|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | |
| final design | **Course:** | **Advance Database Systems** | **Course Code:** | **CS451** |
| **Program:** | **BS(Computer Science)** | **Semester:** | **Spring 2019** |
| **Out Date:** | **12-Feb-2019** | **Total Marks:** |  |
| **Due Date:** | **Tue 19-Feb-2019 *(Start of Class)*** | **Weight:** |  |
| **Section** |  | **Page(s):** | **1** |
| **Assignment:** | **1 (Concurrency Control Techniques)** |  |  |
|  |  | | | |

*Instructions:*

* **This assignment is an individual assignment.**
* **Use proper assignment papers for solving your assignment questions. Assignment done on diary pages, register pages, rough pages will not be credited.**
* **Use any valid assumption where needed.**

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

**Schedule S1: r1(x), w2(y), w1(y), w3(z), c1, w2(z), w2(x), c2, w3(x), c3.**

**Schedule S2:** **r1(y), r1(z), w1(z), w2(z), r2(y), r3(x), w3(x), w1(x), c1, w2(y), r3(z), c2, c3.**

For each of the following concurrency control mechanisms, describe how the concurrency control mechanism handles the schedules.

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. Two-Phase Locking (2PL)
   1. Basic 2PL (use any protocol to deal with deadlock)
   2. Strict 2PL (use any protocol to deal with deadlock)
   3. Rigorous 2PL with protocol based on a timestamp for deadlock avoidance (use wait-die policy)
   4. Rigorous 2PL with protocol based on a timestamp for deadlock avoidance (use wound-wait policy)
   5. Rigorous 2PL with protocol based on a deadlock detection (Use wait-for-graph to deal with deadlock)
2. Timestamp Ordering (TO)
   1. Basic Timestamp Ordering
   2. Strict Timestamp Ordering
   3. Timestamp Ordering using Thomas’s Write Rule (TWR)
3. Multi-version Timestamp Ordering
4. Validation (Optimistic) Concurrency Control Technique (Use defer the validation until a later time when the conflicting transactions have finished)