Course : EECS 495. Intro to Database Systems
Name : Sangrin Lee
Student ID : 2999428

Q1-a. $X \rightarrow Y$ and $Y \rightarrow Z$ imply $X \rightarrow YZ$
1) $X \rightarrow Y$ (Given)
2) $Y \rightarrow Z$ (Given)
3) $X \rightarrow Z$ (Transitivity rule using 1), 2))
4) $X \rightarrow YZ$ (Union rule using 1), 3))

Q1-b. $X \rightarrow Y$ and $Z \rightarrow W$ imply $XZ \rightarrow YW$
1) $X \rightarrow Y$ (Given)
2) $XW \rightarrow YW$ (Augmentation rule using 1))
3) $Z \rightarrow W$ (Given)
4) $XZ \rightarrow XW$ (Augmentation rule using 3))
5) $XZ \rightarrow YW$ (Transitivity rule using 2), 4))

Q1-c. $XY \rightarrow Z$ and $Z \rightarrow X$ imply $Z \rightarrow Y$
It disapproves by setting up a relation which does violate it.

| X | Y | Z |
|---|---|---|
| x1 | y1 | z1 |
| x1 | y2 | z1 |

Both $XY \rightarrow Z$ and $Z \rightarrow X$ hold, but not $Z \rightarrow Y$.

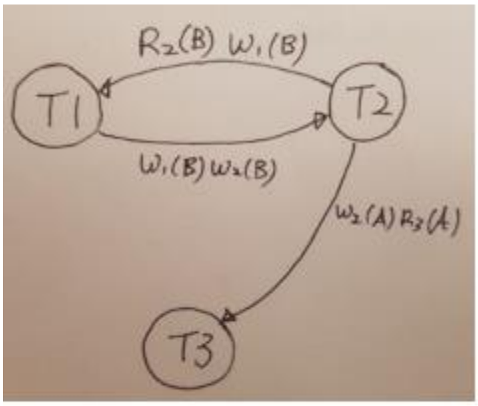
Q2.
1) $ABH \rightarrow C$ does not violate BCNF since attribute closure of ABH is all attributes and therefore ABH is superkey.
2) $A \rightarrow DE$ violates BCNF since attribute closure of A is ADE and therefore A is not a superkey.
(attribute closures of F is also not all attributes, so it is not a superkey as well)
3) Split R into
R1(ADE) with F1 = ($A \rightarrow DE$)
R2(ABCFGH) with F2 = ($ABH \rightarrow C$, $BGH \rightarrow F$, $F \rightarrow AH$, $BH \rightarrow G$)
4) R1 is in BCNF (decomposition is lossless since A is a key of R1)
5) In R2, $ABH \rightarrow C$ or $BGH \rightarrow F$ does not violate BCNF since BH is a key of R2.
6) $F \rightarrow AH$ violates BCNF since attribute closure of F is AH and therefore F is not a superkey.
7) Split R2 into
R21(FAH) with F21 = ($F \rightarrow AH$)
R22(BCFG) with no FDs.
8) Both R21 and R22 are in BCNF (decomposition is lossless since F is a key of R21)
Therefore, BCNF schemas are below
R1(ADE) with F1 = ($A \rightarrow DE$)
R21(FAH) with F21 = ($F \rightarrow AH$)
R22(BCFG)

Q3. Yes.
Case 1) A → B holds, but B → A does not. This means that A is a key, and BCNF condition is satisfied since the only FD is A → B. (Same as A → AB since we can derive it from A → B. Then, A is a key and BCNF is satisfied)
Case 2) B → A holds, but A → B does not. This means that B is a key, and BCNF condition is satisfied since the only FD is B → A. (Same as B → AB since we can derive it from B → A. Then B is a key and BCNF is satisfied)
Case 3) No functional dependencies, so BCNF is basically satisfied.
Case 4) Both A → B and B → A hold. A and B are both keys, therefore they determine all attributes.

Q4.
F = {A → BC, B → AC, C → AB} can be simplified to {A → B, A → C, B → A, B → C, C → A, C → B}
Canonical cover of F is a "minimal" set of functional dependencies equivalent to F, having no redundant dependencies or redundant parts of dependencies.
First, A → C(derived from A → B, B → C), B → A(derived from B → C, C → A),
C → B(derived from C → A, A → B) are redundant.
So, we can get A → B, B → C, C → A.
Second, A → B(derived from A → C, C → B), B → C(derived from B → A, A → C),
C → A(C → B, B → A) are redundant.
So, we can get A → C, B → C, C → B
Therefore, two canonical covers are
1. A → B, B → C, C → A
2. A → C, B → A, C → B

Q5-a.
All Candidate keys of R : A, BC, CD, and E
Q5-b.
A decomposition {R1, R2} is a lossless-join decomposition if and only if at least one of dependencies is (R1 ∩ R2 → R1 or R1 ∩ R2 → R2).
R1 ∩ R2 = A and A is a candidate key. Therefore, R1 ∩ R2 → R1.
In addition, (A → BC) => (A → ABC) => (R1 ∩ R2 → R1) shows that this decomposition is lossless

Q6-a.
Precedence graph for this schedule is below




This schedule is not conflict serializable since its precedence graph for this schedule contains cycle.

6-b.

| T1 | T2 |
|---|---|
| read(X) | |
| | read(X) |
| read(Y) | |
| write(X) | |
| | write(X) |
| write(Y) | |
| | read(Y) |
| | write(Y) |