

IDB Braindump WS16/17

17.02.2017

Inhaltsverzeichnis

1	Zuordnung zu Schichten	3
2	Programmschnittstelle	4
2.1	Programmieren	4
2.2	O/R-Mapping	4
3	Speichern	5
3.1	C-Store Umwandlung	5
3.2	C-Store Theorie	5
3.3	Datenunabhängigkeit	5
3.4	Systemkatalog	5
4	Puffer	6
4.1	Ziel	6
4.2	Seitenzuordnung	6
4.3	Seiteneinbringung	6
4.4	Lokalität	6
4.5	LFU	6
5	Schlüssel	7
5.1	Primärschlüssel	7
5.2	Verzweigungsgrad eines B*-Baums	7
6	Hashing	8
6.1	Erste Teilaufgabe	9
6.2	Zweite Teilaufgabe	9
6.3	Dritte Teilaufgabe	9
7	Transaktionen	10
7.1	ACID	10
7.2	Konsistenz	10
7.3	Abhängigkeitsgraph	10
7.4	Anomalien	10
8	R-Baum	11
8.1	Erstes Einfuegen	12
8.2	Zweites Einfuegen	12
9	SQL-Anfragen verarbeiten	13
9.1	Logisch vs Physisch	13
9.2	Operatorbaum	13
9.3	Optimierung	13

1 Zuordnung zu Schichten

Ordne die Begriffe den jeweiligen Schichten im Modell zu. Unterscheide dabei zwischen Aufgabe und Funktion der Schnittstelle!

1. ACID-Sicherstellung
2. Ausführung von SQL-String
3. Auswahl eines Planoperators
4. Extend-Tabelle
5. Kostenmodelle der Planoperatoren
6. MarkiereAlsVeraendert(Kachel)
7. Mengenorientierte Datenbankschnittstelle
8. Offene Addressierung
9. R*-Baum
10. SchreibeBlock(Zylinder, Puffer)
11. Schreiben(TID)
12. Strategie zur Seitenersetzung

2 Programmschnittstelle

2.1 Programmieren

Gegeben ist eine Relation `Studierende` mit folgenden Attributen:

- `Semester` (Integer)
- `Matrikelnummer` (Integer)
- `Name` (String)
- `Nachname` (String)
- `Geschlecht` (String `["Maennlich", "Weiblich"]`)

Schreibe eine Java-Methode um alle männlichen Studierenden, im 1. oder 2. Semester auszugeben. Der Methodenrumpf sowie die grobe Struktur sind bereits vorgegeben:

```
public void print(){
    Connection con = getConnection();

    Statement s = /* TODO */

    ResultSet r = /* TODO */

    /* TODO */
}
```

2.2 O/R-Mapping

Was ist O/R-Mapping?

3 Speichern

3.1 C-Store Umwandlung

Gegeben die folgenden Tabellen und Anfragen:

$X(\underline{x_id}, x_a_str, x_b_str, x_c_int, x_d_int)$

$Y(\underline{y_id}, fremd[x], y_a_str, y_b_int)$

SELECT **AVG**(x_c_int), x_b_str **FROM** X **JOIN** Y **on**
 $Y.fremd = X.x_id$ **GROUP BY** x_b_str

SELECT x_id, x_a_str **FROM** X **WHERE** $x_d_int > 0$

Geben sie ein mögliches Relationenschema an, um alle Informationen abzuspeichern und die gegebenen Anfragen möglichst effizient ausführen zu können. Geben sie pro Tabelle eine Begründung an, warum diese nötig ist. Geben sie pro Tabelle ebenfalls die Sortierung mit einer kurzen Begründung an.

3.2 C-Store Theorie

C-Store: Wie werden unsortierte Spalten mit vielen verschiedenen Werten gespeichert?
Wie werden sortierte Spalten mit wenig verschiedenen Werten gespeichert?

3.3 Datenunabhängigkeit

Definiere die Voraussetzungen der Datenunabhängigkeit.

3.4 Systemkatalog

Formulieren sie ein normalisiertes Relationenschema, wie der Katalog in normalen Tabellen gespeichert werden kann, um Anfragen der Form 'select-from-where' bearbeiten zu können. Geben sie dafür alle nötigen Tabellen und Attribute an. Auf Indizes und Optimierung soll verzichtet werden.

4 Puffer

4.1 Ziel

Was ist Ziel des Datenbankpuffers, wie erreicht man dieses?

4.2 Seitenzuordnung

Was ist direkte und indirekte Seitenzuordnung?

4.3 Seiteneinbringung

Was ist indirekte Seiteneinbringung und welchen Vorteil hat diese gegenüber direkter?

4.4 Lokalität

Gegeben folgende Referenzfolge, berechne für Fenstergrösse 3 die Lokalität L: 1 2 1 2 1 1 3 1 2 2 2

4.5 LFU

Nutze LFU als Seitenersetzungsstrategie, um die folgenden Seitenzugriffe in die Kacheln einzulagern. Hierbei sind sowohl die eingelagerten Seiten, als auch die Kontrollzustände anzugeben!

Zeit (in Sekunden)		...	7	8	9	10	11
Geforderte Seite		...	2	4	4	4	1
SeitenID	Kachel 1	...	1				
	Kachel 2	...	2				
	Kachel 3	...	3				
	Kachel 1	...	1				
	Kachel 2	...	2				
	Kachel 3	...	3				

5 Schlüssel

5.1 Primärschlüssel

Was ist richtig für Primärschlüssel? Es können mehrere Antworten richtig sein!

- Eindeutig innerhalb einer Relation
- Legt Art der Tupelspeicherung fest
- Legt Reihenfolge der Tupelspeicherung fest
- Ist nie NULL
- Besteht immer aus exakt einem Attribut
- Ist immer numerisch

5.2 Verzweigungsgrad eines B*-Baums

Gegeben seien folgende Größen:

- Blockgröße: 1024 Byte
- TID: 4 Byte
- Ganzzahlwerte: 4 Byte
- Zeigergröße auf einen Block: 8 Byte
- Schlüssellänge: 16 Byte
- Satzlänge: 32 Byte

Benutzen Sie die aus der Vorlesung bekannten Strukturen für innere und äußere Knoten für einen B*-Baum, skizzieren sie diese kurz, und berechnen sie jeweils den Verzweigungsgrad k_{blatt} und k_{in} .

6 Hashing

Für die jeweiligen Teilaufgaben:

- lineares Hashing
- Zwei Sätze pro Bucket
- Hashfunktion $h_j(k) = k \bmod 2^j q$, Initial $q = 2$ Buckets
- Split wenn in Überlaufbereich

Geben sie jeweils die verwendeten Hasfunktionen nach dem Einfügen eines neuen Wertes an, sowie zeichnen sie das Ergebnis nach der angegebenen Operation. Beachten Sie hierbei auch die Position des Zeigers.

6.1 Erste Teilaufgabe

Fuege 37 ein.

Hashfunktion davor: h_0

Zeiger	X							
Index	0	1	2	3	4	5	6	7
Satz 1	0	1						
Satz 2		5						
Ueberlauf 1								
Ueberlauf 2								

6.2 Zweite Teilaufgabe

Fuege 37 ein.

Hashfunktion davor: h_0, h_1

Zeiger		X						
Index	0	1	2	3	4	5	6	7
Satz 1	0	5						
Satz 2		7						
Ueberlauf 1								
Ueberlauf 2								

6.3 Dritte Teilaufgabe

Fuege 37 und 4 ein.

Hashfunktion davor: h_0, h_1

Zeiger		X						
Index	0	1	2	3	4	5	6	7
Satz 1	0	1	2	3				
Satz 2				11				
Ueberlauf 1								
Ueberlauf 2								

7 Transaktionen

7.1 ACID

Erläutere die 4 wichtigen Eigenschaften einer Transaktion in je einem Satz

7.2 Konsistenz

Erkläre den Unterschied zwischen physischer und logischer Konsistenz

7.3 Abhängigkeitsgraph

Zeichne den Abhängigkeitsgraphen und notiere das Objekt: Ist der Ablauf serialisierbar? Begründe!

$r1[B]$, $r2[B]$, $w3[A]$, $w2[C]$, $r2[A]$, $w1[C]$, $r1[A]$, $r3[A]$, $w2[A]$, $c1$, $c2$, $c3$

7.4 Anomalien

Nennen Sie zwei Anomalien mit Definitionen und zeigen sie diese im folgenden Programm. Geben sie hierfür alle Threads sowie beteiligte Elemente inklusive Zeile an!

	1	2	3	4
1	$r1[A]$			
2	$w1[A]$			
3	$c1$			
4		$w2[C]$		
5			$r3[D]$	
6		$r2[B]$		
7		$w2[B]$		
8				$w4[C]$
9				$r4[C]$
10		$r2[C]$		
11			$r3[B]$	
12			$w3[B]$	
13			$a3$	
14				$c4$
15		$c2$		

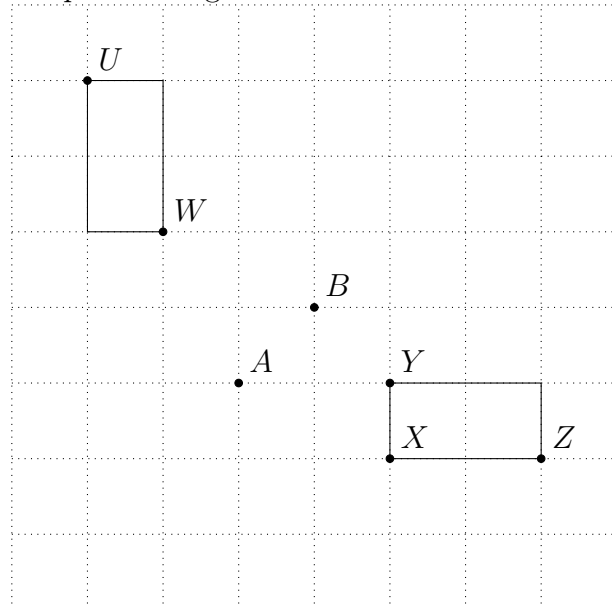
8 R-Baum

Gegeben ist ein R-Baum, $M = 4$, $m = 2$ Eintraege pro Knoten, zwei Dimensionen.

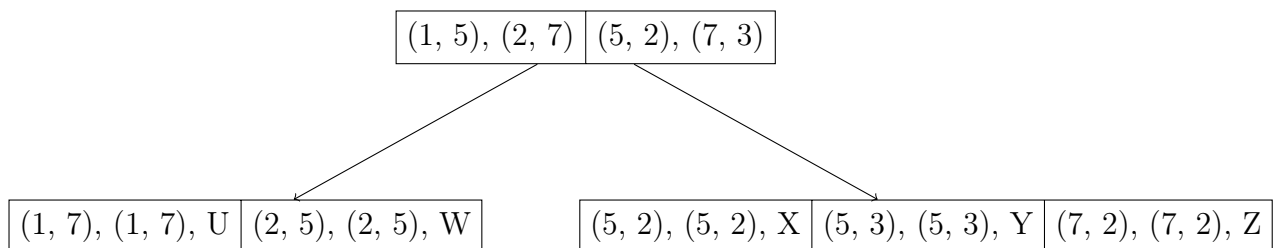
Gegeben folgende Tabelle:

Index	W	V
U	1	7
W	2	5
X	5	2
Y	5	3
Z	7	2
A	3	3
B	4	4

Graphisch dargestellt:



U bis Z wurden bereits in einen R-Baum eingefuegt, dieser sieht so aus:



8.1 Erstes Einfuegen

Fuege den Eintrag A in den Baum ein und zeichne das Endergebnis!

8.2 Zweites Einfuegen

Fuege den Eintrag B in den aus Aufgabe 1 entstandenen Baum ein und zeichne das Endergebnis!

9 SQL-Anfragen verarbeiten

9.1 Logisch vs Physisch

Erkläre logische und physische Planoperatoren sowie Unterschiede zwischen diesen.

9.2 Operatorbaum

```
SELECT a.a, a.b, AVG(b.a)
FROM a, b
WHERE b.b = a.a
GROUP BY a.a, a.b
HAVING COUNT(*) > 3 and a.b = 3;
```

Zeichne den logischen Operatorbaum.

9.3 Optimierung

Nenne zwei konkrete Möglichkeiten, den obigen Operatorbaum zu optimieren. Spezifizieren Sie hierbei die Stellen und Operatoren, auf welche sich diese Möglichkeiten beziehen.