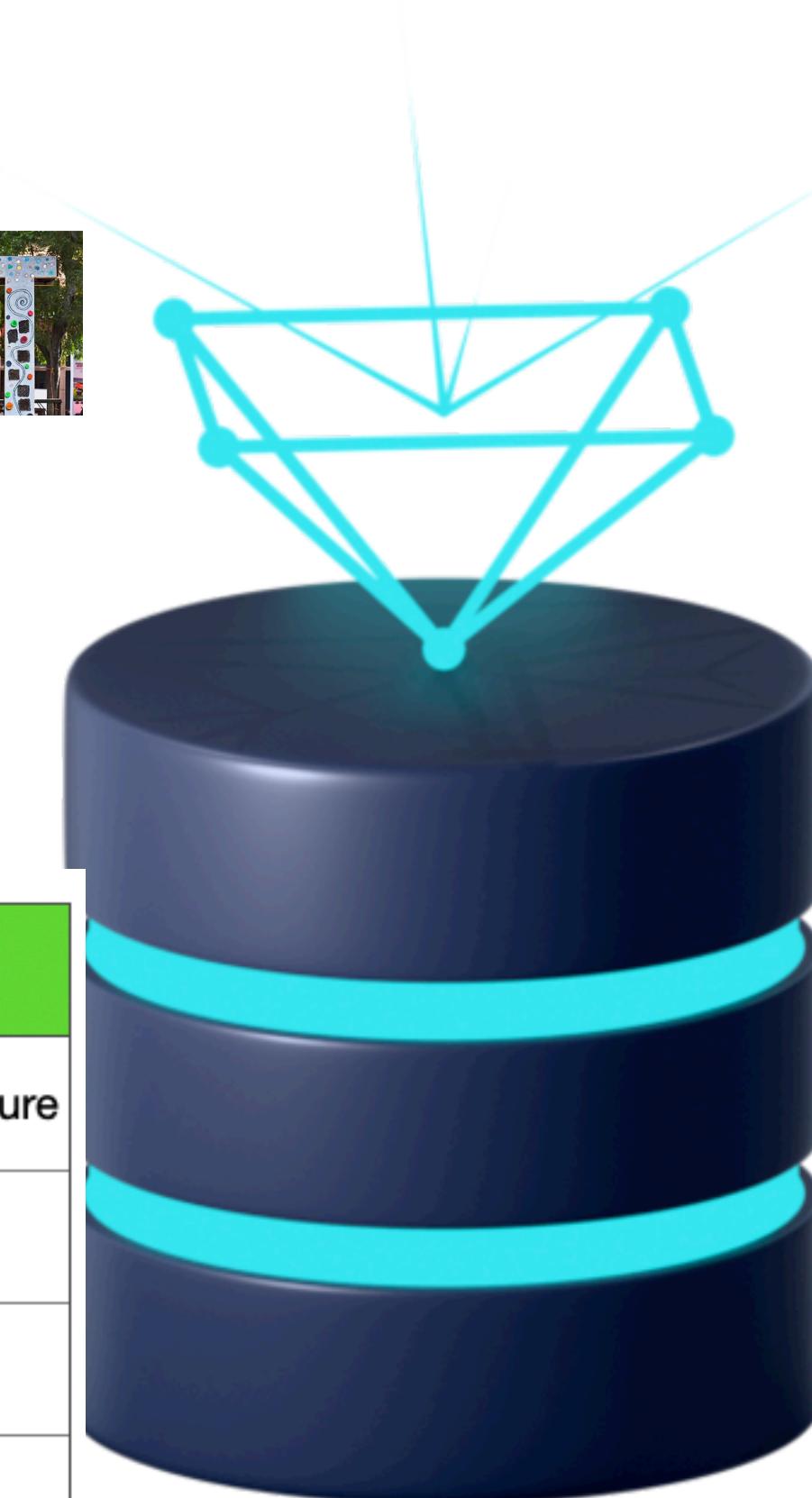


Entity Relationship Model

From RealWorld to Database Schema



Database Schema



Photographer = {[Name: String, Age: Integer, ID: Integer]}
Picture = {[Camera: String, GPS: Point]}

ER model serve as a blueprint for database design

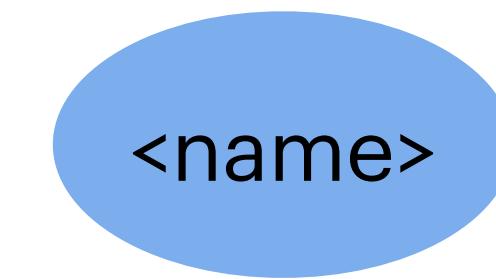
Components of the ER Model

Entities



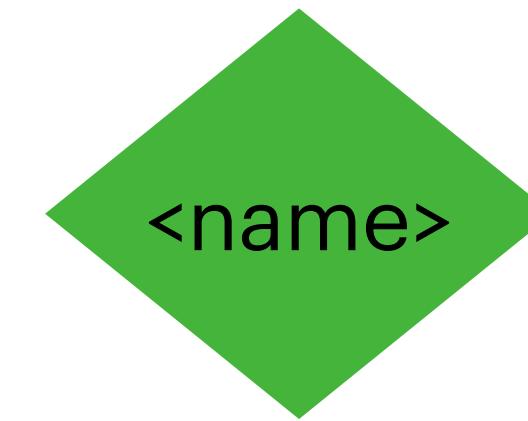
Objects or concepts with data stored about them
Examples: Student, Course, Orders, Supplier, etc.

Attributes



Simple, composite, derived, multi-valued
Examples: Name (simple), address (composite), age (derived)

Relationships



Associations between entities
Examples: “Player plays in a team”, “Customer makes an order”

Know Your Data

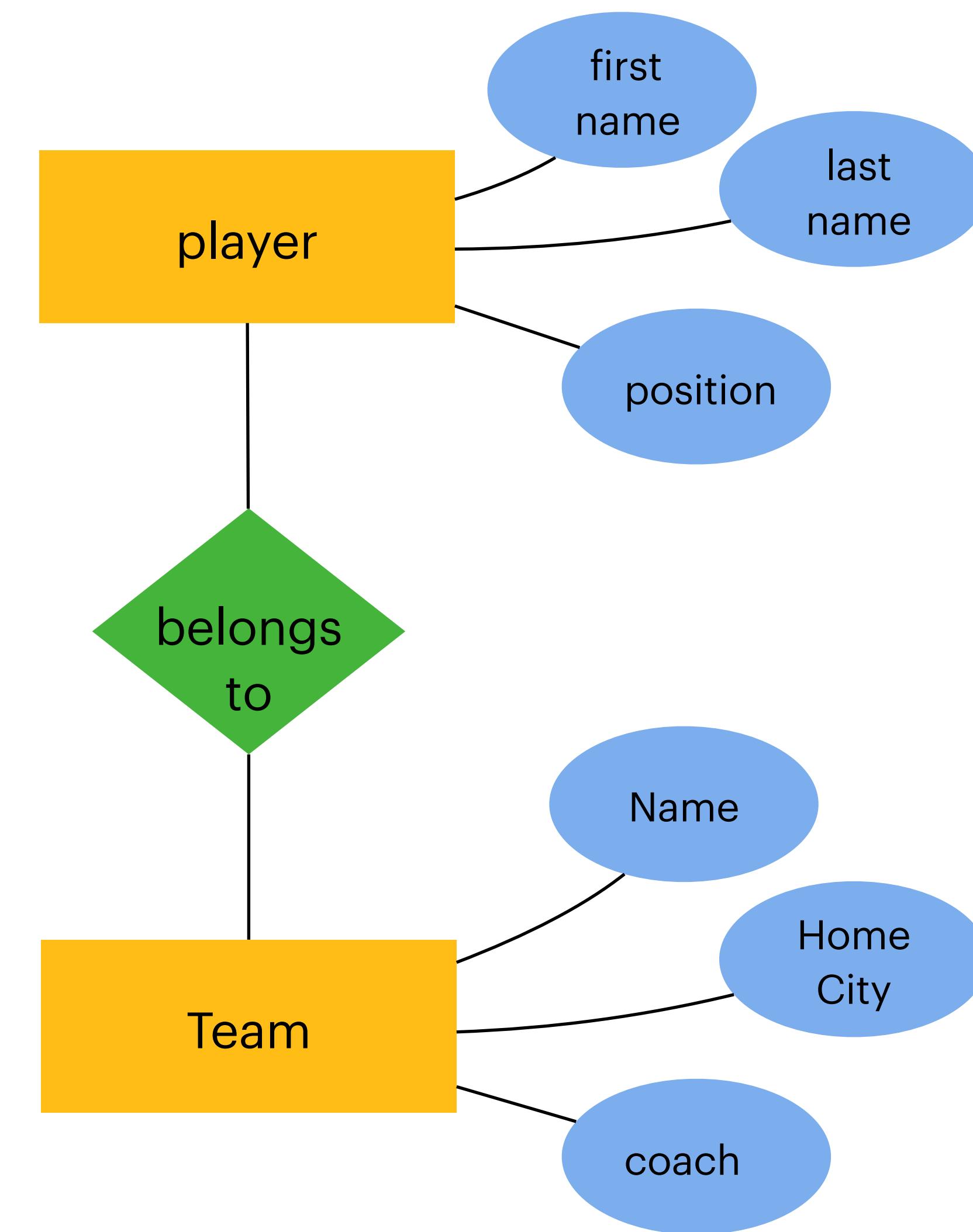
Cricket League - case study

- League has many teams (franchise)
- Each team has a name, home city, coach, captain, and set of players
- A player belongs to a team
- A player has
 - a name
 - position (1-down, 2-down, etc.)
 - skill (batter, blower, wicket keeper)
 - injury records
- Team captain is also a player
- Two teams play a game
 - home team, away team, each game has a score, and date
- ...

Requirement Analysis

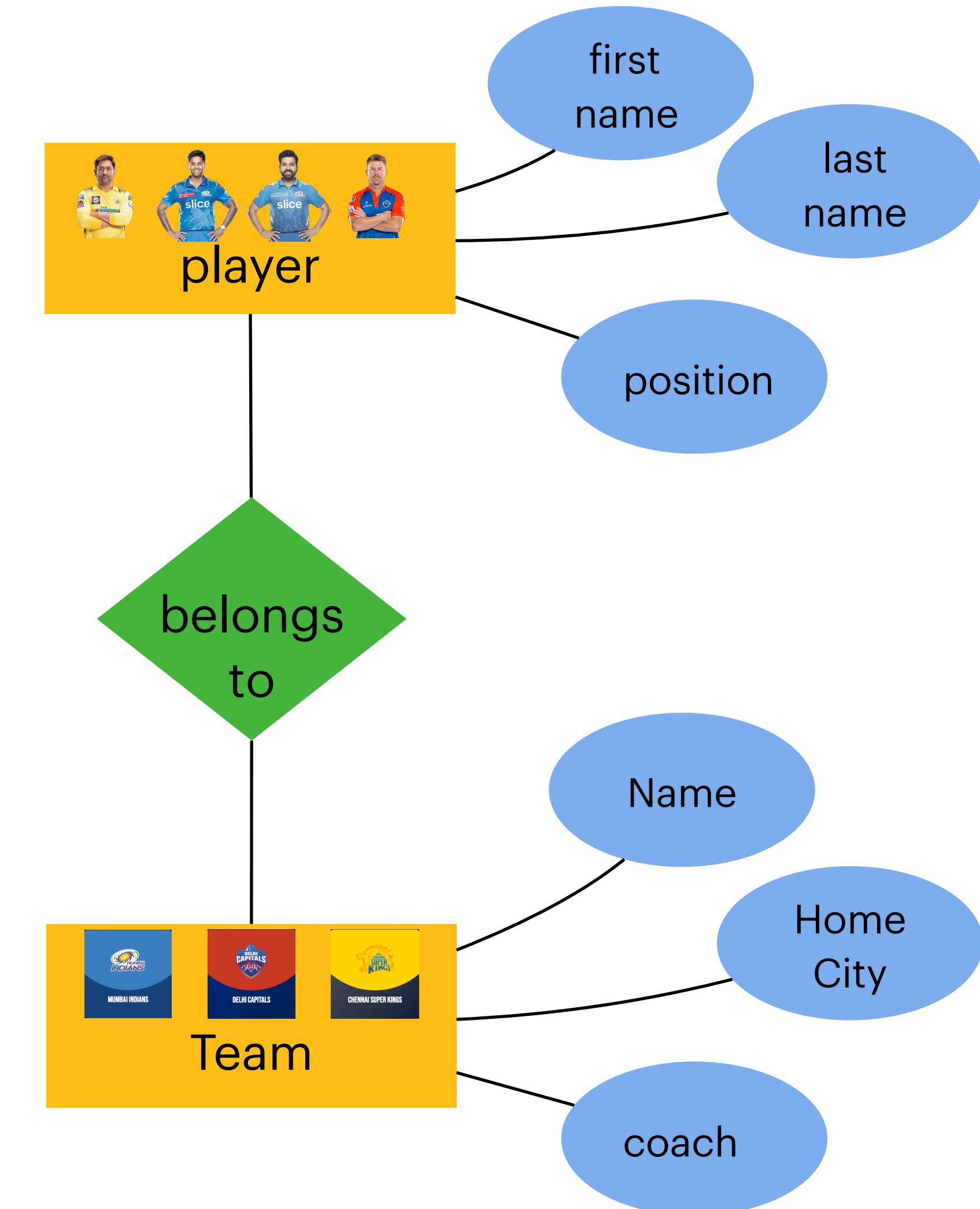


Entities, Relation, and Attributes



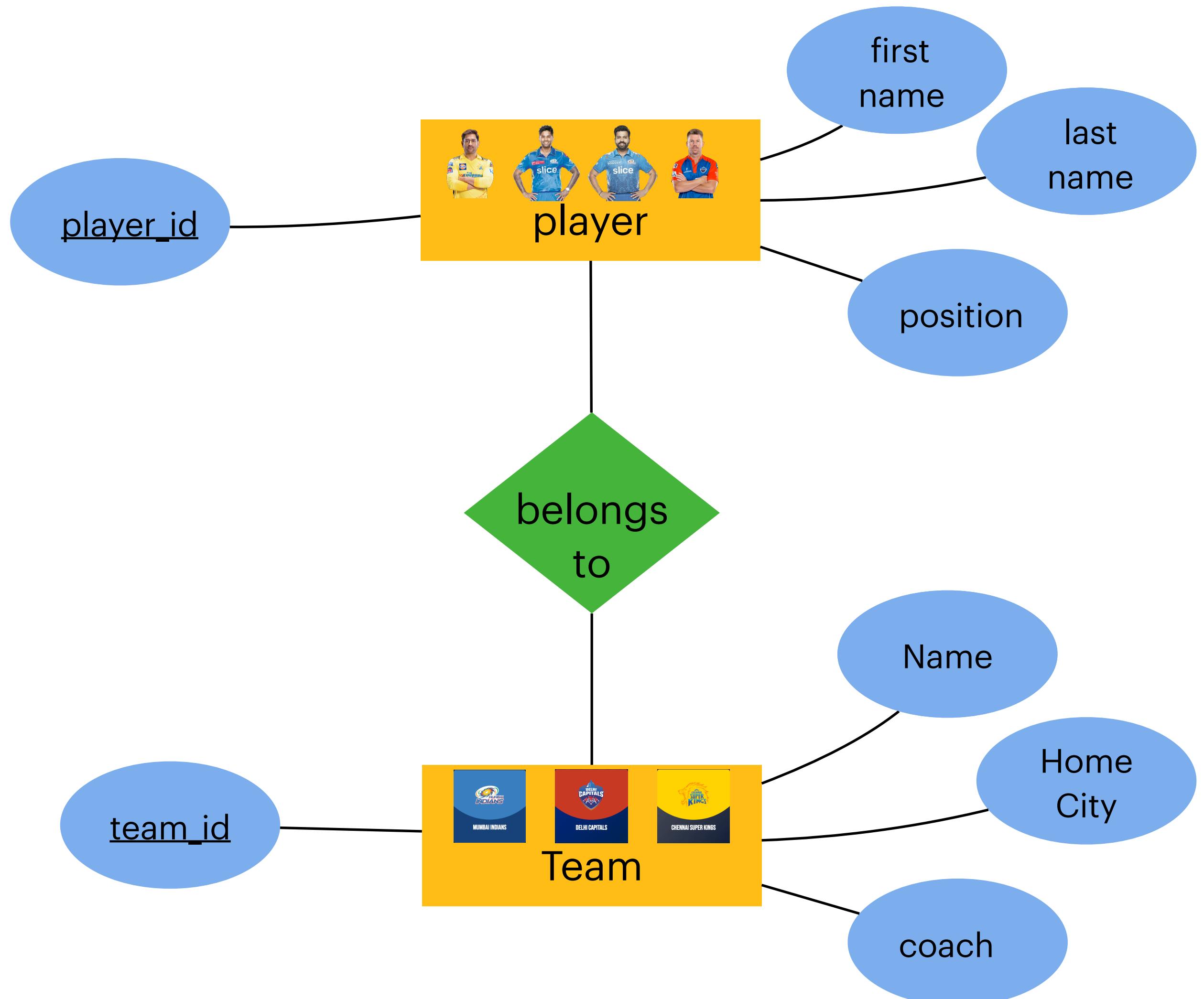
Entity and Entity set

- Entity
 - ▶ An object that is distinguishable from other objects
- Entity Set
 - ▶ Collection of similar entities
 - ▶ Described using set of attributes

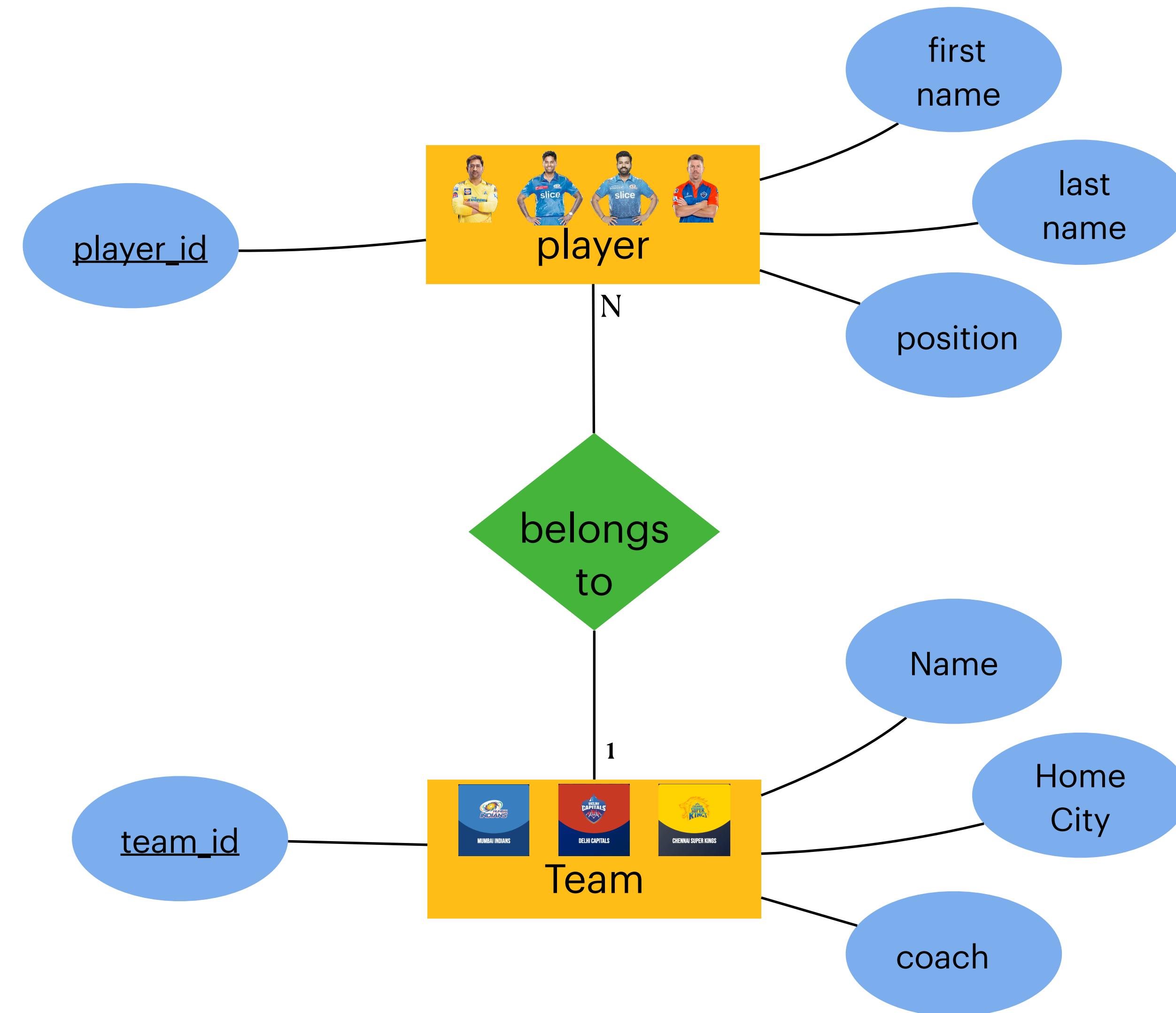


Keys

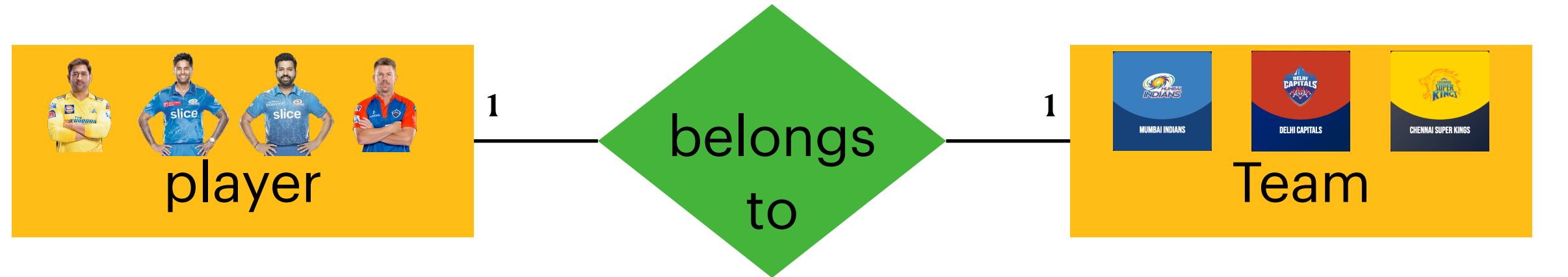
- A key attribute is an attribute that uniquely identifies each instance of an entity
- Value of a key attribute must be unique!
- Cannot by null
- Can be single or composite



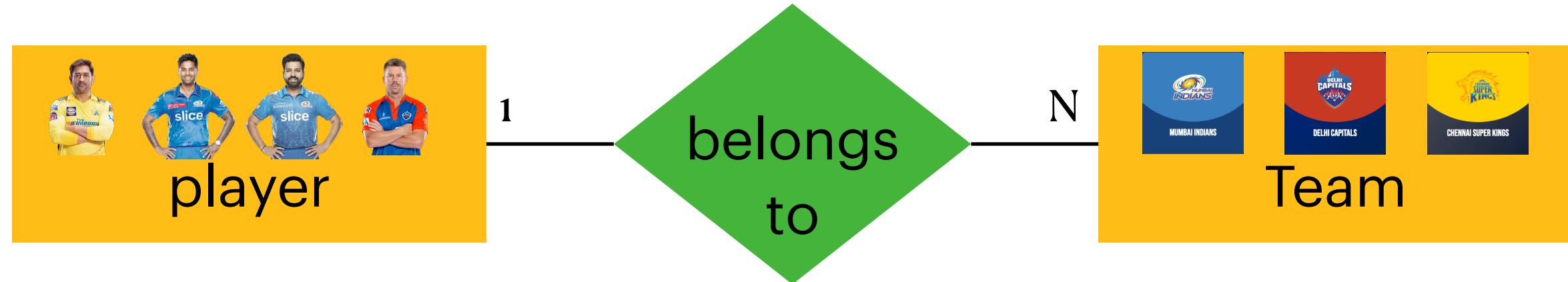
Cardinalities/Multiplicity and Participation Constraints



Types of relationships/ Relationship multiplicity



one-to-one



one-to-many



many-to-one

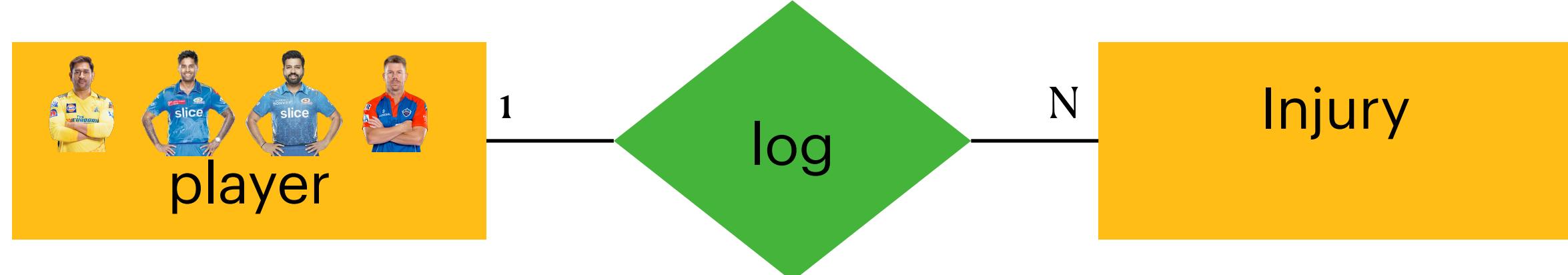


many-to-many

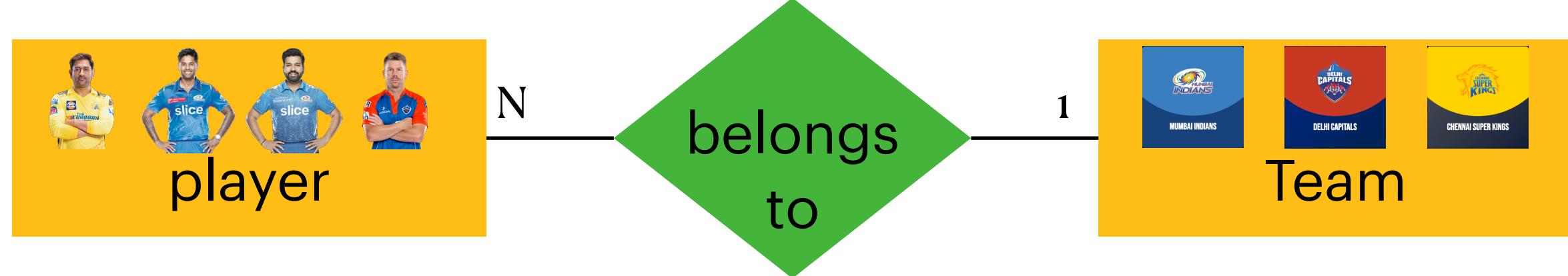
Types of relationships/ Relationship multiplicity



one-to-one



one-to-many



many-to-one



many-to-many

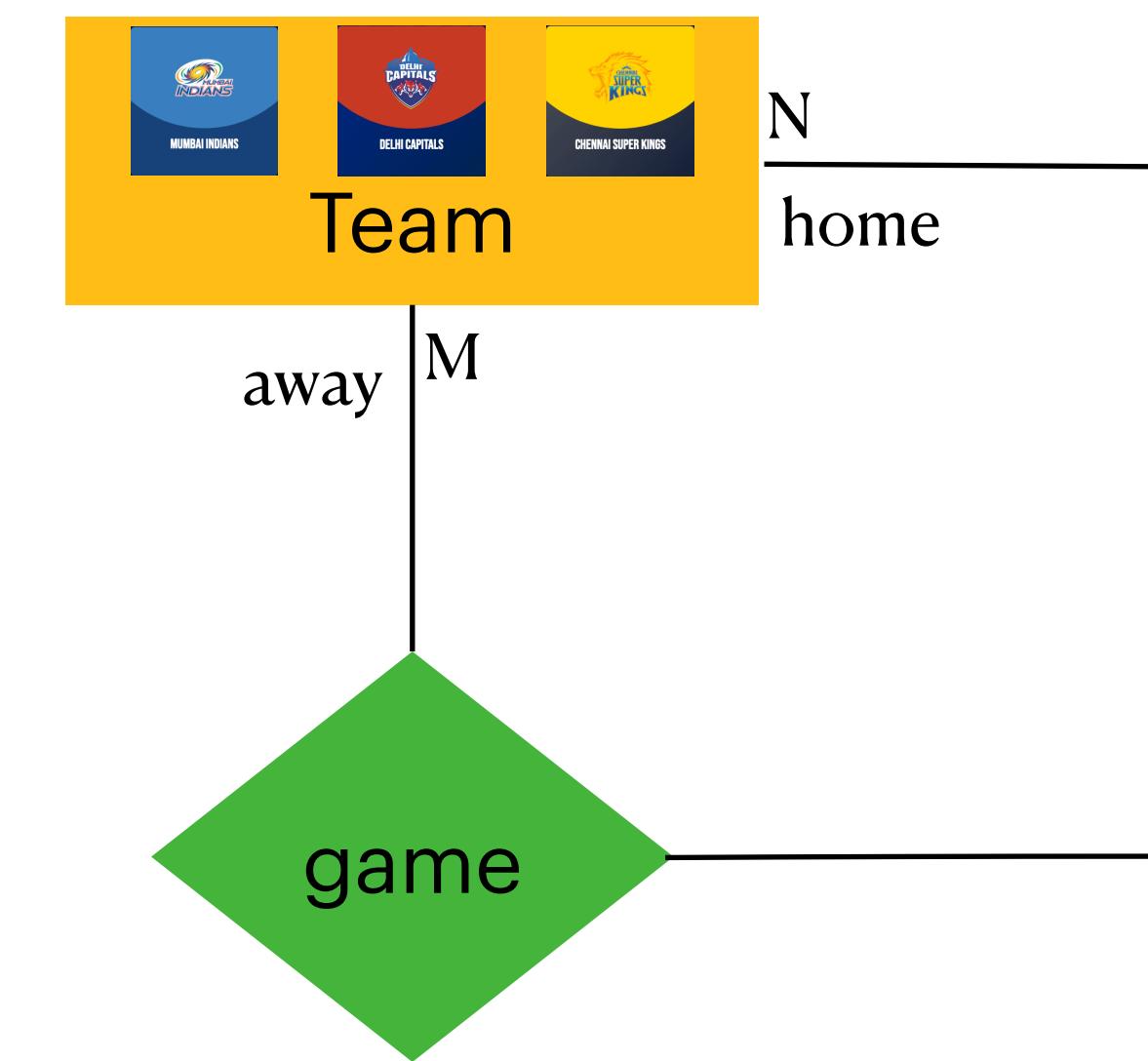
Relations

A Mathematical Definition

- If A and B are sets, then a relational R is a subset of $A \times B$
- Example
 - $A = \{a_1, a_2, a_3, a_4\}$, $B = \{b_1, b_2, b_3\}$
 - $R = \{(a_1, b_1), (a_2, b_1), (a_3, b_2), (a_4, b_3)\} \subseteq A \times B$

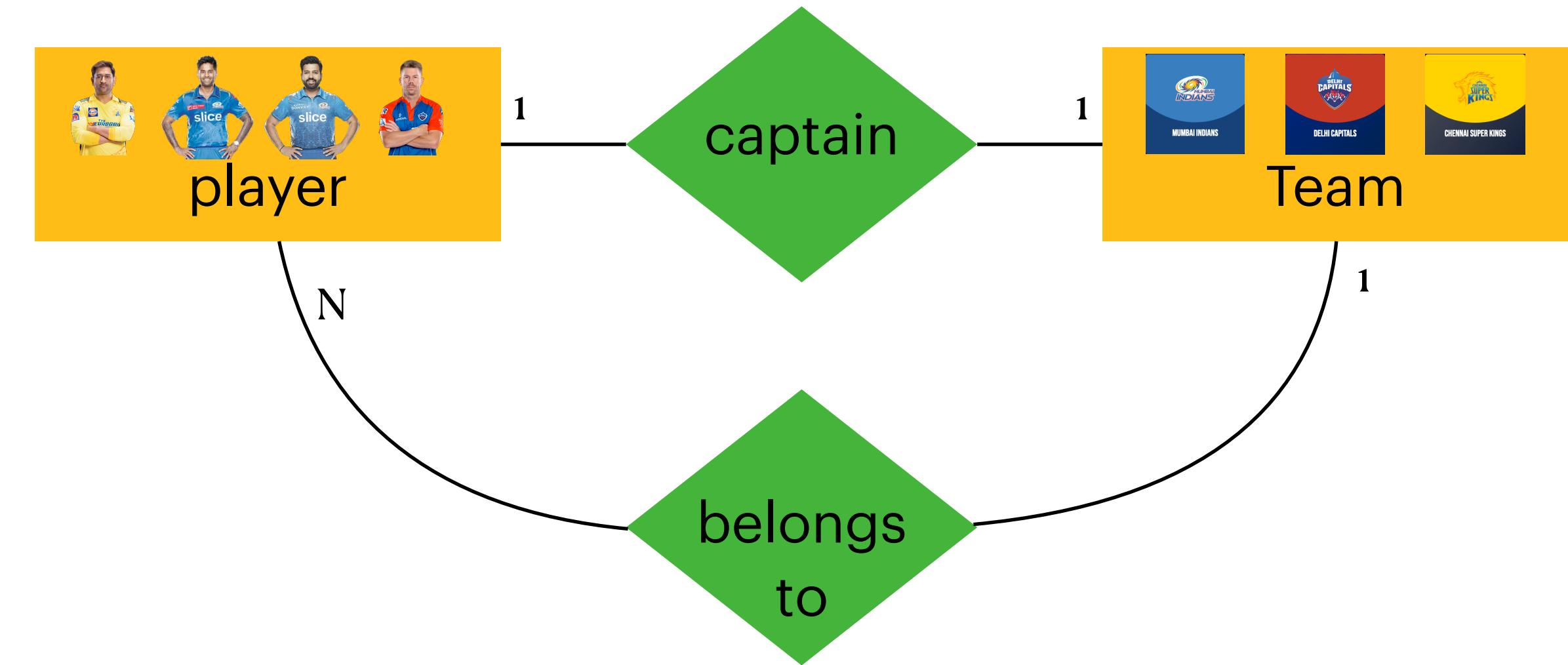
Recursive Relationship/Roles in Relationships

- An entity has a relationship with itself
- Roles: to differential the participation of the entity, roles are assigned to clarify its function in the relationship

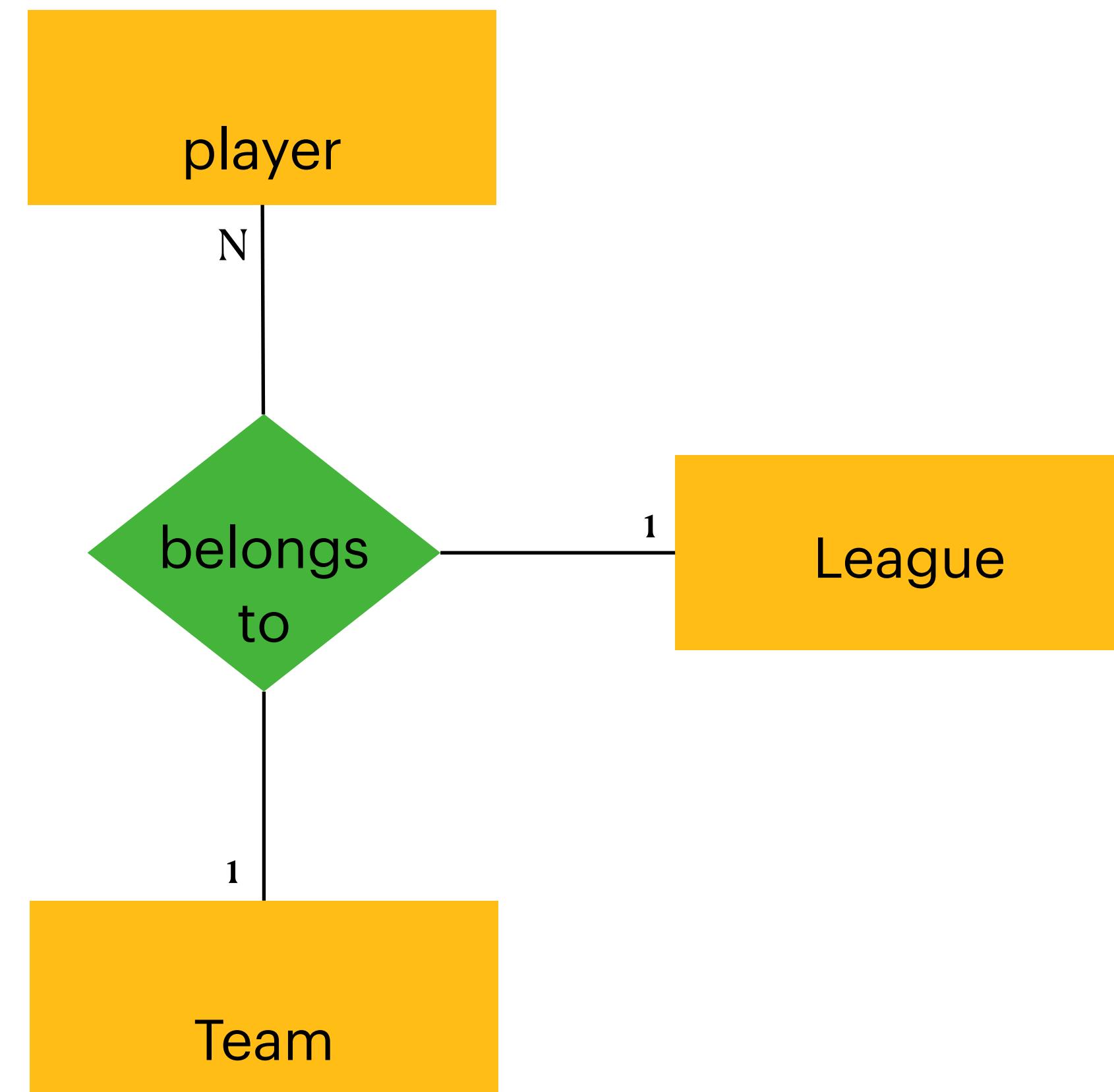


Parallel Relationships

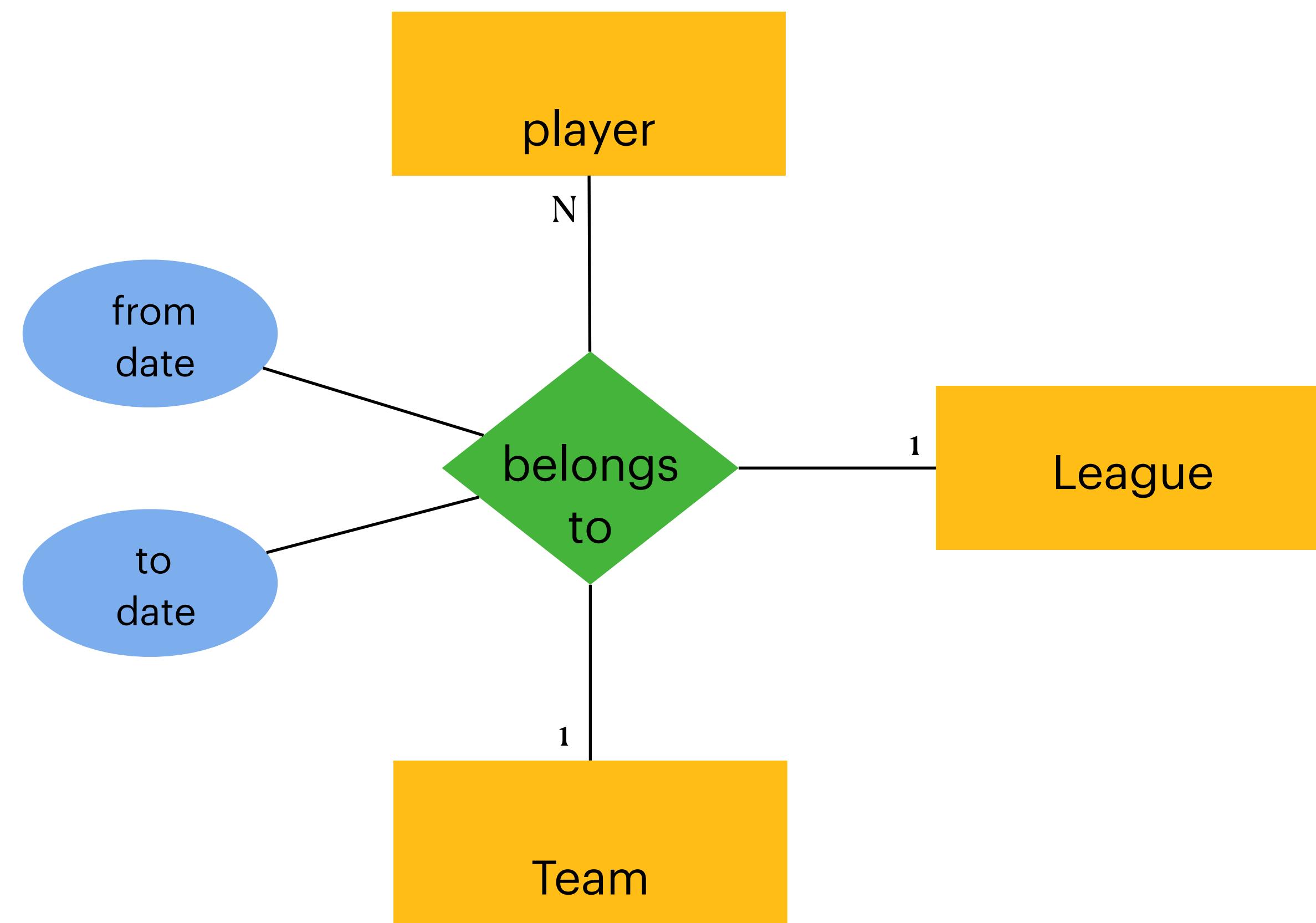
- Two or more relationships exists between the same pair of entities.
- Each represent different associations
- Each relationship has its own distinct meaning or role



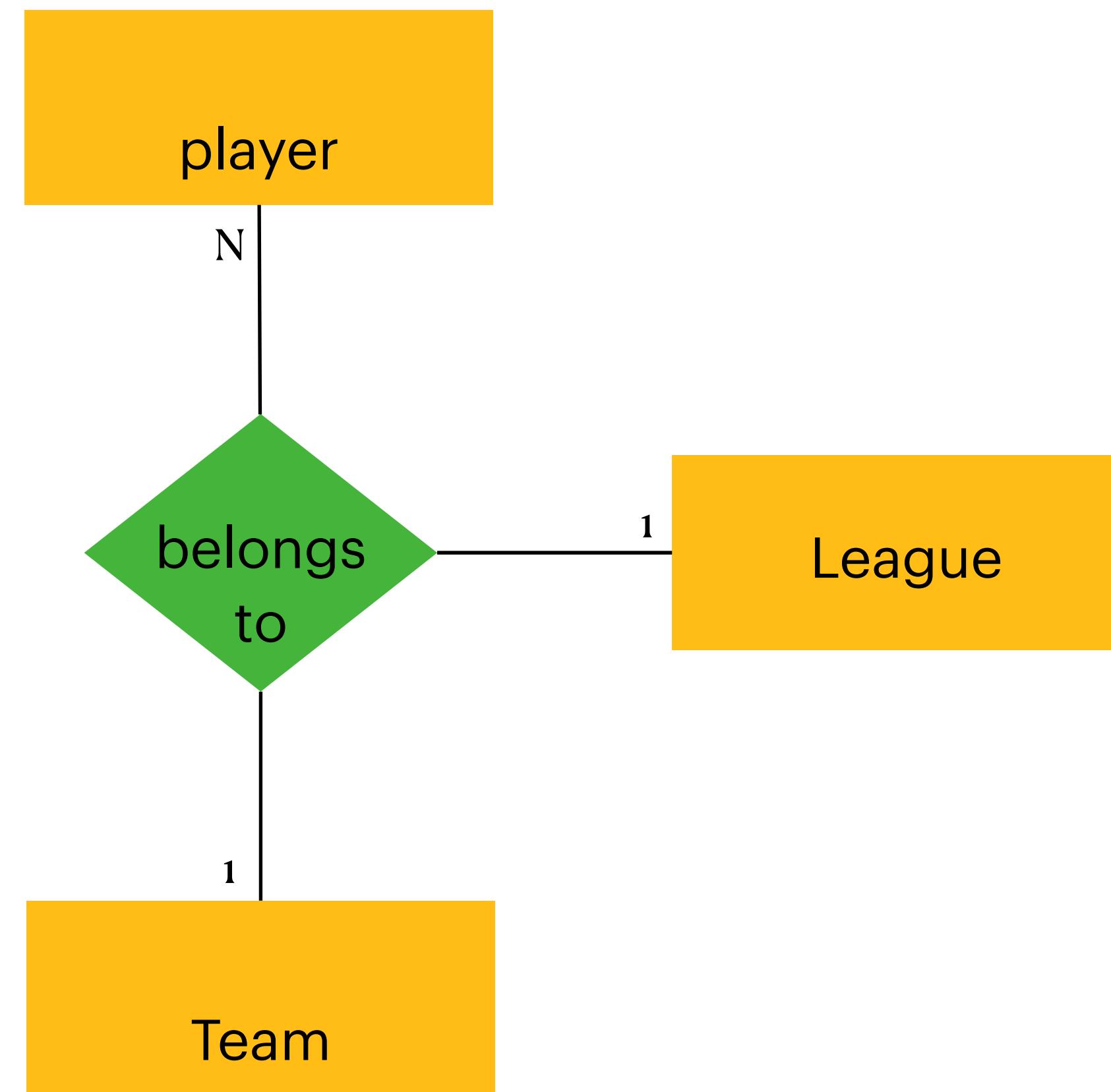
Non-binary (multiway) relationship



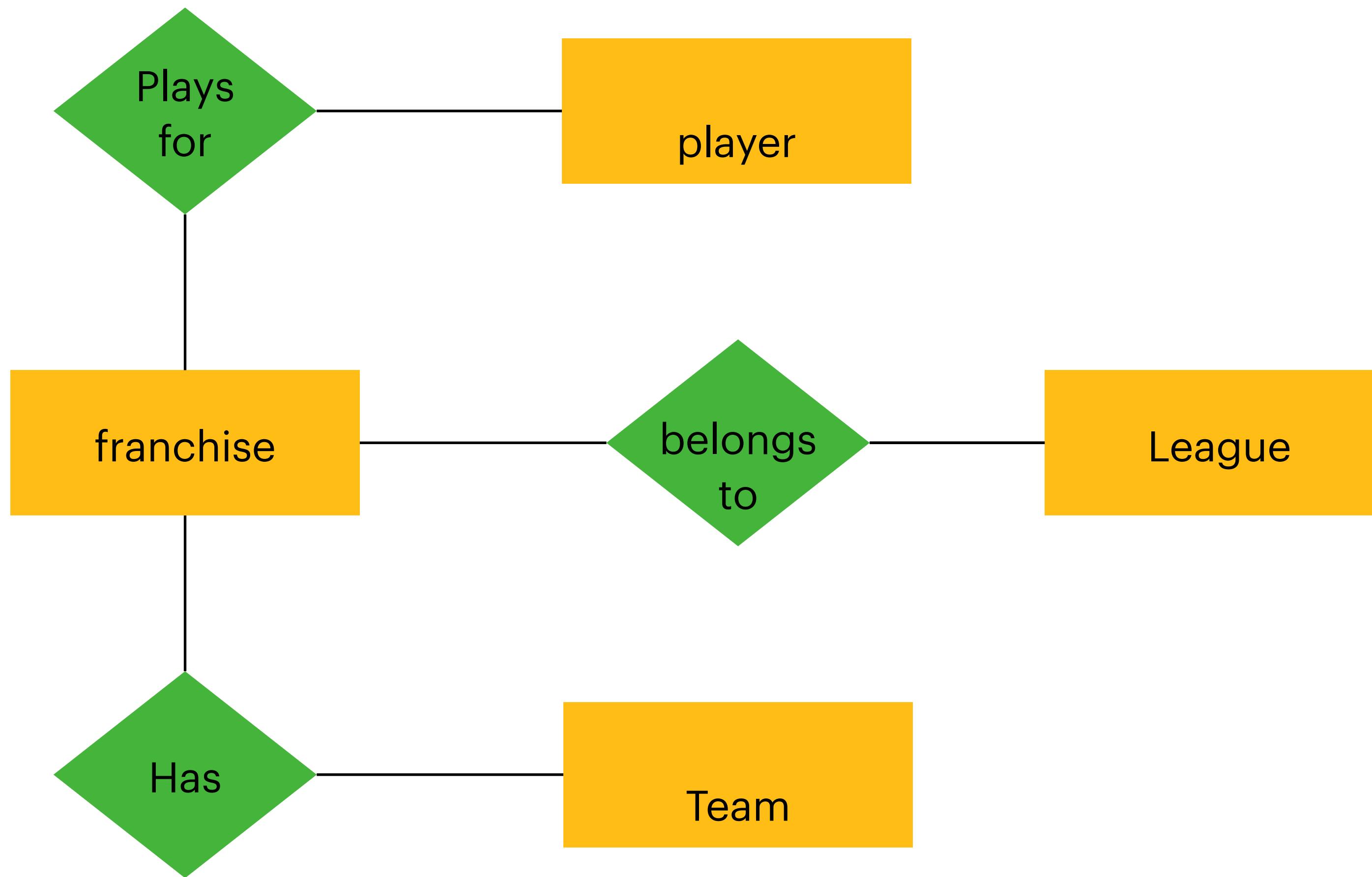
Relationship Attributes



Simplifying multiway relationship

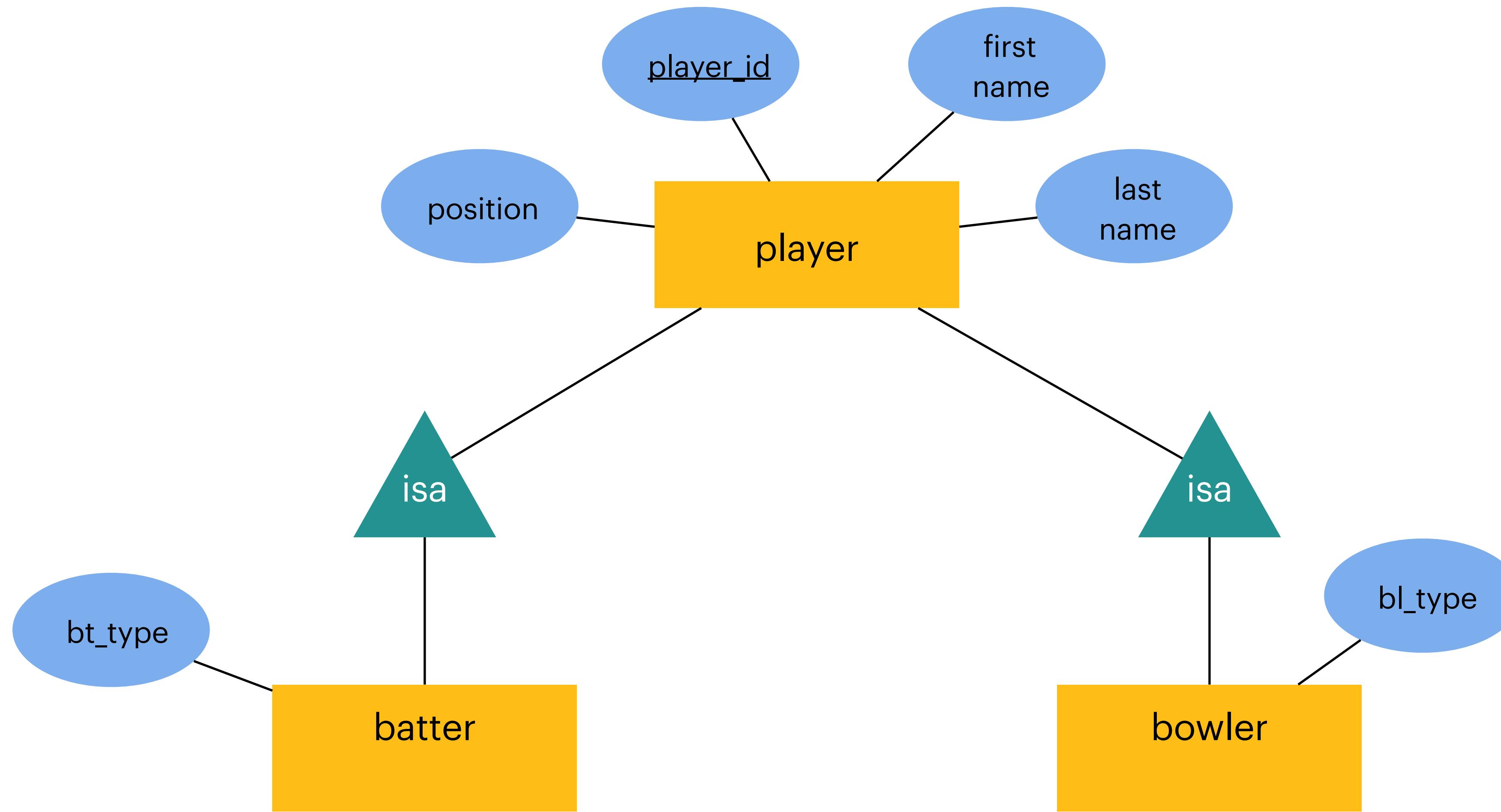


Simplifying multiway relationship



What are the multiplicities here?

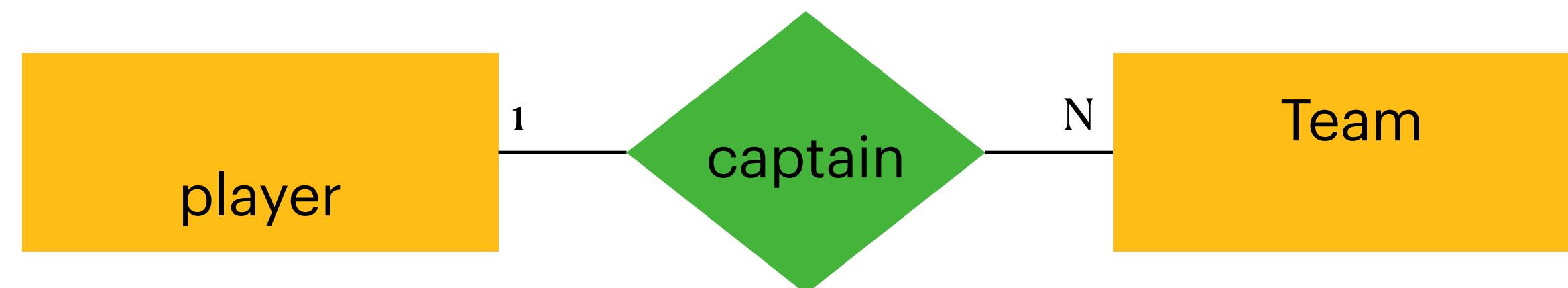
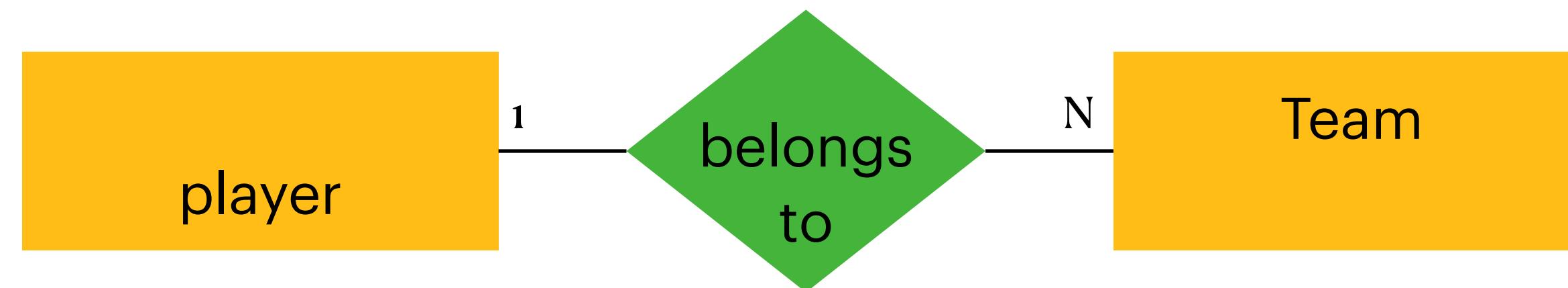
Hierarchies/subclass relationship/isa



Design Principles

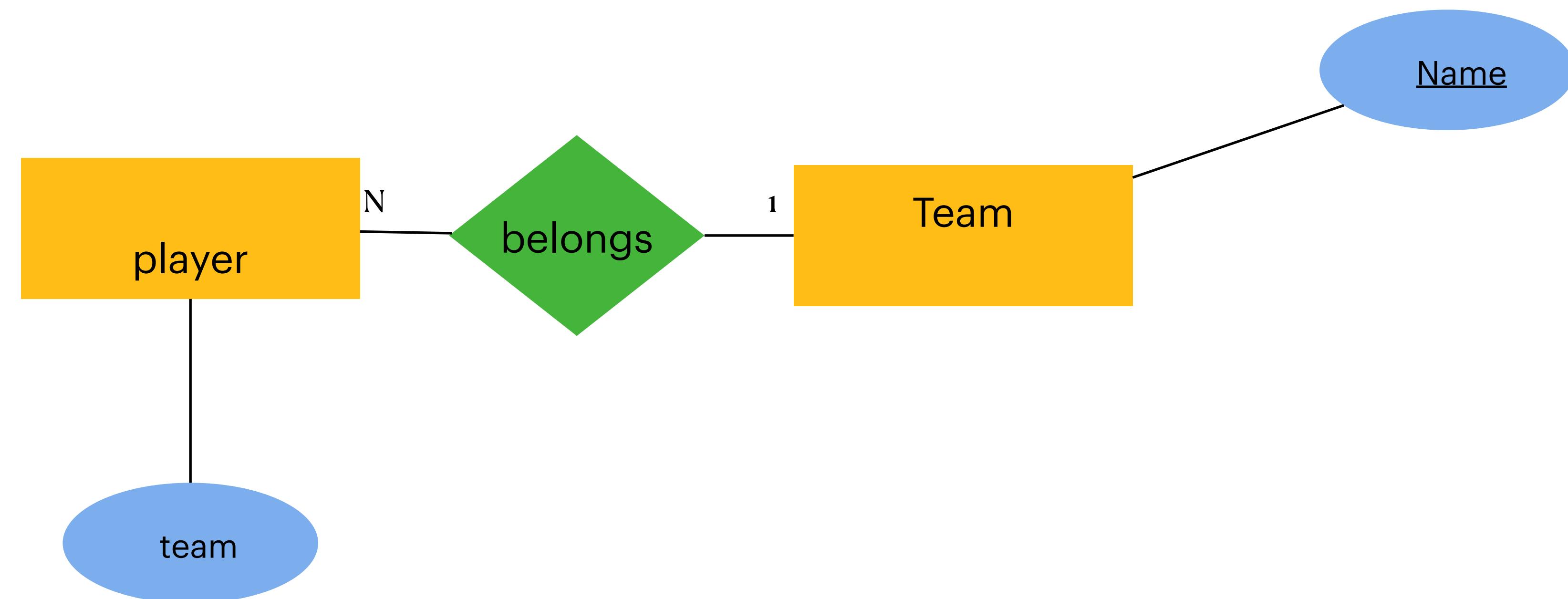
1. Be faithful to the app!

- What is wrong here?



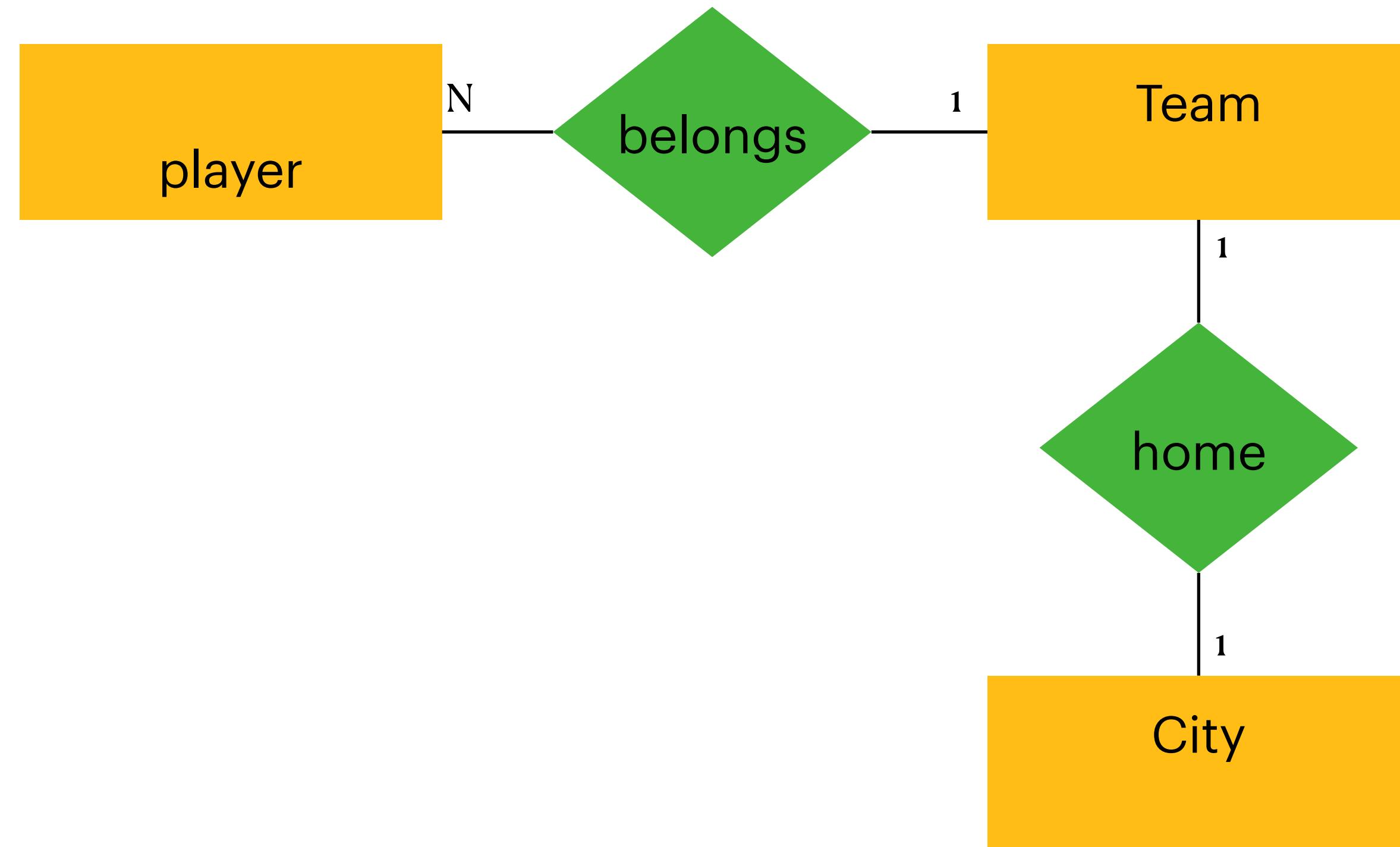
2. Avoid Redundancy

- What is wrong here?



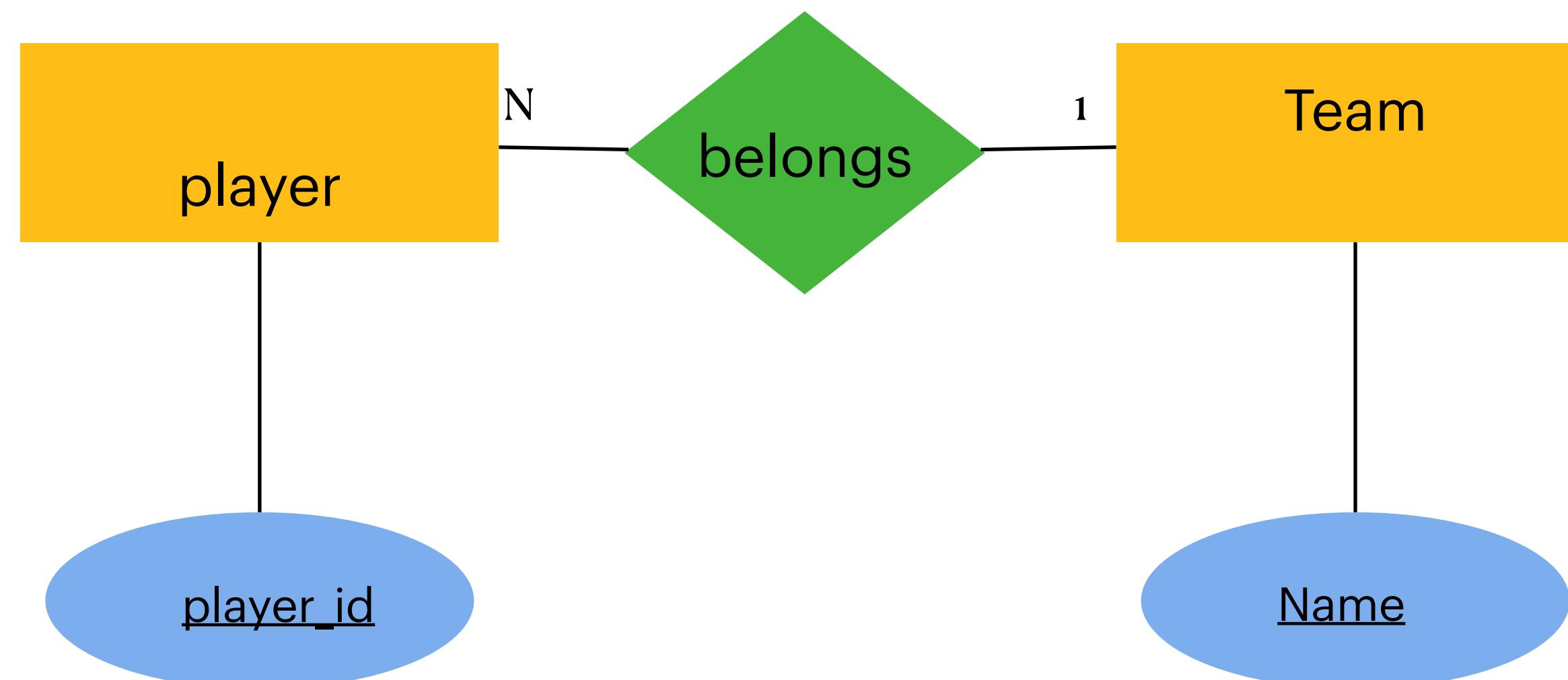
3. Keep it Simple

- What is wrong here?



4. Attributes over entity sets

- What is wrong here?



An entity should satisfy at least one of the two conditions

1. It is more than the name of something; it has at least one non-key attributes
2. It is the many is many-one, many-many relationship

Constraints

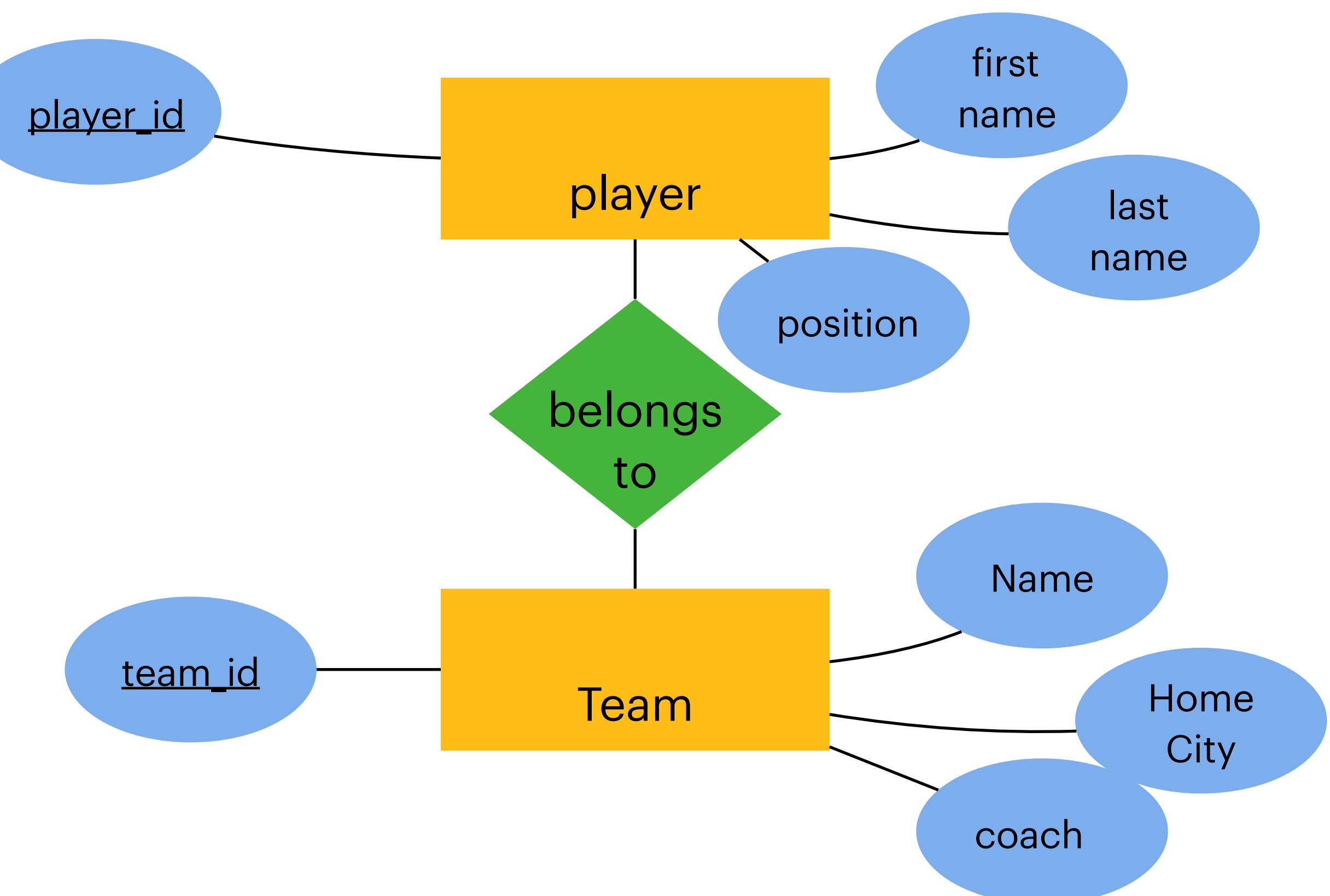
Constraints

- constraint is an assertion that must hold (always!)
 - ▶ Part of the modelling process
 - ▶ Central to your database design
- Gives more semantics to data
- Prevents incorrect data entries
- Enable efficient storage and lookup
- Allows to refer to entities

Types of Constraints

Key constraints

- Keys uniquely identifies an entity (e.g, entry number, player id)
- Every entity set must have a key
- More than one attribute could be a key



Types of Constraints

Single-Value Constraints

- Unique values in given context (keys are single-value constraints)
 - ▶ e.g., place of birth has to be unique
 - ▶ Null values
- Many-one relation \implies single-value constraint

Types of Constraints

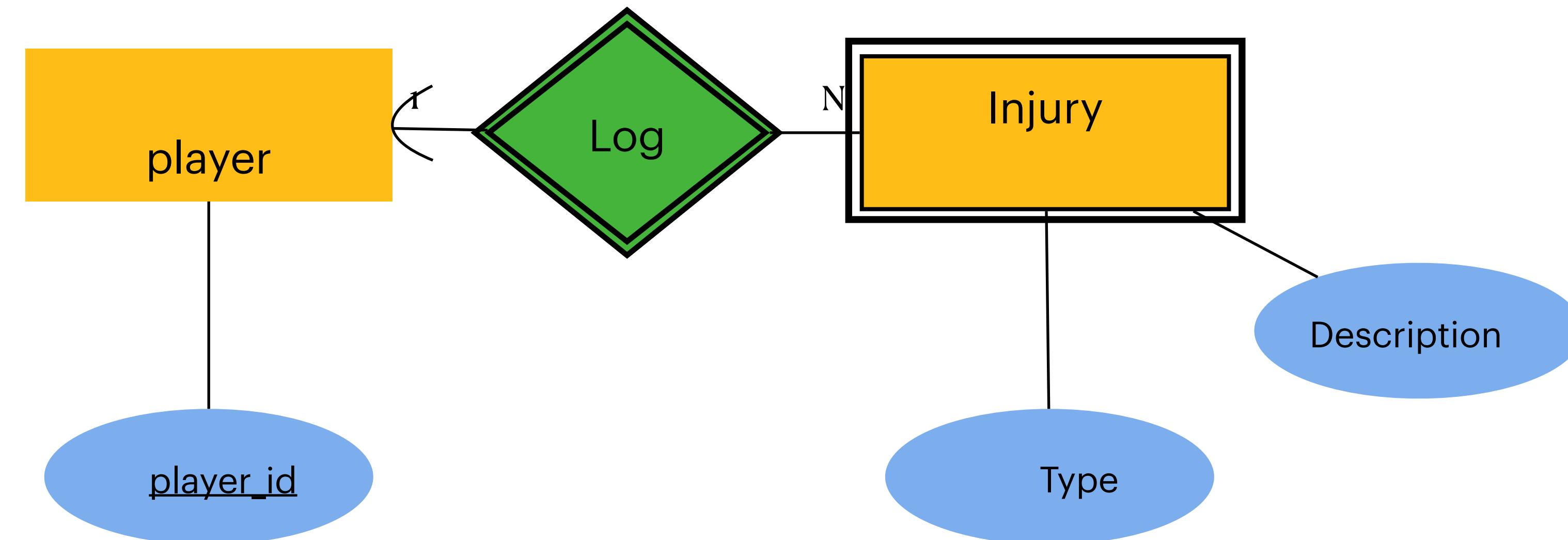
Referential Integrity Constraint

- Remove the null, insist on the value
- A relationship has one value and the value must exist
 - If a player belongs to a team, then the team must exists in the database



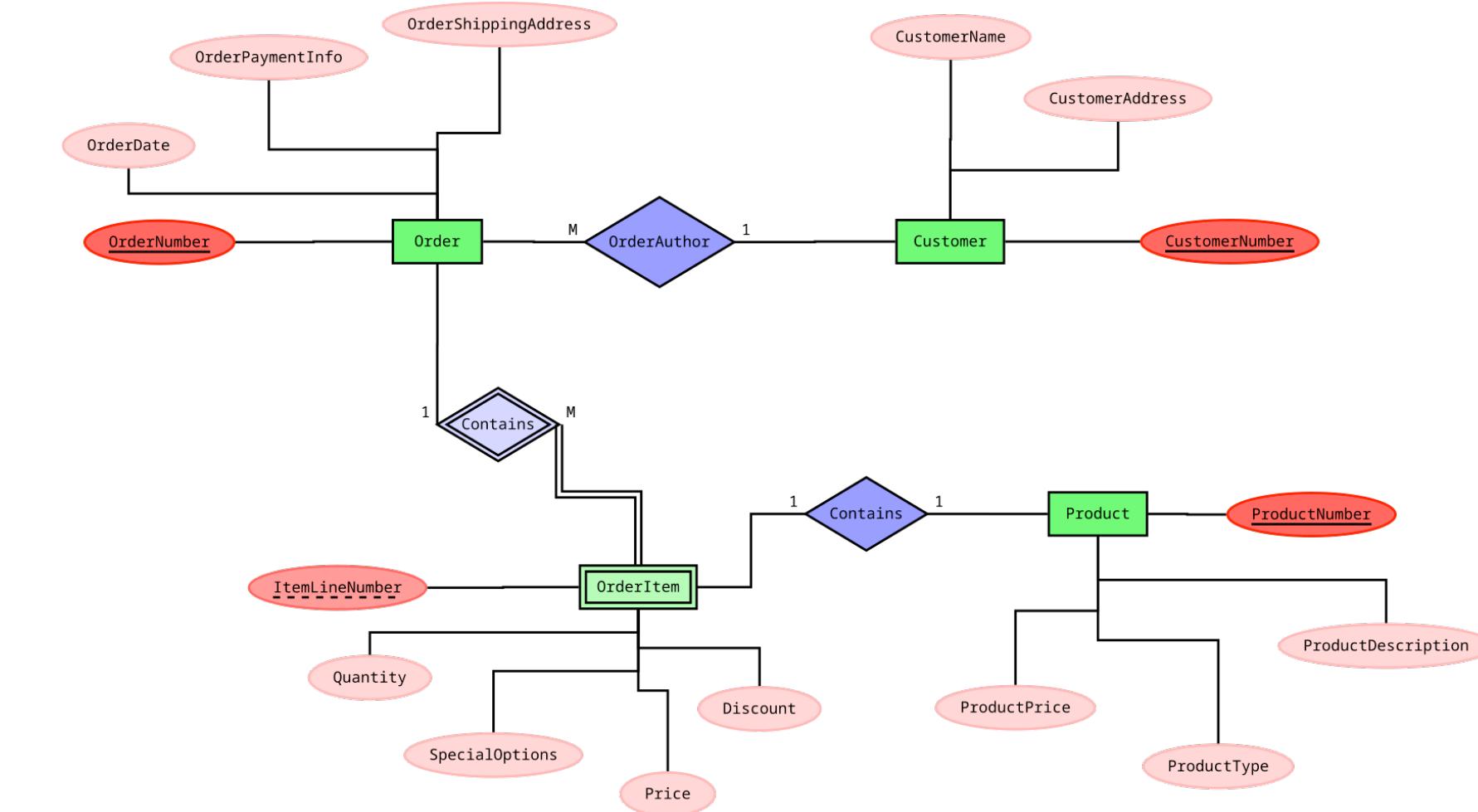
Weak Entity Sets

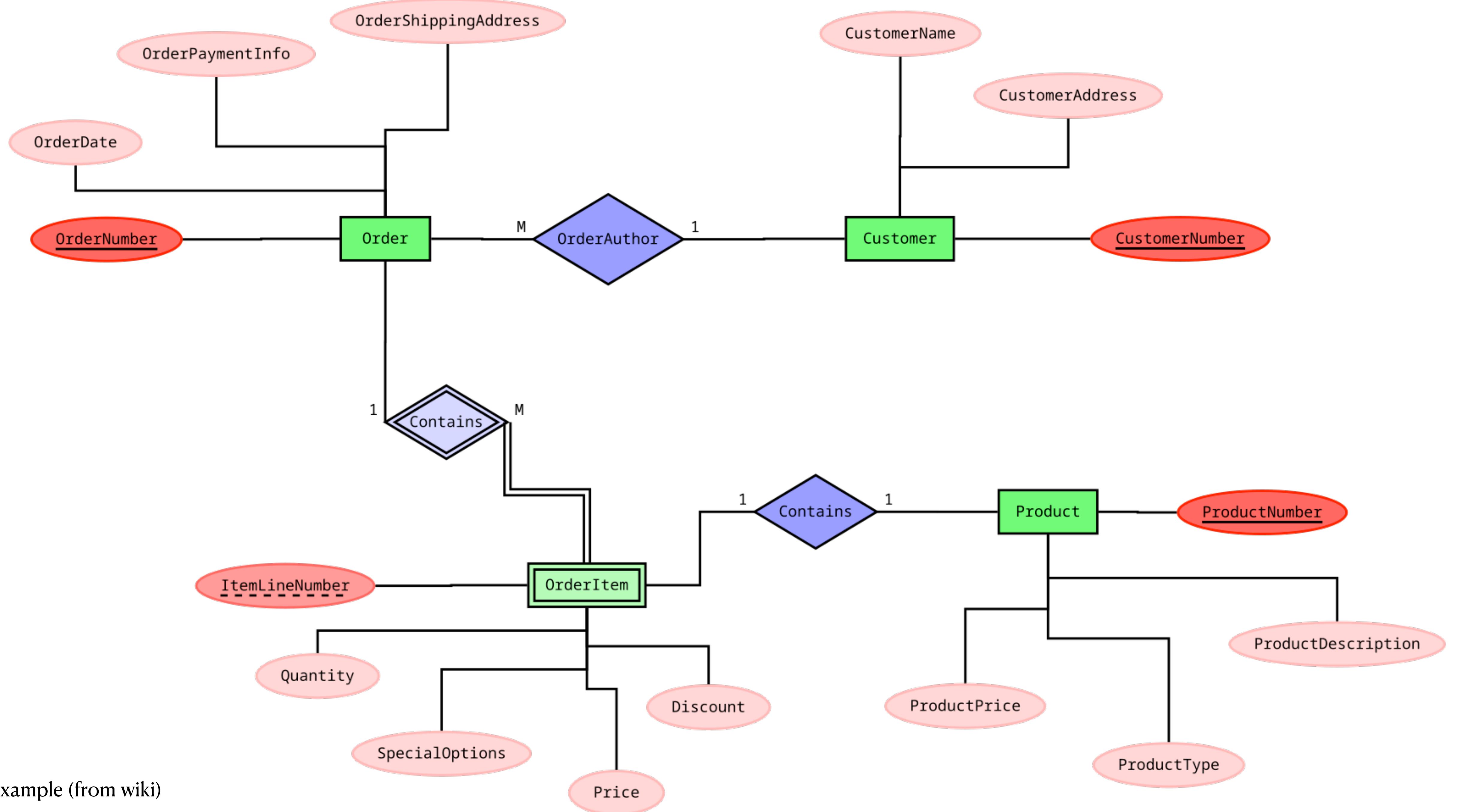
- An entity set whose attributes are not sufficient to form a key



Summary

- ER model is fundamental in database design
- Components of the ER model
 - Entities, relationships, and attributes
- Cardinality and participation constraints
 - 1:1, 1:M, M:N
- Design principles





Example (from wiki)

Next lecture — Relational Model
and ER to Relational Model