

WIEDERHOLUNG

SYSTEMARCHITEKTUR VS SOFTWAREARCHITEKTUR

ZWEI-SCHICHTEN-ARCHITEKTUR

CLIENT-SERVER-ARCHITEKTUR

DREI-SCHICHTEN-ARCHITEKTUR

ANSI-SPARC-ARCHITEKTUR

- 1. INTERNE EBENE/ SCHEMA**
- 2. KONZEPTIONELLE EBENE**
- 3. EXTERNE EBENE**

LOS GEHT'S

\o/

- 1. GRUNDLAGEN: WAS SIND DATENBANKMODELLE**
- 2. ENTITY-RELATIONSHIP-MODELL**
- 3. RELATIONENMODELL**
- 4. HIERARCHISCHES MODELL**
- 5. NETZWERKMODELL**
- 6. OBJEKTORIENTIERTE MODELLE**
- 7. OBJEKTRATIONALE MODELLE**
- 8. XML-BASIERENDE MODELLE**

DATENBANKMODELLE

STATISCHEN EIGENSCHAFTEN

DYNAMISCHEN EIGENSCHAFTEN

UNTERSCHIEDUNG ZWISCHEN OBJEKTEN (DATENBANKSCHEMA ODER DATENBANK) UND KONZEPTEN ZU DEREN DARSTELLUNG (DATENBANKMODELL).

DREISTUFIGE BEZIEHUNG

Enthält		Beispiel
-----	-----	-----
Konzepte zur Darstellung	*Datenbankmodell*	Relationen
-----	-----	-----
Objekte	*Datenbankschema*	Relation "Vorlesung"
-----	-----	-----
Daten	*Datenbank*	"Analysis", "Compilerbau"
-----	-----	-----

WARUM GIBT ES MEHRERE DATENBANKMODELLE?

**WELCHE MODELLE SIND
WICHTIG?**

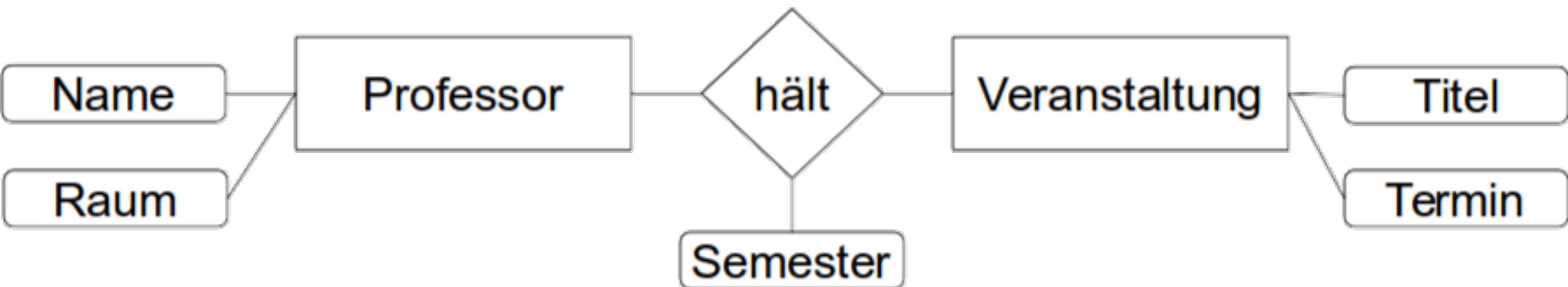
ENTITY- RELATIONSHIP- MODELL

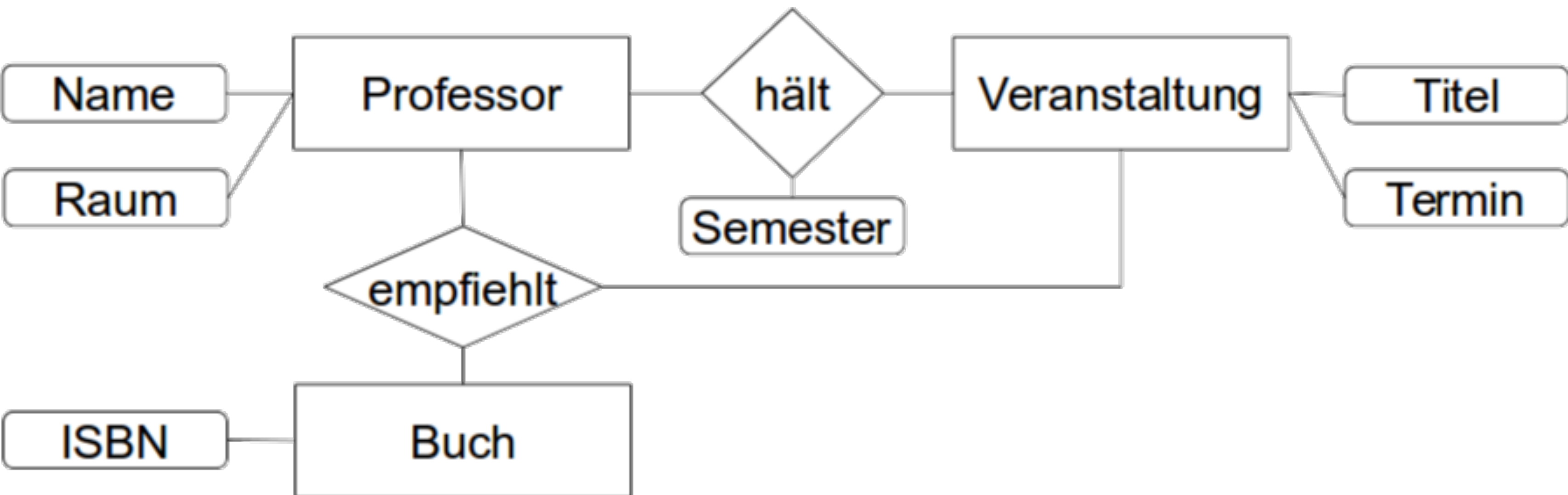
ENTITY

RELATIONSHIP

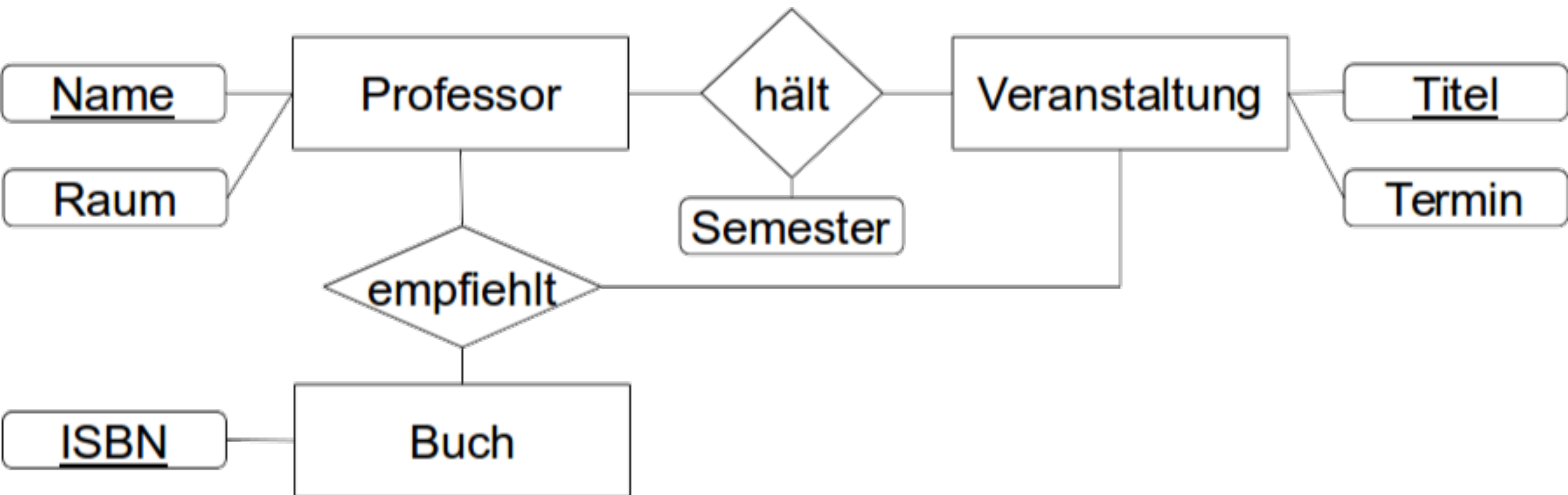
ATTRIBUTE

BEISPIEL

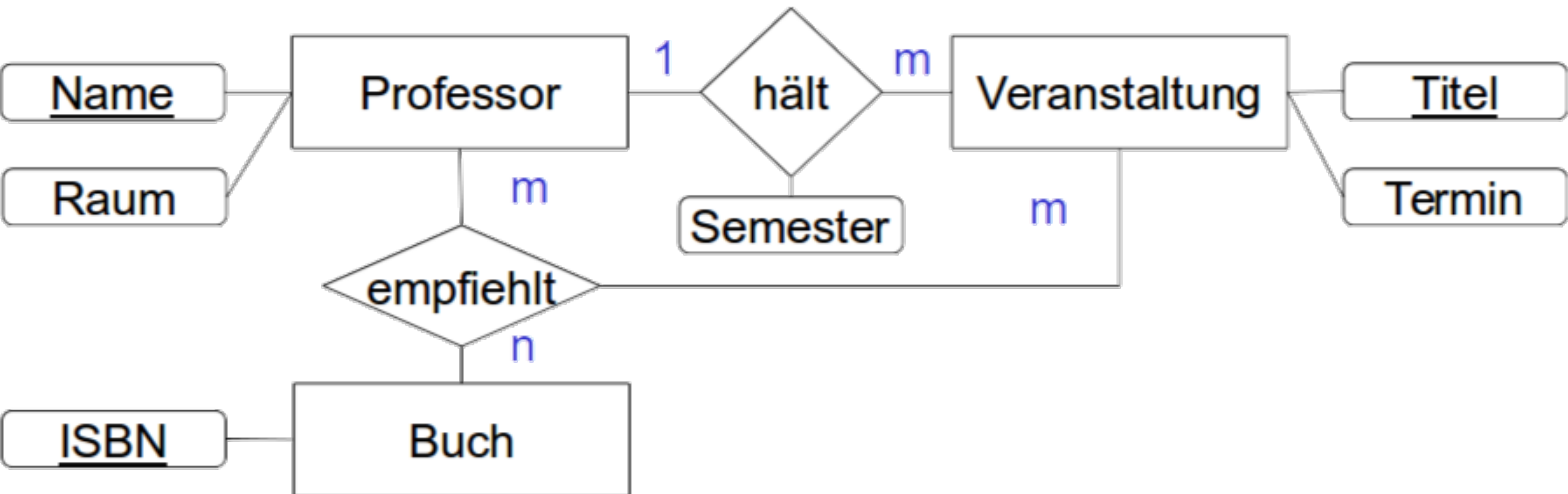




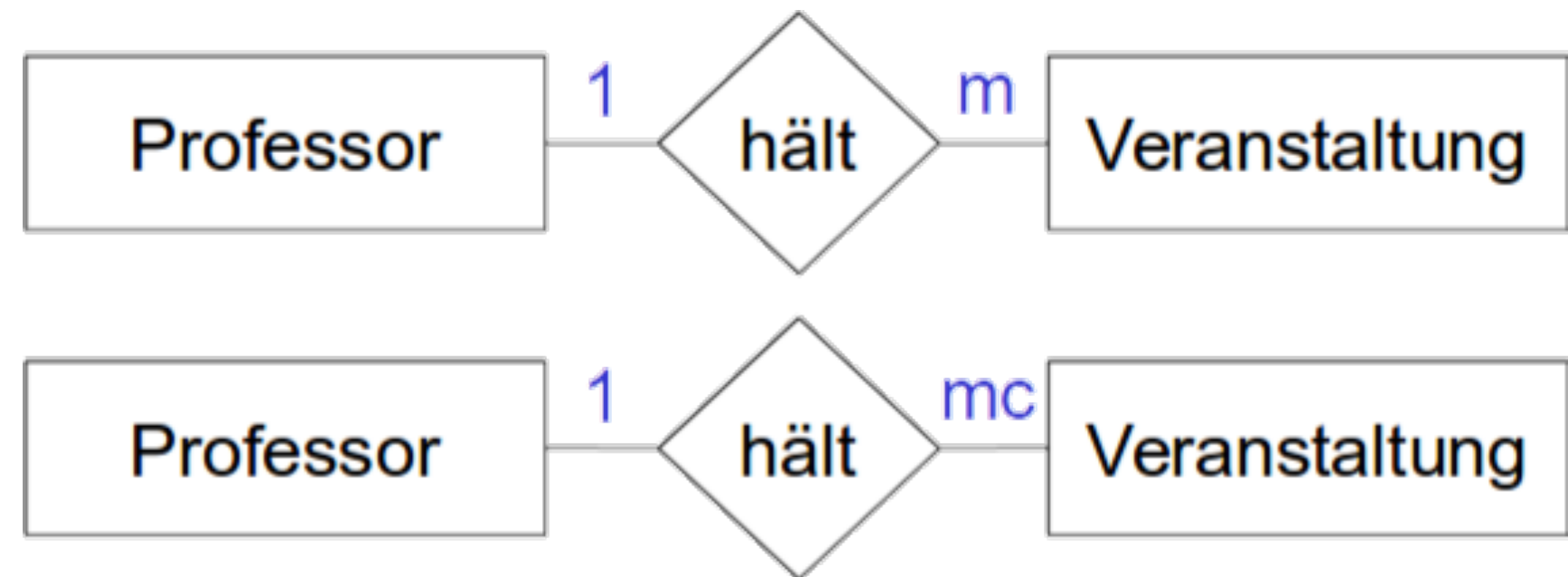
SCHLÜSSEL



KARDINALITÄTEN



- 1 HEIßT GENAU EINS
- N (ODER M) HEIßT EINS ODER MEHRERE
- C HEIßT OPTIONAL UND KANN MIT 1 UND N/
M KOMBINIERT WERDEN.



MANY-TO-MANY-BEZIEHUNG

(N:M-BEZIEHUNG)

MANY-TO-ONE-BEZIEHUNG

(N:1-BEZIEHUNG)

ONE-TO-ONE-BEZIEHUNG

(1:1-BEZIEHUNG)

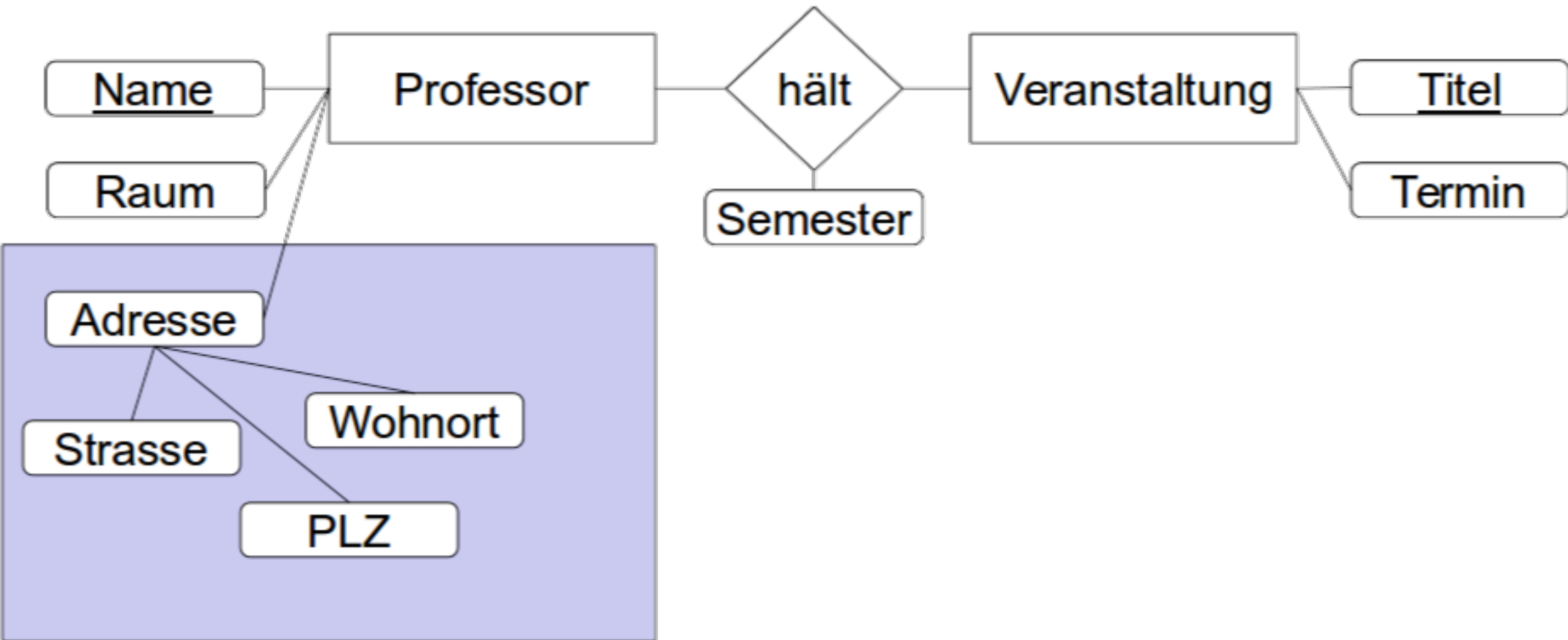
AUFGABE ALS ER-MODELL – 15 MIN

- ES SOLLEN DIE INFORMATIONSZUSAMMENHANGE FÜR EIN FLUGBUCHUNGSSYSTEM EINER FLUGGESELLSCHAFT MODELLIERT WERDEN
- FLÜGE WERDEN DURCH EINE FLUGNUMMER IDENTIFIZIERT, DIE FÜR FLÜGE AM SELBEN TAG EINDEUTIG IST
- PASSAGIERE KÖNNEN EINEN FLUG RESERVIEREN, WAS DURCH EINE RESERVATIONSNUMMER BESTÄTIGT WIRD. EINE RESERVIERUNG WIRD ZU EINER FESTEN BUCHUNG, INDEM MAN EIN TICKET KAUFTE
- BEI DER RESERVIERUNG ODER SPÄTER KÖNNEN PASSAGIERE AUCH EINE SITZPLATZRESERVIERUNG VORNEHMEN
- FÜR TEILNEHMER DES VIELFLIEGERPROGRAMMS IST DIE GESAMTE MIT DER FLUGGESELLSCHAFT GEFLOGENE KILOMETERZAHL VON BEDEUTUNG
 - FLÜGE FLIEGEN VON EINEM BESTIMMTEN FLUGSTEIG AB
- PASSAGIERE MÜSSEN VOR DEM ABFLUG EINE CHECK-IN-PROZEDUR DURCHLAUFEN. DABEI KÖNNEN SIE AUCH GEPÄCKSTÜCKE AUFGEBEN.



ATTRIBUTE

(ERWEITERT)



UML

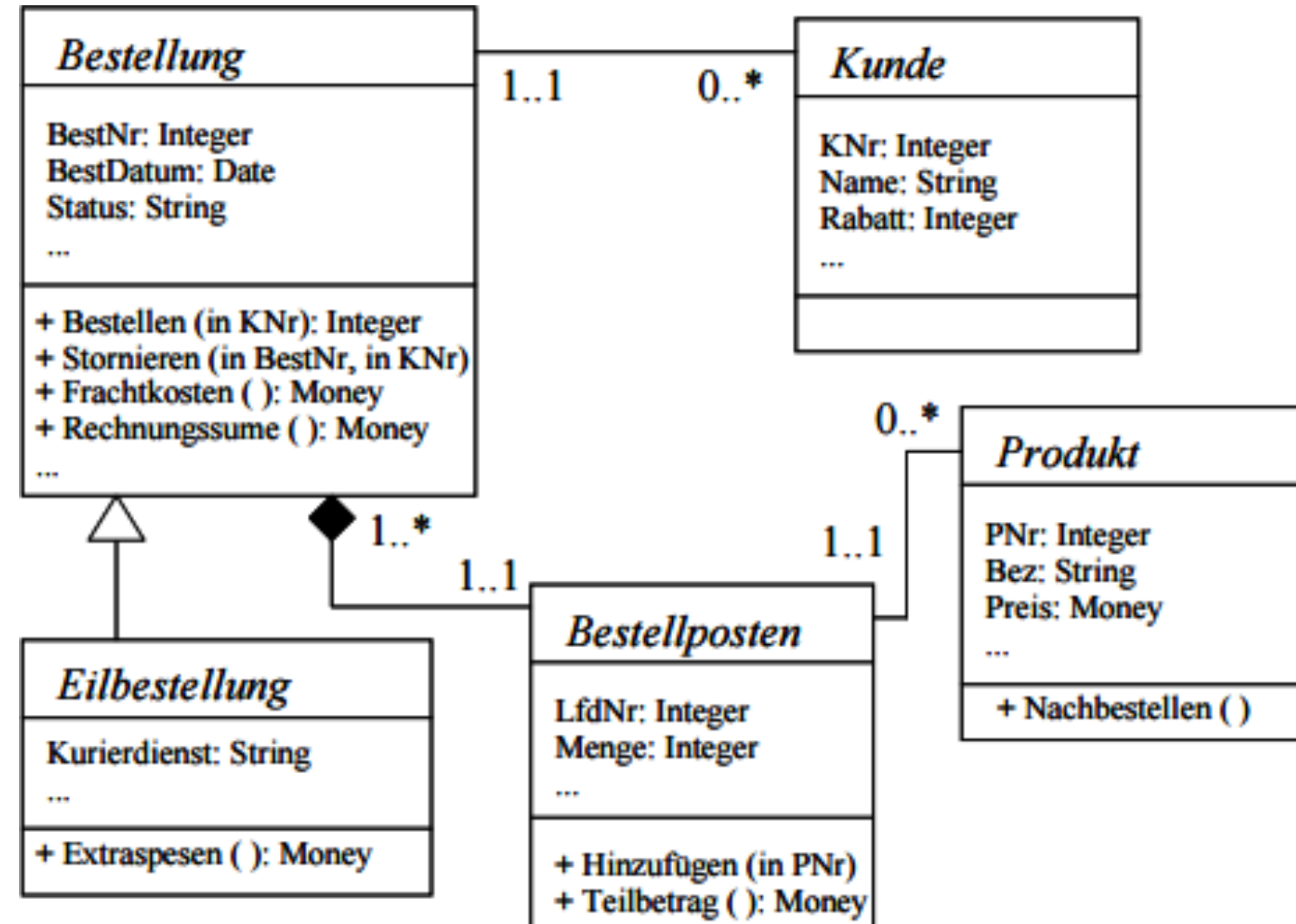
BENUTZUNG

UML VS. ERM

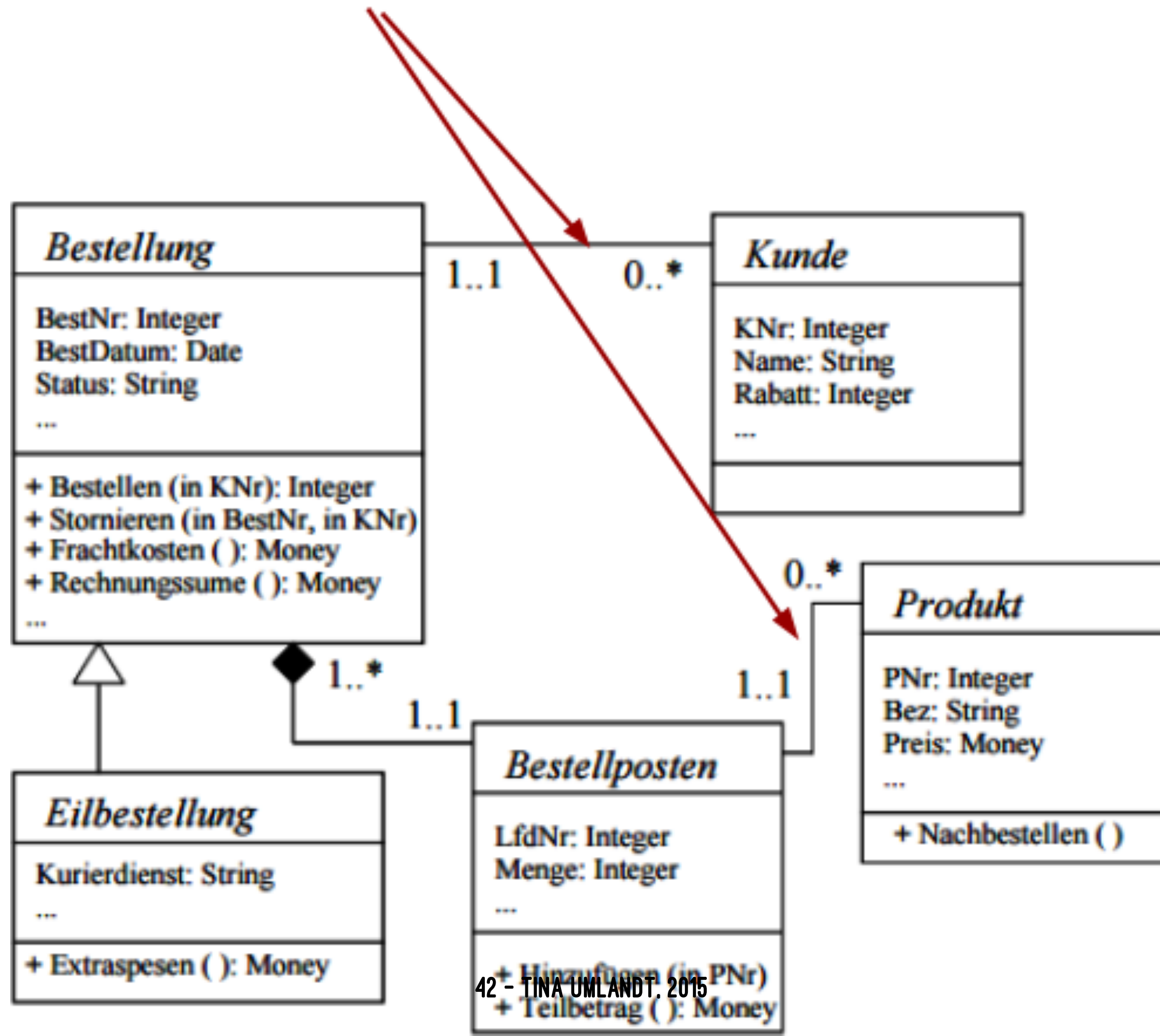
UML

KLASSENDIAGRAMM

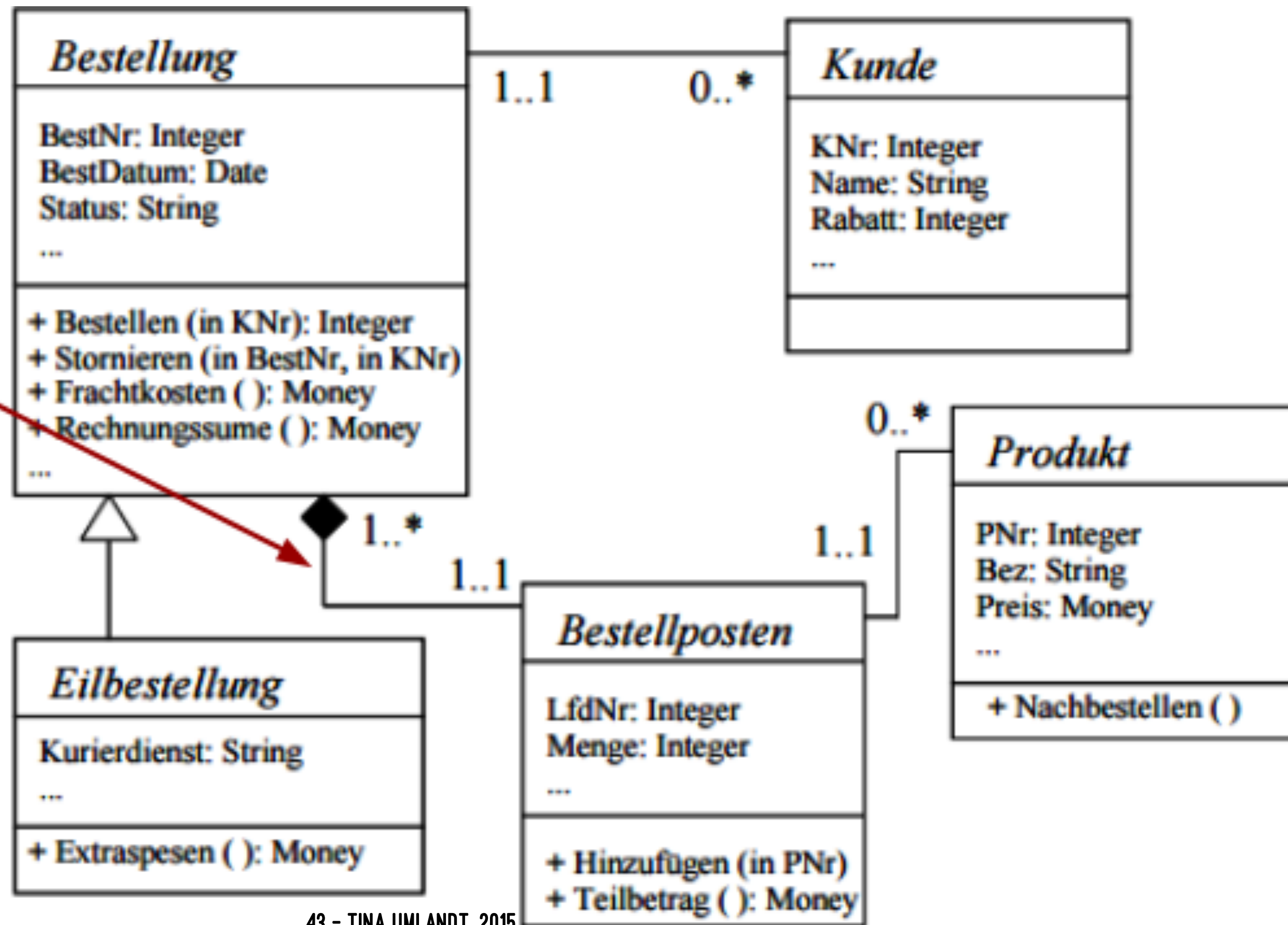
BEISPIEL FUER KLASSENDIAGRAMM



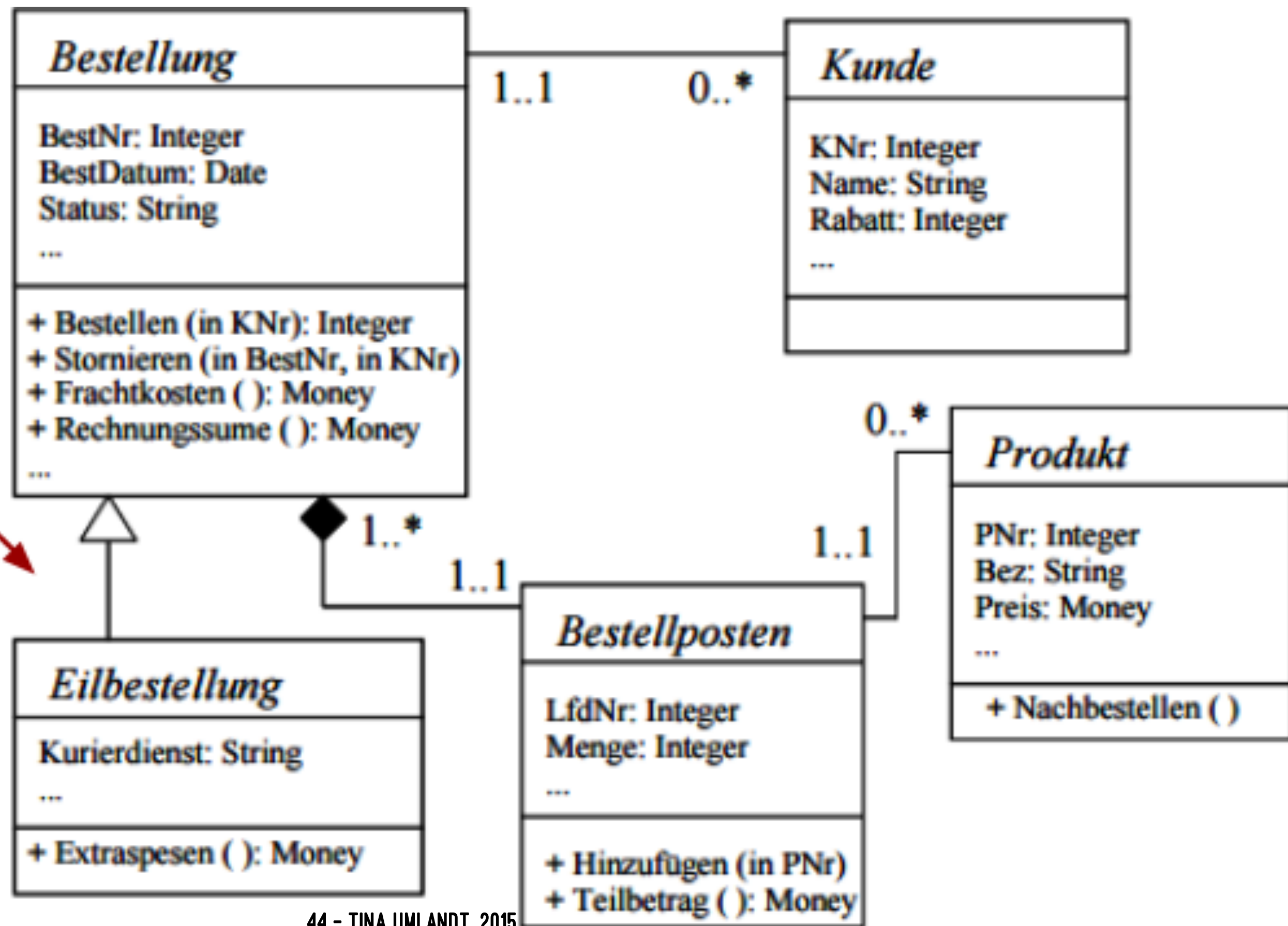
Assoziation



Komposition



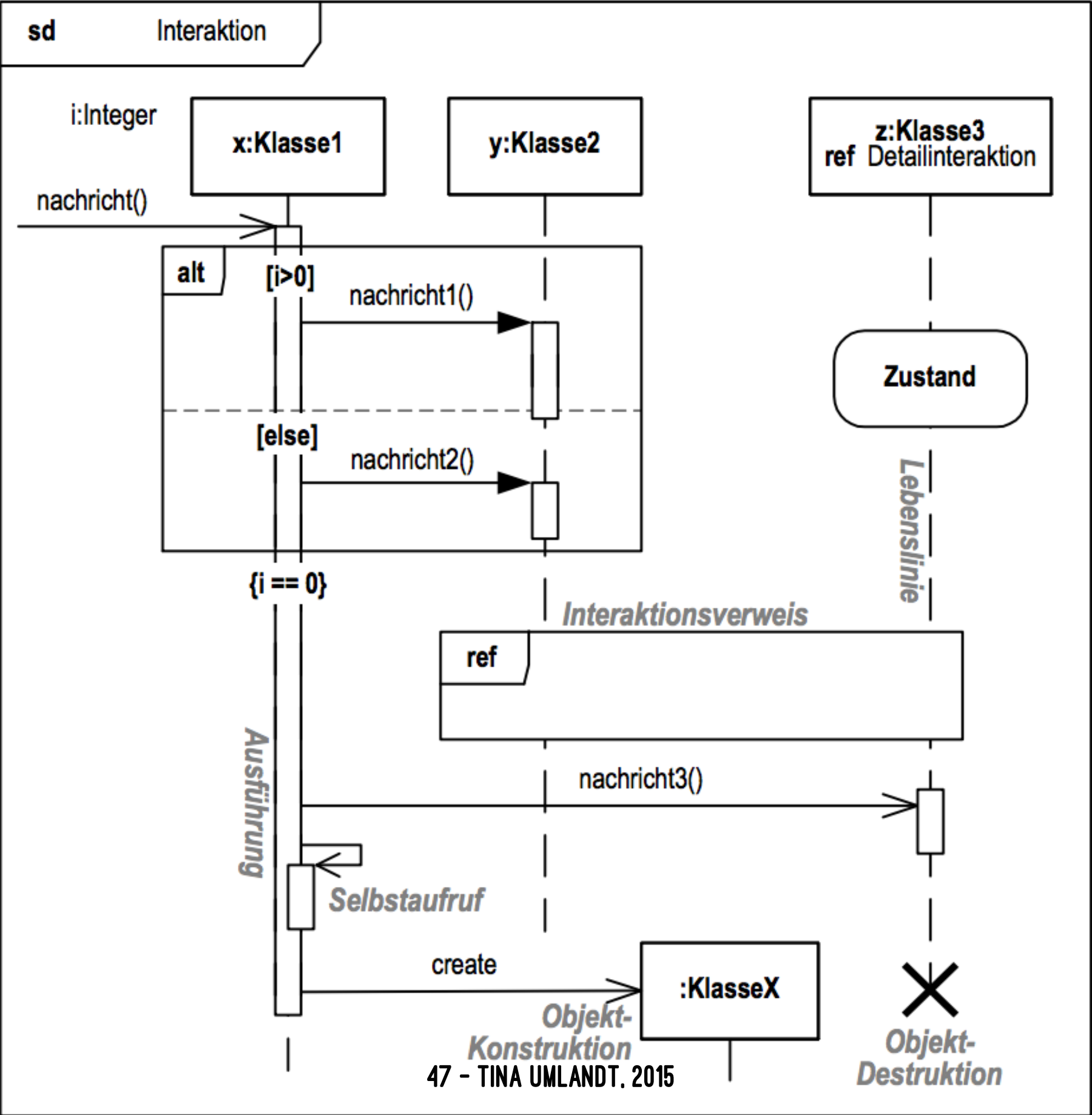
Vererbung



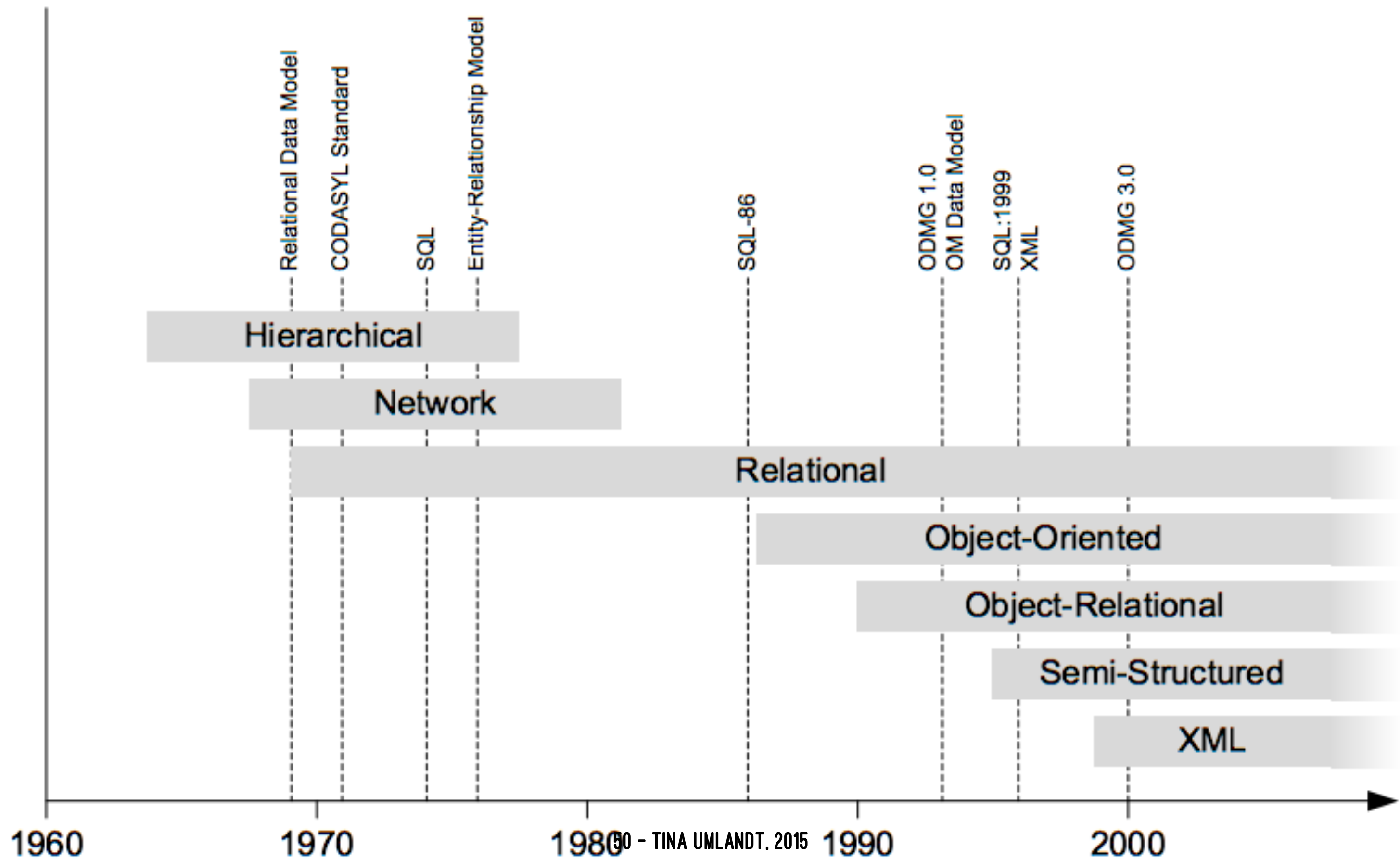
AUFGABE: KONTAKTVERWALTUNG – UML (KLASSENDIAGRAMM)

- **JEDER VERTRIEBSMITARBEITER ERHALT EINE PERSONLICHE KONTAKTVERWALTUNG MIT EINEM ADRESSBUCH**
 - **DIE BUCHHALTUNG KANN BEI BEDARF DAS ADRESSBUCH BENUTZEN**
 - **DER EINTRAG IM ADRESSBUCH (KUNDE) ENTSPRICHT EINER FIRMA BZW. EINER GRÖßEREN EINHEIT**
 - **DER KONTAKT ZU EINEM KUNDEN ERFOLGT ÜBER EINEN SEINER MITARBEITER MIT UNTERSCHIEDLICHEN METHODEN (FAX, TELEFON, E-MAIL USW.)**
- **EIN KONTAKT GEHÖRT ZU EINEM PERSÖNLICHEN ADRESSBUCH. BESUCHEN MEHRERE VERTRIEBSMITARBEITER GLEICHZEITIG DEN KUNDEN, DANN PFLEGEN SIE DEN KONTAKT INDIVIDUELL EIN**
 - **JEDER KUNDE UND JEDER KONTAKT WIRD DURCH KONTAKTGRUPPEN KLASSIFIZIERT**
 - **DIE VERTRIEBSMITARBEITER LEGT DIE KONTAKTGRUPPEN INDIVIDUELL FÜR IHR ADRESSBUCH AN**
 - **DIE KONTAKTGRUPPEN KÖNNEN DABEI IN EINER BAUMARTIGEN HIERARCHIE GELIEDERT WERDEN**
 - **EIN KONTAKT KANN DABEI ZU MEHREREN KONTAKGRUPPEN ZUGEORDNET WERDEN**
- **WIRD DAS ADRESSBUCH GELÖSCHT, SO GEHEN AUCH ALLE KONTAKTE UND KONTAKTGRUPPEN MIT IHM UNTER**
- **WIRD DAGEGEN EINE KONTAKTGRUPPE GELÖSCHT, SO BLEIBEN DIE DARIN ENTHALTENEN KONTAKTE BESTEHEN**

UML SEQUENZDIAGRAMM



DATENMODELLE



RELATIONENMODELL

Messdaten	Tier	Groesse	Gewicht
-----	-----	-----	-----
	Tiger	265 cm	140 kg
-----	-----	-----	-----
	Tiger	230 cm	120 kg
-----	-----	-----	-----
	Leopard	120 cm	40 kg
-----	-----	-----	-----
	Jaguar	165 cm	88 kg
-----	-----	-----	-----
	Jaguar	142 cm	78 kg
-----	-----	-----	-----

RELATIONENSCHEMATA

RELATION

TUPEL

Attribute	Wertebereiche
-----	-----
Tier	String
Groesse	Float
Gewicht	Float

Relationenschema "Messdaten"

<u>Messdaten</u>	Tier	Größe	Gewicht
	Tiger	265 cm	140 kg
	Tiger	230 cm	120 kg
	Leopard	120 cm	40 kg
	Jaguar	165 cm	88 kg
	Jaguar	142 cm	78 kg

Relationenname

Attribute

Tupel

MATHEMATISCH:

RELATION TIER  STRING X FLOAT X FLOAT

INTEGRITÄTSBEDINGUNGEN

SCHLÜSSEL

SCHLÜSSELKANDIDAT > PRIMAERSCHLUESSEL

BEISPIEL ZU SCHLÜSSELN

TABELLE LITERATUR

ISBN	Autor	Buchtitel
0001	Hans	V
0002	Lutz	W
0003	Peter	X
0004	Peter	Y
0005	Ralf	Z

TABELLE KUNDE

Name	Geburtstag	Wohnort
Heinz Hoffmann	01.08.1966	Norden, BBS
Alf Appel	08.11.1957	Mömlingen
Sebastian Sonnenschein	04.08.1979	Hamburg
Klaus Kleber	15.04.1970	Frankfurt
Barbara Bachmann	17.10.1940	Kirchheim

TABELLE ISTCHEF VON

Vorgesetzter	Untergebener
002	104
030	512
115	512
234	993
234	670

FACHLICHE **SCHLÜSSEL**

TECHNISCHE **SCHLÜSSEL**

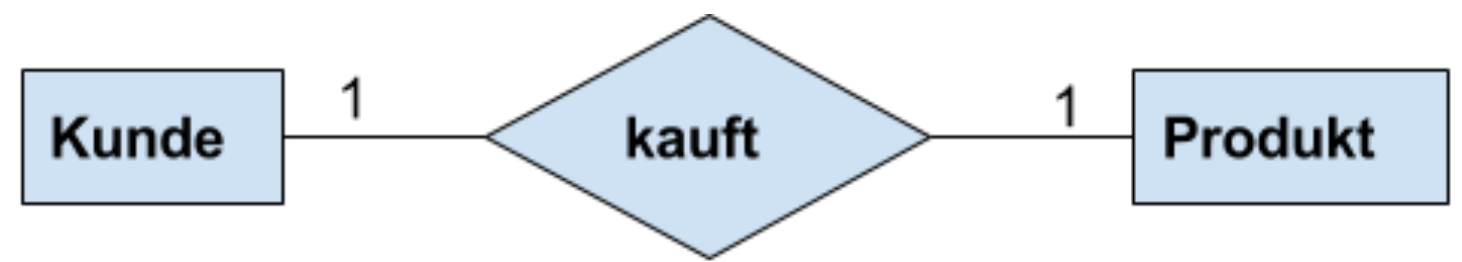
FREMSCHLÜSSEL

Kunde	<u>Name</u>	Adresse
	Meier	Teststr. 42
	Schmidt	Musterweg 85

kauft	Name	Produkt
	Meier	Schal
	Schmidt	Katze

Produkt	<u>Name</u>	Preis
	Schal	120
	Katze	80

TRANSFORMATION ERM → RM



Kunde	<u>Name</u>	Adresse
	Meier	Teststr. 42
	Schmidt	Musterweg 85

kauft	Name	Produkt
	Meier	Schal
	Schmidt	Katze

Produkt	<u>Name</u>	Preis
	Schal	120
	Katze	80

- **JEDER ENTITÄTSTYP WIRD AUF EINE RELATION ABGEBILDET**
 - **DIE ATTRIBUTE DER ENTITÄTSTYPEN WERDEN DEN ATTRIBUTEN DER RELATION ZUGEORDNET**
 - **ES MUSS EIN PRIMÄRSCHLÜSSEL GEWÄHLT WERDEN**
- **ABSCHLIEßEND WERDEN DIE BEZIEHUNGEN TRANSFORMIERT**
- **BEZIEHUNGSTYP INNERHALB DES ERMODELLS WIRD MITTELS EINES EIGENEN FREMDSCHLÜSSELS IM RELATIONENMODELL ABGEBILDET**
- **NM BZW. N:MC – BEZIEHUNGEN WERDEN IN ZWEI 1:N BZW. 1:NC – BEZIEHUNGEN AUFGELÖST.**

AUFGABE

**NEHMEN SIE IHRE OBIGE LOSUNG ZUM ER-MODELL (FLUGHAFEN)
UND WENDEN SIE DAS RELATIONENMODELL FÜR
BEISPIELDATENSÄTZE (DIE SICH AUSDENKEN) AN.**

**FALLS IHNEN AUFFALLT, DASS IHNEN ETWAS IN IHREM ER-MODELL
FEHLT, ERGÄNZEN SIE ES IN BEIDEN MODELLEN.**

OPERATIONEN IM RELATIONENMODELL

SELEKTION

PROJEKTION

VERBUND / JOIN

MENGENOPERATIONEN

UMBENENNUNG

VERGLEICH MIT DEM ER-MODELL

BEZIEHUNGEN

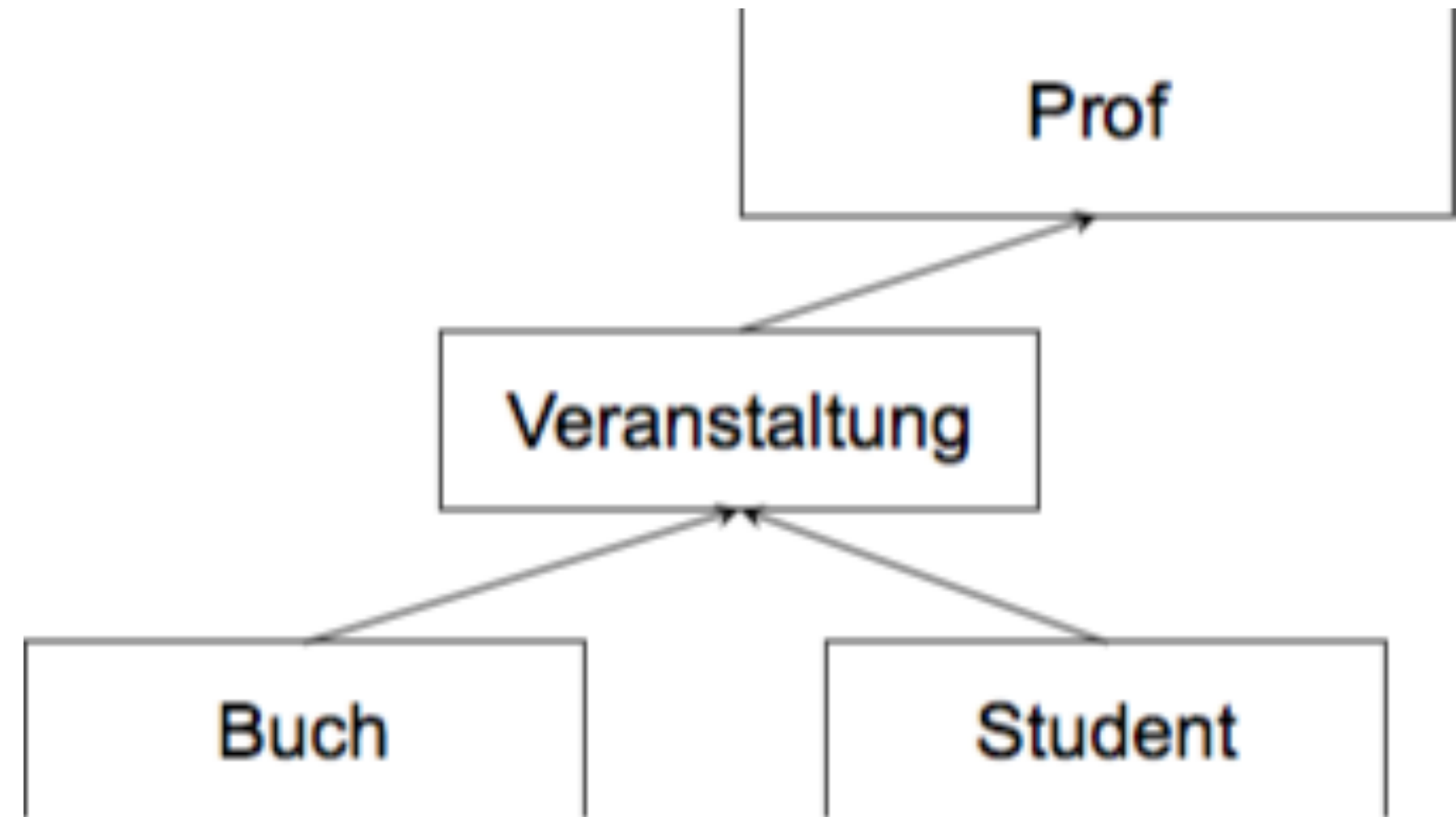
OPERATIONEN

INTEGRITÄTSBEDINGUNGEN

PRÄSENTATION

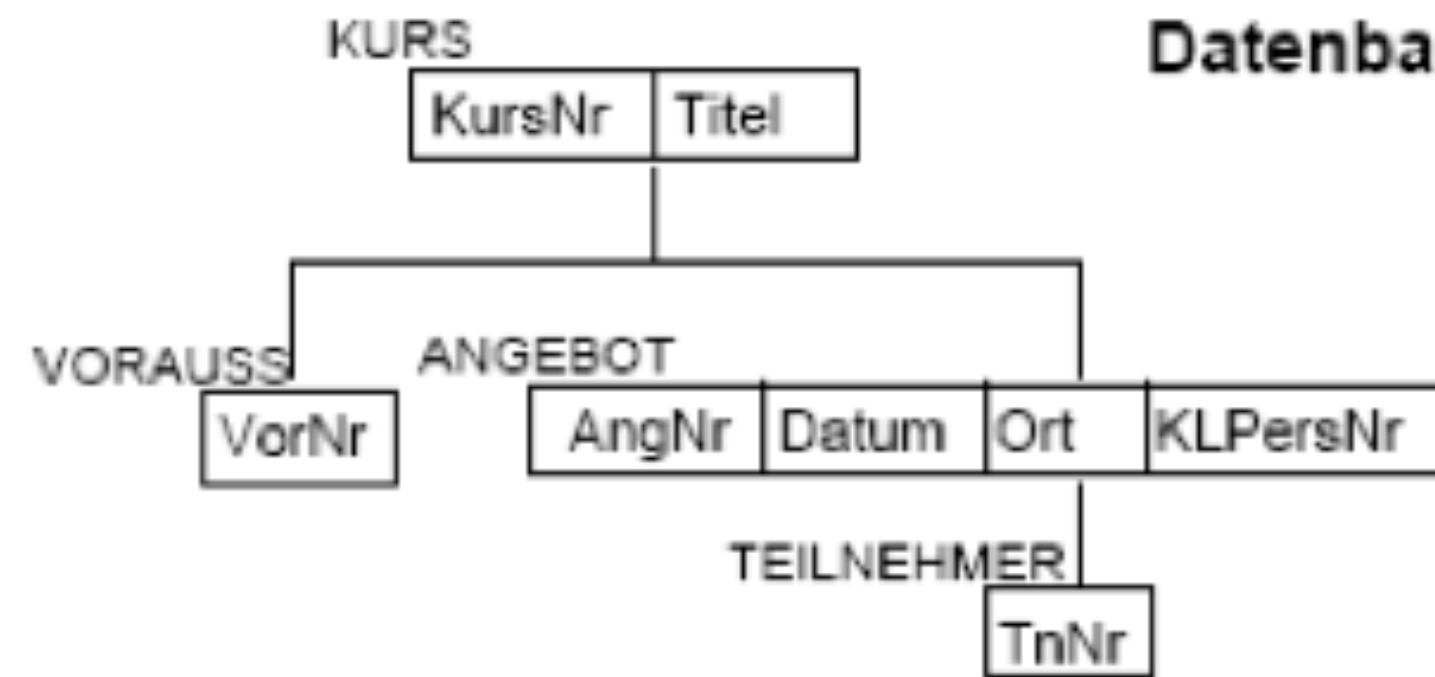
HIERARCHISCHES MODELL

- › DURCHLAUF VON PROF AUSGEHEND ZU VERANSTALTUNG, DANN ZU BUCH UND DANN ZU STUDENT
- › WAS NICHT FUNKTIONIERT: DIREKT ZU STUDENT
- › DESHALB: GUTER ENTWURF DER BÄUME WICHTIG!



KursAngebotDB

Datenbank 1



KursleiterDB

Datenbank 2

KURSLEITER

PersNr	Name	Gehalt
--------	------	--------

TeilnehmerDB

Datenbank 3

TEILNEHMER

TnNr	Name	Ort
------	------	-----

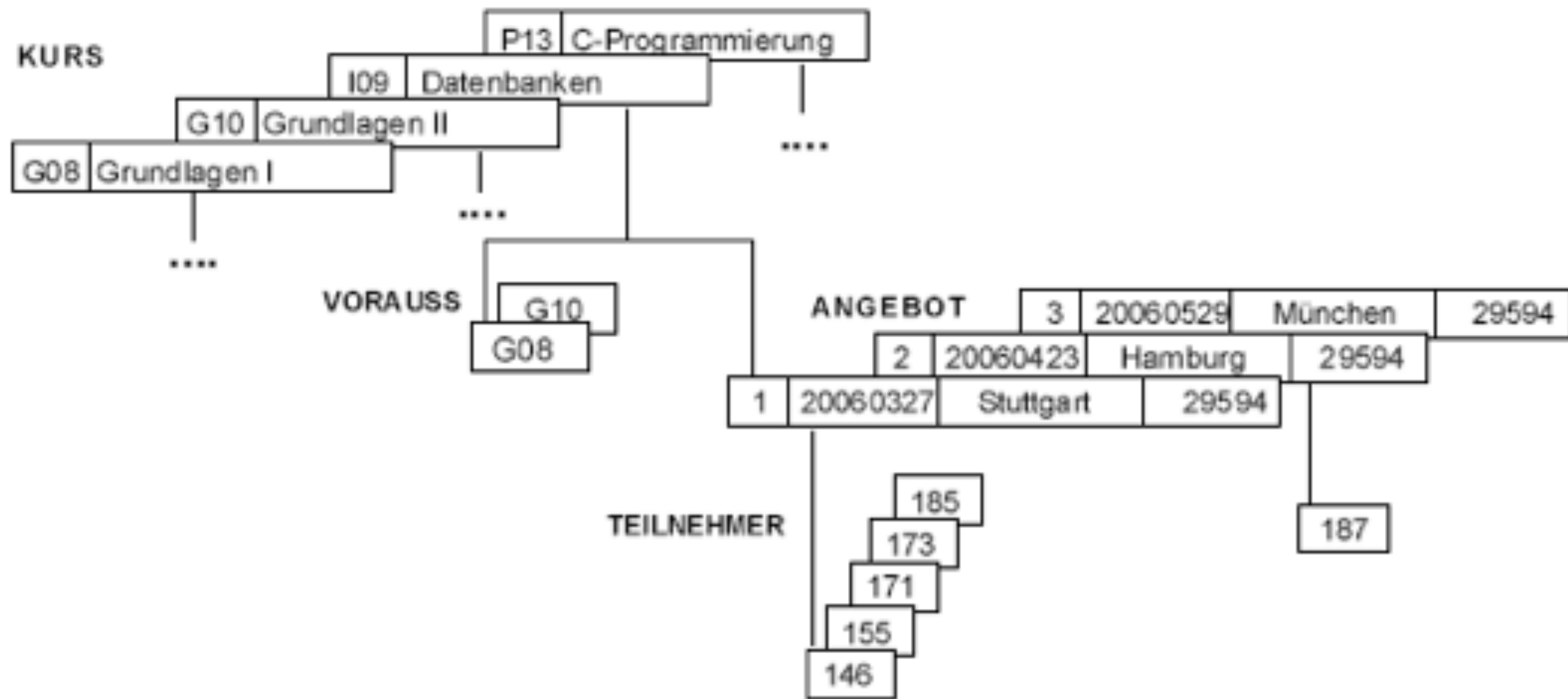


Abb. 7-4: Ausprägungen KursAngebotDB

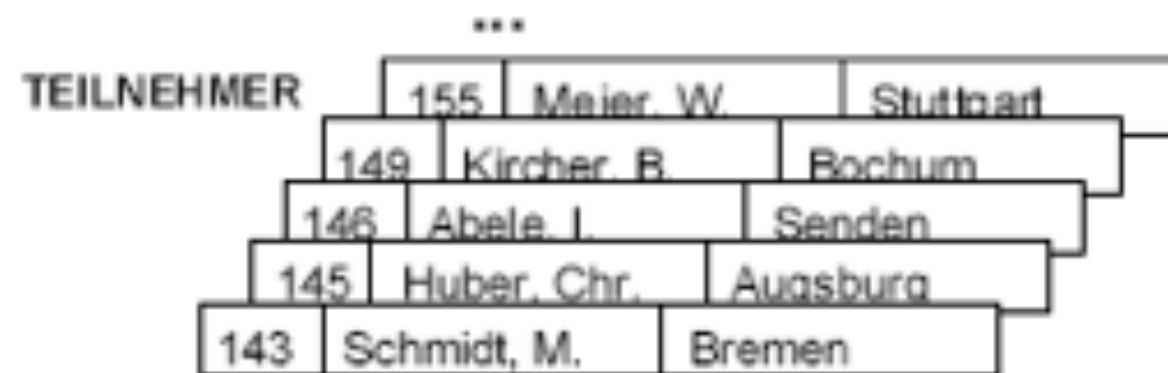


Abb. 7-5: Ausprägungen TeilnehmerDB

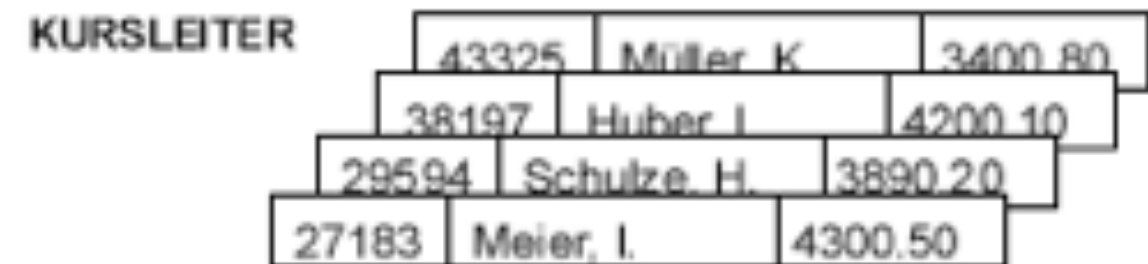
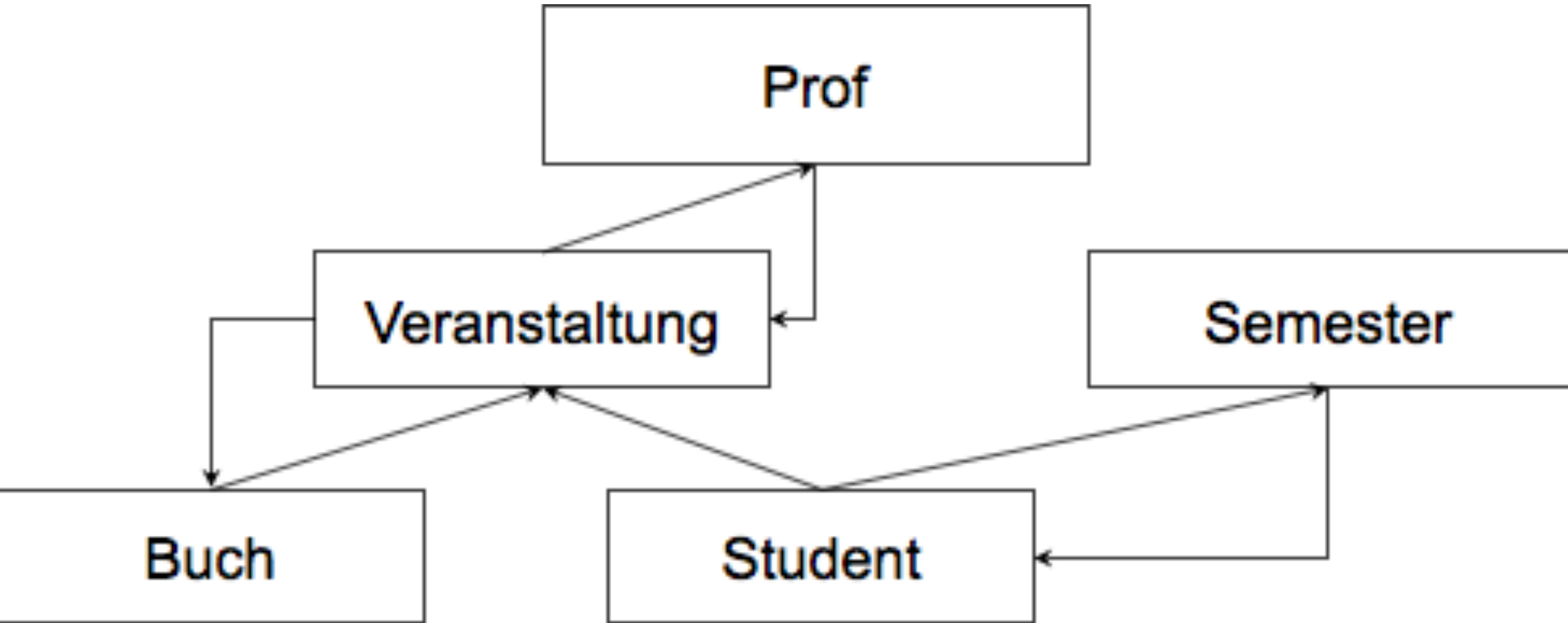


Abb. 7-6: Ausprägungen KursleiterDB

NETZWERKMODELL



OBJEKT-ORIENTIERTES MODELL

ODMG-STANDARD

OBJEKTMODELL

DATENBANKDEFINITIONSSPRACHE ODL

DATENABFRAGESPRACHE OQL

SPRACHEINBETTUNGEN FÜR PROGRAMMIERSPRACHEN (C++, SMALLTALK, JAVA)

VERHALTEN

STRUKTURIERTE TYPKONSTRUKTOREN

ATOMARE DATENTYPEN

BEISPIEL

```
interface Auto: KFZ {  
    extent autos;  
    key fahrgestell_Nr;  
    attribute string autoArt;  
    relationship MotorTyp autoMotor inverse MotorTyp::eingebaut_in;  
    void verkaufen (IN Person) raises (Schrottreif);  
};
```

BEISPIEL

```
Query q = new Query (  
    Employee.class,  
    "manager.salary < salary");  
Collection<Employee> result = q.select(employees);
```

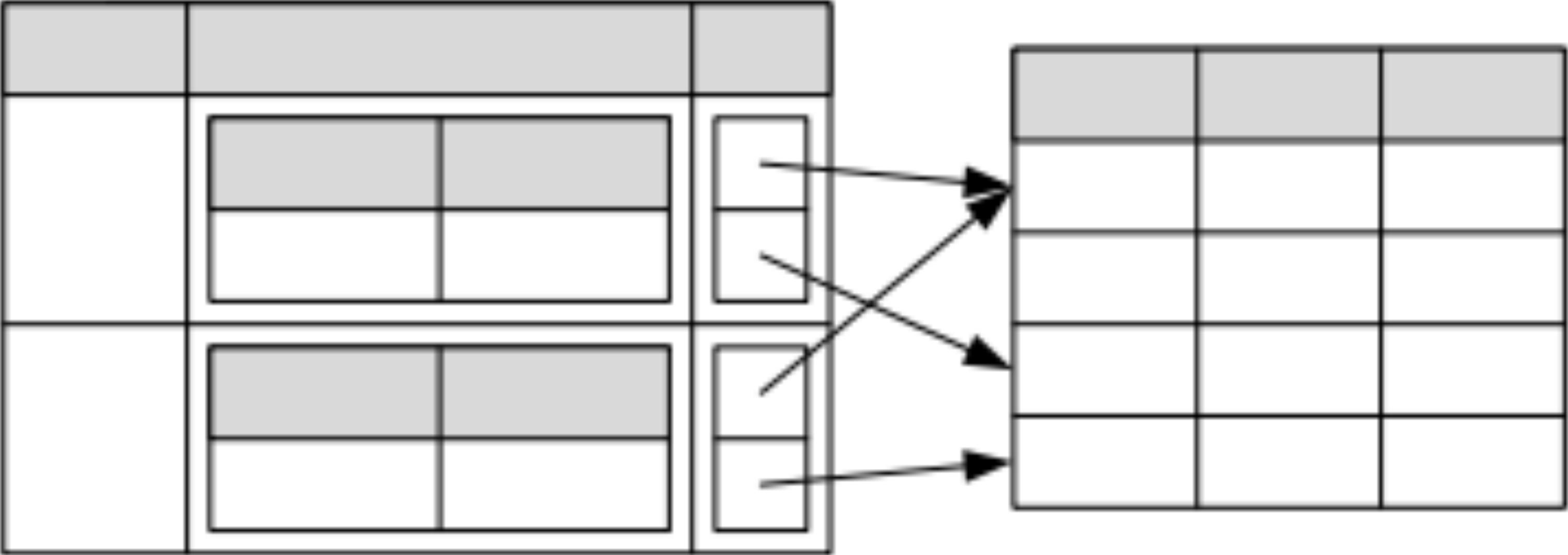
```
SELECT EMP.ID, EMP.NAME  
FROM EMPLOYEE EMP, EMPLOYEE BOSS  
WHERE EMP.BOSS = BOSS.ID AND BOSS.SALARY < EMP.SALARY
```

AUFGABE

**NEHMEN SIE IHRE OBIGE LOSUNG ZUM ER-MODELL (FLUGHAFEN)
UND WENDEN SIE DAS OBJECT-ORIENTIERTE MODELL AN.**

**FALLS IHNEN AUFFALLT, DASS IHNEN ETWAS IN IHREM ER-MODELL
FEHLT, ERGÄNZEN SIE ES IN BEIDEN MODELLEN.**

OBJEKT-RELATIONALES MODELL



UDT

DISTINCT TYPES

```
CREATE TYPE Aepfel AS integer NOT FINAL;
```

```
CREATE TYPE Birnen AS Integer FINAL;
```


STRUCTURED TYPES

```
CREATE TYPE Adresse AS  
  ( PLZ integer,  
    Strasse varchar(50),  
    Nummer varchar (15)  
  ) not final;
```

ARRAYS

```
CREATE TYPE Noten AS  
INTEGER ARRAY[25];
```

```
ARRAY[ ]
```

```
ARRAY[<Werteliste>]
```

```
ARRAY[<SQL-Anfrage>]
```

OBJEKTORIENTIERTE MERKMALE

FINAL ODER NOT FINAL

NOT INSTANTIABLE

METHODEN

```
CREATE TYPE Aepfel ... NOT FINAL
```

```
METHOD ernten() returns integer;
```

```
CREATE METHOD ernten () FOR Aepfel <Methodenrumpf>
```

EINFACHVERERBUNG

```
CREATE TYPE Boskop UNDER Aepfel AS ...  
INSTANTIABLE NOT FINAL
```

```
OVERRIDING METHOD ernten () returns integer;
```


TABELLEN AUF UDT

```
CREATE TABLE Apfeltabelle OF Aepfel
```

```
CREATE TABLE Boskoptabelle OF Boskop UNDER Apfeltabelle
```

AUFGABE:

**NEHMEN SIE IHRE OBIGE LOSUNG ZUM ER-MODELL (FLUGHAFEN)
UND WENDEN SIE DAS OBJECT-RELATIONALE MODELL AN.**

**FALLS IHNEN AUFFALLT, DASS IHNEN ETWAS IN IHREM ER-MODELL
FEHLT, ERGÄNZEN SIE ES IN BEIDEN MODELLEN.**

XML-BASIERENDES MODELL

<Buch>

<Autor>

<Name>Sunzi</Name>

<Geburtsjahr>534 v.Chr</Geburtsjahr>

<Sterbejahr>453 v.Chr</Sterbejahr>

</Autor>

<Titel>Kunst des Krieges</Titel>

<Kapitel>Planung</Kapitel>

<Kapitel>Über die Kriegskunst</Kapitel>

<Kapitel>Taktik</Kapitel>

</Buch>

ANFRAGESPRACHE XQUERY

AUFGABE 1: WAS WILL DAS STATEMENT?

AUFGABE 2: WIE MUSS DAS ZUGEHORIGE XML AUSSEHEN?

```
for $x in doc('books.xml')/bookstore
  where $x/price > 30
  order by $x/title
return $x/title
```

AUFGABE:

**NEHMEN SIE IHRE OBIGE LOSUNG ZUM ER-MODELL (FLUGHAFEN)
UND WENDEN SIE DAS XML-BASIERTE MODELL AN.**

**FALLS IHNEN AUFFALLT, DASS IHNEN ETWAS IN IHREM ER-MODELL
FEHLT, ERGÄNZEN SIE ES IN BEIDEN MODELLEN.**

ZUSAMMENFASSUNG



**DAS WAR'S FÜR
HEUTE**