浙江大学 2014－2015 学年秋季学期

《数据库系统设计》课程期末考试试卷

课程号：__21120302________，开课学院：____计算机学院______

考试试卷：√A 卷、B 卷（请在选定项上打 √）

考试形式：闭、开卷（请在选定项上打 √），允许带__1 张 A4 纸笔记__入场

考试日期：__2014__年__11__月__21__日，考试时间：__120__分钟

诚信考试，沉着应考，杜绝违纪。

考生姓名：______________学号：______________所属院系：_____________

题序	一	二	三	四	五	六	七	八	总 分
得分									
评卷人									

Problem 1: Buffer Manager (12 points, 3 points per part)

Consider DBMS's buffering of disk blocks in memory under the assumption that, to begin with, the buffer is empty, and that the maximum number of blocks that can be held in buffer is four. Assume in the following schedule that each of the data items A, B, C, D, E, F accessed in the schedule lies on distinct blocks.R5（E，8）

R_1(A,1) R_2(B,2) W_1(A,3) R_3(C,4) W_2(B,5) C_2 W_3(C,6) R_4(D,7)

R_5(E,8) W_5(E,9) R_6(A,3) R_3(F,10) W_3(F,11) W_4(D,12)

Where: R_i (X, V) means transaction T_i read value V from data X.

W_i (X, V) means Transaction T_i write value V to data X.

C_i means transaction T_i commits.

1) Name the first operation where an existing block in buffer must be dropped in order that another block can be read in.
2) Blocks in buffer are called dirty if they have been updated in buffer but not yet written back out to their place on disk. What are the dirty blocks in buffer at the time of the operation named in 1)?
3) Assume that DBMS is using an LRU buffer-replacement policy, and that each block is in use only for duration of the read or write that access it. What block will be dropped from buffer at the time of the operation named in 1)

4) Can DBMS simply drops the block mentioned in 3) at that time, forgetting its value, or must some other event take place first? Why?

Answer of Problem 1：

Problem 2: B+ -Tree (16 points, 4 points per part)

1) Construct a B+-tree (n=4) for the following set of key values: (5,10,15,20,25,30,35,40, 45, 50,55) .
   Assume that the tree is initially empty and values are added in ascending order.
2) For the B+-tree constructed in 1), show the form of the tree after the operation "delete 40".
3) Assume that the B+-tree (n=4) contains 2000 index items, please estimate the height of the B+-tree.
4) Assume that the B+-tree (n= 4)) contains 2000 index items, please estimate the size (i.e. the number of nodes) of the B+-tree.

Answer of Problem 2：

Problem 3: Query Optimization (10 points, 5 points per part)

Consider the following relational schema and SQL query:

account(**account_no: char(10)**, customer_name: char(10), branch: char(20), balance: integer)

access (**serial char(10)**, account_no: char(10), amount: integer, access_branch: char(20), access_date: date)

*note: account_no of table **access** references to table **account**.*

select account.account_no, account.customer_name
from account , access as A1, access as A2
where account.account_no = A1.account_no and A1.account_no= A2.account_no
and A1.access_branch =A2.access_branch
and A1.access_date=A2.access_date and account.branch='Hangzhou'
andA1.amount<=1000 and A2.amount>=63500

1) Identify a relational algebra expression that reflects the order of operations that a query optimizer would choose after algebra optimization.
2) What indexes might be of help in processing this query? Explain briefly.

Answer of Problem 3：

Problem 4: Query Size Estimation (10 points)

For the relational schema defined in problem 3, there are following assumptions:

- **account** table has 10,000records
- **access** table has 1,000,000 records
- **branch** attribute has 100 distinct values that are uniformly distributed.
- The value of **access_date** attribute is between '2001-01-01' and '2010-12-31'
- The value of **amount** attribute is between 1 and 100,000 that are uniformly distributed.

Please estimate the size (i.e. number of records) returned by the SQL statement given in the **problem 3**. (Write out your answer step by step).

Answer of Problem 4：

Problem 5: Query Cost Estimation (10 points)

For the relational schema defined in **problem 3**, assume that:

- The **account** table has 10,000 records
- The **access** table has 1,000,000 records
- The file system support 4K byte blocks.
- There are 50 buffer pages (blocks) available in memory for operating join.
- The attribute with ‘integer’ type needs 4 bytes.
- The attribute with ‘date’ type needs 4 bytes.

Please estimate the best cost for evaluating **account** $\bowtie$ **access** with Block Nested-Loop Join method.　(Hint: the cost is measured with the number of blocks transferred to main memory and the times to seek disk.)

Answer of Problem 5：

Problem 6: Two Phase Locking protocol (10 points)

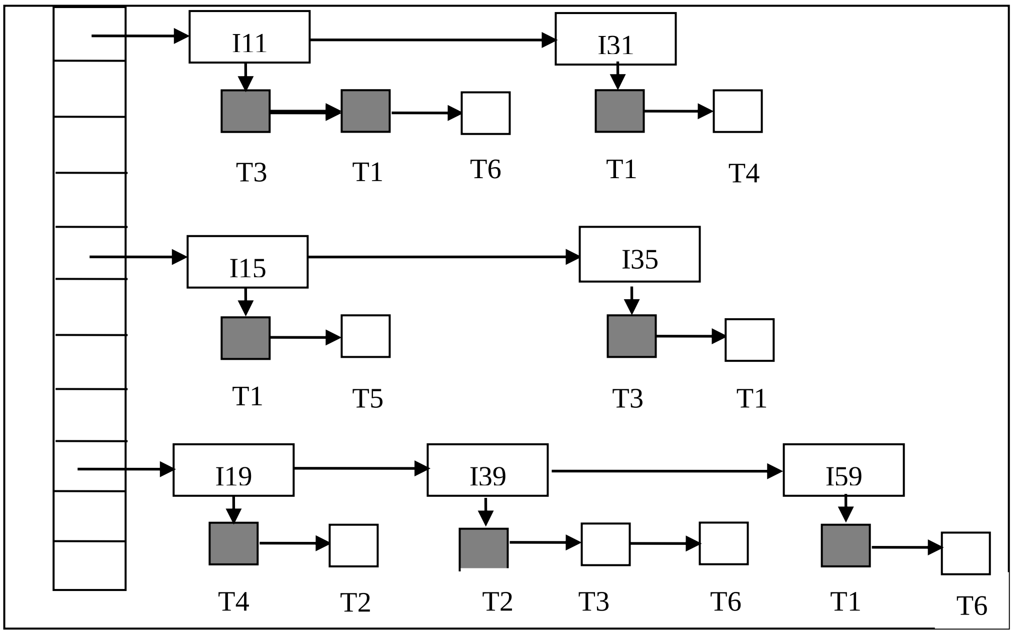
Prove that in a schedule where each transaction obeys the Two Phase Locking protocol, it is possible to move all the read and write actions of the transaction with the first unlock action forward to the beginning of the schedule without passing any conflicting actions.

Answer of Problem 6：

Problem 7: Deadlock Handling (12 points, 4 points per part)

The following figure shows an instance of a lock table. There are 6 transactions (T1-T6) and 7 data items (I11, I31, I15, I35, I19, I39, I59). Granted locks are filled (black) rectangles, while waiting requests are empty rectangles.

1) Which transactions are involved in deadlock?
2) In order to break the deadlock and release most lock resources, which transaction (victim) should be rolled back?
3) Please draw the lock table after the victim transaction of 2) is rolled back.



Answer of Problem 7：

Problem 8: Crash Recovery (20 points, 4 points per part)

Consider the following log sequence of transactions T1, T2, T3, T4 and T5. Supposing the system crashes just after the last log record. Please answer each of the following questions:

1) What are the values of A, B , C and D in the database after system crash?
2) Which transactions should be undone? Which transactions should be redone?
3) What are the start and end points for undo and redo processes respectively?
4) What are the values of A, B , C and D after system recovery?
5) If the system crashes again while the system is doing database recovery from previous crash, how does the DBMS handle such kind situation?

[1] <T1 **start**>
[2] <T1, A, 10, 20>
[3] <T2 **start**>
[4] <T2, B, 30, 40>
[5] <T1, C, 50 ,60>
[6] <T3 **start**>
[7] <checkpoint { T1,T2,T3}>
[8] <T3, D, 70, 80>
[9] <T2 **commit**>
[10] <T3, B, 40, 50>
[11] <T1 **commit**>
[12] <checkpoint {T3}>
[13] <T3, A, 20, 30>
[14] <T4 **start**>
[15] <T4, C, 60, 70>
[16] <checkpoint {T3,T4}>
[17] <T3 **commit**>
[18] <T5 **start**>
[19] <T5 , A, 30, 40>

<u>Answer of Problem 8:</u>