UNIVERSITY OF TWENTE.

Exam Advanced Database Systems (192110902)

Wednesday 4 November 2015 (8:45 – 11:45), Sportcentrum

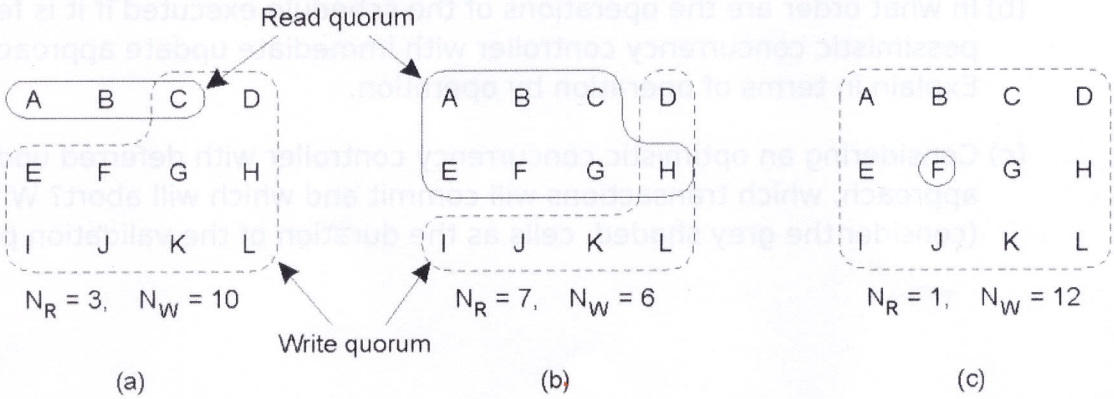
Mena Habib and Djoerd Hiemstra

- The exam consists of 6 questions on 4 pages.
- Motivate yours answers. The motivation / argumentation plays an important role in grading the assignments.
- You may not consult books or notes, but only one page of A4 size, double-sided. The page may contain text (typed or hand-written) and (possibly reduced) images (copied from the book, other sources or hand-made).
- For each assignment, the number of points is given. They add up to 90. You get 10 points for showing up at the exam. The grade for the exam is determined by dividing the number of points by 10.
- The final grade for the course is determined by 0.7 × 'Grade written exam' + 0.3 × 'Grade project'. Both grades need to be ≥ 5.5 and the MiniLab assignment has to be completed satisfactorily.

Question(1) Miscellaneous (15 points)

(a) Describe the differences between the savepoints and chained transactions ? What kind of problems does each approach solve?

(b) Consider the following three possible selections of read and write quorum for a database replication system. Which selections are good for data availability and which are not? Explain.



(c) Write an algorithm (in JAVA or pseudocode) for the locking protocol used in the pessimistic immediate update concurrency controller. Given a lock request, the algorithm should decide whether the request should be allowed or denied.

Question(2) Serializability (15 points)

Consider the three transactions shown below. Transaction T_1 is transferring 100 € from one account with balance bal_x to another account with balance bal_y . T_2 is increasing the balance of these two accounts by 10%, while T_3 is withdrawing 50 € from one account with balance bal_z .

T_1	T_2	T_3
begin_transaction		
read(bal_x)		
bal_x=bal_x - 100		
write(bal_x)		
	begin_transaction	
	read(bal_x)	
	bal_x=bal_x * 1.1	
		begin_transaction
		read(bal_z)
		bal_z=bal_z - 50
		write(bal_z)
		commit_transaction
	write(bal_x)	
	read(bal_y)	
	baly=bal_y * 1.1	
	write(bal_y)	
	commit_transaction	
read(bal_y)		
bal_y=bal_y + 100		
write(bal_y)		
commit_transaction		

(a) Draw the serialization graph of this schedule. Is it serializable or not? Why?

(b) In what order are the operations of the schedule executed if it is fed to a pessimistic concurrency controller with immediate update approach? Explain in terms of operation by operation.

(c) Considering an optimistic concurrency controller with deferred update approach, which transactions will commit and which will abort? Why? (consider the grey shaded cells as the duration of the validation period).

Question(3) Recovery (15 points)

Suppose a crash took place on a server containing a database that uses no-force commit policy. After restart the recovery procedure found the following situation:

Log:

	8	9	10	11	12	13	14	15	16	17	18	19
...	U T1 X 0, 5	CK T1, T3	B T2	U T3 Y 0, 2	C T1	B T4	U T2 X 5, 6	U T3 Z 0, 4	C T3	U T4 Z 4, 8	CL T4 Z 8, 4	A T4

Where each record contains LSN, Transaction type (U=Update, B=Begin Transaction, CK= Sharp Checkpoints, C=Commit, A=Abort, CL= Compensating Log Record), Transaction ID, Item ID, Before and After Images.

DB Pages:

Page 22 LSN:17 X=6 Y=2	Page 25 LSN:17 Z=8

(a) The recovery protocol reconstructs a consistent database state. Describe in details the three phases of the recovery protocol and show the database pages after each phase.

(b) The no force commit policy may cause two problems. What are those two problems. Which of the two problems take place in the aforementioned case?

(c) What is the role of the Compensating Log Record. What would happen if it was not inserted in the log?

Question (4) Indexing (12 points)

Assume a table has 100,000 rows and each row occupies 100 bytes. Disk pages are 4000 bytes. Estimate the number of page transfers needed for an equality search on a search key that has a weight of 50 in the following 3 cases.

(a) An (unsorted) heap file with no index.

(b) An unclustered B+ tree in which each entry occupies 20 bytes.

(c) A clustered B+ tree.

Question (5) Theory (10 points)

Please, explain:

(a) Explain the difference between an equality search and a range search.

(b) Explain why a table can have only one clustered index.

(c) Explain why a secondary, unclustered index must be dense.

(d) Is the projection operation commutative? Explain your answer.

(e) What are the advantages of stored procedures?

Question (6) Query Processing (23 points)

Consider the following relational schema:

```
Professor(ProfId, ProfName, Dept)
Course(CrsCode, CrsName)
Teaching(ProfId, CrsCode, Semester)
```

Assume that keys are underlined and:

- 1000 tuples with 10 tuples per page in Professor;
- 20,000 tuples with 10 tuples per page in Teaching;
- 2000 tuples, 5 tuples per page, in Course;
- A 5-page buffer;
- 50 different values for Dept;
- 200 different values for Semester;
- A clustered 2-level B+ tree index on Professor(Dept);
- An unclustered Hash index on Professor(ProfId);
- Course is a sorted file (no primary index)
- An unclustered Hash index on Course(CrsName);
- A clustered 2-level B+ tree index on Teaching(ProfId);
- An unclustered 2-level B+ tree index on Teaching(Semester).

Answer the following questions:

(a) Write an SQL query that selects the CrsName and ProfName of professors from the 'CS' department (Dept) and their courses that are taught in semester 'F2015'.

(b) Give the "naive" translation of your SQL query into relational algebra (as given by the general translation of SQL to relational algebra.)

(c) For plan of Question (b), choose the most efficient query execution plan. Measure its efficiency in page transfers. Explain your answer.

(d) Draw the fully pushed query tree. Also optimize the projection(s).

(e) For the plan of Question (d), choose the most efficient query execution plan. Measure its efficiency in page transfers. Explain your answer.