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 — Prüfung
Wintersemester 2015/2016 — 03.06.2016

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

Is the following schedule conflict serializable? Draw a precedence graph to verify.

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

Name: Matrikelnummer:

Exercise 2 — 1 Point

Show the commit order (by inserting the commit commands into the schedule) for the following schedule such that the schedule is **cascadeless**.

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

Exercise 3 — 1 Point

Can the following schedule be the output of a two-phase locking scheduler? If so, show the schedule with all required lock and unlock instructions. Otherwise explain why. Could it be the output of a *strict* two-phase locking scheduler?

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

Name: Matrikelnummer:

Exercise 4 — 1 Point

Is the following schedule valid under the timestamp-ordering protocol? If yes, show the timestamps for the affected data items for each operation. If not, mark the first problematic operation and explain the problem.

TS(T1)=1, TS(T2)=2, TS(T3)=3

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

Exercise 5 1 Point

With the following starting values:

A=10, B=20, C=30, D=40, E=50, F=60

write the log file (physical logging) for the following schedule including the log records generated during recovery.

```
T1:          T2:          T3:
start
read(A)
A:=A-5
write(A)
             start
             read(C)
read(B)
B:=B-5
write(B)
read(D)
             C:=C+15
             write(C)
             COMMIT
----------CHECKPOINT------------
                          start
                          read(E)
D:=D-10
write(D)
                          E:=E-15
                          write(E)
                          read(F)
COMMIT
                          F:=F-15
                          write(F)
------------CRASH---------------
```

Name: Matrikelnummer:

Exercise 6 1 Point

Consider the following log. What would happen during the recovery if the system crashes at the end of the log?

```
<T2, start>
<T2, O1, operation-begin>
<T2, X, 200, 250>
<T2, O1, operation-end, (X, -50)>
<T1, start>
<T1, Y, 150, 100>
<T2, Z, 10, 150>
<T1, O2, operation-begin>
<T1, X, 250, 50>
------------CRASH----------------
```

Exercise 7 — 1 Point

Consider a multi-granularity locking protocol with lock modes S, X, IS, IX, SIX. The following object hierarchy is given:

- root R with blocks A and B,
- block A contains records a1 and a2,
- block B contains records b1, b2, and b3.

Write the list of locks (objects being locked and lock type) required to execute the following transactions. Start with the root.

T1: modify record a2

T2: read all records in block B

Can T1 and T2 be executed concurrently? Why?

Name: Matrikelnummer:

Exercise 8 — 1 Point

Answer the following questions (be concise).

- What does Read Committed stand for? **(0.25 Punkt)**
- What is a deadlock (give and example) and how can it be detected? **(0.5 Punkt)**
- What is stable storage and how is it implemented? **(0.25 Punkt)**