

# GDB Tutorium

Woche 14

Jigao Luo

TUM

6. Februar 2020

- Probleme (Nicht abgesicherte Nebenläufigkeit)
- die Isolation Levels
- Historien (Ablauf von Transaktionen)
- Sperrprotokoll

- lost update
- dirty read
- non-repeatable read
- phantom problem
- Beispiele und die Isolation Levels in Slide + Aufgabe1

- Konfliktoperationen: Konflikt, wenn mindestens eine davon eine Schreiboperation mit einem gleichen Objekt
- (Konflikt-)Äquivalenz: Ergebnis nicht zu verändern
- Serialisierbarkeit: eine äquivalente Historie zu einer seriellen Historie
- Serialisierbarkeit  $\equiv$  Serialisierbarkeitsgraph  $SG(H)$  azyklisch ist

- Rücksetzbarkeit (RC): immer  $T_i$  von einer anderen  $T_j$  liest ( $i \neq j$ ) und  $c_i \in H$ , dann  $c_j < c_i$
- RC: TA darf erst commit-en, wenn alle TAs, von denen sie gelesen hat, beendet sind.
- Vermeidung kaskadierenden Rücksetzens(ACA): immer eine  $T_i$  von einer anderen  $T_j$  liest ( $i \neq j$ ), dann  $c_j < r_i[x]$
- ACA:  $T_i$  darf nur Datum vom committed TAs lesen.
- Striktheit (ST):  $w_j[x] < o_i[x]$  (mit  $o_i[x] = r_i[x]$  oder  $w_i[x]$  gilt entweder  $a_j < o_i[x]$  oder  $c_j < o_i[x]$ )
- ST: Nur von bereits erfolgreich abgeschlossenen TAs darf gelesen oder dürfen Datenobjekte überschrieben werden.
- Einordnung-Figure wichtig!

- Zwei-Phasen-Sperrprotokoll (2PL)
- Strenges 2PL
- Wartegraph
- Aufgabe4 + Aufgabe 5

SQL-92 spezifiziert mehrere Konsistenzstufen (isolation level ) durch welche der Benutzer (bzw. die Anwendung) festlegen kann, wie “stark” eine Transaktion von anderen parallel laufenden Transaktionen isoliert werden soll.

- Nennen und erläutern Sie kurz die Isolation Level. Geben Sie an, welche Nebenläufigkeitsprobleme mit dem jeweiligen Level vermieden werden.
- Warum kann zwischen den Konsistenzstufen gewählt werden?

Siehe Blatt



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

$H1 = w1(x); r2(y); w3(y); w2(x); w3(z); c3; w1(z); c2; c1$

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

$H2 = r1(x); r1(y); w2(x); w3(y); r3(x); a1; r2(x); r2(y); c2; c3$

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).

In der Vorlesung haben Sie Serialisierbarkeitsgraphen und den Wartegraphen des (strikten) 2PL kennen gelernt.

- Was bedeutet eine Kante  $T1 \rightarrow T2$  im Serialisierbarkeitsgraphen einer Historie  $H$ ?
- Gehen Sie davon aus, dass die Datenbank die 2PL-Strategie verwendet. Was bedeutet eine Kante  $T1 \rightarrow T2$  in einem Wartegraphen? Worin besteht der Unterschied zu Aufgabe a)?
- Was bedeutet ein Kreis im Serialisierbarkeitsgraphen einer Historie  $H$ ? Was im Wartegraphen? Wo liegt der Unterschied?
- Wie viele neue Kanten werden dem Wartegraphen maximal hinzugefügt, wenn eine Transaktion eine S-Sperre anfordert? Wie viele bei einer X-Sperre?