Homework 8 (by Prashanth Menon)
Due: hard copy, in class at 3:00pm, on Monday, Dec. 5

VERY IMPORTANT: Deposit hard copy of your answers, in class. For ease of grading, please
1. Separate your answers, on different page(s) for each question (staple additional pages, if needed).
2. Type the full info on each page: your name, Andrew ID, course#, Homework#, Question# on each of the 3 pages.

Reminders:
• Plagiarism: Homework is to be completed individually.
• Typeset all of your answers whenever possible. Illegible handwriting may get zero points, at the discretion of the graders.
• Late homeworks: in that case, please email it
  – to all TAs
  – with the subject line exactly 15-415 Homework Submission (HW 8)
  – and the count of slip-days you are using.

For your information:
• Graded out of 100 points; 3 questions total
• Rough time estimate: approx. 6 hours - 1 to 2 hours per question

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serializability and 2PL</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Deadlock Detection and Prevention</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Hierarchical Locking - Return of Bike Sharing</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Question 1: Serializability and 2PL ...................... [33 points]

On separate page, with `[course-id] [hw#] [question#] [andrew-id] [your-name]`

(a) Yes/No questions:
   i. [3 points] Every conflict-serializable schedule is view-serializable.
      □ Yes □ No
   ii. [3 points] In the shrinking phase of strict 2PL, locks cannot be released until
      the end of the transaction.
      □ Yes □ No
   iii. [3 points] Schedules under strict 2PL do not allow dirty reads.
      □ Yes □ No
   iv. [3 points] Schedules under strict 2PL may lead to cascading aborts.
      □ Yes □ No
   v. [3 points] Only schedules under 2PL (and not strict 2PL) may lead to dead-
      locks.
      □ Yes □ No

(b) Serializability:
   Consider the schedule given below in Table 1. R(·) and W(·) stand for ‘Read’ and
   ‘Write’, respectively.

<table>
<thead>
<tr>
<th>time</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
<th>$t_5$</th>
<th>$t_6$</th>
<th>$t_7$</th>
<th>$t_8$</th>
<th>$t_9$</th>
<th>$t_{10}$</th>
<th>$t_{11}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>R(A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R(B)</td>
<td>W(B)</td>
</tr>
<tr>
<td>$T_2$</td>
<td></td>
<td>R(C)</td>
<td>R(A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>W(A)</td>
<td></td>
<td>W(C)</td>
<td></td>
</tr>
<tr>
<td>$T_3$</td>
<td>R(B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>W(B)</td>
<td></td>
<td>R(A)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   Table 1: A schedule with three transactions: $T_1$, $T_2$, and $T_3$

   i. [2 points] Is this schedule serial?
      □ Yes □ No
   ii. [5 points] Give the dependency graph of this schedule.
   iii. [2 points] Is this schedule conflict serializable?
      □ Yes □ No
   iv. [2 points] Is this schedule view serializable?
      □ Yes □ No
   v. [5 points] If you answer “yes” to (iii), provide the equivalent serial schedule.
      If you answer “no”, briefly explain why.
   vi. [2 points] Could this schedule have been produced by 2PL?
      □ Yes □ No

Homework 8 continues...
Question 2: Deadlock Detection and Prevention . . . . . . [34 points]

On separate page, with ‘[course-id] [hw#] [question#] [andrew-id] [your-name]’

(a) Deadlock Detection:
Consider the following lock requests in Table 2. Note that:
- S(·) and X(·) stand for ‘shared lock’ and ‘exclusive lock’, respectively.
- T₁, T₂, and T₃ represent three transactions.
- LM stands for ‘lock manager’.

<table>
<thead>
<tr>
<th>time</th>
<th>t₁</th>
<th>t₂</th>
<th>t₃</th>
<th>t₄</th>
<th>t₅</th>
<th>t₆</th>
<th>t₇</th>
</tr>
</thead>
<tbody>
<tr>
<td>T₁</td>
<td></td>
<td></td>
<td>X(C)</td>
<td>S(C)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T₂</td>
<td></td>
<td>S(B)</td>
<td>S(C)</td>
<td></td>
<td>S(A)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T₃</td>
<td>S(C)</td>
<td></td>
<td>X(B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LM</td>
<td>g</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Lock requests of three transactions: T₁, T₂, and T₃

i. [6 points] For the lock requests in Table 2, 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 (X(A) at time t₁) is marked as granted.

ii. [5 points] Give the wait-for graph for the lock requests in Table 2 at time-tick t₇.

iii. [4 points] Determine whether there exists a deadlock in the lock requests in Table 2 and explain why.

(b) Deadlock Prevention:
Consider the following lock requests in Table 3. As before:
- S(·) and X(·) stand for ‘shared lock’ and ‘exclusive lock’, respectively.
- T₁, T₂, T₃, and T₄ represent four transactions.
- LM represents a ‘lock manager’.

<table>
<thead>
<tr>
<th>time</th>
<th>t₁</th>
<th>t₂</th>
<th>t₃</th>
<th>t₄</th>
<th>t₅</th>
<th>t₆</th>
<th>t₇</th>
<th>t₈</th>
</tr>
</thead>
<tbody>
<tr>
<td>T₁</td>
<td>X(B)</td>
<td></td>
<td>S(A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T₂</td>
<td>X(D)</td>
<td>X(C)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T₃</td>
<td>S(C)</td>
<td></td>
<td>X(B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T₄</td>
<td>X(A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LM</td>
<td>g</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Lock requests of four transactions: T₁, T₂, T₃, and T₄

Question 2 continues...
i. [6 points] For the lock requests in Table 3 determine which lock request will be granted, blocked or aborted by the lock manager (LM), if it has no deadlock prevention policy. Please write 'g' for grant, 'b' for block and 'a' for abort; for 'abort', specify which transaction is aborted - e.g., 'a' (T1 is aborted) An example is given in for time-tick $t_1$.

ii. [5 points] Give the wait-for graph for the lock requests in Table 3. Determine whether there exists a deadlock in the lock requests in Table 3 under LM, and explain why.

iii. [4 points] To prevent deadlock, we use a lock manager (LM) that adopts the Wait-Die policy. We assume the four transactions have priority: $T_1 < T_2 < T_3 < T_4$. Determine which lock request will be granted ('g'), blocked ('b') or aborted ('a'); for 'abort', specify which transaction is aborted - e.g., 'a' (T1 is aborted). Follow the same format as the previous question.

iv. [4 points] In this question, we use a lock manager (LM) that adopts the Wound-Wait policy. We assume the four transactions have priority: $T_1 < T_2 < T_3 < T_4$. Determine which lock request will be granted ('g'), blocked ('b') or aborted ('a'); for 'abort', specify which transaction is aborted - e.g., 'a' (T1 is aborted) Follow the same format as the previous question.

Homework 8 continues...
Question 3: Hierarchical Locking - Return of Bike Sharing [33 points]

On separate page, with ‘{course-id} [hw#] [question#] [andrew-id] [your-name]’

For this problem we consider a modified and simplified version of the bike sharing database from Homework 2. The bike sharing database has the following three tables:

Our bike sharing database (D) contains three tables: Bike (B), Station (S), and Trips (T). Specifically:

- **Bikes**(bid, model, year), that spans 150 pages, namely B1 to B150.
- **Trips**(tid, date, start_city, end_city, distance, bid), that spans 600 pages, namely T1 to T600.

Each page contains 100 records, and we use the notation Bi : j to represent the j\textsuperscript{th} record, 1 ≤ j ≤ 100, on the i\textsuperscript{th} page of table B. For example, B5 : 10 represents the tenth record on the fifth page of the Bikes 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 (B, S, T), (3) page-level (B1 − B150, T1 − T600), (4) record-level (B1 : 1 − B150 : 100, T1 : 1 − T600 : 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. You do not need to list unlock requests.

Please follow the format of the examples listed below:

- Write “IS(D)” to request a database-level IS lock
- Write “X(B2 : 30)” to request a record-level X-lock for the 30\textsuperscript{th} record on the second page of the Bikes table
- Write “S(T2 : 30 − T3 : 100)” to request a record-level S-lock from the 30\textsuperscript{th} record of the second page of the Trips table to the 100\textsuperscript{th} record of the third page of the Trips table.

(a) [7 points] Calculate the average distance of all trips.

(b) [6 points] Read ALL records on page B10 through B70, and modify the record B11 : 44.

(c) [7 points] Modify the date attribute of the last record on EACH and EVERY page of the Trips table to today’s date.

(d) [7 points] Increment the distance attribute of all records from the Trips table whose start_city is ‘Pittsburgh’.

(e) [6 points] Delete ALL the records from ALL tables.

End of Homework 8