|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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:** | **14-Feb-2019** | **Total Marks:** | **10** |
| **Quiz** | **2 (CCT)** | **Weight:** |  |
| **Section** | **CS** | **Max. Time:** | **10 Minutes** |
|  |  |  |  |
|  |  | | | |

**Q:** Consider the following schedule of actions, listed in the order they are submitted to the DBMS:

**Schedule S: S: r2(X); w3(X); w1(Y); r2(Y); r2(Z); r3(Y); c3; c2; r1(Z); c1**

For each of the following concurrency control mechanisms, describe how the concurrency control mechanism handles the schedule. Assume that the timestamp of transaction *Ti* is *i*. For lock-based concurrency control mechanisms, add lock and unlock requests to the above schedule of actions as per the locking protocol. The DBMS processes actions in the order shown. If a transaction is blocked, assume that all its actions are queued until it is resumed; the DBMS continues with the next action (according to the listed schedule) of an unblocked transaction.

1. Rigorous 2PL with timestamps used for deadlock avoidance (use wait-die policy)
2. Rigorous 2PL with deadlock detection. (Show the wait-for-graph in case of deadlock)

**Ans:**

**1) Rigorous 2PL** with timestamps used for deadlock avoidance (**using wait-die policy**)

Time

|  |  |  |
| --- | --- | --- |
| **Transaction T1** | **Transaction T2** | **Transaction T3** |
| x1-lock(Y)  w1(Y)  s1-lock(Z)  r1(Z)  **c1**, releases all locks  unlock(Y)  unlock(Z) | s2-lock(X)  r2(X)  s2-lock(Y) ... **abort** due to T1**,** releases all locks, & restart with same TS.  Unlock(X)  **T2** can be restart here. | x3-lock(X)... **abort** due to T2, releases all locks,& restart with same TS.  **T3** can be restart here. |

**2) Rigorous 2PL** with deadlock detection (Using wait-for-graph)

Time

|  |  |  |
| --- | --- | --- |
| **Transaction T1** | **Transaction T2** | **Transaction T3** |
| x1-lock(Y)  w1(Y)  s1-lock(Z)  r1(Z)  **c1**, releases all locks  unlock(Y)  unlock(Z) | s2-lock(X)  r2(X)  s2-lock(Y)…**wait** for T1 on Y  r2(Y)…**wake-up**  s2-lock(Z)  r2(Z)  **c2**, releases all locks  unlock(X)  unlock(Y)  unlock(Z) | x3-lock(X) .. **wait**  for T2 on X  w3(X) …**wake-up**  s3-lock(Y)  r3(Y)  **c3**, releases all locks  unlock(Y)  unlock(X) |