

## Aufgabe „Tabelle TAB“

Die Transaktionen eines Transaktionsprogramms besteht aus SQL-Befehlen. Die Transaktionen  $T_1$  und  $T_2$  arbeiten auf der Tabelle TAB.

Transaktion $T_1$	Transaktion $T_2$	
BOT	BOT	TAB
SELECT FROM TAB	SELECT FROM TAB	F
NEUF := F+5	NEUF := F*2	2
UPDATE TAB SET F=NEUF	UPDATE TAB SET F=NEUF	
COMMIT WORK	COMMIT WORK	

Die quasiparallele Abarbeitung erfolgt in folgenden Schritten:

	$T_1$	$T_2$
1		BOT
2	BOT	
3	SELCT F FROM TAB	
4		SELECT F FROM TAB
5		NEUF := F*2
6	NEUF := F+5	
7	UPDATE TAB SET F=NEUF	
8	COMMIT WORK	
9		UPDATE TAB SET F=NEUF
10		COMMIT WORK

- (a) Ist die (quasiparallele) Bearbeitung der Transaktionen korrekt? Begründung!

Nein, es liegt ein Lost-Update-Fehlerfall vor. In Schritt 3 bzw. 4 lesen  $T_1$  bzw.  $T_2$  denselben Wert aus der Tabelle TAB. Der von  $T_1$  in Schritt 7 in die Tabelle zurückgeschriebene Wert wird in Schritt 9 von  $T_2$  überschrieben.

- (b) Konstruieren Sie unter Verwendung von  $T_1$  und  $T_2$  einen Dirty-Read-Fehlerfall.

	$T_1$	$T_2$
1		BOT
2	BOT	
3	SELCT F FROM TAB	
4	NEUF := F+5	
5	UPDATE TAB SET F=NEUF	
6		SELECT F FROM TAB
7		NEUF := F*2
8	ABORT	
9		UPDATE TAB SET F=NEUF
10		COMMIT WORK

Dirty-Read bedeutet, dass von zwei gleichzeitig ablaufenden Transaktionen die eine Transaktion Daten liest, die von der anderen Trans-

aktion geschrieben bzw. geändert werden, jedoch noch nicht bestätigt (committed) sind. Somit ist noch nicht sichergestellt, dass diese Daten permanent in die Datenbank übernommen werden. Findet dann ein Abort statt, hat die eine Transaktion Daten ausgelesen, die am Ende nicht in der Datenbank ankommen.

Sobald  $T_1$  committed hat, kann es dazu nicht mehr kommen, da dann die Änderungen von  $T_1$  permanent in der Datenbank festgeschrieben sind und  $T_2$  auf sichere, garantierte Werte zugreift.