19 – Multi-Version Concurrency Control

1. MVCC
   A. DBMS stores multiple physical versions of single logical object.
   B. Read-only txn can see a version that is older than txn itself.
      So, it uses timestamps to check visibility.
   C. If txn commits, versions generated by the txn are writed into DB
2. Design Decisions
   A. Concurrency Control Protocol
      i. Timestamp Ordering
      ii. Optimistic Concurrency Control
      iii. Two Phase Locking
   B. Version Storage
      Version Chain : linked list that manages version pointer
      indexes always point to the “head” of chain
      i. Append-Only : just add new version into chain
      ii. Time Travel : old version are moved to some other table
      iii. Delta : old version’s change attributes are moved to some other table.
   C. Garbage Collection
      Garbage : version that no txn can see.
      i. Tuple level : find old versions directly in table.
         1. Background vacuuming : separate threads scan all table periodically
         2. Cooperative Cleaning : each thread check version if it is reclaimable
      ii. Transaction level : txn keep tracking records old version.

D. Index Management

i. Secondary indexes

1. Logical pointers

2. Physical pointers

3. What is it?

A. How transaction level garbage collection work?

B. Secondary index's purpose…? (I didn't get it)