# Exercise 2 – Teodor Chakarov

**12141198**

## Task 1 – Functional Dependencies

A) No (Advanced – hard, Beginner – easy, Expert - easy)

B) Yes (Alice – Advanced)

C) No (Beginner, Sunny - Climbing)

D) No (Advanced, Climbing - no)

E) Yes (Albert, Rainy - yes)

F) No (Beginner, Sunny – Climbing, no)

## Task 2 – Equivalence of Functional Dependencies

A) We have equivalence when F1+ ⊆ F2+ and vice versa.

We have both RT -> QW

We have T -> AC in F1 which is T-> A and T->C in F2

We have on both A -> Y

We have E->BA which is decomposed on E -> B and E->A. We can see that in F2 we have E -> and on E->YB decomposed is E -> Y and E -> B

Since in F2 we have decomposed E -> Y and E -> B.

We to check TE -> Y attrclosure(F2, {TE})

We have T -> A and E -> YB and decomposed E -> Y and E -> B.

**T -> A and A -> Y transitivity.**

we have that F1: TE -> Y and F2: T -> Y from the transitivity and E -> Y from the decomposition.

B)

F1 = {RT → QW , TE → Y , T → AC , A → Y ,E → BA}

TE → YBC

T -> AC augumentation TE -> EAC

E -> BA augumentation TE -> TBA

TE -> E, TE -> A, TE -> C, TE -> B, TE -> T decomposition

TE -> YBC union

## Task 3: Minimal Cover

A)

F1 = {A -> C, AC -> D, AC -> E, AC -> G, AG -> B, AG -> C, AG -> E, B -> C, B -> F, B -> C, B -> E, CF -> G, CF -> E, E -> B, E -> D, F -> D, F -> E, F -> F}

Left

AC -> D attributclosure (F1, A) YES

AC -> E attributclosure (F1, A) YES

AC -> G attributclosure (F1, A) YES

AG -> B attributclosure (F1, A) YES

AG -> C attributclosure (F1, A) YES

AG -> E attributclosure (F1, A) YES

CF -> G attributclosure (F1, C) No

attributclosure (F1, F) no

CF -> E attributclosure (F1, C) No

attributclosure (F1, F) YES

{A->C, A->D, A->E, A->G, A->B, B->C, B->F, B->E, CF->G, E->B, E->D, F->D, F->E}

Right

A->C attributclosure (F\{A->C}, A) YES

A->D attributclosure (F\{A->D}, A) YES

A->E attributclosure (F\{A->E}, A) YES

A->G attributclosure (F\{A->G}, A) YES

A->B attributclosure (F\{A->B}, A) NO

B->C attributclosure (F\{B->C}, B) NO

B->F attributclosure (F\{B->F}, B) NO

B->E attributclosure (F\{B->E}, B) YES

CF->G attributclosure (F\{CF->G}, CF) NO

E->B attributclosure (F\{E->B}, E) NO

E->D attributclosure (F\{E->D}, E) YES

F->D attributclosure (F\{F->D}, F) NO

F->E attributclosure (F\{F->E}, F) NO

Fc = {A->B, B->CF, CF->G, E->B, F->DE}

B)

F={A->C, ACF->D, AFG->D, AFG->F, B->F, BC->G, BC->E, BC->F, DF->G, F->B}

Left

ACF -> D attributclosure (F, A) NO

Attributclosure (F, C) NO

Attributclosure (F, D) NO

Attributclosure (F, AF) YES

AFG -> D attributclosure (F, A) NO

Attributclosure (F, F) NO

Attributclosure (F, G) NO

Attributclosure (F, AF) YES

AFG -> F attributclosure ( F, A) NO

Attributclosure (F, F) YES

And we remove AFG->F (F -> F)

BC -> G attributclosure (F, B) NO

Attributclosure (F, C) NO

BC -> E attributclosure (F, B) NO

Attributclosure (F, C) NO

BC –> F attributclosure (F, B) YES

DF -> G attributclosure (F, D) NO

Attributclosure (F, G) NO

{A->C, AF->D, B->F, BC->G, BC->E, DF->G, F->B}

Right

A->C attributclosure (F\{A->C}, A) NO

AF->D attributclosure (F\{AF->D}, AF) NO

B->F attributclosure (F\{B->F}, B) BO

BC->G attributclosure (F\{BC->G}, BC) NO

BC->E attributclosure (F\{BC->E}, BC) NO

DF->G attributclosure (F\{DF->G}, DF) NO

F->B attributclosure (F\{F->B}, F) NO

Fc= {A->C, AF->D, B->F, BC->GE, DF->G, F->B}

## Task 4: Identifying Keys and Superkeys

A)

The keys are {B} and the set of superkeys are {B; AB; ABC; ABCD; ABCDE; ABCDEF; ABCDEFG; BC; BCD; BCDE; BCDEF; BCDEFG; BD; BDE; BDEF; BDEFG; BE; BEFG; BF; BFG; BG}

B)

The keys are {AF} and the set of superkeys are {AF, AFB, AFBC, AFBCH, AFBCD, AFBCDE, AFBCDEG, AFC, AFCD, AFCDE, AFCDEG, AFD, AFDC, AFDCH, AFDCHG, AFE, AFEG, AFG }

## Task 5: Normal Forms

Checks:

1 – is the attribute on the right side in in the left side

2 – is the attribute on the left side a super key of R

3 – is the attribute on the right side the FD contained in one of the keys

A)

Keys {AB, BC}

Decomposition:

{AB -> A (1, 2, 3), AB -> C (2, 3), AB -> D (2), C -> A (3), G -> G (1), G -> F (), G -> H (), D -> G (), D -> E ()}

**Neither 3NF nor BCNF**

B)

Keys {ZX, XY, XU}

Decomposed:

{XY -> X (1,2,3), XY -> T (2), XY -> U (2), V -> W, U -> V(2), U -> Z(2), Z -> Y (3)}

**Neither 3NF nor BCNF**

## Task 6: Synthesis Algorithm

**1 - Compute the canonical cover**

* Decomposition: {AF -> D, D -> F, DX -> H, DX -> A, X -> D, D -> G, X -> H, XHF -> A, AD -> S}
* Left reduction:

AF -> D attrcover (F, AF) NO

DX -> H attrcover (F, D) NO

attrcover (F, X) YES - DUPLICATE

DX -> A attrcover (F, D) NO

attrcover (F, X) YES

XHF -> A attrcover (F, X) YES - DUPLICATE

AD –> S attrcover (F,AD) NO

Fc = {AF -> D, D -> FG, X -> ADH, AD -> S}

* Right Reduction – no reduction possible
* Union Fc = {AF -> D, D -> FG, X -> ADH, AD -> S}

**2 – Keys (X)**

**3 – Schemata each element of Fc**

R1 = AFD, R2 = DFG, R3 = XADH, R4 = ADS

Apply FD:

F1 = {AF -> D}, F2 = {D -> FG}, F3 = {X -> ADH}, F4 = {AD -> S}

**4 - Eliminate schemas**

No schemas to eliminate

**5 – Test:** We have the key already so no need for new schema

## Task 7: Decomposition Algorithm

AB -> BG satisfies BCNF,

others not - FG -> ACE

R1 = FGACE F1=F+[R1]={FG -> ACE}, key = FG

R2 = BDFG F2=F+[R2]={FGB->BG, D -> F, B -> DF}, key = BG

R2 not in BCNF

R2,1 = DF F2,1=F+[R2,1]={D->F}, key = D

R2,2 = BG F2,2=F+[R2,2]={}, key = BG