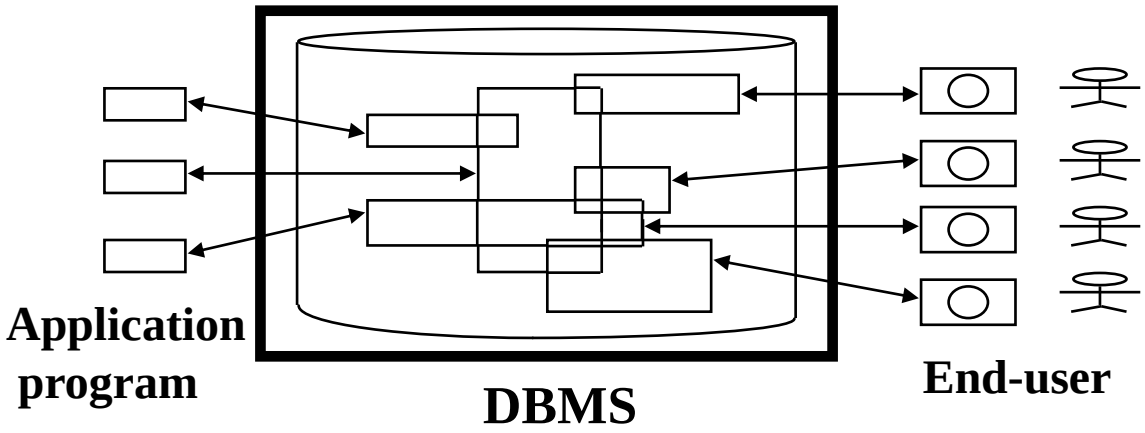
UML Data Modeling

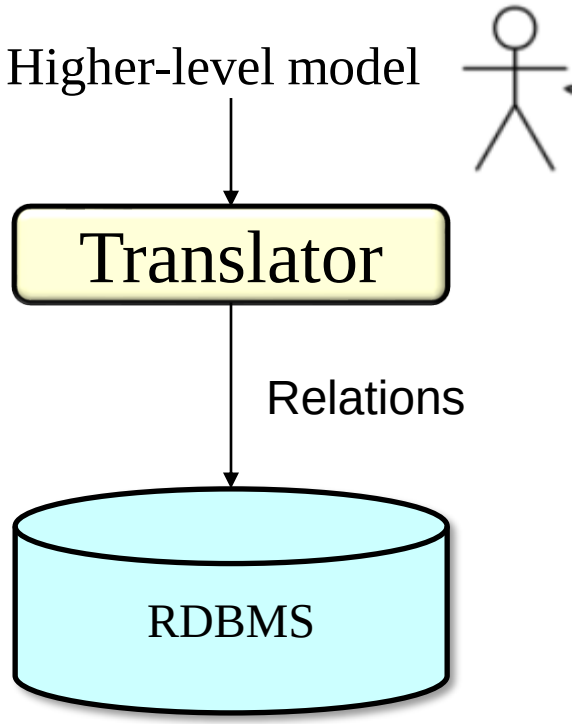
Dr. Shaheen Khatoon



Data Modeling

How to represent data for application

- Relational model – with design principles
- XML
- Database design model
  - Not implemented by system
  - Translated into model of DBMS



Data Modeling

Higher-Level Database Design Models

- *Entity-Relationship Model* (E/R)
- *Unified Modeling Language* (UML)

  Data modeling subset

- Both are graphical
- Both can be translated to relations automatically

  Or semi-automatically

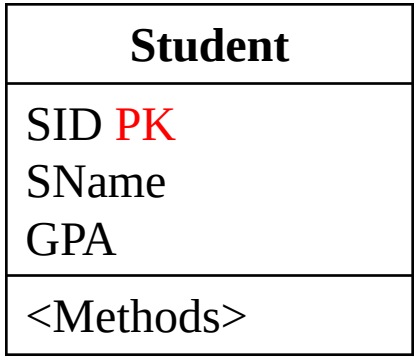
UML Data Modeling: 5 concepts

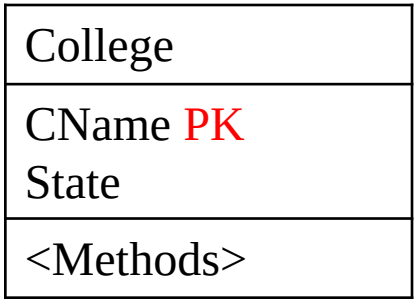
(1) Classes
(2) Associations
(3) Association Classes
(4) Subclasses
(5) Composition & Aggregation

UML Data Modeling: Classes

Name, attributes, methods

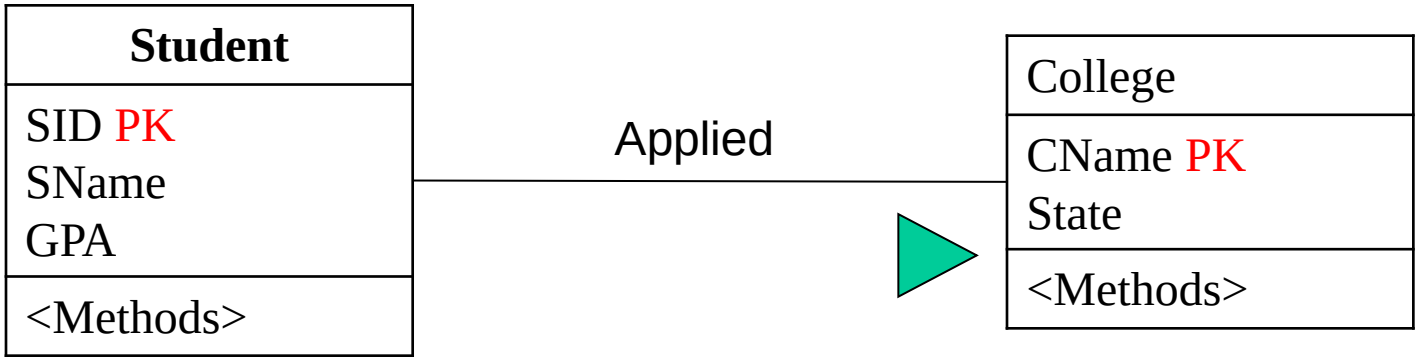
For data modeling: add "pk", drop methods





UML Data Modeling: Associations

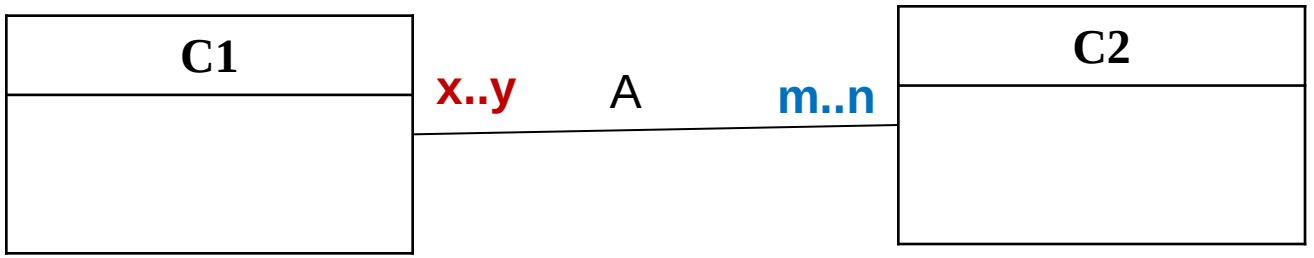
Relationships between objects of two classes

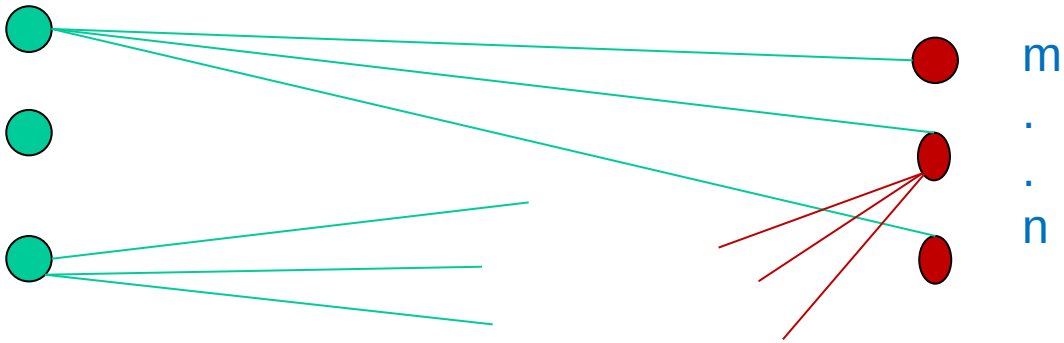


Multiplicity of Associations

Relationships between objects of two classes

Each object of class C_1 is related to at least m and at most n objects of class C_2





m .. *

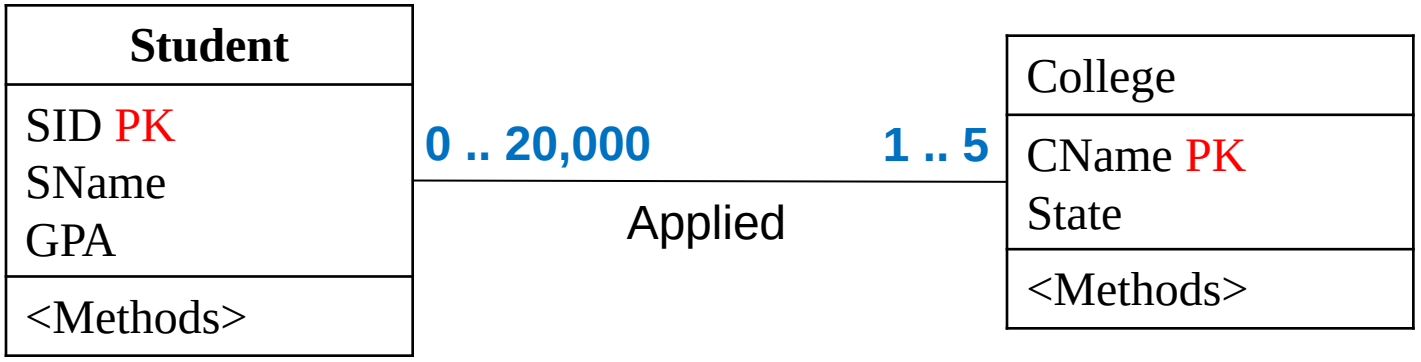
0 .. n

0 .. *

Default
1 .. 1 -> 1

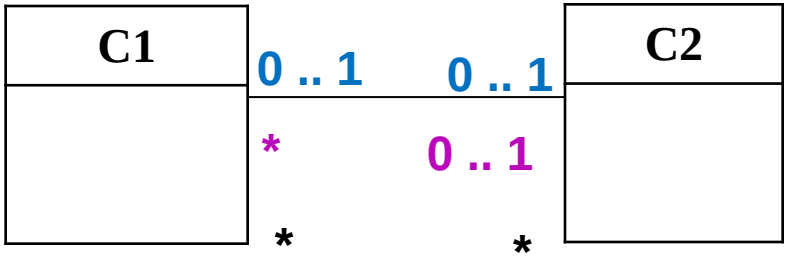
Multiplicity of Associations: Example

Students must apply somewhere and may not apply to more than 5 colleges. No college takes more than 20,000 applications.



Multiplicity of Associations: Types of Relationships

- One-to-One
- Many-to-One
- Many-to-Many
- Complete



Complete One-to-One

1..1 1..1

Complete Many-to-One

1..* 1..1

Complete Many-to-Many

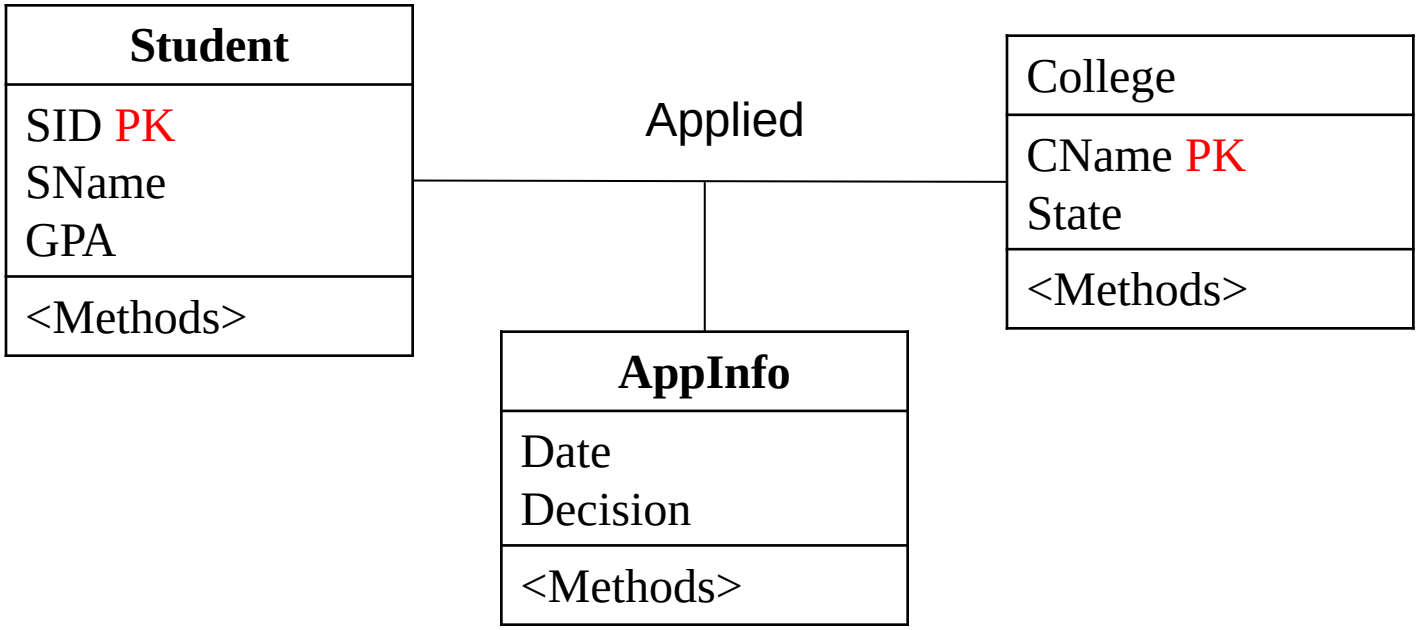
1..* 1..*

UML Data Modeling: 5 concepts

(1) Classes
(2) Associations
(3) Association Classes
(4) Subclasses
(5) Composition & Aggregation

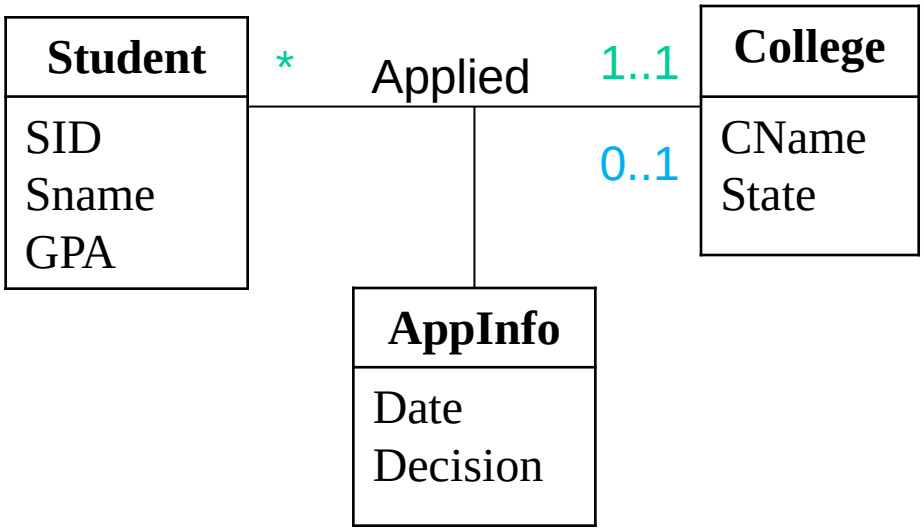
UML Data Modeling: **Association Classes**

Relationships between objects of two classes,
with attributes on relationships



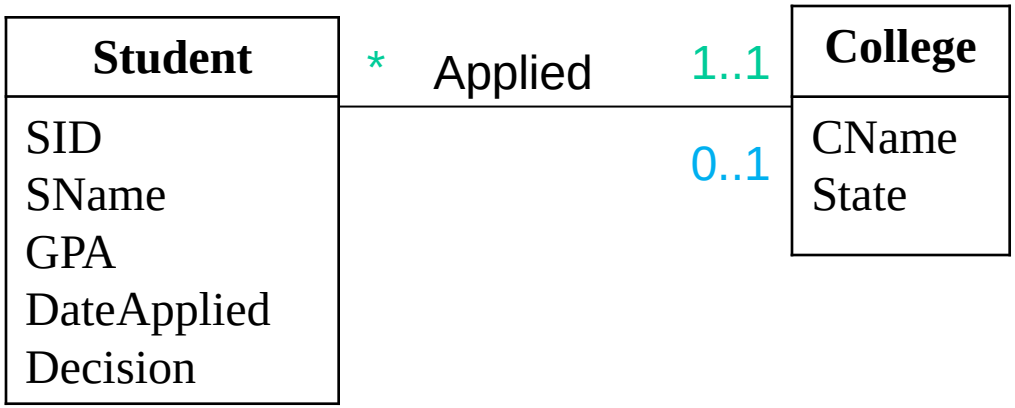
Eliminating Association Classes

Unnecessary if 0..1 or 1..1 multiplicity



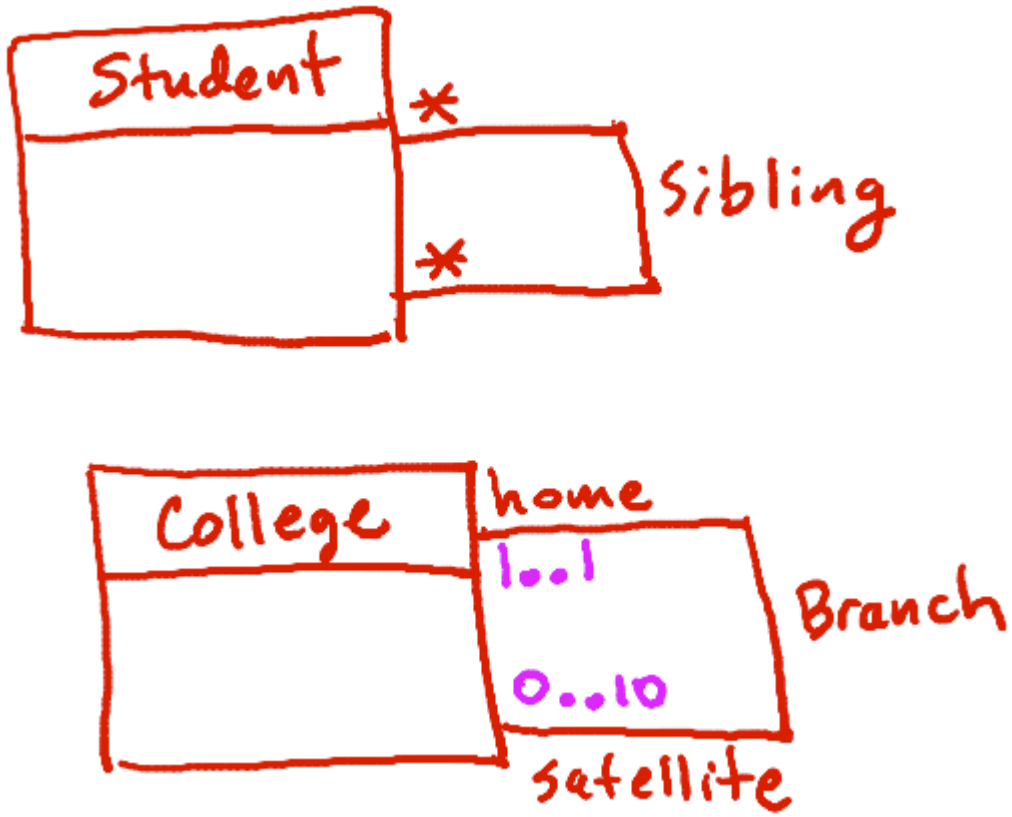
Eliminating Association Classes

Unnecessary if 0..1 or 1..1 multiplicity



Self-Associations

Associations between a class and itself



UML Data Modeling: 5 concepts

(1) Classes

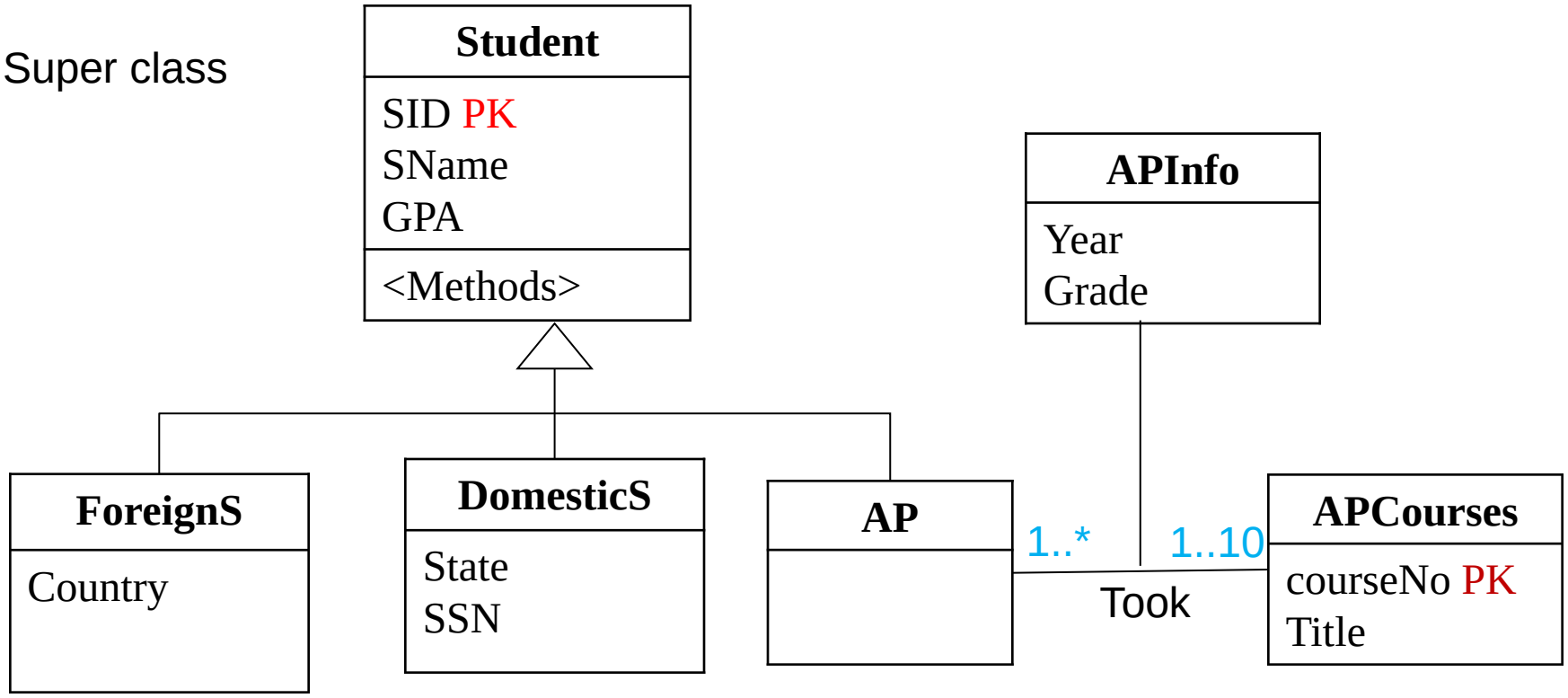
(2) Associations

(3) Association Classes

(4) Subclasses

(5) Composition & Aggregation

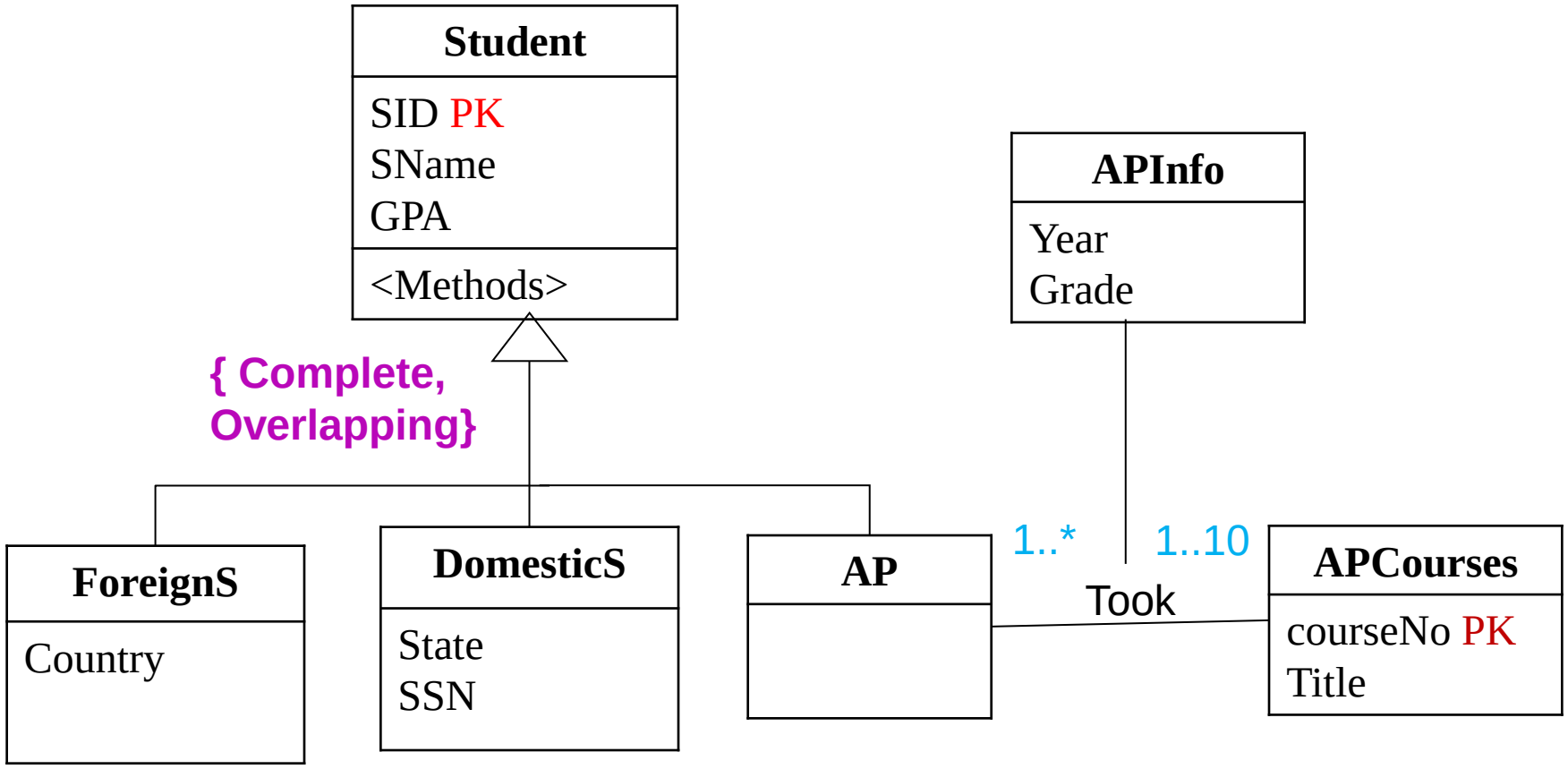
UML Data Modeling: **Subclasses**



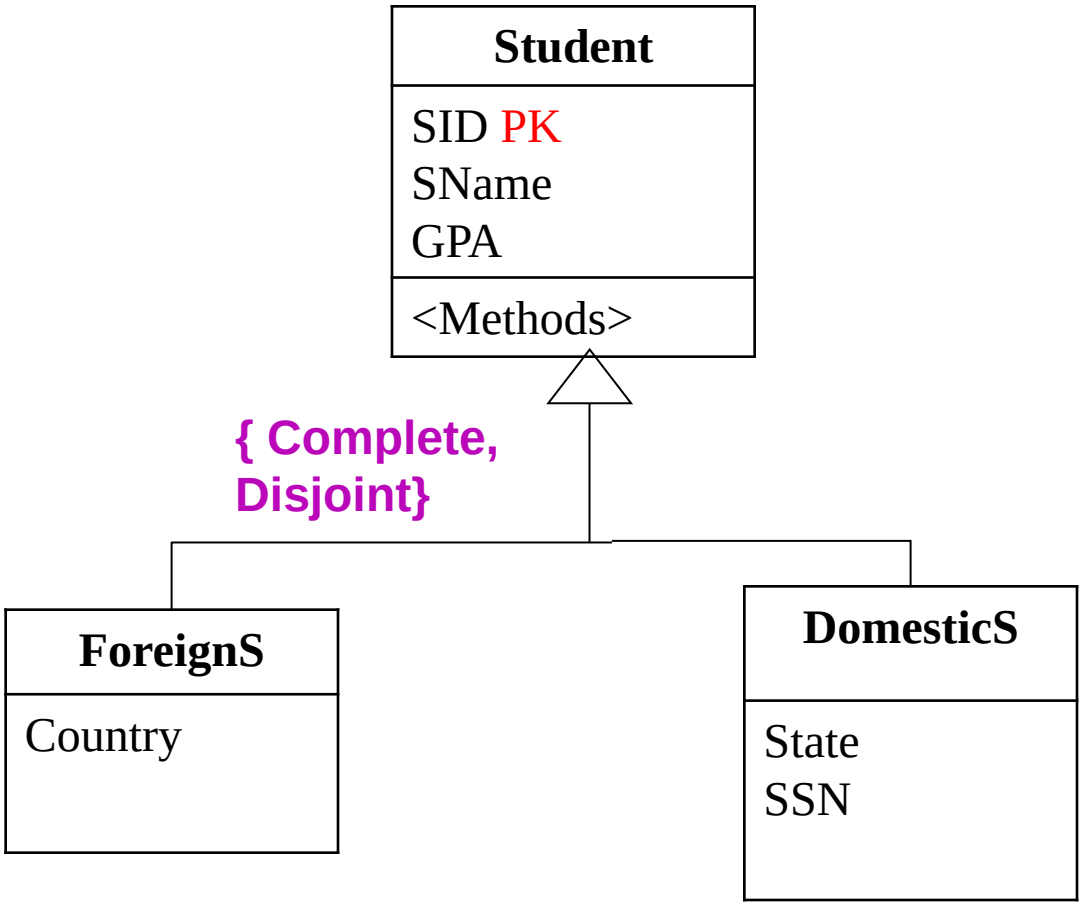
Subclass Terminology & Properties

- *Superclass = Generalization*
- *Subclass = Specialization*
- *Incomplete* (*Partial*) vs. *Complete*
  - *Complete if every object in the superclass is in at least one subclass, otherwise incomplete*
- *Disjoint* (*Exclusive*) vs. *Overlapping*
  - *disjoint if every object is in at most one subclass (we don't have any objects that are in more than one subclass, and that's sometimes called exclusive)*

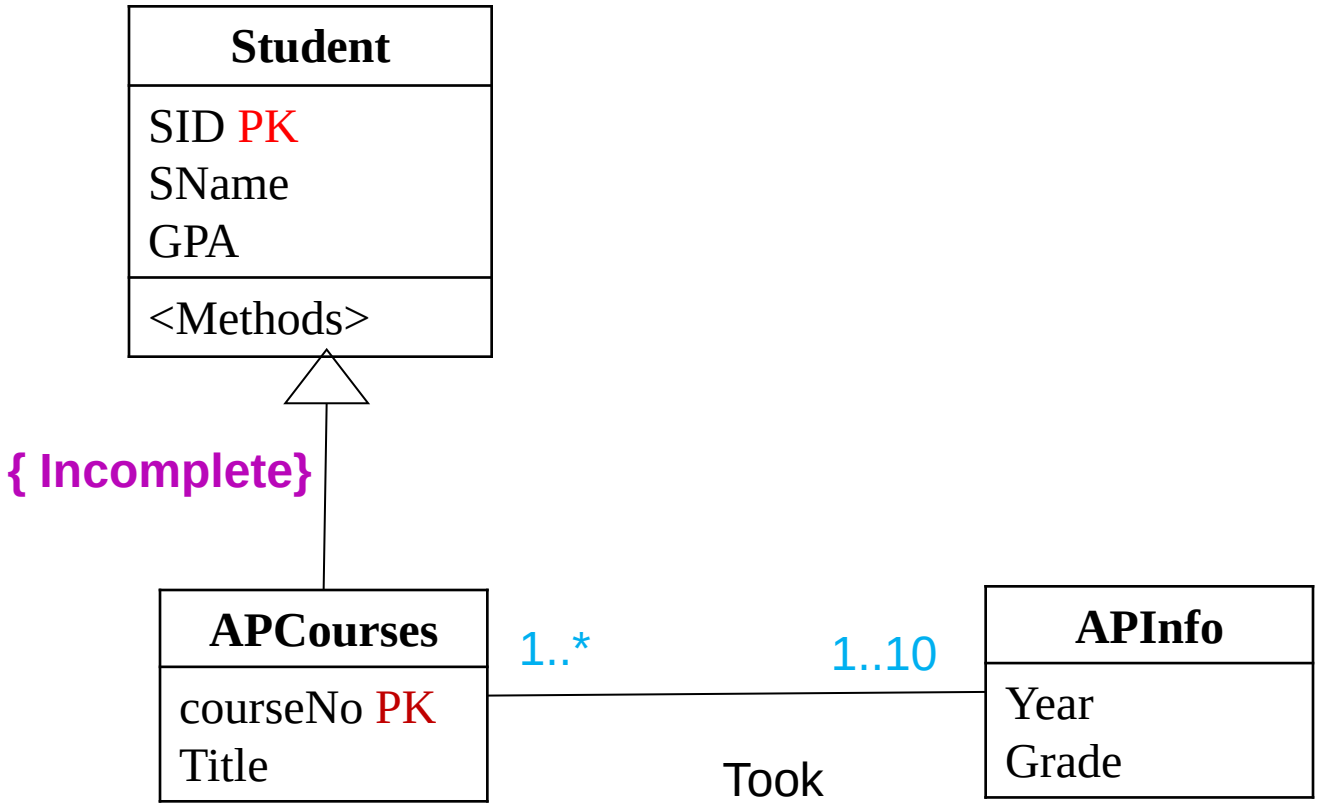
UML Data Modeling: **Subclasses**



UML Data Modeling: **Subclasses**



UML Data Modeling: **Subclasses**



UML Data Modeling: 5 concepts

(1) Classes

(2) Associations

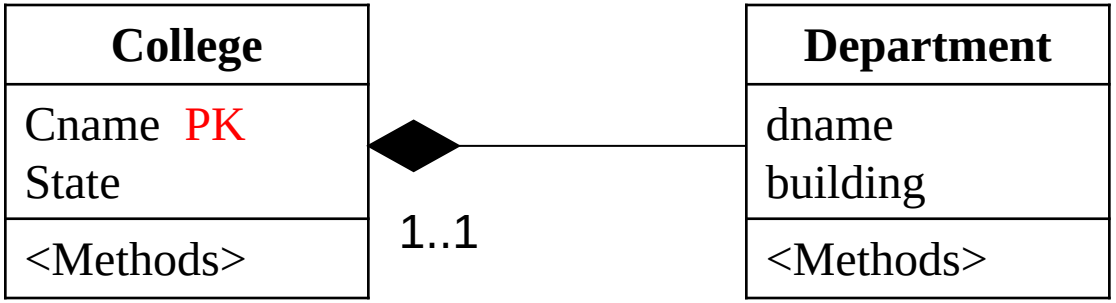
(3) Association Classes

(4) Subclasses

(5) Composition & Aggregation

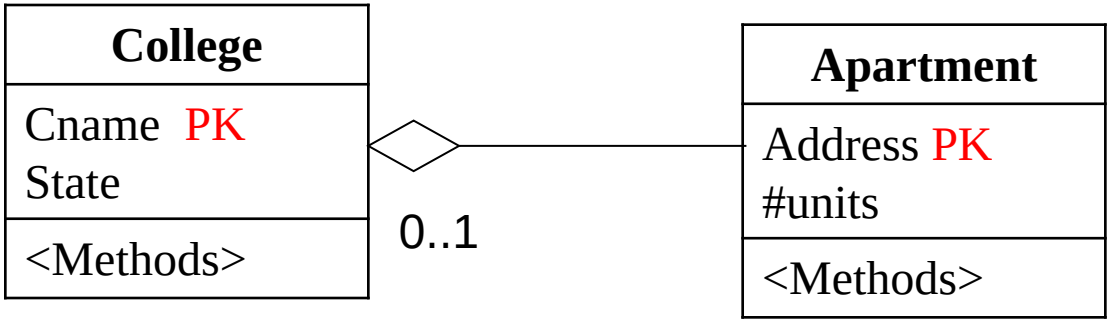
UML Data Modeling:
Composition & Aggregation

Objects of one class belong to objects of another class

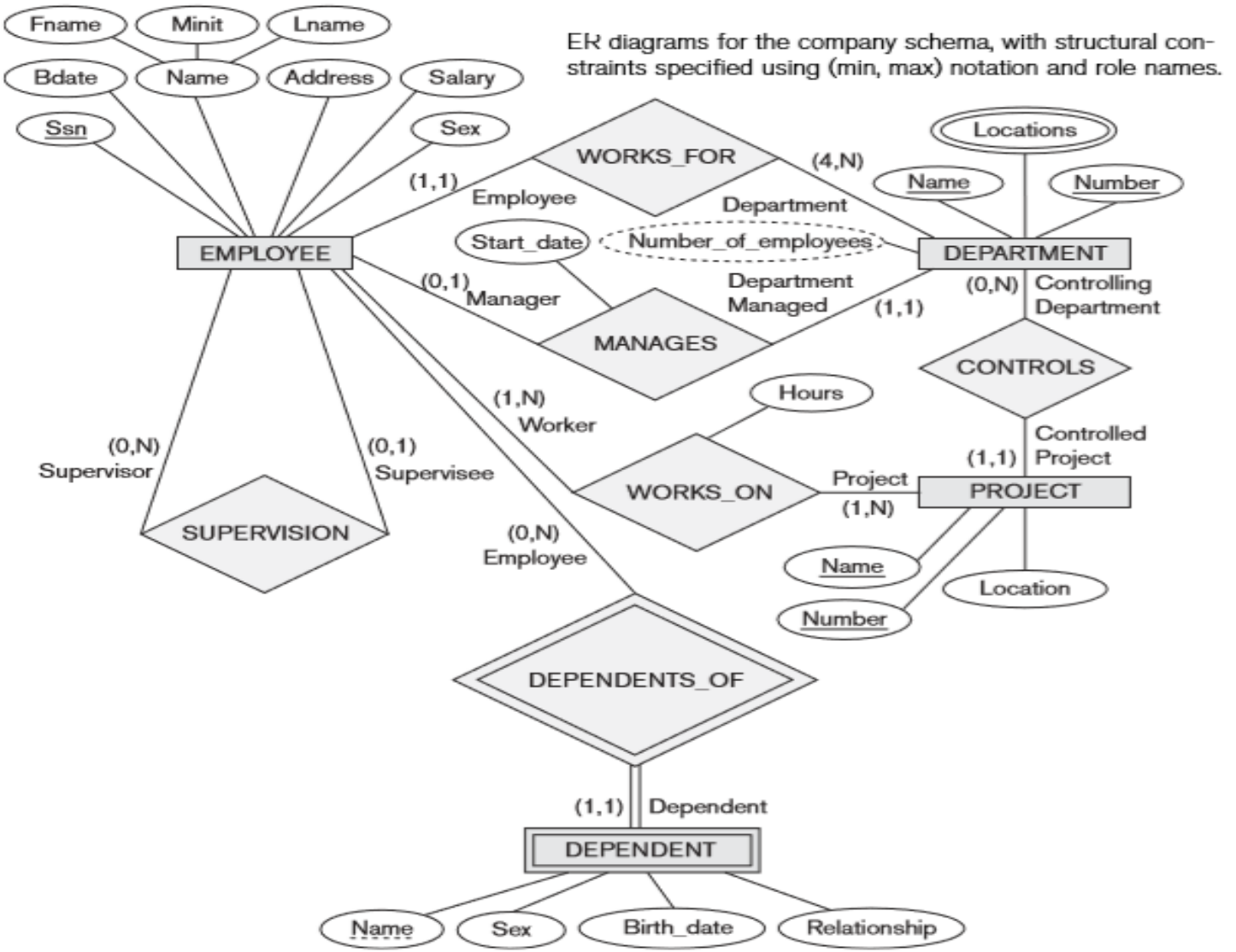


UML Data Modeling:
Composition & Aggregation

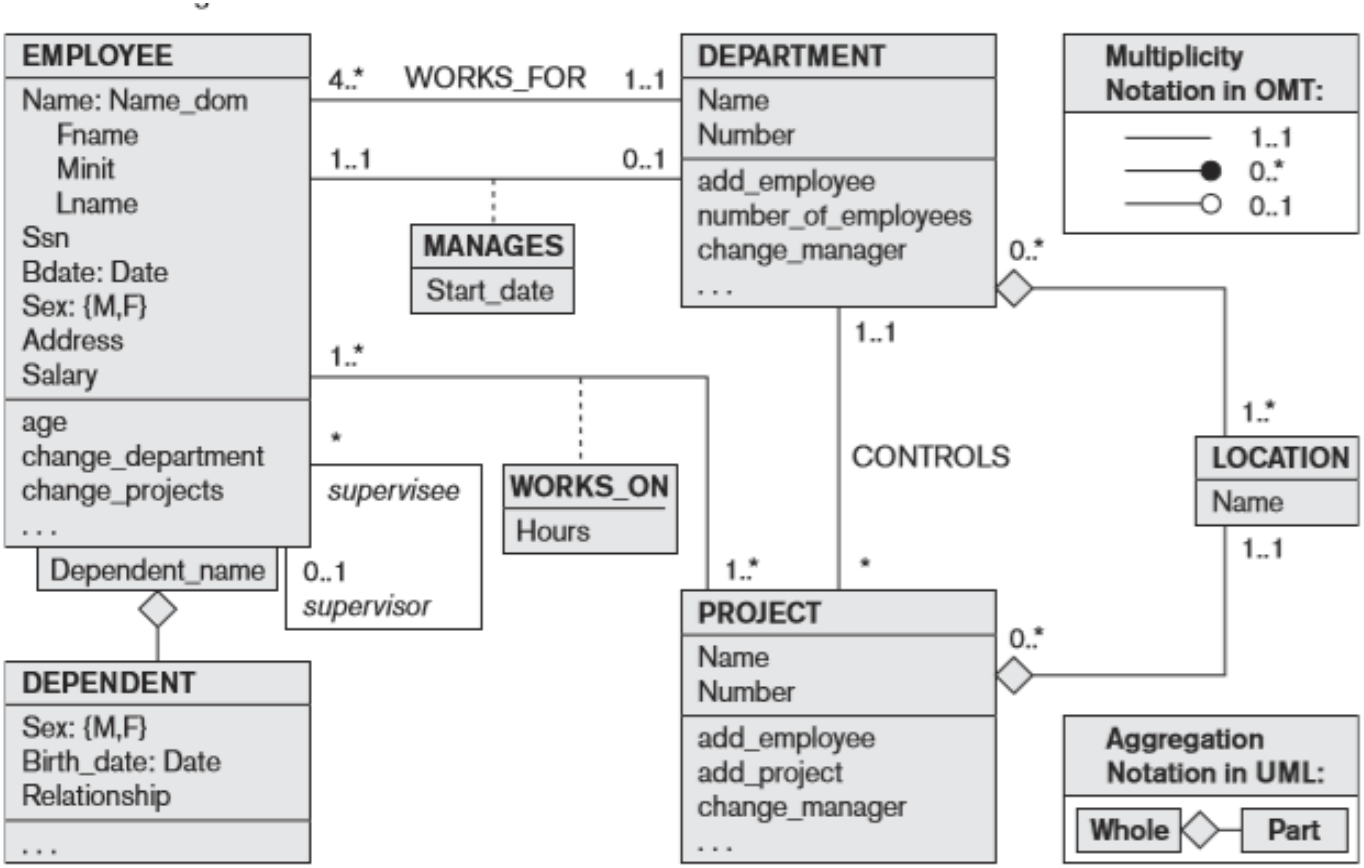
Objects of one class belong to objects of another class



Practice: Map ER to Object Model



The COMPANY conceptual schema in UML class diagram notation

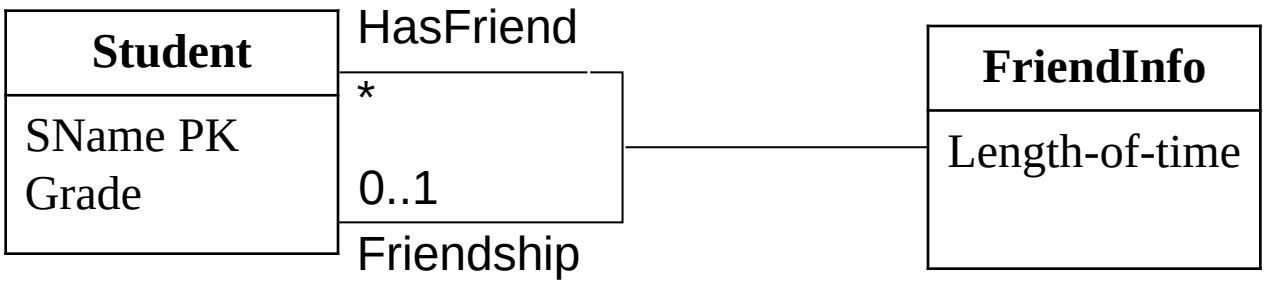


Homework

- Draw a UML diagram for the following scenario:

Consider a tiny social network containing high school students and their “friendship“. Each student may have a friend on at most one other student, and associated with each friendship is the length of time the friendship has been going on. Students have a name and a grade, and names are unique. Draw a UML diagram that models this information. Make sure to capture the asymmetry and multiplicity of the friendship relationship.

Homework: Solution



Summary

Higher-Level Database Design

- *Unified Modeling Language* (UML)
  Data modeling subset
- Graphical
- 5 concepts
  (1) Classes
  (2)Associations
  (3)Association Classes
  (4)Subclasses
  (5)Composition & Aggregation

❖ Can be translated to relations automatically