

Transaction & Recovery Management

1. พิจารณา transactions ต่อไปนี้ แล้วตอบคำถามที่ระบุในตารางนี้

T	Session	Explanation
t0	COMMIT;	This statement ends any existing transaction in the session.
t1	SET TRANSACTION NAME 'sal_update';	This statement begins a transaction and names it sal_update.
t2	UPDATE employees SET salary = 7000 WHERE last_name = 'Banda';	This statement updates the salary for Banda to 7000.
t3	SAVEPOINT after_banda_sal;	1.1 คำสั่งนี้มีไว้เพื่ออะไร
t4	UPDATE employees SET salary = 12000 WHERE last_name = 'Greene';	This statement updates the salary for Greene to 12000.
t5	SAVEPOINT after_greene_sal;	1.2 คำสั่งนี้มีไว้เพื่ออะไร
t6	ROLLBACK TO SAVEPOINT after_banda_sal;	1.3 คำสั่งนี้สั่งให้ทำอะไร
t7	UPDATE employees SET salary = 11000 WHERE last_name = 'Greene';	This statement updates the salary for Greene to 11000 in transaction sal_update.
t8	ROLLBACK;	1.4 คำสั่งนี้สั่งให้ทำอะไร
t9	SET TRANSACTION NAME 'sal_update2';	This statement begins a new transaction in the session and names it sal_update2.
t10	UPDATE employees SET salary = 7050 WHERE last_name = 'Banda';	This statement updates the salary for Banda to 7050.
t11	UPDATE employees SET salary = 10950 WHERE last_name = 'Greene';	This statement updates the salary for Greene to 10950.
t12	COMMIT;	This statement commits all changes made in transaction sal_update2, ending the transaction. The commit guarantees that the changes are saved in the online redo log files.

2. Serializable Schedule

ทรานแซกชันต่อไปนี้เป็นไปโดยประกอบด้วยคำสั่ง read และ write ข้อมูล ซึ่งหมายถึงคำสั่ง select และ insert, update attributes ในฐานข้อมูล

พิจารณาตัวอย่างในรูปต่อไปนี้เป็นการทำงานตามคำสั่งใน 2 ทรานแซกชันตามลำดับดังนี้ นั่นคือเป็น schedule ที่เป็น serializable เนื่องจากไม่มีการใช้ข้อมูลร่วมกัน ดังนั้นผลต่อฐานข้อมูลก็เหมือนกับการ execute T1 ต่อด้วย T2 และเหมือน execute T2 ต่อด้วย T1.

T1	T2
R(A)	
W(A)	
	R(B)
	W(B)
R(C)	
W(C)	
Commit	
	Commit

“A serializable schedule คือการจัดลำดับการทำงานของคำสั่งใน transactions เช่น T1 และ T2 แล้วทำให้ผลกระทบต่อฐานข้อมูลเหมือนกับการ execute T1 ต่อด้วย T2 หรือเหมือน execute T2 ต่อด้วย T1.”

2.1 Is this schedule serializable? Why

T1	T2
R(A)	
W(A)	
	R(B)
	W(B)
	Commit
R(B)	
W(B)	
Commit	

2.2 Is this schedule serializable? Why

T1	T2
R(A)	
	R(B)
W(A)	
	W(B)
	Commit
R(B)	
W(B)	
Commit	

2.3 Is the schedule serializable?

T1	T2
R(A)	
W(A)	
	R(A)
	W(A)
	R(B)
	W(B)
	Commit
R(B)	
W(B)	
Commit	

3. ระบุปัญหาของ schedule ต่อไปนี้ ที่มีผลต่อความถูกต้องของข้อมูลในฐาน

S1

Transaction A	Time	Transaction B
Read SR (10)	t_1	
SR = SR - 1	t_2	
Write SR (9)	t_3	
	t_4	Read SR (9)
Rollback	t_5	

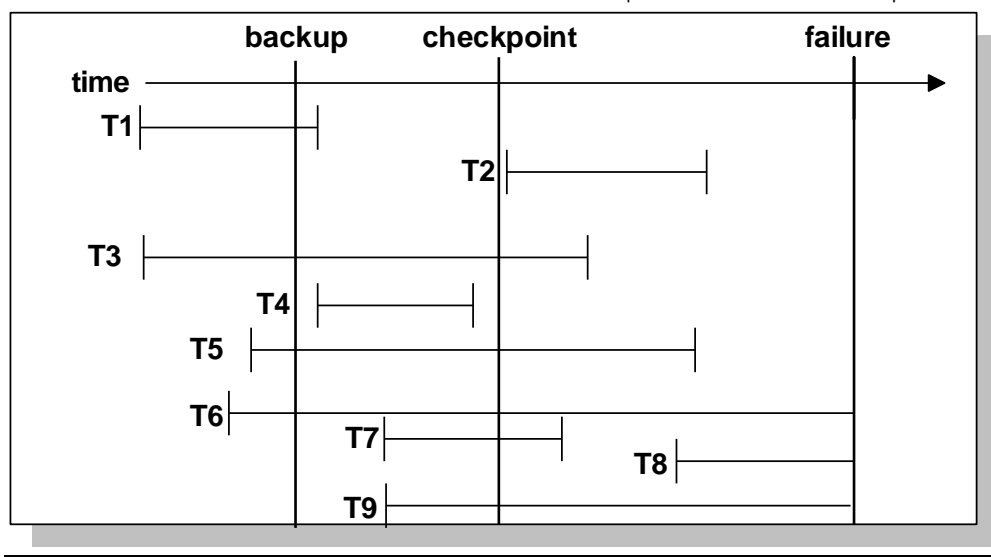
ตอบ:

S2

Transaction A	Time	Transaction B
Read SR (10)	t_1	
	t_2	Read SR
	t_3	SR = SR - 1
	t_4	Write SR (9)
Read SR (9)	t_5	

ตอบ:

4. Timeline ต่อไปนี้ใช้สถานะของ transactions ต่างๆ ณ เวลาที่มีการ backup, checkpoint, failure.



- 4.1 ถ้า transaction T3 ถูก aborted (ยกเลิกก่อนเสร็จด้วยคำสั่ง rollback) หลังจากจุด checkpoint, ถ้าเลือกใช้ Deferred update approach. T3 จะถูก restart หรือไม่ เพราะอะไร
- 4.2 ถ้า transaction T3 ถูก aborted (ยกเลิกก่อนเสร็จด้วยคำสั่ง rollback) หลังจากจุด checkpoint, ถ้าเลือกใช้ Immediate update approach. T3 จะถูกฟื้นฟูอย่างไร (undo redo / restart)
- 4.3 ถ้าเกิด system failure เช่นไฟดับ ซึ่งจะมีผลต่อทรานแซคชันที่ดำเนินการอยู่ ไม่มีผลต่อฐานข้อมูลในดิสก์ และ recovery manager เลือกใช้ Deferred update approach อธิบายสิ่งที่เกิดกับ T1 ถึง T9

Transactions	Undo/Redo/ No action	เหตุผล
T1 and T4		
T6, T8, and T9		
T2, T3, T5, T7		

- Transaction ใดบ้างที่จะถูก restart

- 4.4 ถ้าเกิด system failure และ recovery manager เลือกใช้ Immediate update approach อธิบายสิ่งที่เกิดกับ T1 ถึง T9

Transactions	Undo/Redo/No action	เหตุผล
T1 and T4		
T6, T8, and T9		
T2, T3, T5, T7		

- Transaction ใดบ้างที่จะถูก restart

4.5 ถ้าเกิด a media failure เช่น ดิสก์เสีย recovery manager จะต้องดำเนินการนำ database backup มาติดตั้งเพื่อใช้งาน หลังจากนั้นสิ่งที่ต้องดำเนินการคือ

- นำ transaction log (มีข้อมูลตามรูป)มาดำเนินการอย่างไรกับ transactions T1, T2, T3, T4, T5, T7 ซึ่ง committed หลังการ backup ฐานข้อมูล
- และทำอย่างไรกับ transactions: T6, T8, T9 ซึ่งทำงานยังไม่เสร็จ (uncommitted)

5. T1 และ T2 เป็น 2 Phase Locking หรือไม่ เพราะอะไร

T1	T2
read-lock(X)	read-lock(X)
Read(X)	Read(X)
write-lock(Y)	unlock(X)
unlock(X)	write-lock(Y)
Read(Y)	Read(Y)
$Y = Y + X$	$Y = Y + X$
Write(Y)	Write(Y)
unlock(Y)	unlock(Y)

6. อธิบายโดยใช้ตัวอย่างต่อไปนี้ว่า ทำไม Lost-update problem ไม่สามารถเกิดขึ้นได้เมื่อใช้ 2 Phase Locking (โดยใน 2 Phase Locking ก่อนการ read/write ต้องขอ lock ให้ได้ก่อน)

T1	T2
Read (X)	
$X = X - 5$	
	Read (X)
	$X = X + 5$
Write (X)	Write (X)
COMMIT	COMMIT

7. จงเติมตัวเลขแสดงค่า timestamp ของข้อมูลเมื่อระบบต้อง execute transaction T1 และ T2 โดยใช้ Timestamp Approach สมมติว่า T1 มาก่อน T2 และมีค่า timestamp(TS) เท่ากับ 1 และ 2 ตามลำดับ

	X	Y	Z
R			
W			

	T1	T2
TS		

T1	T2
Read(X)	Read(X)
Read(Y)	Read(Y)
$Y = Y + X$	$Z = Y - X$
Write(Y)	Write(Z)