Chapter 2

Entity-Relationship Model

COMP3278C
Introduction to Database Management Systems

Dr. CHEN, Yi
Email: chenyi1@hku.hk

School of Computing & Data Science, The University of Hong Kong

Acknowledgement: Dr. Chui Chun Kit, Dr. Reynold Cheng, Dr. Ping Luo

We are going to learn …

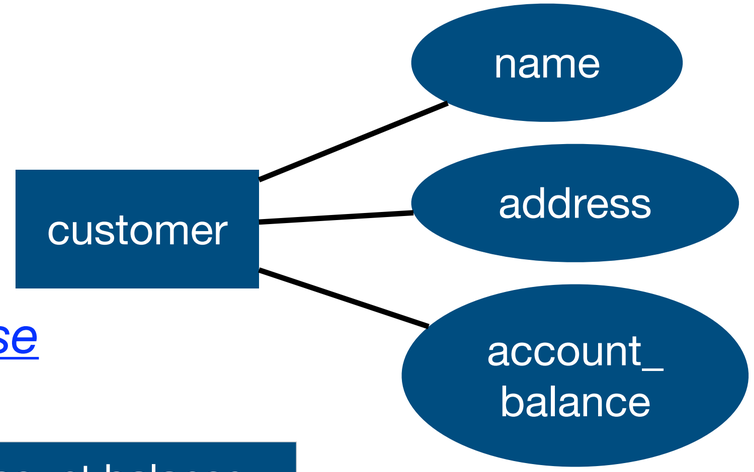
- Outcome 1: **Information Modeling**
  - Able to understand the modeling of real-life information in database systems.

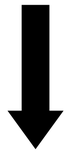
1. Entity-Relation Diagram (E-R diagram)

   Visualizes the data structure and their internal relationships

2. E-R Diagram to Relational Tables

   Represents the actual format used to store data in the database





Name	Address	Account balance
Yi	CB306, HKU, …	$100

- Outcome 3: **System Design**
  - Able to design an efficient and reliable database system.

1. E-R Diagram

- Entity and entity set
- Relationship and relationship set
- Constraints
- Keys
- Weak entity set
- Role
- Specialization
- Different attribute types
- E-R design decision

1. E-R Diagram: Entity and entity set

We have customers Alice, Bob, Cathy …

- Customer Alice is an **Entity**
  - an object that exists and is distinguishable from other objects
- Customers Alice, Bob, Cathy and etc. form to an **Entity set**
  - A set of entities that has the same type and share the same properties
- Customer Alice has a **attribute**, which is the name “Alice”

1. E-R Diagram: Entity and entity set

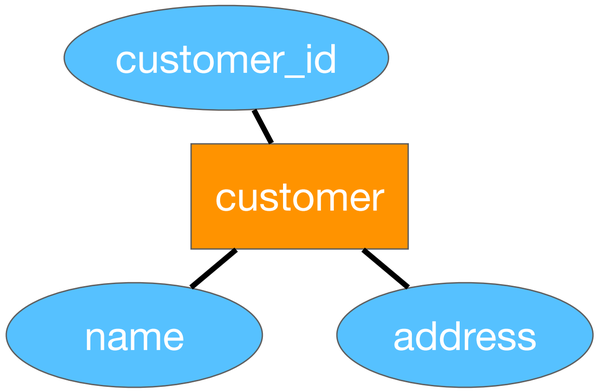
We have customers Alice, Bob, Cathy …

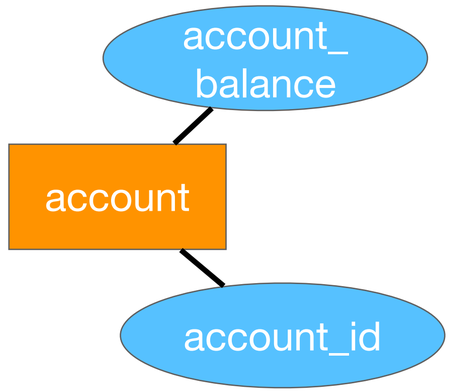
We have accounts 111111, 222222, 333333 … Account 111111 has the balance $100 …

- Customer Alice is an **Entity**
  - an object that exists and is distinguishable from other objects
- Customers Alice, Bob, Cathy and etc. form to an **Entity set**
  - A set of entities that has the same type and share the same properties
- Customer Alice has a **attribute**, which is the name “Alice”

1. E-R Diagram: Entity and entity set

- In E-R Diagram
  - Entity set - **rectangle**
  - Attribute - **ellipse**
  - Relation representing an entity and its attribute - **line** between a rectangle and an ellipse





1. E-R Diagram: Relationship and Relationship set

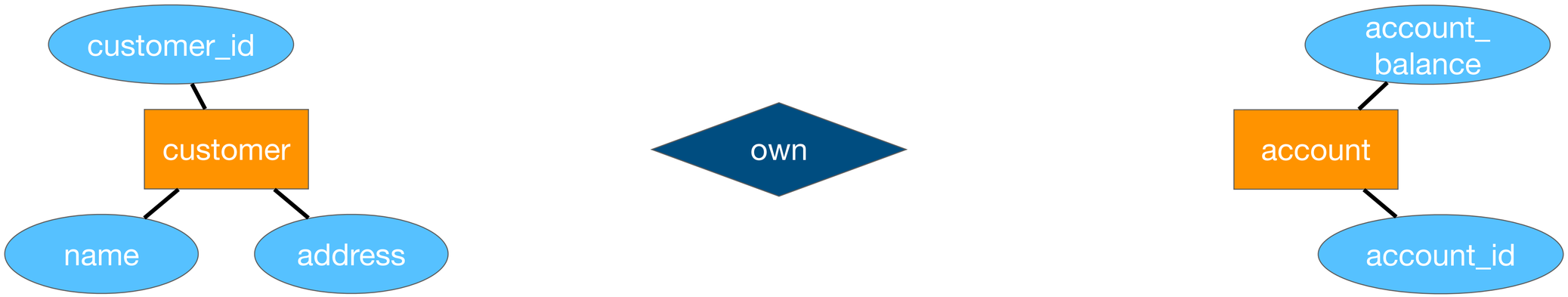
Alice's account is 111111;
Bob's account is 222222;
Cathy's account is 333333 …

- There is an "own" relationship between the customer Alice and the account 111111
- **Relationship**: an association among entities

- **Relationship set**
  - A set of relationships of the same type

1. E-R Diagram: Relationship and Relationship set

- In E-R Diagram
  - Relationship set - diamond



1. E-R Diagram: Constraints

For each customer, how many accounts he/she can have? One or more than one?

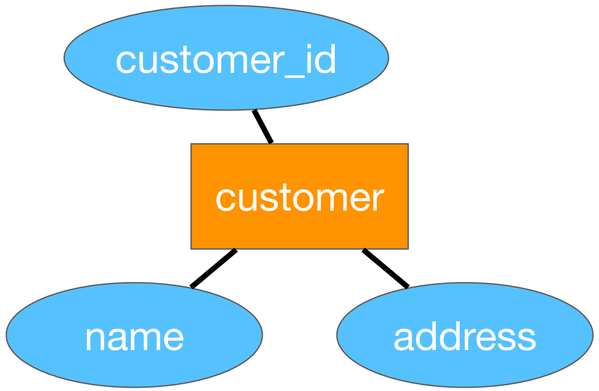
➡ • **Mapping cardinalities**

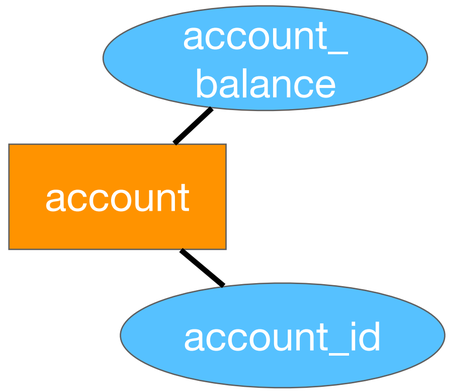
  • Concerns the number of entities to which another entities can be associated via a relationship set

Whether a customer must have an account, or there can be some customer without any accounts?

➡ • **Participation constraints**

  • Concerns whether all entities in the entity set have to participate in the relationship set





1. E-R Diagram: Constraints

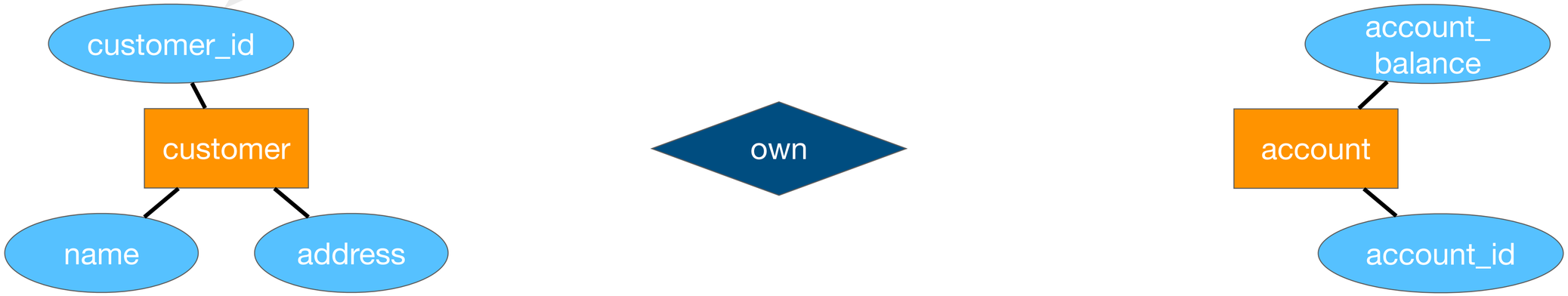
- In E-R Diagram: **Mapping cardinalities**
  - Draw a **directed line "->"**, signifying **one**, between the relationship set and the entity
  - Draw an **undirected line "-"**, signifying **many**, between the relationship set and the entity set.

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

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

An account can be owned by at most **one** customer.

An account can be owned by **more than one** customers.



1. E-R Diagram: Constraints

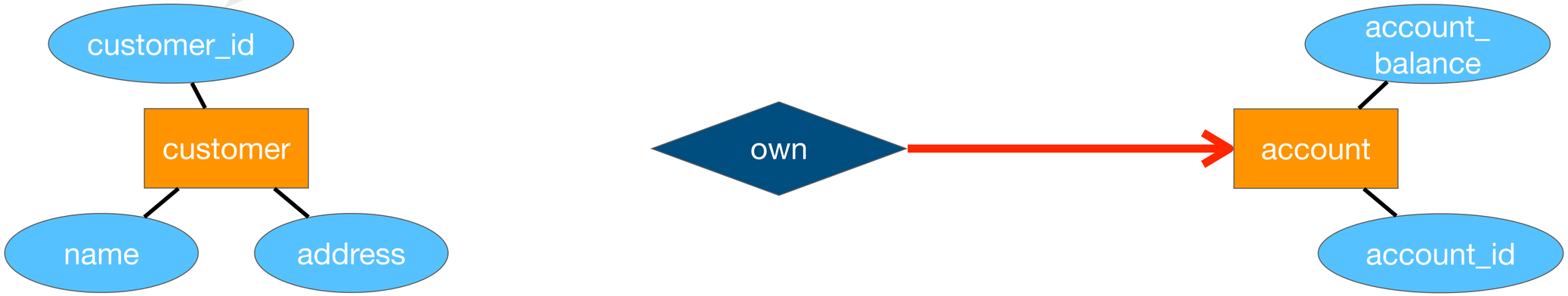
- In E-R Diagram: **Mapping cardinalities**
  - Draw a **directed line "->"**, signifying **one**, between the relationship set and the entity
  - Draw an **undirected line "-"**, signifying **many**, between the relationship set and the entity set.

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

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

An account can be owned by at most one customer.

An account can be owned by **more than one** customers.



1. E-R Diagram: Constraints

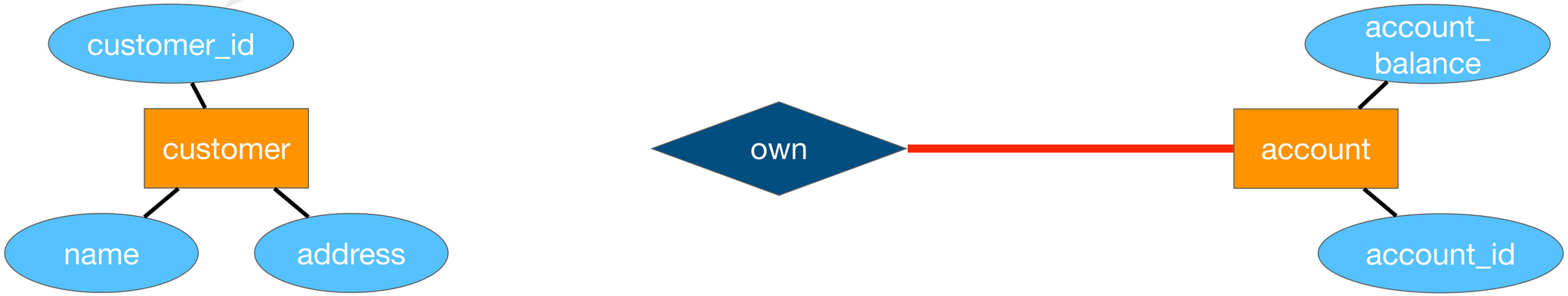
- In E-R Diagram: **Mapping cardinalities**
  - Draw a **directed line "->"**, signifying **one**, between the relationship set and the entity
  - Draw an **undirected line "-"**, signifying **many**, between the relationship set and the entity set.

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

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

An account can be owned by at most one customer.

An account can be owned by **more than one** customers.



1. E-R Diagram: Constraints

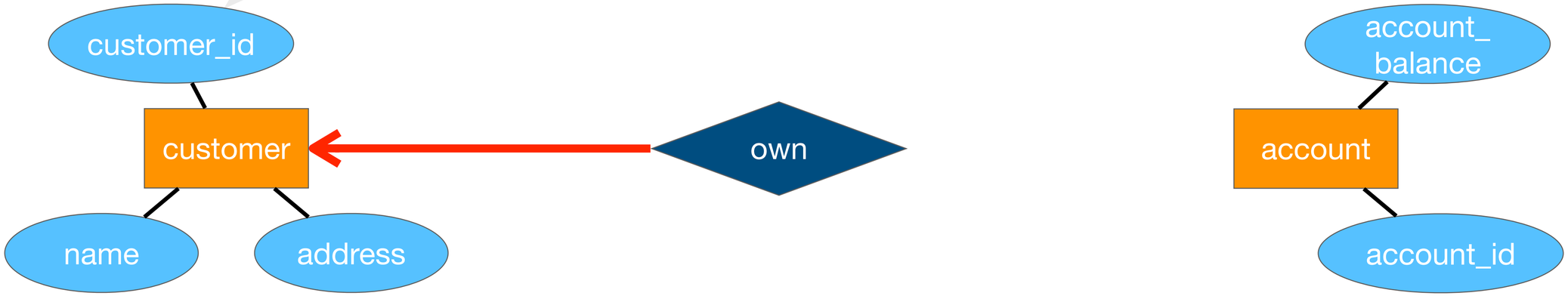
- In E-R Diagram: **Mapping cardinalities**
  - Draw a **directed line "->"**, signifying **one**, between the relationship set and the entity
  - Draw an **undirected line "-"**, signifying **many**, between the relationship set and the entity set.

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

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

An account can be owned by at most **one** customer.

An account can be owned by **more than one** customers.



1. E-R Diagram: Constraints

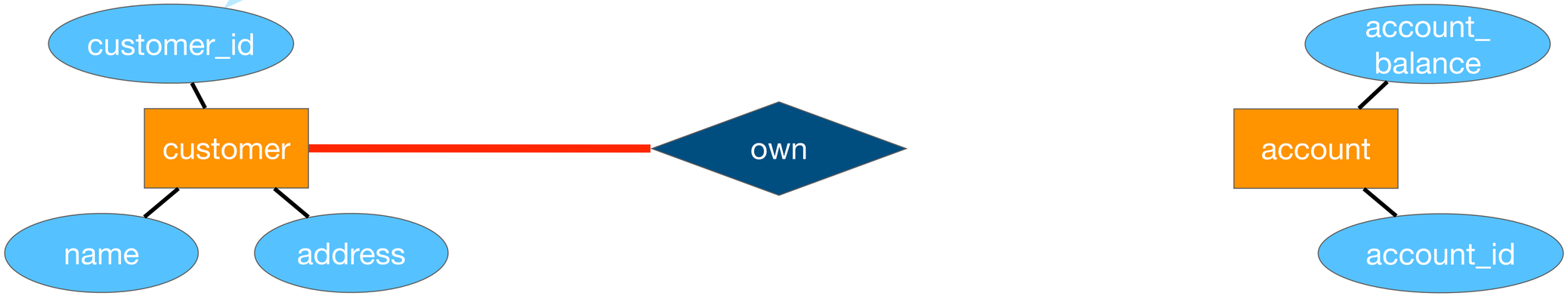
- In E-R Diagram: **Mapping cardinalities**
  - Draw a **directed line "->"**, signifying **one**, between the relationship set and the entity
  - Draw an **undirected line "-"**, signifying **many**, between the relationship set and the entity set.

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

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

An account can be owned by at most one customer.

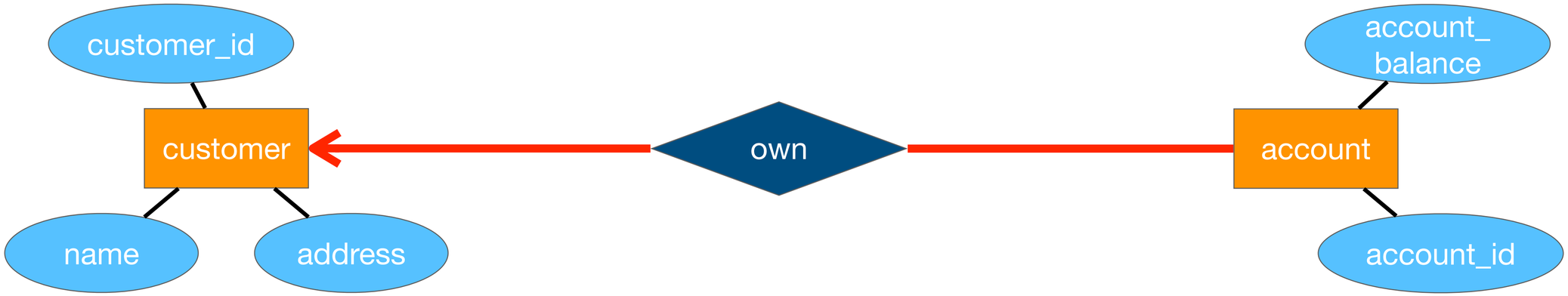
An account can be owned by **more than one** customers.



1. E-R Diagram: Constraints

- In E-R Diagram: **Mapping cardinalities**
  - Draw a **directed line "->"**, signifying **one**, between the relationship set and the entity
  - Draw an **undirected line "-"**, signifying **many**, between the relationship set and the entity set.

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**. Each customer can have one or more accounts, and each account has to be owned by only one customer.

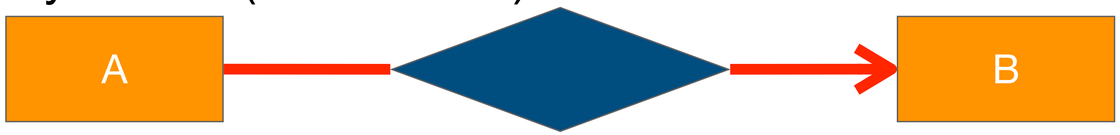


1. E-R Diagram: Constraints

- In E-R Diagram: Mapping cardinalities
  - Four mapping relationships
    - many to many

    - one to many (from A to B)

    - many to one (from A to B)



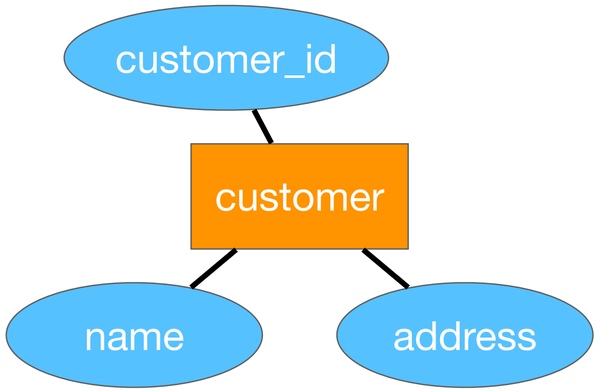
    - one to one (from A to B)

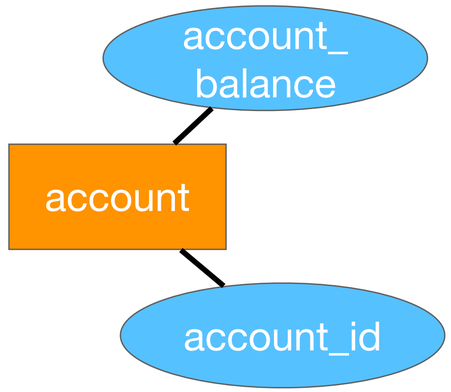
1. E-R Diagram: Constraints

For each customer, how many accounts he/she can have? One or more than one?

Whether a customer must have an account, or there can be some customer without any accounts?

- **Mapping cardinalities**
  - Concerns the number of entities to which another entities can be associated via a relationship set
- **Participation constraints**
  - Concerns whether all entities in the entity set have to participate in the relationship set





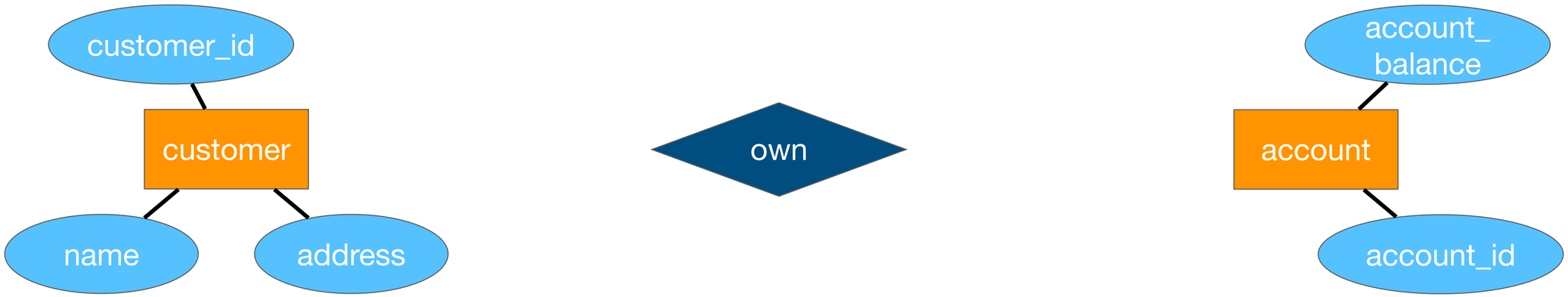
1. E-R Diagram: Constraints

- In E-R Diagram: **Participation constraints**
  - Draw a **double line "="**, signifying **total participation**
  - Draw a **single line "-"**, signifying **partial participation**

Each account **must** be owned by customers.

Each customer **must have an account**.

Not all customers are required to have an account



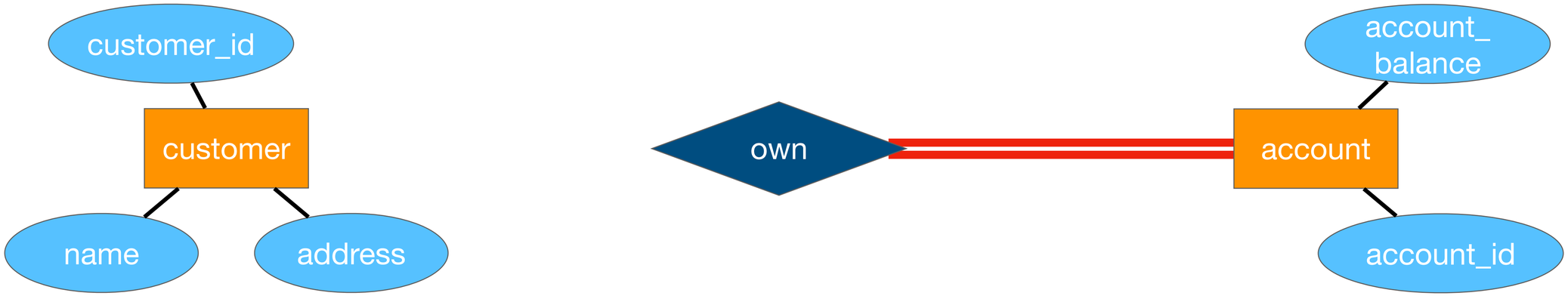
1. E-R Diagram: Constraints

- In E-R Diagram: **Participation constraints**
  - Draw a **double line "="**, signifying **total participation**
  - Draw a **single line "-"**, signifying **partial participation**

Each account **must** be owned by customers.

Each customer **must have an account**.

Not all customers are required to have an account



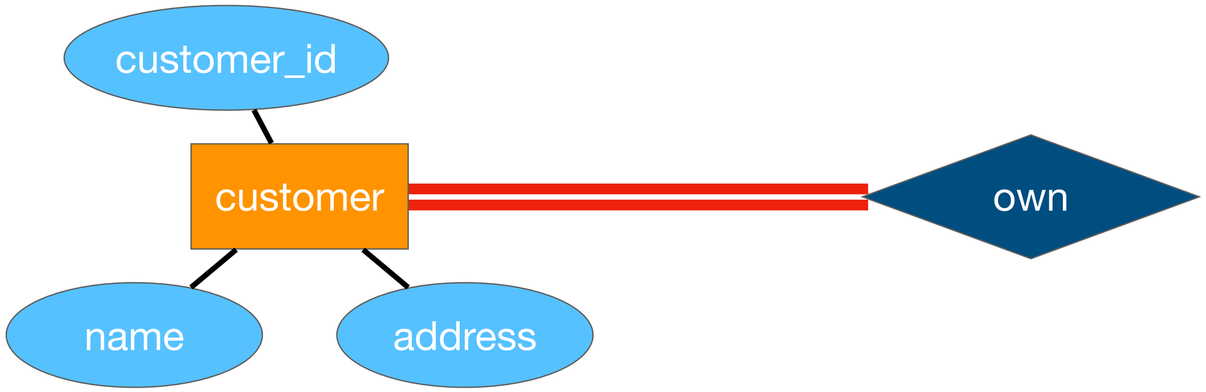
1. E-R Diagram: Constraints

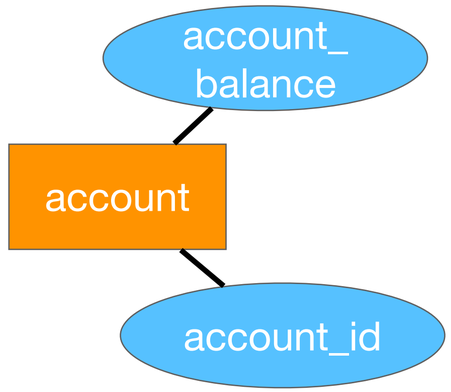
- In E-R Diagram: **Participation constraints**
  - Draw a **double line "="**, signifying **total participation**
  - Draw a **single line "-"**, signifying **partial participation**

Each account **must** be owned by customers.

Each customer **must have an account**.

Not all customers are required to have an account





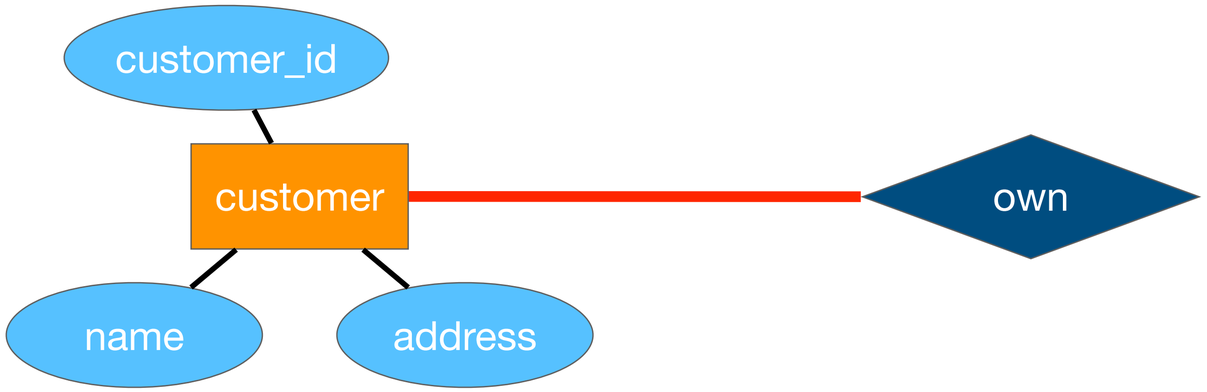
1. E-R Diagram: Constraints

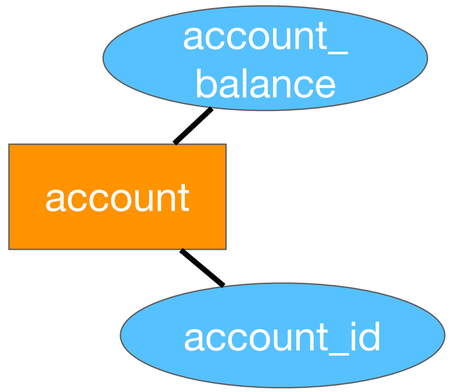
- In E-R Diagram: **Participation constraints**
  - Draw a **double line "="**, signifying **total participation**
  - Draw a **single line "-"**, signifying **partial participation**

Each account **must** be owned by customers.

Each customer **must have an account**.

➡ Not all customers are required to have an account





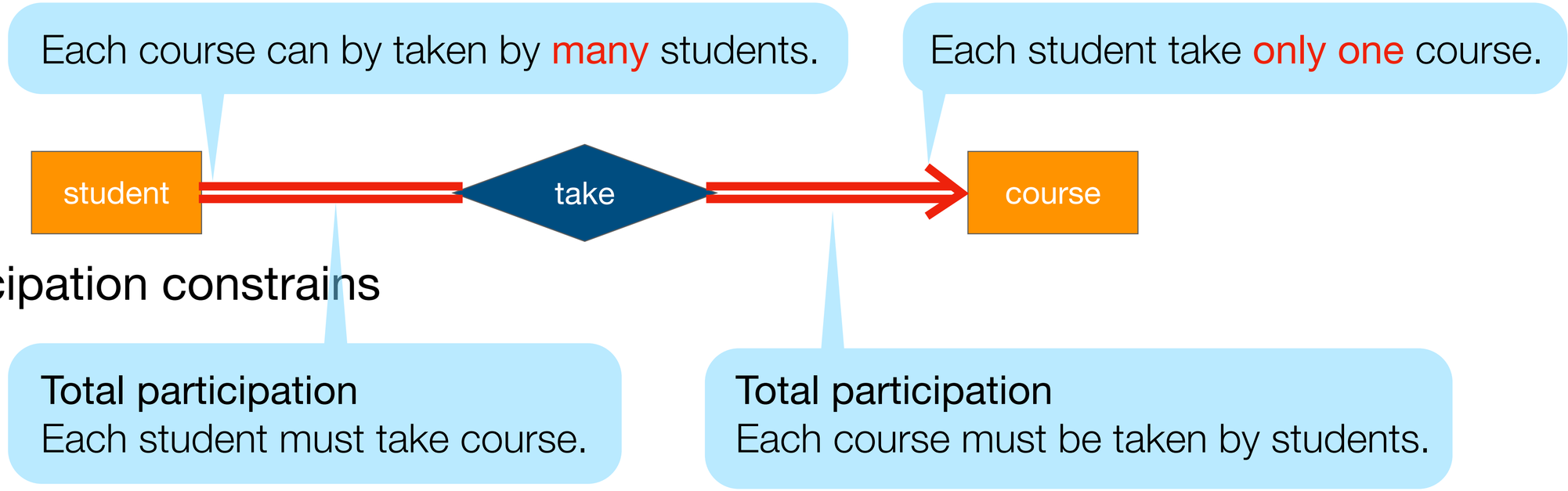
Recap

1. E-R Diagram: Constraints

- In E-R Diagram: Mapping cardinalities
  - Draw a directed line “->”, signifying one
  - Draw an undirected line “-”, signifying many
- In E-R Diagram: Participation constraints
  - Draw a double line “=”, signifying total participation
  - Draw a single line “-”, signifying partial participation

1. E-R Diagram: Constraints (Exercise)

- Mapping cardinality



- Participation constrains

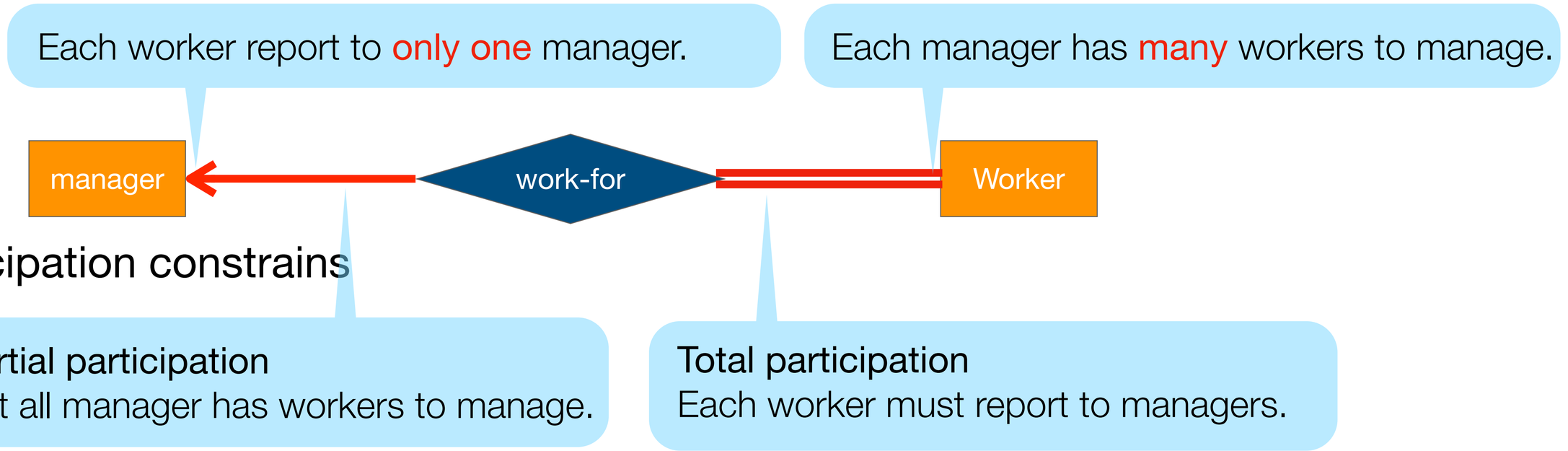
Total participation
Each student must take course.

Total participation
Each course must be taken by students.

1. E-R Diagram: Constraints (Exercise)

Each manager can have no, one or more workers, and each worker has and must to report to only one manager.

- Mapping cardinality



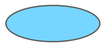
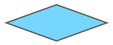
- Participation constrains

Partial participation
Not all manager has workers to manage.

Total participation
Each worker must report to managers.

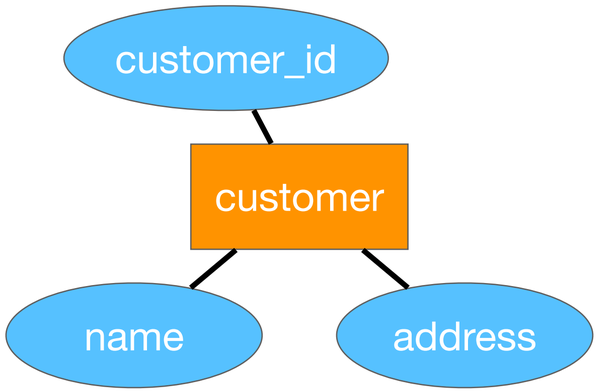
1. E-R Diagram

- Entity and entity set
- Relationship and relationship set
- Constraints
- Keys
- Weak entity set
- Role
- Specialization
- Different attribute types
- E-R design decision

- Entity and entity set: *rectangle*  , Attribute: *ellipse* 
- Relationship and relationship set: *diamond* 
- Constraints
  - Mapping cardinalities:
    - One: *directed line* 
    - Many: *undirected line* 
  - Participation constraints
    - Total: *double line*
    - Partial: *single line* 

1. E-R Diagram: Keys

- How to determine an entity by attributes?
  - Yi; 111111; CB306, HKU …
  - Alice; 222222; CYC301, HKU …
  - …

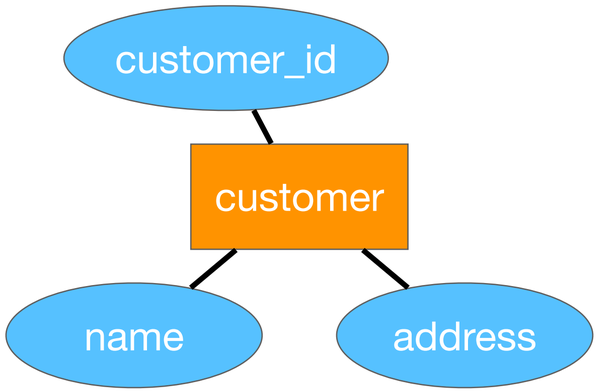


➡

- Keys
  1. Super key
  2. Candidate key
  3. Primary key

1. E-R Diagram: Keys

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



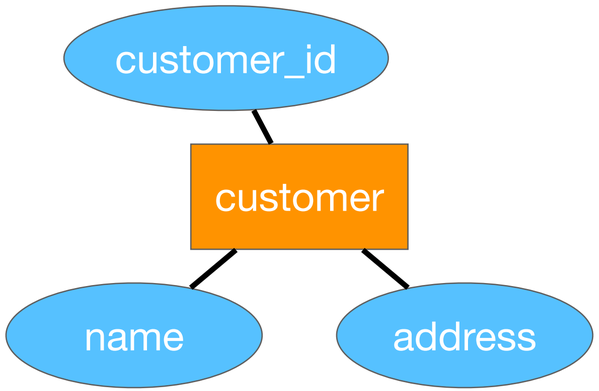
There are redundant attributes.

*If each customer has a **unique** customer_id, then*

- Super key
  - {customer_id}
  - {customer_id, name}
  - {customer_id, address}
  - {customer_id, name, address}

1. E-R Diagram: Keys

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

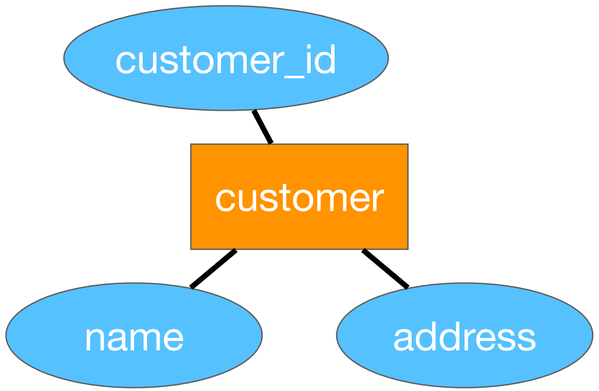


*If each customer has a **unique** customer_id, then*

- Super key
  - {customer_id} ⬅ Candidate key
  - {customer_id, name}
  - {customer_id, address}
  - {customer_id, name, address}

1. E-R Diagram: Keys

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



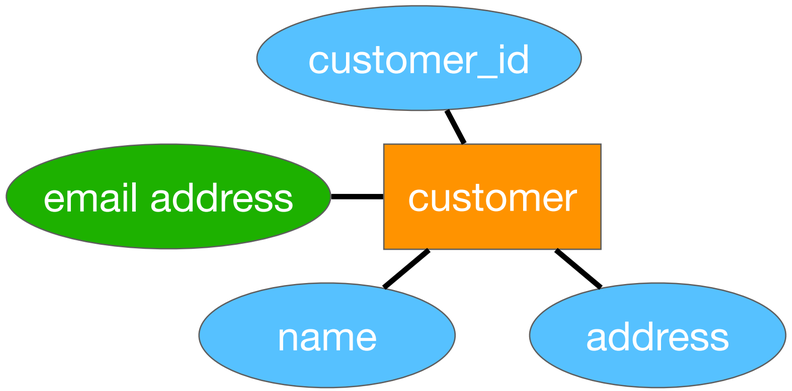
Question: Can it be more than one candidate key?

*If each customer has a **unique** customer_id, then*

- Super key
  - {customer_id} ⬅ Candidate key
  - {customer_id, name}
  - {customer_id, address}
  - {customer_id, name, address}

1. E-R Diagram: Keys

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



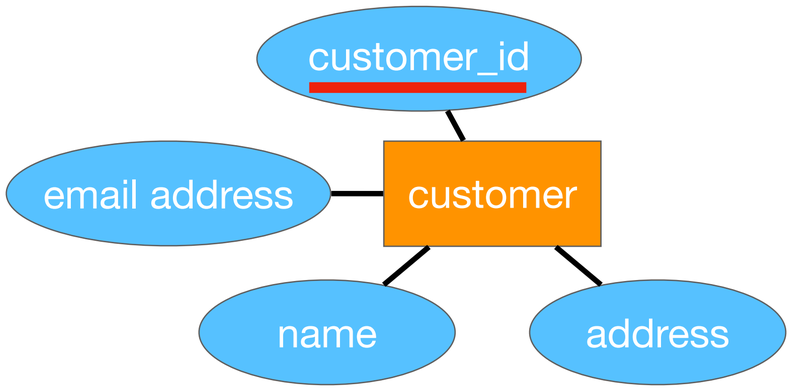
Question: Can it be more than one candidate key?

*If each customer has a **unique** customer_id, then*

- Super key
  - {customer_id} ⬅ Candidate key
  - {customer_id, name}
  - {customer_id, address}
  - {customer_id, name, address}

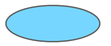
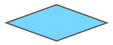
1. E-R Diagram: Keys

3. Primary key
   - one of the candidate keys is selected to be the primary key
   - In E-R Diagram
     - **Underline "-" the attribute**, signifying the **primary key** of an entity

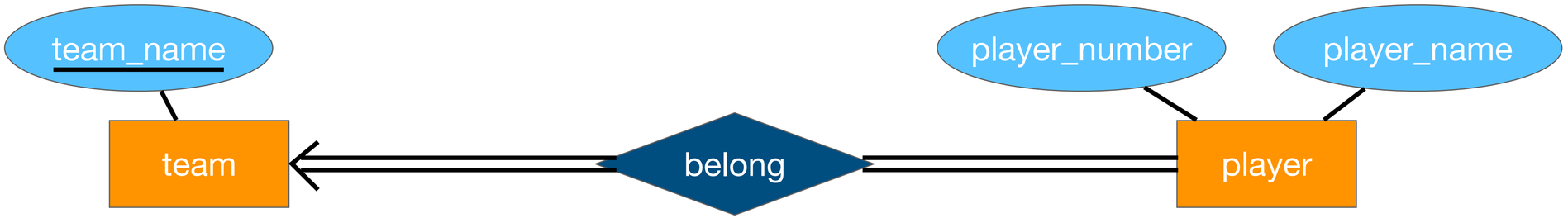


1. E-R Diagram

- Entity and entity set
- Relationship and relationship set
- Constraints
- Keys
- Weak entity set
- Role
- Specialization
- Different attribute types
- E-R design decision

- Entity and entity set: *rectangle*  , Attribute: *ellipse* 
- Relationship and relationship set: *diamond* 
- Constraints
  - Mapping cardinalities:
    - One: *directed line*
    - Many: *undirected line*
  - Participation constraints
    - Total: *double line*
    - Partial: *single line*
- Keys - primary key: *underline the attribute*

1. E-R Diagram: Weak Entity Set

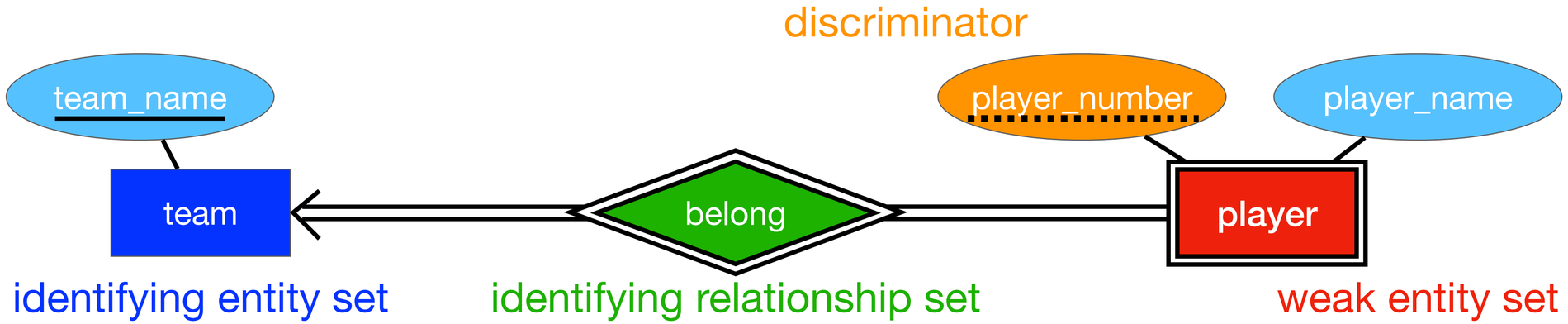


- Entity and entity set: *rectangle* , Attribute: *ellipse*
- Relationship and relationship set: *diamond*
- Constraints
  - Mapping cardinalities:
    - One: *directed line* →
    - Many: *undirected line* —
  - Participation constraints
    - Total: *double line* ═
    - Partial: *single line* —
- Keys - primary key: *underline the attribute*

How to identify a player? There is no attribute can uniquely identify player.

We have this relationship: each player_number belongs to only one player in a unique team. So, we can use the team_name and player_number to uniquely identify a player.

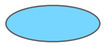
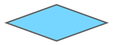
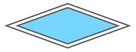
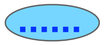
1. E-R Diagram: Weak Entity Set



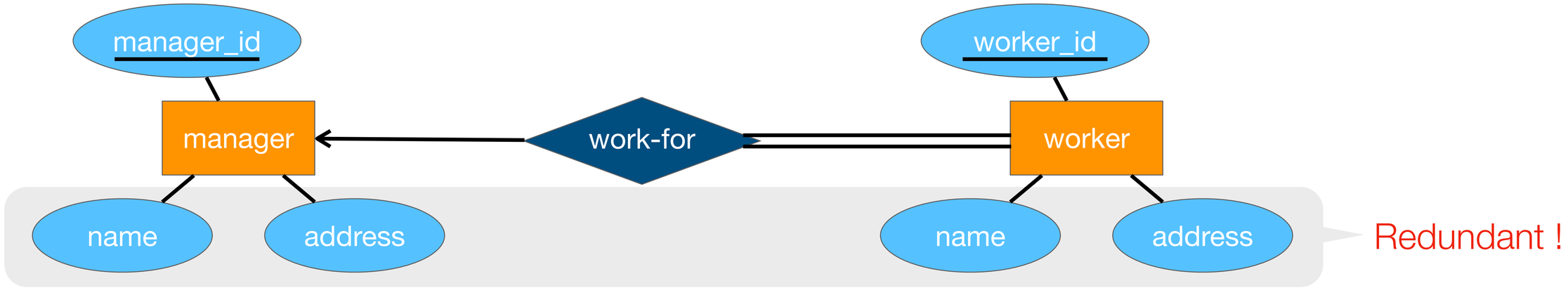
- The primary key of a weak entity set is formed by the primary key of the identifying entity set plus the weak entity set's discriminator.
- In E-R diagram
  - Weak entity set - draw double rectangle
  - Identifying relationship set - draw double diamond
  - Discriminator - draw dashed line
- The weak 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. **[WHY]**

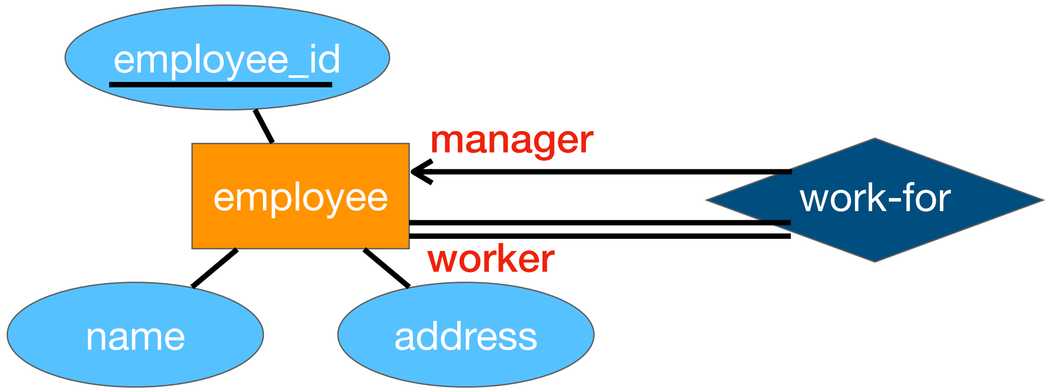
1. E-R Diagram

- Entity and entity set
- Relationship and relationship set
- Constraints
- Keys
- Weak entity set
- Role
- Specialization
- Different attribute types
- E-R design decision

- Entity and entity set: *rectangle*  , Attribute: *ellipse* 
- Relationship and relationship set: *diamond* 
- Constraints
  - Mapping cardinalities:
    - One: *directed line*
    - Many: *undirected line*
  - Participation constraints
    - Total: *double line*
    - Partial: *single line*
- Keys - primary key: *underline the attribute* 
- Weak entity set
  - Weak entity set: double rectangle 
  - Identifying relationship set: double diamond 
  - Discriminator: dashed line 

1. E-R Diagram: Role





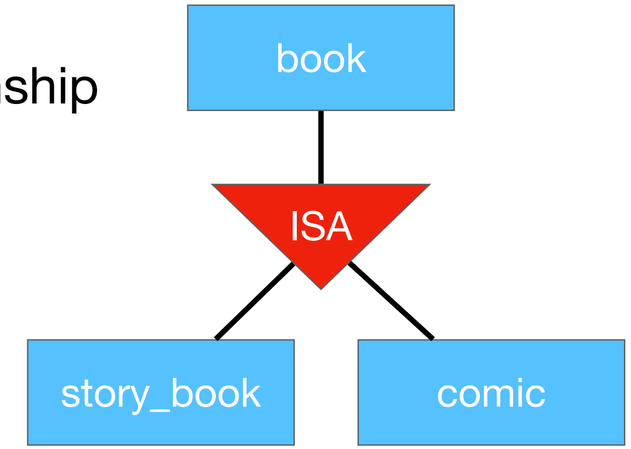
- **Role**
  - In E-R Diagram: The label “manager” and “worker” are called **roles**. They specify how employee entities interact via the “works-for” relationship set.
- **Mapping cardinality**
  - Each employee (manager) can have many workers working for him/her.
  - Each employee (worker) works for only one manager.
- **Participation constraints**
  - Not all employee (manager) has workers working for him/her.
  - Each employee (worker) must work for an employee.

1. E-R Diagram: Specialization

A book may specialized to story book or comics.

- 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.

- In E-R Diagram
  - Draw triangle with IS A relationship



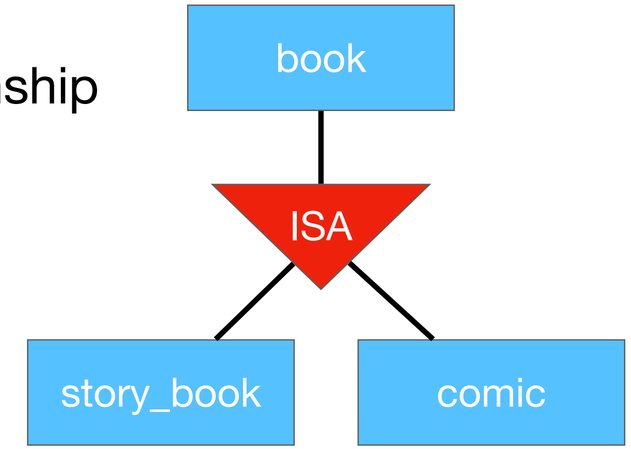
1. E-R Diagram: Specialization

A book may specialized to story book or comics.

Whether a book can be specialized to story book, comics, or none of them?

Whether a book can be both a story book and comic?

- 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.
- In E-R Diagram
  - Draw triangle with IS A relationship



1. E-R Diagram: Specialization

A book may specialized to story book or comics.

- Constraints

Whether a book can be specialized to story book, comics, or none of them?

➡

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

Whether a book can be both a story book and comic?

➡

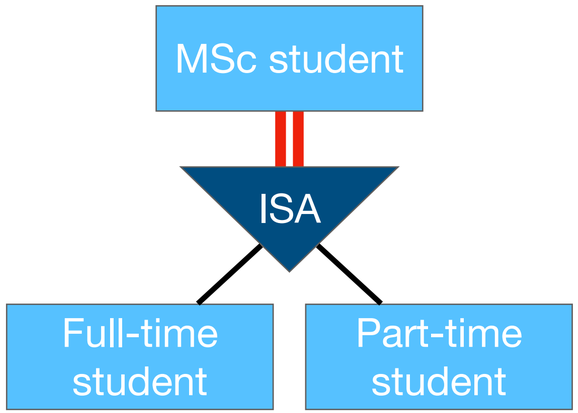
- Disjoint or overlapping specialization
  - *Whether entities may belong to more than one lower-level entity set within a single specialization.*

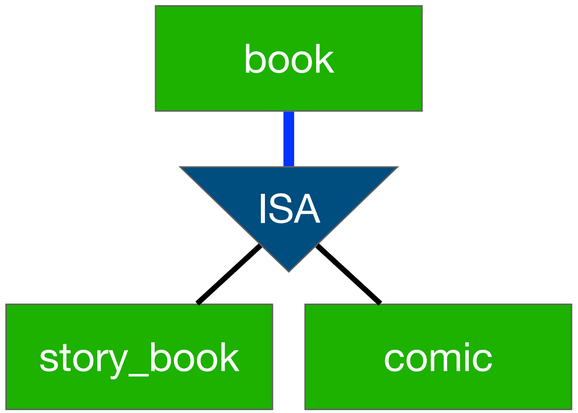
1. E-R Diagram: Specialization

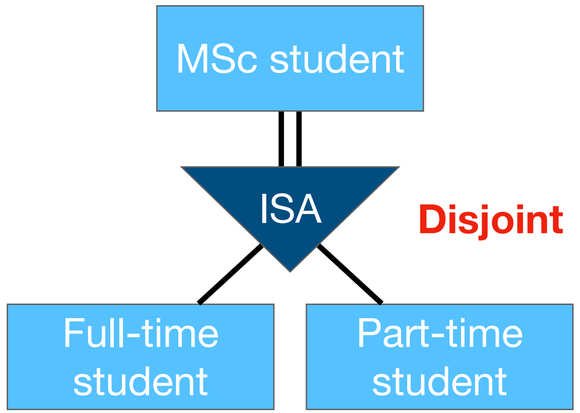
- In E-R diagram
  - Draw a **double line "="**, signifying **total specialization**
  - Draw a **single line "-"**, signifying **partial specialization**
  - Write a keyword **"Disjoint",** signifying **disjoint**
  - No specifying anything, signifying **overlapping specialization**

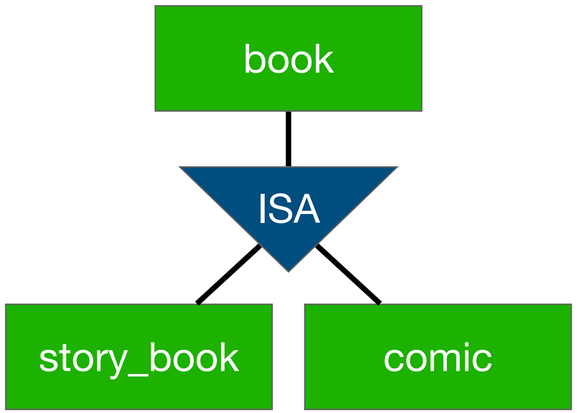
Whether a book can be specialized to story book, comics, or none of them?

Whether a book can be both a story book and comic?



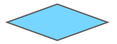
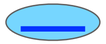
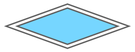
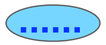






1. E-R Diagram

- Entity and entity set
- Relationship and relationship set
- Constraints
- Keys
- Weak entity set
- Role
- Specialization
- Different attribute types
- E-R design decision

- Entity and entity set: *rectangle*  , Attribute: *ellipse* 
- Relationship and relationship set: *diamond* 
- Constraints
  - Mapping cardinalities:
    - One: *directed line* 
    - Many: *undirected line*
  - Participation constraints
    - Total: *double line* 
    - Partial: *single line*
- Keys - primary key: *underline the attribute* 
- Weak entity set
  - Weak entity set: double rectangle 
  - Identifying relationship set: double diamond 
  - Discriminator: dashed line 
- Role: *write word on line*
- Specialization: *ISA triangle* 

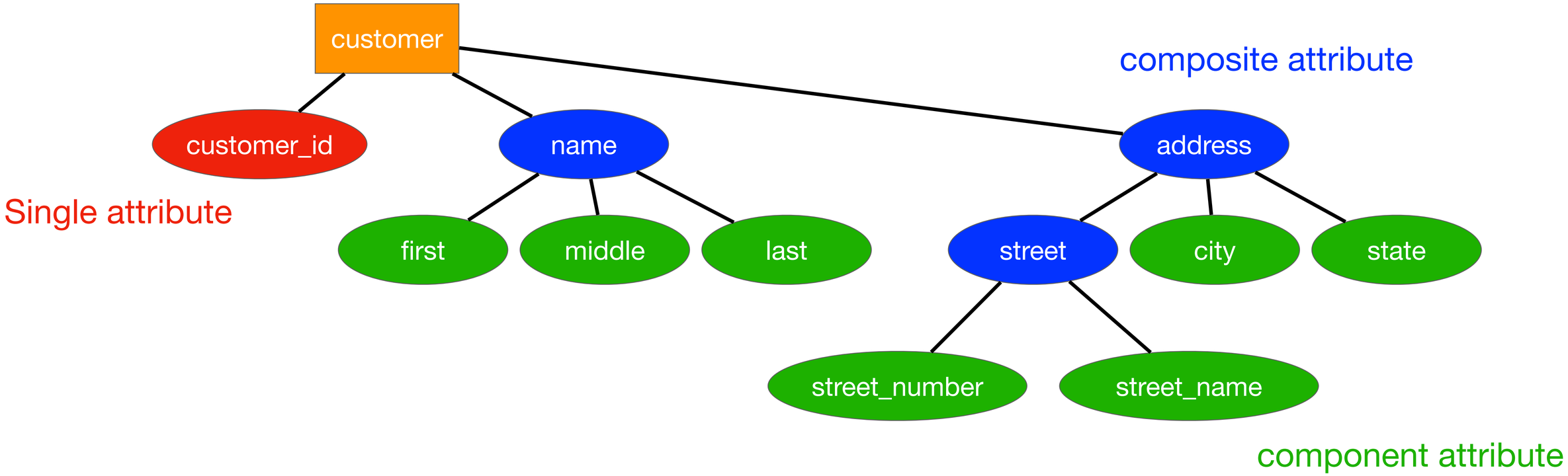
  - Total specialization: *double line* 
  - Partial specialization: *single line*
  - Disjoint specialization: *write a keyword “disjoint”*
  - Overlapping specialization: *write nothing*

1. E-R Diagram: Different attribute type

1. Single attribute v.s. Composite attributes
2. Single-valued attribute v.s. Multi-valued attribute
3. Derived attribute

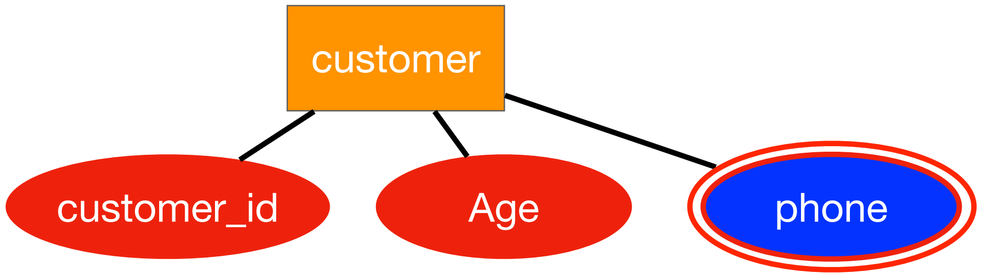
1. E-R Diagram: Different attribute type

1. Single attribute v.s. Composite attributes



1. E-R Diagram: Different attribute type

2. Single-valued attribute v.s. Multi-valued attribute



Single-valued attribute

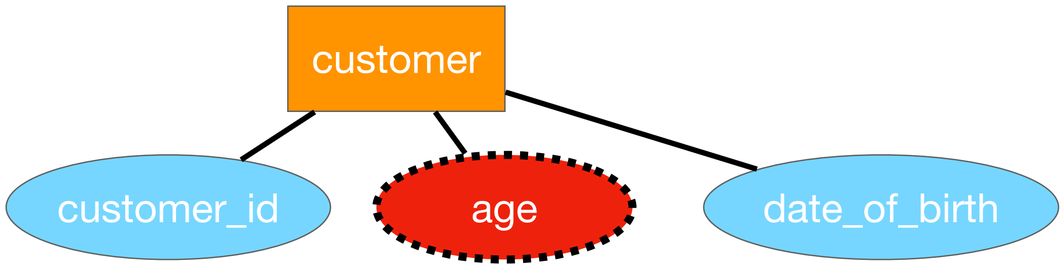
Multi-valued attribute

Customer_id	Name	Phone
111111	Yi	28597070, 65859999, 65880000
222222	Alice	28597071, 24761111

- In E-R Diagram
  - Draw **double ellipses**, signifying the **multi-valued attributes**

1. E-R Diagram: Different attribute type

3. Derived attribute
   - Value in this attribute can be derived from other attributes



Derived attribute

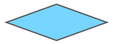
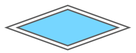
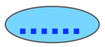
Since “age” can be derived from the “date_of_birth”, “age” is a derived attribute.

   - In E-R Diagram
     - Draw **dashed ellipses**, signifying the **derived attributes**

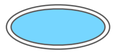
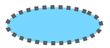
1. E-R Diagram

- Entity and entity set
- Relationship and relationship set
- Constraints
- Keys
- Weak entity set
- Role
- Specialization
- Different attribute types
- E-R design decision

1. E-R Diagram

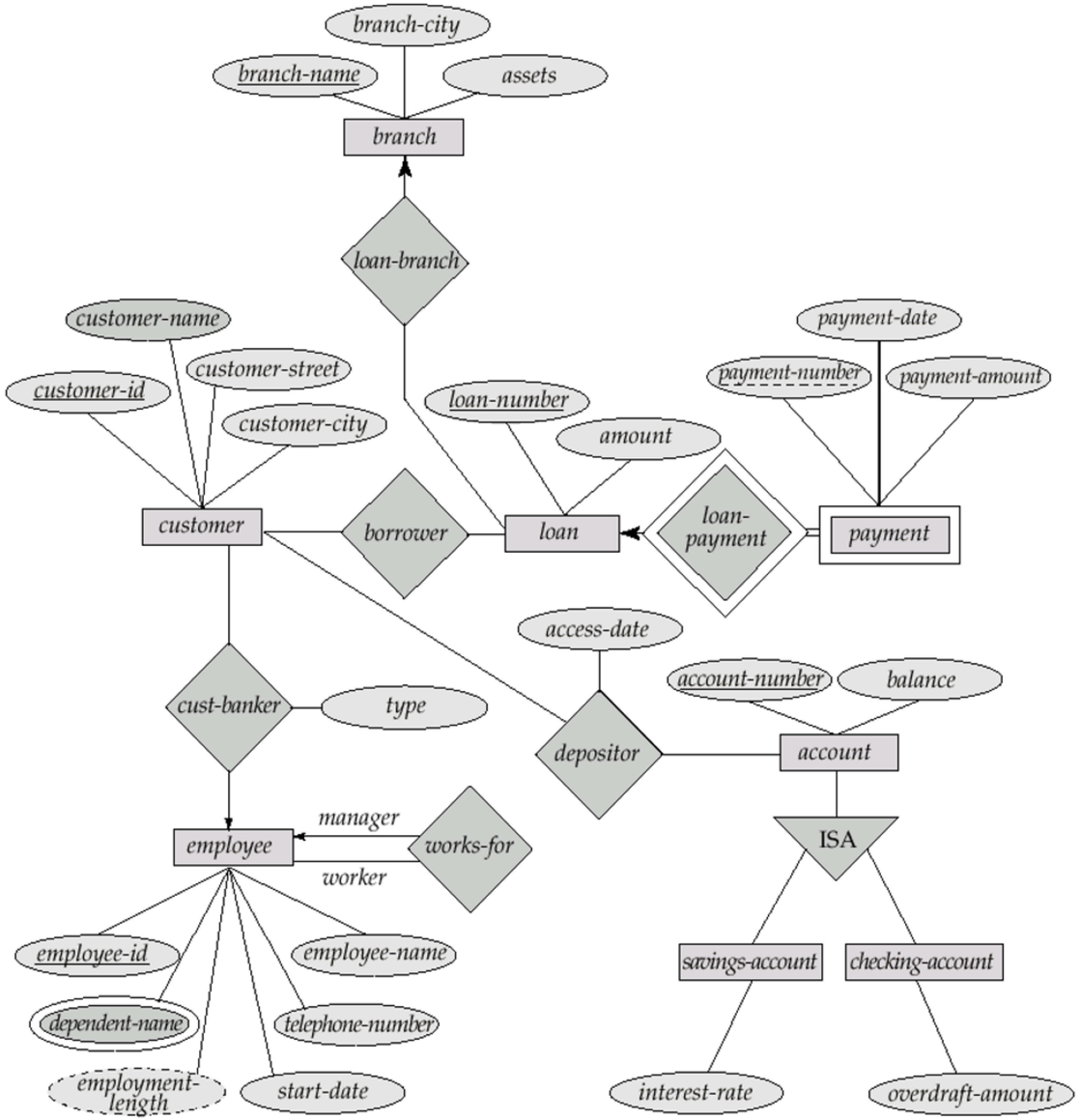
- Entity and entity set: *rectangle*  , Attribute: *ellipse* 
- Relationship and relationship set: *diamond* 
- Constraints
  - Mapping cardinalities:
    - One: *directed line* 
    - Many: *undirected line*
  - Participation constraints
    - Total: *double line* 
    - Partial: *single line*
- Keys - primary key: *underline the attribute* 
- Weak entity set
  - Weak entity set: double rectangle 
  - Identifying relationship set: double diamond 
  - Discriminator: dashed line 

- Role: *write word on line*
- Specialization: *ISA triangle* 

  - Total specialization: *double line*
  - Partial specialization: *single line*
  - Disjoint specialization: *write a keyword “disjoint”*
  - Overlapping specialization: *write nothing*
- Different attribute types
  - Multi-valued attribute: *double ellipse* 
  - Derived-value attribute: *dashed ellipse* 

1. E-R Diagram: Exercise

- Entity and entity set: *rectangle* , Attribute: *ellipse*
- Relationship and relationship set: *diamond*
- Constraints
  - Mapping cardinalities:
    - One: *directed line* →
    - Many: *undirected line* —
  - Participation constraints
    - Total: *double line* ═
    - Partial: *single line* —
- Keys - primary key: *underline the attribute*
- Weak entity set
  - Weak entity set: double rectangle
  - Identifying relationship set: double diamond
  - Discriminator: dashed line
- Role: *write word on line*
- Specialization: *ISA triangle* ISA
  - Total specialization: *double line* ═
  - Partial specialization: *single line* —
  - Disjoint specialization: *write a keyword "disjoint"*
  - Overlapping specialization: *write nothing*
- Different attribute types
  - Multi-valued attribute: *double ellipse*
  - Derived-value attribute: *dashed ellipse*



1. E-R Diagram

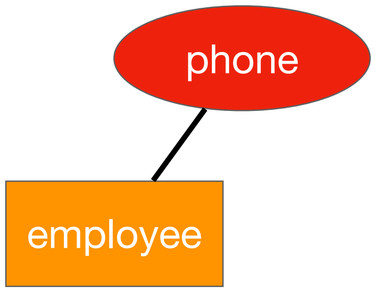
- Entity and entity set
- Relationship and relationship set
- Constraints
- Keys
- Weak entity set
- Role
- Specialization
- Different attribute types
- E-R design decision

1. E-R Diagram: E-R design decision

- Question: Data -> Entity sets? Attributes? Relationship sets?
  1. Entity sets v.s. Attributes (Example)
  2. Entity sets v.s. Relationship sets (Example)

1. E-R Diagram: E-R design decision

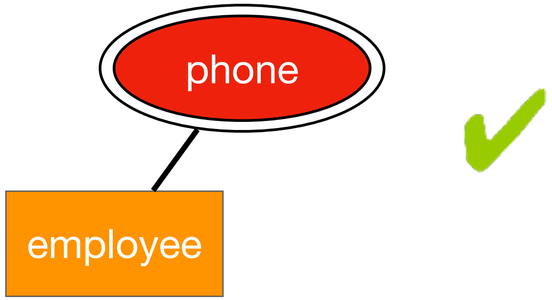
1. Entity sets v.s. Attributes (Example)
   - How do you model an employee and his phone number?
     1. *Treat phone number as an attribute of an employee*
     2. *Treat phone as a separate entity*



1. E-R Diagram: E-R design decision

1. Entity sets v.s. Attributes (Example)
   - How do you model an employee and his phone number?
     1. *Treat phone number as an attribute of an employee*
     2. *Treat phone as a separate entity*

1. In a company, an employee can have multiple phone numbers

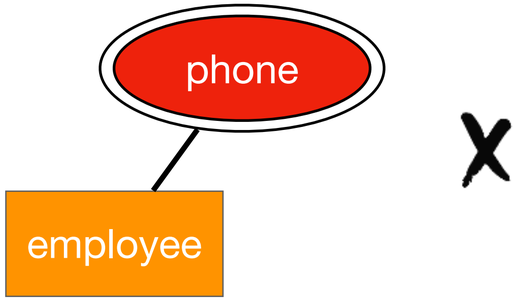


1. E-R Diagram: E-R design decision

1. Entity sets v.s. Attributes (Example)
   - How do you model an employee and his phone number?
     1. *Treat phone number as an attribute of an employee*
     2. *Treat phone as a separate entity*

1. In a company, an employee can have multiple phone numbers

2. In a company, a phone number can be shared by multiple employee …



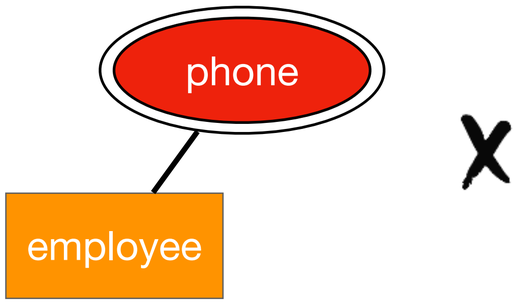
✗

✔

1. E-R Diagram: E-R design decision

1. Entity sets v.s. Attributes (Example)
   - How do you model an employee and his phone number?
     1. *Treat phone number as an attribute of an employee*
     2. *Treat phone as a separate entity*

1. In a company, an employee can have multiple phone numbers

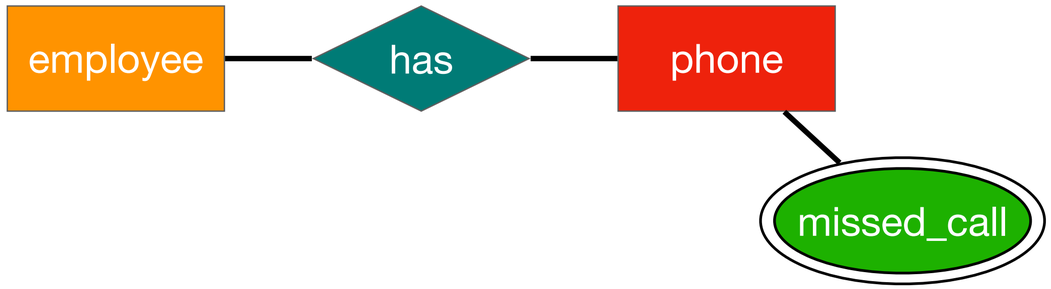


✗

2. In a company, a phone number can be shared by multiple employee …

3. For each phone, I want to keep a list of missed call numbers.

✔



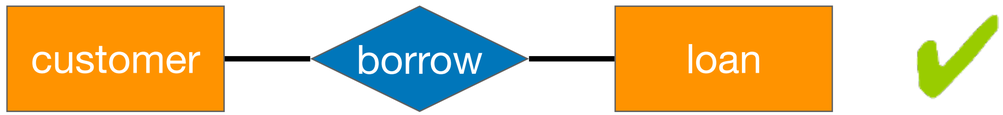
1. E-R Diagram: E-R design decision

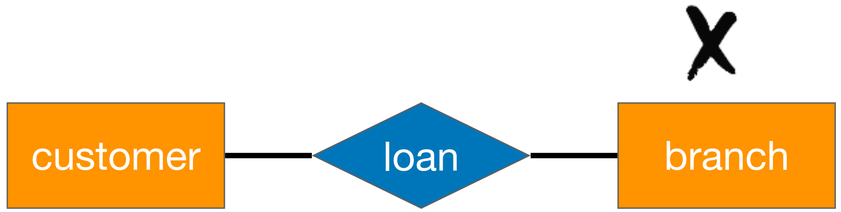
1. Entity sets v.s. Relationship sets (Example)
   - How do you model a loan?
     1. *Treat a Loan as entity.*
     2. *Treat a Loan an a relationship between a customer and a branch.*

1. E-R Diagram: E-R design decision

1. Entity sets v.s. Relationship sets (Example)
   - How do you model a loan?
     1. *Treat a Loan as entity.*
     2. *Treat a Loan an a relationship between a customer and a branch.*

1. Can we have joint loan? E.g., A loan can be associated with multiple customers.





1. E-R Diagram: E-R design decision

1. Entity sets v.s. Relationship sets (Example)
   - Use a relationship set to describe an action that occurs between entities
     - Hint:
       - Entity sets often have “nouns” as name
       - Relationship sets often have “verbs” as name

1. E-R Diagram

- Entity and entity set
- Relationship and relationship set
- Constraints
- Keys
- Weak entity set
- Role
- Specialization
- Different attribute types
- E-R design decision

We are going to learn …

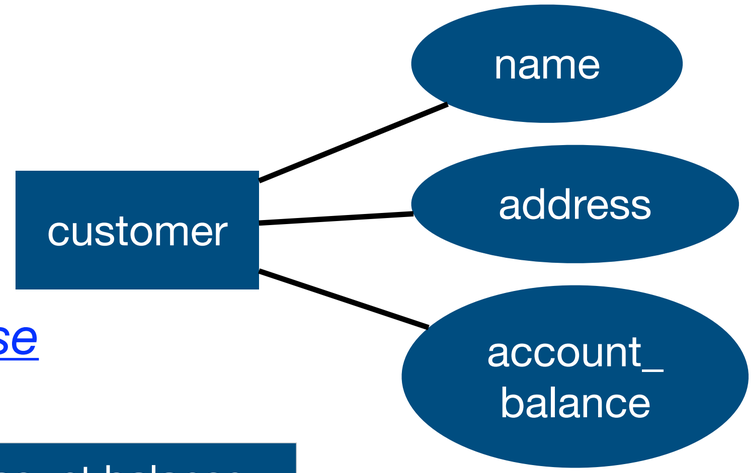
- Outcome 1: **Information Modeling**
  - Able to understand the modeling of real-life information in database systems.

1. Entity-Relation Diagram (E-R diagram)

   Visualizes the data structure and their internal relationships

2. E-R Diagram to Relational Tables

   Represents the actual format used to store data in the database



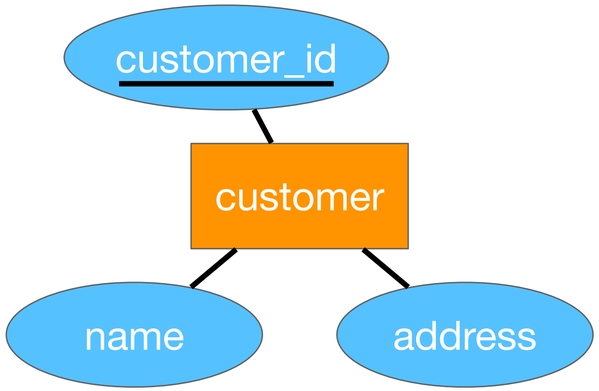
Name	Address	Account balance
Yi	CB306, HKU, …	$100

- Outcome 3: **System Design**
  - Able to design an efficient and reliable database system.

2. E-R Diagram to Relational Tables

1. Entity set
2. Weak entity set
3. Relationship set
4. Specialization

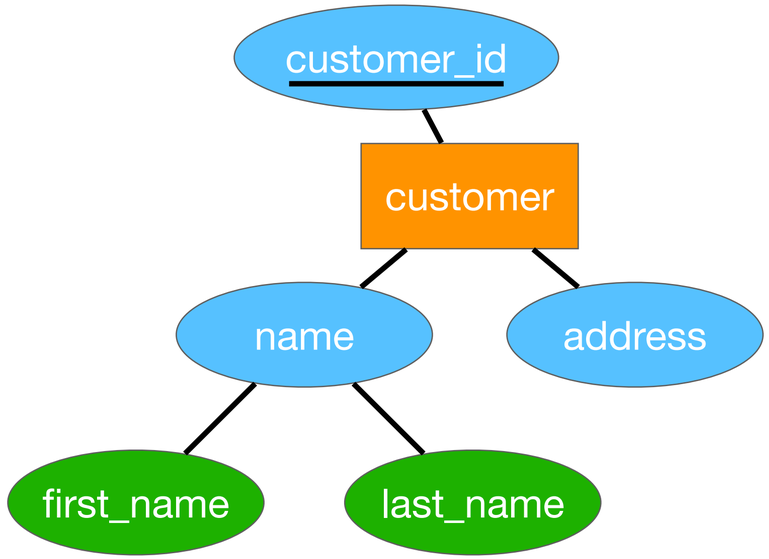
2. E-R Diagram to Relational Tables: Entity Set



- An entity set reduces to a table with the same attributes
  - **Schema**: customer (customer_id, name, address)
  - **Table**:

customer_id	name	address
…	…	…
…	…	…
…	…	…

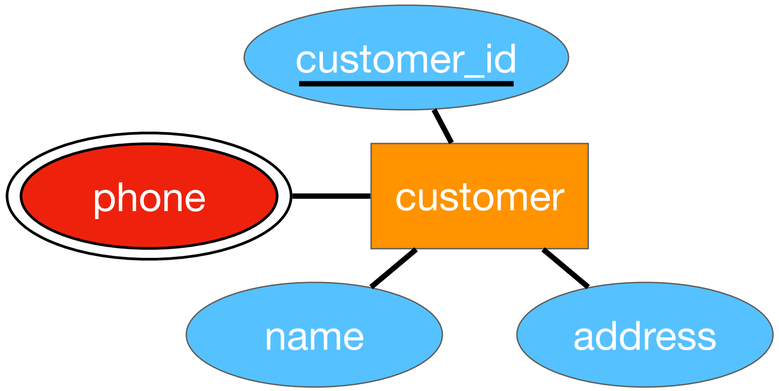
2. E-R Diagram to Relational Tables: Entity Set



- Composite attributes are flattened out by creating a separate attribute for each component attribute.
  - e.g., name -> name.first_name and name.last_name
- **Schema**: customer (customer_id, name.first_name, name.last_name, address)
- **Table**:

customer_id	name.first_name	name.last_name	address
…	…	…	…
…	…	…	…
…	…	…	…

2. E-R Diagram to Relational Tables: Entity Set



- A multi-valued attribute M of an entity set E is represented by a separate table EM, with the primary key of E as one of EM's attribute.

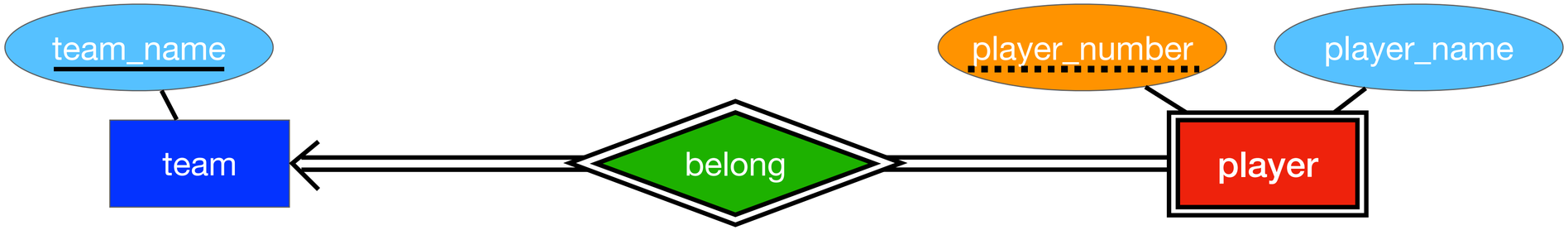
customer_id	name	address
1	Yi	…
2	Alice	…
…	…	…

customer (customer_id, name, address)

customer_id	phone
1	28597070
1	28597080
…	…

CustomerPhone (customer_id, phone)

2. E-R Diagram to Relational Tables: Weak Entity Set



- A weak entity set becomes a table that includes the columns for the primary key of the identifying strong entity set.

team_name	player_number	player_name
1	23	Alice
2	23	Bob
…	…	…

player (team_name, player_number, player_name)

Primary key of the identifying entity set

discriminator (partial key) of the weak entity set

2. E-R Diagram to Relational Tables

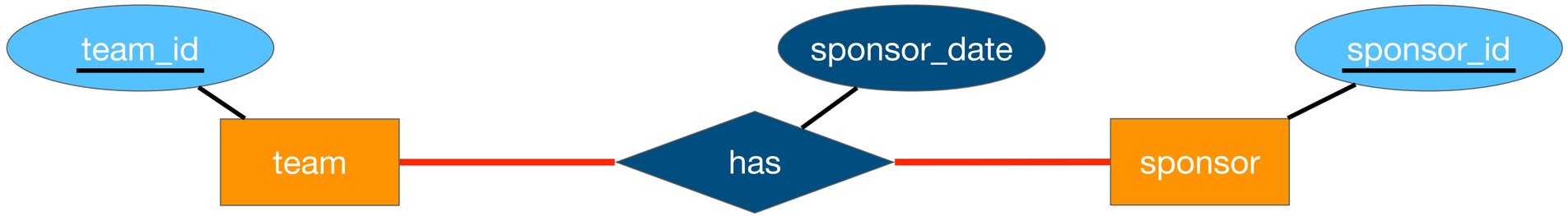
1. Entity set
2. Weak entity set
3. Relationship set
4. Specialization

2. E-R Diagram to Relational Tables: Relationship Set

- The reduction depends on mapping cardinalities
  - Many to many
  - One to many / many to one
  - One to one

2. E-R Diagram to Relational Tables: Relationship Set

- A many-to-many relationship set is a table with columns for the primary keys of the participating entity sets, and any attributes of the relationship set.



team_id	…
1	…
2	…
…	…

team (team_id, …)

team_id	sponsor_id	sponsor_date
1	1	2024-1-1
1	2	2024-5-1
2	1	2024-4-1

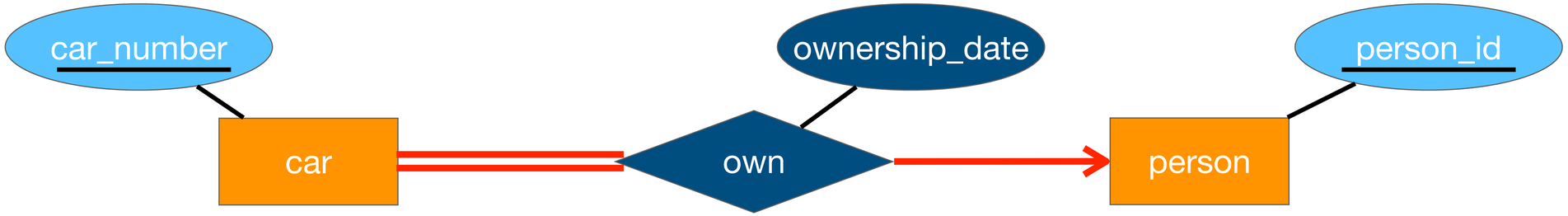
team_asoc_sponsor (team_id, sponsor_id, sponsor_date)

sponsor_id	…
1	…
2	…
…	…

sponsor (sponsor_id, …)

2. E-R Diagram to Relational Tables: Relationship Set

- **Many-to-one** and **one-to-many** relationship sets that are **total on the many-side** can be represented by adding extra attributes to the “many-side”, containing the primary key of the “one-side”.



car_number	…
1	…
2	…

car (car_number, …)

car_number	person_id	ownership_date
1	1	2024-1-1
2	1	2024-9-1

car_asoc_person (car_number, person_id, ownership_date)

person_id	…
1	…
2	…

person (person_id, …)

car_number	ownership_date	person_id	…
1	2024-1-1	1	…
2	2024-9-1	1	…

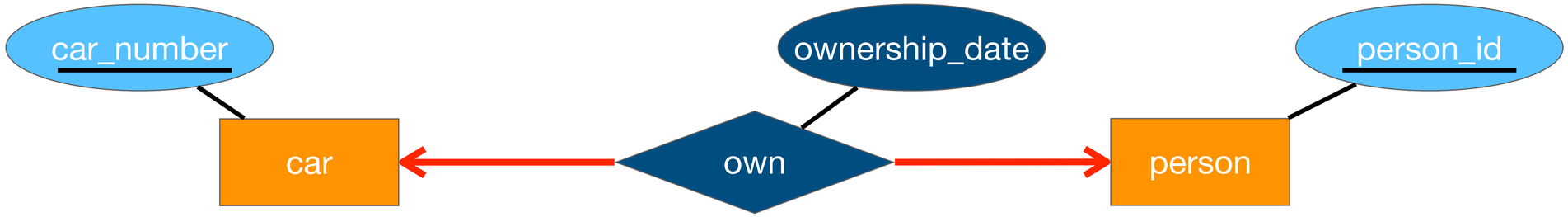
car (car_number, ownership_date, person_id, …)

person_id	…
1	…
2	…

person (person_id, …)

2. E-R Diagram to Relational Tables: Relationship Set

- For **one-to-one** relationship sets, either side can be chosen to act as the "many-side".



car_number	ownership_date	person_id	…
1	2024-1-1	2	…
2	2024-9-1	1	…

car (car_number, ownership_date, person_id, …)

person_id	…
1	…
2	…

person (person_id, …)

OR

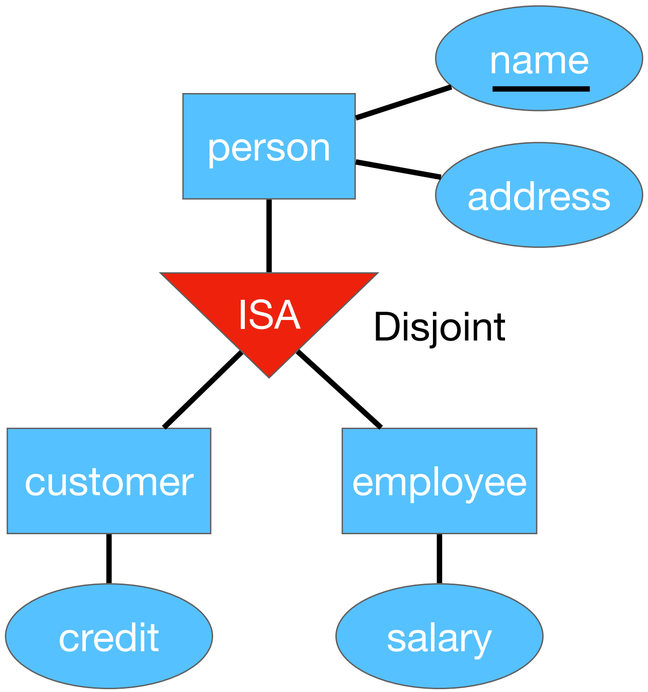
car_number	…
1	…
2	…

car (car_number, …)

person_id	ownership_date	car_number	…
1	2024-9-1	2	…
2	2024-1-1	1	…

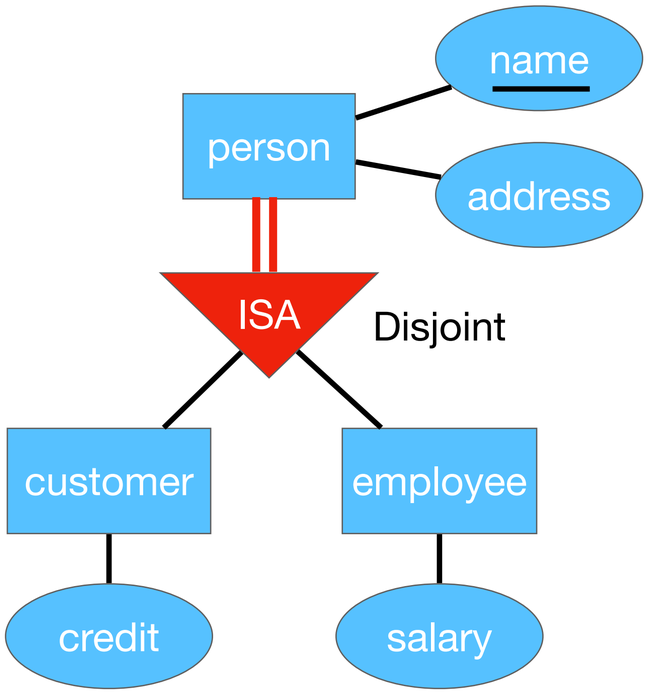
person (person_id ownership_date, car_number, …)

2. E-R Diagram to Relational Tables: Specialization



- Method 1: Form a table from the higher-level entity set; then form tables for each lower-level entity set, which contains the primary key of the higher-level entity set and local attributes.
  - person (name, address)
  - customer (name, credit)
  - employee (name, salary)
- Method 2: Form a table for each entity set with all local and inherited attributes.
  - person (name, address)
  - customer (name, address, credit)
  - employee (name, address, salary)
- Question: What are the advantage and disadvantage of method 1 and 2? Storage redundancy? Efficiency in retrieving data?

2. E-R Diagram to Relational Tables: Specialization



- If the specialization is total?
- Method 2: Form a table for each entity set with all local and inherited attributes.
  - ~~person (name, address)~~
  - customer (name, address, credit)
  - employee (name, address, salary)

We are going to learn …

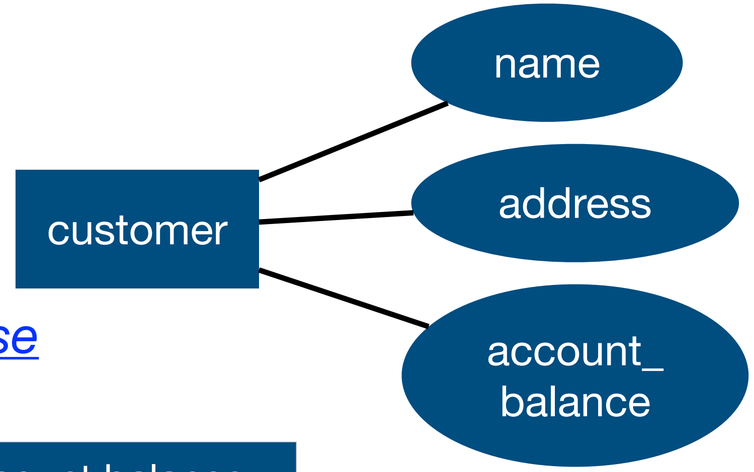
- Outcome 1: **Information Modeling**
  - Able to understand the modeling of real-life information in database systems.

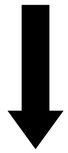
1. Entity-Relation Diagram (E-R diagram)

   Visualizes the data structure and their internal relationships

2. E-R Diagram to Relational Tables

   Represents the actual format used to store data in the database

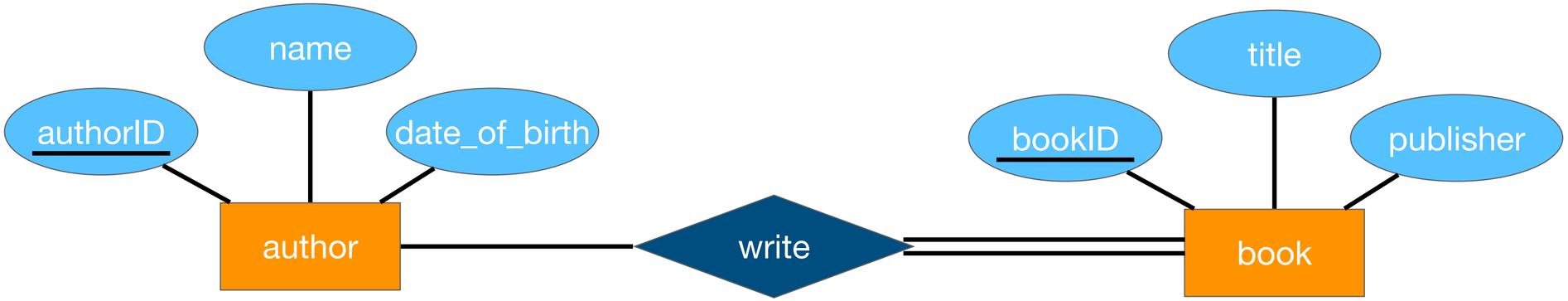




Name	Address	Account balance
Yi	CB306, HKU, …	$100

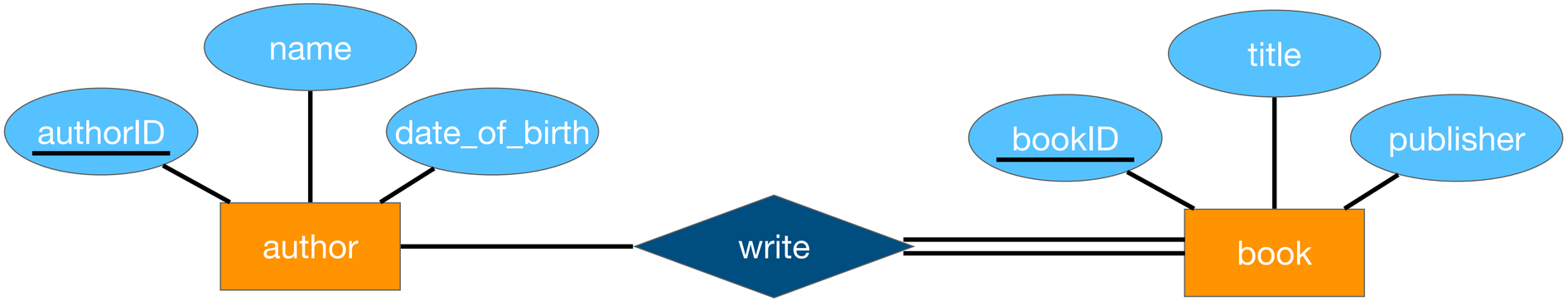
- Outcome 3: **System Design**
  - Able to design an efficient and reliable database system.

What's more? Foreign Key



- Step 1： entity set -> table
  - author (authorID, name, date_of_birth)
  - book (bookID, title, publisher)
- Step 2: relationship set
  - whether a relationship set becomes a table?
    - many to many -> table
  - write (authorID, bookID)

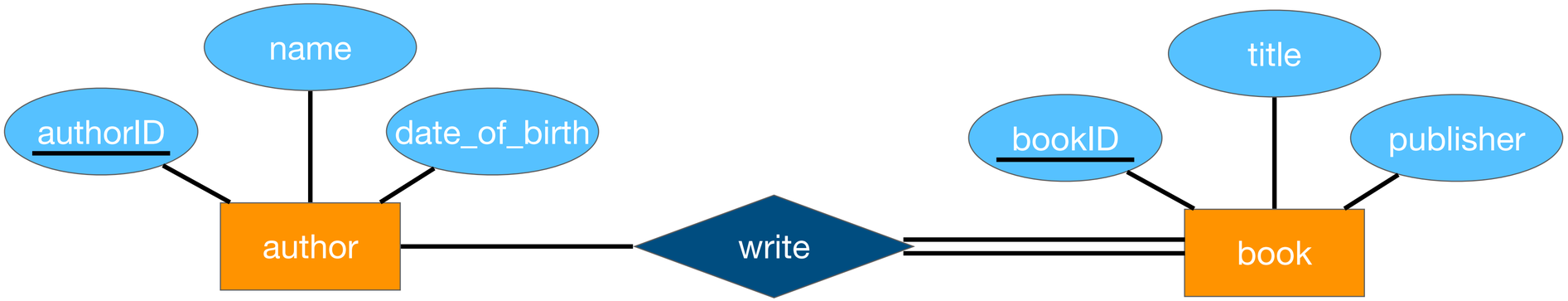
What's more? Foreign Key



- Step 1： entity set -> table
  - author (authorID, name, date_of_birth)
  - book (bookID, title, publisher)
- Step 2: relationship set
  - whether a relationship set becomes a table?
    - many to many -> table
  - write (authorID, bookID)

- **authorID** is a **Foreign key**, referencing the column authorID in the author table.
- **bookID** is another **Foreign key**, referencing the column bookID in the book table.

What's more? Foreign Key



- author (authorID, name, date_of_birth)
  - Foreign key: none
- book (bookID, title, publisher)
  - Foreign key: none
- write (authorID, bookID)
  - Foreign keys: {authorID} referencing author
    {bookID} referencing book

What's more? Foreign Key

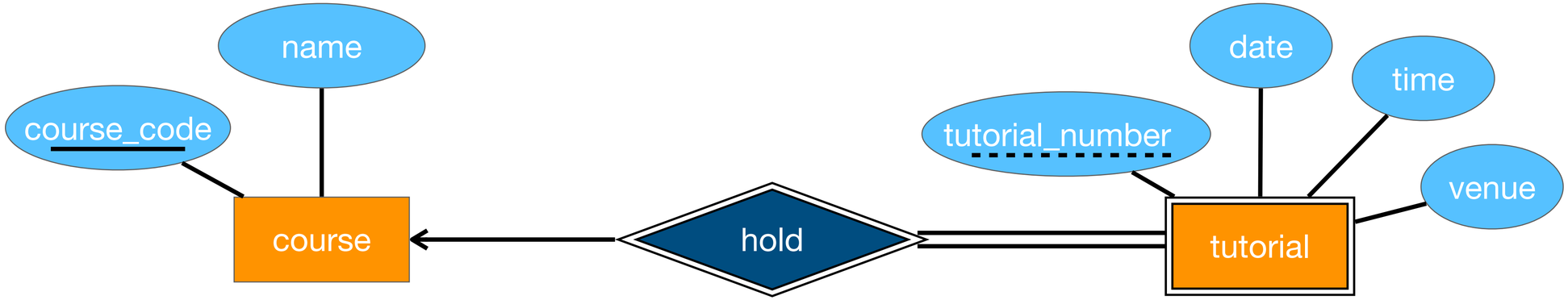
- A **foreign key** is a **referential constraint** between two tables.
- **The foreign key can be used to cross-reference tables.**
  - It is used to link information together.
- A **foreign key** is a field in a relational table that matches a **candidate key** of another table.

- author (authorID, name, date_of_birth)
  - Foreign key: none
- book (bookID, title, publisher)
  - Foreign key: none
- write (authorID, bookID)
  - Foreign keys: {authorID} referencing author
    {bookID} referencing book

What's more? Foreign Key

- *Step 1: entity set -> table*
- *Step 2: relationship set -> table?*
- *Step 3: identify primary key and foreign keys?*

- Weak entity set

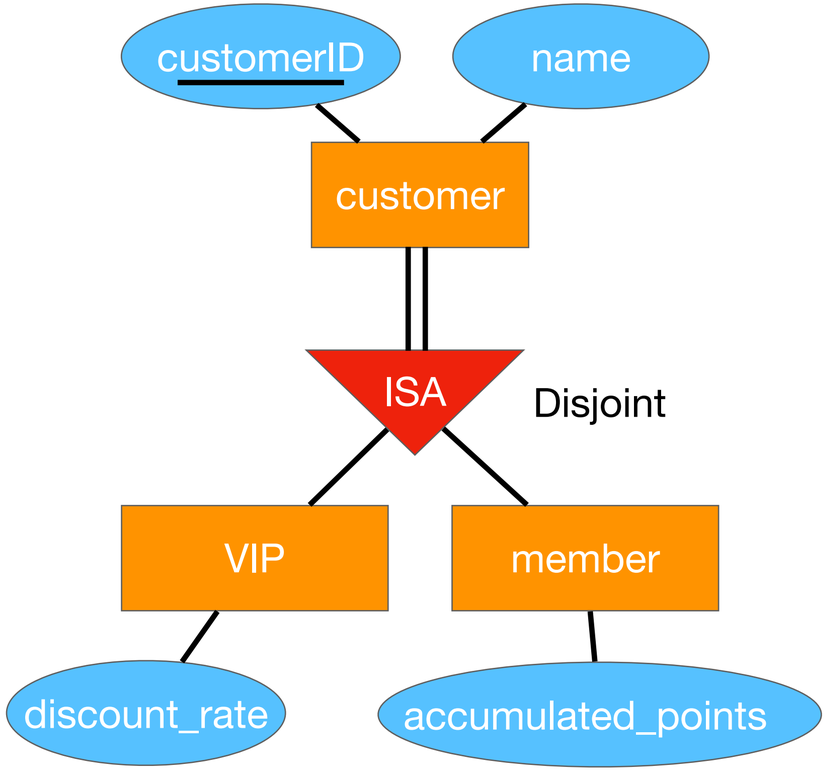


- Method: A weak entity set becomes a table that includes the columns for the primary key of the identifying strong entity set.
- course (course_code, name)
  - Foreign key: none
- tutorial (tutorial_number, date, time, venue, course_code)
  - Foreign key: {course_code} referencing course

What's more? Foreign Key

- *Step 1: entity set -> table*
- *Step 2: relationship set -> table?*
- *Step 3: identify primary key and foreign keys?*

- Specialization

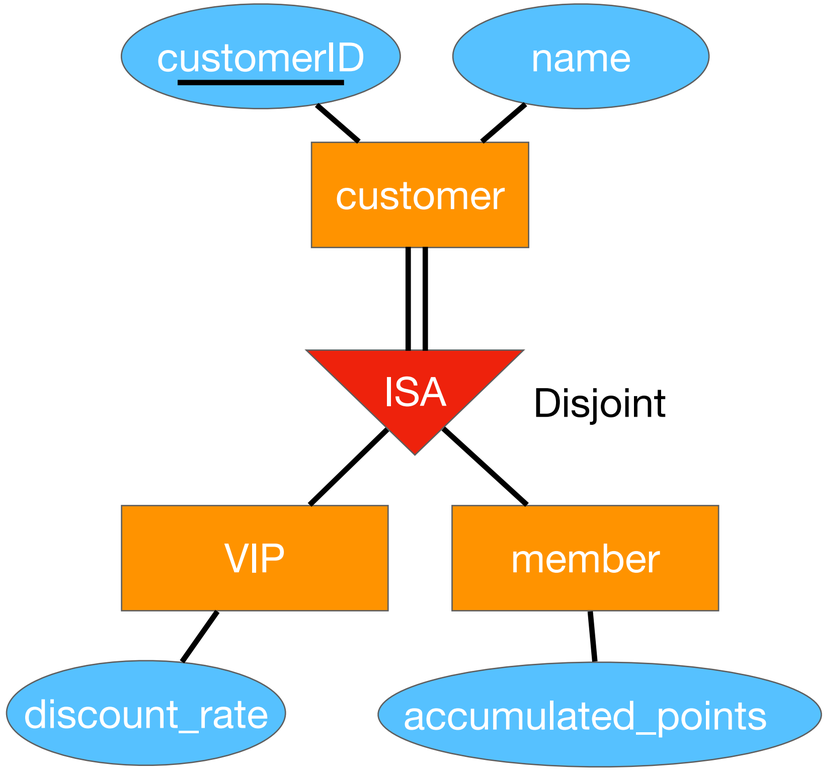


- Method 1: Form a table from the higher-level entity set; then form tables for each lower-level entity set, which contains the primary key of the higher-level entity set and local attributes.
  - customer (customerID, name)
    - Foreign key: none
  - VIP (customerID, discount_rate)
    - Foreign key: {customerID} referencing customer
  - member (customerID, accumulated_points)
    - Foreign key: {customerID} referencing customer

What's more? Foreign Key

- *Step 1: entity set -> table*
- *Step 2: relationship set -> table?*
- *Step 3: identify primary key and foreign keys?*

- Specialization

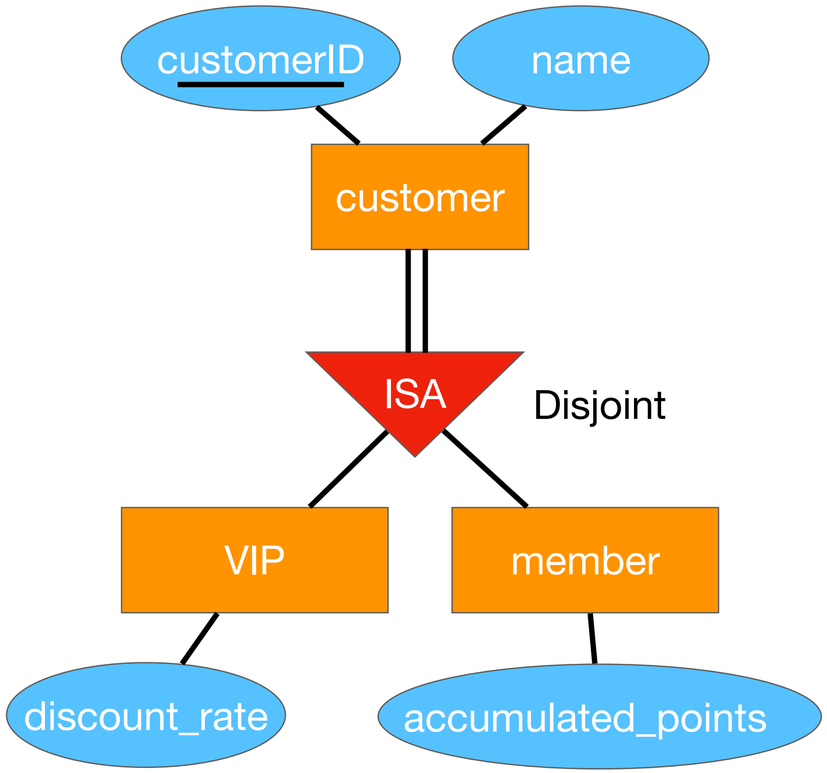


- Method 2: Form a table for each entity set with all local and inherited attributes.
  - customer (customerID, name)
    - Foreign key: none
  - VIP (customerID, name, discount_rate)
    - Foreign key: {customerID} referencing customer
  - member (customerID, name, accumulated_points)
    - Foreign key: {customerID} referencing customer

What's more? Foreign Key

- *Step 1: entity set -> table*
- *Step 2: relationship set -> table?*
- *Step 3: identify primary key and foreign keys?*

- Specialization

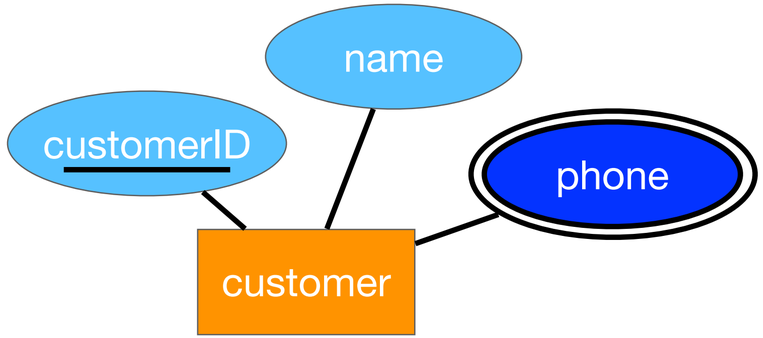


- If the specialization is total?
  - Method 2: Form a table for each entity set with all local and inherited attributes.
    - ~~customer (customerID, name)~~
      - ~~Foreign key: none~~
    - VIP (customerID, name, discount_rate)
      - Foreign key: none
    - member (customerID, name, accumulated_points)
      - Foreign key: none

What's more? Foreign Key

- *Step 1: entity set -> table*
- *Step 2: relationship set -> table?*
- *Step 3: identify primary key and foreign keys?*

- multi-valued attribute



- Method: A multi-valued attribute M of an entity set E is represented by a separate table EM, with the primary key of E as one of EM's attribute.
- customer (customerID, name)
  - Foreign key: none
- customerPhone (customerID, phone)
  - Foreign key: {customerID} referencing customer

Chapter 2

END

COMP3278C
Introduction to Database Management Systems

Dr. CHEN, Yi
Email: chenyi1@hku.hk

School of Computing & Data Science, The University of Hong Kong

Acknowledgement: Dr. Chui Chun Kit, Dr. Reynold Cheng, Dr. Ping Luo