Advanced Database Final 2022
(Dr. Michael Nawar)

Question 1: Various Questions [NAZARI]:

- What are NoSQL Features?
- What is CAP? What is important for NoSQL databases?
- How does the Recovery Manager (Undo/Redo) maintain durability and atomicity?
- Difference and similarities between CAP and ACID ?
- What is the difference between C in CAP and ACID ?
- What is WAL with an example ?
- What is trivial no-steal/force and does it ensure durability and atomicity?
- What are NoSQL databases' different choices in concurrency control?

Question 2:

A- 2 Phase Locking:

- Does 2PL always guarantee no Deadlocks? If yes prove it, if no give example
- What is the type of 2PL with no deadlocks ?
- What is Strict 2 Phase Locking? What can we say about the schedule output from it ?
- Why can't we use 2PL with B+ Trees?

B- Given two schedules [example: w1(A), r2(B), r1(A)...]
For each schedule, draw the precedence graph, and state whether the schedule is serializable or not. If serializable, write its serial equivalent.

C- Given two schedules [example: w1(A), r2(B), r1(A)...]
For each schedule, does multiversion timestamp order ever abort?
Assume TS(Ti) = i, Under what circumstances will it abort?
[Note: We have to draw the table ourselves]

Question 3:

- Given the ARIES LOGs, and a crash happened, make a recovery
[Note: LOGs File contained Checkpoint with dumped Dirty Page Table and Transactions Table]
[Note: LOGs File contained some CLR entries but system crashed before it finished

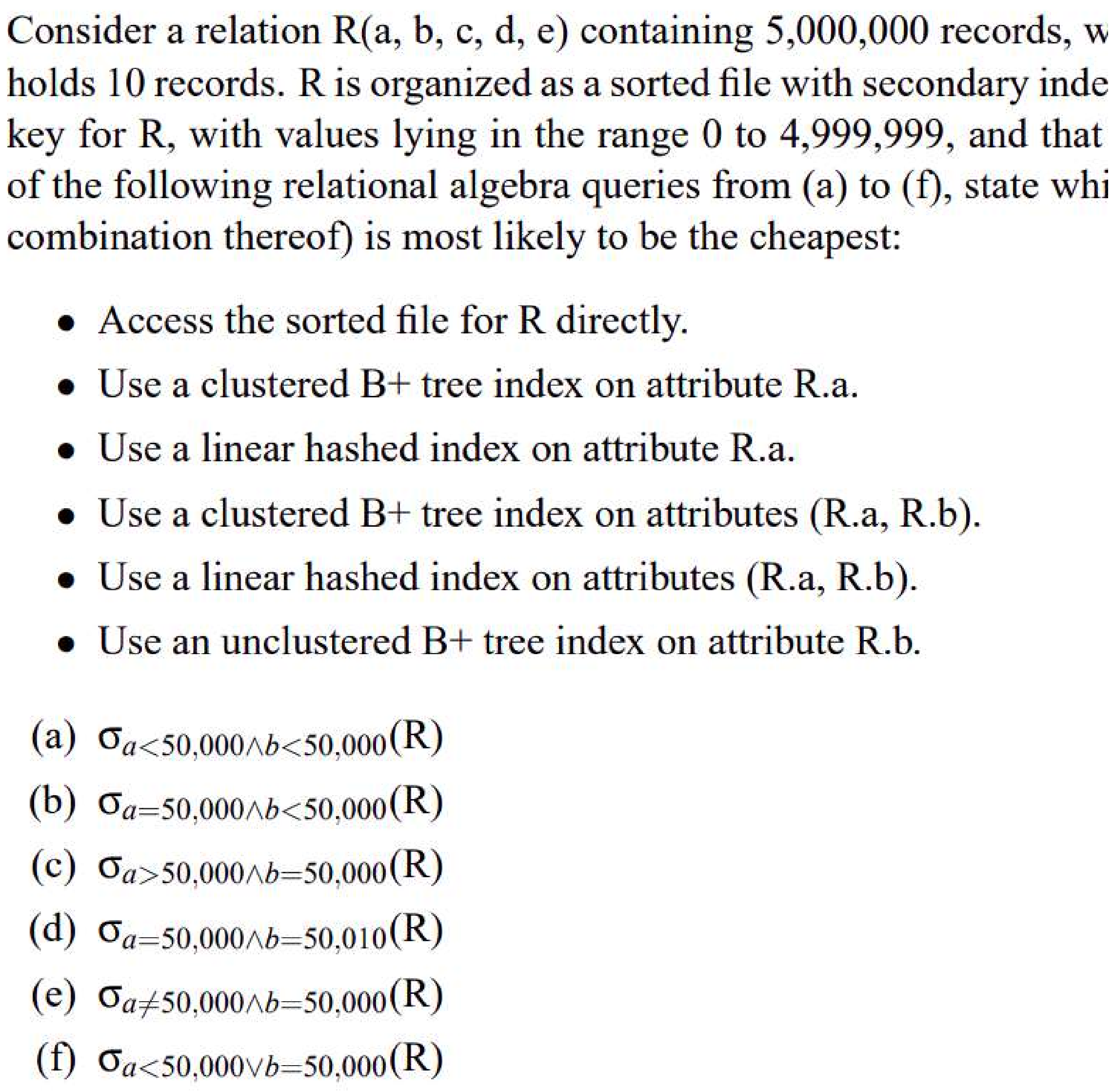
undo]

We are required to make the 3 phases (analysis, undo, redo) to recover from the crash, and to fill given tables for each phase

Question 4:

A- Given a database table schema, some indexes on it, 4 queries.
For each query you should say whether using indices will speedup the query or not **[4 points]**

B-



Consider a relation R(a, b, c, d, e) containing 5,000,000 records, w
holds 10 records. R is organized as a sorted file with secondary inde
key for R, with values lying in the range 0 to 4,999,999, and that
of the following relational algebra queries from (a) to (f), state whi
combination thereof) is most likely to be the cheapest:

- Access the sorted file for R directly.
- Use a clustered B+ tree index on attribute R.a.
- Use a linear hashed index on attribute R.a.
- Use a clustered B+ tree index on attributes (R.a, R.b).
- Use a linear hashed index on attributes (R.a, R.b).
- Use an unclustered B+ tree index on attribute R.b.

(a) $\sigma_{a<50{,}000 \wedge b<50{,}000}(R)$

(b) $\sigma_{a=50{,}000 \wedge b<50{,}000}(R)$

(c) $\sigma_{a>50{,}000 \wedge b=50{,}000}(R)$

(d) $\sigma_{a=50{,}000 \wedge b=50{,}010}(R)$

(e) $\sigma_{a\neq 50{,}000 \wedge b=50{,}000}(R)$

(f) $\sigma_{a<50{,}000 \vee b=50{,}000}(R)$

[Note: This question is taken from this link , solution]

C - 6 True or False **with Correction**
- In extendible hashing, if local depth == global depth and bucket is full, and we did an insertion, this doubles the directory
- Phantom effect is when the result of a query changes from time to time
- Chaining is when collisions occur in hashing, another hash function is applied after

collision.
- Is force same as immediate update in that they both don't need to REDO.
- Strict 2 phase locking prevents dirty read