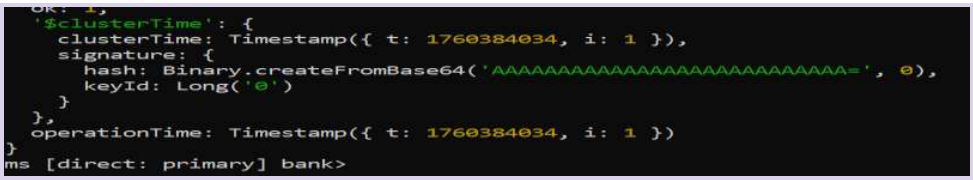
PRACTICAL 8
Implementing transactions to maintain data consistency in MongoDB

In mongosh:

```
C:\Windows\System32>cd C:\Program Files\MongoDB\Server\8.2\bin

C:\Program Files\MongoDB\Server\8.2\bin>start mongod --replSet ms --logpath /data/rs1/1.log --dbpath /data/rs1 --port 27018

C:\Program Files\MongoDB\Server\8.2\bin>start mongod --replSet ms --logpath /data/rs2/2.log --dbpath /data/rs2 --port 27019

C:\Program Files\MongoDB\Server\8.2\bin>start mongod --replSet ms --logpath /data/rs3/3.log --dbpath /data/rs3 --port 27020
```

```
ms [direct: primary] bank> rs.status()
{
  set: 'ms',
  date: ISODate('2025-10-13T19:33:58.762Z'),
  myState: 1,
  term: Long('1'),
  syncSourceHost: '',
  syncSourceId: -1,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
    lastCommittedWallTime: ISODate('2025-10-13T19:33:54.105Z'),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
    appliedOpTime: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
    durableOpTime: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
    writtenOpTime: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
    lastAppliedWallTime: ISODate('2025-10-13T19:33:54.105Z'),
    lastDurableWallTime: ISODate('2025-10-13T19:33:54.105Z'),
    lastWrittenWallTime: ISODate('2025-10-13T19:33:54.105Z')
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1760383994, i: 1 }),
  electionCandidateMetrics: {
    lastElectionReason: 'electionTimeout',
    lastElectionDate: ISODate('2025-10-13T19:22:33.360Z'),
    electionTerm: Long('1'),
    lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1760383343, i: 1 }), t: Long('-1') },
    lastSeenWrittenOpTimeAtElection: { ts: Timestamp({ t: 1760383343, i: 1 }), t: Long('-1') },
    lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1760383343, i: 1 }), t: Long('-1') },
    numVotesNeeded: 2,
    priorityAtElection: 1,
    electionTimeoutMillis: Long('10000'),
    numCatchUpOps: Long('0'),
    newTermStartDate: ISODate('2025-10-13T19:22:33.408Z'),
    wMajorityWriteAvailabilityDate: ISODate('2025-10-13T19:22:33.904Z')
  },
  members: [
    {
      _id: 0,
      name: 'localhost:27018',
      health: 1,
      state: 1,
      stateStr: 'PRIMARY',
      uptime: 1197,
      optime: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
      optimeDate: ISODate('2025-10-13T19:33:54.000Z'),
      optimeWritten: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
      optimeWrittenDate: ISODate('2025-10-13T19:33:54.000Z'),
      lastAppliedWallTime: ISODate('2025-10-13T19:33:54.105Z'),
      lastDurableWallTime: ISODate('2025-10-13T19:33:54.105Z'),
      lastWrittenWallTime: ISODate('2025-10-13T19:33:54.105Z'),
      syncSourceHost: '',
      syncSourceId: -1,
      infoMessage: '',
      electionTime: Timestamp({ t: 1760383353, i: 1 }),
      electionDate: ISODate('2025-10-13T19:22:33.000Z'),
      configVersion: 1,
      configTerm: 1,
      self: true,
```


configTerm: 1,
self: true,
lastHeartbeatMessage: ''
},
{
_id: 1,
name: 'localhost:27019',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 695,
optime: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
optimeDurable: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
optimeWritten: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
optimeDate: ISODate('2025-10-13T19:33:54.000Z'),
optimeDurableDate: ISODate('2025-10-13T19:33:54.000Z'),
optimeWrittenDate: ISODate('2025-10-13T19:33:54.000Z'),
lastAppliedWallTime: ISODate('2025-10-13T19:33:54.105Z'),
lastDurableWallTime: ISODate('2025-10-13T19:33:54.105Z'),
lastWrittenWallTime: ISODate('2025-10-13T19:33:54.105Z'),
lastHeartbeat: ISODate('2025-10-13T19:33:57.461Z'),
lastHeartbeatRecv: ISODate('2025-10-13T19:33:58.317Z'),
pingMs: Long('0'),
lastHeartbeatMessage: '',
syncSourceHost: 'localhost:27018',
syncSourceId: 0,
infoMessage: '',
configVersion: 1,
configTerm: 1
},
{
_id: 2,
name: 'localhost:27020',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 695,
optime: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
optimeDurable: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
optimeWritten: { ts: Timestamp({ t: 1760384034, i: 1 }), t: Long('1') },
optimeDate: ISODate('2025-10-13T19:33:54.000Z'),
optimeDurableDate: ISODate('2025-10-13T19:33:54.000Z'),
optimeWrittenDate: ISODate('2025-10-13T19:33:54.000Z'),
lastAppliedWallTime: ISODate('2025-10-13T19:33:54.105Z'),
lastDurableWallTime: ISODate('2025-10-13T19:33:54.105Z'),
lastWrittenWallTime: ISODate('2025-10-13T19:33:54.105Z'),
lastHeartbeat: ISODate('2025-10-13T19:33:57.462Z'),
lastHeartbeatRecv: ISODate('2025-10-13T19:33:58.316Z'),
pingMs: Long('0'),
lastHeartbeatMessage: '',
syncSourceHost: 'localhost:27018',
syncSourceId: 0,
infoMessage: '',
configVersion: 1,
configTerm: 1
}
],
ok: 1,
'$clusterTime': {
clusterTime: Timestamp({ t: 1760384034, i: 1 }),
signature: {
hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),


'$clusterTime': {
clusterTime: Timestamp({ t: 1760384034, i: 1 }),
signature: {
hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
keyId: Long('0')
}
},
operationTime: Timestamp({ t: 1760384034, i: 1 })
}
ms [direct: primary] bank>