Lesson 8 Supplement
Recovery

The background of recovery must first be explained.

1. The log file and checkpoints.
2. Transactions and commit or abort.
3. Volatile and Stable storage.
4. System buffers and writing buffers to disk.

Then there are two types of crashes:

(a) The database is extensively damaged.
(b) The database survives, but is inconsistent.

For type (a) crashes, must go back to most recent intact backup, then use log to redo committed transactions that are not included in the backup. This assumes the log survives the crash.

For type (b) crashes, recovery strategy depends on the update policy of the DBMS:

Deferred Update:

- Transactions committed before the last checkpoint can be ignored since the updates have already reached the disk (database).
- Transactions committed after the last checkpoint must be redone (using the log) because their updates may have been lost in the system buffers. Redoing performs writes to the database using the after-images from the log, in the order they were written to the log. If this writing has been performed already, before the failure, the write has no effect on the data item, so there is no damage done by writing again.
- Transactions that have started and are active at the time of the failure, but not yet committed, do not have to be undone

since updates are deferred until the commit point, but they must be restarted from the beginning.
- Transactions aborted before the failure can be ignored because no actual writing was done to the database.

Immediate Update:
- Transactions committed before the last checkpoint are ignored (same as Deferred Update).
- Transactions committed after the last checkpoint must be redone (using the log as described above) because their updates may have been lost in the system buffers.
- Transactions that have started but not yet committed must be undone, since updates may have reached the database. Undoing is based on the log and is done in reverse order using the before-images from the log.