

## The Entity-Relationship (ER) Model

### Lecture Topics

- Basic ER modelling
- Extensions to ER modelling
- Designing an ER schema
- Reducing ER diagrams to relational schema

## Overview of ER Model

Proposed by Peter Chen in 1976

Used for database (conceptual schema) design

World/enterprise described in terms of

- entities
- attributes
- relationships

Visualization: **ER-diagram**

## Basic ER Modelling

**Entity:** a *distinguishable* object

**Entity set:** set of entities of same type

**Ex.**

- students currently at the Institute
- flights offered by Air Canada
- burglaries in Ontario during 1994

**Graphical representation of entity sets:**



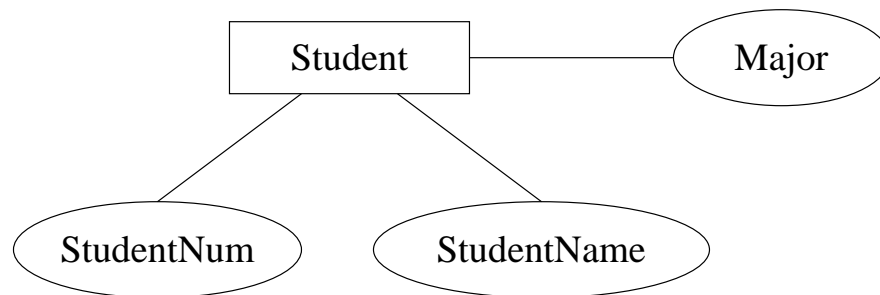
## Basic ER Modelling (cont'd)

**Attributes** describe properties of entities

**Ex.** for Employee-entities: EmpNum, Name, Salary, . . .

**Domain:** set of permitted values for an attribute

**Graphical representation of attributes**



## Basic ER Modelling (cont'd)

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

**Relationship set:** set of relationships of a given type

**Ex.**

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

---

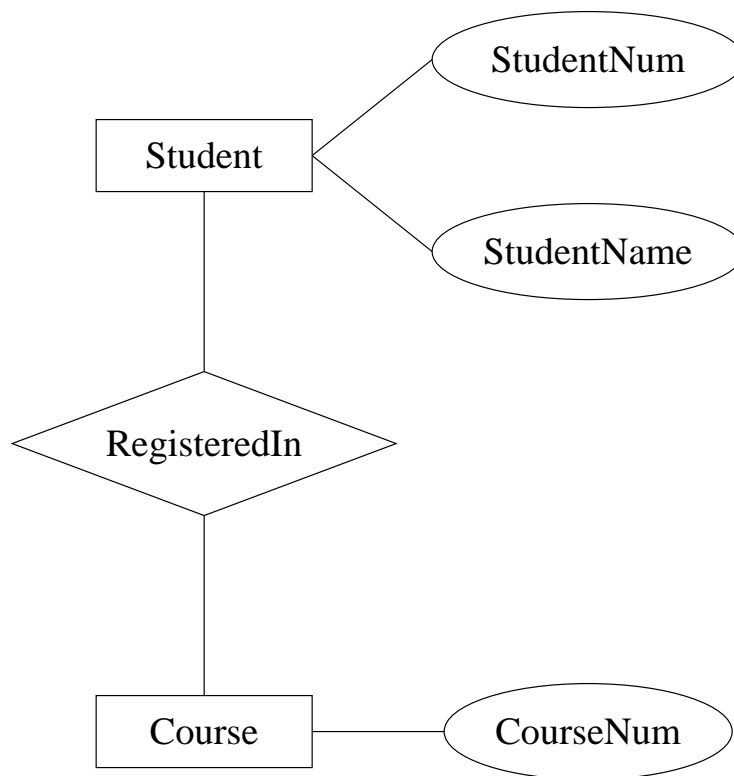
---

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

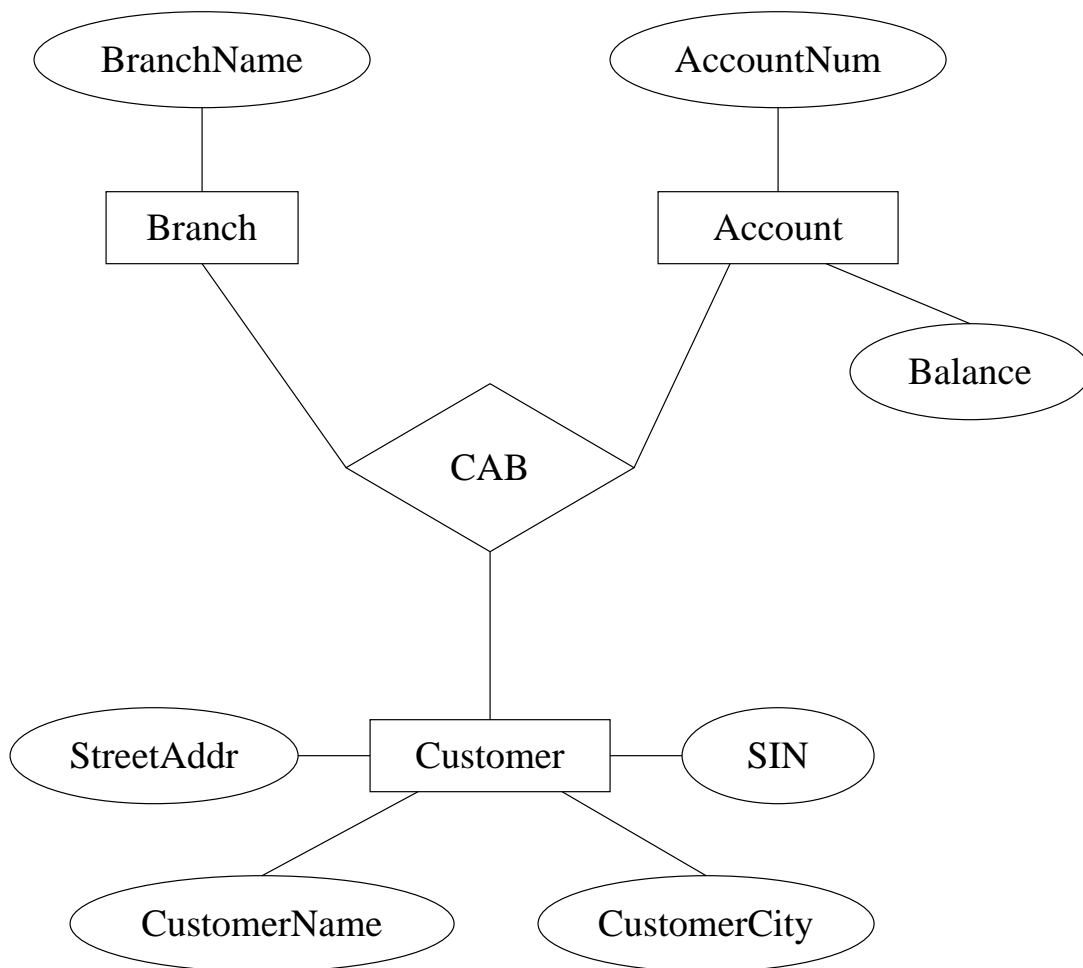
---

---

## Graphical Representation of Relationship Sets



## Graphical Representation of Relationship Sets (cont'd)

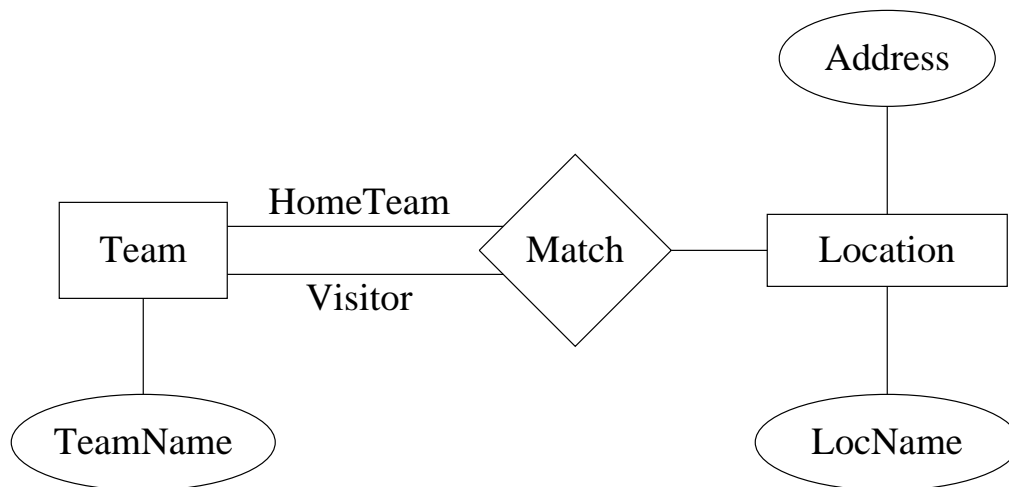


## Recursive Relationships and Role Names

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

**Role name:** an explicit indication of a role

**Ex.**



---

---

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

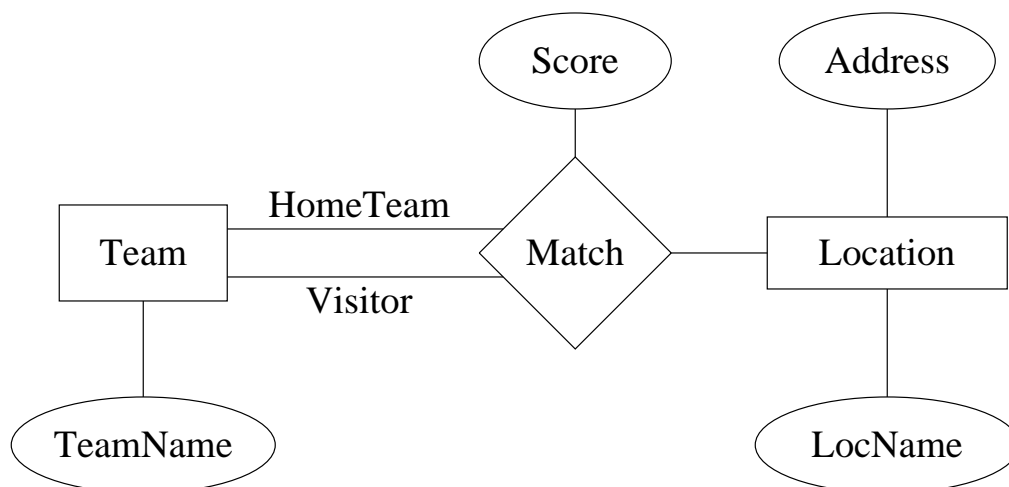
---

---



## Relationships and Attributes

Relationships may also have attributes



## Constraints in ER Models

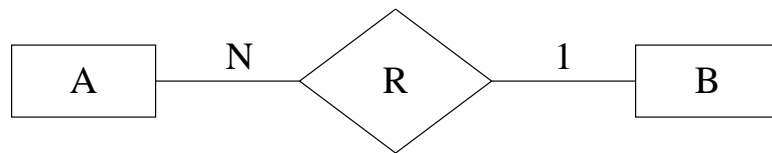
- Binary relationship types
- General cardinality constraints
- Primary keys
- Existence dependencies

## Binary Relationship Types

(relationships between 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:**



**Ex.**



- 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 vice versa

Ex.



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

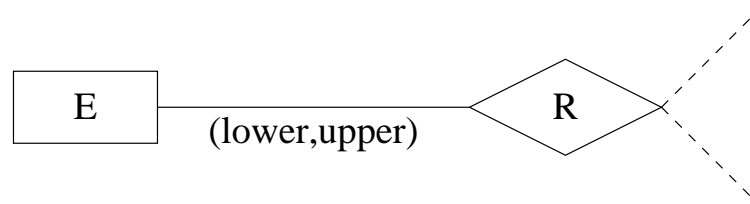
Ex.



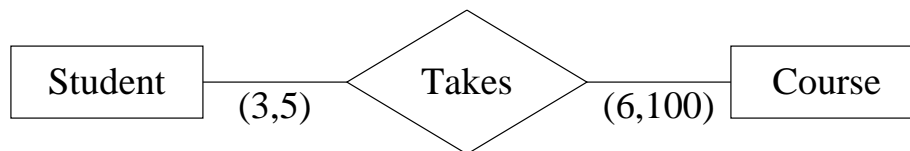
## General Cardinality Constraints

Determine lower and upper bounds on the number of relationships of a given relationship set in which a component entity may participate

### Visualization:



### Ex.

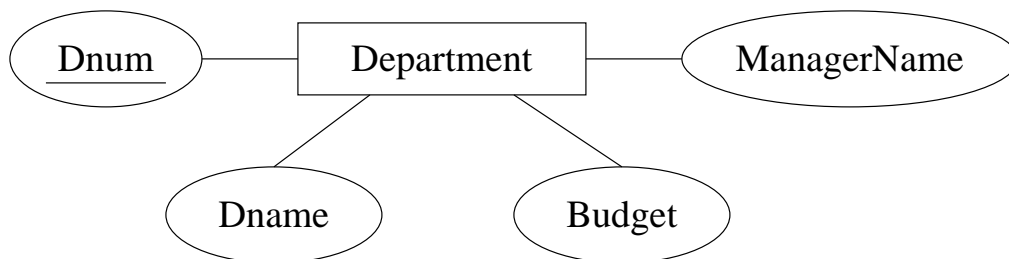


## Primary Keys

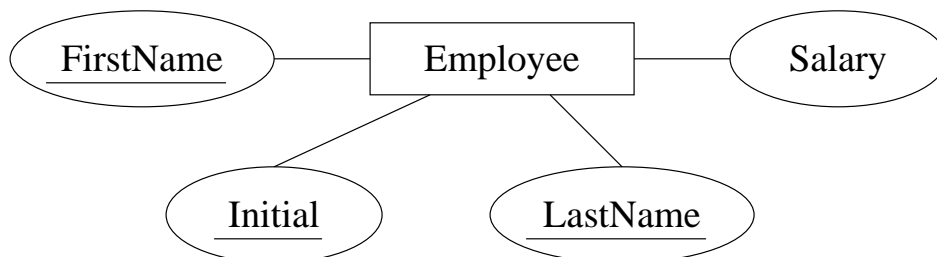
As in relational model, each entity must be distinguishable from any other entity in its set by its attributes

**Primary key:** selection of attributes chosen by designer as a surrogate key of the entity set

**Ex.**



**Ex.**



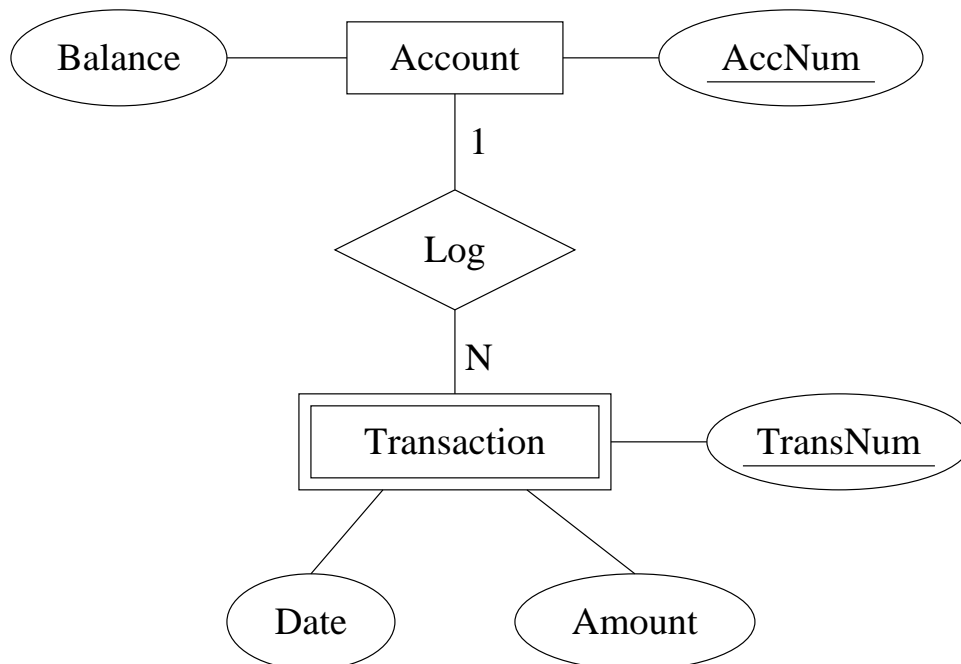
## Existence Dependencies

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**

**Ex.** “Transactions are existence dependent on accounts.”

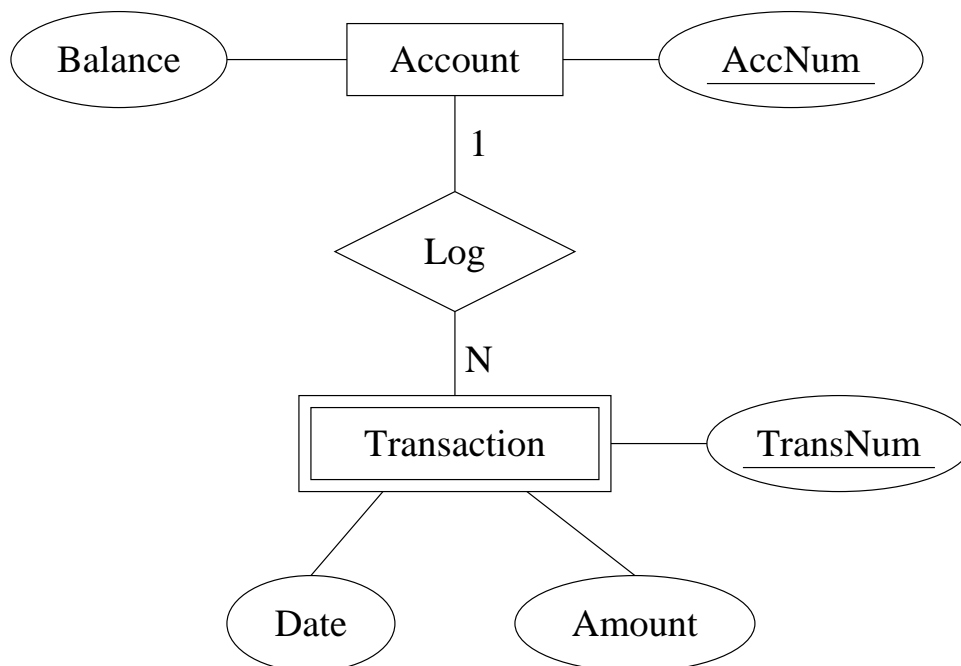


## Identifying Subordinate Entities

Attributes of entity sets with subordinate entities only form key relative to a given dominant entity

- **Weak entity set:** an entity set with subordinate entities
- **Strong entity set:** an entity set with no subordinate entities

**Ex.** “All transactions for a given account have a unique transaction number.”





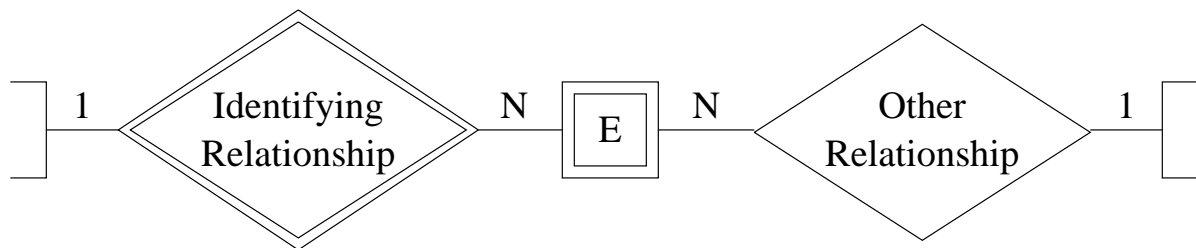
## Identifying Subordinate Entities (cont'd)

A weak entity set must have an N:1 relationship to a distinct entity set

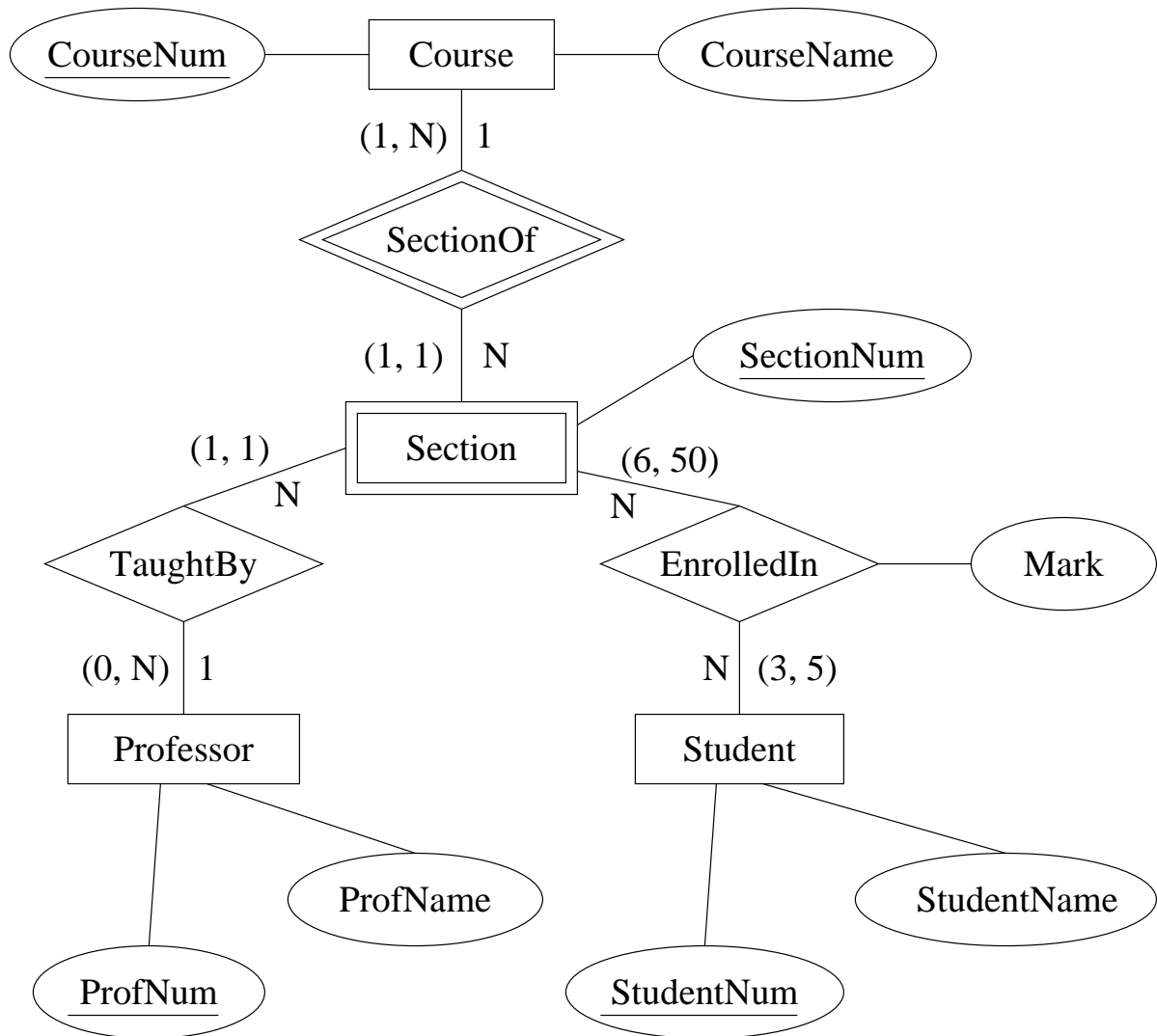
**Discriminator** of a weak entity set: set of attributes that distinguish subordinate entities of the set, for a particular dominant entity

Primary key for a weak entity set: discriminator  $\oplus$  primary key of entity set for dominating entities

**Visualization:** (distinguishing an identifying relationship)



### Example ER Diagram



## **Extensions to ER Modelling**

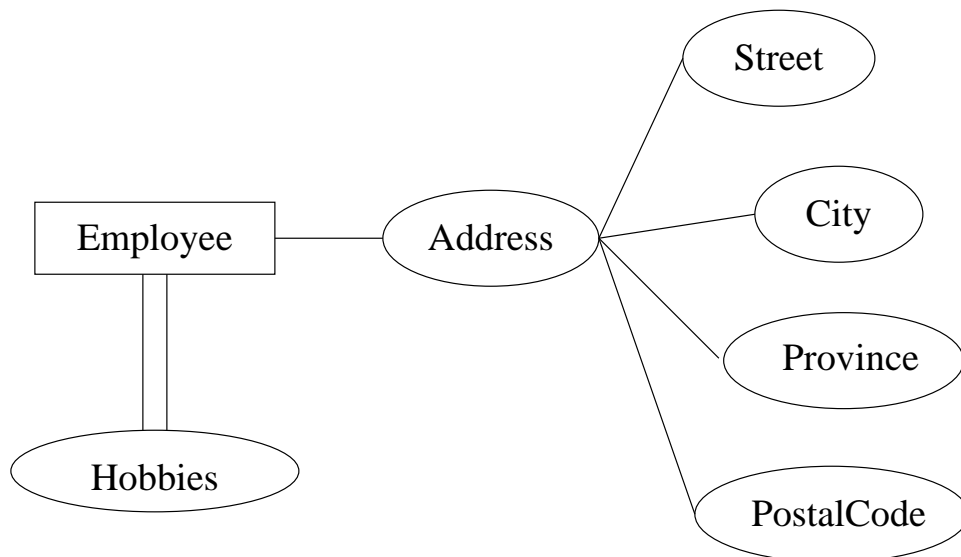
- Structured attributes
- Aggregation
- Specialization
- Generalization

## Structured Attributes

**Composite attributes:** attributes composed of two or more other attributes

**Multi-valued attributes:** attributes that are set-valued

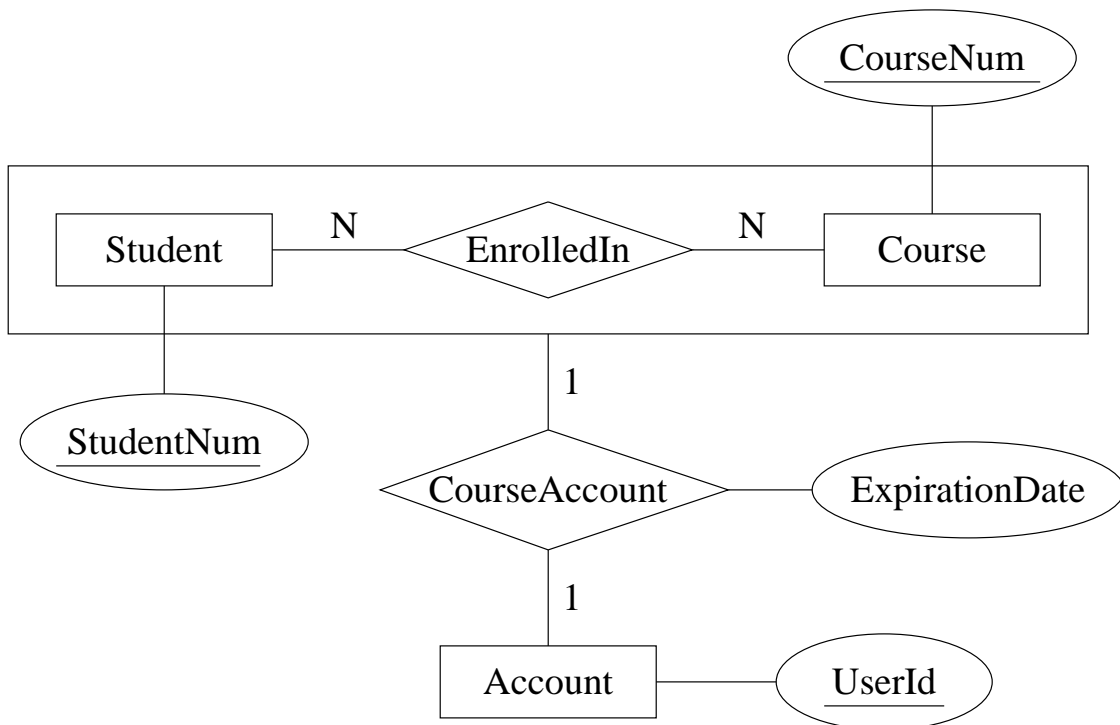
Ex.



## Aggregation

Relationships can be viewed as higher-level entities

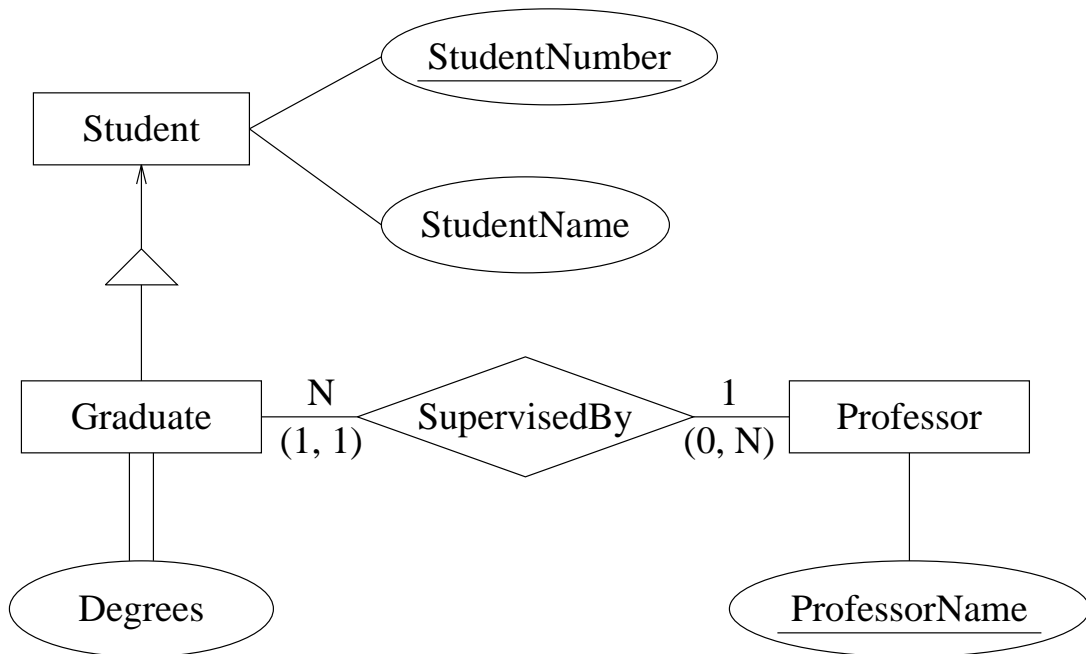
**Ex.** “Accounts are assigned to a given student enrollment.”



## Specialization

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

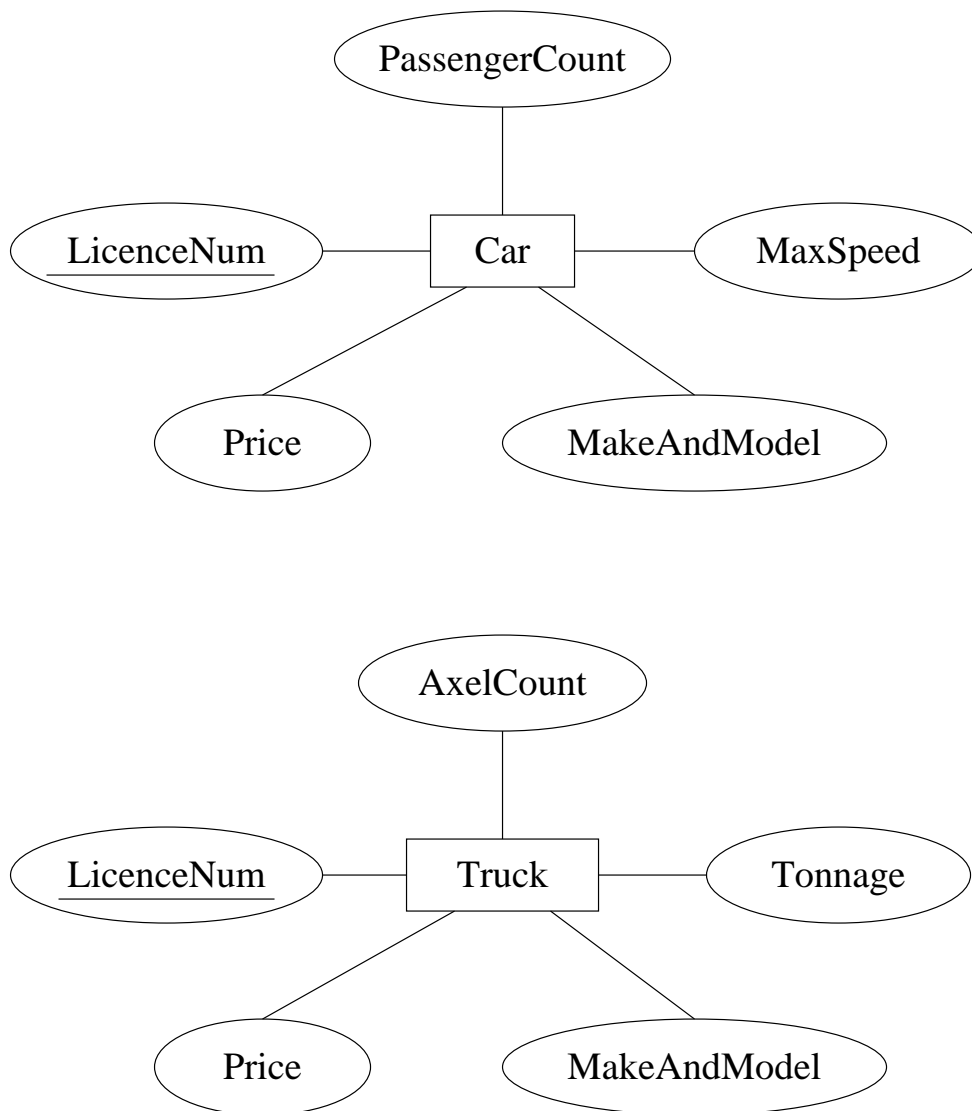
**Ex.** “Graduate students are students that have a supervisor and a number of degrees.”



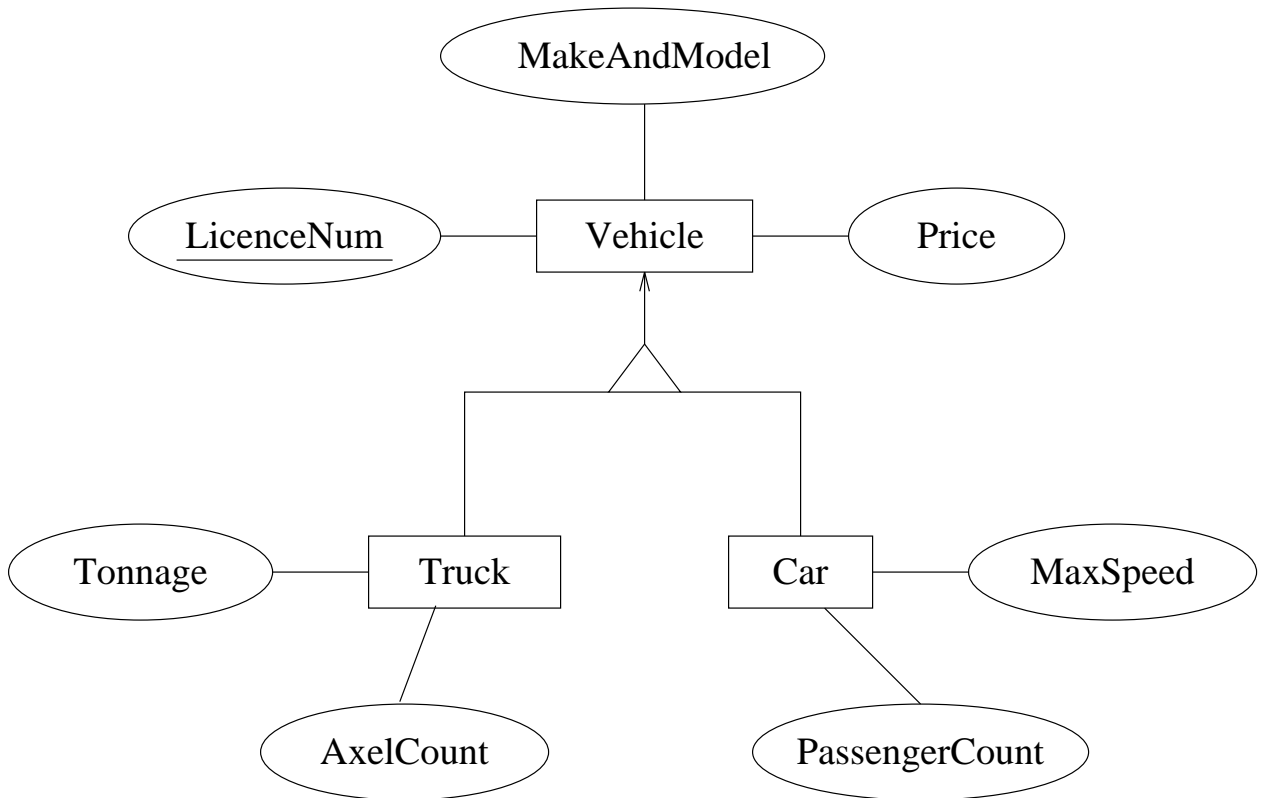
## Generalization

Two or more existing entity sets can be abstracted as a more general kind of entity set

**Ex.** “A vehicle abstracts the notion of a car and a truck.”

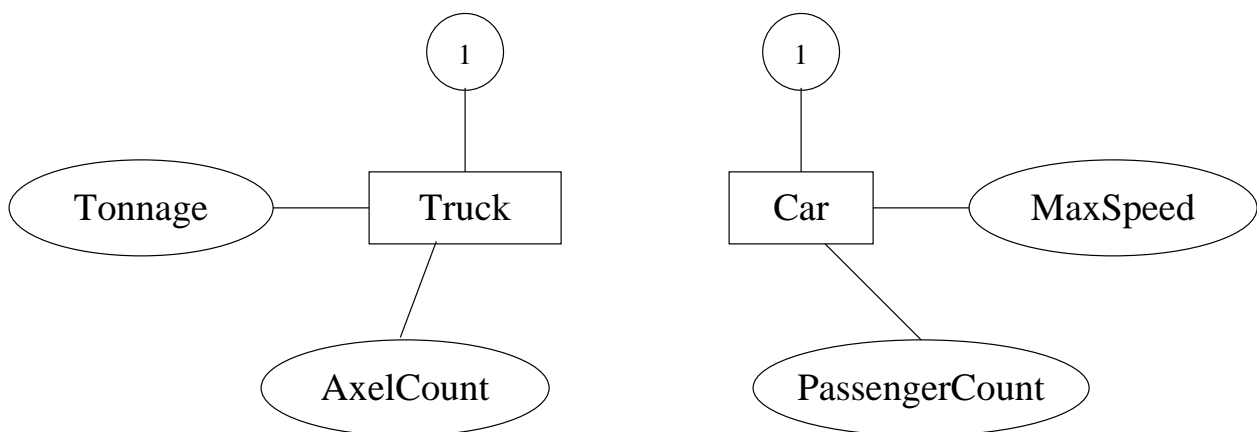


### Generalization (cont'd)





## Expressing Disjointness



## Designing An ER Schema

Usually many ways to design an ER schema

Points to consider

- use attribute or entity set?
- use entity set or relationship set?
- degrees of relationships?
- extended features?

## Choosing Between Attributes and Entity Sets

No simple answer!

**Ex.** 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 implies introducing a new entity set.

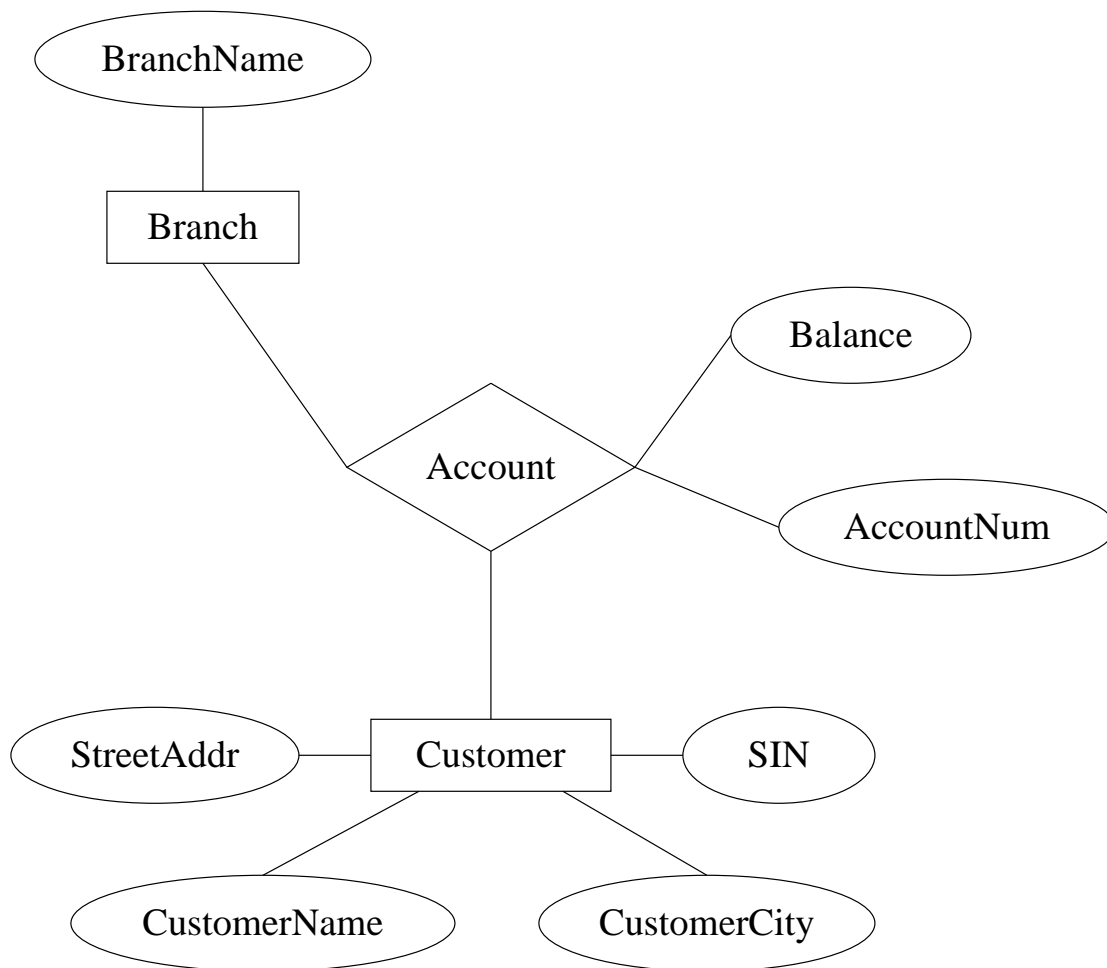
---

---

## Choosing Between Entity Sets and Relationship Sets

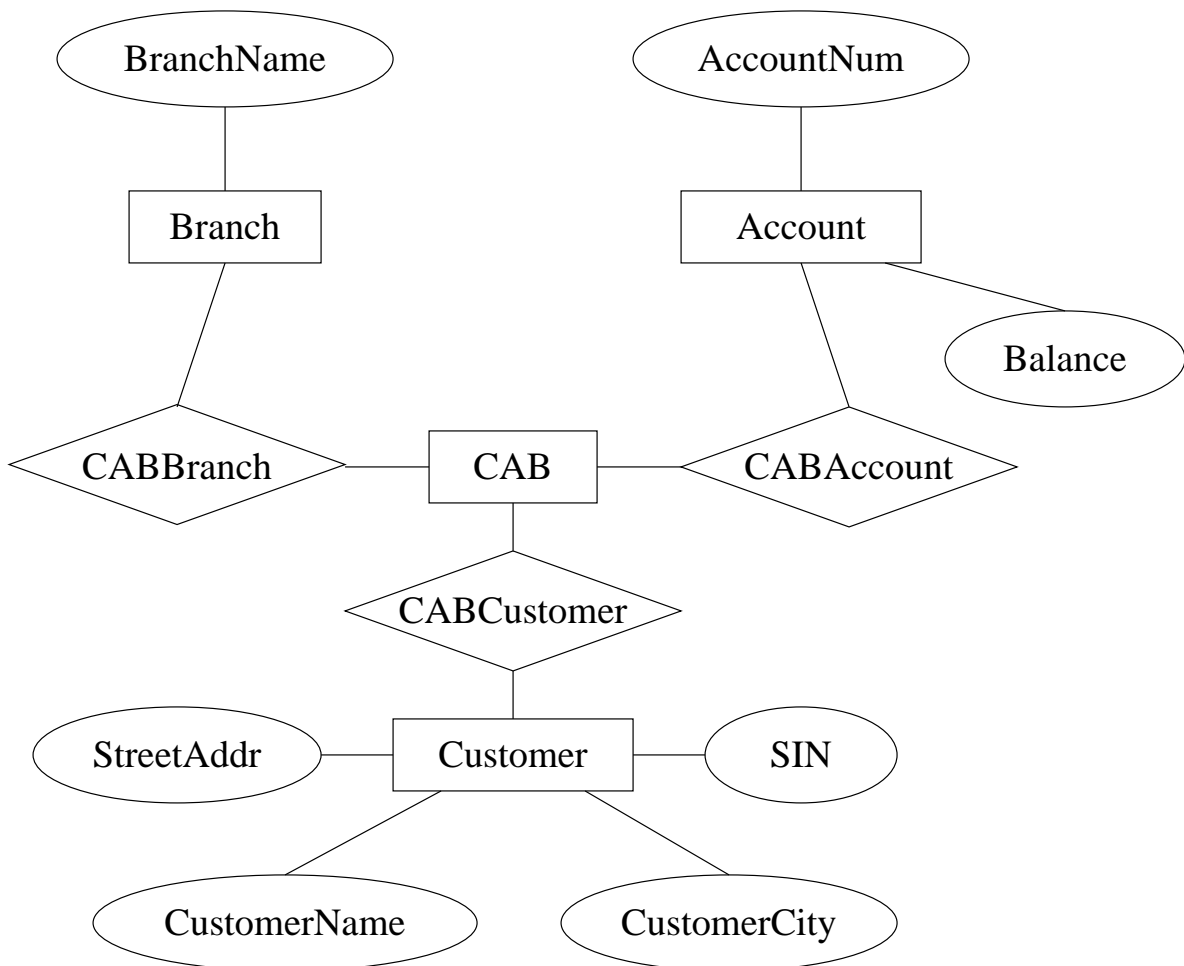
Again no simple answer!

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



## Use of Non-Binary Relationships

Can always represent a relationship on  $n$  entity sets with  $n$  binary relationships



## **Use of Extended ER Features**

Can improve modularity and abstraction if used with care

Excessive use can complicate design

## A Simple 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

---

---

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

---

---