Haozhe Jiang & Emily Hao

CS3431

Assignment 5

**Problem 1.a**

R (A, B, C, D, E) S = { BC 🡪 E, B 🡪 A, E 🡪 D, AD 🡪 B }

A+ =  trivial     AB+ = trivial        BD+ = ABD ABC+ =ABCDE   BCD+ =ABCDE ABCD+ = ABCDE

B+ =  A B AC+ = trivial        BE+ = ABDE ABD+ =trivial BCE+ = ABCDE  ABCE+ = ABCDE

C+ = trivial      AD+ = ABD CD+ = trivial   ABE+ =ABDE BDE+ = ABDE ABDE+ =trivial

D+ =  trivial      AE+ = ABDE  CE+ = CDE ACD+ =ABCDE  CDE+ = trivial ACDE+ =ABCDE

E+ = D E         BC+ = ABCDE DE+ = trivial   ACE+ =ABCDE ADE+ =ABDE BCDE+ =  ABCDE

Super Keys: BC, ABC, ACD, ACE, BCD, BCE, ABCD, ABCE, ACDE, BCDE

Candidate Keys: BC, ACD, ACE

**Problem 1.b**

BC 🡪 E: BC is a super key so does NOT violate

B 🡪 A: B is not a super key so VIOLATE

E 🡪 D: E is not a super key so VIOLATE

AD 🡪 B: AD is not a super key so VIOLATE

Violators: B 🡪 A, E 🡪 D, and AD 🡪 B

**Problem 1.c**

Use B 🡪 A for violation

R1 = (B A) R2 = (B C D E)

For R1: only two attributes so in BCNF

For R2: key = (B C) FD2 = (BC 🡪 E, E 🡪 D)

For FD2: E 🡪 D violates BCNF

R3 = (E D) R4 = (E B C)

For R3: only 2 attributes so in BCNF

For R4: contains the candidate key

**Resulting relations are all in BCNF:**

**R1 = (B A)**

**R3 = (E D)**

**R4 = (E B C)**

**Problem 1.d**

S = { BC 🡪 E, B 🡪 A, E 🡪 D, AD 🡪 B }

Candidate Keys: BC, ACD, ACE

BC 🡪 E: BC is a super key so does NOT violate

B 🡪 A: B is not a super key but A is a part of candidate key so NOT violate

E 🡪 D: E is not a super key and D is a part of candidate key so NOT violate

AD 🡪 B: AD is not a super key but B is a part of candidate key so NOT violate

The FDs in S are already in 3NF.

**Problem 1.e**

S = { BC 🡪 E, B 🡪 A, E 🡪 D, AD 🡪 B }

1. Only one attribute on the right. Check.

2. Can any FD be removed?

a.   BC->E

Remove B->E. C+ = trivial. E is not in the enclosure so cannot remove attribute.

Remove C->E. B+ = AB. E is not in the enclosure so cannot remove attribute.

a.   AD->B

Remove A->B. D+ = trivial. B is not in the enclosure so cannot remove attribute.

Remove D->B. A+ = trivial. B is not in the enclosure so cannot remove attribute.

No left-side attributes can be removed.

3. BC 🡪 E. Can either attribute be removed?

Replace BC -> E with B->E. C+ = trivial. E is not in the closure. Cannot remove E.

Replace BC -> E with C->E. B+ = AB. E is not in the closure. Cannot remove B.

4. Combine any FDs that have the same left side. Nothing to be done.

5. Decompose the relation using the FDs: R1(B, C, E) R2(A, B) R3(D, E) R4(A, B, D)

**Problem 2.a**

S = {ABD 🡪 C, C 🡪 A, B 🡪 D, E 🡪 C }

A+ = trivial   AB+ = ABCD BD+ = trivial ABC+ = ABCD BCD+ = ABCD    ABCD+ = trivial

B+ = BD   AC+ = trivial   BE+ = ABCDE ABD+ = ABCD   BCE+ = ABCDE ABCE+ = ABCDE

C+ = AC AD+ = trivial    CD+ = ACD ABE+ = ABCDE    BDE+ =ABCDE ABDE+ = ABCDE

D+ = trivial AE+ = ACE CE+ = ACE ACD+ = trivial   CDE+ = ACDE ACDE+ = trivial

E+ = ACE   BC+ = ABCD   DE+ = ACDE ACE+ = trivial   ADE+ = ACDE BCDE+ =ABCDE

super keys: BE, ABC, ACE, BCD, BCE, ABCD, ABCE, ACDE, BCDE

Candidate Keys: BE

**Problem 2.b**

ABD 🡪 C: ABD is not a super key so VIOLATES

C 🡪 A: C is not a super key so VIOLATES

B 🡪 D: B is not a super key so VIOLATES

E 🡪 C: E is not a super key so VIOLATES

Violators: all

**Problem 2.c**

1. Use ABD-> C violation to decompose

ABD+ = ABCD Therefore we decompose R into

R1(A, B, C, D) and R2(A, B, E) since ABD -> C, B->D, so AB->C

Use Algorithm 3.12 to discover the projection of FDs S = {ABD -> C, C -> A, B -> D, E -> C} on R1:

A+ = trivial AB+ = ABCD BD+ = trivial ABC+ = ABCD

B+ = BD AC+ = trivial CD+ = ACD ABD+ = ABCD

C+ = AC AD+ = trivial ACD+ = trivial

D+ = trivial BC+ = ABCD BCD+ = ABCD

1. Super keys are AB, BC, ABC, ABD, BCD Candidate keys are AB and BC
2. Determine the FDs for R1
   1. B+ = BD therefore B->D
   2. C+ = AC therefore C->A
   3. AB-> ABCD therefore AB-> C, AB-> D
   4. BC -> ABCD therefore BC-> A, BC-> D

Therefore, S1 {B->D, C->A, AB-> C}

Decompose again because given that the candidate key is AB and BC, C->A is a violation Split R1 into R3 and R4 using violation C->A:   C+ =AC therefore, R3(C, A) R4(B, C, D)

* 1. Using S1 = {AB->C, B->D, C-> A} on R4

B+ = BD BC+ = BCD

C+ = trivial BD+ = trivial

D+ = trivial CD+ = trivial

* 1. S3 = {B->D}
  2. R3 a two-attribute relation which is automatically in BCNF.

1. Decompose again because given that the candidate key is AB and BC, B->D is a violation Split R4 into R5 and R6 using violation B->D:   B+ = BD therefore, R5 (B, D) and R6 (B, C)
   1. Both are two attribute relations which are automatically in BCNF.

4.R2(A, B, E) Using S = {ABD -> C, C -> A, B -> D, E -> C}

A+ = trivial AB+ = trivial

B+ = trivial AE+ = trivial

E+ = AE BE+ = ABE

S2 = {E->A}

Decompose again because given that the candidate key is AB and BC, E->A is a violation Split R4 into R5 and R6 using violation E->A: R7 (E, A) and R8 (B, E)

**As a result, the five relations of the decomposition are R7(E, A) R8(E, B) R3(C,A) R5(B,D) R6(B,C)**

**Problem 2.d**

ABD 🡪 C: ABD is not a super key and C is not in the candidate keys so VIOLATES

C 🡪 A: C is not a super key and A is not in the candidate keys so VIOLATES

B 🡪 D: B is not a super key and D is not in the candidate keys so VIOLATES

E 🡪 C: E is not a super key and C is not in the candidate keys so VIOLATES

Violators: all

**Problem 2.e**

S = {ABD 🡪 C, C 🡪 A, B 🡪 D, E 🡪 C}

1.   Simplify the FDs by having only one attribute on the right side. Already satisfied

2.   Test to see if any FD can be removed.

Remove ABD -> C. ABD+ = ABD. C is not in the enclosure so cannot remove.

Remove C -> A. C+ = C. A is not in the enclosure so cannot remove.

Remove B -> D. B+ = B. D is not in the enclosure so cannot remove.

Remove E -> C. E+ = E. C is not in the enclosure so cannot remove.

3.   Can attributes on the left side be removed?

a.   ABD -> C

Remove A->C. BD+ = trivial. C is not in the enclosure so cannot remove attribute.

Remove B->C. AD+ = trivial. C is not in the enclosure so cannot remove attribute.

Remove D->C. AB+ = ABCD. C is in the enclosure so can remove attribute.

Therefore, changed to AB->C. No other left-side attributes can be removed.

4. Combine any FDs that have the same left side. Nothing to be done.

**Minimal Basis for R: AB->C, C->A, B->D, E->C**

**Problem 2.f**

S = {AB -> C, C -> A, B -> D, E -> C}

Decompose the relation using the FDs: R1 (A, B, C)   R2 (A, C) ,R3 (B, D), R4 (C, E) .

Remove relations that are a subset of another R2

R1 (A, B C), R3 (B, D),  R4 (C, E) .

If none of the sets of relations contains a key for R, add one additional relation whose schema is a candidate key for R: R5(B, E)

Output: the union of the decompositions

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