

---

Aufgabe 1

---

Aufgabe 1 a)

Die Eigenschaften des Transaktionskonzepts werden oft (wie in der Vorlesung) mit der Abkürzung ACID zusammengefasst. ACID steht hierbei für vier grundlegende Eigenschaften, die die Transaktionen in einem Datenbanksystem erfüllen müssen. Nennen Sie diese vier Eigenschaften und diskutieren Sie deren Anforderungen bzw. Bedeutung.

Aufgabe 1 b)

In der Vorlesung wurden die Strategien force /  $\neg$  force und steal /  $\neg$  steal besprochen. Diese Strategien bestimmen das Verhalten eines Datenbanksystems in Bezug auf das Einbringen von Änderungsoperationen auf den Hintergrundspeicher bzw. das Ersetzen von Pufferseiten. Diskutieren Sie die Unterschiede zwischen den einzelnen Strategien bzw. deren Kombinationen. Worin liegen deren spezifische Vor- und Nachteile?

---

Aufgabe 2

---

Aufgabe 2 a)

Welche der ACID-Eigenschaften garantiert die Recovery?

Aufgabe 2 b)

Welche der folgenden Regeln müssen beim Write-Ahead-Logging (WAL) befolgt werden?

1. Vor dem Ausschreiben der Logeinträge einer Seite muss die modifizierte Seite ausgelagert werden.
2. Vor dem Commit einer Transaktion müssen alle Log-Einträge der Transaktion ins Log geschrieben werden.
3. Vor dem Ausschreiben einer modifizierten Seite müssen alle Logeinträge, die zu dieser Seite gehören ins Log geschrieben werden.
4. Vor dem Commit einer Transaktion müssen alle modifizierten Seiten der Transaktion ausgeschrieben werden.

---

### Aufgabe 3

---

#### Aufgabe 3 a)

Nach einem Absturz führt ein Datenbanksystem die Wiederanlaufphase (*Recovery*) durch. Die Recovery kann hierbei konzeptuell in drei Phasen aufgeteilt werden. Nennen Sie diese drei Phasen und beschreiben Sie die im Rahmen jeder einzelnen Phase durchgeführten Operationen.

#### Aufgabe 3 b)

Während der Recovery kann es zu einem erneuten Absturz des Datenbanksystems kommen. Warum kann dies ohne besondere Sicherungsmaßnahmen zu Problemen führen? Wie kann sichergestellt werden, dass auch bei (mehrmaligem) Absturz während der Recovery diese letztendlich erfolgreich abgeschlossen werden kann?

#### Aufgabe 3 c)

Warum müssen in den Kompensationseinträgen im Log keine Undo-Informationen gespeichert werden?

---

### Aufgabe 4

---

In einem Datenbanksystem werden die drei Transaktionen  $T_1, T_2$  und  $T_3$  verzahnt ausgeführt. Die (schreibenden) Zugriffe der Transaktionen auf die Datenelemente  $B, C$  und  $E$  werden in einem Log-File mitprotokolliert (siehe Abbildung 1). Es existieren keine Sicherungspunkte, d.h.  $T_1, T_2$  und  $T_3$  sind die ersten Transaktionen, die auf dem neu gestarteten Datenbanksystem laufen.

#### Aufgabe 4 a)

Angenommen an Abbruchpunkt 1 stürzt das System mit Verlust des Hauptspeicherinhalts ab. Geben Sie die einzelnen Schritte des Wiederanlaufs an.

#### Aufgabe 4 b)

Führen Sie den Wiederanlauf für einen Absturz an Abbruchpunkt 2 durch (Abbruchpunkt 1 wird ohne Absturz durchlaufen). Gehen Sie dabei wie in Teil (b) vor.

#### Aufgabe 4 c)

Welche Art von Logeinträgen verwendet das Datenbanksystem (logisch oder physisch)? Begründen Sie. Wie würden sich die erzeugten Logeinträge verändern, wenn eine andere Protokollierung eingesetzt werden würde?

	$T_1$	$T_2$	$T_3$	Log-Datei								
				[LSN,	TA,	PageId,	Redo,	Undo,	PrevLSN]			
1.	BOT	BOT $r(B, b_1)$ $b_1 = b_1 + 5$ $w(B, b_1)$		[#1,	$T_2,$	$-,$	BOT,	$-,$	0]			
2.				[#2,	$T_1,$	$-,$	BOT,	$-,$	0]			
3.				[#3,	$T_2,$	$P_B,$	$B = B + 5,$	$B = B - 5,$	#1]			
4.				Abbruchpunkt 1								
5.				[#4,	$T_3,$	$-,$	BOT,	$-,$	0]			
6.	$r(B, b_2)$ $w(B, b_2)$ $b_2 = b_2 + 4$ $r(C, c_1)$ $c_1 = c_1 - 3$ $w(C, c_1)$	Commit	BOT $r(E, e_1)$	[#5,	$T_2,$	$-,$	Commit,	$-,$	#3]			
7.				[#6,	$T_1,$	$P_B,$	$B = B + 4,$	$B = B - 4,$	#2]			
8.				[#7,	$T_1,$	$P_C,$	$C = C - 3,$	$C = C + 3,$	#6]			
9.				Abbruchpunkt 2								
10.				Commit		$r(C, c_2)$ $e_1 = e_1 + c_2$ $w(E, e_1)$ Commit	[#8,	$T_3,$	$P_E,$	$E = E + C,$	$E = E - C,$	#4]
11.							[#9,	$T_1,$	$-,$	Commit,	$-,$	#7]
12.							[#10,	$T_1,$	$-,$	Commit,	$-,$	#8]
13.												
14.												

Abbildung 1: Log-File

---

### Aufgabe 5

---

Gegeben seien die Transaktionen  $T_1$ ,  $T_2$  und  $T_3$  mit folgenden Operationen ( $T_1, T_2$  und  $T_3$  bearbeiten die Datenelemente  $A$  und  $B$ ):

- $T_1$  :  $BOT_1$ ;  $Read_1(B)$ ;  $Write_1(A)$ ;  $Read_1(B)$ ;  $Write_1(B)$ ;  $COMMIT_1$
- $T_2$  :  $BOT_2$ ;  $Read_2(B)$ ;  $Read_2(A)$ ;  $Write_2(B)$ ;  $COMMIT_2$
- $T_3$  :  $BOT_3$ ;  $Read_3(B)$ ;  $Write_3(B)$ ;  $ABORT_3$

Verzählen Sie jeweils zwei der drei Transaktionen so, dass ein

Aufgabe 5 a)

“Lost Update” auftritt.

Aufgabe 5 b)

“Dirty Read” auftritt.

Aufgabe 5 c)

“Non-repeatable Read” auftritt.