CS411 Database Systems

Fall 2005, Prof. Chang

Department of Computer Science
University of Illinois at Urbana-Champaign

Final Examination

December 14, 2005
Time Limit: 180 minutes

Problem 1 (*12 points*) Misc. Concepts

(1) False

(2) False

(3) True

(4) False

(5) True

(6) False

(7) False

(8) True

(9) False

(10) True

(11) False

(12) True

Problem 2 (*18 points*) Short Answer Questions

(1) $\pi_a(\sigma_{b<10}R) - \pi_a(\sigma_{b>5}S)$

(2) SELECT b, c FROM R, S WHERE $R.a = S.a$ AND $R.b$>10 AND $S.c$>5

(3) 55

(4) ABCDE

(5) Refer to Figure 1

(6) Atomic (or Atomicity)

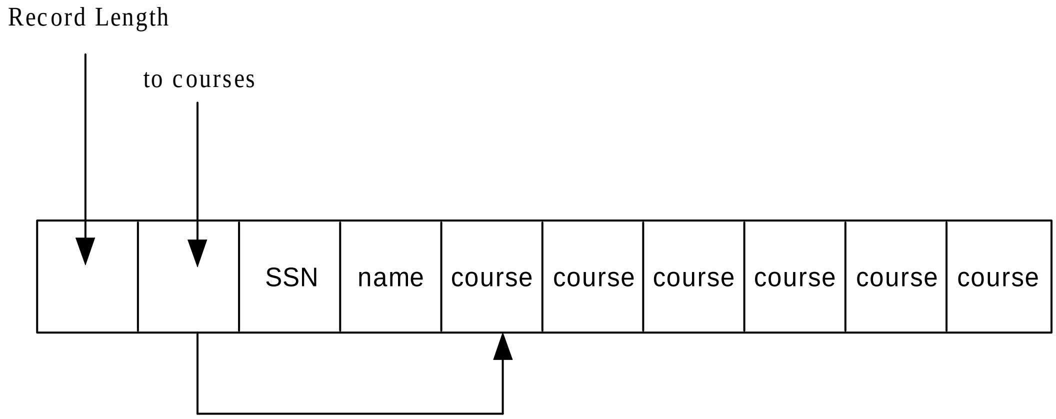

Figure 1: A Professor Record

(7) Optimal plan must come from optimal subplans.

(8) 200

(9) Atomic and Durable (AD)

Problem 3 (*8 points*) Schema Decomposition

(a) CDE, ACD, BCD

(b) Yes, it is in 3NF because B, E and A are all parts of keys.

(C) No. For example, $BC \to E$ violates BCNF. If we decompose using this FD, we get BCE, ABCD. Further decomposing ABCD using $A \to B$ we get AB, ACD. Therefore, it could be decomposed into BCE, AB, ACD.

Problem 4 (*9 points*) Query Languages

(a) $\pi_{name}(\sigma_{age>60}(Customer))$

(b) $\pi_{name}(Customer) - \pi_{name}(Order \bowtie Customer)$

(c) SELECT Product.pid, SUM(quantity) FROM Product LEFT OUTER JOIN Order GROUP BY Product.pid ORDER BY SUM(quantity) DESC

Problem 5 (*10 points*) Indexing: B+ tree

(a) Refer to Figure 2.

(b) $30 < 40 \to$ 1st pointer, goes to 1st leaf, get record with key 20, follow the chain to the 3rd leaf, getting records with keys 30, 40, 50, 60.

(c) Either Figure 3 or 4 is ok.

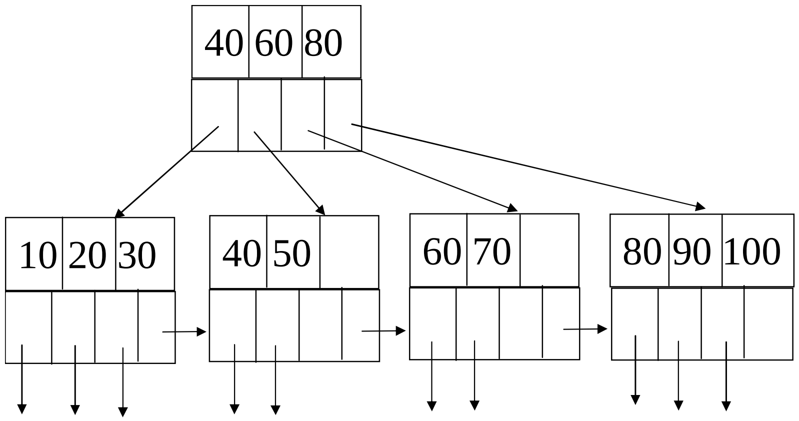

Figure 2: B+ tree after insertion

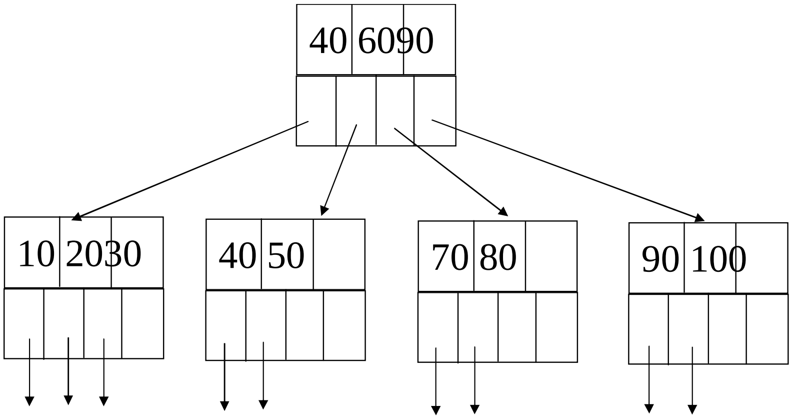

Figure 3: B+ tree after deletion

Problem 6 (*10 points*) Query Processing

(a) $150 + \lceil 150/(14-1) \rceil * 100 = 1250$ blocks

or

$100 + \lceil 100/(15-1) \rceil * 150 = 1300$ blocks

(b) 5(B(R)+B(S)) = 1250 blocks

(c) No. Here we only know the number of blocks for R and S. We need the number of tuples for R and S to estimate the size of the output relation.

Problem 7 (*19 points*) Query Optimization

(a) Joining in different orders will result in different sizes of intermediate relations, thus different processing costs. As an example, suppose the sizes of Student, Enrollment, and Courses are 100, 1000, 1000, respectively. If we join Student with Courses first, which is a cartesian

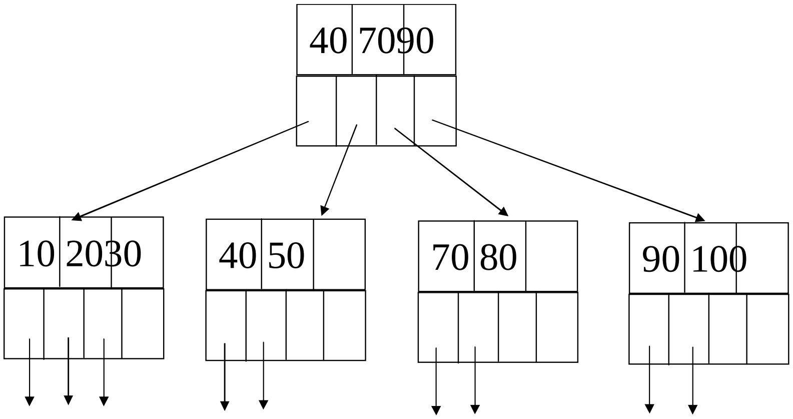

Figure 4: B+ tree after deletion (Alternative)

product, the size of the intermediate results is 100000. We then join it with Enrollment. However, we can also join Student with Enrollment first, with the condition S.sid=E.sid, resulting in 1000 intermediate tuples, which is much smaller than 100000. We then join the intermediate results with courses.

(b) Iteration 1, queries with 1 table: Student, Enrollment, Courses, Instructor. Totally 4.

Iteration 2, queries with 2 tables: SE, SC, SI, EC, EI, CI. Totally 6.

Iteration 3, queries with 3 tables: SEC, SEI, SCI, ECI. Totally 4.

Iteration 4, queries with 4 tables: SECI. Totally 1.

(c) Iteration 1, queries with 1 table: Student, Enrollment, Courses, Instructor. Totally 4.

Iteration 2, queries with 2 tables: SE, EC, CI. Totally 3.

Iteration 3, queries with 3 tables: SEC, ECI. Totally 2.

Iteration 4, queries with 4 tables: SECI. Totally 1.

Problem 8 (*14 points*) Failure Recovery

Log ID	Log
1	$\langle \text{START } T1 \rangle$
2	$\langle T1, A, 10 \rangle$
3	$\langle \text{START } T2 \rangle$
4	$\langle T1, B, 10 \rangle$
5	$\langle \text{COMMIT } T1 \rangle$
—————-	// Output(A); Output(B);
6	$\langle T2, B, 10 \rangle$
7	$\langle \text{COMMIT } T2 \rangle$
—————-	// Output(B);
	// Output(A); Output(B);(move up)
8	$\langle \text{START } T3 \rangle$
9	$\langle T3, A, 10 \rangle$
10	$\langle \text{START } T4 \rangle$
11	$\langle T3, B, 20 \rangle$
12	$\langle \text{COMMIT } T3 \rangle$
—————-	// Output(A); Output(B);
13	$\langle T4, C, 10 \rangle$
—————-	// <START CKPT(T4)>;
14	$\langle \text{START } T5 \rangle$
	// Output(A); Output(B);(move up)
—————-	// <END CKPT>;
15	$\langle \text{COMMIT } T4 \rangle$
	// Output(C)
16	$\langle T5, D, 10 \rangle$
17	$\langle \text{COMMIT } T5 \rangle$
	// Output(D)

(a) No. It can't be undo log. Undo log requires data values be written to disk before the corresponding transactions commit log record is written to disk. There are many counterexamples here.

(b) Output values immediately after the commit log record is written to disk.

(c) Please refer to the log shown above. The <END CKPT> could be put anywhere after log 15 as well. But it is better to put it as early as possible.

(d) We need to redo T4, starting from log 13.