17 – Two-Phase Locking Concurrency Control

1. Lock Types
   A. Shared Locks : can share object with threads which have shared locks
   B. Exclusive Locks : cannot share object with any threads
2. Two-Phase Locking
   A. Just using locks cannot solve isolation problem
   B. Phase
      i. Growing : request locks
      ii. Shrinking : release locks
   C. Another problem : cascading aborts, deadlocks
      i. If T1 aborted at shrinking phase,
         other txn can abort too, because of dirty read.
      ii. 2PL can cause deadlocks at growing phase
   D. Strong Strict 2PL
      i. There is no shrinking phase,
         all locks should be released just before commit.
      ii. It can solve dirty read problem, also solving cascading abort problem.

3. Deadlock Detection + Handling
    A. Deadlock Detection
        i. Make wait-for graph
    B. Deadlock Handling
        i. Select victim
            1. By age, progress, number of locked object, number of rollbacks…
        ii. Abort or restart victim
    C. Deadlock Handling and Detection has overhead,
    frequency of these are important trade-off.
4. Deadlock Prevention
    A. All transaction has priority (= age)
        i. Approach
            1. Wait-die : old txn waits for young txn
            if conflict occurs, and young txn holds the lock, old txn waits.
            if old txn holds the lock, young one aborts.
            2. Wound-wait : young txn waits for old txn
            if conflict occurs, and young txn holds the lock, young one aborts.
            if old txn holds the lock, young one waits.
        ii. One type (old or young) of direction is possible.
        iii. If transaction restarts, its priority should be original age
        because of starvation.
5. Hierarchical Locking
    A. Lock granularities
        i. If table is locked, all tuples in the table should be locked as same type
    B. Intention locks
        i. Lock type that means intention of operation to child object.