# Data Management (A.A. 2024/25) – exam A of 05/06/2025

**Problem 1** Consider a scheduler $D$ that behaves as follows when processing an input schedule $S$: $D$ lets $S$ proceed, dynamically building the precedence graph $P(S)$ by adding nodes and edges when needed and never deleting nodes or edges, and acting only whenever it processes the commit action of a transaction $T_i$. When processing such action, it executes the commit action if $T_i$ is not involved in any cycle in $P(S)$, otherwise it aborts and rollbacks $T_i$. Let $S$ be any complete schedule on transactions $T_1, \ldots, T_n$, where the last action of each $T_i$ is the commit action $c_i$, let $S'$ be the schedule produced in output by $D$ when processing $S$, and let $S''$ be the schedule obtained from $S'$ by ignoring the actions of the transactions aborted by $D$.

1.1 Prove or disprove that $S$ is conflict serializable if and only if $S = S''$.

1.2 Prove or disprove that $S''$ is recoverable, and in case you disproved that $S''$ is recoverable, tell how you would modify $D$ in order to ensure recoverability of $S''$.

**Problem 2** Consider the following schedule $S$:

$$r_1(x)\ w_2(x)\ w_2(y)\ r_3(x)\ w_4(x)\ w_2(z)\ w_3(y)$$

2.1 Is $S$ a 2PL schedule with both shared and exclusive locks? Motivate your answers in detail.

2.2 Is $S$ a strict schedule? Motivate your answers in detail.

2.3 Describe the behavior of the timestamp-based scheduler when processing $S$, assuming that, initially, for each element $\alpha$ of the database, we have rts($\alpha$)=wts($\alpha$)=wts-c($\alpha$)=0, and cb($\alpha$)=`true`, and assuming that the subscript of each action denotes the timestamp of the transaction executing such action.

**Problem 3** Consider the relations `S(A,B)` and `R(B,C,D,E)`, where ($i$) both have `B` as key, ($ii$) `S` is stored in a heap with 60 pages (each page with 40 tuples) with an associated hash index whose search key is `B`, ($iii$) `R` is stored in a heap with 900 pages (each page with 20 tuples) and ($iv$) the buffer has 32 frames available. If your goal is to compute the natural join (equi-join on `B`) between `R` and `S` as efficiently as possible in terms of number of page accesses, which algorithm would you choose among:

3.1 block-nested loop,

3.2 multi-pass based on sorting,

3.3 index-based.

Explain your answer in detail so as to convince that you choice is the right one.

**Problem 4** Consider the relation `MEETING(code,topic,venue,city,date)`, with 800.000 tuples stored in a sorted file with search key `code` (which is also the key of the relation), and with an associated sorted index with search key `venue`. We know that no more than 300 meetings are held in the same venue, that every attribute and pointer in our system occupies 10 Bytes, and that the size of each page in our system is 1.000 Bytes. Consider the following operations (1) given a venue, compute the code of all meetings held in that venue, together with the corresponding topic; (2) insert a new meeting. For each of the two operations, tell which is the worst-case cost of its execution in terms of number of page accesses.

**Problem 5** (only for students who opted for **option 1**, i.e., who do **not** do the project)
Let $B$ be a relational database with relations `Student(id,age)`, `Exam(stid,ccode,mark)`, `Course(ccode,credits)`, `Teaches(pcode,ccode)`, `Prof(pcode,age)`, `Tutoring(pcode,stid,year)`, where ($i$) each exam is given by a student for a given course and with a given mark, ($ii$) each professor can teach several courses and each course can be taught by several professors, ($ii$) each professor can be the tutor of several students and each student can be tutored by several professors.

5.1 Describe how you would organize a property graph database $G$ in order to represent the relational database $B$. In particular, ($i$) specify how nodes, edges, labels, etc. of $G$ are used in order to capture the information stored in the tables of $B$ and ($ii$) choose a few tuples for the relations in $B$, and show the specific property graph database $G$ obtained by applying the chosen representation method.

5.2 Describe how you would organize a document database $D$ in order to represent the relational database $B$. In particular, ($i$) specify how collections, documents, etc. of $D$ are used in order to capture the information stored in the tables of $B$ and ($ii$) choose a few tuples for the relations in $B$, and show the specific $D$ obtained by applying the chosen representation method.