CS143, Fall 2017: Homework 6

Problem A: 2 questions

Take a look at two-Phase Commit Protocol in Pictures described at:

http://www.exploredatabase.com/2014/07/two-phase-commit-protocol-in-pictures.html

and consider the following three types of indefinitely long failures:

F1. Indefinitely long failure of coordinator,

F2. Indefinitely long failure of a cohort,

F3. Indefinitely long failure of communication lines,

Can any to these failures result in:

(A) the non-failing coordinator to halt indefinitely , or

(B) a non-failing cohort to halt indefinitely ?

Describe at which Step of the "Protocol in Picture," F1 or F2 or F3 will cause (A) or (B) and explain the reasons for the problem.

Step 1. A failure of a cohort before step1 is not seen by the coordinator. If the coordinator (A) fails before sending the <prepare to T> a cohort (e.g. B) might decide that it has waited enough and it will abort. In that case when the coordinator comes back it will send the message to B, and thus get an abort message back– which it will then send to the other participants.

Step 2. Failure of coordinator between step 1 and 2, could cause it to miss some <Ready T> message. When the coordinator recovers it can either decide to abort, or restart from Step1.

A failure of a cohort or the communicating with it, can result in the <Ready T> message not be sent or received. After a while, the coordinator might restart from Step1 or decide to abort.

Step 3. Once Step 2 is completed, and the coordinator has received the response by all cohorts it takes a decision and communicates it back to all participants and and also implements the same decision locally. Any failure after Step3 is completed will not change the behavior of the coordinator.

The Step 4 slide on the web assumes that the decision was to commit. The decision is sent to all the cohorts which implement the decision. However, cohort failure or communication line failures might result in some cohort not learning the decision. Then this cohot WILL HAVE to WAIT INDEFINITELY, i.e. until it can contact back the coordinator or even a cohort and learn what was the decision and implement it locally.

Then the last three slides on the web repeat step 2-4 for case where the decision is to abort.

Problem B: 6 questions

We are given the $T(A, C, B, D, E)$

(a) $AD \to CE$

(b) $BC \to D$

(c) $AB \to A$

(d) $B \to E$

Q1. Is any of the previous FDs trivial?

Answer: yes, (c) is trival.

Q2. Is this schema BCNF? (To receive credit you must justify your answer.)

Answer:

Let us replace (a) with:

(a1) $AD \to C$ and

(a2) $AD \to E$

To check whether it is BCNF, let us try the FDs, starting with (a).

AD+ = {A, D, C, E}. Since B is missing, AD is a not a superkey. Thus (a) represents a BCNF violation.

Q3. Are all these FDs key FDs (Key FD: its left side contains a key)?

Answer: No, we have just shown that AD→ CE is not a key FD.

Q4. Is this schema 3NF?

Answer: This is difficult: for every FD $X \to A$ that fails BCNF we have to see if A belongs to some (minimal key).

For instance (a1) $AD \to C$. Well, let us find all the keys that contain C. For instance (A,B,C) is a key since (A,B,C)+= (A,B, C, D,E). Now (B,C)+=(B,C,D,E): A is missing. (A,C)+= (A,C), this is not a key. (A,B)+=(A,B,E) but C and D are missing. **Thus (A,B,C) is a candidate key and thus $AD \to C$ does not violate 3NF.** What about (a2) $AD \to E$? (A,B,C,D,E) is a superkey, but we cannot eliminate A or B because we have no FD with A or B on the right side. But if we keep B then we do not need E. **Thus there is no candidate key that contains E. Thus $AD \to E$ and $B \to E$ violate 3NF. Our original relation is neither BCNF nor 3NF.** Let us decompose it.

Q5. Transform the given FDs into an equivalent set of elementary FDs (no trivial FD, only one attribute on the right side, minimal left side).

Answer: We have done already by replacing (a) by (a1) and (a2).

Q6. Compute a lossless decompositions of T into BCNF relations where **no two relations share the same keys**.

Answer: We can start with (d) B → E and get: R1(B,E), R2(A, B, C, D)

Then ignoring trivial (c) let us use (b) BC → D on R2: (B,C)+ = (B,C, D).

So we decompose R2 into R21(B,C, D), R22(A,B,C)

Now R21 only contains the key dependency (b) and R22 contains no FD.

Thus R1(B,E), R21(B,C, D), R22(A,B,C)

is a lossless BCNF decomposition where no two relations have the same keys.

Other BCNF decompositions are possible.

Q7. Can you reconstruct the original relation from those obtained in the decomposition? What RA operation will you be using for that?

Answer: This is a lossless decomposition where the original relation can be reconstructed by taking the natural join of the BCNF decomposition.

Q8. Is your decomposition FD preserving? To receive credit you must justify your answer.

Answer: From the keys in the normalized relations we can infer B→ E and BC→ D. Can we infer (a1) and (a2) from these? (A,D)+= (A,D). The answer is NO.