

### Aufgabe 1

Im Folgenden gehen wir davon aus, dass die der Aufgabe zugrunde liegende Datenbank auf einem psql-Server liegt. Sie verfügt unter anderem über eine Tabelle Konto. Wir nehmen an, dass der Zähler **XId zu Beginn auf 1** gestellt sei und zudem die Tabelle leer sei. Für die Tabelle Konto gibt es ausserdem den Constraint: **Stand >= 0**.

Nach dem Einfügen der beiden ersten Tupel (siehe Tabelle unten) ist entsprechend der Zähler XId auf 2. Die Abfrage

```
SELECT *, Xmin, Xmax FROM konto
```

liefert folgende Tabelle:

Konto			
KId	Stand	Xmin	Xmax
A	2000	1	-
B	3000	1	-

Verschiedene User greifen nun parallel auf diese Tabelle zu und versuchen, die Transaktionen *T2* bis *T5* nebeneinander durchzuführen:

*T2*  
BEGIN;

UPDATE konto  
SET stand=stand-2500;

COMMIT;

*T3*  
BEGIN;  
SET TRANSACTION  
ISOLATION LEVEL  
REPEATABLE READ;

UPDATE konto  
SET stand = stand\*2  
WHERE kid='A';

COMMIT;

*T4*  
BEGIN;  
SET TRANSACTION  
ISOLATION LEVEL  
REPEATABLE READ;

UPDATE konto  
SET stand = stand+5000

COMMIT;

*T5*  
BEGIN;  
SET TRANSACTION  
ISOLATION LEVEL  
REPEATABLE READ;  
UPDATE konto  
SET stand = stand\*3  
WHERE kid='B';  
COMMIT;

a) Welche Ausgabe liefert am Ende der 4 Transaktionen die SQL-Abfrage  
`SELECT * FROM konto;`?

b) Erstellen Sie die Tabelle pg\_clog, welche am Ende dieser Transaktionen resultiert.

**Aufgabe 2**

Gegeben sei das folgende SQL-Statement.

```
SELECT l.name, a.aid, f.leiter  
FROM l, a, f  
WHERE l.lid = a.lid  
        AND f.fid = a.fid  
        AND f.leiter = 'Momo';
```

- a) Geben Sie die kanonische Übersetzung in relationale Algebra.
- b) Optimieren Sie Ihren relationalen Ausdruck soweit wie möglich.