Database Management Systems
Written Examination

16.06.2006

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:

1. What are the two methods to store variable-length records with a fixed-length representation?
2. What are the two forms of pipelining in the evaluation of complex expressions?
3. Explain the two qualities of an ideal hash function?
4. Which of the two index strategies, hash function or primary B$^+$-tree, is preferable in the following cases (index is on A): (i) Equality selection on A, and (ii) Range selection on A
5. What are the 3 steps in query processing?
6. What are two alternative ways to evaluate the following complex join over relations r and s: $r \times_{\theta_1 \vee \theta_2 \vee \ldots \vee \theta_n} s$
7. Consider the materialized view $v = r \bowtie s$ and an update to r. How is the incremental view maintenance defined for inserts (i_r) and deletes (d_r).
8. What are the states of a transaction and how is the transition between them?
9. What is the definition of a schedule?
10. What is required for a schedule to be cascadeless?

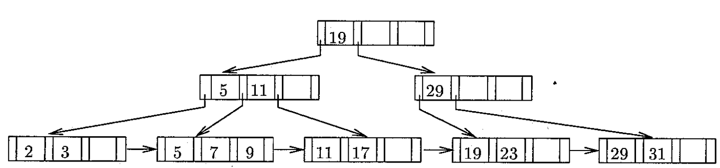
Exercise 2 The following table shows a file organization that represents variable-length records using the pointer method ("$\uparrow r_i$" denotes a pointer to record r_i, and $\perp$ denotes the end of a chain).

r_0	Jan	P1	400	$\uparrow r_2$
r_1	Joe	P3	350	
r_2		P2	500	$\perp$
r_3	Ann	P1	700	$\uparrow r_4$
r_4		P4	900	$\perp$

a. Show the file after the execution of the following steps:
   - Insert(Jan, P7, 800)
   - Insert(Ann, P2, 250)
   - Delete(Jan, P1, 400)

b. Transform the result of a.) into a pointer representation that uses an anchor block and an overflow block.

Exercise 3 Given is the following B^+-tree:



Show the B$^+$-tree after the first and the last of the following operations (applied in that order):

- Insert 10
- Insert 8
- Delete 23

Exercise 4 Consider the following student relation `stud`, where the grade ranges from A to D and the year ranges from 1 to 3:

	Name	Course	Grade	Year
0	Jan	AI	B	2
1	Jan	DB	A	2
2	Jan	ITP	B	1
3	Ann	OS	C	1
4	Bob	CSA	A	1
5	Bob	IC	D	2
6	Sue	BIS	C	2
7	Sue	DBMS	A	2

a. Create a bitmap index on the attributes Grade and Year.

b. Explain the use of this index for the evaluation of $\sigma_{Grade=A \wedge Year=1}(\texttt{stud})$ (all students with an A grade in year 1)

Exercise 5 Consider a relation `emp` with schema (`ID`, `Name`, `Dept`, `Salary`) and the following characteristics: $|\texttt{emp}| = 1,000,000$; the size of the attribute values is: `ID` = 4 bytes, `Name` = 50 bytes, `Dept` = 30 bytes, `Salary` = 6 bytes. The `ID` values are equally distributed between 1 and 10,000,000, and there are no two tuples with identical `ID`. A pointer occupies 6 bytes, and we assume a block size of 2,000 bytes. Further, we assume seek time = 0.016 sec, latency = 0.016 sec, and transfer time = 0.001 sec.

a. Determine the number of blocks needed to store the relation `emp` if 10% of each data block are reserved for future insertions.

b. Assume a B^+-tree index on the attribute `ID`. Determine the number of blocks (= number of nodes) used for the B^+-tree if index blocks are filled up to 75%.

c. Consider the following two queries:

- Q1: $\sigma_{\texttt{ID}=1Mio}(r)$
- Q2: $\sigma_{\texttt{ID}>7.5Mio}(r)$

Determine the number of block IOs (distinguishing between index blocks, data blocks, and total number of blocks) for the following two cases: (i) the B^+-tree is a primary index and (ii) the B^+-tree is a secondary index.

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

Exercise 6 Assume two relations $r(A, C)$ and $s(B, D)$ and the following relational algebra expression:

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

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.

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, 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)	

a. Draw the precedence graph of this schedule.

b. Show that the schedule is not conflict serializable.

c. Design a concurrent schedule of T_1, T_2 and T_3 that is conflict serializable. Specify also the equivalent serial schedule.

Solution 1

1. Reserved space and pointers

2. Demand-driven (or lazy) evalution and produce-driven (or eager) evaluation

3. The distribution is uniform: Each bucket is assigned the same number of search-key values from the set of all possible values.
   The distribution is random: In the average case each bucket will have the same number of records assigned to it irrespective of the actual distribution of search-key values in the file.

4. Equality selection on A: hash function
   Range selection on A: B^+-tree index

5. Parsing and translation, optimization, evaluation

6. Use nested loops/block nested loops
   Compute as the union of the individual joins $r \times_{\theta_i} s$, i.e., $r \times_{\theta_1} s \cup \ldots \cup r \times_{\theta_n} s$

7. For inserts: $v^{new} = v^{old} \cup (i_r \bowtie s)$
   For deletes: $v^{new} = v^{old} - (d_r \bowtie s)$
   where v^{new} is the new view and v^{old} is the old view before the update.

8. States: active, partially committed, committed, failed, aborted
   Transitions: `active` → `partially committed`, `partially committed` → `committed`, `partially committed` → `failed`, `active` → `failed`, `failed` → `aborted`

9. A schedule is a sequence that indicates the chronological order in which instructions of concurrent transactions are executed. It must consist of all instructions of all involved transactions and it must preserve the order of instructions within each individual transaction.

10. For each pair of transactions T_i and T_j such that T_j reads a data item previously written by T_i, the commit operation of T_i appears before the read operation of T_j.

Solution 2

a. File after the 3 update operations:

r_1	Joe	P3	350	
r_2	Jan	P2	500	$\uparrow r_5$
r_3	Ann	P1	700	$\uparrow r_4$
r_4		P4	900	$\uparrow r_6$
r_5		P7	800	$\bot$
r_6		P2	250	$\bot$

b. Pointer representation with anchor block and overflow block:

Anchor block:

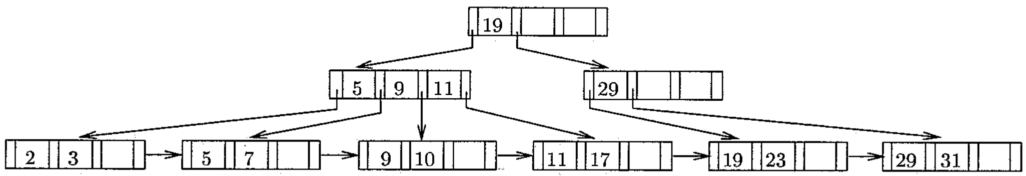
r_0	Joe	P3	350	
r_1	Jan	P2	500	$\uparrow s_0$
r_2	Ann	P1	700	$\uparrow s_1$

Overflow block:

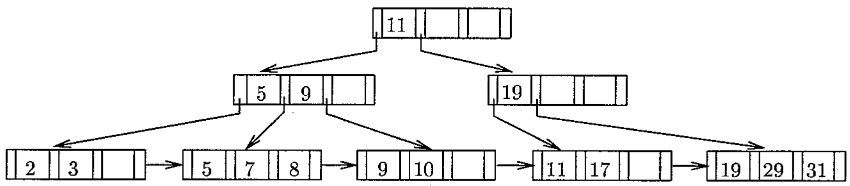
s_0	P7	800	$\bot$
s_1	P4	900	$\uparrow s_2$
s_2	P2	250	$\bot$

Solution 3

- After Insert 10



- After Insert 8 and Delete 23



Solution 4

a. Bitmap indexes:
Grade A: `[0 1 0 0 1 0 0 1]`
Grade B: `[1 0 1 0 0 0 0 0]`
Grade C: `[0 0 0 1 0 0 1 0]`
Grade D: `[0 0 0 0 0 1 0 0]`

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

b. Take the bitmap vector of Grade A and Year 1 and compute the logical AND:
`[0 1 0 0 1 0 0 1]` AND `[0 0 1 1 1 0 0 0]` = `[0 0 0 0 1 0 0 0]`
Retrieve the tuple(s) with a '1' in the resulting bitmap vector, i.e., tuple 4.

Solution 5

a. Data blocks:
$0.9 * 2,000 = 1,800$ bytes used in every data block
$\lfloor 1,800/(4 + 50 + 30 + 6) \rfloor = 20$ tuples/block
$\lceil 1,000,000/20 \rceil = 50,000$ data blocks needed for `emp`

b. Index blocks:
$0.8 * 2000 = 1,600$ bytes used in every index block
$\lfloor 1,500/(4 + 6) \rfloor = 150$ index entries/block
- level 3 (leaf nodes): $\lceil 1,000,000/150 \rceil = 6,667$ blocks
- level 2: $\lceil 6,667/150 \rceil = 45$ blocks
- level 1: $\lceil 45/150 \rceil = 1$ block
$\Rightarrow 6,713$ index blocks are needed

c. $Q1$ – (i) primary index:
3 index blocks (one at each level)
1 data block

$\Rightarrow 3 + 1 = 4$ blocks are transfered in total

$Q1$ – (ii) secondary index: the same as for primary index

$Q2$ – (i) primary index:
3 index blocks (one at each level) to locate the first tuple
$\lceil 250,000/20 \rceil = 12,500$ data blocks to retrieve (on avg. $250,000$ tuples match, and all matching tuples are in contiguous blocks
$\Rightarrow 3 + 12,500 = 12,503$ blocks are transfered in total

$Q2$ – (ii) secondary index:
On avg. $250,000$ data tuples match
3 index blocks (one at each level) to locate the first tuple
$\lceil 250,000/150 \rceil = 1667$ index (leaf) nodes/blocks to locate the other tuples
$250,000$ data blocks to retrieve (in the worst case each matching tuple might be in a different block)
$\Rightarrow 3 + 1,667 + 250,000 = 251,670$ blocks are transfered in total
(a full scan of the data relation without using the index would be more efficient)

d. Execution time:
$Q1$ – (i) primary index: $4 * 0.033 = 0.132$ sec
$Q1$ – (ii) secondary index: the same as for primary index
$Q2$ – (i) primary index: $12,503 * 0.033 = 412.6$ sec
$Q2$ – (ii) secondary index: $251,670 * 0.033 = 8,305.1$ sec

Solution 6 A part of the selection condition ($A < 10$) refers only to relation r and another part ($B > 100$) refers only to relation s. These two conditions can therefore be applied to the argument relations before the evaluation of the Cartesian product, reducing in this way the size of the Cartesian product. Additionally, the conidtion $A+B < 200$ can be moved to the Cartesian product, yielding a join. The transformed expression is: $(\sigma_{A<10}(r)) \bowtie_{A+B<200} (\sigma_{B>100}(s))$

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

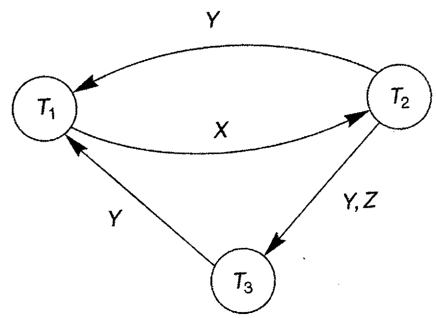
Plan p1:
$cost(p1) = n_{r_1} * b_{r_2} + b_{r_1} = 10,000 * 13 + 2,000 = 132,000$ block transfers

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:
$cost(p3) = 3 * (2,000 + 13) = 6,039$

Solution 8

a. Precedence graph:



b. There is a cycle in the precedence graph: $T_1 \xrightarrow{X} T_2$ and $T_2 \xrightarrow{Y} T_1$. Hence, the schedule is not conflict serializable.

c. The following schedule is conflict serializable to the serial schedule $\langle T_3, T_1, T_2 \rangle$

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