NATIONAL UNIVERSITY OF SINGAPORE

CS3223 – DATABASE SYSTEMS IMPLEMENTATION
(Semester 2: AY2018/19)

Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This assessment paper consists of **TWO** Sections and comprises **THIRTEEN (13)** printed pages. Section **A** has **FOUR (4)** questions, and Section **B** has **SIXTEEN (16)** questions. The maximum score is **FORTY** (40).
2. Answer **ALL** questions. Write your answers within the **space provided** in this booklet.
3. This is an **OPEN BOOK** assessment.
4. You are allowed to use an authorized calculator.
5. Please write your **Student Number** below. Do not write your name.

Student Number:												

EXAMINER'S USE ONLY	
Questions	**Marks**
Section A (24 marks)	
Q1 (9 marks)	
Q2 (5 marks)	
Q3 (5 marks)	
Q4 (5 marks)	
Section B (16 marks)	
Total (40 marks)	

Section A (24 marks)

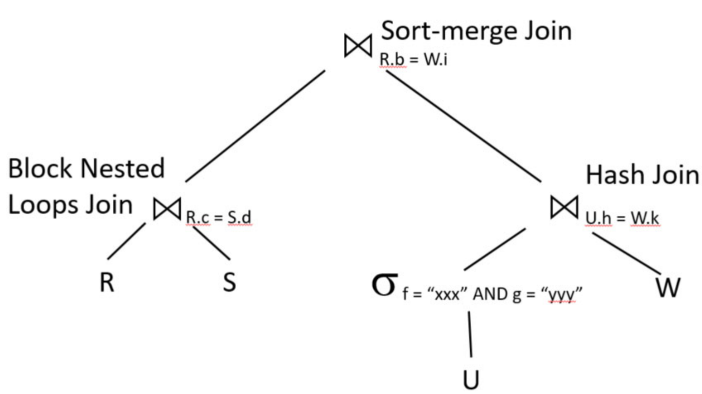
Question 1. (9 marks) Consider the following four relations and the given statistics:

Relation	**Number of tuples**	**Number of pages**	**Unique Values**
R(a, b, c)	1,000	100	
S(d, e)	1,000	100	
U(f, g, h)	100,000	10,000	V(U, f) = 100 V(U, g) = 100 V(U, h) = 10,000
W(i, j, k, l)	10,000	1,000	V(W, k) = 1,000

Histogram of R.c (integer values from 1-500)				
Range	1-100	101-250	251-300	301-500
Number of tuples	200	450	150	200

Histogram of S.d (integer values from 1-500)					
Range	1-150	151-200	201-300	301-499	500
Number of tuples	300	250	200	199	51

We want to compute the cost of the query plan shown below assuming 300 pages of memory. Count only the number of page I/Os.



Follow the various parts listed below. As you compute the cost of the plan in a bottom up fashion, at the output of each operator, consider whether the result can fit in memory and be used directly by the next operator or if you need to write anything to disk. You should also process the plan in an optimal way to minimize the total I/O cost. Consider whether you need one-pass or two-pass algorithms. Moreover, for simplicity, we assume that all attributes should be output.

(A) **(1 mark)** What is the I/O cost for the subtree involving the join between R and S (without the output cost)?

Ans: since memory size is enough, we need to read each table once = R+S = 200

(B) **(2 marks)** What is the size of the join output between R and S (both number of tuples and pages)?

Ans:
R JOIN S = 100*(2*2) + 50*(3*2) + 50*(3*5) + 50*(3*2) + 50*(3*2) + 199*(1*1) + 1*(1*51)
= 400 + 300 + 750 + 300 + 300 + 199 + 51 = 2300

#Pages = 460 (5 tuples per page)

Common mistakes: did not use the distribution table given.

(C) **(1 mark)** What is size of the selection condition on U (in tuples and pages)?

Ans: 100,000/(100*100) = 10 tuples, and 1 page.

(D) **(2 marks)** What is the I/O cost for the subtree involving the join between U and W (without the output cost)? You can assume all memory is available for this operation.

Ans: each tuple of U matches 0 or 10 tuples of W. So,the answer falls in the range of 0 to 100 tuples (10*10)
For simplicity, consider the extreme case of 100 tuples, we have 20 pages.

I/O cost = read U (and perform selection on the fly and build hash table in memory) + read W
= 10,000 + 1,000 = 11,000

Common mistakes: adopt GRACE hash join and did not consider cost to read U

(E) **(3 marks)** What is the (optimized) I/O cost of the query plan? There is no need to consider the output of the plan (you can assume the results are returned to the users as they are produced).

Ans: Part A + output + sort(output) + part (D) + read(sorted output)

Question 2. (5 marks) Suppose we run the following 5 transactions using the validation-based protocol. There are no other transactions in the system. Suppose the database has only the set of given 7 objects: a, b, c, d, e, f, g. For transaction T_5, the values XXX and YYY represent unknown values. The following table lists the transactions involved, together with their read and write sets:

Transaction	Read Set	Write Set
T_1	{a, b}	{c}
T_2	{a, b}	{c, d}
T_3	{d}	{e}
T_4	{f}	{e, g}
T_5	{XXX}	{YYY}

Consider the following schedule of events. For transaction with id i, S_i stands for start, V_i for validation and F_i for finish.

H = S_1; S_2; V_1; F_1; S_3; V_2; S_4; S_5; V_3; V_4; F_2; F_3; V_5; F_4; F_5

(A) **(1 mark)** Does T_2 validate successfully? Explain your answer.

Ans: Yes – since RS(T2) intersect WS(T1) = emptyset

(B) **(1 mark)** Does T_3 validate successfully? Explain your answer.

Ans: No – since RS(T3) intersect WS(T2) = {d}

(C) **(2 marks)** Suppose transaction T_5 validates successfully. What are the largest possible sets of values of XXX and YYY for this to happen?

Ans:
T4 validates successfully, so must compare with T2 and T4

XXX = A, B, F

YYY = A, B, C, D, F

(D) **(1 mark)** Suppose transaction T_5 does not validate successfully. What is the smallest possible write set of value YYY for this to happen?

Ans: YYY = E or G;

I also accept EMPTY set if one assumes conflict at XXX

Question 3. (5 marks) Consider the following transaction log from the start of the run of a database system that is capable of running undo/redo logging with non-quiesce checkpointing. Note that the log record is enclosed by < and >. The number besides each log record is just an identifier to refer to the log record; it is shown separately for clarity.

1 <T1 BEGIN>
2 <T1, X, 2, 1>
3 <T2 BEGIN>
4 <T3 BEGIN>
5 <T2, Y, 2, 1>
6 <T2 COMMIT>
7 <T1, Y, 3, 2>
8 <T3, Z, 2, 1>
9 <Start-CKPT T1, T3>
10 <T1, X, 3, 2>
11 <T1, Y, 4, 3>
12 <End-CKPT>
13 <T3, Z, 3, 2>
14 <T3 COMMIT>
15 <T1, Z, 4, 3>

Assume the log entries are in the format < Tid, Variable, newValue, oldValue >. Further, assume that all the oldValue of objects X, Y, Z are initially on disk.

(A) **(3 marks)** What are the values of X, Y, and Z in the database stored on disk before the DBMS recovers the state of the database? If any of the values cannot be determined, indicate with UNKNOWN and explain why.

X =   Y =   Z =

Ans: All UNKNOWN. The checkpoint flushed everything to disk, but then all the data objects were modified by transactions after the checkpoint. Since we are using NO-FORCE, any dirty page could be written to disk, so therefore we don't know the contents of the database on disk at the time of the crash.

(B) (1 mark) Which of the following statements is TRUE when recovering the database? Write your answer in here – [d]

a. Undo ALL of T1's changes
b. T2 do nothing
c. Redo ALL of T3's changes
d. (a) and (b) only
e. (a) and (c) only
f. (b) and (c) only
g. All of the above
h. None of the above

(C) **(1 mark)** Assume that the DBMS flushes all dirty pages when the recovery process finishes. What are the values of X and Y after the DBMS recovers the state of the database? Fill in your answer here – [b]

a. X = 1, Y = 1
b. X = 1, Y = 2
c. X = 1, Y = 3
d. X = 1, Y = 4
e. X = 2, Y = 1
f. X = 2, Y = 2
g. X = 2, Y = 3
h. None of the above

Question 4 (5 marks) We want to extend the multi-granularity locking scheme to incorporate UPDATE locks. As such, we add "intention update" lock, IU, and "update" lock, U. An IU lock indicates the intention of a transaction to acquire an update lock U at a lower level of granularity. Complete the compatibility table below. For simplicity, we have excluded SIX lock mode. If a cell can be both T and F, then fill in as T/F.

	Requestor					
Holder	**IS**	**IX**	**IU**	**S**	**U**	**X**
IS	T	T	T	T	TT	F
IX	T	T	T	F	FF	F
IU	T	T	TT	T/F	FF	F
S	T	F	TT	T	T	F
U	T/F	FF	FF	F	F	F
X	F	F	F	F	F	F

Section B (16 marks)

This section contains 16 multiple choice questions. Pick the **BEST** answer for each question. Fill in your answers in the following table.

Q1	G		Q9	A/F/G
Q2	D		Q10	E
Q3	A		Q11	D
Q4	B		Q12	D
Q5	E/G		Q13	F
Q6	G		Q14	D
Q7	E		Q15	F
Q8	C		Q16	A

1. Consider the following schedule:

   $S = R_1(B), W_3(A), R_2(C), W_2(C), R_3(C), W_1(B), R_3(B), W_3(C)$

   (A) S is non-serializable
   (B) S is conflict equivalent to T_3,T_2,T_1
   (C) S is conflict equivalent to T_2,T_1,T_3
   (D) S is conflict equivalent to T_1,T_2,T_3
   (E) (B) and (C) are both correct
   (F) (B) and (D) are both correct
   (G) (C) and (D) are both correct
   (H) (B), (C) and (D) are all correct

2. Consider the following schedule:

   $S = R_1(A), W_1(A), R_2(B), W_2(C), R_2(A), R_3(C), W_3(B), R_1(B)$

   (A) S is serializable
   (B) S can be produced by a 2PL scheduler (2PL locking protocol)
   (C) (A) and (B) are both correct
   (D) (A) and (B) are both wrong

3. Consider the following schedule:

   $S = W_2(X), W_2(Y), R_1(X), W_1(Y), W_2(Z), C_2, C_1, R_3(Y), W_3(Z)\ C_3$

   Which of the following statements is TRUE?

   (A) S is recoverable
   (B) S avoids cascading abort
   (C) S is strict
   (D) S is serial
   (E) None of the above.

4. Consider the validation-based protocol. Suppose we have a schedule that shows T1 starts, validates and finishes before all other schedules, e.g.

   T1 starts …(A)... T1 validates ….(B)… T1 finishes ….(C) …

   Where region A consists of other transactions that may have started but have not validated or finished, and region B consists of other transactions that may have started and validated but have not finished, and region C consists of other transactions that may have started, validated and finished. Which of the following statements is TRUE (no assumptions required)?
   (A) Since T1 starts first, it will definitely succeed.
   (B) Since T1 validates first, it will definitely succeed.
   (C) Since T1 finishes first, it will definitely succeed.
   (D) All of the above
   (E) None of the above

5. Consider the multi-granularity locking system. Suppose we have a hierarchy of 3 levels: relation (R), pages (P_1 and P_2) and tuples (t_{11} and t_{12} in P_1, and t_{21} and t_{22} in P_2). Which of the following sequence of locks is valid, non-redundant and correct?

   (A) To read all records in P_2, we have IS(R), S(P_2)
   (B) To modify t_{22}, we have IX(R), IX(P_2), X(t_{22})
   (C) To insert into P_2, we have SIX(R), X(P_2)
   (D) To read all records, but modify t_{22}, we have SIX(R), SIX(P_2), X(t_{22})
   (E) (A) and (B)
   (F) (A), (B) and (C)
   (G) (A), (B) and (D)
   (H) All of the above
   (I) None of the above

6. Let δ denote duplicate elimination. Which of the following statements is TRUE?
   (A) $\delta(R \times S) = \delta(R) \times \delta(S)$
   (B) $\delta(R \text{ JOIN } S) = \delta(R) \text{ JOIN } \delta(S)$
   (C) $\delta(\sigma_s(R)) = \sigma_s(\delta(R))$
   (D) (A) and (B) are both correct
   (E) (A) and (C) are both correct
   (F) (B) and (C) are both correct
   (G) (A), (B) and (C) are all correct

7. In query optimization, which of the following is TRUE?
   (A) The size of a table (in number of tuples) can affect the plan space (e.g., more tuples, larger plan space).
   (B) The number of tables can affect the plan cost (e.g., more tables, the cost will be different).
   (C) Given two possible plans A and B, given all parameters are fixed, manipulating the size of the tables (number of tuples/pages) can change the ranking of the two plans.
   (D) (A) and (B) only
   (E) (B) and (C) only
   (F) (A) and (C) only
   (G) All of the above
   (H) None of the above

8. Consider the greedy paradigm for query optimization. These algorithms typically do not produce good plans, compared to say Dynamic Programming or Randomized algorithms, because (pick the most relevant answer):
   (A) They typically restrict to left-deep trees.
   (B) They build the plans bottom up, and generate only one final plan.
   (C) They search a small space.
   (D) They employ cost models that are error prone.
   (E) They consider cross products.

9. Consider the randomized query optimization algorithm (Local Optimization) that we cover in the lecture. For each "outer" iteration, we need to generate a start state. Let N be the number of start states. For the "inner" loop, we move from one state to another state. Which of the following is TRUE?
   (A) N can be tuned to tradeoff quality of plans and the optimization overhead.
   (B) There will always be N unique local minima, and the one with the minimum cost is the selected plan.
   (C) The inner loop runs for a number of iterations. Given N start states, let the number of iterations be n1 for iteration 1, n2 for iteration 2, and so on, where nN is the number of iterations for the last start state. n1 = n2 = n3 = … = nN.
   (D) In the inner loop, all moves are downward moves.
   (E) (A) and (B)
   (F) (A) and (C)
   (G) (A) and (D)
   (H) (A), (B) and (C)
   (I) (A), (B) and (D)

10. Which of the following is TRUE?
    (A) Under the NO-STEAL policy, a DBMS will need to store the whole table in RAM if a transaction updates all the records of that table.
    (B) STEAL + NO-FORCE policy has the fastest recovery performance.
    (C) Under NO-STEAL + FORCE policy, a DBMS will never need to redo the changes of a committed transaction during recovery.
    (D) (A) and (B)
    (E) (A) and (C)
    (F) (B) and (C)
    (G) (A), (B) and (C)

11. Consider the following four transactions. To prevent deadlocks, the wait-die scheme is used. Suppose these transactions execute in round-robin fashion. When it is a transaction's turn, it executes its next lock or unlock step if it can, and otherwise it takes the appropriate action under the wait-die scheme. Assume that each transaction is given its deadlock timestamp the first time it executes a step. Determine the sequence of steps that the four transactions make, including all die/rollback steps (which take place in a single round of the round-robin) and the restarts of transactions.

    T_1: $L_1(A)$; $L_1(B)$; $U_1(A)$; $U_1(B)$
    T_2: $L_2(A)$; $L_2(C)$; $U_2(A)$; $U_2(C)$
    T_3: $L_3(B)$; $L_3(C)$; $U_3(B)$; $U_3(C)$
    T_4: $L_4(A)$; $L_4(D)$; $U_4(A)$; $U_4(D)$

    (A) T_1 never aborts
    (B) T_2 never waits
    (C) T_3 never dies
    (D) All of the above
    (E) None of the above

12. Referring to Question 11. What is the final ordering of the transactions?
    (A) T_1; T_2; T_3; T_4
    (B) T_1; T_3; T_2; T_4
    (C) T_2; T_3; T_1; T_4
    (D) T_3; T_1; T_2; T_4
    (E) T_4; T_3; T_2; T_1
    (F) T_4; T_2; T_1; T_3
    (G) None of the above

13. Which of the following is TRUE of non-quiesce checkpointing scheme in Recovery Manager?
    (A) It improves performance during recovery as it only needs to examine logs from the most recent checkpoint start onwards.
    (B) It minimizes the amount of storage needed as all older logs prior to the most recent successful checkpoint can be deleted.
    (C) It should be done only once a week as it incurs overhead.
    (D) (A) and (B)
    (E) All of the above
    (F) None of the above

14. Which of the following is TRUE of query optimization?
    (A) Even if statistics are accurate, the estimation of intermediate result sizes may still be inaccurate.
    (B) Even if we do not have a cost model, we can still generate a query plan (although it may not be optimal).
    (C) Even if we do not have an enumeration algorithm, we can still generate a query plan (although it may not be optimal).
    (D) All of the above
    (E) None of the above

15. Which of the following is TRUE of dynamic programming scheme for query optimization (that we studied in the lecture)?
    (A) The principle of optimality is used to prune away plans when considering the same subsets of tables.
    (B) The scheme may not generate globally optimal solution within the left-deep space.
    (C) The scheme minimizes redundant computation by maintaining the best plan for each subset of tables.
    (D) If we include cross-products into the plan space, the scheme will be able to generate the globally optimal solution within the left-deep space.
    (E) (A) and (B)
    (F) (A), (B) and (C)
    (G) (A), (C) and (D)
    (H) All of the above

16. Which of the following is TRUE of Strict 2PL?
    (A) Locks cannot be released until the end of the transaction.
    (B) Schedules under Strict 2PL allows dirty reads.
    (C) Schedules under Strict 2PL may lead to cascading aborts.
    (D) Only schedules under 2PL (and not Strict 2PL) may lead to deadlocks.
    (E) (A) and (B)
    (F) (A) and (C)
    (G) (A) and (D)
    (H) (A), (B) and (C)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ End of Paper ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~