

Der Rechteckige Tod

Gesammelte Aufgabenstellungen ähnlich zur Klausur *Implementierung von Datenbanksystemen* im Wintersemester 2016/17

Sqlinsław Lem

Kapitel 1: Schichtenmodell

Gegeben sind die folgenden Begriffe:

- Seitenersetzung
- `run_query(*sql)`
- `read(file, block, k)`
- ACID-Eigenschaften
- `set_modified(*frame)`
- B-Baum
- Blocktabellen
- Berechnung der Kosten
- Zugriffspfadverwaltung
- Interne Satzchnittstelle
- Sondieren

Ordnen Sie diese Begriffe in das bekannte Schichtenmodell ein.

Kapitel 2: Speicherung

1. Was bedeutet *Datenabstraktion*?
2. Der Systemkatalog wird oft selber auf Tabellen abgebildet. Wie könnte ein normalisiertes Relationenschema hierfür aussehen? Das DBVS muss nur `select from where`-Anfragen ohne Indexe und Optimierung verarbeiten können.
3. Wie werden in C-Store *sortierte/unsortierte Spalten mit vielen/wenigen verschiedenen Werten* gespeichert? (Insgesamt vier Kombinationen)
4. Gegeben sind die folgenden zwei Tabellen. Sie dokumentieren den Konsum von Schafen¹, z.B. kaufen sich Schafe teure Autos um das Schafsein erträglicher zu machen.

```
schaf(sid, name, oppositionell)
konsum(kid, konsument[schaf], beteubungsdauer)
```

Die neue Weltordnung möchte nun folgende Anfragen auswerten lassen:

```
1 SELECT name, SUM(beteubungsdauer)
2 FROM schaf JOIN konsum ON konsum.konsument = schaf.sid
3 GROUP BY oppositionell

1 SELECT sid, name
2 FROM schaf
3 WHERE oppositionell = True
```

Da es schon über acht Milliarden Schafe im Datenbanksystem gibt, sollen die Anfragen effizient durch C-Store implementiert werden. Geben Sie die dafür notwendigen Projektionen an. Es sollen keine Daten verloren gehen. Begründen Sie auch Ihre Projektion in einem Satz.

Kapitel 3: Anfrageverarbeitung

1. Zu der folgenden SQL-Anfrage ist der Operatorgraph zu zeichnen. Optimierungen sollen nicht durchgeführt werden.

```
1 SELECT hitman.id, hitman.town, sum(sale.netvalue)
2 FROM hitman, sale
3 WHERE sale.hitman = hitman.id
4 GROUP BY hitman.id, hitman.town
5 HAVING hitman.town = 'Vladivostok'
```
2. Wie kann der Operatorgraph aus der vorherigen Aufgabe optimiert werden?
3. Welche Arten von Operatoren kennen wir?

¹Wacht auf, Schafe!

Kapitel 4: Schlüsselzugriff

1. Welche Eigenschaften muss ein *Primärschlüssel* auf jeden Fall erfüllen?
2. Gegeben sind die Dimensionen einiger Komponenten eines B*-Baums:
 - Zeigergröße p
 - Satzlänge ℓ einer Tabelle R
 - Integergröße i (Verwendet als Zähler)
 - Blockgröße b
 - Schlüssellänge s für Tabelle R
 - TID-Größe τ

Ein B*-Baum besteht wie bekannt aus inneren Knoten und Blättern. Drücken Sie k_{innen} und k_{Blatt} durch die oben genannten Größen aus.

Kapitel 5: Hashindex

In dieser Aufgabe soll ein Hashindex verwaltet werden. Dieser Hashindex verwendet lineares Hashing und die einfachen Hashfunktionen $h_i(k)$,

$$h_i(k) = k \bmod 2^i q,$$

mit Schritten $i = 0, 1, \dots$ und der initialen Anzahl $q = 2$ an Buckets. Fügen Sie in diesen Hashindex die folgenden Schlüsselwerte ein und zeichnen Sie nach jedem Schritt den Zustand des Indexes. Geben Sie auch immer mit an, welche Hashfunktionen h_i gerade verwendet werden.

- 5
- 4
- 3
- 1337
- 15
- 136
- 7

Anmerkung von Sqlinstaw Lem: In richtigen Klausuren findet sich gewöhnlicherweise ein Vordruck und gleich zu Beginn sind einige Werte schon eingefügt. Diese Übung ist hier dem Leser überlassen.

Kapitel 6: Der Rechteckige Tod

Wir betrachten einen R-Baum mit mindestens $m = 2$ und höchstens $M = 4$ Einträgen je Knoten. Der R-Baum wird als Index der zwei Attribute (X, Y) verwendet. Momentan befinden sich im B-Baum fünf Einträge:

A (1, 10)

B (3, 8)

C (20, 4)

D (20, 2)

E (15, 3)

Im folgenden soll der Aufbau des R-Baums, d.h. dessen Knoten und Kanten, nach den genannten Operationen gezeichnet werden.

1. Ein weiterer Artikel **F** (8, 6) wird eingefügt.
2. Ein weiterer Artikel **G** (12, 15) wird eingefügt. Der R-Baum hat jetzt sieben Einträge.

Anmerkung von Sqlinstaw Lem: Es bietet sich an, den Datenbestand in einem Koordinatensystem mit zwei Achsen zu visualisieren.

Kapitel 7: Transaktionen und Synchronisation

1. Was bedeuten die *ACID-Eigenschaften*?
2. Welche Arten von *Konsistenz* kennen wir?
3. Wann ist ein Ablauf von Transaktionen *serialisierbar*? Falls der vollständige Abhängigkeitsgraph gegeben ist, wie kann man dies schnell erkennen?
4. Welche *Anomalien* können auftreten, wenn nicht korrekt synchronisiert wird?

Kapitel 8: Puffer

1. Warum verwendet man *Datenbankpuffer*? Wie funktionieren sie?
2. Was unterscheidet *direkte Seitenzuordnung* von *indirekter Seitenzuordnung*?
3. Ihr Lieblingskommilitone meint, *indirekte Seiteneinbringung* ist viel besser als *direkte Seiteneinbringung*. Auf welchen Vorteil spielt er an?
4. Gegeben ist die Seitenreferenzfolge

5, 1, 25, 1, 25, 9, 1, 5, 5, 1

Bestimmen Sie die durchschnittliche Lokalität L bei einer Fenstergröße $w = 4$.

Kapitel 9: Programmierschnittstelle

1. Wir betrachten ein Datenbanksystem mit einer Tabelle `Studierende`. Diese Tabelle hat die folgenden Felder:

- Attribut `Nr` vom Typ `Integer`
- Attribut `Name` vom Typ `String`
- Attribut `Geschlecht` vom Typ `Float`

Ergänzen Sie folgende Javamethode `printStudentsElegibleForExtraAuDTutorials`, so dass sie alle Studierende ausgibt, die zumindest 50% weiblich sind. Das Attribut `Geschlecht` ist, falls nicht `NULL`, ein `Float` zwischen 0 und 1, wobei bei einem Cis-Mann `Geschlecht = 0` und bei einer Cis-Frau `Geschlecht = 1` gesetzt ist.

```
1 void printStudentsElegibleForExtraAuDTutorials() {
2     try {
3         Connection c = Mngr.connect("::1", "AzureDiamond", "hunter2");
4         Statement s =
5
6
7         Results r =
8
9
10
11
12
13
14
15
16
17
18
19     } catch (Exception e) {}
20 }
```

2. Was versteht man unter einem *O/R-Mapping*?