Winter 2022 - CS540 – Assignment 3 (Written) (Group 15)

Full Name	OSU Email ID	ONID
Bharghav Srikhakollu	srikhakb@oregonstate.edu	934289891
Akhil Sai Chintala	chintala@oregonstate.edu	934409906

1. **<u>Constraint Reference:</u>**

(a) Given that X, Y, W, Z are attributes in a relation, using Armstrong's axioms, prove that if we have X → Y and Y W → Z, then XW → Z.

Given X → Y and Y W → Z

- ✓ X → Y

  According to Armstrong's axiom – Augmentation

  We can add 'W' to both the sides of the relation and represent as below

  XW → YW

- ✓ Now we have XW → YW and given relation as YW → Z

  According to Armstrong's axiom – Transitivity

  We can conclude XW → YW and YW → Z as below

  XW → Z

Hence, we proved that if we have X → Y and YW → Z, then XW → Z

(b) Given that X, Y, Z are attributes in a relation, using Armstrong's axioms, prove that if we have X → Y and X → Z, then X → YZ.

Given X → Y and X → Z

- ✓ X → Y

  According to Armstrong's axiom – Augmentation

  We can add 'X' to both the sides of the relation and represent as below

  XX → XY which is equal to X → XY

- ✓ X → Z

  According to Armstrong's axiom – Augmentation

  We can add 'Y' to both the sides of the relation and represent as below

  YX → YZ which is equal to XY → YZ

- ✓ Now we have X → XY and XY → YZ

  According to Armstrong's axiom – Transitivity

  We can conclude X → XY and XY → YZ as below

  X → YZ

Hence, we proved that if we have X → Y and X → Z, then X → YZ

2. **<u>Schema Decomposition:</u>**

Consider the relation schema R with attributes A, B, C, and D and the following functional dependencies: AB→C, AC→B, B→D, BC→A.

(a) List all keys for R.

Given FD's are AB→C, AC→B, B→D, BC→A

First, we will find closure.

AB+ = ABCD   BC+ = BDCA   AC+ = ACBD

So, from the above-mentioned closures we can say that AB, BC, AC are the keys for relation schema R.

(b) Is R in BCNF? If it is not, decompose it into a collection of BCNF relations.

According to definition, Definition 11.2.1 (Alice Book), Relation R is in Boyce-Codd normal form (BCNF) if and only if, for each non-trivial Functional Dependency X → Y in R, X is a super key of R. Every attribute depends only on super keys.

Given FD's are AB→C, AC→B, B→D, BC→A

Here B→D we can say that B is not a super key, so relation R is not in BCNF.

Decomposition of BCNF: Since we found out the keys, we will decompose it into BCNF as below. We decompose into two, one with the attributes of FD which violates the BNCF (B→D)

ABC	**BD**
<u>AB</u>→C	<u>B</u>→D
<u>AC</u>→B	
<u>BC</u>→A	

From this decomposition, we can say that R1(ABC) and R2(BD) are in BCNF.

(c) Is R in 3NF? If it is not, convert it into a collection of 3NF relations.

According to definition, Definition 11.2.11 (Alice Book), Relation R is in 3^{rd} Normal Form (3NF) if for each non-trivial Functional Dependency (FD), X → Y in R, X is a super key or Y is a prime attribute (each element of Y is part of candidate key)

Relation schema R is not in 3NF. Given Functional Dependencies are already on a minimal basis. Relation can be converted into a collection of 3NF as below.

R1(ABC) and R2(BD)

(d) Prove that, if relation R has only one simple key, it is in BCNF if and only if it is in 3NF.

Given relation R has only one simple key. To prove it is in BCNF if and only if it is in 3NF condition, let's take R is in 3NF.

Consider a relation $X \rightarrow B$. For this relation to be in 3NF form either X is a super key or B is part of the key. Since R has only one simple key, B has just one attribute and it could be the key which in turn says that X is a super key.

Hence $X \rightarrow B$ is in BCNF in any of these cases. So, relation R with one simple key is in BCNF.

3. **Information Preservation:**

(a) Suppose you are given a relation R (A, B, C, D) with functional dependencies B→C and D→A. State whether the decomposition of R to S1(B, C) and S2(A, D) is lossless, or dependency preserving and briefly explain why or why not.

Given relation R (A, B, C, D) with functional dependencies B→C and D→A

Decomposition of R is S1(B, C) and S2(A, D)

For a relation to be in lossless decomposition, union of decomposition relations (S1 U S2) should be equal to relation R (A, B, C, D). This is true for the given relation, but the intersection of decomposition relations ($S1 \cap S2$) should be equal to null which is not true in this case. Hence for this relation, the decomposition of R is not lossless decomposition.

The functional dependency B→C can be ensured in S1(B, C) and D→A can be ensured in S2(A, D). So, it is dependency preserving decomposition.

(b) Prove that the 3NF synthesis algorithm produces a lossless-join decomposition of the relation containing all the original attributes.

Consider a set of relations R1, R2, … Rn and the Functional Dependencies sets are equivalent. For the relation, key is K, minimal basis is M and set of FD's is U. We can prove that schema is in 3NF if the FD in one set has only one attribute in their right hand side. Let's consider each FD, $X \rightarrow Y$ in M, we get a relation S if XY is not covered by any relation. So, we have tuples which are generated by S, attributes contained in key K and a join on such tuples will generate a lossless decomposition.