

# ACID Properties

Ensuring Reliable Database Operations



r/Bitcoin · 11 yr. ago  
bitcointhief2

...

## How I stole roughly 100 BTC from an exchange and how I could have stolen more!

The reason I'm posting this is not so I can go encourage people to go steal bitcoins from exchanges but to show you how insecure services can be that you need to trust to hold your bitcoins and hopefully so that future bitcoin services do not make this same mistake. I will not post my identity nor will I post which exchange had this particular vulnerability. I gave all the coins I stole back and told the exchange operator how to fix this problem. I will say that this exchange is in the top 10 in terms of volume which is a scary thought.

How I stole roughly 100 bitcoins:

This exploit was very simple. All I did was make a bunch of small sell orders for 0.0001 bitcoins across a few of my own machines at the same time. At the same time I made a big withdraw for at least 100 coins. What happens is sometimes you get lucky and the trade and withdraw execute at the same time. When this happens both the trade and withdraw try to update your balance at the same time and what can happen is the 0.0001 bitcoins gets subtracted from your account balance BUT NOT THE WITHDRAW OF THE 100 BITCOINS! This exploit could have allowed me to withdraw bitcoins from the hot wallet all day. I alerted the exchange operator of this bug and it was quickly fixed by providing locking to accounts and only allowing one action to be performed from an account at a time.

Moral of the story, if you plan on using a bitcoin related service, do your business there and right away withdraw your bitcoins to your own wallet. Don't wait for the exchange to get hacked!

# Agenda

- The First SQL Database
- What is a Transaction?
- The ACID Properties
  - Atomicity
  - Consistency
  - Isolation
  - Durability
- Practical Examples

# The First SQL Database

## A History and Evaluation of System R

Donald D. Chamberlin  
Morton M. Astrahan  
Michael W. Blasgen  
James N. Gray  
W. Frank King  
Bruce G. Lindsay  
Raymond Lorie  
James W. Mehl

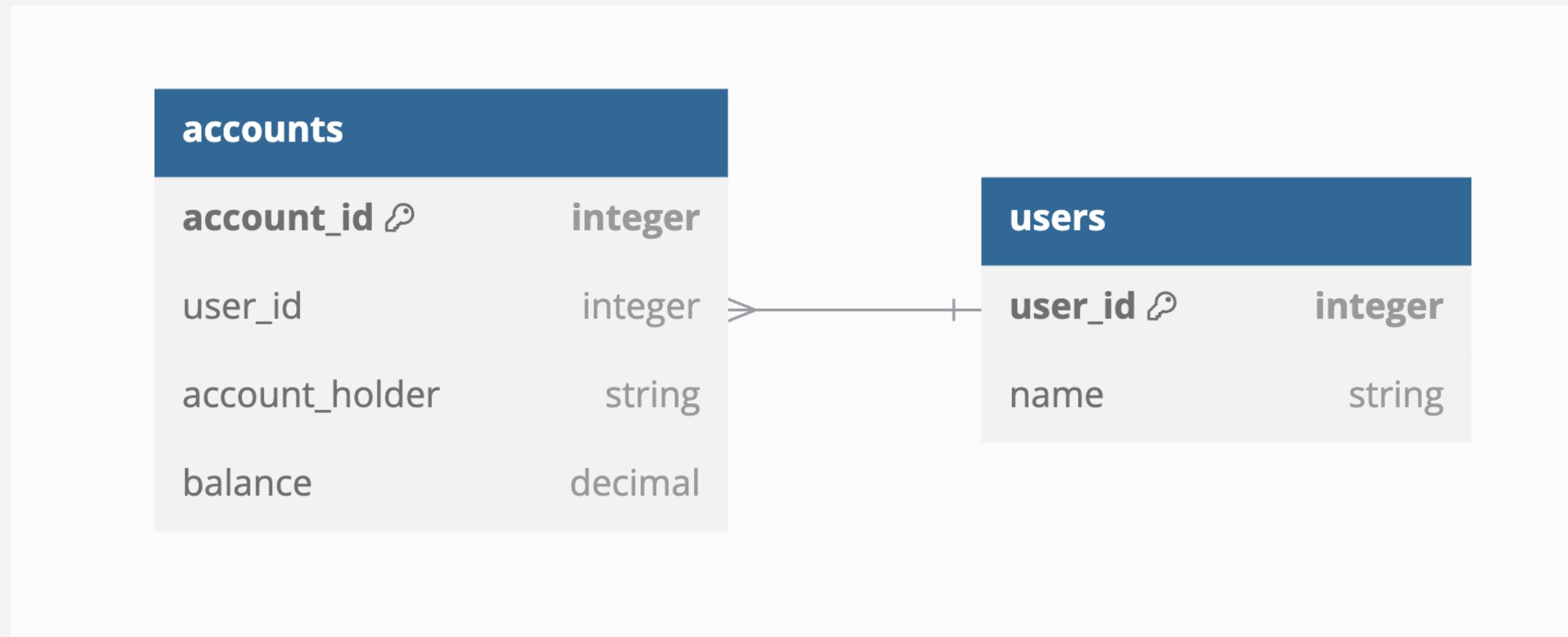
Thomas G. Price  
Franco Putzolu  
Patricia Griffiths Selinger  
Mario Schkolnick  
Donald R. Slutz  
Irving L. Traiger  
Bradford W. Wade  
Robert A. Yost

IBM Research Laboratory  
San Jose, California

```
SELECT MIN(PRICE)
FROM   PRICES
WHERE  PARTNO IN
       (SELECT PARTNO
        FROM   PARTS
        WHERE  NAME = 'BOLT');
```

# **What is a Transaction?**

# ERD



# Real Transaction

Transfer 50 SAR to another account

**begin transaction;**

**Queries & Commands**

**Queries & Commands**

**commit;**

**begin transaction;**

**Queries & Commands**

**Queries & Commands**

**rollback;**

# When Should We Use Transaction?

**When Have a Group of Commands**

**[Write]**

**When We Write or Modify the Data**

**[Read]**

**When We Generate an Extensive Reports**

**When Play With Something**

# ACID Properties

**Atomicity**

**Consistency**

**Isolation**

**Durability**

# ACID Properties

Atomicity

Consistency

Isolation

Durability



Theo Härdter



Andreas Reuter

These four properties, atomicity, consistency, isolation, and durability (ACID), describe the major highlights of the transaction paradigm, which has influenced many aspects of development in database systems. We therefore consider the question of whether the transaction is supported by a particular system to be the ACID test of the system's quality.

In summary, a transaction can terminate in the three ways illustrated in Figure 2. It is hoped that the transaction will reach its commit point, yielding the all case (as in the all-or-nothing dichotomy). Sometimes

Computing Surveys, Vol. 15, No. 4, December 1983

# Atomicity

Consistency

Isolation

Durability

# All or Nothing

Isolation

Durability

# Consistency

1. Defined by the business
2. Referential integrity

# Consistency

Defined by the user  
Referential integrity (foreign keys)

Spot the inconsistency in this data

Users Table

```
test_db=# select * from users;  
  
user_id | name  
-----+-----  
1 | Abdulrahman Mohammed  
2 | Mamdouh Aldhafeeri  
3 | Saud Elabdullah  
(3 rows)
```

Accounts Table

```
test_db=# select * from accounts;  
  
account_id | user_id | account_holder | Balance  
-----+-----+-----+-----  
1 | 1 | Abdulrahman | 500.00  
2 | 2 | Mamdouh | 300.00  
3 | 3 | Saud | 700.00  
3 | 10 | Khaled | 100000.00  
(4 rows)
```

# Isolation

- Concurrent transactions do not interfere with each other
- Isolation Levels

Durability

Isolation

levels

- Read uncommitted
- Read committed
- Repeatable read
- Serializable
- Snapshot

Isolation

levels

Durability

Read uncommitted

Dirty Read  
Dirty Write

balance = 50

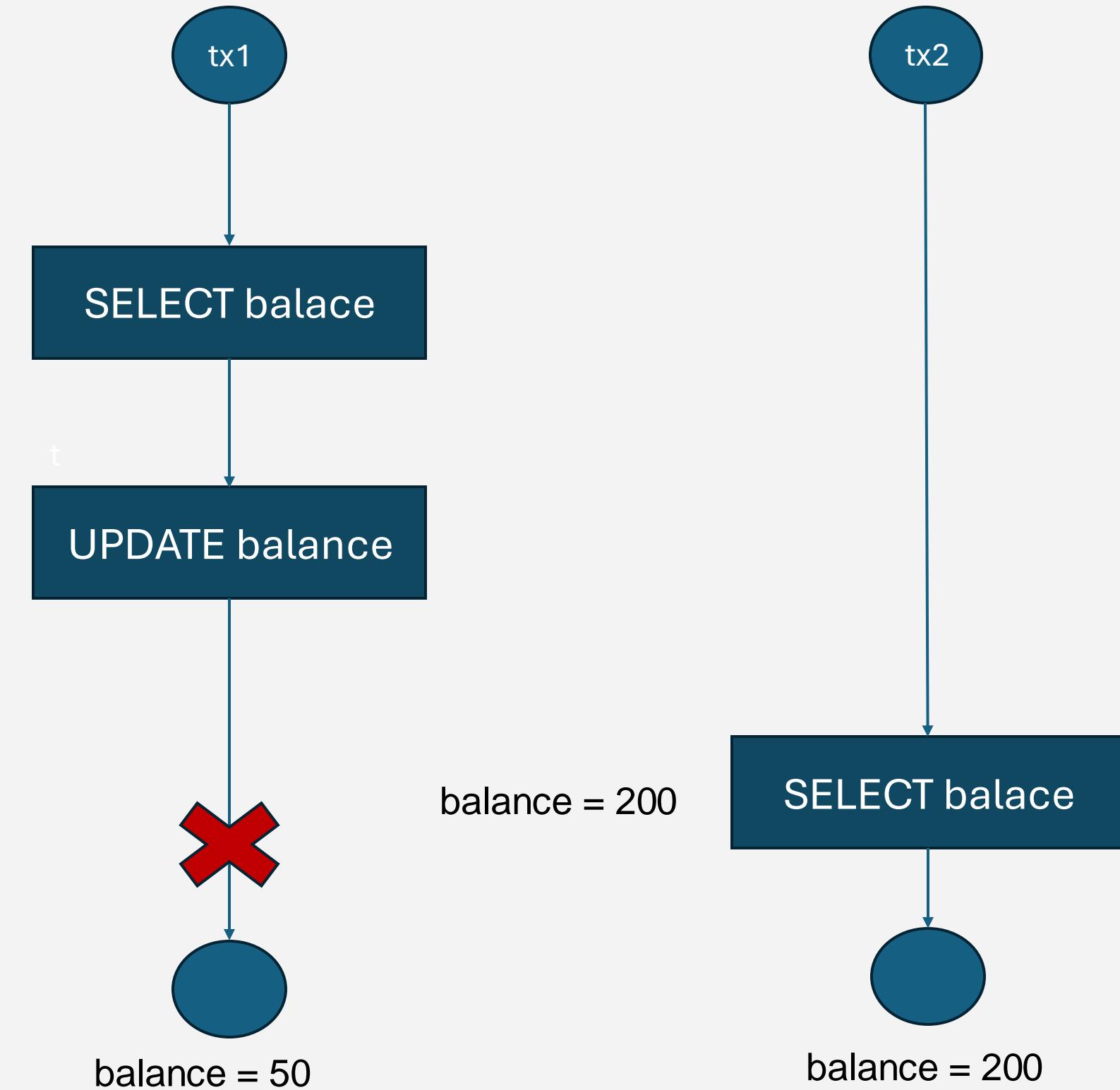
balance = 200

rollback

balance = 50

balance = 200

balance = 200

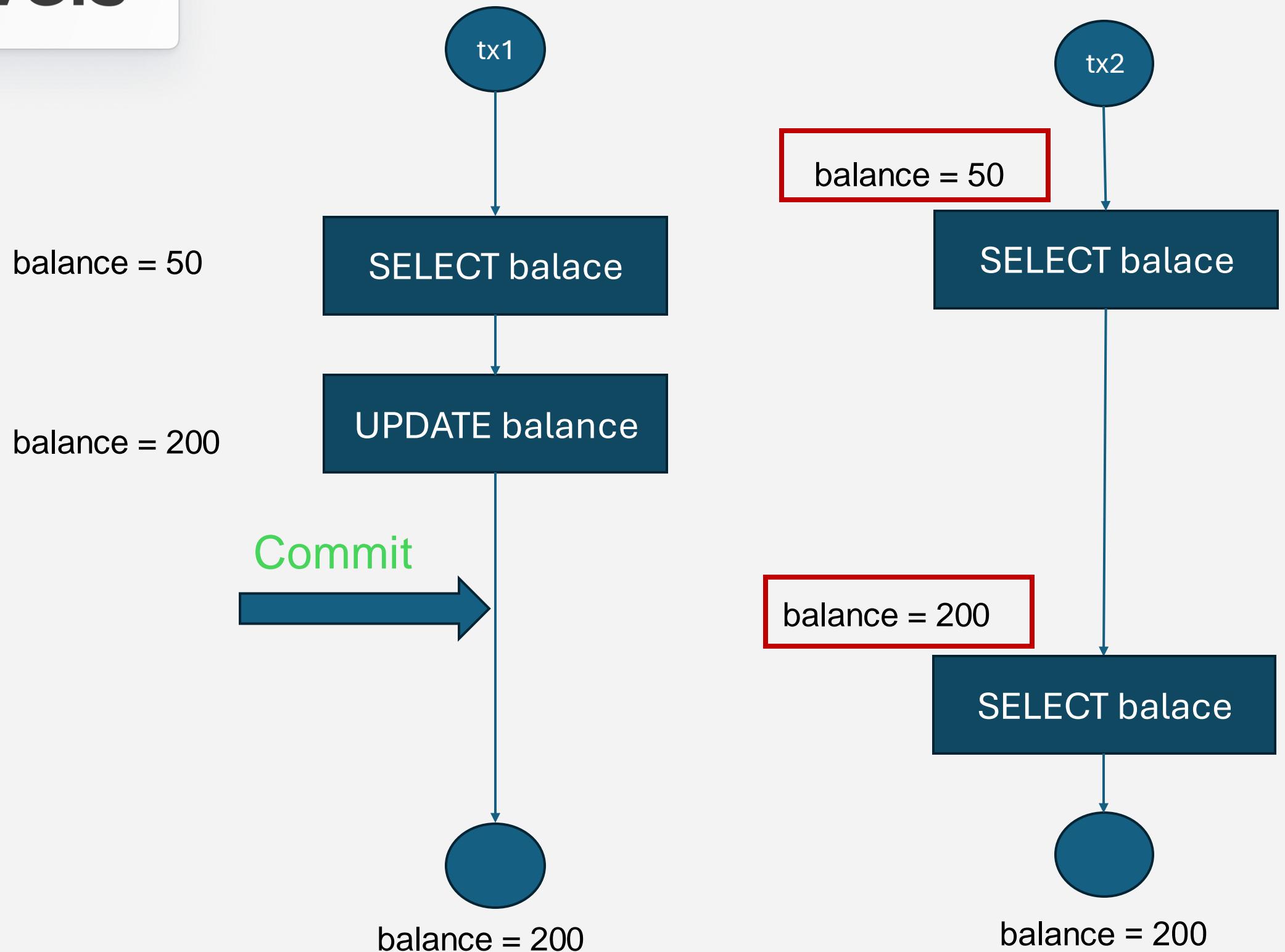


# Isolation levels

Read committed

Non  
Repeatable  
read

Durability



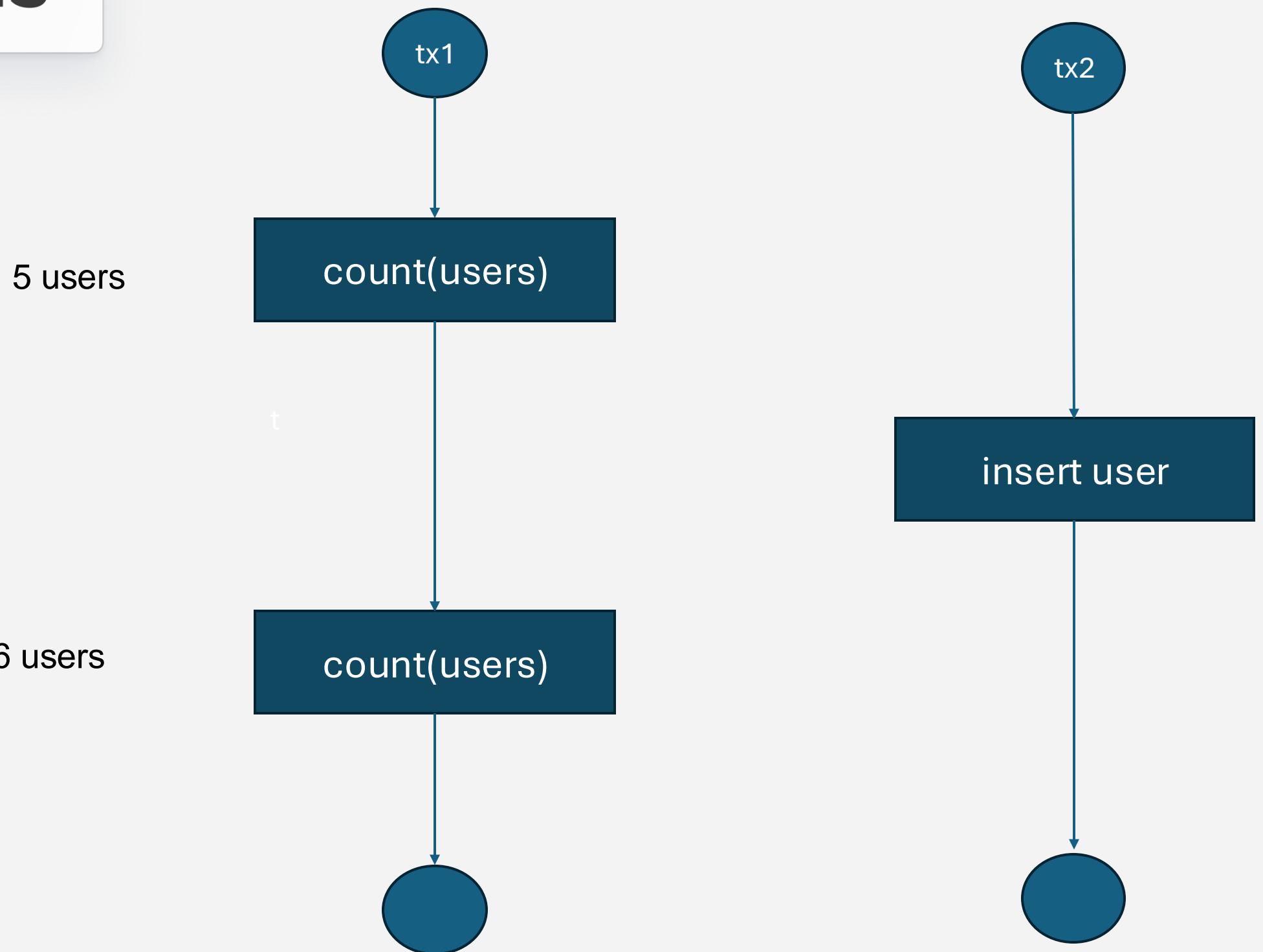
Isolation

levels

Durability

Repeatable read

Phantom  
Read



Isolation

levels

Durability

# **Serializable & Snapshot**

# Isolation

## Phenomena in Concurrent Transactions

Transaction Isolation Levels	Database anomalies		
	Dirty Read	Non Repeatable Read	Phantom Read
Read Uncommitted	Possible	Possible	Possible
Read Committed <small>is used by most of the applications</small>	Solved	Possible	Possible
Repeatable Read	Solved	Solved	Possible
Serializable <small>Very poor performance</small>	Solved	Solved	Solved

## Durability

Once committed, always preserved

What Could Happen if  
We Are Not Using ACID?



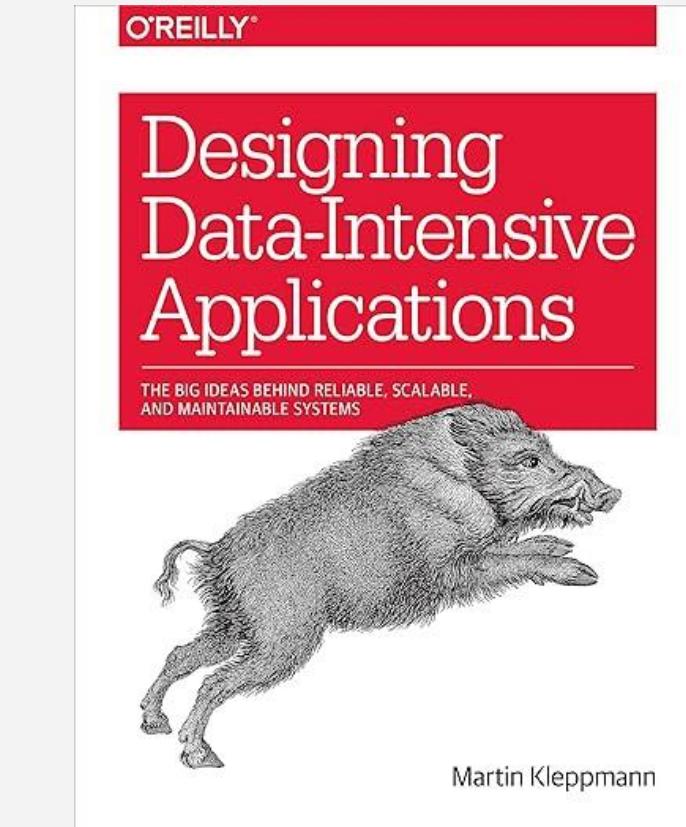
**This is where the fun begins.**

# Next Steps



Hussain  
Nasser

[Fundamentals of Database  
Engineering | Udemy](#)



Martin  
Kleppeman

Chapter 7

[Design Database  
Intensive Applications](#)