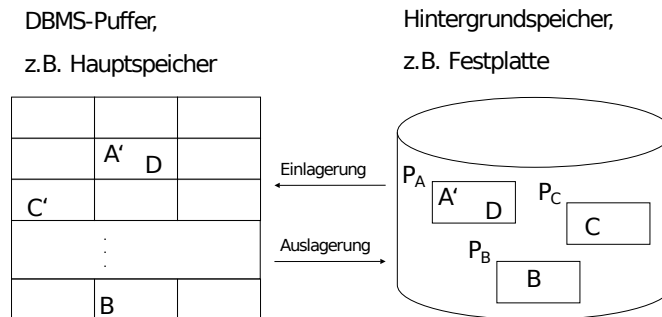


Aufgabe 1: Transaktionen

(1 P.)

- a) Gegeben zwei Transaktionen t_1 und t_2 und folgende Situation der Seiten P_A , P_B und P_C sowie der Datensätze A, B, C und D. Transaktion t_1 hatte bislang Zugriff auf Datensätze A und B. Transaktion t_2 hatte bislang Zugriff auf Datensätze C und D. Durch ‘ markierte Datensätze illustrieren geänderte Datensätze.



Beantworten Sie folgende Fragen:

- Was muss gewährleistet werden, wenn Transaktion t_1 bereits erfolgreich committed hat?
 - Was muss passieren, wenn Transaktion t_1 noch nicht committet hat und abbricht?
 - Was muss passieren, wenn Transaktion t_2 noch nicht committet hat und abbricht?
 - Wie oft musste eine Seite in den Puffer geladen werden?
 - t_2 möchte auch Datensatz B lesen, was muss gelten, damit dies OK ist?
- b) Wenn ein Datenbanksystem mit `-force` und `steal` konfiguriert ist, braucht es UNDO, REDO, beides, oder weder noch? Begründen Sie Ihre Antwort.

Aufgabe 2: Serialisierbarkeit

(1 P.)

- a) Folgende Programme laufen parallel ab, wobei `load(X)` eine Anforderung an das Datenbanksystem ist, um Objekt X zu lesen, und `store(X, val)` die Anweisung das Objekt X mit Wert val zu aktualisieren. Gehen Sie davon aus, dass die Operationen im gleichen Zeitschritt vom Datenbanksystem ausgeführt werden.

Zeit	Programm 1	Zeit	Programm 2	Zeit	Programm 3
1:	BEGIN	1:	BEGIN	1:	BEGIN
2:		2:	x = load(C)	2:	
3:	x = 0	3:	y = load(A)	3:	
4:		4:	z = x + y	4:	store(C, 5)
5:	y = load(A)	5:		5:	
6:	y += 4	6:		6:	x = load(A)
7:		7:		7:	store(A, x+1)
8:	store(A, y)	8:		8:	
9:		9:	x = load(A)	9:	
10:	COMMIT	10:	z = z + x	10:	COMMIT
11:		11:	store(A, z)	11:	
12:		12:	COMMIT	12:	

- i) Geben Sie die Historie an, die beim Datenbanksystem ankommt.
- ii) Ist diese Historie serialisierbar? Zeigen Sie dies anhand des Konfliktgraphen.
- iii) An welchen Stellen ist in diesem Ablauf ACID verletzt? Beschreiben Sie das Problem möglichst genau.
- b) Gegeben der folgende Schedule, wobei b_i den Start der Transaktion i bezeichnet, $rl_i(A)/wl_i(a)$ eine Lese-/Schreibsperre für Objekt a anfordert und $ru_i(A)/wu_i(a)$ die Sperre wieder freigibt:

$b_1 b_2 rl_1(x) r_1(x) rl_2(x) b_3 wl_3(x) r_2(x) wl_1(z) wl_2(x) w_1(z) wu_1(z) ru_1(x)$
 $w_2(x) wu_2(x) ru_2(x) w_3(x) rl_3(y) wu_3(x) r_3(y) ru_3(y) c_1 c_3 c_2$

Ist dieser Schedule durch 2PL oder sogar durch striktes 2PL entstanden?

- c) Gegeben folgende Historie:

$r_1(a) r_5(a) w_3(b) r_2(a) r_5(c) r_3(c) r_4(d) w_2(a) w_5(e) r_4(b) w_1(c) w_4(d) r_2(d) w_2(d)$

Zeichnen Sie den Konfliktgraphen. Ist diese Historie serialisierbar? Falls ja, geben Sie alle möglichen seriellen Ablauffolgen an.

Aufgabe 3: Klassen von Historien

(1 P.)

Gegeben folgende Historien:

- $H_1 = r_1(a), r_1(b), r_2(a), w_1(a), c_1, w_2(a), c_2$
- $H_2 = r_2(a), w_2(a), r_1(a), r_1(b), w_1(c), c_1, r_2(d), c_2$
- $H_3 = r_2(a), w_2(a), r_2(b), w_2(b), c_2, r_1(d), r_1(a), w_1(d), c_1$
- $H_4 = r_2(b), r_1(a), w_1(a), r_2(a), c_1, r_2(c), w_2(c), c_2$
- $H_5 = r_2(b), r_1(a), w_1(a), r_2(a), r_2(c), w_2(c), c_2, r_2(d), w_1(d), c_1$
- $H_6 = r_2(a), r_1(a), w_1(a), c_1, w_2(a), c_2$

- a) Ermitteln Sie für diese Historien, ob sie rücksetzbar (RC) sind, kaskadierendes Rücksetzen vermeiden (ACA) oder strikt (ST) sind, und ob sie serialisierbar oder seriell sind.
- b) Illustrieren Sie an einer Historie, die in RC aber nicht in ACA liegt, wieso die Klasse ACA wünschenswert ist.
- c) Geben Sie zu den serialisierbaren Historien eine serielle Folge an, in der die Transaktionen ablaufen.

Aufgabe 4: Potpourri

(1 P.)

Um diese Aufgabe als gelöst zu markieren, müssen Sie 4 Teilaufgaben lösen.

- Geben Sie ein Beispiel an, bei dem eine rekursive SQL Anfrage mit `UNION` ein anderes Ergebnis liefert als mit `UNION ALL`.
- Wie muss ein Trigger aufgebaut sein, damit er das Datenbanksystem “lahmlegt”, wenn er ausgelöst wird? Illustrieren Sie an einem Beispiel.
- Gegeben zwei Relationen R und S wobei S zwei verschiedene Fremdschlüssel auf R hat, nämlich a und b , wobei $b = a$ oder $b = NULL$. Welches Verhalten bei Änderung der referenzierten Daten muss für diese Schlüssel gelten, damit es egal ist, in welcher Reihenfolge sie überprüft werden?
- Gegeben die MVDs $\text{PersNr} \twoheadrightarrow \text{Sprache}$ und $\text{PersNr} \twoheadrightarrow \text{ProgSprache}$ in der folgenden Relation:

PersNr	Sprache	ProgSprache
1	griechisch	C
1	lateinisch	Pascal
1	griechisch	Pascal
1	lateinisch	C

Welche Tupel müssen eingefügt werden, wenn der Mitarbeiter 1 die Sprache “englisch” und die Programmiersprache “Rust” lernt?

- Geben Sie ein Beispiel an, was schiefgehen kann, wenn eine Zerlegung verlustlos aber nicht abhängigkeiterhaltend ist.
- Gelten die beiden Regeln sowohl unter Mengen- als auch unter Multimengensemantik?
 $T \cap (R \cup S) = (T \cap R) \cup (T \cap S)$ und $R \cup (S \cap T) = (R \cup S) \cap (R \cup T)$
- Geben Sie eine äquivalenzerhaltende Transformationsregel für den Gruppierungsoperator γ (analog zu Vorlesung 3) an.