Middle East Technical University

IS 503 - 2023/2024-Spring
Database Concepts and Applications
Assignment 4
Deadline: May 26th, 23:59

---

1) Consider the schedules S1, S2, and S3 below. State whether each schedule is strict, cascadeless, and recoverable. Explain your reasons by referring to the lecture notes. If a schedule is recoverable/cascadeless/strict you need to explain why it is so, but it is acceptable if you give a non-question-specific generic reason. However, if a schedule is not recoverable/cascadeless/strict, you need to specify the exact reason why it is not, by giving an example from the schedule itself (a single correct example is enough). (3*15 points)

- **S1**: R1(A); R2(B); W1(B); R3(A); R4(B); W4(A); R2(C); W3(B); R1(A); R3(B); C2; A1; A3; C4;
- **S2**: R1(A); R4(A); W2(B); W4(C); R3(A); R2(D); C2; R1(D); R3(B); C3; W1(C); C1; C4;
- **S3**: W2(C); R3(C); R1(B); W2(C); W2(D); R2(B); W4(B); R3(A); W4(C); C4; R1(B); W1(A); C1; C2; C3;

2) Consider the schedules S4, S5, and S6 below. State whether each schedule is conflict-serializable and view-serializable. Draw the precedence graph and show the related data items on arrows. You can simply give a single example of why they are not serializable (you may draw the precedence graph). If it is view-serializable, show the equivalent serial schedule(s). Explain your reasons by referring to the lecture notes. (3*15 points)

- **S4**: W1(B); W3(A); W1(D); R1(B); R2(C); R1(A); R2(B); R3(A); R2(C); R1(C); C3; C1; C2;
- **S5**: R1(A); W2(A); R3(A); W1(B); R2(B); R3(B); R2(C); W3(C); R1(B); C1; C2; C3;
- **S6**: R3(B); W2(C); R4(B); W1(D); R2(C); R3(A); W4(D); W4(C); R3(C); R1(B); R2(B); C2; R1(B); R4(A); R3(C); R1(B); C3; C4; C1;

3) While database management systems automatically handle transactions for us, we can manually create transactions and observe these concurrency problems. Follow the instructions below and observe how a phantom read occurs. You are supposed to use the database we created in your second assignment (10 pts).

- Open MySQL Workbench. If you have not imported the previously provided schema (met), import it.
- Open a new SQL tab. You will write the following statements in the same tab and do not run anything yet.
- Start your transaction with “START TRANSACTION;” since we want to do it manually. You do not need to set the autocommit feature off since this statement temporarily turns it off (after a commit or rollback, it turns back to the automatic mode).
- Using the met schema (the schema given with the second assignment), write an insert query on the user table to add a new user.
- Write a query to select all users.
- Finally, end your transaction with a rollback (“ROLLBACK;”).
- Notice that since we rollback our transaction, nothing should change in the user table. Run this transaction now. It should retrieve the users. Here, check the row you inserted. Although you did not commit, you should still see that your table has a new row. You are looking at a version of your table that officially never existed. Take a screenshot that demonstrates your query tab and the retrieved table. Highlight the row that is seemingly inserted.
- Open another SQL tab and retrieve the user table the way you normally would. Notice that this table does not have the new row since it was not committed. Take a screenshot of this version as well, and highlight how it differs from your previously retrieved table.
- Briefly explain what transactions, commit, and rollback do.

Submission Policy:

- You should name your visuals properly. Your file should be named as “studentId_HW4.pdf”, for example “e123456_HW4.pdf”.
- **No late submissions** will be accepted.