Emirhan UZUN
171044019

CSE 414 DATABASE HOMEWORK #2

1- I will explain all steps below. But for brief, I can say that, I did this question in 5 step which we learnt in lecture. Now first step includes decomposition to see all dependencies.

Step 1: Decomposition right hand side

$A_1, A_3 \rightarrow A_7$

$A_4 \rightarrow A_5$

$A_4 \rightarrow A_7$

$A_2, A_3 \rightarrow A_4$

$A_3, A_7 \rightarrow A_2$

$A_3, A_7 \rightarrow A_4$

$A_1, A_3, A_4 \rightarrow A_2$

$A_3, A_5 \rightarrow A_1$

$A_3, A_5 \rightarrow A_7$

Step 2: Now I will check all functional dependencies and if I find extraneous attribute on left hand side, I will remove. But before check, I can directly say that $A_4 \rightarrow A_5$ and $A_4 \rightarrow A_7$ are not redundant because they have already single attribute in left hand side. I will start with more attributes on left hand side.

$A_1, A_3, A_4 \rightarrow A_2$

Now, my goal is to simplify at last and union that. Now we have $A_1, A_3 \rightarrow A_7$ in first FD. So if we get A_2 without A_4, that means A_4 is redundant. So check this closure:

$A_1, A_3 \rightarrow A_2$

$\{A_1, A_3\}^+ = \{A_1, A_3, A_7, A_2, A_4\}$ Yes, we get A_2 without A_4. So we remove A_4 from FD and our new FD is $A_1, A_3 \rightarrow A_2$

Emirhan UZUN
171044019

Now, we have:

① $A_1, A_3 \rightarrow A_7$

$A_4 \rightarrow A_5$

$A_4 \rightarrow A_7$

② $A_2, A_3 \rightarrow A_4$

③ $A_3, A_7 \rightarrow A_2$

④ $A_3, A_7 \rightarrow A_4$

$A_1, A_3 \rightarrow A_2$ (Changed)

⑤ $A_3, A_5 \rightarrow A_1$

⑥ $A_3, A_5 \rightarrow A_7$

① Now, check $A_1, A_3 \rightarrow A_7$

A_1 without A_3: $\{A_1\}^+ = \{A_1\}$ No, we can't get A_7

A_3 without A_1: $\{A_3\}^+ = \{A_3\}$ No, we can't get A_7. So these attributes aren't redundant

② Now, check $A_2, A_3 \rightarrow A_4$

A_2 without A_3: $\{A_2\}^+ = \{A_2\}$ No, we can't get A_4

A_3 without A_2: $\{A_3\}^+ = \{A_3\}$ No, we can't get A_4. So these attributes aren't redundant

③ Now, check $A_3, A_7 \rightarrow A_2$

A_3 without A_7: $\{A_3\}^+ = \{A_3\}$ We can't get A_2.

A_7 without A_3: $\{A_7\}^+ = \{A_7\}$ So these attributes aren't redundant

④ Now, check $A_3, A_7 \rightarrow A_4$

A_3 without A_7: $\{A_3\}^+ = \{A_3\}$ We can't get A_4

A_7 without A_3: $\{A_7\}^+ = \{A_7\}$ So these attributes are not redundant

⑤⑥ Now, check $A_3, A_5 \rightarrow A_1$ or $A_3, A_5 \rightarrow A_7$

A_3 without A_5: $\{A_3\}^+ = \{A_3\}$ So we can't get neither A_1 nor A_7

A_5 without A_3: $\{A_5\}^+ = \{A_5\}$ These attributes are not redundant

Emirhan UZUN
171044019

Step 3: Now, we have to find redundant functional dependencies and remove that

1. $A_1, A_3 \rightarrow A_7$ (Removed)
   $A_4 \rightarrow A_5$
   $A_4 \rightarrow A_7$
2. $A_2, A_3 \rightarrow A_4$
3. $A_3, A_7 \rightarrow A_2$ (Removed)
4. $A_3, A_7 \rightarrow A_4$
5. $A_1, A_3 \rightarrow A_2$
6. $A_3, A_5 \rightarrow A_1$
7. $A_3, A_5 \rightarrow A_7$ (Removed)

1. $A_1, A_3 \rightarrow A_7$ Find closures from other dependencies and if it contains A_7, remove FD
   $\{A_1, A_3\}^+ = \{A_1, A_3, A_2, A_4, A_5, A_7\}$
   So we have A_7. That means remove this FD

2. $A_2, A_3 \rightarrow A_4$
   $\{A_2, A_3\}^+ = \{A_2, A_3\}$ We don't get A_4, so this will stay.

3. $A_3, A_7 \rightarrow A_2$
   $\{A_3, A_7\}^+ = \{A_3, A_7, A_4, A_5, A_1, A_2\}$ So we get A_2. Remove this FD.

4. $A_3, A_7 \rightarrow A_4$
   $\{A_3, A_7\}^+ = \{A_3, A_7\}$ Since we removed 3, we didn't take A_2 or etc. So this FD are not redun

5. $A_1, A_3 \rightarrow A_2$
   $\{A_1, A_3\}^+ = \{A_1, A_3\}$ We don't get A_2. So this will stay

6. $A_3, A_5 \rightarrow A_1$
   $\{A_3, A_5\}^+ = \{A_3, A_5, A_7, A_4\}$ We don't get A_1. So this will stay
7. $A_3, A_5 \rightarrow A_7$
   $\{A_3, A_5\}^+ = \{A_3, A_5, A_1, A_7, A_4, A_2\}$ We get A_7. So remove this FD.

3

Emirhan UZUN
171044019

Step 4: With remaining functional dependencies, I will apply union if it is possible.

$A_4 \rightarrow A_5$

$A_4 \rightarrow A_7$

$A_2, A_3 \rightarrow A_4$

$A_3, A_7 \rightarrow A_4$

$A_1, A_3 \rightarrow A_2$

$A_3, A_5 \rightarrow A_1$

We have A_4 in two FDs. So we can apply union. The others have not same left hand side for union.

Step 5: Now, the last closure of my answer is:

$A_4 \rightarrow A_5, A_7$

$A_1, A_3 \rightarrow A_2$

$A_2, A_3 \rightarrow A_4$

$A_3, A_5 \rightarrow A_1$

$A_3, A_7 \rightarrow A_4$

//

Emirhan UZUN
171044019

2- $A_1, A_2 \rightarrow A_3$
$A_1, A_4 \rightarrow A_5$
$A_2 \rightarrow A_4$
$A_1, A_6 \rightarrow A_2$

a) $\{A_1, A_2\}^+ = \{A_1, A_2, A_3, A_4, A_5\}$
$\{A_1, A_6\}^+ = \{A_1, A_6, A_2, A_4, A_3, A_5\} = \{A_1, A_2, A_3, A_4, A_5, A_6\}$

b) We use closure more than one purpose. The first one is testing for superkey. To determine which is superkey, we check it contains all attributes of R.

For example if R is $\{A, B, C, D, E\}$ and we check A is superkey or not. If the closure of A contains A, B, C, D, E, that means A is superkey. In our example (2.a), $\{A_1, A_6\}$ is superkey.

The other purpose of closure is testing functional dependencies. For example if a functional dependency $\alpha \rightarrow \beta$ holds, then we have to just find α^+ and check if β is in α^+ or not. It is simple and useful.

The last purpose of closure is computing the closure of F. For every attribute or set of attributes on the left hand side of the arrow in a functional dependency, find the closure. This process results in F^+

Emirhan UZUN
171044019

3-a) Now, we check this part for the R in BCNF or not. Firstly, to be in BCNF, for every functional dependency like $X \to Y$, X should be the super key of the table. Actually when I write the closure of A_1, A_2, I didn't find the whole attributes but if it is super key that means all the functional dependencies are satisfying the condition.

super key $(A_1, A_2) \to \{$All attributes $(A_1, A_2, A_3, A_4)\}$

So yes, the relation R in Boyce-Codd Normal Form.

b) Now the R is $\{A_1, A_2, A_3, A_4\}$

$R_1 = (A_1, A_2)$

$R_2 = (A_1, A_3)$

$R_3 = (A_1, A_4)$

We will check which relations are in BCNF

For R_1, we can find all attributes from A_1. Because $A_1 \to A_2$ and $A_1^+ = \{A_1, A_2\}$. So A_1 is super key, R_1 is in BCNF

For R_2, we can find all attributes from A_1. Because $A_1 \to A_2$, $A_2 \to A_3$ and $A_1^+ = \{A_1, A_2, A_3\}$. So we have A_1, A_3 and A_1 is super key. R_2 is in BCNF.

For R_3, we can't find all attributes from A_1 or A_4. Because the functional dependencies are not enough for that. But, since A_1 is super key in R and we decomposed R into R_3 for this part, that means we can also reach A_4 from A_1. So this R_3 is in BCNF too.

Emirhan UZUN
171044019

c) $R = (A_1, A_2, A_3, A_4)$

$R_1 (A_1, A_2)$

$R_2 (A_1, A_3)$

$R_3 (A_1, A_4)$

$F \{A_1 \rightarrow A_2,\ A_2 \rightarrow A_3\}$

$R_1 (A_1, A_2)$

$A_1^+ = \{A_1, A_2\} \Rightarrow A_1 \rightarrow A_2$

$A_2^+ = \{A_2\}$

F_1

$R_2 (A_1, A_3)$

$A_1^+ = \{A_1, \cancel{A_2}, A_3\} \Rightarrow A_1 \rightarrow A_3$

$A_3^+ = \{A_3\}$

I eliminated A_2 because it is not in R_2

F_2

$R_3 (A_1, A_4)$

$A_1^+ = \{A_1, \cancel{A_2}, \cancel{A_3}\}$

$A_4^+ = \{A_4\}$

No FD

F_3

Now $F_1 \cup F_2 = \{A_1 \rightarrow A_2,\ A_1 \rightarrow A_3\}$

$F = \{A_1 \rightarrow A_2,\ A_2 \rightarrow A_3\}$

So $F_1 \cup F_2$ doesn't cover the F. Because $F_1 \cup F_2$ doesn't have $A_2 \rightarrow A_3$.

So decomposition dependency is not preserving.