Department of Computer Science

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

PARIS
LODRON
UNIVERSITÄT
SALZBURG

Advanced Databases — Exam
Winter Semester 2025/26 — 17.04.2026

Name: _____ **Student ID:** _____

Hints

- Check whether you received all pages of the exam (9 pages).
- Write your name or your student ID on each sheet of the exam and hand in all pages.
- All answers are expected to be written on the exam sheets.
- Clearly highlight and enumerate additional pages that are used for longer answers. Match your text with the according exercise.
- Only use pencils that are permanent and non-red colored.
- Use the notation and techniques discussed in the lecture.
- Exercises with more than one solution are not graded.
- **Multiple-Choice**: Wrong answers result in point deductions!
- You are allowed to use one A4 sheet with your personal notes (both sides, hand written or printed).
- Exam duration: 90 minutes

Signature _____

Grading — Filled by the examiner

Exercise	1	2	3	4	5	6	7	8	Sum
Total points	1	1	1	1	1	1	1	1	8
Points reached									

Exercise 1 — 1 Point

Let S_1 with transactions $T_1, \ldots, T_k$ and S_2 with transactions $T_{k+1}, \ldots, T_n$ be two arbitrary conflict-serializable schedules. **Sketch** why the schedule $S_1 \circ S_2$, resulting from the concatenation (all of S_1 is executed before S_2) of S_1 and S_2, is also **conflict-serializable**.

Exercise 2 — 1 Point

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

```
T1:        T2:        T3:        T4:
──────────────────────────────────────

read(B)

                      read(A)

                                 read(A)

           read(A)

write(B)

           read(C)




                                 read(C)

                                 write(A)

                      write(B)
──────────────────────────────────────
```

Exercise 3 — 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 4 — 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 the use of `select ... for update` for all read operations change in this scenario?

Exercise 5 — 1 Point

Consider the following schedule and a **validation based** scheduler.

T1:	T2:	T3:	T4:
start			
	start		
	validate		
	finish		
		start	
validate			
finish			
			start
		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)={E}, W-set(T2)={A,F}
T3: R-set(T3)={E,F}, W-set(T3)={B,C}

Answer the following questions:

(1) Does T3 successfully validate? Explain which conditions are satisfied or violated.
(2) Suppose that the validation of T4 and all previous validations succeeded. What could have been the largest read set of T4?

Exercise 6 — 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)
                          F:=F-15
                          write(F)
------------CRASH---------------
```

Exercise 7 — 1 Point

Consider a new locking protocol 2PL$^+$ that extends two-phase locking (2PL) and permits *downgrading* an exclusive lock to a shared lock in all phases of the protocol. In 2PL$^+$, a transaction that has an exclusive lock on an item `X` can use the operation `downgrade(X)` to downgrade its exclusive lock on `X` to a shared lock on `X`.

Prove or disprove the following statement: Protocol 2PL$^+$ ensures *conflict-serializable* schedules.

Exercise 8 — 1 Point

The following schedule is **conflict serializable**. Assign timestamps to the transactions T_i, T_j, T_k, T_ℓ such that the schedule is also valid under the **timestamp-ordering (TSO)** protocol. Show the validity of your assignment.

T_i:	T_j:	T_k:	T_l:
			read(B)
		read(A)	
		write(A)	
	read(A)		
			write(B)
read(B)			
		read(C)	
	write(C)		
read(C)			
write(A)			