# COMP 3311: Database Management Systems

## Lecture 21 Exercises
## Concurrency Control: Timestamp-based Protocols

**Exercise 1:** Use the single version, timestamp-ordering protocol to complete the following non-serializable schedule assuming the timestamps 1, 2, and 3 for transactions $T_1$, $T_2$, and $T_3$, respectively. Show where the protocol will fail. Assume initial R/W timestamp of all items is 0.

| X | |
|---|---|
| RTS(X) = | WTS(X) = |

| Y | |
|---|---|
| RTS(Y) = | WTS(Y) = |

| Z | |
|---|---|
| RTS(Z) = | WTS(Z) = |

| $T_1$ [TS=1] | $T_2$ [TS=2] | $T_3$ [TS=3] |
|---|---|---|
| read(X) | | |
| | read(Y) | |
| | write(Y) | |
| | | write(Z) |
| write(X) | | |
| | read(X) | |
| | write(X) | |
| | | read(Y) |
| | | write(Y) |
| write(Z) | | |

**Exercise 2:** Use the multi-version, timestamp-ordering protocol to complete the schedule of Exercise 1 assuming the timestamps 1, 2, and 3 for transactions $T_1$, $T_2$, and $T_3$, respectively. Assume initial R/W timestamp of $X_0$, $Y_0$ and $Z_0$ is 0.

| X | |
|---|---|
| $RTS(X_0) = 0$ | $WTS(X_0) = 0$ |
| $RTS(X_1) =$ | $WTS(X_1) =$ |
| $RTS(X_2) =$ | $WTS(X_2) =$ |

| Y | |
|---|---|
| $RTS(Y_0) = 0$ | $WTS(Y_0) = 0$ |
| $RTS(Y_1) =$ | $WTS(Y_1) =$ |
| $RTS(Y_2) =$ | $WTS(Y_2) =$ |

| Z | |
|---|---|
| $RTS(Z_0) = 0$ | $WTS(Z_0) = 0$ |
| $RTS(Z_1) =$ | $WTS(Z_1) =$ |
| $RTS(Z_2) =$ | $WTS(Z_2) =$ |

| $T_1$ [TS=1] | $T_2$ [TS=2] | $T_3$ [TS=3] |
|---|---|---|
| read(X) | | |
| | read(Y) | |
| | write(Y) | |
| | | write(Z) |
| write(X) | | |
| | read(X) | |
| | write(X) | |
| | | read(Y) |
| | | write(Y) |
| write(Z) | | |

**Do not upload this exercise sheet to Canvas.**

**Name:** _____/_____ **Student#:** _____ **Date:** _____
Family/Last (PRINT) Given/First (PRINT)

# COMP 3311: Database Management Systems

## Lecture 21 Exercises
## Concurrency Control: Timestamp-based Protocols

**Exercise 3:** The following schedule is conflict serializable.

(a) What is the equivalent serial schedule? _____

(b) Assign appropriate timestamps to the transactions $T_1$, $T_2$, $T_3$ and $T_4$ so that the schedule is conflict serializable according to the single version, timestamp-ordering protocol. Assume initial R/W timestamp of all items is 0.

| $T_1$ [TS= ] | $T_2$ [TS= ] | $T_3$ [TS= ] | $T_4$ [TS= ] |
|---|---|---|---|
| read(X) | | | |
| write(X) | | | |
| | read(X) | | |
| | | read(Y) | |
| | | write(Y) | |
| | write(X) | | |
| | | | read(Y) |
| write(Y) | | | |

**Exercise 4:** Use the multi version, timestamp-ordering protocol to complete the conflict serializable schedule of Exercise 3 assuming the timestamps 1, 2, 3, and 4 for transactions $T_1$, $T_2$, $T_3$ and $T_4$, respectively. Show where the protocol will fail. Assume initial R/W timestamp of all items is 0.

| X | |
|---|---|
| $RTS(X_0) = 0$ | $WTS(X_0) = 0$ |
| $RTS(X_1) =$ | $WTS(X_1) =$ |
| $RTS(X_2) =$ | $WTS(X_2) =$ |

| Y | |
|---|---|
| $RTS(Y_0) = 0$ | $WTS(Y_0) = 0$ |
| $RTS(Y_1) =$ | $WTS(Y_1) =$ |
| $RTS(Y_2) =$ | $WTS(Y_2) =$ |

| $T_1$ [TS=1] | $T_2$ [TS=2] | $T_3$ [TS=3] | $T_4$ [TS=4] |
|---|---|---|---|
| read(X) | | | |
| write(X) | | | |
| | read(X) | | |
| | | read(Y) | |
| | | write(Y) | |
| | write(X) | | |
| | | | read(Y) |
| write(Y) | | | |

You must upload this completed exercise sheet to Canvas by **11 p.m. today.**