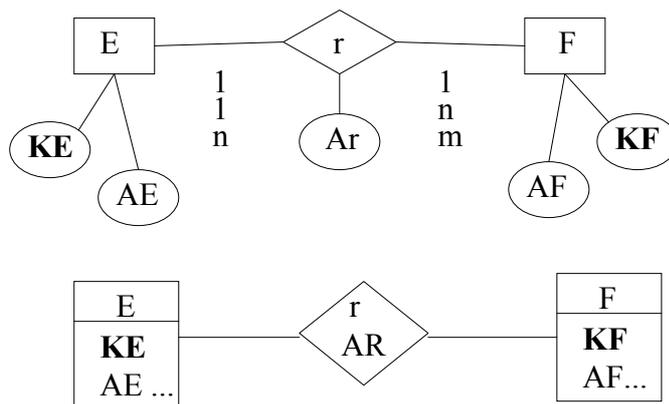


Kap. 1.4 E/R und oo-Modelle

Kap. 1.4.1 Erweiterte E/R Modelle



1

Standard Abbildungen

1 : [0:1] RelErF
 KE, AE, ..., Ar, KF, AF, ...
 mit Schlüssel **KE**

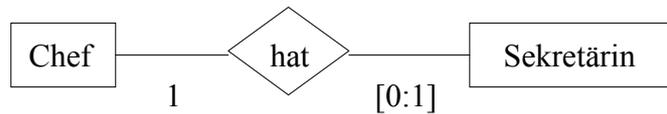
[0:1] : [0:n] RelE
 KE, AE, ...
 RelFr
 KF, AF, ..., Ar, **KE**
 mit Fremdschlüssel **KE**

m : n RelE
 KE, AE, ...
 Relr
 KE, **FE**, AE, ..., AF, ...
 ReF
 KF, AF, ...

2

Abweichung vom Standard:

z.B. wenn Entitäten oder Relationships häufig eigenständig benötigt werden,



2 Relationen, um Projektionen zu vermeiden:
Abwägung zwischen Projektionen und Joins

Beispiel: Folie 2-10 aus Mitschang

Verfeinerung ... Folie 2-11 Mitschang

3

Weitere Verfeinerungen:

Kardinalitätseinschränkungen
mehr Semantik durch Integritätsbedingungen,
z.B. Fremdschlüssel, Typisierung: Tel#, M#, Datum, ...
Prüfungsnote, ...

siehe Vorlesungen DBS I, DBS II, Datenmodellierung, ...

4

Kap. 1.4.2 oo-Entwurf und Hierarchien

Kap.1.4.2.1 Konzeptuelle oo Modelle

A. Abstraktion und Klassifikation ↔ Spezialisierung

isa, io

B. Mengen-Hierarchie (Verband)

ss eo

C. Komponenten-Hierarchie für Komplexe Systeme

co po
component of part of

5

Abstraktion und Klassifikation

- Klassifikations-Hierarchien:
- Personen in Firma
- Lebewesen (Lynnée)
- Dokumente in Bibliothek → Signatur
- ACM - Klassifikation
- Geräte
- Produkte am Markt (GfK Beispiel)
- Fahrzeuge
- ...

6

Klassifikations-Verfahren

Wie findet man Klassifikationen?
„Deutsche Gesellschaft für Klassifikation“

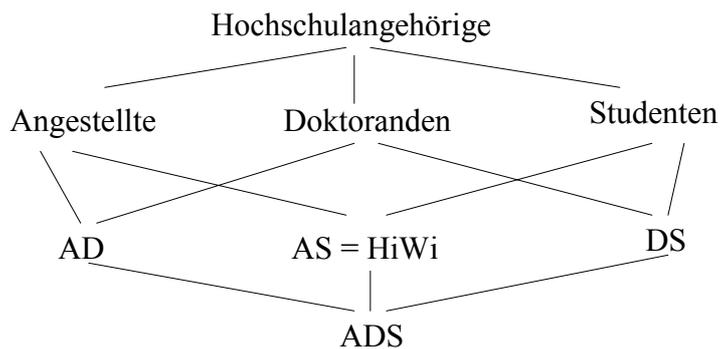
- Lebensraum (Tiere Australiens)
- Verhalten
- Benutzung (Küchengeräte)
- Energieart
- Farbe (brown, white)
- Abstammung (Genom)

gemeinsame Struktur
(Attribute, Properties, Eigenschaften)

Beispiel: Mitschang S. 2-23

7

Alternativ-Lösung:



Semantische Gemeinsamkeit ist erkennbar an syntaktischer
Gemeinsamkeit, das sind die Attribute

Hinweis: durch Spezialisierung werden Attribute der
Vorgänger geerbt.

8

Struktur-Verband:

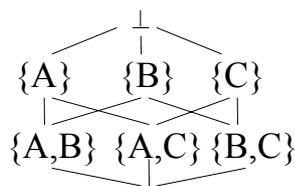
Hinweis: zu gegebenem Strukturverband gibt es viele Klassenhierarchien.

Ref.: Der Strukturverband für eine Menge von Attributen $A = \{A1, A2, \dots AC\}$ ist die Potenzmenge 2^A geordnet nach \subseteq

Beispiel: $A = \{A, B, C\}$

Strukturverband:

$2^A =$



$\{A,B,C\} = T = \text{top, enthält alle Attribute}$

Hinweis: Jede Instanz kann aufgrund ihrer definierten Attribute eindeutig einer Klasse zugeordnet werden

Klassen-Hierarchien

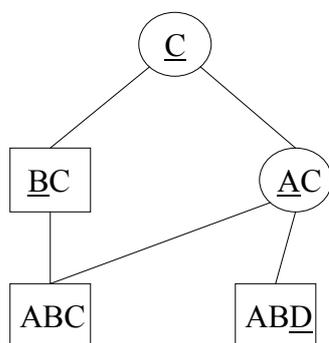
Einige Klassen mit ihren Attributen werden durch die Anwendung vorgegeben, z.B. ABC, ACD, BC

Was ist eine geeignete Klassenhierarchie?

Def: *Owner* eines Attributs ist diejenige Klasse, in der ein Attribut neu definiert wird. Die anderen Attribute werden von Oberklassen *geerbt*.

11

Hierarchie 1:

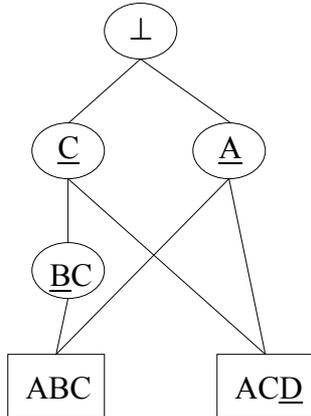


Beispiel: AC ist Owner von Attribut A

5 Klassen
11 Attribute
C mehrfach vererbt

12

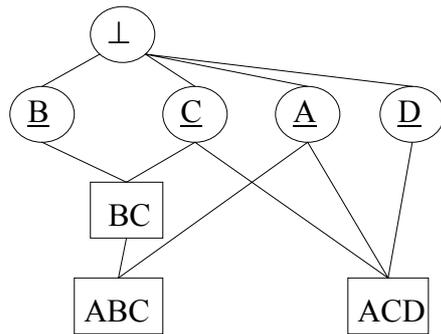
Hierarchie 2



6 Klassen
10 Attribute
Keine mehrfache Vererbung

13

Hierarchie 3



8 Klassen
12 Attribute
Keine mehrfache Vererbung

Frage: weitere Beispiele von Hierarchien?

14

Problem: Durch Anwendung sei eine Menge K von Klassen vorgegeben, gefordert:

$$K = \{K_1, K_2, \dots, K_m\}$$

Was ist eine gute Klassenhierarchie für K unter Verwendung der in den K_i vorkommenden Attributen?

Normalformen, Qualitätsmaß?

15

Normalformen für Klassenhierarchien

KNF1: Jedes Attribut hat genau 1 Owner
(trivialer Algorithmus)

KNF2: KNF1 plus minimale Anzahl von Klassen

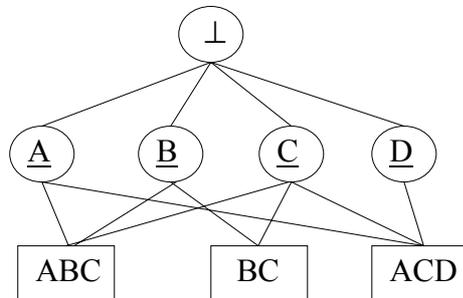
KNF3: KNF2 plus minimale Anzahl von Attributen

KNF4: KNF3 plus minimale Höhe

KNF5: KNF3 plus minimale gewichtete Pfadlängensumme zu den von der Anwendung geforderten Klassen

16

Hinweis zu KNF4: KNF1 plus minimale Höhe würde liefern:



7 Klassen
12 Attribute
keine mehrfache Vererbung
12 Kanten ~ Joins

Lösung entspricht isolierter Attributdefinition,
Attribute als „first class citizens“?

Kantenminimierung als zusätzliches Qualitätsmerkmal?

17

Algorithmus für KNF2:

Gegeben: Anwendungsklassen $K = \{K_1, K_2, \dots, K_m\}$

while $|K| > 1$ **do**

Berechne paarweise Schnitte $V_{ij} = K_i \cap K_j$

{die V_{ij} sind die vererbenden Superklassen}

$V = \{V_{ij} \mid i, j = 1, 2, \dots, m\}$

$B = K - V$ sind Knoten der untersten und dann jeweils nächsthöheren Hierarchieebene.

Falls $K_i \in B$ dann erbt K_i die Attribute von V_{ij}

$K = V$

od

18

KNF2 Beispiel

$K = \{ABC, BC, ACD\}$

$V = \{BC, AC, C\}$

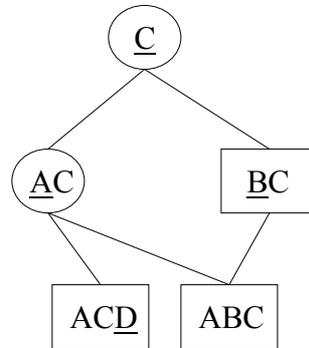
$K-V = \{ABC, ACD\}$

$K = \{BC, AC, C\}$

$V = \{C\}$

$K-V = \{AC, BC\}$

$K = \{C\}$



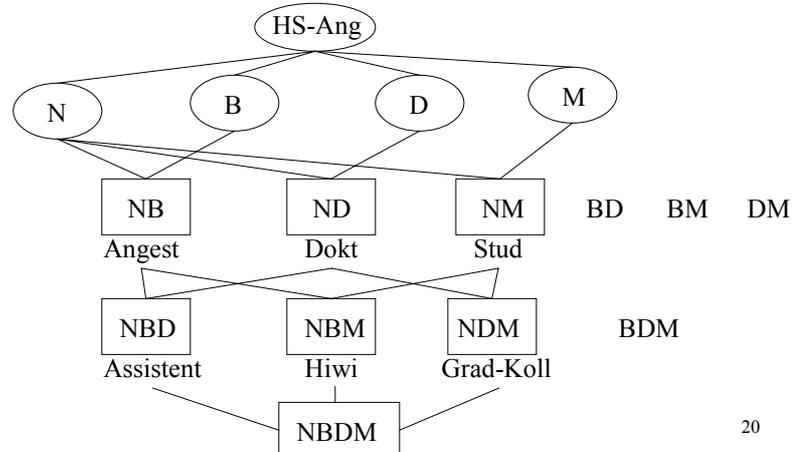
19

Nochmals Hochschul-Angehörige

Attribute: Name, Besoldungsgruppe, Diss-Thema, Matrikel#

N B D M

Vollständiger Klassenverband:

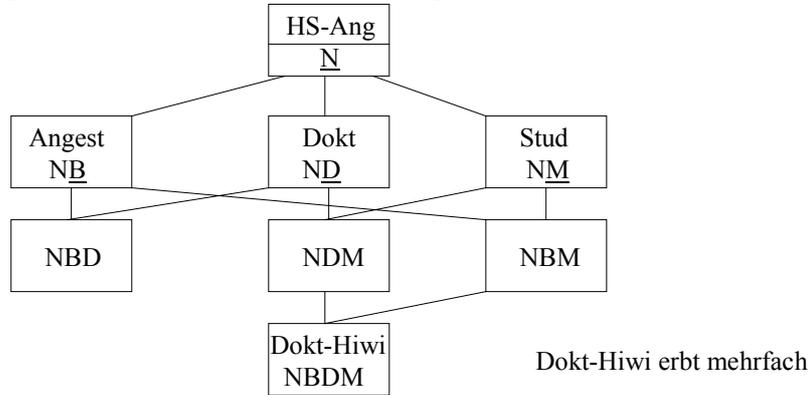


20

Hierarchie für HS-Angehörige

Da Attribut Name allen Personen gemeinsam ist, auf oberste Ebene hochziehen.

Die privaten Attribute bei Owner möglichst weit unten definieren



Übung: Ergänze Dokt und Stud durch Attribut 'Abiturnote' und passe Hierarchie an.
(Evolutionsproblem!)

21

KNF2 Algorithmus für HS-Angehörige

$K = \{NBDM, NBD, NDM, NBM, NB, ND, NM\}$

$V = \{NBD, NDM, NBM, NB, ND, NM, N\}$

$K-V = \{NBDM\}$

$K = \{NBD, NDM, NBM, NB, ND, NM, N\}$

$V = \{NB, ND, NM, N\}$

$K-V = \{NBD, NDM, NBM\}$

$K = \{NB, ND, NM, N\}$

$V = \{N\}$

$K-V = \{NB, ND, NM\}$

$K = \{N\}$

liefert vorherige Hierarchie

22

Einbeziehung von Methoden

Def.: $K_f = \{F1, F2, \dots, F_n\}$

sei die Menge der Attribute, die zur Ausführung der Methode f benötigt werden.

Behandle K_f genau wie eine der vorgegebenen Klassen für den Aufbau einer Klassenhierarchie.

Analog: Owner von f

Vererbung von f

23

Kap. 1.4.2.2 Logische Modelle für oo Hierarchien

Modell V: vollständige Objekte

Modell Z: zerlegte Objekte

Modell U: universelles Modell

Modell T: Tripel Modell

Modell R: redundantes Modell

Modell M: Mischung von Modellen

24

Modell V: genau 1 Relation pro vorgegebener Klasse,
z. B. Dokt mit allen zugehörigen Attributen
(private plus geerbte) plus OID

relation Dokt
(OID : integer,
N : string,
D : string) **key is** OID

relation Hiwi
(OID : integer,
N : string,
B : string,
M : integer) **key is** OID

25

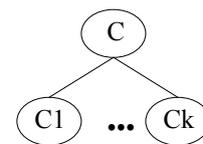
Speicherung einer Instanz bei speziellster Klasse, z.B. nur bei
Hiwi, nicht bei Stud.

Instanzen von C durch view ALLC

```

define view ALLC as
  select * from C           union
  select  $\pi_c$  from ALLC1   union
  ...
  select  $\pi_c$  from ALLCk

```



wobei π_c = Menge der Attribute von C

```

define view ALLStud as
  select * from Stud union
  select OID, N, M from ALLHiwi union
  select OID, N, M from ALLGrad-Koll

```

Hinweis: keine Duplikate in Projektion wegen OID,
d.h. billige Projektionen.

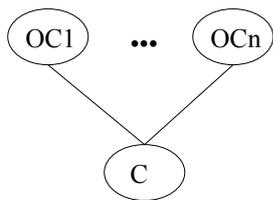
geschachtelte View-Definitionen, Problem für DBS-Optimierer?

26

Modell Z: Speicherung zerlegter Objekte, jedes private Attribut mit seiner Owner-Klasse plus OIDs

Genau 1 Relation pro Klasse in der Hierarchie

Instanzen von C:



```
define view ALLC as
select C.OID, C.A, ...
      ALLOC1. B, ...
      ...
      ALLOCn. D
from C, ALLOC1, ..., ALLOCn
where C.OID = ALLOC1.OID and
      ...
      C.OID = ALLOCn.OID
```

27

Probleme:

- mehrfach vererbte Attribute auch mehrfach im Ergebnis
- viele Joins und geschachtelte Views
- Redundanz bei OIDs
- **insert** Instance führt zu vielen **insert** Tupel

Übungen:

- vollständige view Definition für All Dokt-Hiwi?
- Transaktions-Code für insert Dokt-Hiwi

28

Automatische Konstruktion ohne Views

1. Finde die Menge der Owner Relationen der Attribute, z.B. von N, D, M#, X, aus den Systemtabellen.
2. Alle Owner-Rel. in **from** Teil
3. Alle gewünschten Attribute mit ihren Ownern im **select** Teil
4. Join über alle im Schritt 1 gefundenen Owner Rel. bzgl. OIDs
5. Weitere Attribut-Restriktionen im **where**-Teil

29

Modell U: Universelle Relation:

Speichere alle Objekte in **einer** Relation U, die neben der OID alle in der Hierarchie vorkommenden Attribute enthält.

Die für ein Objekt nicht vorkommenden Attribute werden mit einem **null** value (u. komprimiert) gespeichert.

Die Objekte, die Attribute A, D, E haben, bekommt man mit:

```
select A, D, E from U
      where A  $\neq$  null and D  $\neq$  null
      and E  $\neq$  null
```

30

Query Bearbeitung

- table scan
- Verschneidung von Sekundärindexten und random Zugriff zu U
- Verschneidung von Bitmaps
- mehrdimensionale Indexte

Hinweis: U ~ universellem Blatt in Hierarchie, das alle Attribute erbt
Zwischenknoten der Hierarchie zur Index-Auswahl und
Query-Generierung

31

Modell T: Speicherung als Relation $T = (\text{OID}, \text{Attribut}, \text{Value})$
tripel mit Schlüssel **OID, Attribut**
Objekte mit Attributen A, D, E bekommt man über
3-fachen Self-Join

```
select T1.A, T2.D, T3.E
  from T T1, T T2, T T3
 where T1.Attribut = 'A'   and
        T2.Attribut = 'D'   and
        T3.Attribut = 'E'   and
        T1.OID      = T2.OID and
        T2.OID      = T3.OID
```

Frage: geschickte Berechnung dieser Query?

32

Frage: Konstruktion eines gesamten Objektes
mit Name = 'Bayer' ?

```
select OID, Attribut, Value
from T
where OID in
      select OID from T
      where Attribut = 'Name' and
            Value = 'Bayer'
```

Siehe auch Mitschang S. 6-8 bis 6-11

33

Modell R: redundante Speicherung in jeder Klasse mit allen privaten und geerbten Attributen, z.B. Person wird als

- Student mit allen Attributen gespeichert
- als Hiwi nochmal mit allen Attributen einschl. der studentischen gespeichert.

Anfragen: werden besonders einfach, da alle benötigte Info in der Relation, die zu einer Klasse gehört, enthalten ist.

Nachteil:

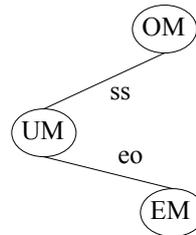
- hohe Redundanz
- aufwendige updates
- klassischer Tradeoff zwischen Update-, Wartungs-, Abfragekosten

34

Mengen Hierarchien

Sie entstehen durch **Wertspezialisierung** im Gegensatz zu Strukturspezialisierung

ss : subset of
eo : element of



$UM \subseteq OM$

UM und OM haben dieselbe syntaktische Struktur.

Def. von UM durch einschränkendes Prädikat $p(x)$
 $UM := \{x | x \in OM \wedge p(x)\}$

35

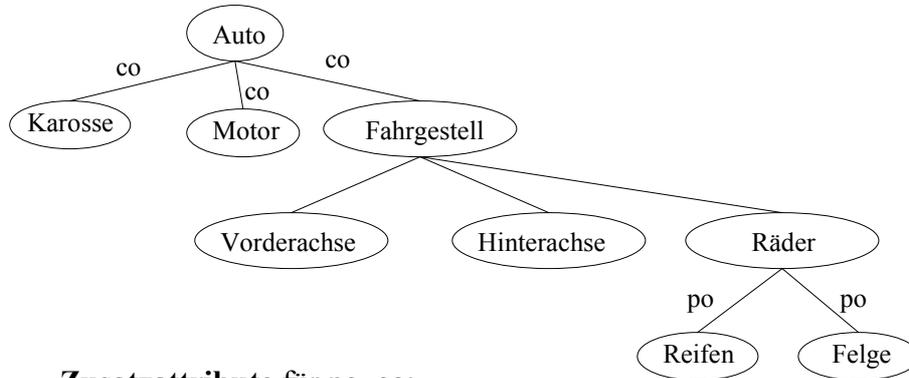
Hinweis: Generalisierungshierarchien und Mengen-Hierarchien oft vermischt

- Längs isa Kanten Vererbung von Struktur, nach unten zusätzliche Attribute
- Längs ss Kanten Vererbung von werteinschränkenden Prädikaten, mit konjunktiver Verknüpfung,
z.B. Beruf = 'Dipl.Inform' \wedge Alter < 32

36

Aggregations-Hierarchien

Zusammensetzung komplexer Systeme aus einfacheren Subsystemen, Komponenten



Zusatzattribute für po, co:

4 Räder

6 Zylinder, etc.

37

Hinweise:

- rein formal ist Aggregationshierarchie eine Umkehrung der Generalisierungshierarchie
- wie modelliert man Sonderausstattungen?
- Kombination von Generalisierungs-, Mengen-, und Aggregations-Hierarchien
- logische Modelle V, Z, U, T, R, M?
- Algorithmen, z.B. Stücklistengenerierung
- Integritätsbedingungen
- Kombination mit Prozessen, z.B. Montage-Reihenfolge

38