

UML Data Modelling

Carla Teixeira Lopes

Bases de Dados

Mestrado Integrado em Engenharia Informática e Computação, FEUP

Based on Jennifer Widom slides

Agenda

Introduction to Database Design Composition & Aggregation

Classes

Constraints

Associations

Derived Elements

Association Classes

Generalizations

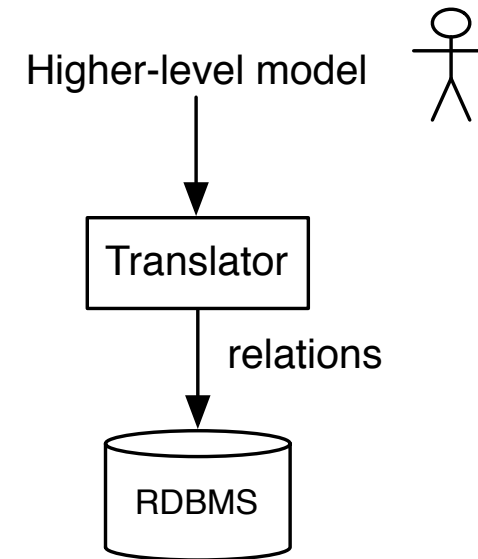
Data modelling

How to represent data for application

Database design model

Not implemented by system

Translated into model of DBMS



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

Key concepts

Classes

Constraints

Associations

Derived Elements

Association Classes

Generalizations

Composition & Aggregation

Classes

Descriptor of a set of objects that share the same properties (semantics, attributes, and relationships)

Concrete things

person, book, car, ...

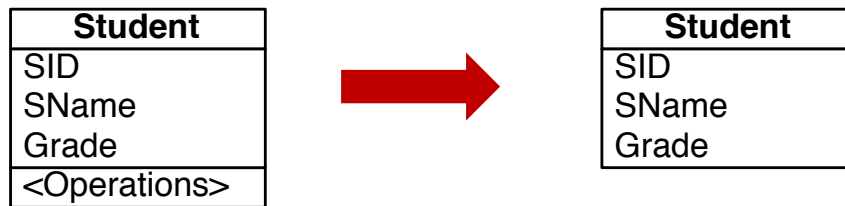
Conceptual things

class, course, profession, ...

Classes

Characterized by name, attributes and operations

For data modeling: drop operations



The class name is usually written in the singular, with the first letter in uppercase

Attributes

Attributes are defined in terms of class, while values of the attributes are defined at the instance level

A student has the attributes: identifier, name and admission grade

John is a student with the id *123*, name *John Smith* and an admission grade of *180*

A class should not have two attributes with the same name

Attributes can be associated with types

Not predefined in UML

Use the ones of the DBMS

Student
SID: integer
SName: string
Grade: integer

In data modelling, we can also specify a primary key

Student
SID <u>pk</u>
SName
Grade

Exercise

Imagine the SIGARRA database.

List 3 classes with attributes that might be in this database.

Key concepts

Classes

Constraints

Associations

Derived Elements

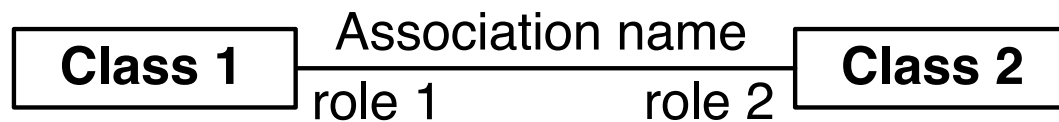
Association Classes

Generalizations

Composition & Aggregation

Associations

Relationships between objects of two classes



As an object is an instance of a class, a **link** is an instance of an association

The name is optional

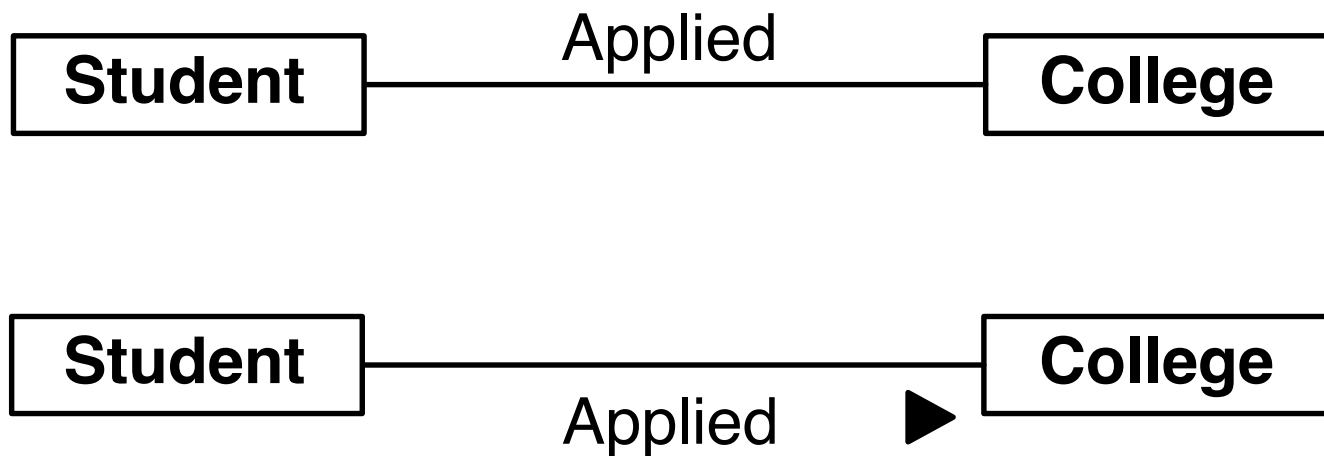
There may be more than one association between the same pair of classes

Having different names

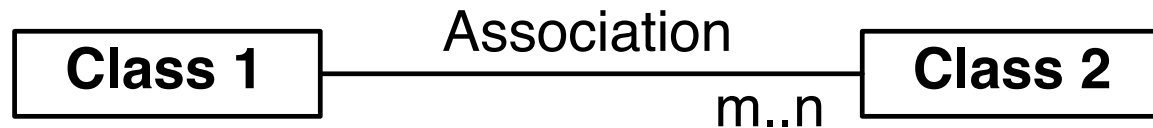
Associations

Imagine a scenario where students apply to colleges.

What classes would we create? And associations?



Multiplicity of Associations



Each object of Class 1 is related to at least *m* and at most *n* objects of Class 2

A * in place of *n* stands for *no upper limit*

Abbreviations

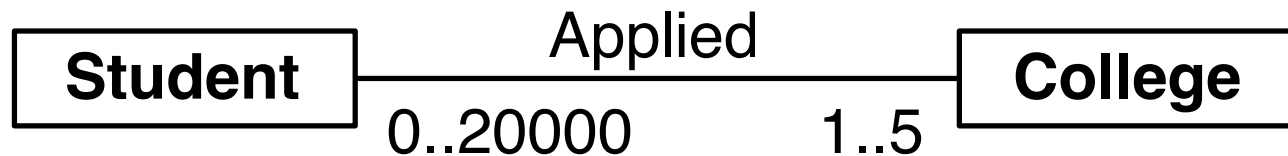
* stands for 0..*, that is, no restrictions

1 stands for 1..1

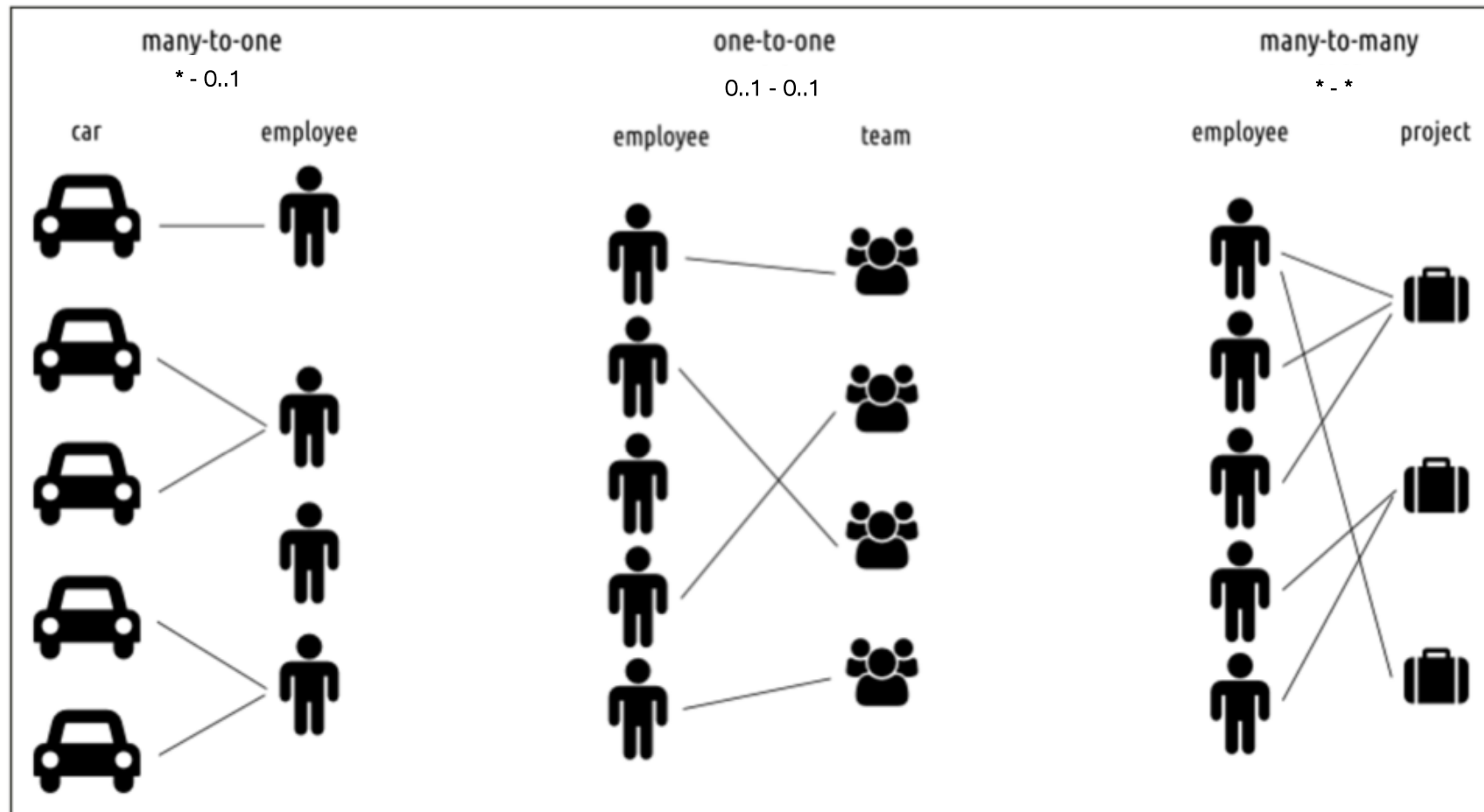
Default: 1..1

Multiplicity of Associations

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



Multiplicity of Associations



Credits: André Restivo

Complete Associations

Every object must participate in the association

Complete many-to-one

1..* - 1..1

Complete one-to-one

1..1 - 1..1

Default association

Complete many-to-many

1..* - 1..*

Kahoot time

Any doubts?

Readings

Jeffrey Ullman, Jennifer Widom, A first course in
Database Systems 3rd Edition

Section 4.7 - Unified Modeling Language

UML Data Modelling

Carla Teixeira Lopes

Bases de Dados

Mestrado Integrado em Engenharia Informática e Computação, FEUP

Based on Jennifer Widom slides

Key concepts

Classes

Constraints

Associations

Derived Elements

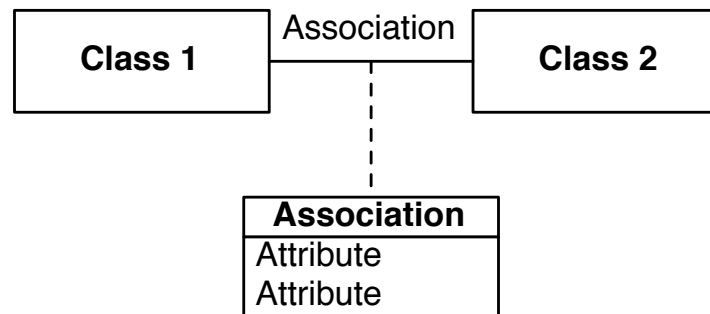
Association Classes

Generalizations

Composition & Aggregation

Association Classes

Relationships between objects of two classes, *with attributes on relationships*

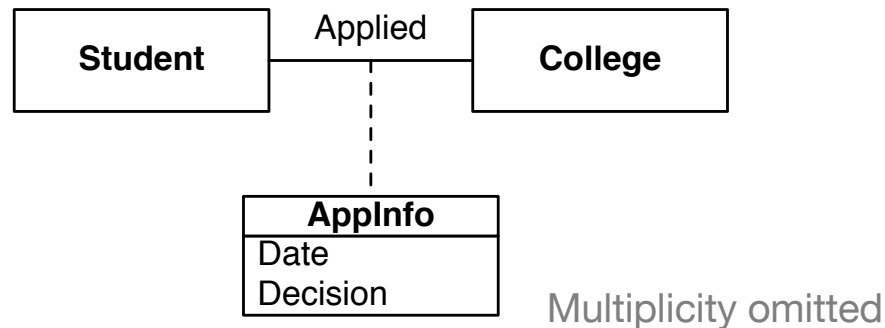


The name may be placed in the association, in the class or in both

It only captures **one** relationship between the two specific objects across the two classes

Association Classes

Suppose that, in the previous example, we also want to have the date the students applied to college and the decision.

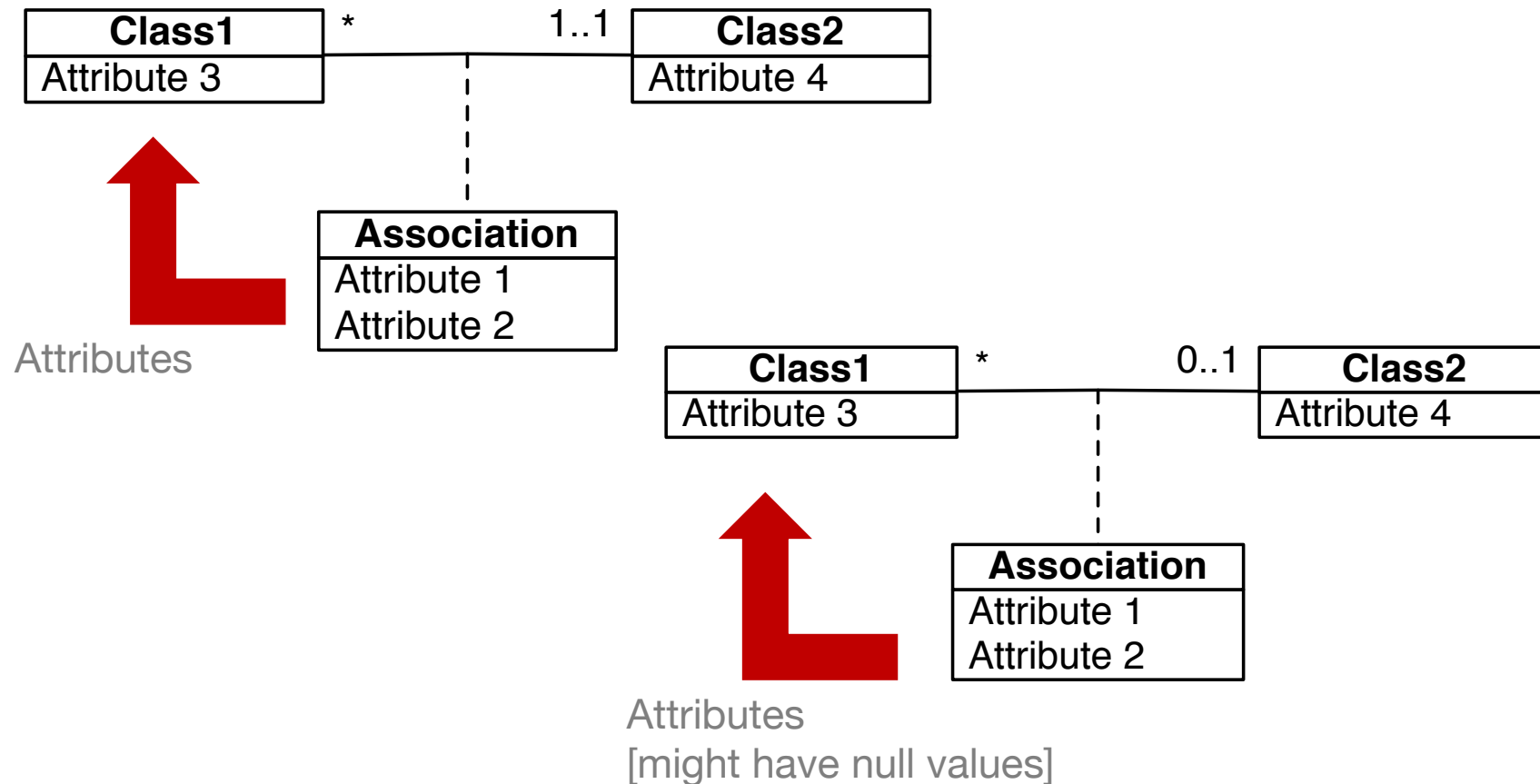


Doesn't allow students to apply multiple times to the same college.

How could we model this situation?

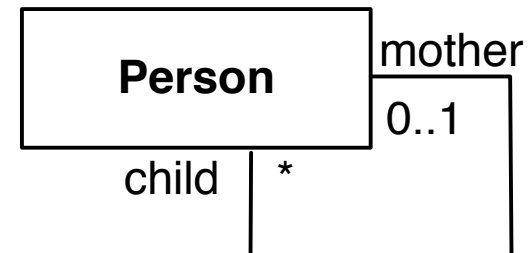
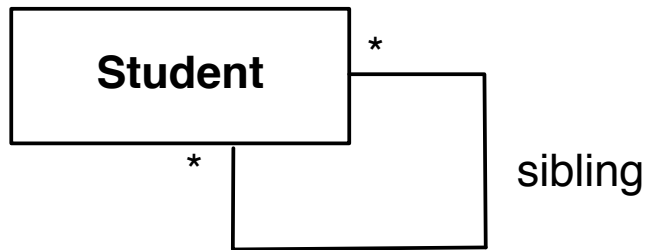
Eliminating Association Classes

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

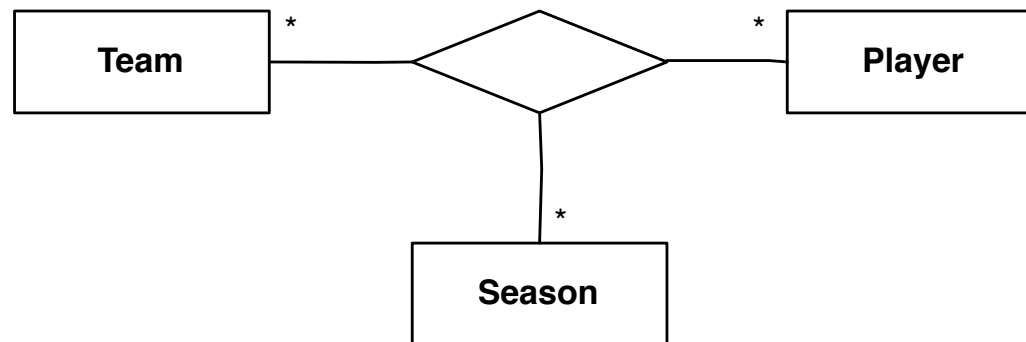
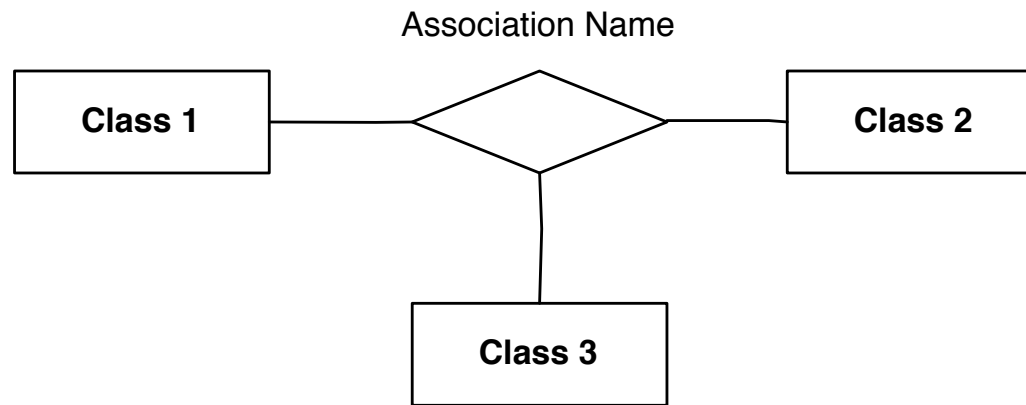


Self-Associations

Associations between a class and itself

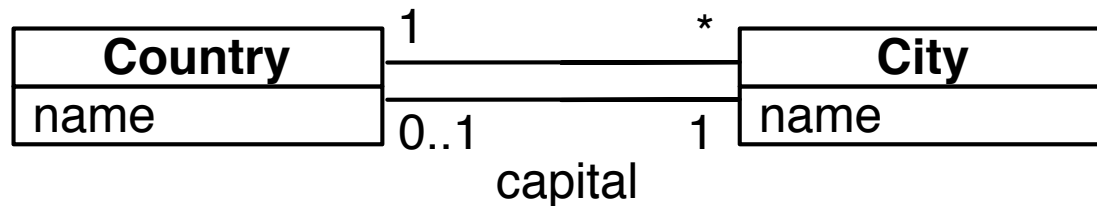
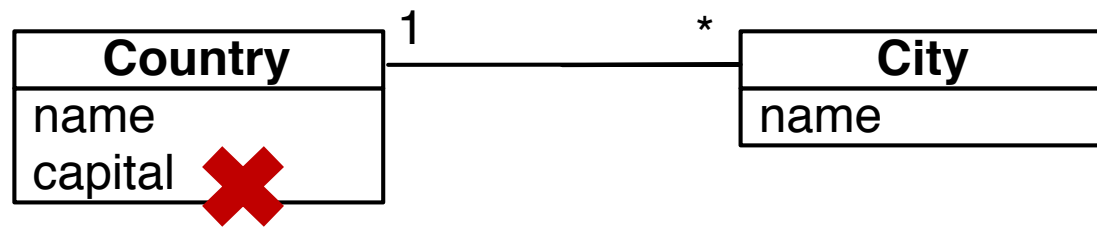


Associations n-arys

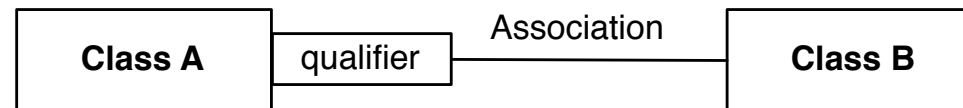


Association versus attribute

An attribute should never be a reference to a class



Qualified associations



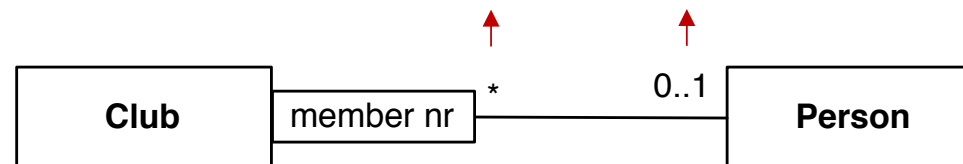
Qualifier

One or more attributes of an association used to navigate from A to B

“Access key” to B from an object of A

A person can be a member of several clubs

For each pair: Club + member nr



Exercise

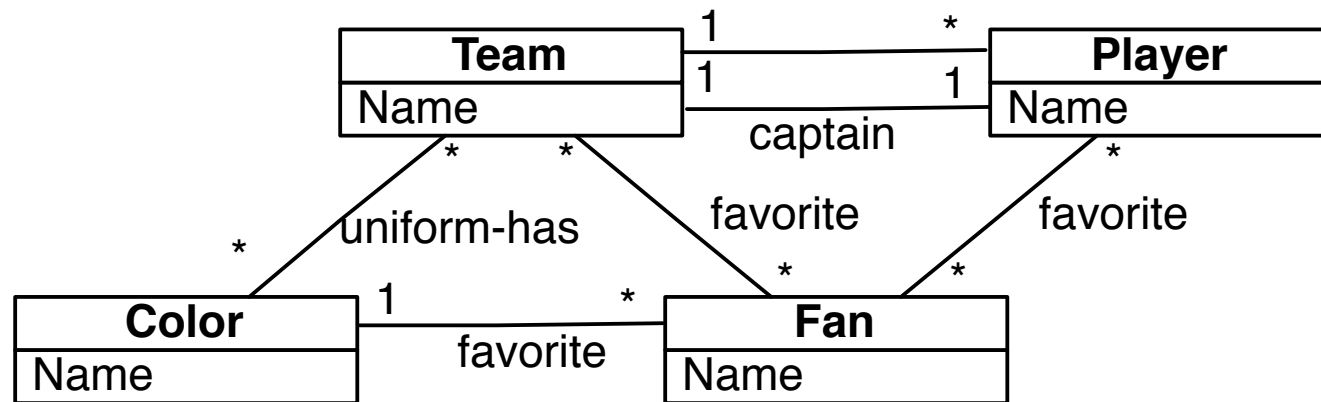
Draw a UML diagram for a database recording information about teams, players, and their fans, including:

For each team, its name, its players, its team captain (one of its players), and the colors of its uniform

For each player, his/her name

For each fan, his/her name, favorite teams, favorite players, and favorite color.

Exercise



Key concepts

Classes

Constraints

Associations

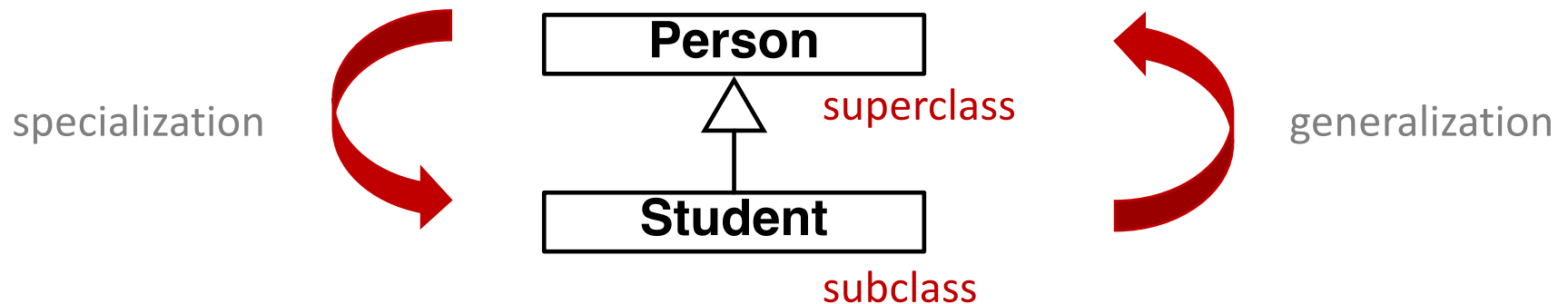
Derived Elements

Association Classes

Generalizations

Composition & Aggregation

Generalizations

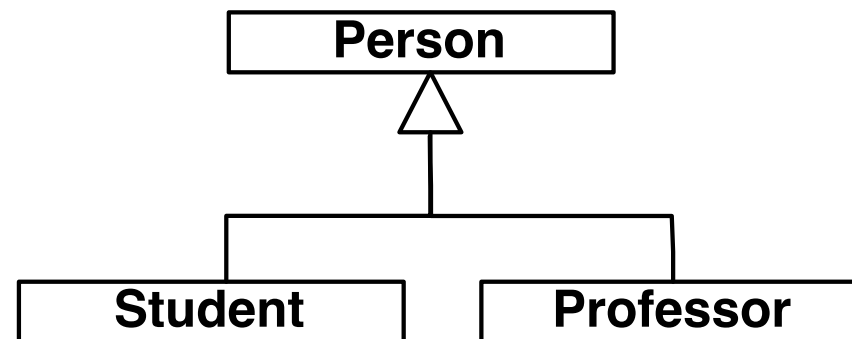
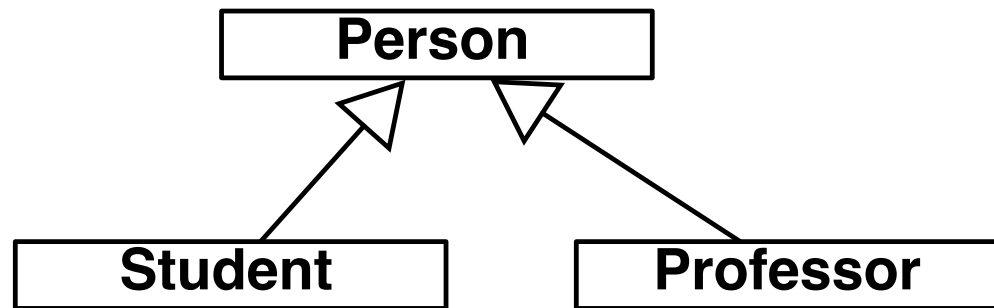


“is a” semantic relationship

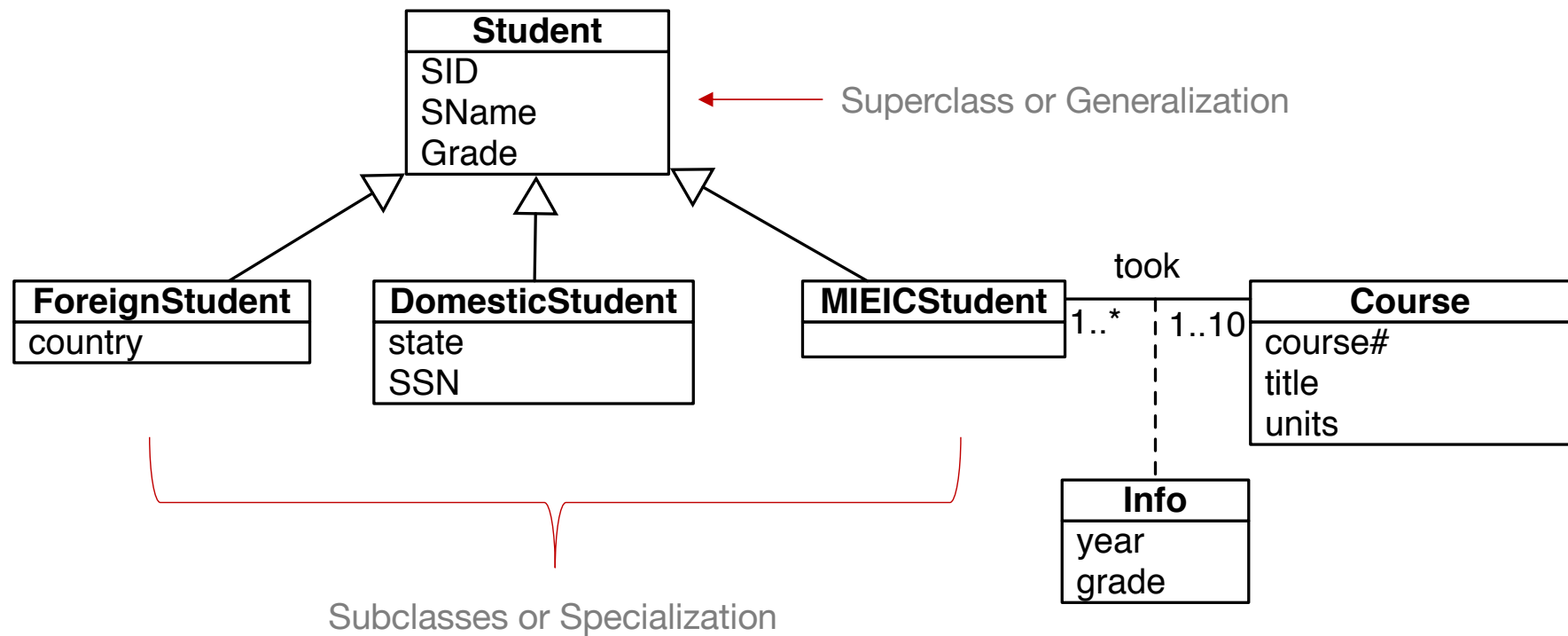
A student “is a” person

The subclass inherits the properties (attributes and relationships) of the superclass and may add other

Generalizations – Alternative Notations



Generalizations Example



Generalizations Properties

Complete

If every object in the super class is in at least one subclass

Incomplete or partial

If it's not complete

Disjoint or Exclusive

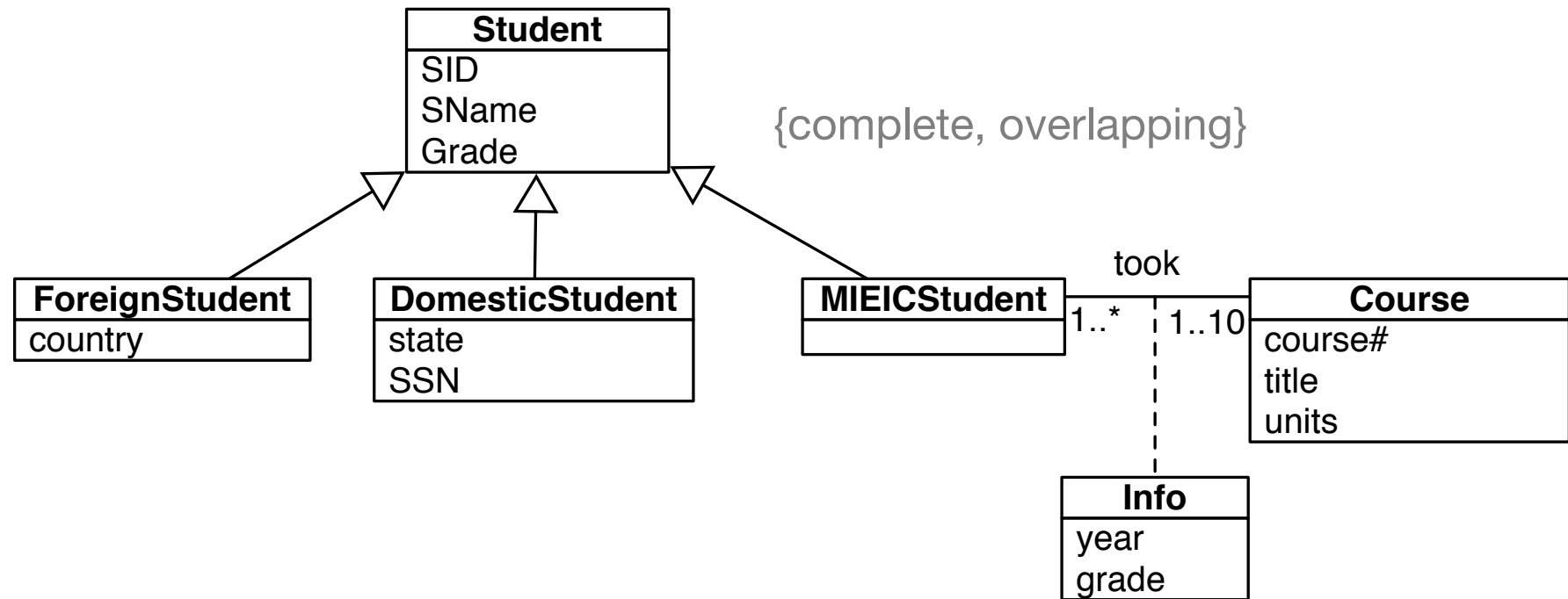
If every object is on at most one subclass

Overlapping

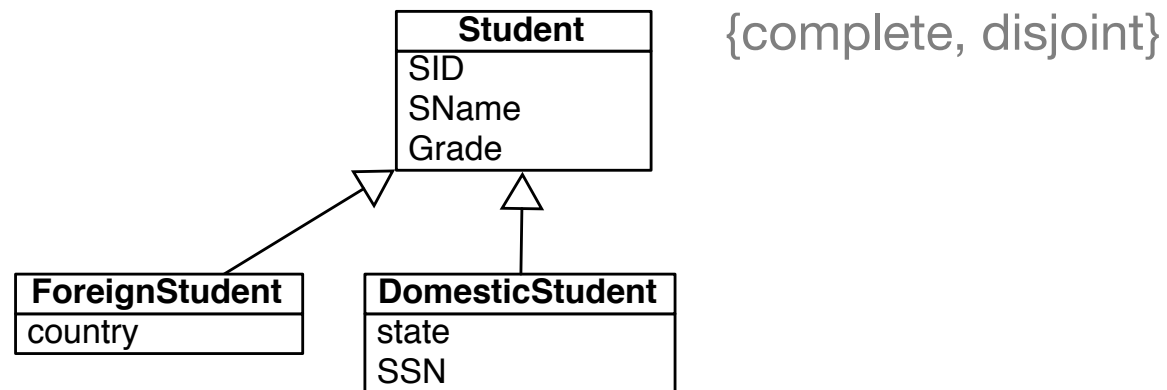
If it's not disjoint

We can have any combination of the first two with the second two.

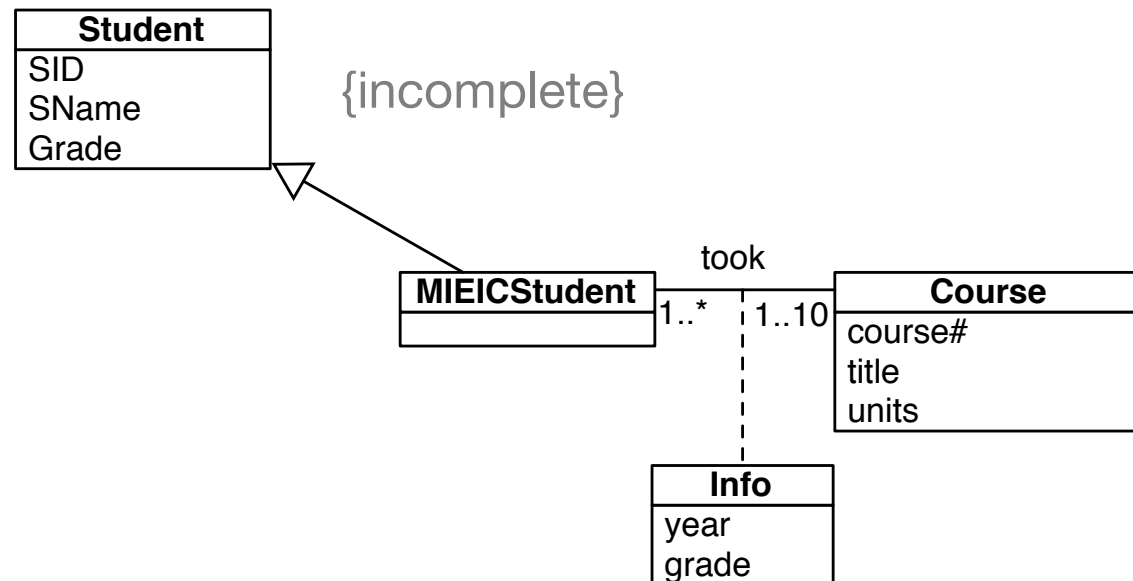
Generalizations Properties



Generalizations Properties



Generalizations Properties



Key concepts

Classes

Constraints

Associations

Derived Elements

Association Classes

Generalizations

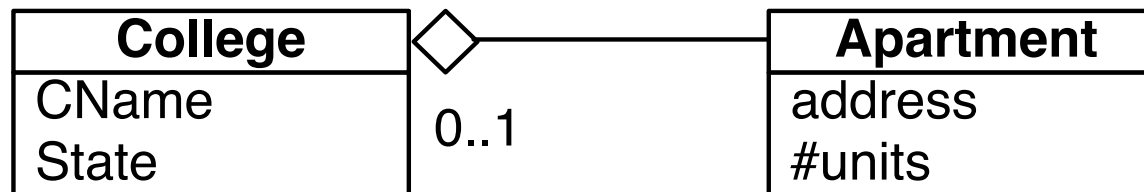
Composition & Aggregation

Aggregation

Special type of an association

Objects of one class belong to objects of another class

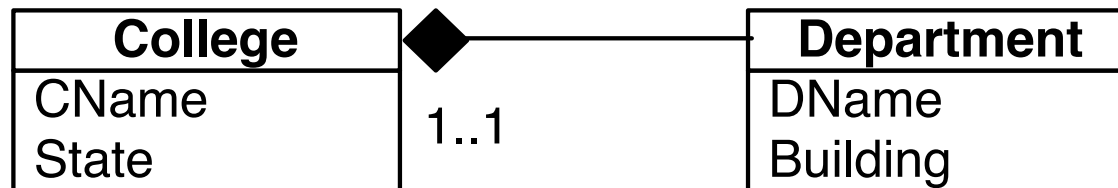
0..1 is implicit



Composition

Strongest form of aggregation

1..1 is implicit



Kahoot time!

Any doubts?

Key concepts

Classes

Constraints

Associations

Derived Elements

Association Classes

Generalizations

Composition & Aggregation

Constraints

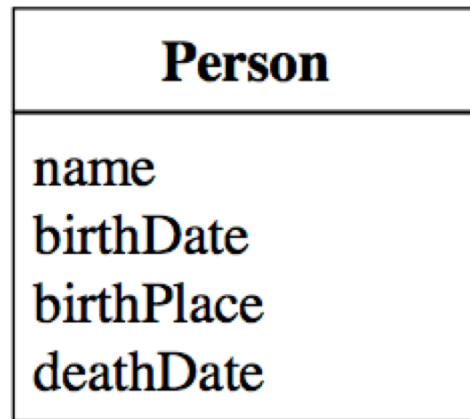
Specifies a condition that has to be present in the system

It is indicated by

- an expression or text between brackets

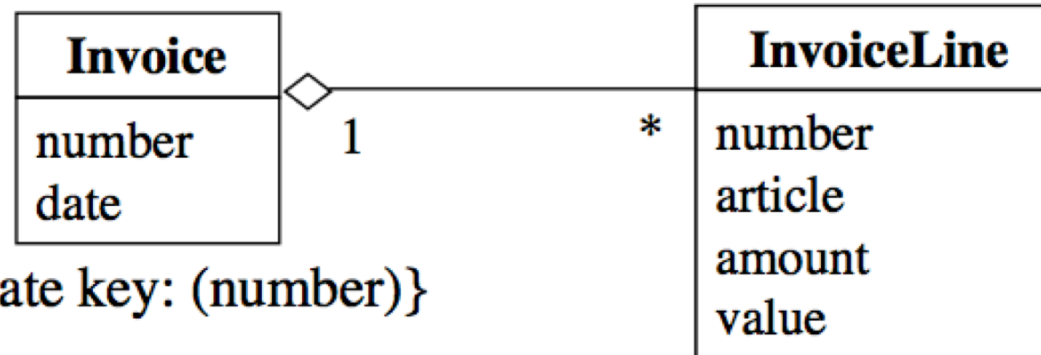
- note placed near (or connected by dotted lines to) the elements to which it relates

Constraints in classes



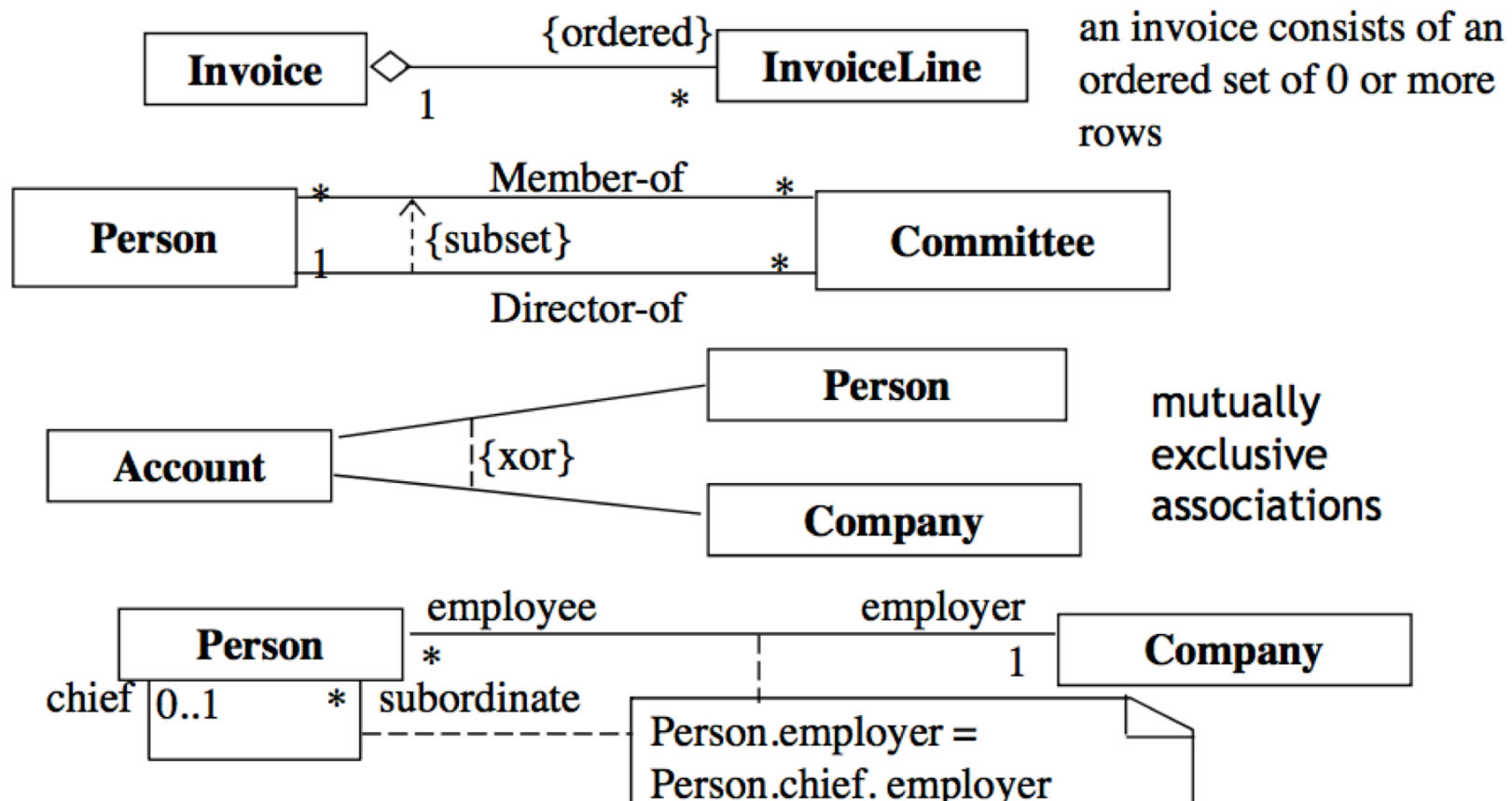
{candidate key: (name, birthDate, birthPlace)}

{deathDate > birthDate}



{candidate key: (number)}

Constraints in associations



Key concepts

Classes

Constraints

Associations

Derived Elements

Association Classes

Generalizations

Composition & Aggregation

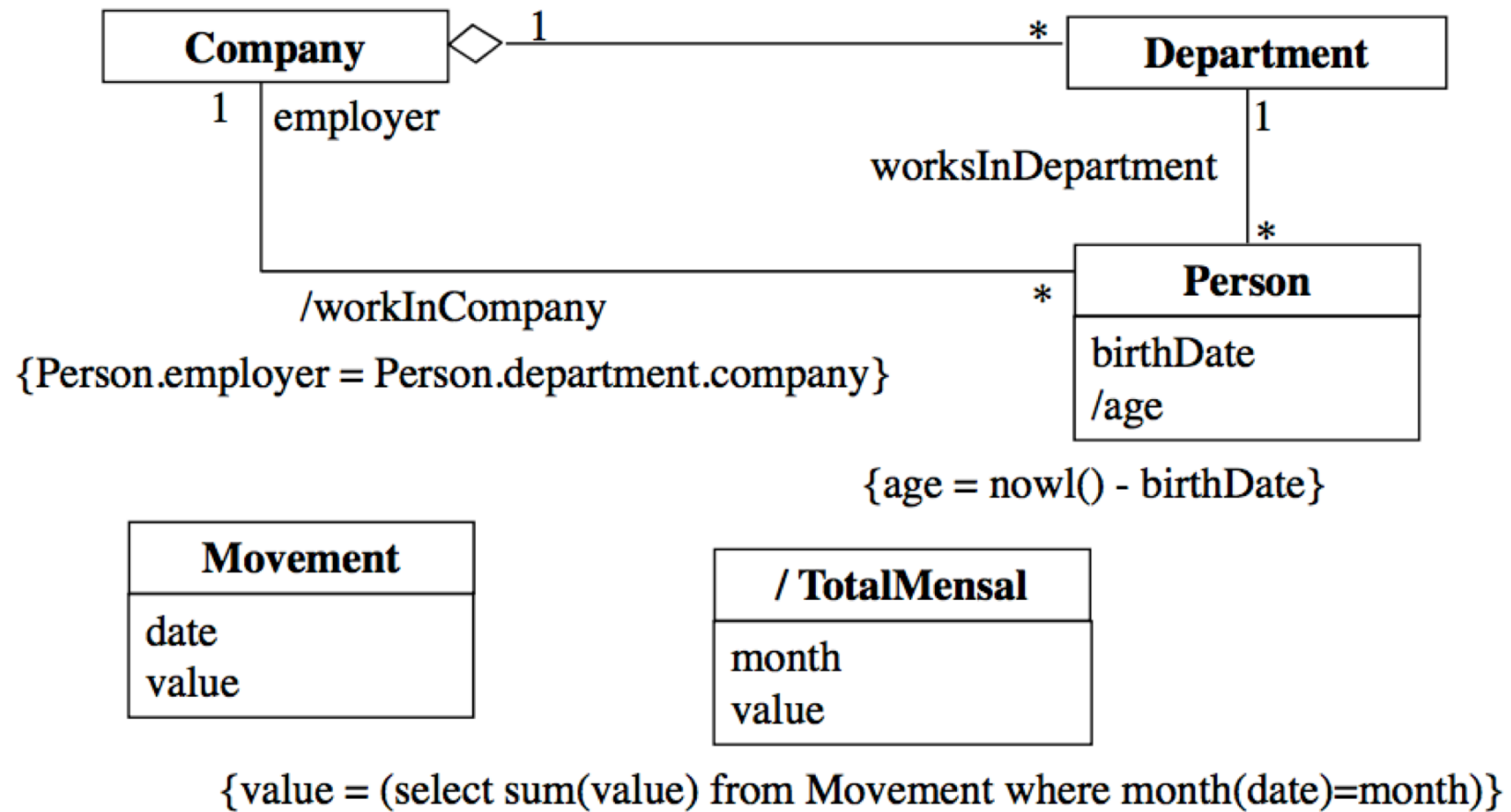
Derived Elements

Element (class, attribute or association) computed using other elements in the model

Notation: ‘/’ before the name of the derived element

Usually have an associated constraint that relates them with other elements

Derived Elements



Higher-Level Database Design

Unified Modeling Language (UML)

Data modeling subset

Graphical

Key concepts

Classes

Associations

Association Classes

Generalizations

Composition & Aggregation

Constraints

Derived Elements

Can be translated to relations automatically

Readings

Jeffrey Ullman, Jennifer Widom, A first course in
Database Systems 3rd Edition

Section 4.7 - Unified Modeling Language