The assignment is to be turned in before Midnight (by 11:59pm) on February 17th , 2015. Late submissions are accepted, but there is a 5 point deduction for each day the assignment is late. You should turn in the solutions to this assignment as a pdf file through the TEACH website. The solutions should be produced using editing software programs, such as LaTeX or Word, otherwise they will not be graded.

1: Indexing (1 point)

Using B+ trees with degrees of two (2) whose keys are integer values, give an example of a B+ tree whose height changes from 2 to 3 when the value 25 is inserted. Show your structure before and after the insertion. A B+ tree with a single node has height of 1.

Solution: One solution is a B+ tree has a root and five leaf nodes. The root node contains four keys of [10, 20, 30, 40]. The five leaf nodes, from left to right, contain [2,6] , [10,13,16,17], [20,21,23,28], [31,32,36,38], and [43,54,69,87]. After inserting 25, the resulting B+ tree will have one root node, two internal nodes, and six leaf nodes. The root node contains a single key 23. The internal nodes, from left to right, contain [10,20] and [30,40]. The leaf nodes, from left to right, contain keys [2,6], [10,13,16,17], [20,21], [23,25,28], [31,32,36,38], and [43,54,69,87].

2: Query processing (2 points)

Consider the join of the relation R and S on attributes R.a=S.b, given the following information about the relations to be joined. Relation R contains 10,000 tuples and has 10 tuples per block. Relation S contains 2000 tuples and also has 10 tuples per block. Attribute b of relation S is the primary key for S. Neither relation has any indexes built on it. 52 buffer pages are available in main memory. What is the cost of joining R and S using a sort-merge join? You should use a version of sort-merge algorithm that provides the minimum cost. The cost metric is the number of block I/Os.

Solution: We have B(R) = 1000 and B(S) = 200. As we have B(R) + B(S) ¡= M^2, we may use the optimized sort-merge join. The cost of the join will be 3 B(R) + 3 B(S) = 3600 number of I/O access.

3: Query optimization (1 points)

Consider the following relational schema and SQL query:

Suppliers(sid, sname, city)
Supply(sid, pid)
Parts(pid, pname, price)

SELECT S.sname, P.pname
FROM Suppliers S, Parts P, Supply Y
WHERE S.sid = Y.sid AND Y.pid = P.pid

How many different join orders, assuming that cross-products are disallowed, does a System R style query optimizer consider when deciding how to process the given query? List the join orders.

Solution: Only left-deep plans are allowed: $((S \bowtie Y) \bowtie P)$ and $((Y \bowtie P) \bowtie S)$.

4: Concurrency control (2 points)

Consider the schedule shown at Table 1.

	T1	T2	T3
0		start	
1		read X	
2	start		
3	read Y		
4		write X	
5			start
6			read X
7			write X
8			Commit
9	read X		
10	write Y		
11	write X		
12	Commit		
13		read Y	
14		write Y	
15		Commit	

Table 1: Transaction schedule

(a) What is the equivalent serialization order for this schedule? If no order is possible, you may state so.

Solution: Because the serialization graph contains both edges T1 → T2 and T2 → T1, this schedule is **not** serializable. Thus, it is not equivalent to any serializable schedule.

(b) When all transactions run in the above schedule, identify the transactions with degree 3 consistency. Answer the same question when transaction T3 does not exist in the schedule.

Solution: Based on the degree of consistency each transaction chooses, other transactions may or may be in degree three consistency. The following is one reasonable scenario. Any other reasonable scenarios is acceptable for solution:

T2 is not in degree 3 consistency as it reads a data item (X), which is dirtied by T1 and T3 before T2 completes. If T2 is in a degree more than or equal to 1, i.e. has long locks on writes, T1 and T3 will not be in degree 3 consistency as they dirty T2s data, i.e. they cannot get a long write lock on X.

5: Recovery (4 points)

In this problem, you need to simulate the actions taken by ARIES. Consider the following log records and buffer actions:

(a) For the actions listed above, show Transaction Table (XT) and Dirty Page Table (DPT) after each action. Assume that DPT holds pageID and recLSN, and XT contains transID and lastLSN.

Solution: Let XT denote xact table and DPT denote dirty page table.

time	LSN	Log	Buffer actions
0	00	update: T1 updates P7	P7 brought in
1	10	update: T0 updates P9	P9 brought in; P9 kicked out
2	20	update: T1 updates P8	P8 brought in; P8 kicked out
3	30	begin_checkpoint	
4	40	end_checkpoint	
5	50	update: T1 updates P9	P9 brought in
6	60	update: T2 updates P6	P6 brought in
7	70	update: T1 updates P5	P5 brought in
8	80	update: T1 updates P7	P6 kicked out
9		CRASH RESTART	

After time 0:

XT

transaction	lastLSN	status
T1	00	active

DPT

page	recLSN
P7	00

After time 1:

XT

transaction	lastLSN	status
T1	00	active
T0	10	active

DPT

page	recLSN
P7	00

After time 2:

XT

transaction	lastLSN	status
T1	20	active
T0	10	active

DPT

page	recLSN
P7	00

After time 3: same as above.

After time 4: same as above.

After time 5:

XT

transaction	lastLSN	status
T1	50	active
T0	10	active

DPT

page	recLSN
P7	00
P9	50

After time 6:

XT

transaction	lastLSN	status
T1	50	active
T0	10	active
T2	60	active

DPT

page	recLSN
P7	00
P9	50
P6	60

After time 7:

XT

transaction	lastLSN	status
T1	70	active
T0	10	active
T2	60	active

DPT

page	recLSN
P7	00
P9	50
P6	60
P5	70

After time 8:

XT

transaction	lastLSN	status
T1	80	active
T0	10	active
T2	60	active

DPT

page	recLSN
P7	00
P9	50
P5	70

Note: P6 is written out with pageLSN = 80.

(b) Simulate Analysis phase to reconstruct XT and DPT after crash. Identify the point where the Analysis phase starts scanning log records and show XT and DPT after each action.

Analysis phase begins by examining the most recent checkpoint and initializing the XP and DPT at the time of the checkpoint, i.e., after time 2 in solution to part (a).

After time 5:

XT

transaction	lastLSN	status
T1	50	active
T0	10	active

DPT

page	recLSN
P7	00
P9	50

After time 6:

XT

transaction	lastLSN	status
T1	50	active
T0	10	active
T2	60	active

DPT

page	recLSN
P7	00
P9	50
P6	60

After time 7:

XT

transaction	lastLSN	status
T1	70	active
T0	10	active
T2	60	active

DPT

page	recLSN
P7	00
P9	50
P6	60
P5	70

After time 8:

XT

transaction	lastLSN	status
T1	80	active
T0	10	active
T2	60	active

DPT

page	recLSN
P7	00
P9	50
P6	60
P5	70

(c) Simulate Redo phase: first identify where the Redo phase starts scanning the log records. Then, for each action identify whether it needs to be redone or not.

Solution: Redo starts from the smallest recLSN in DPT at the end of Analysis, i.e., LSN=00. Table 2 shows whether each action needs to be redone and the reason.

(d) Simulate Undo phase: identify all actions that need to be undone. In what order will they be undone?

Solution: As no transaction committed, all actions will be undone in the decreasing order of LSN. That is UNDO 80, 70, 60, 50, 20, 10, and 00.

h!

LSN	Redone?	Why Not?
00	Yes	
10	No	affected page in DPT but recLSN is greater than 10
20	No	affected page not in DPT
30	No	checkpoint
40	No	checkpoint
50	Yes	
60	No	pageLSN 80 is greater than 60
70	Yes	
80	Yes	

Table 2: Redo operations