RA6 – 25100198

The way Spanner solves this issue is, like many great innovations, stupid simple. It uses the concept of snapshot isolation, which sounds far more complicated than the actual implementation. Essentially, Spanner defines timestamps for every interaction. For R/W (read-write) transactions, the timestamp is the Commit time. For RO (read-only) transactions, the timestamp is the start time. In other words, an RO transaction is timestamped with whatever “wall-clock” time the first read began.
To solve the issue described, RO transactions will only use the record with **the largest** timestamp **less than its own timestamp**. As such, T3 will only use the record of T1, and not use T2, whose timestamp is greater than T3’s. Boom, problem solved, and we now have a serializable order, namely T1, T3, T2.