

## 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			13. 10. 2006
Kennnr.	Matrikelnr.	Familienname	Vorname

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

### Aufgabe 1:

(24)

Der Wiener Tiergarten Schönbrunn und der Innsbrucker Alpenzoo wollen gemeinsam eine verteilte Datenbank ihrer Tiere erstellen. In dieser Datenbank sind folgende Tabellen enthalten:

Tiere(name, art, gewicht, ort, geb, erworben)

Tierarten(art, gattung, platzbedarf, futter)

Die Tabelle Tiere wird (vertikal) fragmentiert in folgende 2 Tabellen:

TiereVerwaltung(name, ort, geb, erworben) und

TiereBiologie(name, art, gewicht).

Außerdem wird die Tabelle TiereVerwaltung(name, ort, geb, erworben) nach dem Attribut "ort" (horizontal) weiter fragmentiert in die zwei Tabellen TiereVerwaltungWien und TiereVerwaltungInnsbruck. Es ist die Anfrage

select name, ort, geb, gewicht

from Tiere t, Tierarten a

where t.gewicht < 30 and a.gattung = 'Raubkatze' and t.art = a.art

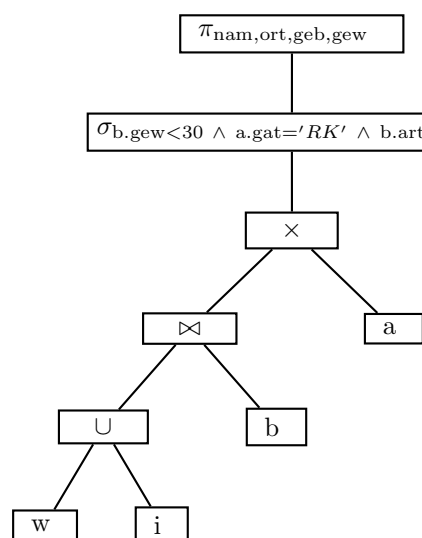
auszuführen (d.h. Informationen über Raubkatzen mit eher geringem Gewicht).

(a) Zeichnen Sie ins erste Kästchen den Operator-Baum für die kanonische Übersetzung. Verwenden Sie für die 3 Fragmente der Tiere-Tabelle sowie für die Tierarten-Tabelle folgende Abkürzungen: **b** (TiereBiologie), **w** (TiereVerwaltungWien), **i** (TiereVerwaltungInnsbruck) und **a** (Tierarten). Außerdem können Sie den String 'Raubkatze' mit 'RK' abkürzen. Auch die Attributnamen können Sie jeweils auf die ersten 3 Buchstaben abkürzen, z.B. gew statt gewicht).

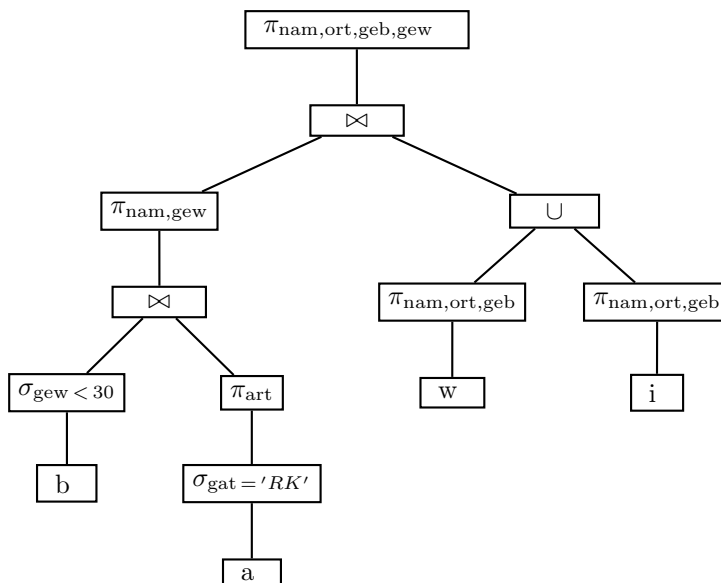
(b) Zeichnen Sie ins zweite Kästchen den Operator-Baum für den optimierten algebraischen Ausdruck. Wenden Sie für die Optimierung folgende Heuristiken an:

- Selektionen so weit wie möglich nach unten verschieben,
- Attribute, die nicht mehr benötigt werden, möglichst früh wegprojizieren.
- Kreuzprodukte durch Joins ersetzen,
- Join-Reihenfolge: Bei einem der Joins ist das Ergebnis des Joins eine Teilmenge eines der Operanden (d.h.: es gilt  $R \bowtie S \subseteq R$  oder  $R \bowtie S \subseteq S$ ). Führen Sie diesen Join zuerst aus.

#### (a) Kanonische Übersetzung:



#### (b) Optimierter Ausdruck:



## Aufgabe 2:

(20)

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			$r(B, b_3)$	
6			$w(A, b_3 * 2)$	[#4, $T_3$ , $P_A$ , A+=300, A-=300, #3] ....
7		$r(A, a_2)$		
8		$w(B, a_2 - 100)$		[#5, $T_2$ , $P_B$ , B+=100, B-=100, #2] ....
9	$w(A, a_1 + 100)$			[#6, $T_1$ , $P_A$ , A-=200, A+=200, #1] ....
10		$r(A, a_2)$		
11			$w(C, b_3 - 50)$	[#7, $T_3$ , $P_C$ , C+=50, C-=50, #4] .....
12		$w(C, 2 * a_2)$		[#8, $T_2$ , $P_C$ , C+=250, C-=250, #5] ....
13	commit			[#9, $T_1$ , commit, #6] .....
14			commit	[#10, $T_3$ , commit, #7] .....
15		rollback ...		[#11, $T_2$ , rollback, #8] .....
16				⟨#12, $T_2$ , $P_C$ , C-=250, #11, #5⟩ .....
17				⟨#13, $T_2$ , $P_B$ , B-=100, #12, #2⟩ .....
18				⟨#14, $T_2$ , (BOT), #13, 0⟩ .....
19				.....

(a) Nehmen Sie an, dass in Zeile 15 auch die Transaktion  $T_2$  mittels commit beendet wird. Ist die resultierende Historie serialisierbar?

☐ ja ☒ nein

Wenn ja, in Reihenfolge  $T - \dots$  vor  $T - \dots$  vor  $T - \dots$ .

Wenn nein: die Historie wird durch das Streichen von zumindest 1 ..... Operationen serialisierbar.

(b) Führen Sie nun – unabhängig davon, ob die Historie serialisierbar ist – in Zeile 15 ein *rollback* für  $T_2$  durch.

Zu Beginn ist der relevante Datenbestand in der Datenbank  $A = 100$ ,  $B = 200$  und  $C = 100$ . Tragen Sie nun das Recovery-Log zu dieser Historie (mit rollback von  $T_2$  in Zeile 15) in die rechte Spalte der obigen Tabelle ein. Dabei sind Undo/Redo-Einträge *relativ* zum Datenbestand anzugeben. Geben Sie in den Zeilen 15 ff. die Log-Einträge für die Recovery an.

Die Werte von  $A$ ,  $B$  und  $C$  nach dem rollback von  $T_2$  sind  $A : 200$  .....;  $B : 200$  .....;  $C : 150$  ......

**Aufgabe 3:**

(20)

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

1. Nehmen Sie an, dass ein Index mit Hilfe von statischem Hashing erzeugt wurde und dass in diesem Index die TIDs (und nicht die Datensätze selbst) gespeichert sind. Dann sind bei einer Punkt-Anfrage mit diesem Index im Idealfall 1.2 Seitenzugriffe notwendig. wahr ☐ falsch ☒
2. Betrachten Sie folgende Historie, bei der nur BOT (= Transaktionsbeginn),  $c_i$  (= Commit) und Sperranforderungen (aber keine Datenzugriffe oder Informationen über Warten bzw. Wecken einer Transaktion) gegeben sind:  $BOT_1, BOT_2, lockX_1(A), lockX_2(B), lockS_1(B), c_2, c_1$ . Diese Historie ist beim Zeitstempel-Verfahren mit wait-die Strategie möglich. wahr ☒ falsch ☐
3. Nehmen Sie an, dass beim Zweiphasen-Commit-Protokoll einer der Agenten abstürzt. Dann kann dieser Agent beim Wiederanlauf mit Hilfe der eigenen Log-Datei eigenständig (also ohne Rückfrage bei einem anderen Agenten oder beim Koordinator) entscheiden, ob eine Transaktion erfolgreich war. wahr ☐ falsch ☒
4. Betrachten Sie drei Relationen  $R(AB), S(AB)$  und  $T(BC)$ . Dann gilt auf jeden Fall folgende Gleichheit:  
 $\pi_C((R \cap S) \bowtie T) = \pi_C([R \bowtie T] \bowtie [S \bowtie \pi_B(T)])$ . wahr ☒ falsch ☐
5. Betrachten Sie drei Relationen  $U(AB), V(BC)$  und  $W(CD)$ . Dann gilt auf jeden Fall folgende Gleichheit:  
 $(U \bowtie V) \bowtie W = (U \times W) \bowtie V$  wahr ☒ falsch ☐
6. Nehmen Sie an, dass eine Relation  $S$  10000 Seiten umfasst und dass die Puffergröße 32 beträgt. Dann ist bei einem Hash-Join von  $R$  mit  $S$  in der Build-Phase möglicherweise aber nicht notwendigerweise ein Re-hashing von  $S$  erforderlich. wahr ☒ falsch ☐
7. Wenn der Abhängigkeitsgraph eines Datalog-Programms maximal 1 Zyklus enthält, dann lässt sich dieses Datalog-Programm auch mittels relationaler Algebra ausdrücken. wahr ☐ falsch ☒
8. Falls das Rückrollen von Datenbank-Änderungen durch einen Systemabsturz unterbrochen wird, muss das Datenbanksystem beim Wiederanlauf in der Lage sein, das Rückrollen abzuschließen. wahr ☒ falsch ☐
9. Die Historie  $r_1(A), r_2(A), w_1(A), w_2(B), w_1(B), c_1, c_2$  ist serialisierbar. wahr ☒ falsch ☐
10. Die Historie  $r_1(A), w_1(A), r_2(A), w_2(A), r_1(A), c_1, c_2$  ist beim Zwei-Phasen Sperrprotokoll aber nicht beim strikten Zwei-Phasen Sperrprotokoll möglich. wahr ☐ falsch ☒

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

**Aufgabe 4:**

(20)

Eine Datenbank von Briefmarkensammlern enthält folgende Relationen:

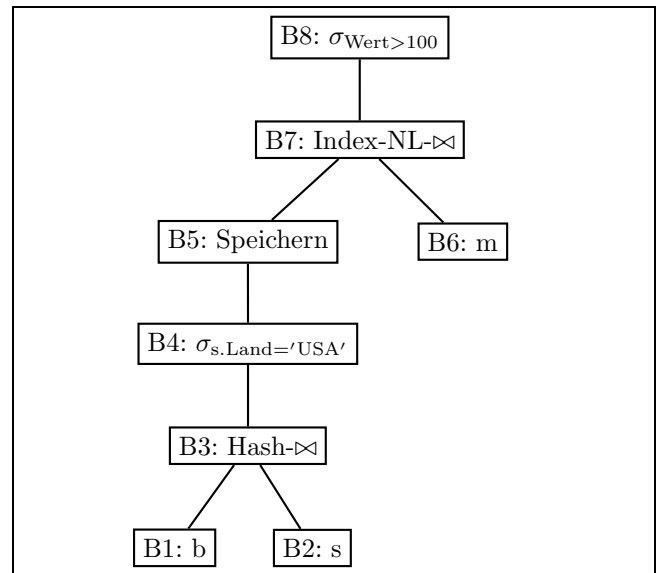
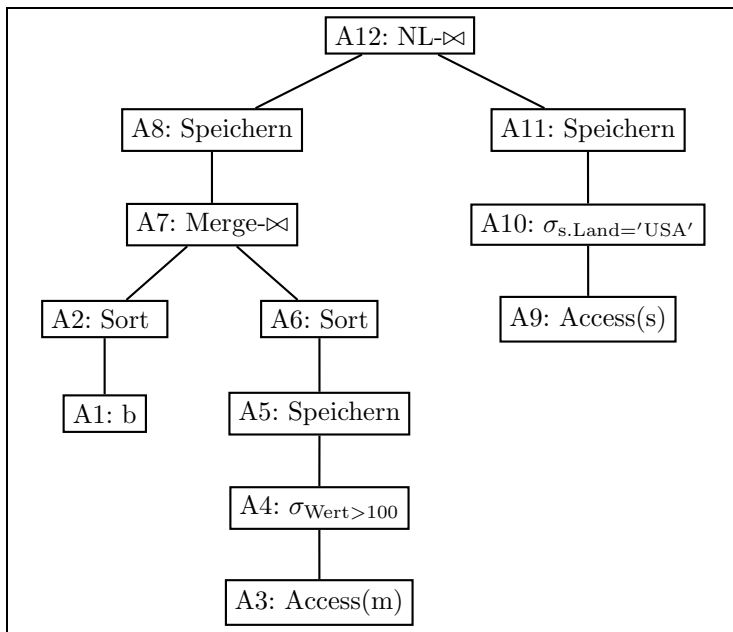
Sammler(SammlerNr, Name, Land, Geburtsdatum) (kurz  $s$ ),  
 Marken(MarkenNr, Name, Land, Jahr, Wert) (kurz  $m$ ) und  
 besitzt(SammlerNr, MarkenNr, Anzahl) (kurz  $b$ ).

Nehmen Sie an, dass  $|s| = 40000$ ,  $|m| = 50000$ , und  $|b| = 1000000$ . Für die durchschnittlichen Tupelgrößen von  $s$ ,  $m$  und  $b$  sind die Werte 100, 120 und 20 Bytes anzunehmen. Nehmen Sie weiters an, dass pro Seite 2000 Bytes an Nutzinformation gespeichert werden können und dass die Hauptspeicher-Puffergröße 128 Seiten beträgt. Es ist die Anfrage

```
select *
from Sammler s, Marken m, besitzt b
where s.SammlerNr = b.SammlerNr
and m.MarkenNr = b.MarkenNr
and s.Land = 'USA'
and m.Wert > 100;
```

auszuführen (d.h. gesucht sind Informationen über teure Briefmarken, die sich im Besitz von Briefmarkensammlern in den USA befinden). Es sind die Selektivitäten  $Sel_{m/b} = 1/50000 = 0.00002$ ,  $Sel_{s/b} = 1/40000 = 0.000025$ ,  $Sel_{s, Land='USA'} = 0.1$  und  $Sel_{m, Wert > 100} = 0.1$  anzunehmen. Für die Primärschlüssel der Relationen  $m$  und  $s$  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 im linken Kästchen und Plan B im rechten Kästchen. Nehmen Sie bei den Block Nested Loop Joins an, dass die äußere Relation jeweils links im Baum steht – unabhängig davon, ob dies optimal ist oder nicht. Bei der Berechnung der benötigten Seiten zum Speichern einer Relation dürfen Sie vereinfachend annehmen, dass die Tupel nicht unbedingt vollständig auf einer Seite Platz haben müssen.



(a) Berechnen Sie für jeden Knoten im Operatorbaum des Auswertungsplans A eine Abschätzung für 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 Joinoperationen ist auch noch die passende Kostenformel anzugeben. Tragen Sie Ihre Berechnungen in folgende Tabelle ein.

Knoten# $i$	Anzahl Tupel $T_i$	Tupel- größe $g_i$	Anzahl Seiten $b_i$	Kostenformel	Kosten (Page I/O)
A1	1000000	20	10000 .....	-	0 .....
A2	1000000 ..	20 .....	10000 .....	$2 * b_1 * (1 + I)$ mit ..... $I = \log_{127}(\lceil b_1/128 \rceil) = 1$ .....	40000 .....
A3	50000	120	3000 .....	-	3000 .....
A4	5000 .....	120 .....	300 .....	-	0 .....
A5	5000 .....	120 .....	300 .....	-	300 .....
A6	5000 .....	120 .....	300 .....	$2 * b_5 * (1 + I)$ mit ..... $I = \log_{127}(\lceil b_5/128 \rceil) = 1$ .....	1200 .....
A7	100000 ....	140 .....	7000 .....	$b_2 + b_6$ .....	10300 .....
A8	100000 ....	140 .....	7000 .....	-	7000 .....
A9	40000	100	2000 .....	-	2000 .....
A10	4000 .....	100 .....	200 .....	-	0 .....
A11	4000 .....	100 .....	200 .....	-	200 .....
A12	10000 ....	240 .....	1200 .....	$b_8 + 1 + \lceil b_8/126 \rceil * (b_{11} - 1)$ ....	18145 .....

Kosten insgesamt (Page I/O):

$$40000 + 3000 + 300 + 1200 + 10300 + 7000 + 2000 + 200 + 18145 = 82145 \dots\dots\dots$$

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

Knoten# $i$	Anzahl Tupel $T_i$	Tupel- größe $g_i$	Anzahl Seiten $b_i$	Kostenformel	Kosten (Page I/O)
B1	1000000	20	10000 .....	-	0 .....
B2	40000	100	2000 .....	-	0 .....
B3	1000000 ..	120 .....	60000 .....	$3 * (b_1 + b_2)$ .....	36000 .....
B4	100000 ....	120 .....	6000 .....	-	0 .....
B5	100000 ....	120 .....	6000 .....	-	6000 .....
B6	50000	120	3000 .....	-	0 .....
B7	100000 ....	240 .....	12000 .....	$b_5 + 1.5 * T_5$ .....	156000 ....
B8	10000 .....	240 .....	1200 .....	-	0 .....

Kosten insgesamt (Page I/O):

36000 + 6000 + 156000 = 198000 .....

**Aufgabe 5:** (16)

(a) Eine Historie sei gegeben durch folgende Folge von Elementaroperationen:  $r_4(A)$  vor  $r_1(A)$  vor  $w_1(B)$  vor  $r_3(B)$  vor  $w_3(C)$  vor  $r_4(B)$  vor  $w_4(D)$  vor  $r_2(B)$  vor  $r_2(D)$  vor  $r_2(E)$  vor  $r_4(C)$  vor  $w_3(E)$  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 “ $\text{lockS}_i(O)$ ” (bzw. “ $\text{lockX}_i(O)$ ”) bedeutet, dass die Transaktion  $T_i$  eine Lesesperre (bzw. eine Schreibsperr) auf das Datenobjekt  $O$  anfordert:  $\text{lockS}_1(A)$  vor  $\text{lockS}_2(B)$  vor  $\text{lockX}_3(C)$  vor  $\text{lockS}_1(C)$  vor  $\text{lockX}_4(D)$  vor  $\text{lockX}_2(A)$  vor  $\text{lockS}_3(D)$  vor  $\text{lockX}_4(B)$ .

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) Ist die Historie aus (a) serialisierbar?    ☐ ja    ☒ nein

Wenn ja, in Reihenfolge  $T - \dots$  vor  $T - \dots$  vor  $T - \dots$ .

Wenn nein: die Historie wird durch das Streichen von zumindest 1 .....