11. Übung zur Vorlesung "Datenbanken" im Sommersemester 2007 – mit Musterlösungen –

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

16. Juli 2007

Aufgabe 1

1. Erklären Sie, was es bedeutet, daß zwei Operationen im Konflikt stehen. Für welche der ACID-Eigenschaften spielt die Definition von "Konflikt" eine Rolle?

   Im Konflikt stehen zwei Operationen verschiedener Transaktionen T_1 und T_2 dann, wenn mindestens eine von beiden, z. B. T_1, schreibt. Dadurch ist es wesentlich, in welcher Reihenfolge sie ausgeführt werden: liest T_2 den von T_1 geschriebenen Wert, dann ist es entscheidend, ob sie dies vor oder nach der Schreiboperation von T_1 tut. Wenn T_2 das gleiche Datenobjekt wie T_1 schreibt, entscheidet die Reihenfolge, welcher Wert nach Ablauf beider Transaktionen sichtbar ist.

   Die Betrachtung von Konflikten dient also im Wesentlichen zum Einhalten der Eigenschaft "Isolation"; durch Umordnen zu einer serialisierbaren Historie wird sichergestellt, daß das Endergebnis das gleiche wie das einer seriellen Historie ist.

2. Auf Folie 15 von Kapitel 11 wird formal definiert, welche drei Eigenschaften eine Menge von Operationen haben muß, um eine *Historie* zu sein. Erklären Sie diese drei Eigenschaften mit eigenen Worten!

   $H = \bigcup_{i=1}^{n} T_i$: Die Historie enthält genau alle Operationen der beteiligten Transaktionen. $<_H$ verträglich mit allen $<_i$: Wenn in einer der beteiligten Transaktionen T_i zwei Operationen in einer bestimmten Reihenfolge vorkommen, so kommen sie auch in der Historie H in dieser Reihenfolge vor. Entweder $p <_H q$ oder $q <_H p$ für Konfliktoperationen: für solche Operationen, die im Konflikt stehen, muß in der Historie festgelegt sein, in welcher Reihenfolge sie vorkommen sollen.

Aufgabe 2

Skizzieren Sie einen Beweis für das Serialisierbarkeitstheorem!

Serialisierbar $\Rightarrow$ keine Zyklen:

Wenn eine Historie H bestehend aus $T_1, \ldots, T_n$ serialisierbar ist, dann hat sie die gleichen Konflikte (und den gleichen Serialisierbarkeitsgraphen) wie eine serielle Historie. Eine Kante $T_i \rightarrow T_j$ in diesem Serialisierbarkeitsgraphen bedeutet, daß T_i in der äquivalenten seriellen Historie vor T_j kommen muß. Hätte der Graph einen Zyklus, müßte es dementsprechend eine Transaktion geben, die vor sich selber kommen muß → Widerspruch.

Azyklisch ⇒ Serialisierbar:

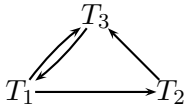
Ein azyklischer Graph ist topologisch sortierbar, d. h. es gibt eine Anordnung $T_{p_1}, \ldots, T_{p_n}$ von Transaktionen für eine Permutation p, so daß $T_i \rightarrow T_j$ im Graphen impliziert, daß $p_i < p_j$. Diese Anordnung von Transaktionen ist eine serielle Historie, die wegen der topologischen Ordnung die gleichen Konflikte wie die Ausgangshistorie haben muß.

Aufgabe 3

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

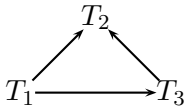
a) $r_3(a)w_3(a)r_1(a)r_1(b)r_2(b)w_2(b)w_3(b)c_1c_2c_3$

Nicht serialisierbar, da der Serialisierbarkeitsgraph zyklisch ist:



b) $w_1(a)r_2(a)r_3(a)w_3(b)r_2(b)w_2(c)c_1c_2c_3$

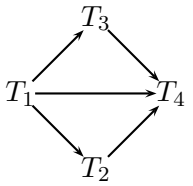
Serialisierbar, da der Serialisierbarkeitsgraph azyklisch ist:



Die einzige mögliche Reihenfolge ist also: T_1, T_3, T_2.

c) $r_2(c)r_1(a)w_2(a)r_4(a)r_1(b)w_3(b)r_4(b)r_4(c)w_4(b)c_1c_2c_3c_4$

Serialisierbar, da der Serialisierbarkeitsgraph azyklisch ist:



Die möglichen Reihenfolgen sind also: T_1, T_2, T_3, T_4 und T_1, T_3, T_2, T_4.

Aufgabe 4

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.

   Betrachte z. B. $r_1(a)r_1(b)r_2(b)r_2(a)w_1(b)w_2(a)c_1c_2$. Hier gibt es die Konflikte $r_1(a) \rightarrow w_2(a)$ und $r_2(b) \rightarrow w_1(b)$, also einen Zyklus im Serialisierbarkeitsgraphen → nicht serialisierbar.

b) Gibt es eine verzahnte Ausführung, deren Historie serialisierbar ist?

   Nein. Man kann sich überlegen, daß jede verzahnte Ausführung – bis auf Symmetrie – im Hinblick auf Konflikte einer der beiden folgenden Historien entspricht:

T_1	T_2
$r_1(A)$	
	$r_2(B)$
$w_1(B)$	
	$w_2(A)$

T_1	T_2
$r_1(A)$	
	$r_2(B)$
	$w_2(A)$
$w_1(B)$	

   Da jeweils der erste Befehl von T_1 mit dem letzten von T_2 in Konflikt steht und umgekehrt, führt also jede verzahnte Ausführung zu Zyklen im Serialisierbarkeitsgraphen.

SQL-Übungsaufgabe

Man möchte wissen, welche Vorlesung der Philosophenuni die grundlegendste ist.

Geben Sie ein SQL-Statement an, daß die Vorlesungen (Vorlnr reicht) mit der Anzahl ihrer Nachfolger angibt. Die Nachfolger sollen dabei auch die indirekten Nachfolger mitzählen.

Die Auflistung der Vorlesungen soll in absteigender Anzahl von Nachfolgern geordnet sein.

```
with transvor(vorgaenger, nachfolger) as
(
```

```
  select * from voraussetzen
    union all
  select t.vorgaenger, v.nachfolger
  from transvor t, voraussetzen v
  where t.nachfolger = v.vorgaenger
) select vorgaenger, count(*) anz from transvor
group by vorgaenger
order by anz desc
```

Lösung zur SQL-Aufgabe vom 2.7.

Formulieren Sie zwei Varianten der Anfrage: Welche Studenten sind über alle Vorlesungen geprüft worden? Stellen Sie den Allquantor einmal über eine Abzählung dar und einmal über eine doppelte Negation!

Variante 1: Abzählen

```
select matrnr from prüfen
group by matrnr
having count(*) = (select count(*) from vorlesungen)
```

Variante 2: Doppelte Negation und Existenzquantor

```
select matrnr from studenten s
where not exists (
   select * from vorlesungen v
   where not exists (
      select * from pruefen p
      where p.matrnr = s.matrnr
      and p.vorlnr = v.vorlnr
   )
)
```