**COMP / IT 420 Database Theory and Design Spring 2020**

**Homework 4: Concurrency**

OPEN: Wednesday, April 7th, 2021 9:00am

DUE: Tuesday, April 21st, 2021 11:59pm

**Introduction**

This lab will focus on numerous aspects of concurrency control including timestamp ordering, deadlock detection and prevention and multi-version concurrency control.

**Section 1: Timestamping (20pts)**

|  |  |  |
| --- | --- | --- |
| *t* | T1, TS(T1) = 1 | T2, TS(T2) = 2 |
| 1 | Read(A) |  |
| 2 | Read(C) |  |
| 3 |  | Read(B) |
| 4 |  | Read(C) |
| 5 |  | Write(C) |
| 6 |  | Read(A) |
| 7 | Write(C) |  |
| 8 |  | Write(B) |

1. Assuming a simple timestamp ordering protocol which ignores locks, fill in the following timestamp table with the appropriate values keyed to time index (*t*) from the table above. (8pts)

|  |  |  |  |
| --- | --- | --- | --- |
| *t* | Object | R-TS | W-TS |
| 1 | A | 1 | 0 |
| 2 | C | 1 | 0 |
| 3 | B | 2 | 0 |
| 4 | C | 2 | 0 |
| 5 | C | 2 | 2 |
| 6 | A | 2 | 0 |
| 7 | X |  |  |
| 8 | B | 2 | 2 |
|  |  |  |  |

1. Assuming a wait / die scheme and locking, answer the following questions about the table below.

|  |  |  |
| --- | --- | --- |
| *t* | T1, TS(T1) = 1 | T2, TS(T2) = 2 |
| 1 | Lock (A) |  |
| 2 | Read (A) |  |
| 3 |  | Lock(B) |
| 4 |  | Read(B) |
| 5 | Lock(B) |  |
| 6 | Read(B) |  |
| 7 |  | Unlock(B) |
| 8 |  | Lock(A) |

1. At what time index(es), if any, will a rollback be triggered for either transaction? (3pts)

**T2 at t = 8 due to lock on A by T1**

1. At what time index will T1 acquire a lock on B? (3pts)

**T = 7 (or 8) due to waiting for T2**

1. Assuming a wound / wait scheme and locking, answer the following questions about the table below.

|  |  |  |
| --- | --- | --- |
| *t* | T1, TS(T1) = 1 | T2, TS(T2) = 2 |
| 1 | Lock (A) |  |
| 2 | Read (A) |  |
| 3 |  | Lock(B) |
| 4 |  | Read(B) |
| 5 | Lock(B) |  |
| 6 | Read(B) |  |
| 7 |  | Unlock(B) |
| 8 |  | Lock(A) |

1. At what time index(es), if any, will a rollback be triggered for either transaction? (3pts)

**T2 at t = 5 due to pre-emption by T1**

1. At what time index will T1 acquire a lock on B? (3pts)

**T1 at t = 5**

**Section 2: Multi-Version Concurrency Control (20pts)**

Follow the guidelines in sections 15.6.1 and 15.6.2 in the posted Database System Concepts 6th edition Chapter 15 on Concurrency.

Given the following transactions, assume:

1. That each write operation is changing the value of its object.
2. Versioning starts with 0 and increments by one.
3. That objects A and B were written in a previous transaction with timestamp 1.

|  |  |  |
| --- | --- | --- |
| *t* | T2, TS(T2) = 2 | T2, TS(T3) = 3 |
| 1 | Write(A) |  |
| 2 | Read(A) |  |
| 3 |  | Read(A) |
| 4 |  | Write(B) |
| 5 | Read(B) |  |
| 6 |  | Write(A) |
| 7 | Read(A) |  |
| 8 |  | Read(B) |

1. Fill in the following table with the correct information regarding the state of each object: (Assume multi-version timestamp-ordering and a declaration of *Read Uncommitted* for T2 and T3.) (10pts)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *t* | Object | Version | W-TS | R-TS |
| 0 | A | 0 | 1 | 1 |
| 0 | B | 0 | 1 | 1 |
| 1 | A | 1 | 2 | 2 |
| 2 | A | 1 | 2 | 2 |
| 3 | A | 1 | 2 | 3 |
| 4 | B | 1 | 3 | 3 |
| 5 | B | 0 | 1 | 2 |
| 6 | A | 2 | 3 | 3 |
| 7 | A | 1 | 2 | 3 |
| 8 | B | 1 | 3 | 3 |

1. Which object and version will be read at each Read timestep? (5pts)

**2: A1, 3:A1, 5:B0, 7:A1, 8:B1**

1. Assuming that each transaction is declared Serializable and strict 2PL is used for concurrency:
   1. At which time indexes will T2 need to wait? ("None" is a possible answer.) (2pts) **None**
   2. At which time indexes will T3 need to wait? ("None" is a possible answer.) (3pts) **T = 6**