Database Management Systems
Written Exam

07.02.2011

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	10	
3	24	
4	20	
5	10	
6	8	
7	8	
Total	100	

Exercise 1 (20 pt) Answer the following questions:

a. Mention two different techniques to optimize disk-block access?

b. For which operation is a clustering file organization advantageous?

c. What are two index structures that can efficiently handle multi-key queries?

d. What are the two steps to evaluate $\sigma_{A \geq v}(r)$ if a primary index on A exists?

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

f. What are the 3 steps of query processing?

g. What is always applicable: materialized evaluation or piplined evaluation?

h. The wait-for graph is used to detect conflict serializaility or deadlocks?

i. What is a cascading rollback?

j. Consider log-based recovery with immediate DB modifications and the following log file: $\langle T_0, start\rangle$, $\langle T_0, A, 1000, 950\rangle$, $\langle T_0, B, 2000, 1950\rangle$. What actions are performed if the system crashes in this situation?

Exercise 2 (10 pt) Consider the following relation:

Branch	Customer	Account
Downtown	Smith	237
Downtown	Jones	222
Mianus	Smith	250
Downtown	Turner	300
Mianus	Jackson	200
Mianus	Hayes	382
Downtown	Williams	180
Brighton	Jackson	290

Suppose that a branch with all its customer and accounts shall be stored in a variable-length record. Show the file organisation for the following methods:

a. Byte string representation

b. Fixed-length representation with pointer (using anchor and overflow block)

c. Slotted page structure

Exercise 3 (24 pt) Consider the following relation r:

	Course	StudID	Grade
r_0	DMS	2100	18
r_1	ITP	2157	18
r_2	ITP	2230	30
r_3	DMS	2177	24
r_4	OS	2340	30
r_5	ITP	2200	23
r_6	DMS	2157	28
r_7	DB	2300	30
r_8	DMS	2263	25
r_9	DB	2299	28

Show the following index structures and file organisations:

a. An index-sequential file organisation with a primary sparse index on *StudID*. For a search-key k, an index entry is created if $k \mod 100 = 0$.

b. Extend the index-sequential structure in a) with a secondary B$^+$-tree index on *Grade*. Assume $n = 3$ for the B$^+$-tree. The tuples are read sequentially as stored in the index-sequential file in a).

c. A hash file organisation using extendable hashing on *Grade* and the hash function $h(n) = n \mod 8$. Each bucket holds at most 2 tuples. Show the structure after inserting $r_0 - r_4$ and after inserting all tuples.

d. A bitmap index on *Course*.

Exercise 4 (20 pt) Assume two identical relations $r(A)$ and $s(A)$ with r being stored in a sequential (ordered) file and s being stored in an unordered file. The block size is 2,000 Bytes, the tuple size 10 Bytes, and the cardinality is 800,000 tuples for both relations. 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 5 (10 pt) Consider a relation $Grades(Stud, Grade)$ that contains the following tuples: $(Jan, 25)$, $(John, 25)$, $(Ann, 25)$, $(Sue, 18)$, $(Pete, 30)$, $(Sarah, 20)$, $(Ron, 27)$, $(Julia, 22)$, $(Bob, 18)$, $(Luk, 23)$, $(Tim, 25)$. Further, assume that only one tuple fits in a block, and the memory holds at most 3 blocks.

a. Show the runs created on each pass of the sort-merge algorithm, when applied to sort the *Grades* relation; sort on the *Grade* attribute.

b. What is the total number of block transfers? Explain your answer.

Exercise 6 (8 pt) Given is the following schedule over transactions T_1, T_2, T_3:

T_1	T_2	T_3
	read(Z)	
	read(Y)	
	write(Y)	
		read(Y)
		read(Z)
read(X)		
write(X)		
		write(Y)
		write(Z)
	read(X)	
read(Y)		
write(Y)		
	write(X)	

Answer the following questions and explain your answers:

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

b. Is the schedule view serializable to $\langle T_1, T_2, T_3 \rangle$?

Exercise 7 (8 pt) Given is the following schedule over transactions T_1, T_2:

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

Answer and explain the following questions:

a. Is this schedule possible under the two-phase locking protocol? If yes, add the lock and unlock instructions.

b. Is the schedule possible under the timestamp protocol? (Assume $TS(T_1) = 1$ and $TS(T_2) = 2$.)

Solution 1

a. Disk-arm scheduling, appropriate file organization (use of write buffers, use of log disks)

b. Join.

c. Bitmap index and grid file index

d. Step 1: Use index to find first tuple with $A \geq v$
Step 2: Scan relation sequentiall from there.

e. $M - 2$

f. (i) Parsing and translation, (ii) optimization, and (iii) evaluation

g. Materialized evaluation

h. Deadlocks

i. A single transaction leads to a series of transaction rollbacks.

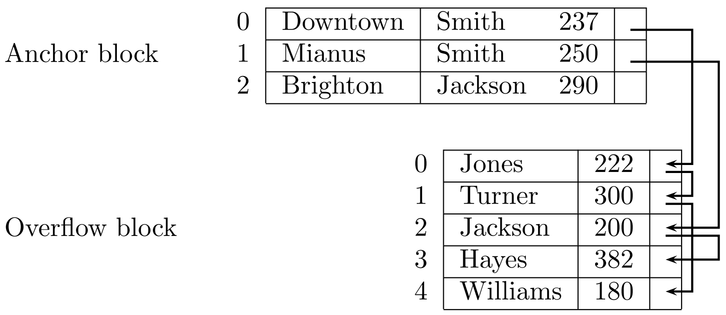
j. $undo(T_0)$, i.e., A is restored to 1000, and B is restored to 2000.

Solution 2

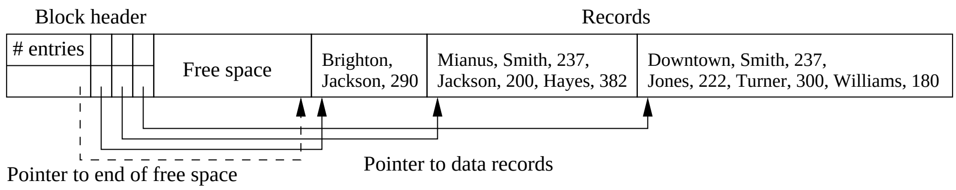
a. Byte string representation

0	Downtown	Smith	237	Jones	222	Turner	300	Williams	180	⊥
1	Mianus	Smith	250	Jackson	200	Hayes	382	⊥		
2	Brighton	Jackson	290	⊥						

b. Fixed-length representation with pointer (using anchor and overflow block):



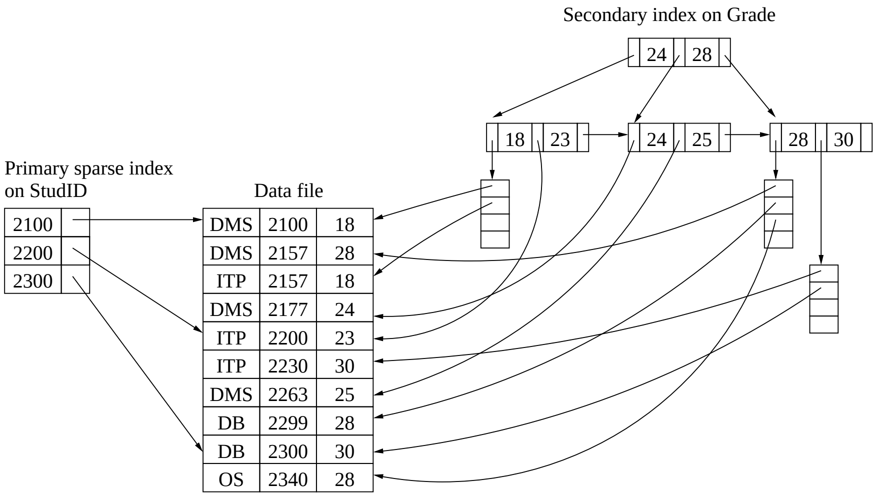
c. Slotted page structure:



Solution 3

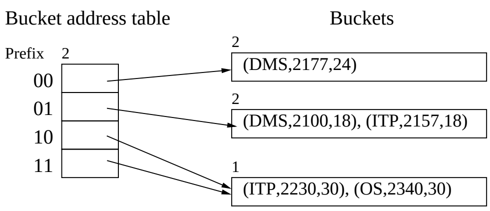
a. Index-sequential file organisation with primary index (see point b.)

b. Secondary B^+-tree index

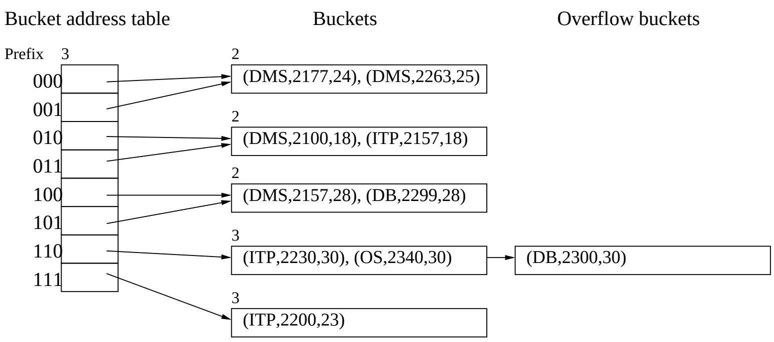


c. Extendable hashing

- after inserting $r_0, \ldots, r_4$:



- after inserting all tuples:



d. Bitmap index for *Course*:

DMS:	[1 0 0 1 0 0 1 0 1 0]
ITP:	[0 1 1 0 0 1 0 0 0 0]
OS:	[0 0 0 0 1 0 0 0 0 0]
DB:	[0 0 0 0 0 0 0 1 0 1]

Solution 4

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

Relation r:
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

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

Relation s:
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: $\sigma_{A<5,009,500}(s)$: 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. Relation r:
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

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

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

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: $(1,900 + 19) \times 0.025 = 47.975sec$

Solution 5

a. In the following we use only the names to refer to the tuples (note that the relation shall be sorted on the *Stud* attribute).

Step 1: Create 4 sorted runs with 3 tuples each:
(Ann, Jan, John), (Pete, Sara, Sue), (Bob, Julia, Ron), (Luk, Tim)

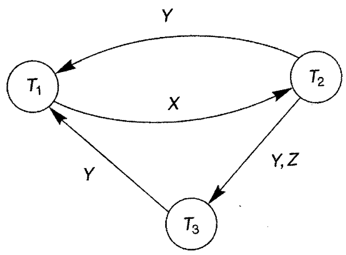
Step 2: Merge pass that merges two runs into one run. Thus the number of runs decreases by the factor of 2:
(Ann, Jan, John, Pete, Sara, Sue), (Bob, Julia, Luk, Ron, Tim)

Step 3: The runs after the second merge pass are:
(Ann, Bob, Jan, John, Julia, Luk, Pete, Ron, Sara, Sue, Tim)

b. Step 1: 11 x 2 = 22 block transfers (read and write)
Step 2: 11 x 2 = 22 block transfers (read and write)
Step 3: 11 x 1 = 11 block transfers (only read)
$\Rightarrow$ 55 block transfers

Solution 6

a. Conflict graph:



The schedule is not conflict serializable, since the conflict graph contains cycles.

b. No.
Example of violating a condition for view serializability: In the concurrent schedule T_2 reads the initial value of Y, and in $\langle T_1, T_2, T_3 \rangle$ the transaction T_2 reads the value of Y which is produced by T_1 (but should read the initial value).

Solution 7

a. Yes.

	T_1	T_2
1	lock-S(A)	
2	read(A)	
3		lock-X(B)
4		write(B)
5		unlock(B)
6	lock-S(B)	
7	read(B)	
8	unlock(A)	
9	unlock(B)	

b. No.
T_2 sets the W-timestamp of B to 2.
Then the *read*(B) operation of T_1 is rejected, since the timestamp of T_1 is smaller than the W-timestamp of B (that is, a younger transaction modified the value of B, and T_1 should have read an older version of B).