

**Datenbanksysteme II**  
SS 2013

**Übungsblatt 6: Hardwaregrundlagen, Logische Anfrageoptimierung**

Besprechung: 17.-19.06.2013

**Aufgabe 6-1**     *Sequentieller Scan vs. Index*

Es werde ein komplexes Selektionsprädikat auf einer Tabelle aus 1.000.000 Tupeln ausgewertet. Ein Tupel belege hierbei 100 Bytes. Für die Operation stehen 10 MByte Datenbankpuffer zur Verfügung. Die Auswertung des Prädikats benötige  $10 \mu\text{s}$  CPU-Zeit. Die Daten des Plattenlaufwerkes seien wie folgt gegeben:

- $t_{seek} = 4 \text{ ms}$
- $t_{lat} = 2 \text{ ms}$
- Transferrate = 50 MByte/s.

(a) Zunächst sei die Tabelle in einer Datei organisiert, die sequentiell gelesen wird.

- (i) Wie viel Zeit benötigt das Einlesen der Datei? Fällt die Latenzzeit ins Gewicht?
- (ii) Wie viel CPU-Zeit benötigt die Selektionsanfrage?

(b) Nun sei die Tabelle in einem Index ( $B^+$ -Baum) organisiert. Eine Seite habe eine Größe von 4 KByte. Der Baum habe drei Directory-levels.

- (i) Der Index habe auf dem Selektionsprädikat optimale Selektivität (*best case*). Das heißt es reicht eine einzige Datenseite zu besuchen. Wie teuer ist die Auswertung (CPU und I/O)? Nehmen Sie an, dass die Auswertung des Prädikats auf einer Directory Seite  $30 \mu\text{s}$  CPU-Zeit benötigt, um den richtigen Teilbaum zu finden.
- (ii) Wie teuer ist die Auswertung (CPU und I/O) bei schlechter Selektivität (*worst case*)? In diesem Fall müssen alle Datenseiten untersucht werden. Ist die Selektion CPU- oder I/O-bound? Sie können ausnutzen, dass beim  $B^+$ -Baum alle Datenseiten linear verlinkt sind.

**Aufgabe 6-2**    *Logische Anfrageoptimierung*

Gegeben sei ein Relationenschema mit folgenden Relationen und (unvollständigen) Ausprägungen:

Relation **Student**:

<u>MatrNr</u>	Name	...
12345	Maier	...
23456	Müller	...
34567	Huber	...
45678	Meier	...
56789	Bauer	...
67890	Klein	...
78901	Schneider	...
89012	Richter	...
90123	Schmitt	...
98765	Schulz	...
87654	Fischer	...
76543	Meyer	...
65432	Weber	...
54321	Wagner	...
43210	Koch	...
32109	Becker	...
21098	Meier	...
10987	Schmidt	...
13579	Meyer	...
24680	Müller	...

Relation **Lehrveranstaltung**:

<u>LVNr</u>	LVTyp	...
1	Vorlesung	...
2	Seminar	...
3	Vorlesung	...
4	Übung	...
5	Übung	...
6	Praktikum	...
7	Seminar	...
8	Vorlesung	...
9	Seminar	...
10	Seminar	...

Relation **Hoert**:

<u>MatrNr</u>	<u>LVNr</u>
54321	2
76543	9
13579	1
76543	3
76543	1
12345	9
13579	4
34567	8
13579	7
54321	1

Relation **Dozent**:

<u>DozNr</u>	Titel	Name	...
1	Dr.	Einstein	...
2	Prof.	Kröger	...
3	PD Dr.	Einstein	...
4	Prof.	Einstein	...
5	Dr.	Schubert	...

Relation **Haelt**:

<u>DozNr</u>	<u>LVNr</u>
4	1
1	2
2	3
3	4
2	5
3	6
4	7
2	8
4	9
2	10

- Geben Sie den kanonischen Operatorbaum für folgende Anfrage an:  
 "Gesucht sind die Namen aller Studenten, die bei Professor Einstein ein Seminar besuchen."  
 Geben Sie dabei die Selektionen nicht zusammengefasst, sondern einzeln an.
- Welche Optimierungsmöglichkeiten ergeben sich?
- Optimieren Sie die Anfrage logisch und geben Sie einen optimierten Operatorbaum an.
- Vergleichen Sie die resultierenden Tupelzahlen der Zwischenschritte der kanonischen Anfragebearbeitung mit den Zahlen Ihrer optimierten Variante.