

**CSE370 : Database Systems**  
**Practice Sheet**  
**Transaction Management**

\*Solutions are at the end of the document. SOLVE IT YOURSELF first.

**Question 1:**

Suppose you and your friend are both buying a train ticket to Sylhet from your own phones. There is one seat left. Both of you try to book that one seat, the database should not allow it, otherwise two people will have booked the same seat. Identify and explain which ACID property is necessary to ensure that the same seat is not booked twice.

**Question 2:**

Two transactions A and B are shown below. Identify which concurrency problem- Lost update, Temporary Update or Incorrect Summary, will occur in the following transaction and explain your answer.

T1	T2
READ_ITEM (X) X=X+15	
	READ_ITEM (X) X=X-25 WRITE_ITEM (X)
WRITE_ITEM (X)	

**Question 3:**

Two transactions A and B are shown below, Transaction A fails in the last step and rolls back. Identify which concurrency problem- Lost update, Temporary Update or Incorrect Summary, will occur in the following scenario and explain your answer.

A	B
Read_Item (X) X= X +100 Write (X)	
	Read_item (X) X = X + 500 Write_item (X)
Read_item (Y) Y = Y + X Write_item (Y)	

**Question 4:**

Two transactions A and B are shown below. In transaction A, the user wants to find the total of X and Y. Identify which concurrency problem- Lost update, Temporary Update or Incorrect Summary, will occur due to the following transactions and explain your answer.

A	B
Read_Item (X) Sum = 0 Sum = Sum + X	
	Read_item (Y) Y = Y + 500 Write_item (Y) Read_Item (X) X = X - 500 Write_item (X)
Read_item (Y) Sum = Sum + Y	

**Question 5:**

You are an employee at a travel agency. You added 5 tickets for a customer on a particular flight. After clicking 'pay' the tickets could not be bought as there were only 4 seats left instead of 5. In this scenario, identify if the DBMS will use the recovery system or the concurrency control mechanism. Explain your answer.

**Question 6:**

You are using your credit card to pay for an item online. You received an OTP on your phone which you entered incorrectly and pressed submit.

- In this scenario, will the transaction execute completely or will it fail? Identify which ACID property ensures that the purchase does not take place and explain your answer.
- If it fails, identify and explain if the DBMS will use the recovery system or the concurrency control mechanism.

**Question 7:**

Suppose you are a student at BRAC university. For the upcoming semester you have added 3 courses and paid your fees. Just before the start of the semester, the BRAC U server caught fire and all data was lost, this has created a huge problem for the university and many students, as many of you are not being able to add the same courses or same sections, or even the same number of courses even though most students have paid for their courses.

- Identify and explain which ACID property is necessary to ensure the above situation does not arise

- b. Identify and explain if the recovery system or the concurrency control mechanism will be required to ensure the property in (a).

**Question 8:**

Imagine a banking application where two users, Alice and Bob, both have access to the same bank account. Let's say the current balance of the account is \$100.

- Alice checks the account balance and sees \$100.
- Bob also checks the account balance and sees \$100.
- Alice wants to transfer \$50 to her friend, so she initiates a transfer transaction.
- At the same time, Bob decides to deposit \$30 into the account.
- Now,
- Alice's transaction deducts \$50 from the account, so the balance becomes \$50.
- Before Alice's transaction is completed, Bob's transaction adds \$30 to the account, making the balance \$130.

Identify what type of concurrency problem the above scenario indicates and explain your answer.

**Question 9:**

Imagine a company's employee database where salary information is stored. Now, suppose Alice, who works in the HR department, is updating the salary of an employee named Bob in the database.

- Alice begins updating Bob's salary, mistakenly setting it from \$50,000 to \$900,000.
  - At the same time, Bob logs into the company's employee portal to check his salary.
  - Now, Alice's update transaction starts and changes Bob's salary to \$900,000 and writes it in the database
  - Before Alice's transaction is committed, Bob checks his salary through the employee portal.
  - Bob sees his salary as \$900,000.
  - Bob's salary is not committed and it is reversed back to the previous value by the DBMS when it encounters a "salaryException" raised by the system when it encounters negative or too high salaries.
- a. Identify the type of concurrency problem indicated by the above scenario and explain your answer with respect to the transaction states.
- b. Explain how the transactions can be handled so that this problem does not occur.

**Question 10:**

Suppose you run an online retail business that manages inventory for various products. Two employees, Sarah and John, both have access to update the inventory of a popular item, a smartphone, in your database. Now,

- Sarah is calculating the average sales revenue for the day by summing the prices of all sold items and then finding the average of that.
- Simultaneously, John restocking shelves by updating the quantities of certain products

While both the transactions are being executed-

- John, is updating the amount of sold Samsung smartphones from 50 to 70 units and Oneplus smartphones from 20 to 27.

- At the same time, Sarah starts calculating the average sale for the day, and she finds 70 Samsung smartphones sold and 20 Oneplus smartphones sold. She calculates the average with this data.

Identify the type of concurrency problem indicated by the above scenario and explain how the transactions can be handled so that this problem does not occur.

### Question 11

Observe the scenarios below, and identify which transaction property has been violated and explain your answer-

- An online shopping platform allows customers to place orders for products. Due to a software bug, a customer manages to place an order for a product that is out of stock. The system does not enforce the constraint to check product availability before accepting orders.
- Sarah and Joe work in the same grocery shop. The database maintained by the grocery shop does not ensure proper database management protocols. Now, Sarah started a transaction to increase apple juice stock from 50 to 100 and reduce orange juice stock from 20 to 15. Meanwhile, Joe tried to count the total stock of both juices in the grocery store. He reads the total is 120.
- A social media platform allows users to post updates and comments. A user posts a status update, and the system confirms the successful posting to the user. However, before the data can be persisted to disk, a sudden power outage occurs, causing the data to be lost.
- In a banking system, Akib attempts to transfer money from his account to his friend's account. However, due to a system error, Akib's transfer transaction deducted the amount from his account but failed to update the friend's account.

### Question 12

What are the differences between the four levels of isolation?

### Question 13

Imagine an online shopping platform where customers can place orders for products. Discuss how each of the ACID properties applies to a transaction involving the placement of an order.

### Question 14

Consider a banking system where customers can transfer money between their accounts. Discuss potential concurrency issues that could arise if multiple customers attempt to transfer money simultaneously and how isolation levels can prevent these issues.

## SOLUTIONS:

### Answer 1:

Isolation.

Isolation means concurrent transactions should not be aware of and interfere with each other. If both the transactions are allowed to execute simultaneously, without isolating them, then one seat will be booked twice. The database should completely execute and commit only one of the transactions first before attempting the other transaction. So either I or my friend only one of them will be booking that available seat and not both.

#### Answer 2:

Lost Update.

Suppose  $x$  has value 50. Transaction A adds 15 to  $x$ , so the value becomes 65, but before transaction A commits the value of  $x$ , Transaction B reads the previous value of  $x(50)$  and subtracts 25 and commits the value 25 in the database. Then transaction A commits its previously calculated value of  $x = 65$  thus overwriting the updated value of transaction B. So the update made by transaction B was lost.

**Note: Explanation should be in your own words, brief and to the point. I used values here to explain, you may not use any values and explain differently as I did for Answer 3 and 4.**

#### Answer 3:

Temporary Update.

Transaction A reads the value of  $x$ , updates it and commits the updated value in the database. After that transaction B reads the updated value of  $x$  and makes some more updates and commits the new update in the database. Then transaction A continues and fails. So it now has to roll back and undo the updates made to  $x$  due to the atomicity property. So,  $x$  will return to the value it had before transaction A started. Thus, transaction B read a temporary value and made changes, which was now reversed. So it causes a temporary update problem.

#### Answer 4:

Incorrect Summary

Transaction A is supposed to add the value of  $X$  and  $Y$ . However, after it reads the value of  $x$ , transaction B starts. Transaction B changes the value of  $Y$  and  $X$  and commits it. Now it returns to transaction A, which reads the value of  $Y$  after it was updated by a different transaction. Thus the summation was incorrect as the value of  $x$  used was before updating and value of  $y$  used was after updating.

#### Answer 5:

The DBMS will use the recovery system.

In this scenario the transaction fails at the time of execution due to insufficient data (i.e. tickets). Since the transaction fails, the entire transaction needs to be rolled back to its previous consistent state, i.e. no tickets should be purchased. Thus the recovery system will allow the database to roll back to its previous state. Here concurrency control is not required, because the issue was not caused by concurrent transactions.

#### Answer 6:

- a. Here the transaction will fail due to incorrect input by the user, so the purchase of the item will not take place. The atomicity property ensures that the purchase does not take place. According to the atomicity property, either the entire transaction will be successfully executed or the entire transaction will be aborted.
- b. The DBMS will use the recovery system. Here concurrency control is not required, because the issue was not caused by concurrent transactions. Instead the issue is that the transaction failed due to an error/exception and so the database has to be returned to its previous consistent state, by undoing any changes/updates made during the transaction.

#### Answer 7:

- a. Durability. Every transaction must be permanently committed in the database. The dbms should have appropriate backups and/or mechanisms in place to ensure that data is not lost
- b. The DBMS will use the recovery system in this scenario to ensure the durability property is enforced. The recovery system can roll back to a consistent state before the fire.

#### Answer 8:

Lost update problem.

Here the update made by Alice i.e. the deduction of \$50 dollars is not committed before adding \$30 dollars, so the change is lost. Either Bob's or Alice's transaction should have been committed before the other's transaction started. If there was concurrency control then the final amount in the account after both transactions would have been \$80 ( $100 - 50 + 30$ ).

#### Answer 9:

- a. The Temporary Update (or Dirty Read) Problem. After updating the salary to \$900,000, the transaction is in the partially committed state, but when the salary exception is raised by the system the transaction aborts and goes to the failed state. So now the recovery system will roll back the transaction to Bob's previous salary before the update. Since Bob already read the temporarily updated salary before it was committed, the temporary update problem occurs.
- b. The concurrency control mechanism should be used to enforce any of these following isolation levels so that the temporary update problem does not occur:
  - Serializable → a transaction will be executed after the other transaction terminates.
  - Read Committed or Repeatable Read → only a committed value can be read by another transaction.

#### Answer 10:

Incorrect summary problem as sarah starts the calculation after John partially updates some of the data but before he completes all of the updates. The concurrency control mechanism should be used to enforce the "serializable" isolation level in this scenario, as sarah's transaction should start only after John's transaction has committed and then terminated.

#### Answer 11:

- a) Consistency. The customer should not be able to order out of stock items, this means the data in the system is not valid/consistent.
- b) Isolation. Joe reads the values after the apple juice stock is updated but before the orange juice stock is updated, so the total is incorrect. Each transaction should be isolated from one another, so they should not interfere with each other, but that is not the case here.
- c) Durability. Every transaction should permanently persist in the database, so here, the user should have been notified of the successful completion only after completing the transaction and permanently committing it to the database.
- d) Atomicity. Either the full transaction should be executed or none at all. However, here partial transaction was completed, but the rest was not. So atomicity was not achieved.

#### Answer 12, 13, 14

For 12, read the slide, it is directly written there. For 13, in the slide there is a similar example for transferring 50\$ between two accounts, use the same format to answer this question for a different scenario.

For 14, you should explain each of the concurrency problems - lost update, temporary update and incorrect summary, with respect to the given scenario.