Ceng352 - Database Management Systems Written Assignment 3

Furkan Doğan / 2098937

**1.**

**a.**

Yes, because schedules are 2PL.

**b.**

Deadlock is possible and T2 would roll back.

**c.**

Not possible because transactions follow Strict 2 PL.

**d.**

Yes, because schedules are 2 PL.

**e.**

Not possible because locks are acquired ordered.

**f.**

Yes, it is possible like below scenario.

X2(A) X2(B) R2(B) R2(A) W2(A) U2(A) X1(A) R1(A) W1(A) W2(B) U2(B)

At above scenario T1 is reading element A which written by T2 and not committed.

**2.**

**a.**

**i.**

TS(T1) = 1, TS(T2) = 2, TS(T3) = 3.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operation** | **A** | | | **B** | | | **C** | | |
| **-** | **RTS** | **WTS** | **C** | **RTS** | **WTS** | **C** | **RTS** | **WTS** | **C** |
| - | 0 | 0 | true | 0 | 0 | true | 0 | 0 | true |
| r1(A) | 1 | 0 | true | 0 | 0 | true | 0 | 0 | true |
| r2(B) | 1 | 0 | true | 2 | 0 | true | 0 | 0 | true |
| r3(A) | 3 | 0 | true | 2 | 0 | true | 0 | 0 | true |
| w1(A) ROLLBACK | 3 | 0 | true | 2 | 0 | true | 0 | 0 | true |
| r2(C) | 3 | 0 | true | 2 | 0 | true | 2 | 0 | true |
| w3(B) | 3 | 0 | true | 2 | 3 | false | 2 | 0 | true |
| w2(C) | 3 | 0 | true | 2 | 3 | false | 2 | 2 | false |
| c1 | 3 | 0 | true | 2 | 3 | false | 2 | 2 | false |
| r2(A) | 3 | 0 | true | 2 | 3 | false | 2 | 2 | false |
| w3(C) | 3 | 0 | true | 2 | 3 | false | 2 | 3 | false |
| c3 | 3 | 0 | true | 2 | 3 | true | 2 | 3 | true |
| w2(B) IGNORED | 3 | 0 | true | 2 | 3 | true | 2 | 3 | true |
| c2 | 3 | 0 | true | 2 | 3 | true | 2 | 3 | true |

**ii.**

TS(T1)=2, TS(T2)=3, TS(T3)=1.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operation** | **A** | | | **B** | | | **C** | | |
| **-** | **RTS** | **WTS** | **C** | **RTS** | **WTS** | **C** | **RTS** | **WTS** | **C** |
| - | 0 | 0 | true | 0 | 0 | true | 0 | 0 | true |
| r1(A) | 2 | 0 | true | 0 | 0 | true | 0 | 0 | true |
| r2(B) | 2 | 0 | true | 3 | 0 | true | 0 | 0 | true |
| r3(A) | 2 | 0 | true | 3 | 0 | true | 0 | 0 | true |
| w1(A) | 2 | 2 | false | 3 | 0 | true | 0 | 0 | true |
| r2(C) | 2 | 2 | false | 3 | 0 | true | 3 | 0 | true |
| w3(B) ROLLBACK | 2 | 2 | false | 3 | 0 | true | 3 | 0 | true |
| w2(C) | 2 | 2 | false | 3 | 0 | true | 3 | 3 | false |
| c1 | 2 | 2 | true | 3 | 0 | true | 3 | 3 | false |
| r2(A) | 3 | 2 | true | 3 | 0 | true | 3 | 3 | false |
| w3(C) ROLLBACK | 3 | 2 | true | 3 | 0 | true | 3 | 3 | false |
| c3 | 3 | 2 | true | 3 | 0 | true | 3 | 3 | false |
| w2(B) | 3 | 2 | true | 3 | 3 | false | 3 | 3 | false |
| c2 | 3 | 2 | true | 3 | 3 | true | 3 | 3 | true |

**b.**

Schedule avoids cascading aborts if whenever a transaction reads an element, then the transaction that wrote it must have already committed and with the help of commit bit we are keep track if the transaction that last wrote X has committed.

**3.**

**a. (JUST BEFORE CRASH)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **LSN** | **transID** | **prevLSN** | **type** | **pageID** | **log entry** | **undoNextLSN** |
| 0 | T1 | - | Update | P1 | Write(A->A0) |  |
| 1 | T2 | - | Update | P1 | Write(B->B1) |  |
| 2 | T2 | 1 | Update | P2 | Write(C->C2) |  |
| 3 | T2 | 2 | Abort | - | - | - |
| 4 | - | - | - | - | CHECKPOINT  (XACT TABLE=[[T1,0],[T2,3]]; DPT=[[P1,1],[P2,2]]) | - |
| 5 | T2 | - | CLR | - | Undo T2 LSN 2 | 1 |
| 6 | T2 | - | CLR | - | Undo T2 LSN 1 | - |
| 7 | T2 | 6 | End | - | - | - |
| 8 | T3 | - | Update | P2 | Write(D->D8) |  |
| 9 | T1 | 0 | Commit | - | - | - |
| 10 | T1 | 9 | End | - | - | - |
| 11 | T4 | - | Update | P1 | Write(A0->A11) |  |
| 12 | T3 | 8 | Update | P1 | Write(B->B12) | - |
| 13 | T4 | 11 | Commit | - | - | - |

|  |  |
| --- | --- |
| pageId | recLSN |
| P1 | 1 |
| P2 | 2 |

|  |  |  |
| --- | --- | --- |
| **transID** | **lastLSN** | **status** |
| T3 | 12 | Unknown |
| T4 | 13 | Committed |

**Active Transaction Table Dirty Page Table**

**Pages in Memory:**

P1: A has value A11, B has value B12. The pageLSN is 12.

P2: C has its initial value, D has value D8. The pageLsn is 8.

**b. (ANALYSIS)**

T4 committed before crush. So, we have all logs in the disk we start analyzing from checkpoint with LSN4.

|  |  |
| --- | --- |
| pageId | recLSN |
| P1 | 10 |
| P2 | 20 |

|  |  |  |
| --- | --- | --- |
| **transID** | **lastLSN** | **status** |
| T3 | 12 | Unknown |
| T4 | 13 | Committed |

**Active Transaction Table Dirty Page Table**

**c. (REDO)**

The REDO phase starts at the firstLSN, which is the smallest LSN in the Dirty Page Table. In our case its 1.

* LSN 1: No need to redo. P1 pageLSN is 1.
* LSN 2: No need to redo. P2 pageLSN is 2.
* LSN 3: redone.
* LSN 3: skipped. It’s for checkpoint.
* LSN 4: skipped. It’s for checkpoint.
* LSN 5: redone.
* LSN 6: redone.
* LSN 7: skipped.
* LSN 8: redone.
* LSN 9: redone.
* LSN 10: skipped.
* LSN 11: redone.
* LSN 12: redone.
* LSN 13: skipped

|  |  |
| --- | --- |
| pageId | recLSN |
| P1 | 1 |
| P2 | 2 |

|  |  |  |
| --- | --- | --- |
| **transID** | **lastLSN** | **status** |
| T3 | 12 | Unknown |
| T4 | 13 | Committed |

**Active Transaction Table Dirty Page Table**

**Pages in Memory:**

P1: A has value A2, B has value B2. The pageLSN is 12.

P2: C has its initial value, D has value D1. The pageLsn is 8.

**d. (UNDO)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| LSN | transId | prevLSN | Type | pageID | Log entry | undoNextLSN |
| 0 | T1 | - | Update | P1 | Write(A->A0) |  |
| 1 | T2 | - | Update | P1 | Write(B->B1) |  |
| 2 | T2 | 1 | Update | P2 | Write(C->C2) |  |
| 3 | T2 | 2 | Abort | - | - | - |
| 4 | - | - | - | - | CHECKPOINT  (XACT TABLE=[[T1,0],[T2,3]]; DPT=[[P1,1],[P2,2]]) | - |
| 5 | T2 | - | CLR | - | Undo T2 LSN 2 | 1 |
| 6 | T2 | - | CLR | - | Undo T2 LSN 1 | - |
| 7 | T2 | 6 | End | - | - | - |
| 8 | T3 | - | Update | P2 | Write(D->D8) |  |
| 9 | T1 | 0 | Commit | - | - | - |
| 10 | T1 | 9 | End | - | - | - |
| 11 | T4 | - | Update | P1 | Write(A0->A11) |  |
| 12 | T3 | 8 | Update | P1 | Write(B->B12) | - |
| 13 | T4 | 11 | Commit | - | - | - |
| 14 | T4 | 13 | End | - | - | - |
| 15 | T3 | - | CLR | - | Undo T3 LSN 12 | 8 |
| 16 | T3 | - | CLR | - | Undo T3 LSN 8 | - |
| 17 | T3 | 16 | End | - | - | - |

**Pages in Memory:**

P1: A has value A2, B has value B. The pageLSN is 15.

P2: C has its initial value C, D has its initial value D. The pageLsn is 16.