10 – Recovery Protocol

1. Logging Schemes
    A. Approach
        i. Physical logging : records the changes made to a specific record
        ii. Logical logging : records the operations
    B. Log flushing
        i. All-at-once : wait until txn's commit
        ii. Incremental : allow system to flush log whenever times
    C. Group commit
       batch logs from multiple txns, flush them together.
    D. Early lock release
       if logs are flushed, records' lock can be unlocked. (is it right? Idk)
    E. MVCC's version records can be physical logging
        i. In-row versioning : not using version storage to reduce traversing time
        ii. Off-row versioning : using version storage to store old versions
2. MSSQL
    A. Recovery protocol
        i. Analysis
        ii. Redo : after this, DB is available
        iii. Undo
            1. Background cleanup (after logical revert, garbage collection)
            2. Aborted version overwrite
               (just overwrite version, not making new version)

3. SILO

    A. Logging

        i. Store per-CPU buffer to store log per txn.

        ii. If buffer is full, give buffer to logger thread, acquire another buffer

        iii. Per 100 epochs, new log file is created, and old file is renamed with max epoch it contains

    B. Persistent epoch

        i. Maintain persistent epoch to track current max epoch.

    C. Recovery

        i. Load last checkpoint

        ii. Log replay
        process log reverse order to find last update of each records

4. Checkpoint Protocols

    A. Ideal :
    no slowdown of txns,
    no latency spikes,
    no excessive memory overhead

    B. Approach
    Consistent checkpoints,
    Fuzzy checkpoints

    C. Contents :
    complete checkpoints,
    delta checkpoints

    D. Frequency
    time based,
    log file size,
    on shutdown