Braden Ackles

CS 440

2/10/2017

Homework #4

1. Problem #1:
   1. Consider the following classes of schedules: serializable and 2PL. For each of the following schedules, state which of the preceding classes it belongs to. If you cannot decide whether a schedule belongs in a certain class based on the listed actions, explain brieﬂy.
   2. The actions are listed in the order they are scheduled and preﬁxed with the transaction name. If a commit or abort is not shown, the schedule is incomplete; assume that abort or commit must follow all the listed actions.
   3. T1:R(X), T2:R(Y), T3:R(X), T1:R(Y)
      1. It is serializable and also 2PL. This is because for 2PL it never try's to have conflicting commands with locks.
   4. T1:R(X), T1:R(Y), T1:W(X), T2:R(Y), T3:W(Y), T1:W(X), T2:R(Y)
      1. It is serializable and is not 2PL because it is not possible to determine if it is recoverable after T2:R(Y) and T3:W(Y).
   5. T1:W(X), T2:R(X), T1:W(X), T2:Commit, T1: Commit
      1. It is neither since the commands overwrite themselves with two T1:W(X). This means that it goes back on its self and writes data that has already been rewritten and there for is not serializable and if it's not serializable it can't be 2PL.
   6. T1:R(X), T2:W(X), T1:W(X), T3:R(X), T1:(Commit), T2(Commit), T3(Commit)
      1. It is neither since the command overwrite its self with the T1:R(X) and T1:W(X). This means that it goes back on its self and writes a command that was already read and there for is not serializable and if it's not serializable it can't be 2PL.
2. Problem #2
   1. Consider a database DB with relations R1 and R2. The relation R1 contains tuples t1 and t2 and the relation R2 contains tuples t3, t4, and t5. Assume that the database DB, relations, and tuples form a hierarchy of lockable database elements. Explain the sequence of lock requests and the response of the locking scheduler to the following schedule. You may assume all lock requests occur just before they are needed, and all unlocks occur at the end of the transaction, i.e., EOT.
   2. T1:R(t1), T2:W(t2), T2:R(t3), T1:W(t4)

|  |  |
| --- | --- |
| T1 | T2 |
| IS.lock(DB)  IS.lock(R1)  S.lock(t1)  Read(t1) |  |
|  | IX.lock(DB)  IX.lock(R1)  X.lock(t2)  Write(t2) |
|  | IS.lock(R2)  S.lock(t3)  Read(t3) |
| X.lock(t4)  Write.(t4) |  |
| S.release-lock(t1)  X.release-lock(t4)  IS.lock(R1)  IS.lock(DB) | X.release-lock(t2)  S.release-lock(t3)  IS.release-lock(R2)  IX.release-lock(R1)  IX.release-lock(DB) |
| Commit | Commit |

This above chart is a breakdown of the steps to show what happens to the database at what time.