SWEN304

Assignment 2

W. Burroughs - 300467432

**Question 1:**

**1)** A → B: The value “a” for attribute A is the same for both tuples, while there are two different values for B “b’ and “d”. The values of A do not uniquely determine the values of B, therefore this function dependency does not hold.

**2)** B → A: Looking at the first tuple, attribute B has the value of “b” while attribute A has the value of “a”. For the second tuple, attribute B has the value of “d” while attribute A has the value of “a”. The values of attribute B uniquely identify values of attribute A, thus this functional dependency holds.

**3)** C → B: For both tuples attribute C has the value of “3”, whereas there are two different values for attribute B: “b” and “d”. The determinant C does not uniquely identify the dependent B, therefore this functional dependency does not hold.

**4)** A → C: Both tuples have the same values for attributes A and C, “a” and “3” respectively. Attribute A therefore uniquely identifies attribute C, therefore this functional dependency holds.

**Question 2:**

**1)** F = {BC→A, A→D}

The attributes A, B, C, D are all atomic - There are no repeating groups, no mixing of data types, and the relation has a key of BC meaning all tuples should be uniquely identifiable. Therefore, N satisfies 1NF.

2NF states that each non-prime attribute must be functionally dependent on the entire candidate key. Since BC→A & A→ D, we can say that D is transitively dependant on the key {BC}. Therefore N satisfies 2NF.

3NF states that no non-prime attribute in the table is transitively dependent on the candidate key. Attribute D is transitively dependent on the candidate key {B C} due to D being functionally dependent on A and A being functionally dependent on BC. Therefore N is not in 3NF.

BCNF requires that every non-trivial functional dependency has a determinant that is a candidate key. The functional dependency A→D violates BCNF since A is not a superkey and it determines a non-prime attribute D. Therefore N is not in BCNF

In summary N is in 1NF & 2NF but does not meet the requirements of 3NF or BCNF.

**2)** F = {BC→D, B→A}

Same kind of case as in 1), attributes A, B, C, D are all atomic - There are no repeating groups, no mixing of data types, and the relation has a key of BC meaning all tuples should be uniquely identifiable. Therefore, N satisfies 1NF.

2NF states that each non-prime attribute must be functionally dependent on the entire candidate key. Since there is only one candidate key: {BC}, the functional dependency B→A violates 2NF since A is only partially dependent (B is a prime-attribute but only part of the candidate key). Therefore N is not in 2NF.

3NF states that no non-prime attribute in the table is transitively dependent on the candidate key. N violates 3NF because attribute A is transitively dependent on the candidate key {BC} through the functional dependency B→A.

BCNF requires that every non-trivial functional dependency has a determinant that is a candidate key. BC→D satisfies this requirement, however attribute B determines the non-prime attribute A - This means A is only partially determined by the candidate key {BC} - Thus in this case N does not satisfy BCNF.

In summary, N is in 1NF, but does not meet the requirements of 2NF, 3NF, or BCNF.

**3)** F = {BC→A, BC→D}

Same kind of case as in 1), attributes A, B, C, D are all atomic - There are no repeating groups, no mixing of data types, and the relation has a key of BC meaning all tuples should be uniquely identifiable. Therefore, N satisfies 1NF.

2NF states that each non-prime attribute must be functionally dependent on the entire candidate key. The non-prime attribute A is determined by the entire key BC (BC→ A) & the other non-prime attribute D is determined by entire key BC as well (BC→D). N therefore satisfies 2NF.

3NF states that no non-prime attribute in the table is transitively dependent on the candidate key. There are no transitive functional dependencies in this case, both non-prime attributes are dependent on the entire candidate key. Therefore, N satisfied 3NF.

BCNF requires that every non-trivial functional dependency has a determinant that is a candidate key. Since every non-trivial functional dependency in the relation has a superkey as its determinant (BC→A, BC→D) - N satisfies BCNF.

In summary, N is in 1NF, 2NF, 3NF, and BCNF.

**4)** F = {BC→AD, A→C}

Same kind of case as in 1), attributes A, B, C, D are all atomic - There are no repeating groups, no mixing of data types, and the relation has keys of {BC,BA} meaning all tuples should be uniquely identifiable. Therefore, N satisfies 1NF.

**Question 3:**

Given the functional dependencies:

F = {AC → B, BD → E, A → D}

1. B → B (Reflexivity)
2. Augment A → D (From F) with B → B gives BA → BD
3. A → A (Reflexivity)
4. Augment AC → B with A → A gives AC → BA
5. Transitivity: AC → BA BA → BD gives AC → BD
6. Transitivity: AC → BD BD → E gives AC → E