

## Database recovery in Multi Database system

Normally a transaction accesses a single database. Sometimes a single transaction, called a multidatabase transaction, may require access to multiple databases. These databases may even be stored on different types of DBMSs; for example, some DBMSs may be relational, whereas others are object-oriented, hierarchical, or network DBMSs. In such a case, each DBMS involved in the multidatabase transaction may have its own recovery technique and transaction manager separate from those of the other DBMSs. This situation is somewhat similar to the case of a distributed database management system where parts of the database reside at different sites that are connected by a communication network.

To maintain the atomicity of a multidatabase transaction, it is necessary to have a two-level recovery mechanism. A global recovery manager, or coordinator, is needed to maintain information needed for recovery, in addition to the local recovery managers and the information they maintain (log, tables). The coordinator usually follows a protocol called the two-phase commit protocol, whose two phases can be stated as follows:

**Phase 1.** When all participating databases signal the coordinator that the part of the multidatabase transaction involving each has concluded, the coordinator sends a message prepare for commit to each participant to get ready for committing the transaction. Each participating database receiving that message will force-write all log records and needed information for local recovery to disk and then send a ready to commit or OK signal to the coordinator. If the force-writing to disk fails or the local transaction cannot commit for some reason, the participating database sends a cannot commit or not OK signal to the coordinator. If the coordinator does not receive a reply from the database within a certain time out interval, it assumes a not OK response.

**Phase 2.** If all participating databases reply OK, and the coordinator's vote is also OK, the transaction is successful, and the coordinator sends a commit signal for the transaction to the participating databases. Because all the local effects of the transaction and information needed for local recovery have been recorded in the logs of the participating databases, recovery from failure is now possible. Each

participating database completes transaction commit by writing a [commit] entry for the transaction in the log and permanently updating the database if needed. On the other hand, if one or more of the participating databases or the coordinator have a not OK response, the transaction has failed, and the coordinator sends a message to roll back or UNDO the local effect of the transaction to each participating database. This is done by undoing the transaction operations, using the log.

The net effect of the two-phase commit protocol is that either all participating databases commit the effect of the transaction or none of them do. In case any of the participants—or the coordinator—fails, it is always possible to recover to a state where either the transaction is committed or it is rolled back. A failure during or before Phase 1 usually requires the transaction to be rolled back, whereas a failure during Phase 2 means that a successful transaction can recover and commit.