

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

---

Parallel and Distributed Data Management  
Summer Semester 2025

Midterm  
29.04.2025

---

Name: \_\_\_\_\_ Student ID: \_\_\_\_\_

---

Hints

---

- Check whether you received all pages of the exam (5 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.
- You are allowed to use one A4 sheet with your personal notes (single-sided, hand written or printed).
- Exam duration: 60 minutes

Signature \_\_\_\_\_

---

Grading

Filled by the examiner

---

Exercise	1	2	3	4	Sum
Total points	3	4	4	4	15
Points reached					

---

**Exercise 1 - Throughput and Response Time.****3 Points**

---

A database system must process transactions  $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$ . For each transaction, Table 1 shows the start time (when the transaction enters the system) and the execution time (the runtime of the transaction in the system when it is not interrupted).

Compute the average response time and the throughput for the following two task scheduling strategies where shorter transactions are given priority over longer transactions (i.e., the waiting queue is ordered by execution time):

1. No running transaction is aborted.
2. Abort the running transaction when a shorter transaction (wrt. its total execution time) enters the waiting queue. Aborted transactions must be restarted from scratch.

Transaction	Start Time	Execution Time
$t_1$	0	3
$t_2$	1	4
$t_3$	2	2
$t_4$	3	1

Table 1: Start and execution time of transactions.

---

**Exercise 2 - Partitioning using Histograms.****4 Points**

---

Consider the following relation consisting of 40 values between 0 and 99. The data should be partitioned onto four processors using range partitioning:

1. Build a histogram with bin size 10 for the relation, i.e., we have the bins

$$[0, 9], [10, 19], \dots, [80, 89], [90, 99]$$

2. Using only the histogram (i.e., using the **number of elements in each bin**), construct a balanced partitioning vector for the relation. Assume a uniform distribution inside each bin as discussed in the lecture.
3. Using your partition vector, how large are the resulting partitions? Do we suffer from partition skew? If yes, why and how could we prevent it?

$$R[A] = [3, 5, 7, 11, 12, 13, 17, 19, 25, 26, 27, 28, 29, 35, 42, 47, 52, 60, 61, 62, \\ 63, 64, 72, 78, 81, 82, 84, 85, 87, 88, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]$$

---

**Exercise 3 - *Parallel External Sort-Merge.***

---

**4 Points**

Execute parallel external sort-merge on 20 values in the range  $0, \dots, 99$  that are horizontally partitioned (round robin) on five processors,  $P_i$ ,  $0 \leq i \leq 4$ , in a shared-nothing environment. The processors  $P'_0$ ,  $P'_1$ , and  $P'_2$  support the sorting.

---

Exercise 4 - *Parallel Join*.4 Points

---

Given a system with 9 processing nodes  $p_i$ ,  $1 \leq i \leq 9$ , and the following two relations

$$R[A] = [4, 7, 13, 14, 16, 24, 25, 36, 44, 55, 57, 62, 68, 72, 78],$$

$$S[A] = [2, 7, 15, 21, 26, 32, 33, 36, 39, 41, 42, 53, 55, 56, 78],$$

where each number is an attribute value of A and forms a unary tuple. The relations are stored on nodes different from the processing nodes  $p_i$ .

(a) Find a parallel join technique for the following query (**ABS** returns the absolute value of a number) and argue why the technique is appropriate.

```
SELECT * FROM R, S
WHERE ABS(R.A - S.A) <= 3;
```

(b) Count

1. the number of tuples that must be transferred over the network,
2. the partial join result size per processing node.