**1.**

1. AB->C, C->DE, D->F

Left: A, B Left and Right: C, D

A+ = {A} B+ = {B} (AB)+ = {A, B, C, D, E, F}

Candidate key is AB

C->DE, D->F are BCNF violations.

Divide S{A, B, C, D, E, F} into S1{C, D, E, F} and S2{A, B, C}

S2 is in BCNF

S1: C->DE, D->F

Candidate key of S1 is C

D->F is BCNF violation.

Divide R1{C, D, E, F} into S11{D, F} and S12{C, D, E}

The collections of BCNF relation:

S11{D, F}

S12{C, D, E}

S2{A, B, C}

1. BC → D, C → AF, AB → CE

Left: B Left and Right: A, C

B+ = {B} (AB)+ = {A, B, C, D, E, F} (BC)+ = {A, B, C, D, E, F}

Candidate key is AB and BC

C->AF is BCNF violation.

Divide S{A, B, C, D, E, F} into S1{A, C, F} and S2{A, B, C, D, E}

S1 is in BCNF

S2: BC->D, AB->CE

Candidate key of S2 is AB

BC->D is BCNF violation

Divide S2{A, B, C, D, E} into S21{B, C, D} and S22{A, B, C, E}

The collections of BCNF relation:

S1{A, C, F}

S21{B, C, D}

S22+{A, B, C, E}

1. ABC → D, CD → E, BD → A

Left: B, C Left and Right: A, D

BC+ = {B, C} (ABC)+ = {A, B, C, D, E} (BCD)+ = {A, B, C, D, E}

Because there is no dependency about F, F is part of the candidate key.

Candidate key is ABCF and BCDF

ABC->D (if do not consider about F, it is not BCNF violation), CD->E, BD->A are BCNF violations.

Divide S{A, B, C, D, E, F} into S1{A, B, C, D} and S2{C, D, E, F}

S1: ABC->D, BD->A

Candidate key of S1 is ABC and BCD

Divide S1 into S11{A, B, D} and S12{B, C, D}

S2: CD->E

Candidate key of S2 is CDF

Divide S2 into S21(C, D, E) and S22(E, F)

The collections of BCNF relation:

S11{A, B, D}

S12{B, C, D}

S21{C, D, E}

S22(E, F)

1. AB → DE, BCD → EF, B → C

Left: A, B, Left and Right: C, D

(AB)+ = {A, B, C, D, E, F}

Candidate key is AB

BCD->EF, B->C are BCNF violations

Divide S{A, B, C, D, E, F} into S1{B, C, D, E, F} and S2{A, B, D}

S2 is in BCNF

S1: BCD->EF, B->C

Candidate key of S1 is BD

Divide S1 into S11{B, C}and S12{B, D, E, F}

The collections of BCNF relation:

S11{B, C}

S12{B, D, E, F}

S2{A, B, D}

**2.**

1. Answer = Proj[person.name](

person join[person.personid = relationpersoninproceeding.personid] relationpersoninproceeding

)

1. temp1 = proj[person.personid, relationpersoninproceeding.inproceedingid](

person join[person.personid = relationpersoninproceeding.personid] relationpersoninproceeding

)

temp 2 = proj[temp1.personid, inproceeding.proceedingid](

inproceeding join[temp1.inproceedingid = inproceeding.inproceedingid] temp1 )

temp3 = proj[temp2.personid](

proceeding join[temp2.proceedingid = proceeding.proceedingid and temp2.personid = proceeding.editorid] temp2

)

answer = proj[person.name](

person join[temp3.personid = person.personid] temp3

)

1. temp1 = proj[personid](

sel[name ~ 'clark$'](person)

)

temp2 = proj[relationpersoninproceeding.inproceedingid](

temp1 join[temp.personid = relationpersoninproceeding.personid]

) relationpersoninproceeding

answer = proj[inproceeding.title](

temp2 join[temp2.inproceedingid = inproceeding.inproceedingid] inproceeding

)

1. Re1 = relationpersoninproceeding

Re2 = relationpersoninproceeding

Inpro = inproceeding

Per = person

Pro = proceeding

temp1 = Proj[personid](Re1) Minus Rename[editorid->personid](Proj[editorid](Pro))

temp2 = Proj[Re1.personid](GroupSelect[size>1](

GroupBy[Re1.personid,Re1.inproceedingid]

(Re1 Join[Re1.inproceedingid = Re2.inproceedingid] Re2))

)

temp3 = temp1 Minus temp2

Answer = Proj[name](temp3 Join[temp3.personid=Per.personid] Per)

**3.**

1. R INTERSECT (S UNION T).

Max(s, t) <= |S union T| <= s+t

0 <= |R intersect (S union T)| <= Min(r, s+t)

The minimum numbers of tuples is 0

The maximum numbers of tuples is Min(r, s+t)

1. Sel[c](R) × S, for some condition c.

0 <= |Sel[c](R)| <= r

0 <= |Sel[c](R) × S | <= r × s

The minimum numbers of tuples is 0

The maximum numbers of tuples is r × s

1. Proj[a](R) - S, for some list of attributes a.

|Proj[a](R)| = r

Max(r-s, 0) <= |Proj[a](R) – s| <= r

The minimum numbers of tuples is Max(r-s, 0)

The maximum numbers of tuples is r

**4**

T3:R(X) T4:W(Y) T4:W(Z) T1:W(Y) T2:R(Y) T3:R(D) T2:R(X) T1:W(A)

T1: W(Y) W(A)

T2: R(Y) R(X)

T3: R(X) R(D)

T4: W(Y) W(Z)



There is no circle in the graph, so it is schedule serialisable.