School of Information Technology IIT Kharagpur

Course Id: IT60113 Advanced Database Systems

End Sem Exam

Date: November 22, 2007 Max. Marks: 100 **Total Time: 3 Hours**

Answer Q. No. 1 and any 3 from the rest. You may answer the questions in any order. However, all parts of the same question must be answered together. <u>Clearly state any assumption that you make.</u>

- 1. Consider the following set of 2-D points: (0,0), (10,10), (20,20), (5,5), (40,40), (25,25).
 - (a) Construct an R-Tree with M=2 and m=1 for the above set of data points, assuming that the data arrived in the above sequence. Show each intermediate step. You do not need to write the algorithms, but must clearly (in one sentence each) explain the steps as you proceed.
 - (b) Once the R-Tree has been constructed, the entry (20,20) is to be deleted. Show the new R-Tree, explaining the individual steps (one sentence each).
 - (c) Consider a query point (16,16). Find its nearest neighbor using the R-Tree constructed in (a) above following MINDIST ordering in the Active Branch List. Apply suitable downward and upward pruning using MINDIST and MINMAXDIST based heuristics. Explain each step in brief as you proceed. [12+4+12=28]
- 2.
- (a) Consider the following two relations r and s available in two sites S1 and S2, respectively. We need to compute the natural join of r and s, with the results made available in site S1. Explain how semi-join processing can be applied, clearly showing the intermediate and the final results.

A	B	С	С	D	E
1	2	3	3	4	5
4	5	6	3	6	8
1	2	4	2	3	2
5	3	2	1	4	1
8	9	1	1	2	3
r			S		

- (b) What are the failures that can occur during two-phase commit for a distributed transaction?
- (c) For each failure, explain how two-phase commit can ensure atomicity.
- (d) Which of these failures is handled in a better way by three-phase commit and how?

[8+6+6+4=24]

3. Consider the following two transactions:

T1:

T2:

Read(A)	Read(A)
Read(B)	Read(B)
A←A+B	A←A-B
Write(A)	Write(A)

For the above two transactions, generate (non-trivial i.e., not serial unless that is the only way):

(a) Conflict serializable schedule using 2PL

- (b) View serializable schedule using Timestamp based protocol with Thomas' Write rule
- (c) Conflict serializable schedule using an optimistic concurrency control scheme

Your schedules should clearly identify the corresponding serial schedules and the conditions that determine the order of the equivalent serial schedule. $[8\times3=24]$

4.

- (a) Which of the following rules and predicates are safe if b1 and b2 are database predicates? [8]
 r1: p(X,X) ← b2(Y,Y,a), b1(X), X>Y
 r2: q(X,Y) ← p(X,Z), p(Z,Y)
 r3: s(X) ← b2(Y,Y,a), X>Y, ¬b1(X)
- (b) Consider a database having two tables: EmpSale (EmpName, Sale) and EmpMgr (EmpName, MgrName). Write suitable Starburst rule(s) which insert/update the sales of an employee's immediate manager (as obtained from EmpMgr table) whenever EmpSale table is updated or rows are inserted in it. You need not handle delete. No penalty for minor syntax errors. [8]
- (c) Consider a temporal database table be defined as follows.
 CREATE TABLE Employee (Name CHAR(30), Salary INTEGER, Designation CHAR(30))
 AS VALID STATE DAY AND TRANSACTION

Assume that the table is initially empty and today (the date of your exam, i.e., 22/11/2007) you are inserting tuples in the table.

- i. Write INSERT statement(s) to insert information about Amit whose salary is 2000 and his designation is Lecturer.
- ii. On 25/11/2007, you come to know that his salary has become 3000 with effect from 23/11/2007. What Temporal SQL statement would you write to put this information in the database? What will be the content of the table after the insert? [3+5=8]
- 5. Briefly explain the following concepts, giving examples wherever applicable:
 - (a) Multi-version scheme for concurrency control
 - (b) Repeatable read isolation level
 - (c) Concurrency control schemes in distributed databases
 - (d) SS-Tree

[6×4=24]