Assignment-4 Solutions

1

Consider the relation scheme $R=\{A,B,C,D,E,F,G,H,I,J\}$ and the set of functional dependencies (FDs) $F=\{AB \to C, B \to EF, AD \to GH, G \to I, H \to J\}$.

What is the key for R?

Hint: For a relation, a key is a set of attributes that functionally determines all attributes and none of its subsets does.

✗ $\{A,B\}$

- $\{A,B\}+ = \{A,B,C,E,F\}$. Clearly, it does not determine all the attributes.

✓ $\{A,B,D\}$

- $\{A,B,D\}+ = \{A,B,D,C,E,F,G,H,I,J\}$. Thus, we get all attributes and it is minimal. So, it is the key.

✗ $\{A,B,D,G,H\}$

- $\{A,B,D,G,H\}+ = \{A,B,D,C,E,F,G,H,I,J\}$. Although we get all attributes, it is not minimal. So, it is not the key.

✗ $\{A\}$

- $\{A\}+ = \{A\}$. Clearly, it does not determine all the attributes.

2

Let $attr(f)$ be the set of attributes that make up the FD f. For e.g., $attr(AB \to C)=\{A,B,C\}$

Suppose a set of FDs, F, holds over a relation R such that, $\forall f_i, f_j \in F$ and $i \neq j \forall fi, fj \in F$ and $i \neq j$, $attr(f_i) \cap attr(f_j) = \emptyset attr(fi) \cap attr(fj) = \emptyset$.

Every possible BCNF decomposition of R will be _____

✓ lossless and dependency-preserving.

- All BCNF decompositions are lossless. Because no two FDs have the same attributes, whenever we decompose a relation using a FD f that violates the BCNF condition, no FD will have its attributes split across relations. This ensures that all FDs will belong to a single relation and all of them can be checked without joining relations.

✗ lossless and not dependency-preserving.

✗ lossy and dependency-preserving.

✗ lossy and not dependency-preserving.

3

If F is a set of FDs, then its closure, F+, is the set of all the FDs that can be derived from the FDs in F.

Suppose $F = \{C \to D, E \to C, A \to D, AC \to E, CD \to B, BC \to A\}$.

Which of the following FD is not in F+?

✓ $A \to DE$

- The only FD involving AA at the LHS is $A \to D$. E cannot be functionally determined by D or AD using Armstrong's axioms and the existing set of FDs.

✗ $E \to BC$

- E is a key, so it functionally determines everything.

✗ $C \to A$

- C is a key, so it functionally determines everything.

❌ C→E

- C is a key, so it functionally determines everything.

❌ BC→AED

- C is a key, so it functionally determines everything.

4

A relation R(A,B,C,D) has the following set of FDs,

F={A→B,A→C,CD→E,B→D,E→A}.

Which of the following functional dependencies is NOT implied by the above set?

Hint: Use Armstrong’s Axioms.

❌ {CD→AC}

- From CD→E and E→A given in the question, we get CD→A by transitivity. CD→C by reflexivity. So, CD→AC follows from the given FDs.

✓ {BD→CD}

- BD→CD cannot hold true since we only have BD→BD from reflexivity. So, BD→CD is not implied and is the correct answer.

❌ {BC→CD}

- B→D is given and C→C holds by reflexivity. So, BC→CD follows from the given FDs.

❌ {AC→BC}

- A→B is given and C→C holds by reflexivity. So, BC→CD follows from the given FDs.

5

Consider the relation $R(A,B,C,D,E)$ and the following set of FDs that hold over R, $F=\{BD \rightarrow C, A \rightarrow BC, ABC \rightarrow E, E \rightarrow D\}$.

Let T be the sum of the number of attributes in the relations obtained from a BCNF decomposition. What is the smallest possible value of T that can be obtained from decomposing R into BCNF? Is there a unique BCNF decomposition that corresponds to this value in this case?

✓ 6, yes

- In general, a BCNF decomposition is not unique but for this case there is no need to try all possible decompositions. Recognize that whenever you split a relation on a FD f, you obtain two relations, $R1$ and $R2$, where $R1$ contains the attributes in f and $R2$ contains the remaining attributes PLUS the attributes on the LHS of f. If a relation is not already in BCNF, then a heuristic to find the smallest T would be to find a FD f such that, 1) splitting on it would prevent further splitting of $R2$ through other FDs (which avoids adding back attributes on the LHS of the FDs that are used to further split $R2$) and 2) f has the fewest number of attributes on its LHS (so that we add back the smallest possible number of attributes). In the above scenario, this is done by recognizing that first splitting on $E \rightarrow D$ would prevent further splitting as the resulting relations will already be in BCNF. The only other FD that causes a BCNF violation is $BD \rightarrow C$ and decomposing R using this FD will only give a larger T.

✗ 6, no

- See explanation for correct answer.

✗ 7, yes

- See explanation for correct answer.

✗ 7, no

- See explanation for correct answer.

✗ 8, no

- See explanation for correct answer.

✕ 5, yes

- See explanation for correct answer.

6

Consider a relation $R(L,M,N,O,P)$ which is decomposed into relations $R1(L,N,O)$ and $R2(M,N,O,P)$.

The FDs $F=\{M \to N, MO \to P, MN \to O, NO \to L\}$ hold over RR.

Is the above decomposition in BCNF?

✓ Yes.

- $NO \to L$ is the only FD that is in $R1$ and NO is a key so there is no BCNF violation. M is a key for $R2$ because using the first three FDs we get $M \to NOP$. This means the first three FDs have superkeys on their left-hand side, which means they do not violate the BCNF condition for $R2$.

✕ No.

7

A relation R is in BCNF if and only if for every non-trivial functional dependency $X \to Y$, X is a superkey.
A relation R is in 3CNF if and only if for every non-trivial functional dependency $X \to Y$, X is a superkey or Y is a prime attribute.

Which of the following are true?

✓ If all the attributes of relation are prime attributes, then the relation is always in 3NF.

- If all the attributes of a relation are prime attributes, then the right hand side of all the functional dependencies will be prime attributes. Therefore, the relation will be in 3NF.

❌ BCNF decompositions are always lossless and dependency preserving.

- In BCNF, every non prime attribute should be functionally dependent on any of super key in schema. If there exists any FD, which don't follow this, then for that case we have to separate it into new relation. Now if any of other FD uses previous FD, Then this creates non preservation of FD in BCNF

✓ Every binary relation (a relation with only 2 attributes) is always in BCNF.

- For a relation with only two attributes, BCNF will hold trivially since there is at most one functional dependency here and the left hand side of the functional dependency will be a superkey by definition.

✓ If a relation is in BCNF form, then 3NF is also satisfied.

- The condition for a 3CNF is a subset of the condition for BCNF. Hence, BCNF implies 3NF.

8

Which of the following normal forms hold in $R(A,B,C,D,E)$ under the following given functional dependencies.

The FDs given on RR are $F=\{CD\to AE, ABC\to D\}$

❌ BCNF

- $ABC\to D$ is in BCNF. Let us check $CD\to AE$. CD is not a super key, so this dependency is not in BCNF. Hence, R is not in BCNF.

❌ 3NF

- $ABC\to D$ we don’t need to check for this dependency as it already satisfied BCNF. Let us consider $CD\to AE$. Since E is not a prime attribute, R is not in 3NF.

✓ 1NF

- Since every attribute is single-valued in this relation, we can say that RR is under 1NF.

9

Consider the following instance of a relation $R(A,B,C,D)$. Which of the following multivalued dependencies does R not satisfy?

✓ $AB \twoheadrightarrow C$

- Missing the tuples (1, 2, 3, 5) and (1, 2, 4, 4).

✗ $AB \twoheadrightarrow CD$

- This is a trivial MVD and is always satisfied as it contains all the attributes of the relation.

✗ $C \twoheadrightarrow D$

- The value of DD for tuples 1 and 2 are the same so swapping them still gives these two tuples.

✗ $CD \twoheadrightarrow AB$

- This is a trivial MVD and is always satisfied as it contains all the attributes of the relation.

10

The relation $R(A,B,C,D,E)$ has MVDs $\{AB \twoheadrightarrow C, C \twoheadrightarrow D\}$ and is known to contain tuples $(1,2,3,4,5)$ and $(1,2,6,7,8)$. Which of the following tuple must also be in RR?

✓ $(1,2,3,7,5)$

- First generate $(1,2,3,7,8)$ using the first MVD then use the second MVD $C \twoheadrightarrow D$ and $(1,2,3,4,5)$ to generate this tuple.

✗ $(1,2,5,6,7)$

- This tuple will not be generated from the given tuples and MVDs so we cannot be sure it will be in R.

✕ (1,2,3,6,7)

- This tuple will not be generated from the given tuples and MVDs so we cannot be sure it will be in R.

✕ (1,2,4,5,7)

- This tuple will not be generated from the given tuples and MVDs so we cannot be sure it will be in R.

✕ (1,2,4,5,6)

- This tuple will not be generated from the given tuples and MVDs so we cannot be sure it will be in R.
