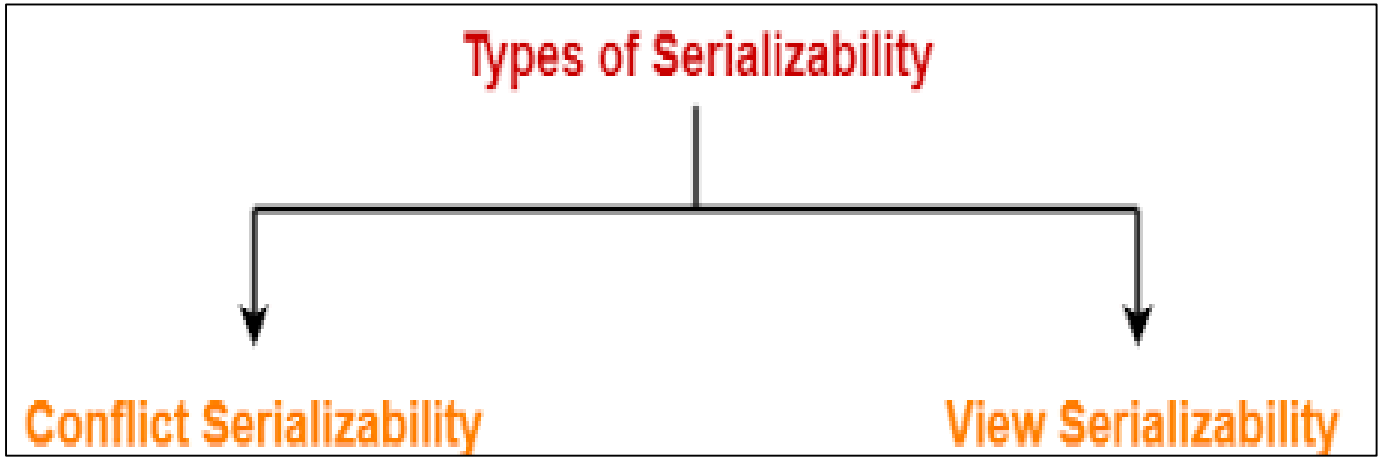
GLA UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)

Accredited with A Grade by NAAC

12-B Status from UGC

Database Management System (BCSC – 0003)

Topic: DBMS Serializability



Nikhil Govil

Assistant Professor, Dept. of CEA, GLA University, Mathura.

Topics to be covered

- Serializability
- Serializability Schedule
- Types of Serializability
- Recoverability of Schedule

Serializability

- When multiple transactions are running concurrently then there is a possibility that the database may be left in an inconsistent state.

- Serializability is a concept that helps us to check which schedules are serializable.

- A serializable schedule is the one that always leaves the database in consistent state.

Serializable Schedule

- A serializable schedule always leaves the database in consistent state.
- A serial schedule is always a serializable schedule because in serial schedule, a transaction only starts when the other transaction finished execution.
- However a non-serial schedule needs to be checked for Serializability.
- A serial schedule doesn’t allow concurrency, only one transaction executes at a time and the other starts when the already running transaction finished.

Types of Serializability

There are two types of Serializability.

1. Conflict Serializability &

2. View Serializability

Conflict Serializability

- A schedule is called conflict serializable if we can convert it into a serial schedule after swapping its non-conflicting operations.

Conflicting operations:

- Two operations are said to be in conflict, if they satisfy all the following three conditions:

1. Both the operations should belong to different transactions.
2. Both the operations are working on same data item.
3. At least one of the operation is a write operation.

Conflict Serializability

Lets see some examples to understand this:

Exp 1: Operation W(X) of transaction T1 and operation R(X) of transaction T2 are conflicting operations, because they satisfy all the three conditions mentioned above. They belong to different transactions, they are working on same data item X, one of the operation in write operation.

Exp 2: Operations W(X) of T1 & W(X) of T2 are conflicting operations.

Exp 3: Operations W(X) of T1 and W(Y) of T2 are non-conflicting operations because both the write operations are not working on same data item so these operations don't satisfy the second condition.

Exp 4: R(X) of T1 and R(X) of T2 are non-conflicting operations because none of them is write operation.

Exp 5: W(X) of T1 and R(X) of T1 are non-conflicting operations because both the operations belong to same transaction T1.

View Serializability

- View Serializability is a process to find out that a given schedule is view serializable or not.

- To check whether a given schedule is view serializable, we need to check whether the given schedule is View Equivalent to its serial schedule.

- A schedule will view serializable if it is view equivalent to a serial schedule.

View Serializability

View Equivalent

- Two schedules S1 and S2 are said to be view equivalent if they satisfy the following conditions:

1. Initial Read: An initial read of both schedules must be the same. Suppose two schedule S1 and S2. In schedule S1, if a transaction T1 is reading the data item A, then in S2, transaction T1 should also read A.

2. Update Read: If in schedule S1, the transaction T1 is reading a data item updated by T2 then in schedule S2, T1 should read the value after the write operation of T2 on same data item.

3. Final write: A final write must be the same between both the schedules. In schedule S1, if a transaction T1 updates A at last then in S2, final writes operations should also be done by T1.

View Serializability

For Example:

T1	T2
Read(A)	
	Write(A)

Schedule S1

T1	T2
	Write(A)
Read(A)	

Schedule S2

Above two schedules are view equivalent because Initial read operation in S1 is done by T1 and in S2 it is also done by T1.

Recoverability of Schedule

- Sometimes a transaction may not execute completely due to a software issue, system crash or hardware failure.
- In that case, the failed transaction has to be rollback.
- But some other transaction may also have used value produced by the failed transaction.
- So we also have to rollback those transactions.

References

- Korth, Silbertz and Sudarshan (1998), “Database Concepts”, 4th Edition, TMH.
- Elmasri and Navathe (2010), “Fundamentals of Database Systems”, 5th Edition, Addision Wesley.
- Date C J,” An Introduction to Database Systems”, 8th Edition, Addision Wesley.
- M. Tamer Oezsu, Patrick Valduriez (2011). “Principles of Distributed Database Systems”, 2nd Edition, Prentice Hall.
- https://www.javatpoint.com/dbms-file-organization/ last accessed on 18 September' 2021.


Thank
you