

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Architektur und Implementierung von Datenbanksystemen

WS 05/06

Dr. Jens-Peter Dittrich
jens.dittrich@inf
www.inf.ethz.ch/~jensdi
Institut für Informationssysteme



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Zugriffssystem

1. Speicherungsstrukturen

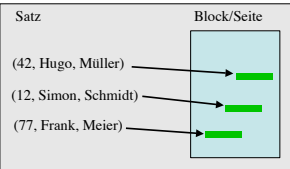


ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Satzverwaltung

Aufgabe: Abbildung von Sätzen auf Blöcke/Seiten

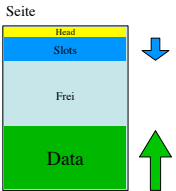
- Agenda
 - Aufbau einer Seite
 - Satzadressierung
 - Abbildung von Sätzen
 - Satzlayout
 - Speichermodelle
 - NSM
 - DSM
 - PAX
 - Komprimierung
 - Lange Felder
 - Freispeicherverwaltung



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Aufbau einer Seite

- Eine Seite besteht aus drei Teilen:
 - fixer Seitenkopf (Metadaten, z.B. Seitennummer, LSN)
 - Slots (Verweise auf einzelne Tupel)
 - Data (Repräsentation der Sätze)
- Slot = (Zeiger, Länge)
- Platz für Slots wird von vorne nach hinten allokiert
- Platz für Sätze wird von hinten nach vorne allokiert

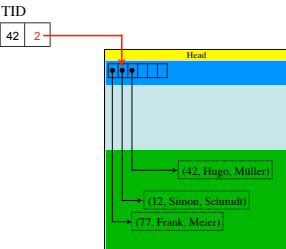


Vorteil: Sätze können problemlos innerhalb einer Seite migrieren

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

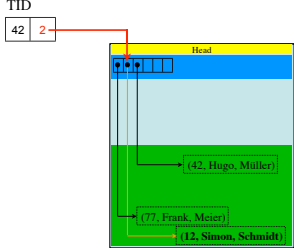
TID-Konzept

- Indirektion über Tuple-ID (TID)
TID = (Seite, Slot)



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Verschieben von Tupeln innerhalb der Seite



ETH
Hörsaalgebäude, Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Verschieben von Tupeln von einer der Seite

TID

42	2
----	---

Seite 42

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Hörsaalgebäude, Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Verschieben von Tupeln von einer der Seite

TID

42	2
----	---

Seite 42

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Hörsaalgebäude, Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

TID-Konzept

- Anmerkungen
 - Zugriff trivial, wenn Satz nicht auf eine andere Seite migriert
 - Verschieben von Sätzen über forward TIDs
 - bei weiterem Verschieben:
Abänderung der ersten forward TID

➔ maximal eine Indirektion durch forward TID

- Bewertung
 - minimal 1 Seitenzugriff erforderlich
 - maximal 2 Seitenzugriffe erforderlich (bei forward)

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Hörsaalgebäude, Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Indirekte Adressierung: Zuordnungstabelle

- Idee:
 - 1. halte separate Zuordnungstabelle
 - 2. gib nur logische Satzadressen nach aussen
 - keine forwards
 - bei Verschieben von Sätzen Eintrag in Tabelle anpassen

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Hörsaalgebäude, Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Verschieben von Tupeln: Zuordnungstabelle

Tabelle

S11	42	2
S43	..	

Seite 42

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Hörsaalgebäude, Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Verschieben von Tupeln: Zuordnungstabelle

Tabelle

S11	42	2
S43	..	

Seite 42

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Hörsaal für Informatiker
ETH Zürich

Verschieben von Tupeln: Zuordnungstabelle

Tabelle

S11	77	3
S43	..	

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Hörsaal für Informatiker
ETH Zürich

Zuordnungstabelle

- Nachteil der Zuordnungstabelle:
 - 2 Seitenzugriffe (1 Zuordnungstabelle + 1 Seite)
- Vorteile:
 - kein Verschnitt in Seiten durch forwards

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Hörsaal für Informatiker
ETH Zürich

Zuordnungstabelle (optimiert, aka PPP)

- Nachteil der Zuordnungstabelle:
 - 2 Seitenzugriffe (1 Zuordnungstabelle + 1 Seite)
- Optimierung:
 - Zugriff auf Zuordnungstabelle kann man einsparen, indem man häufig referenzierte Sätze in separatem Index im Hauptspeicher puffert

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Hörsaal für Informatiker
ETH Zürich

Zuordnungstabelle (optimiert, aka PPP)

Algorithmus zur Suche:

- Falls Satz-Adresse im Cache
 - kein E/A Zugriff auf Zuordnungstabelle
 - Adresse = Cache-Adresse
- Falls Satz nicht an Adresse gefunden:
 - E/A-Zugriff auf Zuordnungstabelle
- Sonst
 - E/A-Zugriff auf Zuordnungstabelle

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Hörsaal für Informatiker
ETH Zürich

Weitere Optreemierung

- Beobachtung: Zuordnungstabelle entspricht grosser Indexseite!
- Warum nicht die Abbildung

logische Satzadresse → physische Satzadresse

 in einer Indexstruktur speichern, z.B. einem B*-Baum?
- Bewertung
 - Vorteil: Sortierung der Einträge garantiert
 - Vorteil: keine zusätzliche Freispeicherverwaltung notwendig
 - Nachteil: teurer Zugriff durch mehrstufigen B*-Baum (für jeden Satzzugriff!)
 - Nachteil: Speicherauslastung schlecht

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Hörsaal für Informatiker
ETH Zürich

Abbildung von Sätzen

- Trennung von Metadaten und Nutzdaten
 - Metadaten: Daten im Katalog
 - Attributname
 - Typ
 - Nutzdaten: Daten auf der Seite
 - Wert
- N.B.
 - In der XML-Welt werden Metadaten und Nutzdaten gemeinsam abgespeichert:


```
<satz>
  <vorname> hugo </vorname>
  <name> müller </name>
</satz>
```

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

Satzlayout

- Teil fixer Länge
 - speichert alle Werte, die einen Typ fixer Länge haben
 - z.B. numeric(10,2), date, char[42]
 - Vorteil: direkte Adressberechnung
address = sizeof(type) * pos
- Teil variabler Länge
 - z.B. varchars
 - speichert Länge und Zeiger im fixen Teil
 - speichert Wert im variablen Teil
 - Nachteil: indirekte Adressberechnung
address = Zeiger
 - Wichtig: die Anwesenheit von varchars stört den direkten Zugriff auf Typen fixer Länge nicht!

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf.ethz.ch 19

Satzlayout

- NULL-Werte
 - kleine Bitmap fixer Länge am Anfang des Satzes
 - "1" falls Attribut den Wert NULL hat, "0" sonst
 - Vorteil: einfach und schnell

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf.ethz.ch 20

Beispiel für Satzlayout

Key	Vorname	Name
77	Frank	Meier
12	Simon	Schmidt
42	Hugo	Müller
11	Hans	Meier
25	Jens	Dittrich
76	Hugo	Schmidt

zeilenweises Auffüllen der Seite

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf.ethz.ch

n-ary Storage Model (NSM)

Key	Vorname	Name
77	Frank	Meier
12	Simon	Schmidt
42	Hugo	Müller
11	Hans	Meier
25	Jens	Dittrich
76	Hugo	Schmidt

- Sätze werden sequentiell abgespeichert
- Alle Attribute eines Satzes werden lokal benachbart gespeichert

Frage: Was passiert in der Speicherhierarchie?

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf.ethz.ch

Decomposition Storage Model (DSM)

RID	Key	Vorname	Name
1	77	Frank	Meier
2	12	Simon	Schmidt
3	42	Hugo	Müller
4	11	Hans	Meier
5	25	Jens	Dittrich
6	76	Hugo	Schmidt

- Teile Ausgangstabelle in 2-attributige Untertabellen auf
- RID muss ggf. nicht separat gespeichert werden

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf.ethz.ch 21

Decomposition Storage Model (DSM)

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf.ethz.ch 21

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Append Only(n)

- Verallgemeinerung von Append Only:
 - Betrachte immer nur die n zuletzt erzeugten Seiten
 - Wenn Satz in eine dieser Seiten passt OK
 - Sonst: erzeuge neue Seite
- Bewertung
 - sehr schnelles Einfügen
 - schlechte Speicherauslastung

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf 31

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Best Fit, First Fit, Next Fit

- Best Fit:
 - suche geeignetste Seite
 - Aufwand = lineare Suche in Liste
- First Fit:
 - nimm erste Seite die passt
- Next Fit:
 - wie First Fit, aber: starte Suche bei letzter Position

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf 32

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Hybride Verfahren HY(n,u)

- Prinzip:
 - Falls Speicherauslastung besser als u:
 - verwende Append Only(n)
 - Sonst
 - verwende Next-Fit
- Bewertung: guter Kompromiss

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf 33

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Freispeichertabelle

- Idee: Speichere für jede Seite verfügbaren Speicherplatz
- Realisierung: Tabelle auf Segmentebene
 - mit exakter Speicherbelegung

Seite	1	2	3	4	5	
Bytes frei	f ₁	f ₂	f ₃	f ₄	f ₅	

 - 2 Byte pro Eintrag
 - 2 Byte pro Eintrag
 - mit unscharfer Speicherbelegung

Seite	1	2	3	4	5	
Bytes frei	f ₁	f ₂	f ₃	f ₄	f ₅	

 - 2 Byte pro Eintrag
 - k Bits pro Eintrag

freier Platz $\leq (f_i / 2^k) * \text{Seitengröße}$

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf 34

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Space Map

- Nachteil der Freispeichertabelle: lineare Suche nach geeigneter Seite
- Lösung "Space Map":
 - Invertieren der Tabelle

Seite	1	2	3	4	5	
Bytes frei	f ₁	f ₂	f ₃	f ₄	f ₅	

Bytes frei	f ₄	f ₂	f ₃	f ₅	
Seite	4	1,2	3	5	

- Index über "Bytes frei" Attribut
- binäre Suche
- einfachste Variante: 0 oder 1, Seite hat Platz oder nicht
- grobgranulare Variante: 00, 01, 10, 11, Seite hat 0%, 25%, ... Platz
- (Diskussion analog zur Freispeichertabelle)

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf 35

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Space Map

- Trade-off: Granularität vs. Speicherverbrauch
- Trade-off: Granularität vs. Performanz
- Vorteil: effiziente Suche für Best Fit: O(1)
- Nachteil: update-Kosten für Space-Map

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf 36

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Zugriffssystem

2. Eindimensionale Zugriffspfade
(Teil 1)

SQL

Datensystem

Elemente (Sätze)

Zugriffssystem

Seiten, Segmente

Speichersystem

Bytes

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Motivation

- Anfragen:
 - Wie ist die Adresse von Student mit Matrikelnummer 424342?
 - Welche Studenten besuchen weniger als 2 Vorlesungen dieses Semester?
 - Welche Studenten wohnen nicht im Kanton Zürich?
- Wie beantwortet das DBMS diese Anfragen?
 - Alle Studenten-Datensätze anschauen (Sequentieller Zugriffspfad, siehe letzte VL-Stunde)
 - Daten geschickt organisieren, so dass die benötigten Sätze schnell gefunden werden (Baumstrukturierte Zugriffspfade, heute)

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Was heisst indizieren?

- Abbildung:
 - Schlüssel → Menge von Sätzen
 - Schlüssel muss nicht mit Primärschlüssel der Relation übereinstimmen

Zugriffspfade sorgen für die effiziente Implementierung dieser Abbildung

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Zugriffspfade

- Oft kann auf denselben Satz über verschiedene "Pfade" zugegriffen werden
- Zugriffspfad = Möglichkeit des Zugriffs auf einen Satz
- Zugriffspfade haben **sehr grossen** Einfluss auf die Effizienz der Anfragebearbeitung des DBMS

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Primärer vs. Sekundärer Zugriffspfad

- Zwei Klassen von Zugriffspfaden
 - Zugriffspfade für Primärschlüssel
Beispiel:

```
SELECT *
FROM Mitarbeiter
WHERE Personalnummer = 42
```
 - Zugriffspfade für Sekundärschlüssel
Beispiel:

```
SELECT *
FROM Mitarbeiter
WHERE Wohnort = 'ZH'
```

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zürich

Sekundäre Zugriffspfade und Invertierung

- Suche über Sekundären Zugriffspfad kann mehr als ein Ergebnis liefern (1:n Beziehung zwischen Schlüssel und Ergebnissen)

Vorname	TID	Key	Vorname	Name
Frank	1,0	77	Frank	Meier
Hans	1,1	12	Simon	Schmidt
Hugo	1,2	42	Hugo	Müller
Jens	1,3	11	Hans	Meier
Simon	1,4	25	Jens	Dittrich
	1,5	76	Hugo	Schmidt

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf

ETH
Hörsaal für Informatiker
Swiss Federal Institute of Technology Zurich

Eindimensionale Zugriffspfade

Eindimensionale Zugriffsstrukturen

- sequentielle Speicherungsstrukturen
 - Ketten
 - logisch
 - physisch
 - Listen
 - logisch
 - physisch
- Baumstrukturen
 - Mehrwegbäume
 - B-Bäume
 - baumstrukturiert
- gesteuerte Speicherungsstrukturen
 - statische Hashverfahren
 - konstant
 - dynamische Hashverfahren
 - dynamisch

fortlaufender (Ketten, Listen)
baumstrukturiert (B-Bäume)
Schlüsseltransformation (statische/dynamische Hashverfahren)
Schlüsselvergleich (Ketten, Listen, B-Bäume)

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf.ethz.ch 41

ETH
Hörsaal für Informatiker
Swiss Federal Institute of Technology Zurich

Eindimensionale Zugriffspfade

Eindimensionale Zugriffsstrukturen

- sequentielle Speicherungsstrukturen
 - Ketten
 - logisch
 - physisch
 - Listen
 - logisch
 - physisch
- Baumstrukturen
 - Mehrwegbäume
 - B-Bäume
 - baumstrukturiert
 - B+-Bäume
 - baumstrukturiert
- gesteuerte Speicherungsstrukturen
 - statische Hashverfahren
 - konstant
 - dynamische Hashverfahren
 - dynamisch

fortlaufender (Ketten, Listen)
baumstrukturiert (B-Bäume, B+-Bäume)
Schlüsseltransformation (statische/dynamische Hashverfahren)
Schlüsselvergleich (Ketten, Listen, B-Bäume, B+-Bäume)

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf.ethz.ch 42

ETH
Hörsaal für Informatiker
Swiss Federal Institute of Technology Zurich

Sequentielle Zugriffspfade: Ketten

- Liste von Sätzen ohne Rücksicht auf Seiten

Seiten

- Bewertung
 - sehr schlechtes E/A-Verhalten
 - Worst Case: 1 wahlfreier Zugriff pro Satz
 - spielt keine Rolle in DBMS

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf.ethz.ch

ETH
Hörsaal für Informatiker
Swiss Federal Institute of Technology Zurich

Sequentielle Zugriffspfade: Liste

- Liste von Sätzen gruppiert in Seiten
- Sätze sind physisch **geclustert**

Seiten

- Bewertung
 - besseres E/A-Verhalten
 - Worst Case: 1 wahlfreier Zugriff **pro Seite**
 - spielt eine Rolle in DBMS

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf.ethz.ch

ETH
Hörsaal für Informatiker
Swiss Federal Institute of Technology Zurich

Sequentielle Zugriffspfade: Sequenz

- Liste von Sätzen gruppiert in Seiten
- Sätze **und** Blöcke sind physisch geclustert

Seiten

- Bewertung
 - bestmögliches E/A-Verhalten
 - Worst Case: 1 wahlfreier Zugriff + sequentielle Reads
 - sehr wichtig für DBMS
 - Nachteil: schwierig aufrecht zu erhalten bei vielen inserts und updates

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf.ethz.ch

ETH
Hörsaal für Informatiker
Swiss Federal Institute of Technology Zurich

Bäume

- Binärbäume
 - Nicht geeignet für DBMS
 - Knoten lassen sich nur schwer auf Seiten abbilden
- Digitalbäume
 - spezielle Anwendungen
 - wichtig für nicht-relationale Daten
- B-Bäume
 - wichtigste Datenstruktur im Datenbankbereich
 - Vorteil: extrem vielseitig, lässt sich leicht erweitern
 - seit über 30 Jahren Forschung, immer noch wichtige Neuerungen
 - viele Indexstrukturen mit ähnlichen Ideen (R-Baum, M-Baum)

11. November 2005 Dr. Jens-Peter Dittrich / Institut für Informationssysteme / jens.dittrich@inf.ethz.ch 43

B-Bäume

- Grundlagen: siehe Algorithmen und Datenstrukturen (Widmayer)
- Agenda (für nächste Woche):
 - Grundlagen (Wiederholung)
 - ISAM
 - Bulk-Loading
 - Präfix B+-Bäume
 - Präfix/Suffix-Komprimierung
 - Cache-sensitive B+-Bäume
 - Primär vs. Sekundärindex
 - Clustered Index
 - Sekundärindexte und Google
 - ...

Nächste Woche

Zugriffssystem

2. Eindimensionale Zugriffspfade (Teil 2)

