

Kap 4: Abbildung des E/R Modells auf das relationale Modell

Verfeinerung von Beziehungsarten

Entity $a \in A$	steht in Bez. r	Anzahl der $b \in B$
		≤ 1
		$= 1$
		≥ 0
		≥ 1

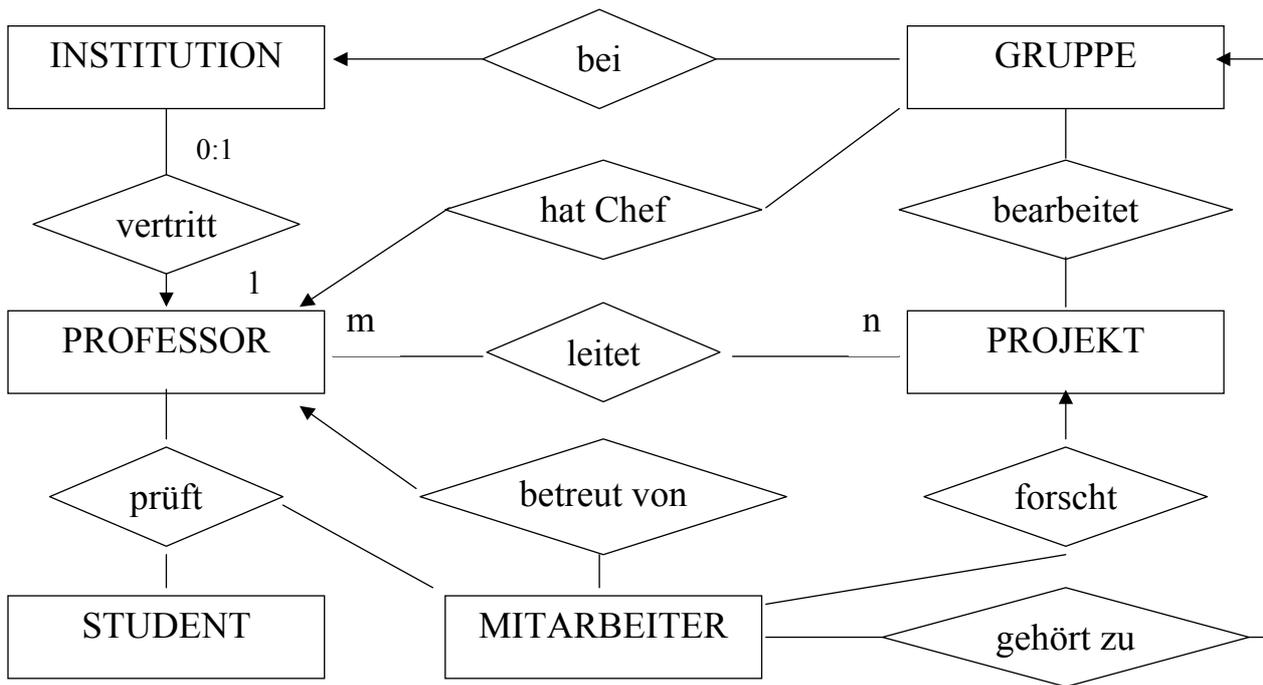
1

Kap. 4.1 Abbildung von Entities

Entity-Schema	→	Relationenschema
Entity-Menge	→	Relation
Entity-Instanz	→	Tupel
Identifikator	→	Primärschlüssel
Attribut	→	Attribut
Domäne	→	Domäne

2

Beispielschema Universität



3

Schritt 1: Relationen mit Attributen, welche Primärschlüssel ?

INSTITUTION

Name
Uni

PROF

P-Name
Tel #
Zi #

PROJEKT

P-Bez
Beginn
Ende

MITARB

Fam-N
Vor-N
ect.

GRUPPE

G-Bez

Student

S-Name ...

4

Schritt 2: Domänen für Attribute

int smallint date time

char (7) varchar (24)

create domain name as Varchar (24)

create domain bez as Varchar (10)

create domain plz as Char (7)

5

Schritt 3: Relationen für Entities vereinbaren

Create table PROF (

 P-Name name **primary key,**

 Tel # **int**

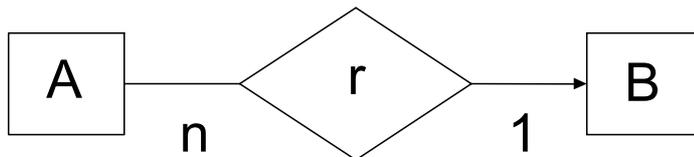
 Zi # **smallint not null)**

6

Kap 4.2 Abbildung von Relationships

Def. : Fremdschlüssel (foreign key) ist ein Primärschlüssel oder ein Schlüsselkandidat einer Rel. R, der außerhalb von R verwendet wird. Dann muss R ein Tupel mit diesem Schlüssel haben. (Referentielle Integrität)

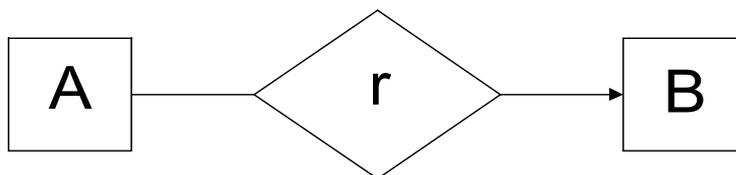
z.B.



Variante 1: Repräsentiere r als Fremdschlüssel von B in A, d.h. Erweiterung von A um Attribute

7

Variante 2: Repräsentiere r als eigenständige neue Relation R mit Attributen (A-Schlüssel, B-Schlüssel)



$\rightarrow \leq 1$

$\xrightarrow[\text{null}]{\text{not}} = 1$

R ≥ 0

R ≥ 1

}

Mit Zusatz-
Bedingung
auf R

8

Charakterisierung durch Matrix:

A \ B	≤ 1	$= 1$	≥ 0	≥ 1	zu $a \in A$ gibt es so viele b
≤ 1	↗	↗	↗	↗	
$= 1$	↗	↗	↗	↗	
≥ 0	↗	↗	R	R	
≥ 1	↗	↗	R	R	

Legende: ↗ foreign key von B in A
 ↗ foreign key, not null!
 R eigene Relation

9

Beispiel:

bei	$= 1$	zu	GRUPPE mit not null
vertritt	$= 1$	zu	INSTITUTION
leitet			
bearbeitet			
forscht	≤ 1	zu	MITARB
prüft			
hat Chef	$= 1$	zu	GRUPPE
betreut	$= 1$	zu	MITARB

nach Variante 1:

Gruppe (...
bei 1 name **not null,**
bei 2 bez **not null,**
hat Chef name **not null)**

INSTITUTION (...
vertritt name **not null)**

MITARB (...
forscht bez,
betreut name **not null)**

11

nach Variante 2: Rel. für Beziehungen:

```
create table leitet (  
  Professor        name,  
  Projekt         Bez,  
  primary key    ( Professor, Projekt    ))
```

```
create table bearbeitet (  
  Gruppe         bez,  
  Projekt        bez,  
  primary key    ( Gruppe, Projekt     ))
```

12

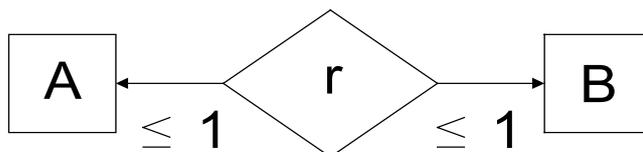
create table prüft (

Professor	name,
Student	name,
Mitarbeiter	name,
primary key	(Professor, Student, Mitarbeiter))

Zusätzlich: weitere Attribute,
z.B. Datum, Note, ...

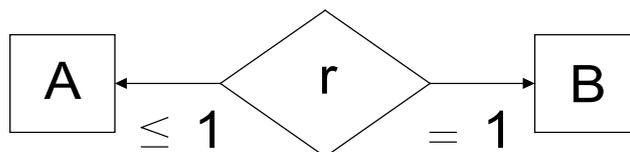
Variante 3: Sonderfälle:

1)



Erweitere A **oder** B um anderen Fremdschlüssel

2)



Erweitere A, Begründung ?

3) = 1 = 1

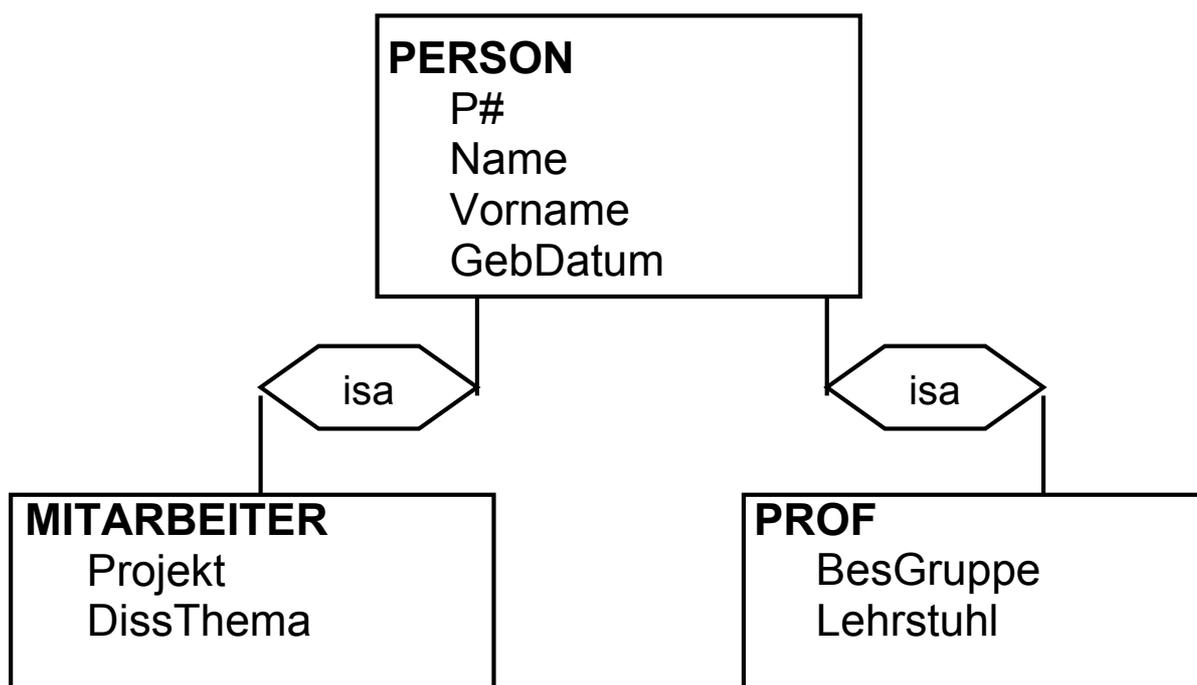
Erweitere A exoder B
exoder fasse A, B zu neuer Entität zusammen,
z.B. bei



spart Platz, einfachere Verarbeitung!

15

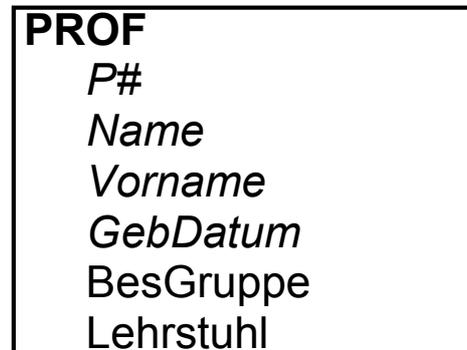
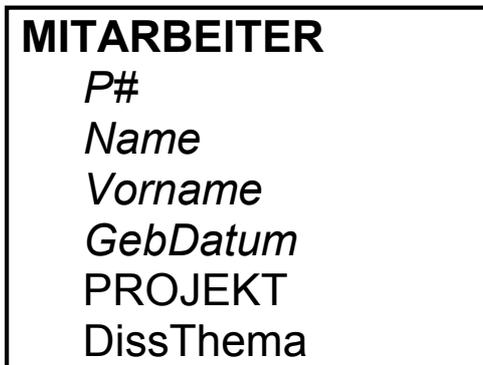
Abbildungen von ISA



16

Variante 1: Abstrakte Klasse Person, ohne entsprechende Relation

isa dient zur Strukturvererbung, Instanzen von PERSON durch Ausdruck berechnet

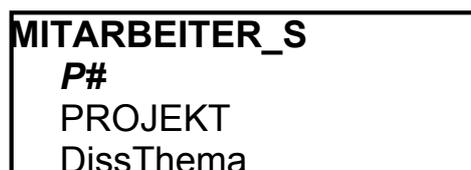
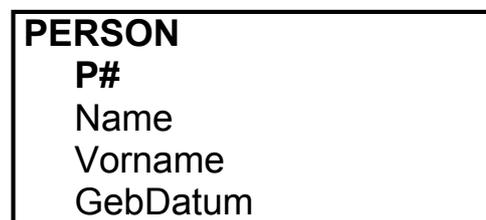


PERSON := π *P#, Name, Vorname, GebDatum* (**MITARBEITER**)
union
 π *P#, Name, Vorname, GebDatum* (**PROF**)

17

Variante 2: Konkrete Klasse Person mit Faktorisierung

alle Instanzen von **MITARBEITER** und **PROF** in PERSON gespeichert,
MITARBEITER-S und **PROF-S** enthalten nur Zusatzinformationen



MITARBEITER := **PERSON** \bowtie **MITARBEITER_S**
PROF := **PERSON** \bowtie **PROF_S**

18

Variante 3: Konkrete Klasse Personen unbekannter Art

MITARBEITER
<i>P#</i>
<i>Name</i>
<i>Vorname</i>
<i>GebDatum</i>
PROJEKT
DissThema

PERSON_U
<i>P#</i>
<i>Name</i>
<i>Vorname</i>
<i>GebDatum</i>

PROF
<i>P#</i>
<i>Name</i>
<i>Vorname</i>
<i>GebDatum</i>
BesGruppe
Lehrstuhl

PERSON := PERSON_U

union

π P#, Name, Vorname, GebDatum (**MITARBEITER**)

union

π P#, Name, Vorname, GebDatum (**PROF**)

Weitere Varianten??

19

Kap. 4.3 Beispiele für Extensionen

INSTITUTION	Name	Uni	vertritt
	Informatik	TUM	Mayr
	Informatik	Passau	Donner
	FORWISS	FAU	Radig
	SFB	TUM	Bode
	ET	TUM	Antreich

GRUPPE	G-Bez	bei 1	bei 2	hat Chef
	Inf III	Informatik	TUM	Bayer
	B	SFB	TUM	Zenger
	Wibas	FORWISS	FAU	Bayer
	Stabi			Leskien

20

PROF	P-Name	Tel #	Zi #
	Bode		
	Bayer	171	108
	Radig	201	
	Paul	161	
	Mitschang	184	118
	Leskien		

PROJEKT	P-Bez	Beginn	Ende	DM
	B2	1989	1998	
	RTB	1994	1996	
	VD17	1994	2002	
	Diadem	1993	1996	

21

MITARB	H-Name	forscht	betreut
	Reiser	B2	Bayer
	Nippl	B2	Mitschang
	Haddouti	VD17	Bayer
	Höfling	Diadem	Bayer
	Wiesener	VD17	Bayer
leitet	Professor	Projekt	
	Paul	B2	
	Bayer	B2	
	Mitschang	B2	
	Bayer	VD17	
	Leskien	VD17	
	Bayer	RTB	

22

Kap. 4.4 Begriffsvergleich

Relationales Modell	Algorithmische Sprachen
Domäne, domain	Art, Sorte
domain name	Art-Indikation
Tupel-Art	struct-Art
Tupel	struct-Objekt, Record
Attribut, Feld, field	Record-Komponente
Field name	Komponenten-Bez.
Column name	Komponenten Bez.
Tupel-Variable	Record Variable
Relation	Menge von Records
Relations-Variable	Var. für Menge von Records
Anfrage, Query	Ausdruck
View	Funktionsdefinition
View-Auswertung	Funktionsaufruf
Query-Optimierung	Ausdrucks-Transformationen