Q2:

1. Transaction Logging (redo log in MySQL):

* **Similarities:**
* A record of every transaction is saved. If there is a system failure, the transactions from the log will be replayed.
* Modifications that did not finish updating data files before an unexpected shutdown are replayed automatically.[[1]](#footnote-1)
* **Differences:**
* If all changes are flushed from the buffer to the tablespaces at the time of the crash, redo log application is skipped. Also, InnoDB skips redo log application if redo log files are missing at startup.
* During recovery, InnoDB scans the redo log to collect counter value changes and applies the changes to the in-memory table object.
* When encountering index tree corruption, InnoDB writes a corruption flag to the redo log, which makes the corruption flag crash safe.[[2]](#footnote-2)

1. Checkpointing:

* **Similarities:**
* During crash recovery, InnoDB looks for a checkpoint label written on the log files. Then InnoDB scans the log files forward from the checkpoint, applying the logged modifications to the database.
* **Differences:**
* InnoDB writes checkpoint information to the first log file at each checkpoint.
* InnoDB creates checkpoints and this often involves flushing of modified database pages to disk.[[3]](#footnote-3)

1. undo log:

* **Similarities:**
* undo log is a collection of undo log records associated with a single read-write transaction.
* **Differences:**
* undo log contains information about how to undo the latest change by a transaction to a clustered index record.[[4]](#footnote-4)

1. <https://dev.mysql.com/doc/refman/8.0/en/innodb-redo-log.html#:~:text=The%20redo%20log%20is%20a,or%20low%2Dlevel%20API%20calls> [↑](#footnote-ref-1)
2. <https://dev.mysql.com/doc/refman/8.0/en/innodb-recovery.html> [↑](#footnote-ref-2)
3. <http://download.nust.na/pub6/mysql/doc/mysql-backup-excerpt/5.1/en/innodb-checkpoints.html> [↑](#footnote-ref-3)
4. <https://dev.mysql.com/doc/refman/8.0/en/innodb-undo-logs.html> [↑](#footnote-ref-4)