Serializability Summary

Murat Kantarcioglu

Scheduling Transactions

- *Serial schedule:* Schedule that does not interleave the actions of different transactions.
- *Equivalent schedules*: For any database state, the effect (on the set of objects in the database) of executing the first schedule is identical to the effect of executing the second schedule.
- *Serializable schedule*: A schedule that is equivalent to some serial execution of the transactions.

(Note: If each transaction preserves consistency, every serializable schedule preserves consistency.)

View Serializability

- Schedules S1 and S2 are view equivalent
  - If Ti reads initial value of A in S1, then Ti also reads initial value of A in S2
  - If Ti reads value of A written by Tj in S1, then Ti also reads value of A written by Tj in S2
  - If Ti writes final value of A in S1, then Ti also writes final value of A in S2

```
T1: R(A)        W(A)
T2:       W(A)
T3:                   W(A)
```

```
T1: R(A),W(A)
T2:             W(A)
T3:                   W(A)
```

Recoverable, Avoids-cascading-abort, Strict

- Recoverable Schedule: For each pair of transaction Ti and Tj, if Tj reads an object previously written by Ti, Tj commits after Ti commits

- Avoids-cascading-abort Schedule: For each pair of transaction Ti and Tj, if Tj reads an object previously written by Ti, Ti commits before the read operation of Tj.
- Strict Schedule: An object written by T cannot be read or overwritten until T commits or aborts

Conflict Serializability

- Two actions Ai and Aj executed on the same data object by Ti and Tj conflicts if either one of them is a write operation.
- Let Ai and Aj are **consecutive non-conflicting actions** that belongs to different transactions. We can swap Ai and Aj without changing the result.
- Two schedules are conflict equivalent if they can be turned one into the other by a sequence of non-conflicting swaps of adjacent actions.

Conflict Serializability

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

Conflict Serializability

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

Conflict Serializability

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

Conflict Serializability

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

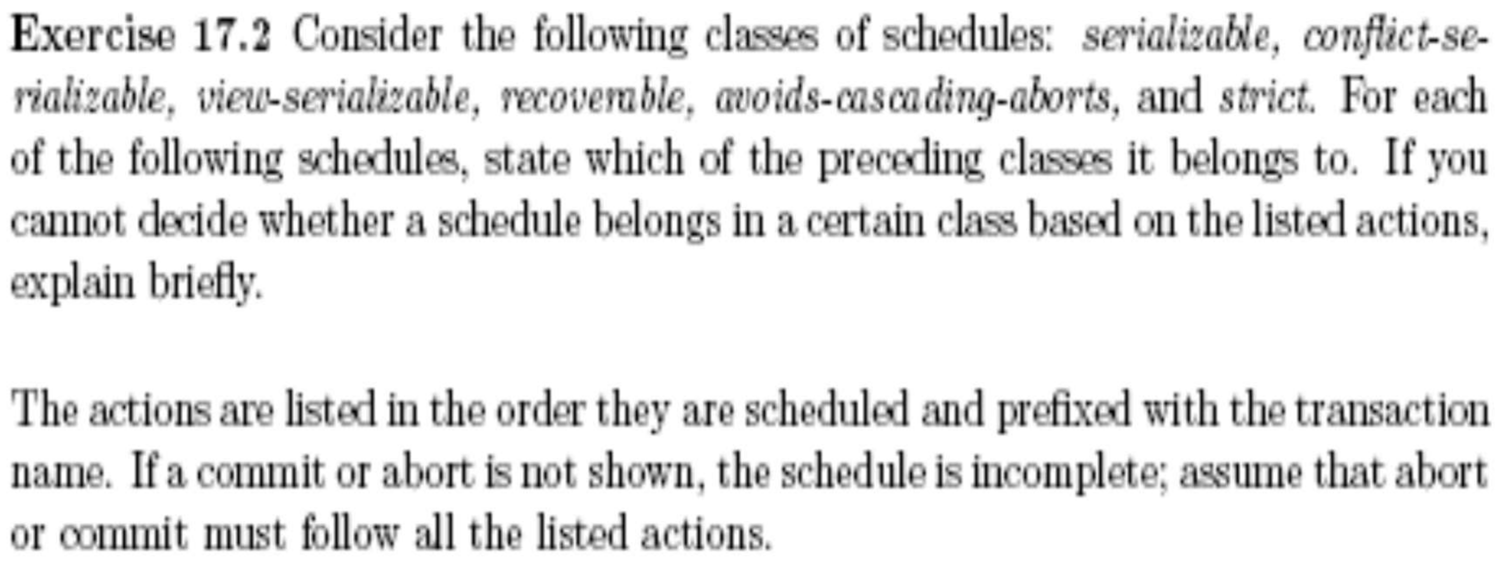
Serial Schedule

Transaction Support in SQL-92

- Each transaction has an access mode, a diagnostics size, and an isolation level.

Isolation Level	Dirty Read	Unrepeatable Read	Phantom Problem
Read Uncommitted	Maybe	Maybe	Maybe
Read Committed	No	Maybe	Maybe
Repeatable Reads	No	No	Maybe
Serializable	No	No	No

Examples



Exercise 17.2 Consider the following classes of schedules: *serializable*, *conflict-serializable*, *view-serializable*, *recoverable*, *avoids-cascading-aborts*, and *strict*. For each of the following schedules, state which of the preceding classes it belongs to. If you cannot decide whether a schedule belongs in a certain class based on the listed actions, explain briefly.

The actions are listed in the order they are scheduled and prefixed with the transaction name. If a commit or abort is not shown, the schedule is incomplete; assume that abort or commit must follow all the listed actions.

T1:W(X), T2:R(Y), T1:R(Y), T2:R(X)

Example

- **T1:W(X), T2:R(Y), T1:R(Y), T2:R(X)**
- Is it conflict serializable, view serializable, serializable, recoverable, avoids cascading aborts, strict?
  - YES! conflict serializable
  - YES! view serializable
  - YES! serializable
  - DO NOT KNOW! recoverable
  - NO! avoids cascading aborts
  - NO! strict