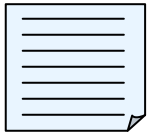
2 Conceptual Database Design

References: Kemper / Eickler chap 2, Elmasri / Navathe chap. 3
Garcia-Molina / Ullmann / Widom: chap. 2

2.1.1. Overview

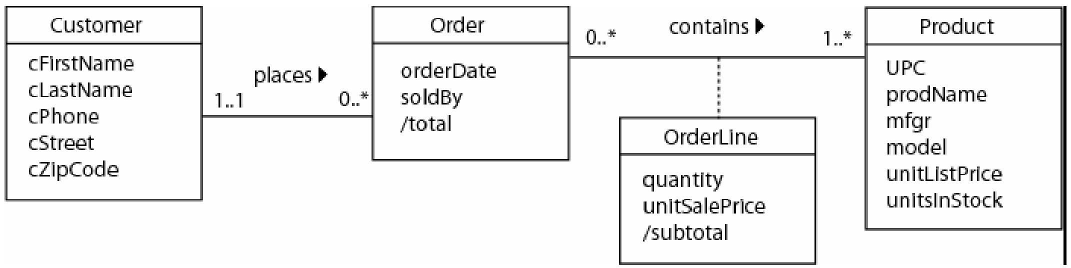
Freie Universität Berlin

Requirement analysis

-> Text 

Conceptual Design

-> Conceptual Model



Logical Schema Design

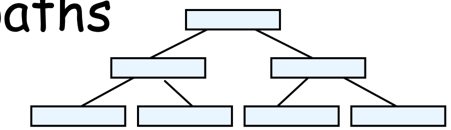
-> Database schema

```
CREATE TABLE Kurs
(KID CHAR(10) PRIMARY KEY,
 Name CHAR(40) NOT NULL,
 Dauer INTEGER);
```

```
CREATE TABLE Order
 ODate  DATE
soldBy INTEGER FOREIGN KEY
       REFEREBCES Peronal(PID)
 CID INTEGER   FOREIGN
 KEY REFERENCES Customer (CID));
```

Physical Schema Design

-> Access paths



Administration

→ Redesign

Database Design:Terminology

Freie Universität Berlin

Def.: Database Design (Modelling)

The process of defining the overall structure of a data-base, i.e. the schema, on **different layers of abstraction.**

Design levels: Conceptual, logical, physical

Includes "Analysis" and "Design" from Software Engineering (SE)

DB Design: defining the "static model" using formal or visual languages

DB design	SE
Requirements	Requirements
Conceptual modeling	Analysis
Logical modeling	Design
Physical modeling	Implementation

2.1.2 Requirement Analysis

Most important: talk with your customers!

Tasks during RA:

- **Identify** essential "real world" information (e.g. interviews)
- **Remove** redundant, **unimportant details**
- **Clarify** unclear natural language statements
- **Fill** remaining **gaps** in discussions
- Distinguish **data** and **operations**

Requirement analysis & Conceptual Design aims at focusing thoughts and discussions !

Example: Geo-DB ("Mondial")

The database we develop will contain data about countries, cities, organizations and geographical facts. In the first step, countries, cities, regions (like "Bundesländer" or geographical regions), and continents are to be represented in the DB.

In the requirements analysis it has to be clarified, **what kind of information** is supposed to be represented, **not how** it should be represented!

First step:	**filter essential information , ignore unimportant details**
Note:	importance of a piece of information depends on the application scenario

Requirement Analysis

- **Clarify unclear statements**
  - what is a country?
    Political unit: compare Korea vs South /North Korea
- **Fill gap**
  - Cities are located in regions. What if
    a country does not have regions?
    $\rightarrow$ region is country itself
  - Can a region belong to different countries? No, but
    there may be regions with the same name in different countries
  - Can a country belong to different continents? Yes.
- **Distinguish data from operations**
  - Gross National Product per inhabitant: calculate
  - "It happens that countries are united" ....

2.2.1 Basic modeling primitives

Freie Universität Berlin

Conceptual modeling

- Distinguish between **types** (classes) and **individual facts** (metadata vs data)
- The name of **this woman** is *Kunz* with first name *Tamara*.
- As opposed to:
- A person is identified by first name, last name and birth date.

- **Describe reality on a type level**

- Use a **graphical language** in order to get an overall impression of the domain modeled.

Modeling language requirements

Freie Universität Berlin

- What is the **right language** for "modeling reality"?
- Which **language primitives** ?

An old problem of **philosophy**: **how to describe the world** in an appropriate, comprehensible way?

One of the answers were **logic** languages.
They allow to express more than we (currently) want to:
facts <u>and</u> rules.

e.g.: human(Plato) , $\forall x\ (\mathrm{human}(x) \Rightarrow \mathrm{mortal}(x))$

Ockham's Razor

Ockham chooses a razor

"Non sunt multiplicanda entia praeter necessitatem"

William van Ockham, English philosopher, 13th century

(Principle of Economy, Law of Parsimony)

Basic modeling primitives

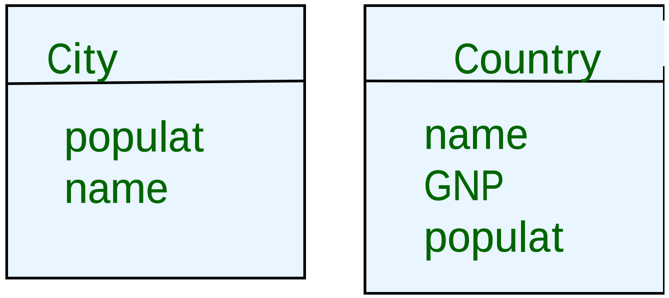
Modeling the "Real World"

Entity (type)

something which exists, has a name

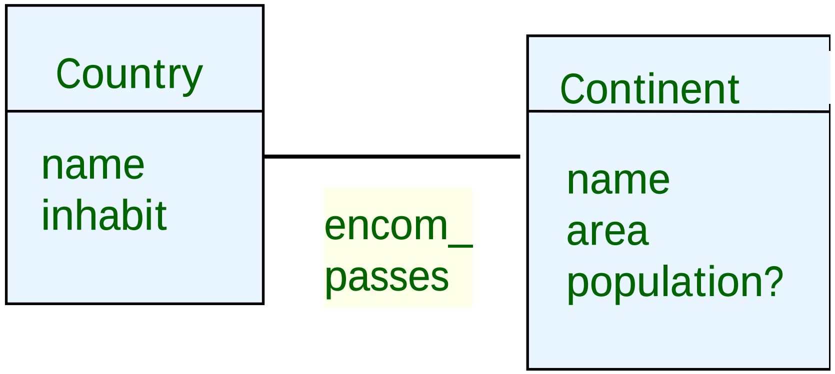
Attribute

property of an entity



Relationship

connects two or more **entities**



En·ti'tät, *die; -,-en* **1.**Dasein eines Dinges **2.** (gegebene) Größe, (Langenscheid)

Basic modeling primitives

Issues

- **Design choices**
  attribute or entity?
  continent: attribute of country or separate entity?

There is never exactly one way of modeling reality.
Many good designs, much more bad designs.

- **Identification**
- e.g. **name** obviously identifies continents but **not cities**

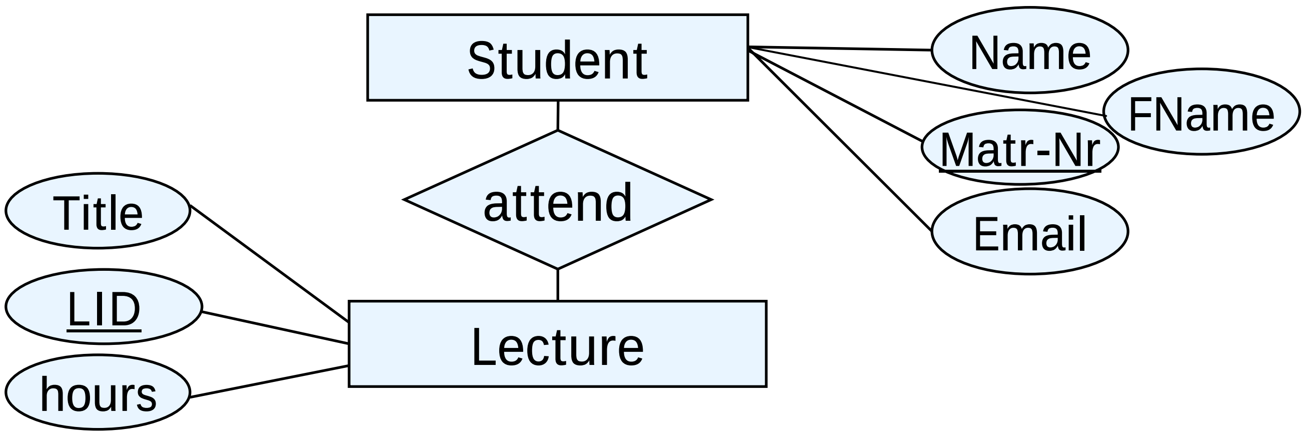
- **Identifying attributes needed at all?**

Freie Universität Berlin

Entity-Relationship-Model (ERM)

- data-oriented: static modeling of data
- 1976 introduced by P.P. Chen
- (Peter P. Chen: The Entity-Relationship Model - Toward a Unified View of Data. ACM TODS 1(1): 9-36, 1976, see Reader)

Traditional graphical notation with squares, bullets and diamond



Graphical modeling languages

Unified modeling language (UML)

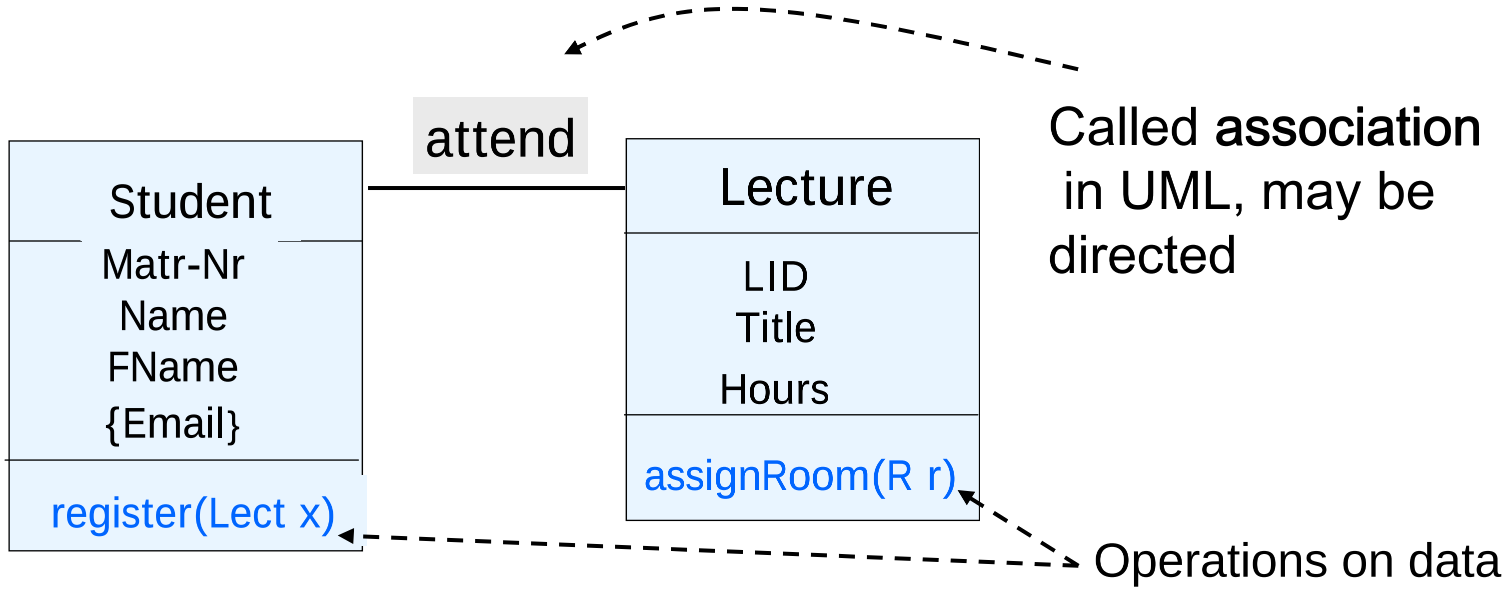
Modeling of **data and operations**

- **Object oriented** flavor
  e.g: each **object (entity) has identity** - a unique pointer
  ERM: entities having the same type and
  the same attribute values are indistinguishable
- **Attributes** may be **constructed** (lists, sets, arrays,…)
- **Relationships** are **directed** (uni- or bidirectional)
  ERM: always bidirectional

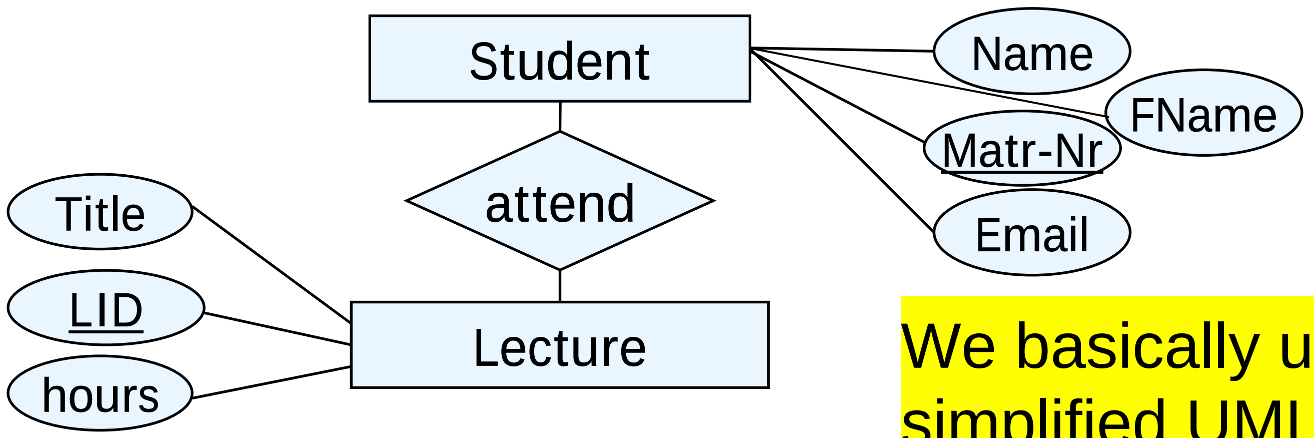
UML versus ERM

Freie Universität Berlin

UML



ERM

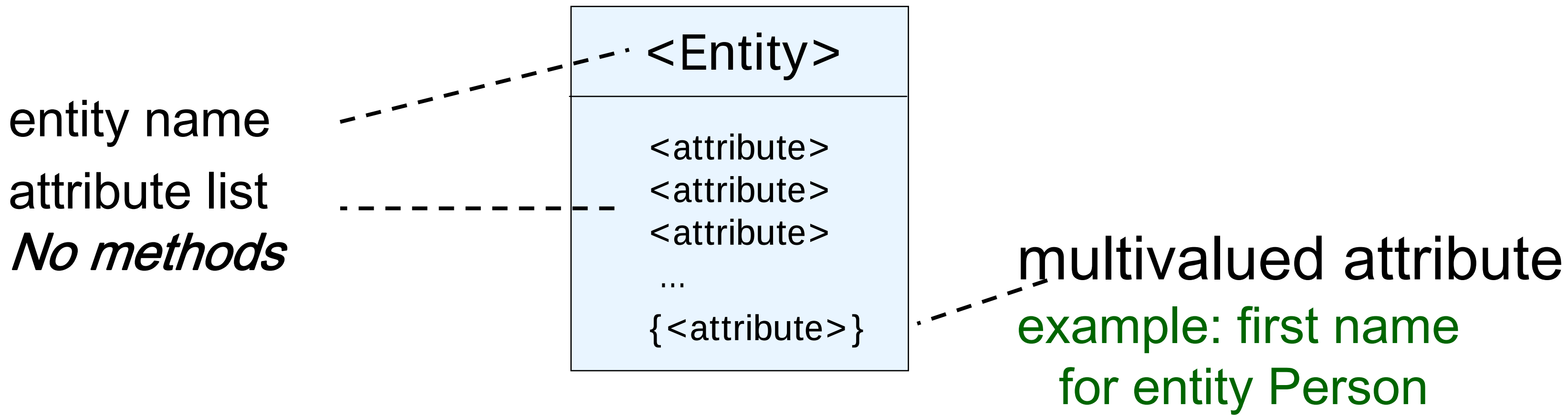


We basically use simplified UML notation

2.2.3 Conceptual Design: Basics

Freie Universität Berlin

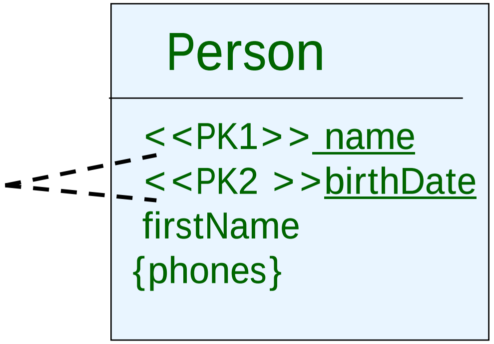
Entities & attributes



Identifying attributes

- "Axiom" of ERM and Relational DB:
  Two individual entities can always be distinguished by the values of some of its attribute(s), together called the key

Key attributes are underlined
or annotated by
<<*PKi*>>



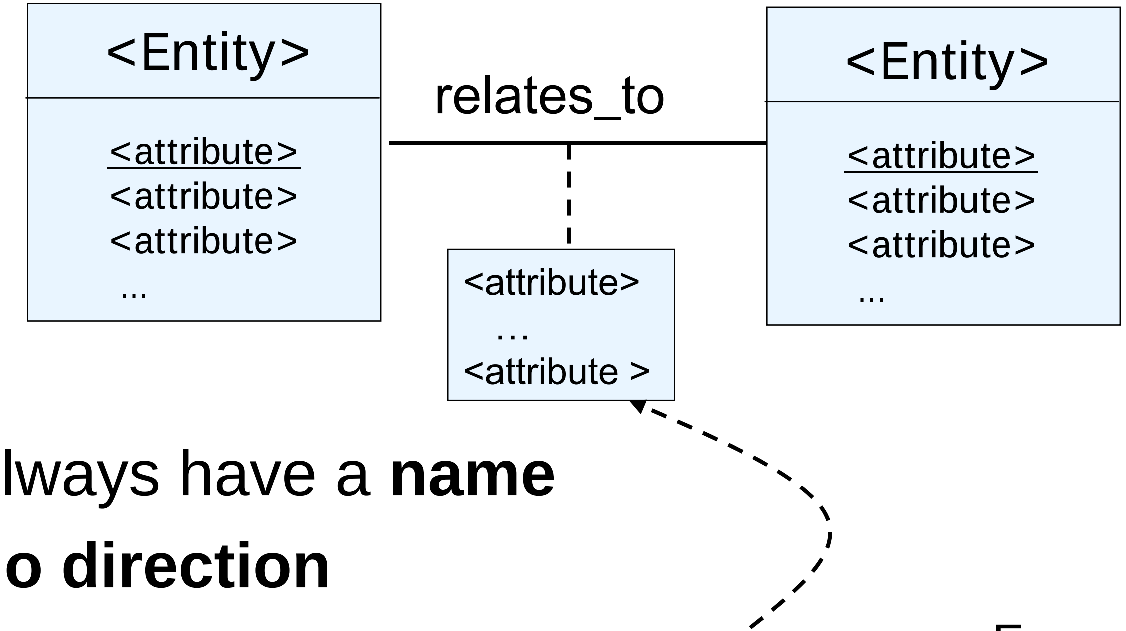
Note: one single attribute may not be identifying for an entity.

<< something>> : UML Stereotype, allows to extend UML – here primary key attributes
Alternative notation: underline all PK attributes (which we use)

Conceptual Design: Basics

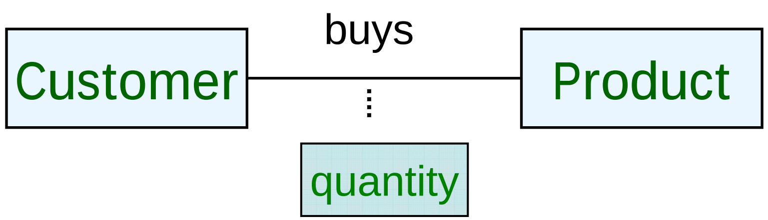
Freie Universität Berlin

Relationships



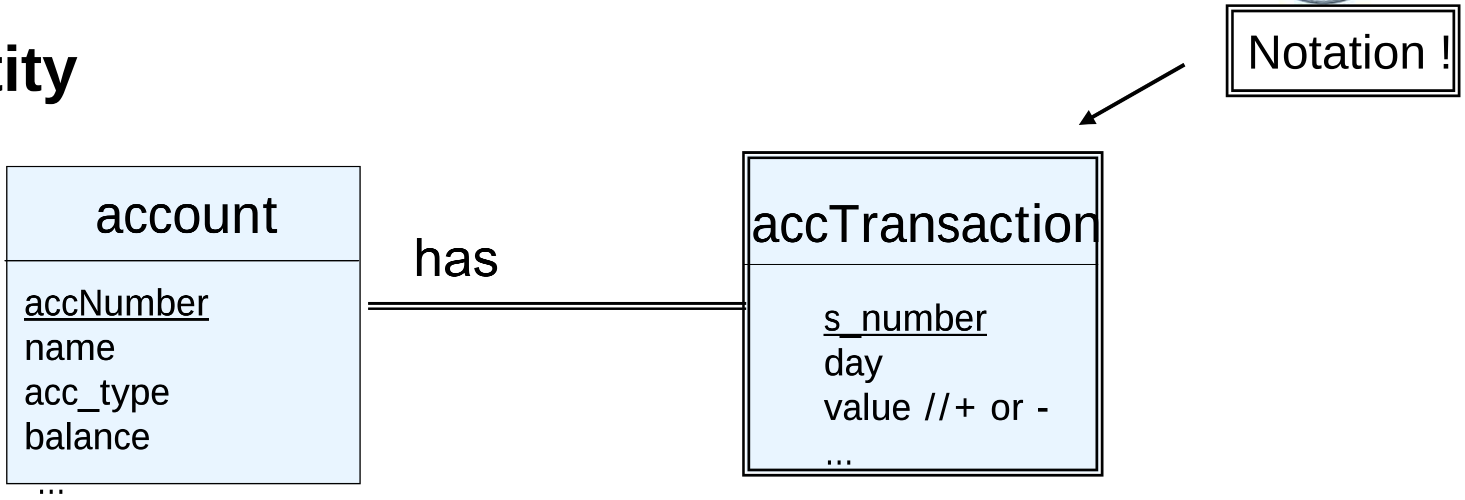
- Always have a **name**
- **No direction**
- May have **attributes**
- **No identifying attributes**

Example:



Freie Universität Berlin

Weak entity



Example: account statement identified by "number" and "acc_number" which is not attribute of 'statement' entity (!)

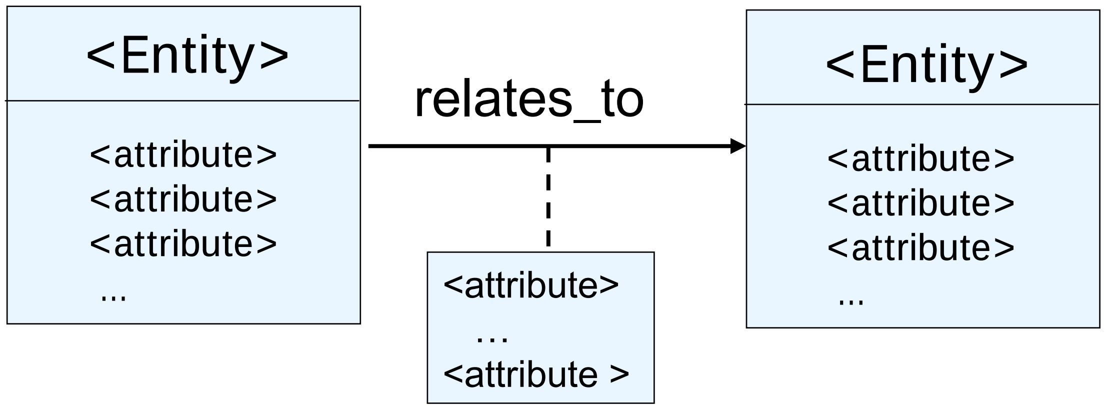
Def:: **A weak entity is** an entity identified by some of its attributes and the relationship to another entity .

Conceptual Design: UML

Freie Universität Berlin

UML-Terminology

- **Class** = **entity type** (UML: attribute = field)
- **Object** = **entity**
- **Association** = **relationship**
- NO keys ("unique address") ⇒ no weak entities
- Relationship may have a direction

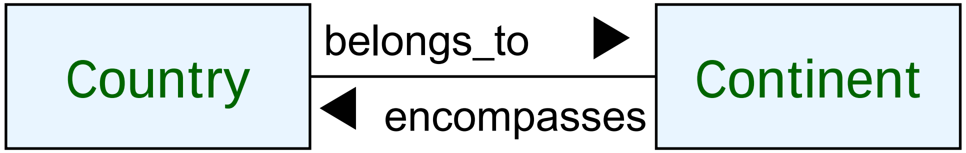


Conceptual Design: Basics

Freie Universität Berlin

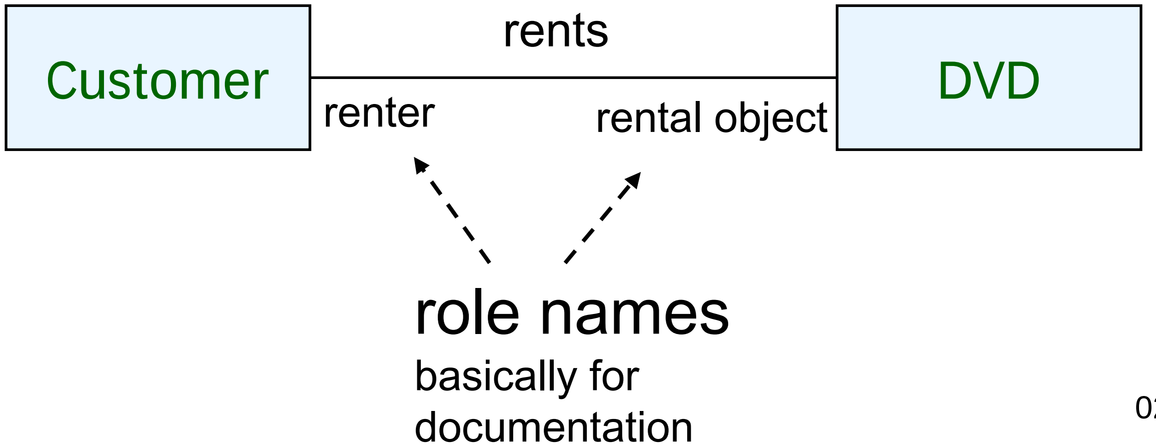
Notation

- Sometimes attributes are omitted
- order of relationship role?



UML-Notation

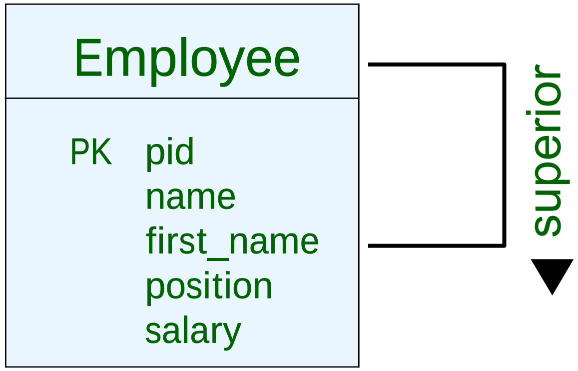
Role names

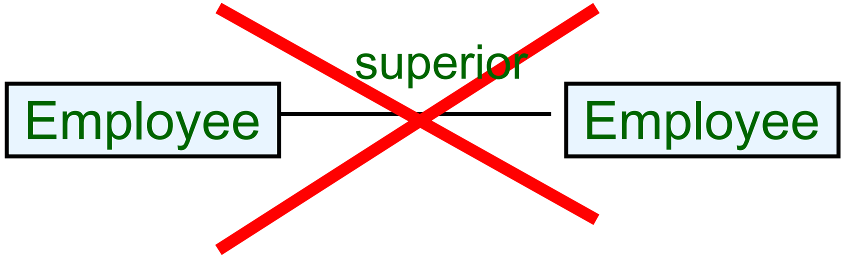


Used in ERM and UML to distinguish the roles of entities in a relationship

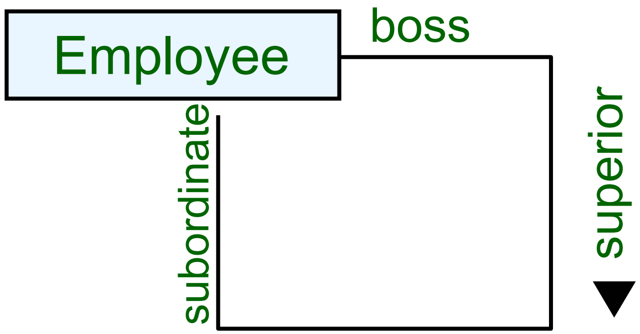
Conceptual Design: Basics

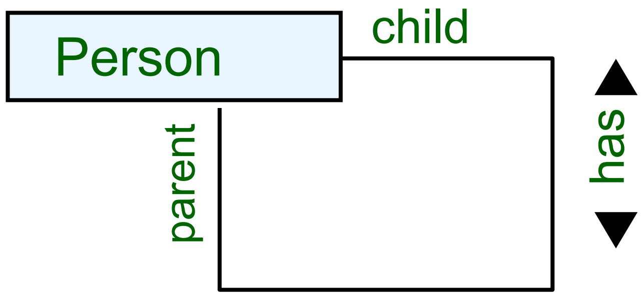
Recursive relationships





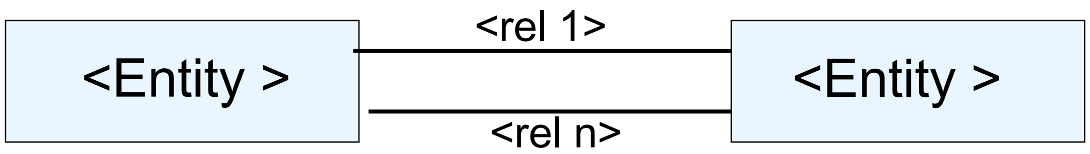
Roles: particularly useful in recursive relations



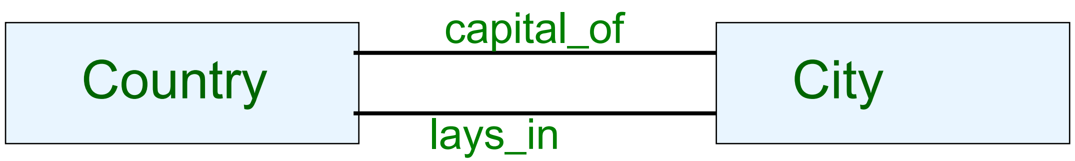


Conceptual Design: Basics

Multiple relationships



There may be **no, one or many relationships between entity types**



2.2.4 From requirements to models

Text to conceptual model

- The only step which **cannot be automated**
- **Requirements** as "cleaned" text
- **conceptual database design**

BTW: nice free graphical Tool for ER and more:
http://dia-installer.de/index_de.html

Text to formal model

Freie Universität Berlin

Rough guideline: correspondence between…

- **entities** - **nouns**

  Every city has...

- **relationships** and **verbs**

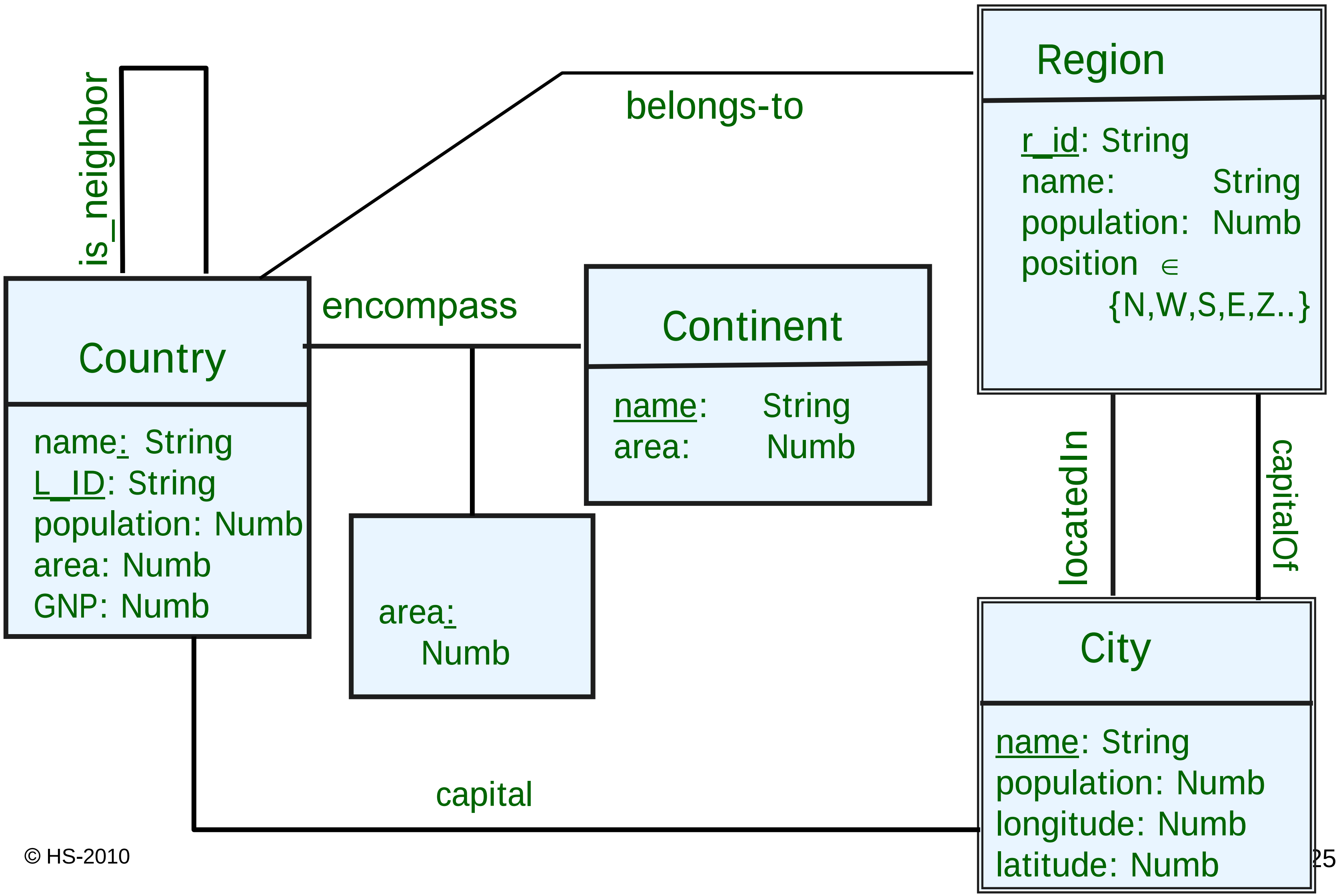
  ...is located in country (exactly one !)

- **attributes** and **adjectives** or phrases like “has a..”, “is...a”

  ...has a GNP (but also: .. has a capital)

Conceptual Design: case study

Freie Universität Berlin



Summary

Freie Universität Berlin

- **Conceptual modeling**: the **art** of structuring the data of an application domain
- Basis: careful **requirement analysis**
- Simple, powerful base constructs: **entities, attributes, relationships**
- Visual (graphical) language
- E-R modeling language and UML related
  - E-R language simpler
  - More appropriate for modeling of data
  - many dialects
  - Compatibility to UML makes sense
  - Some differences, e.g. no keys in UML