# COMP 306: Database Management Systems
# Fall 2023 - Exercise Questions

**Question 1.** Consider a DBMS that uses 2PL. You are given the following transactions with their intended reads and writes on objects A, B, C. Does there exist a schedule in which these transactions would be deadlocked? If so, draw the schedule, show all lock and unlock events, and explain briefly. If not, explain why a deadlock is not possible.

$T_1$: R(A), R(B), R(C)
$T_2$: W(B), R(C), W(A)

**Question 2.** Consider the following schedule with four transactions ($T_1$, $T_2$, $T_3$, $T_4$) and four database objects: $A$, $B$, $C$, $D$. There are two types of locks: S-locks and X-locks. Transactions do not release a granted lock unless they are killed or aborted. For simplicity, only lock-related events are shown in the schedule (reads and writes are not shown).

| time | $T_1$ | $T_2$ | $T_3$ | $T_4$ |
|---|---|---|---|---|
| $t_1$ | S-LOCK(C) | | | |
| $t_2$ | | X-LOCK(A) | | |
| $t_3$ | | | S-LOCK(D) | |
| $t_4$ | S-LOCK(A) | | | |
| $t_5$ | | | | X-LOCK(D) |
| $t_6$ | | | X-LOCK(B) | |
| $t_7$ | | | X-LOCK(C) | |
| $t_8$ | | X-LOCK(B) | | |

(a) Draw the waits-for graph of this schedule with all necessary edges. On each edge, write which database object is causing that edge.

(b) Does this schedule contain a deadlock? Why or why not?

(c) Now consider that the DBMS is using the Wait-Die policy for deadlock prevention. Assume that $T_1$ started first, then $T_2$, then $T_3$, then $T_4$. Also assume all transactions automatically release their locks if they die or abort. Explain what action is taken by the DBMS at each timestamp ($t_1$, $t_2$, $t_3$, ..., $t_8$). Justify briefly when necessary.