NATIONAL UNIVERSITY OF SINGAPORE

CS3223 - DATABASE SYSTEMS IMPLEMENTATION

(Semester 2: AY2015/16)

Time Allowed: 2 Hours

INSTRUCTIONS TO STUDENTS

1. Please write your Student Number only. Do not write your name.
2. This assessment paper contains **SEVEN (7)** questions and comprises **EIGHTEEN (18)** printed pages, including this page.
3. Answer **ALL** questions within the space in this booklet.
4. This is a **CLOSED BOOK** assessment.
5. You are allowed to use an electronic calculator for this assessment.
6. You are allowed to refer to a single, hand-written, A4-sized sheet of notes.

STUDENT NO:

EXAMINER'S USE ONLY

Question:	1	2	3	4	5	6	7	Total
Points:	8	5	4	4	3	10	12	46
Score:								

1. (8 points) Consider the following two schedules, S_1 and S_2, over the same set of transactions $\{T_1, T_2, T_3, T_4\}$:

	Schedule S_1				**Schedule S_2**			
Step	T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4
1	$W_1(a)$				$W_1(a)$			
2				$R_4(a)$	$W_1(c)$			
3	$W_1(c)$							$R_4(a)$
4			$W_3(d)$					$W_4(b)$
5				$W_4(b)$				$W_4(a)$
6				$W_4(a)$			$W_3(d)$	
7			$W_3(b)$				$W_3(b)$	
8		$R_2(a)$			$R_1(b)$			
9		$R_2(d)$				$R_2(a)$		
10				$W_4(c)$		$R_2(d)$		
11	$R_1(b)$						$W_3(c)$	
12			$W_3(c)$			$R_2(c)$		
13			$Commit_3$					$W_4(c)$
14	$R_1(c)$				$R_1(c)$			
15				$Commit_4$		$W_2(c)$		
16		$R_2(c)$					$Commit_3$	
17		$W_2(c)$			$W_1(d)$			
18		$Commit_2$				$Commit_2$		
19	$W_1(d)$				$Commit_1$			
20	$Commit_1$							$Commit_4$

(a) (2 points) State whether S_1 and S_2 are view equivalent schedules.

(b) (2 points) State whether S_1 and S_2 are conflict equivalent schedules.

(c) (1 point) State whether S_1 is a view serializable schedule. If your answer is "Yes", write down a serial schedule that is view equivalent to S_1.

QUESTION 1 CONTINUES ON THE NEXT PAGE

CONTINUATION OF QUESTION 1

The schedules S_1 & S_2 are repeated on this page for convenience.

Schedule S_1

Step	T_1	T_2	T_3	T_4
1	$W_1(a)$			
2				$R_4(a)$
3	$W_1(c)$			
4			$W_3(d)$	
5				$W_4(b)$
6				$W_4(a)$
7			$W_3(b)$	
8		$R_2(a)$		
9		$R_2(d)$		
10				$W_4(c)$
11	$R_1(b)$			
12			$W_3(c)$	
13			$Commit_3$	
14	$R_1(c)$			
15				$Commit_4$
16		$R_2(c)$		
17		$W_2(c)$		
18		$Commit_2$		
19	$W_1(d)$			
20	$Commit_1$			

Schedule S_2

T_1	T_2	T_3	T_4
$W_1(a)$			
$W_1(c)$			
			$R_4(a)$
			$W_4(b)$
			$W_4(a)$
		$W_3(d)$	
		$W_3(b)$	
$R_1(b)$			
	$R_2(a)$		
	$R_2(d)$		
		$W_3(c)$	
	$R_2(c)$		
			$W_4(c)$
$R_1(c)$			
	$W_2(c)$		
		$Commit_3$	
$W_1(d)$			
	$Commit_2$		
$Commit_1$			
			$Commit_4$

(d) (1 point) State whether S_1 is a conflict serializable schedule. If your answer is "Yes", write down a serial schedule that is conflict equivalent to S_1.

(e) (1 point) State whether S_2 is a view serializable schedule. If your answer is "Yes", write down a serial schedule that is view equivalent to S_2.

(f) (1 point) State whether S_2 is a conflict serializable schedule. If your answer is "Yes", write down a serial schedule that is conflict equivalent to S_2.

3. (4 points) Consider a course registration application with a database that consists of two relations, `Student` (sid, numCourses) and `Enroll` (sid, cid), where

   - `sid` represents a unique student identifier;
   - `cid` represents a unique course identifier;
   - each record `(sid,cid)` in `Enroll` indicates that student `sid` has enrolled for course `cid`; and
   - each record `(sid, numCourses)` in `Student` indicates that student `sid` has enrolled a total of `numCourses` courses.

   The database has the following two consistency requirements:

   1. each student can enroll at most once for each course, and
   2. the total number of courses enrolled by each student is at most 20.

   The application program, `RegisterCourse`, is used to register a student with sid S for a course with cid C.

   RegisterCourse (S, C)

```
BEGIN TRANSACTION;
enrolled = 0;
SELECT numCourses FROM Student INTO :num WHERE sid = :S ;
SELECT 1 FROM Enroll INTO :enrolled WHERE sid = :S AND cid = :C ;
if (enrolled != 1) and (num < 20) then
        UPDATE Student SET numCourses = numCourses + 1 WHERE sid = :S ;
        INSERT INTO Enroll VALUES (:S, :C);
endif
COMMIT;
```

   Assume the following for this question:

   1. The application is running on a database system using a lock-based concurrency control protocol with record-level locking and deadlock detection.
   2. Multiple instances of the application could be executing concurrently.
   3. Short duration locks are released right after the completion of the associated operations.

   Consider a schedule H consisting of an instance of the program execution. The following are three possibilities for H, where e denote the Enroll record for S and C, s denote the Student record for S, and $R(\varnothing)$ denote that the execution of the SELECT statement on Enroll relation returns an empty result without locking any record.

   1. If S has already enrolled in C, then $H = R(s), R(e)$.
   2. If S has not yet enrolled in C but S has already enrolled in 20 courses, then $H = R(s), R(\varnothing)$.
   3. If S has not yet enrolled in C and S has enrolled in fewer than 20 courses, then $H = R(s), R(\varnothing), W(s), W(e)$.

QUESTION 3 CONTINUES ON THE NEXT PAGE

CONTINUATION OF QUESTION 3

(a) (2 points) If the application is running with the READ-COMMITTED isolation level and lock upgrades are not used, state whether all executions are conflict serializable. If your answer is "No", show an example READ-COMMITTED schedule that is not serializable.

(b) (2 points) If the application is running with the REPEATABLE-READ isolation level and lock upgrades are used when necessary, state whether all executions are conflict serializable. If your answer is "No", show an example REPEATABLE-READ schedule that is not serializable.

4. (4 points) This question considers two multiversion schedules, S_1 & S_2, over two different sets of transactions.

   (a) (2 points) State whether S_1 is a multiversion view serializable schedule. If your answer is "Yes", write down a serial schedule that is multiversion view equivalent to S_1.

Schedule S_1

Step	T_1	T_2	T_3	T_4	T_5
1			$R_3(b_0)$		
2		$R_2(b_0)$			
3			$R_3(a_0)$		
4		$W_2(b_2)$			
5		$W_2(a_2)$			
6			$W_3(a_3)$		
7					$R_5(b_2)$
8				$R_4(a_2)$	
9	$R_1(a_3)$				
10	$W_1(c_1)$				
11	$W_1(d_1)$				
12				$W_4(b_4)$	
13					$R_5(d_1)$
14				$W_4(d_4)$	
15					$W_5(b_5)$

QUESTION 4 CONTINUES ON THE NEXT PAGE

CONTINUATION OF QUESTION 4

(b) (2 points) State whether S_2 is a multiversion view serializable schedule. If your answer is "Yes", write down a serial schedule that is multiversion view equivalent to S_2.

Schedule S_2

Step	T_1	T_2	T_3	T_4	T_5
1			$R_3(b_0)$		
2		$W_2(d_2)$			
3		$W_2(a_2)$			
4				$R_4(a_2)$	
5				$W_4(a_4)$	
6		$W_2(c_2)$			
7			$R_3(a_4)$		
8			$W_3(a_3)$		
9				$W_4(c_4)$	
10					$R_5(c_2)$
11	$R_1(c_2)$				
12	$W_1(c_1)$				
13					$W_5(b_5)$
14	$W_1(d_1)$				

6. (10 points) Consider a relation R with $||R|| = 200$ and an attribute A with $||\pi_A(R)|| = 20$. The actual distribution of attribute A is shown below.

Value of A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
# of tuples	7	12	9	3	1	30	0	16	3	2	8	25	0	3	40	5	15	2	4	15

This question consists of four parts.

(a) (3 points) Construct an equidepth histogram H with 4 buckets. For each bucket, indicate the (1) range of attribute values and (2) the number of tuples.

(b) (2 points) Use H to estimate the number of tuples in the result of the query: "SELECT * FROM R WHERE A ≥ 5 AND A ≤ 12".

QUESTION 6 CONTINUES ON THE NEXT PAGE

CONTINUATION OF QUESTION 6

The actual distribution of attribute A is repeated on this page for convenience.

Value of A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
# of tuples	7	12	9	3	1	30	0	16	3	2	8	25	0	3	40	5	15	2	4	15

(c) (3 points) Construct an equidepth histogram H' with 3 buckets and top-3 MCV. For each bucket, indicate the (1) range of attribute values and (2) the number of tuples.

(d) (2 points) Use H' to estimate the number of tuples in the result of the query: "SELECT * FROM R WHERE A ≥ 5 AND A ≤ 12".

7. (12 points) Consider a database with the following two relations where the key attributes are shown underlined:

   - S (<u>e</u>, f, g)
   - T (<u>h</u>, j, k)

   Assume the following for this question.

   1. Relation S contains 800 pages with each page containing 50 records.
   2. Relation T contains 600 pages with each page containing 50 records.
   3. Relation S has an unclustered B$^+$-tree index I_g on (g) with at most 60 entries in each leaf page.
   4. Relation T has an unclustered B$^+$-tree index I_j on (j) with at most 60 entries in each leaf page.
   5. Each of the two indexes has two levels of internal nodes.
   6. $||\pi_g(S)|| = 200$.
   7. $||\pi_k(T)|| = 150$, $\pi_j(T) = \{1, 2, 3, 4, 5, 6\}$.
   8. There are 25 buffer pages available.
   9. The database system materializes intermediate results whenever it is beneficial.
   10. The cost metric to use is the number of page I/Os. Ignore the cost of writing out the final result.

   Consider the following two queries.

```
Q1: SELECT  *                Q2: SELECT  *
    FROM    T                    FROM    S, T
    WHERE   T.j ≥ 2              WHERE   S.g = T.k
    AND     T.j ≤ 4              AND     T.j ≥ 2
                                 AND     T.j ≤ 4
```

   Answer the following four parts:

   (a) (3 points) Estimate the number of tuples in the result of Q_2.

   (b) (3 points) What is the cost of evaluating Q_2 using the optimized sort merge join algorithm?

   (c) (3 points) Assume that the database system supports only two join algorithms: block nested-loop join and index nested-loop join. What is the least cost query plan for Q_2? What is its cost?

   (d) (3 points) Assume that the database system supports only two join algorithms: block nested-loop join and index nested-loop join. If there are exactly 100 tuples in the result of Q_1, and each of the result tuples in Q_1 joins with 2 tuples in S (based on $T.k = S.g$), what is the least cost query plan for Q_2? What is its cost?

   For this question, show your working and state any assumptions (uniformity, independence, inclusion) used.

PAGE LEFT BLANK FOR ANSWER TO QUESTION 7(a)

PAGE LEFT BLANK FOR ANSWER TO QUESTION 7(b)

PAGE LEFT BLANK FOR ANSWER TO QUESTION 7(c)

PAGE LEFT BLANK FOR ANSWER TO QUESTION 7(d)

END OF PAPER