The Entity-Relationship (E-R) Model

- An E-R model is used to describe an enterprise that is to be supported by a database management system.
- The enterprise is described as a collection of *entities* and their attributes, and a collection of *relationships* among those entities.
- An E-R model is represented graphically as an *E-R diagram*.

An E-R model can be translated into a relational database schema which, after additional refinement, can serve as the conceptual schema for the underlying database system.

Entities and Relationships

entity: a *distinguishable* object

entity set: set of entities of same type

attribute: a property of an entity

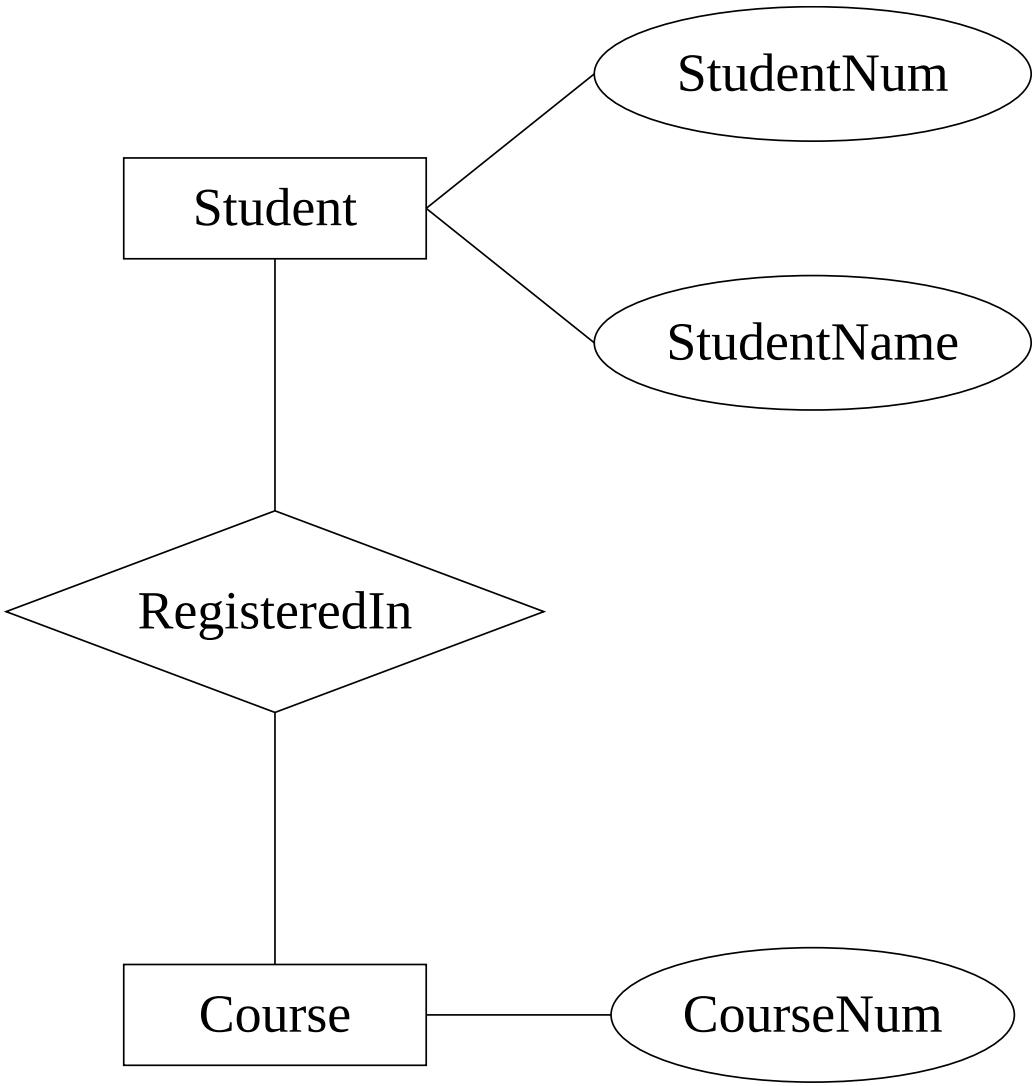
- all entities in an entity set have the same attributes
- each attribute has a name and an associated domain, which specifies the set of permitted values for that attribute

relationship: represents some connection between entities

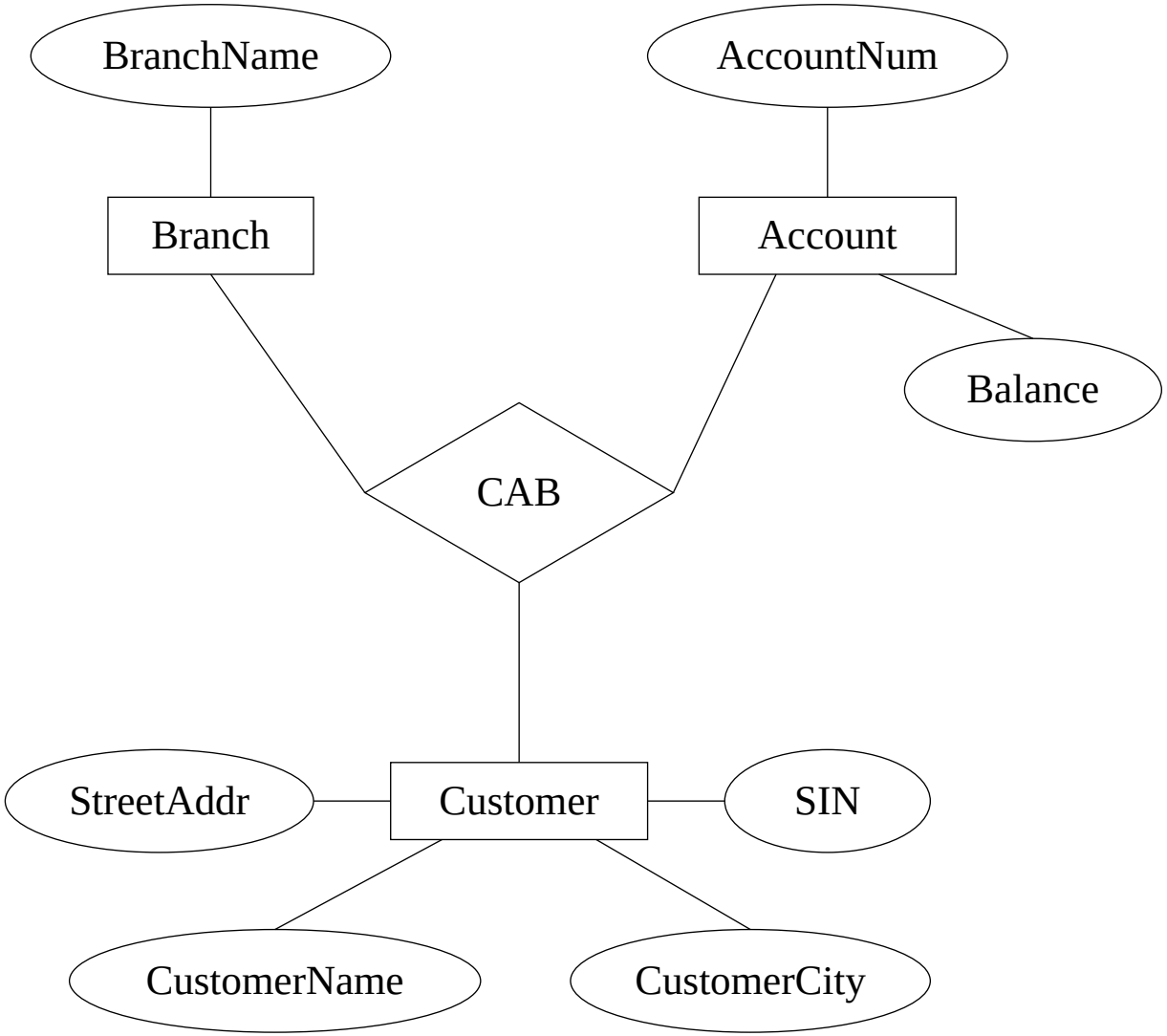
relationship set: set of relationships between entities of one entity set and entities of another

- A relationship can only exist if the entities that it relates exist.

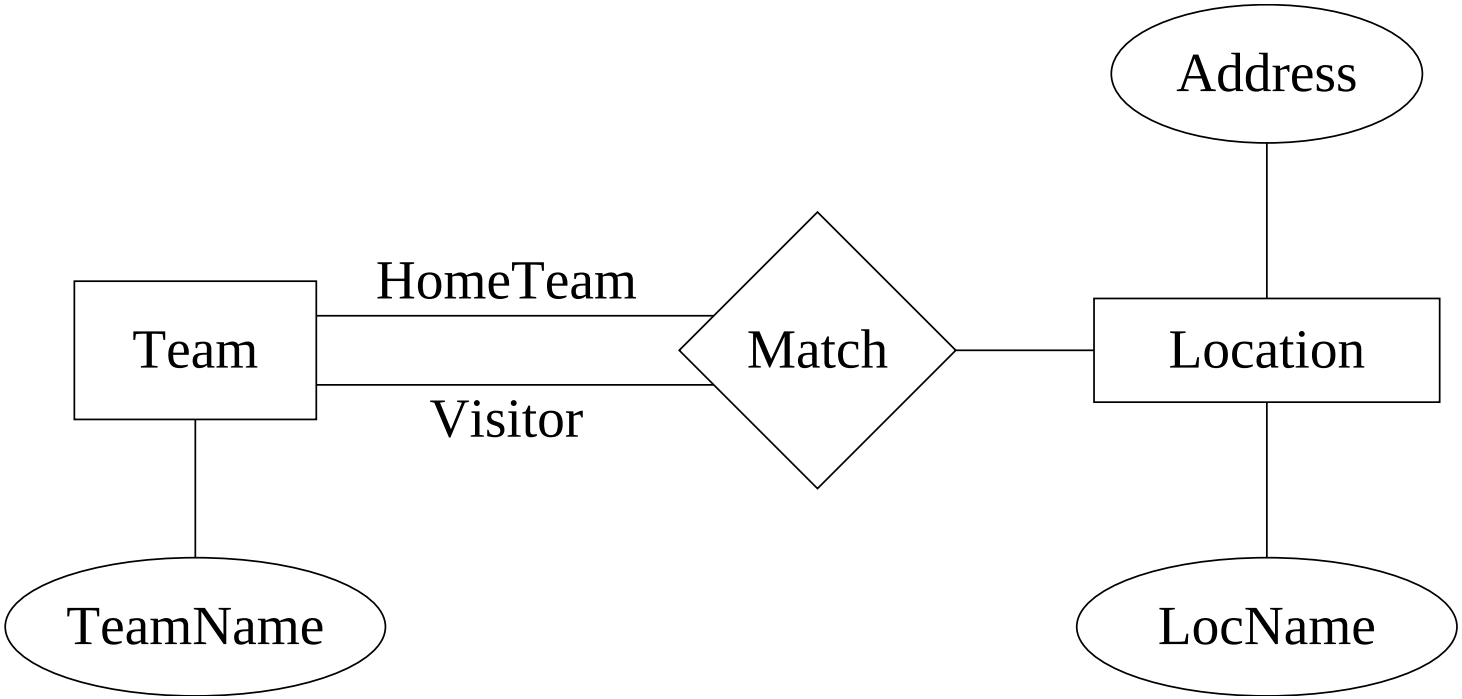
E-R Diagram Example



Another E-R Diagram Example

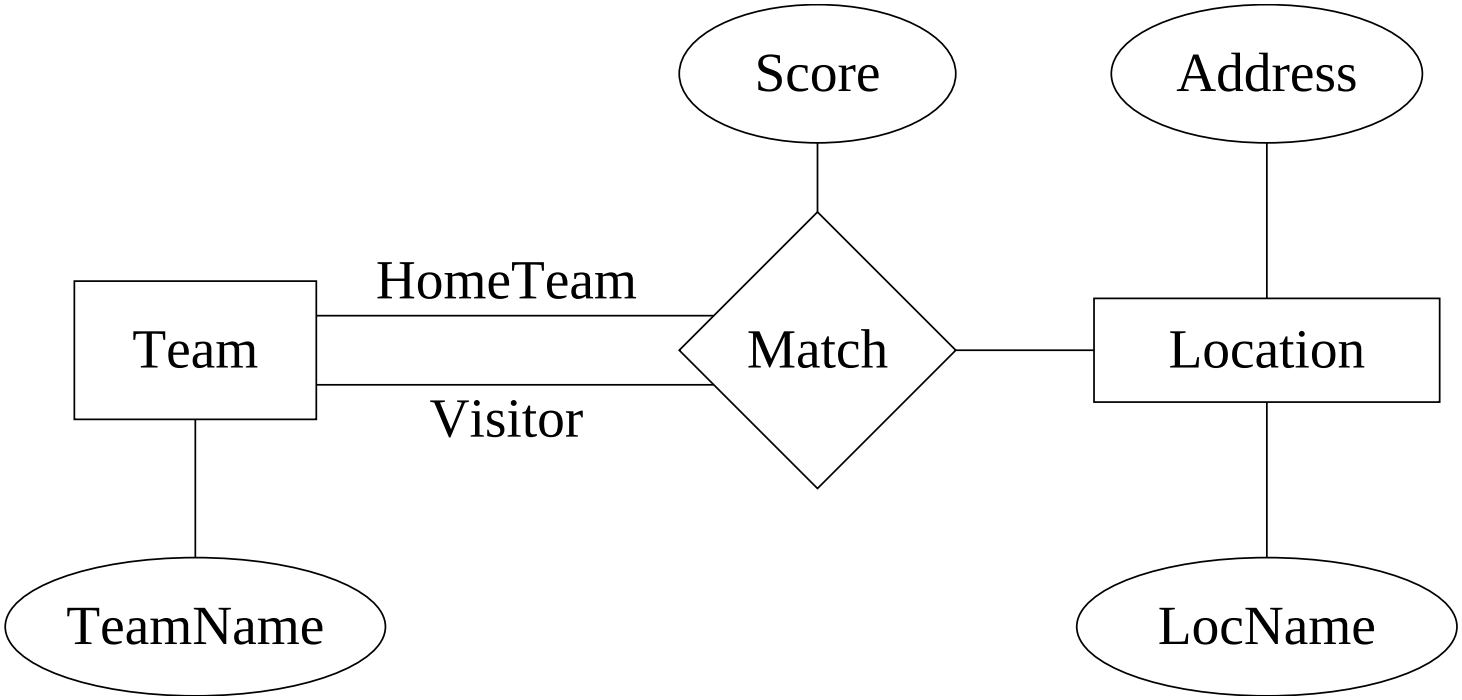


Recursive Relationships and Role Names



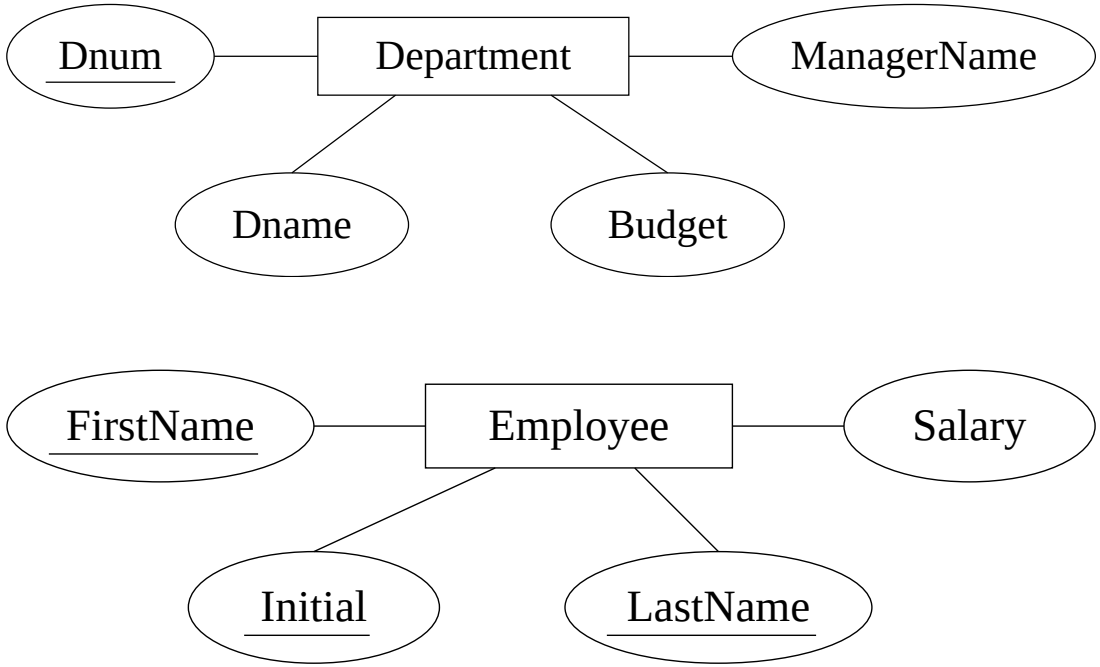
Role labels are needed whenever an entity set has multiple functions in a relationship set.

Relationship Attributes



Relationships, like entities, may have attributes.

Primary Keys in E-R Diagrams



Entities in an entity set must be distinguishable by the values of their key attributes. No two entities in the set may have the same key values.

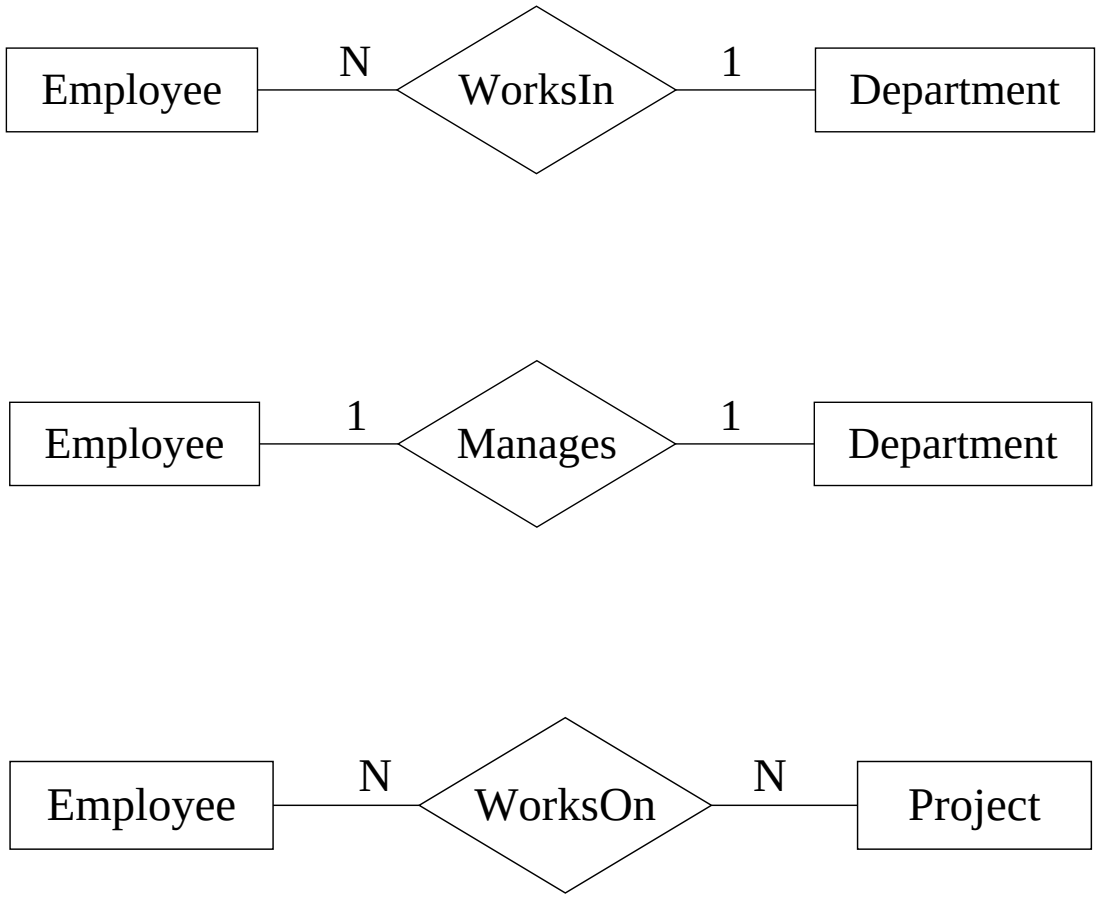
Constraints: Binary Relationship Types

many-to-one (N:1): each entity in set A can be related to at most one entity in set B, but an entity in B may be related to many entities in A

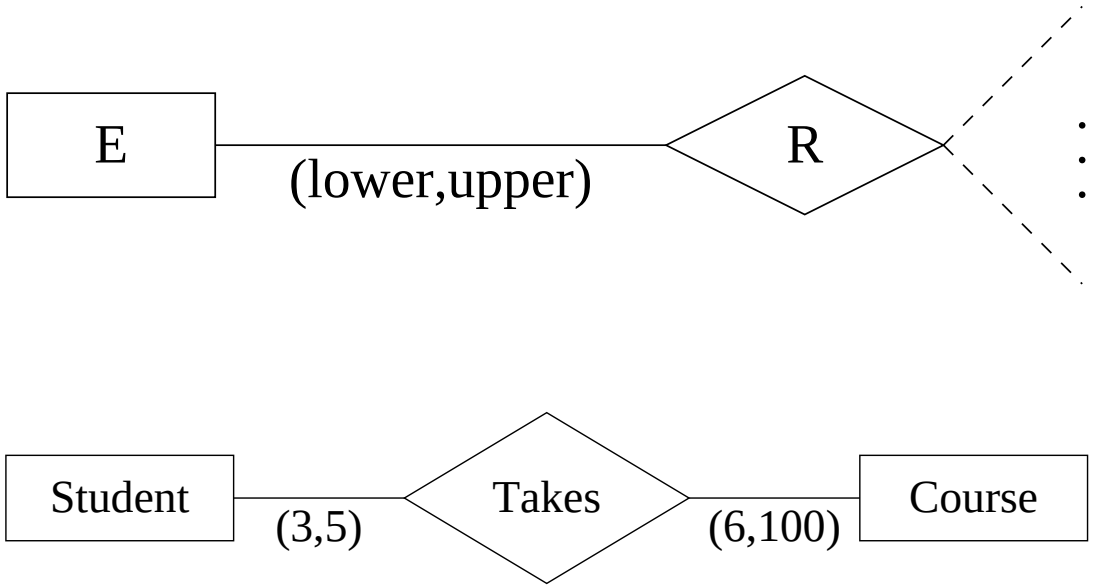
one-to-one (1:1): each entity in set A can be related to at most one entity in set B, and vise versa

many-to-many (N:N): an entity in set A can be related to many entities in set B, and vice versa

Binary Relationship Types in E-R Diagrams



General Cardinality Constraints

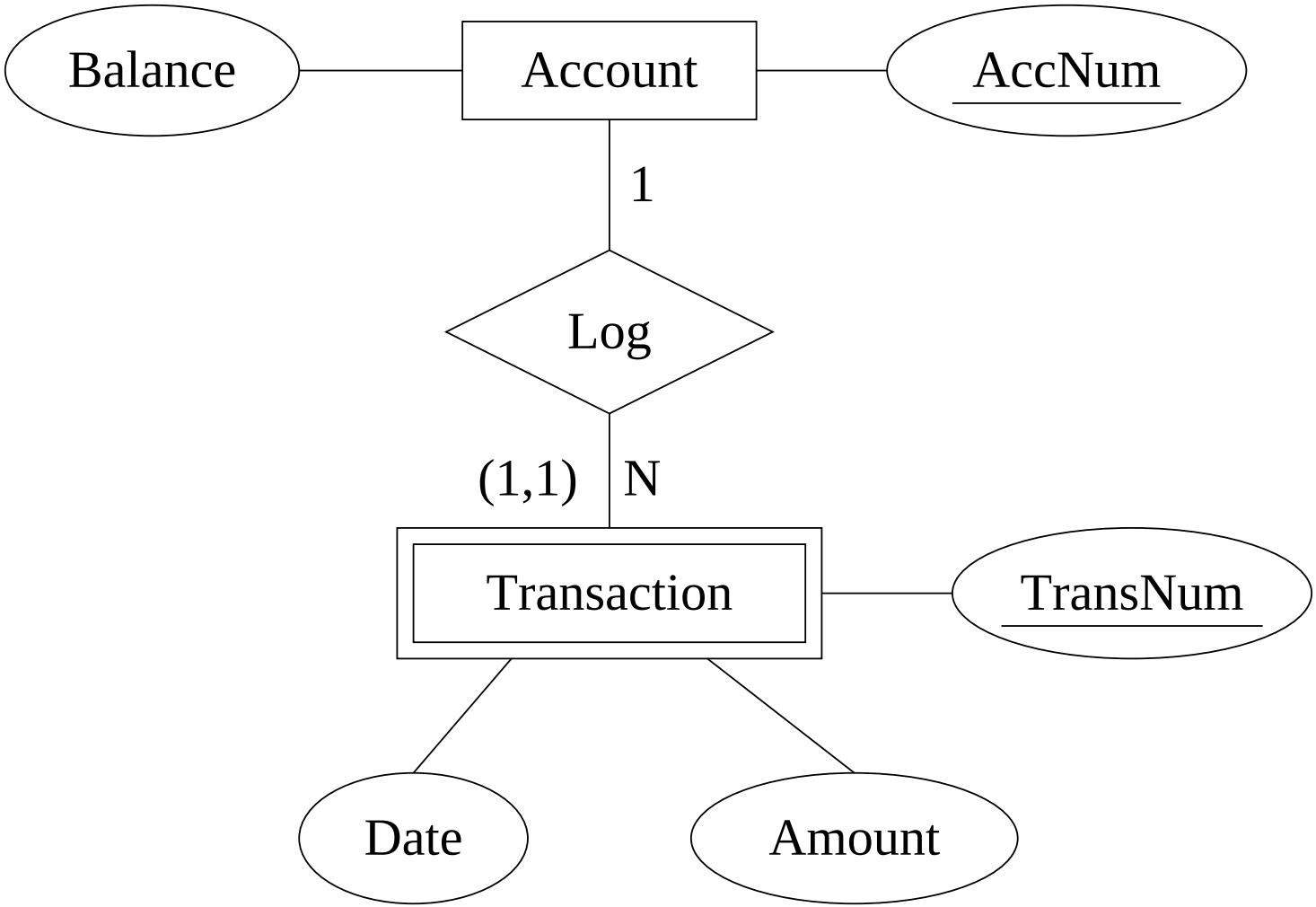


General cardinality constraints define lower and upper bounds on the number of relationships of a given relationship set in which an entity may participate

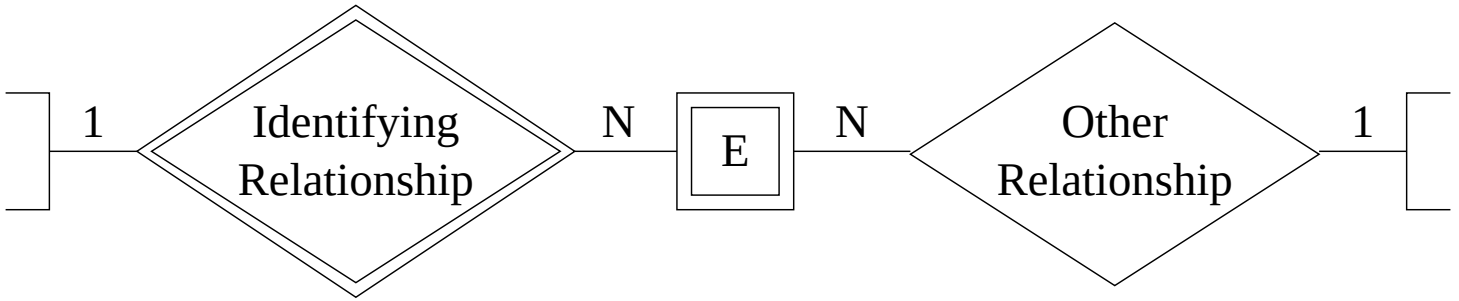
Constraints: Existence Dependencies

- Sometimes the existence of an entity depends on the existence of another entity. The former is called the *subordinate entry*, the latter is called the *dominant entry*.
- A *weak entity set* contains subordinate entities. A *strong entity set* contains dominant entities.
- A weak entity set must have an N:1 or 1:1 relationship to a strong entity set. This is called the *identifying relationship* of the weak entity set.
- The *discriminator* of a weak entity set is a set of attributes that can be used to distinguish among several entities that are subordinate to the same dominant entity. A discriminator is not the same things as a key. (Why?)

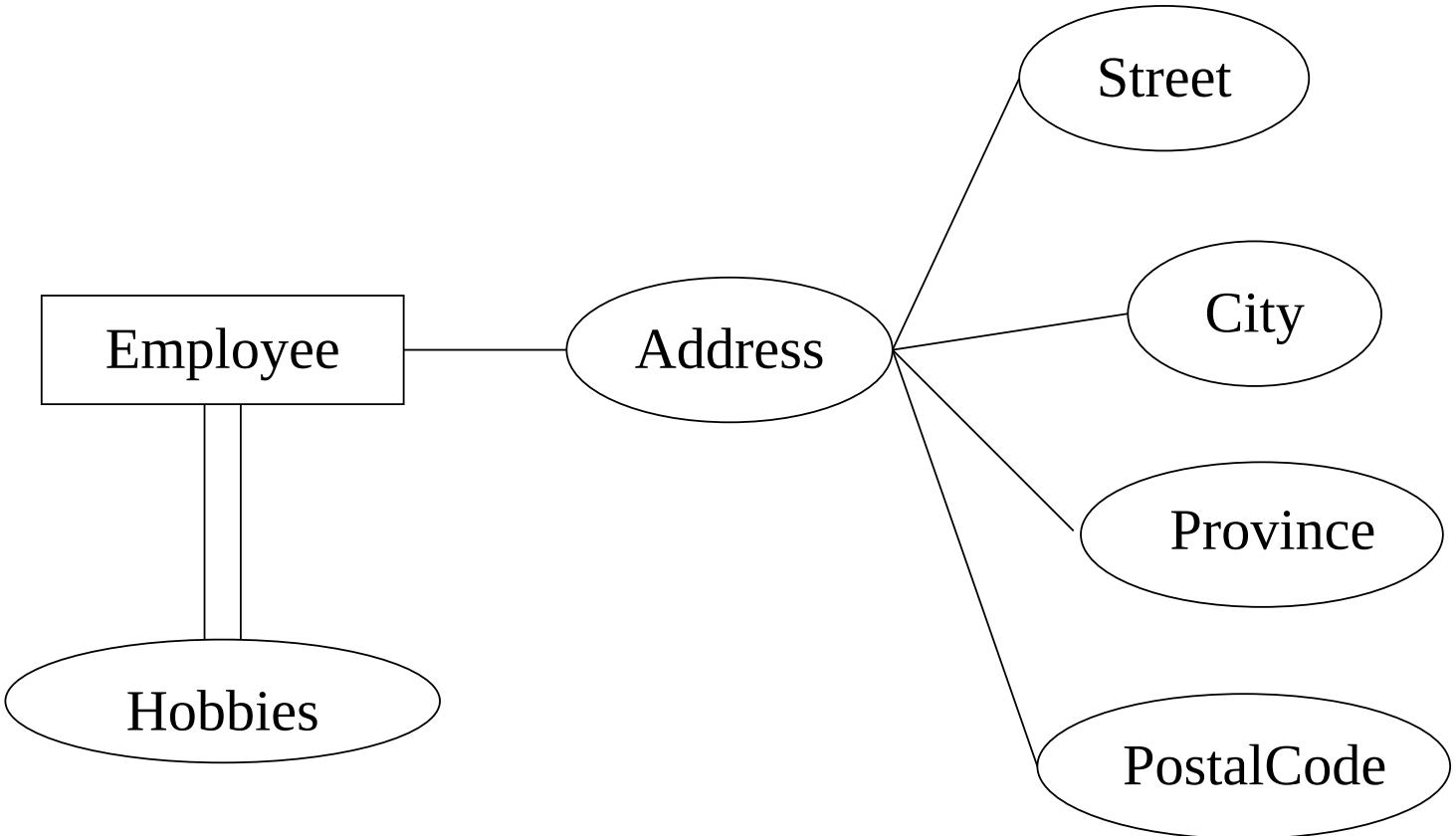
Existence Dependencies in E-R Diagrams



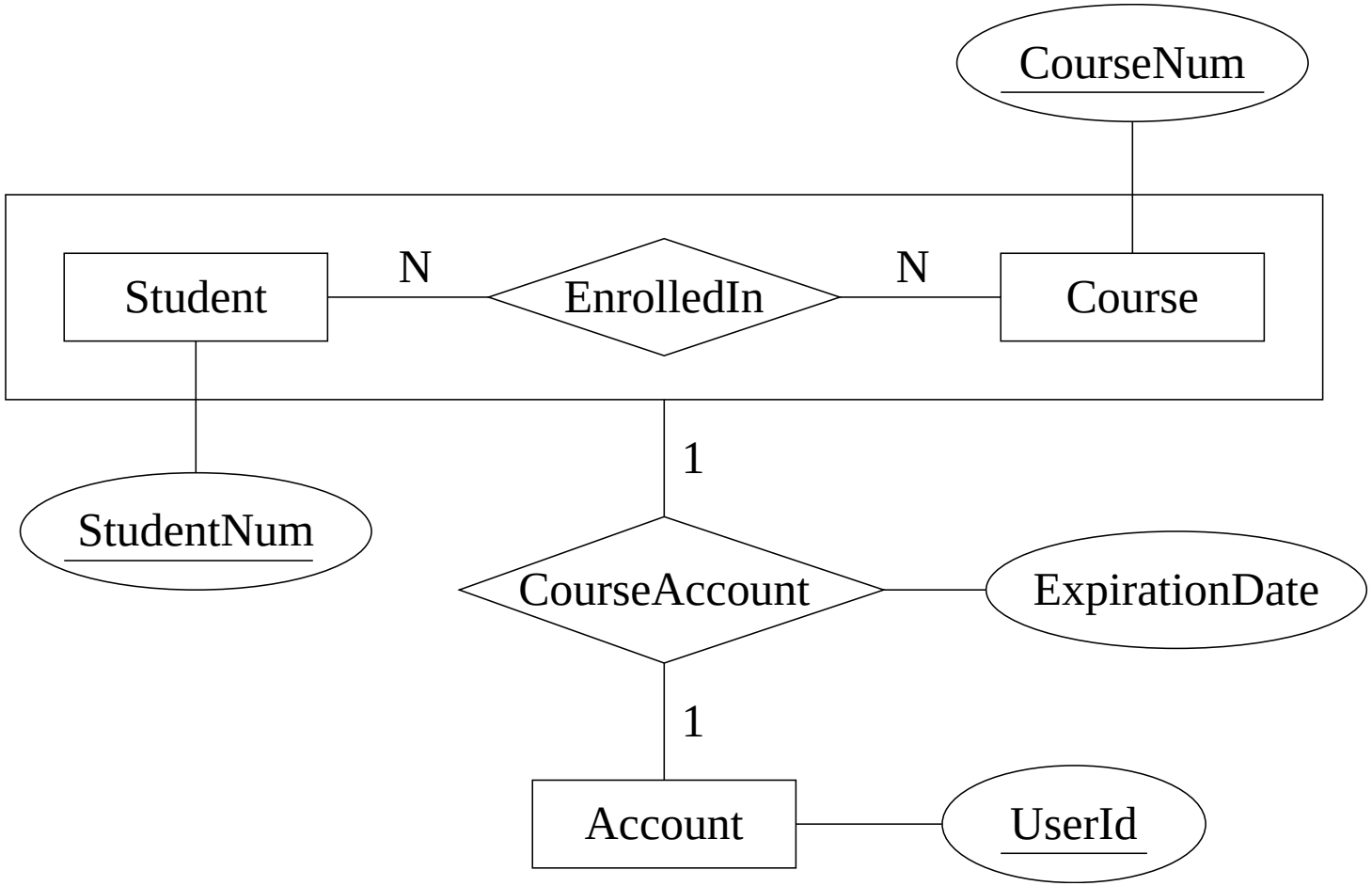
Distinguishing an Identifying Relationship



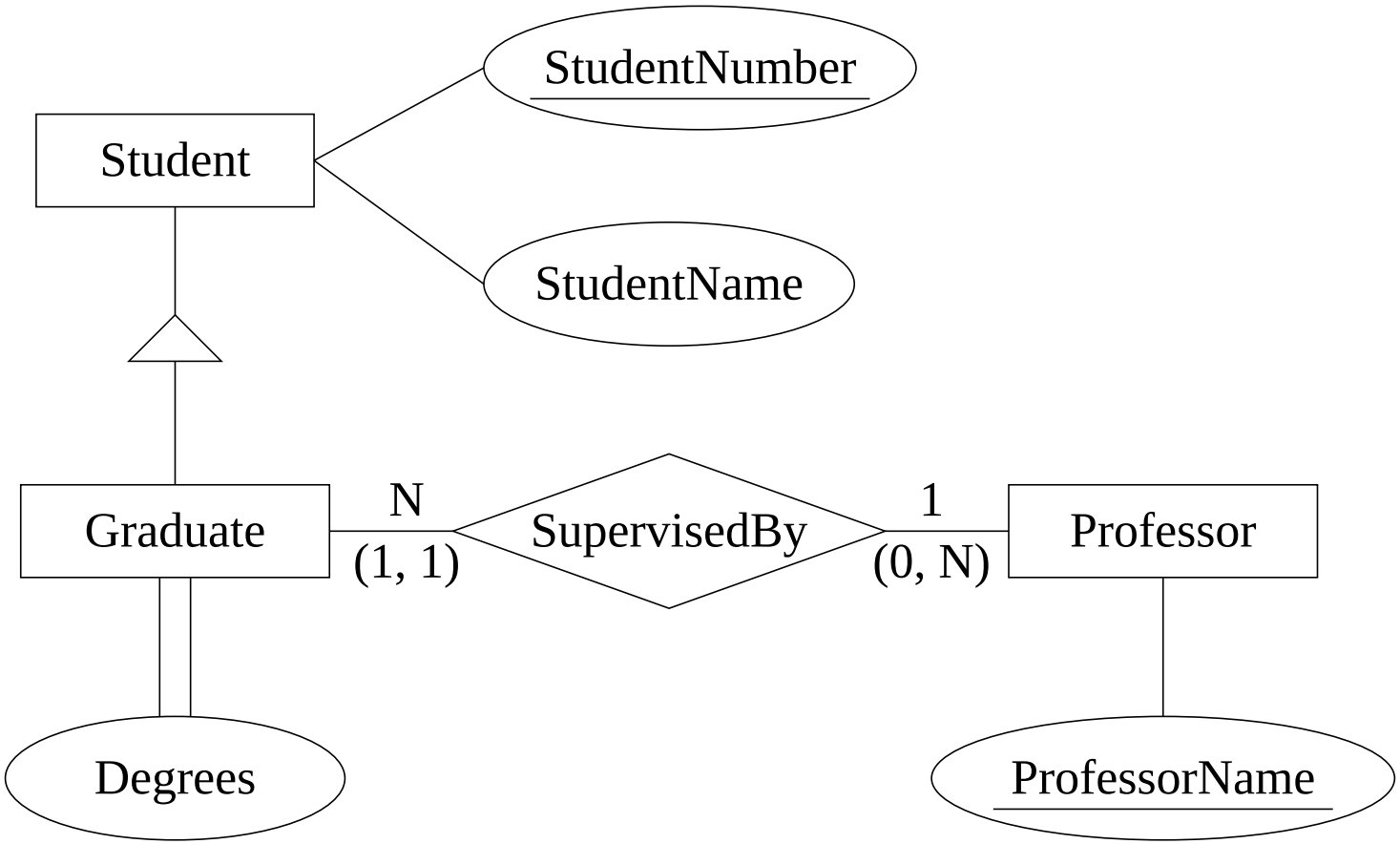
Composite and Multi-Valued Attributes



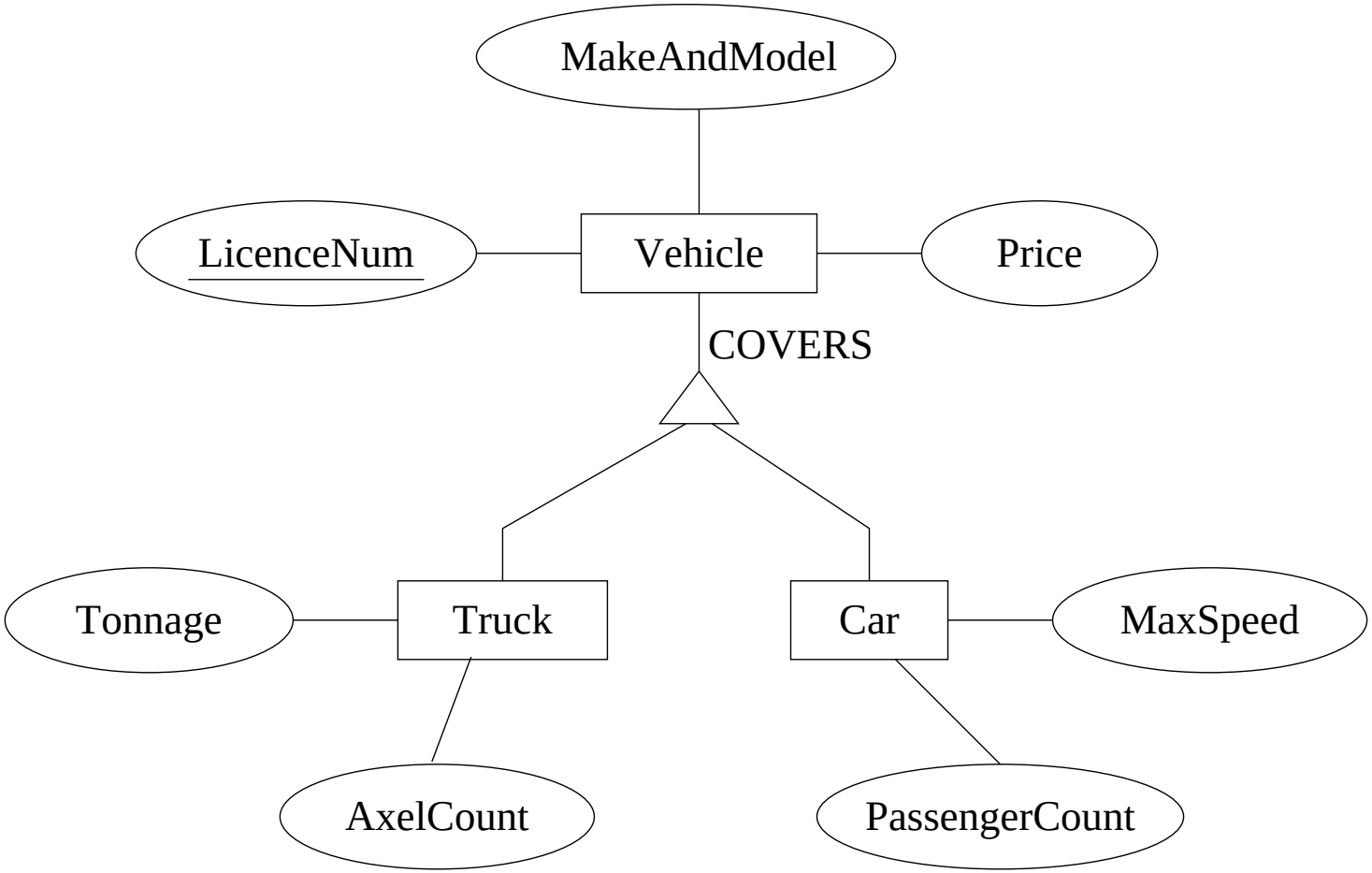
Aggregation



Specialization



Generalization



A Simple E-R Design Methodology

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

Choosing Between Attributes and Entity Sets

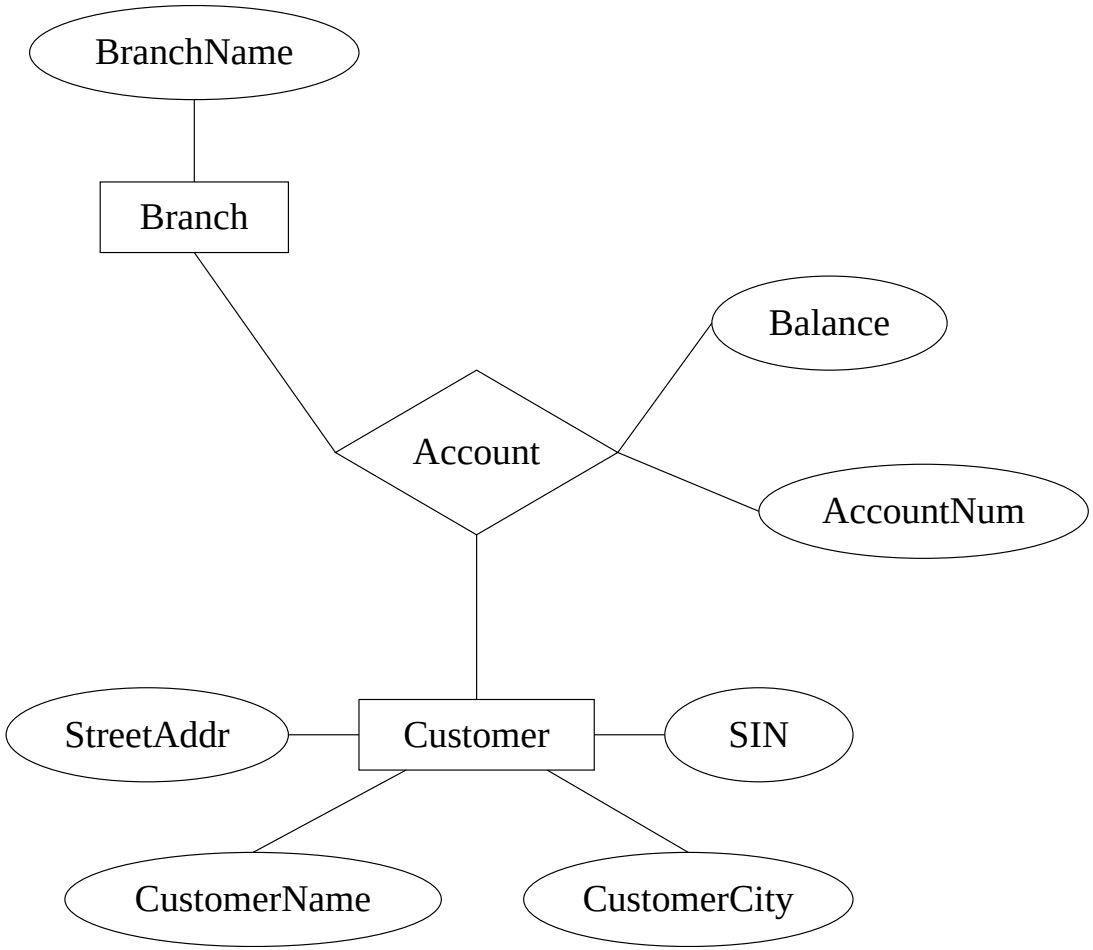
Should one model employees' phones by a PhoneNumber attribute, or by a Phone entity set related to the Employee entity set?

- 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.

Choosing Between Entity Sets and Relationship Sets

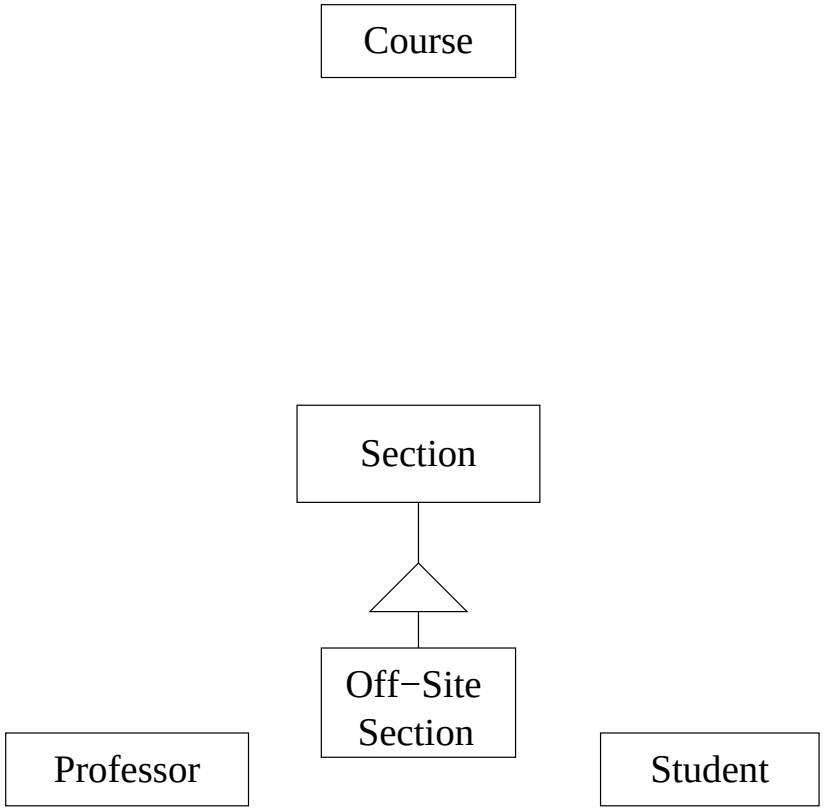
Instead of representing accounts as entities, we could represent them as relationships



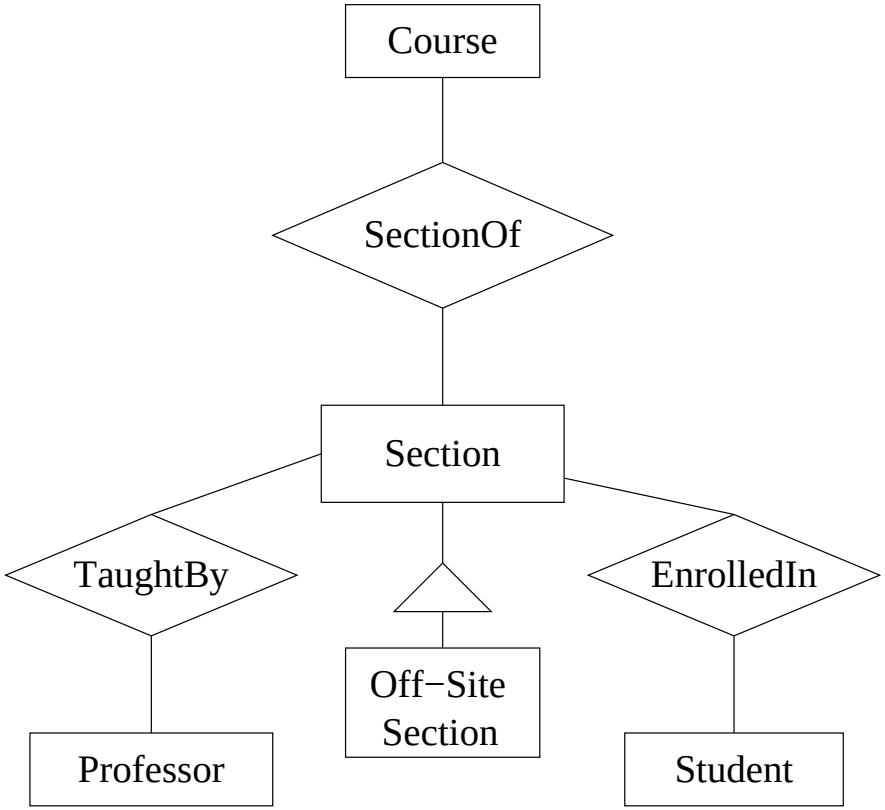
Example: A Registrar's Database

- 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, but if a professor teaches 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.

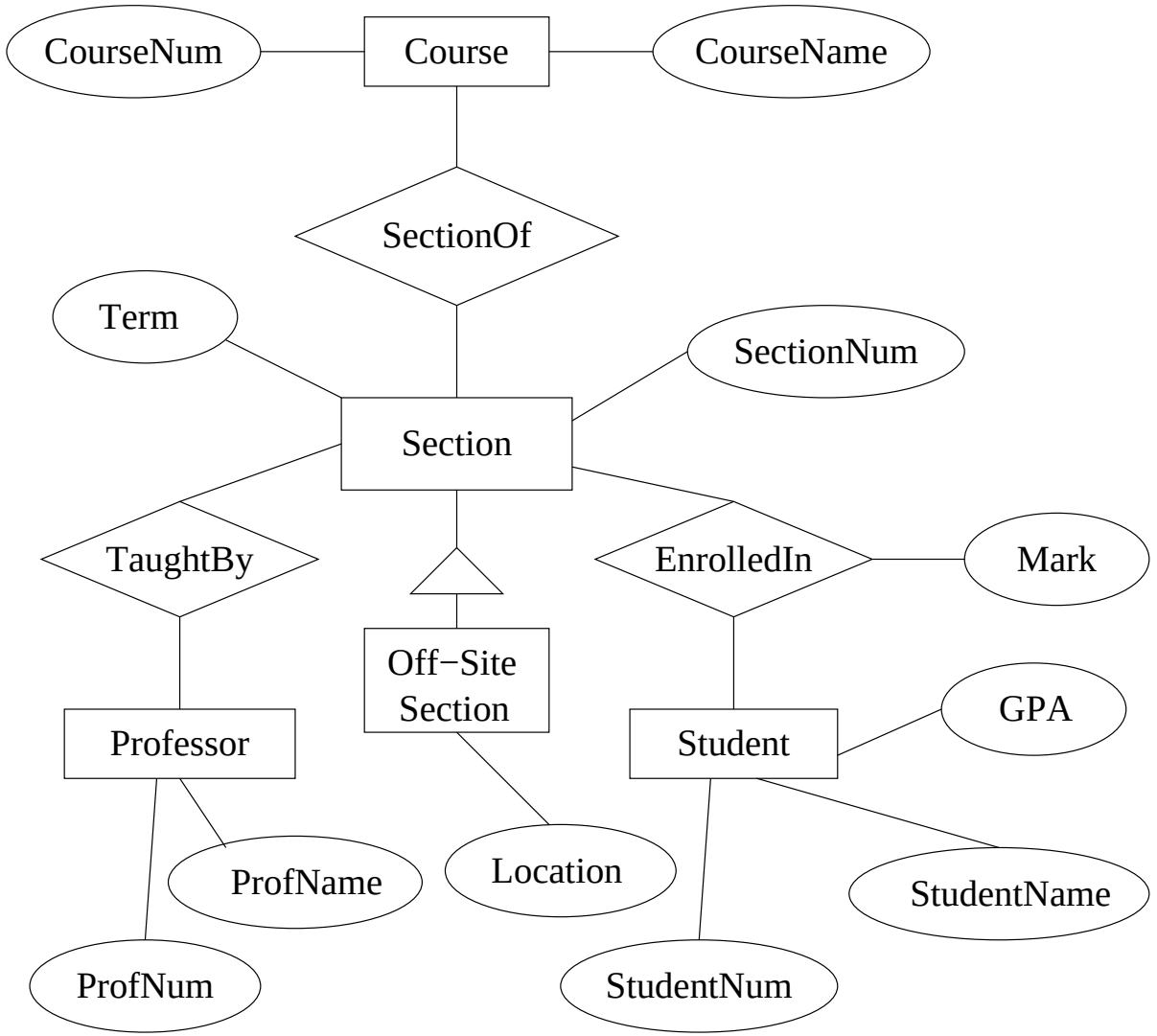
Example: A Registrar's Database (cont'd)



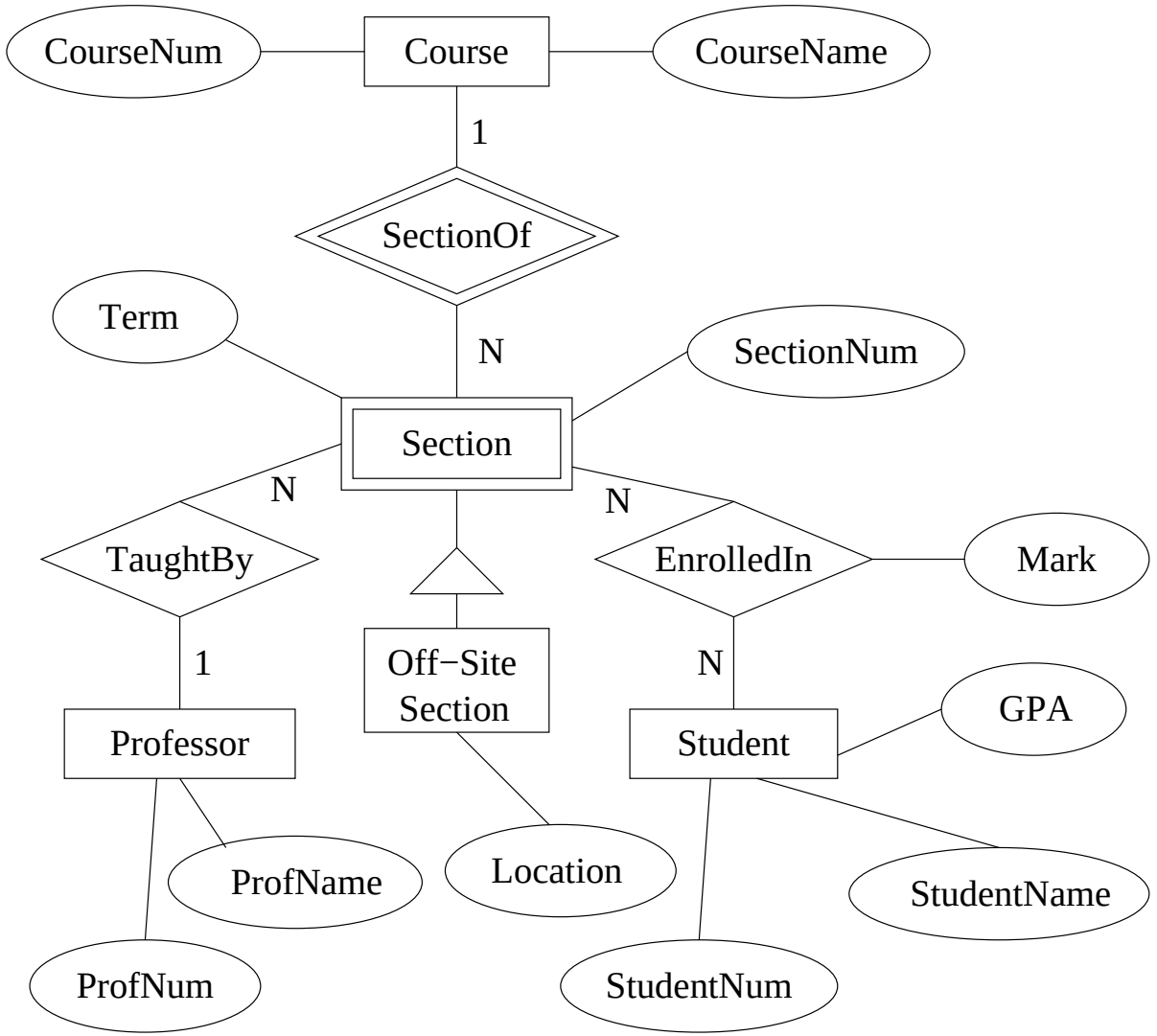
Example: A Registrar's Database (cont'd)



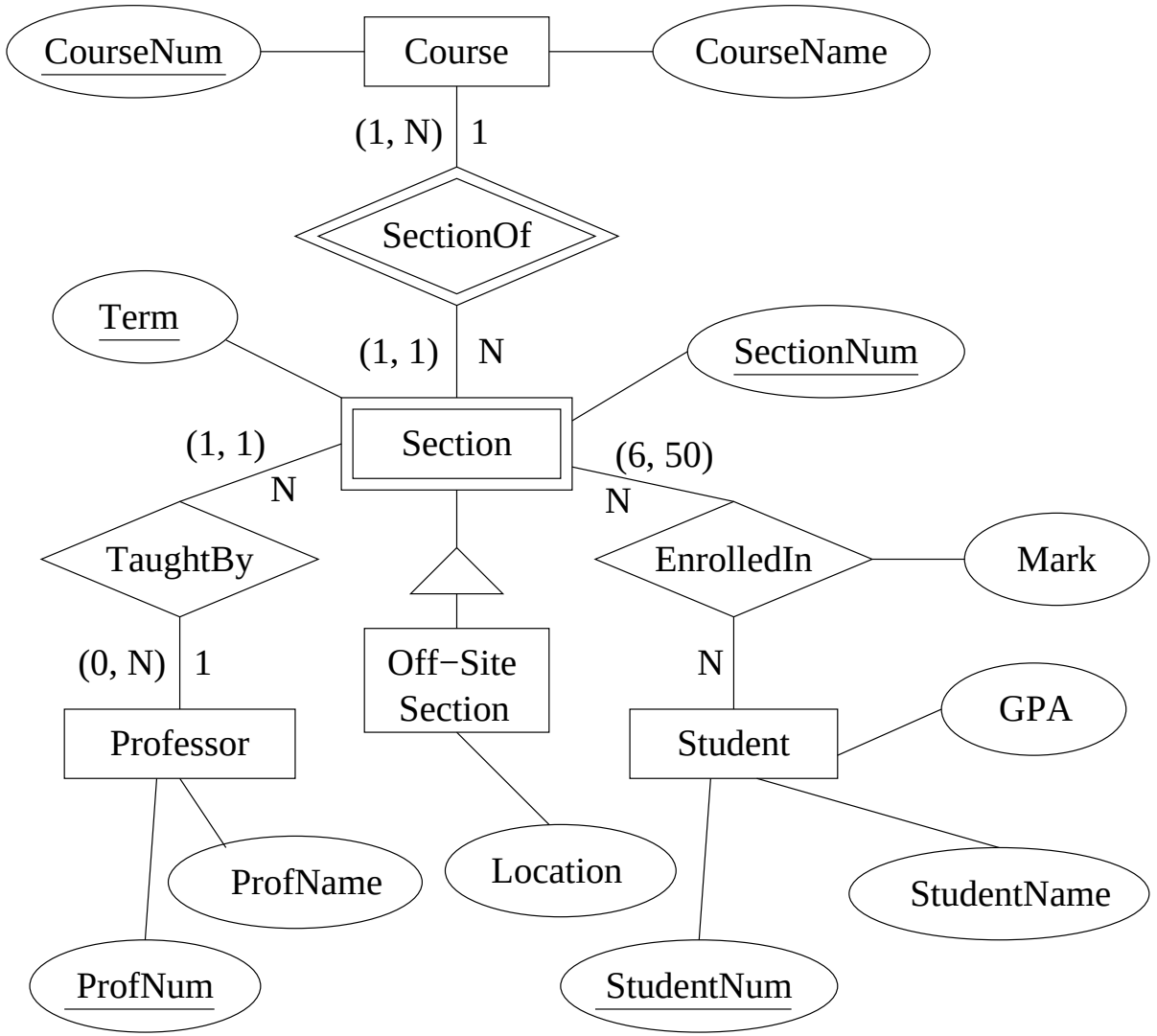
Example: A Registrar's Database (cont'd)



Example: A Registrar's Database (cont'd)



Example: A Registrar's Database (cont.)



Baseball League Example

- The league includes teams from various towns. Some towns may have more than one team. Each team has a unique name. The league's teams are divided into two divisions - each team belongs to one division.
- During a single season each team plays games against other teams in the league. Teams are to be tracked across several seasons.
- Games are played at fields. Fields have names. Each team has a designated home field, which is located in the team's town. A single field may be the home field for more than one team. Every field is home to at least one team.
- Each game is played between two teams. One team is the winner, the other is the loser (no ties).

Baseball League Example (cont'd)

- Each game is played at the home field of one of the two teams involved - that team is called the home team, the other is called the visiting team.
- Each team plays at most one game on any given day. During a game, a each team accumulates certain numbers of hits, runs, and errors.
- Each team has a roster of players. Players do not change teams during the course of a single season, but they may change teams between seasons. Some players may not belong to any team during a particular season.
- Players have names, and, if they are on a team, numbers. Player numbers are unique within a team not unique throughout the league. A player's number may change from season to season even if the player remains on the same team.