Fachbereich Informatik

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

PARIS LODRON UNIVERSITÄT SALZBURG

Datenbanken Vertiefung
Wintersemester 2022/2023

Prüfung
20.04.2023

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**? 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(C)			
			read(A)
		read(C)	
		read(B)	
			read(C)
		write(A)	
		write(B)	
write(B)			

Exercise 2 — 1 Point

Consider the following schedule.

(a) Which transaction must abort to trigger a **cascading rollback** of all other transactions?

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

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

Exercise 3 — 1 Point

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

```
T1:         T2:         T3:
_____________________________

read(X)

read(Y)

                        read(X)

                        write(X)

            read(X)

read(Z)

write(Z)

COMMIT

            write(Y)

            COMMIT

                        read(V)

                        COMMIT
_____________________________
```

Exercise 4 — 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)={B,C}, W-set(T3)={B,F}

Answer the following questions:

1. Does `T3` 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?

Exercise 5 — 1 Point

Consider the following **multiversion two-phase locking** schedule. Assume that the starting value of `ts-counter` and the initial versions of data items A and B are 0. T_i and T_j are read-only transactions and start with their first read operation.

For each row in the schedule, indicate any

- changes to locks,
- waiting transactions,
- newly created data items,
- changes to versions of data items,
- and the data item version for reads.

Also, provide

- the timestamps of the read-only transactions
- and the final value of `ts-counter`.

Ti:	Tj:	Tk:	Tl:
		write(A)	
	read(A)		
			read(A)
	read(B)		
		commit	
			write(B)
read(A)			
			commit
read(B)			

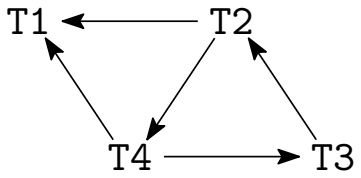
Exercise 6 — 1 Point

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.



Exercise 7 — 1 Point

With the initial values A=50, B=100, C=150, D=200, write the log file (physical logging) for the following schedule including the log records generated during recovery.

```
T1:          T2:          T3:
start
read(A)
                          start
                          read(C)
A:=A+50
write(A)
                          read(B)
                          C:=C-B
COMMIT
----------CHECKPOINT------------
             start
             read(A)
             read(D)
                          write(C)
             D:=D-A
             write(D)
------------CRASH---------------
```

Exercise 8 — 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?