

Gruppe A

Bitte tragen Sie **sofort** und **leserlich** Namen, Studienkennzahl und Matrikelnummer ein und legen Sie Ihren Studentenausweis bereit.

PRÜFUNG AUS DATENBANKSYSTEME VO 181.146			16. 12. 2005
Kennnr.	Matrikelnr.	Familienname	Vorname

Arbeitszeit: 120 Minuten. Aufgaben sind auf den Angabebältern zu lösen; Zusatzblätter werden nicht gewertet.

Aufgabe 1:

(24)

Gegeben ist die folgende Historie von Transaktionen:

Schritt	T_1	T_2	T_3	Log: [LSN, TA, PageID, Redo, Undo, PrevLSN] or ⟨LSN, TA, PageID, Redo, PrevLSN, UndoNextLSN⟩
1	BOT			[#1, T_1 , BOT, 0]
2		BOT		[#2, T_2 , BOT, 0]
3			BOT	[#3, T_3 , BOT, 0]
4	$r(A, a_1)$			
5	$w(A, 2 * a_1)$			[#4, T_1 , P_A , A+=50, A-=50, #1]
6		$r(A, a_2)$		
7		$w(A, a_2 + 100)$		[#5, T_2 , P_A , A+=100, A-=100, #2]
8		$w(B, a_2 + 200)$		[#6, T_2 , P_B , B+=200, B-=200, #5]
9			$r(B, b_3)$	
10	$w(C, a_1)$			[#7, T_1 , P_C , C-=100, C+=100, #4]
11			$r(C, c_3)$	
12			$w(C, b_3 + c_3)$	[#8, T_3 , P_C , C+=300, C-=300, #3]
13		$r(A, a_2)$		
14	commit			[#9, T_1 , commit, #7]
15				⟨#10, T_3 , P_C , C-=300, #8, #3⟩
16				⟨#11, T_2 , P_B , B-=200, #6, #5⟩
17				⟨#12, T_2 , P_A , A-=100, #11, #2⟩
18				⟨#13, T_3 , (BOT), #10, 0⟩
19				⟨#14, T_2 , (BOT), #12, 0⟩

(a) Nehmen Sie an, dass unmittelbar nach Zeile 14 zunächst die Transaktion T_2 und dann die Transaktion T_3 mittels commit beendet wird. Die resultierende Historie ist serialisierbar. Ist sie auch strikt? ☐ ja ☒ nein

Geben Sie eine mögliche Reihenfolge der Serialisierung an.

T_1 vor T_2 vor T_3

(b) Angenommen in Zeile 14 würde an Stelle des commit der Transaktion T_1 ein rollback der Transaktion T_1 erfolgen. Würde dieses Rücksetzen von T_1 ein kaskadierendes Rücksetzen der anderen Transaktionen erfordern?

Kaskadierendes Rücksetzen von T_2 :

☒ ja ☐ nein

Kaskadierendes Rücksetzen von T_3 :

☒ ja ☐ nein

(c) Zu Beginn ist der relevante Datenbestand in der Datenbank $A = 50$, $B = 100$, und $C = 150$. Nehmen Sie an, dass unmittelbar nach Zeile 14 ein System-Absturz mit anschließendem Wiederanlauf passiert. Tragen Sie die entsprechenden Log-Einträge zu dieser Historie in die rechte Spalte der obigen Tabelle ein. Dabei sind Undo/Redo-Einträge *relativ* zum Datenbestand anzugeben. Geben Sie in den Zeilen 15ff. die Log-Einträge für die Recovery an.

Die Werte von A , B und C nach Abschluss der Recovery sind

$A : 100$	$.....$	$; B : 100$	$.....$	$; C : 50$	$.....$
-----------	---------	-------------	---------	------------	---------

.

Aufgabe 2:

(16)

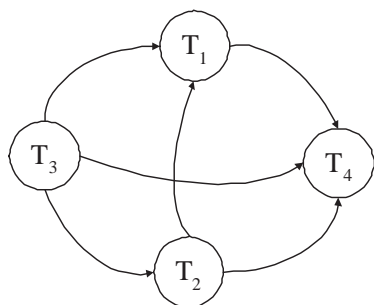
(a) Eine Historie sei gegeben durch folgende Folge von Elementaroperationen: $r_1(A)$ vor $r_2(B)$ vor $r_3(A)$ vor $w_2(B)$ vor $w_3(C)$ vor $r_4(D)$ vor $r_1(B)$ vor $r_1(C)$ vor $r_2(C)$ vor $r_3(C)$ vor $w_4(C)$ vor c_1 vor c_2 vor c_3 vor c_4 .

Zeichnen Sie ins erste Kästchen den Serialisierbarkeitsgraphen für die Transaktionen T_1 , T_2 , T_3 und T_4 . Verwenden Sie dabei folgende Konvention: eine Kante $T_i \rightarrow T_j$ bedeutet, dass in einer äquivalenten seriellen Historie die “Transaktion T_i vor T_j ” ausgeführt werden muss (vgl. VO-Folien bzw. Kemper-Buch).

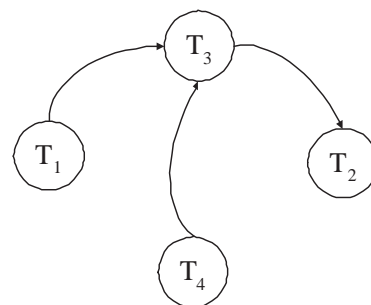
(b) Betrachten Sie die folgende Folge von Sperranforderungen, wobei “lockS_{*i*}(O)” (bzw. “lockX_{*i*}(O)”) bedeutet, dass die Transaktion T_i eine Lesesperre (bzw. eine Schreibsperre) auf das Datenobjekt O anfordert: lockS₁(A) vor lockX₂(B) vor lockX₃(C) vor lockS₁(C) vor lockS₃(B) vor lockS₄(C) vor lockS₂(A).

Zeichnen Sie ins zweite Kästchen den Wartegraphen unter der Annahme, dass zum momentanen Zeitpunkt keine der erhaltenen Sperren wieder zurückgegeben wurde. Verwenden Sie dabei folgende Konvention: eine Kante $T_i \rightarrow T_j$ bedeutet “Transaktion T_i wartet auf die Freigabe einer Sperre durch T_j ” (vgl. VO-Folien bzw. Kemper-Buch).

(a) Serialisierbarkeitsgraph:



(b) Wartegraph:



(c) Geben Sie für den Serialisierbarkeitsgraphen aus (a) eine mögliche topologische Sortierung an:

$T\ 3$	$.....$	vor	$T\ 2$	$.....$	vor	$T\ 1$	$.....$	vor	$T\ 4$	$.....$
--------	---------	-----	--------	---------	-----	--------	---------	-----	--------	---------

Aufgabe 3:

(20)

Eine Bibliotheks-Datenbank enthalte die Relationen

Buecher(BuchNr, Titel, Autor, Kategorie, Jahr) (kurz b),

Leser(LeserNr, Name, Alter, Wohnort) (kurz l) und

Ausleihen(LeihNr, BuchNr, LeserNr, Datum) (kurz a).

Nehmen Sie an, dass $|b| = 9000$, $|l| = 5000$, und $|a| = 300000$. Für die durchschnittlichen Tupelgrößen von b , l und a sind die Werte 100, 80 und 20 Bytes anzunehmen. Nehmen Sie weiters an, dass pro Seite 1000 Bytes an Nutzinformation gespeichert werden können und dass die Hauptspeicher-Puffergröße 24 Seiten beträgt. Es ist die Anfrage

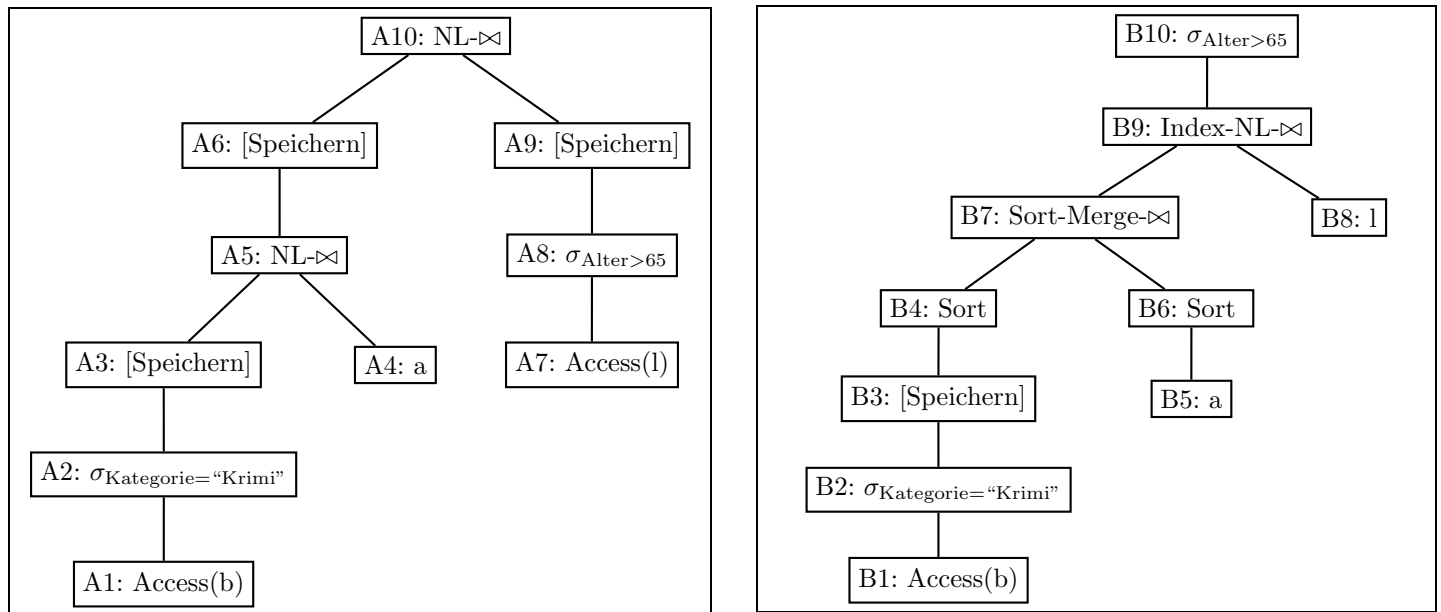
```

select *
from Buecher b, Leser l, Ausleihen a
where b.Kategorie = "Krimi"
and l.Alter > 65
and b.BuchNr = a.BuchNr
and l.LeserNr = a.LeserNr

```

auszuführen (d.h. gesucht sind Informationen über ältere Leser, die sich einen Krimi ausgeliehen haben). Es sind die Selektivitäten $Sel_{b/a} = 1/9000 = 0.0001111$, $Sel_{l/a} = 1/5000 = 0.0002$, $Sel_{b.Kategorie} = 0.1$ und $Sel_{l.Alter > 65} = 0.25$ anzunehmen. Für die Primärschlüssel der Relationen b , l und a sei jeweils ein Hash-Index vorhanden. Nehmen Sie an, dass das Auslesen eines einzelnen Tupels mit einem Hash-Index durchschnittliche Kosten von 1.5 Page I/O erfordert.

Für diese Anfrage sind die Operator-Bäume für 2 Auswertungspläne gegeben: Plan A realisiert alle Joins mittels Block Nested Loop Joins, während Plan B die Joins mittels Sort-Merge Join bzw. Index Nested Loop Join realisiert. Nehmen Sie bei den Block Nested Loop Joins an, dass die äußere Relation jeweils links im Baum steht.



(a) Berechnen Sie für jeden Knoten im Operatorbaum des Auswertungsplans A eine Abschätzung für die die Anzahl der Tupel im Resultat, die Tupelgröße, die Anzahl der Seiten im Resultat, und die Kosten (Page I/O). Für Joinoperationen ist auch noch die passende Kostenformel anzugeben. Tragen Sie Ihre Berechnungen in die folgende Tabelle ein.

Knoten# i	Anzahl Tupel T_i	Tupel- größe g_i	Anzahl Seiten b_i	Kostenformel	Kosten (Page I/O)
A1	9000	100	900	-	900
A2	900	100	90	-	0
A3	900	100	90	-	90
A4	300000	20	6000	-	0
A5	30000	120	3600	$b_3 + 1 + \lceil b_3/22 \rceil * (b_4 - 1)$	30086
A6	30000	120	3600	-	3600
A7	5000	80	400	-	400
A8	1250	80	100	-	0
A9	1250	80	100	-	100
A10	7500	200	1500	$b_6 + 1 + \lceil b_6/22 \rceil * (b_9 - 1)$	19837

Kosten insgesamt (Page I/O):

$$900 + 90 + 30086 + 3600 + 400 + 100 + 19837 = 55013 \dots\dots$$

(b) Berechnen Sie für jeden Knoten im Operatorbaum des Auswertungsplans B eine Abschätzung für die die Anzahl der Tupel im Resultat, die Tupelgröße, die Anzahl der Seiten im Resultat, und die Kosten (Page I/O). Für das Sortieren und für die Joinoperationen ist auch noch die passende Kostenformel anzugeben. Tragen Sie Ihre Berechnungen in die Tabelle auf der folgenden Seite ein.

Knoten# i	Anzahl Tupel T_i	Tupel- größe g_i	Anzahl Seiten b_i	Kostenformel	Kosten (Page I/O)
B1	9000	100	900	-	900
B2	900	100	90	-	0
B3	900	100	90	-	90
B4	900	100	90	$2 * b_3 * (1 + I)$ mit $I = \log_{23}(\lceil b_3/24 \rceil) = 1$	360
B5	300000	20	6000	-	0
B6	300000	20	6000	$2 * b_5 * (1 + I)$ mit $I = \log_{23}(\lceil b_5/24 \rceil) = 2$	36000
B7	30000	120	3600	$b_4 + b_6$	6090
B8	5000	80	400	-	0
B9	30000	200	6000	$b_7 + 1.5 * T_7$	48600
B10	7500	200	1500	-	0

Kosten insgesamt (Page I/O):

900 + 90 + 360 + 36000 + 6090 + 48600 = 92040

Aufgabe 4:

(20)

Kreuzen Sie an, ob die folgenden Aussagen wahr oder falsch sind.

1. Mengenwertige Attribute in einer objektrelationalen Datenbank können für eine effizientere Implementierung von N:M-Beziehungen als in reinen relationalen Datenbanken verwendet werden. wahr ☒ falsch ☐
2. In einem verteilten DBMS benötigt eine einzelne Station unter Umständen eine Kommunikation mit anderen Stationen, um nach einem Absturz die "winner" und "loser" Transaktionen zu erkennen. Die erforderlichen Redo- und Undo-Operationen sind dann aber allein mit Hilfe der lokalen Log-Datei ermittelbar. wahr ☒ falsch ☐
3. Jede Anfrage der relationalen Algebra lässt sich auch als Datalog-Programm ohne Negation und ohne Rekursion darstellen. wahr ☐ falsch ☒
4. Die Privilegien, die man einem Benutzer beim Role-based Access Protocol (RBAC) geben kann, die könnte man ihm auch beim Discretionary Access Protocol (DAC) ohne Rollen-Erweiterung geben. wahr ☒ falsch ☐
5. Beim Discretionary Access Protocol (DAC) hat ein Benutzer immer das Recht, die ihm erteilten Rechte auch an andere weiterzugeben. wahr ☐ falsch ☒
6. Der Abhängigkeitsgraph eines Datalog Programms gibt Auskunft darüber, ob für die rekursiven Regeln ein Fixpunkt existiert. wahr ☐ falsch ☒
7. Nehmen Sie an, dass in einem verteilten DBMS das globale Relationenschema $R(\underline{ABCDE})$ (d.h.: A ist der Primärschlüssel) in die vertikalen Fragmente $R(AB)$, $R(BCD)$ und $R(DE)$ unterteilt wurde. Diese Fragmentierung ist rekonstruierbar. wahr ☐ falsch ☒
8. Ein wesentliches Ziel der Standardisierungsbemühungen von ODMG war die Umsetzung einheitlicher objektrelationaler Erweiterungen in SQL:1999. wahr ☐ falsch ☒

9. Beim Zweiphasen-Commit-Protokoll können die Agenten mittels Timer-Überwachung – durch geeignete Wahl des Timer-Intervalls – verhindern, dass sie bei einem Absturz des Koordinators blockiert werden. wahr ☐ falsch ☒
10. Die transitive Hülle einer binären Relation lässt sich in Datalog ohne Negation aber mit Rekursion darstellen. wahr ☒ falsch ☐

(Pro korrekter Antwort 2 Punkte, **pro inkorrektter Antwort -2 Punkte**, insgesamt mindestens 0 Punkte)

Aufgabe 5:

(20)

Die Datenbank einer Autoverleih-Firma enthalte folgende Relationen:

Autos(AutoNr, Marke, Kategorie, PS, Miete) (kurz a),
 Verkäufer(PersNr, Name, Gehalt, Adresse) (kurz v) und
 Mieten(RechnungNr, AutoNr, KundenNr, VerkäuferNr, Datum) (kurz m).

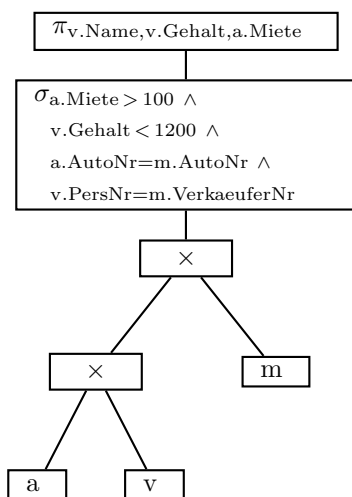
Nehmen Sie an, dass $|a| = 400$, $|v| = 50$, und $|m| = 100000$. Es ist die Anfrage

```
select v.Name, v.Gehalt, a.Miete
from Autos a, Verkäufer v, Mieten m
where a.Miete > 100
and v.Gehalt < 1200
and a.AutoNr = m.AutoNr
and v.PersNr = m.VerkaeferNr
```

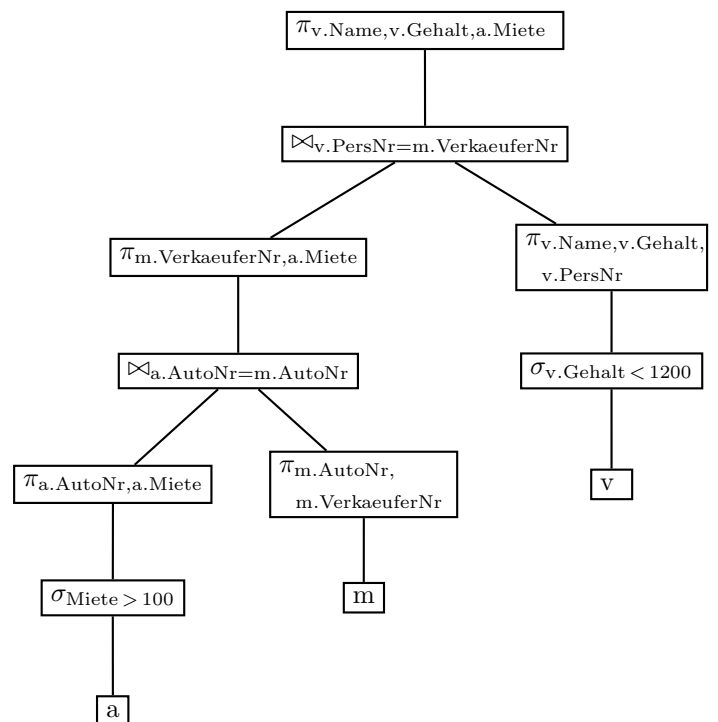
auszuführen (d.h. gesucht sind Informationen über Verkäufer, die teure Autos vermieten, selbst aber eher schlecht verdienen). Es sind die Selektivitäten $Sel_{a/m} = 1/400 = 0.0025$, $Sel_{v/m} = 1/50 = 0.02$, $Sel_{a.Miete > 100} = 0.1$ und $Sel_{v.Gehalt < 1200} = 0.3$ anzunehmen.

- (a) Zeichnen Sie ins erste Kästchen den Operator-Baum für die kanonische Übersetzung.
- (b) Zeichnen Sie ins zweite Kästchen den Operator-Baum für den optimierten algebraischen Ausdruck. Wenden Sie für die Optimierung folgende Heuristiken an:
- Selektionen so weit wie möglich nach unten verschieben,
 - Attribute, die nicht mehr benötigt werden, möglichst früh wegprojizieren,
 - Kreuzprodukte durch Joins ersetzen,
 - Join-Reihenfolge so, dass der erste Join ein möglichst kleines Zwischenergebnis liefert.

(a) Kanonische Übersetzung:



(b) Optimierter Ausdruck:



Gesamtpunkte: 100