SWEN304
Assignment 3

W. Burroughs - 300467432

<u>**Question 1:**</u>

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 dependent 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 the 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:

$$F = \{AC \rightarrow B, BD \rightarrow E, A \rightarrow D\}$$

1. $B \rightarrow B$ (Reflexivity)
2. Augment $A \rightarrow D$ (From F) with $B \rightarrow B$ gives $BA \rightarrow BD$
3. $A \rightarrow A$ (Reflexivity)
4. Augment $AC \rightarrow B$ with $A \rightarrow A$ gives $AC \rightarrow BA$
5. Transitivity: $AC \rightarrow BA$, $BA \rightarrow BD$ therefore we have $AC \rightarrow BD$
6. Transitivity: $AC \rightarrow BD$ $BD \rightarrow E$ therefore we have $AC \rightarrow E$

Question 4:

F = {C → A, B → AC, D → A, AB→ D}

Step 1: Express the functional dependencies in standard form and ensure singleton RHS, reducing the RHS of the FD where possible. B → AC can be decomposed into B → A & B → C giving:

G = {C → A, B → A, B → C, D → A, AB → D}

Step 2: Need to determine whether or not we have any extraneous attributes on the LHS. AB → D is the only one, therefore we need to determine if A+ or B+ contains D.

A+ → A (Reflexivity)

B+ → B (Reflexivity)
B+ → ABC (From B → A & B → C)
B+ → ABCD (From AB → D)

Since B+ contains A, we can remove A from AB → D, which gives us B → D

G1 = {C → A, B → A, B → C, D → A, B → D}

Step 3: Remove the redundant dependencies. Since A is transitively dependent on B due to B → C & C → A the functional dependency B → A is redundant and can be removed from the set, giving us:

For C → A
C+ → C (Reflexivity)
No other way to get A if we ignore C → A, therefore we need C → A in the final G.

G2 = {C → A, B → A, B → C, D → A, B → D}

For B → A
B+ → B (Reflexivity)
B+ → CD (From B → C & B → D)
B+ → CDA (From either C → A or D → A)
B+ contains A, therefore we don't need B → A this in the final G.

G2 = {C → A, B → C, D → A, B → D}

For B → C
B+ → B (Reflexivity)
B+ → BD (From B → D)
B+ → BDA (From D → A)
B+ does not contain C, therefore we can't eliminate it and we need B → C in final G.

$$G2 = \{C \rightarrow A, B \rightarrow C, D \rightarrow A, B \rightarrow D\}$$

For $D \rightarrow A$

$D+ \rightarrow D$ (Reflexivity)

$D+$ does not contain A, therefore we need $D \rightarrow A$ in the final G.

$$G2 = \{C \rightarrow A, B \rightarrow C, D \rightarrow A, B \rightarrow D\}$$

For $B \rightarrow D$

$B+ \rightarrow B$ (Reflexivity)

$B+ \rightarrow C$ (From $B \rightarrow C$)

Does not contain D, therefore we need $B \rightarrow$ In final G.

Therefore, the minimal cover of F is:

$$G = \{C \rightarrow A, B \rightarrow C, D \rightarrow A, B \rightarrow D\}$$

Question 5:

Given N (R, F) where R = {A, B, C, D} and F = {C→A, B→D}

1) To identify all the keys for schema we need to consider all of the functional dependencies and their closure sets.

{A}+ → {A}
{B}+ → {B, D}
{C}+ → {C, A}
{D}+ → {D}
{A, B}+ → {A, B, D}
{A, C}+ → {A, C}
{A, D}+ → {A, D}
{B, C}+ → {A, B, C, D} = Candidate key
{B, D}+ → {B, D}
{C, D}+ → {A, C, D,}
{A, B, C}+ → {A, B, C, D} = Superkey
{A, B, D}+ → {A, B, D}
{A, C, D}+ → {A, C, D}
{B, C, D}+ → {A, B, C, D} = Superkey
{A, B, C, D}+ → {A, B, C, D} = Superkey

The closure sets that contain all of the attributes from relation R (A, B, C, D) are considered to be keys. Therefore there are four keys: {B, C}, {A, B, C}, {B, C, D}, {A, B, C, D} & {B, C} is the minimum candidate key.

2) Determining the highest normal form:

All attributes appear to be atomic, assuming each one only contains one value of appropriate type per row. Therefore the relation is in 1NF.

Taking the candidate key as {B, C} this gives us the non-prime attributes of A & D. In order for the schema to satisfy 2NF, both A & D need to be functionally dependent on the entire candidate key. Both A & D have partial dependencies on the minimum candidate key: C → A & B → D, therefore the relation schema is not in 2NF.

Since the relation does not satisfy 2NF requirements we can deduce that it is not in 3NF. 3NF states that each non-prime attribute must be entirely dependent on the key, essentially disallowing transitive dependencies - This schema does not satisfy 3NF as in all of the given functional dependencies, the dependent attributes have partial dependencies on the minimum candidate key.

The relation does not satisfy 2NF or 3NF, therefore it can not satisfy BCNF requirements. BCNF states that each non-trivial functional dependency must be dependent on the candidate key - This schema does not satisfy BCNF as all of the given functional dependencies have partial dependencies on the minimum candidate key.

3) Decomposing input relation into 3NF:

1. Canonical Functional Dependency Cover:

$$R = \{A, B, C, D\} \text{ and } F = \{C \rightarrow A, B \rightarrow D\}$$

F is already in canonical form, therefore no need to merge based on LHS.

2. Creating 3NF Relations:

Need to create a relation schema for each functional dependency in F:

$$R0 = \{B, D\} \text{ and } F0 = \{B \rightarrow D\}$$
$$R1 = \{A, C\} \text{ and } F1 = \{C \rightarrow A]$$

3. Check for Keys in 3NF Relations:

Need to check if any of the 3NF relations contains a key from the original relation, this is not the case, therefore we need to add another relation whose schema is a key of the original relation and has no functional dependencies. The minimum key was {B, C} therefore:

$$R2 = \{B, C\} \text{ and } F2 = \{\text{No FD's}\}$$

4. Look for Attribute Inclusion:

Need to check if any relation includes all of the attributes found in another relation, this is not the case, therefore no relations need to be removed.

5. Completed Decomposition:

$$R0 = \{B, D\} \text{ and } F0 = \{B \rightarrow D\}$$
$$R1 = \{A, C\} \text{ and } F1 = \{C \rightarrow A]$$
$$R2 = \{B, C\} \text{ and } F2 = \{\text{No FD's}\}$$

4) Verification of Lossless Join:

$$R = \{A, B, C, D\} \text{ and } F = \{C \rightarrow A, B \rightarrow D\}$$

$$R0 = \{B, D\} \text{ and } F0 = \{B \rightarrow D\}$$
$$R1 = \{A, C\} \text{ and } F1 = \{C \rightarrow A]$$
$$R2 = \{B, C\} \text{ and } F2 = \{\text{No FD's}\}$$

The union of the decomposed relations:

$$Temp = R0 \cup R1 \cup R2 = \{B, D\} \cup \{A, C\} \cup \{B, C\}$$
$$Temp = \{A, B, C, D\}$$

Check if the original relation R is equal to Temp:

$$R = \{A, B, C, D\}$$
$$Temp = \{A, B, C, D\}$$

Comparing the Functional Dependencies:

$$F\_temp = F0 \cup F1 \cup F2 = \{B \rightarrow D\} \cup \{C \rightarrow A\} \cup \{\} = \{B \rightarrow D, C \rightarrow A\}$$

Functional dependencies F are preserved:

$$F = \{C \rightarrow A, B \rightarrow D\}$$
$$F\_temp = \{B \rightarrow D, C \rightarrow A\}$$

Since F = F_temp & Since R = Temp we can say that all attributes & functional dependencies from R have been preserved and therefore the decomposition is lossless.

Question 6:

N (R, F), where R = {A, B, C, D} and F = {B→D, A→C, CD→B, CD→A}.

1) To identify all the keys for schema we need to consider all of the functional dependencies and their closure sets.

{A}+ → {A, C}
{B}+ → {B, D}
{C}+ → {C}
{D}+ → {D}
{A, B}+ → {A, B, C, D} = Candidate Key
{A, C}+ → {A, C}
{A, D}+ → {A, B, C, D} = Candidate Key
{B, C}+ → {A, B, C, D} = Candidate Key
{B, D}+ → {B, D}
{C, D}+ → {A, B, C, D} = Candidate Key
{A, B, C}+ → {A, B, C, D} = Superkey
{A, B, D}+ → {A, B, C, D} = Superkey
{A, C, D}+ → {A, B, C, D} = Superkey
{B, C, D}+ → {A, B, C, D} = Superkey
{A, B, C, D}+ → {A, B, C, D} = Superkey

The closure sets that contain all of the attributes from relation R (A, B, C, D) are considered to be keys. Therefore there are 9 keys in total, 4 of which are minimum candidate keys, 5 of which are superkeys.

2) Determining the highest normal form:

All attributes appear to be atomic, assuming each one only contains one value of appropriate type per row. Therefore the relation is in 1NF.

It is in 1NF and there are no partial dependencies on a composite minimum key. For example, taking {C, D} as minimum key, CD directly determines B & A, while transitively determining D & C.Therefore relation is in 2NF.

3NF states that each non-prime attribute must be entirely dependent on the key, essentially disallowing transitive dependencies - If we take {C,D} as our candidate key, both of our non-prime attributes are determined by the entire key (CD → B, CD → A). Therefore this relation is in 3NF.

BCNF states that each non-trivial functional dependency must be dependent on the candidate key - This schema does not satisfy BCNF as attributes C & D are not dependent on the entire candidate key. The functional dependencies that fail this criteria: A → C & B → D.

3) *Determine the functional dependencies:*

Given:

$$R = \{A, B, C, D\} \ \& \ F = \{B \to D, A \to C, CD \to B, CD \to A\}.$$

We form new Relations with a functional dependencies that do not satisfy BCNF (B → D):

$$R0 = \{A, B, C\} \ F0 = \{A \to C, CD \to B, CD \to A\}$$
$$R1 = \{B, D\} \ F1 = \{B \to D\}$$

We form new Relations with a functional dependencies that do not satisfy BCNF (A → C):

$$R0 = \{A, B\} \ F0 = \{CD \to B, CD \to A\}$$
$$R1 = \{B, D\} \ F1 = \{B \to D\}$$
$$R2 = \{A, C\} \ F2 = \{A \to C\}$$

These relations are now in BCNF, decomposition finished:

$$R0 = \{A, B\} \ F0 = \{CD \to B, CD \to A\}$$
$$R1 = \{B, D\} \ F1 = \{B \to D\}$$
$$R2 = \{A, C\} \ F2 = \{A \to C\}$$

4) Verification of Lossless Join:

$$R = \{A, B, C, D\} \ \& \ F = \{B \to D, A \to C, CD \to B, CD \to A\}.$$

$$R0 = \{A, B\} \ F0 = \{CD \to B, CD \to A\}$$
$$R1 = \{B, D\} \ F1 = \{B \to D\}$$
$$R2 = \{A, C\} \ F2 = \{A \to C\}$$

The union of the decomposed relations:

$$Temp = R0 \cup R2 \cup R1 = \{A, B\} \cup \{A, C\} \cup \{B, D\}$$
$$Temp = \{A, B, C, D\}$$

Check if the original relation R is equal to Temp:

$$R = \{A, B, C, D\}$$
$$Temp = \{A, B, C, D\}$$

Comparing the Functional Dependencies:

$F\_temp = F0 \cup F1 \cup F2 = \{CD \rightarrow B, CD \rightarrow A\} \cup \{B \rightarrow D\} \cup \{A \rightarrow C\} = \{B \rightarrow D, A \rightarrow C, CD \rightarrow B, CD \rightarrow A\}$.

Functional dependencies F are preserved:

$$F = \{B \rightarrow D, A \rightarrow C, CD \rightarrow B, CD \rightarrow A\}.$$
$$F\_temp = \{B \rightarrow D, A \rightarrow C, CD \rightarrow B, CD \rightarrow A\}.$$

Since F = F_temp & Since R = Temp we can say that all attributes & functional dependencies from R have been preserved and therefore the decomposition is lossless.