

Review Guide 5: Transactions

1. Explain the distinction between a *serial schedule* and a *serializable schedule*.
2. What is a recoverable schedule? Why is recoverability of schedules desirable? Are there any circumstances under which it would be desirable to allow nonrecoverable schedules? Explain your answer.
3. Consider the following three transactions,

$$\begin{aligned}T_1 &: r_1(B); r_1(A); c_1; \\T_2 &: r_2(A); w_2(B); c_2; \\T_3 &: r_3(A); w_2(A); c_3;\end{aligned}$$

and their schedule, $S : r_3(A); r_2(A); r_1(B); w_2(B); r_1(A); w_2(A); c_2; c_1; c_3;$

- (a) Give the precedence graph for S , and determine whether S is conflict serializable. If S is conflict serializable, give an equivalent serial ordering.
 - (b) Is S possible using (basic) 2PL? If so, indicate in the schedule where locks can be acquired and released by T_1, T_2, T_3 . If not, explain why a 2PL schedule is not possible.
4. Consider a database with objects X and Y and assume that there are two transactions T_1 and T_2 :
$$\begin{aligned}T_1 &: r_1(X); r_1(Y); w_1(X); c_1; \\T_2 &: r_2(X); r_2(Y); w_2(X); w_2(Y); c_2;\end{aligned}$$
 - (a) Give an example nonserial schedule with actions of transactions T_1 and T_2 on objects X and Y that results in a RAW conflict.
 - (b) Give an example nonserial schedule with actions of transactions T_1 and T_2 on objects X and Y that results in a WAR conflict.
 - (c) Give an example nonserial schedule with actions of transactions T_1 and T_2 on objects X and Y that results in a WAW conflict.
 - (d) For each of the three schedules, show that Strict 2PL disallows the schedule.
 5. What benefit does *Strict 2PL* provide? What disadvantages result?