## Section A: Honor Statement

**1.**

BY TAKING THIS ONLINE EXAM, I AFFIRM ON MY HONOR THAT I AM AWARE OF THE STUDENT DISCIPLINARY CODE, AND (I) HAVE NOT GIVEN NOR RECEIVED ANY UNAUTHORIZED AID TO/FROM ANY PERSON OR PERSONS (DIRECTLY OR INDIRECTLY), AND (II) HAVE NOT USED ANY UNAUTHORIZED MATERIALS OR MEANS IN COMPLETING MY ANSWERS TO THIS EXAMINATION.

- ○ I have read and will abide by the Honor Statement.
- ○ I don't care about the Honor code. I have no integrity.

## Section B: Query Optimization (10 Marks)

**2.**

# Fill in the blanks (1 mark)

Consider the following four (4) relations R, S, T, U. The relation schema and statistics of the relations are shown below:

| R(A,B) | S(B, C) | T(C, D) | U(D, E, F) |
|---|---|---|---|
| 100 | 200 | 300 | 400 |
| V(R, A) = 10 | V(S, B) = 50 | V(T, C) = 60 | V(U, D) = 50 |
| V(R, B) = 50 | V(S, C) = 40 | V(T, D) = 100 | V(U, E) = 100 |
| | | | V(U, F) = 200 |

For simplicity, we assume natural joins (all joins are equi-join conditions). Moreover, the resultant schema of any join combination only keeps the non-redundant attributes, e.g., R JOIN S has schema RS(A, B, C).

We also assume all attributes are of the same size, and each page can hold 15 tuples of R.

What is the estimated result size of R JOIN S JOIN T JOIN U?

- Number of tuples: 1
- Number of pages: 2

Enter the correct answer below.

1 Please enter a number for this text box.

2 Please enter a number for this text box.

**3.**

# Fill in the blanks (1 mark)

Using the same setting as Question 2, consider the query: R JOIN S JOIN T JOIN U. Suppose we ignore cross products (i.e., a natural join is a cross product if there is no attribute in common between the left and right relations). What is the number of distinct left-deep trees for this join query? 1

Enter the correct answer below.

1 Please enter a number for this text box.

**4.** In randomized optimization algorithm (e.g., the Local Optimization scheme that we learned in the lecture), we can control the optimization overhead. Which of the following settings is/are reasonably good choice(s)?

(1 mark)

- [ ] Control by a factor based on the number of join operations .
- [ ] Control by a factor based on number of relations.
- [ ] Control by a predetermined (fixed) amount of time (say 10ms).
- [ ] Control by a factor based on the number of tuples.
- [ ] Control by a predetermined (fixed) number of iterations (say, 100 iterations).

**5.** Which of the following statement(s) on Dynamic Programming (DP as taught in the class) is/are true?

(1 mark)

- [ ] Since DP builds the query plan by extending one relation at a time, it eventually considers only one full plan.
- [ ] Although DP's space complexity is exponential with respect to the number of relations, the fact that it can reuse subplans means that, in practice, the space complexity is linear with respect to the number of relations.
- [ ] The principle of optimality in DP allows subplans to be pruned.
- [ ] Since DP maintains only one subplan for each combination of relations, it may miss the global optimal solution even if we just focus on left-deep tree plans.

**6.** Which of the following statements is/are true? (1 mark)

- [ ] A DBMS can generate a query execution plan without knowing the plan space to search.
- [ ] A DBMS can generate a query execution plan without estimating intermediate result sizes.
- [ ] A DBMS can generate a query execution plan without a cost model.
- [ ] A DBMS can generate a query execution plan without knowing the size of the tables involved.
- [ ] A DBMS can generate a query execution plan without a query optimizer.

**7.**

# Fill in the blanks (2 marks)

Consider the following three (3) relations R, S, T. The relation schema and statistics of the relations are shown below:

| R(A,B, C) | S(B, C, D) | T(A, C, E) |
|---|---|---|
| 1000 | 2000 | 3000 |
| V(R, A) = 10 | | V(T, A) = 100 |
| V(R, B) = 50 | V(S, B) = 50 | |
| V(R, C) = 200 | V(S, C) = 40 | V(T, C) = 60 |
| | V(S, D) = 100 | |
| | | V(T, E) = 20 |
| | | |

For simplicity, we assume natural joins (all joins are equi-join conditions). You should make the standard assumptions like independence, uniform, containment of values, etc. For example, whenever two attributes, say A and B, are compared, we can assume that the set of distinct values in A is a subset of that in B (if A is a smaller set). If your answer is a fraction, then you should round up. So, 0.00001 should be 1; 345.7 should be 346.

What is the result size (number of tuples) of the query: SELECT * FROM R WHERE R.A = R.B? 1

Enter the correct answer below.

1 Please enter a number for this text box.

**8.**

# Fill in the blanks (2 marks)

Using the setting as in the previous question, what is the size (number of tuples) of R JOIN S JOIN T? 1

Enter the correct answer below.

1 Please enter a number for this text box.

**9.** Consider two relations R(A, B), S(C, D). In the following statements, we use size to refer to the number of

tuples. Which of the following statement(s) is/are true?

(1 mark)

- [ ] Size of R JOIN(A=C and B \(\not=\) D) S \(\ge\) R JOIN(A=C and B=D) S
- [ ] σ(A=5)(R JOIN(B=C) S) = (σ(A=5)(R)) JOIN(B=C) S
- [ ] Size of R JOIN(A=C) S \(\ge \) Size of R JOIN(A=C and B=D) S
- [ ] (R×S)−(R JOIN(B=C) S) = R JOIN(B \(\not=\) C) S

## Section C: Concurrency Control (18 marks)

**10.** Which of the following schedules is/are Recoverable, Cascadeless and Strict? (1 mark)

- [ ] S1 = w1(X); w2(Y ); w3(Z); c1; r3(X); c2; r3(Y ); c3
- [ ] S3 = w1(X); r2(X); w1(Y ); c1; c2
- [ ] S2 = r1(X); w2(X); w2(Y ); c2; r1(Y ); c1

**11.** Consider the following variant of 2PL: Acquire all locks before a transaction starts; once transaction starts, no more acquiring of locks; locks can be released anytime. Which of the following statement(s) is/are true concerning this variant?

- [ ] The schedule produced under this variant is also 2PL.
- [ ] It is possible for this variant to produce a schedule that is non-recoverable.
- [ ] There will be no deadlock.

**12.** Which of the following statements is/are TRUE? (1 mark)

- [ ] Strict Two-Phase-Locking is guaranteed to produce a serializable schedule.
- [ ] Strict two phase locking generates the same schedule as two phase locking.
- [ ] Strict Two-Phase-Locking is guaranteed to avoid deadlocks.
- [ ] Every transaction that is strict two phase locked is also two phase locked.
- [ ] If we only allow serial executions of transactions, then the ACID properties are guaranteed.

**13.** Consider two transactions T1 and T2 that have exactly the same read set and write set. Moreover, the read and write sets are exactly the same set of elements. Suppose these are the only two transactions in the system. Under the validation-based protocol, which of the following statement(s) is/are true?

(1 mark)

- [ ] It is possible for both transactions to fail to validate successfully.
- [ ] It is possible for both transactions to validate successfully.
- [ ] It is possible for exactly one of the two transactions to validate successfully.

**14.** Consider the following variant of the Validation-based Concurrency Control protocol: Instead of using the Validation Time as the timestamp of transactions, we now use the Start Time of the transactions as its timestamp, i.e., transactions should be ordered by the start time, rather than the validation time. Which of the following statements is/are correct?

(1 mark)

- [ ] This variant is likely to lead to more aborts compared to the method we learned in our lecture.
- [ ] This variant will generate the same transaction ordering as using the method we learn in our lectur

(i.e., order transactions based on Validation Time).

This method does not work since transactions may validate in different order (since a transaction that starts later may validate earlier)!!!

This variant will not have deadlocks.

**15.** 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 6 objects: A, B, C, D, E, F. For transaction $T_4$, the values XXX and YYY represent unknown values (that you are supposed to complete in one of the questions). 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} | {C, D} |
| $T_3$ | {D} | {D, E} |
| $T_4$ | {XXX} | {YYY} |
| $T_5$ | {E} | {C, E} |

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.

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

Which of the following statement(s) is/are true?

(3 marks)

T3 validates successfully.

T1 validates successfully.

T1 does not validate successfully because its write set intersects the write set of T2.

T2 does not validate successfully because its write set intersects the write set of T1.

T2 validates successfully.

- [ ] T5 validates successfully.
- [ ] T3 does not validate successfully because its read set overlaps with the write set of T2.
- [ ] T5 does not validate successfully because it is supposed to finish last.

**16.**

# Fill in the blanks (2 marks)

Refer to the previous question. What is the maximum number of elements in XXX and YYY for T4 to validate successfully:

Max number of elements for XXX = 1

Max number of elements for YYY = 2

Enter the correct answer below.

1 Please enter a number for this text box.

2 Please enter a number for this text box.

**17.** Consider the following schedule S: W1(X); R2(X); W3(Y); W1(Y); W3(Y); W2(X); R3(Y); R2(Y). Which of the following statement(s) is/are true?

(1 mark)

- [ ] S is conflict-serializable, and the serial schedule is T1, T2, T3.
- [ ] S is conflict-serializable, and the serial schedule is T3, T2, T1.
- [ ] S is not conflict serializable.
- [ ] S is conflict-serializable, and the serial schedule is T2, T1, T3.

**18.** Consider the following schedule S: W1(X); R2(X); W3(Y); W1(Y); W3(Y); W2(X); R3(Y); R2(Y). Which of the following statement(s) is/are true?

(1 mark)

- ○ S is not view serializable.
- ○ S is view serializable.

**19.** Consider a DBMS that adopts wait-for-graph deadlock detection method. Which of the following methods may result in aborting a transaction that is not involved in a deadlock?

(1 mark)

- ☐ Pick the transaction with the least number of locks held.
- ☐ Randomly pick an arbitrary transaction from among the transactions involved in a cycle in the graph.
- ☐ Pick the transaction with the most number of transactions waiting for it to release its locks.
- ☐ Pick the transaction with the least work done.

**20.** A live lock may happen when a transaction is repeatedly aborted and restarted many times with no mechanisms to ensure that it can eventually be processed. Which of the following schemes may result in a live lock?

(1 mark)

- ☐ Wait-die deadlock prevention scheme.
- ☐ Wait-for-graph deadlock detection scheme.

- [ ] None of the given optons.
- [ ] Wound-wait deadlock prevention scheme.

**21.** Consider the following variant of wound-wait and wait-die deadlock prevention schemes called wound-die - an older transaction wounds a younger transaction if it needs an object locked by the younger transaction; the younger transaction dies immediately if it needs an object from the older transaction. Which of the following statement(s) is/are true?

(1 mark)

- [ ] Wound-die favors older transactions - older transactions never die or wait!
- [ ] Wound-die is expected to abort more transactions than wound-wait.
- [ ] In Wound-die, younger transactions get aborted immediately, so younger transactions may be starved (i.e., repeatedly aborted and has no chance to complete processing).
- [ ] Wound-die is incorrect since it never let the younger transactions have a chance to wait.

**22.** Which of the following schedules can be generated by 2PL. You should not make any assumptions on when locks can be acquired or released as long as 2PL protocol is observed.

(1 mark)

- [ ] r1(X); w1(X); r2(X); r1(Y); w1(Y); r2(Y); w2(Y); w2(X)
- [ ] r2(Y); w2(Y); r3(Y); r1(X); w1(X); w3(Y); r2(X); r1(Y); w1(Y)
- [ ] r1(X); w1(X); r2(X); w2(X); r3(Y); w3(Y); r1(Y); w1(Y)

**23.** Consider the following sequence of actions: W1(X); W2(Y); W3(Z); W1(Y); W2(Z); W3(X). Suppose our DBMS uses only exclusive locks, and uses wait-die as a deadlock prevention mechanism. We assume the timestamp of a transaction is given by the first action of that transaction. If a transaction gets all its lock, it completes its processing and releases all its locks immediately. Then, any transactions waiting for these

locks can be granted the locks. Moreover, if a transaction is aborted, we assume that it will not be restarted until after all other transactions have completed processing or aborted. If more than one transaction are aborted, then the one that is aborted earlier will be processed first. Which of the following statememt(s) is/are true?

(1 mark)

- [ ] The transactions are executed in the serial order T1, T2, T3.
- [ ] The transactions are executed in the serial order T2, T1, T3.
- [ ] Two transactions have to be aborted.
- [ ] One transaction has to be aborted.
- [ ] The transactions are executed in the serial order T2, T3, T1.

**24.** Consider a DBMS that employs a locking scheme based on the following compatibility matrix involving three lock modes, L1, L2 and L3. A database object can be granted a particular lock mode (as indicated by the column j) if and only if there is no other transaction that holds a lock on the object in some mode k, where the entry in the row k and the column j is "F".

| | L1 | L2 | L3 |
|---|---|---|---|
| L1 | T | T | F |
| L2 | F | T | T |
| L3 | T | F | T |

Which of the following statement(s) is true?

(1 mark)

- [ ] It is possible for different transactions to hold locks on the same object in all three lock modes at the same time.
- [ ] It is possible for different transactions to hold locks on the same database object in modes L1 and L3 at the same time.
- [ ] It is possible for **two transactions** to hold a lock on some object in mode L3, at the same time another transaction holds a lock on the same object in mode L2.

# Section D: Log-based Recovery (12 marks)

**25.** Consider the following log: <START T1>; <T1, X, 10>; <START T2> <T2, Y, 20>; <START CKPT(T1, T2)>; <T2, Z, 30>; <START T3>; <T3, U, 50>; <COMMIT T1>; <END CKPT>; <COMMIT T2>; <COMMIT T3>. What recovery mechanisms is used here?

(1 mark)

- [ ] UNDO logging
- [ ] REDO logging
- [ ] UNDO/REDO logging

**26.** Consider a DBMS that employs Strict 2PL, and log-based recovery. Which of the following statement(s) is/are true on correctness of the recovery process (after a crash)?

- [ ] With REDO logging, the redo process can be done in any order (i.e., not necessarily from earliest log the end of the log).
- [ ] With UNDO logging, the undo process can be done in a reverse order (from latest log to the earliest log).
- [ ] With REDO logging, the redo process can be done in a forward order (from earliest log to the end of the log).
- [ ] With UNDO logging, the undo process can be done in any order (i.e, not necessarily from latest log to the earliest log).

**27.** At the time of a system crash, let the log segment (in the undo/redo logging scheme) be as follows (here the transaction log record is of the form (transaction-id, object, old-value, new-value):

(START S); (S, X, 10, 20); (COMMIT S); (START T); (T, X, 20, 30); (START CKPT(T)); (T, Y, 10, 20); (START U); (COMMIT T); (U, X, 30, 40); (END CKPT); (U, Y, 20, 30); (START V); (START CKPT(U,V)); (COMMIT U); (V, Y, 30, 40)

Immediately after a system crash, what are possible values of X and Y in the database?

(2 marks)

- [ ] Y = 40
- [ ] X = 30
- [ ] X = 40
- [ ] X = 20
- [ ] X = 10
- [ ] Y = 20
- [ ] Y = 10
- [ ] Y = 30

**28.** Which of the following statement(s) is/are TRUE? (1 mark)

- [ ] In redo logging, updating every tuple of a table means the entire table has to be kept in memory til commit time.
- [ ] The primary reason to use undo/redo logging is that recovery after a crash is faster.
- [ ] Undo logging is not as good as undo/redo logging because you can't steal dirty buffer pages from uncommitted transactions if you use undo logging.
- [ ] The weakness of redo logging is that you have to force all the dirty pages of a transaction to disk befo you can write out its COMMIT log record to the log.

**29.** Which of the following statement(s) is/are TRUE? In the statements, the quiescent checkpointing scheme refers

to the naive scheme that we discussed in the lecture.

(1 mark)

- [ ] If you complete a quiescent checkpoint and the system crashes afterward, you will never need to reac the part of the log before the CHECKPOINT log record, no matter what kind of logging you use.
- [ ] For both UNDO and REDO logging schemes, once a non-quiescent checkpoint is successful, you will never need to read the part of the log before the CKPT Start of the most recent successful checkpoint
- [ ] In general, a non-quiescent checkpoint for UNDO logging takes a longer time to finish than a non-quiescent checkpoint for REDO does.

**30.** Consider the following sequence of records present in an undo/redo log (here the transaction log record is of the form (transaction-id, object, old-value, new-value):

(START, S); (S, X, 10, 20); (START T); (COMMIT S); (T, Y, 5,15); (START U); (T, X, 20, 30); (COMMIT T); (U, Y, 15, 25); (U, X, 30, 40).

Let both X and Y fit in the same database page (i.e., both are INPUT from disk to memory together and both are OUTPUT from memory to disk together). Assuming that a quiescent checkpoint occurred just before S started, which of the following are possible pairs of (X, Y) values on disk immediately after the crash.

- [ ] X = 20; Y = 15
- [ ] X = 10; Y = 5
- [ ] X = 20; Y = 25
- [ ] X = 40; Y = 25
- [ ] X = 10; Y = 25
- [ ] X = 30; Y = 15

**31.** In our lecture, we cover REDO logging, UNDO logging and UNDO/REDO logging. However, we did not look at NO UNDO/NO REDO logging method. Under the NO UNDO/NO REDO scheme, there is no need to write any log records. Which of the following statement(s) is/are true?

(1 mark)

- [ ] Although there is no redo and no undo log records, we can still recover the database by using the most recent backup store.
- [ ] Since there is no redo and no undo log records, this means there is no need to do any recovery afte a crash. In other words, we just restart the system.
- [ ] Since there is no REDO log records, there is no way to redo committed transactions. This means the scheme is actually not feasible (i.e., cannot work correctly).
- [ ] Since there is no redo and no undo logs, dirty pages of committed transactions must be written to disk at checkpoints.
- [ ] Since there is no UNDO log records, there is no way to undo uncommitted transactions. This mea the scheme is actually not feasible (i.e., cannot work correctly).

**32.** Consider two objects, X and Y, that are stored in a SINGLE page, i.e., when the page is read/written, both X and Y will be read/written. Moreover, suppose we adopt object-level locking, i.e., it is possible for one transaction to be accessing X, while the other accessing Y. Which of the following statement(s) is/are valid?

(1 mark)

- [ ] We can use REDO logging recovery mechanism.
- [ ] We can use UNDO/REDO logging recovery method.
- [ ] We can use UNDO logging recovery method.

**33.** Consider the following redo log: (1, START S); (2, S, X, 10); (3, COMMIT S); (4, START T); (5, T, X, 20); (6, START CKPT(T)); (7, T, Y, 10); (8, START U); (9, COMMIT T); (10, U, X, 30); (11, END CKPT); (12, U, Y, 20); (13, START V); (14, START CKPT(U,V)); (15, COMMIT U); (16, V, Y, 30). In this log, we have also included the LSN, e.g., consider the log record (5, T, X, 20); here 5 refers to the LSN of the log record (T, X, 20). Which of the following statement(s) is/are true for crash recovery.

(1 mark)

- [ ] REDO X of LSN 2.

- [ ] REDO X of LSN 5.
- [ ] REDO Y of LSN 12
- [ ] REDO X of LSN 10
- [ ] UNDO Y of LSN 16
- [ ] REDO Y of LSN 7

**34.** Consider the following REDO log segment:

<START T1>; <START T2>; <START T3>; <T1, A, 1>; <T2, A, 2>; <T3, A, 3>; <COMMIT T2>;

Which of the following statement(s) is/are true?

- [ ] This is NOT a REDO log since there are logs of uncommitted transactions in the log!
- [ ] The value of A on disk MUST BE 2.
- [ ] The value of A could be 1, 2 or 3.
- [ ] At least one of the transactions performed a dirty read.

Finish Quiz | Save For Later