

# Assignment 3

This homework assignment is an individual assignment.

**Wednesday, 23 November 2016, 23:55**

Rules:

- *Honor system.* The work that you submit must be your own. It must not be copied from anyone else. By handing in a solution to this assignment you sign implicitly that are aware of the regulations of ITU and that you pledge that you live up to the university's expectations of scholarly conduct.
- Submit your work as a single PDF file with a file name that includes your name.
- Upload your work to `learnit.itu.dk`.
- Late hand-ins will not be considered.
- This assignment will be graded and contribute with 15% to your final grade.

## Problem 1: Serializability and 2PL [30 points]

Consider the following schedule. We write  $R(\cdot)$  for read and  $W(\cdot)$  for write operations.

| Time | $T_1$  | $T_2$  | $T_3$  |
|------|--------|--------|--------|
| 1    | $R(A)$ |        |        |
| 2    | $W(A)$ |        |        |
| 3    |        |        | $R(A)$ |
| 4    |        |        | $W(A)$ |
| 5    |        | $R(C)$ |        |
| 6    |        | $R(A)$ |        |
| 7    | $R(B)$ |        |        |
| 8    |        |        | $R(B)$ |
| 9    |        | $W(C)$ |        |
| 10   | $W(B)$ |        |        |
| 11   |        |        | $W(B)$ |
| 12   |        | $R(B)$ |        |

- [10 points] We start off with some general questions about serializability and 2PL. Which ones of the following statements are true and which are false?
  - Schedules under 2PL could have cascading aborts.
  - In the shrinking phase of Strict 2PL, locks cannot be released until the end of the transaction.
  - If a schedule is conflict serializable, then it is also view serializable.
  - Schedules under both 2PL and Strict 2PL may lead to deadlocks.
  - Every serializable schedule is conflict serializable.
- [5 points] Is this schedule given above a serial schedule?
- [5 points] Give the dependency graph of this schedule.
- [5 points]
 

Is this schedule conflict serializable? If you answer yes to the previous question, provide the equivalent serial schedule. If you answer no, briefly explain why.
- [5 points]
 

Could this schedule have been produced by 2PL?

## Problem 2: Hierarchical Locking [50 points]

Consider a Database  $D$  consisting of two tables, Movies  $M$  and PlayIn  $P$ .

$Movies(mid, moviename, movierating)$  spans 300 pages, namely  $M_1$  to  $M_{300}$ .

$PlayIn(mid, actorname, actorrating)$ , spans 600 pages, namely  $P_1$  to  $P_{600}$ .

Furthermore, each page contains 100 records, and we use the notation  $P_3 : 20$  to represent the 20th record on the third page of the PlayIn table. Similarly,  $M_5 : 10$  represents the 10th record on the fifth page of the Movies table.

We use multiple-granularity locking, with S, X, IS, IX and SIX locks, and four levels of granularity: (1) database-level ( $D$ ), (2) table-level ( $M$ ,  $P$ ), (3) page-level ( $M_1 - M_{300}$ ,  $P_1 - P_{600}$ ), (4) record-level ( $M_1 : 1 - M_{300} : 100$ ,  $P_1 : 1 - P_{600} : 100$ ).

For each of the following operations on the database, please determine the sequence of lock requests that should be generated by a transaction that want to carry out these operations efficiently.

Please follow the format of the examples listed below:

- write  $IS(D)$  for a request of database-level IS lock
- write  $X(P_2 : 30)$  for a request of record-level X lock for the 30th record on the second page of the PlayIn table
- write  $S(P_2 : 30 - P_3 : 100)$  for a request of record-level S lock from the 30th record on the second page of the PlayIn table to the 100th record on the third page of the PlayIn table.

1. [5 points] Read ALL records on ALL pages in the Movies table.
2. [10 points] Read ALL records on page  $M_7$  through  $M_{21}$ , and modify the record  $M_{10} : 10$ .
3. [10 points] Modify the first record on EACH and EVERY page of the PlayIn table (these are blind writes that do not depend on the original contents in the pages).
4. [10 points] For EACH record in the Movies table, capitalize the English letters in the movie name if it is not capitalized. That is, “The Hobbit: The Desolation of Smaug” will be modified to “THE HOBBIT: THE DESOLATION OF SMAUG” but “THE HOBBIT: AN UNEXPECTED JOURNEY” will be left unchanged.
5. [10 points] Update the movie rating of EACH movie in the Movies table such that the rating of the movie becomes the sum of the performance “actor-rating” of all the actors/actresses played in the movie. More specific, we use the following formula:

$$\text{movie rating for mid}(M) = \sum_{rating \in \{ \langle r \rangle \mid \exists m, n, r. \langle m, n, r \rangle \in \text{playIn} \wedge m \in M \}} rating$$

6. [5 points]  
Delete ALL the records from ALL tables.

### Problem 3: Deadlock Detection [20 points]

Consider the following lock requests.

| time  | $t_1$  | $t_2$  | $t_3$  | $t_4$  | $t_5$  | $t_6$  | $t_7$  | $t_8$  | $t_9$  |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| $T_1$ | $S(D)$ |        | $S(A)$ |        |        | $X(C)$ |        | $S(B)$ |        |
| $T_2$ |        |        |        | $S(A)$ | $X(B)$ |        |        |        |        |
| $T_3$ |        |        |        |        |        |        | $S(C)$ |        |        |
| $T_4$ |        | $S(C)$ |        |        |        |        |        |        | $X(C)$ |
| LM    | $g$    |        |        |        |        |        |        |        |        |

- $S(\cdot)$  and  $X(\cdot)$  stand for ‘shared lock’ and ‘exclusive lock’, respectively.
  - $T_1$ ,  $T_2$  and  $T_3$  represent three transactions.
  - LM stand for ‘lock manager’.
1. [10 points] For the lock requests in table above, determine which lock will be granted or blocked by the lock manager. Please write ‘g’ in the LM row to indicate the lock is granted and ‘b’ to indicate the lock is blocked. For example, in the table, the first lock ( $S(D)$  at time  $t_1$ ) is marked as granted.
  2. [5 points] Give the wait-for graph for the lock requests in the table above.
  3. [5 points] Determine whether there exists a deadlock in the lock requests in the table above, and briefly explain why.