Bookmark this page

Each multiple-choice quiz problem is based on a "root question," from which the system generates different correct and incorrect choices each time you take the quiz. Thus, you can test yourself on the same material multiple times. We strongly urge you to continue testing on each topic until you complete the quiz with a perfect score at least once. Simply click the "Reset" button at the bottom of the page for a new variant of the quiz.

After submitting your selections, the system will score your quiz, and for incorrect answers will provide an "explanation" (sometimes for correct ones too). These explanations should help you get the right answer the next time around. To prevent rapid-fire guessing, the system enforces a minimum of 10 minutes between each submission of solutions.

Multiple Choice

3/3 points (graded)

[Q1] Consider tables R(A) and S(B) and the following transaction T1:

```
T1: set transaction isolation level repeatable read;
    select * from R;
    select * from R;
    select * from S;
    commit;
```

Suppose table R initially has one tuple with value A=3 and table S initially has one tuple with value B=6. Consider the following possible transactions T2, executed around the same time as T1. Which one of them can cause the two transactions to exhibit nonserializable behavior?

- (●) T2: set transaction isolation level serializable; insert into R values (4); commit; ✔
- ( ) T2: set transaction isolation level serializable; insert into S values (5); delete from R where A=2; commit;
- ( ) T2: set transaction isolation level serializable; update R set A=4; update S set B=5 where B<5; commit;
- ( ) T2: set transaction isolation level serializable; update S set B=7; commit;

[Q2] Consider a relation R(x) containing integers. Suppose Alice runs a transaction that is a query:

```
   select sum(x) from R;
   commit;
```

Betty's transaction is a sequence of inserts:

```
   insert into R values (10);
   insert into R values (20);
   insert into R values (30);
   commit;
```

Carol's transaction is a sequence of deletes:

```
   delete from R where x=30;
   delete from R where x=20;
   commit;
```

Before any of these transactions execute, the sum of the integers in R is 1000, and none of the integers are 10, 20, or 30. If Alice's, Betty's, and Carol's transactions run at about the same time, and each runs under isolation level READ COMMITTED, which of the following sums could be produced by Alice's transaction?

- ( ) 1030
- ( ) 970
- ( ) 1020
- (●) 1010 ✔

[Q3] Consider a relation R(x) containing integers. Suppose Alice runs a transaction that is a query:

```
select sum(x) from R;
commit;
```

Betty's transaction is a sequence of inserts:

```
insert into R values (10);
insert into R values (20);
insert into R values (30);
commit;
```

Carol's transaction is a sequence of deletes:

```
delete from R where x=30;
delete from R where x=20;
commit;
```

Before any of these transactions execute, the sum of the integers in R is 1000, and none of these integers are 10, 20, or 30. Alice's, Betty's, and Carol's transactions run at about the same time. Which of the following sums could be returned by Alice's transaction if all three transactions run under isolation level READ UNCOMMITTED, but not if all three run under isolation level SERIALIZABLE?

- 1010
- 1060
- 1030 ✔
- 1020

Submit