2022

COMPUTER SCIENCE

Paper : CSMC-201

(Advanced Database Management System)

Full Marks : 70

The figures in the margin indicate full marks.

Candidates are required to give their answers in their own words as far as practicable.

Answer ***question no.*** **1**, ***question no.*** **2** and ***any four*** questions from the rest.

1. Answer ***any five*** questions out of the following : 2×5

(a) Consider relation schema R(A,B,C,D,E) with a set of FD, F = { AB → CDE, C → A, D → B}. State the normal form of the relation R.

(b) Consider the relation R(A,B,C). State the utility of finding A+.

(c) Why each leaf node in B+ tree index structure must have (P/2) values? P = No. of Pointers.

(d) Two transaction Tl and T2 are given as follows:

Tl: R1(B), R1(A), W1(A), W1(B) T2: R2(A), W2(A).

Find the possible number of concurrent schedules.

(e) If a multilevel index is to be constructed on an index file consisting of 1024 index blocks where the fan out of each index block is 10, what will be the number of levels in Index tree structure?

(f) State the importance of lock upgradation in a transaction.

(g) Why the conjunctive normal form expression is important in the context of relational query evaluation?

2. Answer ***any five*** questions from the following : 4×5

(a) Consider an initial empty B+ tree of order 6. Write the entries in the root node after inserting 10, 12, 23, 33, 48, 50, 15, 18, 20, 21, 31, 45, 47 and 52. Show the steps of expending the B+ tree.

(b) Explain the term 'Blind Write' through an example.

(c) If the join condition is not equality, what type of join algorithm will be efficient? (Sort-merge, hash join etc) why?

(d) Justify the given statement : "Concurrent execution of transactions is more important when data must be fetched from (slow) disk or when transactions are long, and is less important when data is in memory and transactions are very short."

(e) Describe a situation in which projection should precede selection in processing a project select query, and describe a situation where the opposite processing order is better.

(f) Consider the following two transactions :
T1 : read(A); read(B); if A = 0 then B : = B + 1; write(B).
T2 : read(B); read(A); if B = 0 then A : = A + 1; write(A).
Add lock and unlock instructions to transactions T1 and T2, so that they observe the two-phase locking protocol. Can the execution of these transactions result in a deadlock?

(g) Explain the lifecycle of a transaction.

3. (a) Consider the three transactions T1, T2, and T3, and the schedules S1 and S2 given below. Draw the serializability (precedence) graphs for S1 and S2, and state whether each schedule is serializable or not. If a schedule is serializable, write down the equivalent serial schedule(s).
T1: r1 (X); r1 (Z); w1 (X);
T2: r2 (Z); r2 (Y); w2 (Z); w2 (Y);
T3: r3 (X); r3 (Y); w3 (Y);
S1: r1 (X); r2 (Z); r1 (Z); r3 (X); r3 (Y); w1 (X); w3 (Y); r2 (Y); w2 (Z); w2 (Y);
S2: r1 (X); r2 (Z); r3 (X); r1 (Z); r2 (Y); r3 (Y); w1 (X); w2 (Z); w3 (Y); w2 (Y);

(b) Explain, through an example that how the following differ:
Fragmentation transparency, replication transparency, and location transparency. 5+5

4. (a) An airline reservation system allows many customers to book tickets simultaneously. What are the concurrency related problems that you may encounter in absence of concurrency control mechanism? State a solution to overcome the problem.

(b) Consider the join operation between relation R and S based on the attributes R.a and S.b. The following information is given about the relations to be joined. The cost metric is the number of page I/Os and the cost of writing out the result should be uniformly ignored.

- Relation R contains 10,000 tuples and has 10 tuples per page.
- Relation S contains 2,000 tuples and also has 10 tuples per page.
- Attribute *b* of relation S is the primary key for S.
- Both relations are stored as simple heap files.
- Neither relation has any indexes built on it.
- 52 buffer pages are available.

What is the cost of joining R and S using a page-oriented simple nested loop join? What is the cost of joining R and S using a block nested loop join? If clustered B+ indexes existed on R:a and S:b, is indexed nested loop join provide a cheaper solution than block nested loop join? Explain. 4+6

5. (a) Suppose that a B+ tree index on (branch-name, branch-city) is available on relation branch. What would be the best way to handle the following selection?
σ(branch-city<"Brooklyn") ∧ (assets<5000) ∧ (branch-name = "Downtown")(branch)

(b) What is the system log used for? What are the typical kinds of records in a system log? What are transaction commit points, and why are they important? Explain through an example.

(c) Why is strict or rigorous two-phase locking often preferred? 3+(1+2+2)+2

6. (a) In multiple-granularity locking, what is the motivation behind proposing implicit and explicit locking? Discuss through an example.

(b) When a transaction is rolled back under timestamp ordering, it is assigned a new timestamp? Why can it not simply keep its old timestamp?

(c) Give the DTD for an XML representation of the following nested-relational schema :
Emp = (ename, ChildrenSet setof(Children), SkillsSet setof(Skills))
Children = (name, Birthday)
Birthday = (day, month, year)
Skills = (type, ExamsSet setof(Exams))
Exams = (year, city) 4+3+3

7. (a) State CAP Theorem.

(b) "Availability of Buffer's number affects the performance of external sorting" — comment and justify.

(c) Differentiate between structured, semi-structured and unstructured data through an example.

(d) Mention the advantage(s) behind using semi-join operator in distributed query processing. 2+3+3+2

8. (a) Consider the following figure 1.

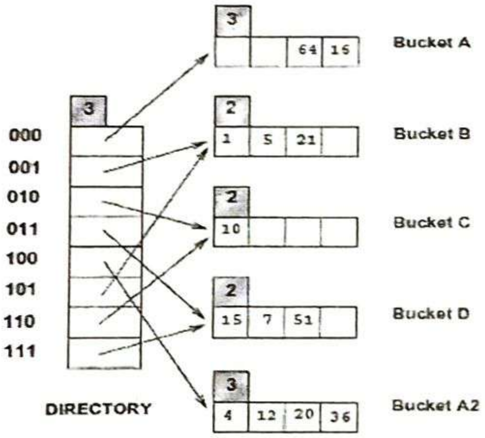

Figure 1

- What can you say about the last entry that was inserted into the index if you know that there have been no deletions from this index so far?
- Suppose you are told that there have been no deletions from this index so far. What can you say about the last entry whose insertion into the index caused a split?
- Show the index after inserting an entry with hash value 68.
- Show the original index after inserting entries with hash values 17 and 69.

(b) Consider the validation-based concurrency-control scheme. Show that by choosing validation (Ti), rather than Start (Ti), as the timestamp of transaction Ti, we can expect better response time provided that conflict rates among transactions are indeed low (1+1+1+2)+5.

$T_B \rightarrow T_1 \rightarrow T_2$