Ludwig-Maximilians-Universität München
Institut für Informatik
PD Dr. Peer Kröger
Daniel Kaltenthaler

SS 2014
08.07.2014

Probeklausur zur Vorlesung Datenbanksysteme II

Die folgenden Aufgaben orientieren sich in Komplexität und Umfang an einer 90-minütigen Klausur. Beachten Sie, dass für die Klausur am 25.07.2014 der gesamte Stoff der Vorlesung und der Übung relevant ist. Insbesondere können in der Klausur Teilgebiete der Vorlesungen geprüft werden, die in dieser Probeklausur nicht geprüft werden. Diese Probeklausur dient lediglich als Wiederholung des Stoffes, und als Orientierung der eigenen Leistung.

Die Probeklausur besteht aus 6 Aufgaben mit insgesamt 54 Punkten. Ein Richtwert zum Bestehen der Klausur liegt bei 50 Prozent der Gesamtpunkte. Bei dieser Probeklausur entspricht dies 27 Punkte.

Die erste Aufgabe enthält Fragen zum Vorlesungsstoff, in denen Sie Ihr Wissen in Multiple Choice-Aufgaben zeigen sollen. In den Aufgaben 2-6 werden praktische Aufgaben gestellt, die Sie bereits aus den Übungen kennen sollten. Diese Probeklausur behandelt nur die Aufgaben 2-6.

Aufgabe	mögliche Punkte
1. Multiple Choice	12
2. Logging und Recovery	13
3. Äquivalenz von Schedules	7
4. Join-Verfahren	5
5. Behandlung von Deadlocks	8
6. Logische Anfrageoptimierung	9
Summe:	54

Aufgabe 1 Gesamter Vorlesungsstoff (12 Punkte)
Multiple Choice

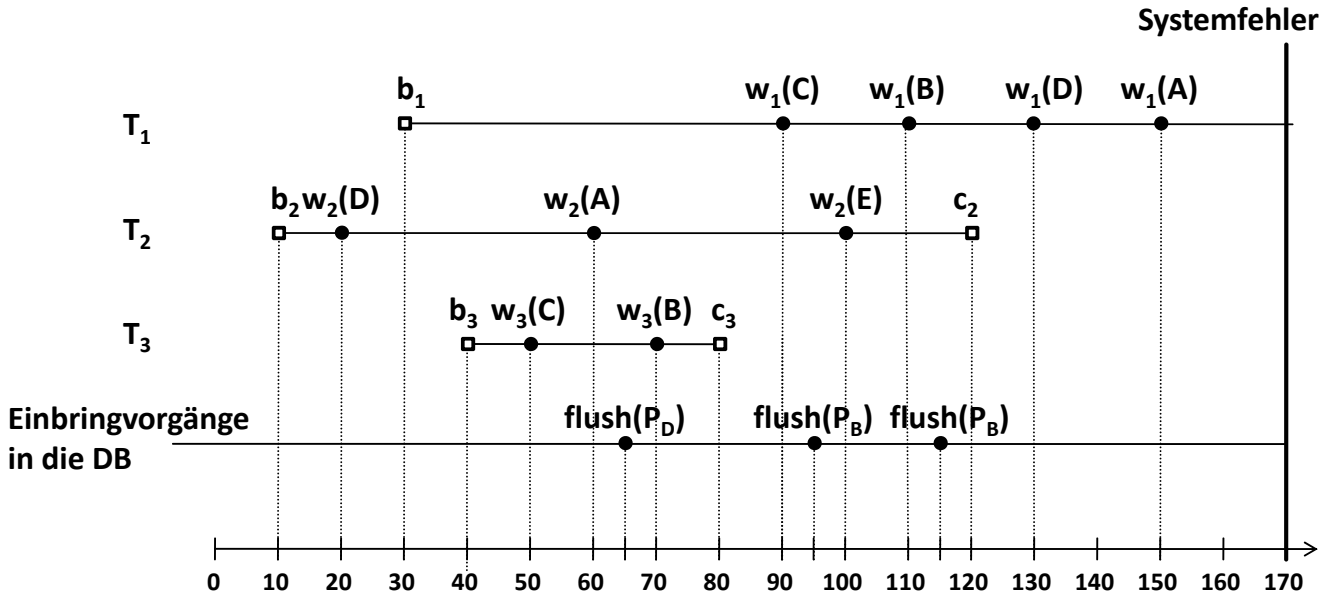
Entscheiden Sie, ob die folgenden Aussagen wahr oder falsch sind. Eine richtige Antwort bedeutet einen Punkt, eine falsche Antwort bedeutet einen Punkt Abzug. Eine unbeantwortete Frage bedeutet 0 Punkte. Sie können in dieser Aufgabe maximal 12 Punkte und minimal 0 Punkte erreichen.

[...]

Aufgabe 2 Logging (8+5 Punkte)

Logging und Recovery

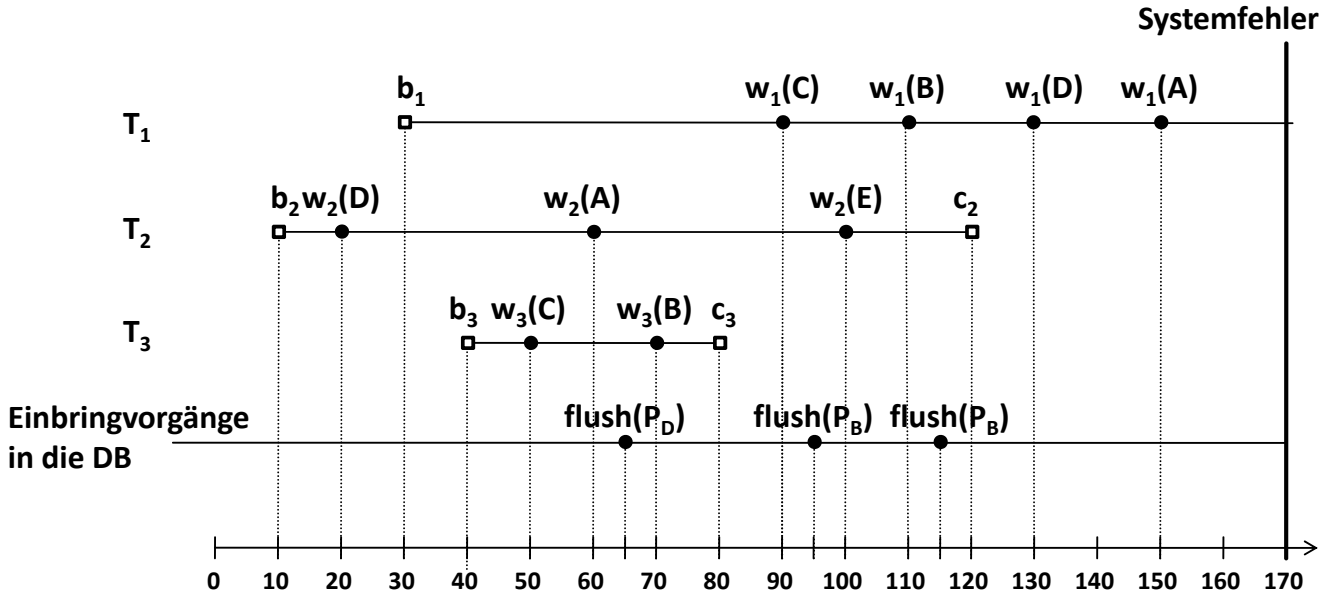
Gegeben sei ein DBMS, das die parallel laufenden Transaktionen T_1, T_2 und T_3 verwaltet. Dabei ändert T_1 die Datenelemente A, B, C und D, T_2 die Datenelemente A, D und E und schließlich T_3 die Datenelemente B und C. Das Datenelement $X \in \{A, B, C, D, E\}$ sei in Seite P_X enthalten. Weiterhin werden die modifizierten Seiten P_D, P_B und P_B jeweils zum Zeitpunkt 65, 95 und 115 aus dem DB-Puffer verdrängt und in die DB eingebracht. Zum Zeitpunkt 170 tritt ein Systemfehler auf.



(a) Führen Sie unter Beachtung der nachfolgenden Punkte die in der Abbildung gezeigten Aktionen der Transaktionen nacheinander durch und vervollständigen Sie dabei die Logeinträge in der Tabelle auf der nächsten Seite.

- Einbringstrategie: Non-Atomic
- Seitenersetzungsstrategie: Steal
- Ausschreibestrategie: No-Force
- Es werden indirekte Sicherungspunkte benutzt, der letzte Sicherungspunkt vor dem Systemausfall war zum Zeitpunkt 0 bereits erfolgreich abgeschlossen.
- Während der Ausführung der Transaktionen werden keinerlei Sicherungspunkte gesetzt.
- Zu Beginn hat die LSN den Wert 0 und alle Seiten-LSNs werden auf 0 gesetzt.
- Hat ein Log-Eintrag keinen Vorgänger, so wird $PrevLSN$ entsprechend mit 0 initialisiert.
- *Es soll so spät wie möglich und so wenig wie möglich ausgeschrieben werden.*
- Vermerken Sie, wenn das Hinzufügen der Log-Einträge zur Log-Datei aufgrund des WAL-Prinzips oder der COMMIT-Regel erfolgt ist.
- Benutzen Sie dabei folgende Notation für die Log-Information:
  BOT Transaktionsbeginn
  EOT Transaktionsende
  $U/R(X)$ UNDO-/REDO-Information für Datenelement X

Zeit *(LSN)*	Aktion	Änderung DB-Puffer *(Page-ID, LSN)* *(Hauptspeicher)*	Änderung in der DB *(Page-ID, LSN)* *(Platte)*	Log-Eintrag im Log-Puffer *(LSN, TA-ID, Page-ID, R(X)/U(X), PrevLSN)* bzw. *(LSN, TA-ID, BOT/EOT, PrevLSN)* *(Hauptspeicher)*	Zur Log-Datei hinzugefügte Log-Einträge *(LSNs)* *(Platte)*
10	b_2			10, T_2, BOT, 0	
20					
30					
40					
50					
60					
65					
70					
80					
90					
95					
100					
110					
115					
120					
130					
150					



(b) Wie sieht der Inhalt von Puffer und Datenbank zum Zeitpunkt des Systemfehlers (170) aus?

Seite	PageLSN (Puffer)	PageLSN (Platte)
P_A		
P_B		
P_C		
P_D		
P_E		

Aufgabe 3 Synchronisation (7 Punkte)

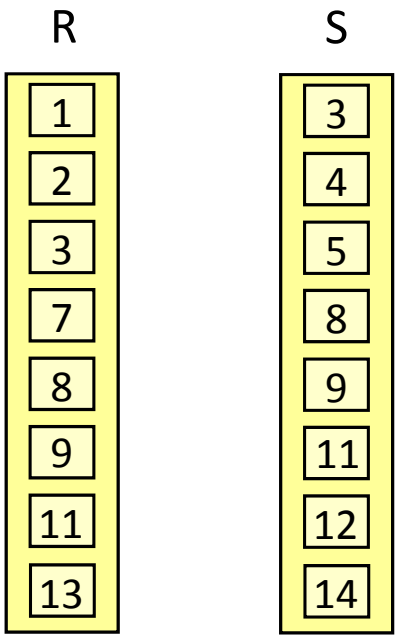
Äquivalenz von Schedules

Geben Sie für den folgenden Schedule S den Abhängigkeitsgraphen sowie, wenn möglich, einen äquivalenten seriellen Schedule an.

$$\begin{aligned} S \quad = \quad & (w_3(a), r_3(d), r_2(b), w_2(c), w_5(d), r_5(a), r_4(d), r_5(c), \\ & w_3(b), w_1(d), w_4(c), r_1(a), r_4(b)) \end{aligned}$$

Aufgabe 4 Anfragebearbeitung (1+2+2 Punkte)
Join-Verfahren

Es soll der Equi-Join der im Folgenden abgebildeten Relationen R und S berechnet werden.



(a) Wieviele Schlüsselvergleiche sind zur Berechnung eines **Nested-Block-Loop Join** notwendig, d.h. wieviele Tupel müssen auf Erfüllung des Joinprädikates hin untersucht werden?
Begründen Sie Ihre Aussage!

(b) Führen Sie den Join mittels des **Sort-Merge-Join** durch. Veranschaulichen Sie alle notwendigen Schlüsselvergleiche beim Durchlaufen der Relationen mit Hilfe der Matrix-Notation. Wieviele Vergleiche sind insgesamt notwendig?

(c) Führen Sie den Join mittels des **einfachen Hash-Join** mit der Hashfunktion $h(x) = x \bmod 3$ durch. Veranschaulichen Sie alle notwendigen Schlüsselvergleiche beim Durchlaufen der Relationen mit Hilfe der Matrix-Notation. Wieviele Vergleiche sind insgesamt notwendig?

Aufgabe 5 Synchronisation (1+3+4 Punkte)
Behandlung von Deadlocks

Gegeben sei folgende zeitliche Reihenfolge R, in der die Transaktionen T_1, T_2 und T_3 Sperren auf den Objekten $\{x, y, z\}$ anfordern:

$$R = (L_1(z), L_2(x), L_3(y), L_1(y), L_2(z), L_1(x), L_3(x))$$

Nehmen Sie an, dass es sich bei T_1 um die älteste Transaktion und bei T_3 um die jüngste Transaktion handelt.

(a) Erstellen Sie den Wartegraphen für den legalen Schedule der sich aus R ergibt und prüfen Sie, ob Verklemmungen vorliegen.

(b) Wie werden die Transaktionen in R bei Anwendung des Wound-Wait Zeitstempelverfahrens behandelt?

(c) Wie werden die Transaktionen in R bei Anwendung des Wait-Die Zeitstempelverfahrens behandelt?

Aufgabe 6 Relationale Anfragebearbeitung (3+2+4 Punkte)
Logische Anfrageoptimierung

Gegeben seien folgende drei Relationen aus der Datenbank zur Fußball-Europameisterschaft 2012:

Team

Key	Land	EM-Titel	Gruppe
CRO	Kroatien	0	C
CZE	Tschechien	1	A
DEN	Dänemark	1	B
ENG	England	0	D
ESP	Spanien	3	C
FRA	Frankreich	2	D
GER	Deutschland	3	B
GRE	Griechenland	1	A
IRL	Irland	0	C
ITA	Italien	1	C
NED	Niederlande	1	B
POL	Polen	0	A
POR	Portugal	0	B
RUS	Russland	1	A
SWE	Schweden	0	D
UKR	Ukraine	0	D

Stadion

Key	Stadt	Plaetze	Land
S1	Breslau	42.771	Polen
S2	Charkiw	38.633	Ukraine
S3	Danzig	41.582	Polen
S4	Donezk	51.504	Ukraine
S5	Kiew	70.050	Ukraine
S6	Lemberg	34.915	Ukraine
S7	Posen	43.090	Polen
S8	Warschau	58.500	Polen

Spiel

Team1	Team2	Stadion	Ergebnis
CZE	POR	S8	0:1
GER	GRE	S3	4:2
ESP	FRA	S4	2:0
ENG	ITA	S5	2:4

Das folgende SQL-Statement liefert die Ländernamen aller Teams, die ein Spiel in einem Stadion in der Ukraine bestritten haben.

```
SELECT t.Land
FROM Team t, Spiel sp, Stadion st
WHERE (t.Key = sp.Team1
OR t.Key = sp.Team2)
AND sp.Stadion = st.Key
AND st.Land = 'Ukraine';
```

(a) Zeichnen Sie den zur obigen Anfrage gehörenden kanonischen Operatorbaum. Geben Sie dabei die Selektionen einzeln an. Die Berechnung von Tupelzahlen ist nicht notwendig.

Team

Key	Land	EM-Titel	Gruppe
CRO	Kroatien	0	C
CZE	Tschechien	1	A
DEN	Dänemark	1	B
ENG	England	0	D
ESP	Spanien	3	C
FRA	Frankreich	2	D
GER	Deutschland	3	B
GRE	Griechenland	1	A
IRL	Irland	0	C
ITA	Italien	1	C
NED	Niederlande	1	B
POL	Polen	0	A
POR	Portugal	0	B
RUS	Russland	1	A
SWE	Schweden	0	D
UKR	Ukraine	0	D

Stadion

Key	Stadt	Plaetze	Land
S1	Breslau	42.771	Polen
S2	Charkiw	38.633	Ukraine
S3	Danzig	41.582	Polen
S4	Donezk	51.504	Ukraine
S5	Kiew	70.050	Ukraine
S6	Lemberg	34.915	Ukraine
S7	Posen	43.090	Polen
S8	Warschau	58.500	Polen

Spiel

Team1	Team2	Stadion	Ergebnis
CZE	POR	S8	0:1
GER	GRE	S3	4:2
ESP	FRA	S4	2:0
ENG	ITA	S5	2:4

```
SELECT t.Land
FROM Team t, Spiel sp, Stadion st
WHERE (t.Key = sp.Team1
OR t.Key = sp.Team2)
AND sp.Stadion = st.Key
AND st.Land = 'Ukraine';
```

(b) Welche Optimierungsmöglichkeiten ergeben sich?

(c) Geben Sie den resultierenden Operatorbaum an. Die Berechnung von Tupelzahlen ist nicht notwendig.