COMP90050 Advanced Database Systems: Tutorial
Winter term, 2023 (Week 3)

Exercises

Part 1

1. Discuss why the isolation property of ACID properties will apply to both an Online shopping platform as well as an Online banking system despite that they are different applications dealing with different data.
2. A bank with millions of customers provides a bonus to each of its customer at the end of the year. The bonus is updated in the database as a flat transaction shown below. Discuss example and the associated issue(s) that can happen with such execution. Is it a good choice to use flat transaction here?

```
GiveEndofYearBonus()
{
      exec sql BEGIN WORK
      for each customer in database
       {
         double bonus = calculate_bonus(customer);
         exec SQL UPDATE customer
           set account = account + :bonus
       }
      exec sql COMMIT WORK
}
```

3. A flat transaction with save-points has the following statements as example. If condition1 is true once and the final commit is successful, what will be printed as the value of count?

```
BEGIN WORK
count = 10
SAVE WORK 1
count = count+10
SAVE WORK 2
count = count+5
SAVE WORK3
count = count+5
If (condition1) ROLLBACK WORK(2)
count = count+1
print count
COMMIT WORK
```

4. In a nested transaction, a transaction PARENT has three sub-transactions A, B, C. For each of the following scenarios, answer which of these four transactions' commits can be made durable, and which ones has to be forced to rollback.

(i) Scenario 1: Commit by A, B, and C; but PARENT rolls back.
(ii) Scenario 2: Commit by A, B, C, and PARENT.

(iii) Scenario 3: Commit by A, B, and PARENT; but C rolls back.

Part 2

5. What is the probability that a deadlock situation occurs?

6. If we use the following comments to lock and unlock access to objects, then which transactions below are in deadlock if they start around the same time?

T1	T2	T3	T4
Begin	Begin	Begin	Begin
LOCK(C)	LOCK(A)	LOCK(C)	LOCK(B)
Write C	Write A	Write C	Write B
UNLOCK(C)	LOCK(B)	UNLOCK(C)	LOCK(A)
End	Write(B)	End	Write A
	UNLOCK(A)		UNLOCK(A)
	UNLOCK(B)		UNLOCK(B)
	End		End

7. Isolation property in ACID properties states that each transaction should run without being aware or in interference with another transaction in the system. If that is the case, we can run transactions sequentially by locking the whole database itself and adhering to the Isolation property through a big lock per transaction. Review why this may not be an ideal solution.

8. Given two transactions, per operation of each transaction, we can use locks to make sure concurrent access is done properly to individual objects that are used in both transactions i.e., they are not accessed at the same time. This is after all what the operating systems do, e.g., lock a file while one program is accessing it so others cannot change it at the same time. Give two transactions showing that this is not enough to achieve the isolation property of transactions for RDBMSs.

Part 3

9. What are the dependencies in the following history (a sequence of tuples in the form (T i, Oi, T j))? Draw the dependency graph mapping to this dependency set as well.

H =< (T1,R,O1),(T3,W,O5),(T3,W,O1),(T2,R,O5),(T2,W,O2),(T5,R,O4),(T1,R,O2),(T5,R,O3) >

10. Given the solution above for the previous question, can we say the history is equal to a serial history? If yes, show one such history. If not, show that there is a wormhole.

11. Assume the following two transactions start at nearly the same time and there is no other concurrent transaction. The 2nd operation of both transactions is Xlock(B). Is there a potential problem if Transaction 1 performs the operation first? What if Transaction 2 performs the operation first?

Transaction 1		Transaction 2	
1.1	Slock(A)	2.1	Slock(A)
1.2	Xlock(B)	2.2	Xlock(B)
1.3	Read(A)	2.3	Write(B)
1.4	Read(B)	2.4	Unlock(B)
1.5	Write(B)	2.5	Read(A)
1.6	Unlock(A)	2.6	Xlock(B)
1.7	Unlock(B)	2.7	Write(B)
		2.8	Unlock(A)
		2.9	Unlock(B)

12. What degree of isolation does the following transaction provide?

Slock(A)
Xlock(B)
Read(A)
Write(B)
Read(C)
Unlock(A)
Unlock(B)

13. The following operations are given with Degree 2 isolation locking principles in place. Convert the locking sequence to Degree 3.

Degree 2

Slock(A)
Read(A)
Unlock(A)
Xlock(C)
Xlock(B)
Write(B)
Slock(A)
Read(A)
Unlock(A)
Write(C)
Unlock(B)
Unlock(C)