Chapter 7 : Transactions, Part-2 Snapshot Isolation

What are we going to cover?

Drawbacks of Read Committed Isolation

What is Snapshot Isolation?

Why it is needed and how is it different from Read Committed Isolation?

How to implement Snapshot Isolation?

Drawbacks of Read Committed Isolation

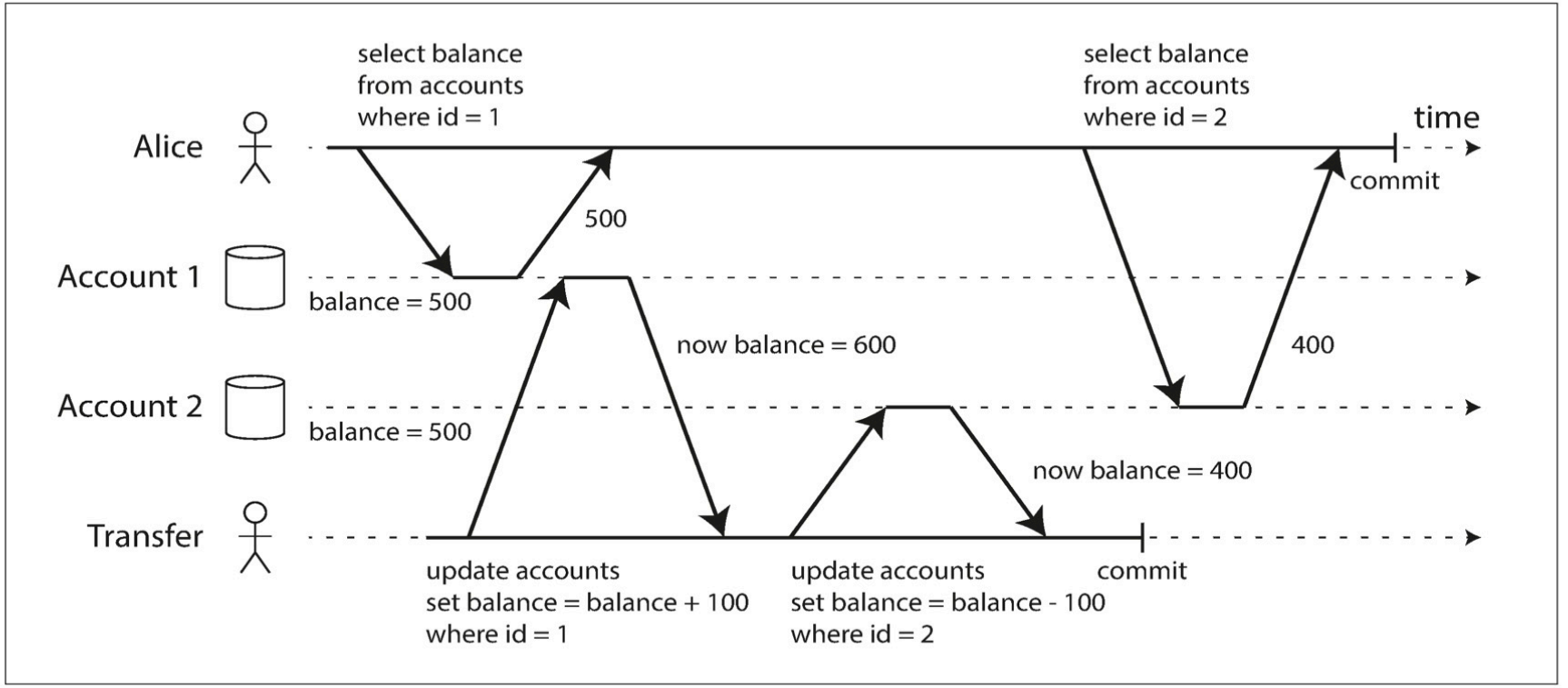


Figure 7-6. Read skew: Alice observes the database in an inconsistent state.

Drawbacks of Read Committed Isolation

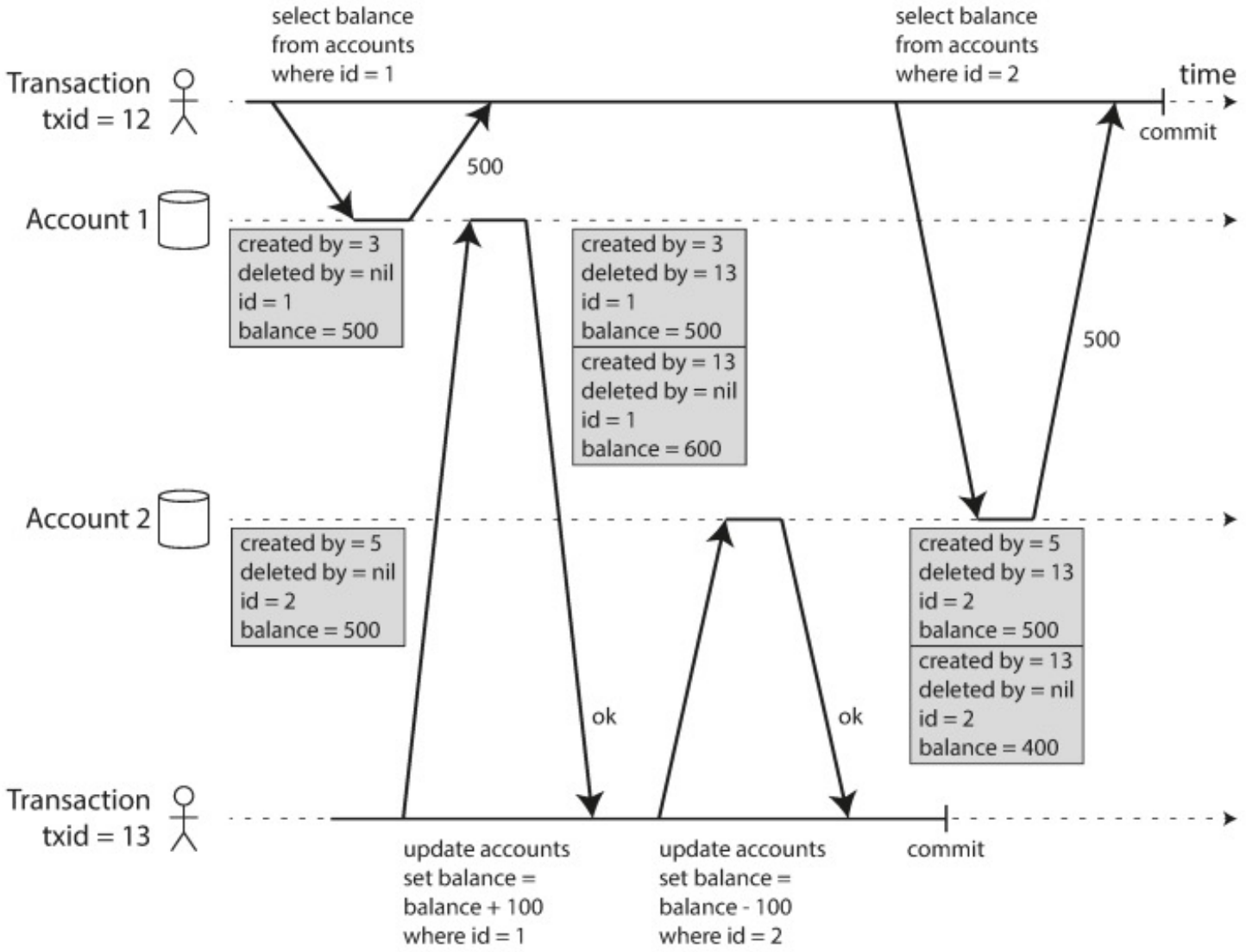
- Non-Repeatable read or read skew
- Some situations cannot tolerate such temporary inconsistencies:
  - Backups
  - Analytic queries and integrity checks

Snapshot Isolation

- Each transaction reads from a consistent snapshot of the database. The transaction sees all the data that was committed in the database at the start of the transaction.
- Good for long-running, read-only queries such as backups and analytics.

Implementing Snapshot Isolation

- Use write locks to prevent dirty writes – transaction that makes a write can block the progress of another transaction that writes to the same object. However, reads do not require any locks.
- Readers never block writers and writers never block readers.
- Database keeps several different committed versions of an object- known as multi version concurrency control(MVCC).



Implementing snapshot isolation using multi-version objects

Visibility rules for observing a consistent snapshot

- At start of each transaction, the database makes a list of all other transactions that are in progress at that time. Any writes that those transactions have made are ignored, even if the transactions subsequently commit.
- Any writes made by aborted transactions are ignored.
- Any writes made by transactions with a later transaction ID after the current transaction are ignored, regardless of whether those transactions have committed.
- All other writes are visible to the application’s queries.

Summary

- Drawbacks of Read Committed Isolation
- Snapshot Isolation
- Why Snapshot Isolation?
- Implementation of Snapshot Isolation

Thank You!