Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science

TU Delft

Examination for

IN4183 Advanced Database Technology

Tuesday, August 19, 2008, 14:00 – 17:00 hours

This examination contains 5 open questions
Total number of pages (including this one): 7

Exercise	Points
1	15
2	15
3	30
4	20
5	20
Total	100

The subquestions within each main question have equal weight
Score for a positive grade: 55 points

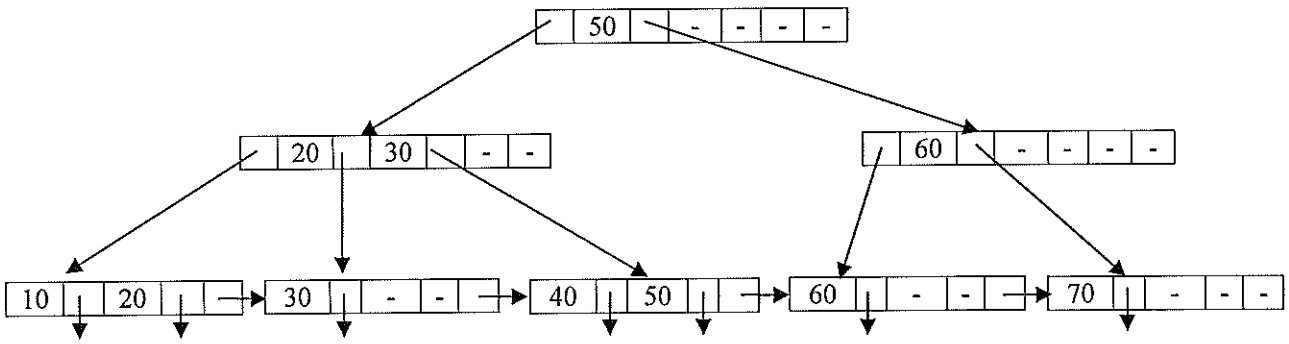
Use of books and readers, as well as calculators is **not** permitted

Specify your name and student number on each answer sheet. You are free to give your answers either in Dutch or in English

This examination covers the material contained in the Lecture Sheets 2007-2008 (cf. Blackboard), and Chapters 13-19 of Elmasri & Navathe, "Fundamentals of Database Systems", 5th ed.

Question 1 (15 points)

Consider a B+-tree of order 4 for the internal nodes (4 pointers possible) and of order 2 for the leaf nodes (2 data pointers and data values possible). Suppose the tree is filled as in the Figure below.



The arrows pointing down in the leaf nodes are pointers to the data file.

1. What is the resulting tree when the value 25 is inserted in the tree of this figure?
2. What is the resulting tree when the value 45 is inserted in the tree **of the figure** (i.e. not in the tree resulting from your answer to the previous question)?
3. What is the resulting tree when the value 50 is deleted from the tree **of the figure**?
4. Same question as 3. but now the value 30 is deleted.
5. Same question as 3. but now the value 70 is deleted.

Draw the tree as it exists after each of these operations. In order to make life easier for you: you do not need to make drawings as extensive as I did in the figure above, it is enough if you give only the values on the different levels, you can omit the pointers. For instance the above figure can be represented as

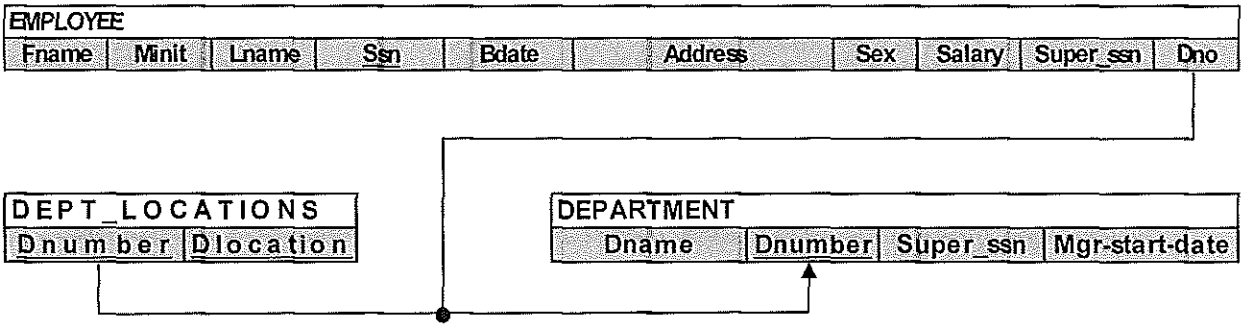
```
                    50,-,-

          20,30,-                   60,-,-

10,20   30,-   40,50        60,-    70,-
```

Question 2 (15 points)

One of the tables in a database of a department store is `PRODUCT`, containing descriptions of the products they sell. The key of this table is the attribute `ProductID`, which has a size of 10 bytes. The table contains 7500 records, each 120 bytes big.
The table is stored on a disk of which the block size is 512 bytes, block pointers for this disk are 7 bytes long. The file organization is unspanned.
In the next subquestions you are asked to give the maximum number of disk block accesses needed to retrieve the record containing a certain value of `ProductID`. You may assume that the file indeed contains such a record.

1. Suppose this data file is sorted on the key `ProductID` and there is no index on this attribute. How many disk accesses are at most needed if the access method is binary search? Explain your answer.
2. Suppose the table is not sorted on `ProductID`, but it has a single level secondary index on this attribute. Index entries are <`ProductID`, block pointer> pairs. How many disk accesses are at most needed if the access method is to exploit this index? Explain your answer.
3. Same situation as in 2., but now the index is multi level.
4. Same situation as in 1., but now there is a multi level clustered index on `ProductID`, and the access method is to use this index.

Question 3 (30 points)



This is part of the schema of the company database from the book, on which the following query is executed:

```
SELECT E.Fname, E.Lname, D.Dname
FROM EMPLOYEE E, DEPT_LOCATIONS DL, DEPARTMENT D
WHERE DL.Dnumber = D.Dnumber AND D.Dnumber = E.Dno AND
      DL.Dlocation = 'Houston' AND E.Salary > 20000;
```

For the database the following facts hold:

a) There are 5 buffers available in main memory. The size of the buffers and disk blocks is 1024 bytes.

b) The table EMPLOYEE consists of 10000 tuples, DEPARTMENT has 50 tuples and DEPT_LOCATIONS contains 150 tuples. A tuple in EMPLOYEE consists of 200 bytes, a tuple in DEPARTMENT is also 200 bytes, and a tuple in DEPT_LOCATIONS is 20 bytes.

c) There is a clustered index on Dno in EMPLOYEE with 2 levels (leaf level included).

d) The attribute Salary in EMPLOYEE has 1000 different values ranging from 10000 (lowest) to 90000 (highest). The attribute Dno in EMPLOYEE has 50 different values. The attribute Dlocation in DEPT_LOCATIONS has 15 different values and Dnumber in DEPT_LOCATIONS has 50 different values. The attribute Dname in DEPARTMENT has 50 different values. Assume a uniform distribution of these values over the tables.

e) The attribute Fname in EMPLOYEE has size 20 bytes, the attribute Lname in EMPLOYEE is 60 bytes big, and the attribute Dname in DEPARTMENT is 20 bytes.

f) The attribute Dnumber in DEPARTMENT is the key and the combination Dnumber, Dlocation is the key in DEPT_LOCATIONS.

Question 4 (20 points)

Consider the following three transactions,

T_1 $r_1(X)$; $w_1(X)$; $w_1(Y)$; $w_1(Y)$;
T_2 $r_2(X)$; $w_2(Z)$;
T_3 $r_3(X)$; $w_3(Z)$; $w_3(Y)$;

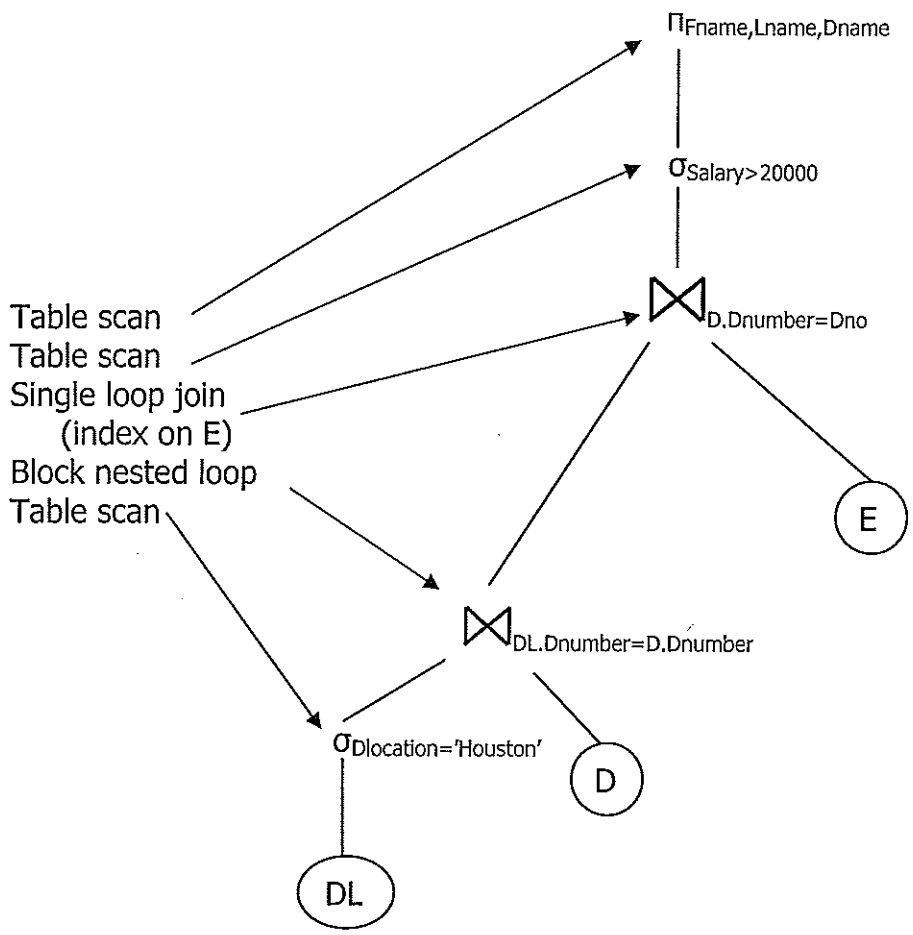
and the following schedule over these transactions

s $r_1(X)$; $r_2(X)$; $r_3(X)$; $w_1(X)$; $w_1(Y)$; $w_2(Z)$; $w_3(Z)$; $w_3(Y)$; $w_1(Y)$;

1. Show that this schedule s is not conflict serializable.
2. Show that s is view serializable, by giving a serial schedule of the form $T_i T_j T_k$ that is view equivalent to s. Explain why your serial schedule is view equivalent.
3. We add commit operators to schedule s obtaining the following schedule s'

$r_1(X)$; $r_2(X)$; $r_3(X)$; $w_1(X)$; $w_1(Y)$; $w_2(Z)$; c_2 ; $w_3(Z)$; $w_3(Y)$; c_3 ; $w_1(Y)$; c_1 ;

Suppose this schedule s' is submitted to a concurrency control with automatic locking (rigorous locking,- all locks are released on commit or abort). Specify for each operation that is submitted in the schedule what locks are obtained, released or denied, and whether the transition is allowed to proceed. Notice that the concurrency control will reorder the schedule s'. In your answer I want to see this reordering as well, i.e. all operations from s' should occur in your answer.



Suppose this query is executed according to the execution plan above (no pipelining). How many disk blocks are read and written when this query is executed according to this plan? Explain your answer. The cost of writing out the final result of the query should be taken into account.

Question 5 (20 points)

Consider the following three transactions (with explicit `begin` operations, here abbreviated as `bg`):

T_1 $bg_1; r_1(X); w_1(X); c_1;$
T_2 $bg_2; r_2(X); w_2(Y); c_2;$
T_3 $bg_3; r_3(X); w_3(X); c_3;$

The following schedule `s` is submitted to a concurrency control that implements strict timestamp ordering:

$$bg_2; bg_1; r_1(X); r_2(X); bg_3; w_1(X); r_3(X); c_1; w_3(X); c_3; w_2(Y); c_2;$$

Suppose at the start of `s` there are no other transactions active. The algorithm manipulates the following variables: $TS(T_1)$, $TS(T_2)$, $TS(T_3)$, `read_TS(X)`, `read_TS(Y)`, `write_TS(X)`, `write_TS(Y)`, `flag(X)` and `flag(Y)`.

1. For each operation in the schedule the concurrency control executes zero or more assignments on the above variables. Give these assignments. Alternatively, the concurrency control can decide to not execute such an operation. In that case, state what the action will be.

2. Now we change T_2 a little bit. It does not write to `Y` but to `X`, turning it into

   T_2' $bg_2; r_2(X); w_2(X); c_2;$

   The schedule `s` now changes accordingly into `s'`:

   $$bg_2; bg_1; r_1(X); r_2(X); bg_3; w_1(X); r_3(X); c_1; w_3(X); c_3; w_2(X); c_2;$$

   Now, the concurrency control will not execute $w_2(X)$; but will abort T_2' and roll this transaction back. Give the values of the above variables on which the concurrency control takes this decision. Also give an interpretation of these values that lead to the abort decision. Show the actions (assignments to the variables) that the concurrency control takes when it rolls back T_2' and executes this transaction again.

End of examination