05.05.2021

EAST WEST UNIVERSITY
Department of Computer Science and Engineering
B.Sc. in Computer Science and Engineering Program
Theoretical Assignment II (Online), Spring 2021 Semester

Course:	**CSE 464 – Advanced Database Systems**
Instructor:	**Mohammad Rezwanul Huq, Ph.D., Associate Professor, CSE Department**
Full Marks:	**100 (15 will be counted for final grading)**
Submission Deadline:	**Wednesday, 05 May 2021, 11:59 PM**

Note: There are **10** (**TEN**) questions. Answer ALL of them. The Mark of each question is mentioned at the right margin.

---

1. Consider the following transactions T1, T2, and T3. [Marks: 10]
T1: r(A), r(c), C = C + A, w(C)
T2: r(C), r(B), B = B + C, w(B), r(A), A = A – C, w(A)
T3: r(C), r(A), A = A * C, w(A), C = C – 10, w(C)

Assume that the initial values of A, B, and C are 50, 20, and 10, respectively.

a) What are their final values of A, B, and C, if we execute the transaction serially using the order T2, T1, and T3?

b) Using the same initial values, what are the final values of A, B, and C, if the order of execution is T1, T2, and T3.

c) Does this result have any implications for serializability?

2. Consider the following schedule S. [Marks: 10]

T1	T2	T3
r(A)		
r(B)		
	r(A)	
		w(B)
		r(C)
	w(A)	
	w(B)	
w(C)		
	r(C)	
commit-req		
	commit-req	
		commit-req

Apply the standard lock, unlock, lock-upgrade, lock-downgrade instructions to the schedule S mentioned above to follow the standard two-phase locking protocol (2PL). Does deadlock occur in that schedule? Explain your answer using the appropriate wait-for graph.

3. Apply the standard timestamp-based ordering protocol to schedule S, as mentioned in Question 2. Maintain a matrix to show the R-timestamp and W-timestamp of each data item when a relevant instruction executes. Will the protocol allow the [Marks: 10] execution of that schedule? Are there any rollbacks? If so, are they cascading rollbacks? Justify your answer.

4. Apply the Thomas' Write Rule to schedule S as mentioned in Question 2. Are there fewer rollbacks than the standard timestamp-based ordering protocol, as you have found in Question 3? Provide an appropriate explanation. [Marks: 10]

5. Consider all three transactions as provided in Question 2 are concurrent. Apply Snapshot Isolation algorithm with first-updater wins rule to the schedule and determine which transactions need to be rolled back. Explain your answer. [Marks: 10]

6. Suppose there are two bank accounts, A and B, held by a single person. The balance of these accounts is referred to as $balance_A$ and $balance_B$. At the time of performing a transaction, the system must satisfy the following constraints – [Marks: 10]

$$balance_A + balance_B \geq 0$$

Both balances are currently \$100. The account holder initiates two transactions concurrently, T1 withdrawing \$200 from account A and T2 withdrawing \$200 from account B.

Explain the differences in the outcome (transaction's commit/abort) between when the database system guarantees serializable schedules and when the database system is under Snapshot Isolation using appropriate terms.

7. Write a schedule with two transactions of your own that shows lost update and non-repeatable read inconsistency problems. How can you avoid these problems? [Marks: 10]

8. Consider the following table. Write yes/no the appropriate cell. [Marks: 10]

Protocol	Problems			
	Serializability	Deadlock	Starvation	Cascading Rollback
Standard 2PL				
Rigorous 2PL				
Standard Timestamp-based Ordering				
Snapshot Isolation				

9. Calculate the optimal number of block transfers and disk seeks required for each of the following algorithms assuming the worst-case scenario, performing the operation: [Marks: 20]

$$\texttt{MovieStar} \bowtie \texttt{StarsIn}$$

i) Nested-loop join
ii) Block nested-loop join
iii) Indexed nested-loop join
iv) Merge Join
v) Hash Join

Assume that relations are unsorted, and the main memory holds only three blocks (M=3), and the buffer memory holds only two blocks (b_b = 2).

Appendix

Consider the following Movie database schema for Question 9.

```
Movie (title, releaseYear, duration, studioName)
MovieStar (starName, address, gender, birthdate, annualIncome)
StarsIn (title, starName)
Studio (studioName, studioAddress, country)
```

Assume that you have the following information about the Movie database.

- All tables are stored in blocks of 4096 bytes.
- Attribute values are uniformly distributed.

Table Name	Number of Tuples	Number of blocks	Number of unique values per attribute	Available Indices
Movie	1000	50	• releaseYear: 50 • duration: 100 • studioName: 90	i. Primary Index on title with h_i = 4 ii. Secondary Index on releaseYear with h_i = 2 iii. Secondary Index on studioName with h_i = 2
MovieStar	3000	400	• address: 2950 • gender: 2 • birthdate: 1500 • annualIncome: 1000	i. Primary Index on annualIncome with h_i = 4 ii. Secondary Index on birthdate with h_i = 4 iii. Secondary Index on starName with h_i = 6
StarsIn	9000	600	• title: 950 • starName: 3000	i. Primary Index on title with h_i = 4 ii. Secondary Index on starName with h_i = 6
Studio	100	5	• studioAddress: 50 • country: 20	i. Primary index on country with h_i = 2 ii. Secondary Index on studioName with h_i = 2

- Maximum and Minimum value of *releaseYear* are 2020 and 1970, respectively.
- Maximum and Minimum value of *duration* are 200 and 50 minutes, respectively.
- Maximum and Minimum value of *annualIncome* are 570000 and 70000 dollars, respectively.