Fachbereich für Computerwissenschaften

Prof. Dr. Nikolaus Augsten
Jakob-Haringer-Str. 2
5020 Salzburg, Austria
Telefon: +43 662 8044 6347
E-Mail: nikolaus.augsten@sbg.ac.at

UNIVERSITÄT SALZBURG

Datenbanken Vertiefung
Wintersemester 2017/2018

Prüfung
02.02.2018

Name: _____ **Matrikelnummer:** _____

Hinweise

- Bitte überprüfen Sie die Vollständigkeit des Prüfungsbogens (9 nummerierte Seiten).
- Schreiben Sie Ihren Namen und Ihre Matrikelnummer auf jedes Blatt des Prüfungsbogens und geben Sie alle Blätter ab.
- Grundsätzlich sollten Sie alle Antworten auf den Prüfungsbogen schreiben.
- Sollten Sie mehr Platz für eine Antwort benötigen, bitte einen klaren Verweis neben die Frage auf die Seitennummer des zusätzlichen Blattes setzen.
- Keinen Bleistift verwenden. Keinen roten Stift verwenden.
- Verwenden Sie die Notation und die Lösungsansätze, die während der Vorlesung besprochen wurden.
- Aufgaben mit mehr als einer Lösung werden nicht bewertet.
- Als Unterlage ist ein beliebig (auch beidseitig) beschriftetes A4-Blatt erlaubt.
- Zeit für die Prüfung: 90 Minuten

Unterschrift _____

Korrekturabschnitt — Bitte frei lassen

Aufgabe	1	2	3	4	5	6	7	8	Summe
Maximale Punkte	1	1	1	1	1	1	1	1	8
Erreichte Punkte									

Exercise 1 — 1 Point

Identify which of the properties: **serial**, **conflict serializable**, **recoverable**, **cascadeless**, are fulfilled by the following schedule. If a property is not fulfilled, explain why.

T1:	T2:	T3:	T4:
			write(B)
		read(C)	
read(A)			
			COMMIT
	read(B)		
	write(A)		
	COMMIT		
		read(A)	
read(B)			
		write(A)	
write(B)			
COMMIT			
		read(B)	
		COMMIT	

Name: Matrikelnummer:

Exercise 2 — 1 Point

Is the following schedule **conflict serializable**? Draw the precedence graph to verify. If it **is not**, explain why. If it **is**, give an equivalent serial schedule.

T1:	T2:	T3:	T4:
	write(C)		
			read(B)
		read(C)	
		write(C)	
			read(A)
read(C)			
			write(A)
	read(A)		
			write(B)
read(B)			
		write(A)	

Exercise 3 — 1 Point

Is the following schedule in a **deadlock state**? Draw the wait-for graph to verify. If it **is** in a a deadlock state, propose a way to recover from the deadlock.

T1:	T2:	T3:
lock-S(A)		
read(A)		
		lock-X(B)
		lock-S(A)
	lock-X(C)	
		read(A)
lock-X(B)		
	read(C)	
	lock-X(A)	
		lock-S(C)

Name: Matrikelnummer:

Exercise 4 — 1 Point

Does the following schedule adhere to the **two phase locking (2PL)** protocol (without the lock conversions)? If **yes**, show the schedule with all required lock and unlock instructions. If **not**, explain why 2PL is violated.

```
T1:          T2:          T3:
_____________________________

                          read(C)

read(A)

write(B)

             read(B)

                          write(A)

_____________________________
```

Exercise 5 — 1 Point

Consider the following schedule and the **validation based** scheduler. The timestamps correspond to the validation order, i.e., `TS(Ti)=validation(Ti)`.

T1:	T2:	T3:	T4:
start			
	start		
			start
		start	
validate			
finish			
	validate		
	finish		
		validate	
		finish	
			validate
			finish

The objects in the database that can be read or written are: A, B, C, D, E, F. The read and write sets of the transactions are:

T1: R-set(T1)={A,B}, W-set(T1)={C,D}
T2: R-set(T2)={A,C}, W-set(T2)={D,E}
T3: R-set(T3)={C,E}, W-set(T3)={B,F}

Answer the following questions:

1. Does `T2` successfully validate? Explain which conditions are satisfied or which one is violated.
2. Suppose that the validation of T4 and all previous validations succeeded. What could have been the largest read set of T4?

Name: Matrikelnummer:

Exercise 6 — 1 Point

Consider the following log.

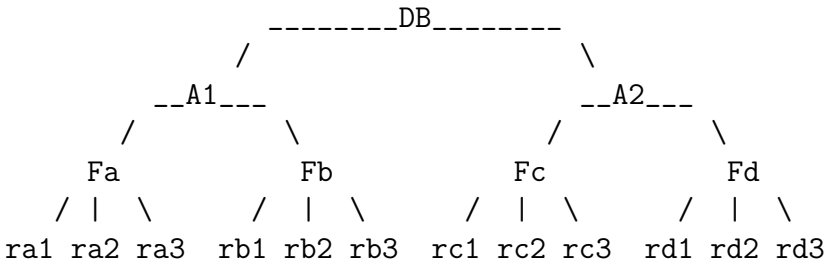
1. Which event has triggered the log record in line 10?
2. What would happen during the recovery if the system crashes at the end of the log? Indicate the generated log records.

```
 1 <T1, start>
 2 <T1, D, 10, 20>
 3 <T1, commit>
 4 <T4, start>
 5 <T4, A, 0, 20>
 6 <checkpoint, {T4}>
 7 <T2, start>
 8 <T2, C, 15, 25>
 9 <T3, start>
10 <T4, A, 0>
11 <T3, B, 20, 30>
12 <T2, commit>
-------CRASH---------
```

Exercise 7 1 Point

Consider the following database object hierarchy and the multi-granularity protocol.

Hierarchy of database elements:



Given are transactions:

T1: Read record rc1.
T2: Modify all records in file Fb.
T3: Read all files in area A2.
T4: Modify record rc3.

1. Indicate the respective locks obtained by transactions T1, T2, T3, T4.

2. Which pairs of transactions T1, T2, T3, T4 **cannot** be executed concurrently?

Name: Matrikelnummer:

Exercise 8 — 1 Point

Give an example of a schedule that adheres to the **multiversion timestamp ordering** protocol **but not** to the **timestamp ordering** protocol. Explain your answer.