Database Management Systems
Written Examination

15.02.2008

First name		Last name	
Student number		Signature	

Instructions for Students

- Write your name, student number, and signature on the exam sheet.
- Write your name and student number on every solution sheet you hand in.
- This is a closed book exam: the only resources allowed are blank paper, pens, and your head. Use a pen, not a pencil.
- Write neatly and clearly. The clarity of your explanations affects your grade.
- You have 120 minutes for the exam.

Reserved for the Teacher

Exercise	Max. points	Points
1	20	
2	8	
3	12	
4	8	
5	20	
6	8	
7	12	
8	12	
Total	100	

Exercise 1 Answer the following questions:

a. What is the buffer manager and what is its goal?

b. What is the main problem with the sequential file organisation?

c. What is a primary index?

d. Which tree has (typically) less nodes: B^+-tree or B-tree?

e. What is the goal of query optimization?

f. Consider a relation $r(A, B)$, which is sorted on A, and the query $\sigma_{B=v}(r)$. Is binary search a valid strategy to evaluate this query?

g. What is a materialized view?

h. Mention two advantages of running multiple transactions concurrently?

i. What are the two phases of the two-phase locking protocol?

j. The wait-for graph is used to detect conflict serializability or deadlocks?

Exercise 2 Given is the following table with project assignments:

Name	Project	Hours
Jan	P7	800
Ann	P2	250
Jan	P1	400
Jan	P3	500

To store this information in a file, we assume the use of variable-length records.

a. Show the file organization using the reserved space method.

b. Show the file organization using the slotted page structure.

Exercise 3 Consider an extendable hash table with a hash function $h(x) = x \bmod 8$ and a bucket size of 2. The following key values are inserted in that order:

$$44, 36, 30, 56, 34, 62, 50, 22$$

a. What is the maximal size of the bucket address table?

b. Show the hash index (both the address bucket table and the buckets) after the first four keys are inserted.

c. Show the hash index (both the address bucket table and the buckets) after all the keys are inserted.

Exercise 4 Consider the following relation `accounts`:

	AccountNumber	BranchName	Balance
0	A-217	Brighton	750
1	A-101	Downtown	500
2	A-110	Downtown	600
3	A-215	Mianus	700
4	A-102	Perryridge	400
5	A-201	Perryridge	900
6	A-218	Perryridge	700
7	A-222	Reedwood	700
8	A-305	Round Hill	350

a. Create a bitmap index over the attributes *BranchName* and *Balance*, where *Balance* is divided into four ranges: [0–250), [250–500), [500–750), [750–1000]

b. Consider a query that requests all accounts with a balance of 500 or more that are not in Downtown. Outline the steps in answering the query, and show the final and intermediate bitmaps constructed.

Exercise 5 Assume two relations $r(A)$ and $s(A)$ with r being stored in a sequential (ordered) file and s being stored in an unordered file on the disk. The block size is 2,000 Bytes, the tuple size 10 Bytes, and the cardinality is 800,000 tuples for both relations (assume identical instances). The values of the integer attribute A are uniformly distributed between 5 Mio. and 9 Mio. and they are unique in both relations. The disk performance is given as follows: latency time = 0.008 sec, seek time = 0.016 sec, transfer time = 0.001 sec.

a. Determine the number of block IOs and the execution time for the following queries on the two relations:
Q1: $\sigma_{A=6{,}000{,}000}(x)$
Q2: $\sigma_{A<5{,}009{,}500}(x)$

b. Consider the creation of a B$^+$-tree index for r and s, where each node contains 100 index entries and fills an entire block. Determine the number of blocks at each level of the two trees.

c. Determine the number of block IOs and the execution time for Q1 and Q2 when the B$^+$-tree index is used.

Exercise 6 Consider two relations $r(A, B, C)$ and $s(C, D, E)$ and the following two queries:

a. Q1: $\pi_{A,E}(\sigma_{A=10}(r \bowtie s))$

b. Q2: $\pi_{A,B,C}(\sigma_{A=10 \wedge (B=100 \vee E=100)}(r \bowtie s))$

Write equivalent but more efficient RA expressions for Q1 and Q2. Explain the optimization step(s) and the evaluation of the new expressions.

Exercise 7 Let relations $r_1(A, B)$ and $r_2(A, C)$ have the following properties: r_1 has $10,000$ tuples and 5 tuples of r_1 fit into one block; r_2 has 125 tuples and 10 tuples of r_2 fit into one block. We have a hash index on attribute A in relation r_1.

Compute the costs of the following evaluation plans for $r_1 \bowtie r_2$:

- Plan p1: Nested loop with r_1 as outer loop
- Plan p2: Nested loop with r_2 as outer loop and hashed lookup in r_1
- Plan p3: Hash join

Exercise 8 Given is the following schedule that involves transactions T_1 and T_2:

T_1	T_2
read(A)	
write(A)	
	read(A)
	read(B)
read(B)	
write(B)	

Answer the following questions and explain your answers:

a. Is the schedule conflict serializable?

b. Is the schedule view serializable?

c. Is the schedule recoverable if both transactions commit immediately after the last operation?

d. Is the schedule cascadeless?

Solution 1

a. Buffer manager is the subsystem which is responsible for buffering disk blocks in main memory. Tries to minimize the number of disk accesses.

b. It is difficult to maintain the physical order as records are inserted and deleted.

c. Index whose search key specifies the sequential order of the file.

d. B-tree.

e. Find the most efficient query evaluation plan (query plan) for a given query, i.e., the one which can be executed most efficiently.

f. No.

g. A materialized view is a view whose contents is computed and stored/cached on disk.

h. Increased processor and disk utilization; reduced average response time.

i. Growing phase and shrinking phase.

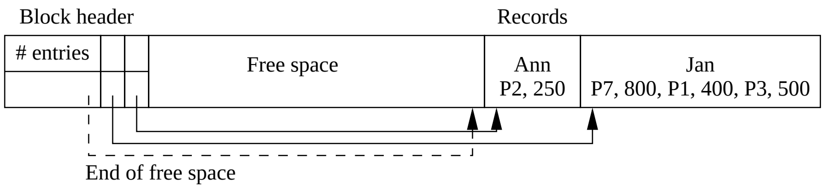
j. Deadlocks.

Solution 2

a. Reserved space method for variable-length records

0	Jan	P7	800	P1	400	P3	500
1	Ann	P2	250	$\perp$	$\perp$	$\perp$	$\perp$

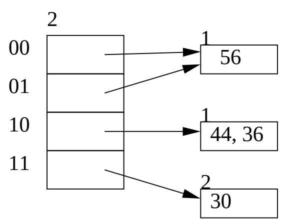
b. Slotted space structure for variable-length records



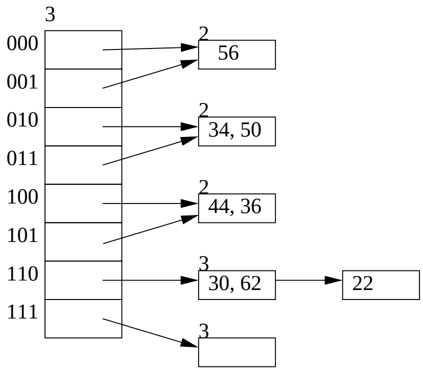
Solution 3

a. Max. size of bucket address table is 3

b. Hash table after inserting the first four keys:
$h(44) = 44 \bmod 8 = 100$
$h(36) = 36 \bmod 8 = 100$
$h(30) = 30 \bmod 8 = 110$
$h(56) = 56 \bmod 8 = 000$



c. Hash table after inserting all keys:
$h(34) = 34 \text{ mod } 8 = 010$
$h(62) = 62 \text{ mod } 8 = 110$
$h(50) = 50 \text{ mod } 8 = 010$
$h(22) = 22 \text{ mod } 8 = 110$



Solution 4

a. Bitmap index for *BranchName*:

Brighthon:	`[1 0 0 0 0 0 0 0 0]`
Downtown:	`[0 1 1 0 0 0 0 0 0]`
Mianus:	`[0 0 0 1 0 0 0 0 0]`
Perryridge:	`[0 0 0 0 1 1 1 0 0]`
Reedwood:	`[0 0 0 0 0 0 0 1 0]`
Round Hill:	`[0 0 0 0 1 0 0 0 1]`

Bitmap index for *Balance*:

$[0 - 250)$:	`[0 0 0 0 0 0 0 0 0]`
$[250 - 500)$:	`[0 0 0 0 1 0 0 0 1]`
$[500 - 750)$:	`[0 1 1 1 0 0 1 1 0]`
$[750 - 1000)$:	`[1 0 0 0 0 1 0 0 0]`

b. Take the bitmap vector of *Balance* in [500-750) and the bitmap vector of *Balance* in [500-1000] and compute the logical OR:
`[0 1 1 1 0 0 1 1 0]` OR `[1 0 0 0 0 1 0 0 0]` = `[1 1 1 1 0 1 1 1 0]`

Take the bitmap vector of *Downtown* and compute the logical NOT:
NOT `[0 1 1 0 0 0 0 0 0]` = `[1 0 0 1 1 1 1 1 1]`

Compute the AND of the two intermediate results:
`[1 1 1 1 0 1 1 1 0]` AND `[1 0 0 1 1 1 1 1 1]` = `[1 0 0 1 0 1 1 1 0]`

Retrieve the records 0, 3, 5, 6, and 7.

Solution 5

a. $2,000/10 = 200$ tuples/block
$800,000/200 = 4,000$ blocks
$1 \text{ IO} = 0.008 + 0.016s + 0.001s = 0.025s$

Q1: $\sigma_{A=6,000,000}(r)$: Binary serach
– Block IOs: $\lceil \log_2 4,000 \rceil = 12$
– Time: $0.025 \times 12 = 0.3$ sec

Q1: $\sigma_{A=6,000,000}(s)$: Sequential search
– Block IOs: on average read 2000 blocks (half of all) to find a unique value
– Time: $0.025 \times 2000 = 50$ sec

Q2: $\sigma_{A<5,009,500}(r)$: Sequential search
– Block IOs:
  avg. distance between values: $4Mio/800,000 = 5$
  # of qualifying tuples: $9,500/5 = 1,900$
  # of qualifying blocks: $\lceil 1,900/200 \rceil = 10$ block IOs
– Time: $10 \times 0.025 = 0.25$ sec

Q2: $\sigma_{A<5,009,500}(s)$: Sequential search
– Block IOs: 4000 blocks
– Time: $0.025 \times 4,000 = 100$ sec

b. Nodes (=index blocks): 100 index entries per node
Index blocks required at each level:
- level 3: $\lceil 800,000/100 \rceil = 8,000$ blocks (leaf nodes)
- level 2: $\lceil 8,000/100 \rceil = 80$ blocks
- level 1: $\lceil 80/100 \rceil = 1$ block
$\Rightarrow 8,081$ index blocks are needed in total
The result is the same for r and s

c. Q1: $\sigma_{A=6,000,000}(r)$: B$^+$-tree search
– Block IOs: 3 index blocks + 1 data block = 4 blocks
– Time: $0.025 \times 4 = 0.1$ sec

Q1: $\sigma_{A=6,000,000}(s)$: the same as for r

Q2: $\sigma_{A<5,009,500}(r)$: Index makes no sense; hence the same result as in (a).

Q2: $\sigma_{A<5,009,500}(s)$: Follow leave nodes in B$^+$-tree
– Block IOs: Follow leave nodes from the beginning to locate the data blocks
  # of qualifying tuples (= # of qualifying key values): $1,900$
  # of qualifying blocks: max. $1,900$ (each qualifying tuple in different block)
  # of qualifying B$^+$-tree nodes: $\lceil 1,900/100 \rceil = 19$ (100 index entries/node)
  Total: $1,900 + 19 = 1,919$ block IOs
– Time: $(1,900 + 19) \times 0.025 = 47.975 sec$

Solution 6

a. Q1: $\pi_{A,E}(\sigma_{A=10}(r \bowtie s))$
– Push down the selection to relation r: $\pi_{A,E}(\sigma_{A=10}(r) \bowtie s)$
– Drop B and D from r and s, respectively: $\pi_{A,E}(\pi_{A,C}(\sigma_{A=10}(r)) \bowtie \pi_{C,E}(s))$

b. Q2: $\pi_{A,B,C}(\sigma_{A=10 \wedge (B=100 \vee E=100)}(r \bowtie s))$
– Push down the selection on A to r: $\pi_{A,B,C}(\sigma_{B=100 \vee E=100}(\sigma_{A=10}(r) \bowtie s))$
– Project s to C and E: $\pi_{A,B,C}(\sigma_{E=100}(\sigma_{A=10}(r) \bowtie \pi C, E(s)))$

Solution 7
Blocks required for r_1: $\lceil 10,000/5 \rceil = 2,000$ blocks
Blocks required for r_2: $\lceil 125/10 \rceil = 13$ blocks

Plan p1:
$cost(p1) = n_{r_1} * b_{r_2} + b_{r_1} = 10,000 * 13 + 2,000 = 132,000$ block transfers

Plan p2:
$cost(p2) = b_{r_2} + n_{r_2} * c = 13 + 125 * 16 = 2,013$ block transfers
(c is the cost for the hashed lookup and retrieval of machting tuples: cost for lookup is 1; each r_2-tuple has on avg. $10,000/125 = 80$ matching r_1-tuples, which corresponds to $80/5 = 16$ blocks)

Plan p3: (the overhead due to partially filled blocks is ignored)
$cost(p3) = 3 * (b_{r_1} + b_{r_2}) = 3 * (2,000 + 13) = 6,039$

Solution 8

a. No. For both possible serial schedules, $\langle T_1, T_2 \rangle$ and $\langle T_2, T_1 \rangle$, we get either a conflict with $write(A) - read(A)$ or with $write(B) - read(B)$.

b. No.
In the serial schedule $\langle T_1, T_2 \rangle$, the following rule is violated for data item B: For each data item Q, if transaction T_i reads the inital value of Q in schedule S, then T_i must in scheule S' also read the initial value of Q.
In the serial schedule $\langle T_2, T_1 \rangle$, the following rule is violated for data item A: For each data item Q, if transaction T_i reads data item Q in schedule S and the value was produced by T_j, then T_i must in scheule S' also read the value of Q that was produced by T_j.

c. No. T_1 might fail after T_2 already committed.

d. No. If T_1 fails after T_2 executed the last operation (not yet committed), it causes T_2 to roll back.