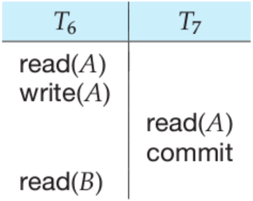
Database Management Systems (CSN-351)
Transactions (Contd.)

BTech 3rd Year (CS) + Minor + Audit

Instructor: **R**anita **B**iswas
Department of Computer Science and Engineering
Indian Institute of Technology Roorkee
Roorkee, Uttarakhand - 247 667, India

Nonrecoverable Schedule

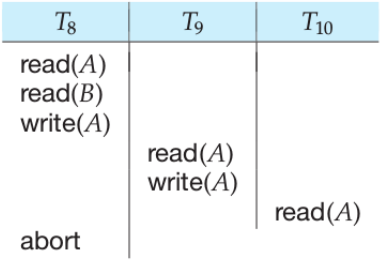


T_6	T_7
read(A)	
write(A)	
	read(A)
	commit
read(B)	

Recoverable Schedule

A **recoverable schedule** is one where, for each pair of transactions T_i and T_j such that T_j reads a data item previously written by T_i , the commit operation of T_i appears before the commit operation of T_j .

Cascading Rollback



T_8	T_9	T_{10}
read(A)		
read(B)		
write(A)		
	read(A)	
	write(A)	
		read(A)
abort		

Cascadeless Schedule

A **cascadeless schedule** is one where, for each pair of transactions T_i and T_j such that T_j reads a data item previously written by T_i, the commit operation of T_i appears before the read operation of T_j.

Isolation Levels

Serializable: ensures serializable execution.

Isolation Levels

Serializable: ensures serializable execution.

Repeatable read: allows only committed data to be read and further requires that, between two reads of a data item by a transaction, no other transaction is allowed to update it.

Isolation Levels

Serializable: ensures serializable execution.

Repeatable read: allows only committed data to be read and further requires that, between two reads of a data item by a transaction, no other transaction is allowed to update it.

Read committed: allows only committed data to be read, but does not require repeatable reads.

Isolation Levels

Serializable: ensures serializable execution.

Repeatable read: allows only committed data to be read and further requires that, between two reads of a data item by a transaction, no other transaction is allowed to update it.

Read committed: allows only committed data to be read, but does not require repeatable reads.

Read uncommitted: allows uncommitted data to be read.

Read Phenomena

A **dirty read** (aka uncommitted dependency) occurs when a transaction is allowed to read data from a row that has been modified by another running transaction and not yet committed.

Read Phenomena

A **dirty read** (aka uncommitted dependency) occurs when a transaction is allowed to read data from a row that has been modified by another running transaction and not yet committed.

A **non-repeatable** read occurs, when during the course of a transaction, a row is retrieved twice and the values within the row differ between reads.

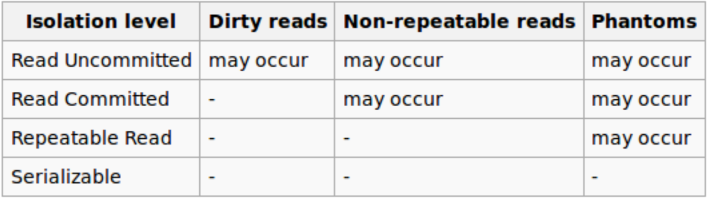
Read Phenomena

A **dirty read** (aka uncommitted dependency) occurs when a transaction is allowed to read data from a row that has been modified by another running transaction and not yet committed.

A **non-repeatable** read occurs, when during the course of a transaction, a row is retrieved twice and the values within the row differ between reads.

A **phantom read** occurs when, in the course of a transaction, two identical queries are executed, and the collection of rows returned by the second query is different from the first.

Isolation Levels vs Read Phenomena



Isolation level	Dirty reads	Non-repeatable reads	Phantoms
Read Uncommitted	may occur	may occur	may occur
Read Committed	-	may occur	may occur
Repeatable Read	-	-	may occur
Serializable	-	-	-

Question 1

Which one of the following is NOT a part of the ACID properties of database transactions?

- ○ Atomicity
- ○ Consistency
- ○ Isolation
- ○ Deadlock-freedom

Question 1

Which one of the following is NOT a part of the ACID properties of database transactions?

○ Atomicity

○ Consistency

○ Isolation

● **Deadlock-freedom**

Question 2

Consider the following transaction involving two bank accounts x and y.
$read(x); x := x - 50; write(x); read(y); y := y + 50; write(y)$
The constraint that the sum of the accounts x and y should remain constant is that of

○ Atomicity

○ Consistency

○ Isolation

○ Durability

Question 2

Consider the following transaction involving two bank accounts x and y.
$read(x); x := x - 50; write(x); read(y); y := y + 50; write(y)$
The constraint that the sum of the accounts x and y should remain constant is that of

○ Atomicity

● **Consistency**

○ Isolation

○ Durability

Question 3

Suppose a database schedule S involves transactions $T_1, ...T_n$. Construct the precedence graph of S with vertices representing the transactions and edges representing the conflicts. If S is serializable, which one of the following orderings of the vertices of the precedence graph is guaranteed to yield a serial schedule?

○ Topological order

○ Depth-first order

○ Breadth-first order

○ Ascending order of transaction indices

Question 3

Suppose a database schedule S involves transactions $T_1, ...T_n$. Construct the precedence graph of S with vertices representing the transactions and edges representing the conflicts. If S is serializable, which one of the following orderings of the vertices of the precedence graph is guaranteed to yield a serial schedule?

- ● **Topological order**
- ○ Depth-first order
- ○ Breadth-first order
- ○ Ascending order of transaction indices

Question 4

Which of the following scenarios may lead to an irrecoverable error in a database system?

- ○ A transaction writes a data item after it is read by an uncommitted transaction
- ○ A transaction reads a data item after it is read by an uncommitted transaction
- ○ A transaction reads a data item after it is written by a committed transaction
- ○ A transaction reads a data item after it is written by an uncommitted transaction

Question 4

Which of the following scenarios may lead to an irrecoverable error in a database system?

○ A transaction writes a data item after it is read by an uncommitted transaction

○ A transaction reads a data item after it is read by an uncommitted transaction

○ A transaction reads a data item after it is written by a committed transaction

● **A transaction reads a data item after it is written by an uncommitted transaction**

Question 5

Consider the transactions $T1, T2$, and $T3$ and the schedules $S1$ and $S2$ given below.
$T1 : r1(X); r1(Z); w1(X); w1(Z)$
$T2 : r2(Y); r2(Z); w2(Z)$
$T3 : r3(Y); r3(X); w3(Y)$
$S1 : r1(X); r3(Y); r3(X); r2(Y); r2(Z); w3(Y); w2(Z); r1(Z); w1(X); w1(Z)$
$S2 : r1(X); r3(Y); r2(Y); r3(X); r1(Z); r2(Z); w3(Y); w1(X); w2(Z); w1(Z)$
Which one of the following statements about the schedules is TRUE?

- ○ Only $S1$ is conflict-serializable.
- ○ Only $S2$ is conflict-serializable.
- ○ Both $S1$ and $S2$ are conflict-serializable.
- ○ Neither $S1$ nor $S2$ is conflict-serializable.

Question 5

Consider the transactions $T1, T2$, and $T3$ and the schedules $S1$ and $S2$ given below.
$T1 : r1(X); r1(Z); w1(X); w1(Z)$
$T2 : r2(Y); r2(Z); w2(Z)$
$T3 : r3(Y); r3(X); w3(Y)$
$S1 : r1(X); r3(Y); r3(X); r2(Y); r2(Z); w3(Y); w2(Z); r1(Z); w1(X); w1(Z)$
$S2 : r1(X); r3(Y); r2(Y); r3(X); r1(Z); r2(Z); w3(Y); w1(X); w2(Z); w1(Z)$
Which one of the following statements about the schedules is TRUE?

● **Only $S1$ is conflict-serializable.**

○ Only $S2$ is conflict-serializable.

○ Both $S1$ and $S2$ are conflict-serializable.

○ Neither $S1$ nor $S2$ is conflict-serializable.

Question 6

Consider the following schedule S of transactions $T1, T2, T3, T4$:

T1	T2	T3	T4
	Read(X)		
		Write(X)	
		Commit	
Write(X)			
Commit			
	Write(Y)		
	Read(Z)		
	Commit		
			Read(X)
			Read(Y)
			Commit

Which one of the following statements is CORRECT?

- S is conflict-serializable but not recoverable
- S is not conflict-serializable but is recoverable
- S is both conflict-serializable and recoverable
- S is neither conflict-serializable nor is it recoverable

Question 6

Consider the following schedule S of transactions $T1, T2, T3, T4$:

T1	**T2**	**T3**	**T4**
	Read(X)		
		Write(X) Commit	
Write(X) Commit			

	Write(Y)
	Read(Z)
	Commit		

			Read(X)
			Read(Y)
			Commit

Which one of the following statements is CORRECT?

○ S is conflict-serializable but not recoverable

○ S is not conflict-serializable but is recoverable

● ***S* is both conflict-serializable and recoverable**

○ S is neither conflict-serializable nor is it recoverable

Question 7

Consider the following four schedules due to three transactions (indicated by the subscript) using read and write on a data item x, denoted by $r(x)$ and $w(x)$ respectively. Which one of them is conflict serializable?

(A) $r_1(x); r_2(x); w_1(x); r_3(x); w_2(x)$

(B) $r_2(x); r_1(x); w_2(x); r_3(x); w_1(x)$

(C) $r_3(x); r_2(x); r_1(x); w_2(x); w_1(x)$

(D) $r_2(x); w_2(x); r_3(x); r_1(x); w_1(x)$

Question 7

Consider the following four schedules due to three transactions (indicated by the subscript) using read and write on a data item x, denoted by $r(x)$ and $w(x)$ respectively. Which one of them is conflict serializable?

(A) $r_1(x); r_2(x); w_1(x); r_3(x); w_2(x)$

(B) $r_2(x); r_1(x); w_2(x); r_3(x); w_1(x)$

(C) $r_3(x); r_2(x); r_1(x); w_2(x); w_1(x)$

(D) $r_2(x); w_2(x); r_3(x); r_1(x); w_1(x)$

ANSWER: **(D)**

Question 8

Consider the following schedule for transactions $T1, T2$ and $T3$:

T1	**T2**	**T3**
Read(X)		
	Read(Y)	
		Read(Y)
	Write(Y)	
Write(X)		
		Write(X)
	Read(X)	
	Write(X)	

Which one of the schedules below is the correct serialization of the above?

○ $T1 \rightarrow T3 \rightarrow T2$

○ $T2 \rightarrow T1 \rightarrow T3$

○ $T2 \rightarrow T3 \rightarrow T1$

○ $T3 \rightarrow T1 \rightarrow T2$

Question 8

Consider the following schedule for transactions $T1, T2$ and $T3$:

T1	**T2**	**T3**
Read(X)		
	Read(Y)	
		Read(Y)
	Write(Y)	
Write(X)		
		Write(X)
	Read(X)	
	Write(X)	

Which one of the schedules below is the correct serialization of the above?

● $T1 \rightarrow T3 \rightarrow T2$

○ $T2 \rightarrow T1 \rightarrow T3$

○ $T2 \rightarrow T3 \rightarrow T1$

○ $T3 \rightarrow T1 \rightarrow T2$

Question 9

Consider the following transactions with data items P and Q initialized to zero:

```
$T1$ :
$read(P)$;
$read(Q)$;
if $P = 0$ then $Q := Q + 1$;
$write(Q)$;
$T2$ :
$read(Q)$;
$read(P)$;
if $Q = 0$ then $P := P + 1$;
$write(P)$;
```

Any non-serial interleaving of $T1$ and $T2$ for concurrent execution leads to

○ A serializable schedule

○ A schedule that is not conflict serializable

○ A conflict serializable schedule

○ A schedule for which a precedence graph cannot be drawn

Question 9

Consider the following transactions with data items P and Q initialized to zero:

```
$T1:$
$read(P);$
$read(Q);$
if $P = 0$ then $Q := Q + 1;$
$write(Q);$
$T2:$
$read(Q);$
$read(P);$
if $Q = 0$ then $P := P + 1;$
$write(P);$
```

Any non-serial interleaving of $T1$ and $T2$ for concurrent execution leads to

○ A serializable schedule

● **A schedule that is not conflict serializable**

○ A conflict serializable schedule

○ A schedule for which a precedence graph cannot be drawn

Question 10

Consider three data items $D1$, $D2$ and $D3$ and the following execution schedule of transactions $T1$, $T2$ and $T3$. In the diagram, $R(D)$ and $W(D)$ denote the actions reading and writing the data item D respectively.

T1	**T2**	**T3**
	R(D3)	
	R(D2)	
	W(D2)	
		R(D2)
		R(D3)
R(D1)		
W(D1)		
		W(D2)
		W(D3)
	R(D1)	
R(D2)		
W(D2)		
	W(D1)	

Which of the following statements is correct about the schedule?

- ○ Serializable as $T2 \rightarrow T3 \rightarrow T1$
- ○ Serializable as $T2 \rightarrow T1 \rightarrow T3$
- ○ Serializable as $T3 \rightarrow T2 \rightarrow T1$
- ○ Not serializable

Question 10

Consider three data items $D1$, $D2$ and $D3$ and the following execution schedule of transactions $T1$, $T2$ and $T3$. In the diagram, $R(D)$ and $W(D)$ denote the actions reading and writing the data item D respectively.

T1	**T2**	**T3**
	R(D3)	
	R(D2)	
	W(D2)	
		R(D2)
		R(D3)
R(D1)		
W(D1)		
		W(D2)
		W(D3)
	R(D1)	
R(D2)		
W(D2)		
	W(D1)	

Which of the following statements is correct about the schedule?

○ Serializable as $T2 \rightarrow T3 \rightarrow T1$

○ Serializable as $T2 \rightarrow T1 \rightarrow T3$

○ Serializable as $T3 \rightarrow T2 \rightarrow T1$

● **Not serializable**