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

PARIS LODRON UNIVERSITÄT SALZBURG

Datenbanken Vertiefung — Prüfung
Wintersemester 2020/2021 — 15.03.2020

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									

Name: Matrikelnummer:

Exercise 1 — 1 Point

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

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

Name: Matrikelnummer:

Exercise 2 — 1 Point

Consider the following schedule.

(a) Which transaction has to abort to trigger **cascading rollback**?

(b) Insert `COMMIT` instructions such that the schedule becomes cascadeless.

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

Name: Matrikelnummer:

Exercise 3 — 1 Point

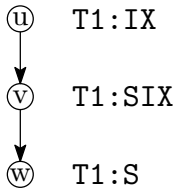
Consider the following schedule and a two-phase locking scheduler with lock conversions. Insert lock and unlock instructions such that all read and write operations in the schedule can be performed in this order.

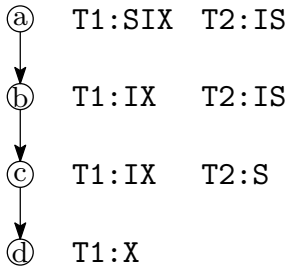
```
T1:        T2:        T3:        T4:
_____________________________________

read(B)

                      write(A)

                                 read(A)

           read(A)

                                 read(B)

                                 write(A)

write(B)

_____________________________________
```

Name: Matrikelnummer:

Exercise 4 — 1 Point

Consider the **multiple granularity** locking scheme with intentional locks. The following figures show data hierarchies. At each node, there is a transaction and a lock type it holds. For each figure, identify which rules of the scheme are violated or state that all rules are obeyed.





Name: Matrikelnummer:

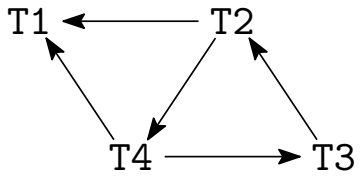
Exercise 5 — 1 Point

Consider the following graph in which a directed edge `Ti`→`Tj` indicates that `Ti` requests a lock currently held by `Tj`.

(a) Which of the transactions would be rolled back under the **wound-wait** deadlock prevention strategy?

(b) With what timestamp is the rolled-back transaction restarted and why?

Transaction timestamps are equal to their numeric identifiers.



Name: Matrikelnummer:

Exercise 6 — 1 Point

Consider the **snapshot isolation** concurrency control scheme and the following transactions.

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

(a) Can `T1` and `T2` both commit? Explain why.

(b) What would using `select ... for update` change in this scenario?

Name: Matrikelnummer:

Exercise 7 — 1 Point

Consider the following schedule. Indicate what happens at each step when the schedule is processed by a **multiversion timestamp-ordering** scheduler. The transactions start in order with TS(T1)=1, TS(T2)=2, TS(T3)=3, TS(T4)=4. Assume that the first write of data item `A` succeeds.

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

Name: Matrikelnummer:

Exercise 8 — 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
                          start
                          read(E)
D:=D-10
write(D)
COMMIT
----------CHECKPOINT------------
             write(C)
             COMMIT
                          E:=E-15
                          write(E)
                          read(F)
                          F:=F-15
------------CRASH---------------
```