# Hands-on 5: Logging

Michael Plasmeier

1. The database only shows studentA since that was the only transaction that ended with end which means the transaction was written to disk.
2. With or without the logging system?

Without the logging, only studentA would be there, since that is the only entry in the DB.

With the logging system, all of the actions should be there, including studentA with 900, studentB with 2000, and studentC with 3100.

1. Again, those changes were never ended (written to disk). This implementation of a WAL only records the values into the DB when end is reached.
2. I would expect that the DB file would stay the same.
3. When it actually runs, we have studentA with 900 and studentC with 3100. This is not what I expected. For some reason, the 3rd transaction has been ended and committed. You can see that in the log.

From reading the textbook, transaction 3 is a “winner” – it has been ordered committed, but has not yet been written to the disk. The WAL recovery process goes ahead and writes these to disk for us.

1. Done are transactions committed and ended/written to disk. Winners transactions that were ordered committed, but never ended/written to disk. Losers are transactions that are not ordered committed or aborted. These are undone as the log is read backwards.
2. Checkpoint **does not** write things to DB. It is a step that speeds recovery. (see below)
3. Transactions 1 and 3 were rolled back. Checkpoint writes all of the currently open transactions (“losers”) at that point in time. When the backwards scan reaches a checkpoint, it only has to scan up to those losers that are listed and no further. This can greatly increase recovery time when most transactions have been ended/written to disk.
4. The DB is recovered to the same state. This is called *idempotent*.
5. The old one was winners: 3, losers: 2, done: 1. The re-recovery had winners: (nil), losers: 2, done: 1, 3. The log file is updated when the file is recovered to clarify that an action was taken (with END). Losers are ENDed so that future recoveries ignore it and don’t try to roll it back again in the future.