The marked answers are not related to the actual Answer Key

CS315A: Fundamentals of Database Systems

End-sem Exam

3rd May, 2023, 8:00–11:00

Instructions

Mark neatly on the question paper *itself*, and return. There is no separate answer sheet. Use a *pen* for marking your choice. If multiple choices are marked, that will be considered a wrong answer.

This question paper contains **36** questions in **8** pages. Questions **1-8** are *true-false* questions with **+1** for correct answer, and **−1** for wrong answer. Questions **9-28** are *multiple-choice* questions with **+3** for correct answer, and **−1** for wrong answer. Questions **29-36** are *fill-in-the-blanks* questions with **+4** for correct answer, and **0** for wrong answer. The total marks is, therefore, $8 \times 1 + 20 \times 3 + 8 \times 4 = 100$.

Please sign below, and write your name and roll number on **every page**.

Name **Roll number** **Signature**

True-False

Q1: None of the 2-phase locking protocols can prevent deadlock.

A. True B. False

Q2: If a database has two tables T_1, T_2 and both of them have the same column C, then C is called a foreign key.

A. True B. False

Q3: A read lock can be granted even if there is another existing lock on that data item.

A. True B. False

Q4: The properties satisfied by the HBase database are consistency and partition tolerance.

A. True B. False

Q5: The *maximum* possible size of operation ($r \bowtie s$) is n_s, where n_s is number of tuples in relation s, and $r(X, Y)$, $s(W, X)$ are two relations with X and W as primary keys respectively, and X is a foreign key in s.

A. True B. False

Q6: Locks are *not* required if all transactions work on different data items.

A. True B. False

Q7: A concurrency control manager *never* sacrifices on the correctness even if there could be a deadlock.

A. True B. False

Q8: The schedule given below is an example of cascadeless schedule.

$R_1(A); W_1(A); R_2(B); R_2(C); W_2(B); W_2(C); (commit\ T_1); R_3(A); (commit\ T_2); R_3(B); W_3(A); W_3(B); (commit\ T_3);$

A. True B. False

Multiple-Choice Type

Q9: Consider the following log record with checkpoints (chk_i):

$\langle T_1, start\rangle$; $\langle T_2, start\rangle$; $\langle T_1, A, 10, 20\rangle$; $\langle T_1, B, 30, 40\rangle$; $\langle chk_1\rangle$; $\langle T_1, commit\rangle$; $\langle T_2, C, 10, 15\rangle$; $\langle T_2, B, 40, 50\rangle$; $\langle T_2, commit\rangle$; $\langle chk_2\rangle$; $\langle T_4, start\rangle$; $\langle T_4, B, 40, 50\rangle$; $\langle T_3, commit\rangle$; $\langle T_4, D, 0, 5\rangle$; $\langle T_5, start\rangle$; $\langle T_5, Q, 15, 50\rangle$; $\langle T_4, commit\rangle$;

After the $\langle T_4, commit\rangle$ statement, the system crashes.

Which of the following *undo* and *redo* lists will be used in the *deferred* database modification technique?

A. *redo-list:* $\{T_3, T_4\}$; *undo-list:* $\{\}$

B. *redo-list:* $\{T_3, T_4\}$; *undo-list:* $\{T_5\}$

C. *redo-list:* $\{T_4\}$; *undo-list:* $\{T_5\}$

D. *redo-list:* $\{T_1, T_2, T_3, T_4\}$; *undo-list:* $\{T_5\}$

Q10: Which of the following statements is incorrect?

A. Thomas' write rule provides more concurrency

B. Wait-die and wound-wait schemes avoid starvation by ensuring that an older process never gets aborted

C. Presence of cycle in wait-for graph of deadlock detection algorithm may not always result in deadlock

D. Redo operation of a recovery scheme should be idempotent

Q11: Consider the following schedules:

1. $r_1(a)r_2(b)r_3(c)r_1(b)w_2(b)w_1(a)w_1(b)$
2. $r_1(a)r_1(b)w_2(b)w_1(a)r_3(c)w_1(b)w_3(c)$

Determine whether the transaction T_1 will be allowed or not with the time-stamp ordering protocol and Thomas' write rule.

A. Time-stamp: allowed in only schedule 1; Thomas' rule: allowed in both the schedules

B. Time-stamp: allowed in only schedule 1; Thomas' rule: allowed in only schedule 1

C. Time-stamp: allowed in neither of the schedules; Thomas' rule: allowed in only schedule 2

D. Time-stamp: allowed in only schedule 2; Thomas' rule: allowed in only schedule 2

Q12: Consider the following log record with checkpoints (chk_i):

$\langle T_1, start\rangle$; $\langle T_2, start\rangle$; $\langle T_1, A, 10, 20\rangle$; $\langle T_1, B, 30, 40\rangle$; $\langle chk_1\rangle$; $\langle T_1, commit\rangle$; $\langle T_2, C, 10, 15\rangle$; $\langle T_2, B, 40, 50\rangle$; $\langle T_2, commit\rangle$; $\langle chk_2\rangle$; $\langle T_4, start\rangle$; $\langle T_4, B, 40, 50\rangle$; $\langle T_3, commit\rangle$; $\langle T_4, D, 0, 5\rangle$; $\langle T_5, start\rangle$; $\langle T_5, Q, 15, 50\rangle$; $\langle T_4, commit\rangle$;

After the $\langle T_4, commit\rangle$ statement, the system crashes.

Which of the following *undo* and *redo* lists will be used in the *immediate* database modification technique?

A. *redo-list:* $\{T_3, T_4\}$; *undo-list:* $\{T_5\}$

B. *redo-list:* $\{T_3, T_4, T_2\}$; *undo-list:* $\{T_5\}$

C. *redo-list:* $\{T_4\}$; *undo-list:* $\{\}$

D. *redo-list:* $\{T_3, T_4, T_5\}$; *undo-list:* $\{\}$

Q13: Which of the following are possible optimizations for disk-block access?

I Disk arm scheduling (*Elevator algorithm*)

II File organization (reduce random I/Os compared to sequential I/O)

III Deferred writes (write buffers)

A. Only I

B. Only I and III

C. Only II and III

D. All of I, II and III.

Q14: Suppose in the midst of a recovery process, the system crashes again.
Which of the following statements is then true?

A. Same *undo* and *redo* list will be used again to recover.

B. All the transactions which have been already undone and redone will not be repeated again.

C. System cannot recover from this state.

D. There is insufficient information to answer.

Q15: Which of the following statements are correct?

(1) 2-phase locking ensures view serialization.
(2) Strict 2-phase locking protocol is deadlock free.
(3) A schedule, following rigorous 2-phase protocol, can be serialized by the commit order of its transactions.
(4) All schedules produced by a strict 2-phase locking protocol are recoverable.

A. Only (1) and (2)

B. Only (3) and (4)

C. Only (1), (3) and (4)

D. All of them

Q16: Consider the following relation schema *r* (with the candidate key underlined) and its functional dependencies.

$$r(\underline{A}, B, C, D)$$
$$A \rightarrow BCD;\ C \rightarrow D$$

Why is *r* not in 3NF?

A. All the non-prime keys depend on prime keys.

B. There is transitive dependency.

C. All the non-prime keys depend on candidate keys.

D. The relation is actually in 3NF.

Q17: Given below is the schema of a library database where the primary keys are underlined and the foreign keys are italicized.

Book(bid, title, *pid*, *aid*)
Author(aid, aname)
Publisher(pid, pname)
Customer(cid, cname, address, phone)
Issued(*bid*, *cid*, issue_datetime, return_datetime)

A tuple is created to record the issuing of a book in table *Issued* at the time of issue with *issue_datetime* as CURRENT_TIMESTAMP and *return_datetime* set as *null*. At the time of return, *return_datetime* is recorded as CURRENT_TIMESTAMP. What is the result of the following query?

```
SELECT *
FROM Book
WHERE bid NOT IN
        (SELECT bid FROM Issued
        WHERE return_datetime = NULL);
```

A. The details of all the books that the library has.

B. The details of books that have never been issued.

C. The details of books that have been issued but never returned.

D. The details of books that are currently available in the library.

Q18: A relation has the property that all its attributes are atomic.

What can be inferred from this about the relation?

A. The relation stores names using at least two attributes, one for first name and other for last.

B. The relation has a foreign key.

C. The relation is in 1NF.

D. None of the above.

Q19: Solve the following:

$$\Pi_{A,D}\left(\sigma_{B<3,C>3}\left((p - \rho_{s(A,B)}(s)) \times (\rho_{q(C,D)}(q) \cup r)\right)\right)$$

p

A	B
1	2
2	4
3	6

q

A	B
2	1
4	2
6	3

r

C	D
1	2
4	2
6	3

s

C	D
2	1
2	4
6	3

A.

A	B	C	D
1	2	4	2
1	2	6	3

B.

A	D
1	2
1	3

C.

A	D
2	4
2	6

D.

A	D
3	1
3	2

Q20: Given below is the schema of a library database where the primary keys are underlined and the foreign keys are italicized.

Book(bid, title, *pid*, *aid*)
Author(aid, aname)
Publisher(pid, pname)
Customer(cid, cname, address, phone)
Issued(*bid*, *cid*, issue_datetime, return_datetime)

A tuple is created to record the issuing of a book in table *Issued* at the time of issue with *issue_datetime* as CURRENT_TIMESTAMP and *return_datetime* set as *null*. At the time of return, *return_datetime* is recorded as CURRENT_TIMESTAMP. Write an SQL query to record the return of book with bid 1234.

A. INSERT INTO Issued VALUES (1234, CURRENT_TIMESTAMP, CURRENT_TIMESTAMP);

B. UPDATE Issued SET return_datetime = CURRENT_TIMESTAMP;

C. UPDATE Issued SET return_datetime = CURRENT_TIMESTAMP where bid = 1234;

D. INSERT INTO Issued (bid, return_datetime) VALUES (1234, CURRENT_TIMESTAMP);

Q21: Consider the following database schema for an online game, where the primary keys are underlined and the foreign keys are italicized.

Account(id, *username*, email, password)
Player(username, *cid*, attack-value, attack-rank, defense-value, defense-rank, gold, rank)
Attack(aid, *attacker*, *defender*, result, gold_stolen)
Clan(cid, cname)

Each account is tied to a player name. Players attack each other to steal gold and raise their *attack-value* and *defense-value* to earn better rank.

Players can belong to up to 1 clan. If a player is not part of any clan, then the *cid* field is set to NULL.

The power of a clan C with players $P_1, \ldots, P_n$ is $P(C) = \Sigma(1/rank(P_i))$.

The clan with the highest power is ranked first.

Which of the following queries calculate the current power of the clans and display them so that the first rank appears first?

A. SELECT cid, 1/SUM(rank) as power
FROM Player
GROUP BY cid
ORDER BY power DESC;

B. SELECT Clan.cid, SUM(1/rank) as power
FROM Player, Clan
GROUP BY cid
ORDER BY power DESC;

C. SELECT Clan.cid, SUM(1/rank) as power
FROM Player, Clan
WHERE Player.cid = Clan.cid
GROUP BY cid
ORDER BY power ASC;

D. SELECT cid, SUM(1/rank) as power
FROM Player
GROUP BY cid
ORDER BY power DESC;

Q22: Consider the following database schema for an online game, where the primary keys are underlined and the foreign keys are italicized.

Account(<u>id</u>, *username*, email, password)
Player(<u>username</u>, *cid*, attack-value, attack-rank, defense-value, defense-rank, gold, rank)
Attack(<u>aid</u>, *attacker*, *defender*, result, gold_stolen)
Clan(<u>cid</u>, cname)

Each account is tied to a player name. Players attack each other to steal gold and raise their *attack-value*, *defense-value* to earn better rank. Players get certain amount of gold every hour, based on their armysize. Players can attack other players. If their attack-value is higher than the defender's defense-value, the attack is considered successful; else, it fails. After the attack is performed, its record is logged in the Attack table, where the *attacker* and *defender* fields refer to *usernames* of the players involved. The result is recorded as 'success' or 'failure' from the attacker's point of view. A successful attack results in *all* of the defender's current gold being stolen by the attacker. A failed attack has *no* effect on anyone's gold.

Given players 'P' and 'Q', which SQL query finds out the *net* amount of gold stolen by 'P' from 'Q', i.e., amount of gold stolen by 'P' from 'Q' minus that stolen by 'Q' from 'P'?

A. SELECT (G1.gold - G2.gold) AS total_gold_stolen
FROM
(SELECT SUM(gold_stolen) as gold FROM Attack WHERE attacker = 'P' AND defender = 'Q') AS G1,
(SELECT SUM(gold_stolen) as gold FROM Attack WHERE attacker = 'Q' AND defender = 'P') AS G2;

B. SELECT SUM(gold_stolen) AS total_gold_stolen
FROM Attack
WHERE attacker = 'P' and defender = 'Q';

C. SELECT SUM(gold_stolen) AS total_gold_stolen
FROM Attack
WHERE (attacker = 'P' AND defender = 'Q') OR (attacker = 'Q' AND defender = 'P');

D. SELECT SUM(G1.gold_stolen) - SUM(G2.gold_stolen) AS total_gold_stolen
FROM Attack as G1, Attack as G2
WHERE ((G1.attacker = 'P' AND G1.defender = 'Q') OR (G2.attacker = 'Q' AND G2.defender = 'P')) AND (result = 'success');

Q23: Which of the following statements is/are *incorrect*?
(I) A commit point is reached right after all the operations have been done correctly.
(II) Every transaction operation is recorded in logs/journals when a shadow database is used.

A. Only (I)
B. Only (II)
C. Both (I) and (II)
D. Neither (I), nor (II)

Q24: The minimal cover for the set of functional dependencies $\{A \rightarrow B, C \rightarrow B, BC \rightarrow A, A \rightarrow E, AC \rightarrow D, CD \rightarrow E\}$ on a relation schema $R(A, B, C, D, E)$ is

A. $\{A \rightarrow B, C \rightarrow A, A \rightarrow E, CD \rightarrow E\}$
B. $\{A \rightarrow B, C \rightarrow A, C \rightarrow E, A \rightarrow E, C \rightarrow D\}$
C. $\{A \rightarrow B, C \rightarrow A, A \rightarrow E, C \rightarrow D\}$
D. $\{A \rightarrow B, C \rightarrow B, A \rightarrow E, C \rightarrow D, CD \rightarrow E\}$

Q25: Consider three relations $r(A, B)$, $s(C, D)$ and $t(E, F)$. Which of the following expression is equivalent to
$(r \bowtie_{A=C} s) \bowtie_{(D=E)\wedge(B=E)} t$?

A. $r \bowtie_{(A=C)\wedge(D=E)} (s \bowtie_{B=E} t)$
B. $r \bowtie_{(A=C)\wedge(B=E)} (s \bowtie_{D=E} t)$
C. $r \bowtie_{(D=E)\wedge(B=E)} (s \bowtie_{A=C} t)$
D. None of the above

Q26: Consider the relation $r(X, Y)$. If the number of tuples in r is $n_r = 1200$ and the number of distinct values of attributes X and Y in r are $d_X = 6$ and $d_Y = 5$ respectively, what are the estimated size of the following queries:

$Q1$: $\sigma_{(X=a)\wedge(Y=b)}$
$Q2$: $\sigma_{(X=a)\vee(Y=b)}$

A. 40 and 625 respectively
B. 30 and 400 respectively
C. 60 and 600 respectively
D. None of the above

Q27: Consider the following schedule S that operates on two data items A and B:
$S: R_2(A), R_2(B), W_1(A), W_1(B), W_2(B), W_3(B)$

Select the most appropriate option.

A. S is conflict serializable
B. S is *not* view serializable
C. Removing $W_3(B)$ from S will make it non view serializable
D. None of the above

Q28: Hashing, in the database context, faces the issue of (i) ______, which can be handled by using (ii) ______.
Which of the following is a valid completion of the above statement?

A. (i) collision, (ii) closed hashing
B. (i) overflow, (ii) chaining
C. (i) collision, (ii) open hashing
D. All the completions are wrong.

Fill-in-the-Blanks

Q29: Consider the following tables and the query q,

$$q = \sigma_{(P \geq M + 30) \wedge (M + N < 40)} (T_1 \times T_2)$$

T_1:

P	Q	R
22	51	12
57	23	92
43	16	17
35	76	34
17	11	14

T_2:

L	M	N
32	22	21
23	13	3
14	23	17
11	5	32
23	31	4

How many rows will be there in the result set of query q? 5

Q30: Consider the following tables and the query q,

$$q = \sigma_{(A = C) \wedge (D > 3)} (T_1 \times T_2)$$

T_1:

A	B
1	3
1	7
2	3
4	5

T_2:

C	D	E
2	9	7
4	3	7
1	5	9
1	3	5

How many rows will be there in the result set of query q? 3

Q31: The number of ways the join operation can be performed between 4 tables when commutativity is *not* considered different is 7.

Q32: The number of ways the join operation can be performed between 4 tables when commutativity is considered *different* is 168.

Q33: Consider three transactions T_1, T_2 and T_3, where the number of operations in transaction T_1 is 4, T_2 is 3, and T_3 is 2. How many *concurrent* schedules are possible for the given transactions T_1, T_2 and T_3? 1260

Q34: What is the total number of conflict equivalent serial schedules possible for the given schedule? 2
$r_1(a), w_1(a), r_2(a), w_2(a), r_4(a), r_3(a), r_3(b), r_4(c), w_4(c), w_3(b), r_5(b), r_5(c), w_5(b), w_5(c)$

Q35: Consider two transactions T_1 and T_2 as given below:
T_1: $r_1(a), w_1(a)$
T_2: $r_2(b), w_2(b), w_2(a)$
The total number of conflict serializable schedules that can be formed by T_1 and T_2 is 10.

Q36: Consider two transactions T_1 and T_2, where the number of operations in transaction T_1 is 6, and T_2 is 3. How many *concurrent* schedules are possible for the transactions T_1 and T_2? 84

Space for Rough Work