Physische Datenorganisation

- Speicherhierarchie
- Hintergrundspeicher / RAID
- B-Bäume
- Hashing
- (R-Bäume)
- Objektballung
- Indexe in SQL

Kapitel 7 (Teil 2)

- Hashfunktion $h(x) = x \bmod 3$

0	
1	(27550, 'Schopenhauer', 6)
2	(24002, 'Xenokrates', 18)
	(25403, 'Jonas', 12)

- Kollisionsbehandlung durch Überlaufseiten

0		
1	(27550, 'Schopenhauer', 6)	
	(26830, 'Aristoxenos', 8)	
2	(24002, 'Xenokrates', 18)	→ (26120, 'Fichte', 10) → ...
	(25403, 'Jonas', 12)	(28106, 'Carnap', 3)

$\Longrightarrow$ ineffizient bei nicht vorhersehbarer Datenmenge

Hashing

- B-Bäume haben den Vorteil, dass sie Bereichsanfragen und Sortierungen unterstützen.
- Aufwand: O(log n)

- Hashing ist vom Aufwand O(1) für das Suchen einzelner Tupel, unterstützt aber keine Sortierung.

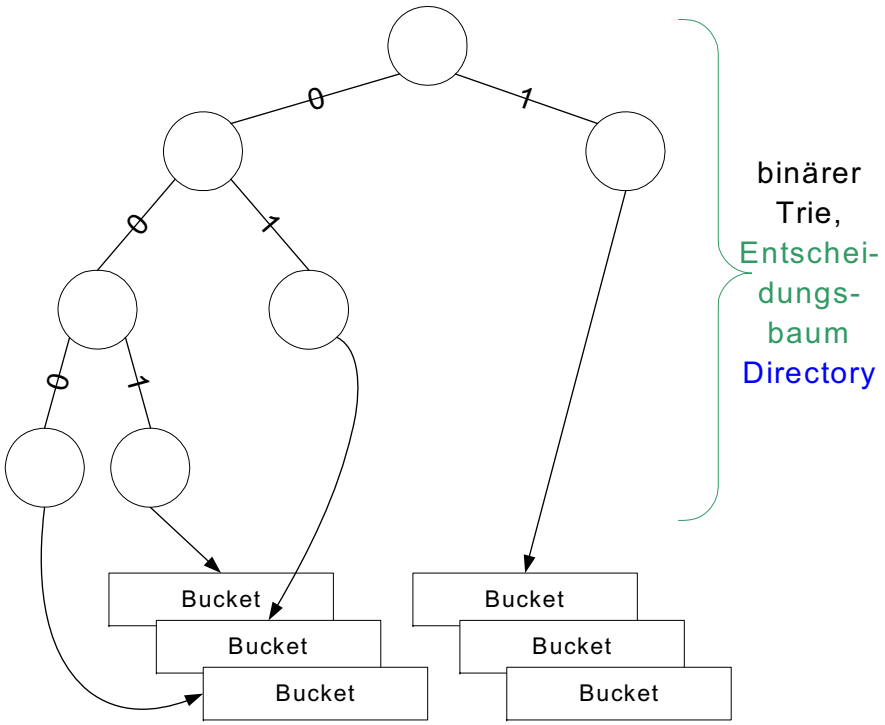
Statisches Hashing

- Einfacher Ansatz:
  - Hashfunktion h(x) = x mod N, wobei N die Größe der Tabelle in Seiten ist.
  - h(x) gibt die Seite an, auf der der Eintrag zu Schlüssel x zu finden ist.

- Probleme:
  - À priori Allokation des Speichers
  - Nachträgliche Vergrößerung der Hashtabelle ist „teuer"
    - Hashfunktion h(...) = ... mod N
    - Rehashing der Einträge
      - h(...) = ... mod M
    - In Datenbankanwendungen viele GB

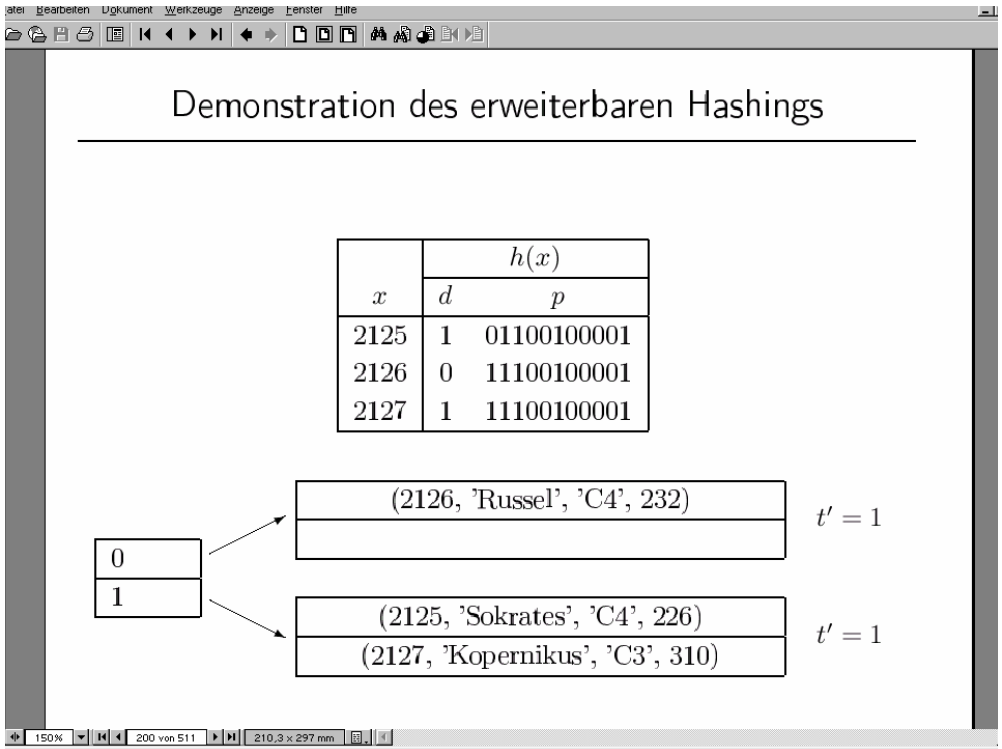
Erweiterbares Hashing

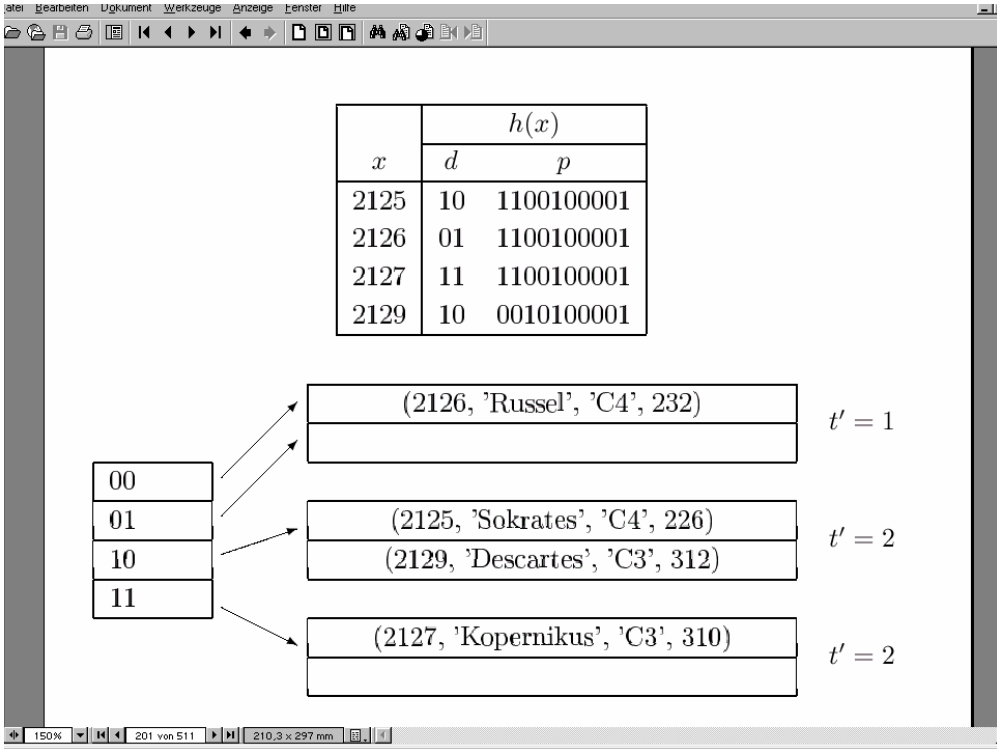
- Zusätzliche Indirektion über ein Directory
- Ein zusätzlicher Zugriff auf ein Directory, das den Zeiger (Verweis, BlockNr) des Hash-Bucket enthält
- Dynamisches Wachsen (und Schrumpfen) ist möglich
- Der Zugriff auf das Directory erfolgt über einen binären Hashcode

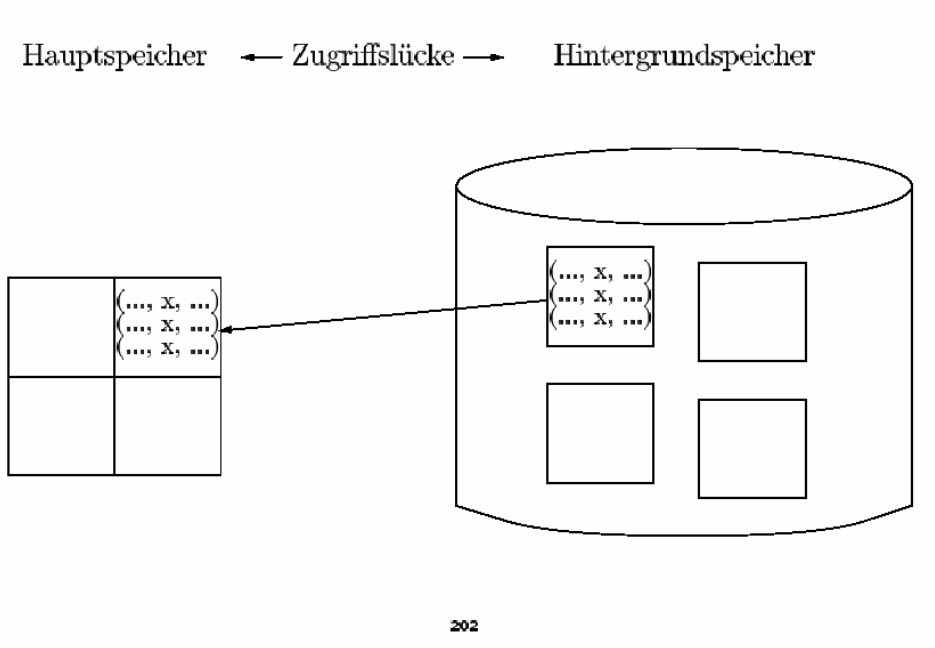


Hashfunktion für erweiterbares Hashing

- h: Schlüsselmenge → {0,1}*
- Der Bitstring muss lang genug sein, um alle Objekte auf ihre Buckets abbilden zu können
- Anfangs wird nur ein (kurzer) Präfix des Hashwertes (Bitstrings) benötigt
- Wenn die Hashtabelle wächst wird aber sukzessive ein längerer Präfix benötigt
- Beispiel-Hashfunktion: gespiegelte binäre PersNr
  - h(004) = 001000000... (4=0..0100)
  - h(006) = 011000000... (6=0..0110)
  - h(007) = 111000000... (7 =0..0111)
  - h(013) = 101100000... (13 =0..01101)
  - h(018) = 0100100000... (18 =0..010010)
  - h(032) = 000001000... (32 =0..0100000)
  - H(048) = 000011000... (48 =0..0110000)



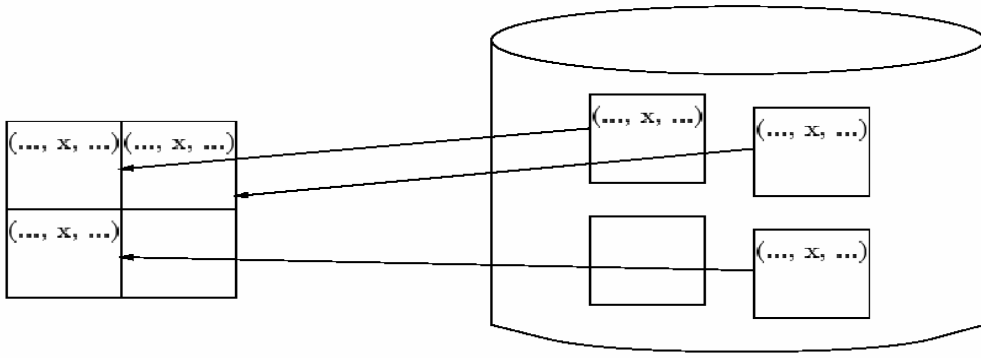




11

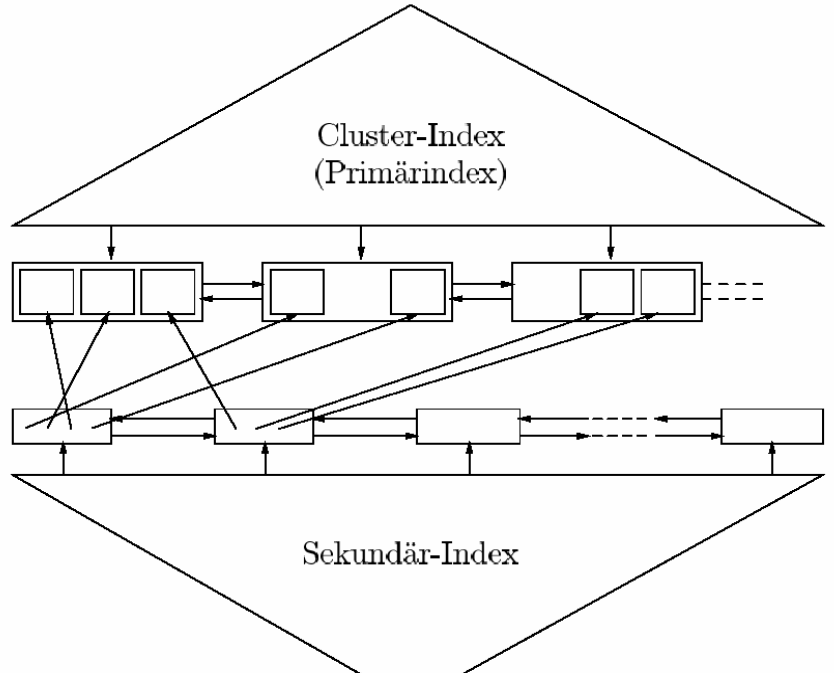
Objektballung / Clustering logisch verwandter Daten

select $*$
from R
where $A = x$;



10

Indexe und Ballung



Seite P_i

2125	Sokrates		C4		226
5041	Ethik		4	2125	
5049	Mäeutik		2	2125	
4052	Logik		4	2125	
2126	Russel		C4		232
5043	Erkenntnistheorie		3	2126	
5052	Wissenschaftstheorie		3	2126	
5216	Bioethik		2	2126	

Seite P_{i+1}

2133	Popper		C3		52
5259	Der Wiener Kreis		2	2133	
2134	Augustinus		C3		309
5022	Glaube und Wissen		2	2134	
2137	Kant		C4		7
5001	Grundzüge		4	2137	
4630	Die 3 Kritiken		4	2137	
	:				

Indexe in SQL

```
Create index SemsterInd
      on Studenten
      (Semester)
```

```
drop index SemsterInd
```


Unterstützung eines Anwendungsverhaltens

```
Select Name
From Professoren
Where PersNr = 2136
```

```
Select Name
From Professoren
Where Gehalt >= 90000 and Gehalt <= 100000
```