

10. Übung zur Vorlesung "Datenbanken" im Sommersemester 2004

Prof. Dr. Gerd Stumme, Dipl.-Inform. Christoph Schmitz

5. Juli 2004

Aufgabe 1 – Logging und Recovery

Obwohl Abstürze natürlich auch während des Wiederanlaufs geschehen können, enthalten CLRs keine Undo-Information. Warum nicht?

Aufgabe 2 – Mehrbenutzer-Synchronisation

Sind die folgenden Historien serialisierbar? Zeichnen Sie den Serialisierbarkeitsgraphen! Falls serialisierbar: geben Sie eine äquivalente serielle Historie an.

- a) $r_3(a)w_3(a)r_1(a)r_1(b)r_2(b)w_2(b)w_3(b)c_1c_2c_3$
- b) $w_1(a)r_2(a)r_3(a)w_3(b)r_2(b)w_2(c)c_1c_2c_3$
- c) $r_2(c)r_1(a)w_2(a)r_4(a)r_1(b)w_3(b)r_4(b)r_4(c)w_4(b)c_1c_2c_3c_4$

Aufgabe 3 – Mehrbenutzer-Synchronisation

Betrachten Sie die folgenden Transaktionen:

T_1	T_2
read(A)	read(B)
read(B)	read(A)
if A = 0 then B := B + 1	if B = 0 then A := A + 1
write(B)	write(A)
commit	commit

- a) Zeigen Sie eine zeitverschränkte Ausführung von T_1 und T_2 , die eine nicht serialisierbare Historie ergibt.
- b) Gibt es eine verzahnte Ausführung, deren Historie serialisierbar ist?

Aufgabe 4 – Mehrbenutzer-Synchronisation

- a) Begründen Sie, warum mit dem Zwei-Phasen-Sperrprotokoll (2PL) nur serialisierbare Historien erzeugt werden können.
Tip: Machen Sie sich bewußt, was Kanten und Pfade im Serialisierbarkeitsgraphen bedeuten!
- b) Beim strikten 2PL werden im Gegensatz zum 2PL alle Sperren atomar beim Ende der Transaktion freigegeben. Begründen Sie, warum dies das Problem des kaskadierenden Rücksetzens behebt.