American University of Beirut

CMPS 377

Advanced Algorithms and Data Structures

Fall 2001-2002

Final Exam

Date: Feb. 6th, 2002, 3:00 – 5:00pm.
Instructor: Jihad Boulos

Name: .....

ID #: .....

AMERICAN UNIVERSITY
LIBRARY
OF BEIRUT

This is an open-book, open-note exam. Your exam should have 16 pages, and there are 10 questions totaling 100 points. You may use your notes, class handouts, and the main course textbook. You are **NOT** allowed to use any external notes. Your answers should be concise, and when possible should be a list of important points rather than prose. Solve as much problems as you can. I advise each one of you to pick the problems that he/she thinks are easiest for him/her and work on them. I also advise you to spend time on understanding the problem and budget your time for solving each problem, or else you will be wasting a lot of time on one problem and will run out of time for other problems.

AMERICAN UNIVERSITY
LIBRARY
OF BEIRUT

Exercise 1 (10 points):

Suppose we have two relations: R(a, b, c) and S(c, d, e). Give 6 valid logical plans (either in relational algebra or as logical plan trees) for the following SQL query:

```
SELECT B, C, D
FROM R, S
WHERE R.c = S.c AND R.a = 5
```

Exercise 2 (5 points):

Suppose we have two relations: R(a, b, c) and S(c, d, e). Consider the following relational algebra expression:

$$\prod_{a,d}[\sigma_{E.e=5 \wedge R.a=10 \wedge R.c=S.c}(R \times S)]$$

Transform this expression into an equivalent expression that:

- Has no Cartesian products
- Performs projections as early as possible
- Performs selections as early as possible

An "early" selection/projection is a selection/projection that occurs as close as possible to the leaves in the logical plan tree form of the expression. Give projections priority over selections.

Exercise 3 (15 points):

Suppose relations R and S are not stored in any particular order. Suppose furthermore that two indexes exist on relation R, one on column *a* and one on column *c*. No other indexes are present. Give 7 valid physical plans (as physical plan trees) for the following logical plan:

$$[\sigma_{a=5}(R)] \bowtie_{R.c=S.c} [S]$$

Exercise 4 (15 points):

Here are the statistics about four relations: W, X, Y, and Z:

W(a, b)	X(b, c)	Y(c, d)	Z(d, e)
T(W) = 100	T(X) = 400	T(Y) = 200	T(Z) = 300
V(W, a) = 80	V(X, b) = 200	V(Y, c) = 200	V(Z, d) = 200
V(W, b) = 10	V(X, c) = 1	V(Y, d) = 120	V(Z, e) = 80

Estimate the sizes of the following expressions:

a) $\sigma_{a=35}(W)$ (assume that 35 is present in column a of relation W)

b) $\sigma_{a=35 \wedge b=5}(W)$ (assume that 35 is present in column a and 5 is present in column b)

c) $W \bowtie X$ (natural join)

d) $X \bowtie Y$

e) $W \bowtie X \bowtie Y \bowtie Z$

Exercise 5 (9 points):

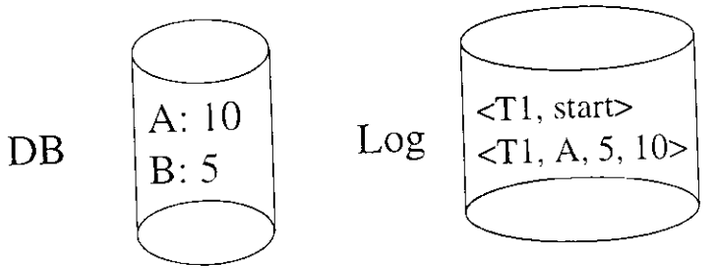
Suppose we have a transaction T1 that performs the following two actions:

A := A + 5; B := B + 5

Say that UNDO/REDO logging is in use, and that initially, A = 5 and B = 5. For each hypothetical disk state shown below, state whether it is a **legal** state for UNDO/REDO logging. If it is not a legal state, explain why not.

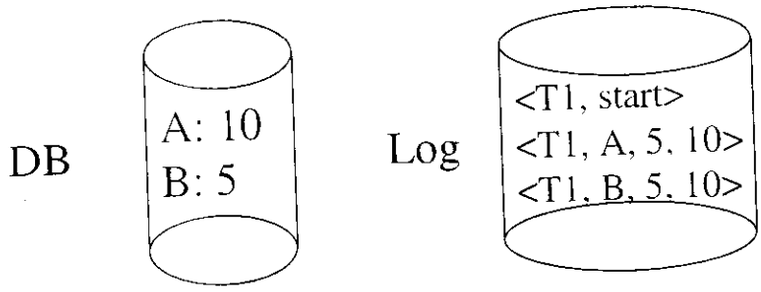
Assume the log entries are in the format
`<Tid, Variable, Old value, New value>`

a)



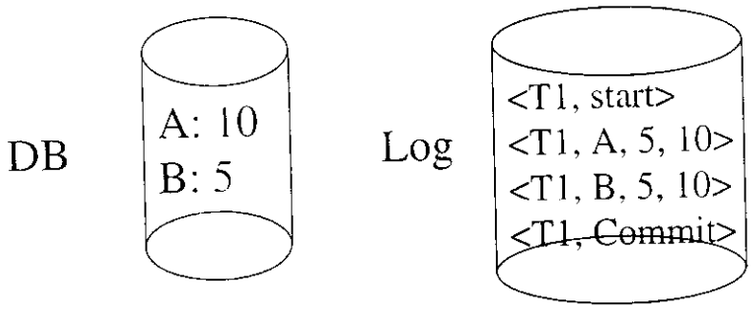
ANSWER:

b)



ANSWER:

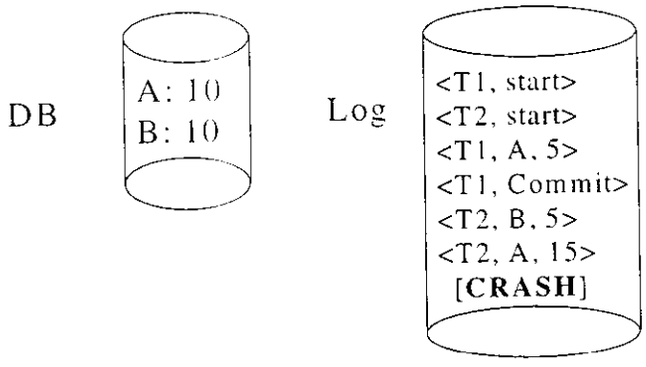
c)



ANSWER:

Exercise 6 (8 points):

Say the system reboots after a crash and finds the following disk state:



a) If the system is using UNDO logging, give the initial state of the database before T1 and T2 began executing (i.e., what were the initial values of A and B on the disk?).

b) If the system is using UNDO logging, what will be the final state of the database after recovery (i.e., what will be the values of A and B on the disk after the recovery process has finished?).

c) If the system is using REDO logging, give the initial state of the database before T1 and T2 began executing (i.e., what were the initial values of A and B on the disk?).

d) If the system is using REDO logging, what will be the final state of the database after recovery (i.e., what will be the values of A and B on the disk after the recovery process has finished?).

Exercise 7 (10 points):

The following table describes the undo/redo log found on a system after a failure.

Action Id	Action Type	Transaction Dd	Other Info
1	trans start	T_1	
2	db write	T_1	
3	trans start	T_2	
4	db write	T_2	
5	start dump	System	T_1 active
6	trans start	T_3	
7	db write	T_3	
8	end dump	System	
9	db write	T_3	
10	trans start	T_4	
11	db write	T_4	
12	db write	T_3	
13	commit	T_4	
14	commit	T_2	
15	start checkpoint	System	T_1, T_3 active
16	db write	T_1	
17	db write	T_3	
18	trans start	T_5	
19	db write	T_5	
20	trans start	T_6	
21	db write	T_6	
22	end checkpoint	System	
23	db write	T_5	
24	db write	T_3	
25	db write	T_1	
26	commit	T_5	
27	db write	T_6	
28	commit	T_3	
29	end of log		

The database dump that occurs between actions 5 and 8 writes a complete copy of the database to tape, to be used in case of medial failure.

a) Assume that at action 29 the system crashes but the database is *not* lost. Assume that a two pass algorithm is run for recovery, where actions are first undone, and then actions are redone. In the two lines below indicate what actions are undone in the first pass, and what actions are redone. Write the actions *in the order* they would be undone or redone (left to right). Identify each action by its id given in the first column in the table.

b) Now assume that the database is lost at action 29. In this case we load the saved database from tape and bring it up to date. In the two lines below again indicate what actions are undone and redone to bring the dumped database up to date.

Exercise 8 (10 points):

Assume the existence of two schemes R1 = {A,B,C,D} and R2 = {E, F, B, C} with respective primary keys of K1 = {A} and K2 = {E} where A, B,..E, F are attributes. A distributed database system stores the relations x(R1) and y(R2). The system also contains the following fragments:

$$u = \sigma_{C='c'}(x) \text{ and } w = \pi_{B,C}(x)$$

Also, the following statistics are available:

$n_x = 3000$, $n_y = 2000$, each attribute is of 10 bytes fixed length, $V(C,x) = V(C,y) = 10$ and $V(B,x) = V(B,y) = 12$.

Also, assume the following distribution of relations and fragments:

i) site2 stores y and u ii) site3 stores w iii) site4 stores x and w

The following query has originated at site1:

$$\pi_{D,F}(\sigma_{C='c'}(x \otimes y))$$

Since site1 does not store any of the relations involved, the site1 query processor must determine a strategy to execute the query.

Devise a strategy which attempts to minimize the size of any temporary relations as well as the number of bytes sent through the network **and** estimate the number of bytes actually sent to site1. You may assume uniform distribution of values.

Exercise 9 (9 points):

Consider the following schedule. Only the pertinent read (RD) and write (WR) instructions are shown with the target data item in parenthesis.

Time	T1	T2	T3	T4
1		RD(A)		
2	RD(A)			
3	WR(C)			
4			RD(C)	
5	WR(B)			
6				RD(B)
7			WR(A)	
8				RD(C)
9		WR(D)		
10		RD(B)		
11				WR(A)
12				WR(B)

a) There are several algorithms to build precedence graphs. Give a brief description of an algorithm that would build the precedence graph.

b) Use the algorithm you described in part a) to build the appropriate precedence graph for the schedule.

c) By reference to the constructed graph, determine whether the schedule is serializable (**justify your decision**) and if so, give its equivalent serial schedule.

Exercise 10 (9 points):

Consider the following function transfer() which transfers money from one account to another:

```
1) EXEC SQL BEGIN DECLARE SECTION;
2)      int acct1, acct2; /* the two accounts */
3)      int balance1; /* amount of money in the first account */
4)      int amount; /* amount of money to transfer */
5) EXEC SQL END DECLARE SECTION;

6) void transfer() {
7)      /* Code to prompt user to enter acct1, acct2 and amount. */
8)      EXEC SQL SELECT balance INTO :balance1
9)              FROM Accounts
10)             WHERE acctNo = :acct1;
11)     if (balance1 >= amount) {
12)             EXEC SQL UPDATE Accounts
13)                     SET balance = balance + :amount
14)                     WHERE acctNo = :acct2;
15)             EXEC SQL UPDATE Accounts
16)                     SET balance = balance - :amount
17)                     WHERE acctNo = :acct1;
        }
18)     else /* Code informing user of insufficient funds. */
    }
```

Suppose that the initial balance of account 1 is \$100 and the initial balance of account 2 is \$200. Transaction T_1 invokes transfer() to transfer \$150 from account 2 to 1; transaction T_2 invokes transfer() to transfer \$150 from account 1 to 2.
(Unlikely? Well, think of it as unimaginative friends each wanting to give each other a holiday gift!)

a) What are the final balances of accounts 1 and 2 if T_1 is executed before T_2 ?

b) What are the final balances of accounts 1 and 2 if T_2 is executed before T_1?

c) Now suppose that each transaction specified

SET TRANSACTION READ WRITE
ISOLATION LEVEL READ UNCOMMITTED

Describe a situation in which the resulting execution would fail to be serializable