CMSC424: Database Design
ER Model

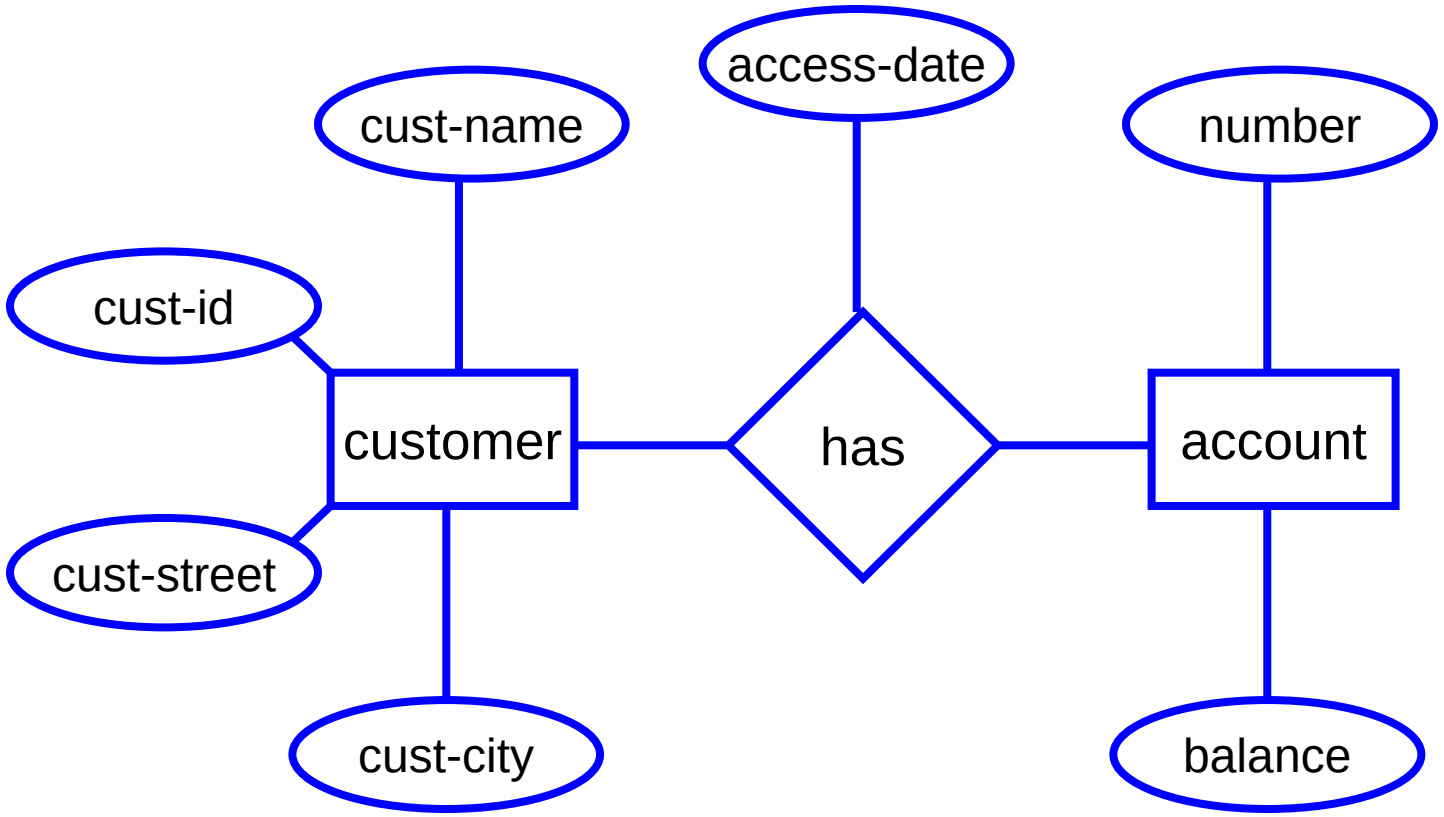
Entity-Relationship Model

- Two key concepts
  - *Entities*:
    - An object that *exists* and is *distinguishable* from other objects
      - Examples: Bob Smith, BofA, CMSC424
    - Have *attributes* (people have names and addresses)
    - Form *entity sets* with other entities of the same type that share the same properties
      - Set of all people, set of all classes
    - Entity sets may overlap
      - Customers and Employees

Entity-Relationship Model

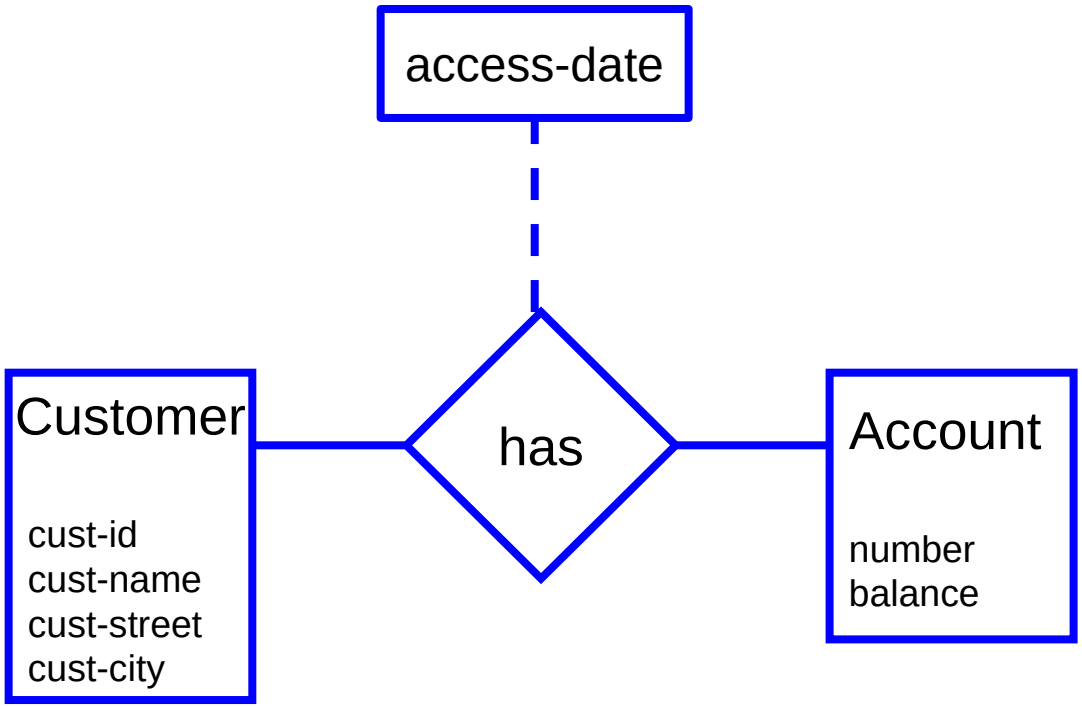
- Two key concepts
  - *Relationships*:
    - Relate 2 or more entities
      - E.g. Bob Smith *has account at* College Park Branch
    - Form *relationship sets* with other relationships of the same type that share the same properties
      - Customers *have accounts at* Branches
    - Can have attributes:
      - *has account at* may have an attribute *start-date*
    - Can involve more than 2 entities
      - Employee *works at* Branch *at* Job

ER Diagram: Starting Example



- Rectangles: entity sets
- Diamonds: relationship sets
- Ellipses: attributes

ER Diagram: Starting Example



- Rectangles: entity sets
- Diamonds: relationship sets
- Ellipses: attributes

Next: Relationship Cardinalities

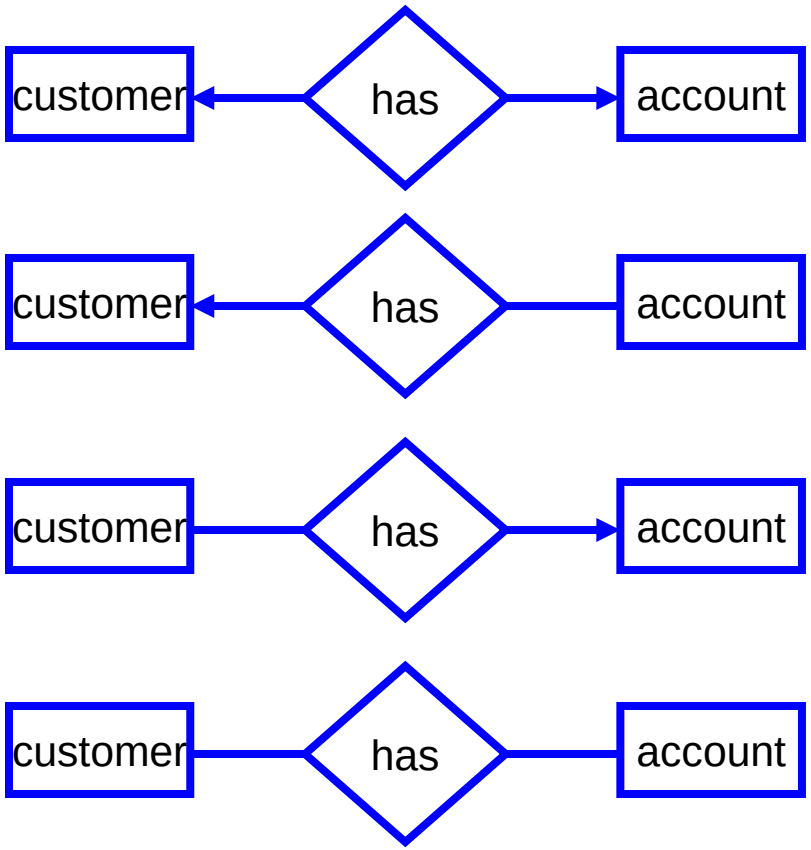
- We may know:
  - One customer can only open one account
  - OR
  - One customer can open multiple accounts
- Representing this is important
- Why ?
  - Better manipulation of data
    - If former, can store the account info in the customer table
  - Can enforce such a constraint
    - Application logic will have to do it; NOT GOOD
  - Remember: If not represented in conceptual model, the domain knowledge may be lost

Mapping Cardinalities

- Express the number of entities to which another entity can be associated via a relationship set
- Most useful in describing binary relationship sets

Mapping Cardinalities

- One-to-One
- One-to-Many
- Many-to-One
- Many-to-Many



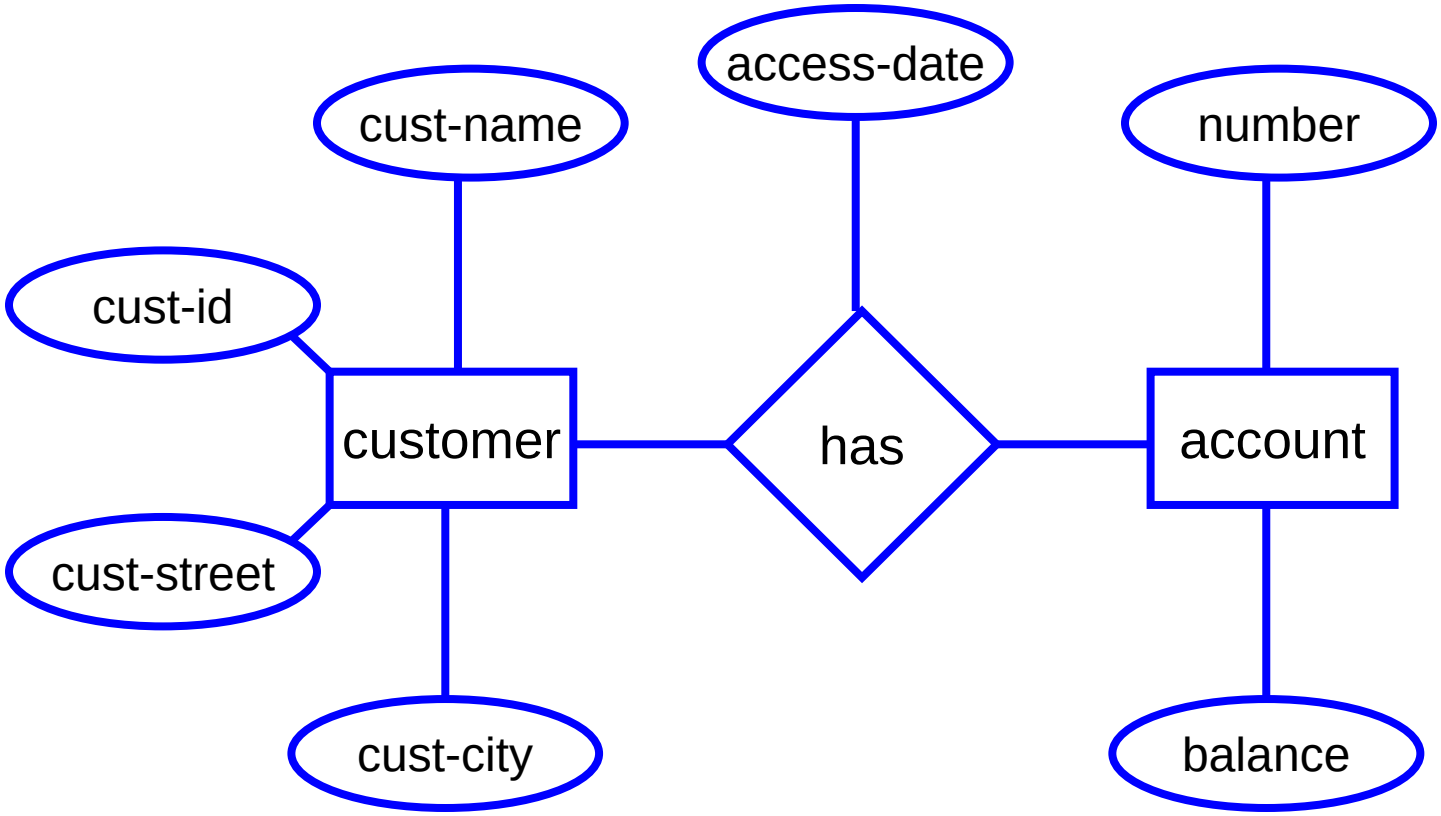
Mapping Cardinalities

- N-ary relationships ?
  - More complicated
  - Details in the book

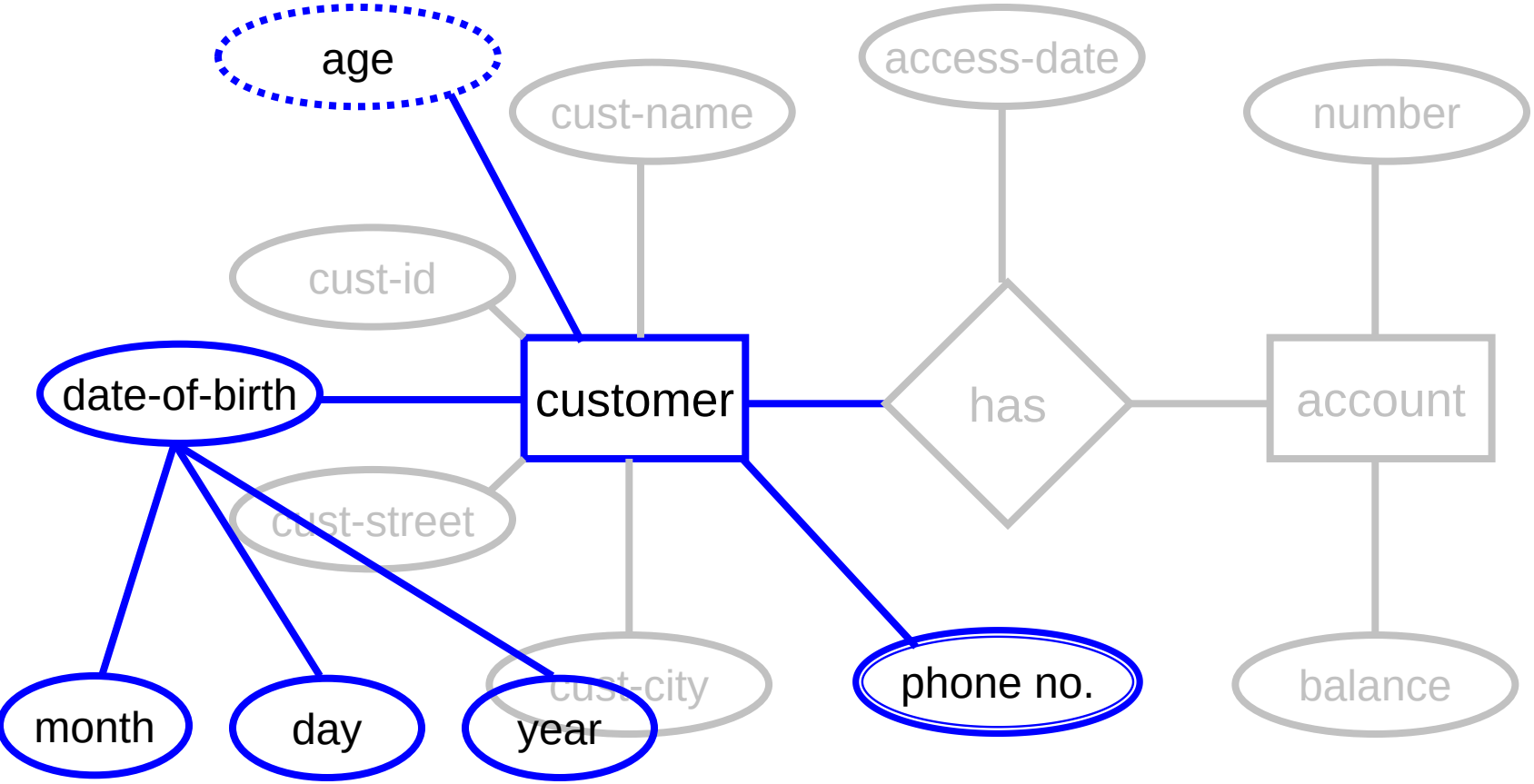
Next: Types of Attributes

- Simple vs Composite
  - Single value per attribute ?
- Single-valued vs Multi-valued
  - E.g. Phone numbers are multi-valued
- Derived
  - If date-of-birth is present, age can be derived
  - Can help in avoiding redundancy, enforcing constraints etc…

Types of Attributes

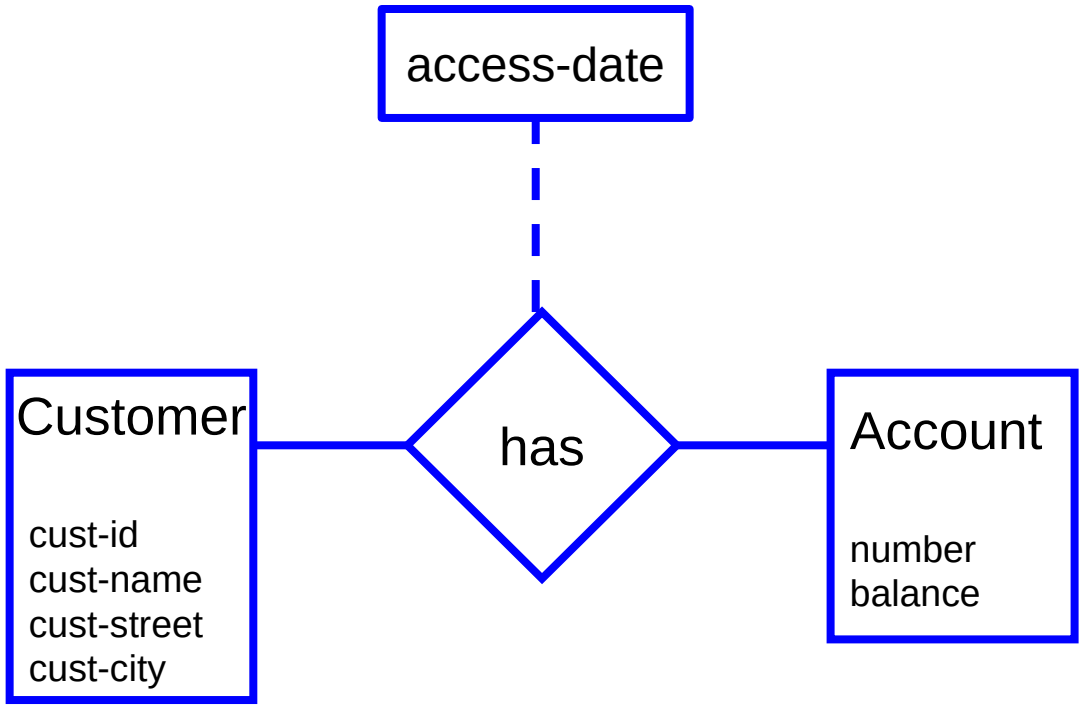


Types of Attributes

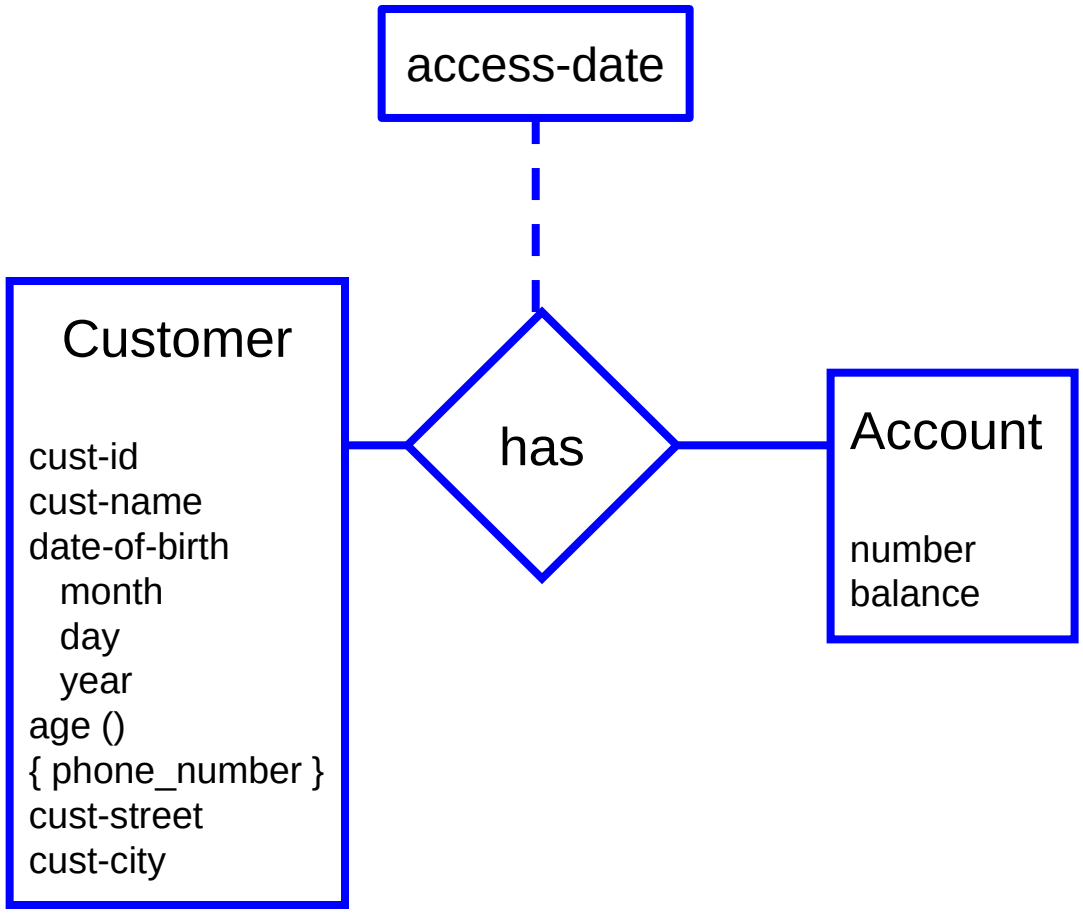


Composite Attribute

ER Diagram: Starting Example



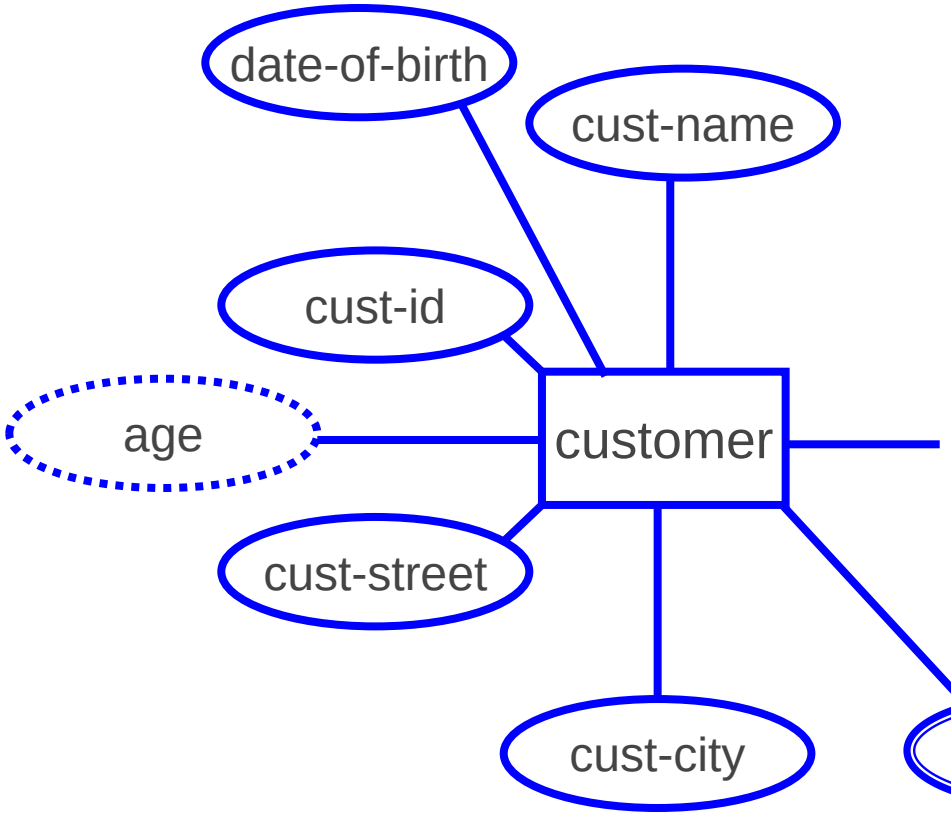
ER Diagram: Starting Example



Next: Keys

- Key = set of attributes that uniquely identifies an entity or a relationship

Entity Keys



Possible Keys:

{cust-id}

{cust-name, cust-city, cust-street}

{cust-id, age}

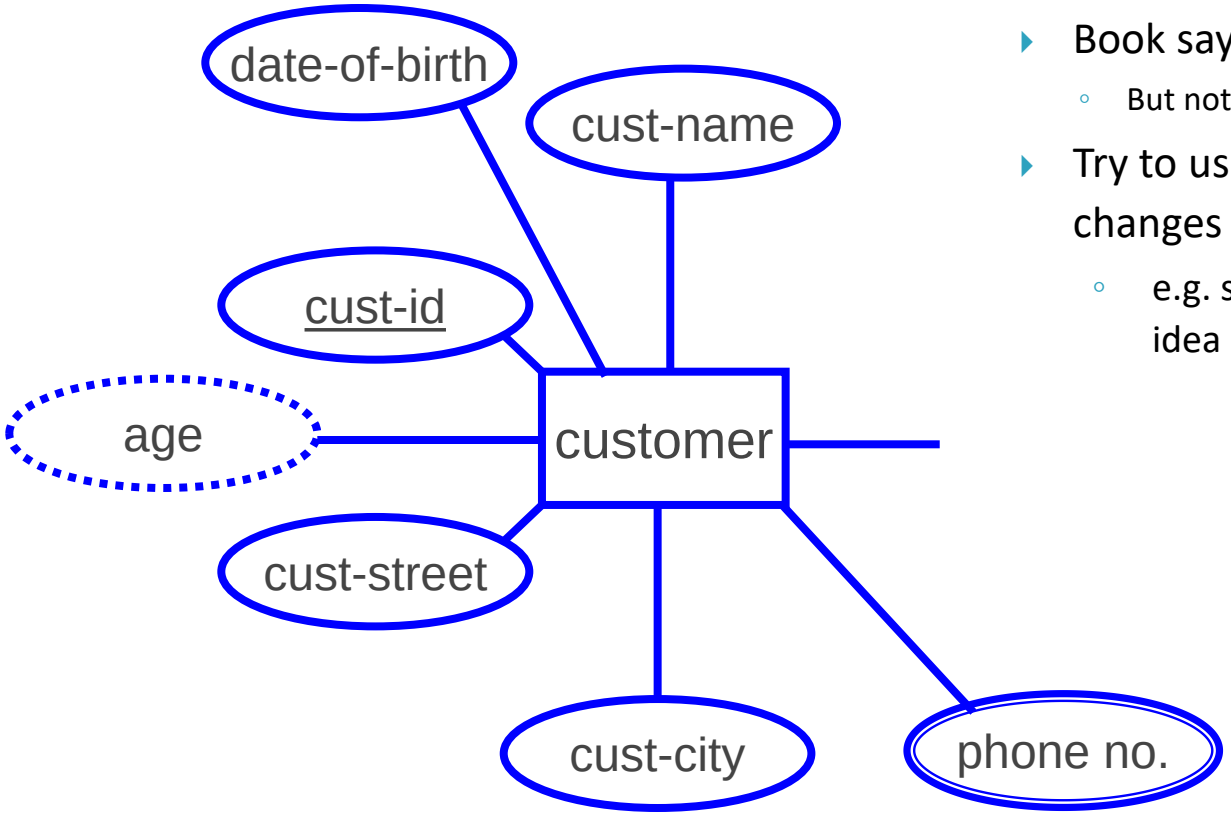
cust-name ?? Probably not.

Domain knowledge dependent !!

Entity Keys

- *Superkey*
  - any attribute set that can distinguish entities
- *Candidate key*
  - a minimal superkey
    - Can't remove any attribute and preserve key-ness
      - {cust-id, age} not a candidate key
      - {cust-name, cust-city, cust-street} is
        - assuming cust-name is not unique
- *Primary key*
  - Candidate key chosen as *the* key by DBA
  - Underlined in the ER Diagram

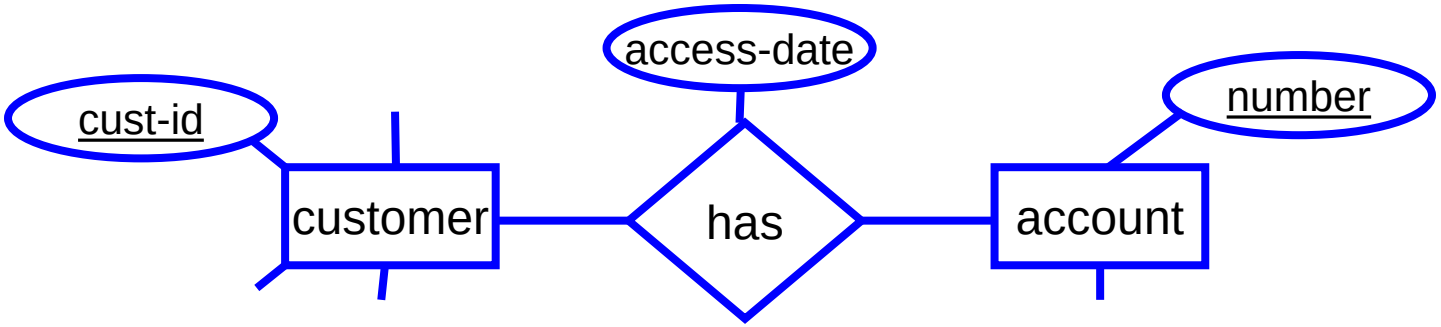
Entity Keys



- {*cust-id*} is a natural primary key
- Book says SSN forms a good primary key
  - But not any more. Can you guess why?
- Try to use a candidate key that rarely changes
  - e.g. something involving address not a great idea

Relationship Set Keys

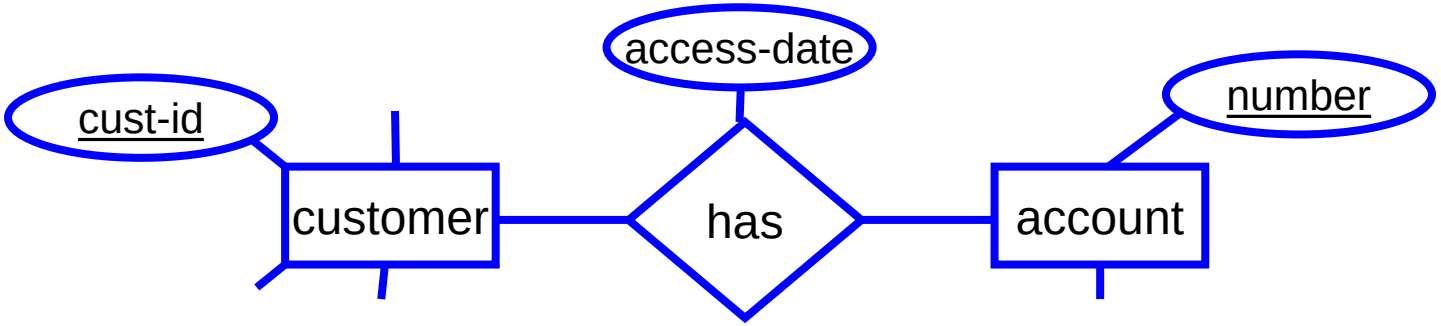
- What attributes are needed to represent a relationship completely and uniquely ?
  - Union of primary keys of the entities involved, and relationship attributes



  - {cust-id, access-date, account number} describes a relationship completely

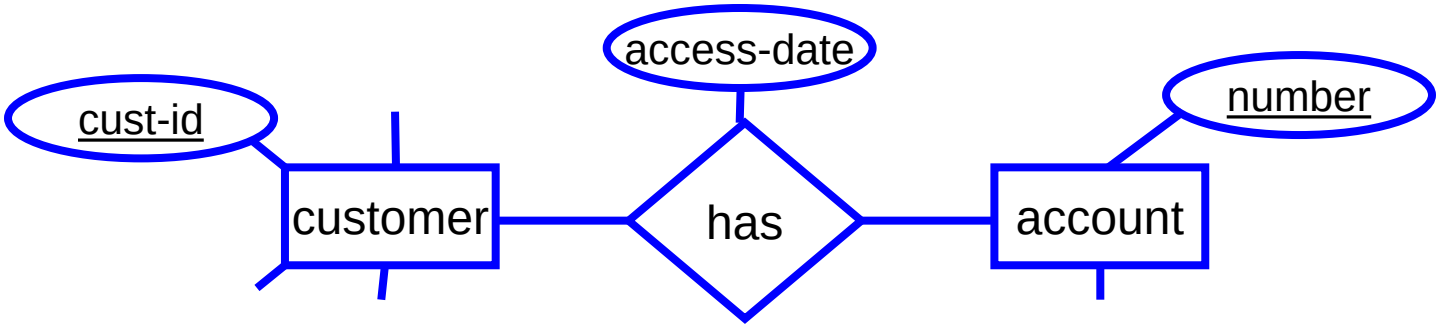
Relationship Set Keys

- Is *{cust-id, access-date, account number}* a candidate key *?*
  - No. Attribute *access-date* can be removed from this set without losing key-ness
  - In fact, union of primary keys of associated entities is always a superkey



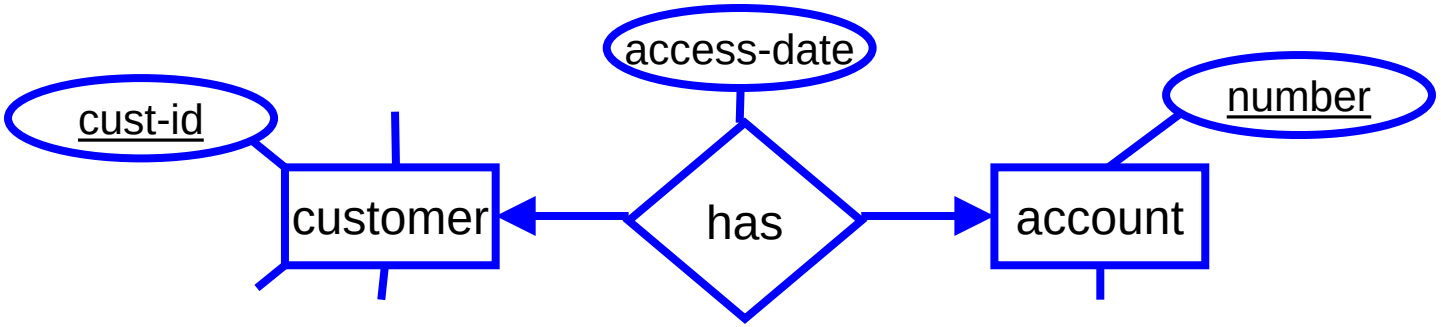
Relationship Set Keys

- Is {cust-id, account-number} a candidate key ?
  - Depends



Relationship Set Keys

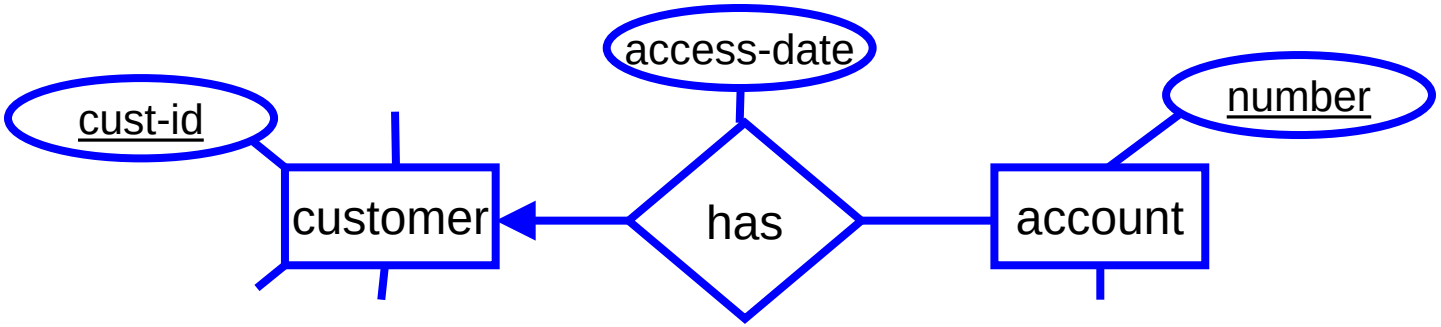
- Is {cust-id, account-number} a candidate key ?
  - Depends



- If one-to-one relationship, either *{cust-id}* or *{account-number}* sufficient
  - Since a given *customer* can only have one *account*, she can only participate in one relationship
  - Ditto *account*

Relationship Set Keys

- Is {cust-id, account-number} a candidate key ?
  - Depends

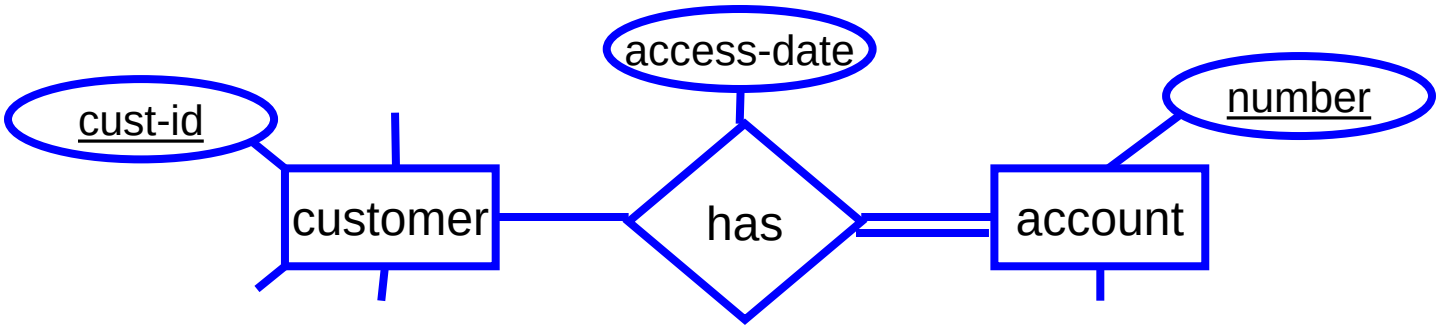


- If one-to-many relationship (as shown), *{account-number}* is a candidate key
  - A given customer can have many accounts, but at most one account holder per account allowed

Relationship Set Keys

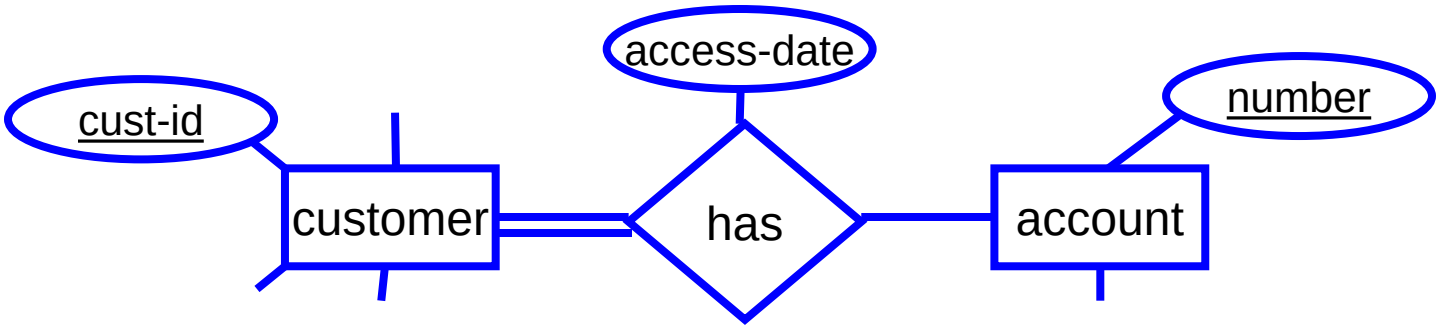
- General rule for binary relationships
  - one-to-one: primary key of either entity set
  - one-to-many: primary key of the entity set on the many side
  - many-to-many: union of primary keys of the associate entity sets
- n-ary relationships
  - More complicated rules

Participation constraints



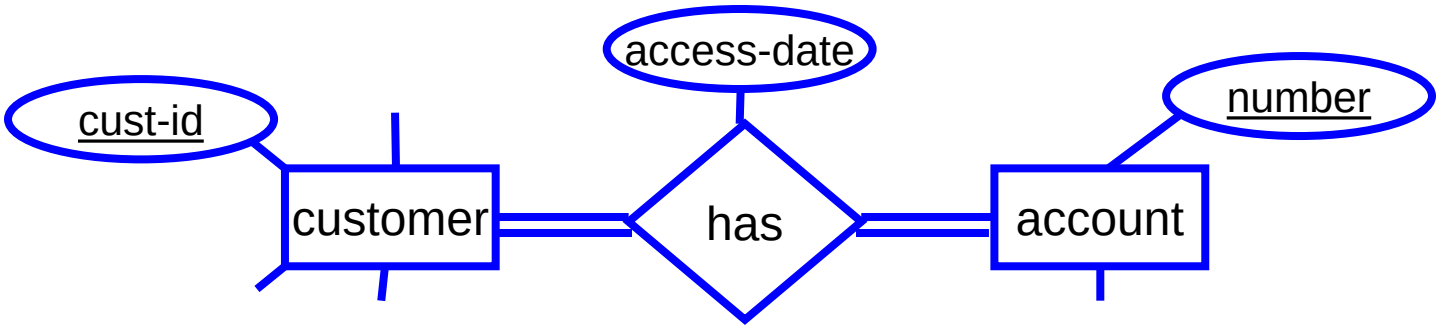
- Every account has at least one customer, but customers may have 0 accounts

Participation constraints



- Every customer has at least one account, but accounts may be ownerless

Participation constraints

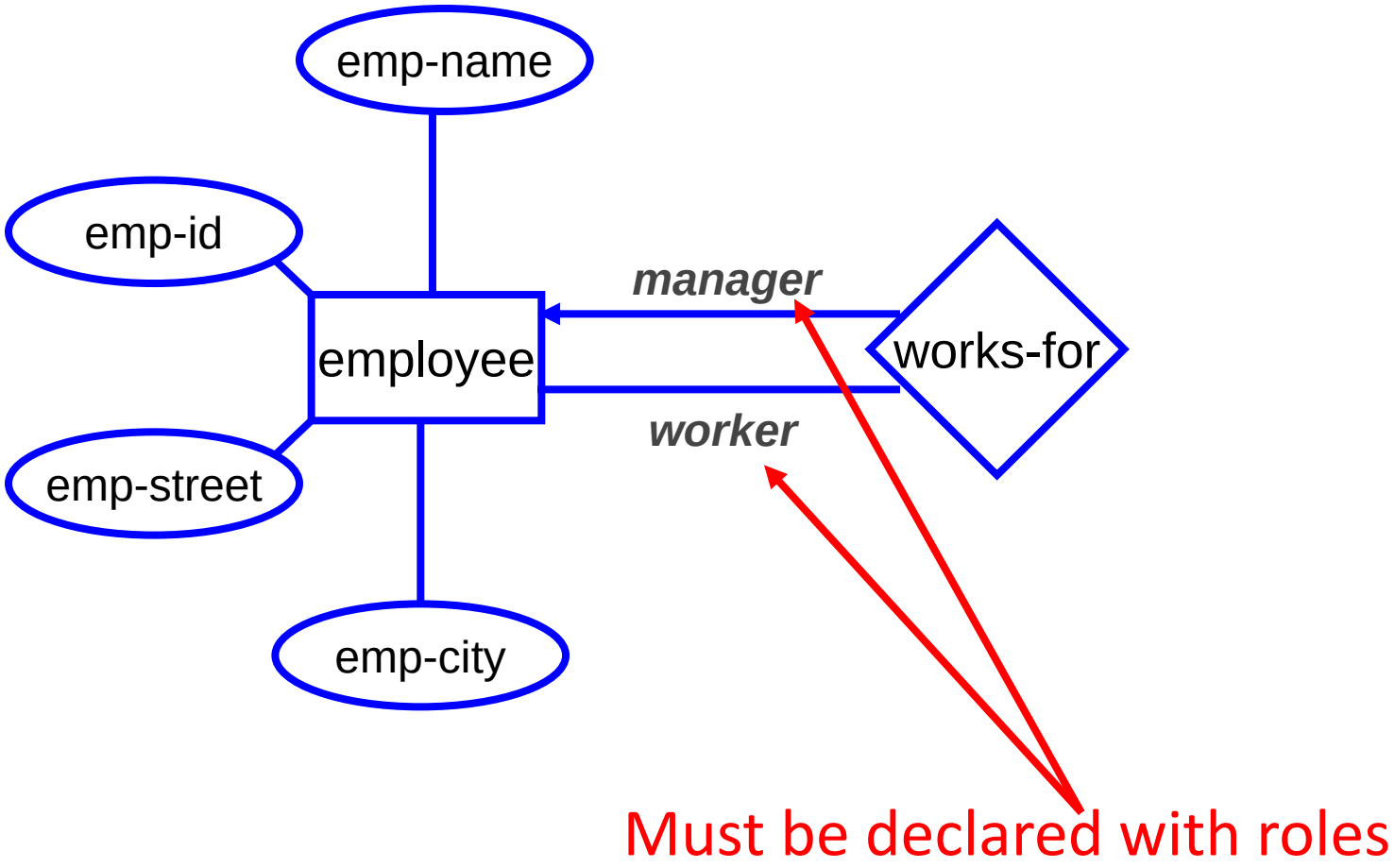


- Every customer has at least one account, and every account has at least one customer

Next: Recursive Relationships

- Sometimes a relationship associates an entity set to itself

Recursive Relationships



Next: Weak Entity Sets

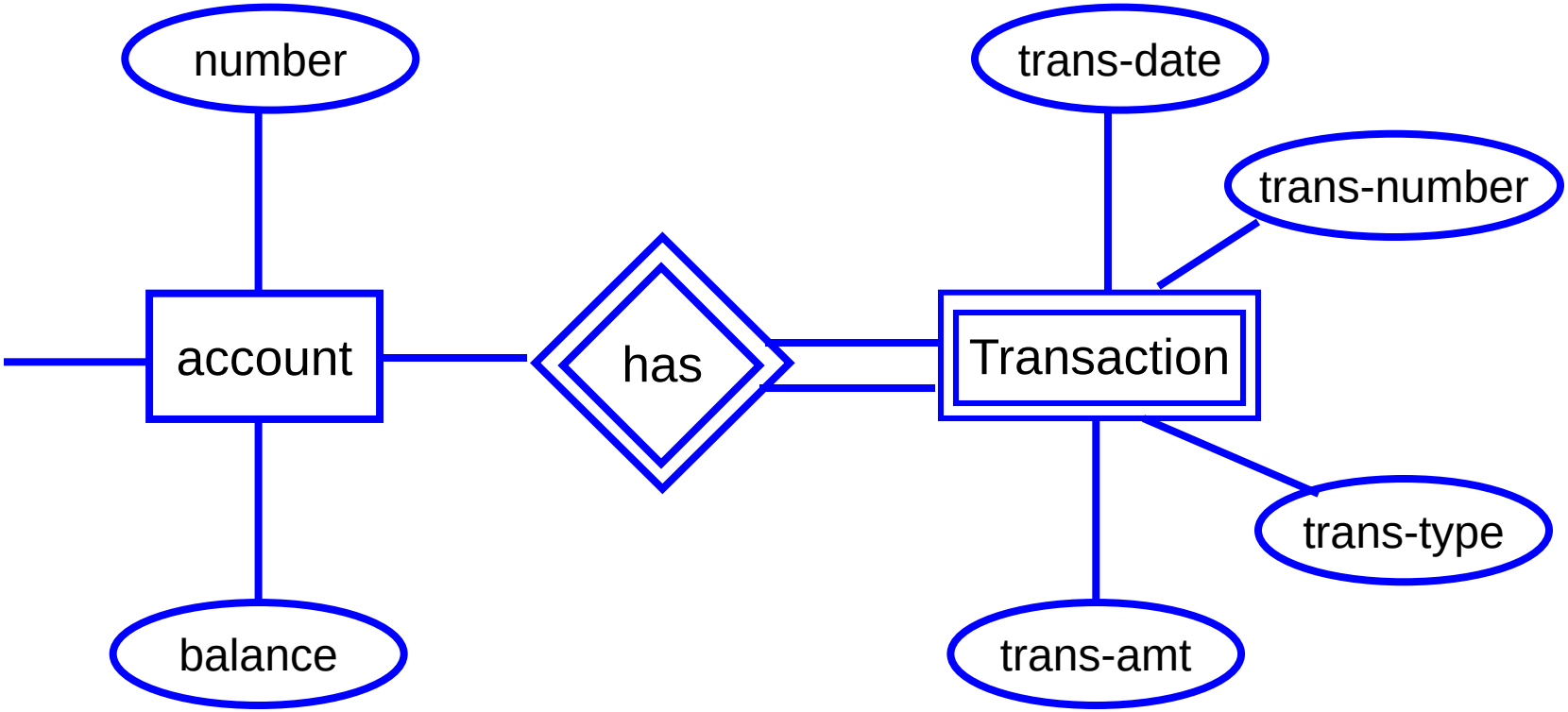
- An entity set without enough attributes to have a primary key
- E.g. Transaction Entity
  - Attributes:
    - transaction-number, transaction-date, transaction-amount, transaction-type
    - transaction-number: may not be unique across accounts

Weak Entity Sets

- A weak entity set must be associated with an identifying or owner entity set
- Account is the owner entity set for Transaction

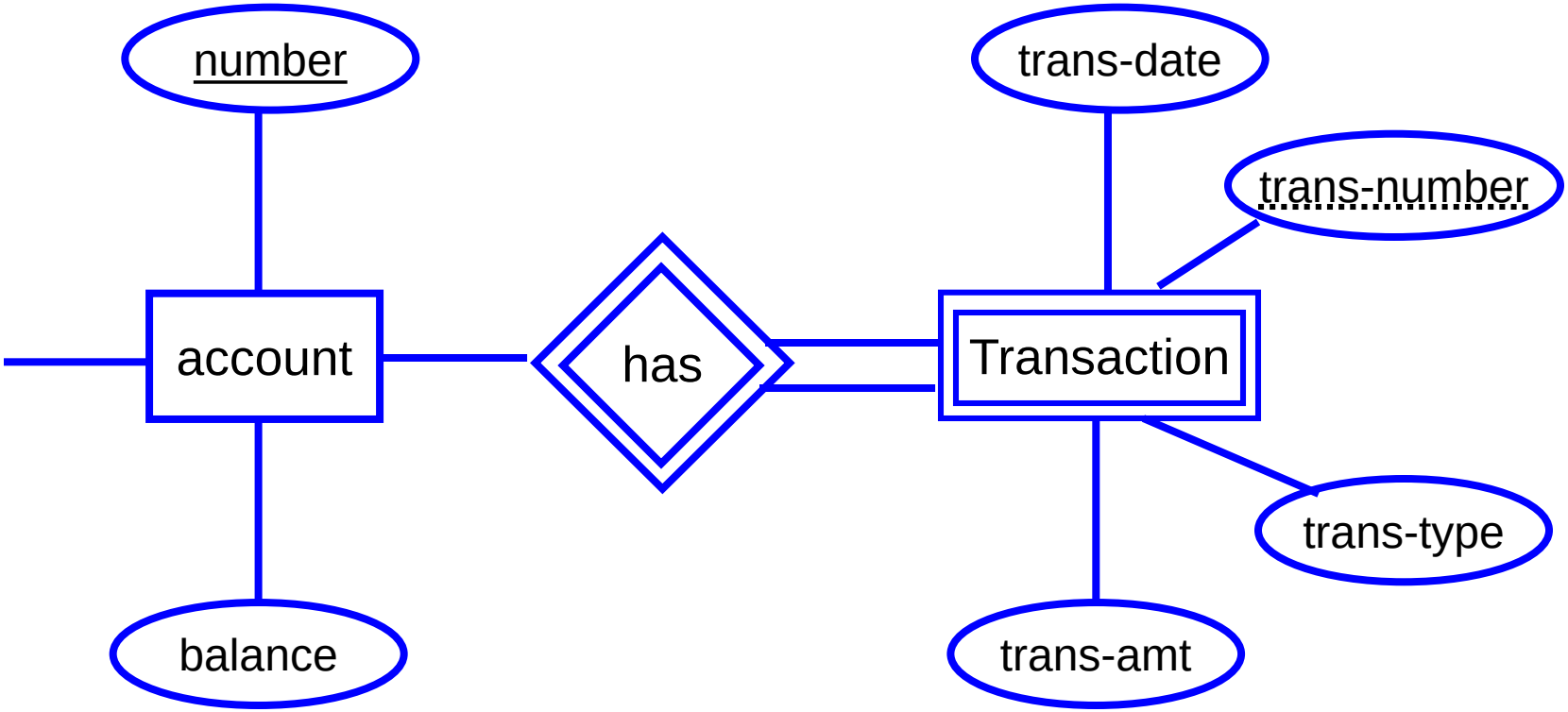
Weak Entity Sets

Still need to be able to distinguish between different weak entities associated with the same strong entity



Weak Entity Sets

Discriminator: A set of attributes that can be used for that



Weak Entity Sets

- Primary key:
  - Combine:
    - associated strong entity
    - discriminator attribute set
  - For Transaction:
    - *{account-number, transaction-number}*

More…

- (optional) Read rest of chapter 7 for:
  - Specialization/Generalization/Aggregation
  - Generalization: opposite of specialization
  - Lower- and higher-level entities
  - Attribute inheritance