



[< Previous](#)



[Next >](#)

Notifications



## Transactions Quiz

[🔖](#) Bookmark this page

### Pursue a verified certificate

- ✓ Earn a **verified certificate** of completion to showcase on your resumé
- ✓ Support our **mission** at edX

Upgrade for \$50



Hide Notes

Quiz due Apr 29, 2022 08:04 IST

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.

## Q1

1/1 point (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; update R set A=4; delete from S where B=6; commit;

☐ T2: set transaction isolation level serializable; update S set B=7; commit;

☐ T2: set transaction isolation level serializable; delete from R where A=3; commit;

☐ T2: set transaction isolation level serializable; update R set A=4; update S set B=5 where B<5; commit;



### Problem Explanation

Nonserializable behavior can be exhibited if transaction T2 performs inserts on R, or if T2 modifies both R and S. On inserts, repeatable reads allow "phantom" tuples to appear during a transaction. If both R and S are modified in T2, it is possible for T1 to read the state of R before T2 and the state of S after T2.

Submit

**i** Answers are displayed within the problem

## Q2

1/1 point (graded)

[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?

☐ 1050

☐ 1030

☒ 1040

☐ 1020



### Problem Explanation

Alice must see all of Betty's inserts or none of them. Alice could compute the sum before Betty or Carol does anything (answer 1000), or after Betty's inserts but before Carol's deletes (answer 1060). Alternatively, Carol could complete her deletes before Betty starts, in which case Carol has no effect (answer 1000 or 1060). Carol could complete her deletes after Betty's inserts, but before Alice reads, in which case Alice sees only 10 from among Betty's inserts (answer 1010). Since the isolation level is READ COMMITTED and not SERIALIZABLE, another possibility is that Carol starts before Betty, deleting 30 (which has no effect on  $R$ ), then Betty does her inserts and commits, then Carol deletes 20, leaving 10 and 30 in  $R$ . Finally, Alice executes (answer 1040). There are no other possibilities, so the possible sums are 1000, 1010, 1040, 1060.

Submit

**i** Answers are displayed within the problem

### Q3

1/1 point (graded)

[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?

☒ 1040

☐ 950

☐ 1000

☐ 1020



#### Problem Explanation

< Previous

Next >

---

# edX

- [About](#)
- [Affiliates](#)
- [edX for Business](#)
- [Open edX](#)
- [Careers](#)
- [News](#)

---

## Legal

- [Terms of Service & Honor Code](#)
- [Privacy Policy](#)
- [Accessibility Policy](#)
- [Trademark Policy](#)
- [Sitemap](#)

---

## Connect

- [Blog](#)
- [Contact Us](#)
- [Help Center](#)
- [Media Kit](#)



© 2022 edX Inc. All rights reserved.  
深圳市恒宇博科技有限公司 [粤ICP备17044299号-2](#)