Data Modeling and the Entity-Relationship Model

School of Computer Science
University of Waterloo

CS 348
Introduction to Database Management
Fall 2007

Outline

Overview of E-R Model

Used for (and designed for) database (conceptual schema) design

$\Rightarrow$ Proposed by Peter Chen in 1976

World/enterprise described in terms of

- entities
- attributes
- relationships

Visualization: **E-R diagram**

Basic E-R Modeling

Entity: a *distinguishable* object

Entity set: set of entities of same type

Examples:

- students currently at University of Waterloo
- flights offered by Air Canada
- burglaries in Ontario during 1994

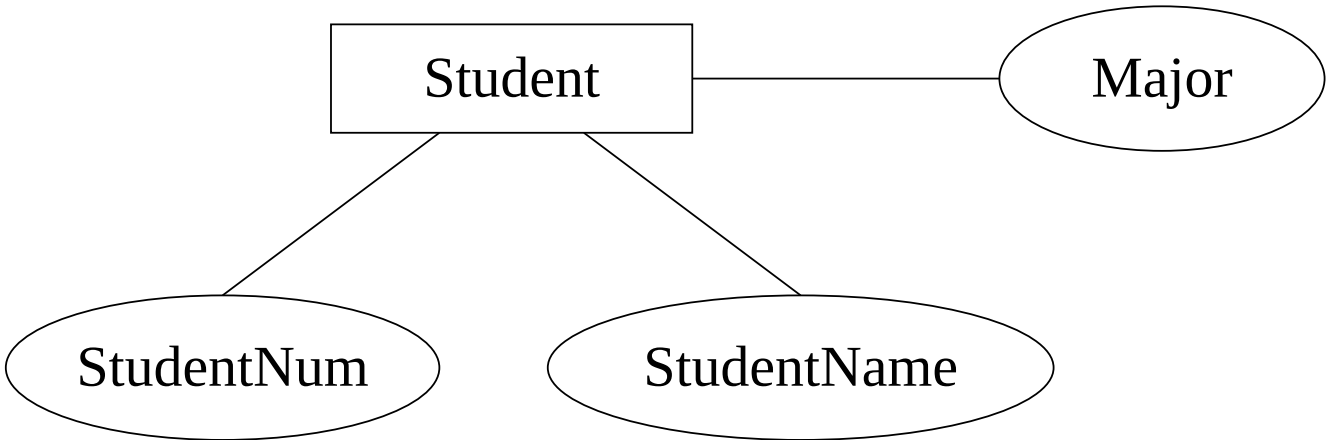
Graphical representation of entity sets:

Basic E-R Modeling (cont'd)

Attributes: describe properties of entities
Examples (for Student-entities): StudentNum, StudentName, Major, ...

Domain: set of permitted values for an attribute

Graphical representation of attributes:



Basic E-R Modeling (cont'd)

Relationship: representation of the fact that certain entities are related to each other

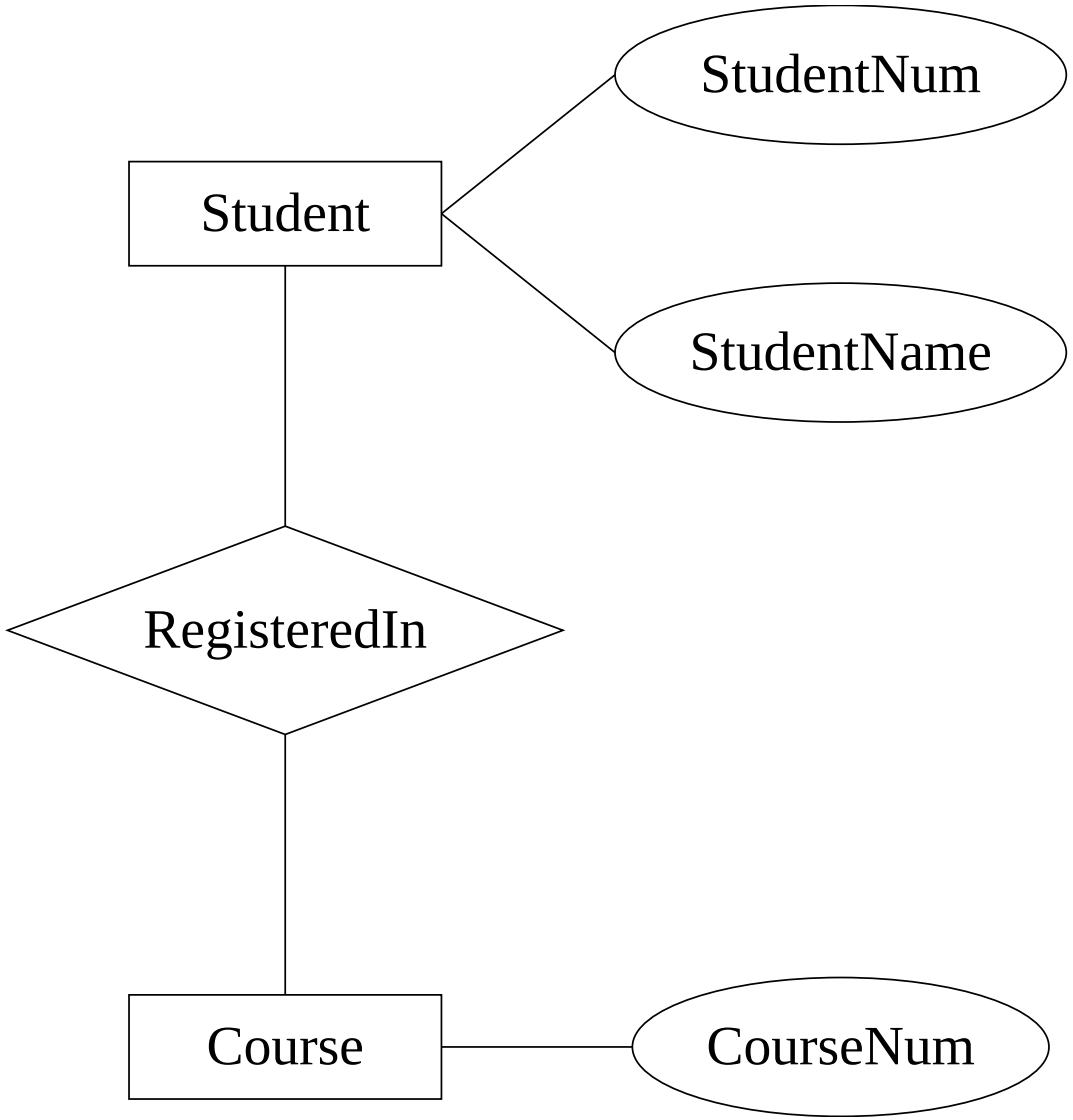
Relationship set: set of relationships of a given type

Examples:

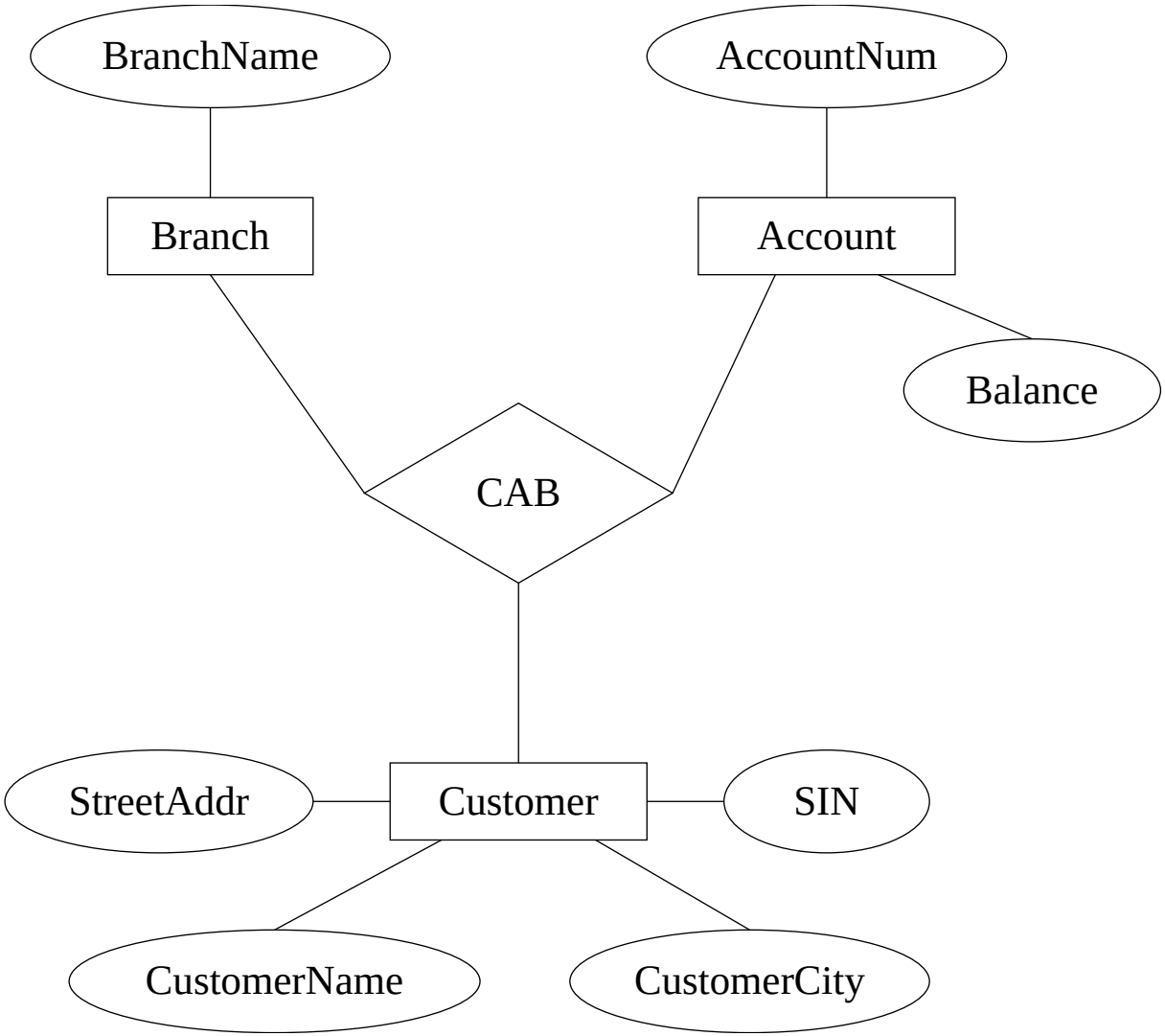
- students registered in courses
- passengers booked on flights
- parents and their children
- bank branches, customers and their accounts

In order for a relationship to exist, the participating entities must exist.

Graphical Representation



Graphical Representation (cont'd)

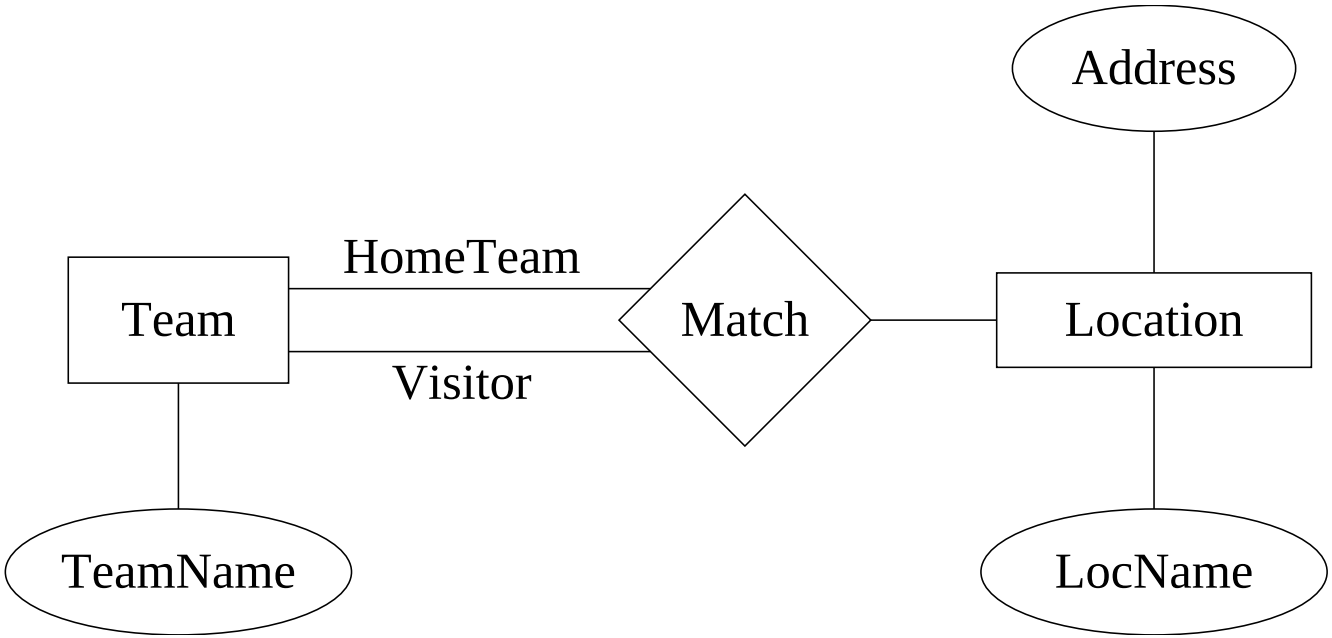


Multiple Relationships and Role Names

Role: the function of an entity set in a relationship set

Role name: an explicit indication of a role

Example:

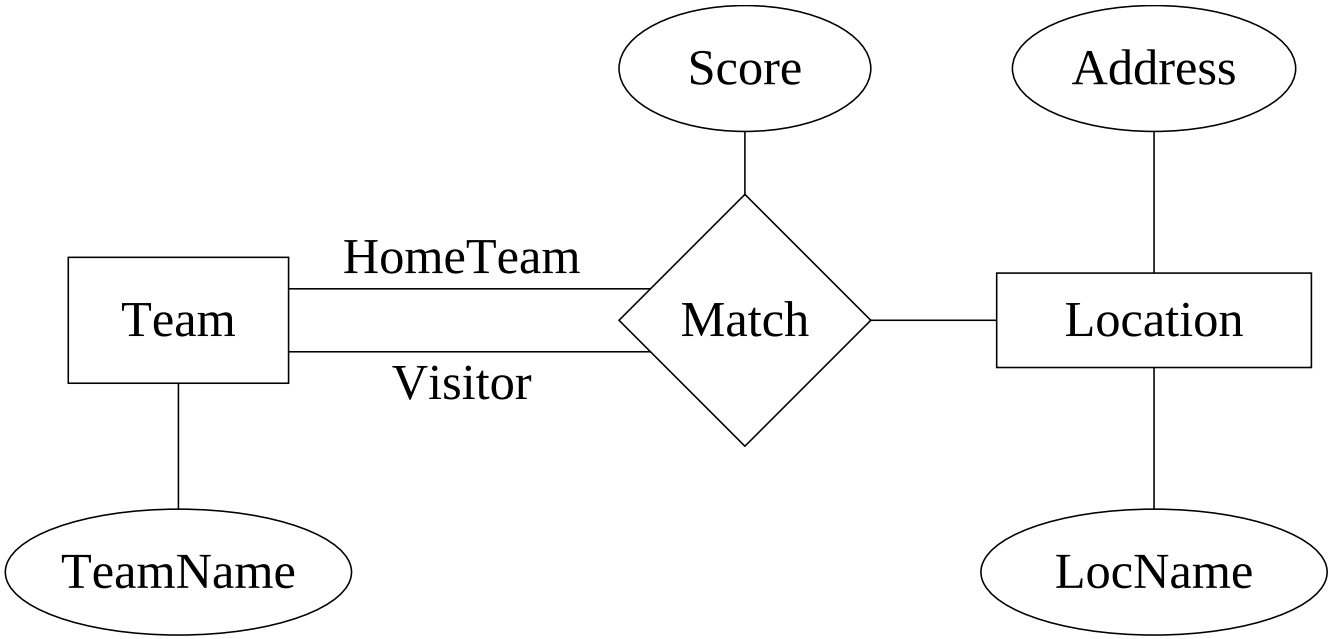


Role labels are needed whenever an entity set has multiple functions in a relationship set.

Relationships and Attributes

Relationships may also have attributes

Example:



Constraints in E-R Models

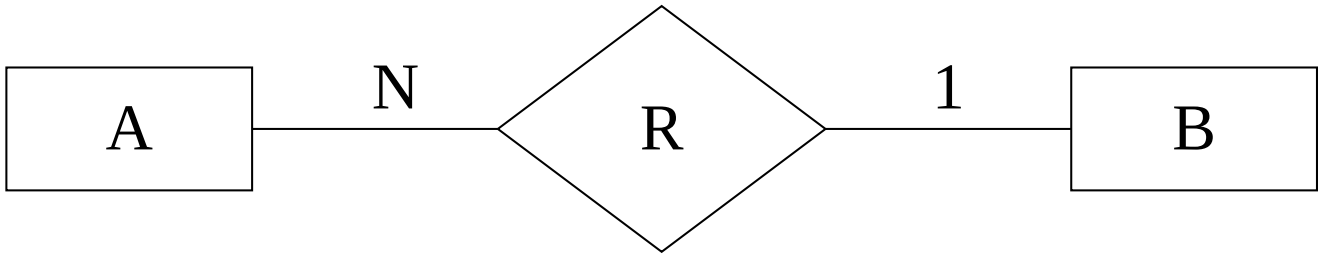
- Binary relationship types
- General cardinality constraints
- Primary keys
- Existence dependencies

Binary Relationship Types

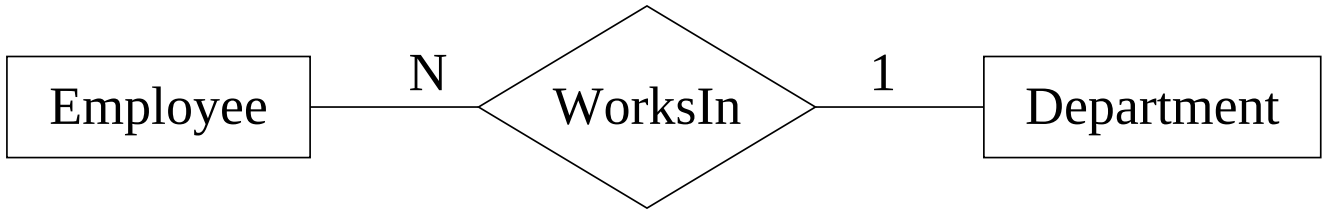
relationships between **TWO** entity sets A and B

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

Visualization:



Example:



- similarly: **one-to-many (1:N)**

Binary Relationship Types (cont'd)

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

Example:

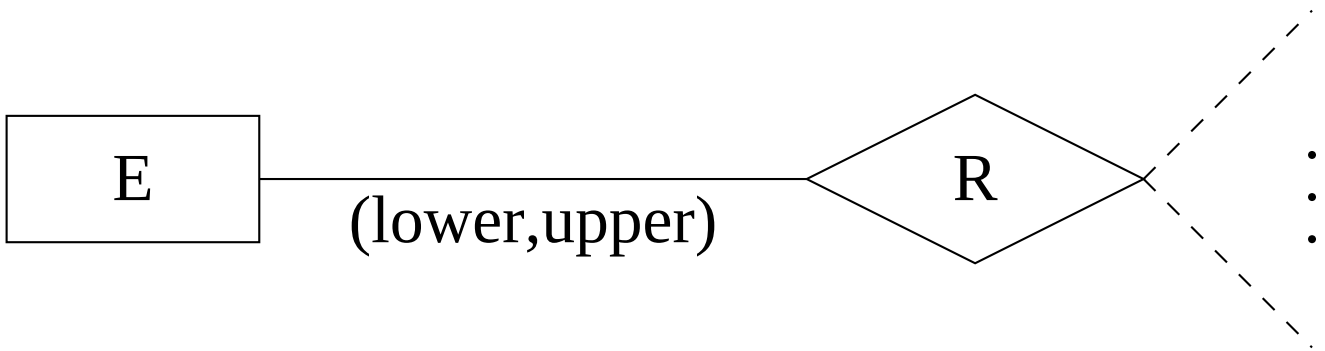
- **many-to-many (N:N):** an entity can be related to many entities in the other set, and vice versa

Example:

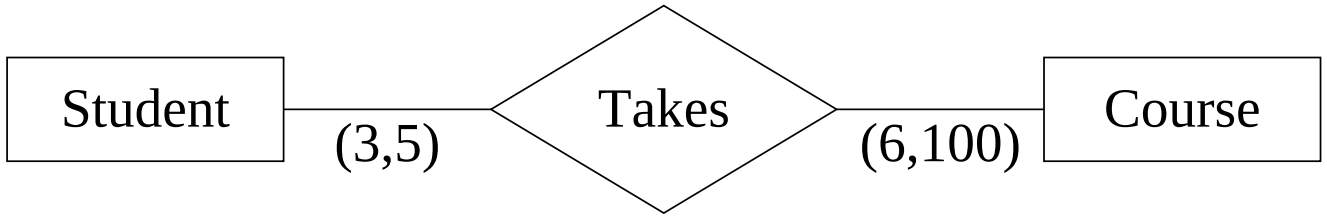
General Cardinality Constraints

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:

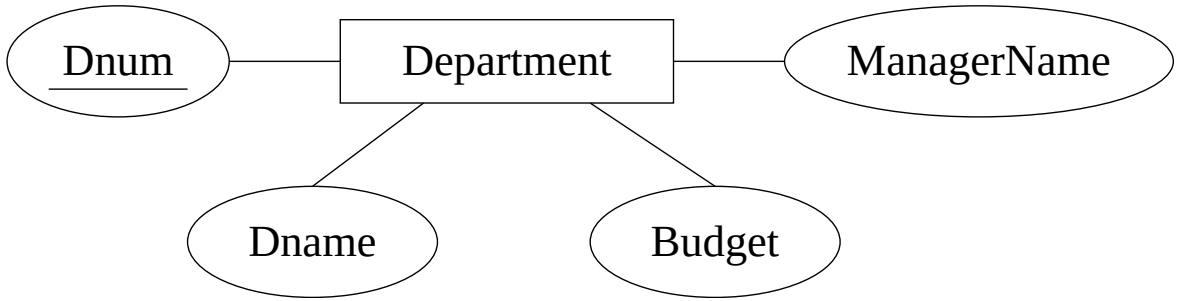


Primary Keys

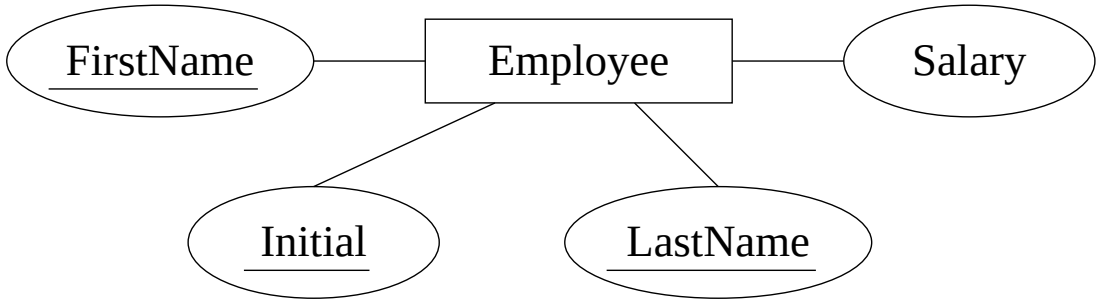
Each entity must be distinguishable from any other entity in an entity set by its attributes

Primary key: selection of attributes chosen by designer values of which determines the particular entity.

Example 1:



Example 2:



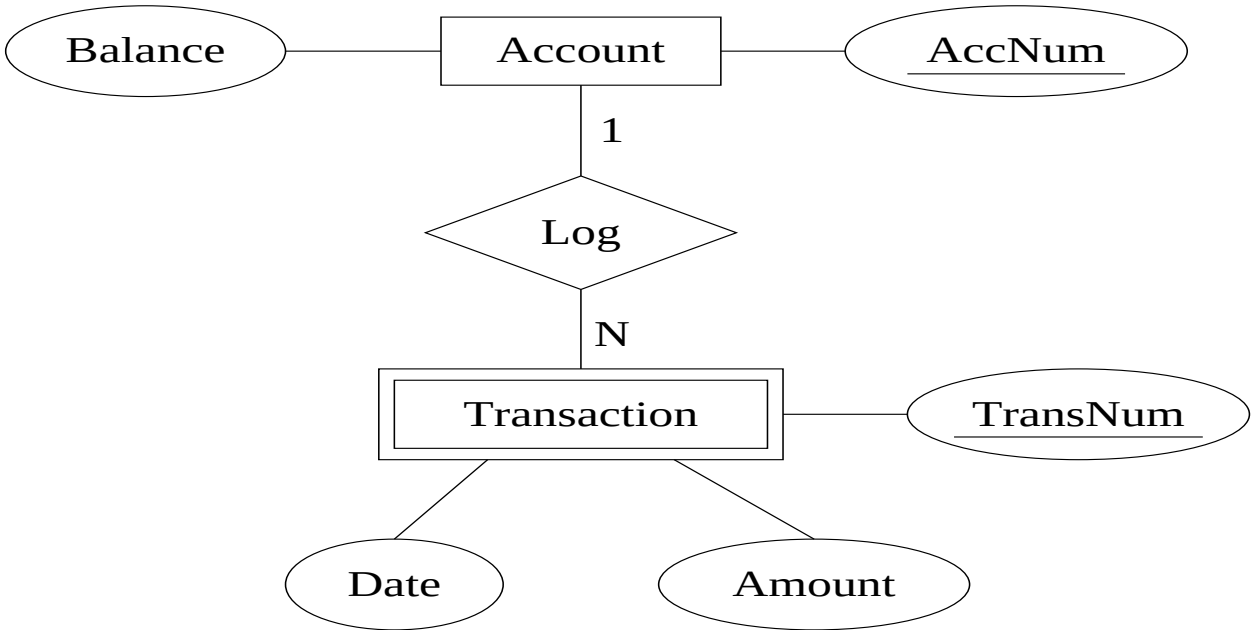
Existence Dependencies

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."



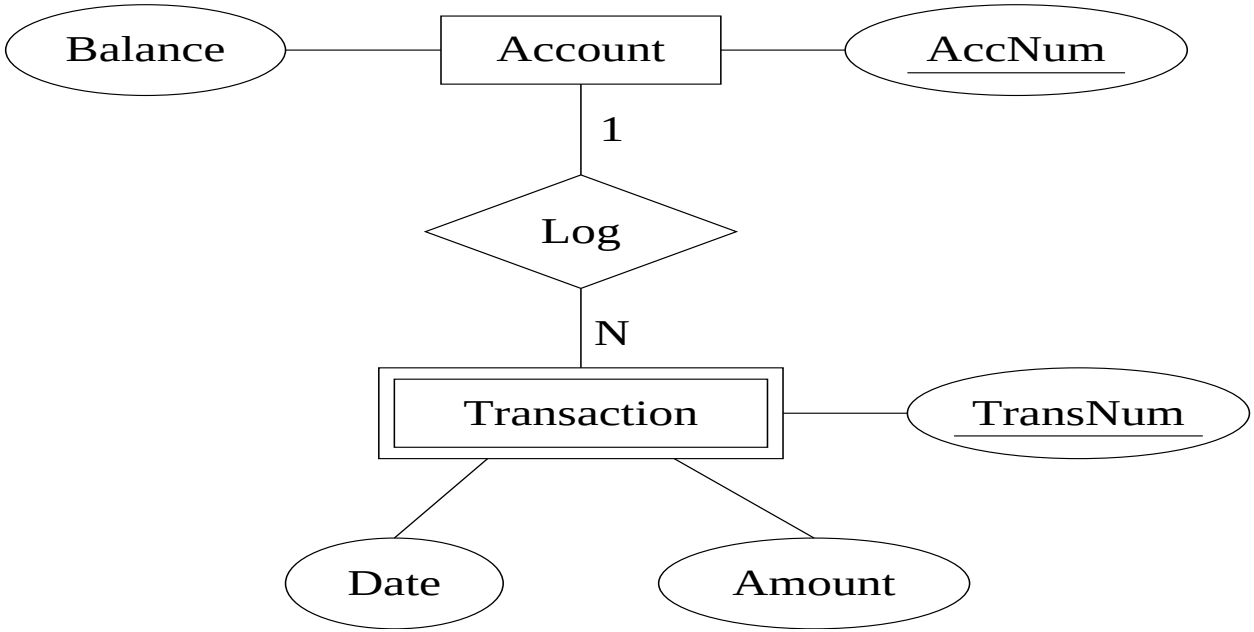
Identifying Subordinate Entities

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 form key relative to a given dominant entity

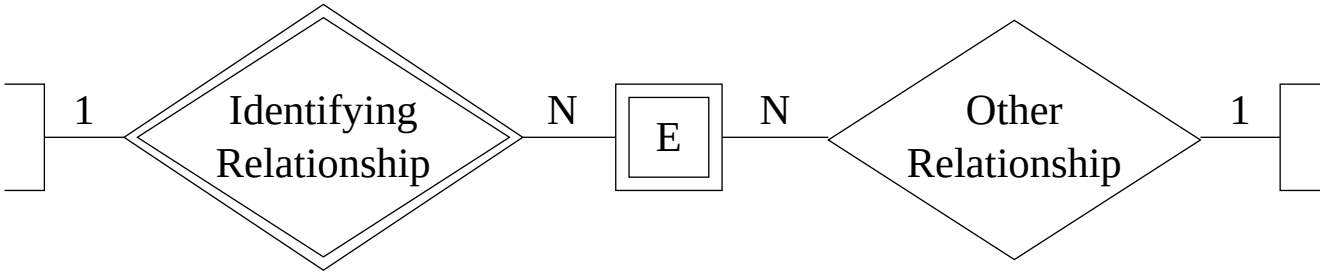
Example: "All transactions for a given account have a unique transaction number."



Identifying Subordinate Entities (cont'd)

A weak entity set must have an N:1 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

Extensions to E-R Modeling

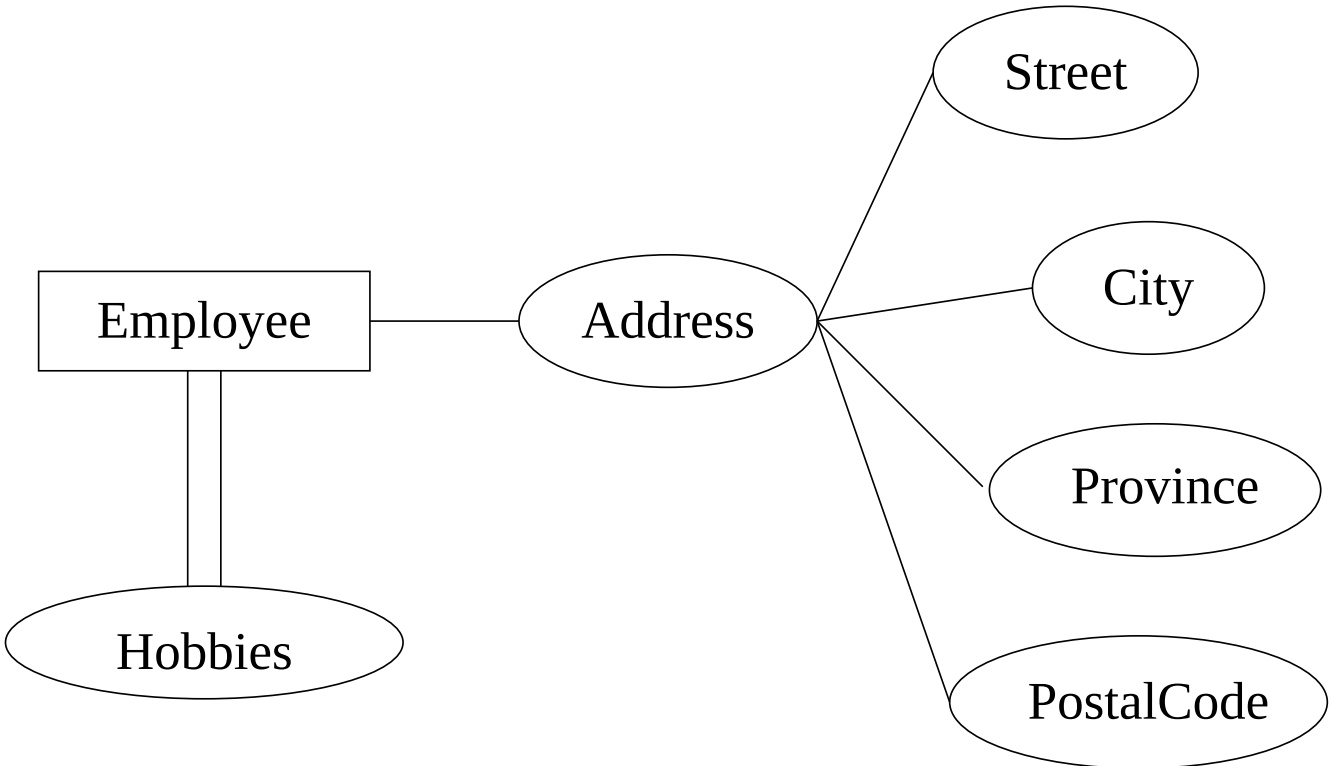
- Structured attributes
- Aggregation
- Specialization
- Generalization
- Disjointness

Structured Attributes

Composite attributes: composed of fixed number of other attributes

Multi-valued attributes: attributes that are set-valued

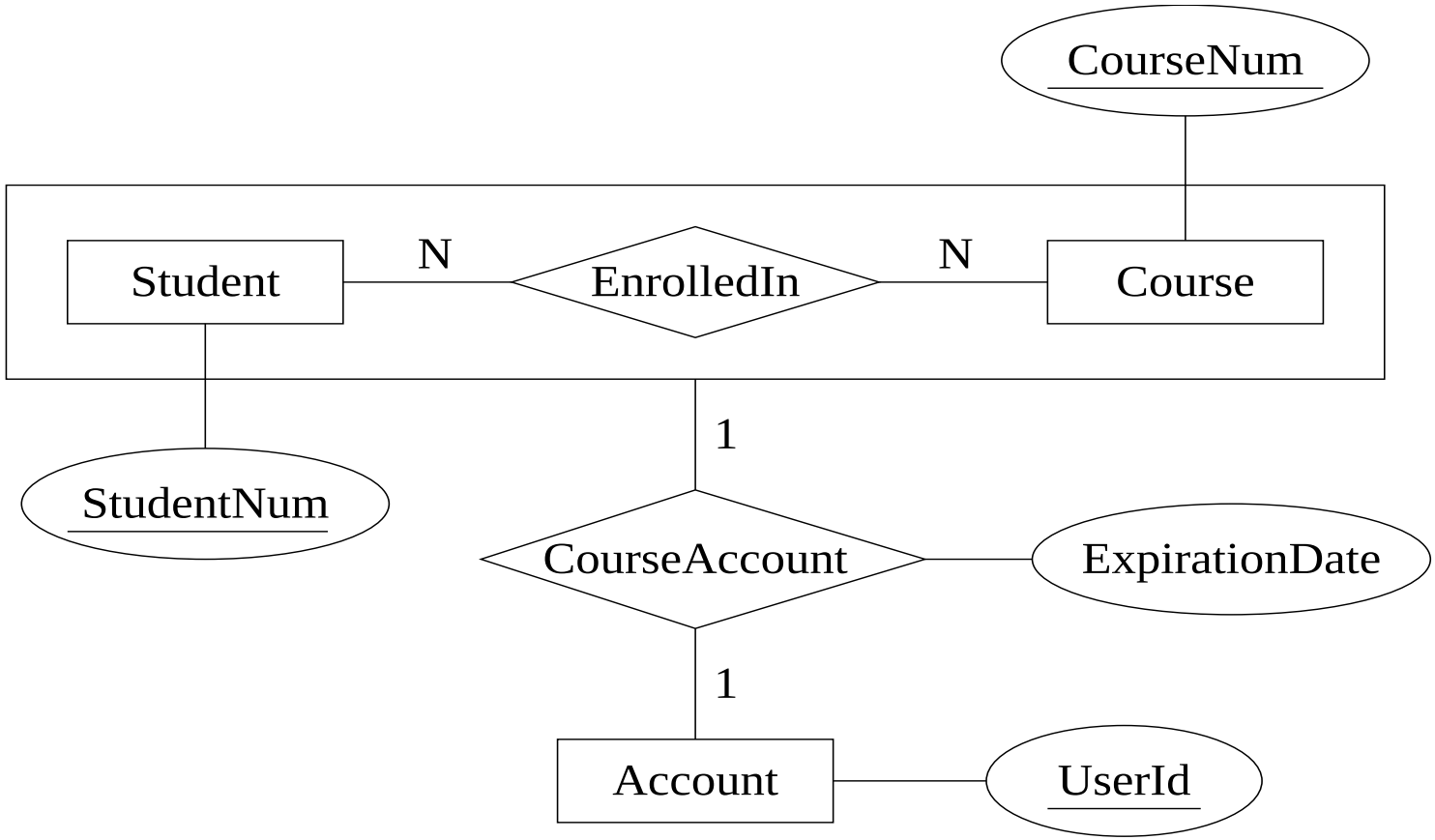
Example:



Aggregation

Relationships can be viewed as higher-level entities

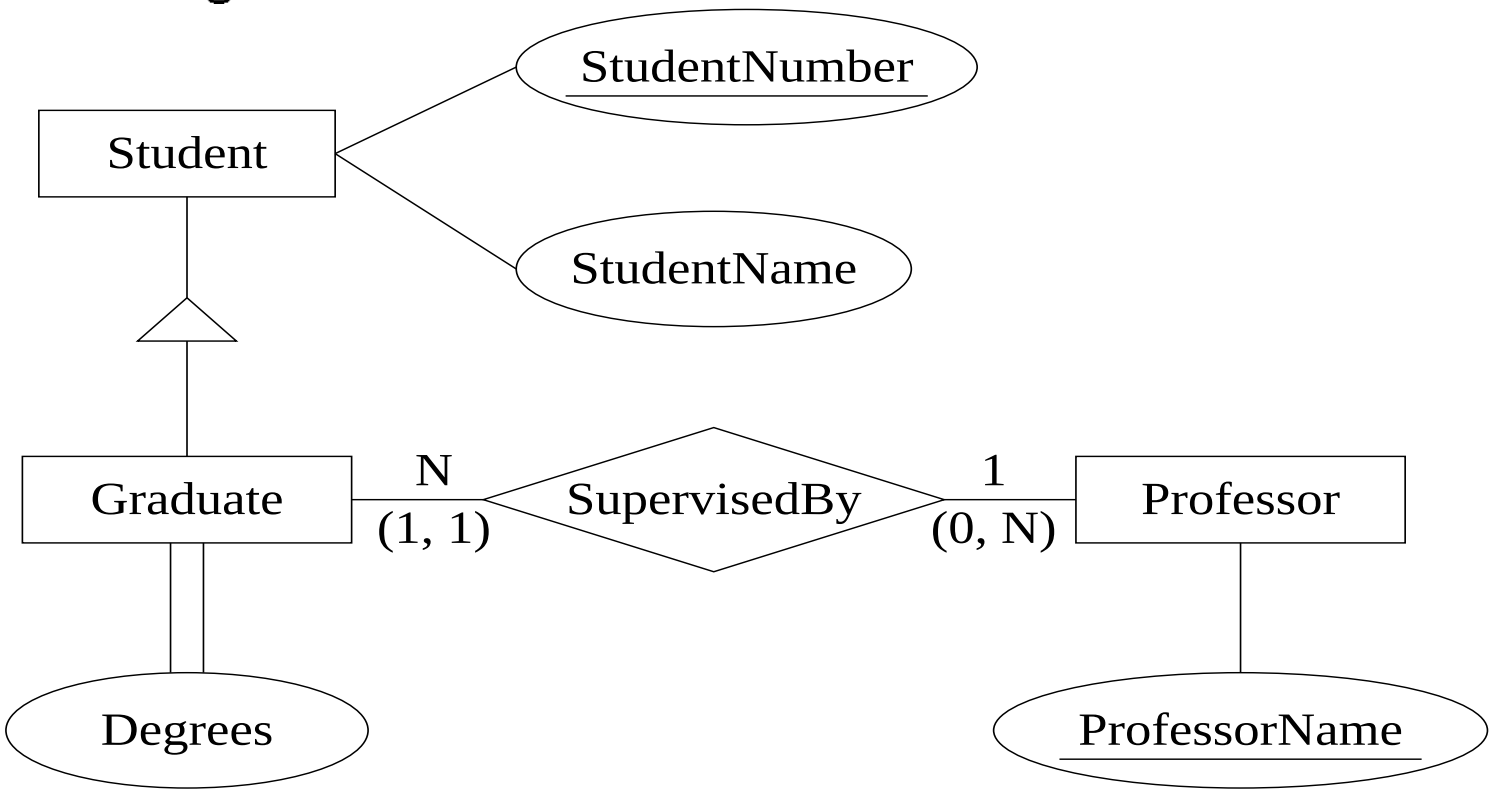
Example: "Accounts are assigned to a given student enrollment."



Specialization

A specialized kind of entity set may be derived from a given entity set

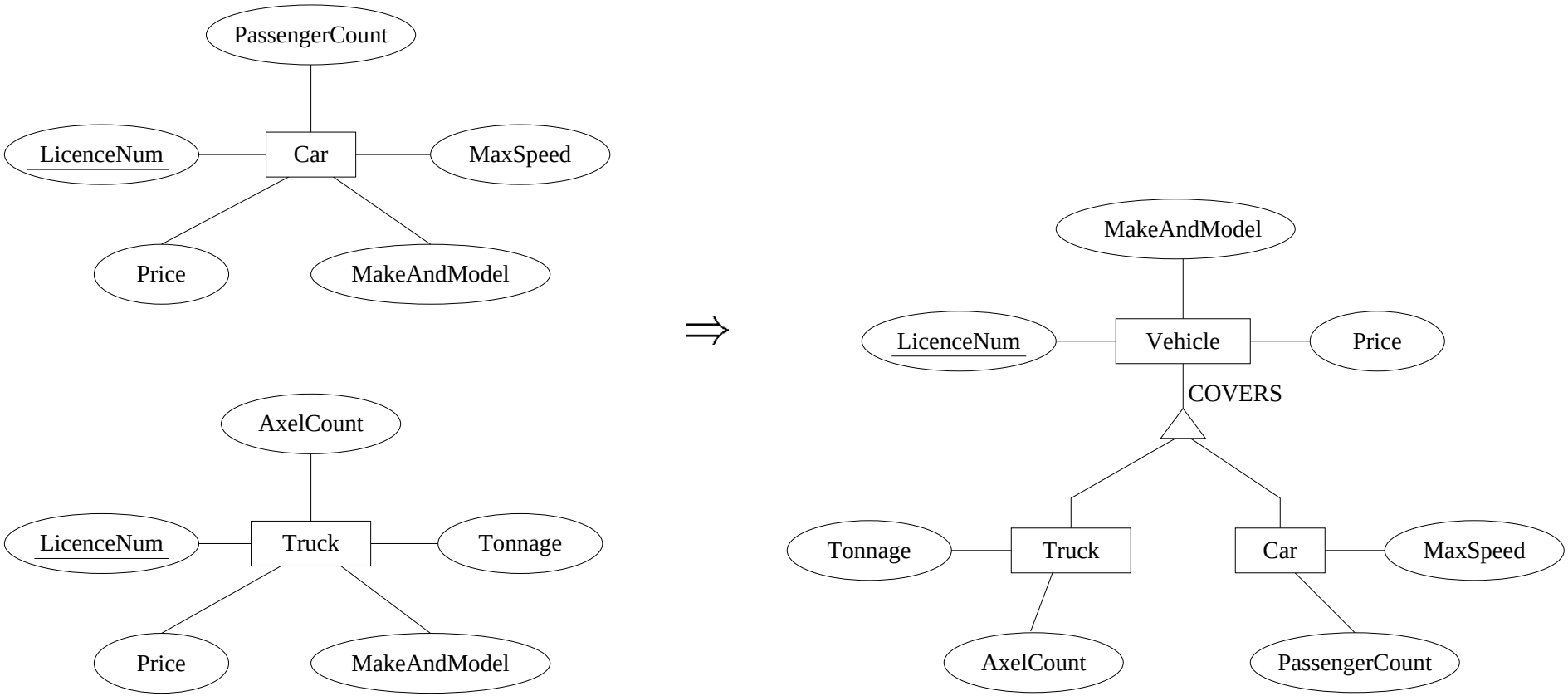
Example: "Graduate students are students that have a supervisor and a number of degrees."



Generalization

Several entity sets can be abstracted by a more general entity set

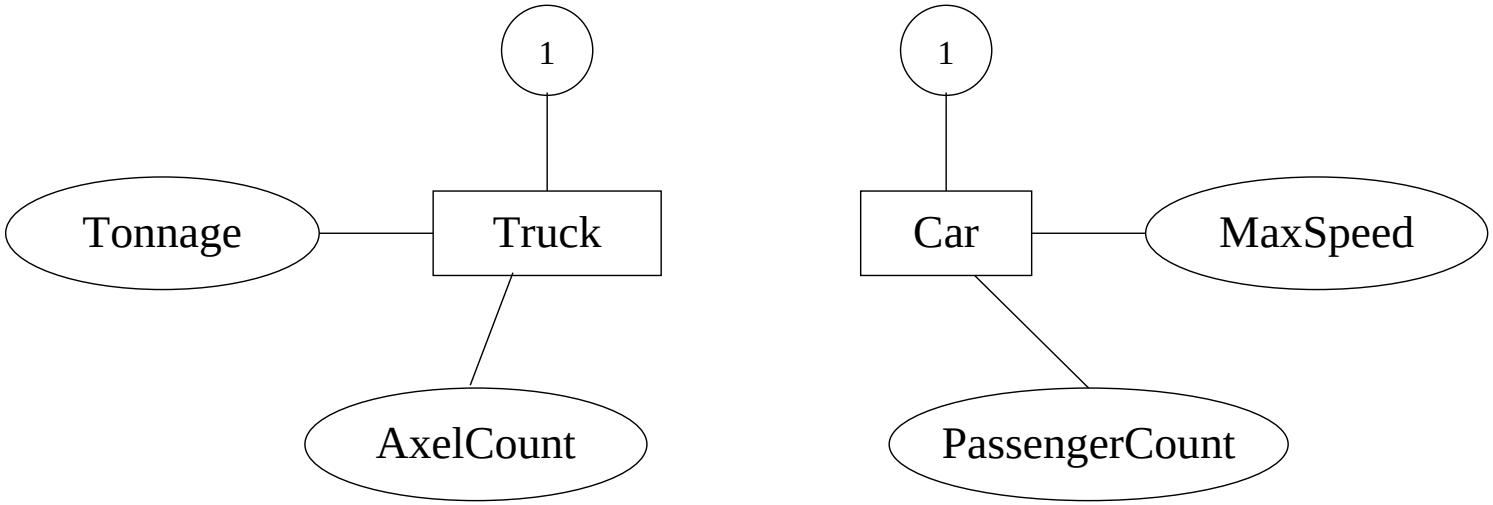
Example: "A vehicle abstracts the notion of a car and a truck."



Disjointness

Two or more entity sets can be declared to have no entities in common between any pair

Example: "Nothing is both a car and a truck."



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?
- extended features?

Attributes or Entity Sets?

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

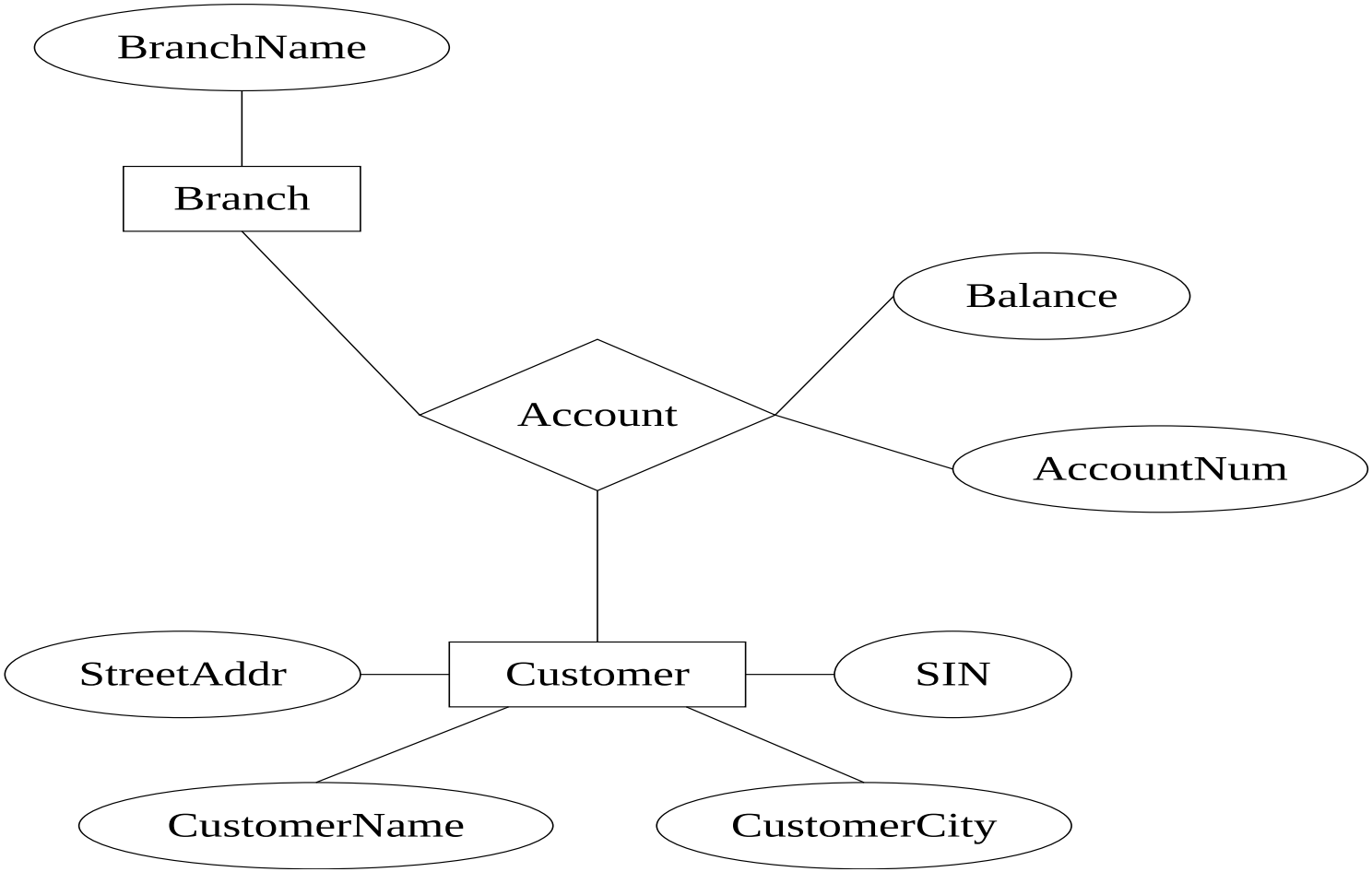
Rules of thumb:

- 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?

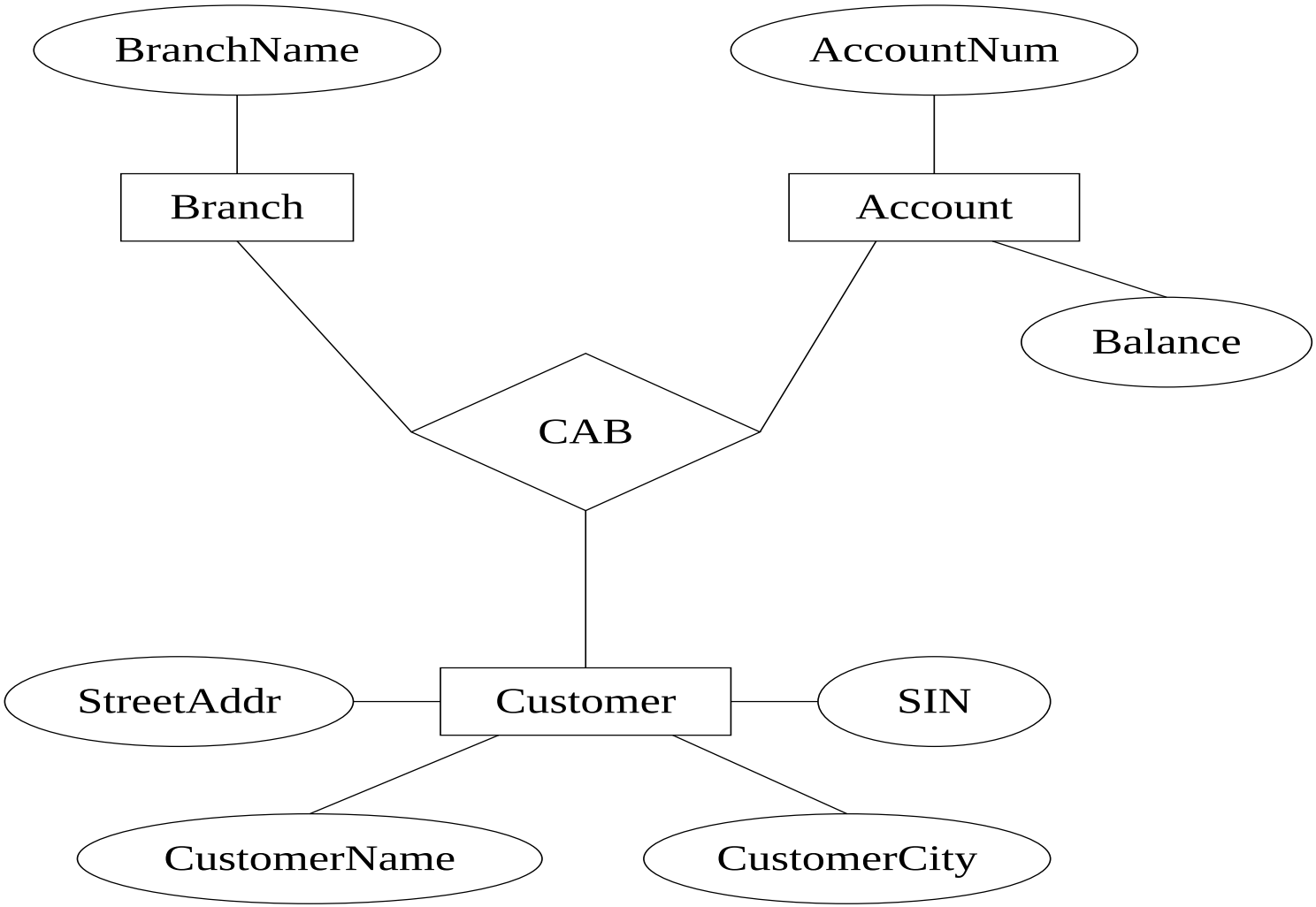
An affirmative answer to any of the above suggests a new entity set.

Entity Sets or Relationships?

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

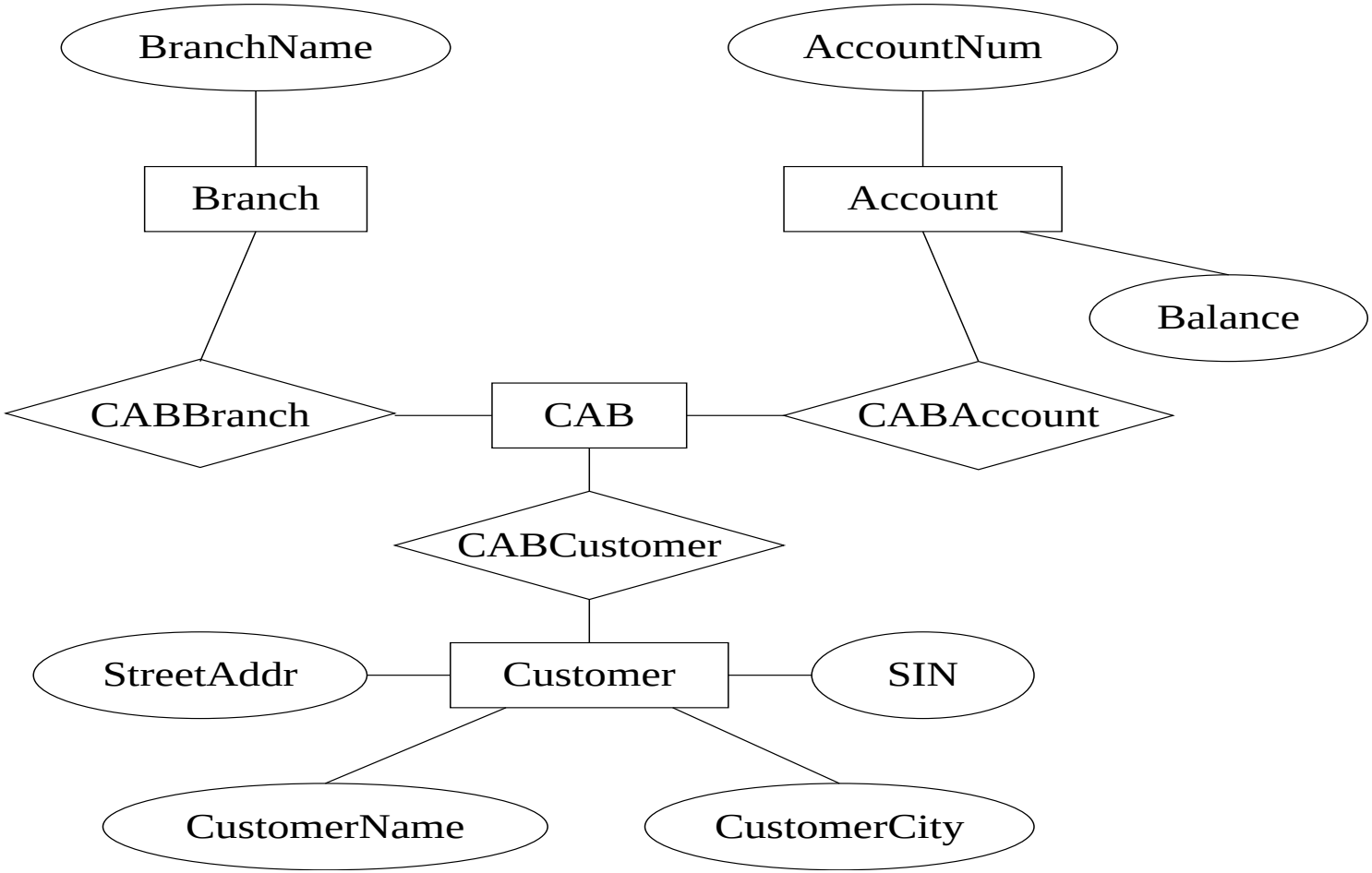


Binary vs. N-ary Relationships?



Binary vs. N-ary Relationships (cont'd)

We can always represent a relationship on n entity sets with n binary relationships



A Simple Methodology

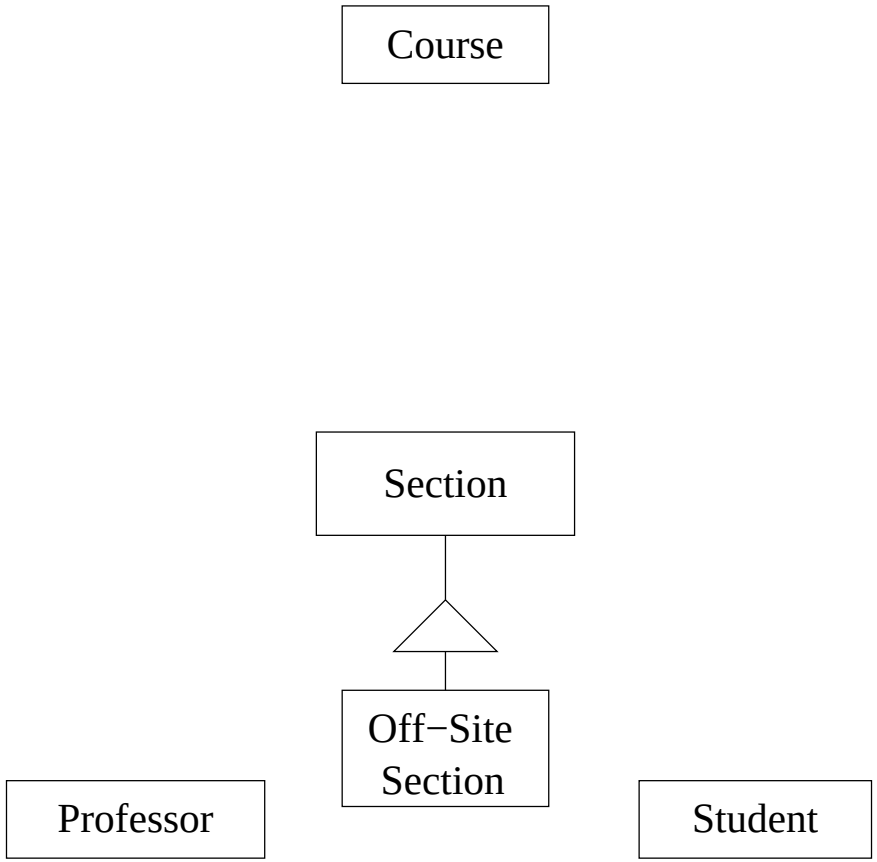
1. Recognize entity sets
2. Recognize relationship sets and participating entity sets
3. Recognize attributes of entity and relationship sets
4. Define binary relationship types and existence dependencies
5. Define general cardinality constraints, keys and discriminators
6. Draw diagram

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

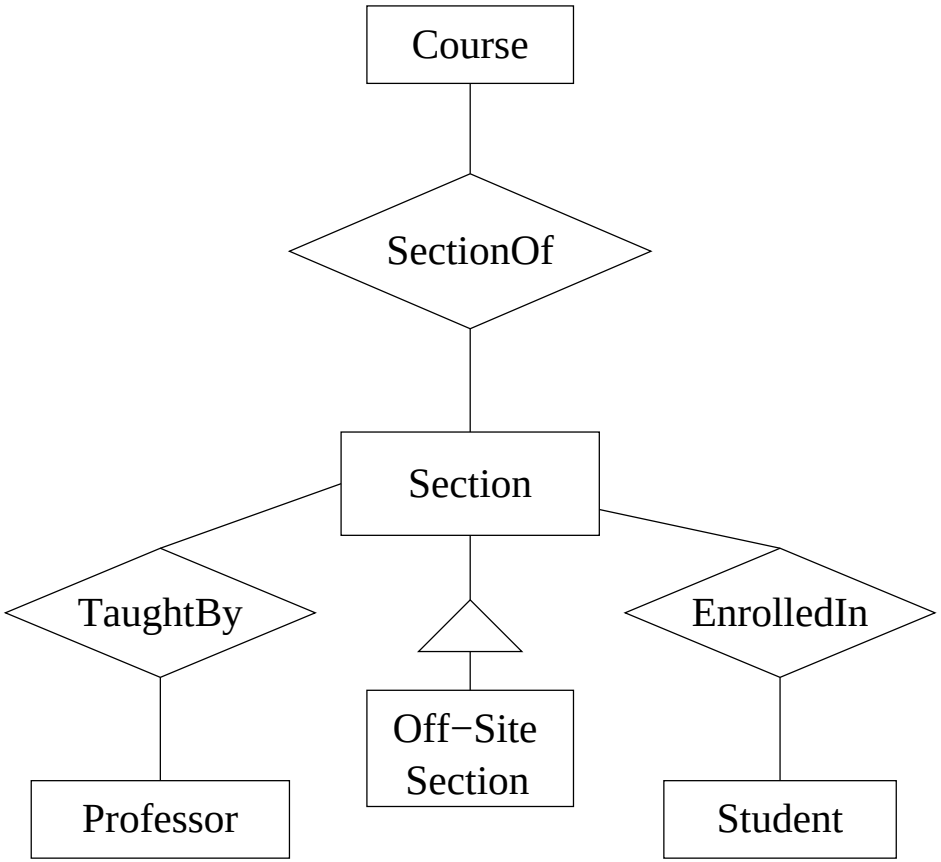
Example: A Registrar's Database

- 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, 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.

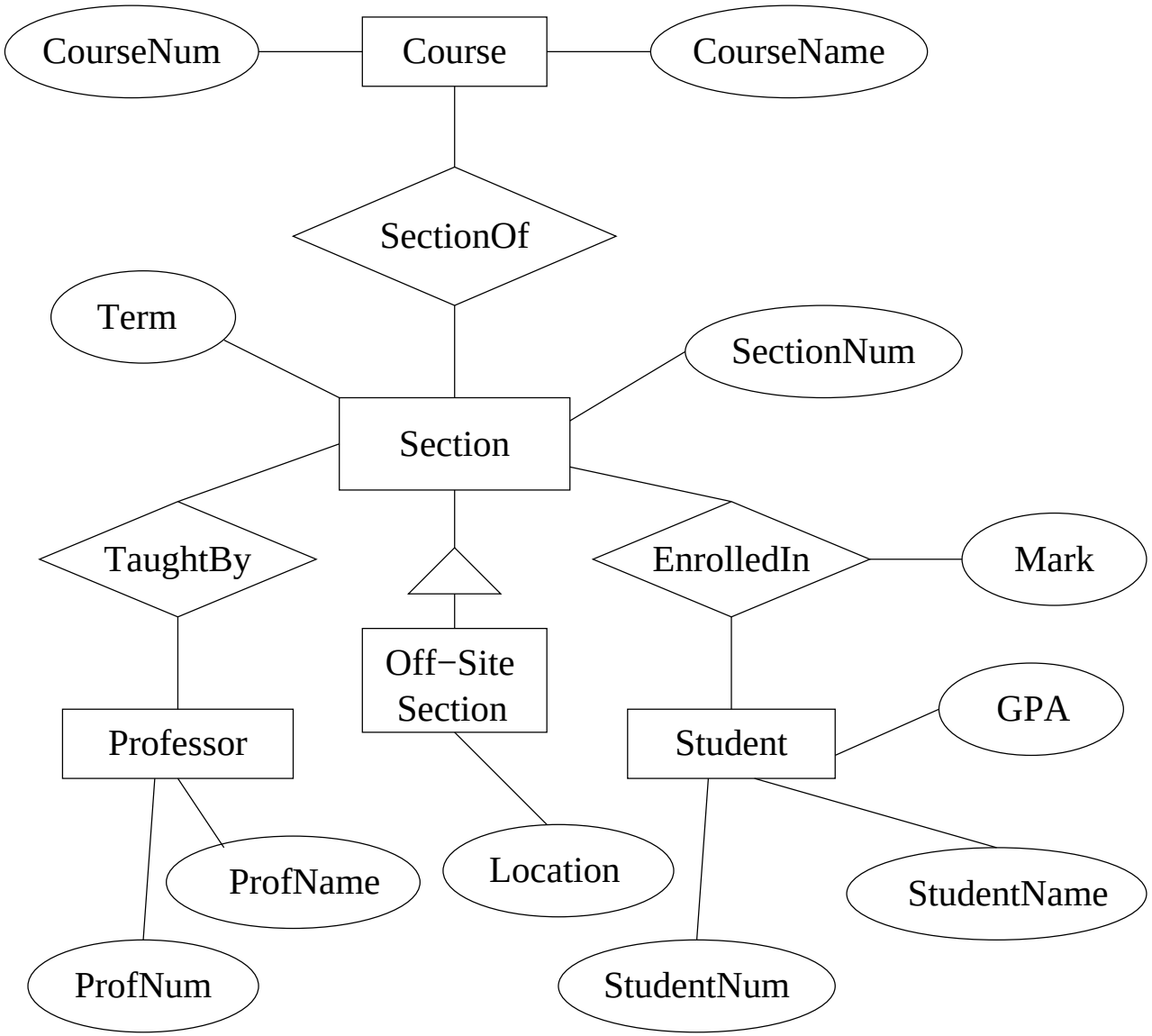
Example: A Registrar's Database (cont'd)



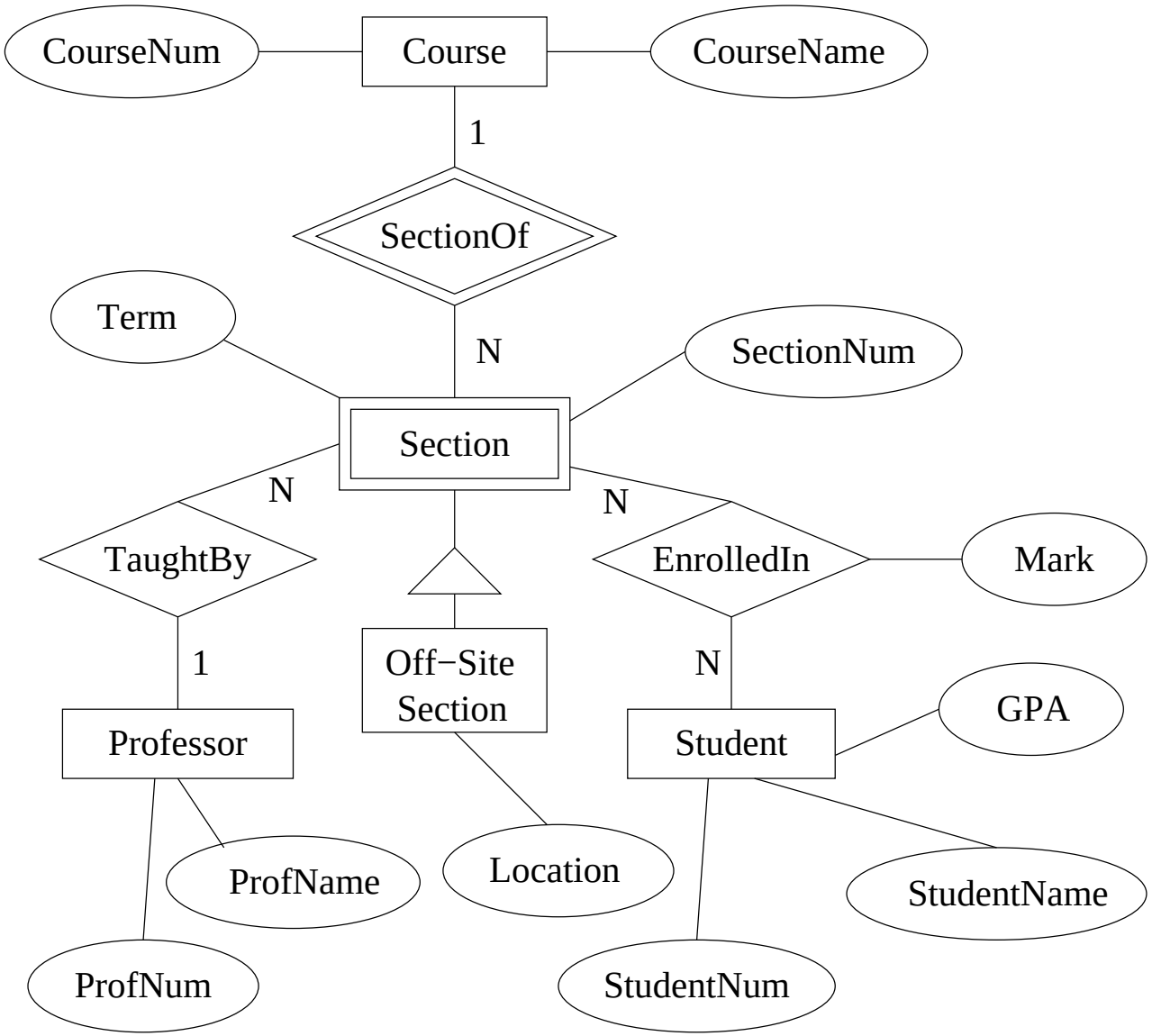
Example: A Registrar's Database (cont'd)



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



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



Example: A Registrar's Database (cont.)