



## Übung zur Vorlesung *Grundlagen: Datenbanken* im WS14/15

Harald Lang (harald.lang@in.tum.de)

<http://www-db.in.tum.de/teaching/ws1415/grundlagen/>

### Blatt Nr. 13

Am 28.01.2015 findet anstelle der Vorlesung eine Zentralübung/Fragestunde statt.

#### Hausaufgabe 1 - Sperrbasierte Synchronisation

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.

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

#### Hausaufgabe 2 - Zwei-Phasen-Sperrprotokoll

- 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 Schreibsperren bis zum EOT (Transaktionsende) zu halten, aber Lesesperren schon früher wieder freizugeben?

#### Hausaufgabe 3 - Verklemmungen (Deadlocks)

Ein inherentes 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 4 - Verklemmungsvermeidung durch Zeitstempel

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, basierend auf dem Alter der Transaktionen, zum Abbruch von einer der betroffenen Transaktionen führen.

- 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 5 - Snapshot Isolation im Vergleich zu anderen Synchronisationsverfahren

Gegeben die Relation „Aerzte“, 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:

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