

Datenbanken 1

Sommersemester 2017

Übung 8

(v3.0 - 9.6.2017)

Übersicht

- Aufgabe 1: **Einfache Transaktionen Model (Lock/Unlock)**
- Aufgabe 2: **2-Phasen-Sperrprotokoll (Two phase locking)**
- Aufgabe 3: **2-Phasen-Sperrprotokoll + Read/Write Lock Model**
- Aufgabe 4: **Baumprotokoll**
- Aufgabe 5: **Hierarchische Sperr-Protokoll**

Frage: Wie können nicht-serialisierbare Ausführungen vermieden werden?

- Scheduler
 - Lock/Unlock
- Protokolle (die Serialisierbarkeit garantieren)
 - **2-Phasen-Sperrprotokoll**
 - **Baumprotokoll**
 - **Das hierarchische Sperr-Protokoll**

Aufgabe 1: Einfache Transaktionen Model (Lock/Unlock)

- Überprüfen Sie den Schedule auf Serialisierbarkeit und begründen Sie Ihre Antwort (durch ein formelles Modell oder vergleichbare Verfahren aus der Vorlesung).

a)

	T1	T2
1	LOCK A	
2	UNLOCK A	
3		LOCK B
4		UNLOCK B
5		LOCK A
6		UNLOCK A
7	LOCK B	
8	UNLOCK B	
9		LOCK B
10		UNLOCK B
11	LOCK A	
12	UNLOCK A	
13	LOCK B	
14	UNLOCK B	
15		LOCK A
16		UNLOCK A

- Für alle Sequenzen von LOCK und UNLOCK assoziiere eine Funktion $f()$!

Einfache Transaktionen Model (Lock/Unlock)

a)

	T1	T2
1	LOCK A (F1)	
2	UNLOCK A	
3		LOCK B (F5)
4		UNLOCK B
5		LOCK A (F6)
6		UNLOCK A
7	LOCK B (F2)	
8	UNLOCK B	
9		LOCK B (F7)
10		UNLOCK B
11	LOCK A (F3)	
12	UNLOCK A	
13	LOCK B (F4)	
14	UNLOCK B	
15		LOCK A (F8)
16		UNLOCK A

Überprüfung für Graph Cycles:

$F6(F1(A)): T1 \rightarrow T2$

$F2(F5(B)): T2 \rightarrow T1$

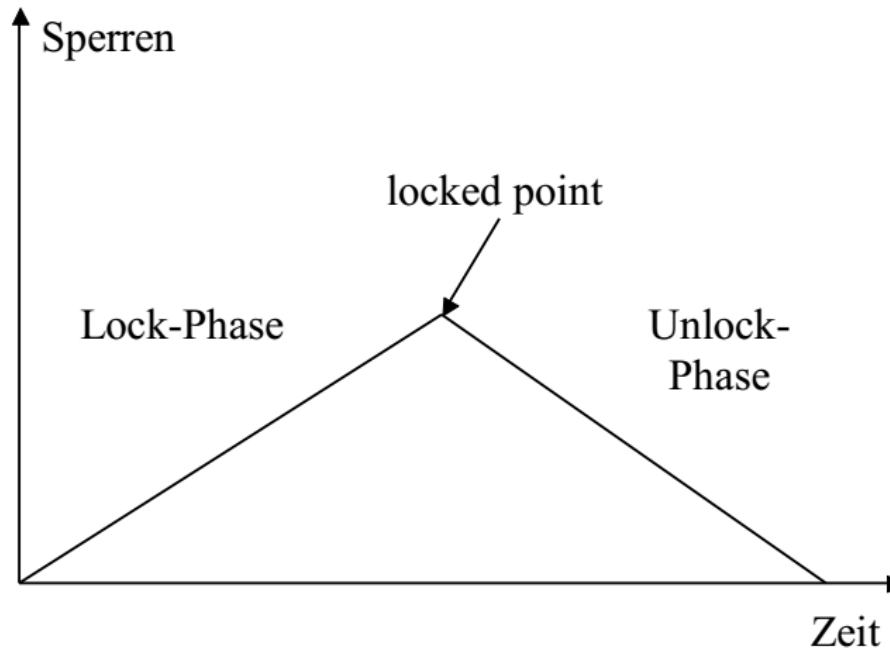
$F7(F2(F5(B))): T1 \rightarrow T2$

...

Diese Ausführung ist nicht-serialisierbar!

Aufgabe 2: 2-Phasen-Sperrprotokoll (Two phase locking)

- Das Protokoll ist so, dass verlangt wird, dass in einer Transaktion **alle Locks** vor **allen Unlocks** stattfinden.
- Wenn in einem Schedule alle Transaktionen zweiphasig sind, ist es **serialisierbar**.

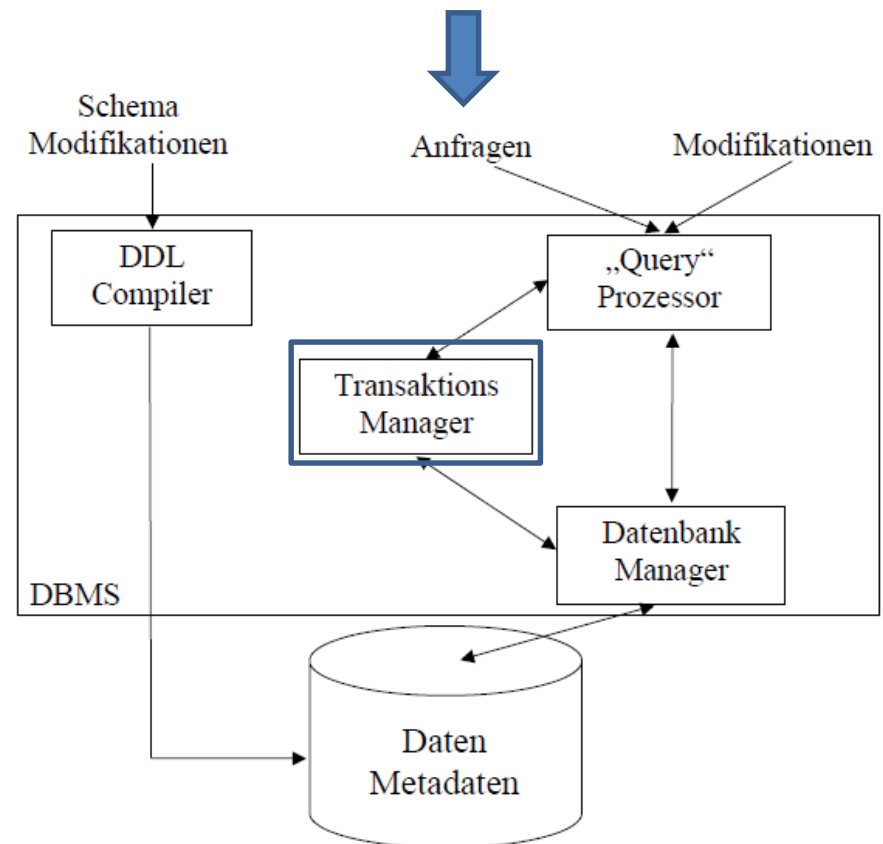


Aufgabe 2: 2-Phasen-Sperrprotokoll (Two phase locking)

- Stream:

T1(Read A); T2(Read B); T2(Read A); T1(Read B);
T2(Write B); T1(Write A); T1(Write B); T2 (Write A);

...



Aufgabe 2: 2-Phasen-Sperrprotokoll (Two phase locking)

a)

	T1	T2
1	LOCK A	
2		LOCK B
3		Wait A
4	Wait B	
5	Abort (T1)	
6		LOCK A
7		UNLOCK B
8		UNLOCK A
9	Restart (T1)	
10	LOCK A	
11	LOCK B	
12	UNLOCK A	
13	UNLOCK B	

Aufgabe 3: 2-Phasen-Sperrprotokoll + Read/Write Lock Model

- Es gibt zwei elementare Typen von Sperren:
 - Read-Locks (Shared Locks): nur zum Lesen.
 - Write-Locks (Exclusive Locks): zum Updaten.

		Bestehende Sperre	
		R-LOCK	W-LOCK
Sperr- anforderung	R-LOCK	☺	☹
	W-LOCK	☹	☹

☺ Verträglich, ☹ Konflikt

- Für jede Transaktion gilt:
 - kein Unlock, falls sie keinen R/W-Lock hält.
 - kein R-Lock, falls sie bereits einen Lock hält.
 - kein W-Lock, falls sie bereits einen W-Lock hält.

Ein Write-Lock kann auf ein Objekt gesetzt werden, für das die Transaktion bereits einen Read-Lock hält.

Aufgabe 3: 2-Phasen-Sperrprotokoll + Read/Write Lock Model

- Stream:

T1(Read A);T2(Read B); T2(Read A); T1(Read B);
T2(Write B); T1(Write A); T1(Write B); T2 (Write A);

...

Aufgabe 3: 2-Phasen-Sperrprotokoll + Read/Write Lock Model

Vorschlag 1 a)

	T1	T2
1	RLOCK A	
2		RLOCK B
3		RLOCK A
4	RLOCK B	
5		Wait B
6	Wait A	
7	Abort (T1)	
8	Restart (T1)	
9		WLOCK B
10	RLOCK A	
11		Wait A
12	Wait B	
13	Abort (T1)	
14	Restart (T1)	
15		WLOCK A
16	Wait A	
17		UNLOCK B
18		UNLOCK A
19	RLOCK A	
20	RLOCK B	
21	WLOCK A	
22	WLOCK B	
23	UNLOCK A	
24	UNLOCK B	

Vorschlag 2 b)

	T1	T2
1	RLOCK A	
2		RLOCK B
3		RLOCK A
4	RLOCK B	
5		Wait B
6	Wait A	
7	Abort (T1)	
8		WLOCK B
9		WLOCK A
10		UNLOCK B
11		UNLOCK A
12	Restart (T1)	
13	RLOCK A	
14	RLOCK B	
15	WLOCK A	
16	WLOCK B	
17	UNLOCK A	
18	UNLOCK B	

Aufgabe 4: Baumprotokoll

Nebenläufigkeit bei hierarchisch strukturierten Objekten

Beispiel: Die Objekte von verschiedener Größe sind definiert als kleinere Objekte in größeren:

- 1) Gesamte Datenbank.
- 2) Jede Relation.
- 3) Jeder Block einer Datei, die mit einer Relation korreliert.
- 4) Jedes Tupel.

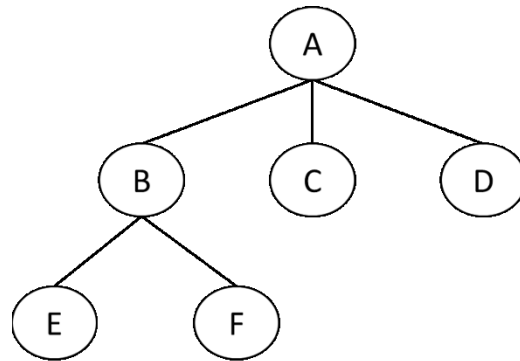
XML und JSON Datei haben auch Baumstruktur!

Aufgabe 4: Baumprotokoll

- Eine Transaktion erfüllt das Baumprotokoll, wenn gilt:
 1. Ein Objekt A wird nur dann von T gesperrt, wenn T bereits einen Lock auf den Vater von A hält. Dies gilt nicht für das erste Objekt (das muss nicht die Wurzel sein) im Baum, das von T gesperrt wird.
 2. Kein Objekt wird von einer Transaktion mehr als einmal gesperrt.

Aufgabe 4

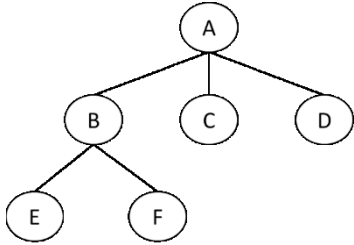
Gegeben ist der unten angezeigte Baum, welcher eine hierarchische Objektstruktur darstellt. Dem Scheduler wird der unten gegebene Schedule der Transaktionen T1 und T2 übergeben. Der Scheduler folgt dem Baumprotokoll.



T1	T2
	WRITE F
WRITE C	
WRITE B	
	WRITE D
COMMIT	
	COMMIT

Aufgabe 4: Baumprotokoll

Vorschlag 1



T1	T2
	WRITE F
WRITE C	
WRITE B	
	WRITE D
COMMIT	
	COMMIT

	T1	T2
1		Lock A
2		Lock B
3		Lock F
4		Write F
5		Unlock F
6		Unlock B
7	Wait A	
8		Lock D
9		Write D
10		Unlock D
11		Unlock A
12	Lock A	
13	Lock C	
14	Write C	
15	Unlock C	
16	Lock B	
17	Write B	
18	Unlock B	
19	Unlock A	
20	Commit	
21		Commit

Vorschlag 2

	T1	T2
1		Lock A
2		Lock B
3		Lock D
4		Unlock A
5		Lock F
6		Write F
7		Unlock F
8		Unlock B
9	Lock A	
10	Lock C	
11	Write C	
12	Unlock C	
13	Lock B	
14	Write B	
15	Unlock B	
16	Unlock A	
17		Write D
18		Unlock D
19	Commit	
20		Commit

Aufgabe 5: Hierarchische Sperr-Protokoll

Modell für Transaktionen T_i :

- LOCK (Exclusive, E-Sperre): Sperrt ein Objekt und alle seine Nachfahren. Zwei Transaktionen können nicht gleichzeitig ein Lock auf ein Objekt halten.
- WARN (Intention, I-Sperre): Keine Transaktion kann ein Objekt sperren, auf dem eine Warnung liegt.
- UNLOCK: Entfernt ein WARN oder ein LOCK

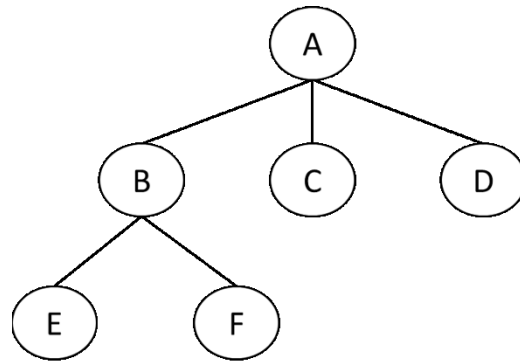
Aufgabe 5: Hierarchische Sperr-Protokoll

Eine Transaktion gehorcht dem Warnungs- oder hierarchischen Sperrprotokoll auf einer Objekt-Hierarchie, wenn sie:

- 1) damit beginnt, eine Warnung oder Lock auf die Wurzel zu setzen.
- 2) keine Sperre oder Warnung auf ein Objekt setzt, solange sie keine Warnung auf den Vorfahren (Parent) hält.
- 3) keine Warnung oder Sperrung von einem Objekt entfernt, solange sie Warnungen oder Sperren auf Nachfahren (Children) des Objektes hält.
- 4) sie dem 2-Phasen Protokoll folgt.

Aufgabe 5

Gegeben ist der unten angezeigte Baum, welcher eine hierarchische Objektstruktur darstellt. Dem Scheduler wird der unten gegebene Schedule der Transaktionen T1 und T2 übergeben. Der Scheduler folgt dem hierarchischen Sperrprotokoll.

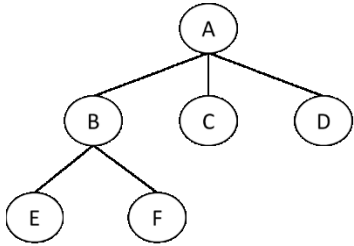


T1	T2
	WRITE F
WRITE C	
WRITE B	
	WRITE D
COMMIT	
	COMMIT

Aufgabe 5: Hierarchische Sperr-Protokoll

Vorschlag 1

Vorschlag 2



T1	T2
	WRITE F
WRITE C	
WRITE B	
	WRITE D
COMMIT	
	COMMIT

	T1	T2
1		Warn A
2		Warn B
3		Lock F
4		Write F
5	Warn A	
6	Lock C	
7	Write C	
8	Lock B geht nicht WAIT	
9		Lock D
10		Write D
11		Unlock D
12		Unlock F
13		Unlock B
14	Lock B	
15	Write B	
16		Unlock A
17		Commit
18	Unlock B	
19	Unlock C	
20	Unlock A	
21	Commit	

	T1	T2
1		Warn A
2		Warn B
3		Lock F
4		Write F
5	Warn A	
6	Lock C	
7	Write C	
8	Lock B geht nicht WAIT	
9		Lock D
10		Write D
11		Unlock D
12		Unlock F
13		Unlock B
14		Unlock A
15		Commit
16	Lock B	
17	Write B	
18	Unlock B	
19	Unlock C	
20	Unlock A	
21	Commit	