

# Grundlagen: Datenbanken

## 3. Zentralübung / Fragestunde

Linnea Passing

Harald Lang

[gdb@in.tum.de](mailto:gdb@in.tum.de)

Diese Folien finden Sie online.

Die Mitschrift stellen wir im Anschluss online.

# Agenda

- ▶ Hinweise zur Klausur
- ▶ Stoffübersicht/-Diskussion
- ▶ Wiederholung: MVDs / 4. Normalform
- ▶ Übung
  - ▶ Höchste Normalform bestimmen
  - ▶ Dekompositionsalgorithmus
  - ▶ Logische Optimierung
  - ▶ Erweiterbares Hashing

# Hinweise zur Klausur

## Termine

- ▶ 1. Klausurtermin
  - ▶ Mi. 24.02.2016, 10:30 Uhr
- ▶ 2. Klausurtermin
  - ▶ noch nicht bekannt. Prüfungsperiode vom 28.03. - 16.04.2016.  
(Anmeldung von 07.03 - 21.03.2016)
- ▶ **Raubekanntgabe**, jeweils (spätestens) eine Woche vorher!  
(via TUMonline sowie auf der Homepage)

## Verschiedenes

- ▶ 90 Minuten / 90 Punkte
- ▶ Sitzplatzvergabe (Aushang:  $MatrNr \mapsto Sitzplatz$ , KEINE Namensnennung)
- ▶ Betrugsfälle
- ▶ Notenbekanntgabe (via TUMonline)
- ▶ Einsichtnahme (Instruktionen auf der Homepage, nach Notenbekanntgabe)
- ▶ Bonus: Gilt für beide Klausuren.

# Stoffübersicht (1)

## Datenbankentwurf / **ER-Modellierung**

- ▶ ER-Diagramme, Funktionalitäten, Min-Max, Übersetzung ER  $\leftrightarrow$  Relational, Schemavereinfachung/-verfeinerung

## Das Relational Modell

- ▶ Stichworte: Schema, Instanz/Ausprägung, Tupel, Attribute,...
- ▶ **Anfragesprachen**
  - ▶ **Relationale Algebra**
    - ▶ RA-Operatoren: Projektion, Selektion, Join (Theta, Natural, Outer, Semi, Anti), Kreuzprodukt, Mengendifferenz/-vereinigung/-schnitt, Division
  - ▶ **Tupelkalkül**, Domänenkalkül

# Stoffübersicht (2)

## SQL

- ▶ ...

## Relationale Entwurfstheorie

- ▶ Definitionen:
  - ▶ Funktionale Abhängigkeiten (FDs), Armstrong-Axiome (+Regeln), FD-Hülle, Kanonische Überdeckung, Attribut-Hülle, Kandidaten-/Superschlüssel, Mehrwertige Abhängigkeiten (MVDs), Komplementregel, Triviale FDs/MVDs,...
- ▶ **Normalformen\***: 1., 2., 3.NF, BCNF und 4. NF
- ▶ Zerlegung von Relationen
  - ▶ in 3.NF mit dem **Synthesealgorithmus**
  - ▶ in BCNF\*/4.NF (zwei Varianten des **Dekompositionsalgorithmus**)
  - ▶ Stichworte: Verlustlos, Abhängigkeitsbewahrend

\* Es folgt noch eine Übungsaufgabe dazu.

## Stoffübersicht (3)

- ▶ Physische Datenorganisation
  - ▶ Speicherhierarchie
  - ▶ HDD/**RAID**
  - ▶ TID-Konzept
  - ▶ **Indexstrukturen (Bäume, Hashing\*)**
  
- ▶ Anfragebearbeitung
  - ▶ **Kanonische Übersetzung\*** (SQL → Relationale Algebra)
  - ▶ Logische **Optimierung\*** (in relationaler Algebra)
    - ▶ Frühzeitige Selektion, Kreuzprodukte durch Joins ersetzen, Joinreihenfolge
  - ▶ Implementierung relationaler Operatoren
    - ▶ ...
    - ▶ **Nested-Loop-Join**
    - ▶ **Sort-Merge-Join**
    - ▶ Hash-Join
    - ▶ Index-Join

\* Es folgt noch eine Übungsaufgabe dazu.

# Stoffübersicht (4)

- ▶ Transaktionsverwaltung
  - ▶ BOT, read, write, commit, abort
  - ▶ Rollback (R1 Recovery)
  - ▶ ACID-Eigenschaften
- ▶ Fehlerbehandlung (Recovery)
  - ▶ Fehlerklassifikation (R1 - R4)
  - ▶ Protokollierung: Redo/Undo, physisch/logisch, Before/After-Image, WAL, LSN
  - ▶ Pufferverwaltung: Seite, FIX, Ersetzungsstrategie steal/ $\neg$ steal, Einbringstrategie force/ $\neg$ force
  - ▶ Wiederanlauf nach Fehler, Fehlertoleranz des Wiederanlaufs, Sicherungspunkte
- ▶ Mehrbenutzersynchronisation
  - ▶ Formale Definition einer Transaktion (TA)
  - ▶ Historien (Schedules)
    - ▶ Konfliktoperationen, (Konflikt-)Äquivalenz, Eigenschaften von Historien
  - ▶ Datenbank-Scheduler
    - ▶ pessimistisch (sperrbasiert, zeitstempelbasiert), optimistisch

# **Relationale Entwurfstheorie**

# Normalformen: 1NF $\supset$ 2NF $\supset$ 3NF $\supset$ BCNF $\supset$ 4NF

- ▶ **1. NF:** Attribute haben nur atomare Werte, sind also nicht mengenwertig.
- ▶ **2. NF:** Jedes Nichtschlüsselattribut (NSA) ist voll funktional abhängig von jedem Kandidatenschlüssel.
  - ▶  $\beta$  hängt **voll funktional** von  $\alpha$  ab ( $\alpha \xrightarrow{\bullet} \beta$ ), gdw.  $\alpha \rightarrow \beta$  und es existiert kein  $\alpha' \subset \alpha$ , so dass  $\alpha' \rightarrow \beta$  gilt.
- ▶ **3. NF:** Frei von transitiven Abhängigkeiten (*in denen NSAe über andere NSAe vom Schlüssel abhängen*).
  - ▶ für alle geltenden nicht-trivialen FDs  $\alpha \rightarrow \beta$  gilt entweder
    - ▶  $\alpha$  ist ein Superschlüssel, oder
    - ▶ jedes Attribut in  $\beta$  ist in einem Kandidatenschlüssel enthalten
- ▶ **BCNF:** Die linken Seiten ( $\alpha$ ) aller geltenden nicht-trivialen FDs sind Superschlüssel.
- ▶ **4. NF:** Die linken Seiten ( $\alpha$ ) aller geltenden nicht-trivialen MVDs sind Superschlüssel.

# Mehrwertige Abhängigkeiten

multivalued dependencies (MVDs)

“Halb-formal”:

- ▶ Seien  $\alpha$  und  $\beta$  disjunkte Teilmengen von  $\mathcal{R}$
- ▶ und  $\gamma = (\mathcal{R} \setminus \alpha) \setminus \beta$
- ▶ dann ist  $\beta$  mehrwertig abhängig von  $\alpha$  ( $\alpha \twoheadrightarrow \beta$ ), wenn in jeder gültigen Ausprägung von  $\mathcal{R}$  gilt:
- ▶ Bei zwei Tupeln mit gleichem  $\alpha$ -Wert kann man die  $\beta$ -Werte vertauschen, und die resultierenden Tupel müssen auch in der Relation enthalten sein.

Wichtige Eigenschaften:

- ▶ Jede FD ist auch eine MVD (gilt i.A. nicht umgekehrt)
- ▶ wenn  $\alpha \twoheadrightarrow \beta$ , dann gilt auch  $\alpha \twoheadrightarrow \gamma$  (Komplementregel)
- ▶  $\alpha \twoheadrightarrow \beta$  ist trivial, wenn  $\beta \subseteq \alpha$  ODER  $\alpha \cup \beta = \mathcal{R}$  (also  $\gamma = \emptyset$ )



## Übung: Höchste NF bestimmen

$$\mathcal{R} = \{ A, B, C, D, E \}$$

$$A \rightarrow BCDE$$

$$BC \twoheadrightarrow E$$

- 1. NF
- 2. NF
- 3. NF
- BCNF
- 4. NF
- keine der angegebenen

# Schema in BCNF überführen

## BCNF-Dekompositionsalgorithmus (nicht abhängigkeitsbewahrend)

- ▶ Starte mit  $Z = \{\mathcal{R}\}$
- ▶ Solange es noch ein  $\mathcal{R}_i \in Z$  gibt, das nicht in BCNF ist:
  - ▶ Finde eine FD  $(\alpha \rightarrow \beta) \in F^+$  mit
    - ▶  $\alpha \cup \beta \subseteq \mathcal{R}_i$  (FD muss in  $\mathcal{R}_i$  gelten)
    - ▶  $\alpha \cap \beta = \emptyset$  (linke und rechte Seite sind disjunkt)
    - ▶  $\alpha \rightarrow \mathcal{R}_i \notin F^+$  (linke Seite ist kein Superschlüssel)
  - ▶ Zerlege  $\mathcal{R}_i$  in  $\mathcal{R}_{i.1} := \alpha \cup \beta$  und  $\mathcal{R}_{i.2} := \mathcal{R}_i - \beta$
  - ▶ Entferne  $\mathcal{R}_i$  aus  $Z$  und füge  $\mathcal{R}_{i.1}$  und  $\mathcal{R}_{i.2}$  ein, also  $Z := (Z - \{\mathcal{R}_i\}) \cup \{\mathcal{R}_{i.1}\} \cup \{\mathcal{R}_{i.2}\}$

# Schema in 4.NF überführen

## 4NF-Dekompositionsalgorithmus (nicht abhängigkeitsbewahrend)

- ▶ Starte mit  $Z = \{\mathcal{R}\}$
- ▶ Solange es noch ein  $\mathcal{R}_i \in Z$  gibt, das nicht in 4NF ist:
  - ▶ Finde eine **MVD**  $\alpha \twoheadrightarrow \beta \in \mathcal{F}^+$  mit
    - ▶  $\alpha \cup \beta \subset \mathcal{R}_i$  (FD muss in  $\mathcal{R}_i$  gelten)
    - ▶  $\alpha \cap \beta = \emptyset$  (linke und rechte Seite sind disjunkt)
    - ▶  $\alpha \rightarrow \mathcal{R}_i \notin \mathcal{F}^+$  (linke Seite ist kein Superschlüssel)
  - ▶ Zerlege  $\mathcal{R}_i$  in  $\mathcal{R}_{i.1} := \alpha \cup \beta$  und  $\mathcal{R}_{i.2} := \mathcal{R}_i - \beta$
  - ▶ Entferne  $\mathcal{R}_i$  aus  $Z$  und füge  $\mathcal{R}_{i.1}$  und  $\mathcal{R}_{i.2}$  ein, also  $Z := (Z - \{\mathcal{R}_i\}) \cup \{\mathcal{R}_{i.1}\} \cup \{\mathcal{R}_{i.2}\}$

## Übung: BCNF-Dekompositionsalgorithmus

$$\mathcal{R} = \{A, B, C, D, E, F\}, F_{\mathcal{R}} = \{B \rightarrow AD, DEF \rightarrow B, C \rightarrow AE\}$$

# **Anfragebearbeitung/-optimierung**

## Übung: Anfrageoptimierung

Geben Sie die kanonische Übersetzung der folgenden SQL-Anfrage an und optimieren Sie diese logisch:

```
SELECT DISTINCT s.name
  FROM studenten s, hören h, vorlesungen v
 WHERE s.matrnr = h.matrnr
       AND h.vorlnr = v.vorlnr
       AND v.titel = 'Grundzüge'
```

## Übung: Anfrageoptimierung (2)

Angenommen

- ▶  $|s| = 10000$
- ▶  $|h| = 20 * |s| = 200000$
- ▶  $|v| = 1000$
- ▶ 10% der Studenten haben 'Grundzüge' gehört

Dann ergeben sich

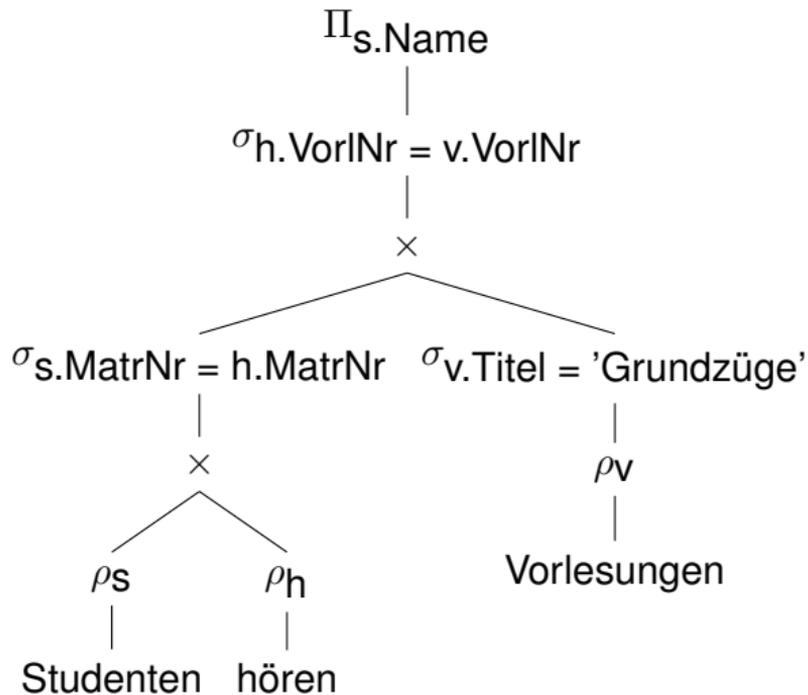
- ▶  $|s \times h \times v| = 10000 \cdot 20 \cdot 10000 \cdot 1000 = 2 \cdot 10^{12}$

Nach der Selektion verbleiben noch

- ▶  $|\sigma_p(s \times h \times v)| = 0,1 \cdot |s| = 1000$

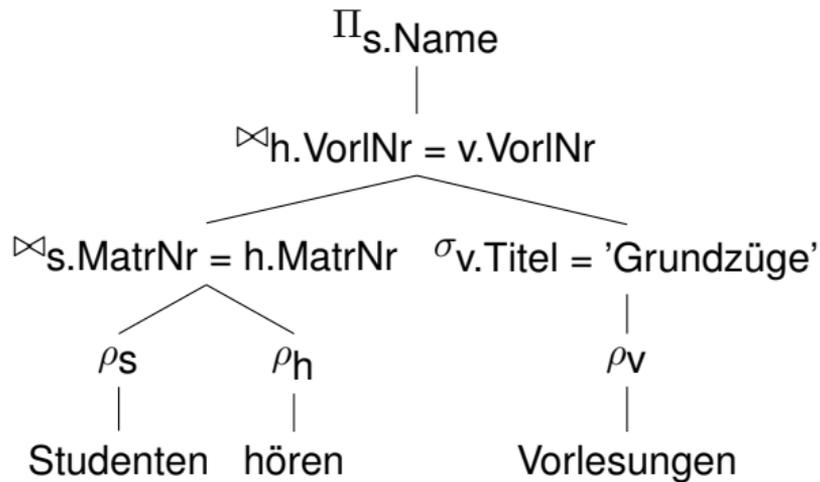
## Übung: Anfrageoptimierung (3)

**Optimierung 1:** Selektionen frühzeitig ausführen (*push selections*):



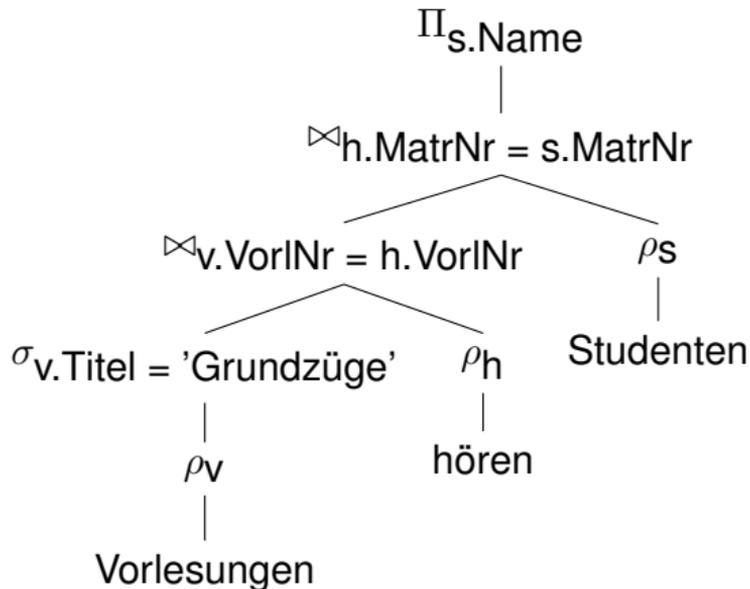
## Übung: Anfrageoptimierung (4)

**Optimierung 2:** Kreuzprodukte durch Joins ersetzen (*introduce joins*):



## Übung: Anfrageoptimierung (5)

**Optimierung 3:** Joinreihenfolge optimieren (*join order optimization*), so dass die Zwischenergebnismengen möglichst klein sind:



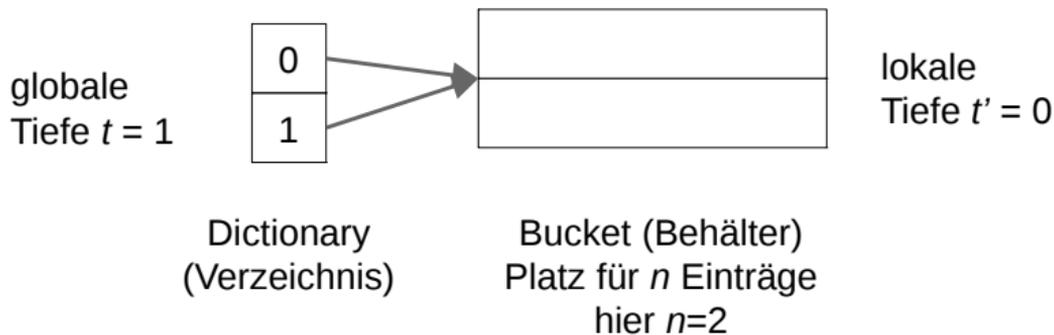
# **Erweiterbares Hashing**

# Erweiterbares Hashing

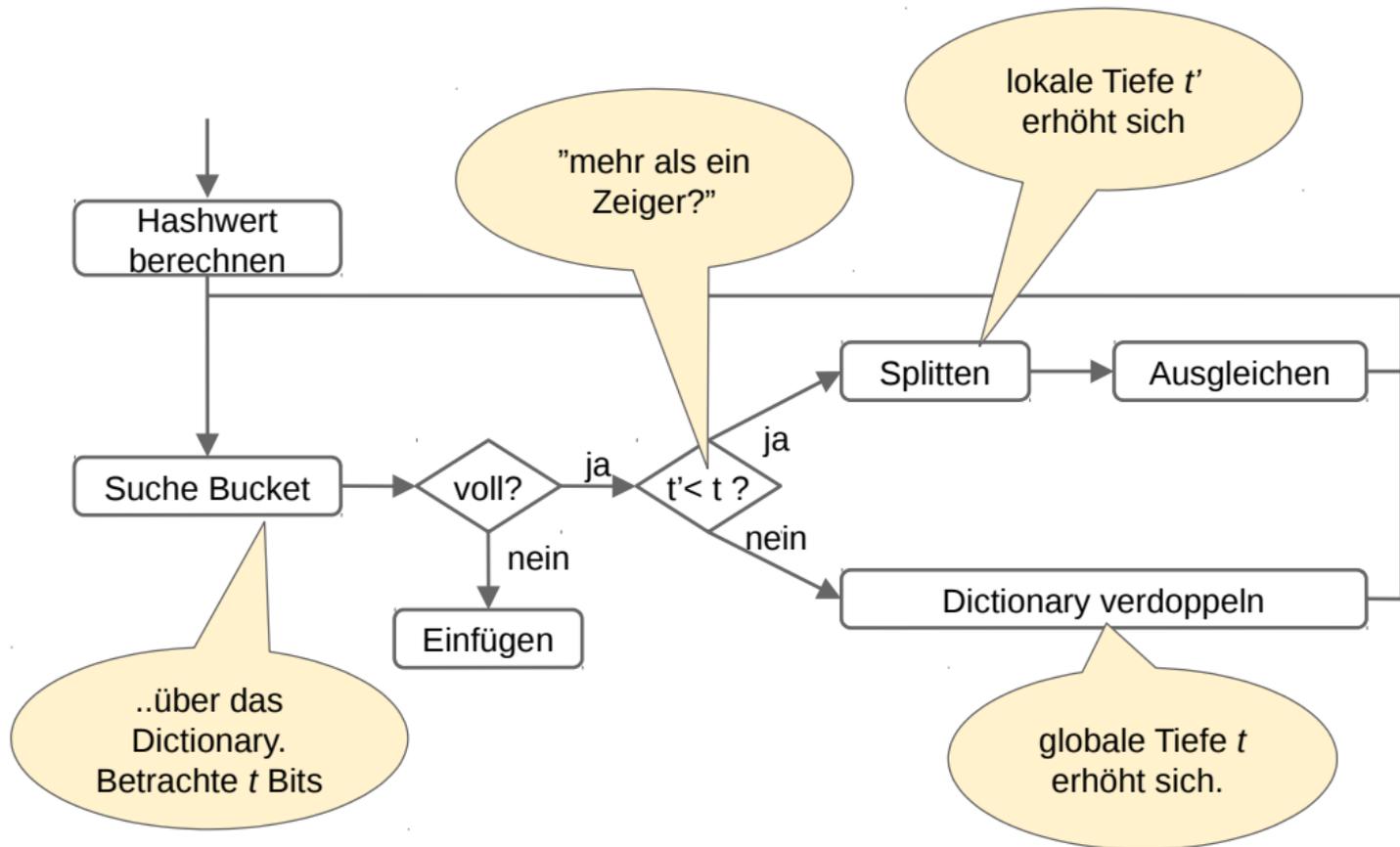
Hashfunktion  $h: \mathbf{S} \rightarrow \mathbf{B}$   
Schlüssel                      Bucket

wir betrachten die **Binärdarstellung** des Hashwerts

$h(x) = dp$   
Anzahl betrachteter Bits  
= globale Tiefe des Dictionaries  
unbenutzte Bits

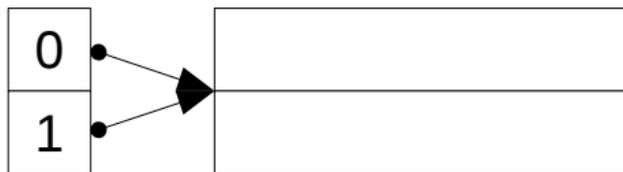


# Erweiterbares Hashing / Einfügen



## Übung: Erweiterbares Hashing / Einfügen

$x$	$h(x)$
A	1100
B	0100
C	1101
D	1010



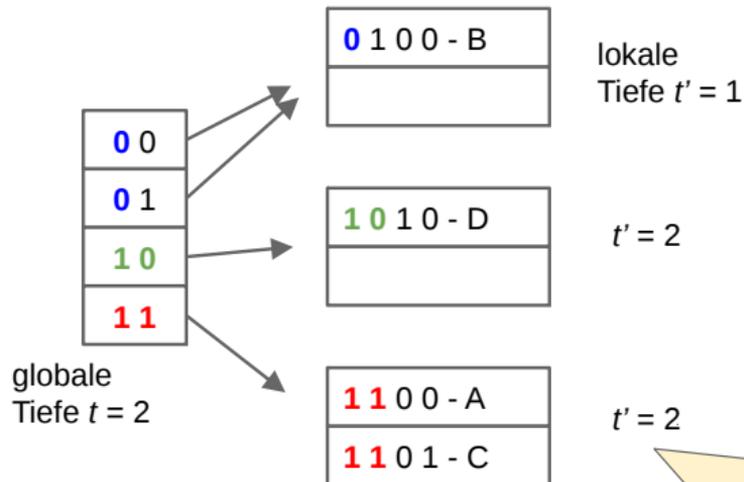
## Übung: Erweiterbares Hashing / Einfügen

$x$	$h(x)$
A	1100
B	0100
C	1101
D	1010

# Erweiterbares Hashing / Lösung

$x$	$h(x)$
A	1100
B	0100
C	1101
D	1010

$\underbrace{\hspace{1.5cm}}_d \quad \underbrace{\hspace{1.5cm}}_p$



in einem Bucket mit Tiefe  $t'$ , stimmen (mindestens) die  $t'$  führenden Bits der Hashwerte überein