

SFWRENG 3DB3 Tutorial

Anjola Adewale

Week 2 Tutorial

Agenda

- Review of Relational Model and Keys
- E-R Diagram
 - Notations
 - Relationship Types
 - Examples

Relational Model

- A relation is a table
 - Schema: relation name and attribute list
 - Ex: Students (id, name)
- Relation model is simple, but effective

Relation Name

Students

Attribute Names

Tuples (Records)

ID	Name
2225555	Peter Jones
1234567	Amber Smith

Keys

- A **key** is a set of attributes that uniquely identifies tuples in a relation
- A set of attributes **K** is a superkey for a relation **r** if **r** cannot contain two distinct tuples t_1 and t_2 such that $t_1[K] = t_2[K]$
- **K** is a candidate key for **r** if **k** is a minimal superkey

Example of keys

<i>sid</i>	<i>name</i>	<i>login</i>	<i>age</i>	<i>gpa</i>
50000	Dave	dave@cs	19	3.2
53666	Jones	jones@cs	18	3.3
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.7
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

- Superkeys:

- {name, age}
- {login}
- {name, login}
- {sid}
- {sid, login, name, age, gpa}
- Etc...

- Keys:

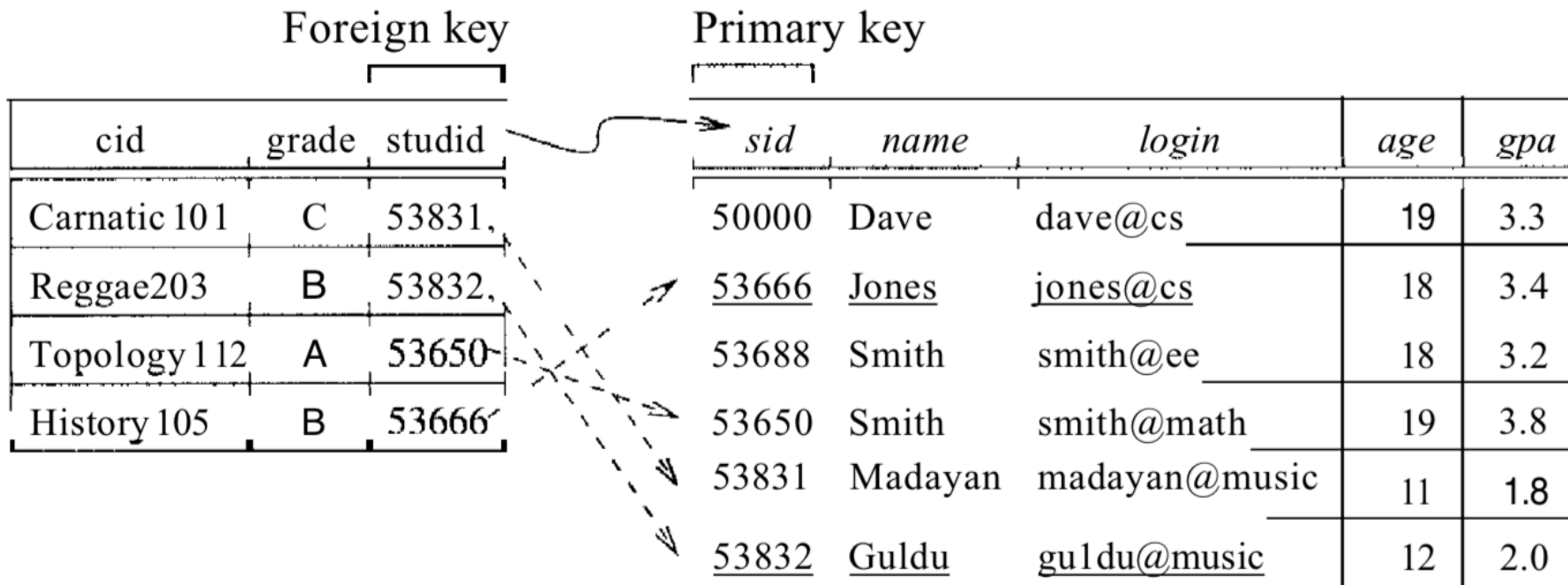
- {login}
- {sid} : Primary key
- {name, age}
- {age, gpa}

Caution!

- The **red keys** could be candidate keys of above **given** data.
- However, they might **prevent the addition** of **new students**.
 - New student who has same name and age **cannot be added**.
 - Two students who have same age **cannot have same gpa**.

Foreign Key

- A foreign key is an attribute (or a set of attributes) in one table that uniquely identifies a row of another table. It establishes a relationship between two tables.
- A foreign key must correspond to the primary key of the referenced table.



Referential Integrity

- Referential integrity is used to guarantee that attributes in one relation refer to existing tuples in another relation referenced in a relationship

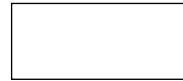
Employee ID	Name	DOB	Department No
123	John Smith	02/25/1978	5
456	Alice Doe	04/06/1984	3
789	John Smith	09/17/1990	3

Department No	Name	Manager	#employees
5	Finance	Jane Mae	10
2	Sales	Bob Brown	15
4	Development	Sally Roe	30

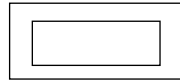
Is there a foreign key or referential integrity violation between these relations?

ER Diagram Notations

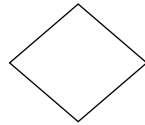
- Entity



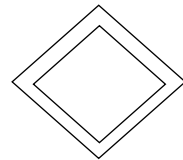
- Weak Entity



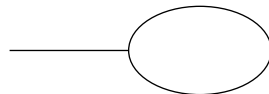
- Relationship



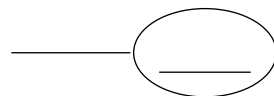
- **Supporting** Relationship



- Attribute

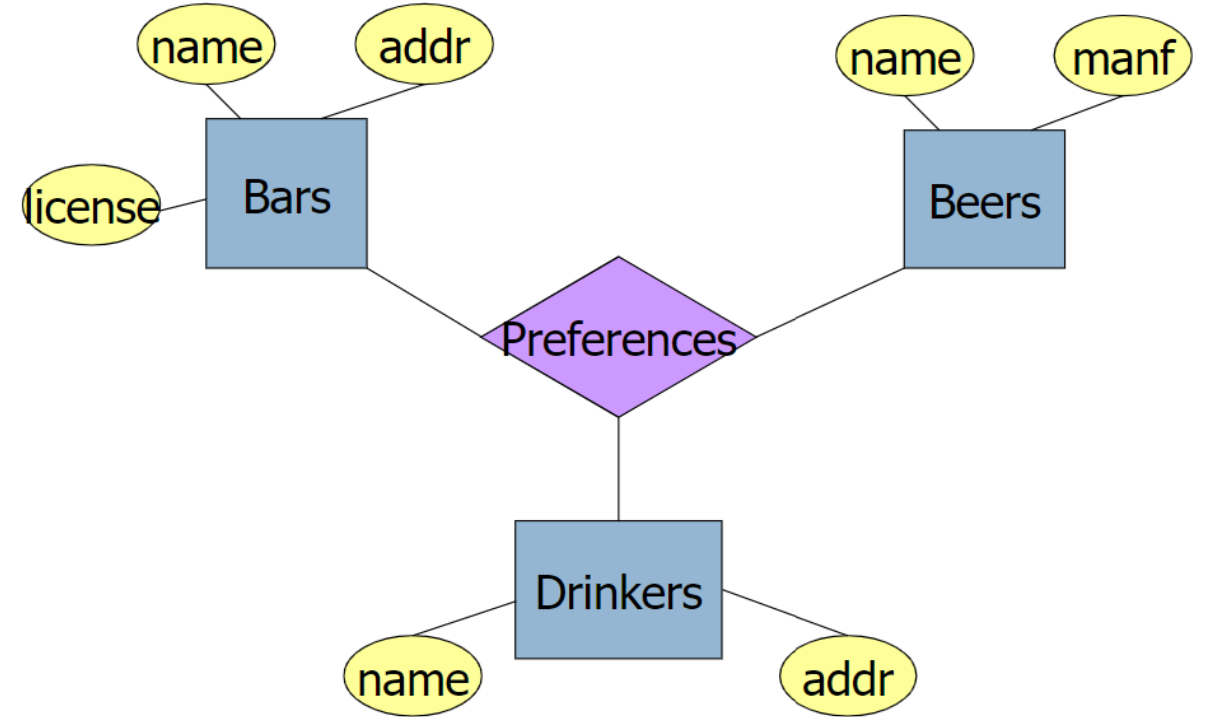


- **Primary Key** Attribute



ER Terms

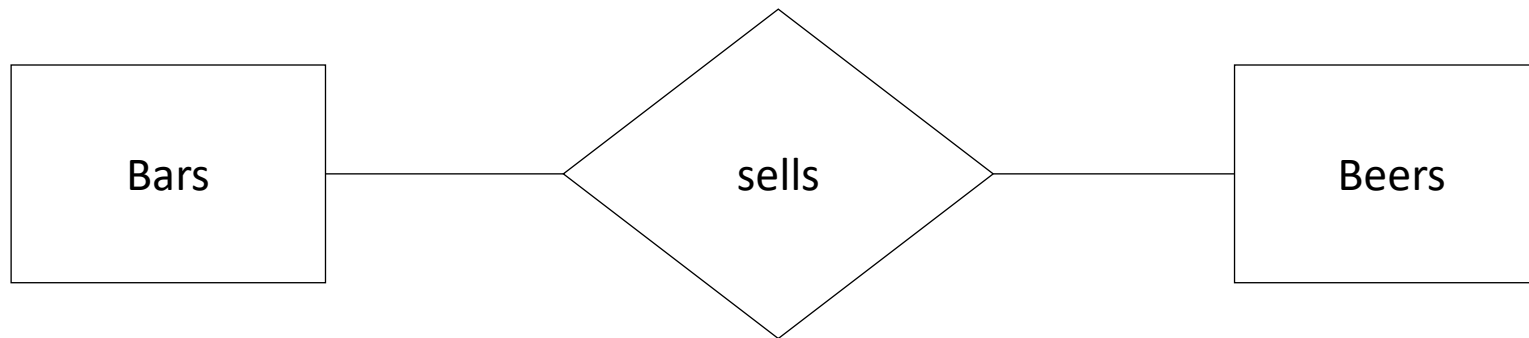
- Entity: Is a “thing” or object
- Attribute: Is a property of an entity set
- Relationship: Association between entity sets



List entities, attributes, and relationships

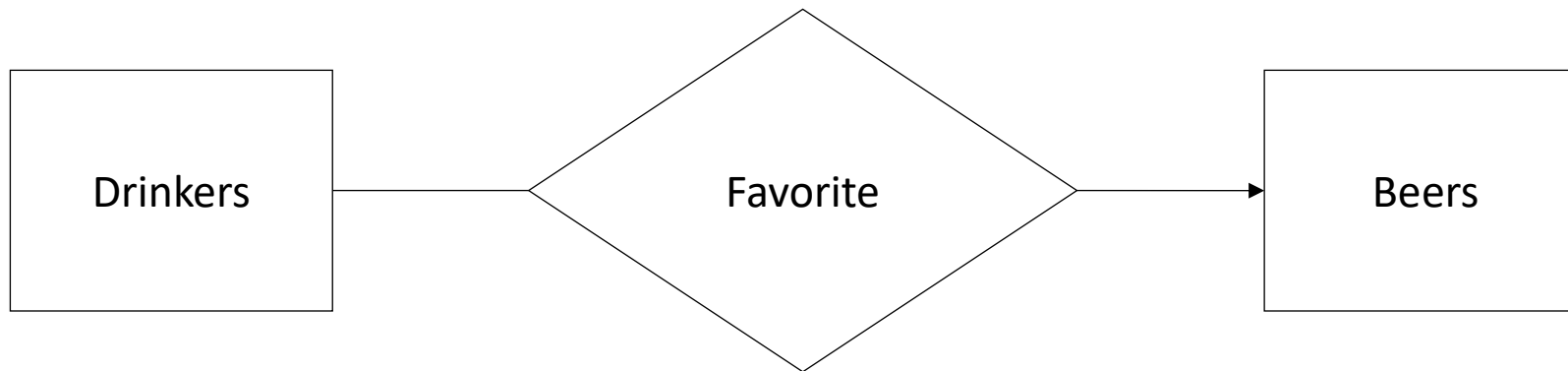
Many-to-Many Relationship

- An entity of either set can be connected to many (zero, one or more) entities of the other set
 - E.g., a bar sells many beers, and a beer is sold by many bars



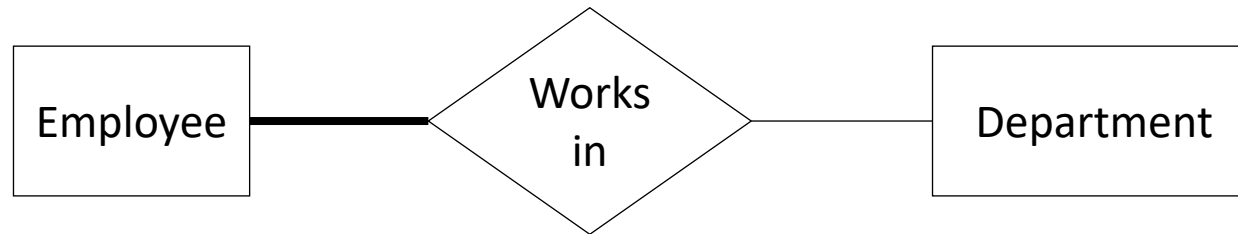
Many-to-One Relationship

- Each entity of the first set is connected to **at most one entity (zero or one)** of the second set. But an entity of the second set can be connected to zero, one, or many entities of the first set
 - E.g., a drinker has at most one favorite beer, and a beer can be the favorite of many drinkers



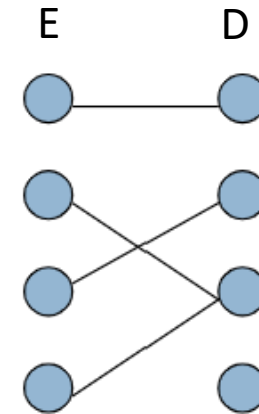
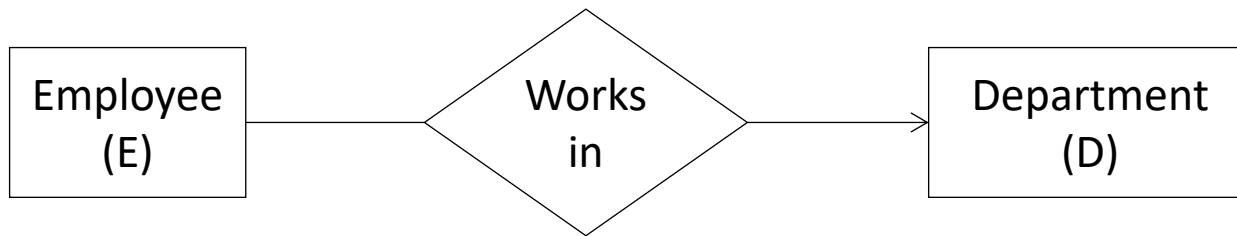
Participation

- Total Participation: At least one entity in the set participates in the relationship
 - E.g., Department must have **at least one** Employee



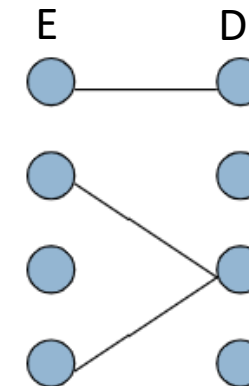
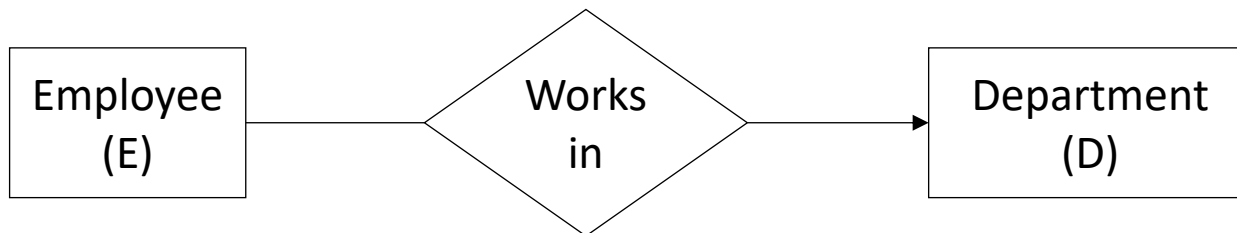
Many-to-One Relationship (Participation)

- Participation of the entity is total: open arrow
 - E.g., **Every** employee works in **one** department (**exactly one**)



Participation of Employee entity is total

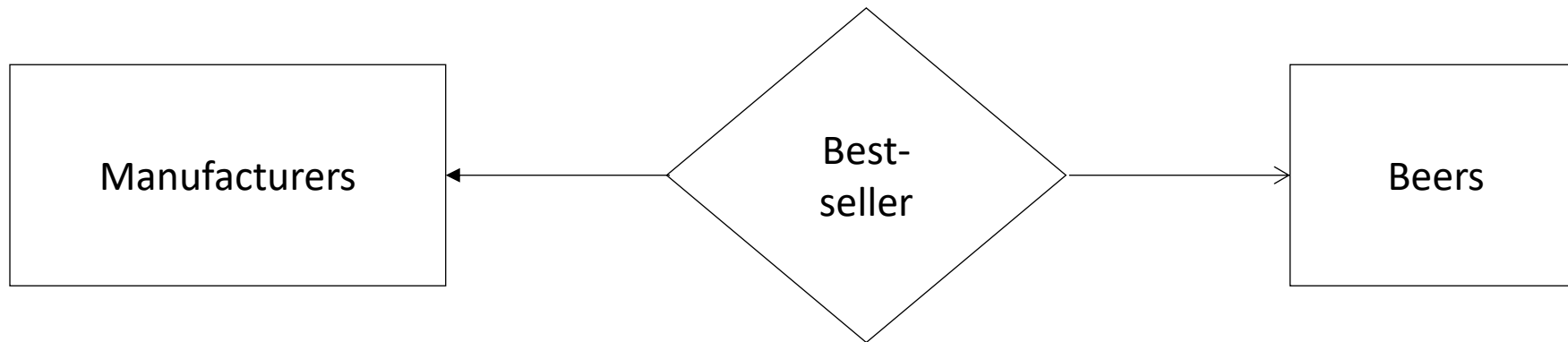
- Participation of the entity is partial
 - E.g., Employee has **at most one** department (**zero or one**)



Participation of Employee entity is partial

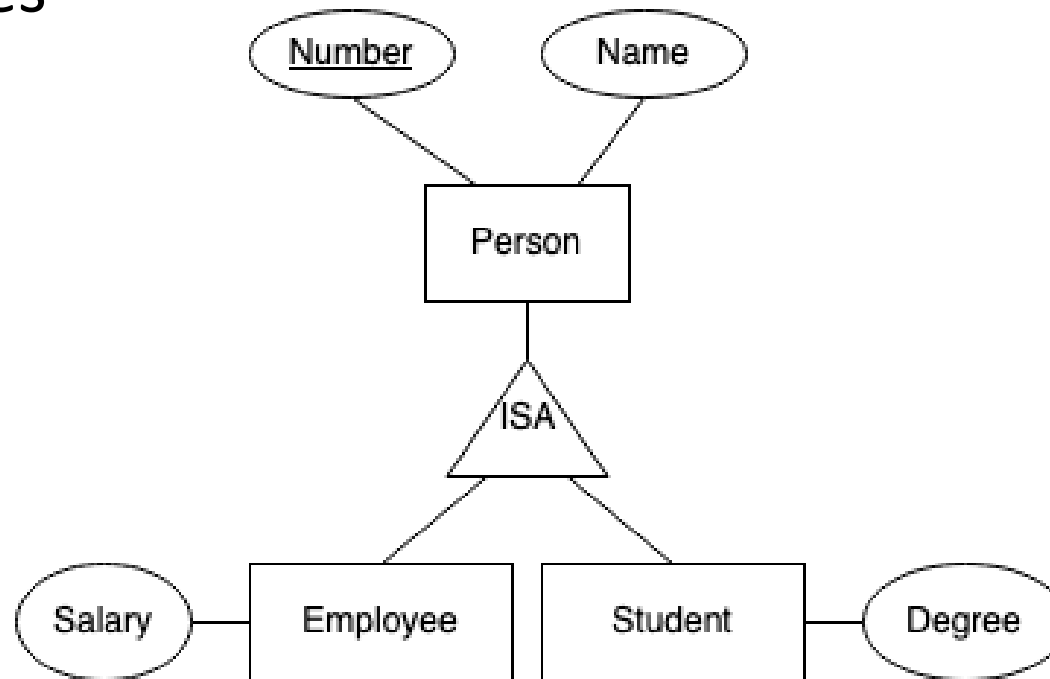
One-to-One Relationship

- Each entity of either entity set is related to at most one entity of the other set
 - E.g., a manufacturer has exactly one best-seller beer, and a beer is the best-seller of at most one manufacturer
 - **Caution:** carefully draw the **type of arrow** (w.r.t. **relationship**)



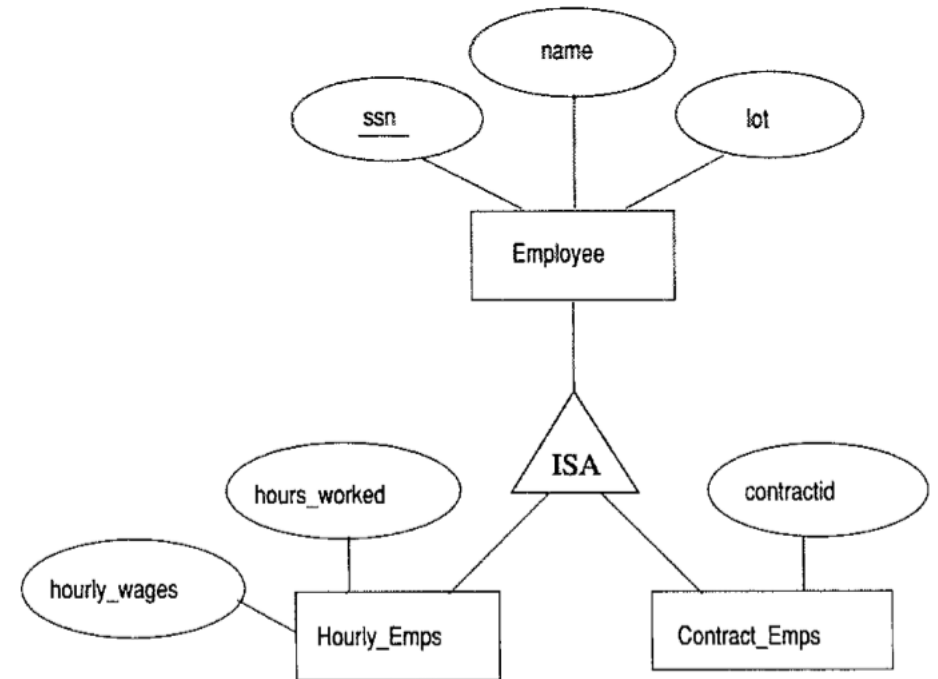
Subclasses/ISA Relationship

- Subclass are a special type of entity set that **inherit properties** from a parent entity
- The subclass must have all attributes of the parent as well as having additional properties



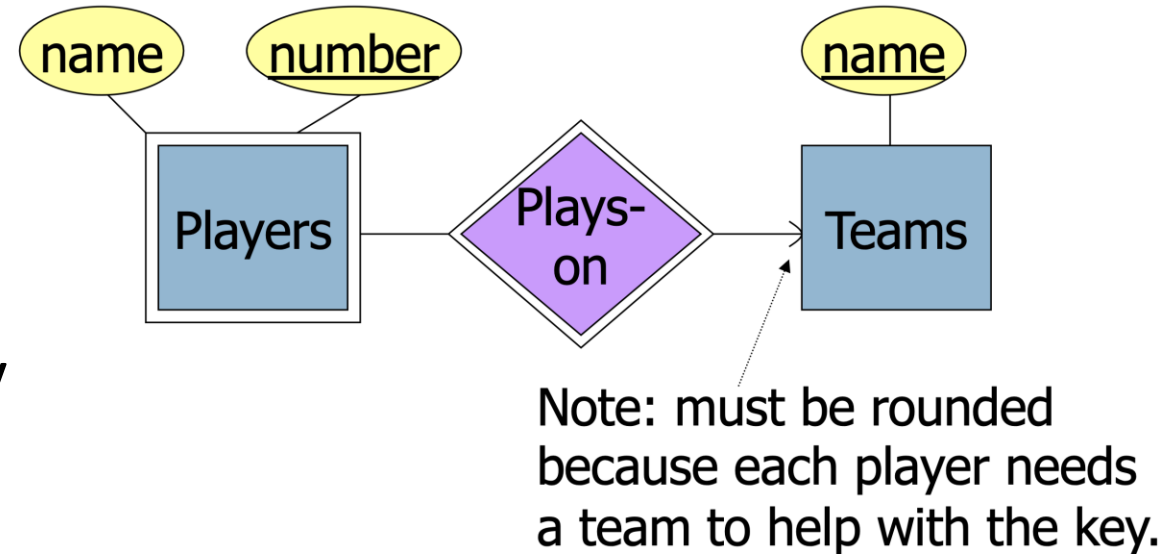
Subclasses/ISA Relationship

- Overlap constraints
 - Can two sub-classes contain the same entity?
 - E.g., Can Joe be an Hourly_Emps as well as a Contract_Emps entity?
 - By default: No
- Covering constraints
 - Does every Employee entity have to be an Hourly_Emps or a Contract_Emps entity?
 - By default: No



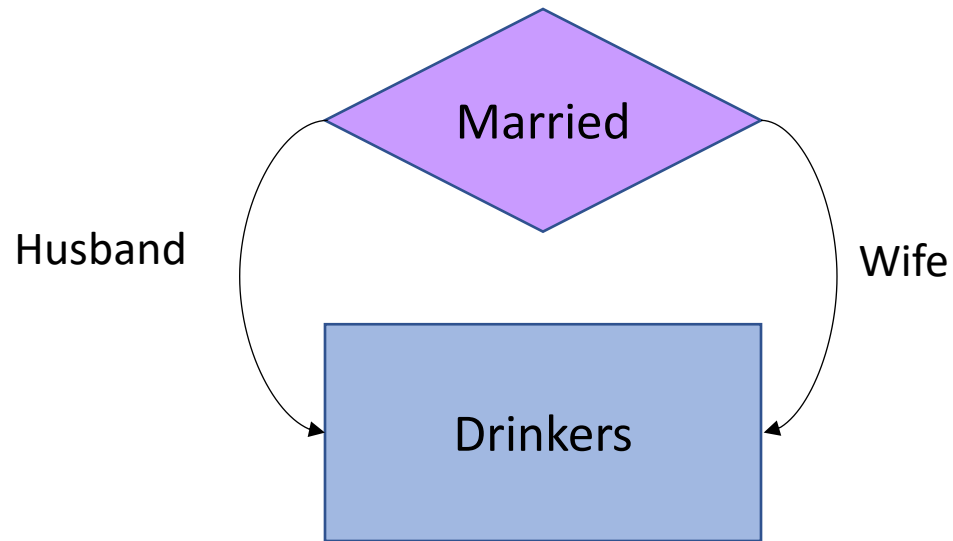
Weak Entity

- A weak entity does not have enough information to have its own primary key and relies on a supporting entity for unique identification
- To identify the (**weak**) entity, we need one (or more) many-to-one (**supporting**) relationship(s) to other (supporting) entity set(s)
 - E.g., 'number' in 'Players' together with the 'name' in 'Teams' related to the 'Plays-on' should be unique
 - Supporting relationship is many-to-one



Role

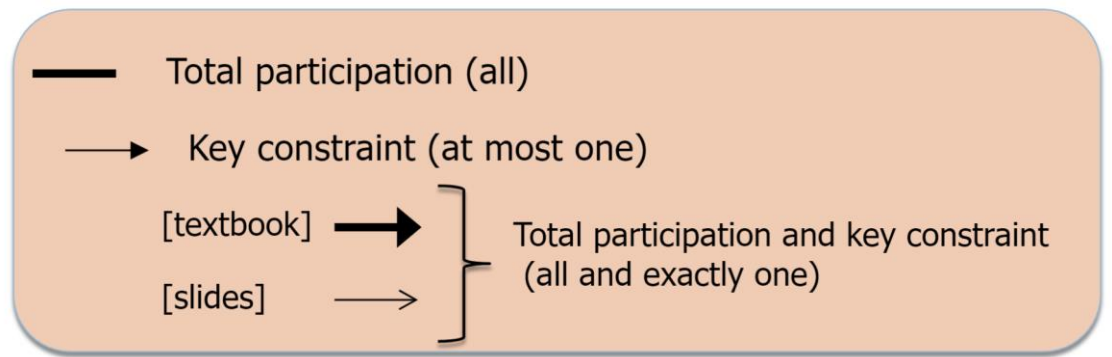
- An entity set may appear more than once in a relationship
 - Label the edges between the relationship



Relationship Set

Husband	Wife
Bob	Ann
Joe	Sue
...	...

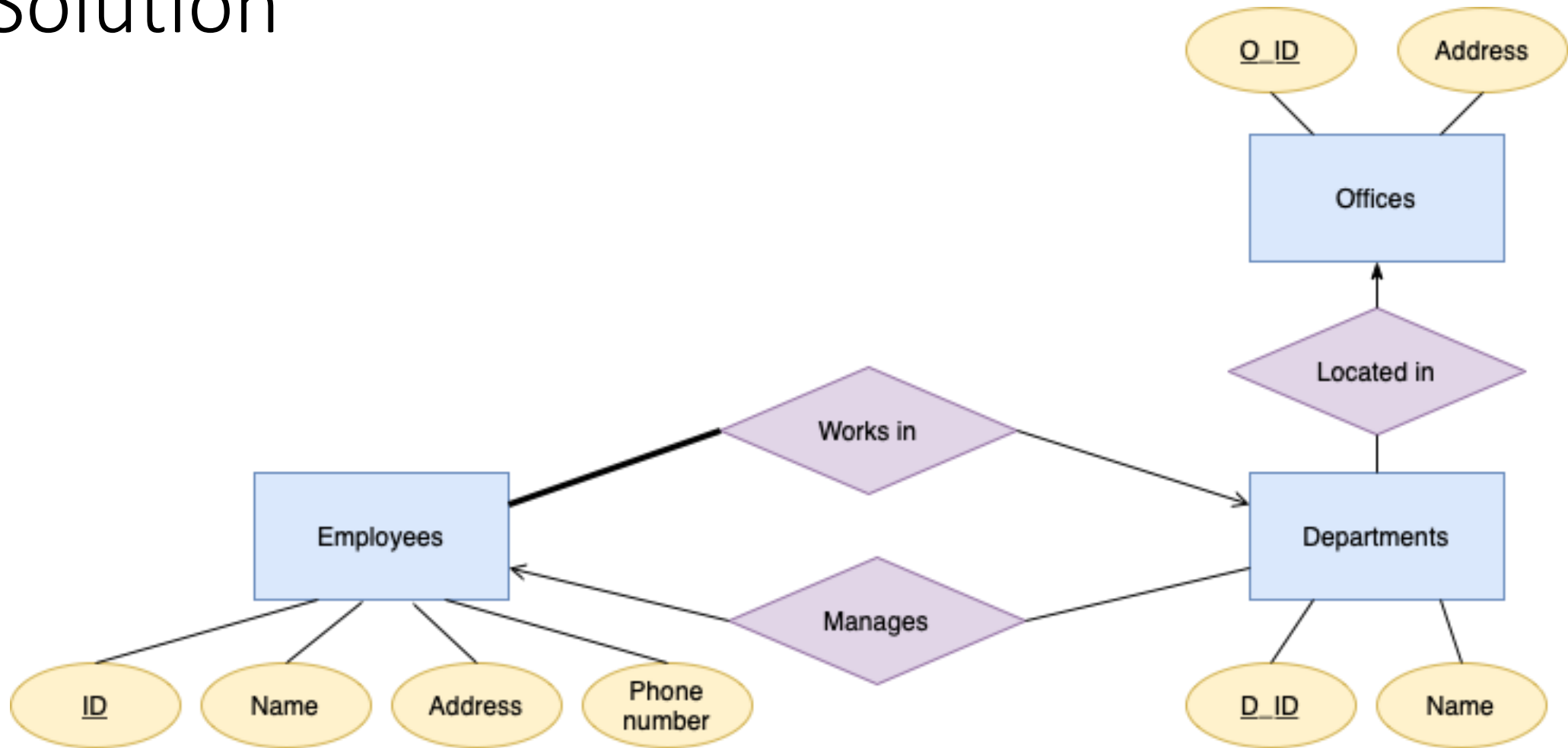
Example 1



Suppose you are given the following requirements for a simple database for the industry:

- the Industry has many departments,
- each department has an ID, name, a manager, an office and a set of employees,
 - each department must have at least one employee
 - each department is located in at most one office
- each employee belongs to exactly one department,
- each department has exactly one manager and a single employee is allowed to manage many departments,
- each employee has an ID, a name, an address and a phone number,
- each office has an ID and an address.
- **Note: ID is the primary key for department, employee and office**

Solution



Example 2

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

- The NHL has many teams and each team has:
 - A name, city, coach, and a set of players.
 - Each team must have at least one player.
- Each player belongs to exactly one team.
- Each player has:
 - A name, position, skill level, and injury records.
- Injury records include a description and are logged with a date of injury for each player.
- Games are played between two teams (referred to as `host_team` and `guest_team`) and have a date and a score.
- **Note: Name is the primary key for the team and player entities.**

Solution

Will be posted at the end of Tutorial