**CS3431-A20 Wong**

**Assignment 5: Functional Dependencies and Normalization**

Due: T 10/13 at **noon!** No late submissions accepted after noon because solutions will be posted then! This assignment will be done in the same teams of two as in your projects. Submission: Upload your typed Word or PDF document (with both of your names on it) to Canvas using the Assignment 5 button.

For attribute closures and relational schemas, list the attributes in alphabetical order for grading purposes. For example, instead of AE+ = AED, please write AE+ = ADE. And instead of R1(D, B, A), write R1(A, B, D).

**Problem 1 (50 Points)**

For the relational schema given below and its corresponding functional dependencies (FDs):

R(A, B, C, D) S = { AB 🡪 D, B 🡪 C, CD 🡪 A } answer the following questions:

1. (10 Points) find all candidate keys of the relation through an exhaustive set of attribute closures. Specify when an attribute set closure is trivial.

Trivial Cases are denoted by T

A+ = T AB+ = ABCD BD+ = ABCD ACD+ = T

B+ = BC AC+ = T CD+ = ACD BCD+ = ABCD

C+ = T AD+= T ABC+ = ABCD

D+ = T BC+= T ABD+ = ABCD

Super Keys: AB,BD,ABC,ABD,BCD (anything with a “B” that gives us ABCD)

Candidate Key: AB, BD (the only attribute that has nothing pointing to it)

1. (5 Points) Given the keys you defined in step a, find the FDs in order that they appear in S that violate BCNF.

B 🡪 C and CD 🡪 A violate BCNF since its left side does not have a superkey.

1. (20 Points) Decompose the relations to satisfy BCNF always using the left-most FD violation. For example, if AB 🡪 D is in BCNF but the other two FDs are in violation, you would use B 🡪 C for the decomposition. Specify which FD is used to make the decomposition. If there is multi-step decomposition, then indicate each step along with which FD is used for the decomposition.

**BCNF Decomposition: (leftmost violation is B 🡪 C)**

Using B 🡪 C as the violation

B+ = BC

R1 = (B, C) - two attributes, so R1 is already in BCNF

R2 = (A, B, D) - now we must check to see if R2 is in BCNF

Trivial Cases are denoted by T

A+ = T AB+ = ABD

B+ = T AD+ = T

D+ = T BD+ = T

Candidate Key: AB

AB 🡪 D

There are no more rules that are a violation here because AB is a valid key.

1. (5 Points) If the FDs in S are not in 3NF, calculate a minimal basis for the FDs. If the FDs in S are already in 3NF, explain why.

The FDs are in 3NF when each attribute that is on the right side of S is a part of the candidate key. Since none of the candidate keys include C then we are not in 3NF.

Remove AB 🡪 D: AB+ = ABC. D is not in the closure so we cannot remove. AB 🡪 D is part of the minimal basis.

Remove B 🡪 C: B+ = trivial. C is not in the closure so we cannot remove. B🡪 C is part of the minimal basis.

Remove CD 🡪 A: CD+ = trivial. A is not in the closure so we cannot remove. CD 🡪 A is part of the minimal basis.

AB 🡪 D: attribute removal

A+ = trivial. Since D is not in the closure A 🡪 D does not exist. Therefore, A cannot be removed. Part of basis.

B+ = BCD. D is in the closure, B 🡪 D does exist. We **can** remove B. B🡪D is not part of basis. It **can** be removed.

CD 🡪 A: attribute removal

C+ = trivial. Since A is not in the closure C 🡪 A does not exist. Therefore, C cannot be removed. Part of basis.

D+ = trivial. Since A is not in the closure D 🡪 A does not exist. Therefore, D cannot be removed. Part of basis.

Resulting FD’s

AB 🡪 D

B 🡪 D

B 🡪 C

CD 🡪 A

1. (10 points) Decompose the relation R to satisfy 3NF

The first and the last functional dependencies are already in 3NF since their right-hand side is part of the candidate key. However, it only takes one FD that is not in 3NF to show that S is not in 3NF. We created a minimal basis above. In this situation, we have 4 FD’s, 2 of which can be combined, creating 3 relation tables for each FD.

Combine Left Sides:

R1 (A,B,D) R2(B,C, D) R3 (A,C,D)

**Problem 2 (50 Points)**

For the relational schema given below and its corresponding functional dependencies (FDs):

R(A, B, C, D, E) S = { BC 🡪 D, A 🡪 E, B 🡪 E, DE 🡪 C }answer the following questions:

1. (10 Points) find all candidate keys of the relation R through an exhaustive set of attribute closures. Specify when an attribute set closure is trivial.

Trivial cases are denoted by T

A+ = AE AB+ = ABE BD+ = BCDE ABC+ = ABCDE ADE+ = ACDE ABCD+ = ABCDE

B+ = BE AC+ = ACE BE+ = T ABD+ = ABCDE BCD+ = BCDE ABCE+ = ABCDE

C+ = T AD+ = ACDE CD+ = T ABE+ = T BCE+ = T ABDE+ = ABCDE

D+ = T AE+ = T CE+ = T ACD+ = ACDE BDE+ = BCDE ACDE+ = T

E+ = T BC+ = BCDE DE+ = CDE ACE+ = T CDE+ = T BCDE+ = T

Super Keys: ABC, ABD, ABCD, ABCE, ABDE (anything with “AB” that gives us ABCDE)

Candidate Key: AB (the only two attributes that have nothing pointing to it)

1. (5 points) List the dependencies, in the order given in S, that violate **BCNF**.

All functional dependencies violate BCF here since none of the functional dependencies have a superkey on the left side.

BC 🡪 D

A🡪 E

B 🡪 E

DE 🡪 C

1. (20 points) If R is not in **BCNF**, provide decomposition into multiple relations where each one is in BCNF. For each decomposition step, use the left-most FD violation following the FD order given in S. For example, if BC 🡪 D is in BCNF but the other three FDs are in violation, then you would use A 🡪 E for the decomposition. Make sure to specify which FD is used to make the decomposition.

**BCNF Decomposition: (leftmost violation is** BC🡪D**)**

Using BC 🡪 D as the violation

BC+ = BCDE

R1 = (B,C,D,E) neither R1 nor R2 are in BCNF

R2 = (A,B,C)

R1: (T denotes trivial)

B+ = BE BC+ =BCE CE+ = T BDE+ = BCDE

C+ = T BD+ = BCDE DE+ = CDE CDE+ = T

D+ = T BE+ = T BCD+ = BCDE

E+ = T CD+ = T BCE+ = T

Super Keys: BD, BCD, BDE

Candidate Key: BD

B 🡪 E

BC 🡪 E remove because B 🡪 E

BD 🡪 CE

DE 🡪 C

BCD 🡪 E remove because BC 🡪 E

BDE 🡪 C remove because DE 🡪 C

S1 = {B🡪E, BD 🡪 CE, DE 🡪 C), R1’s only candidate key is BD

B 🡪 E and DE 🡪 C are violations because they are not superkeys

We break R1 into **R3(B,C,D)** and R4 (B, C, E) – both are still not in BCNF

We use B 🡪 E as a violation

R3: (T denotes trivial)

B+ = T BC+ = BCD

C+ = T BD+ = T

D+ = T CD + = T

Candidate Key: BC

BC 🡪 D

S3 = {BC🡪D} BC is a valid key, so that means S3 is in BCNF composition

R4:

Using B🡪 E violation as before

B+ = BE BC+ = BCDE

C+ = T BE+ = T

E+ = T CE+ = T

We cannot generate a superkey for R4 so it violates BCNF.

R2: (T denotes trivial)

Using BC 🡪 D violation as before

A+ = AE AB+ = ABE

B+ = BE AC+ = ACE

C+ = T BC+ = BCE

We cannot generate a superkey for R2 (A,B,C) so it violates BCNF.

Final Decomposition: **R3(B,C,D)**

1. (10 Points) If the FDs in S are not in 3NF, calculate a minimal basis for the FDs. If the FDs in S are already in 3NF, explain why.

The FDs in S are not in 3NF because at least one of the functional dependencies that are included do not have a left side that is included in the candidate key. We calculate a minimal basis.

Remove BC 🡪 D: BC+ = BCE, since D is not in BC+, then BC 🡪 D is part of minimal basis. Cannot be removed.

Remove A🡪 E: A+ = trivial, since E is not in A+, then A🡪E is part of minimal basis. Cannot be removed.

Remove B 🡪 E: B+ = trivial, since E is not in B+, then B🡪 E is part of minimal basis. Cannot be removed.

Remove DE 🡪 C: DE+ = trivial, since C is not in DE+, then DE🡪 C is part of minimal basis. Cannot be removed.

Remove BC 🡪 D: Attribute removal

B+ = BE, since D is not in the closure then B🡪D does not exist. Part of minimal basis.

C+ = trivial, since D is not in the closure then C🡪D does not exist. Part of minimal basis.

Remove DE 🡪 C: Attribute removal

D+ = trivial, since C is not in the closure then D🡪 C does not exist. Part of minimal basis.

E+ = trivial, since C is not in the closure then E🡪C does not exist. Part of minimal basis.

Resulting FD’s are the same as beginning FD’s, we have to create a new relation that encompasses our candidate key.

1. (5 points) Decompose the relation R to satisfy 3NF.

The original FD’s are a minimal basis, no attributes or relations can be removed. In addition, no left sides can be combined. So, we must create a relation for each FD:

R1 (B,C,D) R2(A,E) R3 (B, E) R4(C,D,E)

Because the candidate key does not appear in the above relations: we create a new relation encompassing the candidate key: R5(A,B).

The final decomposition is:

R1 (B,C,D) R2(A,E) R3 (B, E) R4(C,D,E) R5(A,B)