

Purdue University
Computer Science Department
CS44800 Homework 4 (Feedback)
Fall 2019

Just a reminder, if you are planning to submit your assignment handwritten, please write legibly. As usual, provide enough detail to support your answers. We encourage you to itemize your answers. **Also, we encourage you to answer the questions using your own words and not just copy text from the Textbook or Online sites.**

Make sure to answer all the questions.

Most common errors were:

Question 2.

- Many of you had issues with Q2a and Q2b.
- The sources needed to be clear and informative. You were asked to include an example for each one of your two sources. We were also looking for explanations which could explain how this will define FD.
- Few of you just gave the definition of FD and did not clearly explain how you will build those FDs. Points were deducted for unclear explanations. If ‘discussion about semantics’ were included, we considered it unclear for only one source.
- Few of you have given different types of FDs as answers. Essentially, you answered with the question. Points were deducted considering both sources included unclear explanations.
- Few of you mentioned attributes as a source but did not justify your answer how attributes can define FD. Points were deducted as unclear explanation for one source.
- For Q2b, many of you have ignored the difference between closure of a set of functional dependencies ($F: A \rightarrow B, B \rightarrow C, F^+ = \{A \rightarrow C\}$) and closure of a set of attributes ($A^+ = \{A, B, C\}$).
- For Q2c, few of you have not explained, or not clear enough why is lossless decomposition important.

Question 3.

- Very few of you have mentioned only integrity and constraint violations with dirty read violation. Need to include phantom violations which is prevented by serializable isolation.
- A small number of you have not even attempted to answer this question.

Question 4.

- Many of you did not get schedule b and d right. For example, for schedule b, the serial schedule would be $T3 \rightarrow T1 \rightarrow T2$ or, $r3(Y); w3(X); r1(Y); w1(X); r2(X); w2(Y)$.
- Some of you just printed out the transaction table, remember that does not explain the equivalent serial schedule or if there is a cycle or not.
- Remember, a schedule is conflict serializable if there is no cycle.
- We recommend you to practice building the precedence graph.
- If you are swapping operations, build a precedence graph after swapping to double check if there is any loop or not.
- Practice converting to equivalent serial schedule. You should start from the node with indegree 0. You can find a clear algorithm with example in this link - <https://www.geeksforgeeks.org/precedence-graph-for-testing-conflict-serializability-in-dbms/>

Question 5.

- Some of you have given a serial schedule as an example of conflict serializable i.e., $T1 \rightarrow T2 / T2 \rightarrow T1$. In detail, all operations from $T1$, and then added all operations from $T2$. This is not a valid example. We expected an example schedule which can be converted to serial schedule. See: <https://piazza.com/class/jyw0kgmjx1d3a1?cid=325>
- Few answers did not provide any explanation for non-conflict serializable example. Explain reasons why a serial is not conflict serializable.