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 — 28.01.2026

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

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

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

Exercise 2 — 1 Point

Consider the following relation $R[A, B, C]$.

A	B	C
1	2	3
1	3	4

Provide two transactions, each as a sequence of SQL statements, that can result in a phantom read when executed with transaction isolation level **read committed** if they are executed concurrently. Provide a sample output showing the phantom read. How can this problem be avoided?

Transaction 1:

```
BEGIN TRANSACTION ISOLATION LEVEL READ COMMITTED
```

```
COMMIT
```

Transaction 2:

```
BEGIN TRANSACTION ISOLATION LEVEL READ COMMITTED
```

```
COMMIT
```

Exercise 3 — 1 Point

Consider relation $R[\underline{A}, B, C]$ and multi-granularity locking. We have $|R| = 10^6$. A single file can store 10^3 records. A is the primary key of R. There is a sparse index on attribute B, i.e., R is physically sorted by attribute B. The values of A, B, and C have the following distributions:

- A is uniformly distributed between 1 and $2 \cdot 10^6$.
- B is uniformly distributed between 1 and 10^3.
- C is uniformly distributed between 1 and 10^5.

The hierarchy of granularities (from coarse to fine) is: (1) database, (2) area (storing a single and complete relation), (3) file (each storing 10^3 records of a relation), (4) record.

For the following SQL statements, state which lock modes should be used on which level of the hierarchy (including intention lock modes). Explain your answer.

Query 1

```
SELECT SUM(B) FROM R;
```

Query 2

```
UPDATE R SET C = 42 WHERE A = 1234;
```

Query 3

```
UPDATE R SET C = 42 WHERE B = 123;
```

Exercise 4 — 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. State if a transaction must be rolled back and also consider cascading rollbacks.

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

Exercise 5 — 1 Point

Either prove or disprove the following statement:

Every schedule allowed by timestamp ordering with ***Thomas' write rule*** *is conflict serializable.*

Exercise 6 — 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 7 — 1 Point

With the initial values A=100, B=200, C=300, write the log file for the following schedule. What happens during the recovery according to the recovery algorithm? Specify the resulting log records.

T1:	T2:	T3:
START		
		START
	START	
		read(C)
	read(B)	
read(A)		
	B:=B-90	
A:=A-40		
		C:=C+70
		write(C)
write(A)		
	write(B)	
		COMMIT
----------CRASH----------		

Exercise 8 — 1 Point

Is the following schedule valid under the **timestamp-ordering** protocol? Show the timestamps for the affected data items for each operation. If the schedule is not valid under timestamp ordering, 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(A)		