Chapter 4 High-level Database Models

- Entity/Relationship Models (E/R diagram)
- How to transfer to a relational model
- Unified Modeling Language (UML)
- Object Definition Language (ODL)
- Hot to Transfer them to a relational model

Introduction

- 现实世界：客观存在的世界。
- 信息世界：现实世界在人们头脑中的反映。
- 机器世界：信息世界的信息在机器世界中以数据的形式存放。

---

reality-》information world-》 machine world

E-R data model　　relational model√

UML　　object-relational model

ODL　　object-oriented model

Purpose of E/R Model

- Sketch database schema designs
  - Includes some constraints, but not operations.
- Designs are pictures called entity-relationship diagrams.
- convert E/R designs to relational DB designs.

Entity/Relationship Model

- Entity like objects, =things
- Entity set like class = set of similar Entity or objects
- Attribute=property of entities in an entity set, similar to fields of a struct.
- Relation=connect two or more entity set

In diagrams,

entity set : rectangle;

attribute: oval,

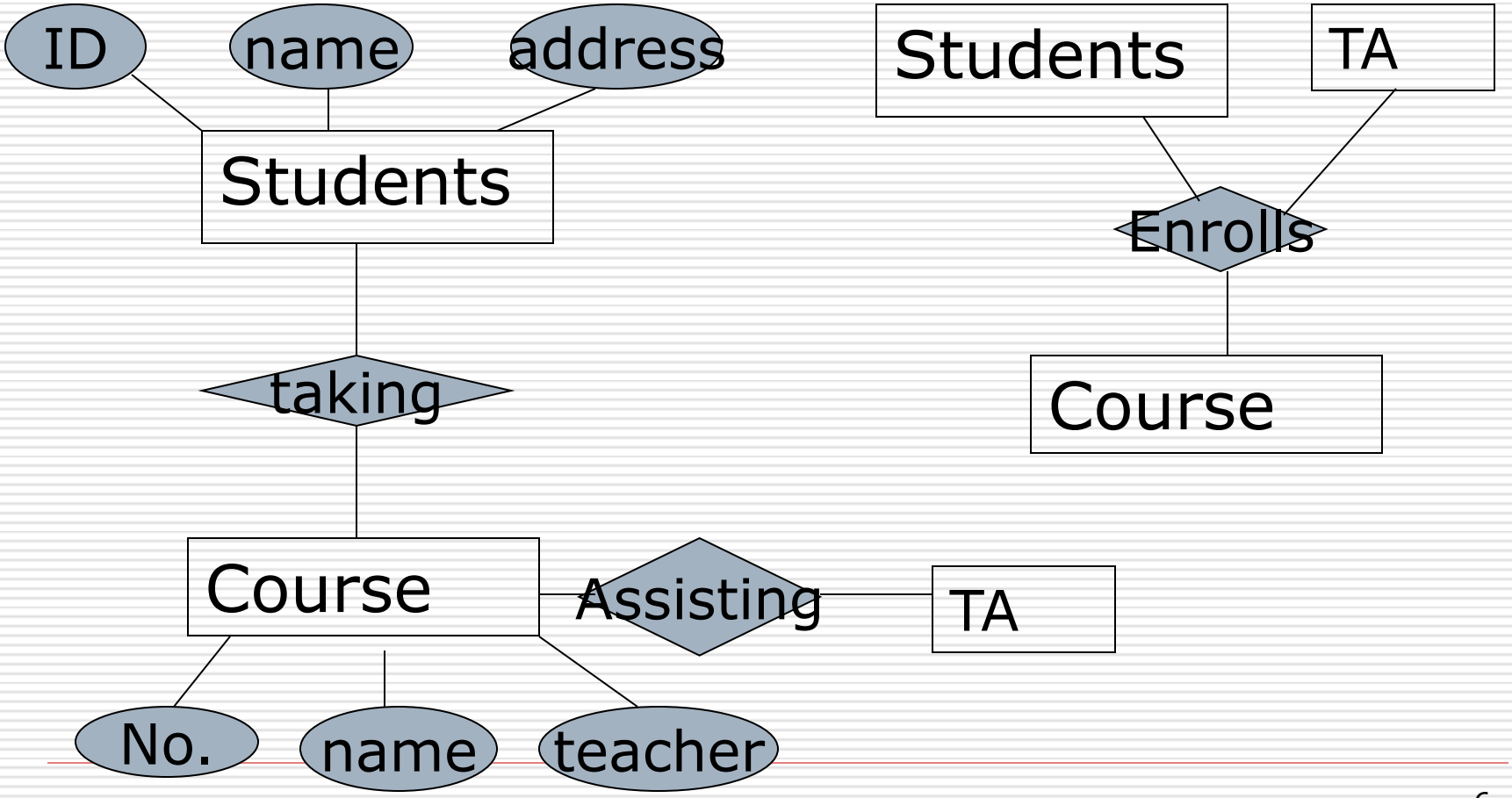
relation: diamonds

Relationships

1. Binary (relation between two entity sets)
2. Multiway (relation between more than two entity sets)

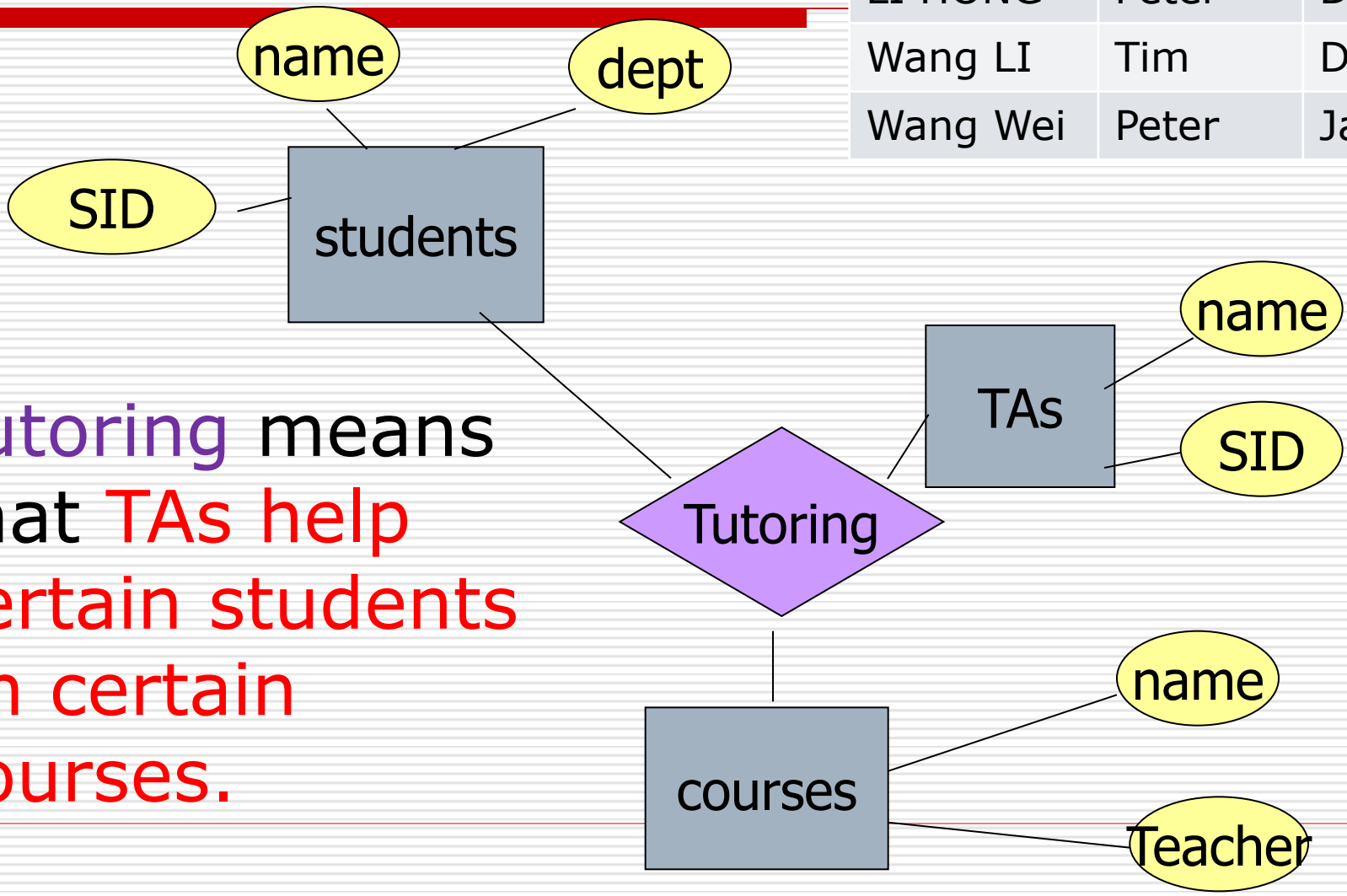
- Multiplicity of relationships: Express **the number of entities** to which another entity can be associated via a relationship set.

Binary & Multiway Relationships



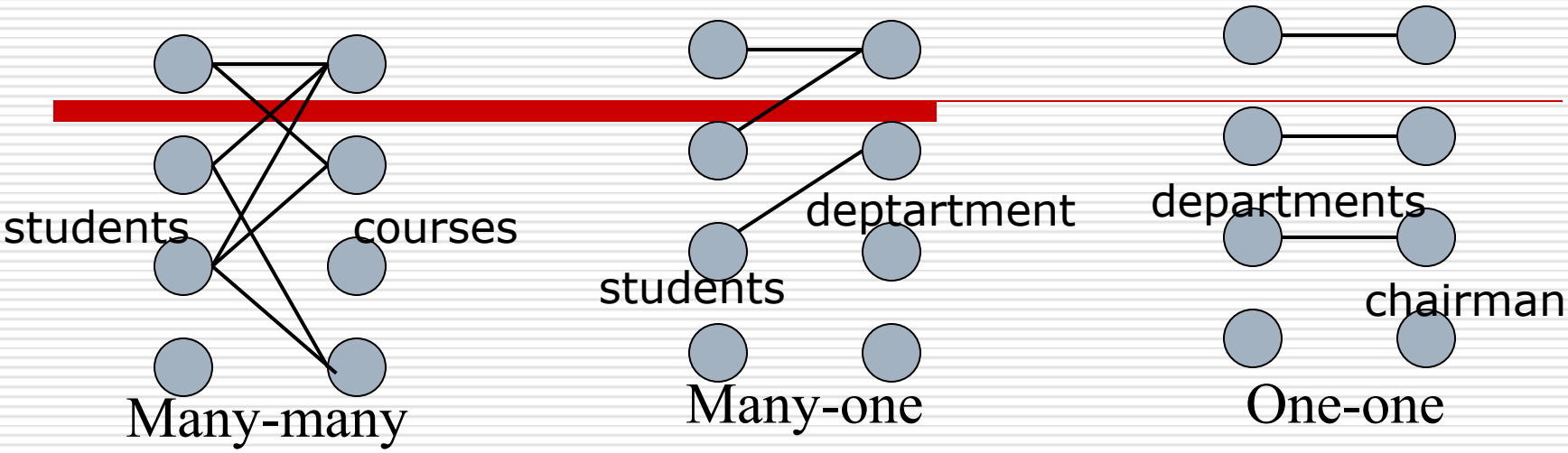
Example: 3-Way Relationship

Students Name	TAs Name	Courses Name
LI HONG	Peter	DB
Wang LI	Tim	DB
Wang Wei	Peter	Java



Tutoring means that TAs help certain students on certain courses.

Multiplicity of Relationships

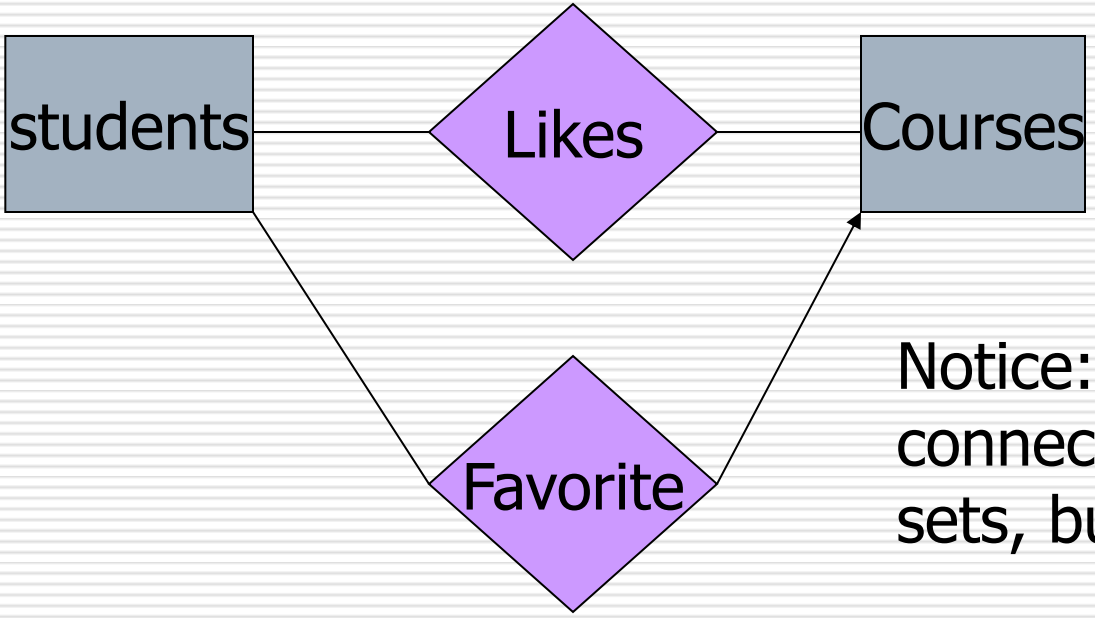


- In a *many-many* relationship, an entity of either set can be connected to many entities of the other set. E.g., a bar sells many beers; a beer is sold by many bars.
- In a many-one relationship, each entity of the first set is connected to at most one entity of the second set. But an entity of the second set can be connected to zero, one, or many entities of the first set.
- In a *one-one* relationship, each entity of either entity set is related to at most one entity of the other set.

Representing "Multiplicity"

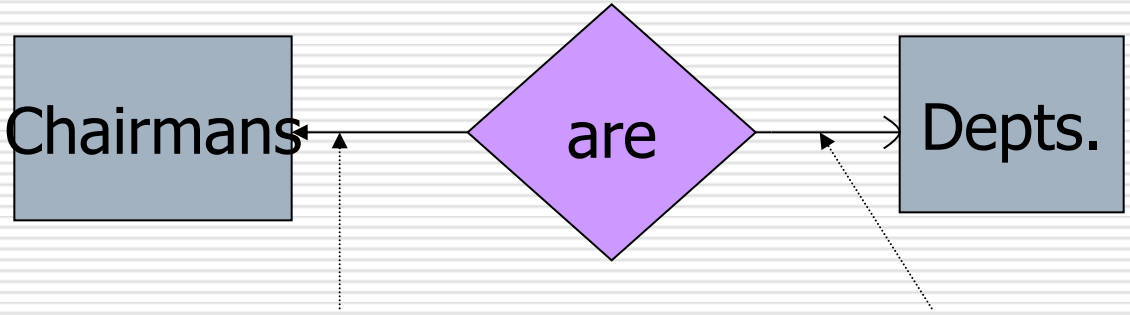
- Show a many-one relationship by an arrow entering the "one" side.
- Show a one-one relationship by arrows entering both entity sets.
- Rounded arrow = "exactly one," i.e., each entity of the first set is related to exactly one entity of the target set.

Example: Many-Many and Many - One Relationship



Notice: two relationships connect the same entity sets, but are different.

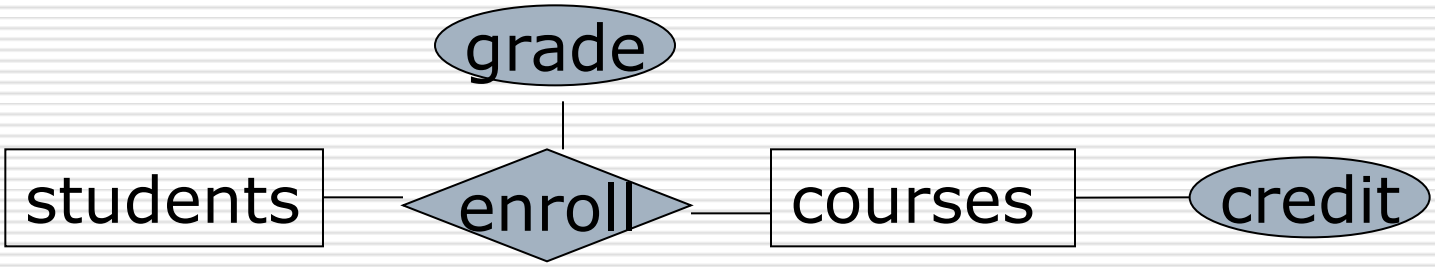
Example: One-One Relationship



A chairman should be in exactly one department.

A department has one chairman.

Attributes on Relationships



- Grade depends jointly on students and courses
- Credit depends only on courses.

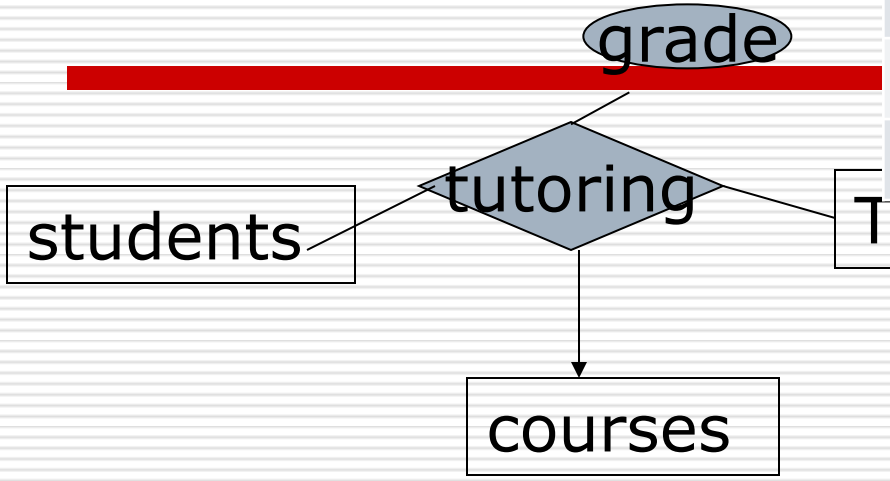
Converting Multiway to 2 way

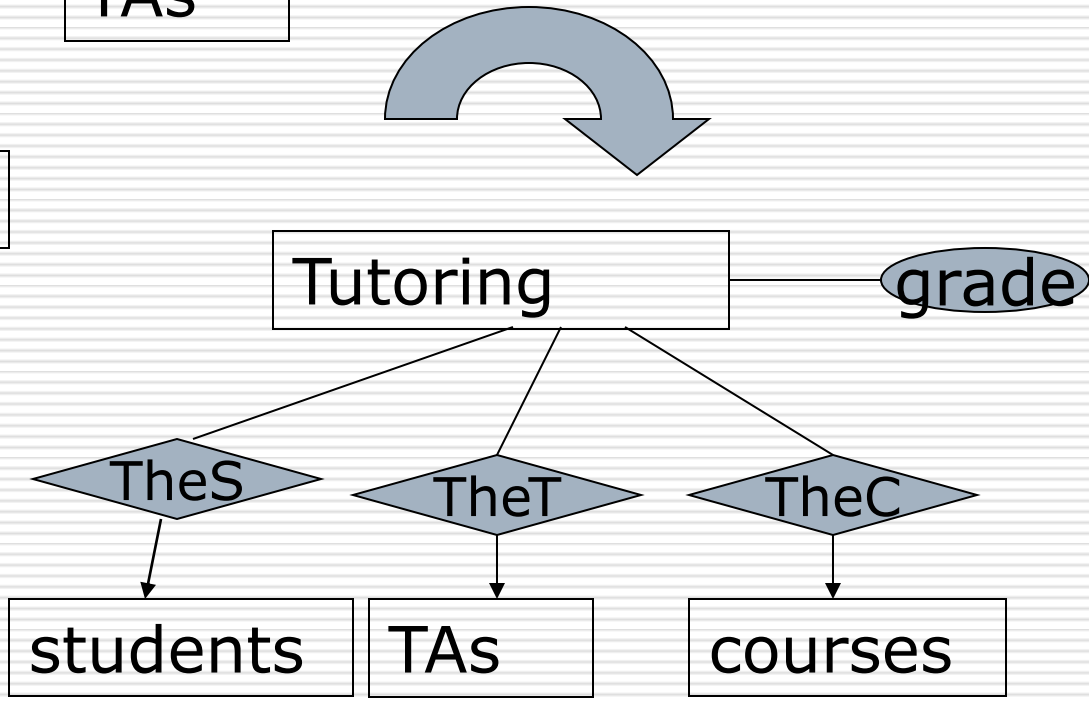
- Motivation: some model can not solve multiway relationship.
- Method:

1. Creating a new connecting E.S. to represent the rows of a relationship set
2. Many-one relationships from the connecting E.S. to the others

Example

Students Name	TAs Name	Courses Name	Grade
LI HONG	Peter	DB	78
Wang LI	Tim	DB	67
Wang LI	Tim	Java	90





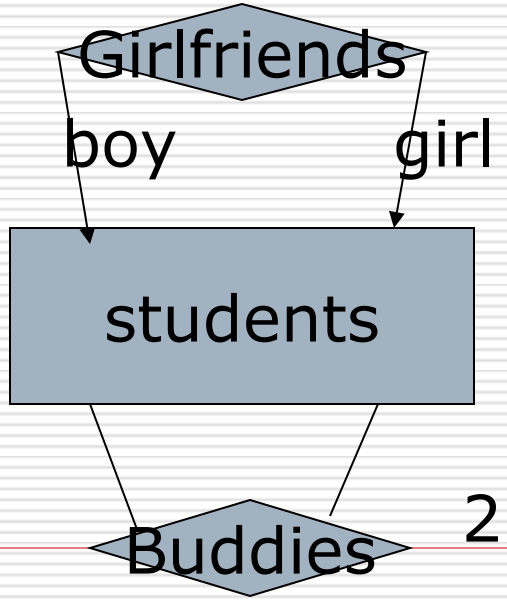
- Relationship Set: like an entity set.
- Each row: an entity
- TheS: many-to relationship

Roles

- an E.S.participates more than once in a relationship.
- Label edges with roles to distinguish.

Married RelationshipSet

Boy	Girl
Bob	Ann
Joe	Sue
…	…



Buddies RelationshipSet

Buddy1	Buddy2
Bob	Ann
Joe	Sue
Ann	Bob
Joe	Moe
…	

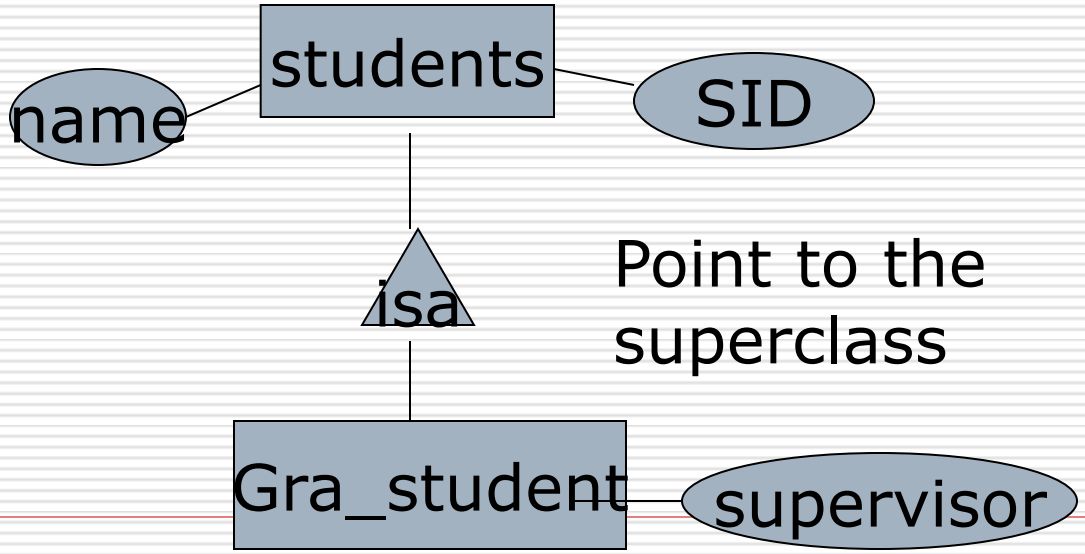
Subclasses

- Subclass = special case = fewer entities = more properties
- Example

  Graduate students are a kind of students with a supervisor.

E/R Subclasses

- Assume subclasses form a tree (no multiple inheritance)
- Isa triangles indicate the subclass relation.



Different Subclass Viewpoints

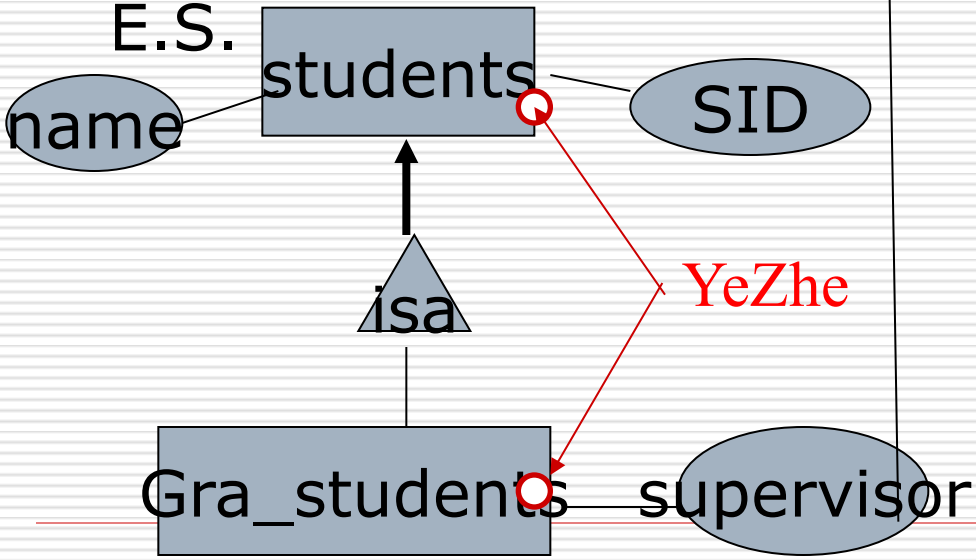
- E/R viewpoint: E/R entities have ***representatives*** in all subclasses to which they belong
  - ✓ Rule: if entity *e* is represented in a subclass, then *e* is represented in the superclass. Its properties are the union of the properties of these E.S.

- Object-oriented viewpoint: An object (entity) belongs to exactly one class. It inherits properties of its superclasses.

Examples for both viewpoints

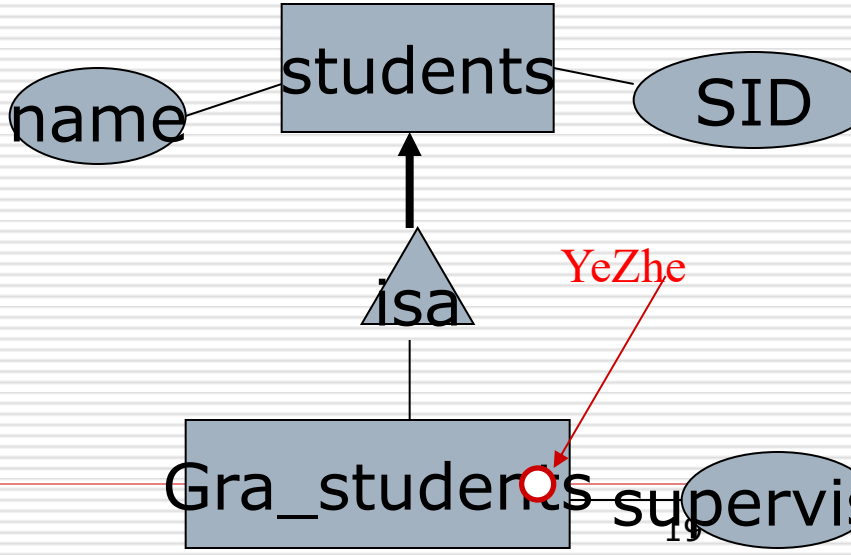
- ER viewpoint:

Its properties are the union of the properties of these E.S.



- OO viewpoint:

subclass inherited all properties of super class



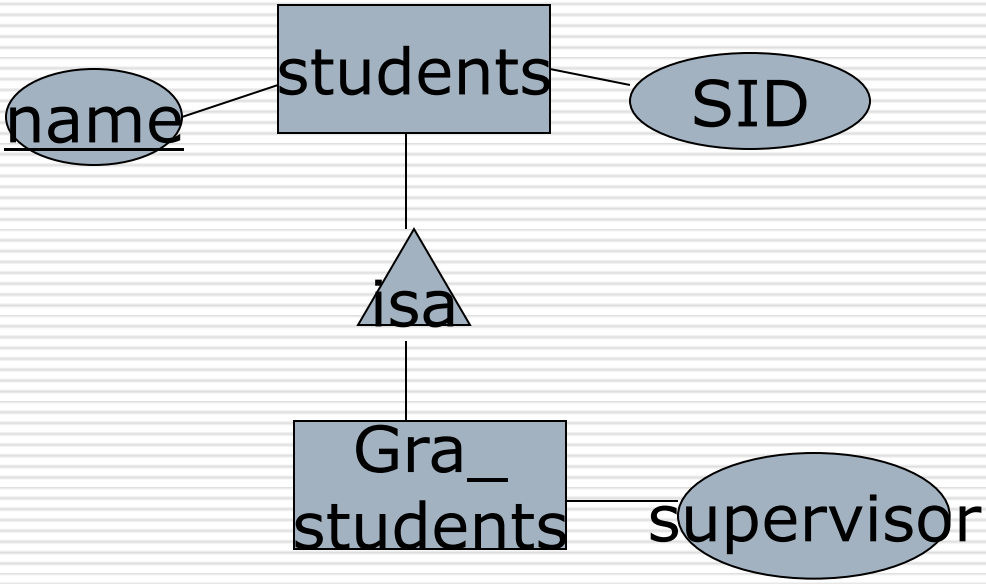
Keys

A key is a set of attributes such that no two entities agree on all these attributes.

- In E/R model, every E.S. must have a key. It could have more than one key, but one set of attributes is the “designated”key.
- In E/R diagrams, you should underline all attributes of the designated key.

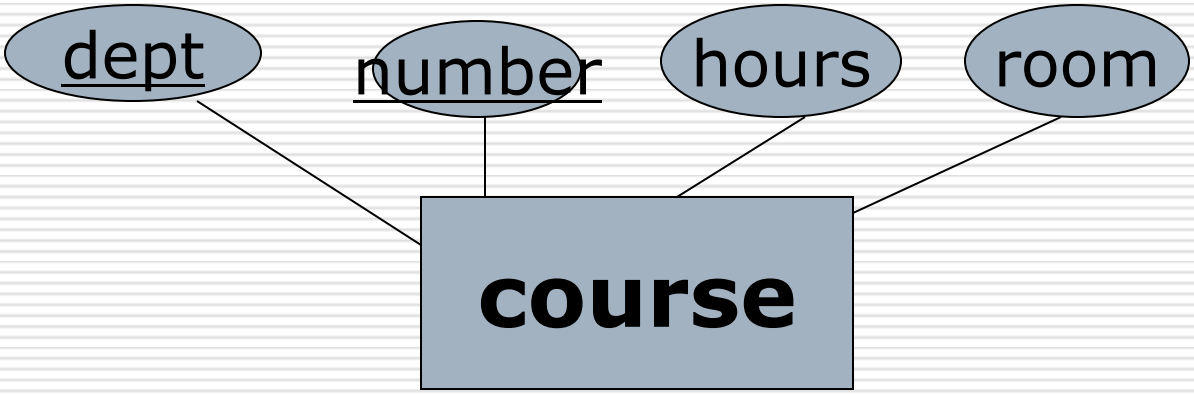
Example

Suppose SID is key for students



SID is also key for Gra_students. In general, key at root is key for all.

Example: A Multiattribute Key



Possibly, hours+room also forms a key, but we have not designed it as such.

Weak Entity Sets

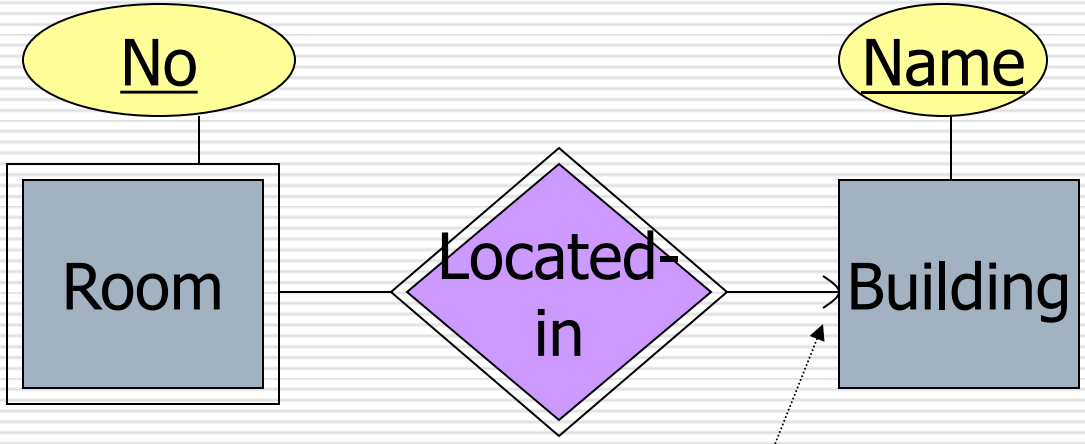
an E.S. E's key comes not (completely) from its own attributes, but from the keys of one or more E.S's to which E is linked by a supporting many-one relationship.

- Represented by putting double rectangle around E and a double diamond around each supporting relationship.
- **Many-one-ness** of supporting relationship (includes 1-1) essential.
- "**Exactly one**" also essential, or else we might not be able to extract key attributes by following the supporting relationship.

Example of Weak Entity Sets

- name is almost a key for football players, but there might be two with the same name.
- number is certainly not a key, since players on two teams could have the same number.
- But number, together with the team name related to the player by Plays-on should be unique.

In E/R Diagrams



(rooms) number and (building) name is a key for Room

Note: must be rounded because each room locates in a building.

- Double diamond for *supporting* many-one relationship.
- Double rectangle for the weak entity set.

Summarization of weak entity set

Suppose E is a weak entity set, R is a supporting relationship, F is the another entity set:

- The key of E consists of its own attributes and key attributes of F.
- R must be a many-one relationship.
- The attributes that F supplies for the key of E must be key attributes of F.
- Weakness can be chained.
- connecting entity set is a weak entity set.

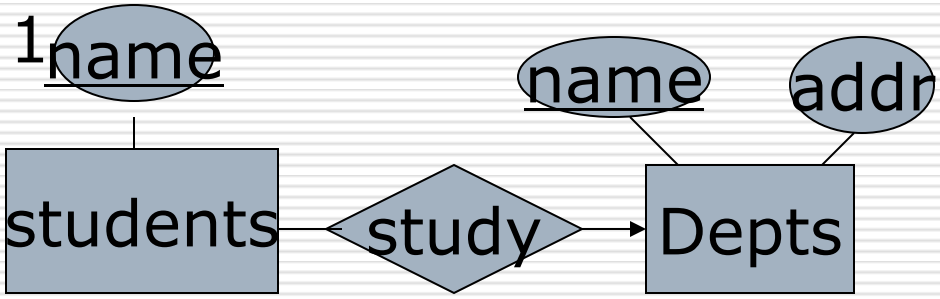
Design Techniques

1. Avoid redundancy.
2. Limit the use of weak entity sets.
3. Don’t use an entity set when an attribute will do.

Avoiding Redundancy

- *Redundancy* = saying the same thing in two (or more) different ways.
- Wastes space and (more importantly) encourages inconsistency.
  - Two representations of the same fact become inconsistent if we change one and forget to change the other.
  - Recall anomalies due to FD's.

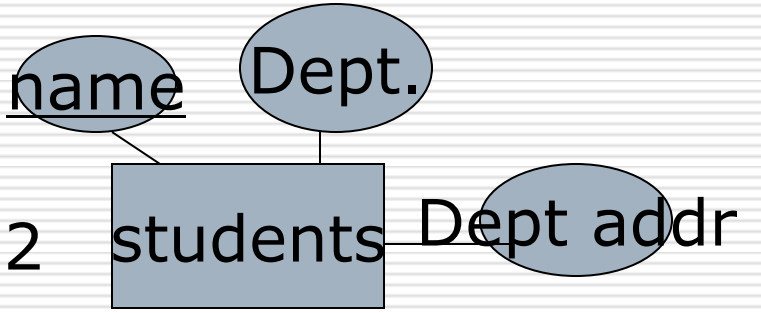
Example: Which one is better?

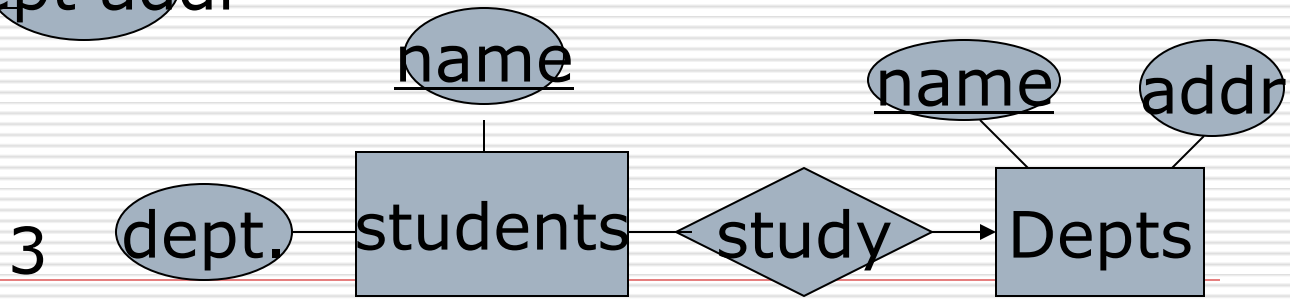


1:good

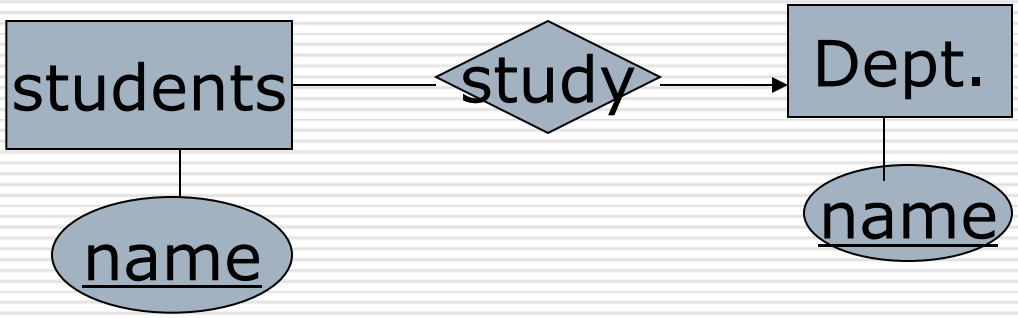
2:repeats dept address for each students in the department.

3:dept's name said twice.

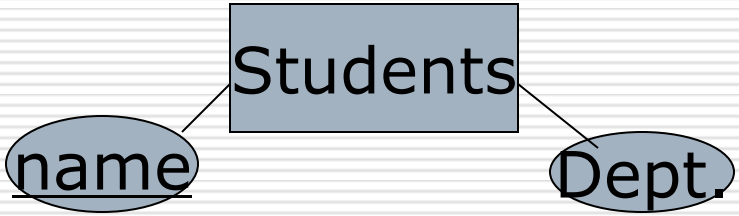




Entity Sets Vs. Attributes



Wrong ??



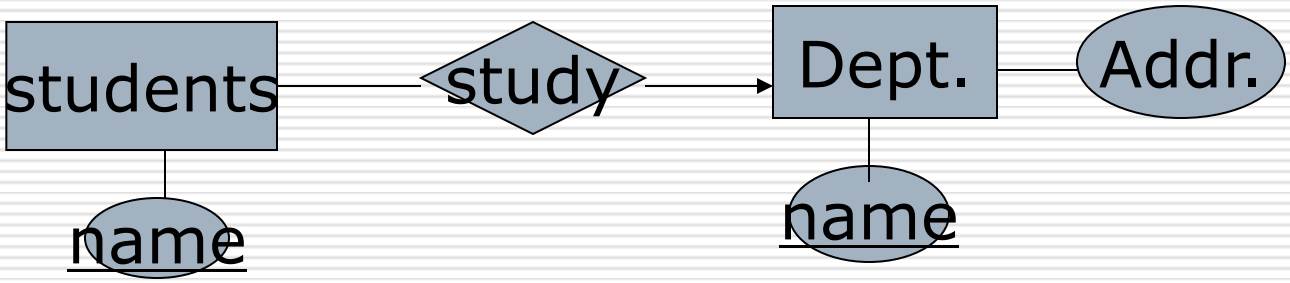
Right !!

Intuitive Rule for E.S. vs. Attribute

Make an entity set only if it either:

1. Is more than a name of something; i.e., it has nonkey attributes or relationships with a number of different entity sets, or
2. Is the "many" in a many-one relationship.

Example



- Dept. Deserves to be an E.S. because we record addr, a nonkey attribute.
- students deserves to be an E.S. because it is at the end of the “many” end.

Don't overuse Weak E.S.

- Create unique ID's to compensate: Product-ID, Player-ID, etc.
- Weak E.S.'s are necessary when:

1. Such ID's are not easily created; e.g., "species ID" as part of the standard nomenclature.
2. No global authority to create them, e.g., crews and studios

Constraints in the E/R Model

- Key constraints
- Single-value constraints
- Referential integrity constraints
- Other constraints

Constraints are part of the model.

Key constraints

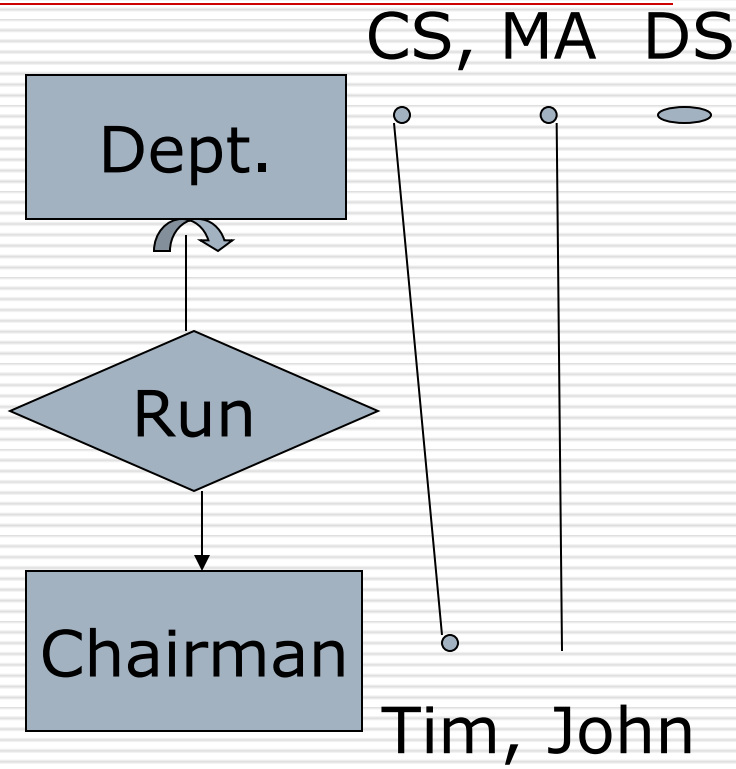
- No two entities may agree in their values for all of the attributes that constitute a key.
- A key may consist of more than one attribute.
- There can also be more than one possible key for an entity set.

Single-value constraints

- Many ways to express:
- Each attribute of an entity set has a single value. (not null)
- A relationship R that is many-one from entity set E to entity set F implies a single-value constraint. (at most one, or exactly one)

Referential integrity constraints

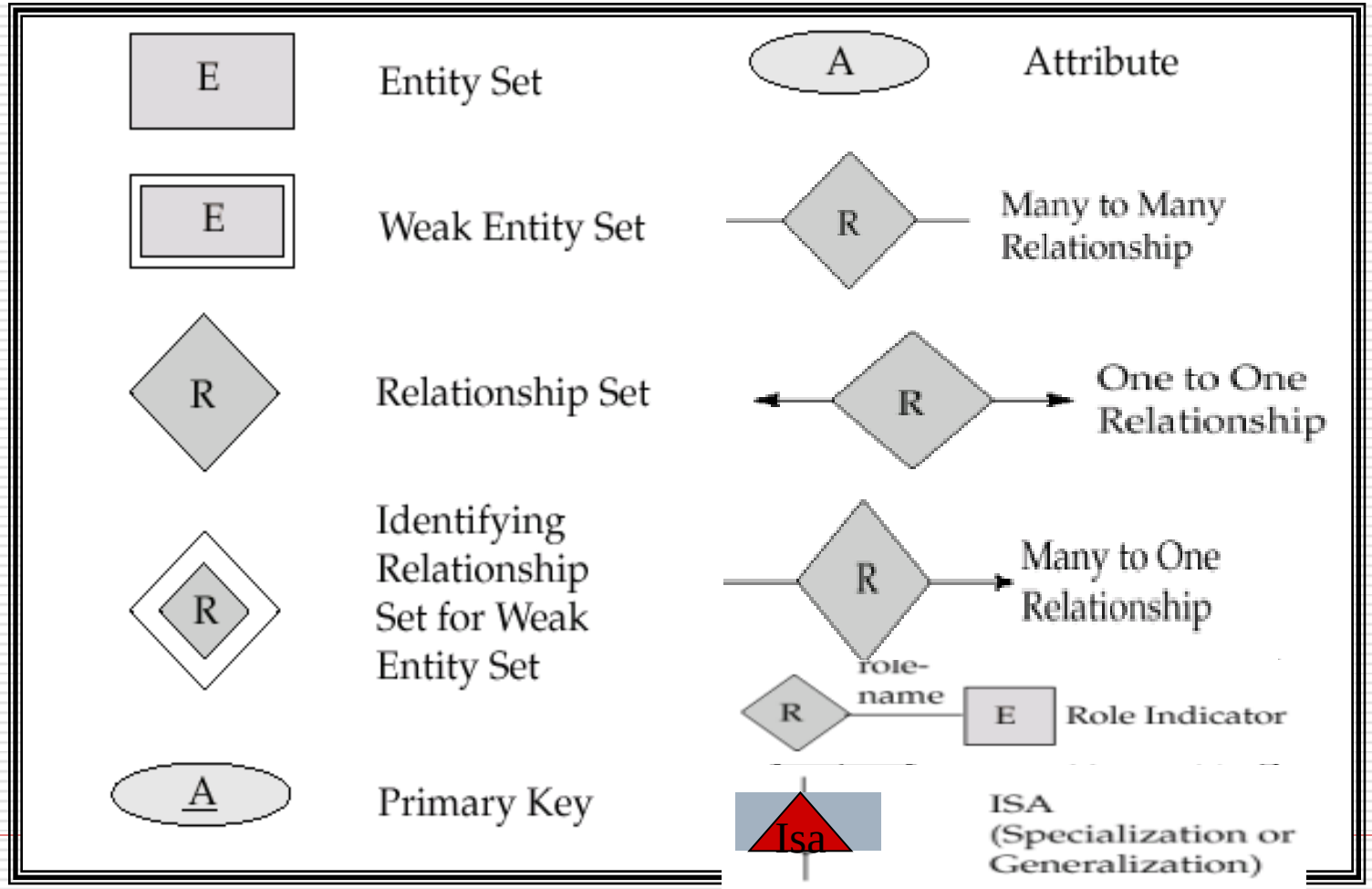
- Every chairman must duty for a department.
- Every department has at most one chairman, that means sometimes, no chairman.
- A rounded arrowhead indicate exactly one (not allowed zero)



Other constraints

- Domain constraints restrict the value of an attribute to be in a limited set.
- General constrains, such as placing a constraint on the degree of a relationship, number constraints and so on.

Summary of Symbols Used in E-R Notation



Summary of E/R diagram

- Entity-Relationship Diagrams

Entities & Attributes & Relationships
Binary and Multiway of Relationships
Multiplicity, role of relationship
Weak Entity Sets, Supporting Relationship
Subclasses, key

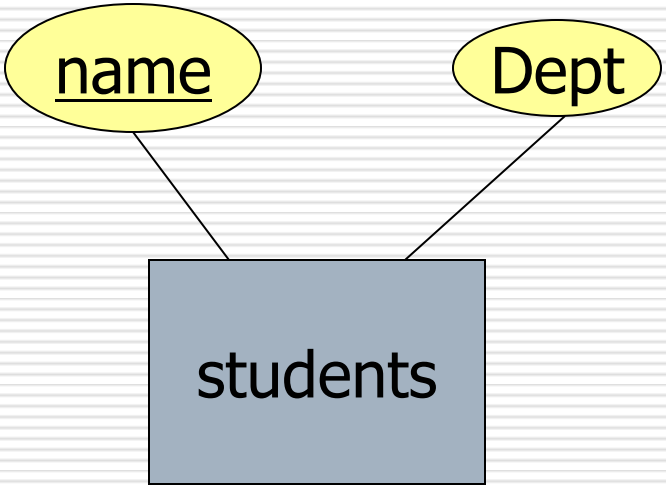
- Good Design

Faithfully represent
Avoid redundancy
Choose appropriate elements

From E/R Diagrams to Relations

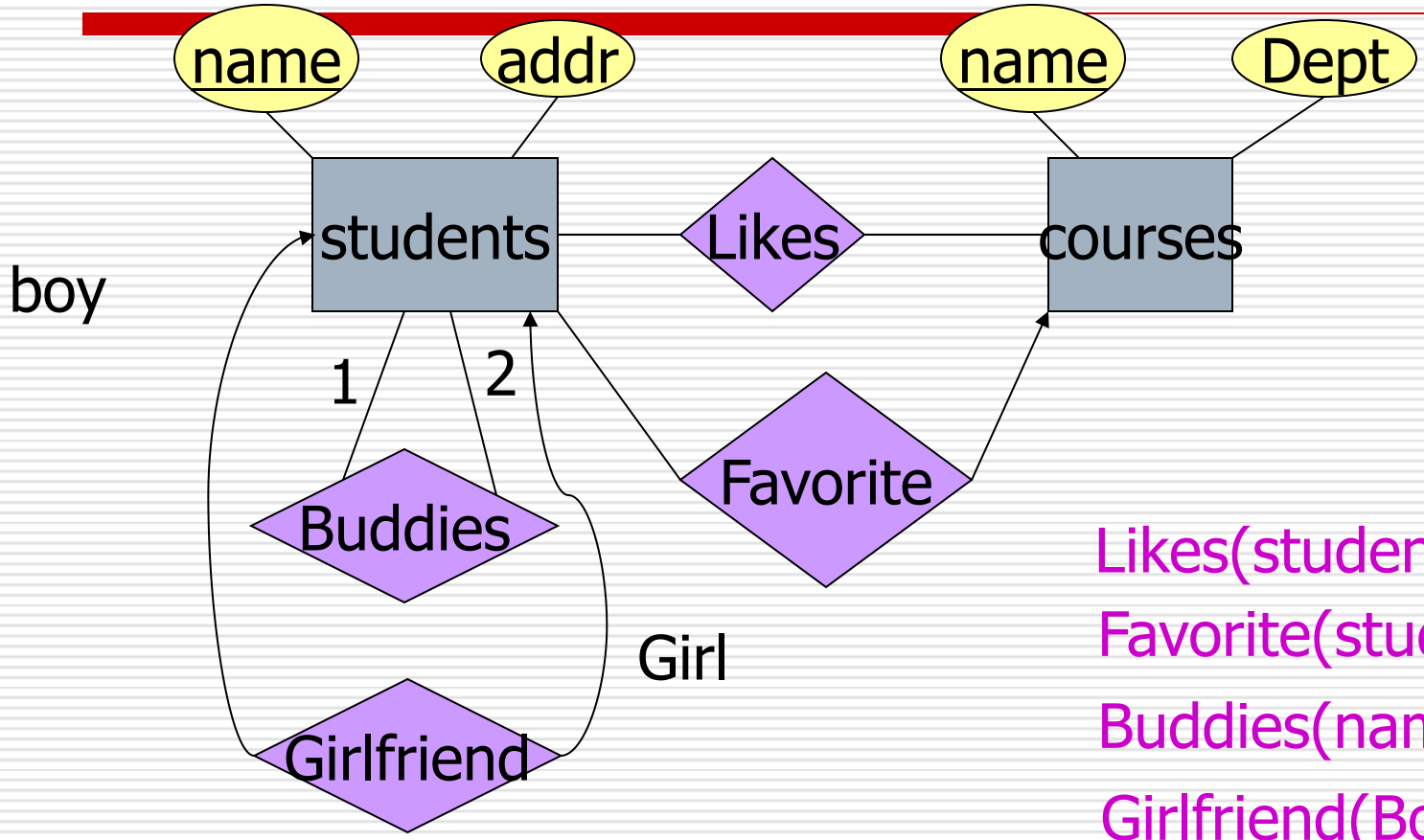
- Entity set -> relation.
  - Attributes -> attributes.
- Relationships -> relations whose attributes are only:
  - The keys of the connected entity sets.
  - Attributes of the relationship itself.

Entity Set -> Relation



Relation: Students(name, Dept)

Relationship -> Relation



Likes(student, course)

Favorite(student, course)

Buddies(name1, name2)

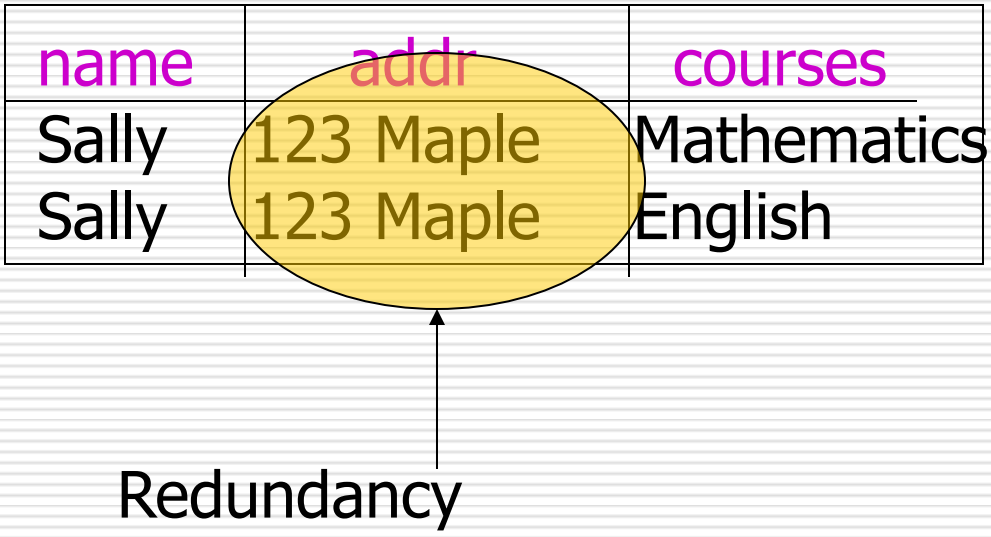
Girlfriend(Boy, Girl)

Combining Relations

- OK to combine into one relation:
  1. The relation for an entity-set *E*
  2. The relations for many-one relationships of which *E* is the "many."
- Example: Students(name, addr) and Favorite(student, course) combine to make students1(name, addr, favCourse).

Risk with Many-Many Relationships

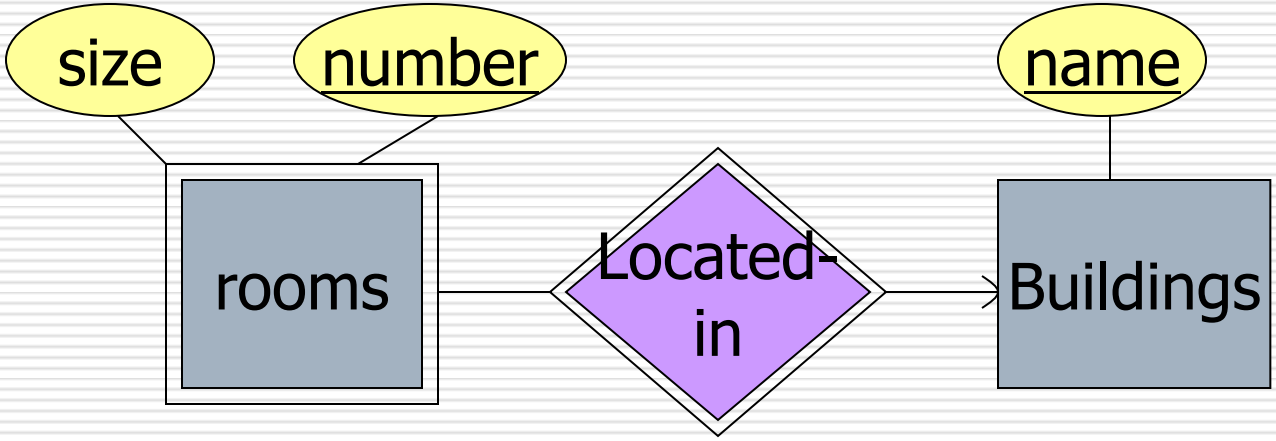
- Combining Courses with Courses would be a mistake. It leads to redundancy, as:



Handling Weak Entity Sets

- Relation for a weak entity set must include attributes for its complete key (including those belonging to other entity sets), as well as its own, nonkey attributes.
- A supporting relationship is redundant and yields no relation (unless *it* has attributes).

Example: Weak Entity Set -> Relation



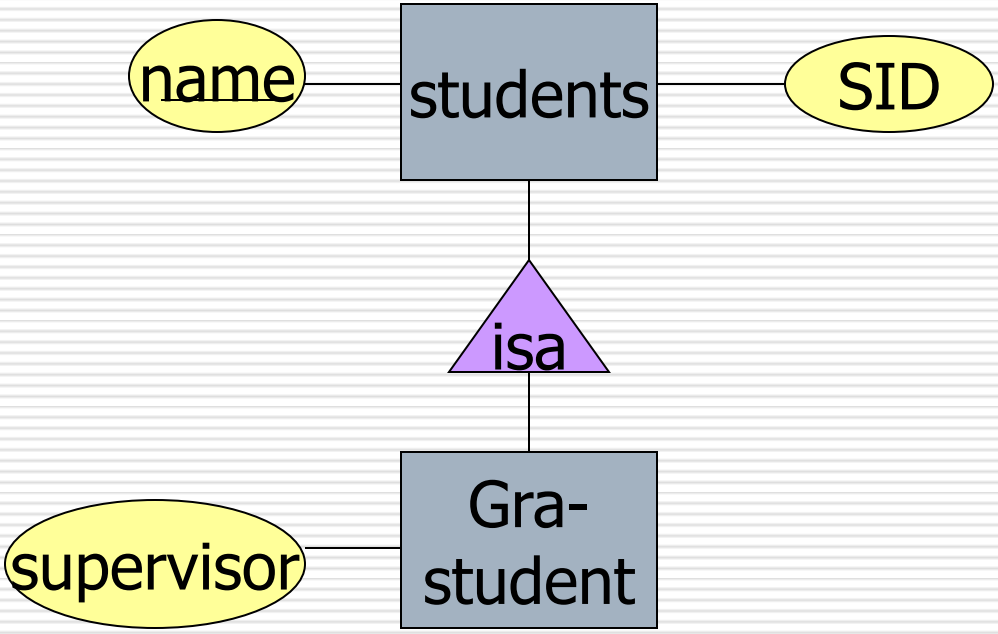
- Rooms(name,number,size)
- Buildings(name)
- ~~Located-in(number,name)~~

Located-in becomes part of rooms

Subclasses: Three Approaches

1. *Object-oriented* : One relation **per subset of** subclasses, with all relevant attributes.
2. *Use nulls* : One relation; entities have NULL in attributes that don't belong to them.
3. *E/R style* : One relation for **each subclass**:
   - Key attribute(s).
   - Attributes of that subclass.

Example: Subclass -> Relations



Object-Oriented

students:

name	SID
LI hone	030001

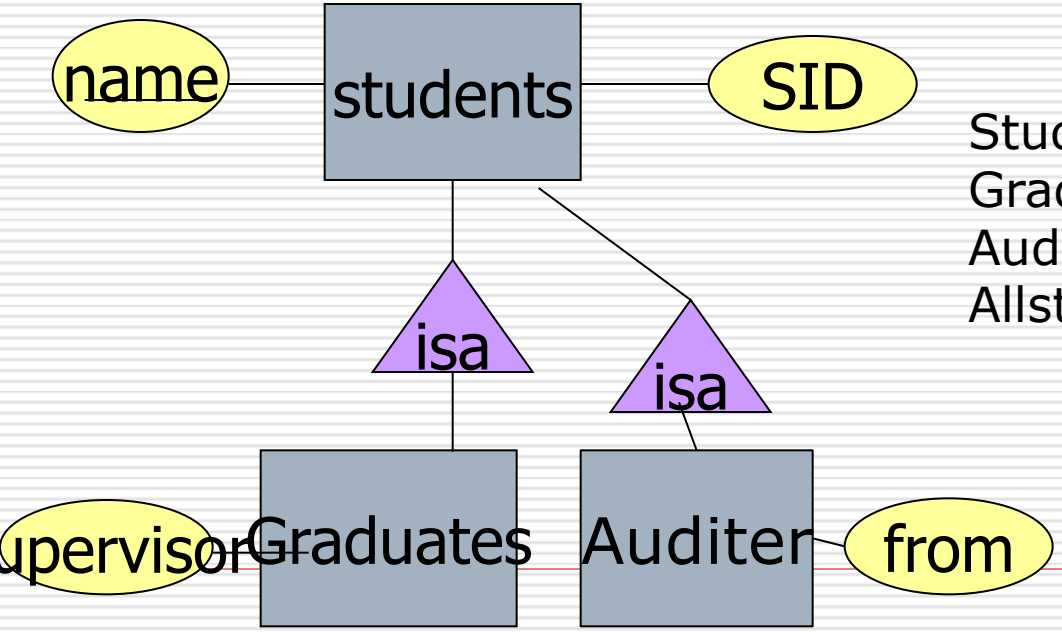
Graduates:

name	SID	Supervior
YeZhe	030987	LF

Good for queries like “find the supervisor of YeZhe”

Object-Oriented (cont.)

- Converting isa-hierarchies to relations is to enumerate all the possible subtrees of the hierarchy.



Students(name,SID)
Graduates(name,SID,supervisor)
Auditer(name,SID,from)
Allstudents(name,SID,supervisor,from)

E/R Style

students

LI Hone	030001
YeZhe	**030987**

Graduates

Yezhe	**030987**	**LF**

Good for queries like “find all students (including Graduates students)”

Using Nulls

students:

name	SID	Supervisor
Li hone	030001	NULL
Yezhe	030987	LF

Saves space unless there are *lots* of attributes that are usually NULL.

Homework

- Read section 4.6.1 and 4.6.2