CS143: Database Systems

Homework #7

1.(b)
switch: r1(A),c3
switch: r4(B),c2
results:
w3(A)c3r1(A)w1(B)c1r2(B)w2(C)c2r4(B)c4

1. Consider the following schedule:

$$w_3(A)r_1(A)c_3w_1(B)c_1r_2(B)w_2(C)r_4(B)c_2c_4$$

   (a) Is it a serial schedule? No

   (b) Is the schedule conflict serializable? If so, what are all the equivalent serial schedules? Yes

   (c) Is the schedule recoverable? If not, can we make it recoverable by moving a single commit operation to a different position? Yes

   (d) Is the schedule cascadeless? If not, can we make it cascadeless by moving a single commit operation to a different position? No, the schedule isn't cascadeless. But we can make it cascadeless by moving c3 to the second position to make it cascadeless.

2. Consider the relation `Employee(name, salary)` where `name` is the key. The following three transactions are being executed:

2000,2300,3400,3700. When all transactions run under SERIALIZABLE, any possible schedule is conflict equivalent to a serial schedule. Possible schedules for T1,T2 and T3 are: T1T2T3, T1T3T2, T2T1T3, T2T3T1, T3T1T2, and T3T2T1. The outputs from T1 are 2000, 2000, 3400, 3700, 2300, 3700, respectively.

T_1:

```
SELECT SUM(salary) FROM Employee;
COMMIT;
```

T_2:

```
UPDATE Employee SET salary = salary + 200;
UPDATE Employee SET salary = salary + 1000 WHERE name = 'Tony';
COMMIT;
```

T_3:

```
UPDATE Employee SET salary = salary + 100 WHERE name = 'James';
UPDATE Employee SET salary = salary + 200 WHERE name = 'Tony';
COMMIT;
```

The table `Employee` originally has two tuples, ('Tony', 1000) and ('James', 1000). Please assume that individual SQL statements are executed atomically.

   (a) Assume that all three transactions run under the isolation level SERIALIZABLE. List all possible values that can be returned by T_1. Briefly explain your answer.

   (b) Assume that T_1 runs under the isolation level READ UNCOMMITTED and T_2 under REPEATABLE READ and T_3 under SERIALIZABLE. List all possible values that can be returned by T_1. Briefly explain your answer.

2000,2100,2300,2400,2700,3400,3500,3700. Only T2 and T3 are updating values, so let us focus on these two transactions first. Under REPEATABLE READ, the only exception to ACID is phantom, but because T2 and T3 do not insert any tuple, we do not need to worry about ACID exceptions for the two. So the possible schedules for T2 and T3 are equivalent to T2T3 or T3T2. Regarding T1, since it is READ UNCOMMITTED, its SELECT statement may do a dirty read. Now let us consider the schedule T2T3. Under this schedule, the total salary value changes from 2000 —> 2400 —> 3400 —> 3500 —> 2700. T1 may read any of these salary sum values. For the schedule T3T2, the total salary value changes from 2000 —> 2100 —> 2300 —> 2700 —> 3700. Again, T1 may read any of these salary sum values. Therefore, possible outputs from T1 are 2000,2100,2300,2400,2700,3400,3500,3700.