CS 411: Database Systems

Spring 2023

Homework 4 (Due by 23:59 CT on 28th March 2023)

Logistics

1. This homework is due on March 28th at 23:59 CT. **We DO NOT accept late homework submissions.**
2. You will be using Gradescope to submit your solutions. Answer each sub-question (e.g. "a.") on a **new page** and submit your solution as a single PDF file on Gradescope. All registered students should enroll on Gradescope.
3. Please submit the PDF to "Homework 4".
4. *IMPORTANT*: Please make sure to link PDF pages with the corresponding question outline on GradeScope.
5. The answers can be written electronically or they can be hand-written, but if we cannot read your submissions, we won't be able to grade them.
6. Please write down any intermediate steps to receive full credit.
7. Keep your solutions brief and clear.
8. Please use Campuswire if you have questions about the homework but **do not post answers.** Feel free to use private posts or come to office hours.
9. Please tag your submission questions with the Gradescope rubric correctly; failure to do so will result in a grade deduction of 10 pts.
10. DO NOT PLAGIARIZE. The following are written in the course syllabus:

> 1. **Assignments are individual work.**
> 2. Collaboration is NOT allowed when working on the assignments.
> 3. Discussions are allowed if and only if these discussions regard only high-level concepts and general ideas. Discussion cannot involve answers to the questions on the homework. Checking answers/part of the solutions among peers are **not** allowed. Sharing answers on any public/private electronic platform, including but not limited to email, messenger, Facebook groups, discord chat, etc., are **not** allowed.
> 4. If you discussed questions with your classmates, you **must** include their names and the questions you discussed. Not including students' names will be considered a violation of the course's academic integrity policy.
> 5. You are allowed to submit regrade requests within the time frame listed on Campuswire. Typically we allow up to one week after the HW grades are released if not explicitly mentioned.
> 6.. Uploading your assignment questions to public platforms (i.e., shared drive, course hero, etc.) is prohibited. Such violations are copyright infringements and possible violations of academic honesty. We will process these strictly.

Question 1 : Basics of Transactions (5 points)

A Transaction **T1**, executed by a mobile phone store, performs the following steps:

1. The customer is searching for a desired mobile specification and brand. Information about the desired products is located in database objects (eg., table rows) **A** and **B**, which the system retrieves from disk and then the customer is told about the product options. The customer thinks about the options and decides to book an upcoming phone that accepts pre-orders. The information about the preorders of the chosen phone is stored in **B**. A pre-order request is made for the customer.
2. The customer selects a delivery date and time for the mobile phone to arrive at their delivery address from the available slots; delivery data is available in the database element **C**.
3. The system gets the customer's credit card number and generates a bill for the purchase. It adds the bill for the mobile phone to the list in database element **D**.
4. The system fetches the list on database element **E** that contains details for emails to be sent out as purchase confirmations. The customer's phone number, email address, and bill are added to this list so that an email can be sent confirming this purchase.

Express transaction **T1** as a sequence of Read and Write actions.

Your solution should have a format somewhat similar to:

*Ri(X); Wi(X); …. where **Ri** and **Wi** correspond to the read and write actions to the database item **X** by the transaction **Ti**.*

Question 2 : Isolation levels and Locking (20 points)

Consider the schedules **S1** and **S2**, followed by transactions **T1** and **T2** operating on database objects **A**, **B**, and **C**. For each part, explain your answer with the placement of locks to determine the feasibility (or infeasibility) under the assumed isolation levels. Note that the locks must be released as early as possible while respecting the isolation level.

S1: R1(A), R2(C), R1(B), W1(C), R2(A), W2(B), R1(A), R1(B), R2(A)

S2: R2(A), R1(B), R2(C), R1(A), W1(C), R2(C), W2(B), R2(A)

Determine whether the given schedules (**S1** and **S2**) are possible under the following isolation levels:

a. T1: Repeatable Read, T2: Read Committed (5*2=10 points)
b. T1, T2: Repeatable Read (5*2=10 points)

Your solution should have a format somewhat similar to something like this:

S1: SLOCK1(A); R1(A); SLOCK1(B); R1(B); REL1(A,B); etc. and an explanation if infeasible
S2: SLOCK1(A); R1(A); SLOCK1(B); R1(B); REL1(A,B); etc. and an explanation if infeasible

*If a transaction **Ti** is denied a database item **X**, use the syntax **DENIEDi(X)** to denote the same.*

Question 3: Isolation levels and conflicts (15 points)

Consider the following queries:

Q1: SELECT * FROM Temperatures WHERE cityName='Portland';
Q2: INSERT INTO Temperatures (tempId, temperature, cityName) VALUES (10001, 60, 'Portland');

Consider this as the initial state of the table Temperatures:

tempId	temperature	cityName
1	50	Portland
100	60	Champaign
1000	70	Mumbai
200	90	Portland

a. Let **S1** be the schedule defined as:

S1: Q1(T2), Q2(T1), Commit(T1), Q1(T2)

Assuming tuple-level locking during insertions, explain why phantom reads may occur under the Repeatable Read isolation level for **T1** and **T2**. Provide a table that includes locking operations and results for the actions in the schedule. (7.5 points)

b. Let **S2** be the schedule defined as:

S2: Q1(T2), Q2(T1), Q1(T2), Abort(T1)

Assume that the isolation level for **T1** and **T2** is Read Uncommitted. Briefly explain the occurrence of dirty reads in this scenario. Provide a table that includes locking operations and results for the actions in the schedule. (7.5 points)

Question 4 : Precedence graphs, Conflict-serializability (18 points)

Consider the given Schedules **S1** and **S2** consisting of transactions **T1**, **T2,** and **T3**.

S1: R1(A); R1(B); W1(B); R2(B); W2(B); R1(C); W1(C); R2(C); W2(C); R2(A); R3(A); W3(A);
S2: R3(C); W1(B); W2(B); W3(A); R1(C); W2(A); W1(A);

Test the given schedules for conflict-serializability using precedence graphs.

If a schedule is conflict-serializable, convert the schedule into an equivalent serial schedule and represent it in a tabulated form.
If a schedule is not conflict-serializable, explain why. Also, state the transactions and the pairs of actions (**Ri(X)** and/or **Wi(Y)**) that make the schedule not conflict-serializable, i.e., the transactions represent the edges that are part of the cycle. (9 * 2 = 18 points)

Question 5 : 2PL and Strict 2PL (42 points)

Consider a set of schedules **S1**, **S2** and **S3** by transactions **T1**, **T2** and **T3** on database objects **A**, **B** and **C**.

S1: W3(C), W1(A), R2(B), W3(B), R2(B), W1(B)
S2: W1(C); W3(B); R3(B); W2(A); R1(A); R3(A); R2(A); R2(C); R2(B)
S3: R1(A); R3(B); R2(C); R1(B); R3(C); W1(A); R1(A); R3(B); W2(C)

a. Determine whether schedules **S1**, **S2,** and **S3** can be produced by a Two-Phase Lock (2PL) scheduler. Provide tables with placements of shared locks (**SLOCK**), exclusive locks (**XLOCK**), and lock releases (**REL**) that follow 2PL (with downgrade) and obey lock-compatibility restrictions. (The formatting should be correct e.g. , a shared lock by **T1** on **B** is given by **SLOCK1(B)** and an exclusive lock by **T1** on **B** is given by **XLOCK1(B)**). Each column must correspond to a single transaction with a single operation in each row. **In the case of infeasibility, mark the first point of the schedule where a transaction fails to get the required lock and you can stop there**. Denote that a transaction has reached its end successfully by committing it. Note: Locks must be acquired only when necessary; however, you may assume the predictability of actions within each transaction to determine lock releases. (7 * 3 = 21 points)

b. For the same schedules **S1**, **S2,** and **S3**, determine whether they can be produced by a Strict Two-Phase Lock (2PL) scheduler. **If not, enforce Strict 2PL to obtain different schedules.** Proceed with the next transaction in the schedule if a transaction is blocked on some operation. In case there is a deadlock (i.e., all transactions are dependent on one another to release locks), you can stop there. Similar to part (a), provide the corresponding lock placement table. Denote that a transaction has reached its end successfully by committing it.
NOTE: Actions within each transaction should not be reordered. (7* 3 = 21 points)