**CS561WS Final Exam Name:Zhenjie Zhao**

1. Consider the following table schema and functional dependencies:

- Table4(A, B, C, D, E, F, G)

- A->B

- GB->C

- GB->E

- GB->F

- C->D

- D->E

Find and remove all redundant left hand side attributes. Be sure to show all the details of how you arrived at your answer. (No credit otherwise)

1. redundant left hand side attribute

Is G redundant in GB->C?

F1 = {A->B, GB->C, GB->E, GB->F, C->D, D->E}

F2 = {A->B, B->C, GB->E, GB->F, C->D, D->E}

A+F1 = AB A+F2 = ABCDE the two sets of fd’s are not equivalent

Is B redundant in GB->C?

F1 = {A->B, GB->C, GB->E, GB->F, C->D, D->E}

F3 = {A->B, G ->C, GB->E, GB->F, C->D, D->E}

A+F1 = AB A+F3 = AB

G+F1 = G G+F3 = GCDE the two sets of fd’s are not equivalent

Is G redundant in GB->E?

F1 = {A->B, GB->C, GB->E, GB->F, C->D, D->E}

F4 = {A->B, GB->C, B->E, GB->F, C->D, D->E}

A+F1 = AB A+F4 = ABE the two sets of fd’s are not equivalent

Is B redundant in GB->E?

F1 = {A->B, GB->C, GB->E, GB->F, C->D, D->E}

F5 = {A->B, GB->C, G ->E, GB->F, C->D, D->E}

A+F1 = AB A+F5 = AB

GB+F1 = GBCEFD GB +F5 = GBCEFD

G+F1 = G G+F5 = GE the two sets of fd’s are not equivalent

Is G redundant in GB->F?

F1 = {A->B, GB->C, GB->E, GB->F, C->D, D->E}

F6 = {A->B, GB->C, GB->E, B->F, C->D, D->E}

A+F1 = AB A+F6 = ABE the two sets of fd’s are not equivalent

Is B redundant in GB->F?

F1 = {A->B, GB->C, GB->E, GB->F, C->D, D->E}

F7 = {A->B, GB->C, GB->E, G ->F, C->D, D->E}

A+F1 = AB A+F7 = AB

GB+F1 = GBCEFD GB +F7 = GBCEFD

G+F1 = G G+F7 = GF the two sets of fd’s are not equivalent

1. elimination of redundant functional dependencies

if A->B is redundant?

F1 = {A->B , GB->C, GB->E, GB->F, C->D, D->E}

F8 = { , GB->C, GB->E, GB->F, C->D, D->E}

A+F1 = AB A+F8= A the two sets of fd’s are not equivalent

if GB->C is redundant?

F1 = {A->B, GB->C , GB->E, GB->F, C->D, D->E}

F9 = { A->B, , GB->E, GB->F, C->D, D->E}

A+F1 = AB A+F9= AB

GB +F1 = GBCEFD GB +F9= GBEF the two sets of fd’s are not equivalent

if GB->E is redundant?

F1 = {A->B, GB->C, GB->E , GB->F, C->D, D->E}

F10 = { A->B,GB->C , GB->F, C->D, D->E}

A+F1 = AB A+F10= AB

GB +F1 = GBCEFD GB +F10= GBCFDE

C+F1 = CDE C +F10= CDE

D+F1 = DE D +F10= DE SO THE TWO SETS ARE EQUIVALENT.

if GB->F is redundant?

F10 = { A->B,GB->C, GB->F ,C->D, D->E}

F11 = { A->B,GB->C , C->D, D->E}

A+F1 = AB A+F9= AB

GB +F10 = GBCFDE GB +F11= GBCDE the two sets of fd’s are not equivalent

if C->D is redundant?

F10 = { A->B,GB->C, GB->F ,C->D ,D->E}

F11 = { A->B,GB->C, GB->F , D->E}

A+F1 = AB A+F9= AB

GB +F10 = GBCFDE GB +F11= GBCF the two sets of fd’s are not equivalent

if D->E is redundant?

F10 = { A->B,GB->C, GB->F ,C->D ,D->E}

F11 = { A->B,GB->C, GB->F ,C->D }

A+F1 = AB A+F9= AB

GB +F10 = GBCFDE GB +F11= GBCFD the two sets of fd’s are not equivalent

So. GB->E is redundant. The final Set is { A->B,GB->C, GB->F, C->D, D->E}

2. Consider the following table schema and functional dependencies:

- Table5(A, B, C, D, E, F, G)

- A->B

- GB->C

- GB->E

- GB->F

- C->D

- D->E

Find a key for Table5 and show that it’s actually a key. (Recall that a set of attributes is not a key if some proper subset implies all the attributes.)

{ A->B,GB->C, GB->E, GB->F, C->D, D->E}

{ A->B,GB->CEF, C->D, D->E }

 (AG)+ = AGB = AGBCEF = AGBCEFD = ABCDEFG

So, AG is a super key of the set.

Check the subset of AG:

 (A)+ = A ≠ ABCDEFG

 (G)+ = G ≠ ABCDEFG

So the key for table5 is AG

3. Consider the following table schema and sets of functional dependencies:

-Table6(A, B, C, D, E, F, G)

-F1 = {GFE->D, G->C, F->BA, CB->D}

-F2 = {G->C, F->BA, CB->D}

Determine if F1 is equivalent to F2. Be sure to show all the details of how you arrived at your answer. (No credit otherwise)

1un-bundle” fd’s with multiple right-hand side attributes

F1 = {GFE->D, G->C, F->BA, CB->D}

= { CB->D , F->A, F->B, G->C, GFE->D}

F2 = { CB->D , F->A, F->B, G->C}

2 Check FD set for equivalence

(CB)+F1= CBD (CB)+F2=CBD

(F)+F1= FAB (F) +F2=FAB

(G)+F1= GC (G) +F2=GC

(GFE) +F1= GFEABCD=ABCDEFG (GFE) +F2= GFEABCD=ABCDEFG

SO F1 is equivalent to F2.

4. Consider the following table schema and functional dependencies:

- Table7(A, B, C, D, E)

- A->B

- B->C

- C->DE

- A->D

- A->E

Find and eliminate all redundant functional dependencies using the algorithm – not eyeballing, even if you’re able to. Be sure to show all the details of how you arrived at your answer. (No credit otherwise)

1un-bundle” fd’s with multiple right-hand side attributes

- Table7(A, B, C, D, E) {A->B, B->C, C->DE, A->D,A->E}

F1 = {A->B, A->D, A->E, B->C, C->D, C->E }

2 elimination of redundant functional dependencies

if A->B is redundant?

F1 = {A->B , A->D, A->E, B->C, C->D, C->E }

F2= { A->D, A->E, B->C, C->D, C->E }

A+F1 = ABDEC A+F2= ADE the two sets of fd’s are not equivalent

if A->D is redundant?

F1 = {A->B , A->D , A->E, B->C, C->D, C->E }

F3= {A->B , , A->E, B->C, C->D, C->E }

A+F1 = ABDEC A+F3= ABECD

B+F1 = BCDE B+F3= BCDE

C+F1 = CDE C+F3= CDE SO THE TWO SETS ARE EQUIVALENT.

if A->E is redundant?

F3 = {A->B , A->E , B->C, C->D, C->E }

F4 = {A->B , B->C, C->D, C->E }

A+F3 = ABECD A+F4= ABCDE

B+F3 = BCDE B+F4= BCDE

C+F3 = CDE C+F4= CDE SO THE TWO SETS ARE EQUIVALENT.

if B->C is redundant?

F4 = {A->B , B->C , C->D, C->E }

F5 = {A->B , C->D, C->E }

A+F4 = ABCDE A+F5= AB the two sets of fd’s are not equivalent

if C->D is redundant?

F4 = {A->B , B->C,C->D, C->E }

F6 = {A->B , B->C , C->E }

A+F4 = ABCDE A+F6= ABCE the two sets of fd’s are not equivalent

if C->E is redundant?

F4 = {A->B , B->C,C->D, C->E }

F6 = {A->B , B->C,C->D }

A+F4 = ABCDE A+F6= ABCD the two sets of fd’s are not equivalent

So. A->D, A->E are redundant functional dependencies.

The final Set is {A->B , B->C, C->DE }

5. Consider the following table schema and functional dependencies:

- Table8(A, B, C, D, E, F)

- A->B

- B->C

- C->D

Normalize Table8. (Note that there are no redundant left hand side attributes and no redundant functional dependencies.)

From the fds, we get A-> BCD. And E, F are orphan attributes, so create extra tables AEF.

Normalizing table8 we get:

T1 (A, B) with functional dependencies A -> B

T2 (B, C) with functional dependencies B -> C

T3 (C, D) with functional dependencies C -> D

T4 (A, E, F) with no functional dependencies – meaning that {A, E, F} is the table’s key

There is a foreign key reference from T2.B to T1.B

There is a foreign key reference from T3.C to T2.C

There is a foreign key reference from T4.A to T1.A

11. Consider the following table schema and functional dependencies:

- Table9(A, B, C, D)

- A->BC

- B->A

- C->D

Normalize Table9. (Note that there are no redundant left hand side attributes and no redundant functional dependencies.)

(A+) = ABCD

(B+) = ABCD

Normalizing table9 we get:

T1 (A, B, C) with functional dependencies A -> BC , B -> A

T2 (C, D) with functional dependencies C -> D

There is a foreign key reference from T2.C to T1.C