Lecture 2

Entity- Relationship Model

COMP3278B

Introduction to Database Management Systems

Dr. Ping Luo
Email : pluo@cs.hku.hk

Department of Computer Science, The University of Hong Kong

Acknowledgement: **Dr. Chui Chun Kit**

Outcome based learning (OBL)

- Outcome 1. **Information Modeling**
  - Able to understand the modeling of real life information in a database system.
- Outcome 2. **Query Languages**
  - Able to understand and use the languages designed for data access.
- Outcome 3. **System Design**
  - Able to understand the design of an efficient and reliable database system.
- Outcome 4. **Application Development**
  - Able to implement a practical application on a real database.

We are going to learn…

- Introduce the Entity-Relationship (E-R) Diagram
- More examples using the E-R Diagram

Section 2.1

E-R Diagram

E-R Diagram

Four concepts

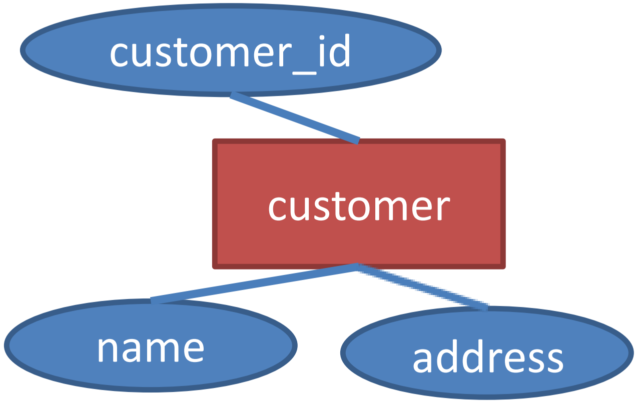
- 1. Entity and Entity set
- 2. Relationship and Relationship set
- 3. Constraints
- 4. Keys

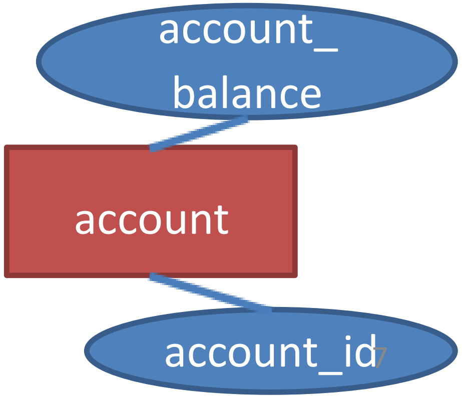
1. Entity and Entity set

- **Entity**
  - An object that exists and is distinguishable from other objects.
  - E.g. A customer, an account, a department, etc.
- **Entities have Attributes**
  - People have names and address.
- **Entity set**
  - A set of entities of the same type that share the same properties (or attributes).
  - E.g., a set of all customers, all saving accounts, all departments in the company, etc.

1. Entity and Entity set

- **In the E-R Diagram**
  - **Rectangles** – entity sets.
  - **Ellipses** – attributes.
  - **Line between a rectangle and an ellipse** – link between an attribute and an entity set.





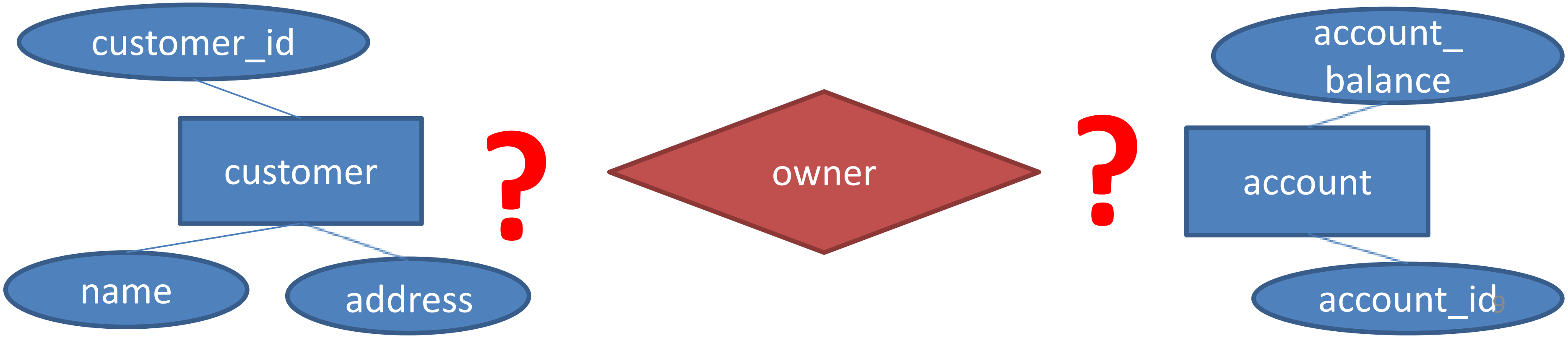
2. Relationship and Relationship set

- **A relationship is an association among entities.**
  - E.g., the relationship between the customers and the accounts.
- **A relationship set is a set of relationships of the same type.**

2. Relationship and Relationship set

- **In the E-R Diagram**
  - **Diamond** – a set of relationships

What are the constraints in specifying the relationship between two entity sets?



customer_id

customer

name

address

?

owner

?

account_balance

account

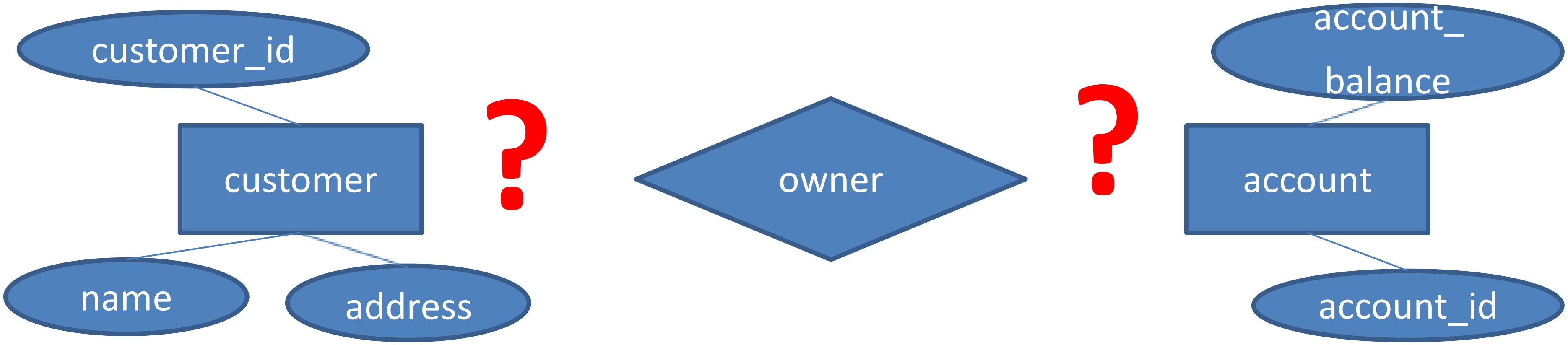
account_id

3. Constraints

- **Mapping cardinalities**
  - Concerns the number of entities to which another entity can be associated via a relationship set.
  - E.g. For each customer, how many accounts he/she can have? **One or more than one**?
- **Participation constraints**
  - Concerns whether all entities in the entity set have to participate in the relationship set.
  - E.g. whether a customer **must have** an account record, or there can be some customers **without** any accounts?

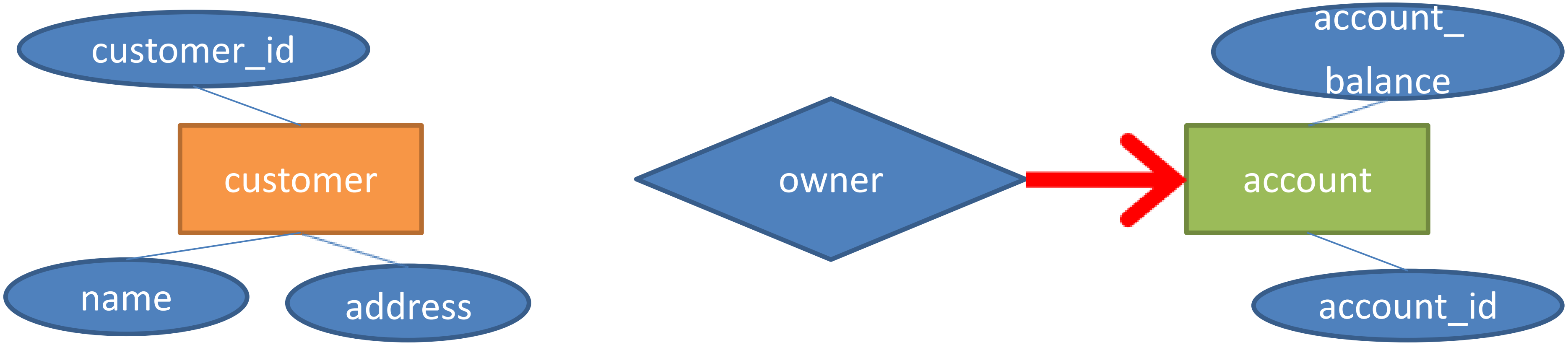
Mapping cardinalities

- We express **cardinality constraints** by drawing either **a directed line (→)**, signifying "**one**," or **an undirected line (—)**, signifying "**many**," between the relationship set and the entity set.



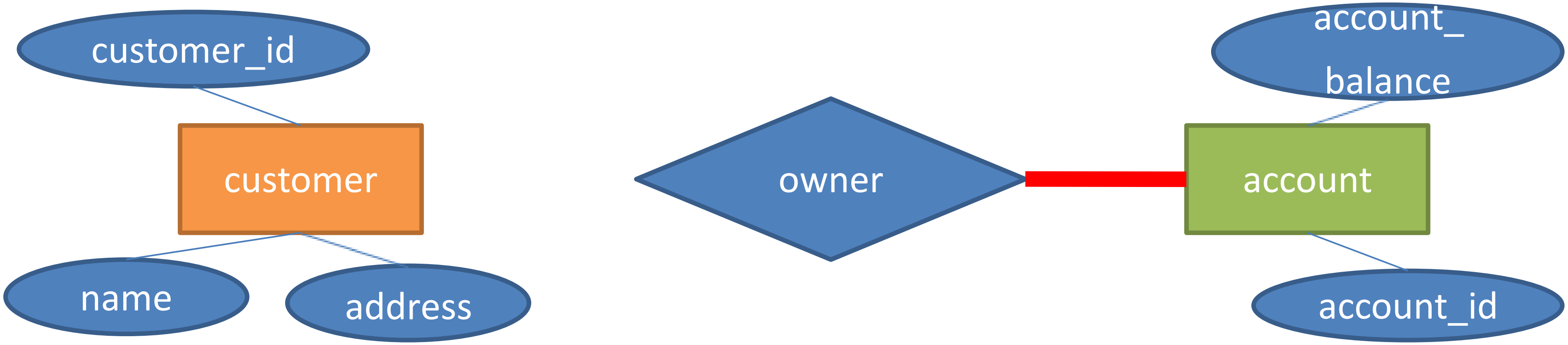
Mapping cardinalities

- A **customer** can have at most **one** **account**.



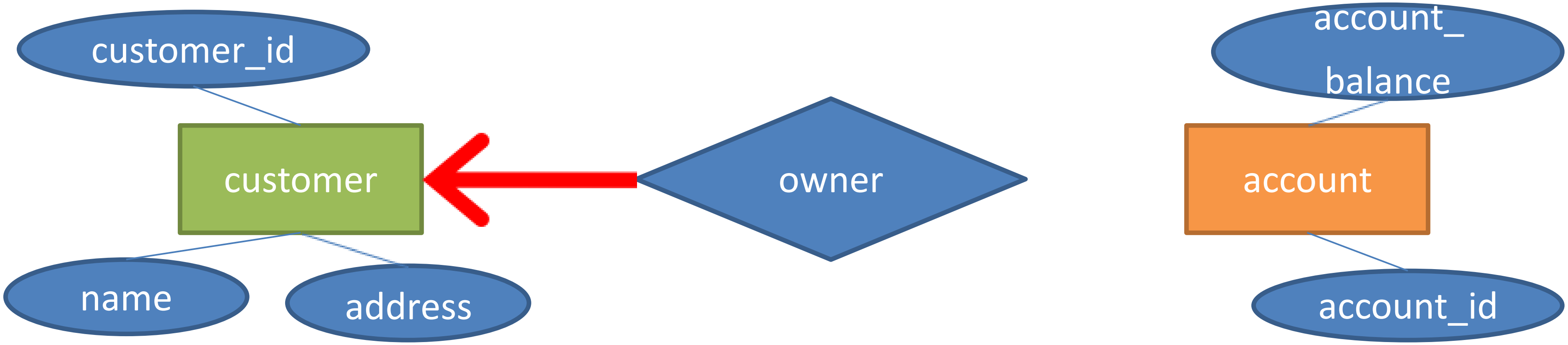
Mapping cardinalities

- A **customer** can have **more than one** **accounts**.



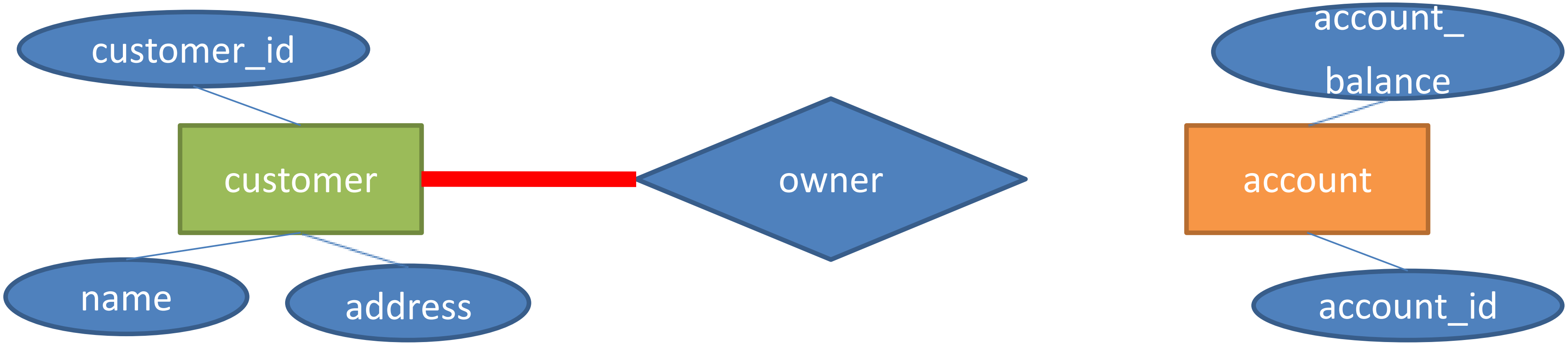
Mapping cardinalities

- An **account** can be associated with at most **one** **customer**.



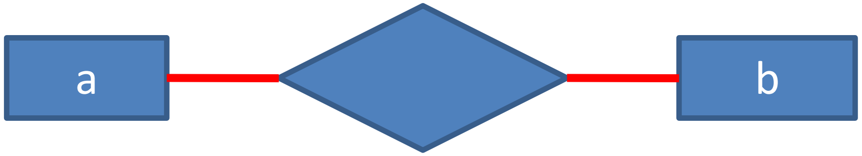
Mapping cardinalities

- An account can be associated with **more than one** **customers**.

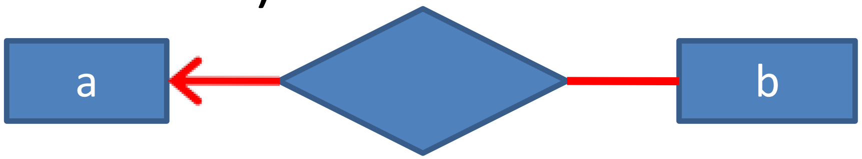


Mapping cardinalities

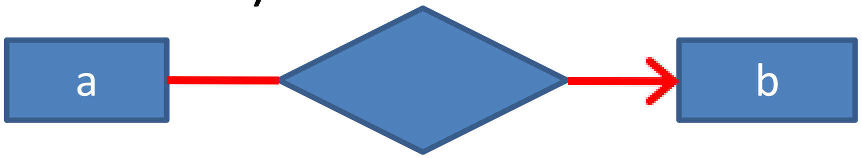
- **Different mapping relationships:**
  - Many to many.



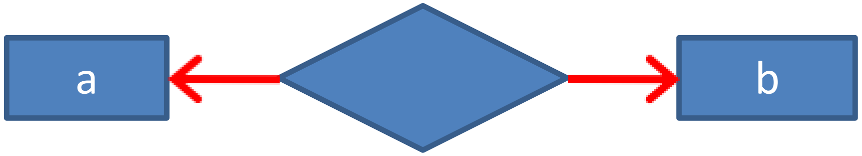
  - One to many (from a to b).



  - Many to one (from a to b).



  - One to one.



Mapping cardinalities

Please build a system to store the **customer** and **account** information of UBank. For each customer, we record his/her **customer ID**, **name** and **address**; for each account, we record its **account ID** and **account balance**. **Each customer can have one or more accounts, and each account has to be owned by only one customer**.

Step1. Identify the Entity sets.

Mapping cardinalities

Please build a system to store the **customer** and **account** information of UBank. For each customer, we record his/her **customer ID**, **name** and **address**; for each account, we record its **account ID** and **account balance**. **Each customer can have one or more accounts, and each account has to be owned by only one customer**.

Step1. Identify the Entity sets.

customer

account

Mapping cardinalities

Please build a system to store the **customer** and **account** information of UBank. For each customer, we record his/her **customer ID**, **name** and **address**; for each account, we record its **account ID** and **account balance**. **Each customer can have one or more accounts, and each account has to be owned by only one customer**.

Step2. Identify the Relationship sets.

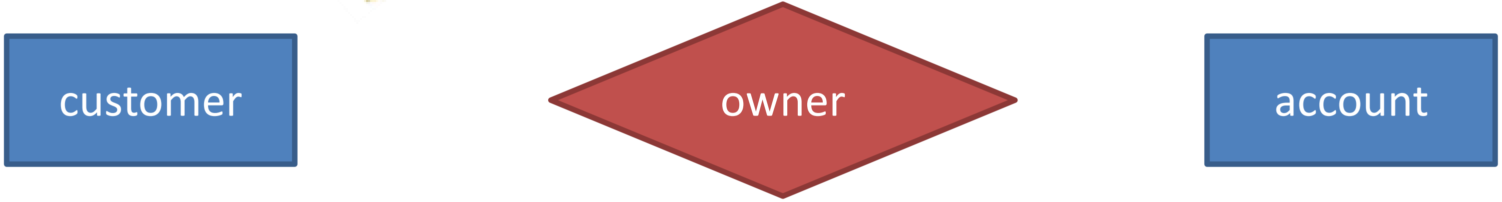
customer

account

Mapping cardinalities

Please build a system to store the **customer** and **account** information of UBank. For each customer, we record his/her **customer ID**, **name** and **address**; for each account, we record its **account ID** and **account balance**. **Each customer can have one or more accounts, and each account has to be owned by only one customer**.

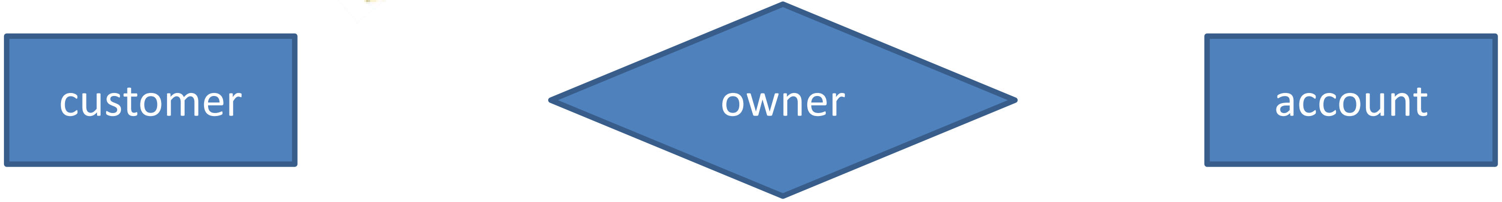
Step2. Identify the Relationship sets.



Mapping cardinalities

Please build a system to store the **customer** and **account** information of UBank. For each customer, we record his/her **customer ID**, **name** and **address**; for each account, we record its **account ID** and **account balance**. **Each customer can have one or more accounts, and each account has to be owned by only one customer**.

Step3. Identify the attributes.



customer

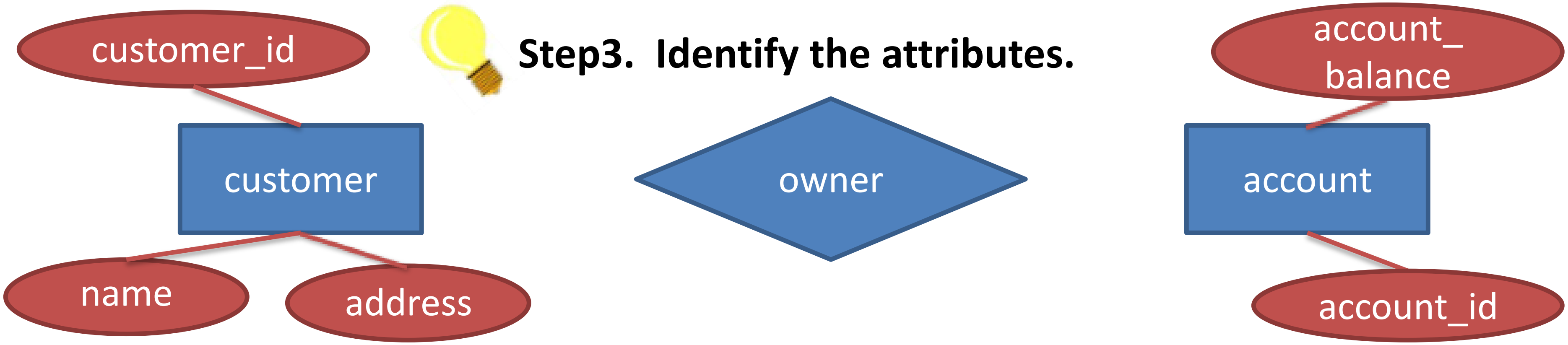
owner

account

Mapping cardinalities

Please build a system to store the **customer** and **account** information of UBank. For each customer, we record his/her **customer ID**, **name** and **address**; for each account, we record its **account ID** and **account balance**. **Each customer can have one or more accounts, and each account has to be owned by only one customer**.

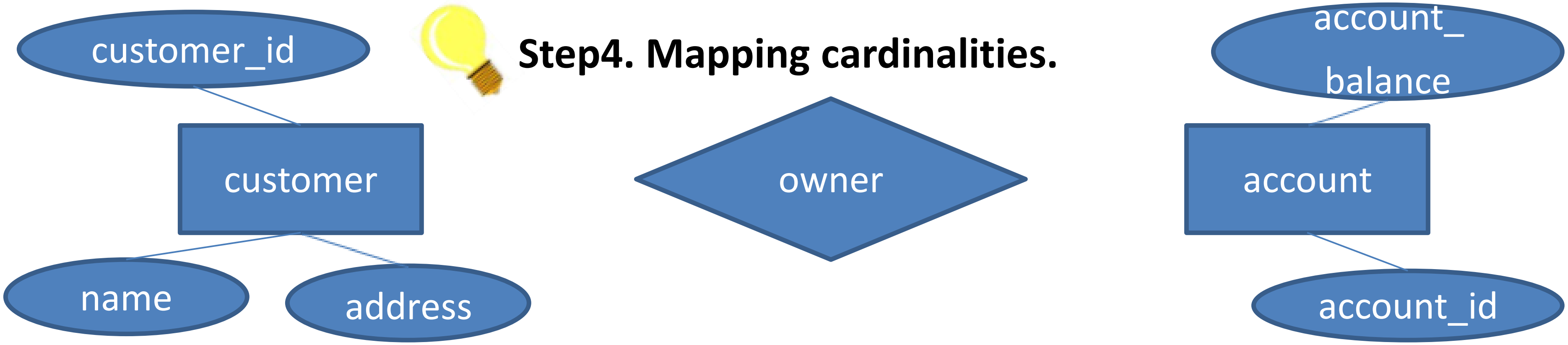
Step3. Identify the attributes.



Mapping cardinalities

Please build a system to store the **customer** and **account** information of UBank. For each customer, we record his/her **customer ID**, **name** and **address**; for each account, we record its **account ID** and **account balance**. **Each customer can have one or more accounts, and each account has to be owned by only one customer**.

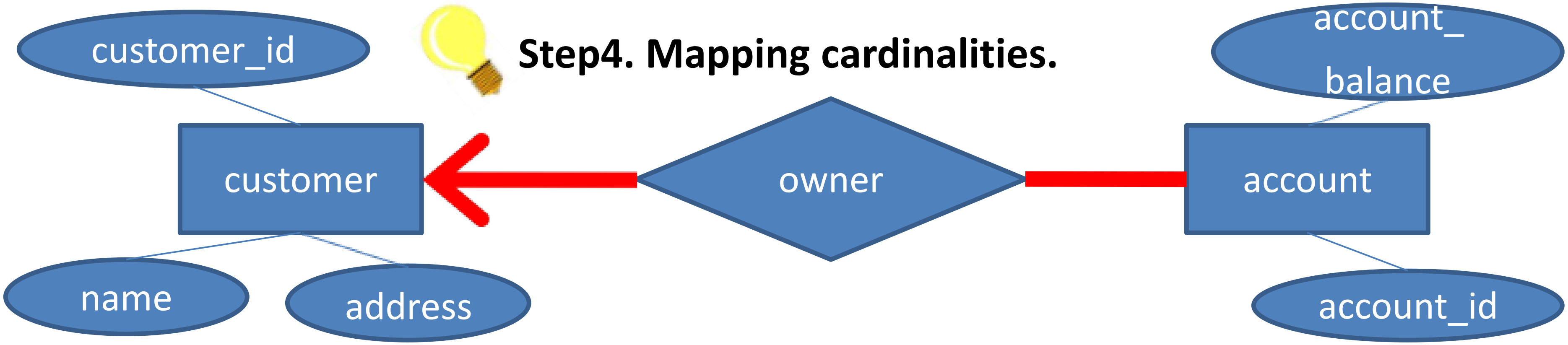
Step4. Mapping cardinalities.



Mapping cardinalities

Please build a system to store the **customer** and **account** information of UBank. For each customer, we record his/her **customer ID**, **name** and **address**; for each account, we record its **account ID** and **account balance**. **Each customer can have one or more accounts, and each account has to be owned by only one customer**.

Step4. Mapping cardinalities.



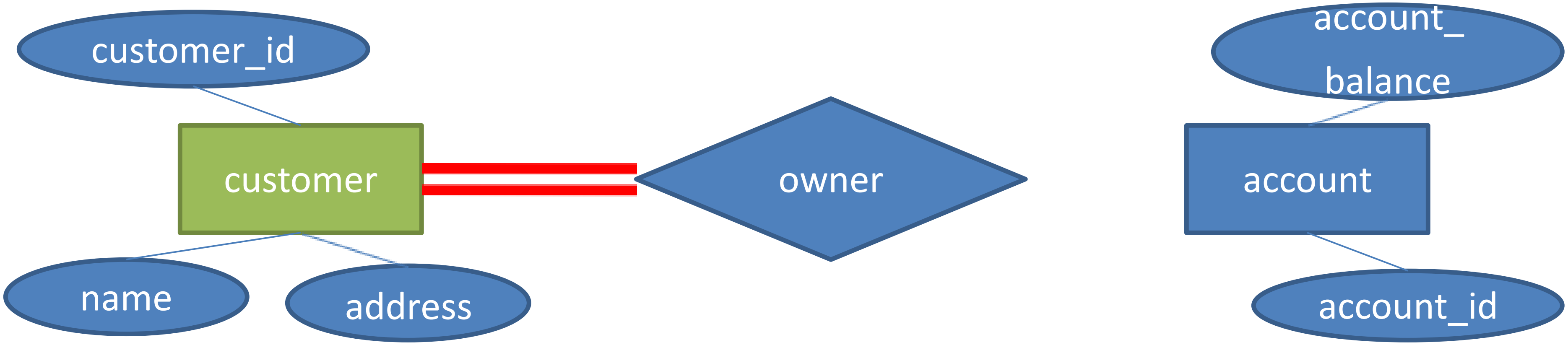
Note: this ER-Diagram is **incomplete**! **Some more steps in the next few slides including the participation, primary keys …etc**

Participation constraints

- Concerns whether all entities in the entity set have to participate in the relationship.
  - **Total participation** (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set.
  - **Partial participation** (indicated by single line): some entity may not participate in any relationship in the relationship set.

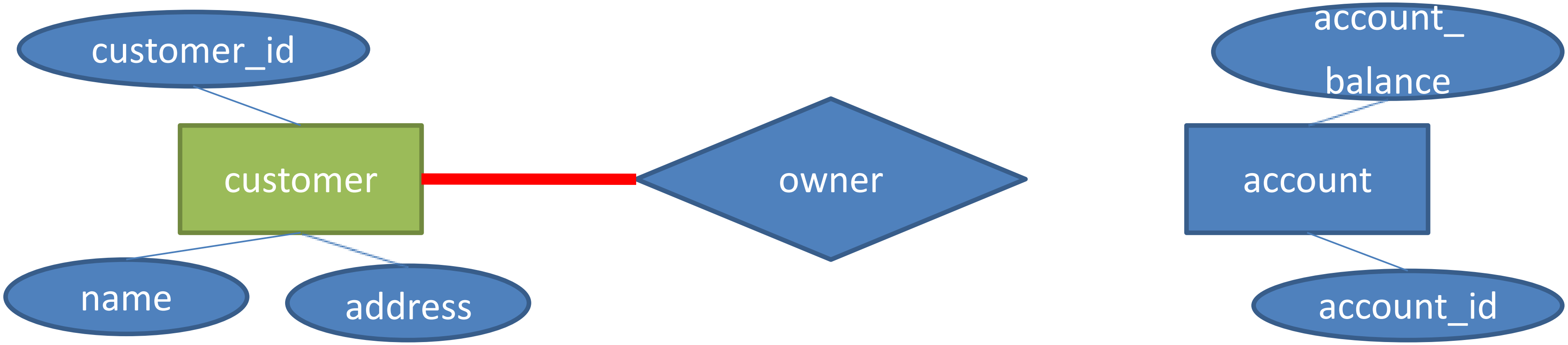
Participation constraints

- Each customer **must have an account**.
  - **Total participation**: All **customers** must participate in the owner relationship.



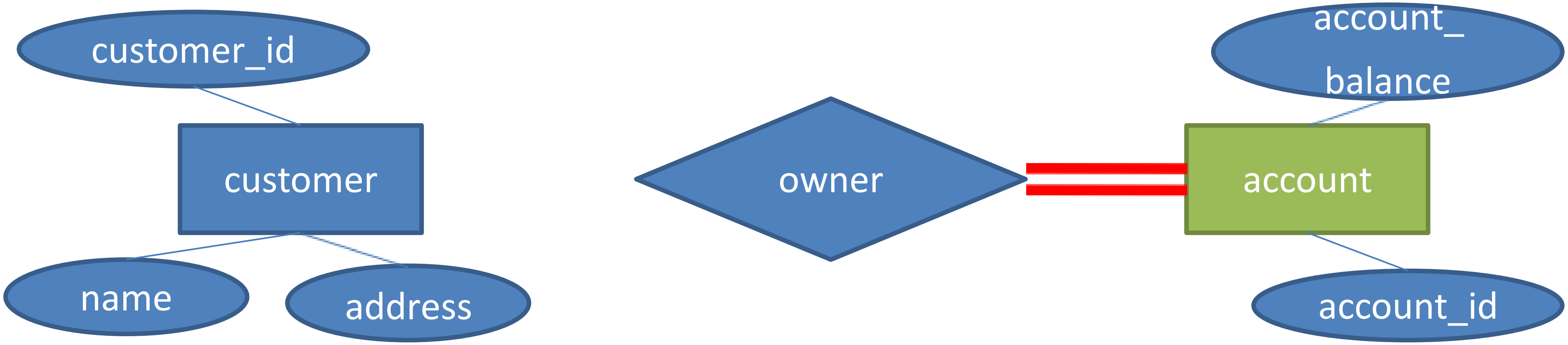
Participation constraints

- Not all customers are required to have an account.
  - **Partial participation**: Not all **customers** participate in the owner relationship.



Participation constraints

- Each account must be owned by customers.
  - **Total participation**: All **accounts** participate in the owner relationship.



Practical issues

- **As a professional DB designer, you have to be able to:**
  - Understand and model the data of an application using a E-R diagram.
  - Interact with the client to work out a clear problem definition.
  - Realize the missing information and ask your client for clarification.
  - Provide professional suggestions to better design the database for the specific application.

Practical issues

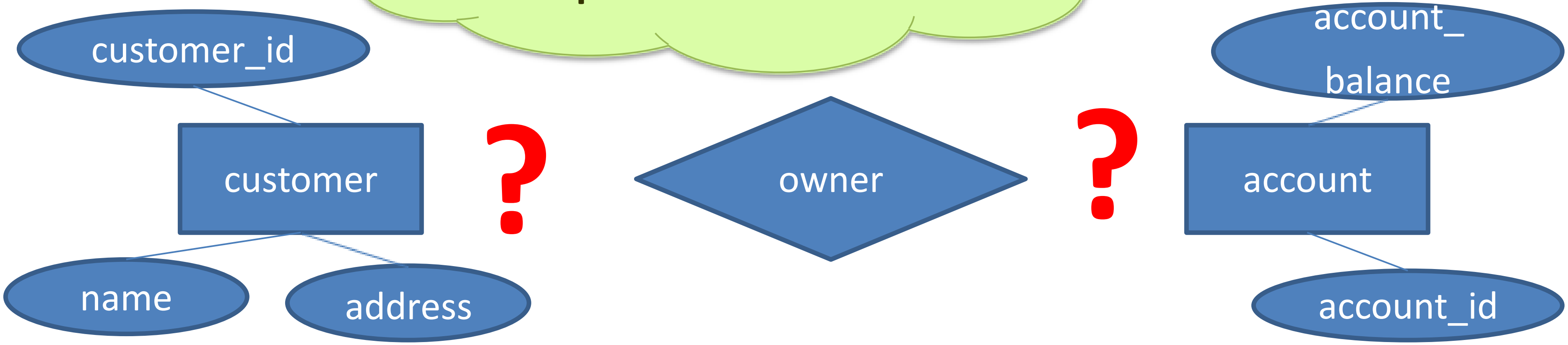
Please build a system to store the customer and account information of our bank. For each customer, we record his/her customer ID, name and address; for each account, we record its account ID and account balance.

Do we have enough information to model the data of this application? What questions should I ask?

Practical issues

Please build a system to store the customer and account information of our bank. For each customer, we record his/her customer ID, name and address; for each account, we record its account ID and account balance.

Do we have enough information to model the data of this application? What questions should I ask?



To Recap

- **Entity and entity set, Entities have Attributes**
- **Relationship and relationship set**
- **In the E-R Diagram**
  - **Rectangles** – entity sets.
  - **Ellipses** – attributes.
  - **Line between a rectangle and an ellipse** – link between an attribute and an entity set.

To Recap

- **Mapping cardinalities (many, one)**
  - Concerns the number of entities to which another entity can be associated via a relationship set.
  - E.g. For each customer, how many accounts he/she can have? **One or more than one**?
- **Participation constraints (full, partial)**
  - Concerns whether all entities in the entity set have to participate in the relationship set.
  - E.g. whether a customer **must have** an account record, or there can be some customers **without** any accounts?

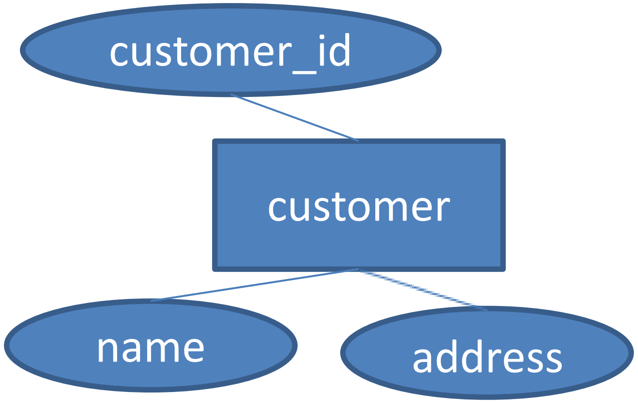
Keys

Attributes

- Super key
- Candidate keys
- Primary key

Super key

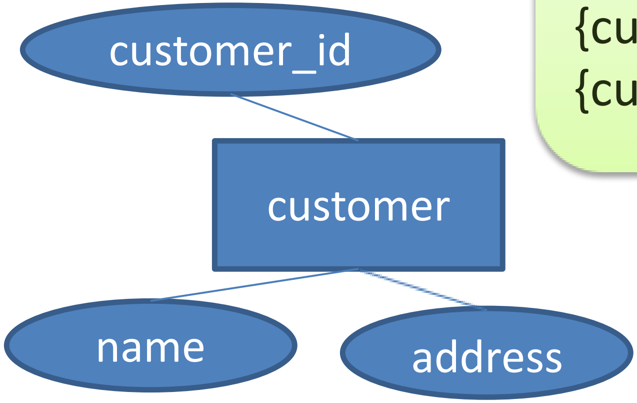
- A **super key** of an entity set is a set of one or more attributes whose values uniquely determine each entity.
  - No two entities have exactly the same values in super key.



Super key

- A **super key** of an entity set is a set of one or more attributes whose values uniquely determine each entity.
  - No two entities have exactly the same values in super key.

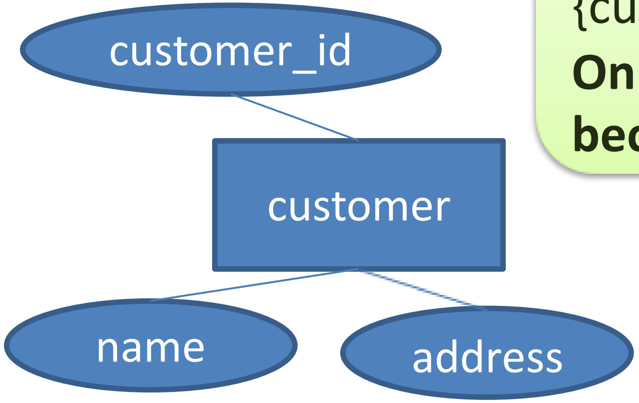
In this example, if each customer has his/her **unique customer_id**, then
{customer_id, name} is a super key.
{customer_id, address} is another super key.
{customer_id, name, address} is also a super key.



Candidate key

- A **candidate key** of an entity set is a **minimal** super key.
  - Minimal – no redundant attributes, i.e., no subset of a candidate key is still a key.

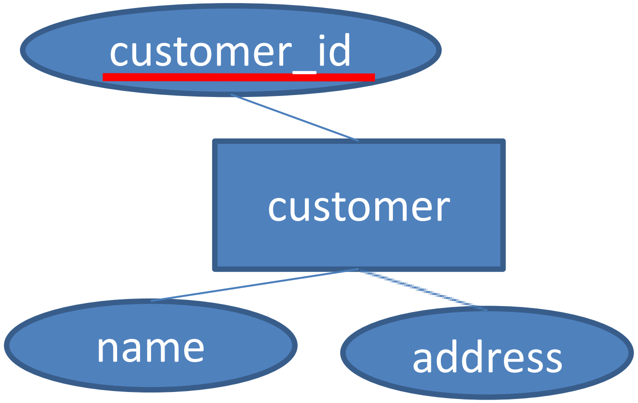
Although the following are super keys:
{ customer_id, name}, {customer_id, address},
{customer_id, name, address}
Only the {customer_id} is a candidate key because it is minimal.



Question:
Can it be more than one candidate key?

Primary key

- Although several candidate keys may exist, one of the candidate keys is selected to be the **primary key**.
- **In the E-R Diagram**
  - **Underline the attribute** – The attribute is a primary key of the entity.

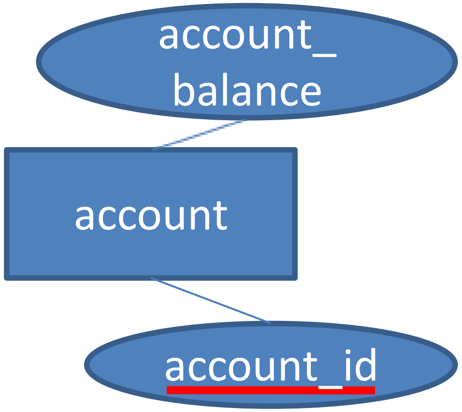


customer_id

customer

name

address



account_ balance

account

account_id

Section 2.2

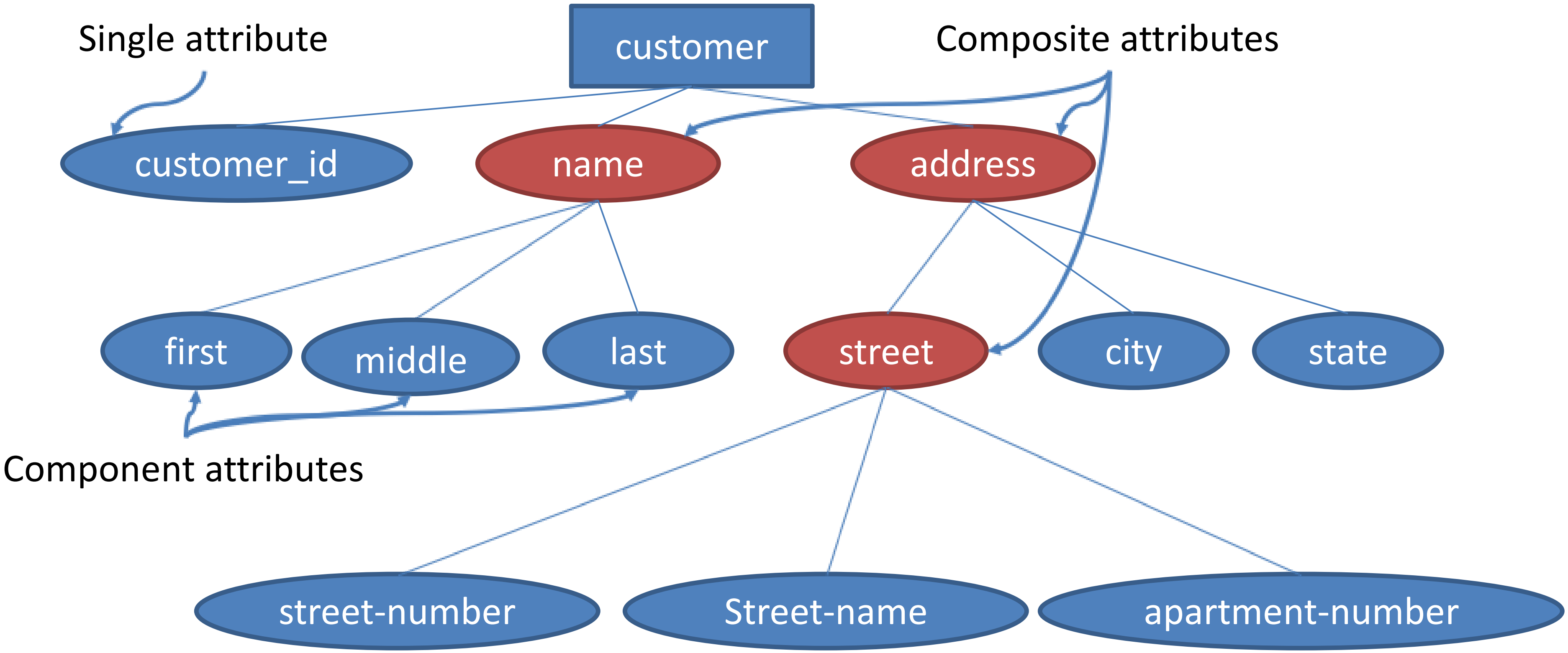
More on E-R Diagram

More on E-R Diagram

- Different attribute types
- Weak entity set
- Role
- Specialization

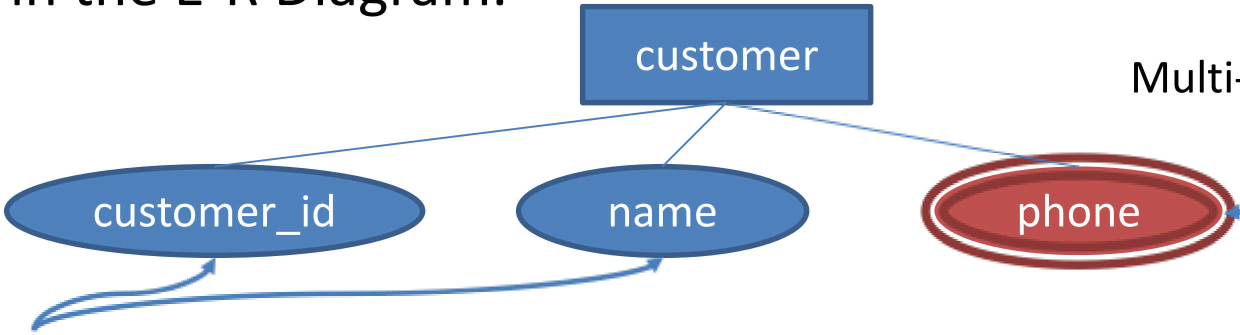
Different attribute types

- **Single v.s. Composite attributes**



Different attribute types

- **Single-valued v.s. Multi-valued attributes**
  - Multi-valued attributes are represented by **double ellipses** in the E-R Diagram.



Multi-valued attribute

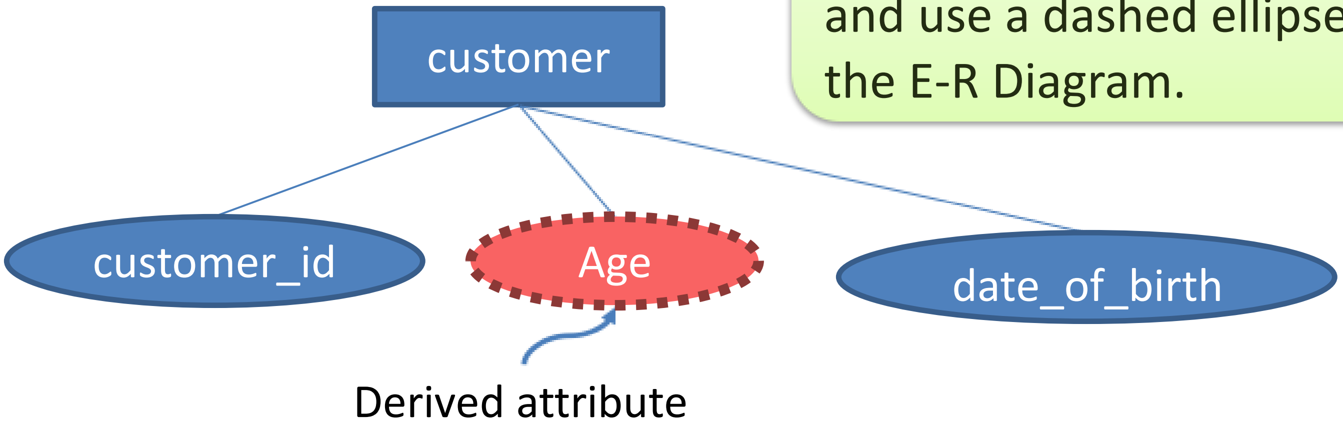
Single-valued attributes

customer_id	name	phone
1	Kit	6012 3456, 9888 8888, 2857 8435
2	Yvonne	6987 6543, 2859 1104
3	Jolly	9876 1234, 2857 8434

Different attribute types

- **Derived attribute**
  - Values in this attribute can be **derived** from other attributes.
  - Derived attributes are represented by **dashed ellipses** in the E-R Diagram.

Since “**age**” can be derived from the “**date of birth**”, we treat “age” as a derived attribute, and use a dashed ellipse to represent it in the E-R Diagram.



Weak Entity Set

- **An entity set that does not have a primary key is referred to as a weak entity set.**
- **The existence of a weak entity set depends on the existence of an identifying entity set.**
  - We need the identifying entity set to help to uniquely identify the entities in the weak entity set.

Create a database for storing the NBA teams and the NBA players…

Weak Entity Set

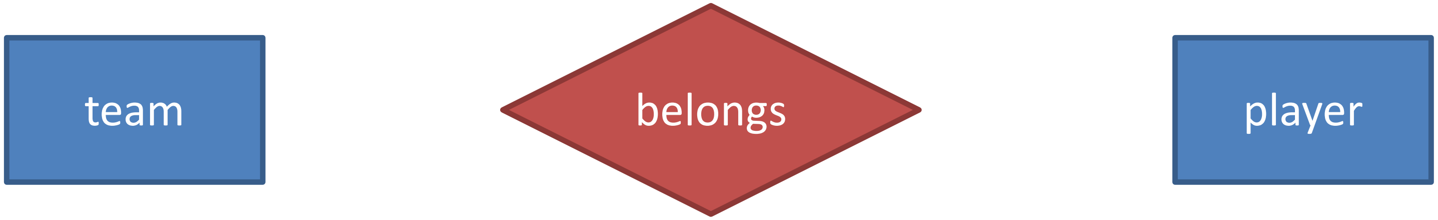
team

player

1. Identify Entity sets:
We have to store the “NBA Teams” and “NBA Players”.

Create a database for storing the NBA teams and the NBA players…

Weak Entity Set


team

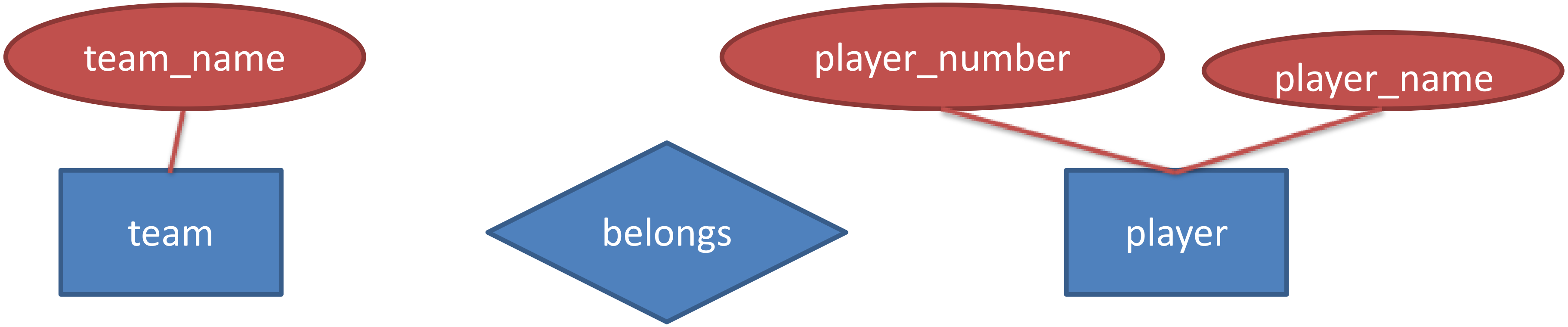
belongs

player

2. Identify relationship set:
Each player **belongs** to a team.

Create a database for storing the NBA teams and the NBA players…

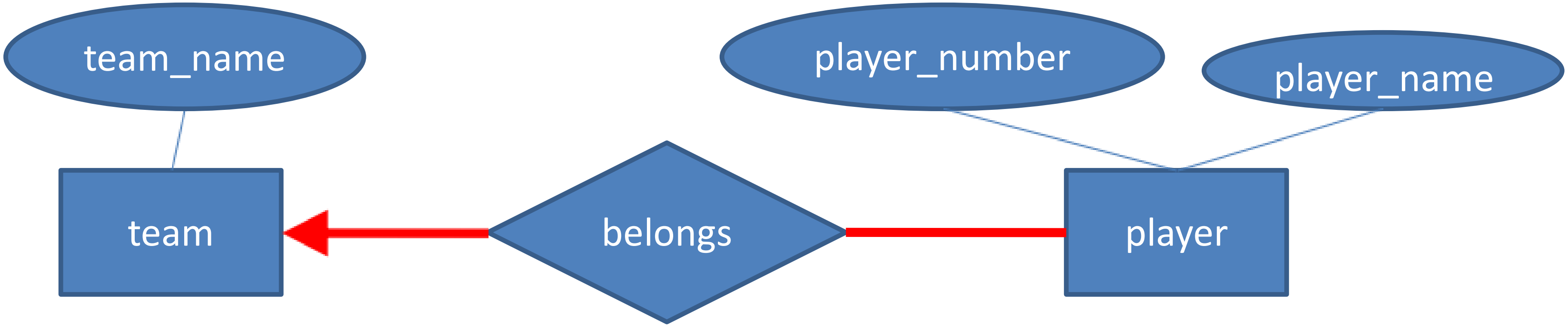
Weak Entity Set



3. Identify the attributes of the entity sets: For each NBA team, we store the team’s **name**; for each player, we store his **number** and **name**.

Create a database for storing the NBA teams and the NBA players…

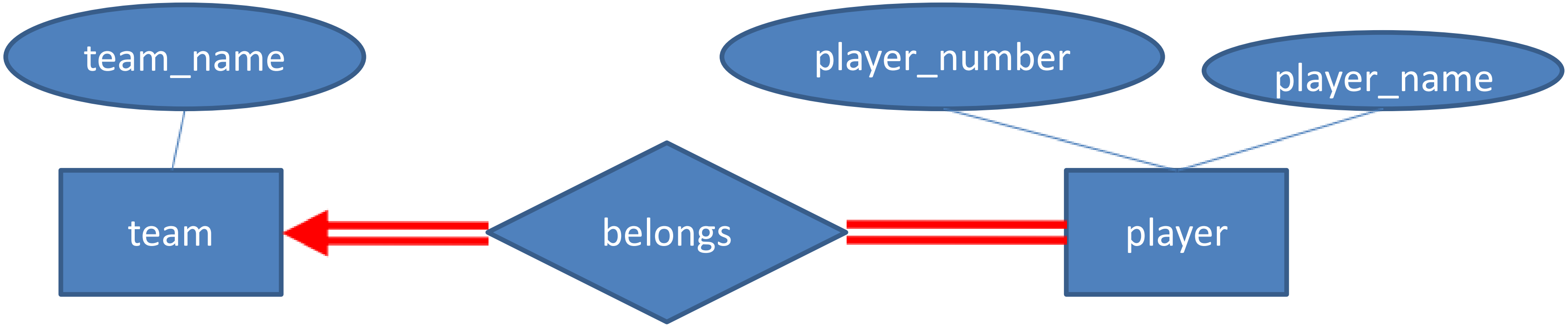
Weak Entity Set



4. Identify the mapping cardinalities:
A team can have more than one players.
A player can belongs to only one team.

Create a database for storing the NBA teams and the NBA players…

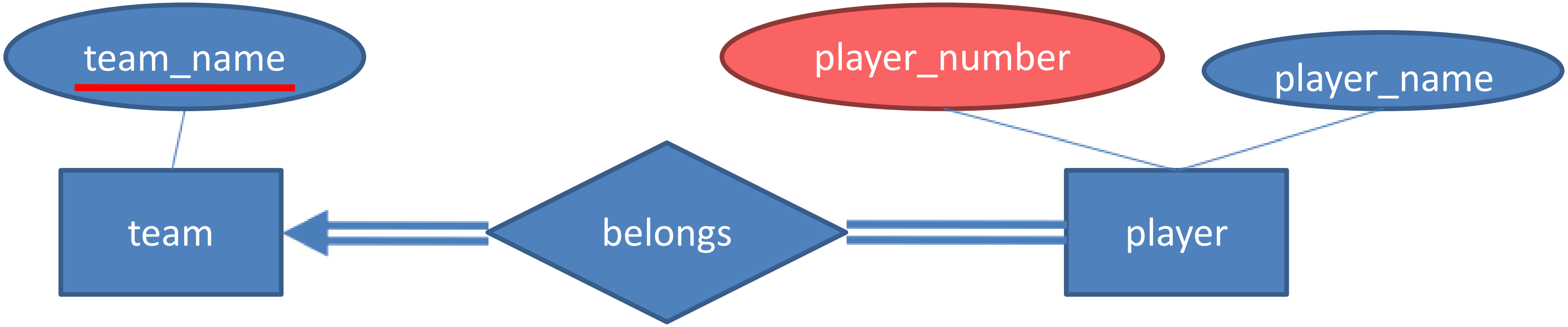
Weak Entity Set



5. Identify the total/ partial participation:
A team must have some players.
A player must belong to a team.

Create a database for storing the NBA teams and the NBA players…

Weak Entity Set



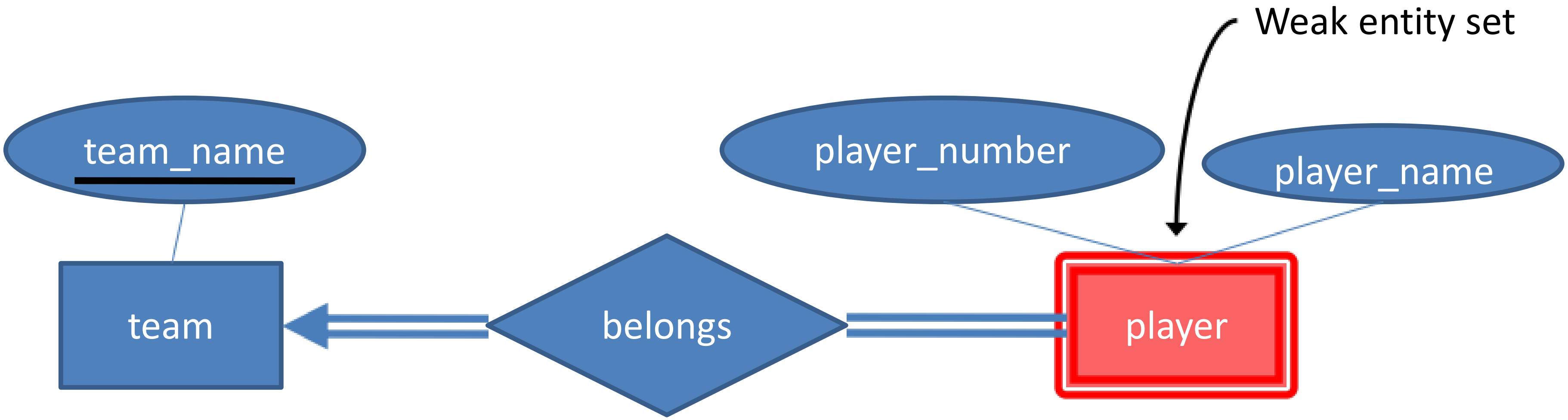
6. Identify the primary key of the entity sets:

The team name can uniquely identify a team.

Problem: The player_number cannot uniquely identify a player!

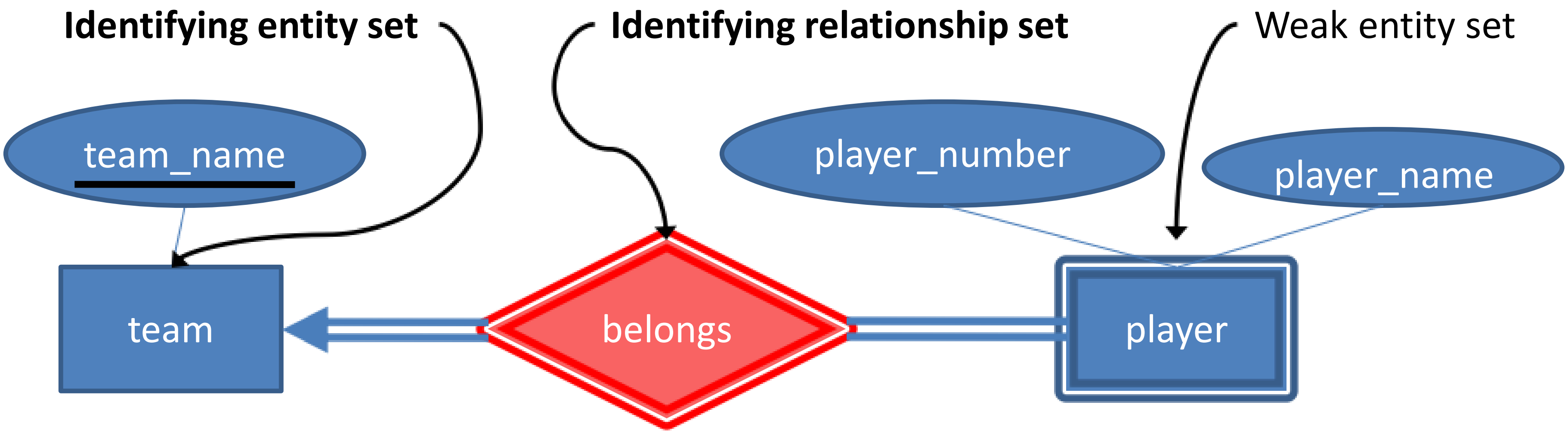
E.g., Michael Jordan and LeBron James were both #23!

Weak Entity Set



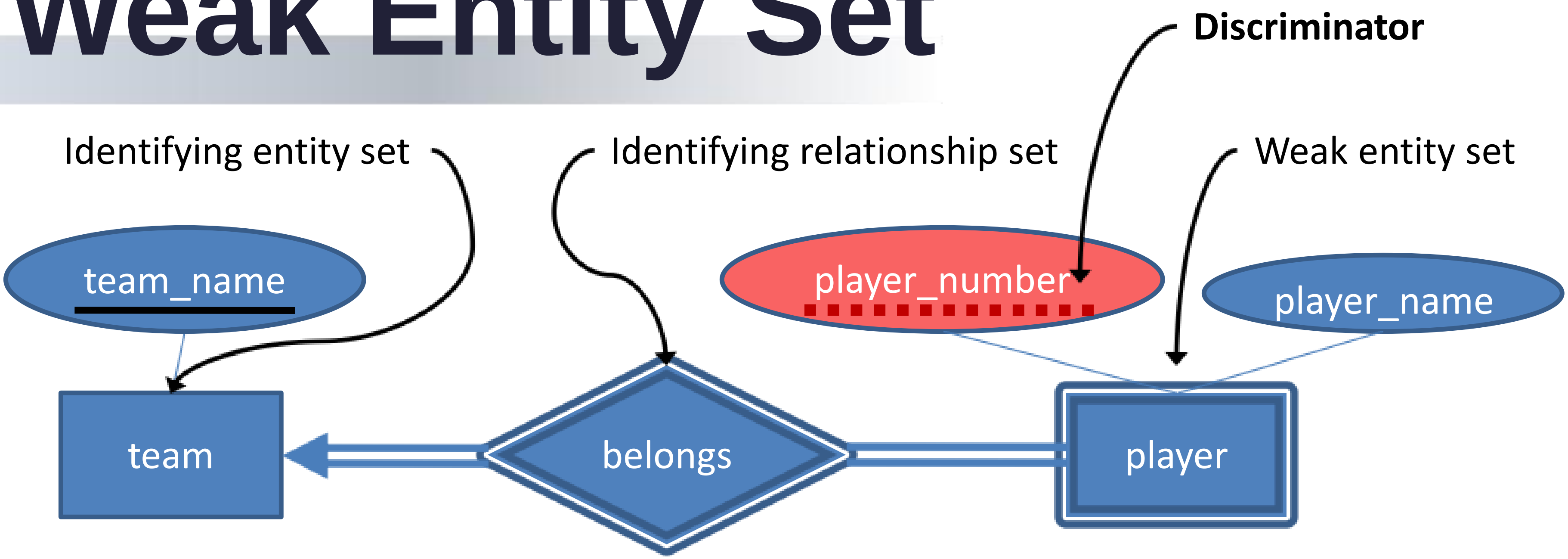
- **An entity set that does not have a primary key is referred to as a weak entity set.**
  - We depict a weak entity set by a **double rectangle**.

Weak Entity Set



- **The existence of a weak entity set depends on the existence of an identifying entity set.**
  - The weak entity set must relate to its identifying entity set via a **total**, **many-to-one** **identifying relationship set** from the weak entity set to the identifying entity set.
  - We depict an identifying relationship set as a **double diamond**.

Weak Entity Set



- The **discriminator** (or, partial key) of a weak entity set is a set of attributes that distinguish among the weak entities that depend on the same identifying entity.
  - The primary key of a weak entity set is formed by the primary key of the identifying entity set plus the weak entity set’s discriminator.

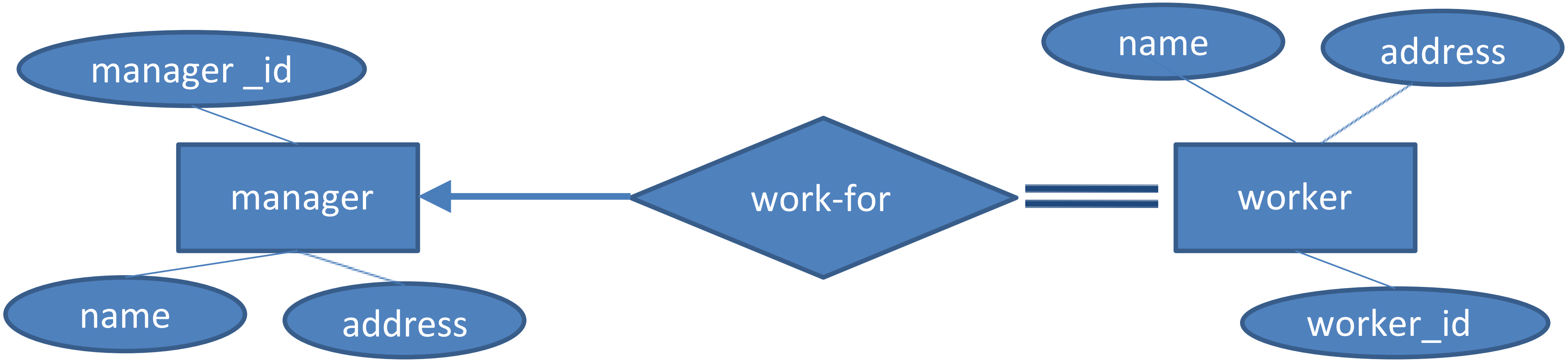
Role

- Entity sets of a relationship need not be distinct.

Role

Entity sets of a relationship need not be distinct.

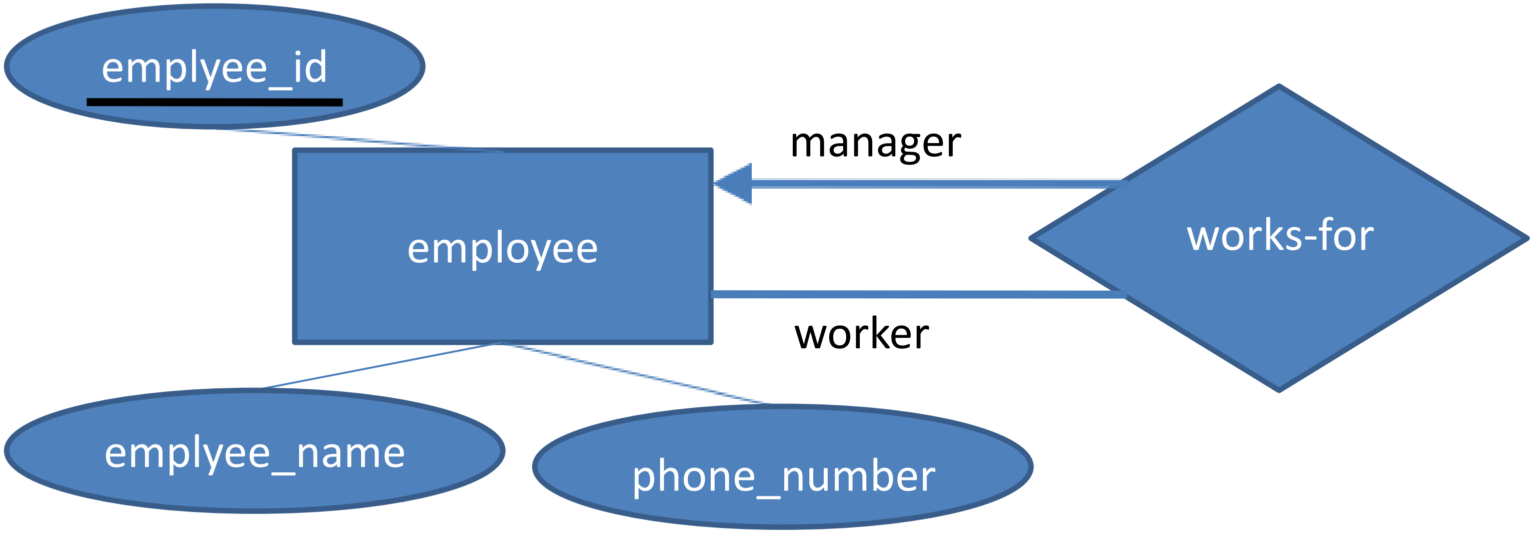
Please build a system to store the **manager** and **worker** information for UBank. For each manager, we record his/her **manager ID**, **name** and **address**; for each worker, we record its **worker ID, name** and **address**. **Each manager can have no, one or more workers, and each worker has to report to only one manager.**



Any problems in the above ER diagram?

Role

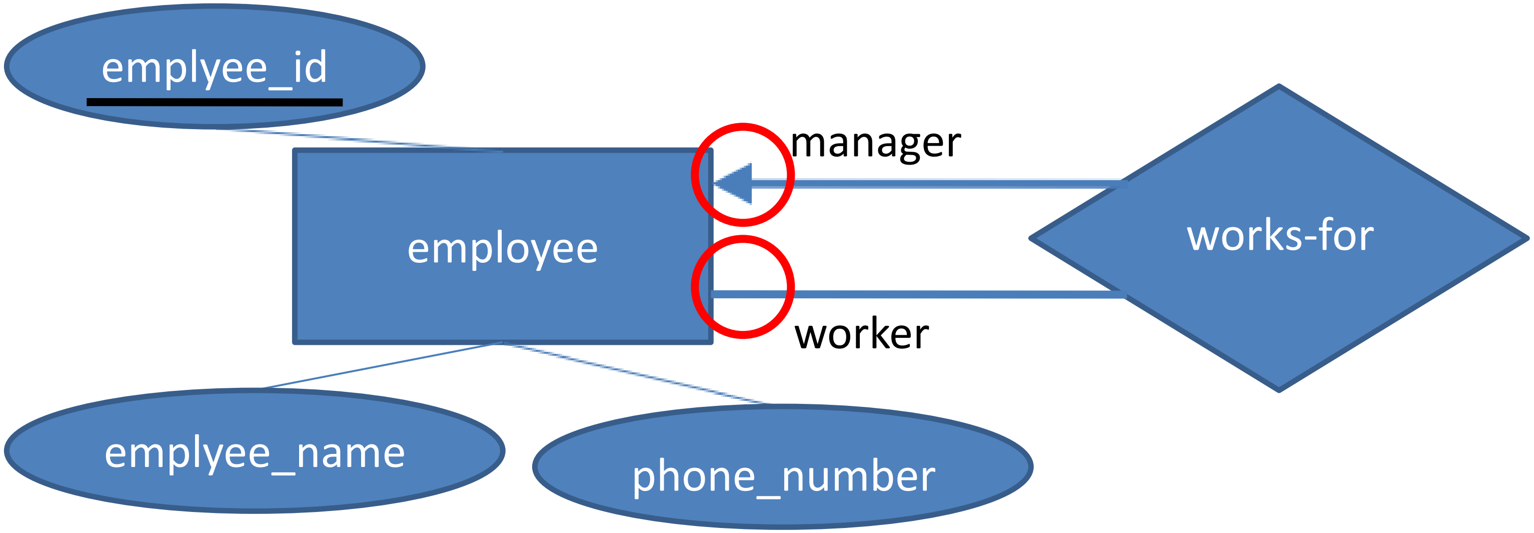
- **Entity sets of a relationship need not be distinct.**



- The label “manager” and “worker” are called **roles**. They specify how employee entities interact via the “works-for” relationship set.

Role

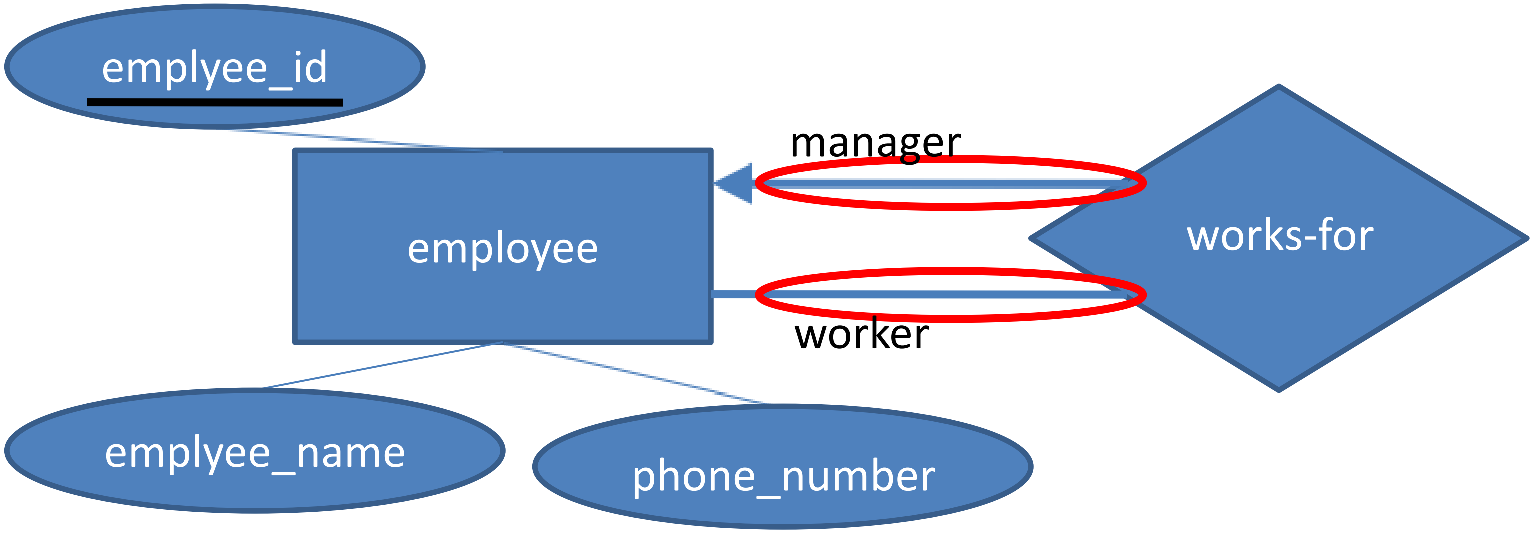
- **Entity sets of a relationship need not be distinct.**



- **Cardinality**
  - An employee (worker) works for one manager.
  - An employee (manager) can have more than one workers work for him/her.

Role

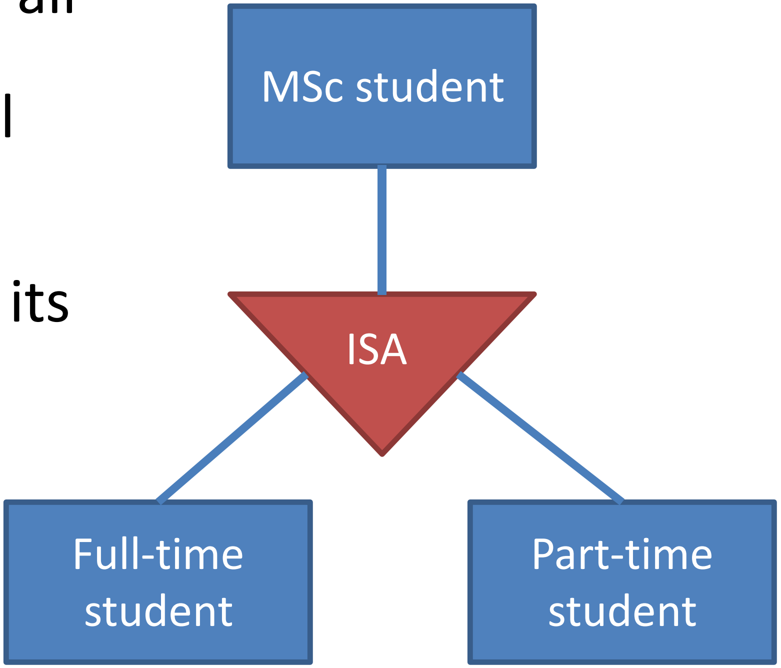
- **Entity sets of a relationship need not be distinct.**



- **Participation**
  - An employee (worker) may not work for any manager.
  - An employee (manager) can have no workers work for him/her.

Specialization

- **Specialization**
  - We designate sub-groupings within an entity set that are distinctive from other entities in the set.
  - A lower-level entity set inherits all attributes and relationship set participation of the higher-level entity set to which it is linked.
  - Lower-level entity set can have its own attributes.

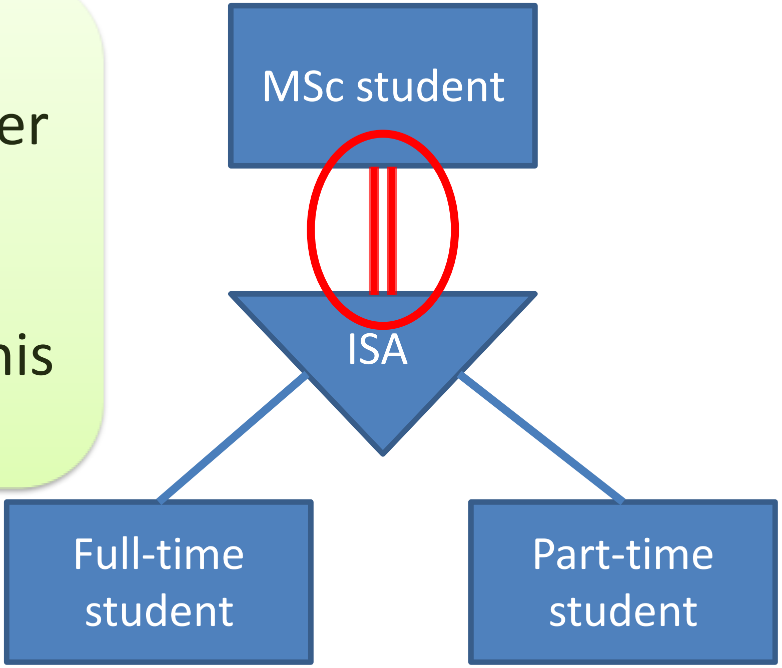


Specialization

Total or partial

- Specifies whether an entity in the higher level-entity set must belong to at least one of the lower-level entity sets within a specialization.

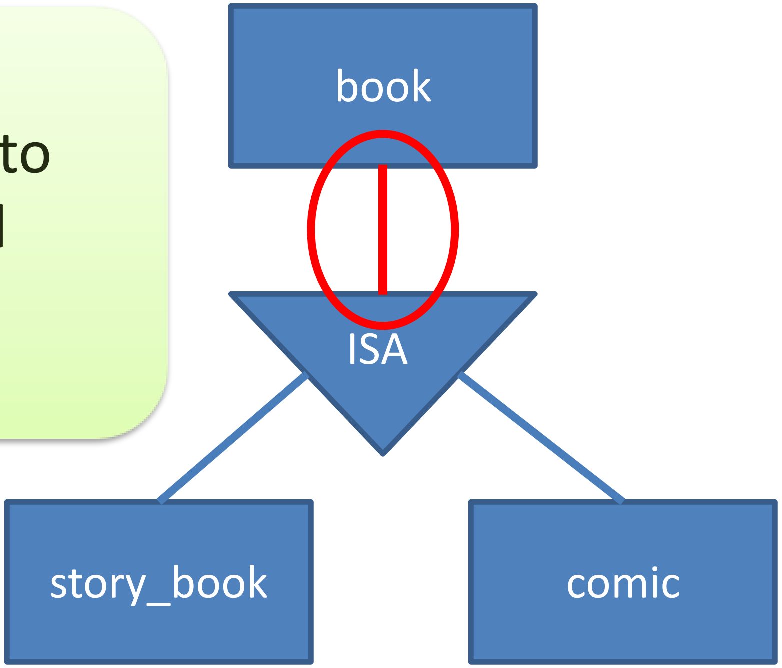
Total specialization:
An MSc student MUST BE either a full-time student or a part-time student, so all MSc students must participate in this specialization.



Specialization

- **Total or partial**
  - Specifies whether an entity in the higher level-entity set must belong to at least one of the lower-level entity sets within a specialization.

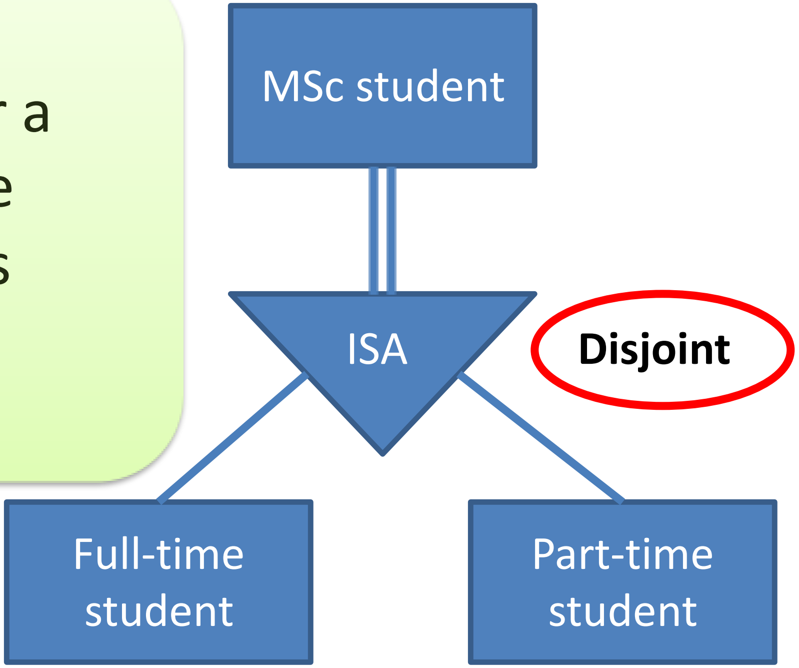
Partial specialization:
A book may not be specialized to story book or comics, so **not all books** are participating in this specialization.



Specialization

- **Disjoint or overlapping**
  - Constraints on whether entities may belong to more than one lower-level entity set within a single specialization.

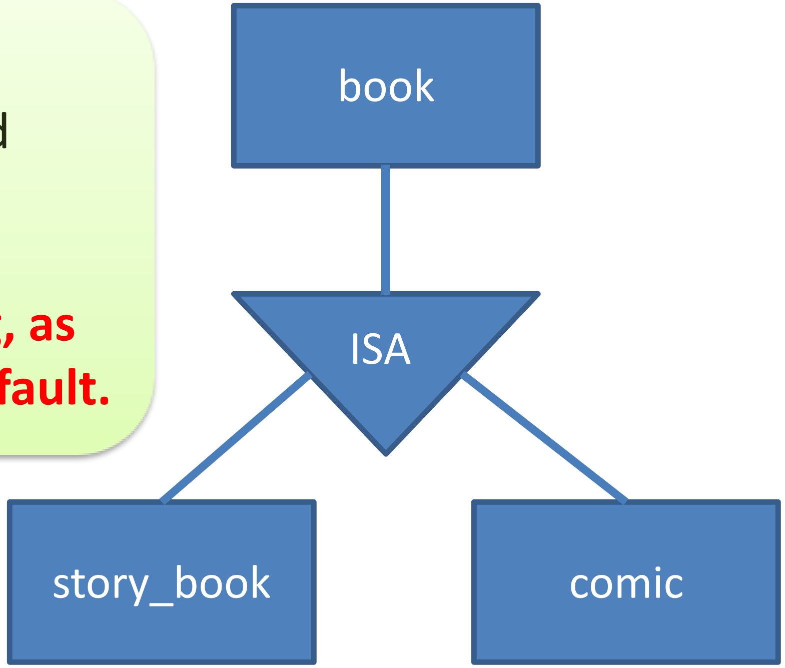
Disjoint specialization:
An MSc student must be either a full-time student or a part-time student, so the specialization is disjoint. **We use a keyword "Disjoint" to indicate it.**



Specialization

- **Disjoint or overlapping**
  - Constraints on whether entities may belong to more than one lower-level entity set within a single specialization.

Overlapping specialization:
A book can be both a story book and comic, so the specialization is overlapping.
We do not need to specify anything, as overlapping specialization is the default.



To Recap

- **Entity and entity set, Entities have Attributes**
- **Relationship and relationship set**
- **In the E-R Diagram**
  - **Rectangles** – entity sets.
  - **Ellipses** – attributes.
  - **Line between a rectangle and an ellipse** – link between an attribute and an entity set.

To Recap

- **Mapping cardinalities (many, one)**
  - Concerns the number of entities to which another entity can be associated via a relationship set.
  - E.g. For each customer, how many accounts he/she can have? **One or more than one**?
- **Participation constraints (full, partial)**
  - Concerns whether all entities in the entity set have to participate in the relationship set.
  - E.g. whether a customer **must have** an account record, or there can be some customers **without** any accounts?

To Recap

- Super key
- Candidate keys
- Primary key
- Different attribute types (Single/Composite/component, single-valued, multi-valued, derived)
- Weak entity set
- Role
- Specialization (ISA, Total/partial, disjoint/overlap)

Lecture 2

END

COMP3278B

Introduction to Database Management Systems

Dr. Ping Luo
Email : pluo@cs.hku.hk

Department of Computer Science, The University of Hong Kong