Assignment 4

Q1[4]. Create the precedence graph for the schedule: w1(X), r1(X), w2(Y), r3(X), w1(Y), w1(X). Based on the graph, determine whether the schedule is conflict-serializable.

Q2[4]. What is the difference between serializable and conflict-serializable schedule. Why are they not the same thing?

Q3[4]. Consider the following schedule:

r1(A);r2(B);???;w1(C);w2(A);

What value for ??? will make the schedule not conflict-serializable. Give one such possible value.

Q4[4]. Consider the transactions T_1 and T_2.

$T_1 : r_1(A); w_1(A); r_1(B); w_1(B);$

$T_2 : r_2(B); w_2(B); r_2(A); w_2(A);$

Give an example of a conflict-serializable schedule and non conflict-serializable schedule for the eight actions.

Q5[4]. Show how a scheduler that follows the two-phase-locking protocol will work if the following operations come in this order: $r_1(A)$; $w_1(\mathrm{B})$; $r_2(\mathrm{B})$; $w_2(\mathrm{C})$; $r_3(\mathrm{C})$; $w_3(\mathrm{A})$;

Q6[4]. Are the following schedules conflict-serializable?

a) $r_1(A)$; $w_1(\mathrm{B})$; $r_2(\mathrm{B})$; $w_2(\mathrm{C})$; $r_3(\mathrm{C})$; $w_3(\mathrm{A})$;

b) $w_3(A)$; $r_1(\mathrm{A})$; $w_1(\mathrm{B})$; $r_2(\mathrm{B})$; $w_2(\mathrm{C})$; $r_3(\mathrm{C})$;

Q7[8]. Create two triggers that work with the SQL tables from the labs. Add the first trigger to the **transaction** table. The idea is that every time a tuple is inserted in the transaction table, the current balance of the credit card that is involved in the transaction should be incremented by the dollar amount of the transaction.

Add a second trigger to the **credit card** table. If a **credit card** tuple is updated, then the trigger should make sure that the new credit card balance is less than the new credit limit. If it is not, then you should signal the exception 12345.

Make sure your triggers work on the MySQL database. Submit only the code for the two triggers.