



5. Übungsblatt

Für die Übung am Donnerstag, 30. November 2006,
von 15:30 bis 17:00 Uhr in 13/222.

Aufgabe 1: Zugriffskostenvergleich mit und ohne Clustering

In einer DB sei ein Segment mit M Seiten gegeben, in dem mehrere Relationen und die zugehörigen Indexstrukturen gespeichert sind. Unter anderem befindet sich in diesem Segment folgende Relation:

```
PERSONAL (PNR, NAME, ADRESSE, GEHALT, MNR, ...)
```

mit Attributen für die Personalnummer, den Namen des Angestellten, die Adresse, das Gehalt sowie den Manager, bei dem der Mitarbeiter beschäftigt ist. Daneben sei noch weitere Information enthalten. Die Relation PERSONAL enthalte P Datensätze. Bei Clusterbildung seien in den zum Cluster gehörenden Seiten durchschnittlich C Datensätze zu finden. Weiterhin habe ein Manager im Mittel Q Mitarbeiter.

Wie viele Seitenzugriffe sind notwendig, um folgende Anfragen zu beantworten:

```
SELECT * FROM PERSONAL WHERE MNR = X, wobei X eine Personalnummer ist.
```

```
SELECT * FROM PERSONAL ORDER BY MNR
```

jeweils unter Berücksichtigung nachfolgender Situationen:

Es existieren keinerlei Indexstrukturen auf der Relation PERSONAL.

Es gibt eine Indexstruktur auf PNR von PERSONAL.

Es gibt eine Indexstruktur mit Clusterbildung auf PNR von PERSONAL.

Es gibt eine Indexstruktur auf MNR von PERSONAL.

Es gibt eine Indexstruktur mit Clusterbildung auf MNR von PERSONAL.

Man benutze dabei einen B*-Baum als Indexstruktur, bei der nur Zeiger auf Seiten, in denen sich die zugehörigen Datensätze befinden, in Blättern gespeichert sind. Anschließend berechnen Sie die jeweiligen Kosten für folgende Größen:

$M = 10^6$ Seiten, $P = 10^5$ Datensätze, $C = 5$ Datensätze, $Q = 100$ Datensätze.

Die Seitengröße S_L beträgt 4 KB.

Die Eintragslänge im B*-Baum E_L beträgt 10 Bytes.

Was passiert bei Änderungen, wenn eine Indexstruktur mit Clusterbildung vorliegt?

Aufgabe 2: Kostenmodelle für die Selektionsoperation

Gegeben sei eine Tabelle R mit den Attributen $A_1, A_2, A_3, \dots, A_n$, die zusammenhängend in den Seiten des Segments S gespeichert ist.

$R(A_1, A_2, A_3, \dots, A_n)$

Das Segment S habe $M_S = 10^4$ Seiten. Die Tabelle R habe $N_R = 10^5$ Sätze und ggf. (für die entsprechenden Aufgabenstellungen) einen Cluster-Faktor $c_R = 50$.

Weiterhin seien die Indizes $I_R(A_1)$ mit $j_{A_1} = 100$ und $I_R(A_2)$ mit $j_{A_2} = 10$ für die Attribute A_1 und A_2 angelegt. Bei Indizes sind jeweils als B*-Bäume mit der Höhe $h_B = 2$ und $N_B = 100$ Blattseiten realisiert.

a) Wie teuer (Anzahl der Seitenzugriffe) ist die Auswertung der SQL-Anfrage

```
SELECT *  
FROM R  
WHERE A3='x'
```

- (1) bei einem Tabellen-Scan?
- (2) bei Nutzung des Indexes $I_R(A_1)$?
- (3) bei Nutzung des Indexes $I_R(A_2)$ mit Cluster-Bildung?
- (4) wenn die Tabelle als Hash-Struktur mit A_3 als Primärschlüssel angelegt ist?

b) A_1 habe 100 Werte, die von 1 bis 100 gleichverteilt vorkommen ($j_{A_1} = 100$). Wie teuer ist die Auswertung der SQL-Anfrage

```
SELECT *  
FROM R  
WHERE A1 > 50
```

- (1) bei Nutzung des Indexes $I_R(A_1)$?
- (2) bei Annahme einer Cluster-Bildung bei $I_R(A_1)$?
- (3) ohne Indexnutzung?

c) Welche Kosten verursacht die SQL-Anfrage

```
SELECT *  
FROM R  
WHERE A1 = 50 AND A2 = 10
```

- (1) bei Nutzung von $I_R(A_1)$ und $I_R(A_2)$ jeweils ohne Cluster-Bildung?
- (2) bei gemeinsamer Nutzung von $I_R(A_1)$ und $I_R(A_2)$ mit Cluster-Bildung?
- (3) bei Zugriff nur über $I_R(A_2)$ mit Cluster-Bildung?
- (4) bei Zugriff nur über $I_R(A_1)$ mit Cluster-Bildung?

Aufgabe 3: Allgemeines zum Sicht-Konzept

Welche Maßnahmen erfordert die Definition einer speziellen Sicht, wenn als Sichtsemantik

- a) eine Kopie des aktuellen Sichtinhalts
- b) ein dynamisches Fenster auf die Basisrelation unterstellt wird?

Diskutieren Sie verschiedene Fälle von Manipulationen an der Basisrelation ANGESTELLTE für das folgende Beispiel:

```
CREATE VIEW UNTERBEZAHLT AS
SELECT PNR, NAME, GEHALT
FROM ANGESTELLTE
WHERE BERUF='PROGRAMMIERER' AND GEHALT<20000
```

Aufgabe 4: CHECK OPTION bei Sichten in SQL

Gegeben seien folgende SQL-Anweisungen:

```
CREATE TABLE T (S1 INT, S2 INT, S3 INT, S4 INT, S5 INT);
CREATE VIEW V1 AS
SELECT * FROM T WHERE S1=1;
CREATE VIEW V2 AS
SELECT * FROM V1 WHERE S2=2 WITH LOCAL CHECK OPTION;
CREATE VIEW V3 AS
SELECT * FROM V2 WHERE S3=3;
CREATE VIEW V4 AS
SELECT * FROM V3 WHERE S4=4 WITH CASCADED CHECK OPTION;
CREATE VIEW V5 AS
SELECT * FROM V4 WHERE S5=5;
```

Ist die Ausführung der nachfolgenden INSERT-Anweisungen erfolgreich? Geben Sie jeweils eine kurze Begründung an.

- a) INSERT INTO V1 VALUES (2, 1, 3, 2, 5);
- b) INSERT INTO V2 VALUES (2, 1, 3, 2, 5);
- c) INSERT INTO V2 VALUES (2, 2, 3, 2, 5);
- d) INSERT INTO V3 VALUES (2, 2, 4, 2, 5);
- e) INSERT INTO V3 VALUES (1, 3, 3, 2, 5);
- f) INSERT INTO V4 VALUES (2, 2, 3, 2, 5);
- g) INSERT INTO V4 VALUES (2, 1, 3, 4, 5);
- h) INSERT INTO V4 VALUES (1, 2, 2, 4, 5);
- i) INSERT INTO V4 VALUES (2, 2, 3, 4, 5);
- j) INSERT INTO V5 VALUES (1, 2, 3, 4, 6);
- k) INSERT INTO V5 VALUES (1, 2, 4, 4, 5);