



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS17/18

Harald Lang, Linnea Passing (gdb@in.tum.de)

<http://www-db.in.tum.de/teaching/ws1718/grundlagen/>

Blatt Nr. 14

**Hausaufgabe 1**

- Erläutern Sie kurz die zwei Phasen des 2PL-Protokolls.
- Inwiefern unterscheidet sich das *strenge* 2PL?
- Welche Eigenschaften (SR,RC,ACA,ST) haben Historien, welche vom 2PL und vom strengen 2PL zugelassen werden?
- Wäre es beim strengen 2PL-Protokoll ausreichend, alle Schreibsperrern bis zum EOT (Transaktionsende) zu halten, aber Lesesperrern schon früher wieder freizugeben?

**Hausaufgabe 2**

- Geben Sie alle Eigenschaften an, die von der Historie erfüllt werden.

$$H_1 = w_1(x), r_2(y), w_3(y), w_2(x), w_3(z), c_3, w_1(z), c_2, c_1$$

richtig	falsch	Aussage
		$H_1$ ist serialisierbar (SR)
		$H_1$ ist rücksetzbar (RC)
		$H_1$ vermeidet kaskadierendes Zurücksetzen (ACA)
		$H_1$ ist strikt (ST)

- Geben Sie alle Eigenschaften an, die von der Historie erfüllt werden.

$$H_2 = r_1(x), r_1(y), w_2(x), w_3(y), r_3(x), a_1, r_2(x), r_2(y), c_2, c_3$$

richtig	falsch	Aussage
		$H_2$ ist serialisierbar (SR)
		$H_2$ ist rücksetzbar (RC)
		$H_2$ vermeidet kaskadierendes Zurücksetzen (ACA)
		$H_2$ ist strikt (ST)

- Gegeben die unvollständige Historie:

$$H_3 = w_1(x), w_1(y), r_2(x), r_2(y)$$

- Fügen Sie **commits** in  $H_3$  so ein, dass die Historie RC aber nicht ACA erfüllt.
- Fügen Sie **commits** in das ursprüngliche  $H_3$  so ein, dass die Historie ACA erfüllt.

### Hausaufgabe 3

Bei der sperrbasierten Synchronisation hat jedes Datenobjekt eine zugehörige Sperre. Bevor eine Transaktion zugreifen darf, muss sie eine Sperre anfordern. Dabei unterscheiden wir zwei Sperrmodi: Lese- und Schreibsperre.

- a) Erläutern Sie kurz die Unterschiede.
- b) Geben Sie deren Verträglichkeiten an (wenn mehrere Transaktionen Sperren auf dem selben Datenobjekt anfordern).

### Hausaufgabe 4

Ein inhärentes Problem der sperrbasierten Synchronisationsmethoden ist das auftreten von Verklemmungen (Deadlocks). Zur Erkennung von Verklemmungen wurde der Wartegraph eingeführt. Dabei wird eine Kante  $T_i \rightarrow T$  eingefügt, wenn  $T_i$  auf die Freigabe einer Sperre durch  $T$  wartet.

Skizzieren Sie einen Ablauf von Transaktionen, bei dem ein Deadlock auftritt, der einen Zyklus mit einer Länge von mindestens 3 Kanten im Wartegraphen erzeugt.

### Hausaufgabe 5

Bei der Verklemmungsvermeidung durch Zeitstempel wird jeder Transaktion ein eindeutiger Zeitstempel (*time stamp*,  $TS$ ) zugeordnet. Die  $TS$  werden (streng) monoton wachsend vergeben, so dass gilt:  $TS(T_{alt}) < TS(T_{jung})$ . Eine Sperranforderung auf ein bereits gesperrtes Datenobjekt führt nicht mehr zwangsläufig dazu, dass eine Transaktion in den Wartezustand übergeht. Stattdessen kann es, abhängig vom Alter der Transaktionen, zum Abbruch von einer der betroffenen Transaktionen führen.

- a) Erläutern Sie die zwei Strategien *wound-wait* und *wait-die*.

Ein Nachteil der Zeitstempelmethode ist, dass es zu Transaktionsabbrüchen kommen kann, obwohl eine Verklemmung nie aufgetreten wäre.

- b) Geben Sie für *wound-wait* und *wait-die* jeweils eine verklemmungsfreie Historie an, die dennoch zu einem Transaktionsabbruch führt.

### Hausaufgabe 6

Gegeben die Relation „Ärzte“, die den Bereitschaftsstatus von Ärzten modelliert

Name	Vorname	...	Bereit
House	Gregory	...	ja
Green	Mark	...	nein
Brinkmann	Klaus	...	ja

sowie die folgende Transaktion in Pseudocode:

```
dienstende(arzt_name)
  select count(*) into anzahl_bereit from aerzte where bereit='ja'
  if anzahl_bereit > 1 then
    update aerzte set bereit='nein' where name=arzt_name
```

Die Transaktion soll sicherstellen, dass immer mindestens ein Arzt bereit ist.

Betrachten Sie einen Ablauf, bei dem zwei zur Zeit bereite Ärzte zum gleichen Zeitpunkt entscheiden, ihren Status auf „nein“, d.h. nicht bereit zu ändern:

$T_1$ : execute dienstende('House')

$T_2$ : execute dienstende('Brinkmann')

Gehen Sie beispielsweise davon aus, dass das DBMS versucht, die Transaktion jeweils abwechselnd zeilenweise abzuarbeiten.

Diskutieren Sie:

- a) Was kann bei Snapshot Isolation passieren?
- b) Warum ist dies bei optimistischer Synchronisation nicht möglich?
- c) Wie verhält sich die Zeitstempel-basierte Synchronisation?
- d) Wie verhält sich das strenge 2PL?