

COMP207 Lab Exercises

Tutorial 3 - Week 5

The exercises below provide the opportunity to practice the concepts and methods discussed during the lectures. Don't worry if you cannot solve all the exercises during the lab session, but try to tackle at least one or two of them. If at some point you do not know how to proceed, you could review the relevant material from the lecture notes and return to the exercise later.

Logging and Recovery

Exercise 1 (Exercise 17.2.4/17.2.5 in [2]). The following is a sequence of undo-log records written by two transactions, T_1 and T_2 :

<START T_1 >
< $T_1, X, 10$ >
<START T_2 >
< $T_2, Y, 20$ >
< $T_1, Z, 30$ >
< $T_2, U, 40$ >
<COMMIT T_2 >
< $T_1, V, 50$ >
<COMMIT T_1 >

- (a) Describe the action of the recovery manager, including changes to both the database and the log on disk, if there is a system failure and the last log record to appear on disk is:
- (i) <START T_2 >
 - (ii) <COMMIT T_2 >
 - (iii) < $T_1, V, 50$ >
 - (iv) <COMMIT T_1 >
- (b) For each of the situations (i)–(iv) in (a), describe what values written by T_1 and T_2 *must* appear on disk? Which values *might* appear on disk?

Exercise 2 (Exercise 19.1.2/19.1.3 in [2]). Consider the following schedules:

- $S_1: r_1(X); r_2(Y); w_1(Y); w_2(Z); r_3(Y); r_3(Z); w_3(U)$
- $S_2: r_1(X); w_1(Y); r_2(Y); w_2(Z); r_3(Z); w_3(U)$
- $S_3: r_2(X); r_3(X); r_1(X); r_1(Y); r_2(Y); r_3(Y); w_2(Z); r_3(Z)$
- $S_4: r_2(X); r_3(X); r_1(X); w_1(Y); r_3(Y); w_2(Z); r_3(Z)$

- (a) Suppose that each of the schedules is followed by an abort operation for transaction T_1 . Tell which transactions need to be rolled back.
- (b) Now suppose that all three transactions commit and write their commit record on the log immediately after their last operation. However, a crash occurs, and a tail of the log was not written to disk before the crash and is therefore lost. Tell, depending on where the lost tail of the log begins:
- (i) What transactions could be considered uncommitted?
 - (ii) Are any dirty reads created during the recovery process? If so, what transactions need to be rolled back?
 - (iii) What additional dirty reads could have been created if the portion of the log lost was not a tail, but rather some portions in the middle?

Recoverable, Cascadeless, and Strict Schedules

Exercise 3 (Exercise 20.24 in [1]). For each of the following schedules, determine if the schedule is (A) recoverable, (B) cascadeless, (C) strict, (D) non-recoverable. Try to determine the strictest recoverability condition that each schedule satisfies.

- (a) $S_1: r_1(X); r_2(Z); r_1(Z); r_3(X); r_3(Y); w_1(X); c_1; w_3(Y); c_3; r_2(Y); w_2(Z); w_2(Y); c_2$
- (b) $S_2: r_1(X); r_2(Z); r_1(Z); r_3(X); r_3(Y); w_1(X); w_3(Y); r_2(Y); w_2(Z); w_2(Y); c_1; c_2; c_3$
- (c) $S_3: r_1(X); r_2(Z); r_3(X); r_1(Z); r_2(Y); r_3(Y); w_1(X); c_1; w_2(Z); w_3(Y); w_2(Y); c_3; c_2$

Exercise 4 (Exercise 19.1.1 in [2]). What are all the ways to insert lock operations (of the simple lock type only), unlock operations, and commit operations into

$$r_1(X); r_1(Y); w_1(X); w_1(Y)$$

so that the transaction T_1 is:

- (a) Two-phase locked, and strict two-phase locked.
- (b) Two phase locked, but not strict two-phase locked.

References

- [1] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems*. Pearson Education, 7th edition, 2016.
- [2] Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. *Database Systems - The Complete Book*. Pearson Education, 2nd edition, 2009.