Data Management – exam of 18/02/2010

LAST NAME: ______________________ FIRST NAME: ______________________ STUDENT CODE: ______________________	I allow the publication of the grade I will get in this exam on the web page `http://www.dis.uniroma1.it/~lenzerini`, according to the "decreto legislativo 196/2003" (regarding the rules for the privacy of personal data) which I hereby confirm to know. Sincerely, *(Signature)* ______________________

Problem 1 Consider the following schedule

$$S = r_1(A)\, r_3(C)\, w_3(B)\, r_2(A)\, w_1(B)\, w_1(A)\, w_2(A)\, r_1(C)\, w_3(C)\, r_3(A)\, r_2(D).$$

1. Tell whether S is view-serializable or not. Explain the answer in detail.
2. If the answer to the previous question was yes, then show a serial schedule that is view-equivalent to S, otherwise exhibit a minimal set (i.e., a set of minimal cardinality) of actions of S to remove so that the remaining schedule is not only view-serializable, but also conflict-serializable.

Problem 2 Consider again the schedule S shown in Problem 1, and tell whether there is a way to insert the "commit" command for every transaction in S such that the resulting schedule S' is ACR (Avoid Cascading Rollback). If the answer is yes, then show the schedule S', otherwise explain the answer.

Problem 3 We define WOB to be the class of schedules defined as follows: a schedule S on transactions $T = \{t_1, \ldots, t_n\}$ belongs to WOB if and only if (i) S is a complete schedule constituted only by "write" actions, and (ii) there exists a function $\psi : T \rightarrow \{1..n\}$ such that for every pair $\langle \alpha, \beta \rangle$ of conflicting actions in S, where β is an action of transaction t_k occurring in S after the action α of transaction t_h, we have that $\psi(t_k) > \psi(t_h)$.

1. Provide the definition of view-equivalence and view-serializability.
2. By using *only* the definition of view-equivalence and view-serializability, and the definition of WOB, prove or disprove that every schedule in WOB is view serializable.

Problem 4 Consider the relation `MATCH(referee,date,hometeam,hostteam)`, updated monthly, that stores about 1.000.000 tuples, where each tuple $\langle r, d, h, t \rangle$ means that r was the referee of a match played at date d between teams h and t. We assume that (i) no referee can be in more than 20 matches with the same home team, (ii) the pages in our system contains 100 data entries, and 100 index entries, (iii) there is no good hash function for the search key $\langle$`referee`, `hometeam`$\rangle$, and (iv) the following are the most important queries on `MATCH`:

Query 1	**Query 2**
select `referee` from `MATCH` where `date` $\geq \alpha$ and `date` $\leq \beta$	select `hostteam`, `date` from `MATCH` where `referee` $= \gamma$ and `hometeam` $= \delta$

Tell which is the method for representing the relation `MATCH` you would choose in order to optimize the computation of the above queries, explaining in detail your answer. Also, tell how many pages are accessed during the execution of Query 2.

Problem 5 Let T be a relation such that $B = 666.666.667$ (number of pages), and $R = 100$ (number of records per page). Assuming that $D = 15ms$ (time for accessing a page), $C = 100ns$ (time to process one record), and that we have a clustered B*-tree index using alternative 1, with search key equal to the key of T, and fan-out equal to 100, tell which is the time needed for inserting a new tuple in T, explaining in detail your answer.