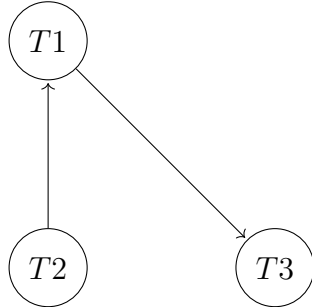


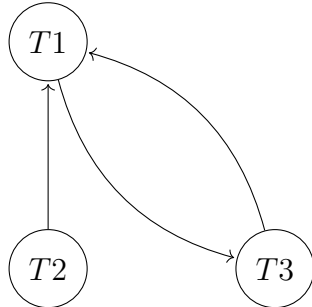
## Big Data Engineering – 7. Übungsblatt

### Aufgabe 1

- a) •  $AP_1$  ist konfliktserialisierbar.

$$\begin{aligned}
 & r_3(C) \rightarrow r_2(B) \rightarrow r_1(A) \rightarrow w_2(B) \rightarrow w_3(C) \rightarrow w_1(A) \rightarrow r_1(B) \rightarrow r_3(A) \rightarrow w_1(B) \\
 & \rightarrow w_3(A) \\
 & \equiv r_3(C) \rightarrow r_2(B) \rightarrow w_2(B) \rightarrow w_3(C) \rightarrow r_1(\mathbf{A}) \rightarrow w_1(A) \rightarrow r_1(B) \rightarrow r_3(A) \rightarrow \\
 & w_1(B) \rightarrow w_3(A) \\
 & \equiv r_3(C) \rightarrow r_2(B) \rightarrow w_2(B) \rightarrow w_3(C) \rightarrow r_1(A) \rightarrow w_1(A) \rightarrow r_1(B) \rightarrow w_1(\mathbf{B}) \rightarrow \\
 & r_3(A) \rightarrow w_3(A) \\
 & \equiv r_3(C) \rightarrow r_2(B) \rightarrow w_2(B) \rightarrow r_1(A) \rightarrow w_1(A) \rightarrow r_1(B) \rightarrow w_1(B) \rightarrow w_3(\mathbf{C}) \rightarrow \\
 & r_3(A) \rightarrow w_3(A) \\
 & \equiv r_2(B) \rightarrow w_2(B) \rightarrow r_1(A) \rightarrow w_1(A) \rightarrow r_1(B) \rightarrow w_1(B) \rightarrow r_3(\mathbf{C}) \rightarrow w_3(C) \rightarrow \\
 & r_3(A) \rightarrow w_3(A)
 \end{aligned}$$


- $AP_2$  ist nicht konfliktserialisierbar, da die Reihenfolge von  $r_1(A) \rightarrow r_3(A) \rightarrow w_3(A) \rightarrow w_1(A)$  darf nicht verändert werden.



- b) 1.  $T_1 \rightarrow T_2 \rightarrow T_5 \rightarrow T_4 \rightarrow T_{13} \rightarrow T_6 \rightarrow T_{10} \rightarrow T_7 \rightarrow T_9 \rightarrow T_8 \rightarrow T_{12} \rightarrow T_{11} \rightarrow T_3 \rightarrow T_{14}$

2. •

	1	2	3	4	5	6	7
Thread 1	$T_{10}$	$T_9$	$T_8$	$T_{12}$	$T_7$	$T_{11}$	$T_{14}$
Thread 2	$T_1$	$T_2$	$T_{13}$	$T_5$	$T_4$	$T_6$	$T_3$

	1	2	3	4	5	6
• Thread 1	$T_1$	$T_2$	$T_7$		$T_{11}$	$T_3$
Thread 2	$T_{10}$	$T_9$	$T_8$	$T_{12}$	$T_{14}$	
Thread 3	$T_{13}$	$T_6$				
Thread 4	$T_5$	$T_4$				

## Aufgabe 2

### 1. Read Uncommitted

- Diese Isolationsstufe ermöglicht Dirty Reads,  $T_4$  kann die von  $T_3$  durchgeführte nicht festgeschriebene Aktualisierung lesen.
- $T_2$  erhält keine Schreibsperre von  $T_3$ , um Zeile 8 auszuführen.
- => festgeschrieben:  $T_1, T_2, T_3, T_4, T_5$ .

### 2. Read Committed

- Diese Isolationsstufe ermöglicht Dirty Reads nicht. Zeile 10 von  $T_4$  kann nicht ausführen werden, da  $T_3$  noch nicht festschreiben.
- $T_2$  erhält keine Schreibsperre von  $T_3$ , um Zeile 8 auszuführen.
- => festgeschrieben:  $T_1, T_3, T_5$ ; abbrechen:  $T_2, T_4$ .

### 3. Repeatable Read

- Diese Isolationsstufe ermöglicht Dirty Reads nicht. Zeile 10 von  $T_4$  kann nicht ausführen werden, da  $T_3$  noch nicht festschreiben.
- Zeile 4 von  $T_1$  erstellt Non-Repeatable Read für  $T_2$ .
- $T_2$  erhält keine Schreibsperre von  $T_3$ , um Zeile 8 auszuführen.
- => festgeschrieben:  $T_3, T_5$ ; abbrechen:  $T_1, T_2, T_4$ .

### 4. Serializable

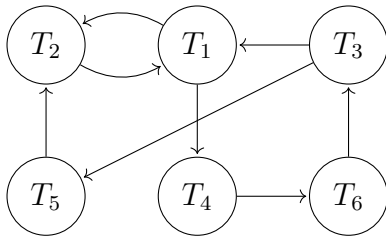
- Diese Isolationsstufe ermöglicht Dirty Reads nicht. Zeile 10 von  $T_4$  kann nicht ausführen werden, da  $T_3$  noch nicht festschreiben.
- Zeile 4 von  $T_1$  erstellt Non-Repeatable Read für  $T_2$ .
- Zeile 12 von  $T_5$ , 14 von  $T_4$  erzeugen Phantom beim Lesen  $r(Z)$  von  $T_3$ .
- $T_2$  erhält keine Schreibsperre von  $T_3$ , um Zeile 8 auszuführen.
- => festgeschrieben:  $T_3$ ; abbrechen:  $T_1, T_2, T_4, T_5$ .

## Aufgabe 3

- Lücke  $T_2$ :  $u(X.x > 20)$   
 $T_4$  bricht ab, aber  $T_3$  schreibt in Isolationsstufe **Read Committed** fest, d.h gibt es Dirty Read in  $r(X.x \geq 20)$  aber nicht in  $r(X.x = 20)$ .
- Lücke  $T_3$ :  $u(Y.y > 30)$   
 $T_3$  bricht ab, d.h Lücke  $T_3$  erstellt Non-Repeatable Read.  $T_4$  schreibt vor Lücke  $T_3$  fest, es gibt nur  $r(Y.y > 30)$  zu berücksichtigen.

- Lücke  $T_3$ :  $r(Z.z=4)$   
 $T_2$  bricht ab, d.h  $T_2$  erstellt Phantom mit  $i(Z.z=4)$ .

#### Aufgabe 4



Deadlocks:  $T_1$ - $T_2$  und  $T_1$ - $T_3$ - $T_4$ - $T_6$

kleinste Menge an Transaktionen, die die Deadlocks durch Zurücksetzen auflösen:  $\{T_1\}$