Dr. N. Ritter
Fachbereich Informatik
Arbeitsgruppe Datenbanken und Informationssysteme
Universität Kaiserslautern

Übungsblatt 8

für die Übung am Donnerstag, 8. Februar 2001, 15.30 Uhr in 36/265

Unterlagen zur Vorlesung: „http://wwwdbis.informatik.uni-kl.de/courses/DBS/“

Aufgabe 1: Serialisierbarkeit

Gegeben sind die drei Transaktionen T1, T2 und T3:
T1: x := x*2; z := z+1
T2: y := y*2; x := x+1
T3: z := z*2; y := y+1

Die Anfangswerte vor Start der drei Transaktionen seinen x=10, y=20 und z=30. Nach Beendigung der parallelen Ausführung aller drei Transaktionen gelte x=22, y=41 und z=62.

a) Geben Sie an, in welcher Reihenfolge die einzelnen Aktionen der Transaktionen T1 bis T3 ausgeführt wurden. Sind die Transaktionen serialisierbar?

b) Was passiert, wenn alle drei Transaktionen zunächst ihre erste Aktion jeweils abschließen, und anschließend die zweite Aktion ausführen?

Aufgabe 2: Serialisierbarkeit von Schedules 86

Gegeben sind die folgenden 3 Transaktionen:
T1: BOT r(x) r(y) w(x) w(y) EOT
T2: BOT r(x) r(y) r(z) EOT
T3: BOT r(x) r(z) w(z) EOT

Diese drei Transaktionen laufen folgendermaßen parallel ab:

T1	T2	T3
BOT		
r(x)		
	BOT	
	r(x)	
r(y)		BOT
		r(x)
w(x)		
w(y)		
EOT	r(y)	
		r(z)
		w(z)
		EOT
	r(z)	
	EOT	

a) Ist der Schedule serialisierbar?

b) Welche Operationen müßten vertauscht werden, um ihn serialisierbar zu machen?

Aufgabe 3: Serialisierbarkeit von Schedules 84

Gegeben seien die Transaktionen $T_1 = \langle r_1(a), w_1(a), w_1(b)\rangle$ und $T_2 = \langle r_2(a), r_2(b), w_2(a)\rangle$. T_1 lese den Wert a, erhöhe diesen um 100 und schreibe ihn nach a zurück. Außerdem schreibe T_1 den alten Wert von a nach b. T_2 erhöhe den gelesenen Wert von a um 200 und lese zusätzlich b. Der Anfangswert von a sei 10, der von b 5. Betrachtet werden die folgenden Schedules:

	Endwert von	
	a	b
$S1 = \langle r_1(a), w_1(a), w_1(b), r_2(a), r_2(b), w_2(a)\rangle$	310	10
$S2 = \langle r_2(a), r_2(b), w_2(a), r_1(a), w_1(a), w_1(b)\rangle$	310	210
$S3 = \langle r_1(a), r_2(a), r_2(b), w_1(a), w_1(b), w_2(a)\rangle$	?	?
$S4 = \langle r_1(a), w_1(a), r_2(a), w_1(b), r_2(b), w_2(a)\rangle$	?	?
$S5 = \langle r_1(a), r_2(a), r_2(b), w_1(a), w_2(a), w_1(b)\rangle$	?	?
$S6 = \langle r_1(a), r_2(a), w_2(a), r_2(b), w_1(a), w_1(b)\rangle$	?	?

a) Ermitteln Sie alle Endwerte von a und b.

b) Welche Schedules sind seriell?

c) Welche Schedules sind serialisierbar?

Aufgabe 4: Serialisierbarkeit von Schedules 85

Überprüfen Sie mit Hilfe von Abhängigkeitsgraphen die folgenden beiden Schedules von je drei Transaktionen $T_i = \langle r_i(x), w_i(x)\rangle$ (i=1,2,3) auf Serialisierbarkeit und geben Sie ggf. eine äquivalente serielle Ausführungsreihenfolge an.

a) $\langle r_1(x), r_2(x), w_1(x), r_3(x), w_2(x), w_3(x)\rangle$

b) $\langle r_1(x), r_2(x), w_2(x), r_3(x), w_1(x), w_3(x)\rangle$

Aufgabe 5: Konsistenzstufen von Transaktionen

Das Bankhaus „Mikrobank“ verwaltet Kundenkonten in einer Relation, die nach folgendem Schema aufgebaut ist: Konto (<u>Kontonr</u> , Kundennr , Kontostand)

Weiterhin hat die Relation „Konto“ der Datenbank anfangs folgenden Zustand:

(1,	4711,	500.0)
(2,	4711,	400.0)
(3,	4711,	500.0)
(4,	4712,	600.0)

Folgende Transaktionen sollen auf der Datenbank ausgeführt werden.

a) Überweisung „100.0“ von Konto 1 nach Konto 2
   1. BOT;
   2. UPDATE Konto SET Kontostand = Kontostand – 100.0 WHERE Kontonr = 1;
   3. UPDATE Konto SET Kontostand = Kontostand + 100.0 WHERE Kontonr = 2;

   4. ABORT;

b) Überweisung „200.0“ von Konto 1 nach Konto 2
   1. BOT;
   2. UPDATE Konto SET Kontostand = Kontostand – 200.0 WHERE Kontonr = 1;
   3. UPDATE Konto SET Kontostand = Kontostand + 200.0 WHERE Kontonr = 2;
   4. COMMIT;

c) Kontoausgabe Konto 1
   1. BOT;
   2. SELECT Kontostand FROM Konto WHERE Kontonr = 1;
   3. COMMIT;

d) Doppelte Kontoausgabe Konto 1
   1. BOT;
   2. SELECT Kontostand FROM Konto WHERE Kontonr = 1;
   3. SELECT Kontostand FROM Konto WHERE Kontonr = 1;
   4. COMMIT;

e) Ausgabe von Konten mit Kontostand von mehr als 499.0
   1. BOT;
   2. SELECT Kontonr FROM Konto WHERE Kontostand > 499.0;
   3. SELECT Kontonr FROM Konto WHERE Kontostand > 499.0;
   4. COMMIT;

1. Führen Sie die Transaktionen a) bis e) seriell nacheinander auf der Datenbank aus. Notieren Sie sich dabei die Ausgaben bzw. die Änderungen am Datenbankzustand.
   Was ändert sich, wenn die Reihenfolge der Transaktionen (*Schedule*) verändert wird?

Im Folgenden sollen Transaktionen verzahnt ausgeführt werden. Die konkret zu betrachtenden Verzahnungen sind als Ausführungsfolgen in Form von Wertepaaren angegeben, die den Aufbau <Transaktion| Schritt> besitzen. So steht <a1> z. B. für den ersten Schritt von Transaktion a), also das „BOT“
Betrachten Sie die Ergebnisse (Ausgaben) der Transaktionen c) bis e). Sind diese korrekt? Wenn nicht, so benennen Sie das Phänomen und geben an, wie es sich beheben läßt.

2. Wie sieht die Ausgabe von Transaktion c) bei der Ausführungsfolge <a1>, <a2>, <c1>, <c2>, <a3>, <c3>, <a4> aus?
3. Welche Ausgaben liefert Transaktion d) bei folgender der Ausführungsfolge <b1>, <d1>, <d2>, <b2>, <b3>, <b4>, <d3>, <d4>?
4. Welche Ausgabe liefert Transaktion e) Ausführungsfolge <b,1>, <e,1>, <e,2>, <b,2>, <b,3>, <b,4>, <e,3>, <e,4>?

Aufgabe 6: 2-Phasen-Commit x

Geben Sie einen kurzen Überblick über das 2-Phasen-Commit Protokoll. Spielen Sie das Protokoll durch für den Fall, daß Gesamttransaktion

a) erfolgreich beendet wird?

b) abgebrochen wird?

Berücksichtigen Sie dabei, daß b) mehrere Ursachen haben kann.