Yasemin Akın 22101782

CS 353 Spring 2024
Homework 5

Q1. (b), (c), (e), (f), and (h) are violated.

Q2.

(a) Holds.

(b) Does not hold.

(c) R is not in BCNF. Both B → D and C → D violate BCNF.
Decomposing B → D into two relations: **R1(B, D)** containing all attributes of the violating functional dependency and **R2(A, B, C)** containing remaining attributes plus the determinant of the violating functional dependency.
R1 and R2 are now both in BCNF.

Q3.

(a) It is lossless. Because for a decomposition to be lossless, one of the following should hold: C → A, B, C or C → C, D [From the lossless join test: (R1∩R2) → R1or (R1∩R2) → R2, here R1∩R2 = C]. We already have C → A in the given dependencies, which implies C → A, B, C. Therefore, the decomposition is lossless.

(b) It is not lossless. Because for a decomposition to be lossless, one of the following should hold: B → A, B or B → B, C, D [From the lossless join test: (R1∩R2) → R1or (R1∩R2) → R2, here R1∩R2 = C]. We do not have functional dependencies B → A or B → C, D. Therefore, the decomposition is not lossless. Example R instance:

A	B	C	D
a1	b1	c1	d1
a2	b1	c2	d2

Decomposing R into R1 and R2:
R1: {(a1,b1), (a2, b1)}
R2: {(b1, c1, d1), (b1, c2, d2)}
Then, when we perform a natural join on R1 and R2:
R’: {(a1, b1, c1, d1), (a1, b1, c2, d2), (a2, b1, c1, d1), (a2, b1, c2, d2)}
R' contains tuples that were not in the original relation R, which means the join of R1 and R2 has extra information not present in the original relation. Thus, we have lost information about which tuples existed. This demonstrates that the decomposition of R into R1(A, B) and R2(B, C, D) is not lossless.

Q4.

(a) To be in BCNF, for every non-trivial functional dependency, X → Y, X must be a superkey of R. At A → BC, A is not a superkey because it does not determine all attributes of R. Since at least one of the functional dependencies has a determinant that is not a superkey, R is not in BCNF.

(b) Decomposing R by the violating FDs:
Decomposing R according to A → BC:
R1(A, B, C) with A → BC is in BCNF because A is a superkey of R1.
R2(A, D, E, F) with AD → F and AF → E, if AD or AF are superkeys of R2, then R2 is in BCNF.
If AD and AF are not superkeys, further decomposition is needed, which we would continue iteratively until all relations are in BCNF. Since neither AD nor AF alone can determine all attributes in R2 (they each determine one other attribute but not both), neither is a superkey, and R2 is not in BCNF.
Decomposing R2 according to AD → F:
R3(AD, F) with only FD AD → F is in BCNF because AD is a key for R3.
R4(A, D, E) is still not in BCNF because AF → E violates it.
Decomposing R4 according to AF → E:
R5(AF, E) in BCNF because AF is a key for R5.
R6(A, D) in BCNF because it has no non-trivial FDs.
The final BCNF relations resulting from this decomposition:
R1(A, B, C)
R3(AD, F)
R5(AF, E)
R6(A, D)
Each of these relations is in BCNF because they each have one functional dependency, and the left-hand side of each dependency is key to that relationship.

(c) AD is the candidate key.

(d) Only FD A → BC violates the condition for 3NF because A is not a superkey, and B and C are non-prime attributes. Therefore, we can say that R is not in 3NF due to dependency A → BC.

(e) The FD set itself is the canonical cover because all of them are necessary.
In the given set, there are no obvious extraneous attributes since none of the dependencies share common attributes on the right side.
A → BC is not redundant because we can not determine BC from A using the other dependencies.
AD → F is not redundant because we can not determine F from AD using the other dependencies.
AF → E is not redundant because we can not determine E from AF using the other dependencies.
Also, no FDs can be combined because none has the same determinant.

(f) Using the canonical form:

A → BC creates R1(A, B, C)
AD → F creates relation R2(A, D, F)
AF → E creates relation R3(A, F, E)
Also, two candidate keys, AD and AF, are already included in R2 and R3, respectively, so no new relation needs to be added, and there are no redundant relations since none is a subset of another. Each relation reflects one of the FDs. So, final 3NF relations are R1, R2 and R3.

Q5. First, it is needed to check if R is in BNF:

A → B: A is a superkey because it is on the left side of both multivalued dependencies and functional dependencies that include all other attributes.
B → E: B is not a superkey since it doesn't determine A, C, or D. However, since B is dependent on A (because A → B), and A is a superkey, the relation can still be in BCNF.

Secondly, it is necessary to check if all multivalued dependencies are trivial or if each one is a dependency on a superkey:

A →> C and A →> D have A as their left-hand side, and A is a superkey.

Since R satisfies both the BCNF condition and the 4NF condition regarding multivalued dependencies, the relation R is in 4NF.