

Aufgabe 1: Indexe

(1 P.)

a) Geben Sie SQL Statements an, die Indexe kreieren, die für die folgenden Anfragen geeignet sind:

- i) `SELECT matnr FROM studenten WHERE name = 'Thomas'`
- ii) `SELECT matnr FROM studenten WHERE strasse = 'Hauptstraße' AND nummer <= 50`
- iii) `SELECT matnr FROM studenten s WHERE semester > 10 AND (SELECT COUNT(*) FROM pruefungen p WHERE p.matnr = s.matnr) = 0`

Begründen Sie Ihre Wahl.

b) Gegeben die folgende Anfrage:

```
SELECT * FROM professoren WHERE stufe IN('W2', 'W3') AND alter < 50
AND nachname LIKE 'm%' AND fachbereich = 8
```

Welche der folgenden Indexe können benutzt werden, um diese Anfrage schneller zu beantworten, und wieso?

- i) B⁺-Baum-Index auf (Fachbereich, Nachname, Vorname)
 - ii) B⁺-Baum-Index auf (Vorname, Nachname, Fachbereich)
 - iii) B⁺-Baum-Index auf (Fachbereich DESC, AngestelltSeit, Alter)
 - iv) Hash-Index auf (Stufe, Nachname)
 - v) Hash-Index auf (Stufe, Fachbereich)
- c) Fügen Sie die folgenden Schlüssel in einen anfangs leeren B⁺Baum vom Typ (2,2) mit zusammengesetzten Schlüsseln ein, einmal durch Einfügen einzelner Einträge und einmal durch die in der Vorlesung vorgestellte Bulk-Loading-Technik. Zeichnen Sie den Baum jeweils so, dass deutlich wird, in welcher Reihenfolge neue Knoten angelegt werden.
(A,2) (M, 5) (D,8) (M, 9) (H, 4) (J, 9) (K, 4) (N,8) (M, 7) (J,1)

Aufgabe 2: Optimierung und Kostenmodelle

(1 P.)

Gegeben die Relationen $R(A, B, C)$, $S(D, E, F)$ und $U(A, D, G)$ mit folgenden Statistiken:

$T(R) = 1000$	$V(R, A) = 1000$	$V(S, D) = 1000$	$V(U, A) = 5$
$T(S) = 1000$	$V(R, B) = 10$	$V(S, E) = 50$	$V(U, D) = 4$
$T(U) = 20$	$V(R, C) = 200$	$V(S, F) = 10$	$V(U, G) = 20$

$U.A$ wird als Fremdschlüssel referenziert von $R.A$, $U.D$ von $S.D$. Die Einträge pro Attribut sind zwischen 1 und Kardinalität, z.B. nimmt $S.F$ Werte zwischen 1 und 10 an. Weiterhin gibt es keine NULL-Werte und Sie können von gleichverteilten und unabhängigen Werten ausgehen.

Betrachten Sie die folgenden Anfragen:

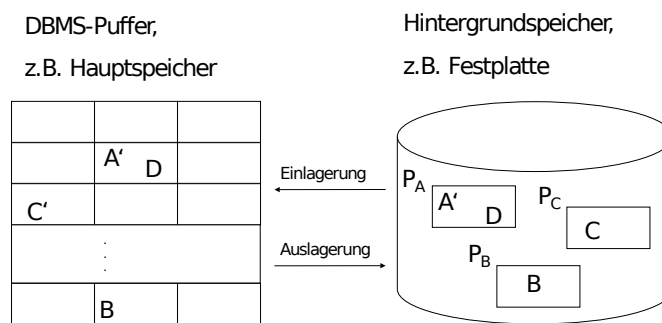
- i) $\sigma_{R.A=U.A}(\sigma_{U.D=S.D}(\sigma_{R.B=6}(\sigma_{U.G \neq 10}(R \times (S \times U))))))$
- ii) $\pi_{R.A, R.B}(\sigma_{R.B=2}(\sigma_{U.G \leq 10}(\sigma_{U.A=R.A}((R \times U) \cup (U \times R))))))$

Optimieren Sie die Anfragen nach der in der Vorlesung vorgestellten Vorgehensweise ("Regelbasierte Optimierung"). Benutzen Sie dabei die im Kapitel über Relationale Algebra vorgestellten Äquivalenzregeln. Geben Sie für Ihren finalen Plan die Größe des Ergebnisses sowie der Zwischenergebnisse in Tupeln an.

Aufgabe 3: Transaktionen

(1 P.)

- a) Beschreiben Sie mit eigenen Worten, was die Buchstaben von ACID bedeuten.
- b) Gegeben zwei Transaktionen t_1 und t_2 und folgende Situation der Seiten P_A , P_B und P_C sowie der Datensätze A, B, C und D. Transaktion t_1 hatte bislang Zugriff auf Datensätze A und B. Transaktion t_2 hatte bislang Zugriff auf Datensätze C und D. Durch ' markierte Datensätze illustrieren geänderte Datensätze.



Beantworten Sie folgende Fragen:

- Was muss gewährleistet werden, wenn Transaktion t_1 bereits erfolgreich committed hat?
 - Was muss passieren, wenn Transaktion t_1 noch nicht committet hat und abbricht?
 - Was muss passieren, wenn Transaktion t_2 noch nicht committet hat und abbricht?
 - Wie oft musste eine Seite in den Puffer geladen werden?
 - t_2 möchte auch Datensatz B lesen, was muss gelten, damit dies OK ist?
- c) Wenn ein Datenbanksystem mit \neg force und steal konfiguriert ist, braucht es UNDO, REDO, beides, oder weder noch? Begründen Sie Ihre Antwort.