Database Management Systems
Written Exam

15.02.2012

First name		Last name	
Student number		Signature	

Instructions for Students

- Write your name, student number, and signature on the exam sheet and 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 2 hours for the exam.

Good luck!

Reserved for the Teacher

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

Exercise 1 (20 pt) Answer the following questions:

a. What is minimized by disk-arm-scheduling algorithms (e.g., by the elevator algorithm)?

b. What is the difference between a dense and a sparse index?

c. Can bucket overflow be avoided in hashing?

d. Assume $M \geq 3$ main memory blocks. How many blocks shall be used for the outer relation in a (block) nested loop join?

e. Which strategy for the evaluation of complex expression is more efficient: materialization or pipelining?

f. Is $E_1 \bowtie_\theta E_2 = E_2 \bowtie_\theta E_1$ a correct equivalence rule (for query optimization)?

g. Describe the A property of the ACID properties?

h. Is the following statement correct: Every view serializable schedule is also conflict serializable?

i. What is stored in the lock table?

j. For log-based recovery with deferred DB modifications: What actions are performed if a transaction is rolled back?

Exercise 2 (8 pt) Consider the following file of variable-length records that stores the `exams` relation of exercise 4 by using the byte string representation.

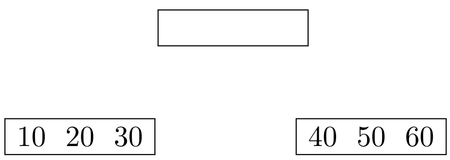
r_0	Jan	AI	C	3	DB	A	2	DMS	B	2	⊥
r_1	Ann	AI	A	3	CSA	D	1	⊥			
r_2	Bob	OS	C	1	DB	C	2	⊥			
r_3	Sue	DMS	A	2	ITP	A	1	⊥			
r_4	Joe	DMS	B	2	⊥						

a. Show the file structure with fixed-length representation with reserved space.

b. Show the file structure with fixed-length representation with pointers and anchor and overflow block.

Exercise 3 (12 pt) Consider a B$^+$-tree, where at most 4 pointers fit into a single node.

a. The following is an incomplete B$^+$-tree for the keys $\{10, 20, 30, 40, 50, 60\}$. Complete the tree.



b. Insert 25 and 18 (in that order). Show the resulting tree.

c. Delete 30. Show the resulting tree.

Exercise 4 (8 pt) Consider the following relation `exams`:

	Name	Course	Grade	Year
0	Jan	AI	C	3
1	Jan	DB	A	2
2	Jan	DMS	B	2
3	Ann	AI	A	3
4	Ann	CSA	D	1
5	Bob	OS	C	1
6	Bob	DB	C	2
7	Sue	DMS	A	2
8	Sue	ITP	A	1
9	Joe	DMS	B	2

a. Create a bitmap index over the two attributes which are most suitable for such an index. Explain your choice.

b. Describe the evaluation of $\sigma_{Course=DMS \wedge Grade=A \wedge Year=2}(\texttt{exams})$ using the index.

Exercise 5 (20 pt) Consider a relation $r(A, B)$ with a primary B^+ tree index on attribute A and the following characteristics: $|r| = 4,000,000$; 100 index entries per block; 20 data tuples per block; the values of both attributes A and B are uniformly distributed in the range $[1 - 10,000,000]$; A forms a key; seek time $= 0.016$ sec; latency $= 0.008$ sec; transfer time $= 0.001$ sec.

a. Determine the number of blocks (= number of nodes) used for the B^+-tree if index blocks are filled up to 80%.

b. Describe the evaluation and determine the number of block IOs (distinguishing between index blocks and data blocks) for the following queries:

- Q1: $\sigma_{A>2Mio \wedge A<4Mio}(r)$
- Q2: $\sigma_{A<2Mio \vee A>4Mio}(r)$
- Q3: $\sigma_{A=5Mio \wedge B>5Mio}(r)$
- Q4: $\sigma_{A>5Mio \vee B>5Mio}(r)$

c. Determine the execution time for the queries in b.)

Exercise 6 (12 pt) Let relations $r(A, B)$ and $s(A, C)$ have the following properties: r has 10,000 tuples and 5 tuples of r fit into one block; s has 125 tuples and 10 tuples of s fit into one block. In relation r we have a hash index on attribute A, where 100 (search-key,pointer)-pairs fit into one bucket. Furthermore, the A-values in r are equally distributed over the A-values in s, and 20 blocks fit into main memory. Compute the cost of the following evaluation plans for $r \bowtie s$:

a. Plan p1: Nested loop with r as outer loop

b. Plan p2: Hash join

c. Plan p3: Merge join (assume r and s are already sorted)

Exercise 7 (8 pt) Assume two relations $r(A, C)$ and $s(B, D)$ and the following relational algebra expression:

$$\sigma_{A<10 \wedge B>100 \wedge A+B<200}(r \times s)$$

a. Transform this selection statement into a more efficient expression by applying some equivalence rules. Explain your choice and why the new expression is more efficient.

b. Assume that you can create one single-attribute index on either relation r or s to improve the evaluation of the expression obtained in a.). Which index (type, relation, attribute) would you create? Motivate your choice and briefly describe the evaluation strategy with this index.

Exercise 8 (12 pt) Given is the following schedule that involves transactions T_1 and T_2:

	T_1	T_2
1	read(A)	
2	write(A)	
3		read(A)
4		read(B)
5	read(B)	
6	write(B)	

Answer the following questions and explain your answers:

a. Draw the conflict graph and show whether the schedule is conflict serializable or not?

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

c. Is the schedule possible under the timestamp protocol?

Solution 1

a. Disk-arm movement

b. A dense indes has an index record for every search-key value. A sparse index has an index record only for some search-key values.

c. No

d. $M - 2$

e. Pipelining

f. Yes

g. Atomicity: A transaction's changes to the state of a DB are atommic, i.e., either all operations of a transaction are properly reflected in the DB or none are.

h. No

i. Granted locks and pending requests for locks

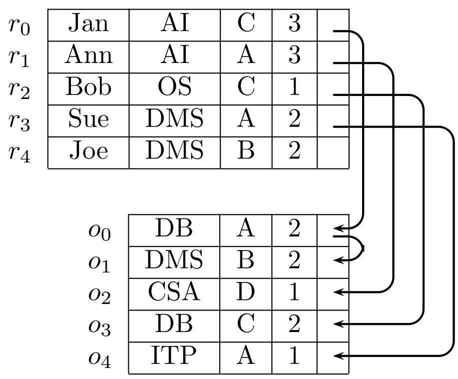
j. No actions need to be done

Solution 2

a. Fixed-length representation with reserved space

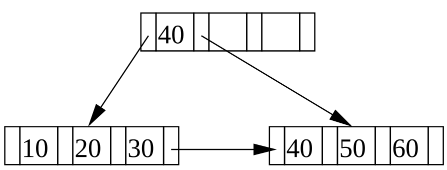
0	Jan	AI	C	3	DB	A	2	DMS	B	2
1	Ann	AI	A	3	CSA	D	1	⊥	⊥	⊥
2	Bob	OS	C	1	DB	C	2	⊥	⊥	⊥
3	Sue	DMS	A	2	ITP	A	1	⊥	⊥	⊥
4	Joe	DMS	B	2	⊥	⊥	⊥	⊥	⊥	⊥

b. Fixed-length representation with pointers ($\uparrow x$ denotes a pointer to record x)

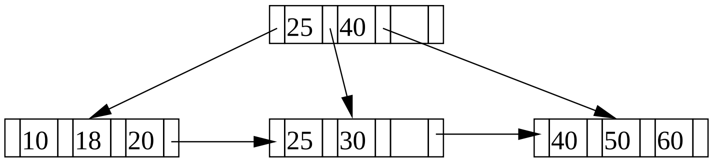


Solution 3

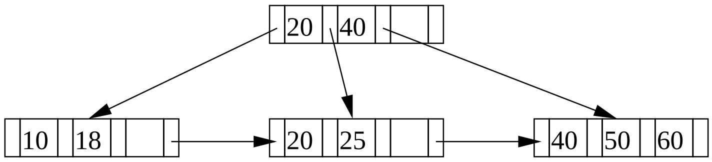
a.



b.



c.



Solution 4

a. *Grade* and *Year* are the most suitable attributes for a bitmap index, since they have the smallest number of different values keeping the index as small as possible.

Grade A: `[0 1 0 1 0 0 0 1 1 0]`
Grade B: `[0 0 1 0 0 0 0 0 0 1]`
Grade C: `[1 0 0 0 0 1 1 0 0 0]`
Grade D: `[0 0 0 0 1 0 0 0 0 0]`

Year 1: `[0 0 0 0 1 1 0 0 1 0]`
Year 2: `[0 1 1 0 0 0 1 1 0 1]`
Year 3: `[1 0 0 1 0 0 0 0 0 0]`

b. Take the bitmap vector of Grade A and Year 2 and compute the logical AND:

```
[0 1 0 1 0 0 0 1 1 0] AND
[0 1 1 0 0 0 1 1 0 1]
---------------------
[0 1 0 0 0 0 0 1 0 0]
```

Retrieve the tuple(s) with a '1' in the resulting bitmap vector, i.e., tuple no. 1 and tuple no. 7, and then select those tuples that have Course=DMS, i.e., $(Sue, DMS, A, 2)$.

Solution 5

a. Index blocks:
$0.8 * 100 = 80$ index entries per block
Index blocks required at each level:
- level 4: $\lceil 4,000,000/80 \rceil = 50,000$ blocks (leaf nodes)
- level 3: $\lceil 50,000/80 \rceil = 625$ blocks
- level 2: $\lceil 625/80 \rceil = 8$ blocks
- level 1: $\lceil 8/80 \rceil = 1$ block
$\Rightarrow 50,634$ index blocks are needed in total

b. Total number of data blocks: $\lceil 4,000,000/20 \rceil = 200,000$

$Q1$: Use index to locate the first tuple with $A > 2$ Mio, then scan the data relation sequentially.
4 index blocks (one at each level)

$40,000$ data blocks (= 20% of the data blocks)
$\Rightarrow 4 + 40,000 = 40,004$ blocks in total

$Q2$: Scan data relation from the beginning until $A \geq 2Mio$; then use index to locate the first tuple with $A > 4$ Mio and scan the data relation sequentially from that point.
Condition $A < 2$ Mio: $40,000$ data blocks
Condition $A > 4$ Mio: 4 index blocks + $120,000$ data blocks = $120,004$ blocks
$\Rightarrow 160,004$ blocks in total

$Q3$: Use the index to find the tuple (if there is one) with $A = 5$ Mio and check this tuple for the second condition, $B > 5$ Mio.
4 index blocks (one at each level); 1 data block
$\Rightarrow 4 + 1 = 5$ blocks in total

$Q4$: Scan all data blocks; index is not useful.
$\Rightarrow 200,000$ data blocks in total

c. Execution time: 1 IO $= 0.016 + 0.008s + 0.001s = 0.025s$
Q1: $40,004 \times 0.025s = 1,000.1s$
Q2: $160,004 \times 0.025s = 40,000.1s$
Q3: $5 \times 0.025s = 0.125s$
Q4: $200,000 \times 0.025s = 5,000s$

Solution 6
Blocks required for r: $\lceil 10,000/5 \rceil = 2,000$ blocks
Blocks required for s: $\lceil 125/10 \rceil = 13$ blocks

a. Plan p1: $cost(p1) = n_r * b_s + b_r = 10,000 * 13 + 2,000 = 132,000$ block transfers

b. Plan p2: $cost(p3) = 3 * (b_r + b_s) = 3 * (2,000 + 13) = 6,039$

c. Plan p3: $cost(p4) = b_r + b_s = 2,000 + 13 = 2,013$

Solution 7

a. First, since condition $A < 10$ refers only to relation r, it can be pushed down to r. Similar, $B > 100$ can be pushed down to s, and we get $\sigma_{A+B<200}(\sigma_{A<10}(r) \times \sigma_{B>100}(s))$, which produces a smaller Cartesian product.

   Second, the condition $A + B < 200$ can be pushed down to transform the Cartesian product into a join. The final expression is then: $\sigma_{A<10}(r) \bowtie_{A+B<200} \sigma_{B>100}(s)$

b. Create an ordered index on attribute B of relation s. The index can then be used to locate the first tuple with $B > 100$ and then continue to scan the relation. An index for the condition on relation r is not useful, since the relation is anyway scanned from the beginning.

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. T_1 might fail after T_2 already committed (and T_2 used A which was produced by T_1).

c. No. Assume $ts(T_1) = 1$ and $ts(T_2) = 2$. Then T_2 sets the read timestamp of B to $R - ts(B) = 2$. When T_1 wants to write B, we have $ts(T_1) < R - ts(B)$, thus the write operation is rejected and T_1 is rolled back.
If $ts(T_2) = 1$ and $ts(T_1) = 2$ the same situation appears with the data item A.