

Hauptdiplomklausur

Datenbanksysteme I

Wintersemester 2000/2001

Name:

Vorname:

Matrikelnummer:

Studienfach:

Wichtige Hinweise:

1. Prüfen Sie Ihr Klausurexemplar auf Vollständigkeit (16 Seiten).
2. Es sind keine Hilfsmittel zugelassen.
3. Die Klausur dauert 100 Minuten.
4. Jede Aufgabe ist auf dem zugehörigen Aufgabenblatt (und ggf. auf separaten Lösungsblättern) zu bearbeiten.
5. Vermerken Sie Ihren Namen und Ihre Matrikelnummer auf jedem Aufgaben- (bzw. Lösungsblatt). Blätter ohne Namens- und Matrikelnummerangabe werden nicht bewertet.
6. Das Deckblatt sowie alle Aufgabenblätter (evtl. Lösungsblätter) sind abzugeben.

	maximale Anzahl Punkte	erreichte Anzahl Punkte
Aufgabe 1	7,5	
Aufgabe 2	14	
Aufgabe 3	30	
Aufgabe 4	11,5	
Aufgabe 5	16	
Aufgabe 6	9	
Aufgabe 7	12	
	100	

1. (je 1,5 Punkte)

Geben Sie für folgende Aussagen an, ob sie wahr oder falsch sind.

Bei einer falschen Antwort werden Punkte abgezogen, die Gesamtpunktzahl kann jedoch nicht unter 0 Punkte sinken.

	wahr	falsch
Schwache Entities besitzen keinen eigenständigen Schlüssel. Erst in Kombination mit dem Schlüssel des übergeordneten Entities sind sie eindeutig voneinander unterscheidbar.	<input type="checkbox"/>	<input type="checkbox"/>
Das relationale, das objekt-orientierte und das verteilte Datenmodell sind alles logische Datenmodelle.	<input type="checkbox"/>	<input type="checkbox"/>
Zu den Optimierungstechniken beim Umwandeln von kanonischen Operatorbäumen gehört das Zusammenfassen von Selektionen und Kreuzprodukten zu Joins und das Verschieben von Selektionen nach unten im Operatorbaum.	<input type="checkbox"/>	<input type="checkbox"/>
Nach einer Einfügeoperation auf einer erweiterbaren Hashtabelle bei der das Verzeichnis verdoppelt wird, existieren genau zwei Behälter (Buckets) auf die jeweils nur ein Zeiger verweist.	<input type="checkbox"/>	<input type="checkbox"/>
In verteilten Datenbanksystemen können Relationen entweder horizontal oder vertikal fragmentiert werden, aber nicht horizontal und vertikal.	<input type="checkbox"/>	<input type="checkbox"/>

2. (a) (5 Punkte)

Gegeben sei ein Relationenschema $\mathcal{R}(A, B, C, D, E, F, G, H)$ mit folgenden funktionalen Abhängigkeiten $\mathcal{F}_{\mathcal{R}} = \{AC \rightarrow D, A \rightarrow F, D \rightarrow GH, B \rightarrow C\}$. In welcher höchsten Normalform befindet sich \mathcal{R} ? Begründen Sie Ihre Antwort kurz.

(b) (5 Punkte)

Gegeben sei ein Relationenschema $\mathcal{S}(A, B, C, D, E, F)$ mit folgenden funktionalen Abhängigkeiten $\mathcal{F}_{\mathcal{S}} = \{AD \rightarrow E, A \rightarrow B, B \rightarrow C, C \rightarrow ADF\}$. In welcher höchsten Normalform befindet sich \mathcal{S} ? Begründen Sie Ihre Antwort kurz.

(c) (4 Punkte)

Welche Gründe sprechen dafür ein Relationenschema, das in 1NF oder 2NF ist, in 3NF zu zerlegen und nicht in noch höhere Normalformen zu überführen?

3. Ein Partyservice speichert seine Daten in einem relationalem Datenbanksystem mit folgendem Schema:

- Einzelposten(ENr, Name, Menge, Preis, Vorrat)
- Buffetangebot(BNr, Name, PreisProPerson)
- besteht_aus(BNr, ENr, Anzahl)
- Einzelorder(Kunde, Lieferdatum, ENr, Anzahl)
- Buffetorder(Kunde, Lieferdatum, BNr, AnzahlPersonen)

In der Relation *Einzelposten* werden die Daten zu den angebotenen Lebensmitteln gespeichert. Beispiel für ein Tupel wäre (0815, "Apfelsaft", 0.2, 2.50, 200), was angibt, daß Apfelsaft in 0,2 Liter Flaschen zu je 2.50 angeboten wird und 200 dieser Flaschen in Vorrat sind. Ein Buffetangebot besteht aus mehreren Einzelposten. Die Angaben in *besteht_aus* beziehen sich darauf, wieviele Einzelposten pro Person für ein Buffetangebot vorgesehen sind (z.B. könnte das Buffet mit der Nummer 45 aus 1 x Krabbencocktail, 1 x Lachshäppchen und 2 x Brötchen bestehen). Kunden können komplette Buffets oder einzelne Posten bestellen.

Formulieren Sie folgende Anfragen in SQL:

(a) (4 Punkte)

Geben Sie Lieferdaten und Buffetnamen für alle Kunden aus, die Buffets für mehr als 50 Personen bestellt haben.

(b) (5 Punkte)

Geben Sie für jedes Buffet die Nummer, den Namen und die Gesamtanzahl der Einzelposten aus denen es besteht an. Sortieren Sie die Liste absteigend nach Anzahl der Einzelposten.

Einzelposten(ENr, Name, Menge, Preis, Vorrat)
Buffetangebot(BNr, Name, PreisProPerson)
besteht_aus(BNr, ENr, Anzahl)
Einzelorder(Kunde, Lieferdatum, ENr, Anzahl)
Buffetorder(Kunde, Lieferdatum, BNr, AnzahlPersonen)

(c) (9 Punkte)

Geben Sie für jedes Buffetangebot die Differenz des Preises zwischen *PreisProPerson* und der Summe der Einzelposten aus denen es besteht an, d.h. bestimmen Sie für jedes Buffetangebot wieviel man im Vergleich zur Bestellung der Einzelposten spart.

(d) (12 Punkte)

Bestimmen Sie, ob alle Order mit den bevorrateten Einzelposten erfüllt werden können, oder ob nachbestellt werden muß.

4. Folgende Einträge befinden sich in der Log-Datei eines Datenbanksystems:

1	T_1	-	BOT	-	0
2	T_2	-	BOT	-	0
3	T_3	-	BOT	-	0
4	T_3	P_C	$C:=C+100$	$C:=C-100$	3
5	T_2	P_B	$B:=B-20$	$B:=B+20$	2
6	T_1	P_A	$A:=A+30$	$A:=A-30$	1
7	T_2	-	COMMIT	-	5
8	T_1	P_A	$A:=A-30$	6	1
9	T_3	P_C	$C:=C-100$	4	3

(a) (4 Punkte)

Es gab einen Absturz im Datenbanksystem. Inzwischen wurde das Datenbanksystem wieder eingeschaltet und der Wiederanlauf wurde gestartet. Seitdem läuft das System ohne Absturz. In welcher Phase des Wiederanlaufs befindet sich das Datenbanksystem im Moment (wenn die Log-Datei wie oben abgebildet aussieht)? Begründen Sie Ihre Antwort kurz.

(b) (3,5 Punkte)

Rekonstruieren Sie die fehlenden Teile des untenstehenden Schedules aus der Log-Datei.

Schritt	T_1	T_2	T_3
1			
2	$r(A, a_1)$		
3			
4			
5			$r(C, c_3)$
6		$r(B, b_2)$	
7			$c_3 := c_3 + 100$
8		$b_2 := b_2 - 20$	
9			
10			
11	$a_1 := a_1 + 30$		
12			
13			
14	Absturz		

(c) (4 Punkte)

Angenommen beim Absturz eines Datenbanksystems befinden sich n Log-Records in der Log-Datei. Was ist die maximale Anzahl von Log-Records die sich in der Log-Datei nach erfolgreichem Abschluß des Wiederanlaufs befinden können? (Gehen Sie davon aus, daß das Datenbanksystem beim Wiederanlauf selbst wieder beliebig oft abstürzen kann.) Begründen Sie Ihre Antwort kurz.

5. In Abbildung 1 sehen Sie eine Historie, in der noch nicht alle Operationen in eine endgültige Reihenfolge gebracht wurden. Entscheiden Sie für die folgenden Eigenschaften, ob eine Historie mit den genannten Eigenschaften durch geeignete Anordnung der Operationen aus der Historie in Abbildung 1 gebildet werden kann. Falls ja, geben Sie diese Historie an. Falls nein, begründen Sie Ihre Antwort kurz.

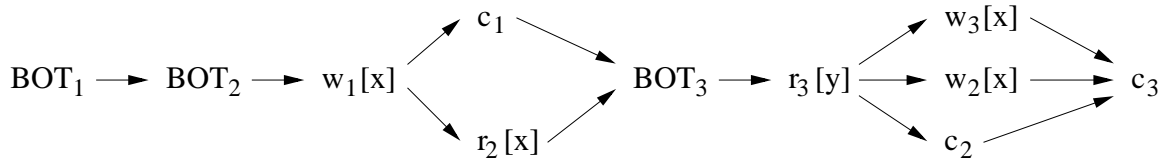


Abbildung 1: Unvollständig sortierte Historie

- (a) (4 Punkte)
 serialisierbar und strikt:

Ja, nämlich

BOT₁ BOT₂ w₁[x] BOT₃ r₃[y] c₃

Nein, denn

- (b) (4 Punkte)
 nicht serialisierbar und ACA, aber nicht strikt:

Ja, nämlich

BOT₁ BOT₂ w₁[x] BOT₃ r₃[y] c₃

Nein, denn

(c) (4 Punkte)

serialisierbar und rücksetzbar, aber nicht ACA:

Ja, nämlich

BOT₁ BOT₂ w₁[x] BOT₃ r₃[y] c₃

Nein, denn

(d) (4 Punkte)

nicht serialisierbar und strikt:

Ja, nämlich

BOT₁ BOT₂ w₁[x] BOT₃ r₃[y] c₃

Nein, denn

6. Für einen Zugriff auf einen bestimmten Block auf einer Festplatte werden drei Arbeitsvorgänge benötigt. Zunächst muß der Schreib-/Lesekopf auf die entsprechende Spur plaziert werden. Die dazu benötigte Zeit wird als *Seek Time* bezeichnet. Dann wird gewartet, bis sich durch die Rotation der Platte der gesuchte Block am Kopf vorbeibewegt. Die zu erwartende Verzögerung wird *Latenzzeit* genannt. Im dritten Schritt wird der Block gelesen (*Lesezeit*).

Moderne Festplattenlaufwerke haben mehr Blöcke auf den äußeren Spuren als auf den inneren. Da die Rotationszeit jedoch konstant ist, bedeutet dies, daß hintereinanderliegende Blöcke auf äußeren Spuren schneller eingelesen werden können. Die *Seek Time* und die *Latenzzeit* bleiben davon unbeeinflusst.

Wo auf der Platte (innere, mittlere, äußere Spuren) würden Sie die folgenden Dateien abspeichern, um die Zugriffszeiten zu optimieren? Begründen Sie Ihre Antwort kurz.

(a) (3 Punkte)

Häufige, zufällige Zugriffe auf kleine Dateien (z.B. Katalogrelationen).

(b) (3 Punkte)

Sequentielle Scans auf großen Dateien (z.B. Selektionen auf großen Relationen ohne Index).

(c) (3 Punkte)

Zufällige Zugriffe auf große Dateien über einen Index (z.B. Selektionen auf großen Relationen über einen Index). Wo speichern Sie die Dateien, wo den Index?

7. Betrachten Sie folgendes relationales Schema:

- Angestellte(PNr, Name, Alter, Einkommen)
- arbeitet_in(PNr, ANr)
- Abteilung(ANr, Budget, ManagerPNr)

(a) (2 Punkte)

Angenommen auf *Angestellte* wäre folgende Sicht definiert:

```
create view SeniorAng(SName, SAlter, SEinkommen) as
select Name, Alter, Einkommen
from   Angestellte
where  Alter > 50;
```

Nehmen Sie weiterhin an, daß die Sicht nicht materialisiert ist. Das bedeutet, daß eine Anfrage, die die Sicht benutzt, vor der Bearbeitung vom Datenbanksystem umgeschrieben wird, um die Sicht miteinzubeziehen. Wie wird die folgende Anfrage vor der eigentlichen Bearbeitung vom Datenbanksystem modifiziert?

```
select SName
from   SeniorAng
where  SEinkommen > 100.000;
```

(b) (2 Punkte)

Die oben definierte Sicht *SeniorAng* ist nicht updatefähig. Warum? Welche kleine Änderung können Sie an der Definition vornehmen, damit sie updatefähig wird?

(c) (2 Punkte)

Geben Sie eine weitere Sicht auf *Angestellte* an, die aus einem anderen Grund nicht updatefähig ist.

(d) (je 3 Punkte)

Betrachten Sie folgende Sicht auf dem oben definierten Relationenschema.

```
create view AbtInfo(ANr, Manager, AnzAng, SummeEink) as
select B.ANr, B.ManagerPNr, count(*), sum(A.Einkommen)
from   Angestellte A, arbeitet_in I, Abteilung B
where  A.PNr = I.PNr
and    I.ANr = B.Anr
group by B.ANr, B.ManagerPNr;
```

Diese Sicht ist ebenfalls nicht updatefähig. Grundsätzlich gilt aber, daß die gemäß SQL änderbaren Sichten eine Untermenge der theoretisch änderbaren Sichten darstellen. D.h. es gibt Sichtdefinitionen, bei denen theoretisch die eindeutige Propagierung von Änderungen auf die Basisrelationen möglich wäre, SQL diese Änderungen aber dennoch ausschließt.

- i. Geben Sie eine Update-Operation auf *AbtInfo* an, die theoretisch auf die Basisrelationen abbildbar wäre. Geben Sie für diese Operation an, wie die entsprechenden Update-Operationen auf den Basisrelationen aussehen.

- ii. Geben Sie eine Update-Operation auf *AbtInfo* an, die überhaupt nicht abbildbar ist. Begründen Sie Ihre Antwort kurz.