|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | |
| final design | **Course:** | **Advance Database Concepts** | **Course Code:** | **CS451** |
| **Program:** | **BS(Computer Science)** | **Semester:** | **Spring 2019** |
| **Date:** | **31-Jan-2019** | **Total Marks:** | **10** |
| **Quiz** | **1 (Transactions)** | **Weight:** |  |
| **Section** | **CS** | **Max. Time:** | **15 Minutes** |
|  |  |  |  |
|  |  | | | |

**Q:** Consider the following transaction schedule, where time increases from left to right:

T1: R(A) W(B) Commit

T2: R(B) R(A) Commit

T3: R(B) W(A) Commit

**a)** Draw the precedence graph for this schedule. State whether this schedule is (conflict) serializable or not. If the schedule is serializable, write down the equivalent serial schedule(s) otherwise explain why it is not.

**b)** Determine whether schedule is strict, cascadeless, recoverable, or nonrecoverable. (Determine the strictest recoverability condition that schedule satisfies.)

**c)** State whether this schedule is view serializable or not. If the schedule is view serializable, write down the equivalent serial schedule(s) otherwise explain why it is not.

**Ans:**

**a) Conflict operations: T2🡪T1(B); T3🡪T1(B); T1 🡪T3(A); T2 🡪T3(A).**

**Not conflict-serializable as cycle exist; T1🡪T3🡪T1.**

**b) Strict Schedule**

**c) Not view-serializable schedule.**