17.1 Suppose that there is a database system that never fails. Is a recovery manager required for this system?

Even in this case the recovery manager is needed to perform roll-back of aborted transactions.

17.3 Database-system implementers have paid much more attention to the ACID properties than have file-system implementers. Why might this be the case?

Database systems usually perform crucial tasks whose effects need to be atomic and durable, and whose outcome affects the real world in a permanent manner. Examples of such tasks are monetary transactions, seat bookings etc. Hence the ACID properties have to be ensured.

17.15 Consider the following two transactions:

T13:

read(A);

read(B);

if A = 0 then B := B + 1;

write(B).

T14:

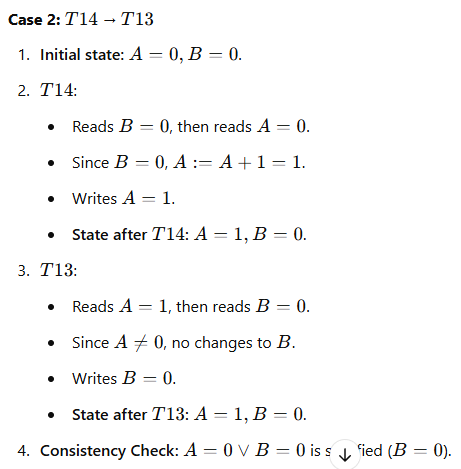
read(B);

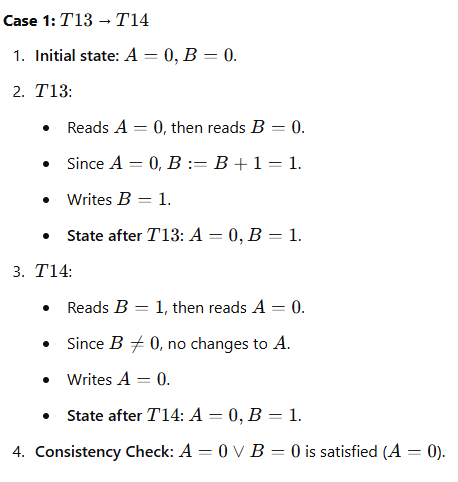
read(A);

if B = 0 then A := A + 1;

write(A).

Let the consistency requirement be A = 0 ∨ B = 0, with A = B = 0 as the initial values.

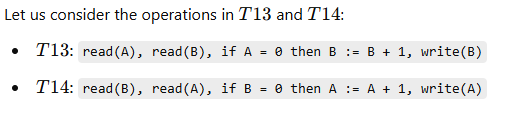
a. Show that every serial execution involving these two transactions preserves the consistency of the database.

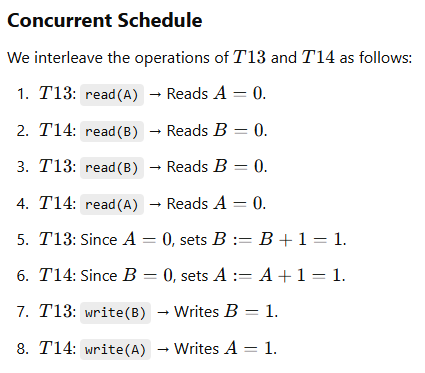


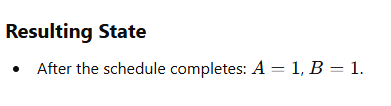
Conclusion

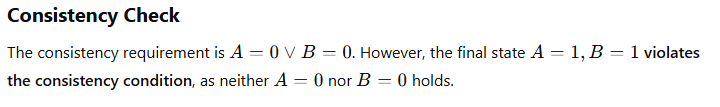
In both serial executions, the consistency requirement A=0∨B=0 is preserved. This shows that every serial execution involving these transactions maintains the consistency of the database.

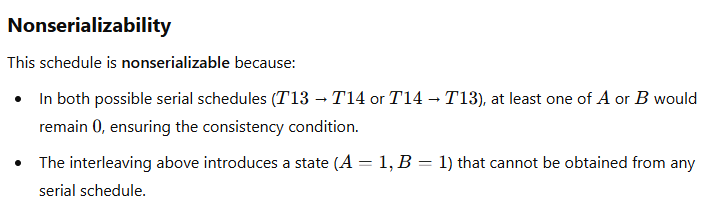
b. Show a concurrent execution of T13 and T14 that produces a nonserializable schedule.











c. Is there a concurrent execution of T13 and T14 that produces a serializable schedule?

For part **(c)**, the answer is **no**, there is no concurrent execution of T13 and T14 that produces a **serializable schedule,** There is no concurrent execution of T13 andT14 that pr**serializable schedule**.

17.16 Give an example of a serializable schedule with two transactions such that the order in which the transactions commit is different from the serialization order.

