Lecture 02: Data Modeling and the Entity-Relationship Model

EGCI 321: LECTURE02 (WEEK01)

Outline

1. Basic E-R Modeling
   - Entities
   - Attributes
   - Relationships
   - Roles
2. Constraints in E-R Models
   - Primary Keys
   - Relationship Types
   - Existence Dependencies
   - General Cardinality Constraints
3. Extensions to E-R Modeling
   - Structures Attributes
   - Aggregation
   - Specialization
   - Generalization
   - Disjointness
4. Design Considerations

Overview of E-R Model

- Used for (and designed for) database (conceptual schema) design
- World/enterprise described in terms of:
  - Entities
  - Attributes
  - Relationships
- Visualization E-R diagram
- Many variant notations are in common use

Basic E-R Modeling

- **Entity:** a distinguishable object
- **Entity set:** set of entities of same type

Examples:

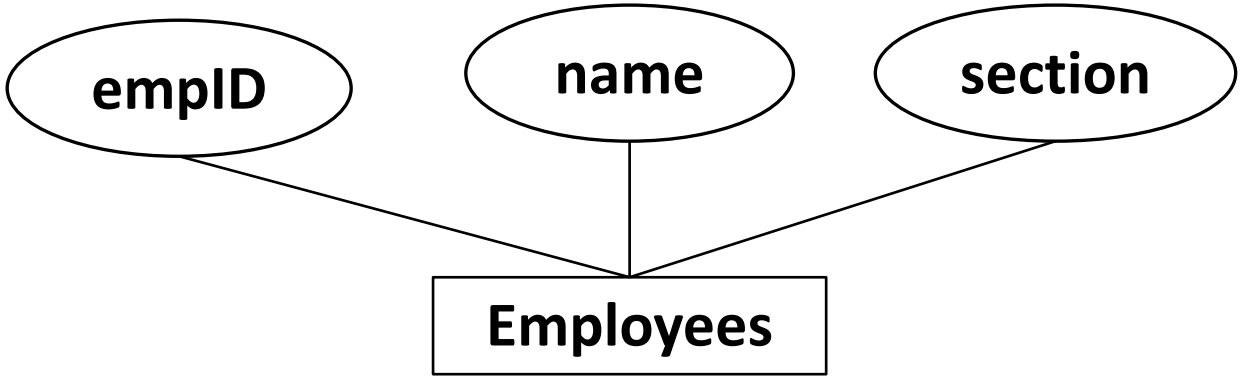
- Students currently at Mahidol University
- Flights offered by Thai Airway
- Burglaries in Bangkok during 1994

- Graphical representation of entity sets:

Basic E-R Modeling (cont.)

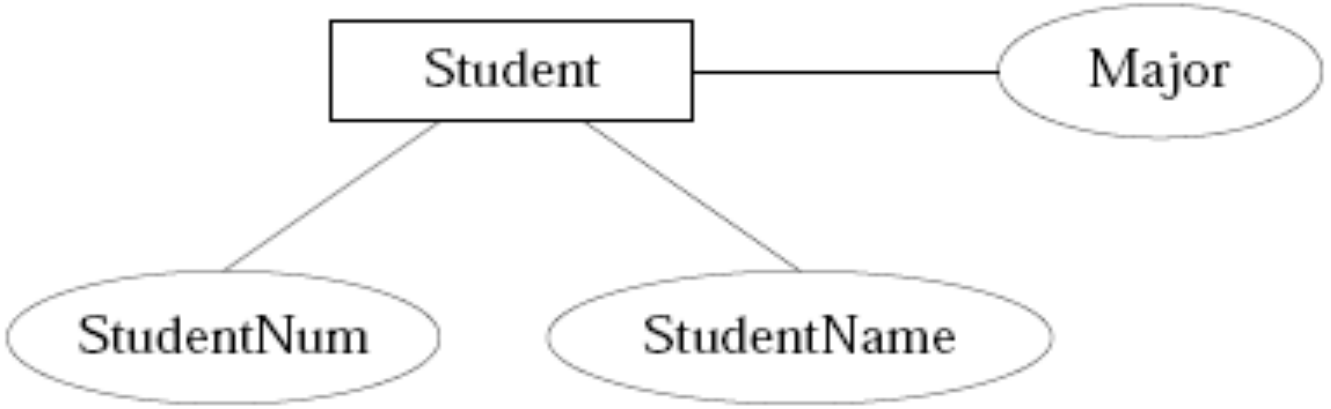
Entity Set: A collection of similar entities

- E.g. all employees
- All entities in an entity set have the same set of attributes.
- Each entity set has a key.
- Each attribute has a domain.



Basic E-R Modeling (cont.)

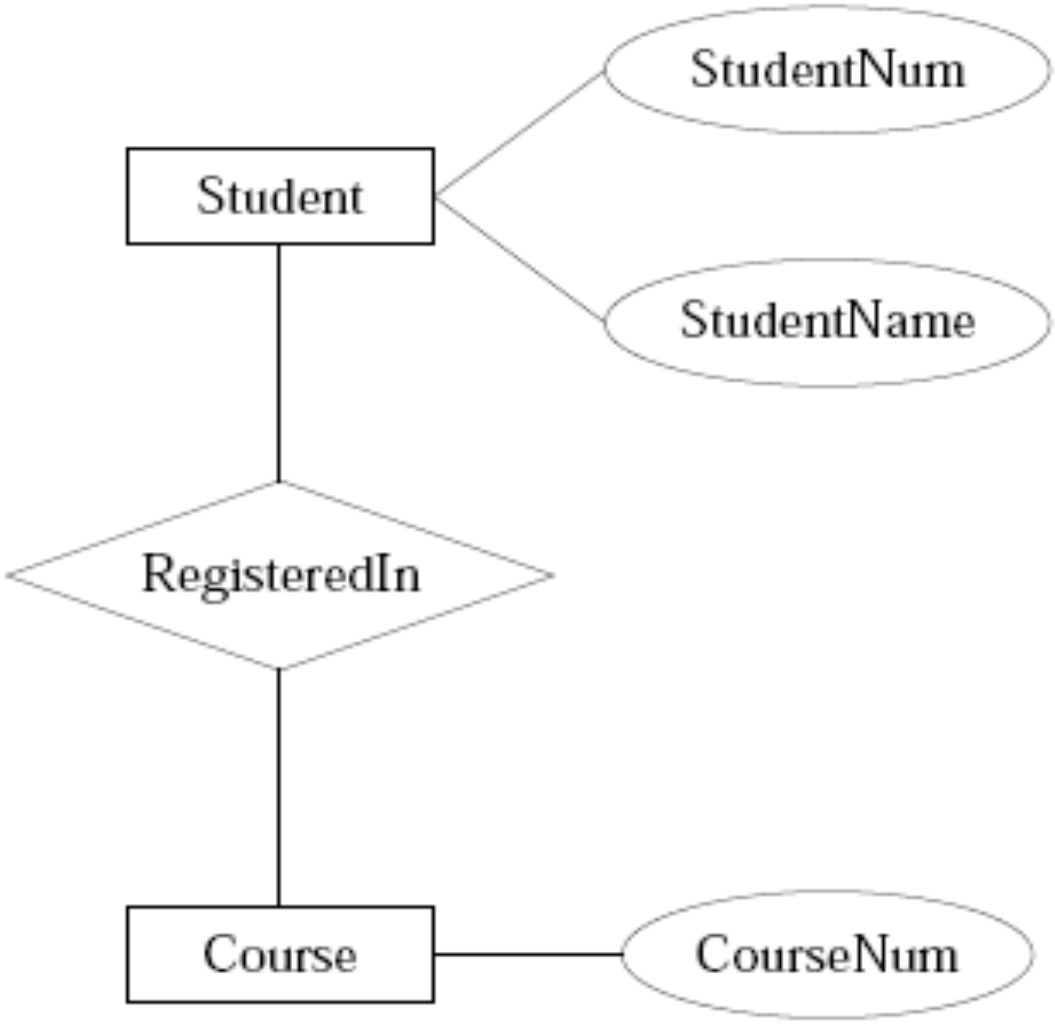
- **Attributes:** describe properties of entities
  Examples (for Student-entites): StudentNum, StudentName, Major,...
- **Domain:** set of permitted values for an attribute
- Graphical representation of attributes:



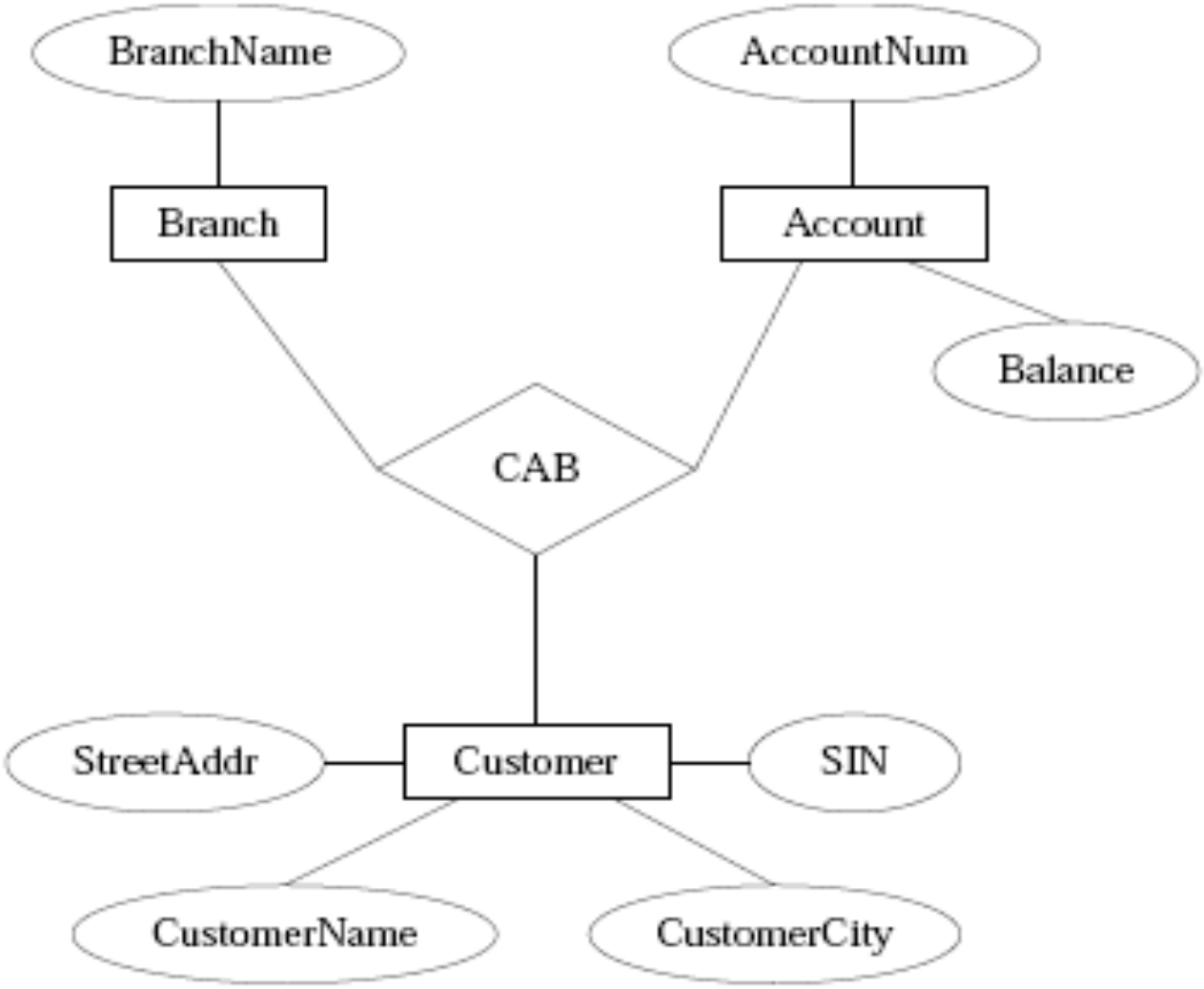
Basic E-R Modeling (cont.)

- **Relationship:** representation of the fact that certain entities are related to each other
  - Association among two or more entities.
  - E.g., Bob works in Pharmacy department.
- **Relationship set:** set of relationships of a given type
  - Examples:
    - Students registered in course
    - Passengers booked on flights
    - Parents and their children
    - Bank branches, customers and their accounts
- In order for a relationship to exist, the participating entities must exist

Graphical Representation



Graphical Representation (cont.)



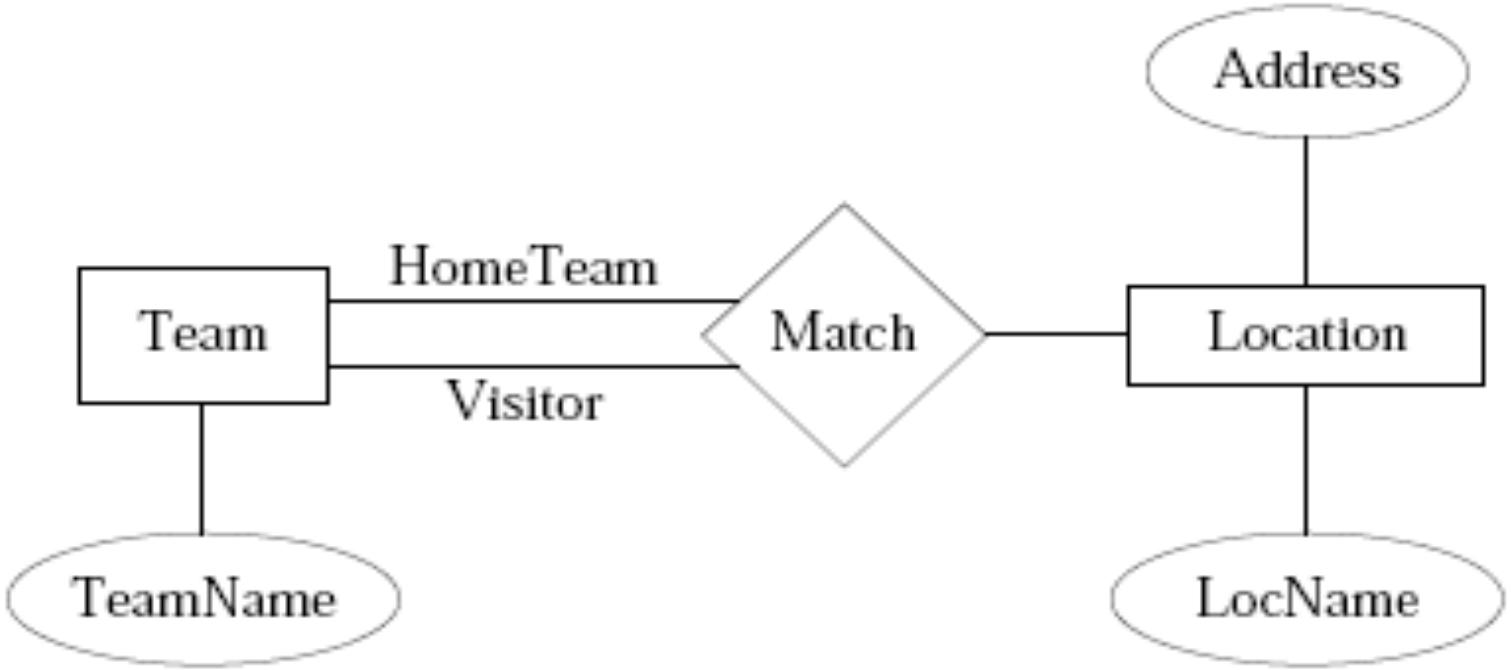
Relationship Set

- An n-ary relationship set *R* relates *n* entity sets *E1 ... En*;
- each relationship in *R* involves entities $e1 \in E1, ..., en \in En$
- Same entity set could participate in different relationship sets, or in different “roles” in same set.

Multiple Relationships and Role Names

- **Role:** the function of an entity set in a relationship set
- **Role name:** an explicit indication of a role

Example:

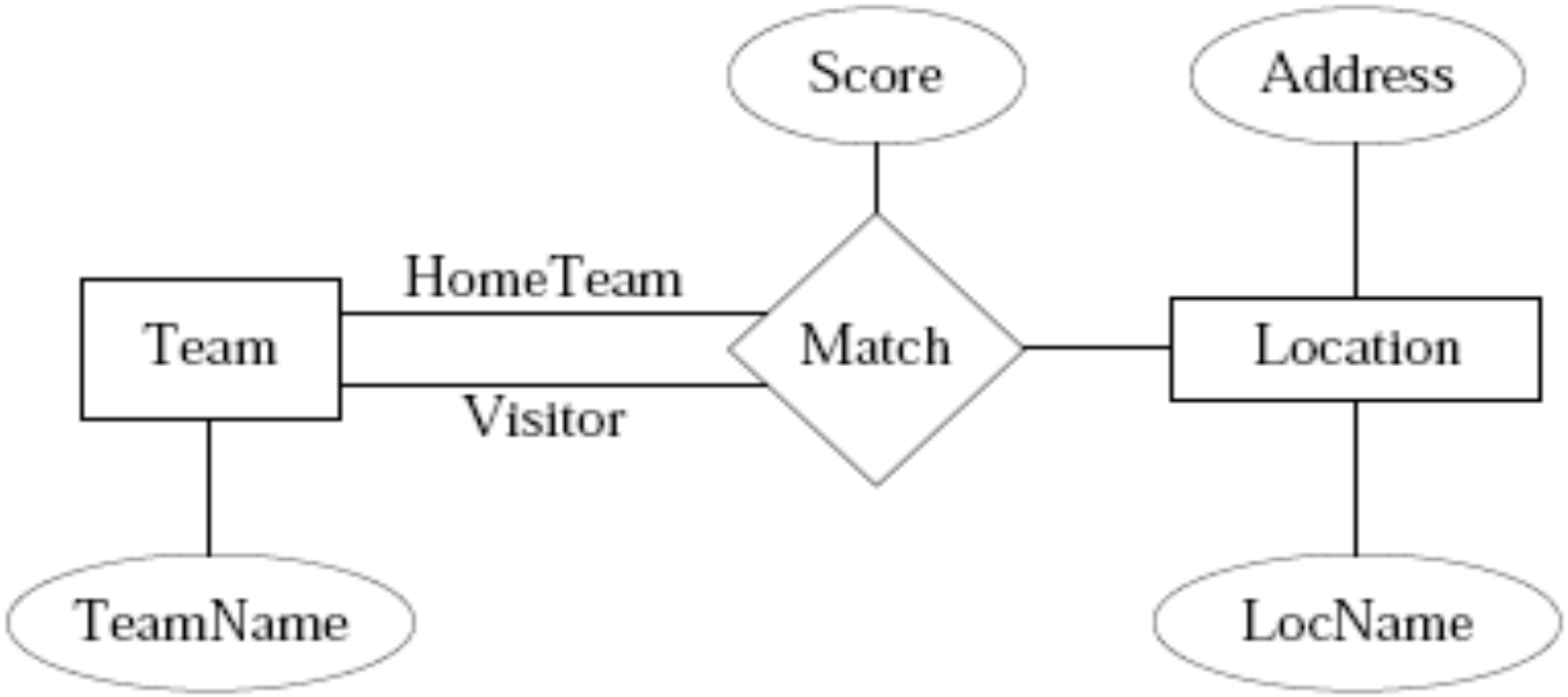


- Role labels are needed whenever an entity set has multiple functions in a relationship set.

Relationships and Attributes

- Relationships may also have attributes

Example:



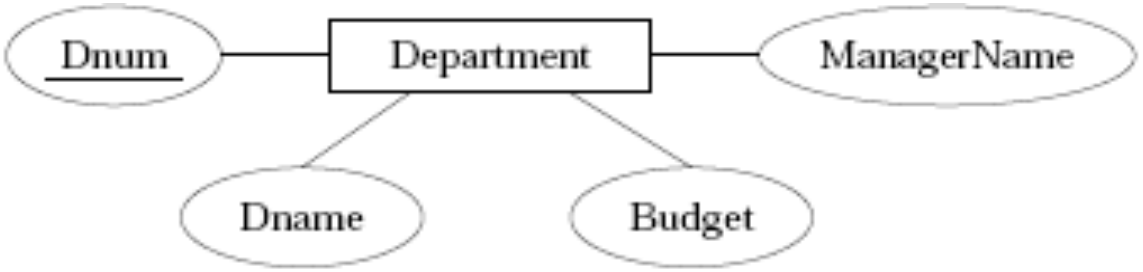
Constraints in E-R Models

- Primary keys
- Relationship types
- Existence dependencies
- General cardinality constraints

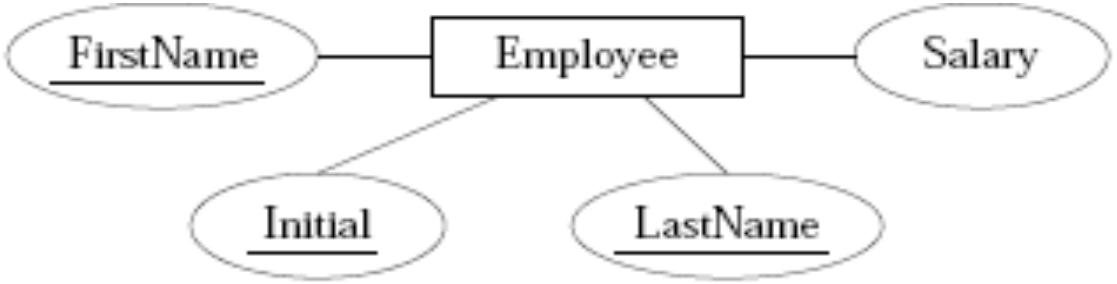
Primary Keys

- Each entity must be distinguishable from any other entity in an entity set by its attributes
- **Primary keys:** selection of attributes chosen by designer values of which determines the particular entity.

Example 1:



Example 2:



Relationship Types

- Many-to-many (M:N): an entity in one set can be related to many entities in the other set, and vice versa

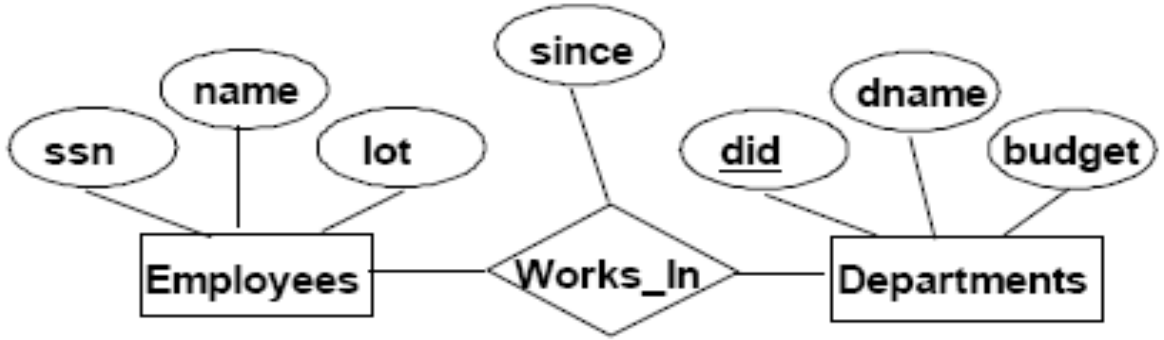
- Many-to-many (N:1) each entity in one set can be related to at most one entity in the other, but an entity in the second set may be related to many entities in the first

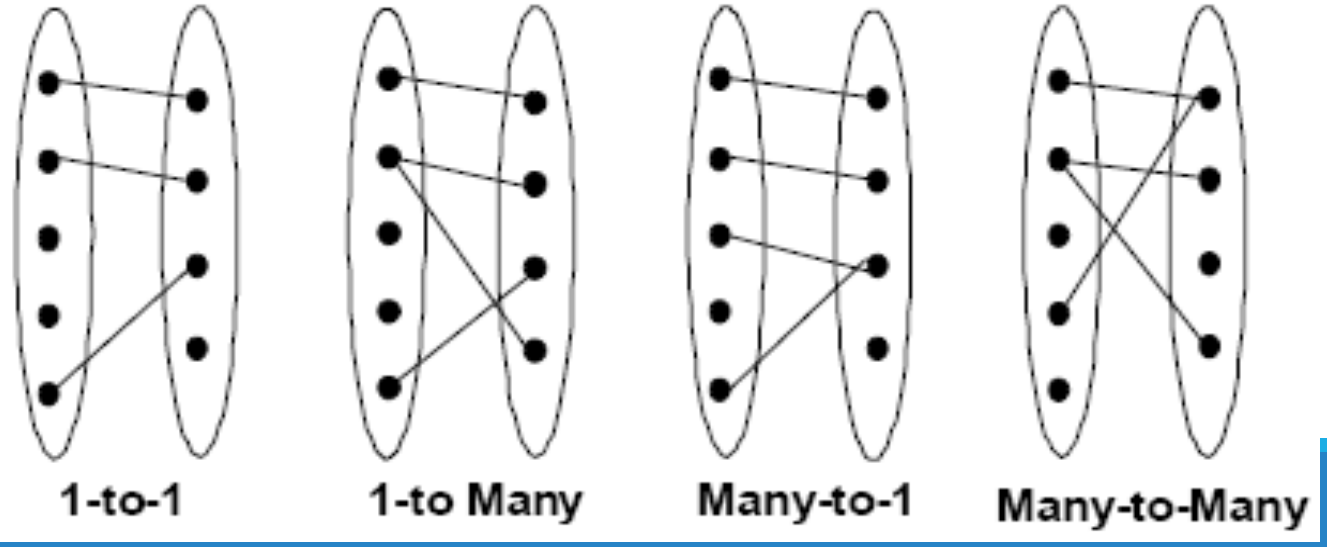
Relationship Types (cont.)

- One-to-many (1:N): similar to Many-to-One (N:1)
- One-to-one (1:1): each entity is one set can be related to at most one entity in the other, and vice versa
- (Arrow's Head :1 , Arrow's Tail: many)

Key Constraints

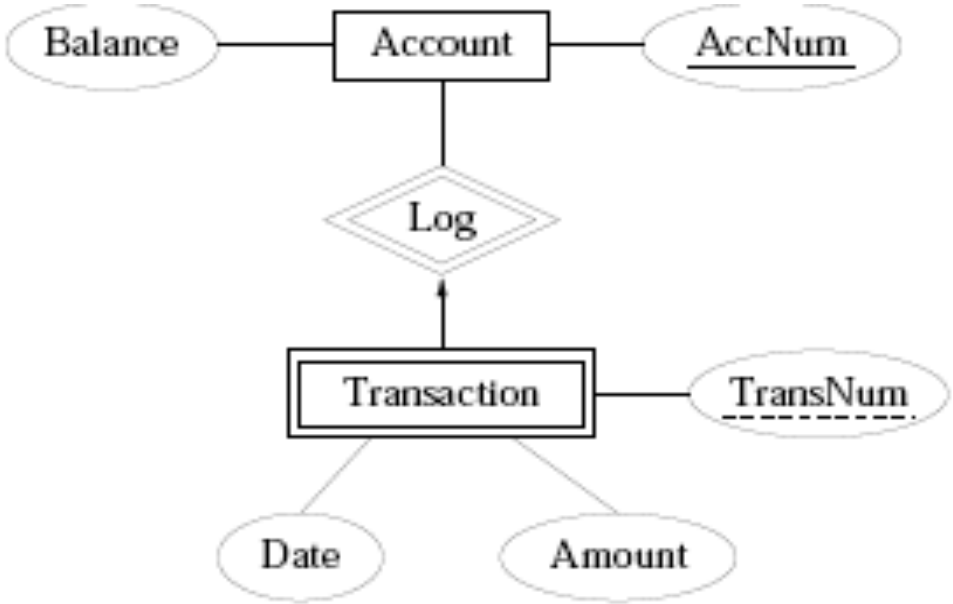
Example: Relationship "*Works_In*" , an employee can work in many departments; a department can have many employees.





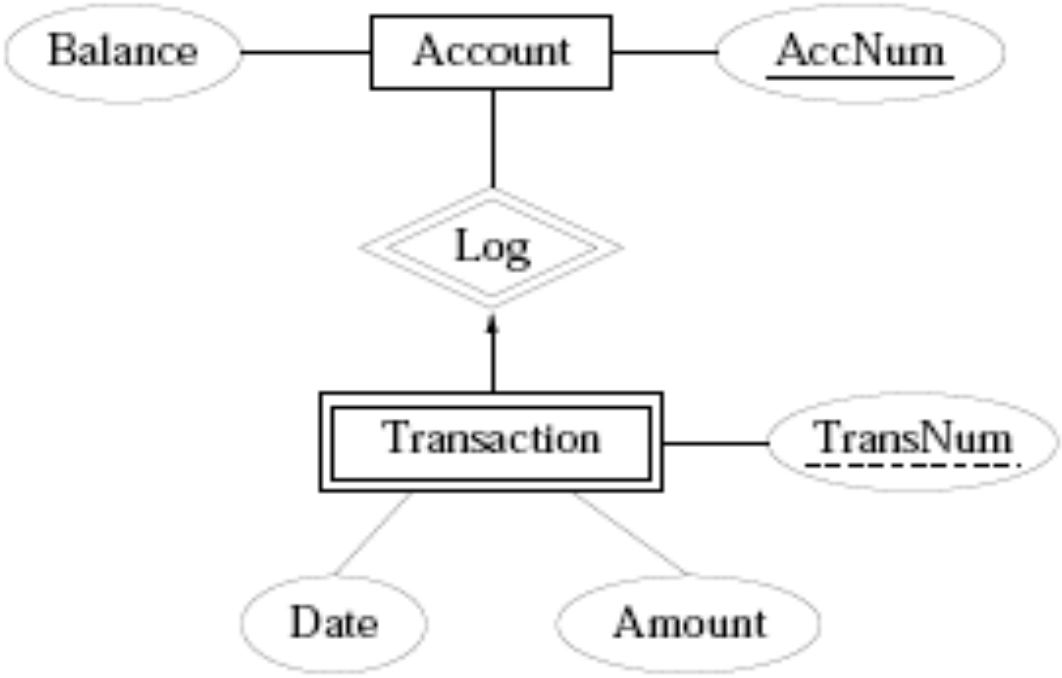
Existence Dependencies

- Sometimes the existence of an entity depends on the existence of another entity
- If x is existence dependent on y, then
  - y is a dominant entity
  - x is a subordinate entity
- Example: "Transactions are existence dependent on accounts"



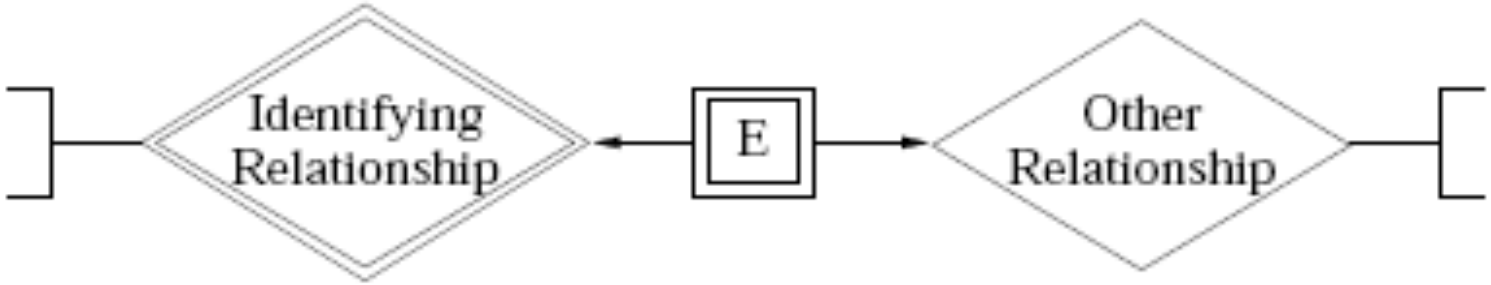
Identifying Subordinate Entities

- Weak entity set: an entity set containing subordinate entities
- Strong entity set: an entity set containing no subordinate entities
- *Attributes of weak entity sets only from key relative to a given dominant entity*
- Example: "All transactions for a given account have a unique transaction number"



Identifying Subordinate Entities (cont.)

- A weak entity set must have a many-to-one relationship to a distinct entity set
- **Visualization:** (distinguishing an identifying relationship)

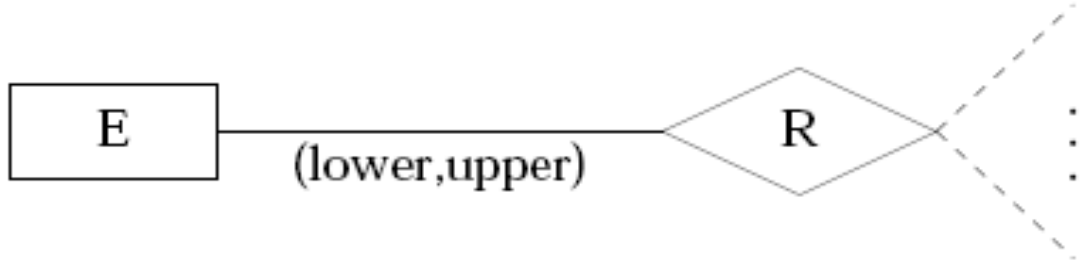


- **Discriminator** of a weak entity set: set of attributes that distinguish subordinate entities of the set, for a particular dominant entity
- **Primary key** for a weak entity set: discriminator + primary key of entity set for dominating entities

General Cardinality Constraints

General cardinality constraints determine lower and upper bounds on the number of relationships of a given relationship set in which a component entity may participate

Visualization:



Example:

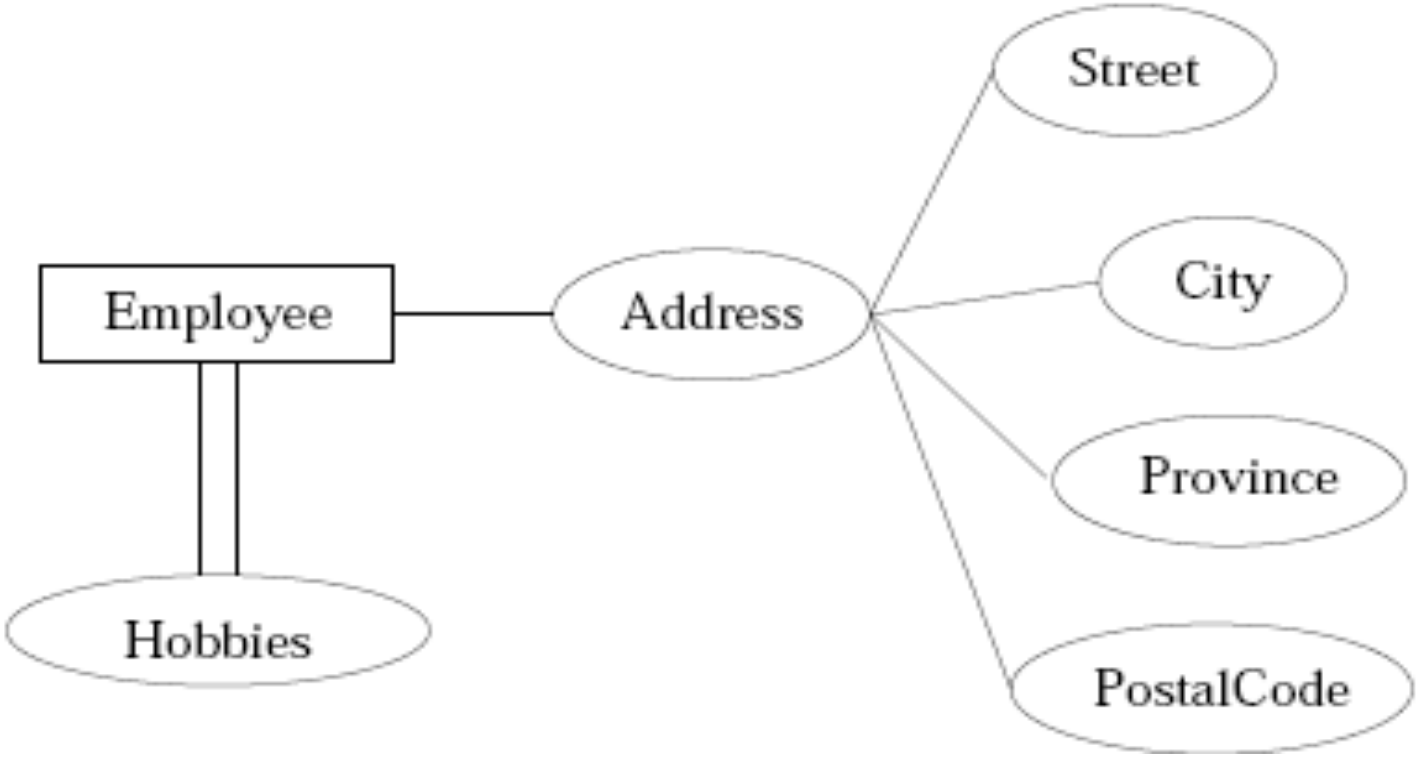
Extension to E-R Modeling

- Structured Attributes
- Aggregation
- Specialization
- Generalization
- Disjointness

Structured Attributes

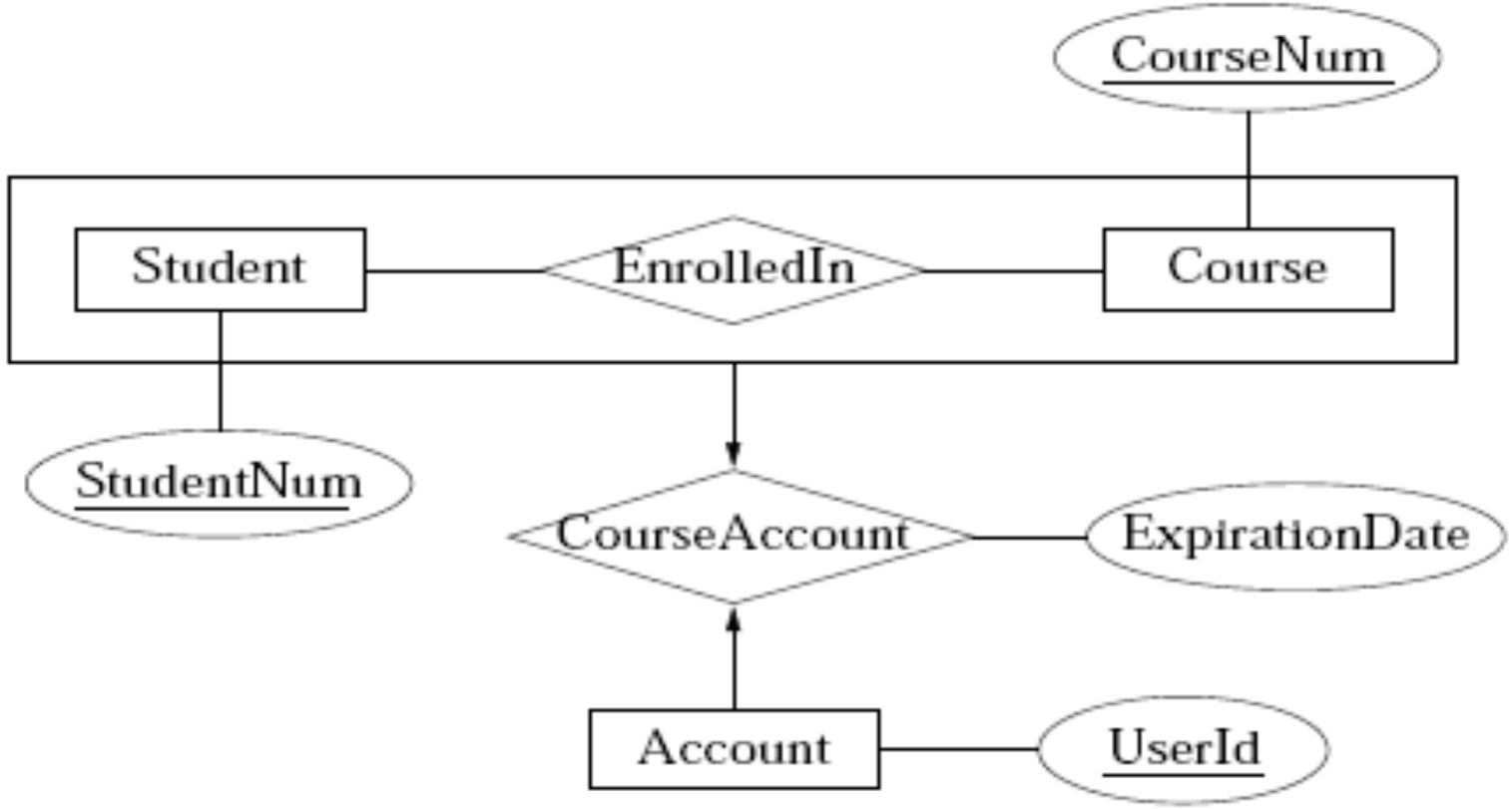
- Composite attributes: composed of fixed number of other attributes
- Multi-valued attributes: attributes that are set valued

Example:



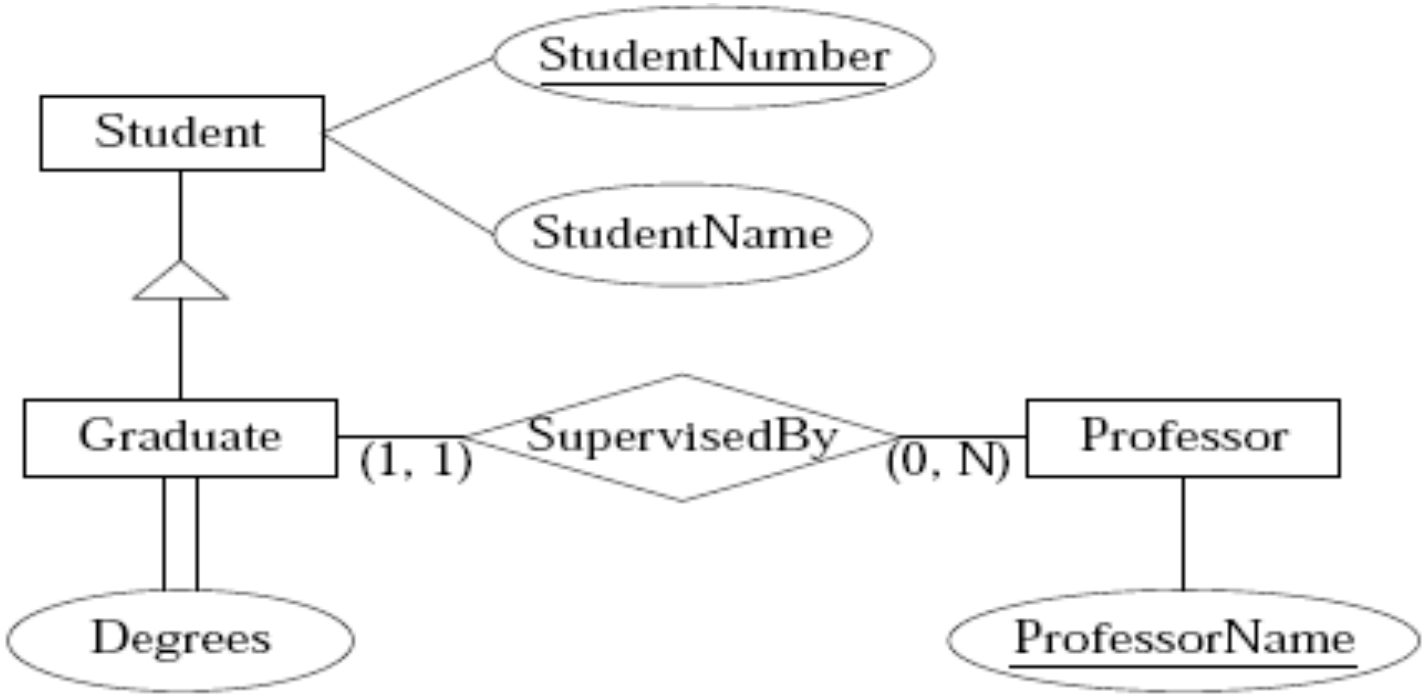
Aggregation

- Relationships can be viewed as higher-level entities
- **Example:** “Accounts are assigned to a given student enrollment.”



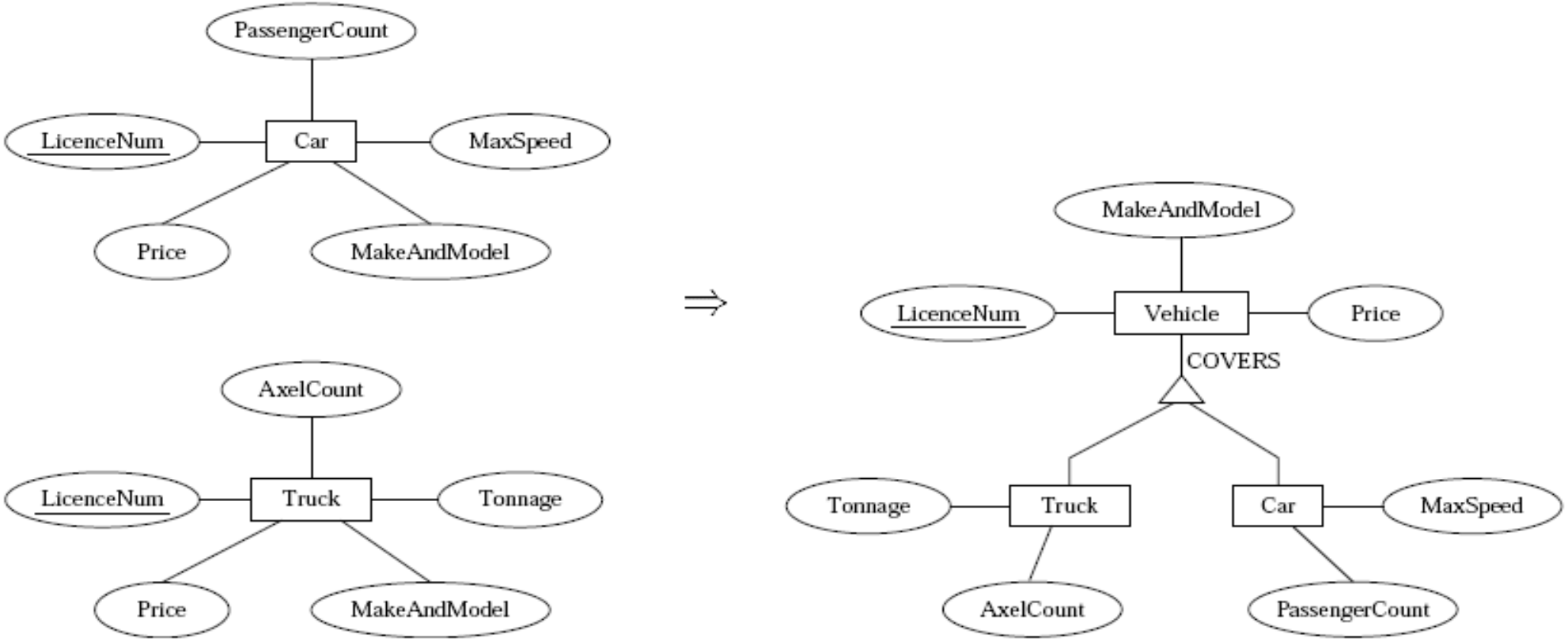
Specialization

- A specialized kind of entity set may be derived from a given entity set
- Example: “Graduate students are students who have a supervisor and a number of degrees.”



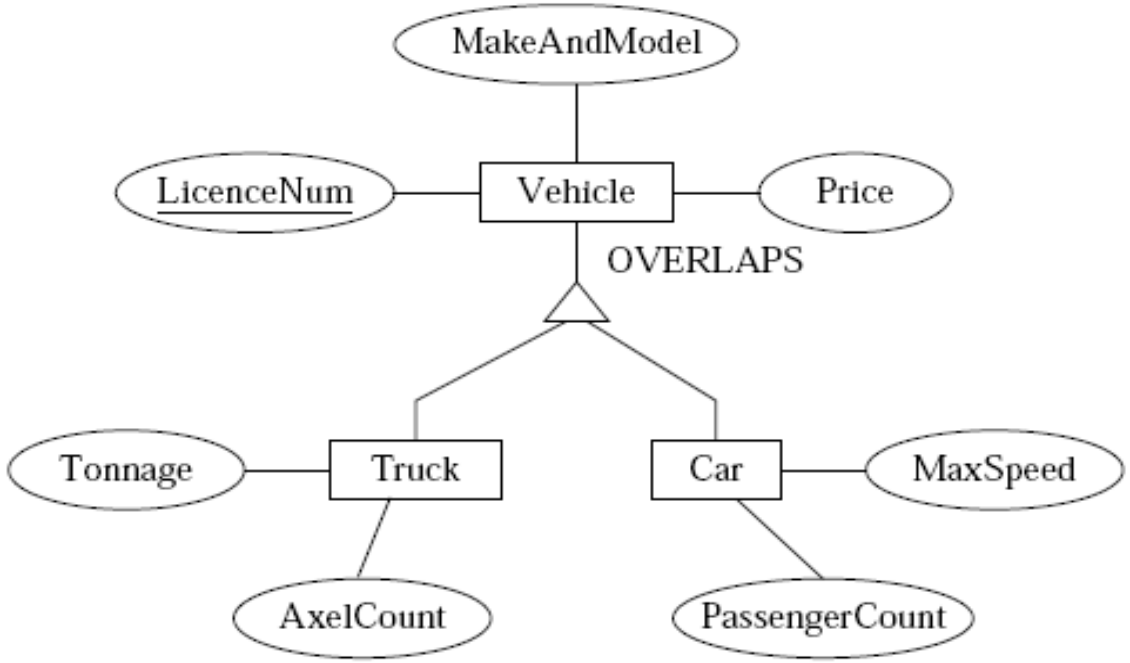
Generalization

- Several entity sets can be abstracted by a more general entity set
- **Example:** “A vehicle abstracts the notion of car and truck.”



Disjointness

- Specialized entity sets are usually disjoint but can be declared to have entities in common
  - By default, specialized entity sets are disjoint.
  - **Example:** We may decide that nothing is both a car and truck.
  - However, we can declare them to overlap (to accommodate utility vehicles, perhaps).



Designing An E-R Schema

- Usually many ways to design an E-R schema
- Points to consider
  - Use attribute or entity set?
  - Use entity set or relationship set?
  - Degrees of relationships?
  - Extend features?

Attributes or Entity?

Example: Should one model employees' phones by a PhoneNumber attribute, or by a Phone entity related to the Employee entity?

Rules of thumb:

- Is it a separate object?
- Do we maintain information about it?
- Can several of its kind belong to a single entity?
- Does it make sense to delete such an object?
- Can it be missing from some of the entity set's entities?
- Can it be shared by different entities?

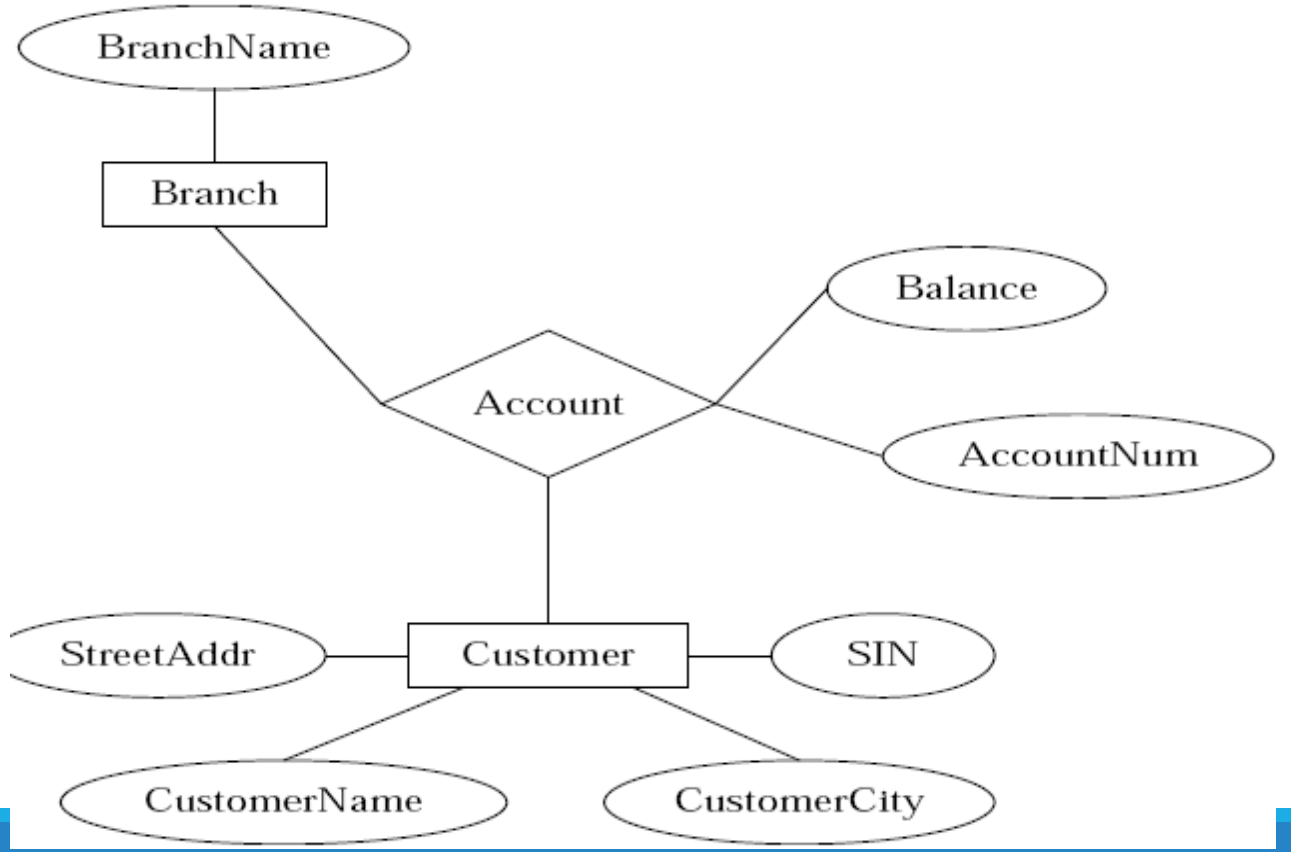
An affirmative answer to any of the above suggests a new entity set.

Entity vs. Attribute

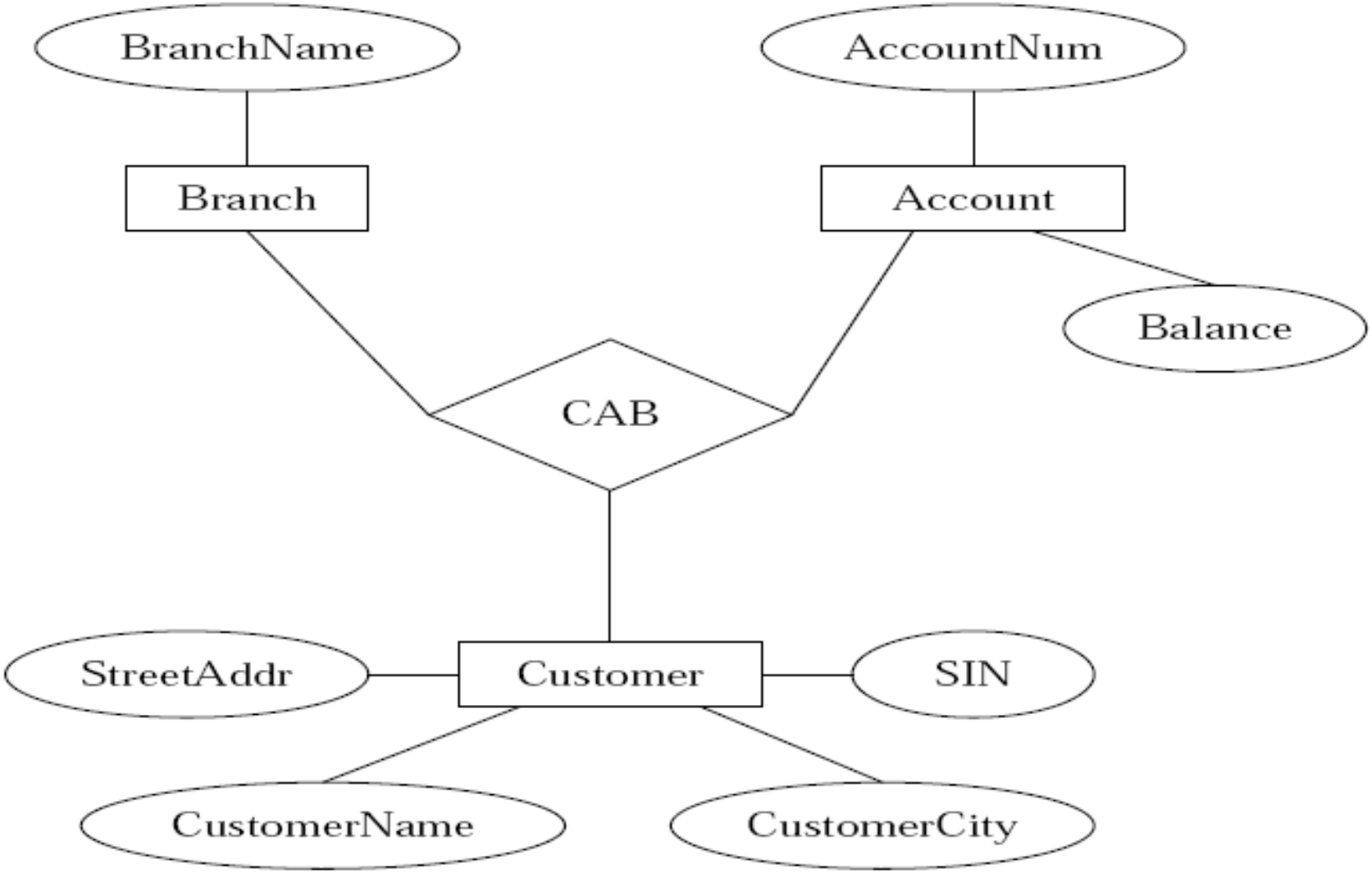
- Should address be an attribute of Employees or an entity (connected to Employees by a relationship)?
- Depends upon the use we want to make of address information, and the semantics of the data:
  - If we have several addresses per employee, address must be an entity (since attributes cannot be set valued).
  - If the structure (city, street, etc.) is important, e.g., we want to retrieve employees in a given city, address must be modeled as an entity (since attribute values are atomic)

Entity Sets or Relationships?

- Instead of representing accounts as entities, we could represent them as relationships



Binary vs. N-ary Relationships?



Design E-R Diagram: A Simple Methodology

1. Recognize entity sets
2. Recognize relationship sets and participating entity sets
3. Recognize attributes of entity and relationship sets
4. Define relationship types and existence dependencies
5. Define general cardinality constraints, keys and discriminators
6. Draw diagram

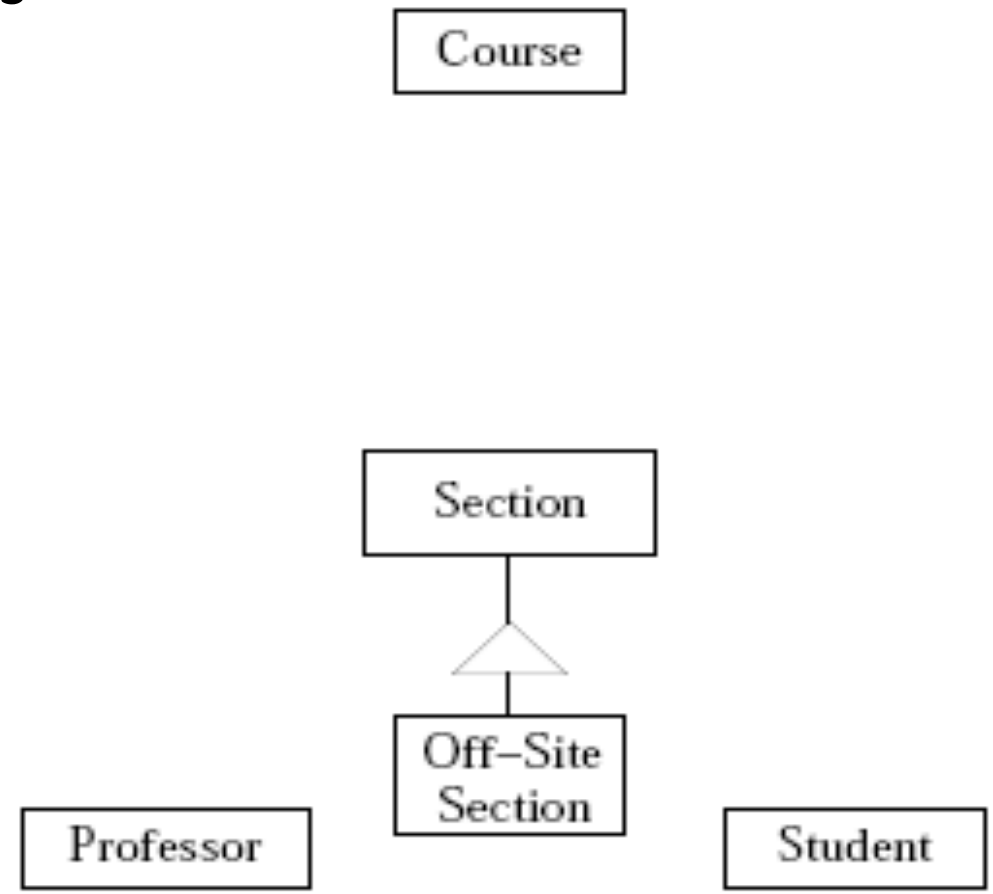
For each step, maintain a log of assumptions motivating the choices, and of restrictions imposed by the choices

Example: A Registrar's Database

- Zero or more sections of a course are offered each term. Courses have names and numbers. In each term, the sections of each course are numbered starting with 1.
- Most course sections are taught on-site, but a few are taught at off-site locations.
- Students have student numbers and names
- Each course section is taught by a professor. A professor may teach more than one section in a term, they are always sections of the same course. Some professors do not teach every term.
- Up to 50 students may be registered for a course section. Sections with 5 or fewer students are cancelled.
- A student receives a mark for each course in which they are enrolled. Each student has a cumulative grade point average (GPA) which is calculated from all course marks the student has received

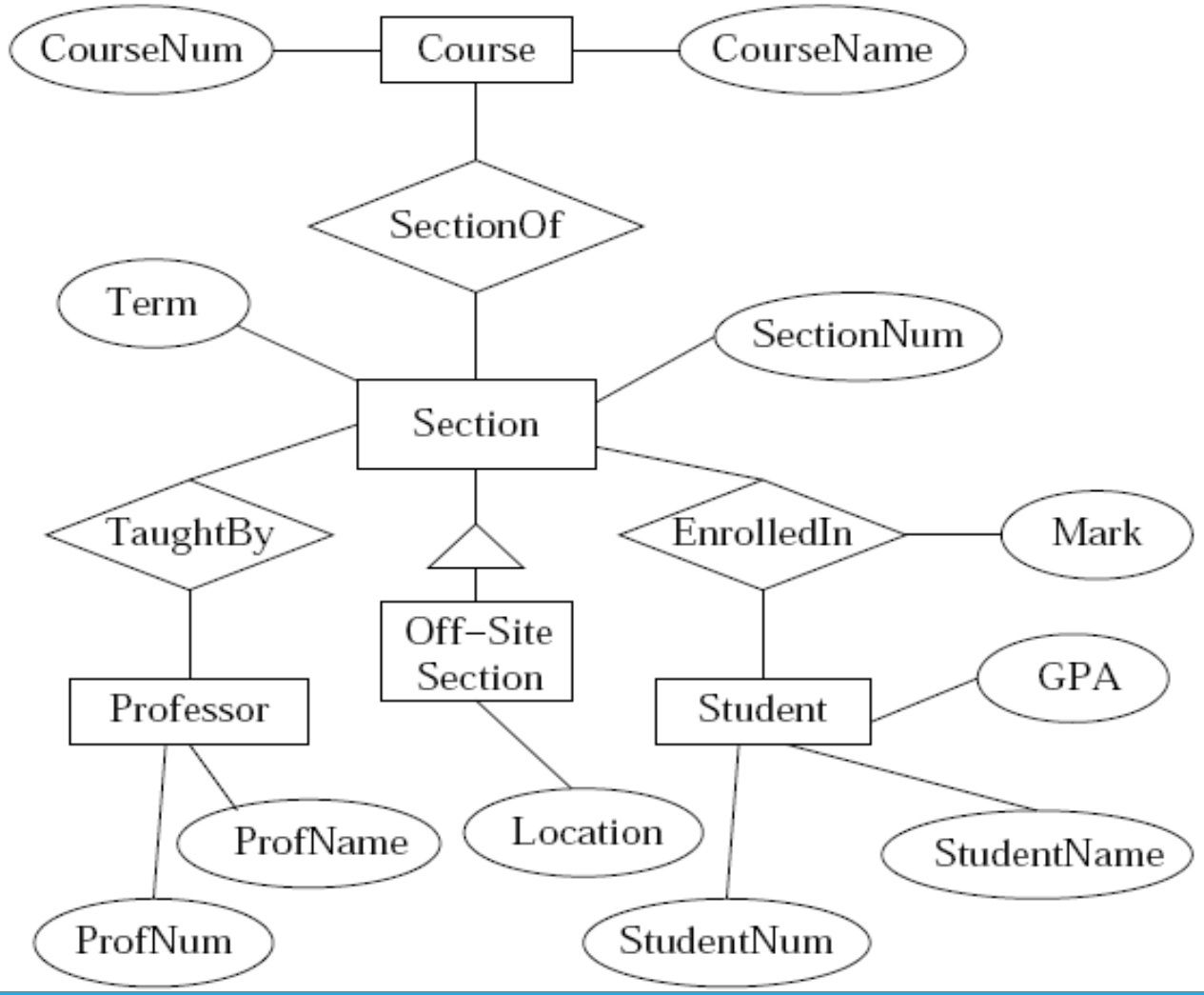
Example: A Registrar's Database (cont.)

(1) Define entities



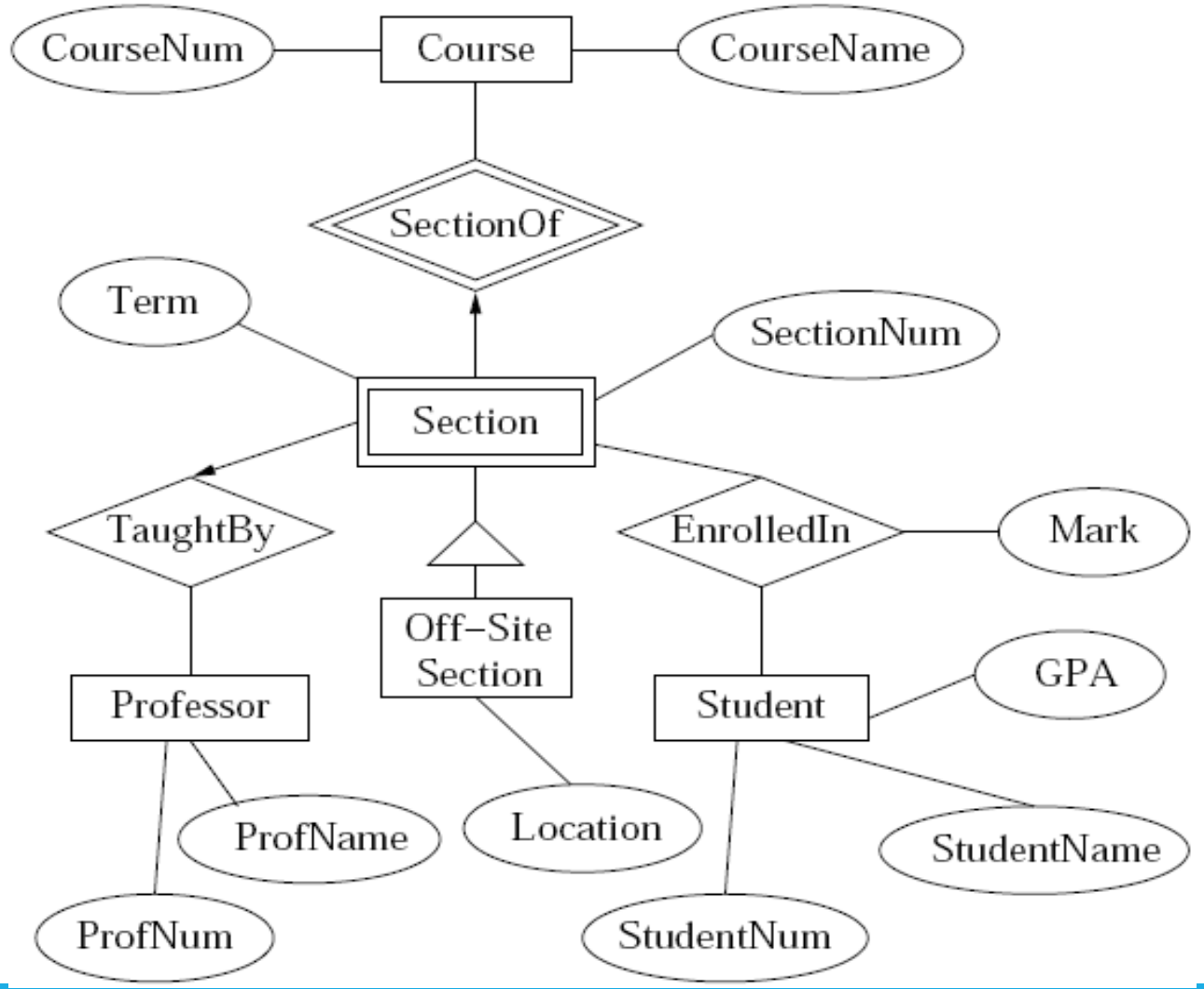
Example: A Registrar's Database (cont.)

(2) Define relationships
(3) Define attributes



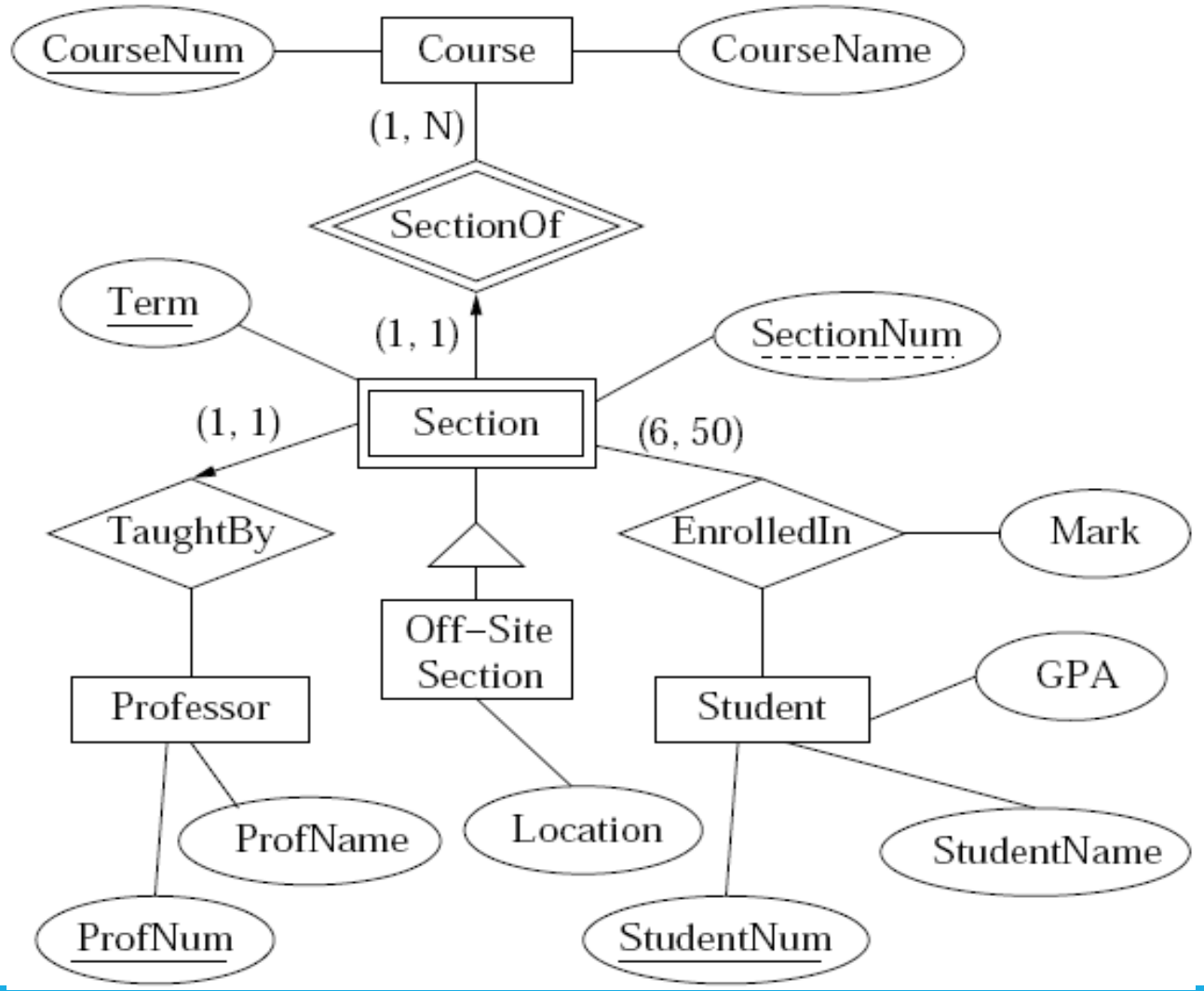
Example: A Registrar's Database (cont.)

(4) Define relationship types and dependencies



Example: A Registrar's Database (cont.)

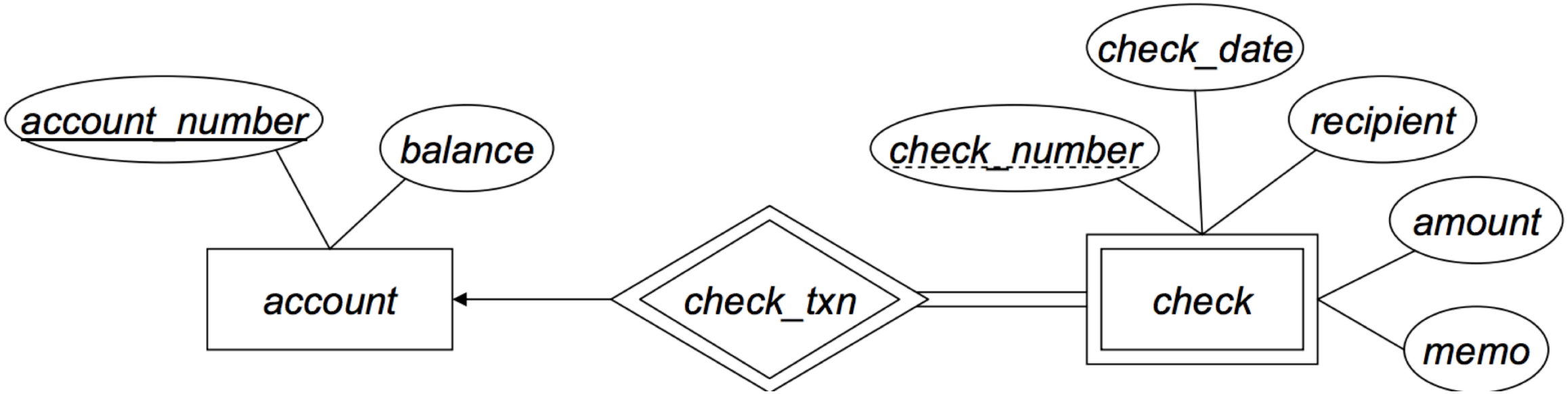
(5) Define constraints



Reference

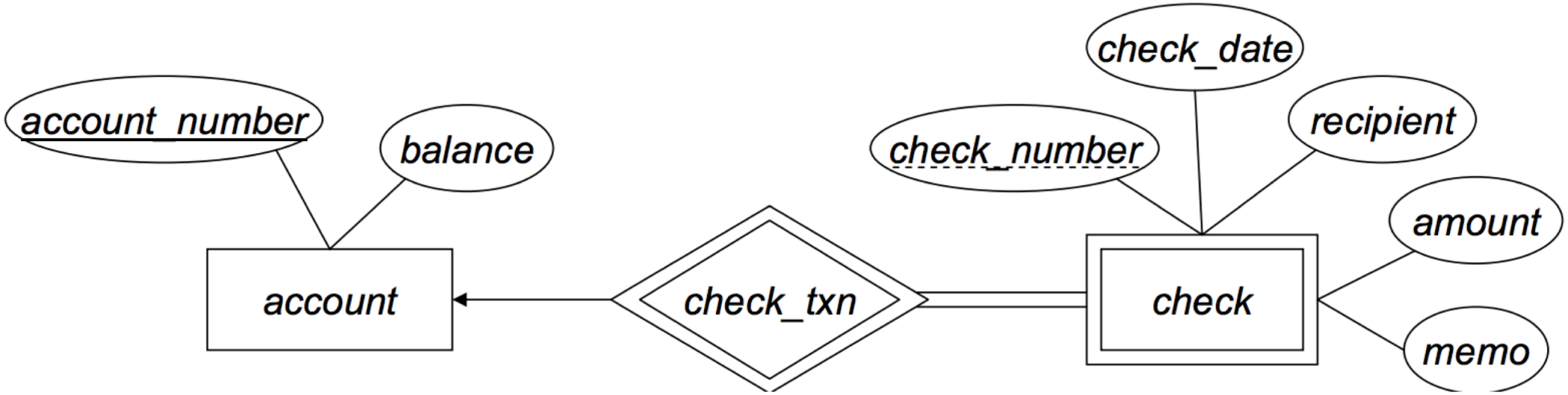
1. Ramakrishnan R, Gehrke J., Database management systems, 3rd ed., New York (NY): McGraw-Hill, 2003.

Weak Entity-Set Example

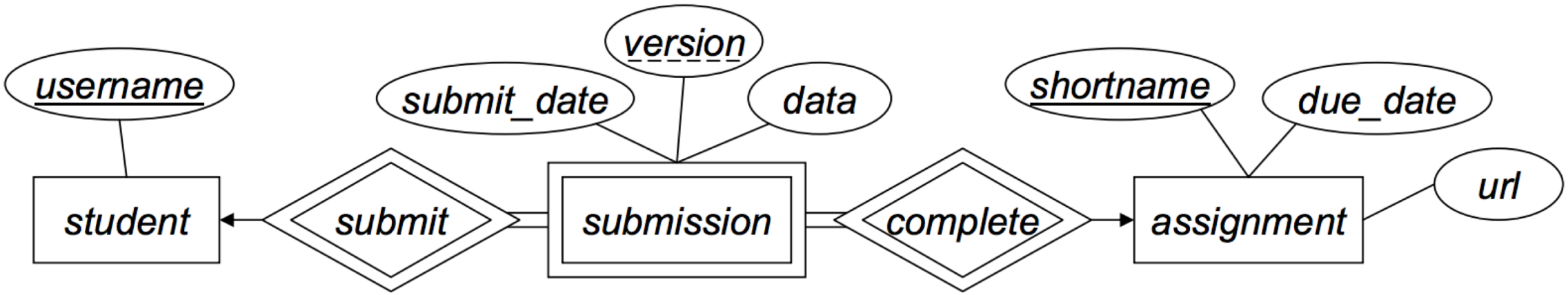


Weak Entity-Set Example

- account schema: account (account_number, balance)
- check schema: – Discriminator is check_number
  - Primary key for check is: (account_number, check_number)
  - check (account_number, check_number, check_date, recipient, amount, memo)

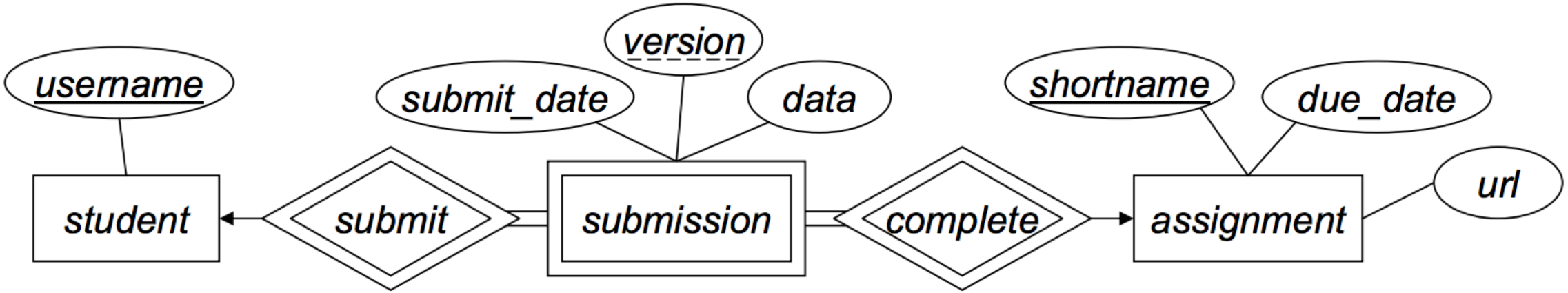


Weak Entity-Set Example (2)



Weak Entity-Set Example (2)

- Schemas for strong entity-sets:
  - student (username)
  - assignment (shortname, due_date, url)
- Schema for submission weak entity-set:
  - submission (username, shortname, version, submit_date, data)

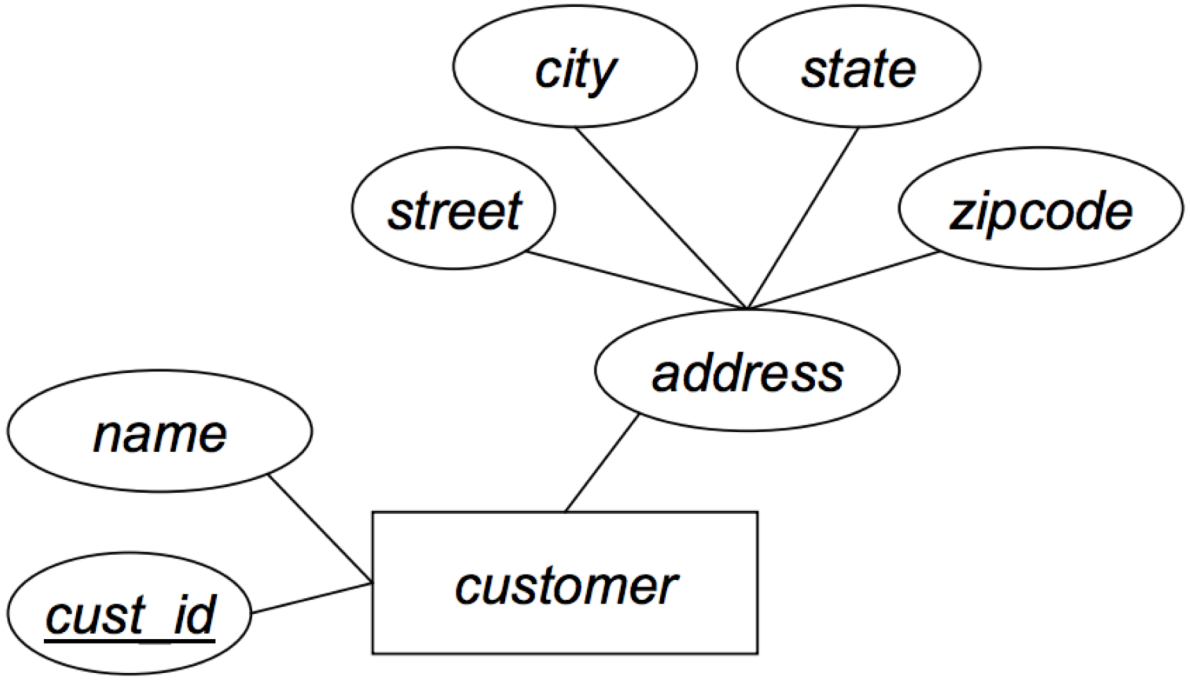


Composite Attribute Example

- Customers with addresses

Composite Attribute Example

- Customers with addresses
- Each component of address becomes a separate attribute customer(cust_id, name, street, city, state, zipcode)

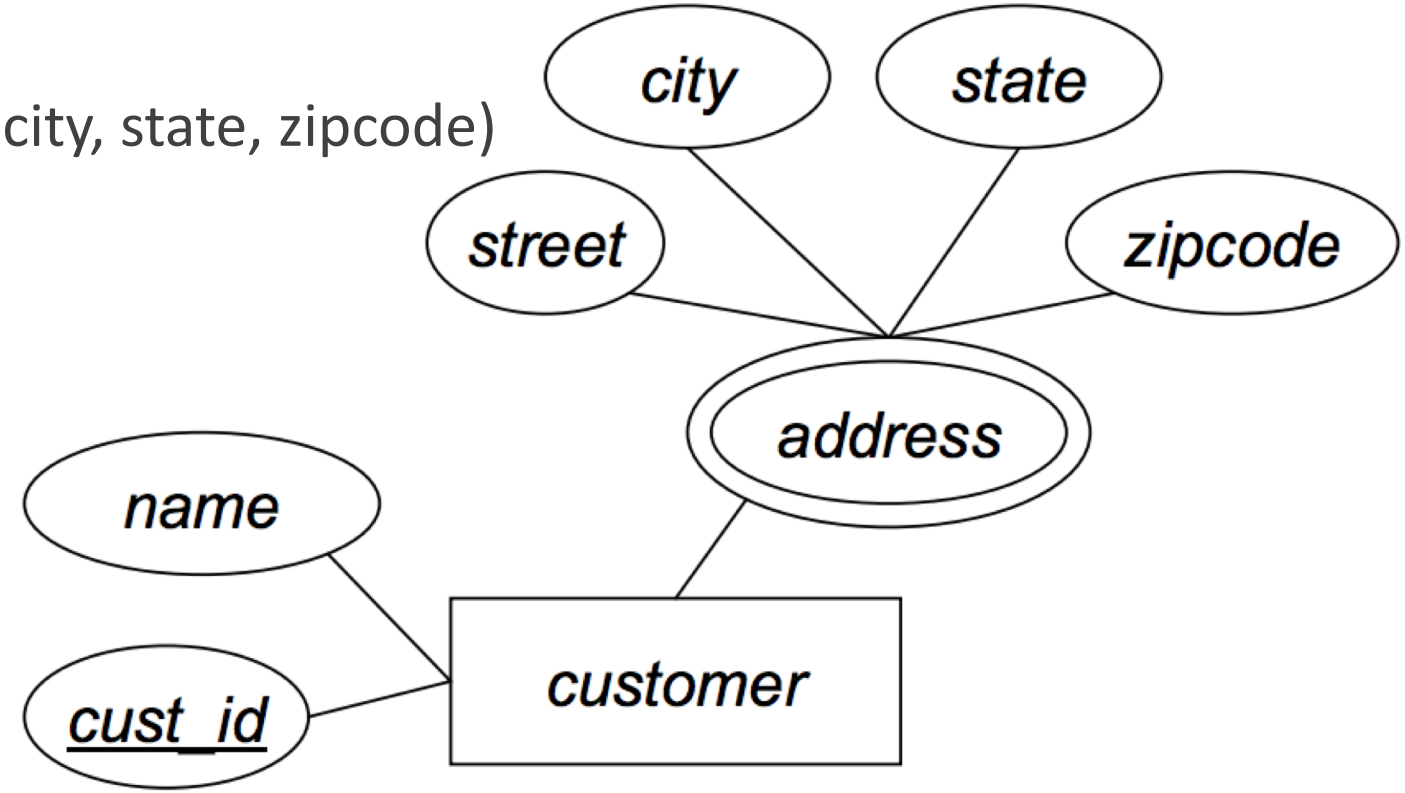


Multivalued Attribute Example

- Customers with multiple addresses

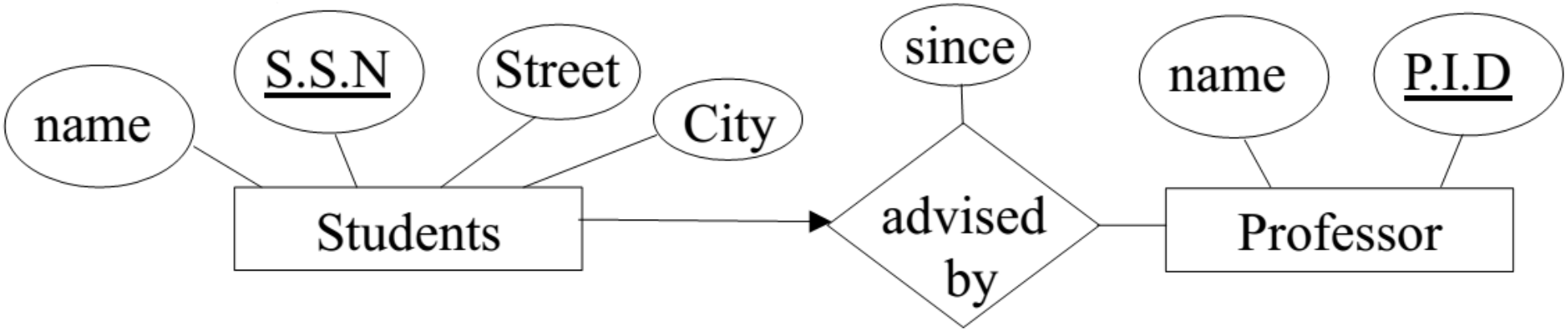
Multivalued Attribute Example

- Customers with multiple addresses
- Create separate relation to store each address
  - customer(cust_id, name)
  - cust_addrs (cust_id, street, city, state, zipcode)



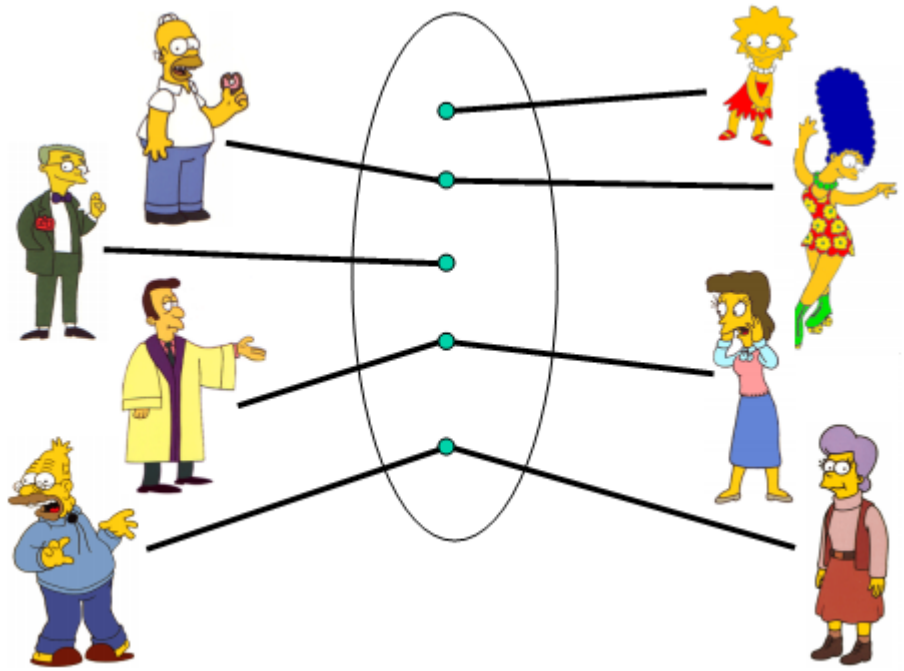
Many-to-One Relationship

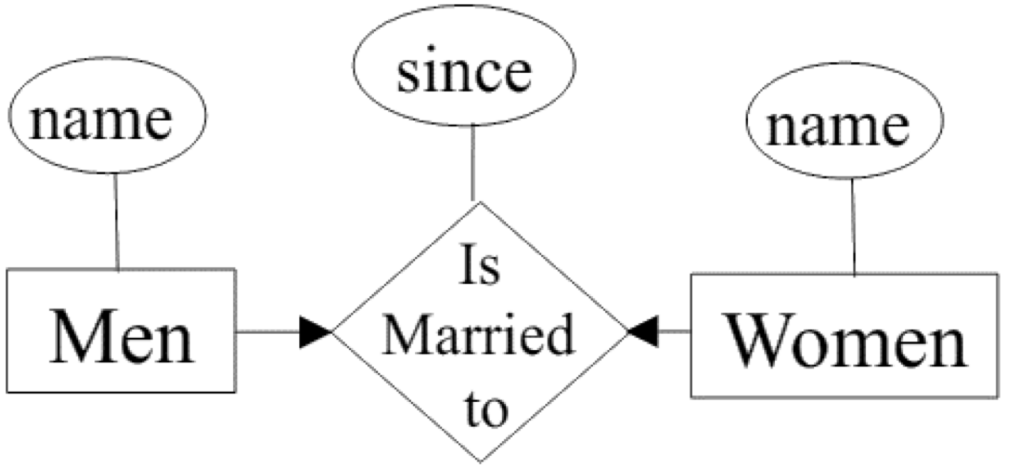
- This is an example of a many-to-one constraints, that is, many students can be advised by one professor, but each student can only have (at most) one advisor. We can represent this with an arrow as shown below. (Arrow's Head :1 , Arrow's Tail: many)



One-to-One Relationship

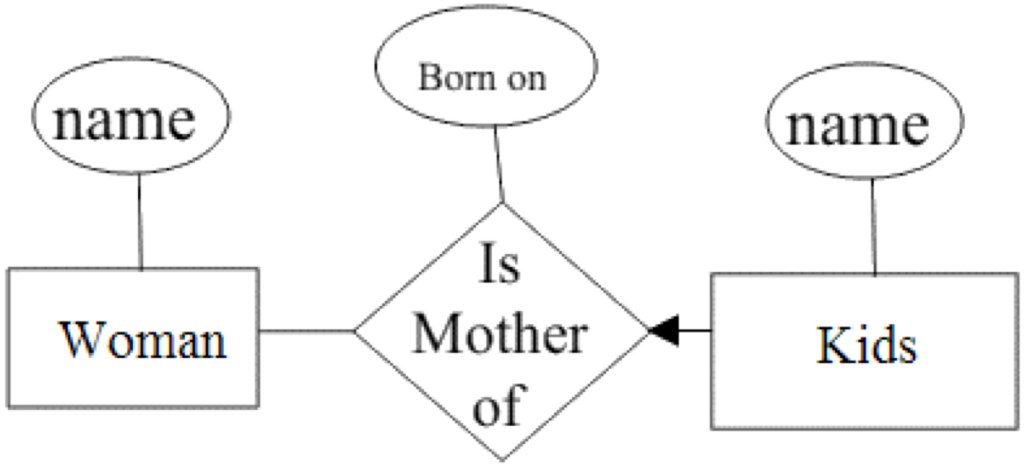
- One-to-one: An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.
- A man may be married to at most one woman, and a woman may be married to at most one man (both men and women can be unmarried)

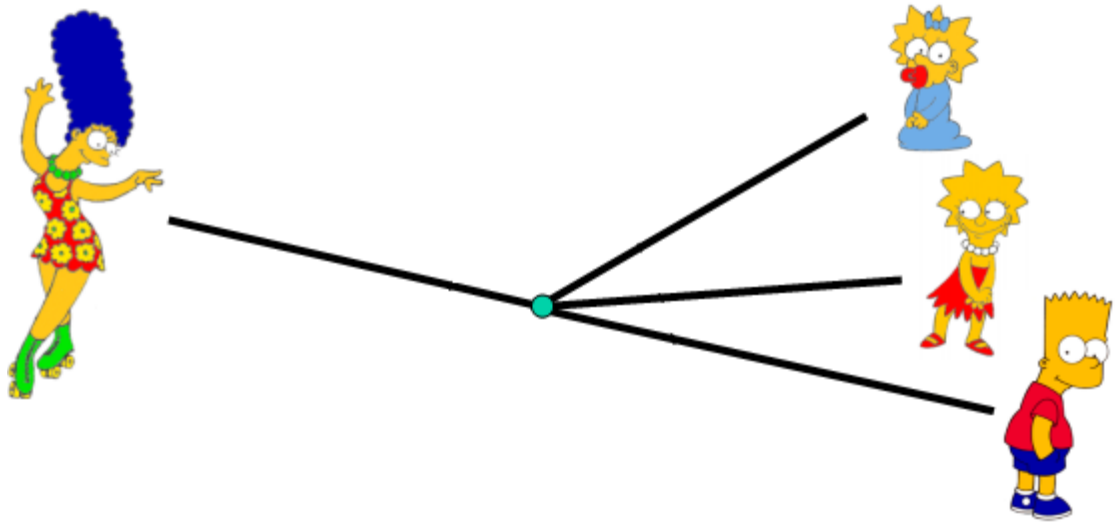




One-to-Many Relationship

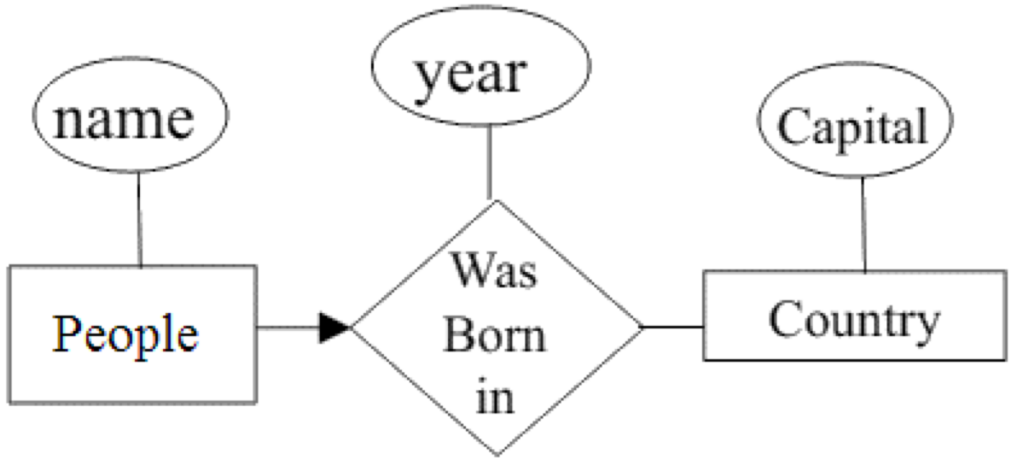
- One-to-many:An entity in A is associated with any number in B. An entity in B is associated with at most one entity in A.
- A woman may be the mother of many (or no) children. A person may have at most one mother.

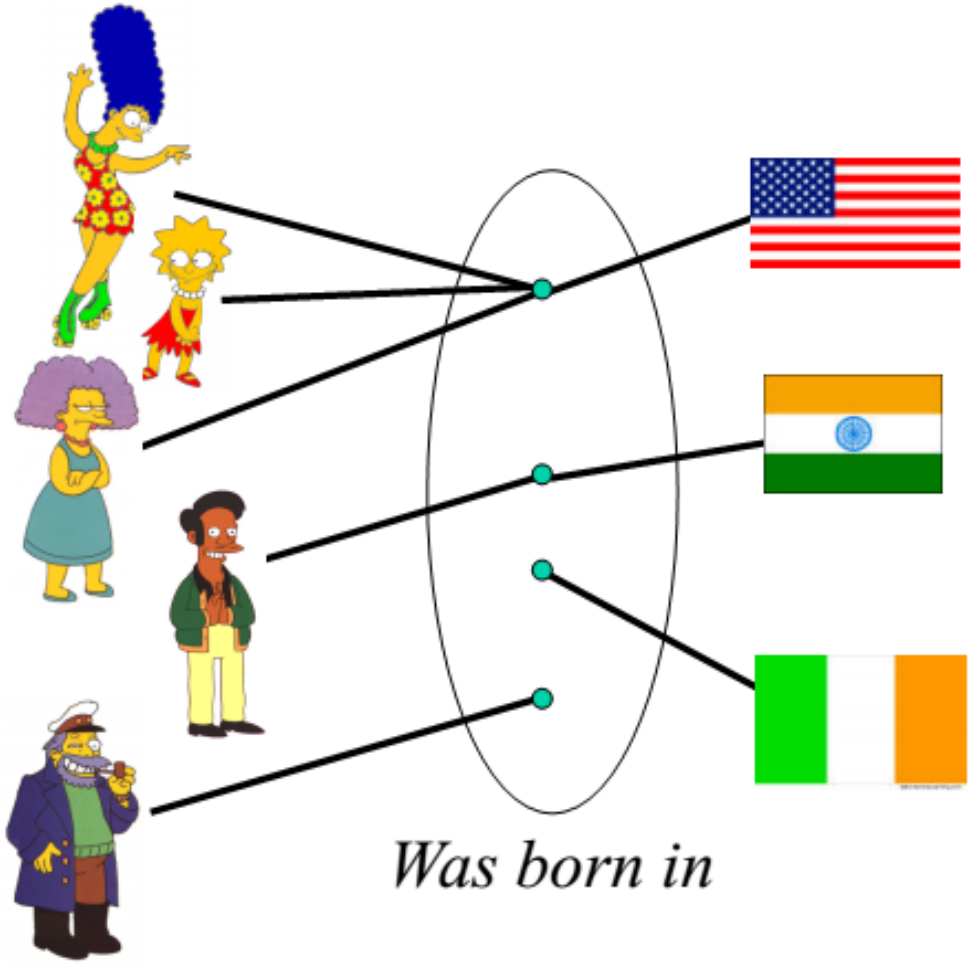




Many-to-One Relationship

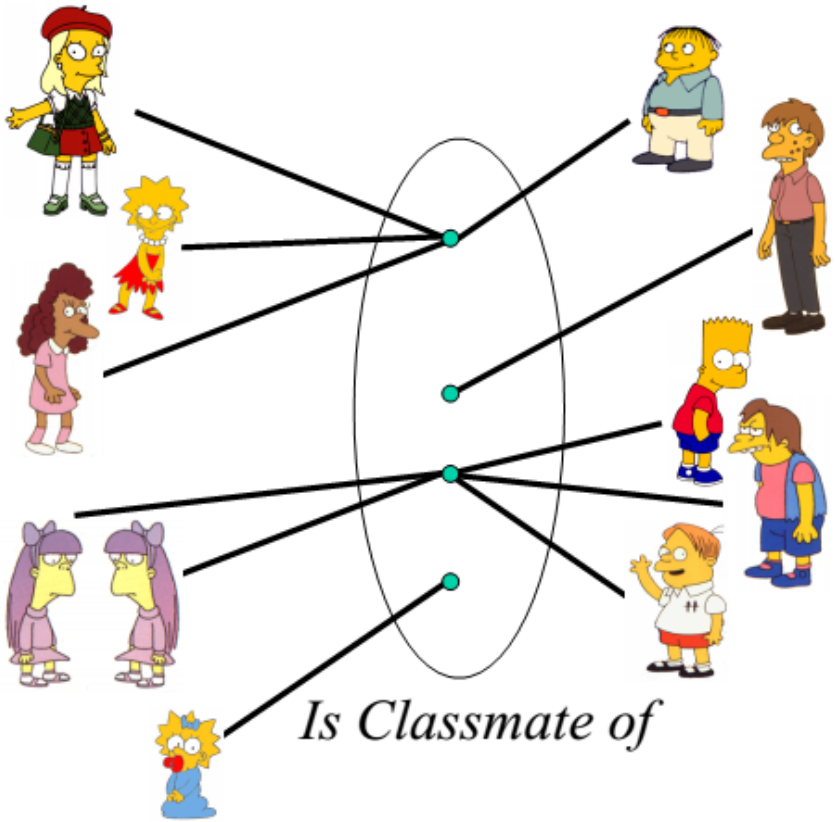
- Many-to-one: An entity in A is associated with at most one entity in B. An entity in B is associated with any number in A.
- Many people can be born in any county, but any individual is born in at most one country.

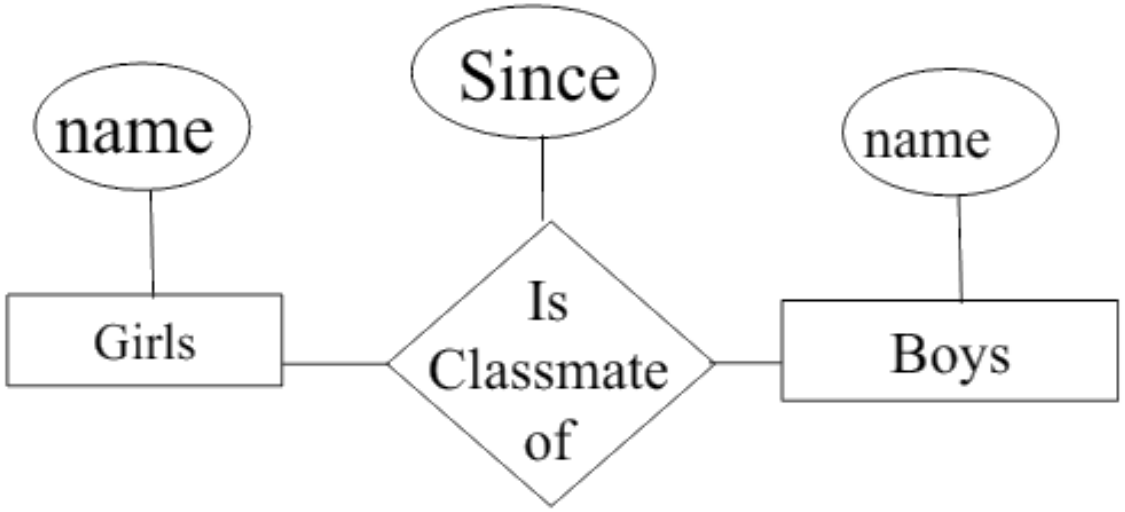




Many-to-Many Relationship

- Many-to-many:Entities in A and B are associated with any number from each other





References

Jessica Lin, “The Entity-Relationship (ER) Model”, http://www.cs.gmu.edu/~jessica/cs450_f11/cs450_ER1.pdf