

Kap. 2 Entity/Relationship Modell

Kap. 2.1. Begriff der Entity

siehe: Chen: The Entity-Relationship Model: Toward a unified view of data. ACM TODS, 1,1, März 76, S. 9-36

Grundidee:

- Entities (Objekte)
- Beziehungen zwischen Entities

Entity –Mengen

(Objekt-Arten)

Personen

Autos

Mütter

Kinder

Babynahrung

Meistens Substantiva !



Beziehungen

fährt

verkauft

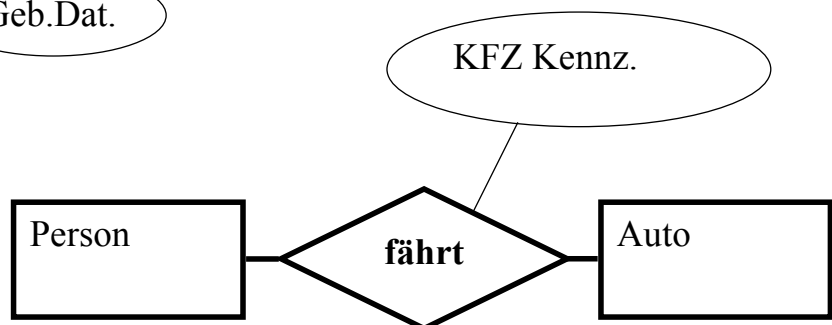
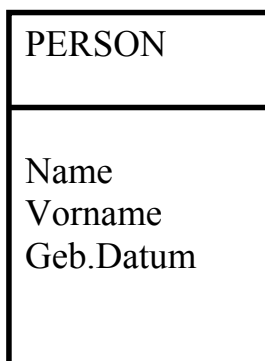
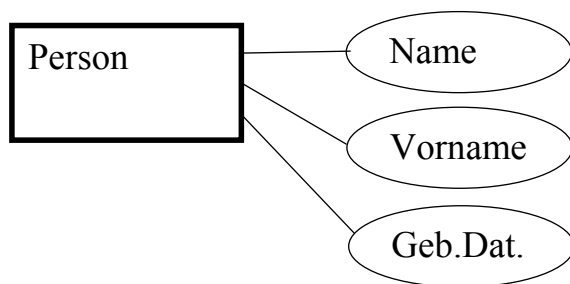
füttert

meistens Verben !



1

Attribute von Entity Mengen und Beziehungen



2

Def. 2.1.1: eine *Entity* ist ein eigenständiges Objekt (Individuum) mit Namen, Attributen und Artzugehörigkeit, das außerdem Beziehungen zu Entities haben kann.

Wesentliche Aspekte:

Identifikation: \Rightarrow Identifikator

OID: intern, extern, von Benutzer definiert, inhärent?

Schlüssel: Kombination von identifizierenden Eigenschaften, Attributen. Minimalität? Schlüsselkandidat? Primär-, Sekundärschlüssel
invariant über Lebensdauer
eindeutig über E-menge oder DB?
wiederverwendbar?

3

Klassifikation: \Rightarrow Entity-Menge E_i

$e \in E_i$?

durch Prädikate: $e_i - E_i (e)$

durch Zuordnung, Auflistung, d.h. per Definition

\Rightarrow Attribute mit

\Rightarrow Wertemenge des Attributes
(Art, Sorte, Typ, value-set)

Beziehungen: \Rightarrow Relationships

explizit festgelegt, aufgelistet, n-stellig,
meistens 2-stellig

siehe auch folgende Abbildungen!

4

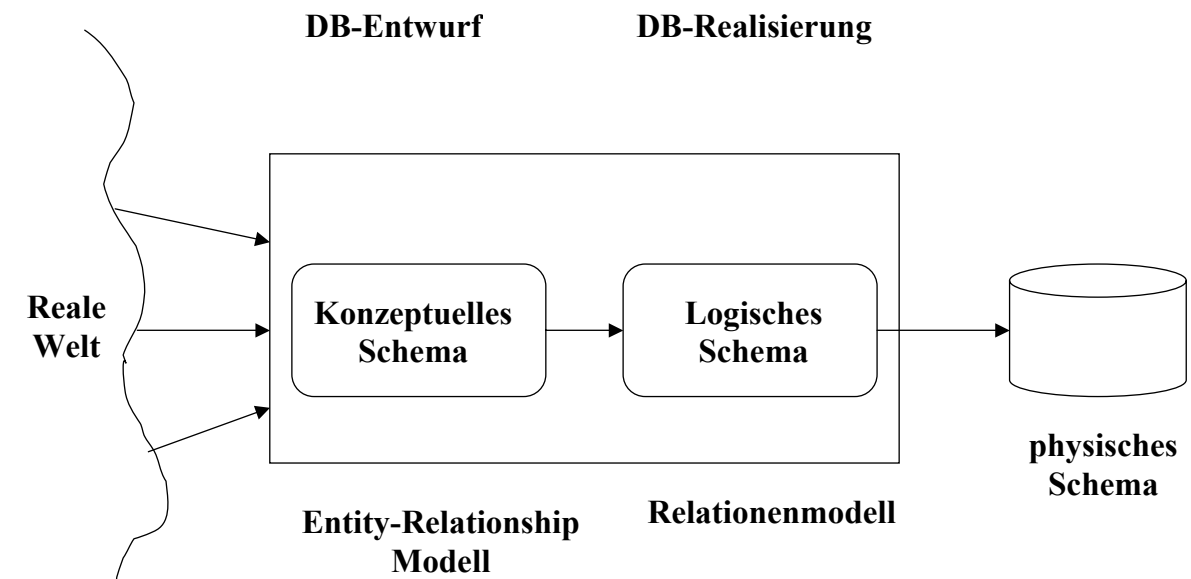


Abb. 2.1.3: Datenbankentwurf mit dem ER-Modell

5

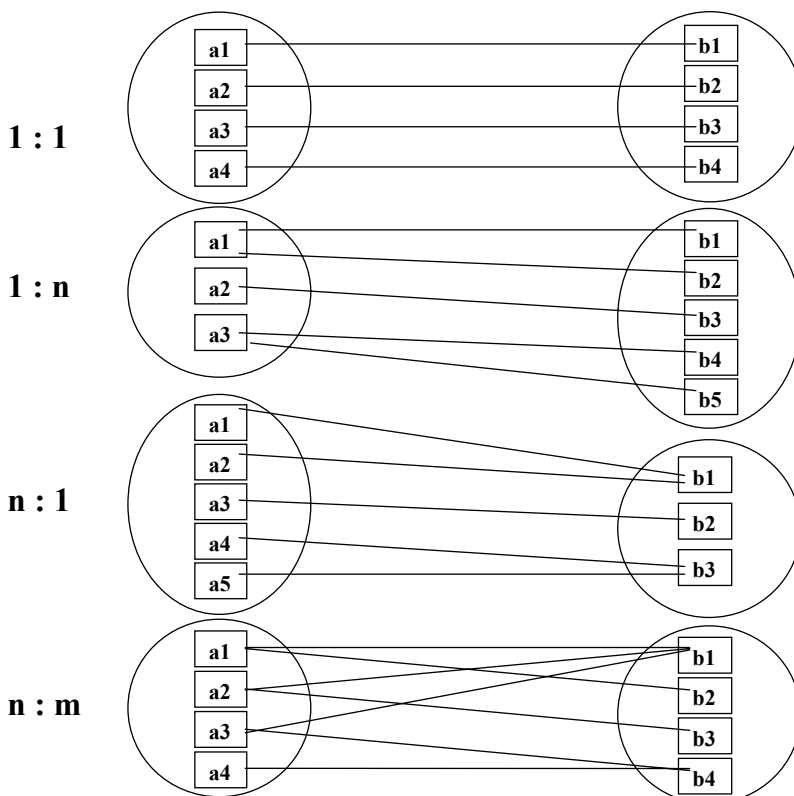


Abb. 2.1.4 Funktionalität von Relationship

6

Kap. 2.2 Notationen

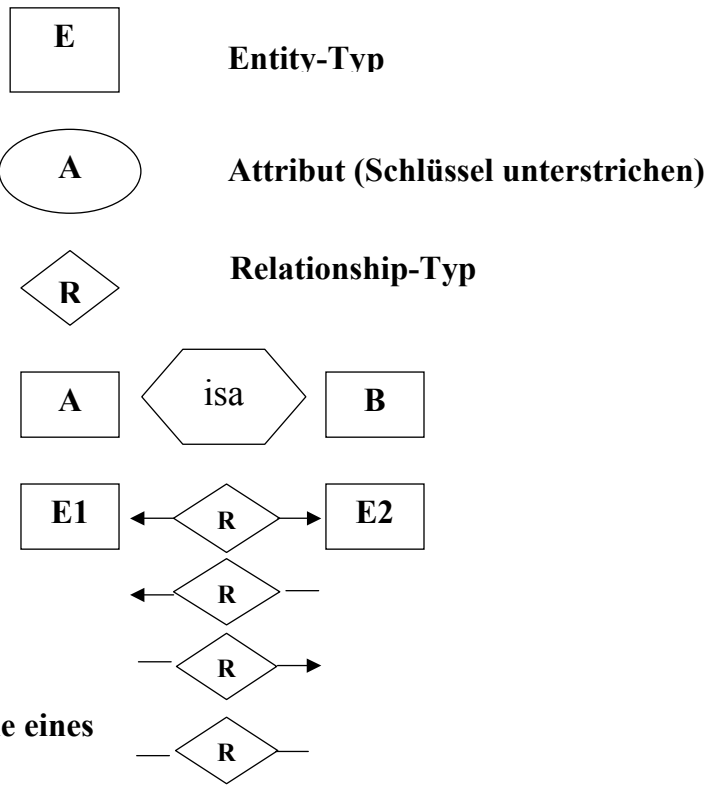
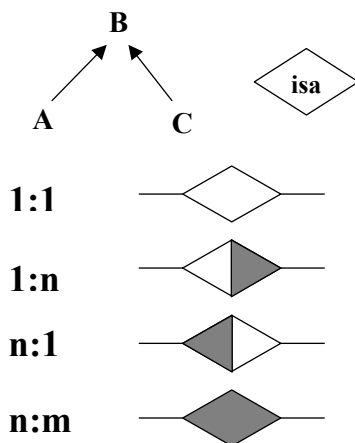
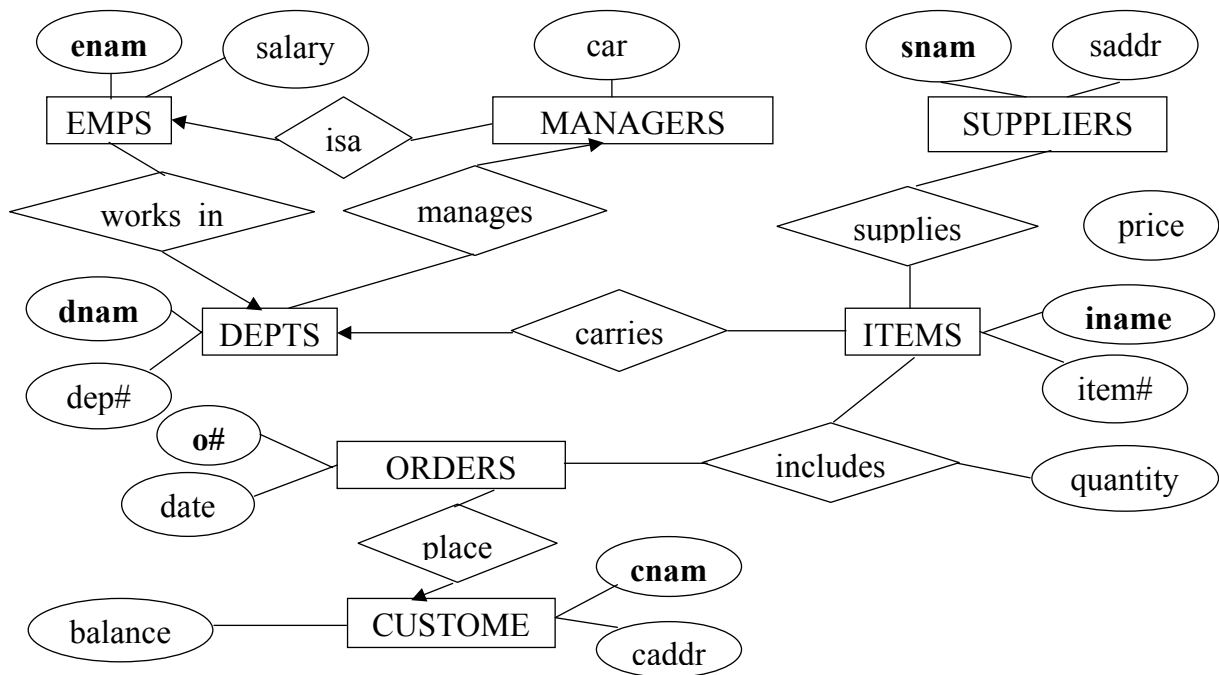


Abb. 2.2.1 Grundbestandteile eines ER-Diagramms

Kießling Notation





nur 2-stellige Beziehungen, typisch für Praxis, siehe später

Abb. 2.2.2: ER-Diagramm für Yuppie Valley Culinary Boutique (YVCB)

9

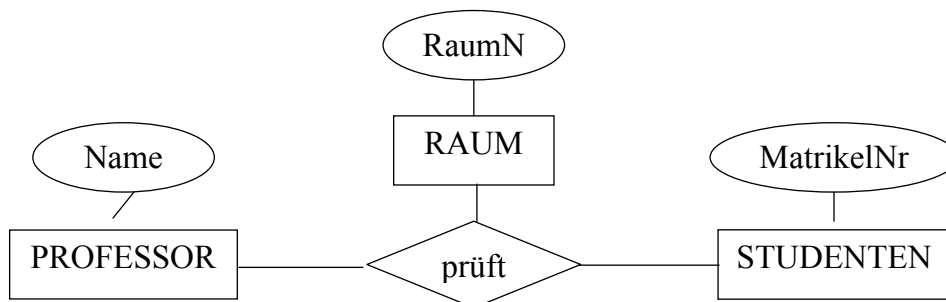
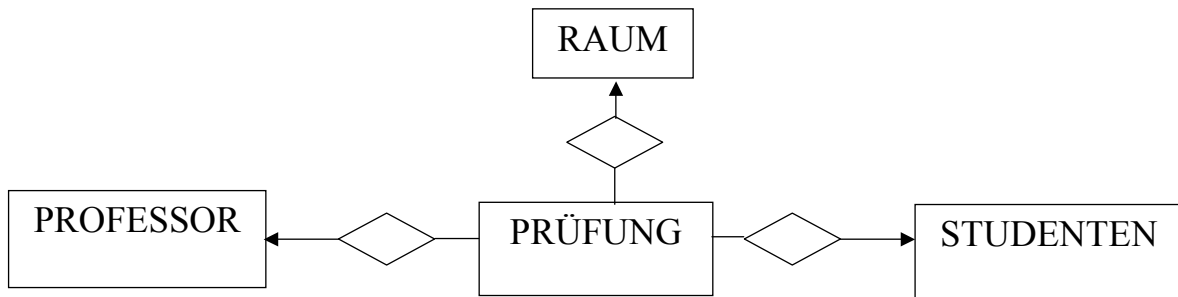


Abb. 2.2.3: Nicht durch mehrere zweistellige Relationships darstellbare Beziehung

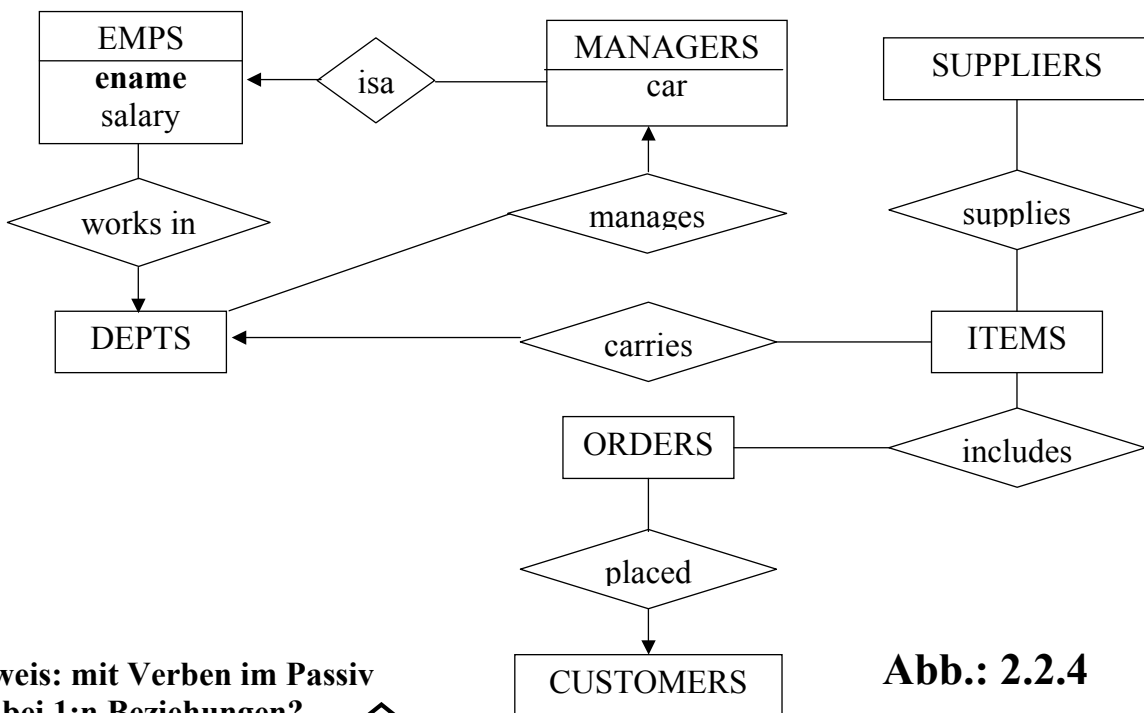
10



Zerlegung in 3 n :1 Beziehungen

11

Schema aus Abb. 2.2.2 ohne äußere Attributsymbole



Hinweis: mit Verben im Passiv
bei 1:n Beziehungen?



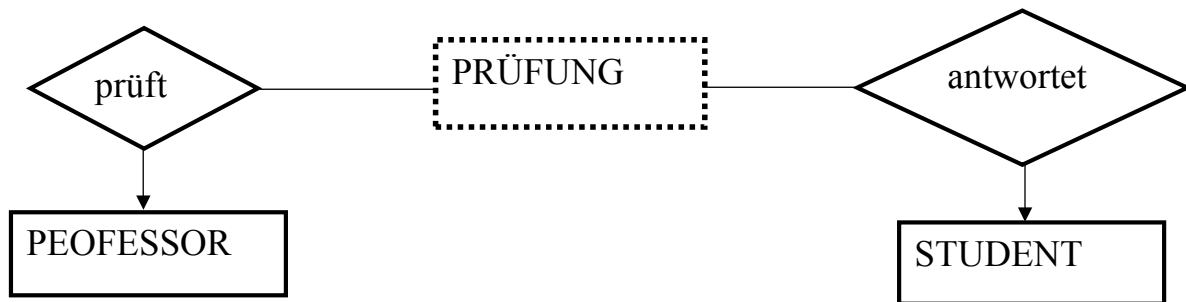
Abb.: 2.2.4

12

Kap. 2.3 Beziehungen als Entities, schwache Entitäten

Def. 2.3.1: Entity E_1 heißt *existenzabhängig* von E_2 falls es eine $n : 1$ Relationship R gibt mit $R \subseteq E_1 \times E_2$, so daß $e_1 \in E_1$ nur existieren kann, wenn $\exists e_2 \in E_2 : (e_1, e_2) \in R$

E_1 heißt auch „**schwache Entity**“



13

Höhere (schwache) Entities: durch Zusatzattribute bei Relationships

gekennzeichnet durch „Substantivierung von Verben“! z.B.

prüft → PRÜFUNG mit
Zeitangabe
Ortsangabe
Note etc.

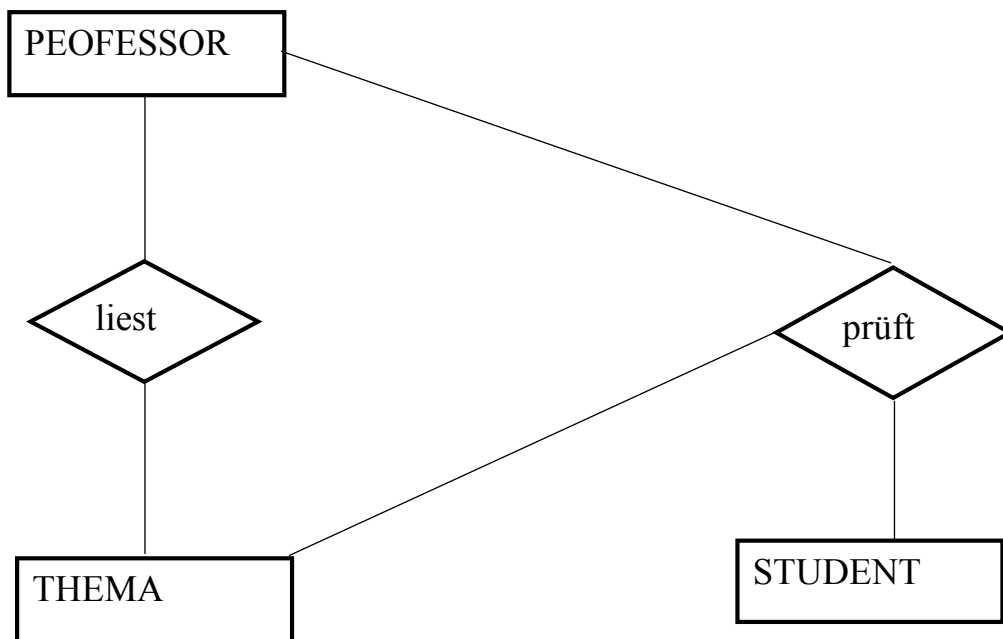
heiratet → Ehe (mehrfach)

Beziehungen zwischen Beziehungen möglich, z.B.

der Ehe mit Kennedy
folgt die Ehe mit Onassis

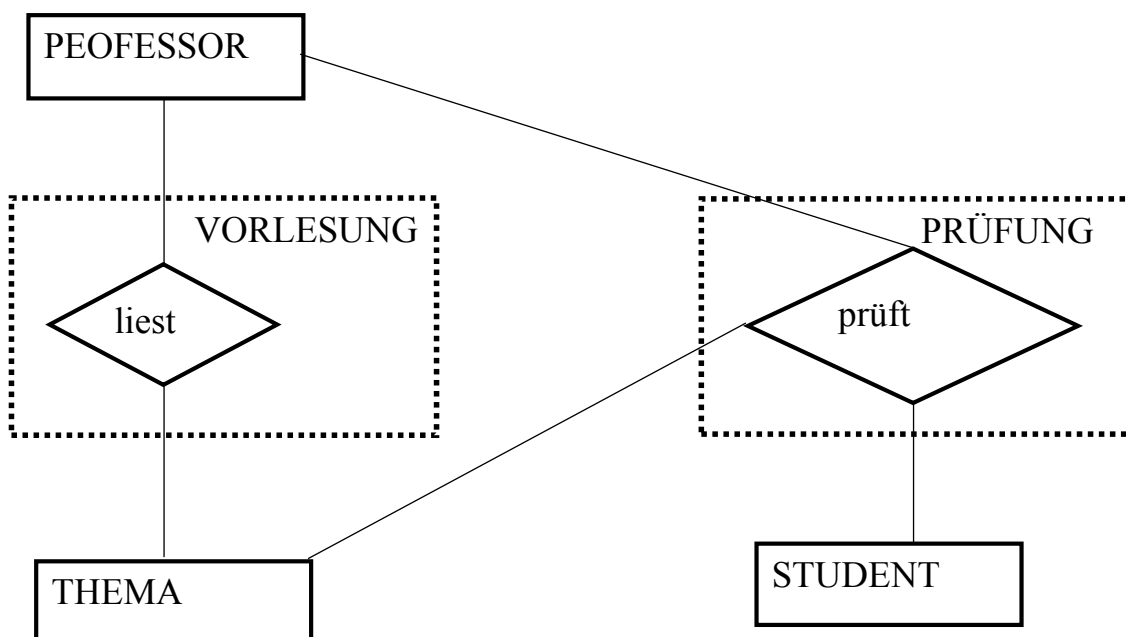
14

Entstehung schwacher Entitäten



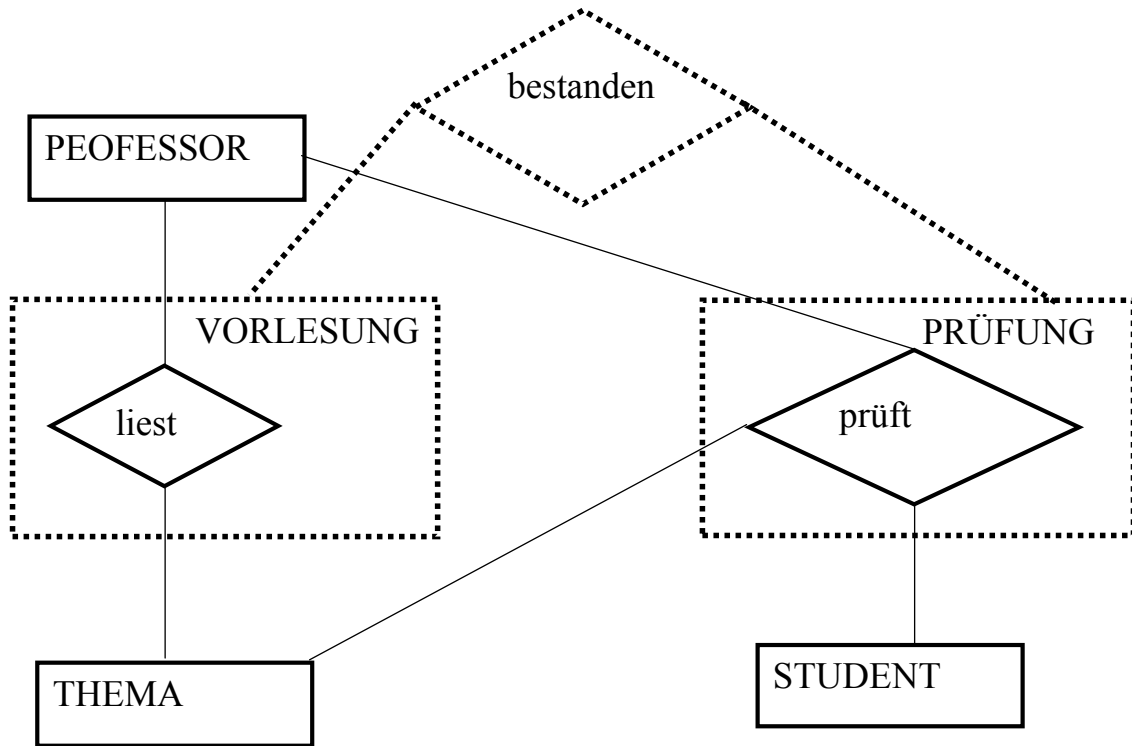
15

Schwache Entitäten 1. Stufe



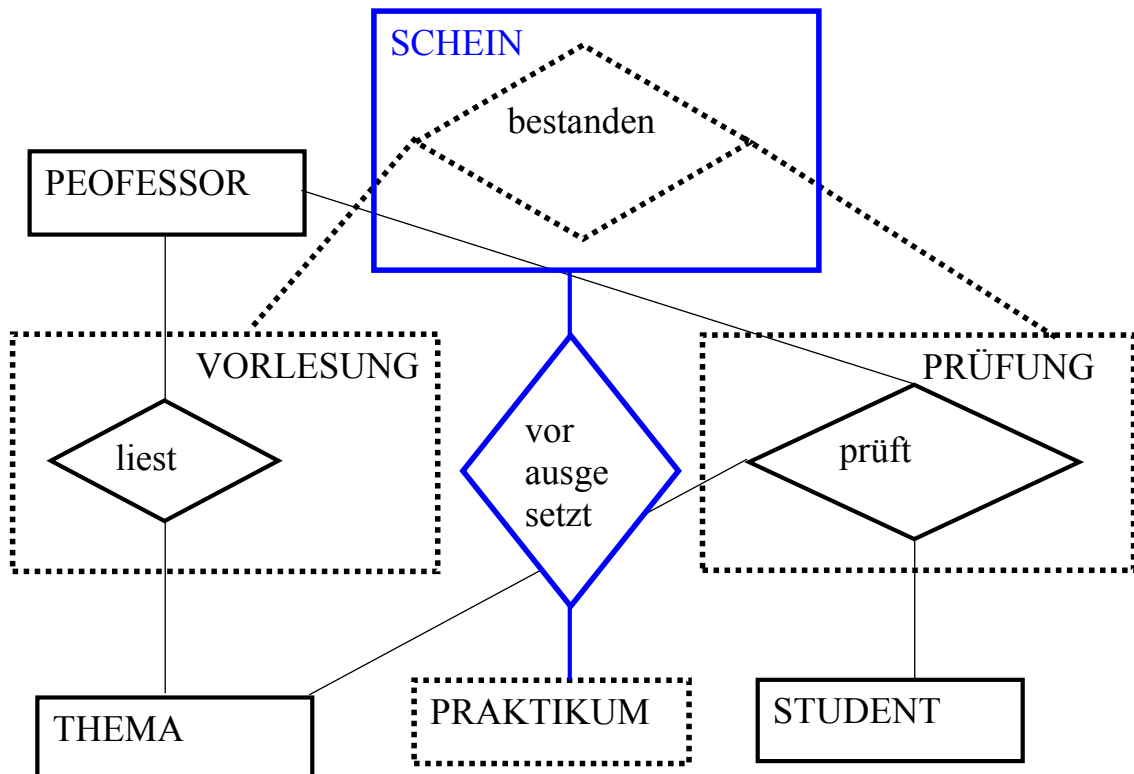
16

Beziehungen zwischen schwachen Entitäten 1. Stufe



17

Schwache Entität 2. Stufe und Beziehungen



18

Kap. 2.4 Instanzen

2.4.1 Instanzen von Entities

EMPS mit Attributen:

e # integer $\geq 10\ 000$
ename string $\neq \varepsilon$
salary integer ≥ 0

Instanzen in Langnotation: z.B. mit XML Tags?

((e #: 10123), (ename: „Otto“), (salary: 75.000)) \in EMPS

((e #: 10124), (ename: „Julia“), (salary: 84.100)) \in EMPS

Kurznotation: mit fester Struktur!

EMPS: 10123 „Otto“ 75.000
10124 „Julia“ 84.100

19

Unterscheide!

Extension

Auflistung

Tabelle

Menge von Instanzen

Intension

Art

Typ

Klasse

Intension: Menge aller möglichen d.h. strukturell korrekten Instanzen, meistens unendlich

Extension: Menge der in DB konkret repräsentierten (i.e. meist gespeicherten) Instanzen, immer endlich.

Hinweis: Wiederverwendung einer Entity-Def. für mehrere, unterschiedlich benannte Extensionen in E/R-Modell nicht vorgesehen

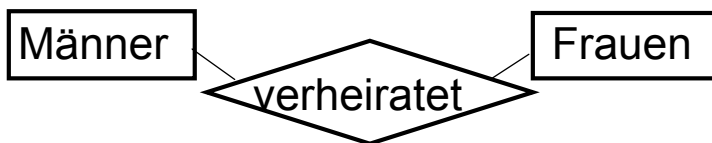
20

2.4.2 Instanzen von Beziehungen

Unterscheide wieder Extension, Intension
z.B. verheiratet
managed

Relationship-Instanzen

Paare, Tripel, ..., n-Tupel von Entities
erweitert um Werte für Zusatzattribute, z.B.



((Mann (...“Otto“ ...)), (Frau (...“Maria“ ...))
(H-Jahr 1983), (S-Jahr ?))

21

⇒ Redundanz, verwende Identifikatoren
10123 für (Mann (...“Otto“...))
10124 für (Frau (...“Maria“...))

als Platzhalter für ganze Entity

(10123, 10124, 1983, ?)

von Schema her bekannt: 10123 identifiziert Mann
10124 identifiziert Frau

Hinweis: bei E/R Eindeutigkeit von Identifikator
nur innerhalb einer E-Menge

Bei oo-DBS gilt OID systemweit: Universal-Identifikator!

22

Kap. 2.5 Operationen

Hinweis: nur prinzipielle Überlegungen, keine existierende Sprache!

2.5.1 Schema-Operationen

declare-E : Entity mit Attributen und Typ

drop-E

modify-E : Struktur ändern,
Attribut hinzufügen,
Attribut löschen, Typ ändern

declare-R : Relationship mit beteiligten Entities
und Zusatzattributen

drop-R

modify-R

23

Hinweis:

Einschränkung der Intension durch Integritätsbedingung, z.B.

„Manager verdienen ≥ 100 TDM“

„Hiwis arbeiten ≤ 19 h/Woche“

⇒ Intension nicht einfach Cartesisches Produkt der Wertemengen (Domänen) der Attribute

24

2.5.2 Instanz-Operationen

Datenmanipulation

create (insert) z.B.
EMPS (10123, „Otto“, 75.000)

delete (remove)

update z.B.
EMPS (10123).salary: = 77.200

25

Query-Sprache deklarativ!!!

Typ A: einfach, nur 1 E-Menge:

„finde alle Instanzen aus Extension von E mit bestimmten Eigenschaften“

$$\{x : x \in E \wedge P(x)\}$$

$P(x)$ ist offene Formel, Prädikat mit freier Variable x

Def: Antwortmenge sind alle Instantiierungen der freien Variablen x , so daß $P(x)$ **true** ist.

⇒ Selektionen, Restriktionen

Notation: $P_E(x)$, um Individuenbereich von x zu spezifizieren

26

Typ B: komplex, mehrere E-Mengen involviert:

„finde alle Instanzen aus E1, E2, i.a. aus E1 ... En,
die in Beziehung T zueinander stehen
und bestimmte Eigenschaften haben“

$$\{(x,y) : P_{E1}(x) \wedge Q_{E2}(y) \wedge T(x,y)\}$$

⇒ führt bei relationalen DBSen zu sog. Join-Operationen

Antwortmenge:

Alle Paare (Tupel) von Instanzen (x,y), die prädikatenlogische Formel erfüllen.

Antwort Präsentation:

a) Paar (tupel) von vollständigen Entities, zu viel!

b) Tupel von Identifikatoren, z.B. (1023, 1079) zu wenig

c) $\{(x.a, x.b, y.c, y.d, y.e) : F(x,y)\}$

⇒ Projektion von a)

2.6 Höhere Stufen von Entities und Beziehungen

Beobachtung: zwischen Beziehungen (als schwache Entities aufgefaßt, können wieder Beziehungen aufgebaut werden, z.B.

Ehen: Ehe \subset Person x Person
I \subset Ehe x Ehe

$i \in I \Leftrightarrow i = (e_1, e_2) \wedge$
 e_1 geschlossen vor $e_2 \wedge$
 e_1, e_2 haben gemeinsamen Partner

flachklopfen? $I \subset E \times E \subset (P \times P) \times (P \times P)$
 $= P \times P \times P \times P$ **Verlust von Semantik!**

29

Komplexere Beispiele

Job-Kandidaten:

Studium: $S \subset$ Person x Univ. x Vorlesungen x... x Fachabschluß

Firmen: F ...

Kandidaten: $K \subset$ Person x Firma

können über Suchprädikat aufgefunden werden

Beispiel Familie:

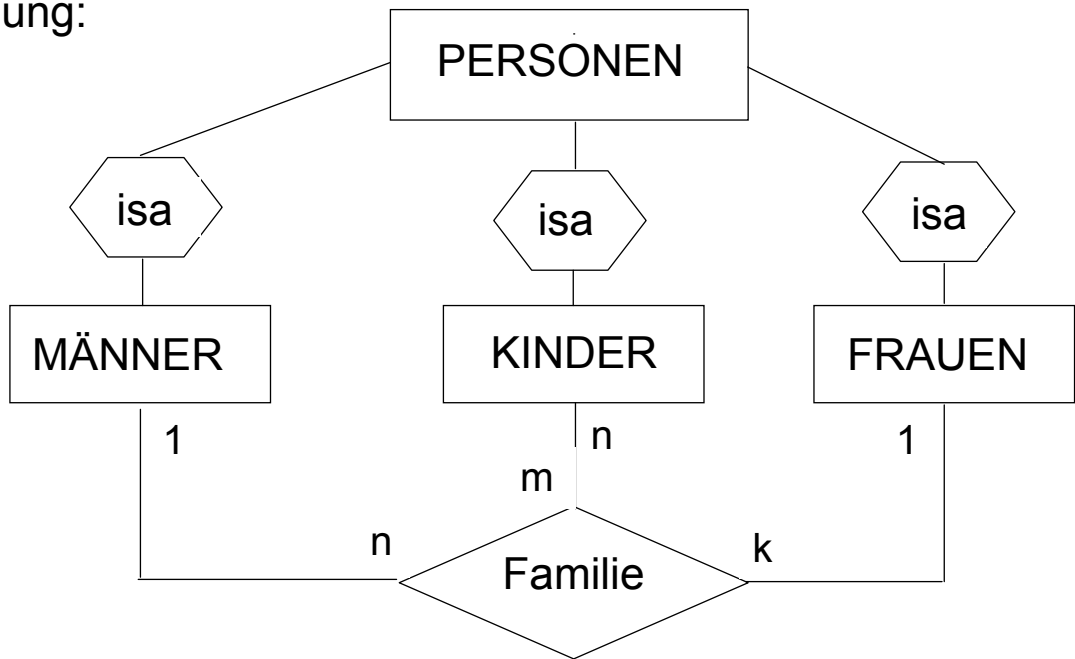
erste Stufe: als Beziehung zwischen Personen, Kleinfamilie

höhere Stufen: als Entity mit weiteren Beziehungen (Familie bekommt Namen, Identifikator), z. B. Freunde, Feinde, Großfamilie

30

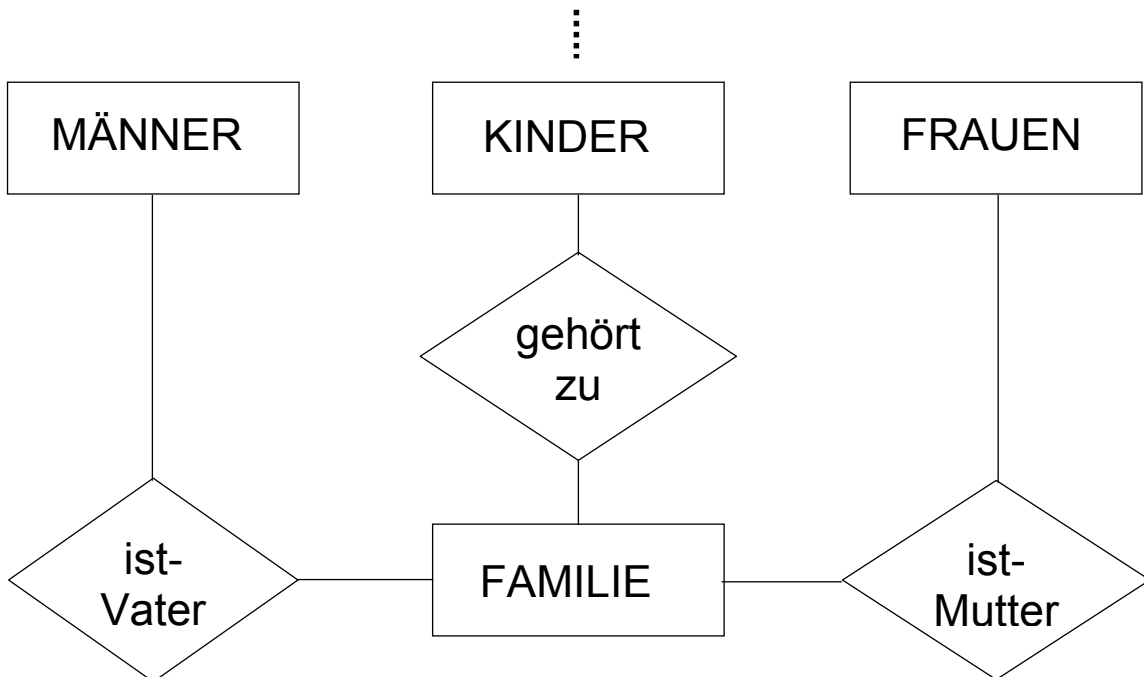
Beispiel Familie:

als Beziehung:



31

als Entity mit weiteren Beziehungen
(Familie bekommt Namen, Indentifikator)

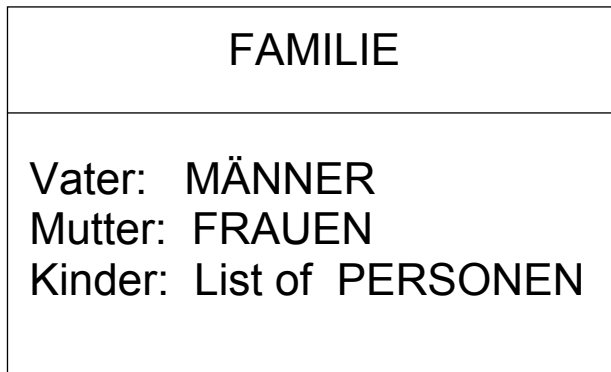


32

als komplexe Entity:

Höhere Stufen von Beziehungen,
z.B.

Nachbarn
Mafia-Koop



Kap.2.7 Entities als Beziehungen

(Umkehrung der Betrachtungsweise)

Mensch M (Geb.Datum, Adresse, Beruf)

G A B

$M \subset G \times A \times B$

d.h. Mensch als Relationship über einfacheren Wertmengen

Problem: eindeutige, d.h. identifizierende Merkmale?

Frage: sind G, A, B einfache, atomare Wertmengen oder selbst wieder Entities mit Beziehungen?

⇒ **freie Modellierungsentscheidung, abhängig von Anwendungen!!!**

G, A, B als Entities:

Datum: Info-System über Geschichte 333

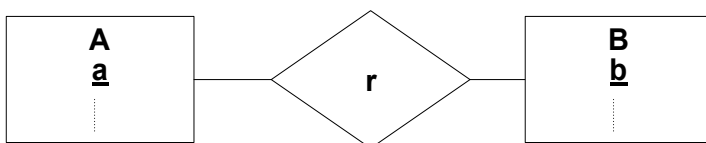
Beruf: Personalsystem

Adresse: polizeiliches Fahndungssystem

⇒ formal keine Unterscheidung zwischen Entities, Beziehungen.
Anwendung entscheidet!!

35

Identifikation von Relationships



(a, b)

zusammengesetzter Schlüssel (a, b)
aus Fremdschlüsseln (a bzw. b)

Relationship (a, b) kann nur existieren, wenn e_a und e_b existieren

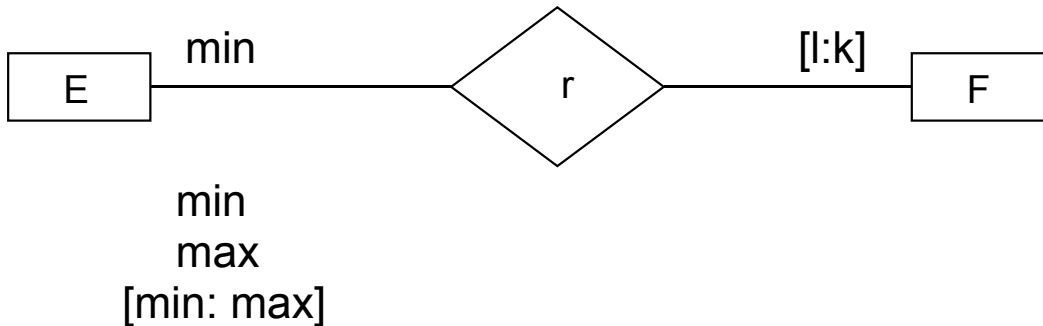
⇒ referentielle Integrität

⇒ schwache Entitäten aus Relationships, wenn diese als Entitäten aufgefaßt werden, haben Paare als Identifikatoren (Schlüssel)

36

Kap. 2.8 Erweiterungen, Überblick

Kardinalität von Beziehungen



$$\forall e \in E : \min \leq |\{(e,y):(e,y) \in r \wedge y \in F\}| \leq \max$$

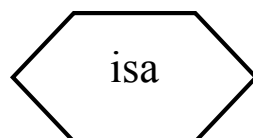
37

Zusammengesetzte Entities:

- ~ Beziehung als Entity aufgefaßt
- ⇒ durch Iteration (Rekursion)
beliebig viele Stufen
- ⇒ part-of Beziehung in oo-DBS bzw.
component-of : po co

ISA-Hierarchien

Generalisierung:



führt zu Klassen-Bäumen bzw.
Klassen-Hierarchien

38

Spezialisierung:

Inversion der Generalisierung

- ⇒ Vererbung der Struktur
- ⇒ Vererbung von Werten
- ⇒ Vererbung von Invarianten (Integritätsbed.)
- ⇒ einfache oder multiple Vererbung
- ⇒ Vererbung von Operationen
Methoden

Aggregation, Mengenbildung

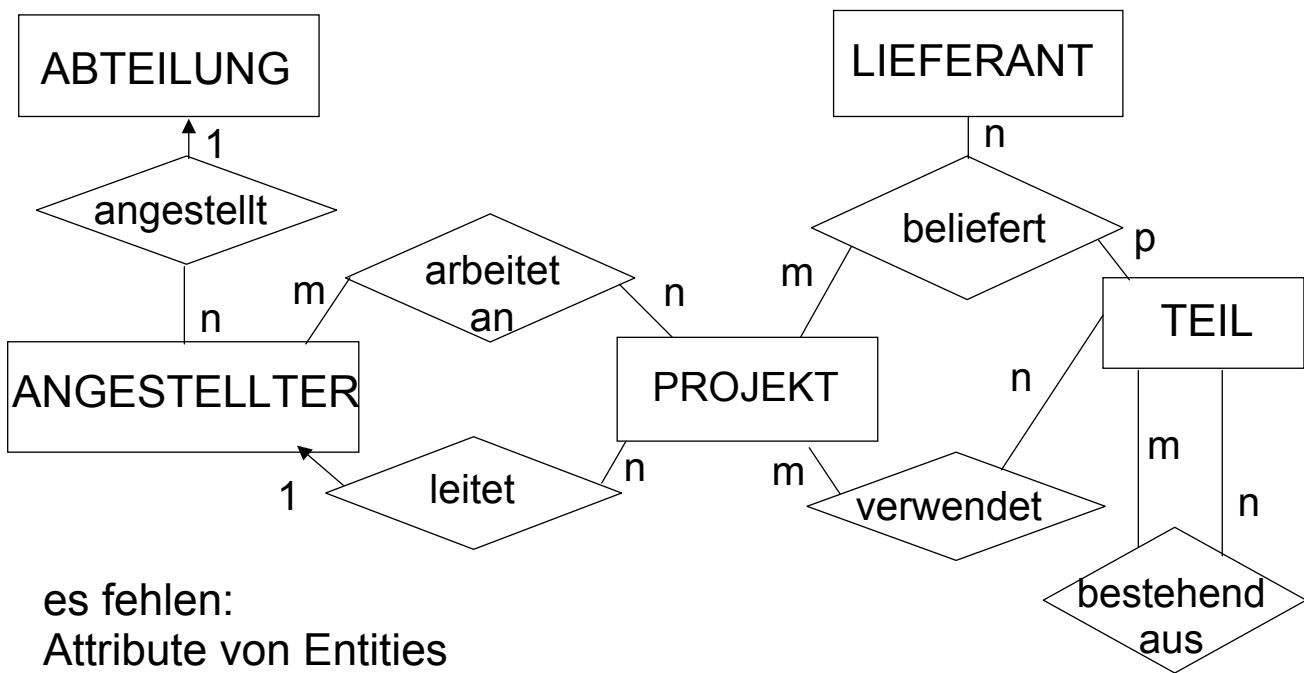
E-Mengen sind homogen, gleichstrukturierte Objekte

Aggregationen sind inhomogen,
z.B. Fußballmannschaft
Schiffsbesatzung

siehe späteres Kapitel über UML-Modellierung

und Spezialvorlesungen
oo-Datenbanksysteme
Datenmodellierung

Arbeitsbeispiel Modell Firma



es fehlen:
Attribute von Entities
Attribute von Relationships