# Exercise Session 6

# Locking-based Concurrency Control Techniques

## 6.1 The Two-Phase-Locking Protocol (2PL)

a) Show that the set of histories produced by 2PL schedulers is a strict subset of CSR.

b) Give an example of a history that is produced by a 2PL scheduler, but is not strict. What extension of 2PL could you suggest, that would guarantee that only ST histories are produced?

c) Show that 2PL preserves the order of transactions, i.e. $2PL \in OCSR$

## 6.2 Deadlock Management

a) What is understood by a deadlock? Show how deadlocks could occur under the 2PL protocol.

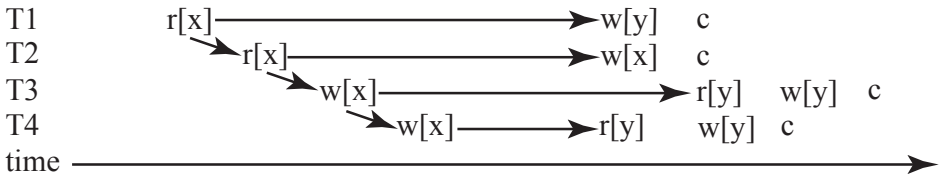
b) How can we detect / eliminate deadlocks in locking-based schedulers?

c) Discuss the different ways in which deadlocks can be avoided.

d) Is it possible that a transaction participates in more than one deadlock?

## 6.3 2PL Scheduler

a) Given is the following partially ordered history over the transactions T1, T2, T3 and T4:



Assuming that Strict 2PL is used for concurrency control, describe the corresponding locking operations and managed state information (locks, waiting queues). Operations of individual transactions can be executed at the times shown in the diagram the earliest. Delays caused by blocking are potentially possible.

b) Given is the history:
H = r1[x] r2[z] r3[y] r3[z] w2[z] c3 r1[z] w1[y] r2[x] c1 c2
Can this be a history produced by a 2PL Scheduler? Is this a strict history?

## 6.4 Transaction Isolation Levels

In the SQL-92 standard the 4 isolation levels were defined, which are often implemented in the following way using locking-based techniques:

| **Isolation Level** | **Row/Predicate Exclusive** | **Row Shared** | **Predicate Shared** |
|---|---|---|---|
| Read Uncommitted | Long term (or not allowed) | None (latch) | None (latch) |
| Read Committed | Long term | Short term | Short term |
| Repeatable Read | Long term | Long term | Short term |
| Serializable | Long term | Long term | Long term |

Isolation Level 1 is usually only used for read-only transactions. "Row-exclusive" and "row shared" correspond to write and read locks respectively. The table shows when locks are set and how long they are held: "long term" means that locks are held until Commit, "short term" means that locks can be released earlier (usually after the operation completes). A latch is a short-duration lock set only for the duration of a physical I/O operation to ensure atomicity.

a) Which isolation levels preclude (do not preclude) the "Dirty-Read", "Dirty-Write" and "Non-Repeatable Read" phenomena respectively?

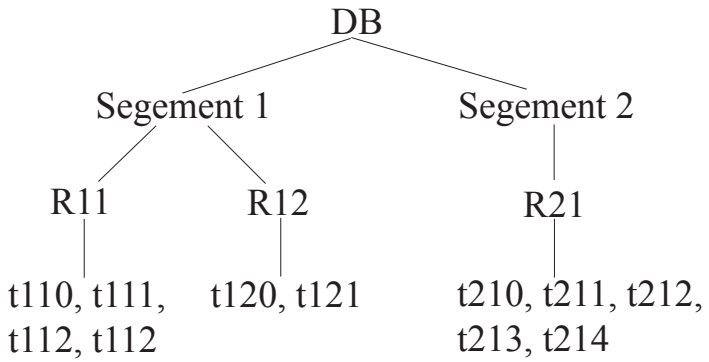
b) Which isolation levels preclude (do not preclude) the "Inconsistent Analysis" anomaly? Describe the problems caused by it using practical examples.

c) Which isolation levels preclude (do not preclude) the "Lost Update" and "Write Skew" anomalies? Describe the problems caused by them using practical examples.

d) Which isolation levels preclude (do not preclude) the "Phantom" phenomenon? Describe the anomalies caused by it using practical examples.

## 6.5 Multi-Granularity Locking (MGL)

The diagram below shows the data graph of a RDBMS.




a) Explain how the MGL Method works.

- Discuss the lock types and the compatibility matrix for MGL.
- In what order must locks be set / released in MGL?

b) What locks need to be set if the following transactions are executed:

- T1 reads tuples t110, t111, t112, t113
- T2 reads tuples t210 and writes t121
- T3 reads tuples t120, t121 updates either t120 or t121
- T4 performs a nested-loop-join of R11 and R12.

c) To what extent does MGL provide a solution for the Phantom problem?