

浙江大学 2011 - 2012 学年秋季学期

《数据库系统设计》课程期末考试试卷

课程号: 21120301, 开课学院: 计算机, 任课教师:

考试试卷: ☒ A 卷、B 卷

考试形式: ☒ 闭、开卷, 允许带 1 张 A4 纸笔记入场

考试日期: 2011 年 11 月 7 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	总 分
得分						
评卷人						

Problem 1 : Index and Query (20 points)

Consider the following relational schemas:

Create table emp(id char(20) primary key,
 name char(30),
 age int,
 salary real,
 address char(50),
 dname char(30) ,
 foreign key (dname) references dept);

Create table dept(dname char(30) primary key,
 location char(50));

Assume table *emp* has 6000 records, which are stored sequentially based on their ids; table *dept* has 200 records, which are stored sequentially based on their *dnames*; each (data or index) block can hold 2K (2048) bytes, in which 48 bytes are reserved for the block header. The attribute with *real* type is 8 bytes and *int* type is 4bytes. No records span two or more blocks. Each block pointer(which points to a block) is 6 bytes. Each record pointer (which points a data record) is 8 bytes (includes a block pointer + 2bytes offset within a block). Index entries do not span blocks.

(1) Suppose we want to build a conventional primary index INDEX1 on *emp.id* using a dense

index. What is the minimum number of blocks needed for INDEX1?

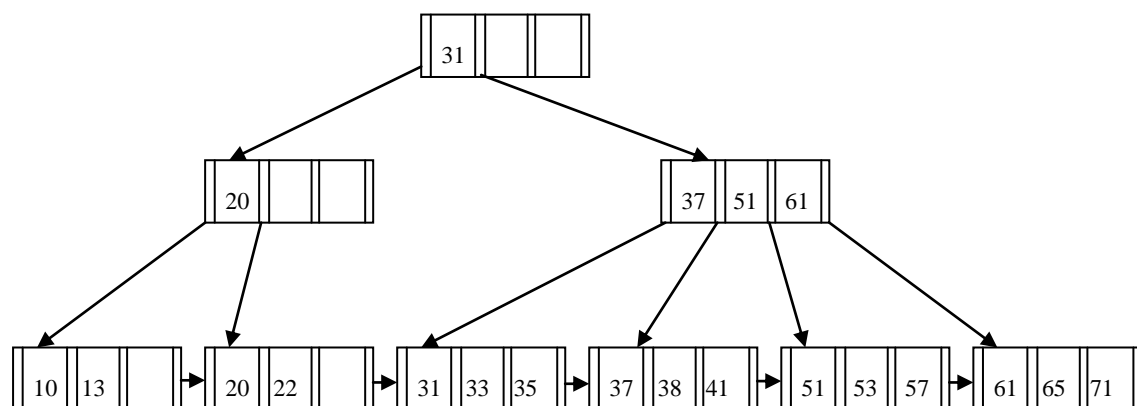
(2) For the same primary index INDEX1 on *emp.id*, we now want to use a sparse index. What is the minimum number of blocks needed for INDEX1?

(3) Suppose we want to build a secondary index INDEX2 on *dept.location*. What type of INDEX2 makes sense, sparse or dense? What is the minimum number of blocks needed for INDEX2?

(4) Suppose now we want to do the natural join $emp \bowtie dept$. The available memory buffer size is 10 blocks. Estimate the cost for the join, and explain the algorithm to get the lowest cost.

Problem 2 : B+ Tree (24 points)

For the following B+ tree (n=4):



Answer the following questions:

- (1) Draw the B+ tree after insert an index item with key '55' to the above tree.
- (2) Draw the B+ tree after delete an index item with key '13' from the original tree.
- (3) Suppose the B+ tree contains 900 index items, please estimate the height of the B+ tree. (Consider the tree with half-filled and full-filled respectively.)
- (4) Suppose that the B+ tree contains 900 index items, please estimate the size (i.e. the number of nodes) of the B+ tree. (Consider the tree with half-filled and full-filled respectively.)

Problem 3. Query Optimization (12 points)

Suppose we have the database tables $R(\underline{ID}, A)$ and $S(\underline{ID}, B)$ in which IDs are keys, and we would like to execute the following query:

SELECT A, B

FROM R, S

WHERE $R.ID = S.ID$ AND $A = 'a1'$ AND $B = 'b1'$;

We have the following statistics and assumptions:

R stored contiguously in $b_r = 1000$ blocks;

S stored contiguously in $b_s = 600$ blocks;

$V(R, ID) = 15000$; // $V(R, ID)$ is the number of distinct values that appear in the

relation R for attribute ID .

$V(R, A) = 50;$

$V(S, ID) = 10000;$

$V(S, B) = 100;$

Assume that the values in R.A and S.B are uniformly distributed, in which value 'c1' and 'b1' exist. The available memory buffer size is 30 blocks.

Please answer the following questions:

- (1) Please give out an execution plan with the lowest cost for the query at the relational-algebra level.
- (2) Estimate the cost for the query with your optimized algorithm.(Note: the cost is measured with the number of blocks transferred between disk and memory buffer and the seek times of disk. As usual, do not count the cost to write out the final query result.)

Problem 4: Concurrent Control (24 points)

There are two concurrent schedules:

$S1 : w1(Y); w2(X); r3(X); r1(Z); w4(Z); r4(X); r2(Y).$

$S2 : r1(X); r3(X); r1(Z); r4(V); w3(Z); w2(Y); w2(X); w4(V); r3(Y).$

For each of the above schedules $S1$ and $S2$, answer the following questions respectively:

- (1) Draw the precedence graph for the schedule.
- (2) Is the schedule conflict-serializable? If yes, describe all the conflict-equivalent serial schedules. (You do not need to explicitly write out the serial schedules, just use the form like $\langle T1, T3, T2 \rangle$.) If no, explain why it is not.
- (3) Could the schedule be created by the 2PL protocol? If yes, put the lock and unlock actions (e.g. use 'Ls(Q)' for Lock-s(Q), 'Lx(Q)' for Lock-x(Q), 'UL(Q)' for Unlock(Q)) into the schedule to show it can be generated by 2PL. If not, explain briefly why.

Problem 5 Crash Recovery (20 points)

Consider the following transaction log of a database system. Suppose the system crashes just after the last log record.

- 1) $\langle T1 \text{ START} \rangle$
- 2) $\langle T1, A, 20, 30 \rangle$
- 3) $\langle \text{CKPT} (T1) \rangle$
- 4) $\langle T2 \text{ START} \rangle$
- 5) $\langle T1, A, 30, 50 \rangle$
- 6) $\langle T2, B, 52, 66 \rangle$
- 7) $\langle T3 \text{ START} \rangle$
- 8) $\langle T3, E, 0, 15 \rangle$
- 9) $\langle T2 \text{ COMMIT} \rangle$
- 10) $\langle \text{CKPT} (T1, T3) \rangle$

- 11) < T3, E, 15, 40 >
- 12) < T1, C, 45, 90 >
- 13) < T1 COMMIT >
- 14) < T3 ABORT >
- 15) < T4 START >
- 16) < T4, D, 39, 0 >

Assume the log entries are in the format <T_{id}, Variable, Old value, New value >.

Please answer each of the following questions:

- (1) What are the values of A, B, C, D and E in the database just 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, D and E after system recovery ?